

Algoritmo de Composición Musical Autónoma sin Entrenamiento, aplicando Procesamiento de Lenguaje Natural, Cómputo Evolutivo, Metaheurísticas y LWRS

**Iván Alejandro
Ramos Herrera**

Junio, 2025



UNIVERSIDAD AUTONOMA DE
AGUASCALIENTES

CENTRO DE CIENCIAS BÁSICAS

DEPARTAMENTO DE CIENCIAS DE LA
COMPUTACIÓN

Tesina

**Algoritmo de Composición Musical Autónoma sin Entrenamiento,
aplicando Procesamiento de Lenguaje Natural, Cómputo Evolutivo,
Metaheurísticas y LWRS**

Presenta

Iván Alejandro Ramos Herrera

Para Obtener el Grado de Ingeniería en Computación Inteligente

Directora:

Dra. Aurora Torres Soto

Codirectora:

Dra. María Dolores Torres Soto

Comité Tutorial:

Ing. Braulio Jesús Montoya Padilla

Dr. Francisco Álvarez Rodríguez

Dr. Alejandro Padilla Díaz

Aguascalientes, Ags., Junio 2025

Aguascalientes, Ags., 9 de junio de 2025

A quien corresponda:

Por medio de la presente me permito informar que el alumno(a) **Iván Alejandro Ramos Herrera** de la carrera de Ingeniería en Computación Inteligente con ID: **243681**, ha terminado satisfactoriamente su tesina titulada: **"Algoritmo de Composición Musical Autónoma sin Entrenamiento, aplicando Procesamiento de Lenguaje Natural, Cómputo Evolutivo, Metaheurísticas y LWRS"**, correspondiente a la materia de Seminario de Investigación II.

Para los fines que al interesado convengan.

ATENTAMENTE



Dra. Aurora Torres Soto
Directora

Aguascalientes, Ags., 9 de junio 2025

A quien corresponda:

Por medio de la presente me permito informar que el alumno(a) **Iván Alejandro Ramos Herrera** de la carrera de Ingeniería en Computación Inteligente con ID: **243681**, ha terminado satisfactoriamente su tesina titulada: **"Algoritmo de Composición Musical Autónoma sin Entrenamiento, aplicando Procesamiento de Lenguaje Natural, Cómputo Evolutivo, Metaheurísticas y LWRS"**, correspondiente a la materia de Seminario de Investigación II.

Para los fines que al interesado convengan.

ATENTAMENTE



Dra. María Dolores Torres Soto
Codirectora

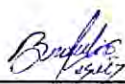
Aguascalientes, Ags., 9 de junio de 2025

A quien corresponda:

Por medio de la presente me permito informar que el alumno(a) **Iván Alejandro Ramos Herrera** de la carrera de Ingeniería en Computación Inteligente con ID: **243681**, ha terminado satisfactoriamente su tesina titulada: **"Algoritmo de Composición Musical Autónoma sin Entrenamiento, aplicando Procesamiento de Lenguaje Natural, Cómputo Evolutivo, Metaheurísticas y LWRS"**, correspondiente a la materia de Seminario de Investigación II.

Para los fines que al interesado convengan.

ATENTAMENTE



Ing. Braulio Jesús Montoya Padilla
Miembro del Comité Tutorial

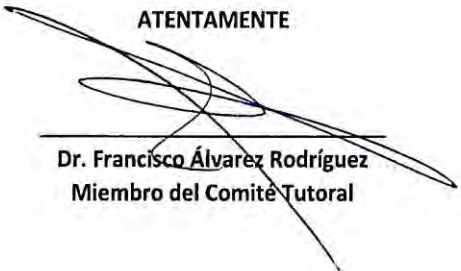
Aguascalientes, Ags., 11 de junio de 2025

A quien corresponda:

Por medio de la presente me permito informar que el alumno(a) **Iván Alejandro Ramos Herrera** de la carrera de Ingeniería en Computación Inteligente con ID: **243681**, ha terminado satisfactoriamente su tesina titulada: **"Algoritmo de Composición Musical Autónoma sin Entrenamiento, aplicando Procesamiento de Lenguaje Natural, Cómputo Evolutivo, Metaheurísticas y LWRS"**, correspondiente a la materia de Seminario de Investigación II.

Para los fines que al interesado convengan.

ATENTAMENTE


Dr. Francisco Álvarez Rodríguez
Miembro del Comité Tutorial

Aguascalientes, Ags., 11 de junio de 2025

A quien corresponda:

Por medio de la presente me permito informar que el alumno(a) **Iván Alejandro Ramos Herrera** de la carrera de Ingeniería en Computación Inteligente con ID: **243681**, ha terminado satisfactoriamente su tesina titulada: **“Algoritmo de Composición Musical Autónoma sin Entrenamiento, aplicando Procesamiento de Lenguaje Natural, Cómputo Evolutivo, Metaheurísticas y LWRS”**, correspondiente a la materia de Seminario de Investigación II.

Para los fines que al interesado convengan.

ATENTAMENTE



Dr. Alejandro Padilla Díaz
Miembro del Comité Tutorial

TESIS TESIS TESIS TESIS TESIS



TESIS TESIS TESIS TESIS TESIS

Prefacio

La creatividad de un músico jamás será replicada computacionalmente mientras no se construya una máquina que sea capaz de vivir el día a día como un humano: sufrir dolor de espalda, quemarse con metal caliente, sentir alergia al pasto, anhelar reconocimiento o amar a su prójimo. Construí MIA moviéndome inicialmente sobre un juego de probabilidad. Pensé en un algoritmo que, con la aleatoriedad de lanzar prácticamente cualquier combinación de sonidos, fuese capaz de, aun con pequeña posibilidad, crear cualquier canción humana que exista o no; eso sí, favoreciendo la probabilidad ante aquellas combinaciones que de alguna manera le resulten agradables al oído. Creo, sin embargo, haber descubierto que, para el buen compositor, el mecanismo probabilístico que le lleva a crear obras de arte no depende enteramente de la teoría, sino de algo irreplicable, que es su vida entera condensada en el placer efímero del inicio y final de cada nota acomodada. Creo entender un poco sobre cómo la consciencia de una vida, llena de sus triunfos y fracasos, de sus sueños y desencantos, llena de anhelos, desdichas y dramas, logra la verdadera combinación correcta en la creación de una obra, cuestiones inexistentes para una máquina que puede apagarse y encenderse sin más. Creo que si queremos que una máquina componga música como un ser humano, primero tendremos que darle una vida, recuerdos y sueños, y eso quizás no vaya a pasar jamás. Podríamos simularlo, pero mientras el arte no nazca como una necesidad del cuerpo de ser expulsada al mundo, y sin necesidad de un comando previo, sólo será eso: una simulación. Más allá de los aspectos técnicos, de la teoría, de la matemática asociada a la estética del sonido, y más allá de todo lo plasmado en la literatura musical, queda un plano que fluye únicamente sobre el eje del oyente, siendo el oyente quien, a mi parecer, le da verdadero sentido a la música, siendo siempre el primer oyente de una pieza, su creador. Es el oyente quien en conjunto a su historia de vida, genera o no un apego a los sonidos, que mueven o no su subconsciente en pos de darle un sentido totalmente único a lo que escucha y/o compone. Esa es, en mi opinión, la esencia de la música: el efecto involuntario producido en el interior del escucha. Efecto que cual virus informático actúa sin autorización, corrompiendo todo el mecanismo interno de la víctima, que sólo puede reaccionar al reflejo involuntario que produce.

Podemos sin embargo crear artificialmente parte del proceso. No se reemplaza al compositor, quien es el principal generador de sentido, sino que se le ofrece una herramienta, que, tal como un pájaro que silba una melodía aleatoria, puede inspirarle a construir algo nuevo, que no habría nacido de su cabeza, pero que al venir del exterior, permite mezclarse con su mecanismo probabilístico interno (llámese *consciencia*) para formar algo único, que al igual que las mejores obras, depende totalmente de las circunstancias específicas en que se dio. Con esta idea del *“pájaro que silba e inspira”*, el proyecto MIA nace un lunes 21 de noviembre de 2022 y se finaliza parcialmente el lunes 16 de junio de 2025.

Agradecimientos

El logro tras este trabajo, así como el estudio de mi carrera se lo debo a mis padres: Edaena Margarita Herrera Torres y Gerardo Ramos Martínez. Agradezco a ellos dármeles todo para vivir cómodamente y para desarrollarme profesionalmente en lo que hoy soy. Agradezco a mis abuelos, por haber formado una familia con valores claros, y con un deber hacia la educación individual y colectiva. Agradezco a mi hermano, Gerardo Daniel Ramos Herrera, por acompañarme a lo largo de la vida, por inspirarme a ser mejor, y por compartir conmigo su gusto por el arte y la filosofía.

Agradezco a todos quienes me acompañaron en el proceso, a Ruth Montserrat Serafio Méndez, por ser parte, y por estar ahí durante las tantas horas de trabajo en la biblioteca de la universidad. Agradezco a Ruth por hacerme aprender tanto acerca de la palabra escrita y hablada, así como de la filosofía. Agradezco especialmente a Alejandra Hernández Celestino, por regalarme momentos que siempre tendré presentes en mi corazón, por mostrarme lo que es el cariño y la fraternidad, y por todo lo que pasamos a través de esta travesía profesional llamada *“Ingeniería”*. Agradezco a mi mejor amigo, Jorge Antonio Hernández Magallanes, por tantos momentos de plática absurda y profunda; agradezco por mostrarme qué es la lealtad, el honor y la integridad, y por estar ahí en cada ocasión en la que hizo falta. Agradezco a Ángel Enrique Chávez Ponce, por demostrarme cómo la gratitud y la fe pueden crear comunidades excepcionales. Gracias,

TESIS TESIS TESIS TESIS TESIS

Ángel, por apoyarme y quererme siempre, sé que lograrás cualquier cosa que te propongas, porque sabes trabajar duro y aprender para crecer. Agradezco a María Fernanda Álvarez Barrón, por acompañarme y quererme durante todos estos años; nunca te permitas dar el mínimo, cuando tienes tanto por demostrar. Agradezco a mi gran amigo Abel Soto Valdez, por compartir conmigo tanto conocimiento, sabiduría y filosofía. Agradezco también con especial aprecio a todos quienes fueron miembros alguna vez del *Club de Programación Creativa*, por creer en mí, y por permitirme dar lo mejor.

Agradezco a las doctoras Aurora Torres Soto y María Dolores Torres Soto, por enseñarme cuán lejos se puede llegar cuando uno se esfuerza trabajando con empeño y honestidad. Gracias, maestras, por instruirme con su vasto conocimiento y experiencia, a través de su calidez humana. Gracias además por tener la paciencia de asesorarme a lo largo de este, el proyecto más importante (hasta la fecha) de mi vida.

Agradezco a aquellos que nunca he conocido en persona, pero quienes han influenciado decisiones en mi camino: A Jaime Altozano, quien es un divulgador que me ha enseñado a *ver* la música con cada uno de mis sentidos, a Nobuo Uematsu, por crear la música tan bella que me hizo querer ser un compositor, a Ángel David Revilla Lenoci por recomendar obras literarias, musicales y videojugabilísticas, que han marcado mi forma de ser, y finalmente a Toby Fox, quien no sólo me ha cautivado con su música, sino que, gracias a su trabajo como programador y desarrollador independiente, me hizo ver que yo también podía hacerlo, que yo también podía componer bandas sonoras memorables para historias conmovedoras; esto último aún no se logra, sin embargo el camino de programador lo tomé, y de no haber tomado ese camino, este texto no estaría plasmado aquí y ahora.

Agradezco finalmente a *ellos*, quienes no han llegado y ya forman parte del propósito de mi vida.

Dedicatoria

Dedico este trabajo al profesor *José Guadalupe Lucero Rodríguez*, quien es el motivo por la cual hoy soy músico. Le doy gracias, no sólo por enseñarme las bases de la música, sino por reconocer talento en mí al recomendarme al programa de gobierno *APTES*, por el cual viví mi infancia formándome en las artes. Gracias a la música me convertí en programador, y gracias a la programación soy ahora un ingeniero. Le agradezco, profesor, por empujarme a encontrar las dos pasiones que le dan forma a mi vida: la música y la programación. Le agradezco por la calidez con la que ha compartido a todos la belleza y la magia que trae consigo la música. Donde sea que esté, gracias de corazón.

16 de junio de 2025.

Resumen

MIA Harmfirst a1 es una arquitectura algorítmica diseñada para componer piezas musicales originales y aleatorias de manera autónoma, que no requiere de entrenamiento a partir de piezas reales, y que posee parámetros que al ser configurados permiten dirigir el estilo musical en sus resultados. El sistema genera las llamadas “*Ideas musicales*”, en archivos de audio-partitura MIDI, con dos componentes principales: melodía (de una sola voz y sin contrapunto) y armonía. El algoritmo recibe su nombre “*Harmfirst*” debido a que construye una armonía y con base en esta musicaliza su melodía (“*Harmony first*”). La melodía se construye utilizando una *Gramática Generativa* y un banco de palabras en español, que en conjunto generan oraciones completamente aleatorias y sintácticamente correctas, que posteriormente son procesadas mediante una metaheurística de *Recocido Simulado* para asignar un ritmo a cada sílaba, cuadrando las oraciones en un número definido de compases, mediante una función objetivo que maximiza la aplicación de reglas prosódicas como la colocación de sílabas tónicas en tiempos fuertes, buscando fluidez y uniformidad rítmica. La armonía, por su parte, se genera mediante un *Algoritmo Genético*, optimizando secuencias de acordes generados aleatoriamente, maximizando principios de estética musical en la armonía tonal, como la presencia de cadencias y dinámicas de tensión/resolución. Posteriormente, utilizando el algoritmo de selección *LWRS* (*Logarithmic Weighted Random Selector*) y creando una jerarquía de tonos basada en los principios de la música tonal, se asignan sonidos a cada sílaba en función de los acordes subyacentes, para finalmente conformar la *Idea musical*. El algoritmo se construyó con tres estructuras distintas de generación. Primeramente una estructura que recibe versos escritos por el usuario (en lugar de utilizar la *Gramática Generativa*) y los musicaliza (llamada *Estructura TEXTO*), otra que compone piezas musicales de la forma “*abab*’”: es decir, una frase melódica “*a*” y una frase melódica “*b*” con una progresión de acordes para ambas, y seguidamente una repetición de ambas frases, pero siendo “*b*’” una variación de “*b*” y teniendo una variación (o no) de los acordes de “*a*” y “*b*” (*Estructura abab*’), y finalmente una estructura que compone de manera libre (*Estructura libre*).

Adicional al algoritmo se diseñó una métrica experimental de evaluación estética basada en la percepción de un oyente (*Métrica de Estética Musical Perceptual*), y se aplicó en evaluaciones realizadas por cinco estudiantes del octavo semestre de la licenciatura en música, quienes analizaron siete piezas generadas por el algoritmo y siete piezas de compositores humanos galardonados. Se les indicó a los músicos que todas las piezas fueron generadas por un algoritmo, para evitar sesgos de evaluación propiciados por el origen de cada pieza. Los resultados de evaluación demostraron un puntaje global de 50% de calidad estética para las composiciones de *MIA Harmfirst a1*, en comparación al 92% de las piezas musicales humanas, con una diferencia de 15% entre la peor pieza de compositores y la mejor del algoritmo. La evaluación demostró además un 13.4% de desviación estándar en los puntajes de las piezas del algoritmo, en contraposición con el 3.4% para piezas musicales humanas, demostrando, por un lado una alta variabilidad en la calidad de las piezas del algoritmo, y por otro, la buena objetividad en la métrica de evaluación, debido a la consistencia de puntajes entre las piezas de compositores, que fueron seleccionadas especialmente por ser de excelente calidad. La fiabilidad y objetividad de la métrica de evaluación se corrobora además gracias al 5.2% de desviación estándar en los puntajes de piezas de los compositores seleccionados, pero en este caso, para una pregunta de evaluación subjetiva basada en el gusto de la pieza. Es decir, existe mayor variabilidad con los puntajes subjetivos que con la métrica objetiva.

Se logró construir un algoritmo de composición musical automática, que no depende del uso de piezas musicales reales para su entrenamiento, y que evidenció un 64.2% de aprobación entre los músicos evaluadores a los que se les pidió responder, para cada pieza, si esta poseía potencial de convertirse en una obra completa de ser tomada y ligeramente modificada a su gusto. Se encontró además que cada módulo del algoritmo tuvo un desempeño similar, aunque con diferencias puntuales. El *Recocido Simulado* (junto a la implementación de “*múltiples Recocidos*”) destacó por ser el modelo que más aportó a la calidad de los resultados generados, sobre todos para las piezas con estructura *abab*’, que tuvieron las mejores puntuaciones, con un 60% de calidad estética promedio y con un 77.5% de calidad específicamente en el *ritmo*. Se encontró que la repetición de componentes generados (como frases o grupos de acordes) dentro de una misma pieza propicia al aumento en la calidad, principalmente en *ritmo* y en *coherencia estructural*. La composición libre se encontró como la estructura peor puntuada, con 10.84% menos que el promedio de calidad en las piezas. El módulo *LWRS* tuvo un gran desempeño, pues la *melodía* fue el segundo componente mejor evaluado después del *ritmo*, para las estructuras *abab*’ y *libre*, aunque siendo menor en desempeño global con respecto al *Algoritmo Genético*, que demostró un desempeño que se puede mejorar. La categoría peor evaluada para las composiciones de *MIA* fue el *Impacto Emocional y Expresivo*. El algoritmo *MIA* logra componer piezas musicales de hasta un 71% de calidad estética, dejando cabida a que, mediante la reutilización de componentes y el acomodo manual de los mismos a criterio del usuario, se pueda mejorar incluso más la calidad de los resultados finales.

Contenido

Autorizaciones.....	I
Prefacio.....	VII
Agradecimientos.....	VII
Dedicatoria	VIII
Resumen	IX
Contenido	X
Figuras	XIII
Tablas	XIV
1 Introducción.....	1
2 Definición del problema	2
3 Justificación	2
4 Objetivos.....	3
4.1 Objetivo general	3
4.2 Objetivos específicos.....	4
5 Preguntas de investigación.....	4
6 Marco teórico.....	4
6.1 Teoría musical	4
6.1.a Conceptos básicos.....	4
6.1.b Conceptos compuestos.....	14
6.1.b.1 Melodía	14
6.1.b.2 Motivo.....	14
6.1.b.3 Armonía	15
6.1.b.4 Tonalidad	16
6.1.b.5 Reposo	18
6.1.b.6 Tensión.....	18
6.1.c Teoría tonal.....	19
6.1.c.1 Música tonal.....	19
6.1.c.2 Círculo de quintas	21
6.1.d Reglas de composición	22
6.1.d.1 Para armonía	23
6.1.d.2 Para melodía.....	24
6.1.e Idea musical	25
6.2 Teoría computacional	25
6.2.a Conceptos básicos.....	25
6.2.a.1 Algoritmos	25
6.2.a.2 Modelos	26
6.2.b Procesamiento de Lenguaje Natural	30
6.2.b.1 Lingüística.....	30

6.2.b.2	Prosodia	32
6.2.b.3	Gramáticas Generativas	32
6.2.c	Cómputo Evolutivo.....	34
6.2.c.1	Algoritmo Genético.....	35
6.2.d	Metaheurísticas	37
6.2.d.1	Recocido Simulado	38
6.2.e	Algoritmo LWRS.....	39
7	Proyectos relacionados.....	41
7.1	Modelos con entrenamiento.....	41
7.2	Modelos sin entrenamiento.....	45
7.3	Composición basada en frases	46
7.4	Resumen de proyectos	46
7.5	Oportunidades de contribución.....	49
8	Metodología.....	50
9	Desarrollo	52
9.1	Arquitectura del algoritmo.....	52
9.1.a	Entradas y parámetros de preferencias.....	52
9.1.b	Módulo de Frases Melódicas: Gramáticas Generativas	56
9.1.c	Módulo de Ritmo: Recocido Simulado.....	59
9.1.d	Módulo de Armonía: Algoritmo Genético.....	62
9.1.e	Módulo de Melodía: Algoritmo LWRS	68
9.1.e.1	Primera versión	68
9.1.e.2	Segunda versión	72
9.1.e.3	Variante funcional.....	73
9.1.f	Módulo MIDI.....	78
9.1.g	Salidas.....	78
9.2	Implementación del algoritmo.....	81
9.2.a	Estructura TEXTO	82
9.2.b	Estructura abab'	82
9.2.c	Estructura libre.....	83
9.3	Métrica de Estética Musical Perceptual.....	83
9.3.a	Coherencia Estructural.....	84
9.3.b	Integración Armónico-Melódica.....	85
9.3.c	Fluidez Rítmica.....	85
9.3.d	Impacto Emocional y Expresivo	86
9.3.e	Elementos individuales	87
9.3.f	Cómputo de puntuación	87
9.3.g	Cuestionamientos extra.....	88
9.4	Implementación de la métrica.....	88
10	Experimentos	90

10.1	Planeación	90
10.1.a	Piezas musicales	90
10.1.a.1	Piezas de MIA Harmfirst.....	90
10.1.a.2	Piezas de compositores reales	93
10.1.a.3	Conjunto de piezas de evaluación	96
10.1.b	Sujetos evaluadores.....	96
10.2	Ejecución	97
10.2.a	Sujeto 01	97
10.2.b	Sujeto 02	99
10.2.c	Sujeto 03.....	100
10.2.d	Sujeto 04	101
10.2.e	Sujeto 05.....	102
11	Resultados.....	104
11.1	Resultados objetivos globales.....	104
11.2	Resultados subjetivos globales	106
11.3	Resultados por sección evaluada	106
11.3.a	Estructura TEXTO	107
11.3.b	Estructura abab'	108
11.3.c	Estructura libre.....	109
11.4	Resultados de utilidad.....	110
11.5	Recuento de estadísticas	111
12	Conclusiones.....	111
12.1	Objetivos	111
12.2	Preguntas de investigación	112
12.3	Contribuciones.....	112
12.4	Trabajo futuro	113
12.4	Reflexiones finales	114
	Bibliografía.....	115
	Anexos.....	124
A.1	Calendario de actividades.....	124
A.2	Cartas de consentimiento de participación	125
A.2.1	Carta de consentimiento del Sujeto 01.....	125
A.2.2	Carta de consentimiento del Sujeto 02.....	126
A.2.3	Carta de consentimiento del Sujeto 03.....	127
A.2.4	Carta de consentimiento del Sujeto 04.....	128
A.2.5	Carta de consentimiento del Sujeto 05.....	129

Figuras

Figura 1. Frecuencias asociadas a cada nombre de tono, con C4 y D3 resaltados.....	5
Figura 2. Figuras musicales y sus duraciones.....	7
Figura 3. Acorde de Do con quinta justa.	9
Figura 4. Acorde de Do mayor.	10
Figura 5. Acorde de Do menor.	10
Figura 6. Acorde de Do aumentado.	10
Figura 7. Acorde de Do disminuido.....	11
Figura 8. Acorde de Do séptima.	11
Figura 9. Acorde de Do con séptima mayor.	11
Figura 10. Acorde de Do menor séptima.	12
Figura 11. Acorde de Do menor con séptima mayor.	12
Figura 12. Acorde de Do suspendido con cuarta.	12
Figura 13. Acorde de Do suspendido con segunda.	13
Figura 14. Inversiones del acorde mayor de Do.	13
Figura 15. Motif y Melodía de la Quinta Sinfonía de Beethoven.	14
Figura 16. Escala de Re# (Mib) mayor.....	16
Figura 17. Círculo de quintas.....	21
Figura 18. Círculo de quintas con los acordes de la escala de Mib mayor seleccionados.	22
Figura 19. Círculo de quintas con los acordes de la escala de Mib mayor seleccionados.	22
Figura 20. Jerarquía de tonos en Do mayor y Do menor según el estudio de Krumhansl y Kessler.	24
Figura 21. Representación de árbol de una gramática generativa.	34
Figura 22. Visualización del “cruzamiento” de dos individuos en el Algoritmo Genético.	36
Figura 23. Escala logarítmica de 5 elementos.	41
Figura 24. Metodología experimental del proyecto MIA.	50
Figura 25. Arquitectura del Algoritmo MIA Harmfirst a1.	52
Figura 26. Preferencias de selección en el algoritmo EWRS con valores x entre 1 y 10, para 10 objetos.	59
Figura 27. Posible salida de oración convertida en frase rítmica mediante Recocido Simulado.	61
Figura 28. Probabilidades de selección para 10 elementos usando CWRS, para complejidades de 0, 20, 40, 60, 80 y 100.	65
Figura 29. Escala de Do mayor hacia arriba ignorando tonos del acorde Do mayor en segunda inversión.	70
Figura 30. Tonos restantes cromáticos en una escala de Do mayor.....	70
Figura 31. "Melodic Jude", primer tema compuesto por el algoritmo MIA.	71
Figura 32. "Desaparecer": tema compuesto con la segunda variante del módulo LWRS, con versos de un poema.	73
Figura 33. "Aceptación 2a1": Pieza generada por la versión final de MIA Harmfirst a1.	77
Figura 34. "Aceptación 4a8": Octava pieza generada por la versión final de MIA Harmfirst a1.	77
Figura 35. Piano roll para la pieza "Aceptación 4a8" visto en el DAW Reaper.	78
Figura 36. Cuestionario inicial de la interfaz web para la Métrica de Estética Musical Perceptual.	88
Figura 37. Interfaz principal del instrumento web de evaluación de la Métrica de Estética Musical Perceptual.	89
Figura 38. Bloques con las preguntas extra en la herramienta web de evaluación.	90
Figura 39. Pieza de algoritmo P02: “¿Cuántas pasquinase?” 91	91
Figura 40. Pieza de algoritmo P04: "Mi dulce Tierra".	91
Figura 41. Pieza de algoritmo P06: "¿Cuáles esos morigeran?..... 91	91
Figura 42. Pieza de algoritmo P09: "Ajuntases"..... 92	92
Figura 43. Pieza de algoritmo P10: "¿Quién refrigerador enzurdecieres?".	92
Figura 44. Pieza de algoritmo P14: "Me acompañó".	92
Figura 45. Piezas musicales elegidas, del compositor Nobuo Uematsu, del videojuego Final Fantasy VI.	94
Figura 46. Piezas musicales elegidas, del compositor Toby Fox, del videojuego Undertale..... 95	95
Figura 47. Pieza musical humana extra P08: "Sandy".	95
Figura 48. Sujeto 01 realizando evaluación.	98

Figura 49. Sujeto 03 realizando evaluación.	100
Figura 50. Sujeto 04 realizando evaluación.	101
Figura 51. Sujeto 05 realizando evaluación.	103
Figura 52. Aulas de evaluación A-3 y A-1.	104
Figura 53. Resultados objetivos globales de evaluación: MIA vs compositores.	104
Figura 54. Resultados subjetivos globales de evaluación: MIA vs compositores.	106
Figura 55. Puntaje porcentual promedio en cada sección evaluada para piezas de MIA.	107
Figura 56. Puntaje promedio en cada sección evaluada para piezas de MIA TEXTO.	108
Figura 57. Puntaje promedio en cada sección evaluada para piezas de MIA abab'.	108
Figura 58. Puntaje promedio en cada sección evaluada para piezas de MIA libre.	109

Tablas

Tabla 1. Nombre de las doce notas musicales.	5
Tabla 2. Nombres y funciones de los grados asignados a los acordes en una tonalidad diatónica.	17
Tabla 3. Recopilación de proyectos de composición musical con cómputo inteligente y sus características.	47
Tabla 4. Acordes disponibles para el algoritmo MIA.	63
Tabla 5. Conjunto completo de piezas musicales a evaluar.	96
Tabla 6. Resultados de evaluación del sujeto 01.	98
Tabla 7. Resultados de evaluación del sujeto 02.	99
Tabla 8. Resultados de evaluación del sujeto 03.	100
Tabla 9. Resultados de evaluación del sujeto 04.	102
Tabla 10. Resultados de evaluación del sujeto 05.	103
Tabla 11. Puntajes promedio para cada pieza musical y sus secciones.	111

1 Introducción

El estudio de la creación artística es complejo en sí mismo, pues se trata de un fenómeno que depende casi enteramente de cada creador que la lleva a cabo. La composición musical es un proceso único para cada artista, pues precisa de la inspiración, de las motivaciones individuales, del estado de ánimo, y de muchos otros factores subjetivos únicos e imposibles de plasmar en un esquema al que se le pueda llamar “*proceso creativo universal*”. Bajo este precedente, es notable la alta complejidad de un proceso creativo como lo es la composición musical, y es fácil dilucidar cuán vulnerable es este a factores tan variados que afecten la psicología de su creador, hablando de circunstancias que puedan llevarlo a bloqueos que le impidan crear. Es ante estos bloqueos que surge como solución natural el concepto de *inspiración*, entendiéndole en este contexto como aquellas ideas que, a pesar de no provenir del creador de una obra, sino de su exterior, gatillan el inicio de un proceso creativo que le lleva a explorar conceptos que sí provienen de sí, y de los que nace una composición. Esta inspiración, para un músico puede encontrarse hasta en lo más simple, como en el pájaro que silva una melodía aleatoria, en el galope de un caballo o hasta en el ritmo de gotas de agua que caen. Depende enteramente del creador reinterpretar estas observaciones en codificaciones musicales, y aun con ello es difícil superar el bloqueo de una “*hoja en blanco*”.

La fuente principal de inspiración de un músico es la música. Resulta imposible imaginar cómo un compositor puede crear algo que intrínsecamente no posea ideas tomadas de música a la que ha sido expuesto. Es inasequible desconectar lo que crea un artista de lo que ha consumido, y en este contexto, la música es la principal inspiración para crear más música. Esto anterior puede dejar al descubierto un dilema relacionado a qué tanto de lo que crea un músico es verdaderamente original. La *inspiración* mal ejecutada puede convertirse en *plagio*, y es por ello importante controlar de dónde se toma inspiración; si de un artista comercial reconocido, de un pájaro que canta de manera natural, o de un sistema computacional generador de música.

Habiendo definido la complejidad intrínseca de un proceso de composición musical y habiendo mencionado la posibilidad latente en un músico a sentirse bloqueado y sin capacidad de componer, se idea que las herramientas computacionales de asistencia jugarían un papel importante en el proceso creativo de un músico. A través de este trabajo se explorará cómo es un hecho que la computación ofrece una alternativa a esta necesidad de obtener inspiración, con herramientas de composición musical autónomas que le dan al músico ideas de las cuales gatilla su proceso creativo. Herramientas que no son un remplazo del creador, sino que, como se menciona en “*Artificial Intelligence for Music Composition*”, capítulo del libro “*Handbook of Artificial Intelligence for Music*” (Gioti, 2021), ayudan al compositor a liberarse de su pensamiento habitual y explorar nuevos espacios conceptuales.

La creación de herramientas computacionales que asistan el proceso de composición de un músico es un tópico ya bien estudiado y desarrollado (como se verá a lo largo de este trabajo). Y se sabe, por lo analizado en investigaciones como “*Algorithmic music composition based on artificial intelligence: A survey*” de (Lopez-Rincon et al., 2018), que una importante mayoría de estos trabajos basan sus soluciones en modelos computacionales dependientes enteramente de entrenamiento con datos; es decir, modelos que aprenden a componer utilizando conjuntos de canciones y composiciones de artistas reales. Los trabajos aquí revisados exponen una visible preferencia en los investigadores por modelos de *Machine Learning* (Aprendizaje Automático), y más importante que eso, sin considerar implicaciones éticas relacionadas al uso no autorizado de música de artistas para el entrenamiento de modelos generativos. En el antes mencionado capítulo de Gioti se expone que estos modelos tienden a imitar sus ejemplos de entrenamiento, limitando su “*creatividad*”, y sin discutir si a ello se le consideraría o no plagio. Gioti menciona igualmente que la mayoría de estos trabajos buscan la extrema automatización en el proceso creativo, dando como resultado herramientas que necesitan poca interacción y configuración del usuario; sin dar espacio al músico a plasmar parte de su creatividad en el proceso de creación de una pieza generada por estas herramientas casi totalmente autónomas.

Tomando como base lo antes expuesto, es notable un espacio potencial para la aportación, en el que modelos computacionales que asistan el proceso de composición de un músico, como apoyo ante la complejidad del mismo, puedan ser funcionales en su objetivo a la par de no depender del uso de piezas musicales reales; esto para, por un lado no caer en el robo de arte y a su vez no limitar las composiciones de estas herramientas a sólo un género o artista de referencia, dando la posibilidad de ser flexible ante las configuraciones de decisiones creativas del

TESIS TESIS TESIS TESIS TESIS

artista en cuestión, para darle a este un control real sobre lo que necesita de inspiración. Es por ello que este trabajo buscará atender esta necesidad de herramientas de apoyo creativo en la composición musical, pero con un enfoque que no dependa de uso de música real para entrenamiento de modelos, de manera que este sistema sea tan eficiente por sí mismo únicamente explorando y explotando reglas de teoría y composición musical, así como técnicas computacionales que no dependan de datos de entrenamiento, como en el *Cómputo Evolutivo*, e incorporando sobre el camino técnicas interesantes del *Procesamiento de Lenguaje Natural* para mejorar la expresividad en el proceso generativo de este sistema en cuestiones como la composición de frases melódicas, además de la exploración del *Recocido Simulado* como heurística que permita obtener ritmos consistentes, y el algoritmo novedoso *LWRS* (Ramos Herrera, 2025b) para incorporar decisiones creativas aleatorias que tengan la suficiente envergadura para abarcar cualquier posibilidad musical al componer una melodía y a la vez una convergencia preferencial a aquellas decisiones que resulten estéticas según la teoría.

2 Definición del problema

Uno de los principales desafíos éticos, legales y técnicos en la generación de contenidos mediante Inteligencia Artificial refiere a garantizar la originalidad de sus resultados, en medida que estos no constituyan una forma de plagio o de reutilización no autorizada de obras existentes. Esta exigencia resulta especialmente relevante en el contexto artístico, donde la mayoría de las herramientas disponibles se crean mediante el uso de extensos conjuntos de obras reales, lo que, por un lado genera descontento entre los creadores quienes observan cómo su trabajo alimenta estos sistemas sin obtener reconocimiento ni compensación alguna, y por otro, restringe la creatividad genuina en las piezas generadas, puesto que estos modelos tienden a imitar patrones de las piezas que se usaron para generárseles. En un contexto musical, los modelos entrenados con música mayormente limitan el género, el autor de referencia o el estilo compositivo, lo que coarta la diversidad creativa y el grado de control que el usuario puede ejercer sobre el resultado, así como reduce la utilidad del mismo. Estos inconvenientes evidencian una necesidad de prescindir del uso de obras reales en la creación de estos modelos generadores, y en ello es que el marco de este trabajo encuentra un enfoque.

En la literatura relacionada, adicional a la evidente preferencia hacia los modelos entrenados con música real, se nota una carencia metodológica referente a la medición de la calidad estética en las obras generadas; casi ningún trabajo en este campo ha propuesto una métrica clara para evaluar la calidad estética y la capacidad inspiradora y útil de la música generada por Inteligencia Artificial. Quienes proponen indicadores de evaluación suelen basarse en métricas estadísticas propias de los algoritmos de aprendizaje, que cuantifican similitudes estadísticas, pero no proporcionan criterios musicales objetivos ni la percepción real de un músico. Sin protocolos de evaluación rigurosos, resulta imposible comparar de manera sistemática y transparente las composiciones automáticas con obras humanas, ni diagnosticar con precisión las áreas potenciales de mejora en las herramientas automáticas; tampoco medir la utilidad de la herramienta creada con valoraciones de su público objetivo, reflejando así otra necesidad técnica a abordar en el marco de este trabajo.

3 Justificación

El estudio de las Ciencias de la Computación concibe algo fascinante y único: la veloz y constante innovación de su estado del arte. Desde cualquier parte del mundo y a un paso complejo de seguir, se hacen mes a mes avances revolucionarios. La Inteligencia Artificial es un área de estudio con frecuentes resultados disruptivos, con avances que toman notoriedad incluso en el conocimiento popular, debido principalmente a populares herramientas únicas abiertas al público, como los denominados *Modelos de lenguaje de gran tamaño* (véase los modelos *GPT* creados por *Openai* y cuyo propósito es la generación de texto artificial realista), o como los sistemas generativos *DALL-E* (dedicados a la creación de imágenes), que sumando su utilidad, su efectividad y su publicidad, convierten el propio concepto de “*Inteligencia Artificial Generativa*” en un común para el usuario promedio del internet occidental.

La creación de herramientas de Inteligencia Artificial (IA) pueden tener motivaciones nobles, pues a menudo se habla de ofrecer herramientas que faciliten y optimicen las labores de cualquier usuario. Sin embargo, en paralelo a la creación de estos sistemas es que surgen debates sociales y dilemas éticos. El más relevante dilema es el que parte del hecho de que, para la creación de estos sistemas tan complejos hace falta entrenar modelos computacionales con grandes cantidades de datos de ejemplo en aquellas tareas que pretende emular. Por ejemplo, un modelo generativo de imágenes se entrenará con conjuntos de millones de ilustraciones, para las cuales es natural suponer que no se solicita autorización de uso a sus creadores. Además, dado que los modelos de aprendizaje empleados en estas herramientas tienden a imitar aquello con lo que fueron alimentados, resulta lógico cuestionar si a aquel proceso de generación se le puede llamar *plagio*.

En el artículo de 2024 “*AI Art is Theft: Labour, Extraction, and Exploitation: Or, On the Dangers of Stochastic Pollocks*”, por su traducción al español: “*El arte de la IA es un robo: trabajo, extracción y explotación, o sobre los peligros de los Pollocks estocásticos*”, Goetze (2024) analiza las implicaciones morales asociadas a la creación de modelos como DALL-E, Midjourney y Stable Diffusion, fundamentado y concluyendo que el uso de piezas de arte para el entrenamiento de estos modelos “*implica un tipo de robo de mano de obra poco ético*”.

En contraposición, Shoemaker (2024) argumenta y concluye en el trabajo “*Is AI Art Theft? The Moral Foundations of Copyright Law in the Context of AI Image Generation*”, (“*¿La IA es un robo de arte? Los fundamentos morales de la legislación sobre derechos de autor en el contexto de la generación de imágenes con IA*”) que no hay fundamento legal ni ético para etiquetar como “*robo*” el uso de arte en el entrenamiento de modelos generativos de Inteligencia Artificial, si bien justifica y describe el descontento generalizado de la comunidad de artistas visuales ante la existencia de estas aplicaciones.

Si bien el debate es variado de opiniones, tanto en el análisis legal como en el moral, hay un par de cuestiones que son constantes entre quienes ofrecen análisis sobre este tema (destacando que, pesar de estar principalmente enfocado en los modelos de generación de imágenes, es igualmente aplicable para los modelos de generación de música, texto o casi cualquier tipo de contenido artístico): en primer lugar, que la creación de estas herramientas genera un descontento en la comunidad de artistas (independientemente de si se argumenta como justificable o no), y en segundo lugar, que estos sistemas son hechos con el fin de servir como herramientas de apoyo para artistas. Resulta irónico que algo creado para servir como apoyo genere descontento en aquellos para quienes está dirigido, y esto como un costo asociado a crear modelos cuyas funcionalidades son posibles únicamente gracias al entrenamiento con conjuntos enormes de creaciones de artistas reales, para los que se sabe que no se les solicitó autorización de uso, sabiendo además que no recibirán ningún tipo de compensación ni crédito por su trabajo.

Es con este panorama bajo el cual el presente trabajo plantea abordar ambos fuertes: en primer lugar, aportar a la comunidad artística musical, creando una herramienta que le sea de ayuda al proceso creativo un músico de cualquier nivel (o incluso a no-músicos con el interés de crear música), y en segundo lugar, explorando alternativas que no generen debates éticos ni descontentos respectivos al uso no autorizado de piezas artísticas. MIA se define y se propone entonces como un sistema de ayuda a compositores en su proceso creativo, siendo un sistema que no se nutre de piezas de artistas reales y que no imita el estilo de ningún género ni compositor en específico.

4 Objetivos

Con base en el problema planteado, este trabajo se guía por los siguientes objetivos.

4.1 Objetivo general

Diseñar y desarrollar un algoritmo autónomo de composición musical que, sin entrenamiento previo con música real, genere piezas equiparables a ejemplos de la música tonal contemporánea y que incorpore una métrica validada mediante evaluación de músicos expertos para medir su calidad estética.

4.2 Objetivos específicos

1. Explorar e incorporar reglas de composición musical, técnicas de Procesamiento de Lenguaje Natural, Cómputo Evolutivo y Metaheurísticas, a la composición autónoma de piezas musicales originales.
 2. Crear una métrica de puntos basada en la percepción de escucha, para evaluar las cualidades estéticas en los componentes estructurales de composiciones musicales.
 3. Recopilar una selección de obras de la música tonal contemporánea y una selección de piezas generadas por el algoritmo diseñado, para evaluar su calidad estética con un grupo de músicos estudiados en el análisis musical.
 4. Analizar los resultados de evaluación de las piezas para entender fortalezas y deficiencias en la arquitectura del algoritmo diseñado.
-

5 Preguntas de investigación

Descritos los objetivos, este reporte pretende responder las siguientes preguntas:

- ¿Qué arquitectura de algoritmo computacional permitiría la composición musical autónoma, sin que dependa de entrenamiento con música real?
 - ¿De qué manera es posible adaptar técnicas de Procesamiento de Lenguaje Natural y Metaheurísticas, a la composición autónoma de melodías musicales?
 - ¿De qué manera el Cómputo Evolutivo puede ser parte del proceso de composición de la armonía de una pieza musical?
 - ¿Qué elementos cualitativos podrían evaluar la calidad estética de una composición musical?
-

6 Marco teórico

El trabajo aquí presentado se construye a partir de una intersección entre dos áreas del conocimiento; la música y la computación inteligente. Es entonces que lo construido funge como vinculator entre las artes y la ciencia, de modo que, para la comprensión de esta misma, es preciso el entendimiento de conceptos básicos y complejos de ambas áreas.

6.1 Teoría musical

6.1.a Conceptos básicos

La música como un concepto vasto se forma a partir de elementos muy básicos. La unidad más básica es la *Nota*, y es definida en el libro *“The concise Oxford dictionary of music”* (Scholes & Ward, 1977) como *“un símbolo que denota un sonido musical; también el sonido en sí”*. Es descrita también como *“los signos que representan las duraciones y los sonidos”* (Danhauser, 2008). Se entiende entonces que *“nota musical”* es esta unidad básica formadora de la música, y está compuesta de dos elementos: duración y tono.

El *Tono* es descrito; según el libro *Física y música*, del autor De la Herrán (2007), como el sonido que nace de la vibración de un cuerpo físico que propaga dichas vibraciones a través de algún medio como el aire o el agua; de manera que, según la frecuencia de vibración del objeto, es decir, su velocidad de vibración medida, se le da un nombre asociado a un *tono musical*. Es entonces que se entiende *tono musical* como la velocidad de vibración de un sonido (medida popularmente en Hertz como unidad de frecuencia), y a su vez como el nombre que se le da en notación musical. Por ejemplo, una frecuencia de 261 Hertz es llamada por convención *Do-4* (escrito *C4* en notación inglesa), y una de 146 Hertz es llamada *Re-3* (*D3*), bajo un estándar de afinación de *A4=440 Hertz*, en el que la frecuencia de 440 se nombra *La-4* (*A4*) y el resto de las frecuencias se nombran a partir de esta. En palabras simples, el *Tono* indica cuál es el sonido de una nota, primeramente midiendo la frecuencia de vibración de la nota, y seguidamente asignando como convención un nombre a distintos valores de frecuencia, en donde estos nombres son parte de un estándar definido llamado *Temperamento Igual*.

El "Temperamento igual" fue impuesto hasta la época de Bach, como convención para dividir el rango de frecuencias audibles por el ser humano, en valores manejables y nombrables (De la Motte, 2007). El proceso de identificación de sonidos con base en su frecuencia data de la antigua Grecia, en donde los pitagóricos notaron que, al hacer vibrar una cuerda tensada, se producía un sonido; que al sonar simultáneamente con el de otra cuerda tensada con la mitad del tamaño de la primera, crearía una armonía agradable al oído. Este proceso descubrió que; una cuerda vibrando a cierta frecuencia, por ejemplo, 440 Hz, y otra vibrando a una frecuencia con una relación matemática de 2 respecto a la primera (mitad, doble, cuádruple, etc.), por ejemplo, 880 Hz, produce un sonido que para el oído se denominaría "similar", lo que llevó al *Temperamento igual* a nombrar a dichas dos notas con el mismo nombre, pero diferenciándolas con un número de "octava".

La **Figura 1** muestra los nombres determinados por el Temperamento igual, en una tabla en la que, tomando como base la afinación estándar, es decir, que $A4 = 440\text{Hz}$, se muestran los nombres asignados a las frecuencias en que fue dividido el espectro audible. Notando que la nota nombrada *D#2* (como ejemplo) tiene una frecuencia de 77.8 Hz, y que *G5* equivale a 784 Hz. Se nota además que el tono sobre la columna 4 y la fila C, es decir, el tono *C4* (*Do-4*) corresponde a la frecuencia de 261.6 Hz, de igual manera que la nota *D3* (*Re-3*) tiene la frecuencia de 146.8 Hz. También se nota que se le da un mismo nombre (*A* en el caso de la columna naranja resaltada a frecuencias distintas, pero que tienen relación matemática de 2).

FREQUENCY CHART // $A_4 = 440\text{ Hz}$

	0	1	2	3	4	5	6	7	8	Octave
C	16.4	32.7	65.4	130.8	261.6	523.3	1046.5	2093.0	4186.0	
C#	17.3	34.7	69.3	138.6	277.2	554.4	1108.7	2217.5	4434.9	
D	18.4	36.7	73.4	146.8	293.7	587.3	1174.7	2349.3	4698.6	
D#	19.5	38.9	77.8	155.6	311.1	622.3	1244.5	2489.0	4978.0	
E	20.6	41.2	82.4	164.8	329.6	659.3	1318.5	2637.0	5274.0	
F	21.8	43.7	87.3	174.6	349.2	698.5	1396.9	2793.8	5587.7	
F#	23.1	46.3	92.5	185.0	370.0	740.0	1480.0	2960.0	5919.9	
G	24.5	49.0	98.0	196.0	392.0	784.0	1568.0	3136.0	6271.9	
G#	26.0	51.9	103.8	207.7	415.3	830.6	1661.2	3322.4	6644.9	
A	27.5	55.0	110.0	220.0	440.0	880.0	1760.0	3520.0	7040.0	
A#	29.1	58.3	116.5	233.1	466.2	932.3	1864.7	3729.3	7458.6	
B	30.9	61.7	123.5	246.9	493.9	987.8	1975.5	3951.1	7902.1	Hz

Figura 1. Frecuencias asociadas a cada nombre de tono, con *C4* y *D3* resaltados.

Nota: Figura con autoría de Matzmatzmatzmatz (2020).

El Temperamento Igual define el nombre de las frecuencias usando dos elementos: nombre de la nota y octava. Para esto define 12 nombres únicos de notas, que se repiten de manera cíclica según su número de octava. La **Tabla 1** muestra los 12 nombres únicos para las notas, así como su nombre en notación inglesa.

Tabla 1. Nombre de las doce notas musicales.

Nombre	Notación inglesa
Do	C
Do sostenido / Re bemol	C# / Db
Re	D
Re sostenido / Mi bemol	D# / Eb
Mi	E
Fa	F

Fa sostenido / Sol bemol	F# / Gb
Sol	G
Sol sostenido / La bemol	G# / Ab
La	A
La sostenido / Si bemol	A# / Bb
Si	B

Cuando se habla de “Octava”, se hace referencia al número que representa la agrupación de doce notas a la que representa dicha nota. Por ejemplo, la nota *C4* es aquella de 261.6 Hz, de manera que es la nota *Do* (C) de la octava número 4. La nota *C3* entonces es la nota *Do* de la tercer octava. De manera que, si se tiene una nota con el doble de frecuencia de otra, se dice que dicha nota está una octava por encima, del mismo modo que si su frecuencia es la mitad, se dice que está una octava por debajo.

Sabiendo entonces que el sonido de una nota se distingue mediante: el nombre de la nota y la octava (y que a su vez esto indica la frecuencia de dicha nota), conceptos adyacentes a esto surgen cuando se quieren medir distancias entre notas. Por ejemplo, el “*Semitono*”, que es el nombre que recibe la distancia que hay entre dos notas consecutivas. Por ejemplo, entre un *D#5* (*Re sostenido-5*) y un *E5* (*Mi-5*) existe una distancia de semitono. Al igual que entre el *E5* (*Mi-5*) y *F5* (*Fa-5*). El semitono es entonces la distancia que hay entre cada una de todas las notas en que se divide el espectro sonoro según el Temperamento Igual. Existe además el concepto “*Tono*”, que no se refiere a lo mismo que el concepto anterior de “*Tono*”, sino que en el contexto de distancias entre notas, un *Tono* es la distancia de dos semitonos entre dos notas nombradas. Por ejemplo, entre las notas *C, D, M, F, G, A, B* existe una relación de distancia de *tono*, y se le llama “*Intervalo*” a la distancia medida en tonos o semitonos entre dos notas¹.

Así como el sonido, la “*Duración*” es otro componente que define a una nota musical, y se entiende como el tiempo que suena la misma al ser interpretada por un instrumento. Para formalizar la duración de notas, se ha creado la llamada “*notación clásica de solfeo*”, que sirve como convención. En esta notación se utilizan las llamadas “*figuras*”, que son símbolos con nombres distintivos y con una respectiva duración basada en la “*unidad de valor*” (Danhauser, 2008). La notación clásica de solfeo nombra las siguientes figuras, ordenadas de manera que cada siguiente figura dura la mitad de la que le precede, y, por lo tanto, el doble de la que sigue.

1. **Redonda (1 entero):** Figura de más larga duración. Se le considera *la unidad de valor*, siendo las siguientes figuras fracciones de esta unidad. Vale 2 blancas o 4 negras u 8 corcheas o 16 semicorcheas o 32 fusas o 64 semifusas. Llamada en inglés *whole note* (nota completa).
2. **Blanca ($\frac{1}{2}$):** Vale 2 negras o 4 corcheas u 8 semicorcheas o 16 fusas o 32 semifusas. Llamada en inglés *half note* (media nota).
3. **Negra ($\frac{1}{4}$):** Vale 2 corcheas o 4 semicorcheas u 8 fusas o 16 semifusas. Llamada en inglés *quarter note* (cuarto de nota).
4. **Corchea ($\frac{1}{8}$):** Vale 2 semicorcheas o 4 fusas u 8 semifusas. Llamada en inglés *eighth note* (octavo de nota).
5. **Semicorchea ($\frac{1}{16}$):** Vale 2 fusas o 4 semifusas. Llamada en inglés *sixteenth note* (dieciseisavo de completa).
6. **Fusa ($\frac{1}{32}$):** Vale 2 semifusas. Llamada en inglés *thirty-second note* (treintaidosavo de nota).
7. **Semifusa ($\frac{1}{64}$):** Vale un 64ava parte de una redonda y la mitad de una fusa. Llamada en inglés *sixty-fourth note* (sesentaicuatroavo de nota).

La **Figura 2** muestra una tabla (separada en dos) con los símbolos asociados a cada figura de duración, en donde la primer columna corresponde al nombre de la figura y la segunda corresponde al símbolo de la figura.

¹ Cuando se hable de “*Tono (sonido)*” se hace referencia al nombre que recibe una nota según su frecuencia, y cuando se mencione “*Tono (distancia)*” se refiere a la distancia de dos semitonos.

Además, esta ilustración posee una tercer columna, con el símbolo asociado al *Silencio*. Si una nota es definida como “*silencio*”, implica que no tiene un tono asociado, sino que al reproducirse con un instrumento, el intérprete no debe producir sonido. A pesar de no tener sonido un silencio, sí tiene una duración, por lo que el símbolo de la tercer columna es el que denotará cuánto dura dicho silencio. La cuarta columna de la ilustración posee su duración en la unidad llamada “*Tiempos*”. Sabiendo que, como antes fue descrito, la *Redonda* representa la unidad de valor, se define que la unidad de valor es equivalente a 4 tiempos².

NOMBRE	FIGURA	SILENCIO	TIEMPOS
REDONDA (Unidad)			4 Tiempos
BLANCA (Mitad)			2 Tiempos
NEGRA (Cuarto)			1 Tiempo
CORCHEA (Octavo)			1/2 Tiempo

NOMBRE	FIGURA	SILENCIO	TIEMPOS
SEMICORCHEA (Dieciseisavo)			1/4 Tiempo
FUSA (Treintadosavo)			1/8 Tiempo
SEMIFUSA (Sesentaicuatavo)			1/16 Tiempo

Figura 2. Figuras musicales y sus duraciones.
Nota: Figura extraída y modificada de Sr. Imposible (2024).

Las figuras no son el único indicador de duración en una nota. Además de la figura, se puede agregar una “*Alteración de duración*”. Cuando en notación musical se agrega un punto al costado del símbolo correspondiente a la nota, a la duración de esta se aumenta en la mitad de la duración de la misma nota. Por ejemplo, si se tiene una figura de *Redonda* con un puntillo al costado, esta durará 6 tiempos, es decir, los respectivos 4 tiempos de la figura, y además la mitad de su duración, es decir, 2 tiempos más.

La medida de “*Tiempo*” no es constante, y depende de la manera en que se agrupe el transcurso del tiempo. En música existe una manera de dividir el paso del tiempo en segmentos que tiempo que agrupan notas. A estos segmentos se les llama “*Compás*”. Se entiende como compás a la agrupación de notas en una pieza musical, y su duración está delimitada por un valor llamado “*Indicador de compás*”. Un compás es entonces una delimitación de tiempo que agrupa una secuencia de *notas*. Si un compás se define con duración de una *redonda* (es decir, de una *unidad de valor*), entonces ese compás agrupará notas cuya secuencia de valores fraccionarios sume el valor de la unidad. Dicha secuencia de notas (que como se mencionó anteriormente, incluye tanto el *tono* como la *duración*), conforman un compás (Danhauser, 2008). El “*Indicador de compás*” por su parte, según Read (1974), es un símbolo que se presenta al inicio de una obra musical o cuando se requiere un cambio en la estructura rítmica durante la pieza. El indicador de compás está compuesto por dos números que funcionan como una fracción: el numerador (número superior) y el denominador (número inferior). Estos números juntos determinan cómo se agrupan y estructuran las duraciones de las notas dentro de cada compás. Es decir, se mencionó anteriormente que para entender la duración de una nota se usan figuras, y cada figura tiene un valor de cantidad de *Tiempos*. Estos valores de tiempo están dados por el indicador de compás, en donde el denominador (número inferior) indica qué figura rítmica representa un tiempo (por ejemplo, si el denominador es 4, una negra equivale a un tiempo), y el numerador indica cuántos de esos tiempos caben en cada compás. Así, la duración en tiempos de cualquier figura se calcula en relación con la figura que representa la unidad de tiempo según el compás³.

² Esta aseveración aplica para compases 4/4 (consulta enseguida el concepto de “*Indicador de Compás*”). El llamado “*Tiempo*” se puede asociar a cada “*pulso*” en el “*Tempo*”, es decir, si una *Redonda* vale 4 tiempos, implica que si el *Tempo* es de 100 pulsaciones por minuto (bpm), la *redonda* tomará el espacio de 4 de estas pulsaciones, pero esto puede cambiar según el indicador de compás.

³ A menudo en la literatura suele referírsele al indicador de compás, solamente como compás.

- **Numerador:** Indica la cantidad de tiempos o pulsos que se agrupan en cada compás. Por ejemplo, en un compás de 4/4, el numerador "4" establece que cada compás contendrá cuatro tiempos o pulsos.
- **Denominador:** Indica qué figura rítmica representa un tiempo o pulso dentro del compás. Está relacionado directamente con la *duración* de las figuras musicales, que define cuánto tiempo dura cada nota en relación a la redonda, que es la *unidad de valor*. Por ejemplo, si el denominador es "4", esto indica que la negra (1/4 de una redonda) representa un tiempo en ese compás. Si el denominador es "8", significa que la corchea (1/8 de una redonda) es la que ocupa un tiempo en el compás.

Para entender mejor los conceptos de *Tiempo* y *Compás*, hace falta entender que, así como el *Tono* utiliza una medida física para delimitar el sonido, para la duración se utiliza otra medida física para delimitar el transcurso del tiempo: El "*Tempo*". El *Tempo* es la medida cuyo valor determina la velocidad con la que se ejecutan los "*Pulsos*" (o "*Tiempos*") en una obra musical; es decir, la velocidad a la que se suceden los tiempos dentro de un compás. El tempo suele representarse en la medida "*pulsos por minuto*", *beats per minute (bpm)* en inglés. Por ejemplo, si una pieza musical tiene un tempo de 60 bpm, significa que cada pulso del compás, es decir, cada *Tiempo* tiene como duración la 60va parte de un minuto (que es equivalente a un segundo), si en cambio el tempo es de 120 bpm, los pulsos ocurren el doble de rápido, pues cada uno es la 120va parte de un minuto. Es entonces que la manera de entender la duración de una nota depende, primeramente del *Tempo* (pues este indica cuánto dura cada "*Tiempo*" o "*Pulso*" con respecto a un minuto) y finalmente del indicador de compás, que define, con su denominador, qué figura rítmica equivale a un Tiempo, y con su numerador, cuántos de esos Tiempos constituyen un compás. Por ejemplo, en un compás de 4/4 con un tempo de 60 bpm, cada *negra* (1/4 de una *redonda*, como lo indica el denominador "4") durará exactamente un segundo, ya que el denominador señala que la *negra* es la figura que representa un pulso. Si el compás es de 3/8 y el tempo es de 120 bpm, entonces cada *corchea* (1/8 de una *redonda*, según el denominador "8") durará medio segundo, alineando los pulsos al tempo.

El "*Ritmo*" es un concepto que nace de entender las duraciones de las notas en una pieza. Moncada García (1966) define el ritmo como "*Orden y la proporción en que se agrupan los sonidos en el tiempo*". Y se pueden entender como los patrones en que se agrupan las notas según sus duraciones en una pieza musical. El ritmo considera además ciertas reglas de acentuación. Ya se mencionó que en la música el paso del tiempo se divide en fragmentos de tiempo llamados *compases* y estos *compases* están formados por *tiempos*, en donde el *indicador de compás* define cuántos *tiempos* tiene cada *compás* y qué duración tiene cada tiempo; bajo este hecho, la teoría de acentuación en la música designa jerarquías a los tiempos de un compás, dándole a cada uno una importancia entre tres categorías: *tiempo fuerte*, *tiempos semi-fuerte* y *tiempo débil*. Las reglas para designar qué categoría de tiempos se tienen, son, según Pattison (1991) y Teoría Musical (2021):

1. El primer tiempo de un compás siempre es *fuerte*.
2. No puede haber dos tiempos *fuertes* seguidos.
3. No puede haber más de dos tiempos *débiles* seguidos.

Por ejemplo, para compases según su clasificación:

1. **Compas binario:** Cuando hay 2 tiempos, estos son: *Fuerte, Débil*.
2. **Compas ternario:** Cuando hay 3 tiempos, estos son: *Fuerte, Débil, Débil*.
3. **Compas cuaternario:** Cuando hay 4 tiempos, estos son: *Fuerte, Débil, Semi-fuerte, Débil*.
4. **Compás irregular (amalgamas):** Cuando la cantidad de tiempos es distinta a 2, 3 o 4, para interpretarse se debe subdividir en grupos de 2, 3 y 4. Por ejemplo, un compás con 5/4 tiempos puede ser representado como 2/4 + 3/4, o bien como 3/4 + 2/4, de manera que cada subdivisión seguirá las reglas de categorización anteriores. Esta manera de subdividir dependerá del compositor y de cómo quiera acentuar el tiempo en su obra.

Habiendo entendido a la *Nota*⁴ como una unidad básica compuesta por *Tono* y *Duración*, nace el concepto de "*Acorde*". "*La palabra Acorde, puede definirse como cualquier combinación de notas interpretadas*

⁴ En la literatura, así como en este trabajo, a menudo se usa el término "*Nota*" para referirse exclusivamente al "*Tono*".

simultáneamente” (Scholes & Ward, 1977). Los acordes son el corazón de la *armonía* (consultar definición en **6.1.b Armonía**), son estos los que les dan sentido a las *melodías*. Un acorde es en esencia una agrupación de varias notas que se ejecutan al mismo tiempo, formando un acompañamiento a frases melódicas. Los acordes reciben un nombre respectivo a las notas que suenan, tomando una de estas notas (llamada *nota base* o *raíz*) y dándole al acorde el nombre de dicha nota, además de un apellido que depende del resto de notas que estén sonando. Según Steinke & Harder (2009), esta combinación de notas puede categorizar los acordes en los distintos tipos, según la cantidad de estas:

1. Acordes de dos notas: Los más comunes son:

- a. **Acordes de quinta justa (power-chords):** Acordes cuyas dos notas tienen una distancia de *quinta justa*, es decir, siete semitonos. Se conforma de una *nota base* (que le da el nombre al acorde) y la nota que está una quinta por arriba. Por ejemplo: *C4* y *G4*. En notación inglesa suelen representarse como; la nota base (que nombra al acorde) y el número 5. En este caso, el acorde formado por *C4* y *G4* es llamado acorde de quinta justa sobre *Do4* (*C5*). No importa de qué octava sea la nota base, el nombre será igual si tiene sonando su respectiva quinta⁵. La **Figura 3** contiene una ilustración con las notas del acorde de quinta justa sobre *Do*.



Figura 3. Acorde de Do con quinta justa.

- b. **Acorde de octava justa:** Cuando las dos notas que suenan tienen el mismo nombre, por ejemplo: *C4* y *C5*. No suele tener notación específica para referirse a este, puesto que al no aportar mucha información armónica (pues ambas son la misma nota, pero de distintas octavas), no se le refiere usualmente como acorde.
 - c. **Acordes de cuarta justa:** Acordes cuya nota base suena a la par que la nota a una cuarta justa por encima (5 semitonos). Al igual que los acordes de quinta justa, se simbolizan por la nota base y un número 4. Ejemplo: *C5* y *F5* forman el acorde *C4*.
- 2. Triadas:** Son los acordes más comunes en la *música tonal popular*. Se forman de tres notas, en donde una de las notas (*nota base*) da nombre al acorde y el resto le dan apellido al acorde:
- a. **Acordes mayores:** Los más comunes en la música popular, se forman por una nota base y agregando las notas que estén a distancia de una tercera mayor y quinta justa (es decir, las notas a 4 semitonos y 7 semitonos). Su sonoridad suele estar asociada a la *felicidad*. Ejemplo: *C*, *E*, *G* forman el acorde de *C* (*Do mayor*). En notación inglesa sólo se coloca la letra correspondiente al acorde, de manera que *D* representa un acorde de *Re mayor*, *Bb* un acorde de *Si bemol mayor*, *G#* un acorde de *Sol sostenido mayor*, etc. La **Figura 4** ilustra en color rojo las notas del acorde de *Do mayor*.

⁵ Es importante entender que la notación musical puede ser ambigua, pues “*C5*” podría referirse a la nota *Do-5* o bien a un acorde de quinta justa sobre *Do*. De la misma manera que, “*Do*” (*C*) puede referirse a un acorde mayor de *Do*, o a la escala mayor de *Do*, o simplemente a una nota llamada *Do*.

Acorde de C (Do mayor)



Figura 4. Acorde de Do mayor.

- b. **Acordes menores:** Tan comunes como los acordes mayores. Su sonoridad suele asociarse a la *tristeza*. Se forman con una nota base y las notas a distancia de tercera menor y quinta justa (3 semitonos y 7 semitonos respectivamente). Estos acordes se señalan en notación inglesa usando la nota base del acorde y una letra “m” minúscula. Por ejemplo: C, Eb y G forman el acorde Cm (Do menor)⁶. Acorde ilustrado en la **Figura 5**.

Acorde de Cm (Do menor)



Figura 5. Acorde de Do menor.

- c. **Acordes aumentados:** Mucho menos comunes que los acordes mayores y menores. Su sonoridad suele asociarse al sentimiento de *intigra* y *tensión*. Se forman con una nota base, una tercera mayor y una quinta aumentada (quinta aumentada un semitono). En notación inglesa se representan comúnmente como la nota del acorde y “aug” (y aunque menos común, también se suele utilizar el símbolo “+” en lugar del “aug”). Ejemplo: C, E, G# forman el acorde de Caug (Do aumentado). La **Figura 6** le ilustra.

Acorde de Caug (Do aumentado)

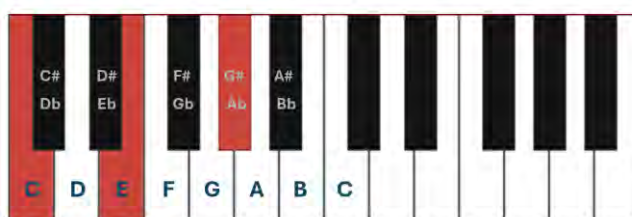


Figura 6. Acorde de Do aumentado.

- d. **Acordes disminuidos:** Denotados en notación por el nombre de la nota y “dim” (o también el símbolo “-“) y tan poco comunes como los acordes aumentados. Se conforman de una nota base, la tercera menor y *tritono*. El *tritono* es un intervalo separado por seis semitonos, también es llamado cuarta aumentada y quinta disminuida. El tritono aporta un sentimiento de *tesión* y *terror*, y es por ello por lo que en la edad media se le consideraba un intervalo diabólico. Ejemplo: C, Eb, Gb forma el acorde de Do disminuido: Cdim, ilustrado en la **Figura 7**.

⁶ Como regla general, los acordes mayores y menores son en esencia acordes de quinta justa, a los que se le agrega una nota a intervalo de tercera, que, según su tipo, le dará sonoridad de mayor. Esta es llamada “nota distintiva”.

Acorde de Cdim (Do disminuido)

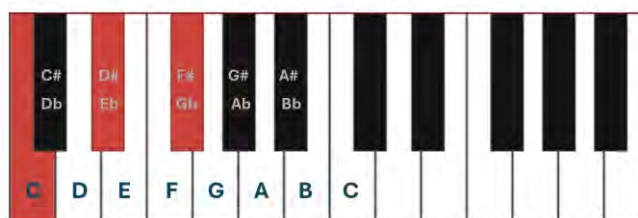


Figura 7. Acorde de Do disminuido.

3. **Tetracordes:** Acordes con cuatro notas. Son casi tan comunes como los acordes mayores y menores y usualmente suelen ser una extensión de estos mismos, con alguna nota extra que agrega sonoridad al mismo. Los más comunes son los acordes de séptima; acordes mayores o menores a los que se les agrega una séptima menor o mayor (nota a intervalo de 10 y 11 semitonos de la nota base respectivamente), y que por su naturaleza provocan un sonido de “*preparación*”, es decir, que generan una tensión que prepara el oído para recibir enseguida un acorde que resuelva dicha tensión:

- a. **Acorde mayor de séptima menor:** También llamados únicamente *acordes de séptima*. Son acordes mayores a los que se les agrega una séptima menor (nota a intervalo de 10 semitonos de la nota raíz). Su sonoridad genera una tensión que provoca al oído a querer escuchar el siguiente acorde. Cuando el acorde es el quinto grado de una escala, se le llama *acorde de dominante* y genera una tensión que fuerza al escucha a querer escuchar el acorde de *tónica* de la tonalidad para *resolver* (consulta 6.1.b Funciones tonales). Ejemplo: C, E, G, Bb, que forma el acorde que se denota como un acorde mayor y el número 7: C7 (acorde de Do séptima). El acorde se muestra en la Figura 8.

Acorde de C7 (Do séptima)

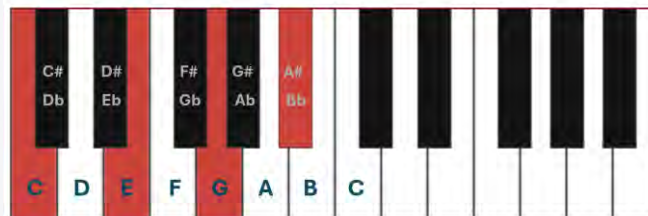


Figura 8. Acorde de Do séptima.

- b. **Acorde mayor de séptima mayor:** Formado por un *acorde mayor* al cuál se le agrega una *séptima mayor* (nota a intervalo de 11 semitonos por encima de la nota raíz). Su sonoridad es “*cristalina*”. Es un acorde muy agradable a la escucha. Ejemplo: C, E, G, B; que se denota como Cmaj7 en notación inglesa (acorde de Do con séptima mayor), aunque también suele representarse con el símbolo “ Δ ” (C Δ). Este acorde se ilustra en la Figura 9.

Acorde de Cmaj7 (Do con séptima mayor)

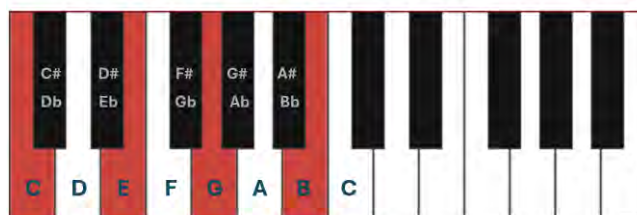


Figura 9. Acorde de Do con séptima mayor.

- c. **Acorde menor con séptima menor:** Acorde formado por un acorde menor al que se le agrega una séptima menor. Su sonoridad suele ser asociada a la *calma*. Por ejemplo: C, Eb, G, Bb, denotado como Cm7, visible en la **Figura 10**.

Acorde de Cm7 (Do menor séptima)



Figura 10. Acorde de Do menor séptima.

- d. **Acorde menor con séptima mayor:** Acordes menores a los que se les agrega una séptima mayor. Su sonoridad suele estar asociada al *misterio*. Ejemplo: C, Eb, G, B. Su notación es confusa, pero se les puede denotar como un acorde menor al cuál se le agrega “(maj7)” al costado o “(M7)”. En este caso, el acorde sería Cm(maj7) o Cm(M7) y las notas que lo conforman se ilustran en la **Figura 11**.

Acorde de Cm(maj7) (Do menor con séptima mayor)

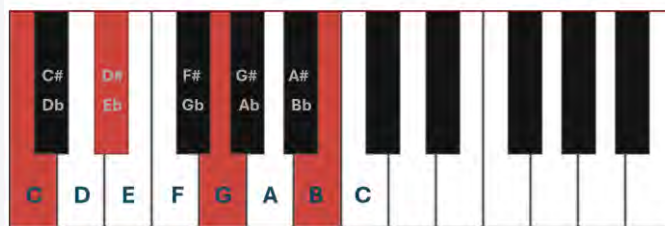


Figura 11. Acorde de Do menor con séptima mayor.

4. **Acordes suspendidos:** Acordes de triada cuya sonoridad genera una tensión muy grande que busca enseguida resolver a un *acorde mayor*, debido a la disonancia (cercanía muy corta entre un par de notas) generada con la *nota raíz* (para un acorde sus2) o la *quinta justa* (para un acorde sus4).

- a. **Acorde sus4:** Formado por la nota raíz, la cuarta justa y la quinta justa. Es como un *acorde mayor*, pero la *tercera mayor* se ha aumentado a *cuarta*, lo que genera en la escucha una necesidad de escuchar enseguida el acorde con su *tercera mayor* (*acorde mayor*). Ejemplo: C, F, G (acorde Csus4), que busca *resolver* a C, E, G (acorde C). Se ilustra en la **Figura 12**.

Acorde de Csus4 (Do suspendido con cuarta)



Figura 12. Acorde de Do suspendido con cuarta.

- b. **Acorde sus2:** Misma idea que el acorde sus4, pero en lugar de generar tensión con una *cuarta justa*, genera tensión con una *segunda mayor* (nota a dos semitonos de la nota base), que

igualmente busca resolver en el acorde mayor. Ejemplo: C, D, G (acorde Csus2) que busca resolver a C, E, G (acorde de C), mostrado en la **Figura 13**.

Acorde de Csus2 (Do suspendido con segunda)

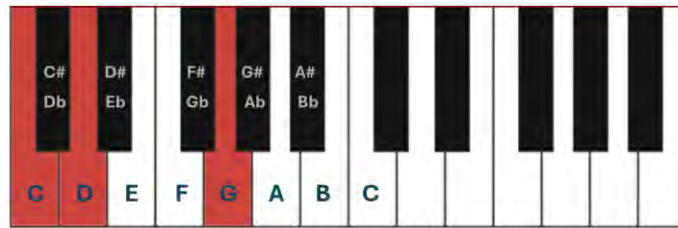


Figura 13. Acorde de Do suspendido con segunda⁷.

5. **Acordes invertidos:** Se ha definido que los acordes se forman tomando una nota base y agregando notas usando intervalos hacia arriba; por ejemplo, con el acorde mayor de Do se toma la nota Do (C4), después su tercera mayor por encima (E4) y después su quinta siguiente (G4). Esto nos dice que un acorde de Do mayor está formado por las notas C, E y G y esto es verdad para cualquier contexto, sin importar la octava en la que suenen ni el instrumento que las ejecute, si estas tres notas en cualquiera de sus frecuencias es interpretada simultáneamente, a esto se le llamará *acorde de Do mayor*. Esto implica que existirán escenarios en los cuales las notas del acorde suenen en distinto orden, a esto se le llaman inversiones. Si se tiene el acorde de Do mayor sonando con la nota fundamental (Do) como la nota más baja, se dice que es un acorde en su *forma fundamental* (C, E y G en este caso). Si se toma la segunda nota del acorde (que en este caso es la tercera mayor, es decir E) y se recorre de manera que esta tenga la frecuencia más baja, entonces se dice que el acorde está en su *primera inversión* (E, G, C). Si en cambio se toma la tercer nota (la quinta, que en este caso es G) y se recorre siguiendo la misma lógica, entonces se dice que se tiene el acorde de Do en *segunda inversión* (G, C, E). Si bien todos estos acordes siguen siendo el acorde de Do mayor, su sonoridad es ligeramente distinta. Se muestran en la **Figura 14**.

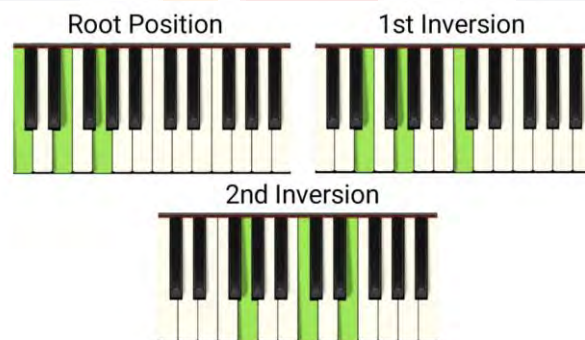


Figura 14. Inversiones del acorde mayor de Do.

Nota: Figura tomada de PianoChordsPlays (2023).

6. **Acordes de más de cuatro notas:** Es importante hacer la anotación de que cualquier combinación de notas sonando al simultáneo puede representar un acorde, y por ende hay una infinita posible combinación de acordes, aunque sus clasificaciones serán finitas.

Al hablar de acordes, surge el término de “*Disonancia*”. La disonancia implica que cuando un sonido junto a otro u otros produce una sensación incómoda al oyente, principalmente relacionado a cuando las frecuencias de dichos sonidos no tienen ninguna relación numérica (De la Herrán, 2007). Si bien la disonancia

⁷ Las figuras desde la 3 hasta la 13 fueron extraídas y modificadas desde C chords (s/f).

La **Figura 15** muestra la representación de la partitura el primer movimiento de la quinta Sinfonía de Beethoven, en donde se encierra en color azul el *motif* de la obra, siendo este *motif* una agrupación de tres notas consecutivas iguales de misma duración que enseguida tienen una nota de mayor duración por debajo de las tres primeras. El *motif* es entonces esta agrupación específica de cuatro notas, que se repite a lo largo de toda la obra con variaciones de altura. Sin importar qué notas se elijan y a qué altura, este motivo será el mismo, y en suma con el resto de las repeticiones y variaciones (considerando qué notas se usarán en cada caso) se forma la melodía.

6.1.b.3 Armonía

Definida como: “*Arropamiento de la melodía*” según Scholes & Ward (1977). Y definida además como “*parte de la música que estudia la formación y combinación de acordes*” por Piston et al. (1987). La armonía es un concepto muy extenso en la teoría musical, pues es, en sí misma, un área completa de estudio, que se enfoca en las diversas estrategias en la combinación de acordes y las sensaciones que producen. Piston et al. desarrollan que la armonía constituye la combinación de notas independientemente de su función; es decir, sin importar si se trata de melodía u sólo acordes, todo lo relacionado a la combinación de notas que suenen en simultáneo se le considera armonía. Por ejemplo, si se colocasen dos melodías que suenen simultáneamente, aquellas notas que suenan a la par generan una armonía en sí, y por lo revisado en el concepto de *acorde*, se puede decir que esto en sí mismo es un acorde. Para el propósito de este trabajo, armonía se referirá simplemente a la sucesión de acordes que se utilicen para acompañar a una melodía.

- **Progresión de acordes:** También llamadas “*progresiones armónicas*”, es entendida simplemente como una sucesión de acordes, principalmente asociada a una melodía, pero no necesariamente. En este trabajo se utilizará en ocasiones como sinónimo de armonía, aunque el concepto de armonía es mucho más extenso en sí mismo.
- **Escala:** El espectro sonoro comprende un rango de frecuencias infinitas del cuál el ser humano sólo puede escuchar un rango entre 20Hz y 20,000Hz, equivalente a diez octavas completas (Rumsey & McCormick, 2004). Este rango es dividido en grupos de tonos a los cuales se les da un nombre en específico; de estos se tienen doce tonos, que repiten su nombres según qué octava. Si bien se tienen doce notas (y sus tonos homónimos en el rango de frecuencias audibles) son utilizables en la creación de piezas musicales, según la pieza no todas son utilizadas. Cuando se selecciona sólo una serie de las doce notas, es que se construye lo llamado *escala*. A cualquier subconjunto de estas notas se le llama *escala*, y según el criterio para seleccionarse, es que se llaman de distinta manera. El diccionario de música de Oxford define las escalas como una “*disposición escalonada de todas las notas principales de un pasaje musical en particular o del sistema musical de un período o de un pueblo.*” (Scholes & Ward, 1977), y por ende habla que la selección de estas notas depende de la cultura, la época o el sistema musical, y es según estas selecciones que, en el ya definido sistema de *Temperamento igual* se conocen algunas escalas de las cuales a continuación se enlistan las más importantes y representativas. Una notación posible para definir el patrón de selección de las notas se hace tomando como referencia una *nota base* (al igual que con los acordes) que irá en el nombre de la escala y contando los intervalos para seleccionar las notas subsecuentes, con conteo en *Semitonos (S)* y *Tonos (T)*:

1. **Escala cromática:** Llamada a la escala que posee todas las doce notas.

2. **Escala diatónica:** Escala que posee siete sonidos. Las más representativas son las escalas mayores y menores, a partir de las cuales se constituye la gran mayoría de la música tonal popular.

- **Escala diatónica mayor:** Se toma una nota base y su patrón de selección de notas hacia arriba es: *T-T-S-T-T-T-S*. Por ejemplo, una escala mayor de *C* se construye tomar la nota *C* como base y seleccionar enseguida con base a su patrón; en este caso se seleccionarían las notas: *C, D, E, F, G, A, B* (y de nuevo *C* para repetir la selección sobre todo el espectro de notas audibles), siendo esta la escala llamada “*Escala mayor de Do*” o simplemente “*Escala de Do mayor*”. La **Figura 16** destaca en un teclado de piano todas las notas que componen una escala de ejemplo (*D#* para este caso).



Figura 16. Escala de Re# (Mib) mayor.

- **Escala diatónica menor:**
 - a. **Escala diatónica menor natural:** También llamada sólo *Escala menor*, se construye con el patrón T-S-T-T-S-T-T. Por ejemplo, tomando como base la nota A, se encuentran las notas A, B, C, D, E, F, G, siendo la “Escala menor de La” o simplemente “Escala de La menor”⁸.
 - b. **Escala diatónica menor armónica:** Derivada de la escala menor natural, pero su séptimo grado (es decir, la séptima nota contando la de base como la primera) está aumentada un semitono más; es decir: T-S-T-T-S-3S-T.
 - c. **Escala diatónica menor melódica:** Derivada de la escala menor natural, su sexto grado (sexta nota contando desde la base) está aumentado semitono, al igual que el séptimo: T-S-T-T-T-T-S.
- 3. **Escala pentatónica:** Escala que posee cinco sonidos. Las más representativas:
 - **Escala pentatónica mayor:** Patrón T-T-3S-T-3S.
 - **Escala pentatónica menor:** Patrón 3S-T-T-3S-T.
 - **Escala de blues mayor:** Patrón T-3S-T-T-3S.
 - **Escala de blues menor:** Patrón 3S-T-3S-T-T.
 - **Escala egipcia (pentatónica suspendida):** T-3S-T-3S-T.
 - **Escala japonesa:** T-2T-T-T-2T.
- 4. **Escalas modales:** También llamadas *grados* o *modos griegos*, son tipos de escalas diatónicas con nombres y sonoridades definidas:
 - **Modo Jónico:** Es la escala mayor.
 - **Modo Dórico:** T-S-T-T-T-S-T.
 - **Modo Frigio:** S-T-T-T-S-T-T.
 - **Modo Lidio:** T-T-T-S-T-T-S.
 - **Modo Mixolidio:** T-T-S-T-T-S-T.
 - **Modo Eólico:** Es la escala menor melódica.
 - **Modo Locrio:** S-T-T-S-T-T-T.

Todos estas escalas pueden ser consultadas en el libro “*The Complete Book of Scales, Chords, Arpeggios & Cadences: Includes All the Major, Minor (natural, Harmonic, Melodic) & Chromatic Scales Plus Additional Instructions on Music Fundamentals*” (Palmer et al., 1994).

6.1.b.4 Tonalidad

El autor Dionisio de Pedro (2008a) define a la *tonalidad* como “un conjunto de sonidos ordenados mediante relaciones mutuas, estando estas determinadas por un sonido básico llamado *Tónica*”. Se mencionó en la definición de *escala* que cuando se compone una canción, por lo general se utilizan únicamente una selección de

⁸ Es necesario notar que; siguiendo las reglas de selección de los patrones mayores y menores, es posible conseguir las mismas escalas partiendo de distintos puntos; por ejemplo, las escalas de Do mayor y La menor contienen exactamente las mismas notas, si bien su nota fundamental es distinta (Do y La respectivamente). A estas escalas se les llama “Escala relativas” y se nombran respecto la una a la otra. La escala de La menor es la Relativa menor de Do y la escala de Do mayor es la Relativa mayor de La. Si bien ambas escalas tienen las mismas notas, no se les considera iguales, pues al tener de nota base una nota distinta (también llamada *Tónica*), su contexto armónico será distinto en términos de Tonalidad (consultar los conceptos de tonalidad, funciones tonales, reposo y tensión en este mismo apartado).

las doce notas del *temperamento igual*, cuyo nombre es *escala*, y que se seleccionan partiendo de una nota base llamada *tónica*, que le da nombre a dicha escala. Esta nota tónica es el centro de una escala y hacia esta es que todo el resto de las notas de la escala direcciona generando lo que se denomina *reposo* (definido posteriormente). Una pieza musical en la *música tonal* (definida en **6.1.c.i Música Tonal**) tiene definida una *tonalidad*, es decir, al igual que con una escala, la tonalidad define qué notas se utilizarán en dicha pieza, siendo la tónica la nota que da también el nombre a la tonalidad y hacia donde todas las notas y acordes en la pieza querrán *resolver* para terminar y generar *reposo*. Se podría decir que la *tonalidad* y la *escala* de una pieza están relacionadas, y si bien esto es verdad, la tonalidad es más extensa en su definición, pues no sólo dictamina qué notas se deberían utilizar, sino qué acordes. En el diccionario de música de Oxford se le define como “*la observancia de lo que puede llamarse 'lealtad a una tónica', es decir, al esquema tonal de una composición*” (Scholes & Ward, 1977), por lo que se entiende que la tonalidad es cómo una pieza musical, sus notas y sus acordes girarán en torno a una nota tónica. En inglés el concepto se llama “*Key*”. “*Se entenderá que todas las tonalidades mayores son exactamente iguales, al igual que todas las menores, de modo que la tonalidad particular elegida para una composición es más una cuestión del nivel de tono deseado que de algo más importante*”, de ahí que el concepto de *transposición* refiere a que la relación entre las notas de una pieza en la tonalidad de, por ejemplo, *Do mayor* es igual que si la misma pieza se interpretara en *Sol mayor*. Al ser ambas tonalidades mayores, la relación de distancia entre sus notas será la misma en una o en otra, si bien cada una utilizará notas y escalas totalmente diferentes.

- **Funciones tonales:** La relación de los acordes dentro de una escala y la función que cumplen estos dentro de la tonalidad de una pieza (Piston et al., 1987). Se mencionó en el concepto de *tonalidad* que cada pieza en la *música tonal* posee una tonalidad definida y que esto definirá qué notas se poseen y qué acordes se pueden usar, respondiendo siempre a una nota *tónica* que le da nombre a la escala y a la tonalidad. Cada tonalidad contiene la lista de acordes que preferentemente estarán en la obra, sin embargo, estos acordes tienen una jerarquía según su función con respecto a la nota tónica. Por ejemplo, una pieza en tonalidad de *Do mayor* posee los acordes: *Do mayor*, *Sol mayor*, *Fa mayor*, *La menor*, *Mi menor*, *Re menor* y *Sí aumentado*, sin embargo, cada acorde tiene una función distinta con respecto a la tónica (*Do*). Los acordes se obtienen a partir de generar las reglas de generación de acordes (consultar **6.1.a acordes**) tomando como base cada una de las notas de la escala de la tonalidad. Dado que las notas de la escala de *Do mayor* son *Do*, *Re*, *Mi*, *Fa*, *Sol*, *La* y *Si*, los acordes de dicha tonalidad se construirán tomando de nota base cada uno de esos acordes de la escala, dándole un número denominado *grado*, a cada nota y por ende a cada acorde generado, siendo cada grado un número romano dado por el orden de las notas de la escala, donde para *Do mayor*, los grados son: *Do*: I, *Re*: II, *Mi*: III, *Fa*: IV, *Sol*: V, *La*: VI y *Si*: VII. Dado que la música tonal está compuesta principalmente de tonalidades mayores y menores (es decir, de escalas diatónicas), cada tonalidad mayor o menor tiene siempre siete grados, donde la función de cada uno está nombrada en la siguiente **Tabla 2**.

Tabla 2. Nombres y funciones de los grados asignados a los acordes en una tonalidad diatónica.

Grado	Nombre / Función	Descripción
I	TÓNICA	Nombre del tono
II	Supertónica	Encima de la tónica
III	MEDIANTE / Modal	En medio del I y V (determina el modo)
IV	SUBDOMINANTE	Por debajo de la dominante
V	DOMINANTE	Domina el mecanismo tonal (da tensión ante la tónica)
VI	Superdominante	Encima de la dominante

VII	SENSIBLE / Subtónica	Debajo de la tónica
VIII (I)	TÓNICA	Repetición del I

Nota: Tabla extraída de “*Teoría Completa de la Música I*” de Dionisio de Pedro (2008a).

La **Tabla 2** muestra para una tonalidad mayor o menor cualquiera (diatónica con siete notas en su escala y por ende siete grados) los nombres que recibirá cada uno de los acordes de la tonalidad/escala. Estos nombres le dan la función cumplirá cada acorde en una pieza musical en dicha tonalidad con respecto a la tónica, también conocida como la nota/acorde de reposo.

- **Clasificación de grados diatónicos:** En escalas diatónicas, cada grado tiene una clasificación según su función, teniendo las siguientes (Moncada García, 1966):
- **Grados tonales:** Son el I, V y IV. Establecen la tonalidad de la pieza y proporcionan un sentido de estabilidad y reposo, siendo la *tónica* (grado I) el reposo, el *dominante* (V) el direccionador⁹ hacia el reposo, y el *subdominante* (IV) que prepara la resolución para el *dominante* o la tónica.
- **Grados modales:** Son el III y VI. Aportan color y carácter al modo. El III (llamado *Modal Primario*) define el modo de la escala (mayor o menor) y el VI (*Modal secundario*) refuerza su carácter modal y sirve como puente armónico hacia otros grados.
- **Grados atractivos:** Son VII, II y IV, generan tensión y movimiento hacia los grados tonales, creando dinamismo en la progresión armónica. El grado VII tiene a direccionar y resolver hacia el I (tónica), el II hacia el V (dominante) y el IV puede conducir al V o directamente a la tónica¹⁰.

6.1.b.5 Reposo

Es el estado de “*estabilidad*” y resolución armónica en una pieza musical. Ocurre cuando los grados tonales, especialmente el grado I (tónica), son utilizados como punto de llegada o conclusión de una *progresión de acordes*. La sensación de reposo proporciona calma y un cierre perceptible en la estructura musical, funcionando como el “hogar” armónico de la tonalidad. El direccionamiento de los acordes (grados) de una tonalidad tenderán a querer *resolver* hacia la tónica, generando reposo. Entendiendo *resolver* como la sensación producida por terminar una progresión en un acorde que genere reposo, como lo puede ser el primer grado (Piston et al., 1987).

6.1.b.6 Tensión

Es el estado de inestabilidad o expectativa armónica que impulsa el movimiento hacia el reposo. Se genera principalmente mediante los *grados atractivos* (como el VII o el II) y los acordes que contengan disonancias o resoluciones pendientes. La tensión es un recurso clave para crear interés y dirección en una pieza musical, ya que establece un contraste que lleva a la resolución armónica y melódica e impulsa al oyente a sentir una sensación de avance armónica que le quiera hacer llegar al reposo (Piston et al., 1987).

6.1.b.7 Cadencia

Es un tipo de progresión de acordes especial, en donde “*el encadenamiento de dos o más acordes coinciden con una detención definitiva o momentánea y que además debe tener un sentido tonal conclusivo o suspensivo*” (Dionisio de Pedro, 2008b). Cuando se habla de cadencias se habla de configuraciones específicas de progresiones de acordes que por su dirección tonal generan una sensación de resolución (reposo), pues marcan el final de una frase musical, ya sea de forma definitiva o momentánea, estas diferentes tipos de configuraciones favorecen la creación de armonías con sentido melódico. Se tienen distintos tipos de cadencias:

⁹ Cuando se habla de dirección en los acordes se habla de la sensación que se genera en el oyente al oír un acorde y presentir cuál será el siguiente acorde. El V grado se dice que tiene fuerte dirección al grado I, pues al oír una pieza en cierta tonalidad y escuchar el acorde de grado V, el oído interpreta que enseguida viene el grado I. Este grado de certidumbre del oyente es la medida de dirección que tiene un acorde con otro en determinada tonalidad.

¹⁰ Estas y más funciones de los grados pueden ser estudiadas de (Schoenberg, 1911, 1983).

1. **Cadencias conclusivas:** Conllevan una plena confirmación tonal, pues el acorde de tónica coincide con el punto de reposo de la progresión:
 - **Cadencia perfecta (auténtica):** Acorde de Dominante (triada mayor o bien triada mayor con séptima), con el acorde de Tónica enseguida, ambos en estado fundamental: V – I.
 - **Cadencia imperfecta:** Cadencia perfecta cuando alguno de los dos acordes está invertido. Cuando el acorde de tónica está en estado fundamental, la cadencia imperfecta conserva su carácter conclusivo, y este se pierde si se invierte. En esta cadencia el acorde de Dominante puede ser sustituido por el acorde de quinta disminuido con su nota más baja siendo la *sensible*: VII 5 – I.
 - **Cadencia Plagal:** Encadenamiento del acorde de Subdominante junto al acorde de Tónica, estando ambos en estado fundamental: IV – I. El Subdominante también podría ser sustituido por los acordes de II o VI: II – I o VI – I. Existe una variante “frecuente en el lenguaje romántico” (Dionisio de Pedro, 2008b), llamada *cadencia plagal suavizada*, que consiste el acorde perfecto mayor del cuarto grado del modo mayor en un acorde perfecto menor.
 - **Cadencia Plagal imperfecta (plagal invertida):** Cadencia plagal en la que alguno de los acordes se encuentra invertidos. Igual que la *cadencia imperfecta*, su carácter conclusivo se mantendrá siempre que el acorde de tónica esté en estado fundamental.
2. **Cadencia completa:** Sucesión de una cadencia perfecta y una cadencia plagal, en cualquier orden: V – I – IV – I o también IV – I – V – I, bien I – IV – V – I.
3. **Cadencia suspensivas:** Aquellas cuyo punto de reposo no está ocupado por el acorde de tónica, por lo que no generan completamente una sensación de confirmación tonal.
4. **Cadencia rota (interrumpida):** Encadenamiento del acorde de Dominante y cualquier otro distinto del acorde de Tónica, usualmente VI, IV o II. Por ejemplo: V – VI, o también V – IV o bien V – II. Se convierte en imperfecta con la inversión del acorde de Dominante (V). “*Suelen derivar a acordes pertenecientes a otras tonalidades, aumentando así su carácter sorpresivo*” (Dionisio de Pedro, 2008b).
5. **Semicadencia:** Genera un reposo momentáneo sobre un grado distinto a la tónica, siendo los más característicos el V y el IV, aunque también puede producirse dicho reposo sobre II, VI o III, procediéndoles cualquier acorde posible: IV – V – I – VI, o también IV – V. Se convierte en imperfecta cuando se invierte el acorde característico. Esta cadencia también puede recibir el nombre de *Frigia*, cuando se produce entre el IV grado en primera inversión y el V grado, y su uso proviene de la costumbre de convertir en acorde mayor el último acorde de un modo Frigio, “*en la música de J.S. Bach es frecuente ver este acorde (sobre todo en los corales), como acorde final, debido a su carácter eminentemente modal de sus melodías*” (Dionisio de Pedro, 2008b).
6. **Acorde de 6/4 cadencial:** “*Cualquiera de las cadencias descritas pueden ir precedidas de cualquier acorde, ahora bien, es el acorde de cuarta y sexta sobre la Dominante uno de los más frecuentemente utilizados*” (Dionisio de Pedro, 2008b). Este acorde es un acorde de tónica en segunda inversión y su función es la misma que de un Dominante, de ello que posea un poder tonal en direccionar a la tónica.
7. **Tercera de picardía:** Se le llama a la terminación de una pieza en modo menor, estando inicialmente en un modo mayor.
8. **Cadencia andaluza:** Cadencia construida sobre una escala Frigia con terminación en Tónica mayor: IV – III – II – I o bien I – VII – VI – V – I. No es igual a la *cadencia frigia*.

6.1.c Teoría tonal

Ya definidos los conceptos básicos que le dan forma a la música y a lo que compete en este trabajo, en este apartado de discutirán conceptos y teorías relacionadas a cómo se construyen composiciones a través de estos conceptos básicos integrados en la idea de la música tonal.

6.1.c.1 Música tonal

Los autores Steinke & Harder describen en “*Harmonic Materials in Tonal Music, A Programmed Course*” (2009) todo el sistema bajo el que se construye la llama *música tonal*. En este material se define que “*la mayor*

parte de la música compuesta entre 1600 y 1900 se basa en escalas mayores y menores. La música de este tipo se denomina generalmente música TONAL”. Steinke & Harder describen como, en la música tonal se posee una nota llamada tónica, que predomina sobre el resto de los tonos de la escala. Esta nota no sólo le da nombre a la escala/tonalidad, sino que, determina qué función tendrá cada uno de los demás tonos en una pieza musical (consultar **6.1.b Funciones tonales**). El pianista Leo Ravera describe una distinción de música según el cómo se construye, en donde, al igual que en buena parte de la literatura, se distinguen dos enfoques distintos: *música modal* y *música tonal*. Según Ravera, “casi toda la música que escuchamos y conocemos es música tonal. La música clásica de 1600 a 1900 es tonal. Por ejemplo, la música de Bach, Beethoven, Mozart, Chopin, así como todo el repertorio lírico y operístico, pertenecen a la categoría de música tonal”¹¹ (2022). Se explica que la música tonal es aquella que toma una escala principal (o tonalidad) y compone con base a solamente notas de esa escala. El enfoque de la música tonal prioriza la elección de las notas y acordes pertenecientes a la escala/tonalidad, pero sin excluir las demás. Este sistema de composición designa que de una tonalidad se construyen una serie de acordes que “pertenecen” a la misma. El enfoque tonal fuerza a que, además de tener una nota como eje de direccionamiento, también se tenga un acorde principal, llamado y definido anteriormente como *primer grado de la escala* (denotado con el número romano I), el resto de los acordes (grados) se forman a partir de cada nota de la escala, y estos acordes serán los principales a elegir en la composición, y siempre serán una lista finita y definida. Por ejemplo, una escala de *Do mayor*, o bien dicho de otra manera, una pieza musical compuesta en tonalidad de *Do mayor* contiene siete grados, dado que su escala contiene siete notas, y los acordes formados sobre cada una de las notas de dicha escala siempre serán *Do mayor* (grado I), *Re menor* (grado ii), *Mi menor* (grado iii), *Fa mayor* (IV), *Sol mayor* (V), *La menor* (vi), *Si disminuido* (VII)¹². Estos acordes se forman generando triadas de acordes utilizando sólo las notas de la escala.

Ravera describe las jerarquías que reciben las notas dentro de una pieza musical. Las jerarquías según su orden de importancia son:

1. Las notas del acorde que se está tocando en un momento dado.
2. Las notas de la tonalidad de la canción, es decir, las siete notas de la escala.
3. Las notas restantes que no pertenecen a la escala y, por lo tanto, están fuera de la tonalidad.

Se nota por esta jerarquía que, a la hora de componer una pieza de la música tonal, ciertas notas tendrán más importancia de aparecer en ella. Los grupos 2 y 3 ya están definidos al comienzo de una pieza, al elegir la tonalidad y la escala, sin embargo el grupo 1 cambia a lo largo de la pieza.

Tanto Steinke & Harder como Ravera coinciden en que los acordes de una tonalidad tienen jerarquía de importancia, donde el acorde de Tónica (I) y el acorde de Dominante (V) son los más importantes. Esta importancia de los acordes de Tónica y Dominante, y que relacionan directamente la construcción de la música a través de los conceptos de *tensión* y *reposo* (consultar **6.1.b.5 Reposo** y **6.1.b.6 Tensión**), dado que si el acorde de Dominante genera tensión, el acorde de Tónica resuelve la tensión. Además, en “*Techniques and Materials of Music: From the Common Practice Period Through the Twentieth Century, Enhanced Edition*”, por su traducción “*Técnicas y Materiales de la Música: Desde el Período de Práctica Común hasta el siglo XX, edición mejorada.*”, Benjamin et al., escriben que “la música tonal se construye alrededor de estos puntos de llegada tónicos y dominantes (cadencias), y forman uno de los bloques fundamentales de la estructura musical”. Estos principios permiten establecer reglas que benefician la composición (al menos en el contexto armónico), en donde dichas funciones de los acordes se aprovechen para propiciar estructuras armónicas clásicas como la *Cadencia* (consultar **6.1.d.1 Para armonía**).

¹¹ Se nombrará a lo largo de este trabajo la música tonal contemporánea como aquella música tonal entre el 1900 y la actualidad: 2025.

¹² Al representar los grados de una tonalidad con números romanos, es costumbre colocar mayúscula cuando el acorde es mayor y minúscula cuando el acorde es menor.

Resumiendo los aportes de Steinke & Harder, Ravera y Benjamin et al., se extraen los siguientes principios fundamentales de la armonía tonal:

1. “Casi toda la música que escuchamos y conocemos es música tonal”.
2. En la música tonal, las notas y acordes se eligen teniendo preferencias definidas.
3. La armonía en la música tonal se construye a partir de *reposo* y *tensión*, a través del uso de *Tónica* y *Dominante* respectivamente, y de ello se derivan las cadencias.

Steinke & Harder definen cómo el oído comprende estéticas a partir de reglas de construcción definidas, utilizando el siguiente párrafo:

“Como puntos de reposo, las cadencias son extremadamente importantes para el estudio de la armonía. Debido a que son objetivos de la frase en sí, las cadencias tienen un significado estructural especial. El oído tiende a escuchar estos objetivos y relaciona el material circundante con ellos. Los finales de las frases se denominan cadencias” (Steinke & Harder, 2009).

6.1.c.2 Círculo de quintas

El círculo de quintas es un diagrama que organiza en una circunferencia las notas en una secuencia de quintas perfectas. Siendo que si se lee en sentido horario las notas tienen una distancia de quinta, y si se leen en sentido antihorario tendrán distancia de cuarta (Pilhofer & Day, 2011). Esta herramienta está principalmente enfocada en los principios de la música tonal, como aquel en que se sabe que una tonalidad posee únicamente una serie de acordes asociados a ella. Este diagrama contiene dos líneas de notas (niveles), en donde el nivel superior (exterior) posee notas que representan acordes y tonalidades mayores y el nivel inferior (interior) posee los relativos menores a los acordes del nivel superior.



Figura 17. Círculo de quintas.

Nota: Figura extraída de Musicca (s/f).

En la **Figura 17** se tiene el círculo de quintas, en donde el nivel superior (representado en un color verde) contiene notas (o también se puede interpretar como acordes mayores) en secuencia de quintas. El nivel inferior (representado en color grisáceo) contiene las notas/acordes correspondientes al relativo menor de cada nota/acorde el nivel superior. Si se toma el acorde de Do del nivel superior, se sabe que su relativo menor es La menor, al igual que si se toma Reb se sabe que su relativo menor es Sib menor.

Una utilidad importante del círculo de quintas es la de agrupar los acordes de una tonalidad. Si se quisiera saber cuáles son los acordes de, por ejemplo, la tonalidad de Mib mayor, el círculo de quintas agrupará a su alrededor los correspondientes acordes. Como se puede ver en la **Figura 18**.



Figura 18. Círculo de quintas con los acordes de la escala de Mib mayor seleccionados.

Nota: Figura extraída de Musicca (s/f).

En la **Figura 19** se seleccionó una tonalidad mayor de Mib (en color naranja) y se seleccionaron los acordes pertenecientes a dicha tonalidad, por un lado los acordes que se encuentran alrededor (en color morado) y por otro lado un acorde sobre la esquina inferior derecho de dicha selección (en color amarillo), este acorde inferior derecho siempre será disminuido. Dado que la selección de acordes será la misma si se toma la tonalidad de Dom, se entiende que las escalas de Mib y Dom contienen los mismos acordes; es por ello que se les llama relativos. Siendo los acordes elegidos: Mib – Lab – Sib – Fam – Dom – Solm, o en notación inglesa: Eb – Ab – Bb – Fm – Cm – Gm.



Figura 19. Círculo de quintas con los acordes de la escala de Mib mayor seleccionados.

Nota: Figura extraída de Musicca (s/f).

Se nota en la **Figura 19** que la selección de acordes de una tonalidad sigue la misma lógica en una tonalidad menor que en una mayor. Siendo los acordes seleccionados: Sim – Mim – Fa#m – Sol – Re – La, o en notación inglesa: Bm – Em – F#m – G – D – A.

6.1.d Reglas de composición

Con base en lo revisado a lo largo de la sección **6.1 Teoría musical**, a continuación se enlistan hechos que definen patrones presentes en las composiciones musicales. Lo que denota la cualidad de música agradable al oído está dado por estos patrones y cualidades que los autores señalan como presentes en la música tonal contemporánea, y que se pueden interpretar como reglas, que de ser aplicadas en la composición, garantizarían a una pieza su pertenencia y similitud a composiciones populares reales.

6.1.d.1 Para armonía

1. Según la Teoría Tonal (consultar **6.1.c.i Música tonal**), y según lo descrito por (Ravera, 2022), se sabe que una pieza de la música tonal posee una *Tonalidad*, lo que implica que dicha pieza utilizará casi exclusivamente las nota de la *Escala* de dicha tonalidad, y a su vez esto generará una serie de acordes (consultar **6.1.c.ii Círculo de quintas**). La pieza, según la teoría, utilizará casi exclusivamente los acordes dados por dicha tonalidad, por lo que se crea el hecho de que, si una pieza utiliza sólo los acordes de su escala, se garantiza que el resultado armónico será fiel a los ejemplos reales de la música tonal.
2. Existen patrones comunes al construir progresiones tonales, como es la presencia de *Cadencias* y todas sus variantes (Dionisio de Pedro, 2008b). Al existir estas en una progresión, se garantiza que la misma tendrá un carácter resolutivo (consultar **6.1.b.7 Cadencia**), es decir, transmitirá una sensación de *Tensión-Resolución* a la escucha (consultar **6.1.b.5 Reposo** y **6.1.b.6 Tensión**), y por ende, esto dará estética a la escucha y mostrará signos de intencionalidad.
3. Dado que la música tonal da jerarquía a un acorde llamado *Tónica* (consultar **6.1.b.4 Tonalidad**), se sabe que cualquier progresión de acordes que finalice o inicie con el acorde de tónica dará la sensación de *Resolución* (Piston et al., 1987), entonces, la presencia de esta cualidad en una armonía garantizará que la misma se sienta parte de una armonía común de la música tonal.
4. Dado que los acordes tienen una *función*, según la tonalidad (consultar **6.1.b.4 Tonalidad – Funciones tonales**), la manera en que se agrupen definirá carácter de una armonía, conociendo hechos como que los *Grados Atractivos* (VII, II, IV) generarán *tensión* y movimiento hacia los *Grados Tonales*, por lo que al ser colocados previos a los *Grados Tonales* (I, V, VI), se producirá una sensación de dirección intencionada hacia el *reposo*. Del mismo modo los *Grados Modales* (III, VI) aportarán color la *reposo* (Piston et al., 1987).
5. Si un acorde de *Dominante* es complementado con su séptima menor (séptima nota de la escala construida sobre ese acorde si es una escala mayor o menor, y por ende la que está un tono por debajo de la nota raíz del acorde), entonces se construirá un acorde denotado como V7 y llamado acorde de séptima de dominante. Este acorde genera más tensión que direcciona al primer grado, forzando a querer escuchar enseguida el acorde de tónica (Piston et al., 1987).
6. Se le llama *Dominante secundario* o *Dominante auxiliar* al recurso estético de utilizar un acorde (llamado igualmente *dominante secundario*) para transicionar a otro, en donde este acorde de transición es el *Dominante* del siguiente acorde y no necesariamente de la tonalidad (Piston et al., 1987). Este acorde de dominante es siempre mayor. Por ejemplo, si se tiene la siguiente progresión en la tonalidad de Do mayor (C): C – Em – F – G (Do mayor – Mi menor – Fa mayor – Sol mayor), podrían agregarse dominantes secundarios previos al Em y el G, agregando para cada uno su acorde de Dominante correspondiente (es decir, el acorde de grado V si se considerara dicho acorde a transicionar como la tónica). En este caso siendo B (Si) el Dominante de Em y siendo D (Re) el Dominante de G; siendo alterada la progresión como: C – B – Em – F – D – G. Si bien los acordes B y D no pertenecen a la tonalidad de C, estos generan una sensación agradable al escucha, pues generan tensión que se espera resolver al sonar el siguiente acorde (al que se planeaba transicionar). Si a un Dominante secundario se le agrega su séptima, el efecto de tensión es mayor; por ejemplo: C – B7 – Em – F – D7 – G, aplicando el principio antes descrito en el punto 6.
7. El llamado *Direccionamiento de voces* (Schoenberg, 1983) se refiere al movimiento coherente y perceptible de las notas que componen una progresión de acordes, particularmente cuando dichas notas —ya sea el bajo, la voz superior u otras voces internas— siguen una dirección definida, ascendente o descendente. Schoenberg dedica todo el capítulo “VI. CONNECTION OF CHORDS THAT HAVE TO COMMON TONE (HARMONISCHES BAND)” (por su traducción “VI. CONEXIÓN DE ACORDES QUE TIENEN UN TONO COMÚN (BANDA ARMONÍCA)”) a explicar cómo este recurso contribuye a la estética de una composición. La idea central es destacar que, más allá de la sucesión armónica en sí, el movimiento de las voces dentro de los acordes puede crear un sentido de dirección, tensión y resolución. Por ejemplo, se habla de *direccionamiento de bajo* cuando se observa que la nota más grave de cada acorde (la voz del bajo) en una progresión,

forma una línea melódica con dirección clara, ya sea descendente, ascendente o con algún contorno definido, lo que aporta coherencia estructural y expresividad musical, lo que también puede ser aplicado a la *voz principal* (la nota más alta de cada acorde), y también a las demás notas del acorde.

8. Otro patrón estético mencionado por Schoenberg (1983) en *“Theory of Harmony”* (*“Teoría de la Armonía”*) es el de los denominados *“Common Tones”*, o en español, *“Tonos Comunes”*, que se manifiestan cuando los acordes en una progresión comparten notas en común. Este recurso aporta estética a una armonía. Por ejemplo, un acorde de F, que contiene las notas F, A, C, sonará bien adyacente a un acorde de Am (A, C, E), pues contiene 2 tonos comunes (A y C). Este patrón de estar presente (e incluso complementarse con el direccionamiento de voces) nos podría garantizar que una armonía será estética a la escucha.

6.1.d.2 Para melodía

1. Toda la Teoría Tonal aplicada a la música tonal contemporánea nos explica que una pieza musical en cierta *Tonalidad/Escala* usa casi exclusivamente una selección de tonos. Por lo cual, incluso aunque la música tonal real no se limita sólo a las notas de la escala, se sabe que de utilizar únicamente estas, el resultado resultará coherente a la estética tonal (consultar **6.1.b.4 Tonalidad**).
2. Existe una jerarquía para la selección de las notas de una melodía tomando como base su armonía, en la que (según lo descrito en **6.1.c.i Música tonal**) las notas que pertenecen al acorde que suena, son las prioritarias, después, las notas de la tonalidad, y finalmente el resto de las notas (Ravera, 2022). Esto nos indica que de tener la necesidad de componer una melodía sobre una armonía de base, seguir este sesgo de preferencia nos garantizaría un resultado (Krumhansl & Kessler, 1982).
3. El estudio de Krumhansl & Kessler (1982) también respalda la existencia de una jerarquía tonal internalizada en la percepción de los oyentes. En su investigación se tomaron participantes que calificaron la adecuación de cada tono a diversos contextos armónicos, tanto en tonalidades mayores como menores. Los resultados mostraron una preferencia constante por ciertas notas sobre otras, en estrecha consonancia con la estructura de la tonalidad. Estas calificaciones promedio formaron lo que los autores denominaron jerarquías tonales: clasificaciones de tonos basadas en su estabilidad percibida y pertinencia dentro de una tonalidad. Esta jerarquía corrobora el orden teórico de tonos de acordes primero, luego tonos de escala y finalmente tonos cromáticos (el resto de los 12 tonos) y ofrece una base perceptual para la construcción de melodías. La **Figura 20** muestra una representación gráfica de estas jerarquías tonales en *Do mayor* y *Do menor*.

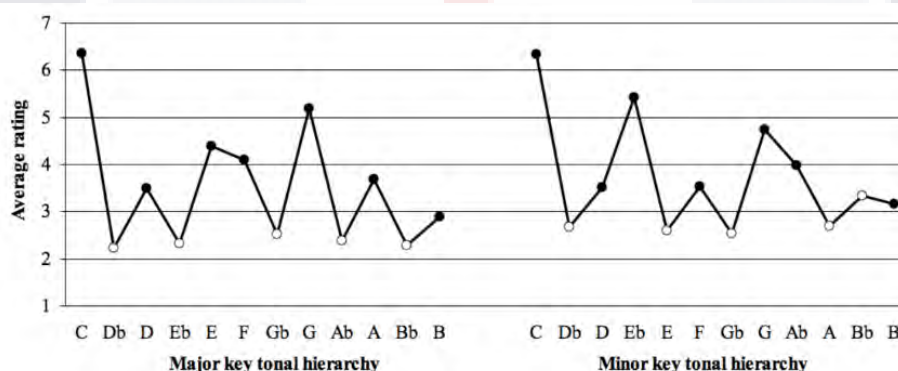


Figura 20. Jerarquía de tonos en *Do mayor* y *Do menor* según el estudio de Krumhansl y Kessler.

Nota: Figura extraída del trabajo de Woolhouse & Cross (2008)¹³.

4. Según lo revisado en **6.1.b.2 Motivo** y de acuerdo con las definiciones de Bribitzer-Stull (2015), The Editors of Encyclopaedia Britannica (2025) y Schoenberg (1967), se entiende el concepto de *motivo* como un eje central en la composición de una melodía memorable para el oyente. El concepto de

¹³ La figura fue tomada del artículo de 2008 de Woolhouse, Matthew y Cross, pero esta es sólo un rediseño del gráfico original de Krumhansl y Kessler's de 1982.

motivo evoca a la repetición de células rítmicas o melódicas, lo que implica que en la construcción de frases melódicas que pretendan ser atractivas a la escucha, es vital la repetición de componentes estructurales de las frases, tomando de un grupo de notas, ya sea sus duraciones, sus tonos o sus intervalos, para ser repetidos y variados.

5. El libro “*Teoría Generativa de la Música Tonal*” (Lerdahl & Jackendoff, 1985), propone un principio importante para la generación de melodías en la música tonal. Este trabajo relaciona el proceso generativo de una melodía con el *Lenguaje Natural*, más específicamente con la *Gramática Generativa* (Chomsky, 1956, 1957). La “*Gramática Generativa Musical*” (GTTM) de Lerdahl y Jackendoff sugiere que las melodías, al igual que las oraciones, pueden construirse a partir de reglas jerárquicas. En la GTTM, la estructura de agrupación segmenta la música en *motivos*, *frases* y *secciones*, y la estructura métrica impone una jerarquía regular de pulsos *fuertes* y *débiles*, de forma directamente análoga a cómo la sintaxis construye cláusulas y frases, y cómo la *prosodia* establece patrones de acentuación en el habla. L&J señalan que la forma musical refleja la prosodia del habla, por lo que el ritmo y la entonación de una oración guían naturalmente el ritmo de una melodía. Al utilizar la *sintaxis* para la estructura de las frases, la *prosodia* para el ritmo y el acento, y el *acento musical* para la colocación métrica, sería posible producir frases melódicas memorables a la escucha.
6. Finalmente, teniendo reglas para la creación rítmica de frases y para la selección de tonos en las notas de la misma, una manera adicional de reflejar intencionalidad en las notas de una melodía tiene que ver con el mismo principio de *direccionamiento de voces* descrito por Schoenberg (1983) en la teoría de la armonía, de manera que un escalamiento ascendente o descendente en el desarrollo de una melodía a través de la selección de sus tonos podría proveer de estética a la misma, al sentirse un camino claro y fluido a través de la escala de la pieza, propiciando incluso en estos patrones de dirección, posibles apariciones de *motivos* en las frases.

6.1.e Idea musical

Se definirá en este trabajo como la abstracción simbólica de una canción, a través de dos elementos: *Melodía* y *Armonía*, de manera que, el algoritmo aquí desarrollado pretende generar piezas musicales abstraídas a modo de *ideas musicales*, es decir, una unión de: Frases formadas por notas y una progresión de acordes que acompañe a dichas frases. Se busca que estas *ideas musicales* representen el potencial para, gracias a la mano del usuario, perfeccionarse, instrumentalizarse y agruparse con otras para formar obras musicales completas.

6.2 Teoría computacional

Ya definido el marco teórico musical, esta sección introduce el segundo enfoque en el marco de este trabajo: la computación. A continuación se presenta toda la teoría necesaria para comprender la lógica computacional detrás de la construcción del algoritmo MIA.

6.2.a Conceptos básicos

6.2.a.1 Algoritmos

“*Algoritmo*” surge como el concepto clave del presente trabajo, y es incluso anterior al primer computador electrónico creado. El concepto *algoritmo* puede ser encontrado en trabajos académicos de hasta los años veinte; por ejemplo, en el artículo “*Sobre soluciones positivas de un sistema de ecuaciones lineales*” (Dines, 1926), en donde se describe un “*procedimiento*” para llegar a soluciones en sistemas de ecuaciones, describiendo el paso a paso de su método y refiriéndose a este como “*algoritmo*”. Si bien Dines habla del concepto de algoritmo, este no lo define; lo da por entendido como un concepto más del Álgebra. Revisando literatura más hacia el pasado, el concepto *Algoritmo* resulta tan longevo como la propia Álgebra.

Al'Khowârizmî fue un astrónomo y matemático y árabe, que en el siglo IX escribió “*Compendio de cálculo por reintegración y comparación*”, un libro en el que describió una serie de procedimientos a los que llamó “*al-jabr*” (del árabe: *Reintegración y Recomposición*), que aplicados a la matemática, exploraban los primeros intentos sobre resolución de sistemas de ecuaciones, en lo que se considera la fundación de lo actualmente conocido como Álgebra (Maher, 1998). Al'Khowârizmî (cuya pronunciación podría ser escrita como “*Al-*

Juarismi”) en el año 820 definió los principios de un nuevo tipo de aritmética que se basaba en métodos constituidos de procedimientos definidos con operaciones matemáticas básicas. Estos procedimientos, denotados tanto como fórmulas matemáticas, así como por instrucciones de operaciones, definieron no sólo el concepto del Álgebra, sino de una completa área de estudio llamada “*Algoritmia*”, cuyo nombre proviene de la pronunciación latina de “*Al-Juarismi*”, es decir, “*Algorithmi*”.

Tanto con su origen en el siglo XI, como en trabajos del siglo XX, *Algoritmo* se entiende como un concepto puramente matemático cuya base es la definición de procedimientos para resolver sistemas de ecuaciones. Sin embargo, es haciendo una revisión de trabajos más recientes (del siglo XXI) cuando se encuentran definiciones más generalizadas al cómputo y a la resolución de problemas en general (entendiendo aquí cómputo como el cálculo numérico y no como procesos de computadores electrónicos). Kenneth Rosen, por ejemplo, escribe en “*Discrete Mathematics and Its Applications*” (“*Matemáticas discretas y sus aplicaciones*” por su traducción al español): “*Un algoritmo es una secuencia finita de instrucciones precisas para realizar un cálculo o resolver un problema*”, en donde “*resolver un problema*” no se limita específicamente al Álgebra. Rosen define también el concepto paralelo “*Pensamiento Algorítmico*”, para el cuál escribe: “*Ciertas clases de problemas se resuelven mediante la especificación de un algoritmo. Tras la descripción de un algoritmo, se puede construir un programa informático que lo implemente*” (2002). Con esto, Rosen sugiere que los algoritmos son adaptables a la computación, mediante programas informáticos que los implementen, evidenciando una potencial conexión con la informática (también llamada computación). El autor Dan Myers, en el libro “*Data Structures and Algorithms in Java*” y su capítulo “*Algorithms + Data Structures = Programs*” hace contundente la relación entre los algoritmos y la informática, pero sin limitarlos a la misma, con la definición: “*Un algoritmo, en términos sencillos, es un procedimiento especificado con precisión para resolver un problema. Los programas informáticos son algoritmos, pero el término podría aplicarse a cualquier serie de pasos que alcancen un objetivo, desde ensamblar muebles modulares hasta hornear un pastel de lima*” (2024).

A pesar de los 1204 años de diferencia entre Myers y Al-Juarismi, y a pesar de las grandes diferencias entre sus culturas (Estadounidense e Iraquí), el concepto de algoritmo es consistente en tres cuestiones: 1. Serie de instrucciones para 2. Resolver un problema, 3. Mediante cálculos y/o procesos. Es de ello que, acorde a la definición del problema de este trabajo (consultar **2. Definición del problema**), se recuerda el propósito de “*formular un algoritmo de composición musical autónoma, que, sin necesidad de piezas musicales reales de entrenamiento, sea capaz de generar ideas musicales originales*”. Con este preámbulo, y haciendo un adelanto de la sección **7. Proyectos relacionados**, a continuación se hace una revisión sobre las técnicas utilizadas históricamente para abordar esta problemática.

6.2.a.2 Modelos

Los algoritmos y los llamados “*Modelos*” comparten similitudes en su definición. Atendiendo a la diferenciación de los conceptos, Alexander Stevenson (2024), Ingeniero de Inteligencia Artificial en Cisco, menciona que a menudo suelen ser mencionados como sinónimos, pero bajo el concepto de *Aprendizaje automático*, son diferentes. Stevenson menciona que un algoritmo es un “*proceso o conjunto de reglas que deben seguirse en los cálculos u otras operaciones de resolución de problemas*”, mientras que un *Modelo* es “*el producto final resultante de la aplicación de un algoritmo a un conjunto de datos*”. También menciona que, mientras el modelo “*contiene los parámetros y la estructura aprendidos que se pueden utilizar para realizar predicciones*”, el algoritmo “*proporciona los pasos necesarios para entrenar un modelo*”. Se han mencionado varios conceptos, como “*parámetros y estructura aprendidos*”, “*entrenamiento*” o “*predicciones*”. Todos estos se comprenden mediante dos conceptos fundamentales: “*Inteligencia Artificial*” y “*Aprendizaje Automático*”.

Alan Turing, considerado el padre de la informática y el pilar de la computación moderna, publica en 1950 el artículo “*Computing Machinery and Intelligence*” (“*Maquinaria informática e inteligencia*”), un trabajo en el que, a pesar de ser escrito durante los inicios de la computación, propone una pregunta vigente a la actualidad: “*¿Pueden pensar las máquinas?*”. Turing, pensando en el futuro de la computación desarrolla todo un cuestionamiento acerca de cómo podría ser identificada la inteligencia en una entidad. Él propuso la “*Prueba de Turing*” (también conocida como “*Test de Turing*”) como una metodología mediante la cual se podría identificar

inteligencia en un sistema computacional, en donde, el parámetro a evaluar es la capacidad de una máquina para exhibir un comportamiento inteligente comparable al de un humano, mediante un interrogatorio en el que un humano, después de interactuar con una máquina y con otro humano (sin conocer cuál de los dos es una persona real) debe concluir quien de ambos es la máquina y quién es el humano. Si la máquina es capaz de engañar al interrogador, entonces se concluye que el sistema *“posee inteligencia”*. Si bien Turing nunca definió el concepto *“Inteligencia Artificial”*, con su trabajo sentó la base de cómo sería esta.

Turing murió en 1954, y no pasó mucho tiempo (concretamente dos años) para que se comenzara a estudiar académicamente el concepto de Inteligencia Artificial. Fue John McCarthy, profesor del *Dartmouth College*, quien en 1955 organizó un taller de verano para aclarar y desarrollar ideas sobre las máquinas pensantes, proyecto para el cual eligió de nombre: *“Inteligencia Artificial”* (Lawrence Livermore National Laboratory, s/f). En agosto de 1955, después de dos meses de trabajo con diez personas, el artículo *“A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence”* (cuya traducción es *“Una propuesta para el Proyecto de Investigación de Verano de Dartmouth sobre Inteligencia Artificial”*) definió a la Inteligencia Artificial como: *“La ciencia y la ingeniería de crear máquinas inteligentes, especialmente programas informáticos inteligentes”* (McCarthy et al., 2006).

Desde entonces, el concepto *“Inteligencia Artificial”* ha sido definido múltiples veces. Una de las tantas descripciones popularmente aceptadas es de Bartneck et al., quienes en el trabajo *“What is AI?”* (*“Qué es IA”*) le definen como *“El estudio de agentes [inteligentes] que reciben percepciones del entorno y realizan acciones”* (2021). En cambio, autores como Legg & Hutter describen más propiamente la *inteligencia* en sí misma como: *“La capacidad de alcanzar objetivos en una amplia gama de entornos”* (2007). Otras definiciones importantes son de ISO (Organización Nacional de Estandarización), que, en su norma *ISO/IEC TR 24030:2021* describe Inteligencia Artificial como: *“Capacidad de adquirir, procesar, crear y aplicar conocimientos, mantenidos en forma de modelo, para llevar a cabo una o más tareas dadas”* (2021). Igualmente le define como: *“La capacidad de las máquinas para pensar, aprender y adaptarse”* ISO (s/f). Las dos definiciones anteriores nos permiten extraer dos conceptos importantes que componen la Inteligencia Artificial: *Modelo y Aprendizaje*.

El *“Aprendizaje automático”* o, por su nombre en inglés *“Machine Learning”* es otro concepto que nace naturalmente de Inteligencia Artificial, y este implica que para lograr Inteligencia hace falta aprendizaje. Se considera que Arthur Samuel fue el primero en acuñar el término *“Machine Learning”*, cuando en 1959 publicó *“Some Studies in Machine Learning Using the Game of Checkers”* (*“Algunos estudios sobre aprendizaje automático usando el juego de damas”*), en donde escribió: *“La programación de computadoras para que aprendan de la experiencia debería eventualmente eliminar la necesidad de gran parte de este esfuerzo de programación detallado”*. Este texto planta la base del concepto mediante dos puntos calve: *“Aprender de la experiencia”* y *“No necesitar programación detallada”*. Samuel (1959) describió además el aprendizaje de máquina como un proceso con dos enfoques posibles:

1. **Enfoque de Red Neuronal:** *“Aborda la posibilidad de inducir un comportamiento aprendido en una red de conmutación conectada aleatoriamente (o su simulación en una computadora digital) como resultado de una rutina de recompensa y castigo”*. Escribió que este enfoque conduciría a máquinas de aprendizaje de propósito general, utilizando redes de un tamaño comparable incluso a la de los animales, no posibles para la creación de dispositivos prácticos según la capacidad técnica de su propia época.
2. **Enfoque de Equivalentes de Red:** *“Producir el equivalente de una red altamente organizada, diseñada para aprender solo ciertas cosas específicas”*. Sobre este enfoque fue capaz de construir aplicaciones prácticas, aunque con la limitante que estas requerían de reprogramación para cada nuevo tipo de aplicación.

Bajo el trabajo de Samuel, el aprendizaje automático comenzó como área de estudio, y esta creció hasta la actualidad, estableciéndose como un pilar de la Inteligencia Artificial, gracias a; no sólo la aparición de nuevos avances, sino el uso de modelos antiguos que, gracias a la computación, ya tenían posibilidad de ser aplicados. Un ejemplo de esto es la llamada *“Neurona artificial”*; un concepto introducido en 1943 por Warren McCulloch

y Walter Pitts, en “*A logical calculus of the ideas immanent in nervous activity*” (“*Un cálculo lógico de las ideas immanentes a la actividad nerviosa*”). Los autores McCulloch & Pitts describieron un modelo matemático que, inspirado en las neuronas humanas y su respuesta binaria (de todo-o-nada), simularían la lógica proposicional e imitar el comportamiento de operadores lógicos. Los principios que fundamentaron fueron:

1. **Disparo total o nulo:** La neurona es binaria, por lo que se activa completamente o no se activa en absoluto.
2. **Recuento sináptico fijo:** Una neurona se excita sólo cuando un número fijo de sinapsis (excitadoras) se activa, dentro del período de adición latente.
3. **Inhibición absoluta:** Cualquier activación de una sinapsis inhibidora bloquea completamente la neurona en ese instante.
4. **Estructura constante:** La estructura de la red es estática a lo largo del tiempo.
5. **Retardo sináptico únicamente:** El único retraso en el sistema es el tiempo de transmisión sináptica.

Se constituye entonces “*Neurona artificial*” como un objeto matemático de naturaleza binaria que posee una estructura que, gracias a la estimulación con entradas numéricas de datos, puede simular distintos tipos de operadores lógicos (mismos que dan forma a las operaciones en los computadores modernos).

En 1943, cuando el modelo de neurona de McCulloch & Pitts fue propuesto, no existía manera práctica de aplicársele. No fue sino hasta 1957, cuando Frank Rosenblatt, un investigador del Laboratorio Aeronáutico de Cornell, casi dos años después de la formulación del concepto de *Inteligencia Artificial* y casi dos años antes del concepto de *Machine Learning*, publicó “*El perceptrón: un autómata que percibe y reconoce*” (de título original “*The Perceptron: a perceiving and recognizing automaton*”). En esta publicación, Rosenblatt logró simular el modelo de neurona artificial en un computador IBM 704, bajo un modelo que llamó “*Perceptrón*”, que era capaz de “*aprender*” los “*pesos de conexión*” que definen las categorías de patrones a través de la experiencia. Rosenblatt en 1958 publicó “*El perceptrón: Un modelo probabilístico para el almacenamiento y la organización de la información en el cerebro*”, en donde describió todos los principios del modelo de perceptrón, como uno capaz de aprender patrones en grupos de datos. El modelo de Perceptrón es la base principal de los actuales modelos llamados “*Redes Neuronales Artificiales*”¹⁴, cuyo nombre es inspirado en los enfoques de aprendizaje automático propuestos por McCulloch & Pitts.

El modelo de Perceptrón se perfeccionó a lo largo de la década de los sesentas y setentas, conservando tres de sus principios: disparo total o nulo, recuento sináptico fijo e inhibición absoluta, pero agregando nuevas capacidades. Por ejemplo, el recuento sináptico fijo se definió como una función matemática denominada “*Función de activación*”, a través de la cuál no sólo se recibiría la excitación sináptica (que en términos de una Red Neuronal se traducen en las entradas numéricas de los datos a las neuronas), sino que se ajustaría su valor mediante estas funciones de activación (cuyo objetivo es agregar no-linealidad en las salidas). Otra mejora fue lo que Rosenblatt llamaba “*corrección de errores por retropropagación*”, misma que nunca logró desarrollar (1961), y que no fue sino hasta 1976, que Seppo Linnainmaa; aplicando diversos métodos matemáticos y teorías de derivación desarrolladas desde los años cincuenta por Robbins & Monro (1951), Hecht-Nielsen, Bryson (1961), Yu-Chi Ho's, Kelley (1960), (Dreyfus, 1962, 1973), logró concretar un algoritmo que corregía errores en los ajustes del perceptrón, creando lo que hoy es conocido como el algoritmo de “*Retropropagación*” (“*Backpropagation*”), que permite calcular derivadas de los pesos sinápticos del perceptrón para; con respecto a una función de coste en la salida de una Red Neuronal (que mide el error de aprendizaje en la Red) aplicar un principio de *optimización* para ajustar los pesos (sinapsis) de las neuronas, acorde al “*gradiente*”, es decir, el impacto de cada neurona en el error de la red (Linnainmaa, 1976b).

La mejora continua del modelo de Perceptrón dio cabida a los modelos ya bien definidos como *Redes Neuronales Artificiales*. En la actualidad el modelo de Red Neuronal (también llamado *Perceptrón Multicapa* o *Red Neuronal FeedForward*) funciona estructurando distintos perceptrones agrupados en las llamadas “*capas*”.

¹⁴ Las “*Redes Neuronales Artificiales*” también pueden ser identificadas por sus siglas en inglés: NN, al igual que las *Redes Recurrentes (RNN)* y las *Redes Convolucionales (CNN)* o bien por sus siglas en español (RN).

Una Red posee la llamada “*capa de entrada*”, que consiste en una serie de perceptrones cuya entrada es la codificación de las instancias individuales de datos de los cuáles se pretende aprender patrones (el conjunto de datos de entrada, o también llamados anteriormente como *experiencia*). Esta capa, así como las capas aledañas (llamadas “*capas ocultas*”) filtran su salida utilizando la antes mencionada “*función de activación*”, la cual representa la salida de la capa y la entrada de los perceptrones de la siguiente capa. Las salidas de las capas ocultas llegan hasta la denominada “*capa final*”, que activa o desactiva ciertas neuronas para definir una “*predicción*” (es decir, el resultado de la Red respecto a la entrada de datos), siendo este valor numérico de predicción la salida de la Red, y finalmente, utilizando una “*función de coste*” es decir, una función que permita evaluar la diferencia entre el resultado predicho por la Red y el resultado real esperado de los datos (Montesinos López et al., 2022), y posterior a utilizar Retropropagación para calcular el gradiente de los pesos sinápticos de cada capa (derivada con respecto al error), aplica un algoritmo de optimización actualizando sus pesos sinápticos y esperando que ante las siguientes entradas de datos, el error de la Red se haya minimizado.

Tom Mitchell escribe en el libro “*Machine Learning*”: “*Se dice que un programa de computadora aprende de la experiencia E con respecto a alguna clase de tareas T y medida de desempeño P, si su desempeño en las tareas en T, medido por P, mejora con la experiencia E*” (1997). Cuando se habla de experiencia *E*, se habla del conjunto de datos que representan la realidad que se pretende enseñar a la máquina, específicamente la tarea *T* que se busca que imite. La medida de desempeño *P* es la antes mencionada *función de coste*, que numéricamente informa qué tan bien actuó la máquina respecto a lo que se esperaba que imitara. Finalmente, el proceso de aprendizaje es el ciclo en el que, mediante la experiencia *E*, la máquina mejora su desempeño. Por ejemplo, si la tarea que se espera que desempeñe nuestra máquina es componer música, entonces como *experiencia* se necesitará una colección de composiciones musicales. El enfoque de utilizar música para entrenar un modelo de composición musical ha demostrado ser efectivo y valioso (consultar el próximo apartado **7. Proyectos relacionados**) utilizando no sólo las clásicas Redes Neuronales, sino variantes útiles como; por ejemplo, las llamadas “*Redes Neuronales Recurrentes*”, diseñadas por Hopfield (1982) como un modelo especialmente adaptado a datos de naturaleza secuencial y de series de tiempo (como bien lo es la música), o como las “*Redes Neuronales Recurrentes LSTM*”, que son una variante más específica de esta, que introdujo células de memoria controladas para abordar las dependencias de largo alcance (Hochreiter & Schmidhuber, 1997). Otras variantes de Red Neuronal son las “*Redes Neuronales Convolucionales*”, adaptadas específicamente para aprender patrones en imágenes (Fukushima, 1980), o las “*Redes Neuronales Deep Belief*” (Hinton et al., 2006), que se centran en aprender los patrones subyacentes de los datos sin necesidad de estar estos etiquetados con valores que nos permitan conocer si su desempeño y/o predicciones fueron correctos.

Existiendo la posibilidad de abordar el problema de creación de composiciones musicales por una máquina, usando el enfoque de entrenamiento mediante datos de ejemplo, resulta relevante estudiar si también existen modelos capaces de lograrlo sin la necesidad de datos de entrenamiento. Dado que el objetivo de este proyecto es utilizar técnicas no basadas en conjuntos de música real de artistas, entre los métodos posibles aplicables se encuentran los “*Modelos de Markov*”. Más específicamente, las “*Cadenas de Markov*” ofrecen una manera simple de crear secuencias aleatorias, como melodías, siguiendo reglas simples y fijas y sin depender de grandes conjuntos de datos ni parámetros aprendidos. Una Cadena de Markov se define como un conjunto finito de “*estados*”; siendo estos una definición de los posibles escenarios de decisión en la generación de la secuencia, además de un conjunto de probabilidades de transición para los estados posibles. En esencia, cada paso depende únicamente del estado actual y no del historial completo de estados anteriores (Markoff, 1910).

En la composición musical, los modelos de Cadenas de Markov basados en reglas se encuentran entre los primeros enfoques algorítmicos computacionales utilizados. Volviendo al 1957, cuando, a la par que la implementación computacional del Perceptrón de Rosenblatt, Hiller e Isaacson crearon la denominada “*primera partitura compuesta por una computadora electrónica*” (Sandred et al., 2009): la “*Suite Illiac*” para cuarteto de cuerdas, implementando un sistema con Cadenas de Markov. El sistema generaba melodías lanzando “*datos ponderados*” contra tablas de transición de altura (o dicho también “*tono*”: consulte **6.1.a Conceptos básicos**) y ritmo preprogramados. Con este modelo se compusieron otras piezas musicales como el “*Cuarteto de cuerdas estocástico*” (“*Stochastic String Quartet*”) de 1963 (Tenney, 1988). Compositores posteriores, como Theodore

DeLio, ampliaron esta idea creando manualmente matrices de transición para altura (*tono*), duración y dinámica (entiéndase como la variación en la intensidad de interpretación), utilizándolas como reglas de composición en lugar de modelos aprendidos (Ames, 1987, 1990). DeLio compuso “*Serenade*” con este trabajo.

Dado que los sistemas de Cadenas de Markov se basan únicamente en probabilidades especificadas por quien las diseña, evitan la necesidad de grandes conjuntos de datos musicales, lo que les convierte en una alternativa ligera y transparente para la generación de música estocástica (aleatoria). Sin embargo, en este trabajo, las Cadenas de Markov se estudiarán sólo como modelos utilizados en proyectos relacionados. Los algoritmos de interés para ser formulados se encuentran a continuación.

6.2.b Procesamiento de Lenguaje Natural

El concepto “*Procesamiento de Lenguaje Natural*” (NLP por sus siglas en inglés), a pesar de ser catalogado actualmente como parte del estudio de Inteligencia Artificial, tiene un origen incluso previo a la misma (1948). Se considera que la idea que impulsó la primera aplicación informática de NLP (un traductor), fue el “*Modelo de Shannon-Weaver*” (Shannon, 1948). Se menciona que Weaver escribió un memorándum proponiendo utilizar computadoras para traducir lenguajes humanos. Él sugirió técnicas criptográficas que habían sido utilizadas durante la Segunda Guerra Mundial, para construir un sistema de traducción de idiomas, y si bien, el término NLP no fue definido, sentó precedente sobre el uso de computación aplicado al lenguaje. El antes mencionado “*Test de Turing*” también exploraba la posibilidad de una computadora interactuando a manera de texto humano. El memorándum de Weaver inspiró proyectos, como el “*Experimento de IBM-Georgetown*”, un proyecto entre la empresa IBM y la Universidad de Georgetown, que en 1954 demostró exitosamente la traducción de más de 60 oraciones del ruso al inglés (Dostert, 1955; Macdonald, 1954), en el que se codificaron reglas lingüísticas para ambos idiomas. El experimento de IBM-Georgetown demostró, en primero lugar, la posibilidad de codificar reglas lingüísticas computacionalmente, y en segundo lugar, la complejidad de hacerlo, puesto que los resultados de dicho experimento se consideraban deficientes, pues a falta de reglas más completas al lenguaje, existían problemas de traducción, como la ambigüedad léxica.

No fue sino hasta 1956 que Noam Chomsky introdujo una teoría sólida para el tratamiento de reglas del lenguaje aplicadas a contextos matemático-computacionales. Se introdujo el concepto “*Gramática Generativa*”, como un sistema de estructuras que describen la sintaxis de un lenguaje (consultar **6.2.b.3 Gramáticas Generativas**), y a pesar de no tenerse claro cuándo se definió el concepto por primera vez, se sabe que es a partir del trabajo de Chomsky, que se le considera al “*Procesamiento de Lenguaje Natural*” como un área de estudio enfocada en el lenguaje humano (lenguaje natural) como objeto de modelación computacional y matemática. El NLP estudia no sólo la implementación del lenguaje natural dentro de contextos computacionales, sino también cómo aplicar sus reglas lingüísticas (Chomsky, 1956, 1957).

6.2.b.1 Lingüística

La Real Academia Española (RAE) define la *lingüística general* como el “*estudio teórico del lenguaje que se ocupa de métodos de investigación y de cuestiones comunes a las diversas lenguas*” (2001). A pesar de ser esta una definición simple, abarca tanto el análisis de las lenguas particulares (su estructura, historia, variación, etc.) como las propiedades generales del lenguaje (sintaxis, morfología, fonética y fonología)¹⁵.

Para el desarrollo de este trabajo, los conceptos de lenguaje y la lingüística necesarios son:

- **Oración:** En lingüística, una “*oración*” es la unidad básica que expresa un significado completo. Formalmente, la RAE define la oración como una “*estructura de predicación en la que se pone en relación un sujeto con un predicado*” (2001). Esto significa que toda oración reúne un sujeto (explícito o implícito) y un predicado cuyo núcleo suele ser un verbo conjugado. Las oraciones se clasifican de varias formas. Según el número de predicados, se distinguen las oraciones “*simples*”

¹⁵ La lingüística tiene distintas variantes. En el contexto de este trabajo, se hablará de lingüística refiriendo exclusivamente a la lingüística general.

(un solo predicado/verbo) y las “*compuestas*” (dos o más predicados). Por su modalidad las oraciones pueden ser enunciativas (afirmativas o negativas), interrogativas, exclamativas o imperativas etc. (Bosque & Demonte, 1999).

- La oración entonces puede ser definida como una unidad finita compuesta por palabras, que expresa una idea. Se recuerda como Schoenberg (1967) define en un contexto musical la “*frase*” y la relaciona al concepto de “*oración*” (consultar **6.1.b.1 Melodía**), describiéndole como “*una unidad aproximada a la que podría cantarse en una simple respiración. Su final sugiere una forma de puntuación tal como una coma*”. Es esta relación la que se considera relevante, pues, en la construcción de melodías musicales (compuestas por frases), se utilizarán oraciones como equivalencia.
- **Palabra:** Es el constituyente principal de una oración. La palabra como “*Sonido o conjunto de sonidos articulados que expresan una idea*” (Real Academia Española, 2001). Esta se puede categorizar en distintos tipos según la idea que represente, y a estos tipos se les llama “*Categoría Gramatical*”. Según Bosque (2015), las categorías gramaticales en el español son:
 - **Sustantivos:** Palabras que nombran personas, animales, lugares o conceptos.
 - **Adjetivos:** Palabras que describen o modifican las cualidades y características de los sustantivos.
 - **Verbos:** Palabras que indican acciones, estados o procesos.
 - **Adverbios:** Palabras que modifican al verbo o al adverbio, indicando cuándo, dónde, cómo y en qué medida.
 - **Pronombres:** Palabras que reemplazan a los sustantivos.
 - **Preposiciones:** Palabras que estructuran relaciones entre palabras o grupos de palabras.
 - **Conjunciones:** Palabras que conectan palabras u oraciones.
 - **Interjecciones:** Palabras que expresan sentimientos o emociones.
 - **Determinantes:** Palabras que acompañan al sustantivo para indicar su cantidad, género, número o tipo.
- **Sintaxis:** Los autores Bosque & Gutiérrez-Rexach, en “*Fundamentos de sintaxis formal*” definen: “*La sintaxis es la parte de la gramática que estudia la forma en que se combinan las palabras*” (2020). Bajo el objetivo de generar oraciones (para hacerles equivalencias a frases melódicas), entender y modelar la sintaxis de los distintos tipos de oraciones, es una manera de implementarlo, y será próximamente analizado en **2.6.b.3 Gramáticas Generativas**.
- **Sílaba:** El diccionario de la Real Academia Española describe “*palabra*” como un “*conjunto de sonidos*”, al igual que define “*sílaba*” como: “*Sonido o sonidos articulados que constituyen un solo núcleo fónico*” (2001). La palabra es, como se describió antes, la que al agruparse con otras palabras (sintaxis), constituye una oración, y a su vez, lo que constituye una palabra es la sílaba, siendo una agrupación de sílabas lo que la forma.
- **Sílaba prosódica (sílaba tónica):** Definido como la sílaba “*que tiene el acento prosódico*” (Real Academia Española, 2001), y definiendo el acento prosódico como el “*relieve que en la pronunciación se da a una sílaba de la palabra, distinguiéndola de las demás por una mayor intensidad o por un tono más alto*”. Cada palabra en el español contiene exactamente una sílaba tónica, siendo esta la que distingue la intensidad con la que se le pronuncia. Esta definición también nos permite relacionar la sílaba tónica con las notas en la melodía musical, pues en una melodía, ciertas notas específicas nos pueden indicar un relieve en la misma. Las sílabas *átonas* son aquellas sílabas que no son tónicas.
- **Sinalefa:** Según la Real Academia Española: “*Enlace de sílabas por el cual se forma una sola de la última de un vocablo y de la primera del siguiente, cuando aquel acaba en vocal y este empieza con vocal, precedida o no de h muda*”. La sinalefa es entonces el caso en el que dos sílabas separadas se agrupan como una sola, siendo estas de dos palabras distintas, ocurriendo cuando la terminación de al primer y la iniciación de la segunda, es igualmente una vocal (o una vocal y una *h muda*).

6.2.b.2 Prosodia

El Centro Virtual Cervantes define “*prosodia*” como “*el conjunto de fenómenos fónicos que abarcan más de un fonema o segmento -entonación, acentuación, ritmo, velocidad de habla, etc.-*” (s/f), aunque se le puede entender no sólo como los fenómenos, sino el área de estudio de los mismos. La prosodia estudia el ritmo; un concepto antes mencionado en el apartado de teoría musical (6.1.a Conceptos básicos), y que refiere a los patrones en las duraciones de las notas, sin embargo, el ritmo es aplicable también en el lenguaje, y este refiere al patrón temporal de sílabas acentuadas y no acentuadas en las oraciones.

En música, como se mencionó anteriormente en 6.1.a Conceptos básicos, existe el concepto de “*compás*”, siendo este una manera en la que se agrupa el tiempo en una pieza musical. El compás es una división del tiempo, a su vez que este compás es dividido por “*Tiempos*” o “*Pulsos*”. La teoría de la acentuación en música indica que estos pulsos en el compás no tienen la misma importancia acentual, sino que se categorizan como Tiempos: “*fuertes*”, “*semi-fuertes*” o “*débiles*”. Es gracias a esta categorización, que es posible establecer reglas de ritmo en las sílabas según la acentuación del compás y la acentuación de las sílabas (Pattison, 1991):

1. Las sílabas tónicas deben coincidir con los tiempos fuertes del compás.
2. Las sílabas átonas deben ubicarse en los tiempos débiles del compás.
3. Se debe evitar colocar sílabas tónicas en tiempos débiles, a menos que se busque un efecto expresivo específico. Ubicar sílabas acentuadas en tiempos débiles puede generar una sensación de desplazamiento rítmico o énfasis no natural¹⁶.

En un todo, se debe mantener la coherencia entre la acentuación lingüística y la estructura métrica musical cuando se quiere incorporar el texto a la música. La alineación adecuada entre los acentos del texto y los acentos del compás contribuye a una interpretación más natural y comprensible de la música vocal (Pattison, 1991). Considerando lo antes mencionado, resulta interesante que según Pattison & Welch, autores de “*Writing Better Lyrics*” (“*Escribiendo mejores letras*”), la *prosodia* se define cuando, tanto los elementos lingüísticos como los elementos musicales se combinan para justificar el mensaje de una canción, su ritmo y sus emociones; una definición que interseca la lingüística y la música (2009).

6.2.b.3 Gramáticas Generativas

La “*Gramática Generativa*”, introducida por Noam Chomsky en 1956, considera a un “*lenguaje*” como un conjunto (finito o infinito) de oraciones construidas a partir de un alfabeto finito. En el trabajo “*Three models for the description of language*”, por su traducción: “*Tres modelos para la descripción del lenguaje*”, define la “*gramática*” como un mecanismo formal que “*produce todas las cadenas que son oraciones de L y solo estas*” (refiriéndose en este caso a L como el lenguaje). En este caso, una “*gramática generativa*” es un sistema finito de reglas (también llamadas “*reglas de producción*”) que enumera con precisión las oraciones gramaticales de una lengua. El objetivo principal es distinguir las oraciones bien formadas de las secuencias agramaticales mediante reglas, de modo que la gramática “*genere todas las secuencias gramaticales de L y ninguna de las agramaticales*” (Chomsky, 1957). Desde esta perspectiva, la gramática puede modelar una lengua completa utilizando una descripción finita de sus reglas de producción; lo que implicaría representar en ella, su sintaxis.

En su trabajo “*Syntactic Structures*”, o por su traducción, “*Estructuras Sintácticas*” (1957), Chomsky describe las “*reglas de producción*” (también mencionadas como “*reglas de reescritura*”). En una Gramática Generativa, cada regla de producción especifica cómo una categoría sintáctica de alto nivel puede expandirse en una secuencia de categorías o palabras de nivel inferior. Las “*categorías sintácticas*” (también llamados

¹⁶ En la música existe la “*síncopa*”, un recurso rítmico que consiste en desplazar el acento del ritmo de los tiempos fuertes a los tiempos débiles, creando un efecto de movimiento y sorpresa, lo que la vuelve dinámica y atractiva. La síncopa está presente principalmente en géneros latinos como la salsa, el tango, la milonga y el choro brasileño (Fernández, 2021).

“símbolos no-terminales”) son representaciones textuales de elementos gramaticales. Un ejemplo de representación podría ser:

- **S:** Representa a una oración.
- **SN:** Representa el sintagma nominal (es decir, el sujeto u objeto directo de la oración).
- **SV:** Representa el sintagma verbal (es decir, el predicado de la oración).
- **Det:** Representa una palabra de la categoría gramatical “*determinante*”.
- **V:** Representa una palabra de la categoría gramatical “*verbo*”¹⁷.

Habiendo definido los símbolos no-terminales, las reglas de producción se construyen como instrucciones de *reescritura* de símbolos no-terminales, por otros símbolos terminales (o como se verá a continuación, con *símbolos terminales*). Por ejemplo, la regla “ $SV \rightarrow V SN$ ” puede interpretarse como: SV (que aquí representa al *sintagma nominal*) se reescribe como V (un *verbo*) seguido de SN (un *sintagma nominal*). En general, una regla de la forma “ $A \rightarrow B C \dots$ ” significa que la categoría “A” se expande en la secuencia “B C ...”, lo que define que “B C ...” es una estructura constituyente legítima de tipo “A”. Por lo tanto, cada producción nos indica cómo se construyen constituyentes mayores a partir de constituyentes menores. Finalmente, los *símbolos terminales* son aquellos que están en el nivel más bajo de las producciones; es decir, las palabras finales sobre las cuales no se puede reescribir más.

Agrupando distintas reglas de producción, se consigue describir la sintaxis de un lenguaje o de cierto tipo específico de oraciones del mismo. Por ejemplo:

1. $S \rightarrow SN SV$.
2. $SN \rightarrow Det Sus$.
3. $SV \rightarrow V SN$.
4. $Det \rightarrow “el” \mid “un”$.
5. $Sus \rightarrow “gato” \mid “perro”$.
6. $V \rightarrow “persiguió” \mid “vio”$.

Donde las interpretaciones de las reglas son:

1. Una oración S se construye de *sintagma nominal* (SN) y *sintagma verbal* (SV).
2. Un *sintagma nominal* (SN) se construye de *determinante* (Det) y *sustantivo* (Sus).
3. Un *sintagma verbal* (SV) se construye con un *verbo* (V) y un *sintagma nominal* (SN).
4. Un *determinante* (Det) se construye con “el” o bien con “un”.
5. Un *sustantivo* (Sus) se construye con “gato” o bien con “perro”.
6. Un *verbo* (V) se construye con “persiguió” o bien con “vio”.

Las reglas de producción implican que, cualquier oración posible a formarse con sus reglas de producción, pertenece al lenguaje, de la misma manera en que, siguiendo reglas de producción aleatorias, se pueden construir oraciones aleatorias válidas para el lenguaje descrito. Por ejemplo, una oración posible con las anteriores reglas de producción sería “*el perro persiguió un gato*”.

El símbolo “ $\rightarrow$ ” se utiliza como estándar y significa “*se reescribe como*”, y el símbolo “ $\mid$ ” en las reglas de producción implica que, o bien puede ser uno u otro. Por ejemplo, una regla escrita como “ $SN \rightarrow Sus Adj \mid Det Sus$ ” es equivalente a las reglas individuales “ $SN \rightarrow Sus Adj$ ” y “ $SN \rightarrow Det Sus$ ”. Es decir, el símbolo “ $\mid$ ” nos permite colocar en una sola línea todos los valores posibles de producción (Chomsky, 1957).

¹⁷ La elección de símbolo para nombrar un no-terminal se hace a criterio del modelador, con la única condición de que sea un texto sin espacios, pues los espacios indicarán el nombramiento de otro terminal.

Cada derivación (como se le llama a las maneras de describir las posibles producciones en reglas) podría visualizarse como un “árbol sintáctico”¹⁸ que explicita la estructura sintáctica de la oración. En un “árbol sintáctico”, cada nodo está etiquetado con una categoría sintáctica o no-terminal, y las ramas muestran cómo se expanden las categorías. Por ejemplo, las reglas anteriores producirían un árbol con la raíz “S” ramificándose en “SN” y “SV”, luego “SN” ramificándose en “Det” y “Sus”, y “SV” ramificándose en “V” y “SN”, etc. Visualmente, este árbol mostraría cómo se construye la oración a partir de constituyentes más pequeños.

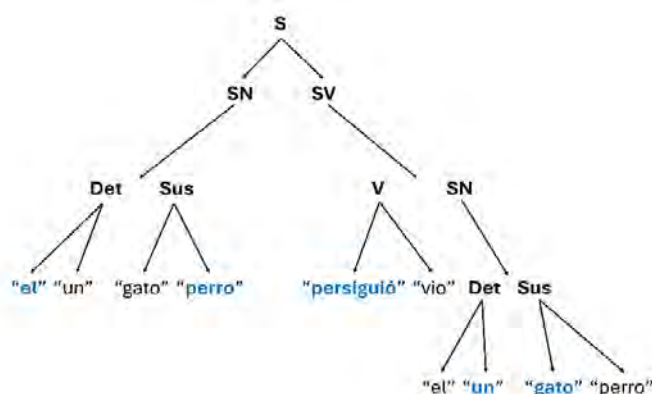


Figura 21. Representación de árbol de una gramática generativa.

La **Figura 21** muestra una representación visual de la Gramática Generativa definida anteriormente mediante reglas de producción (que es la manera en que se definen), al igual que se señala en color azul, cómo al seguir un camino aleatorio hasta el nivel más bajo de cada rama se consigue una oración aleatoria con sintaxis correcta; en este caso, la antes mencionada: “el perro persiguió un gato”.

Este formalismo de convertir símbolos abstractos en oraciones concretas mediante la reescritura jerárquica fue la idea central de los primeros trabajos de Chomsky. Las gramáticas generativas hacen que la sintaxis sea explícita y computacional: a partir de “S” (“sentence”), se reescriben repetidamente los *no-terminales* mediante reglas, hasta que solo quedan los *terminales* (palabras). El árbol sintáctico resultante por lo tanto, incorpora los pasos de derivación y muestra cómo cada elemento de la oración encaja en una estructura mayor. Este modelo además puede ser perfectamente aplicado al objetivo del proyecto: la generación de oraciones aleatorias, habiendo la necesidad de definir las reglas de producción de las oraciones a generar, para lograrlo.

La Gramática Generativa se ha aplicado ya a la teoría musical, desde 1985, cuando se publicó “Teoría Generativa de la Música Tonal” de Lerdahl & Jackendoff. En esta teoría se planteó la relación entre la estructura de las oraciones del lenguaje, con las melodías musicales (este trabajo también se referencia en **7 Proyectos relacionados - 7.3 Composición basada en frases**).

6.2.c Cómputo Evolutivo

Para comprender el concepto “Cómputo Evolutivo” es preciso introducir primero los llamados “Problemas de Optimización”. Los autores Dagdia & Mirchev definen: “Un problema de optimización se refiere a la tarea de encontrar la mejor solución de un conjunto de soluciones factibles que satisfacen todas las restricciones definidas por el problema. Implica identificar los valores óptimos para las variables de decisión con el fin de minimizar o maximizar una función de aptitud, teniendo en cuenta también cualquier restricción especificada” (2020). Esta definición enfoca algunos de los términos más relevantes en la optimización: “función de aptitud”, “restricciones”, “soluciones factibles”, “mejor solución”. En suma, Tsai & Chiang le definen como: “Encontrar la mejor solución entre un gran número de posibles soluciones en un espacio de soluciones dado. Una forma de resolver un problema de optimización es, primero, encontrar buenas soluciones; luego, estimarlas; y, finalmente,

¹⁸ Entendiendo “árbol” como la estructura computacional que define elementos (llamados “nodos”) de los cuales se desprenden líneas que llevan a otros nodos.

usar la información así obtenida para decidir entre buscar otras posibles soluciones” (2023). Con base a estas redacciones y enfocando directamente en la *optimización*, se puede decir que esta se trata de un área conceptual de estudio con interés en los problemas cuya naturaleza parte de tener un “*conjunto de soluciones factibles*”, para los cuales usualmente no resulta factible (incluso computacionalmente) buscar entre todas las candidatas, y por ello su enfoque utiliza “*funciones de aptitud*” como mecanismo para explorar entre las posibles soluciones, siendo esta función una manera de evaluar la efectividad de cada una para dirigir la búsqueda hacia *soluciones mejores*.

Uno de los principios más importantes en los problemas de optimización es que, a pesar de no poseer métodos fijos que lleguen siempre a la mejor solución, sí se posee una manera de medir qué tan buena es una solución de su espacio de búsqueda (se les puede llamar igualmente *funciones de coste*, *funciones objetivo* o *funciones de aptitud*), así que los métodos para optimizar intentan llegar a la mejor solución posible con base a la medición dada por la *función objetivo*. Otro principio indispensable es que exista una manera de representar simbólicamente cualquier solución del espacio de búsqueda. Por ejemplo, si el problema de optimización fuese encontrar la ruta más rápida entre dos lugares en una ciudad, un mecanismo para representar las posibles soluciones sería una secuencia de calles a transitar para llegar del primer lugar al segundo, de manera que gracias a esta representación es posible identificar cada posible solución del espacio de búsqueda; es decir, todas las posibles combinaciones de calles que lleven del primer al segundo lugar se pueden identificar con esta representación, permitiendo además que para estos problemas sea incluso factible generar soluciones aleatoriamente, creando, por ejemplo, combinaciones de calles, en un intento por tener soluciones únicas obtenidas de manera sencilla, por las cuáles comenzar a buscar, recordando la estrategia de Tsai & Chiang: “*primero, encontrar buenas soluciones; luego, estimarlas; y, finalmente, usar la información así obtenida para decidir entre buscar otras posibles soluciones*”, siempre que cumplan con la *restricción* de llevar del primer lugar al segundo. En este mismo ejemplo, una posible *función objetivo* puede ser la *distancia recorrida*, de modo que si queremos encontrar la ruta más rápida entre dos lugares, la distancia nos dirá qué tan buena fue la ruta elegida.

Dentro de los métodos para abordar problemas de optimización, se encuentra el *Cómputo Evolutivo*, que, según Gao, es “*un modelo de inteligencia computacional que imita los procesos de evolución biológica mediante algoritmos como el algoritmo genético, la programación evolutiva, las estrategias evolutivas y la programación genética*” (2018). Gao agrega igualmente, que este “*implica la creación de soluciones iniciales, la evaluación de sus propiedades, la selección de soluciones con base en los resultados de la evaluación, la realización de operaciones evolutivas y la obtención de la siguiente generación de soluciones hasta que se cumplan los requisitos*”. Por lo que, la Computación Evolutiva comprende el desarrollo y definición de una serie de métodos que, imitando la evolución como mecanismo de mejoramiento, funcionan creando una “*población*” inicial de soluciones, sometiéndolas a operadores evolutivos a través de “*generaciones*” y mejorando su rendimiento mediante una “*función de aptitud*”, para resolver problemas de optimización. El *Cómputo Evolutivo* nace como una manera de abordar estos problemas, que por su naturaleza podrían representar retos enormes para otras técnicas de solución de problemas.

6.2.c.1 Algoritmo Genético

El “*Algoritmo Genético*” (AG o “GA” por sus siglas del inglés) surge como una técnica del *Cómputo Evolutivo*, cuyo fin es encontrar soluciones óptimas en problemas para los cuáles existe un número enorme de posibles soluciones y cuya inspiración de solución es el fenómeno natural de la evolución, que explica que con el paso de las generaciones, los individuos de una especie mejoran sus características en función de la adaptación a su entorno. El Algoritmo Genético fue propuesto originalmente en 1975 por John Henry Holland, en su trabajo “*Adaptation in Natural and Artificial Systems: An introductory analysis with applications to biology, control, and Artificial Intelligence*”. Su trabajo original lo ha ampliado en diversos trabajos posteriores (Holland, 1984, 1992a, 1992b), por ejemplo, en “*Algoritmos Genéticos: Programas informáticos que evolucionan de maneras que se asemejan a la selección natural pueden resolver problemas complejos que incluso sus creadores no comprenden del todo*” (1992b).

El Algoritmo Genético comienza generando una “*población inicial*” de soluciones, codificadas numérica o simbólicamente acorde al tipo de problema, esta población de soluciones puede ser total o parcialmente aleatoria

y su codificación, también llamada “*mecanismo de representación*” debe describir computacionalmente las características de las posibles soluciones. Un mecanismo común (que propone Holland) es una cadena de dígitos binarios (es decir, 0’s y 1’s). Holland explica que “*se parte de una población de cadenas aleatorias de unos y ceros y se califica cada cadena según la calidad de su resultado*” (1992b). En la práctica, cada cadena de números representa una posible solución, que se evalúa mediante la “*función de aptitud*” específica del problema (por ejemplo, beneficio, precisión o recompensa, costo). Esta *función de aptitud* debe ser capaz de, dado un *individuo* cualquiera de la población, medir cuál es el valor numérico del objetivo que se plantea minimizar o maximizar, de manera que el paso siguiente a la creación de la *población inicial* aleatoria (en donde la cantidad de individuos creados es un parámetro que decide quien implementa el algoritmo) es asignarle a cada miembro de esta su respectivo valor de aptitud, recordando que dicho valor de aptitud revela la eficacia para resolver el problema planteado. Esto guiará los siguientes pasos del algoritmo, pues aunque en este primer paso las soluciones son aleatorias y sus aptitudes probablemente no sean buenas, con el uso de los operadores genéticos se lograrán mejorarlas (Holland, 1992b).

Enseguida a la creación y evaluación de una población inicial, viene la selección de “*progenitores*”. Tras la evaluación, el algoritmo “*selecciona*” individuos para “*reproducir*”, favoreciendo a aquellos con mayor aptitud. En palabras de Holland: “*las cadenas de alta calidad se aparean; las de baja calidad se pierden*”. En otras palabras, las cadenas con mejor aptitud tienen más probabilidades de ser elegidas como progenitores. Un método común (como sugiere el autor) es la “*selección proporcional a la aptitud*”, para el que la probabilidad de elegir una cadena es proporcional a su aptitud. Este sesgo de selección significa que, a lo largo de las generaciones, las soluciones con buena puntuación tienden a conservarse. De hecho, el autor señala que a medida que pasan las generaciones, predominarán las cadenas asociadas con soluciones mejoradas, lo que refleja cómo los individuos con mayor aptitud dominan gradualmente la población (Holland, 1992a).

Posterior a la selección de progenitores, viene el paso de “*cruzamiento*” (“*crossover*” por su término en inglés). Las cadenas progenitoras seleccionadas se “*aparean*” para producir descendencia. Este “*apareamiento*” consiste en, tomando dos individuos entre los progenitores seleccionados, se corta aleatoriamente un segmento de la codificación del primer individuo, y un segmento del segundo, combinándose para generar una nueva solución; es decir, un *nuevo individuo*. Según Holland, el “*cruzamiento*” es “*el mecanismo fundamental del reordenamiento genético tanto para organismos reales como para algoritmos genéticos*”. Este paso garantiza que de los mejores individuos de una población (los seleccionados como progenitores) sus características (o al menos parte de estas) se pasarán a las siguientes generaciones; de manera que los individuos generados en el cruzamiento heredan parte de las características de los individuos que lo generaron.

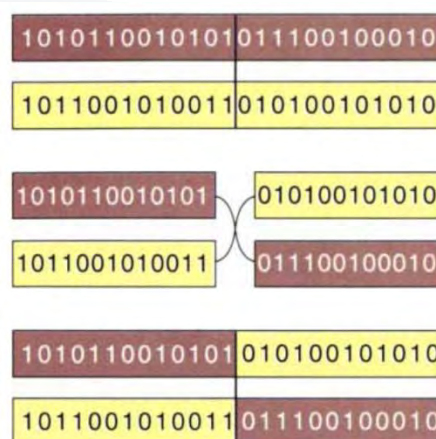


Figura 22. Visualización del “cruzamiento” de dos individuos en el Algoritmo Genético.

Nota: Figura extraída del trabajo de Holland (1992b).

La **Figura 22** muestra visualmente a dos individuos de codificación de 0's y 1's (uno en color rojo y otro en color amarillo), que al cortar sus cadenas y mezclarse, forman dos nuevos individuos, cuyas cadenas son combinaciones de rojo y amarillo.

La “*Mutación*” es el operador genético siguiente a la generación de un nuevo individuo. Tras el cruce, se decide, para cada individuo generado (con una pequeña probabilidad, designada por el implementador del algoritmo) si se desea que el individuo *mute*. Si es el caso, se toma aleatoriamente un segmento de la codificación del individuo y se intercambia su valor. Por ejemplo, en el caso en que la codificación de un individuo fuera una cadena de 0's y 1's, una posible mutación aplicada a un dígito de la cadena al azar es intercambiarlo; si es 0, cambiarlo por 1, si es 1, cambiarlo por 0. La mutación introduce nuevas variaciones aleatorias en la población, como un mecanismo para explorar más alternativas en el espacio posible, y se encarga de no estancar a la población en sólo ciertas características, lo que en términos técnicos se refiere a mantener la diversidad genética y ayudar a evitar la convergencia prematura (Holland, 1992a).

Ya teniendo los nuevos individuos generados (tanto los que se mutaron como los que no), el siguiente paso es la “*formación de la siguiente generación*”. La descendencia producida constituye la *población de la siguiente generación*. Normalmente, toda la nueva descendencia reemplaza a la población anterior (modelo generacional) (1992a), y en otras ocasiones es posible conservar algunos individuos de la población previa al cruzamiento, como una técnica denominada “*elitismo*”, que permitiría que, si durante alguna generación existió un individuo excepcional (o un grupo de varios), este se conserve y se incluya en la formación de la siguiente generación. El AG entonces repite el ciclo (una cantidad de “*generaciones*” decididas por el implementador): la nueva población se evalúa, se selecciona y se reproduce de nuevo. A lo largo de generaciones sucesivas, este proceso iterativo impulsa a la población hacia mejores soluciones. Como señala el autor, con cada generación, “*predominarán las cadenas asociadas con soluciones mejoradas*” (Holland, 1992b). En otras palabras, las características más aptas se vuelven más comunes a medida que el algoritmo continúa.

A través de estos pasos (inicialización aleatoria, evaluación de la aptitud, selección, cruce, mutación y generación de nuevas poblaciones), el Algoritmo Genético imita la evolución natural. En palabras del propio Holland, este proceso “*explota las regiones de mayor rentabilidad del espacio de soluciones*”. En cada iteración, la población de soluciones se filtra y se varía según la aptitud, de manera que con el tiempo los mejores rasgos tienden a acumularse. Esta descripción conceptual y textual del proceso sigue fielmente la formulación original de Holland: partir de una población aleatoria, evaluar repetidamente la aptitud, seleccionar progenitores según su aptitud, aplicar cruces y mutaciones, y formar la siguiente población, impulsando así la búsqueda de soluciones de mayor calidad (Holland, 1992b).

El Algoritmo Genético parece especialmente útil en problemas de optimización en los que el espacio de búsqueda es enorme (sobre todo cuando ni siquiera los recursos computacionales le puedan explorar completamente), pues los mecanismos genéticos no sólo provocan la “*explotación*” de buenas características encontradas en la aleatoriedad (con la selección basada en aptitud y el elitismo), sino que también propicia a la “*exploración*” de nuevas características no encontradas (con la mutación), de manera que, regresando al contexto de la composición musical, el Algoritmo Genético puede ofrecer solución en un contexto en el que, las posibilidades al elegir notas y/o acordes en la composición de una pieza son prácticamente infinitas, y en donde es posible distinguir (mediante reglas de teoría musical) si las elecciones dadas aleatoriamente, fueron buenas. Siendo más específicos, el Algoritmo Genético puede servir en la construcción de una armonía, pues esta se procesa como una cadena de acordes que se puede evaluar utilizando reglas de composición como las analizadas en **6.1.d.1 Para armonía**).

6.2.d Metaheurísticas

Hablando nuevamente de *problemas de optimización*, y sobre todo en el contexto de problemas de gran magnitud (con espacios de solución enormes), las llamadas “*Heurísticas*” surgen como atajos o técnicas que ayudan a encontrar rápidamente buenas soluciones, aun si no son necesariamente óptimas. Autores como Glover (1986), explican que las heurísticas sacrifican las garantías de obtener la mejor solución, por la velocidad y la

practicidad. La *heurística* como definición puede resultar ambigua, pues las técnicas y atajos pueden ser aplicadas para problemas de cualquier tipo. En cambio, cuando el objetivo es encontrar soluciones a problemas planteados computacionalmente, el término “*Metaheurística*” suele ser más específico en el contexto de la optimización, y se le define como una estrategia específica de alto nivel que aprovecha una o más heurísticas para mejorar la búsqueda en problemas de optimización. Por ejemplo, la técnica llamada “*Búsqueda Tabú*” (Glover, 1986) utiliza reglas basadas en memoria para evitar la revisión de movimientos recientes, y Glover la describe explícitamente como “*una metaheurística superpuesta a otra heurística*”. Las Metaheurísticas guían así las heurísticas subyacentes para evitar los óptimos locales y explorar el espacio de soluciones de forma más amplia. Estos métodos no garantizan un óptimo global, pero proporcionan procedimientos robustos para problemas difíciles y de gran tamaño.

6.2.d.1 Recocido Simulado

Un ejemplo de Metaheurística es el “*Recocido Simulado*”, que fue introducido por Kirkpatrick, Gelatt y Vecchi en “*Optimization by Simulated Annealing*” (1983), basándose en una analogía con el algoritmo de Metropolis en la mecánica estadística (Metropolis et al., 1953). El método de Metropolis et al. fue creado para simular átomos en equilibrio térmico y su idea básica es la siguiente:

1. Se tiene una configuración de átomos a cierta energía.
2. Se hace un pequeño cambio al azar y se calcula cuánto cambia la energía.
3. Si la energía baja, se acepta el cambio en el sistema de configuración de átomos.
4. Si la energía sube, se acepta el cambio sólo en ciertas ocasiones, con base a la probabilidad dada por el factor de Boltzmann ($e^{-\frac{\Delta E}{kT}}$), donde ΔE es la diferencia de energía, k es la constante de Boltzmann y T es la temperatura del sistema.
5. Se repite el proceso múltiples veces a la misma temperatura.

Al hacerlo, las configuraciones que generan energías bajas aparecerán con más frecuencia, de modo que indirectamente estas reglas propiciarían la minimización de energía. Kirkpatrick et al. adaptaron esta idea para la optimización al tratar la “*función objetivo*” como energía, además de introducir una “*temperatura*” ficticia T . Como escriben, uno “*simula el movimiento térmico de los átomos*” al aceptar movimientos ascendentes (peores) de manera probabilística para escapar del mínimo local (1983).

El *Recocido Simulado* funciona mediante los siguientes pasos:

1. **Inicialización:** Se crea aleatoriamente una solución inicial para el problema (llamada “*configuración*”) y se establece una temperatura T alta.
2. **Solución vecina:** Se realiza un pequeño cambio aleatorio en la solución actual y se calcula el coste (es decir, la función objetivo o función de aptitud del problema) para la solución original y la versión modificada.
3. **Criterio de aceptación:** Si la función de coste demuestra que la solución generada (solución vecina) tiene mejor desempeño, esta se acepta siempre y se reemplaza por la actual. Si la solución generada tiene igual o peor desempeño que la actual, esta se aceptará bajo la probabilidad $e^{-\frac{\Delta E}{T}}$. En la práctica, se genera un número aleatorio r entre 0 y 1 y se acepta si $r < e^{-\frac{\Delta E}{T}}$. Esta es exactamente la regla de Metropolis: “*Si $\Delta E < 0$, permitimos el movimiento, y si $\Delta E > 0$, permitimos el movimiento con probabilidad $e^{-\frac{\Delta E}{kT}}$* ”. Es decir, este criterio acepta la solución vecina como nueva solución siempre que sea mejor, pero en los casos en que esta es peor, decide aleatoriamente si aceptarle, calculando el cociente entre la diferencia de “*energía*” (es decir, la diferencia de aptitud) y la temperatura actual.
4. **Iteración y enfriamiento:** Se repiten los pasos 2 y 3 un número definido de veces (como parámetro de quien implementa el algoritmo), y en cada paso se reduce la temperatura. Kirkpatrick et al. priorizan una reducción lenta: “*la temperatura se reduce por etapas lentas*”

hasta que el sistema se congela y no se producen más cambios". El enfriamiento nos asegura que, conforme la temperatura baja, la probabilidad de aceptar soluciones peores es más baja, por lo que progresivamente no habrá muchos cambios en la solución.

5. **Terminación:** Se continúa hasta que la temperatura T sea muy baja, y la solución se estabilice y no cambie más, lo que se entiende como que la configuración se *congeló*. En ese momento, la solución actual sería la mejor solución encontrada.

Kirkpatrick et al., señalan las características clave de este esquema. A diferencia de la búsqueda voraz pura, el recocido simulado "*no necesita atascarse*" en un óptimo local porque se permiten movimientos ascendentes (a soluciones peores, pero que le permiten explorar más el espacio de búsqueda) mientras T no sea cero. En sus palabras, "*las transiciones fuera de un óptimo local siempre son posibles a temperatura diferente de cero*". La regla de aceptación de Metropolis et al., asegura que el algoritmo muestree configuraciones de acuerdo con la distribución de Boltzmann a cada temperatura. Como explican Kirkpatrick et al.: "*al repetir el paso básico muchas veces [...] el sistema evoluciona hacia una distribución de Boltzmann*" (1983), lo que significa que las soluciones de menor energía (mejores) se vuelven exponencialmente más probables a medida que T disminuye. La medida en que el algoritmo reduzca la temperatura en cada repetición la debe modelar quien implementa el algoritmo, y cuando el límite de enfriamiento es muy lento, se garantiza que el método (teóricamente) se acerque al óptimo global, pero requiriendo mayor cantidad de iteraciones y por ende, más tiempo. Así, el algoritmo original de Recocido Simulado, siguiendo el algoritmo de Metropolis et al., funciona imitando el recocido físico de los sólidos: comienza "*derretido*" a alta temperatura y siendo muy maleable y progresivamente se "*congela*" hasta convertirse en una solución "*sólida*".

El *Recocido Simulado* podría resolver el problema de, teniendo una oración, ajustar el tamaño de sus sílabas, con una función objetivo que aplique los principios de la prosodia (consultar **6.2.b.2 Prosodia**) para construir el indicio de una melodía con buen ritmo. Es decir, si se consideran las sílabas de la oración como notas, y las duraciones de estas se codifican como figuras rítmicas (consultar **6.1.a Conceptos básicos**), entonces se podría modelar un problema de optimización en donde el Recocido Simulado encuentre la mejor selección de figuras rítmicas para adaptar la frase según las reglas de la prosodia.

6.2.e Algoritmo LWRS

El Algoritmo "*LWRS*" ("*Logarithmic Weighted Random Selector*" o por su traducción: "*Selector Aleatorio Logarítmicamente Ponderado*") es un mecanismo de selección cuyo propósito es ponderar la selección de elementos en una lista, asignando preferencia respecto al orden en la misma. Es propuesto por Ramos Herrera (2025b), quien es el mismo autor del presente trabajo. *LWRS* surge, según su autor, bajo la necesidad de ponderar los pesos de selección (es decir, la probabilidad de ser seleccionada cualquiera de las opciones disponibles), en fenómenos para los cuáles no se poseen valores numéricos que definan dicho peso de elección, pero para los cuáles sí se conoce su orden de preferencia.

En **6.2.a.2 Modelos** se mencionaban los Modelos de cadenas de Markov, que se construyen definiendo una serie de estados posibles de decisión y formulando una tabla de probabilidades que indica, para cada estado posible, la probabilidad de cambiar a cuál otro estado, de manera que definiendo probabilidades de transición y transaccionando aleatoriamente bajo esas probabilidades (como lanzando un dado), se conseguía un modelo capaz de tomar decisiones y generar secuencias de estados aleatorios. En **6.2.c.1 Algoritmo Genético** se describió un mecanismo de "*selección proporcional a la aptitud*", el cual, teniendo una población de elementos con cada uno una calificación de aptitud, se ponderaba la selección con respecto a dicha calificación, favoreciendo a los más aptos, pero sin quitarle posibilidad a posibles eslabones débiles. Tanto en las cadenas de Markov como en la *selección proporcional a la aptitud* se necesita que los elementos a elegir posean un valor numérico que defina su probabilidad de selección. *LWRS* surge como una alternativa a la selección aleatoria de elementos, pero conociendo solamente su orden de preferencia, y no la magnitud de preferencia; esto para escenarios en los que no se posean dichos valores numéricos.

Un caso práctico y aplicable es el de la música; por ejemplo, para decidir las notas de una melodía. Si se quisieran formular probabilidades, sería necesario hacer un estudio completo con ejemplos de música real, ya sea con modelos entrenados que extraigan los patrones de selección, o bien con definiciones estadísticas en Modelos de Markov que enumeren los posibles escenarios de selección definiéndoles valores constantes de probabilidades. El Algoritmo *LWRS* asigna automáticamente los pesos de selección con base en una escala logarítmica basada en el orden, en donde el primer elemento posee la mayor probabilidad de selección, después el segundo con un poco menos, etc. Esta probabilidad está ajustada al tamaño de la lista de elementos, por lo que para cada tamaño de lista, la magnitud de ponderación cambiará (Ramos Herrera, 2025b). Esta estrategia de ponderar según el orden resulta útil cuando se sabe que en la música tonal existen jerarquías tonos (consultar **6.1.c.i Música tonal**), y recordando además que en una melodía tonal la prioridad principal de selección será para los tonos del acorde que suena, después los tonos de la escala, y finalmente el resto de las notas cromáticas. Esta jerarquía y los elementos que conforman cada grupo es variable según el escenario de composición, por ejemplo, el acorde puede tener 3, 4, 5 o más notas, además de que, estas mismas pueden pertenecer o no a la escala, además, distintas escalas pueden tener distintas cantidades de notas. Esto provoca que las implementaciones de *LWRS* manejen probabilidades dinámicas ajustadas al escenario preciso de cada nota a elegir, y no con probabilidades fijas, agregando factores extras como la preferencia en el orden con respecto a la cercanía con la nota anterior seleccionada (para priorizar fluidez melódica por encima de saltos interválicos bruscos). Además, aplicando el algoritmo *LWRS* se priorizan selecciones que la teoría respalda que serán parte de la música tonal popular, pero a la vez, no perderá la variedad en explorar posibles escenarios que la teoría podría catalogar como poco estéticos, pero que en su ejecución demuestren ser agradables a la escucha. Es decir, el favorecer decisiones estéticas, pero permitiendo (aunque con baja probabilidad), que cualquier decisión posible sea tomada.

El Algoritmo *LWRS*, para asignar las probabilidades de selección a cada elemento construye una escala logarítmica con sus respectivos espacios y distancias; es decir, su probabilidad de decisión. Esta escala se construye como un conjunto de puntos $\mathbb{P}$ definido por Ramos Herrera como:

$$P = \{p_i: p_i = (n \log_{n+1}(i + 1), 0), \quad n, i \in \mathbb{N}, \quad 1 \leq i \leq n\} \quad (1)$$

En donde:

- p_i es la coordenada (x, y) del punto en la escala.
- $n \log_{n+1}(i + 1)$ es la distancia desde el inicio de la escala $(0, 0)$ hasta el punto i .
- n es la cantidad de elementos en la lista.
- i es el número de punto.

Por ejemplo, una escala logarítmica construida a partir de una lista de 5 elementos ($n = 5$) sería así:

$$p_1 = (5 \log_{5+1}(1 + 1), 0) = (1.934, 0) \quad (2)$$

$$p_2 = (5 \log_{5+1}(2 + 1), 0) = (3.066, 0) \quad (3)$$

$$p_3 = (5 \log_{5+1}(3 + 1), 0) = (3.869, 0) \quad (4)$$

$$p_4 = (5 \log_{5+1}(4 + 1), 0) = (4.491, 0) \quad (5)$$

$$p_i = (5 \log_{5+1}(5 + 1), 0) = (5, 0) \quad (6)$$

Una representación visual del ejemplo de escala Logarítmica construida para 5 elementos de selección se muestra a continuación en la **Figura 23**.

Logarithmic scale for n = 5

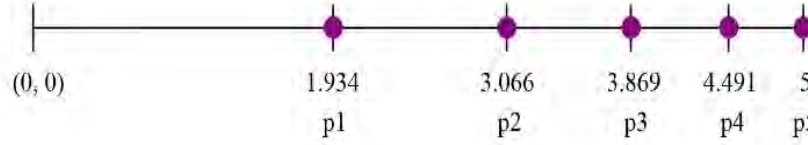


Figura 23. Escala logarítmica de 5 elementos.

Nota: Ilustración extraída del trabajo de Ramos Herrera (2025b).

La escala logarítmica mostrada en la **Figura 23** muestra cuál sería el espacio de probabilidad de selección para cada elemento situado en esta. Notando que entre $(0,0)$ y p_1 existe una distancia de 1.934 de 5 que es el tamaño total de la lista, por lo que ese espacio de selección $(1.934/5 = 38.68\%)$ para el primer elemento. El segundo elemento estaría en el espacio entre 1.934 (p_1) y 3.066 (p_2), es decir, su espacio es $3.066 - 1.934 = 1.132$ de 5, es decir, un 22.64%. Los elementos 3, 4 y 5 tendrían respectivamente las probabilidades de 16.06%, 12.44% y 10.18%. Es decir, para una selección LWRS con 5 elementos, las probabilidades serían: 38.68%, 22.64%, 16.06%, 12.44% y 10.18%, demostrando que existe una preferencia respecto al orden.

La fórmula (7) calcula la probabilidad de cada elemento en una lista LWRS sería:

$$w_i = \frac{n \log_{n+1}(i+1) - n \log_{n+1}(i)}{n} * 100 \quad (7)$$

Donde:

- w_i es la probabilidad de selección (medida en unidades de cientos) del elemento i .
- n es la cantidad de elementos en la lista.
- i es el número de elemento.

Con el uso del Algoritmo LWRS (Ramos Herrera, 2025b), se hace factible programar la toma de decisiones de, por ejemplo, tonos en las notas de una melodía, en donde sea posible aplicar las reglas de composición respectivas (consultar **6.1.d.2 Para melodía**) de modo que, sin necesidad de definir valores numéricos de peso para las posibles notas en cada caso posible, se puedan tomar decisiones basadas en el contexto armónico de la misma, creando para cada caso una lista de selección con los tonos ordenados por jerarquía, esperando mayormente decisiones estéticas según la teoría de la música tonal, aunque dejando espacio a que cualquier posible decisión creativa suceda, aun si la probabilidad es ínfima.

7 Proyectos relacionados

Como se mencionó anteriormente, el presente proyecto encuentra una intersección de conocimientos entre el par de áreas de las *Ciencias de la Computación* y la *Teoría Musical*. Bajo este preámbulo se plantea la creación de un algoritmo con la capacidad de componer piezas musicales de música tonal, adaptando las técnicas computacionales: *Gramáticas Generativas*, el método de *Cómputo Evolutivo*, *Algoritmo Genético*, la Metaheurística *Rocido Simulado*, el algoritmo LWRS, junto a las reglas de estética musical anteriormente estudiadas. Con estos tópicos en mente y teniendo como fin explorar la generación autónoma de música, se puede revisar la literatura para saber qué se ha implementado en la investigación y resolución del problema planteado.

7.1 Modelos con entrenamiento

“Algorithmic Music Composition Based on Artificial Intelligence: A Survey”

- **Autores:** Omar Lopez-Rincon, Oleg Starostenko, Gerardo Ayala-San Martín.

- **Año:** 2018.
- **Referencia:** (Lopez-Rincon et al., 2018).

Este artículo hace una recopilación de tamaño considerable en la que se distinguen las técnicas aplicadas en la composición musical basada en Inteligencia Artificial. Agrupando los trabajos más relevante recopilados en este trabajo, se obtienen las siguientes categorías:

1. Métodos basados en Deep Learning (aprendizaje profundo):

- a. **Redes Deep Belief:** Bickerman et al. (2010) utilizaron *Redes Neuronales Artificiales probabilísticas profundas* para la creación automatizada de improvisaciones melódicas de jazz sobre secuencias de acordes. Este enfoque permite crear *riffs (motifs)* de jazz.
- b. **Red Neuronal Convolutiva:** Grachten & Chacón (2017) experimentaron con dos enfoques distintos de aprendizaje: *REUSE ALL* (reanudar el aprendizaje con parámetros ajustados) y *RESET* (reinicio con inicialización previa), entrenando modelos de *Red Neuronal Convolutiva* representando las notas musicales como píxeles para predecir y decodificar melodías de un conjunto de piezas reales.
- c. **Redes Neuronales Recurrentes:**
 - i. En el trabajo de Huang & Wu (2016) se creó un modelo generativo a partir de la arquitectura de una red neuronal profunda, con el enfoque de que se genere música que tenga tanto armonía como melodía, y que sea tan aceptable como música compuesta por humanos. Este trabajo destaca el aporte basado en que trabajos anteriores se han basado en únicamente creación de melodía. Este enfoque implementa redes *LSTM*.
 - ii. Bretan et al. (2016) se describe un modelo generativo que combina el llamado *Modelo Semántico Estructurado (DSSM)* y una red *LSTM*. Se utilizaron métricas objetivas que incluyen la clasificación media y la precisión y con una prueba de escucha subjetiva en la que se les pide a músicos expertos que completen una tarea de clasificación de elección forzada.
 - iii. En el trabajo de Chu et al. (2016) se presenta un “nuevo marco de generación de música pop”, utilizando una red recurrente jerárquica, donde capas inferiores generan una melodía, mientras que capas superiores producen percusiones y acordes. Su métrica muestra que los usuarios escuchas tuvieron preferencia por la música generada por este modelo, que por el más reciente modelo (para aquel entonces) de Google.
 - iv. El artículo de Cox (2010) explora un concepto de Meyer (1956) que estipula que “*el significado de la música está directamente relacionado con la entropía*”, como una manera de establecer que los modelos estadísticos actuales de la música se limitan a sólo una línea melódica, lo que impide una investigación más amplia de la hipótesis de Meyer. Se propone un modelo de *Red Neuronal Recurrente* que produce estimaciones de entropía para música con múltiples melodías, analizando en el proceso un cuarteto de cuerdas compuesto por Haydn.
 - v. En el trabajo de Hadjeres et al. (2016) se presenta “*DeepBach*”, un modelo gráfico destinado a modelar música polifónica y específicamente a piezas de tipo himno. Se entrenó con piezas musicales, en específico con armonizaciones corales de J.S. Bach y genera corales convincentes al estilo de Bach. Incorpora igualmente una extensión al software de edición de música *Muscore*, para ser utilizado por un usuario músico.

2. Métodos de Composición Estocástica

- a. **Modelos de Markov:**
 - i. El proyecto “*ALYSIA*” (Ackerman & Loker, 2016) presenta una aplicación para la composición lírica automatizada. *ALYSIA* se basa en un modelo de aprendizaje automático que utiliza bosques aleatorios. Utiliza cadenas de Markov para componer melodías basadas en frases líricas.
 - ii. En la investigación de M. Kaliakatsos-Papakostas & Cambouropoulos (2014) se introducen restricciones armónicas en un modelo oculto de Markov. Este modelo se

centra en armonizar melodías. “Este trabajo está motivado por el hecho de que algunas partes de una frase (como la cadencia) o una pieza (por ejemplo, puntos de modulación, picos de tensión, cadencias intermedias, etc.) son características de la identidad idiomática de la frase o pieza”. En cambio, en su trabajo (2011) se expone igualmente un modelo de Markov, pero para la identificación de compositores con base en capturar la estructura de fraseo de sus melodías. Este trabajo igualmente propone *modelos evolutivos* para ajustar los pesos introducidos para explotar la identificación de compositores.

- iii. En el texto de Papadopoulos et al. (2016) se presenta “*FlowComposer*” como una aplicación web que permite a usuarios componer partituras. Este trabajo al igual que el presente habla de música como “*melodías etiquetadas con acordes*”. El usuario observa el proceso de generación, en donde este especifica el estilo de la partitura con base a seleccionar de un corpus de partituras ya existentes o una introducida por el usuario. Este sistema produce a partir de entrenamiento con piezas reales, pero a partir de un modelo gráfico que combina cadenas de Markov.
- iv. El proyecto de Scirea & Barros (2015) presenta el sistema “*SMUG*” plantea la problemática de que, a diferencia de los compositores reales, los modelos no se inspiran de la vida y del mundo como “*entrada*”. Este sistema genera letras y melodías a partir de datos del mundo real, usando artículos académicos como referencia. Aprovecha cadenas de Markov y utiliza cómputo evolutivo para evitar sobreajuste.

3. Modelos de agentes:

- a. Eigenfeldt & Pasquier (2012) proponen el concepto de *agentes creativos*, exponiendo que una interacción entre el usuario creador y el sistema puede mejorar los resultados en el proceso de composición musical, con base en que los propuestos denominados “*curadores*” permiten la toma de decisiones basadas en heurísticas para componer en tiempo real.
- b. Sobre el trabajo de Navarro et al. (2014) se presenta un sistema *multi-agente* enfocado enteramente en la composición de armonías; es decir, en progresiones de acordes.

“*Artificial Intelligence for Music Composition*”, capítulo del libro “*Handbook of Artificial Intelligence for Music*”

- **Autores:** Artemi-Maria Gioti.
- **Año:** 2021.
- **Referencia:** (Gioti, 2021).

El libro de 2021 “*Handbook of Artificial Intelligence for Music*” contiene un capítulo enfocado en explorar el estado actual (para 2021) en la literatura relacionada a Inteligencia Artificial enfocada en composición musical. Sus secciones siguientes resumen relevantemente el estado del arte:

- 1. ***Artificial Intelligence and Distributed Human-Computer Co-creativity:*** En este apartado se habla sobre cómo el marco típico en la Inteligencia Artificial enfocada en la composición musical es la automatización, cuestión que limita el aprovechamiento de la creatividad humana. Herramientas como *WaveNet*, de Van Den Oord et al. (2016), permiten generación de música a partir de datos de audio sin procesar, yendo más allá de las representaciones tradicionales *MIDI*¹⁹. Estos sistemas carecen de interacción humana. Se habla de la “*co-creatividad*” como una interacción entre el sistema y el músico. Acá se nombra cómo a pesar de los avances en Inteligencia Artificial (IA), los sistemas de composición autónoma enfrentan limitaciones en términos de innovación artística, ya que principalmente imitan estilos existentes en lugar de superar los límites, además de carecer capacidades interactivas que los harían más útiles para los artistas. En lugar de centrarse en sistemas

¹⁹ “*MIDI*” es un formato computacional para archivos que almacenan música a través de canales que representan instrumentos, y las notas musicales que interpretan, dando para cada nota, tono, duración, intensidad y demás controles y características.

completamente autónomos, el capítulo aboga por un enfoque ecosistémico que ve a la IA como una herramienta para mejorar la creatividad humana en lugar de reemplazarla. En este enfoque, la IA ayuda a ampliar las posibilidades creativas y desafía las nociones tradicionales de autoría.

2. **Machine Learning: Applications in Music and Compositional Potential:** En esta sección se hace una descripción de los métodos de aprendizaje automático y se hace especial énfasis en el potencial de los *Modelos de Aprendizaje por Refuerzo*, dado que estos funcionarían como agentes que configuren su capacidad de composición a través de la retroalimentación positiva o negativa del músico.
3. **Computational Aesthetic Evaluation:** Explora las técnicas para evaluar la calidad estética de composiciones humano-computadora, desde teorías formuláticas hasta modelos psicológicos y estudios empíricos (Galanter, 2012). Se destaca además cómo ha habido críticas generales en los intentos de definir valores estéticos “universales” (Martindale, 1998). Se habla además de cómo implementaciones podrían aprovechar estas fórmulas de preferencias estéticas ajustadas por un humano, para sesgar las composiciones de un modelo, como en el proyecto *BIAS* de la misma Gioti (s/f), para el cual se creó un modelo de *Red Neuronal* que predice las preferencias estéticas de un compositor para piezas de clarinete y bajo, en donde se grabaron sesiones de improvisación del mismo y después de segmentaron partes de su interpretación, para pedirle a este mismo intérprete que etiquetara cada segmento en una escala de 1 (nada interesante) al 5 (extremadamente interesante), de manera que este algoritmo aprendió los patrones de preferencia estética del compositor para generar piezas con dichos patrones.
4. **Human-Computer Co-exploration:** Enfatiza la importancia de estudiar enfoques de la ya mencionada *co-creatividad*, y la *co-exploración*, mencionando sistemas como el de *Sonic Xplorer* (Tsiros, 2017), que modela una red neuronal que se entrena con ejemplos elegidos por el usuario y este adapta su creatividad a dichos ejemplos, junto a la interacción del usuario al ajustar controles deslizantes de una aplicación para describir las cualidades de sonido deseadas.
5. **The Computer as a Compositional Prosthesis:** Se menciona nuevamente que un sistema de IA debe funcionar como una “prótesis” del compositor, haciendo el relevante razonamiento de que una herramienta como esta debe ayudar al compositor a liberarse de su pensamiento habitual, para explorar nuevos espacios conceptuales, definiendo que debe ser el usuario el que acepte o rechace las sugerencias de la máquina.
6. **Limitations of Machine Learning:** Se estudia cómo los modelos de *Machine Learning* poseen una limitante importante, pues estos modelos imitan las obras de muestra y carecen de capacidades interactiva o de mecanismos de retroalimentación del usuario.
7. **Composition and AI - The Road Ahead:** Menciona y concluye que muchas de las herramientas existentes actuales se centran enteramente en la automatización y creación de arquitecturas de “*caja negra*” (es decir, modelos de los cuales no se entiende el funcionamiento interno, sino sólo la entrada y la salida). Se hace de nuevo énfasis en la necesidad de interactividad y flexibilidad, pues siendo que estas herramientas generativas a menudo priorizan la imitación de estilos, no tienen la capacidad de alinearse con la naturaleza abierta y exploratoria de la creación artística.

“Text-based LSTM networks for Automatic Music Composition”

- **Autores:** Keunwoo Choi, George Fazekas, Mark Sandler.
- **Año:** 2016.
- **Referencia:** (Choi et al., 2016).

Este trabajo se centra en el uso de modelos de *Red Neuronal Recurrente LSTM*, entrenando con el fin de generar dos productos: acordes de jazz (armonía) y pistas de batería de rock, tomando como base representaciones de texto. Se utilizaron 2,486 canciones transpuestas a una tonalidad estándar de Do y con sus respectivas codificaciones de acordes y percusiones en texto. En este caso de estudio sólo funcionó la creación de acordes.

No se establecieron métricas de estética para medir la capacidad de composición.

“Design and Implementation of Automatic Music Composition System Based on Deep Learning”

- **Autores:** Fei Wang.
- **Año:** 2024.
- **Referencia:** (Wang, 2024).

Wang explora el uso de *Redes Neuronales Recurrentes LSTM*. A diferencia de otros estudios, en este se propone como contribución la mejora en los optimizadores y un estudio más riguroso del ajuste del modelo, proveyendo una amplia variedad de géneros musicales en el entrenamiento para mejorar su capacidad creativa. Se presume más eficiencia en el tiempo de ejecución comparado a otros trabajos de la literatura.

No se exponen o desarrollan métricas para medir la calidad de las piezas generadas, sino que las métricas están basadas puramente en el desempeño del algoritmo con base en sus datos de entrenamiento.

7.2 Modelos sin entrenamiento

“Composición Musical usando algoritmos genéticos”

- **Autores:** Carvajal Ramírez, J. C., Giraldo Giraldo.
- **Año:** 2012.
- **Referencia:** (Carvajal Ramírez & Giraldo Giraldo, 2012).

Se utilizó un algoritmo genético en el que una función objetivo premia con base en reglas de composición del compositor del barroco Johan Sebastian Bach. Este modelo no requiere de piezas reales de entrenamiento, sino que las reglas en la función objetivo premian la generación de piezas basadas puramente en el estilo del compositor, generando únicamente piezas en este estilo.

“Composición Musical a Través del Uso de Algoritmos Genéticos”

- **Autores:** Ezequiel Moldaver, Hernán Merlino, Enrique Fernández.
- **Año:** 2014.
- **Referencia:** (Moldaver et al., 2014a).

Combina Algoritmo Genético con una Red Neuronal que toma como entrada una serie de canciones ya existentes y el algoritmo se entrena generando variaciones de las piezas ya existentes, con un fitness basado en que las piezas generadas se separen lo suficiente de las iniciales, pero siendo buenas. Aquí se plantea la cuestión de evaluar la calidad de una pieza con base en la concordancia de todos no disonantes entre la melodía y la armonía. Este trabajo en esencia utiliza piezas reales para su entrenamiento, pero estas son dadas por el usuario a diferencia de los modelos de *Machine Learning*, que se entrenan por aquellas piezas seleccionadas por el investigador y desarrollador del modelo.

“Application of genetic algorithm in model music composition innovation”

- **Autores:** Delei Xu, Haoming Xu.
- **Año:** 2024.
- **Referencia:** (Xu & Xu, 2024).

Algoritmo Genético que rellena según la cantidad de compases elegidos, la tonalidad y el compás. Sólo compone melodía. En este trabajo, los autores Xu exponen una métrica de evaluación para los resultados del algoritmo con base en escuchas de dos tipos de sujetos de prueba: compositores y público general. Se hicieron gráficos con los resultados de puntuación dados por ambos tipos de sujeto.

7.3 Composición basada en frases

“Expressing and Developing Melodic Phrases in Gamelan Skeletal Melody Generation Using Genetic Algorithm”

- **Autores:** Ahmad Zainul Fanani; Arry Maulana Syarif; Guruh Fajar Shidik; Aris Marjuni.
- **Año:** 2024.
- **Referencia:** (Fanani et al., 2024).

Expone un novedoso sistema de composición de melodías basado en la creación de frases de preguntas y respuestas adaptadas a melodías, con un sistema que se entrena de unas iniciales diez composiciones originales y de ello tiene la capacidad de extender su capacidad compositiva más allá de ellas. Para la creación de frases utiliza una técnica llamada *“Gamelan Skeletal”* en conjunto con un Algoritmo Genético. Sólo compone melodías.

El trabajo de Fanani et al. es relevante para el presente trabajo, pues acá se establece una relación entre frases del Lenguaje Natural y las melodías de una pieza musical. Acá la generación de frases es el paso relevante en la creación de melodías, al igual que en la presente propuesta.

“Controllable Syllable-Level Lyrics Generation from Melody with Prior Attention”

- **Autores:** Zhe Zhang, Yi Yu, Atsuhiko Takasu.
- **Año:** 2024.
- **Referencia:** (Zhang et al., 2024).

El trabajo de Zhang et al. no responde a la generación de piezas musicales, como el presente trabajo, sin embargo, se establece una relación entre oraciones del *Lenguaje Natural* y la melodía de una canción, de modo que su propuesta genera líricas y oraciones teniendo como entrada melodías; siendo esto lo contrario a la propuesta en el algoritmo MIA.

“A Generative Theory of Tonal Music”

- **Autores:** Fred Lerdahl, Ray S. Jackendoff.
- **Año:** 1985.
- **Referencia:** (Lerdahl & Jackendoff, 1985).

Libro que desde 1985 propuso una teoría completa para la generación de música tonal. Sin implementaciones en modelos computacionales y sin aplicaciones directas, la teoría completa de este libro formula cuestiones relevantes como la generación de melodías utilizando oraciones del *Lenguaje Natural* y su separación silábica. Esta propuesta resulta totalmente relevante al trabajo aquí propuesto, y continuamente se discutirán aplicaciones de la misma adaptadas al algoritmo MIA.

7.4 Resumen de proyectos

La siguiente **Tabla 3** contiene una recopilación de los proyectos revisados en esta sección, puntualizando características relevantes para este trabajo, como la utilización (o no) de datos de entrenamiento, los modelos implementados, la generación (o no) de melodías y/o armonías y si existe el uso de alguna técnica de Procesamiento de Lenguaje Natural (NLP). La tabla contiene 18 proyectos, que de entre los revisados anteriormente, son aquellos que desarrollan propiamente un sistema de composición musical autónoma. Para cada caso se describe brevemente el trabajo realizado.

Tabla 3. Recopilación de proyectos de composición musical con cómputo inteligente y sus características.

Proyecto	Descripción	¿Usa datos?	Modelo	¿Compone melodía?	¿Compone Armonía?	¿Usa NLP?
(Bickerman et al., 2010)	Red <i>Deep Belief</i> para creación de riffs improvisados sobre acordes de jazz	✓	RN Deep Belief	✓		
(Grachten & Chacón, 2017)	RN Convolutacional para predicción y generación de melodías	✓	RNA Convolutacional	✓		
(Huang & Wu, 2016)	Generación de melodías y armonías tan aceptables como música real	✓	RN Recurrente LSTM	✓	✓	
(Bretan et al., 2016)	Métricas objetivas de escucha subjetiva con músicos expertos	✓	RN Recurrente LSTM	✓		
(Chu et al., 2016)	Generación de música pop con melodía, armonía y percusiones	✓	RN Recurrente Jerárquica	✓	✓	
(Cox, 2010)	Múltiples líneas melódicas con base a un cuarteto de cuerdas de Haydn	✓	RN Recurrente	✓		
(Hadjeres et al., 2016)	" <i>Deep Bach</i> " es un modelo entrenado con música de J.S. Bach	✓	RN Recurrente	✓		
(Choi et al., 2016)	Genera armonía de jazz y pistas de batería. Sólo funcionó la armonía	✓	RN Recurrente LSTM	✓		

(Wang, 2024)	LSTM cuyos optimizadores están mejor ajustados	✓	RN Recurrente LSTM	✓	✓	
(Ackerman & Loker, 2016)	“ALYSIA” es una aplicación completa. Compone melodías con base en líricas	✓	Markov y Random Forest	✓	✓	✓
(M. Kaliakatsos-Papakostas & Cambouropoulos, 2014)	Se centra en armonizar melodías		Markov		✓	
(Papadopoulos et al., 2016)	“FlowComposer” es una aplicación web: el usuario selecciona el estilo de las partituras de un corpus existente	✓	Markov	✓	✓	
(Scirea & Barros, 2015)	El sistema “SMUG” construye líricas y melodías con base a datos del mundo real como “entrada”	✓	Markov y Cómputo Evolutivo	✓		✓
(Carvajal Ramírez & Giraldo Giraldo, 2012)	Algoritmo Genético que premia reglas de composición del estilo barroco de J.S. Bach para imitar su estilo		Algoritmo Genético	✓		
(Moldaver et al., 2014)	Algoritmo Genético y Red Neuronal que toma de entrada piezas ya existentes y las varía	✓	Algoritmo Genético y RN	✓	✓	
(Xu & Xu, 2024)	Propone evaluación de piezas con: compositores y público general		Algoritmo Genético	✓		

(Fanani et al., 2024)	Se entrena con diez composiciones originales y las extiende. Compone frases melódicas a partir de la técnica “Gamelean Skeletal”	✓	Algoritmo Genético y Gamelean Skeletal	✓	✓
(Zhang et al., 2024)	Genera frases de lenguaje con base en melodías	-	-	-	✓

7.5 Oportunidades de contribución

De la literatura de proyectos similares revisada, se encontraron 18 trabajos (listados en la **Tabla 3**) que describen el desarrollo de una herramienta de inteligencia computacional para la composición musical. Al extraer características de sus implementaciones y haciendo un recuento de las mismas, es posible encontrar posibles espacios a la contribución. De los 18 trabajos listados, sus estadísticas son:

- El 77% utilizan datos de entrenamiento.
- El 33% utilizan *Redes Neuronales Recurrentes*.
- El 88% componen melodía.
- El 38% componen armonía.
- El 22% utilizan técnicas de *Procesamiento de Lenguaje Natural*.
- El 55% se limitan a un género, estilo o compositor específico (Bickerman et al., 2010; Carvajal Ramírez & Giraldo Giraldo, 2012; Choi et al., 2016; Chu et al., 2016; Cox, 2010; Hadjeres et al., 2016; M. Kaliakatsos-Papakostas & Cambouropoulos, 2014; Moldaver et al., 2014b; Papadopoulos et al., 2016; Zhang et al., 2024).

Se puede concluir de los 18 proyectos listados que: 1. La mayoría de ellos se centra en modelos entrenados con datos, 2. Un gran porcentaje utilizan Redes Neuronales Recurrentes, 3. La gran mayoría componen melodía, pero sólo el 38% componen igualmente una armonía, 4. El Procesamiento de Lenguaje Natural no está muy presente en los proyectos listados.

Considerando además lo descrito a lo largo de la sección, y haciendo especial énfasis en las conclusiones de la amplia revisión del Estado del Arte de Lopez-Rincon et al. (2018) y Gioti (2021), se identifican a continuación las áreas de mayor repercusión en las que el presente proyecto destacaría con contribución:

1. **Algoritmo independiente de datos de entrenamiento:** Evitando el enfoque más común, en el que se utiliza música real de artistas como entrenamiento para modelos, este trabajo explora de manera novedosa técnicas diversas e independientes de datos de entrenamiento.
2. **No-especificidad en estilos y géneros:** Dado que el entrenamiento con datos propicia a la imitación, y tal como mencionaba Gioti, la gran mayoría de trabajos enfrentan limitaciones en términos de innovación artística, ya que principalmente imitan estilos existentes en lugar de superar los límites. Es por ello que el uso de un enfoque basado en patrones musicales presentes en la *Música Tonal*; que según decía Ravera (2022), es casi toda la música que escuchamos, permite no limitar los resultados a un solo género o estilo específico, como sí lo hacen la mayoría de los trabajos listados (55%). Este enfoque podría aproximarse a cualquier género o estilo que sea parte de la música tonal.
3. **Música con coherencia:** La literatura revisada señala que a menudo los sistemas de generación de música no tienen una manera de limitar la creación de notas en las melodías, de manera que se demuestran mucha aleatoriedad en sus resultados. No se proponen mecanismos que propicien la

percepción de “*intencionalidad*”, como sí lo podría ser la implementación de *direccionamiento de voces* (consultar **6.1.d.2 Para melodía**), o la generación de oraciones basadas en reglas sintácticas, que, a pesar de ser estas aleatorias, son finitas en su cantidad de notas. Otros mecanismos que propicien la repetición de frases para el establecimiento de *motivos* (consultar **6.1.b.2 Motivo**) podrían ser un espacio de contribución no explorado completamente.

4. **Métrica de evaluación estética:** Entre los trabajos revisados, únicamente la publicación de Bretan et al. (2016) y Xu & Xu (2024) ofrecen alguna métrica de evaluación experimental que destaque la calidad en las piezas producidas por sus métodos. Estas métricas se basaron en la evaluación de piezas utilizando personas (tanto expertos en música como personas comunes). Entre los demás trabajos revisados, sólo algunos ofrecen métricas basadas en evaluaciones conceptuales de funciones de coste en sus modelos, que no demuestran verdaderamente la calidad de los resultados generados por sus métodos. Este trabajo desarrolla una métrica y un método experimental que sí permita evaluar la calidad musical de sus composiciones.
5. **Co-creatividad:** El problema principal en los sistemas de composición musical autónoma, según Gioti (2021), es que la mayoría de los modelos son “*cajas negras*” que reciben una entrada muy simple del usuario y arrojan un resultado, sin dar espacio a la configuración, o al ajuste de los resultados del modelo (dado que la mayoría son solamente entrenados). Escribe Gioti que existe una gran deficiencia de sistemas que permitan al usuario músico ajustar parámetros para propiciar su propia creatividad en los resultados del modelo, y no sólo una instrucción de generación.
6. **Algoritmo LWRS:** El algoritmo LWRS (Ramos Herrera, 2025b) es una propuesta novedosa no utilizada hasta la fecha en ninguna aplicación documentada, por lo que este se presentaría como el primer trabajo en implementarle.

8 Metodología

Para abordar los objetivos del proyecto y responder las preguntas planteadas, se plantea una metodología *experimental*, basada en la definición de González Mares (2019), que implica que durante el desarrollo, se estará configurando y manipulando la “*variable de estudio*”, que bajo este contexto es la configuración y arquitectura del algoritmo MIA; además de que, se formularán experimentos para la evaluación de los resultados obtenidos.

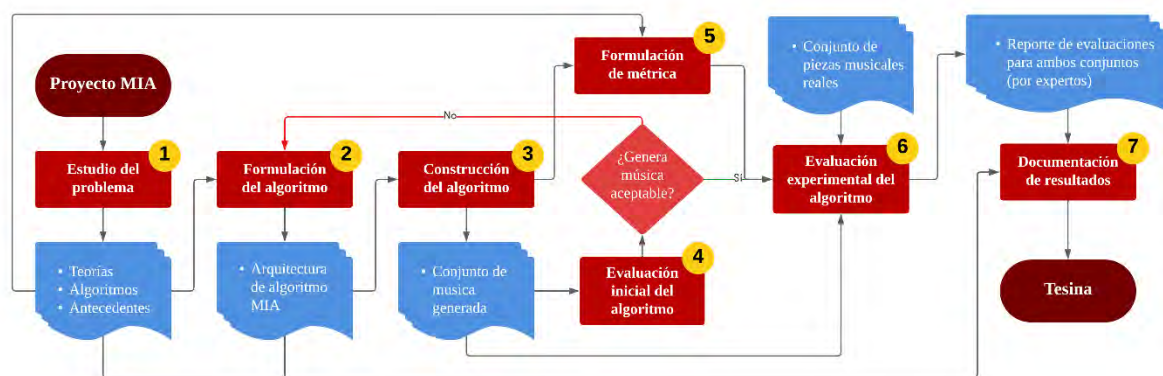


Figura 24. Metodología experimental del proyecto MIA.

La **Figura 24** muestra los pasos durante el desarrollo del presente proyecto, separando este en siete fases cuya definición y características son las siguientes:

- [1] **Estudio del problema:** Esta fase consiste en la recopilación y lectura de fuentes para la obtención de teorías de música y computación, algoritmos de potencial uso y antecedentes de proyectos relacionados, para comprender el estado actual del problema y los avances de cada área del mismo.

En esta área se comprenden las posibles áreas de contribución, con base tanto en las deficiencias como las fortalezas encontradas en la literatura del problema.

- [2] **Formulación del algoritmo:** Teniendo teorías, antecedentes, algoritmos potenciales y definidas las oportunidades de contribución, se idea y estructura una potencial arquitectura de algoritmo. En esta fase se construye un esquema de cómo se pretende abordar la composición musical a través de entradas, procesos y salidas de un algoritmo que potencialmente logre el objetivo planteado.
- [3] **Construcción del algoritmo:** Se implementa en código de programación la estructura de arquitectura formulada en la fase 2, teniendo como resultado un sistema computacional que utilice el algoritmo y que arroje los resultados esperados respecto a composición musical basada en melodía y armonía.
- [4] **Evaluación inicial del algoritmo:** Esta fase utiliza la implementación en código desarrollada en la fase 3 y la ejecuta para obtener 10 piezas musicales, que a través de una evaluación informal basada en la percepción y el criterio del autor e implementador del algoritmo, se usan para responder:
 - **¿El algoritmo implementado genera música aceptable?:** Esta pregunta, basada en el criterio personal del implementador, responde con base en si a las piezas musicales generadas se les considera lo suficientemente consistentes respecto a características de la música tonal de la teoría estudiada, como para concluir que el algoritmo cumple el objetivo de *“generar piezas comparables estéticamente con composiciones reales de la música tonal contemporánea”*. De responderse que *“no”* a la pregunta, el desarrollo se direcciona a la fase 2, y de responderse *“sí”*, se continúa en la fase 6.
- [5] **Formulación de métrica:** Esta fase puede desarrollarse en paralelo o incluso antes que la fase 4, pues esta depende principalmente de la teoría recopilada en la fase 1. En esta fase se formula la *“métrica de estética basada en la percepción”*, es decir, un método para la evaluación que permita identificar características importantes en la música tonal que propicien la estética de la misma. Es preciso que esta fase sea independiente de la 4, pues la métrica que se formula no debe estar basada en la construcción del algoritmo, sino puramente en la teoría musical y en aquellas reglas que describan qué características se esperarían en piezas de la música tonal. De igual manera la formulación y construcción del algoritmo (de las fases 2 y 3) debe estar basada puramente en la teoría revisada y no en la métrica formulada.
- [6] **Evaluación experimental del algoritmo:** Esta fase recibe como entrada: 1. La métrica de estética basada en la percepción, 2. El conjunto de piezas musicales generadas por el algoritmo en la fase 4, y 3. Una lista de piezas musicales reales de la música tonal contemporánea, elegidas de compositores humanos galardonados. Esta fase reúne a una serie de evaluadores con conocimientos académicos en la composición y análisis musical (estudiantes de alto grado de carreras afines a la licenciatura en Música) y se les pide utilizar la métrica para, teniendo el conjunto de piezas reales y piezas del algoritmo (ordenadas aleatoriamente) y según su percepción y expertís, hacer una evaluación estética basada en su percepción. Los músicos elegidos desconocerán que las piezas recopiladas de compositores reales lo son; es decir, se les indicará que absolutamente todas las piezas son generadas por un algoritmo computacional, de manera que se puedan extraer estadísticas con base en sus evaluaciones, comparando música real y música computacional, pero sin que estos tengan el sesgo de evaluar distinto a cada conjunto con base en su origen. De esta manera, el conjunto de música de artistas reales funge como grupo de control y comparación estadística. Se debe asegurar que el músico no conozca de antemano el trabajo de los compositores utilizados, para evitar que durante su evaluación, note música que ya conocía con anterioridad.
- [7] **Documentación de resultados:** Tomando los resultados de evaluación del conjunto de expertos musicales sobre los conjuntos de piezas, de la fase 6, se hacen estadísticas de comparación para concluir sobre la calidad estética de los resultados del algoritmo. Estas estadísticas permiten documentar conclusiones sobre el desarrollo del proyecto.

En paralelo a las siete fases de la metodología y como procedimientos externos a esta, se desarrolla la documentación de tesina (el presente documento), en donde se escribe acerca de todo el proceso, fuentes, resultados y conclusiones generados a lo largo de la metodología, además de la presentación y discusión de todo ello ante investigadores relacionados en el monitoreo y evaluación del desarrollo del proyecto.

9 Desarrollo

9.1 Arquitectura del algoritmo

La arquitectura final para abordar el problema definido fue nombrada *MIA Harmfirst a1* (“*Harmony first*”, es decir, “*Armonía primero*”). Las partes de la arquitectura se muestran en la siguiente **Figura 25**.

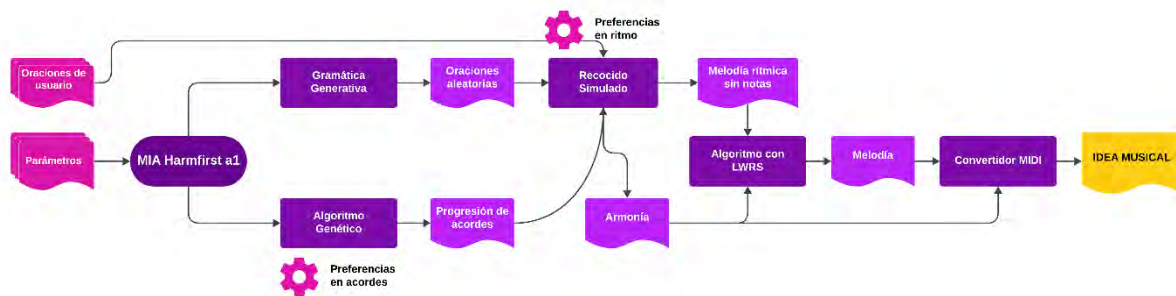


Figura 25. Arquitectura del Algoritmo MIA Harmfirst a1.

El algoritmo se separa en distintos módulos y secciones que será descritos a continuación.

9.1.a Entradas y parámetros de preferencias

El algoritmo tiene dos posibles entradas por parte del usuario (entrada de *parámetros* y entrada de *versos personalizados*), de entre ellas, cada una tiene sus distintos valores requeridos, de los cuales, los que a continuación son marcados con “(*)” son obligatorios, y el resto pueden ser elegidos aleatoriamente.

1. Parámetros: Bajo este tipo de entrada, la composición es totalmente libre:

- Cantidad de frases en la melodía (*):** Este parámetro dicta la cantidad de frases que tendrá la melodía, recordando que cada frase representa una idea única que se interpreta con una respiración. Las canciones populares suelen construirse mediante secciones denominadas, por ejemplo, los “*versos*” y “*estribillos*”, y a menudo cada uno suele estar compuesto por cuatro oraciones, por lo que un valor recomendado y **el valor por defecto es 4²⁰**. Bajo esta lógica, una idea musical sería como un verso o un estribillo, y una canción compuesta por versos y estribillos podría ser compuesta con dos ideas musicales, una para cada uno y repetidas en una estructura clásica, como lo puede ser *verso, verso, estribillo, verso, estribillo, estribillo*.
- Duración de compases de cada frase (*):** Este parámetro se da para cada una de las oraciones elegidas, en donde implica cuántos compases tendrá cada frase. Lo común y **el valor por defecto es 2** compases para cada oración.
- Acordes por cada frase (*):** Este parámetro implica la cantidad de acordes asignados para cada frase, de manera que en suma formarían la progresión de acordes para la idea musical. Su **valor por defecto es 1**.
- Patrón para cada frase (*):** Este parámetro permite decidir si cada oración es totalmente original o si en cambio repite algún componente de otra oración. Los componentes repetidos pueden ser:
 - Frase:** Si se repite este componente de alguna otra frase, entonces se utilizará la misma frase que la seleccionada para repetir. Esto significa que este patrón repetirá la cantidad de sílabas y su distribución, aunque el ritmo y los tonos serán distintos.

²⁰ Todos los valores por defecto fueron elegidos a criterio del autor, ya sea por tratar de tener un estándar basado en la música popular, o porque en la prueba y error de ejecutar el algoritmo, demostraron ser buenos valores de parámetro.

- ii. **Ritmo:** Si se repite este componente, la frase no sólo copiará a otra frase, sino el ritmo (es decir, la selección de duraciones para cada sílaba), de manera que esta nueva frase será una copia rítmica de la seleccionada.
 - e. **Forma para cada frase:** Este parámetro indicará la silueta musical que tendrá cada frase:
 - i. **Ascendente:** Los tonos de la frase mayormente tienen una dirección hacia arriba.
 - ii. **Descendente:** Los tonos de la frase mayormente tienen una dirección hacia abajo.
 - iii. **Arco:** Los tonos mayormente siguen la forma de, ascender hasta un pico y descender.
 - iv. **Valle:** Los tonos mayormente siguen la forma de, descender hasta un pico y ascender.
 - f. **Tipo de oración para cada frase:** Al generar cada oración, se tiene 4 tipos disponibles (no hacen mucha diferencia en cuanto al resultado, más allá de qué tan extensas sean):
 - i. **Oración simple declarativa.**
 - ii. **Oración simple interrogativa.**
 - iii. **Oración simple exclamativa.**
 - iv. **Oración simple Imperativa.**
 - g. **Tonalidad (nota):** El nombre de la nota que será la tonalidad, entre las posibles opciones: C, C# (Db), D, D# (Eb), E, F, F# (Gb), G, G# (Ab) A, A# (Bb), B. Su **valor por defecto es C**.
 - h. **Tonalidad (escala):** El tipo de escala de la tonalidad: *Mayor, Menor, Menor melódica, Menor armónica, Dórica, Frigia, Lidia, Mixolidia, Lócrica*. El **valor por defecto es mayor**.
 - i. **Indicador de compás:** El indicador de compás de la pieza, separado en dos valores para el numerador y denominador. El **valor por defecto es (4, 4)**.
 - j. **Anacrusa:** La anacrusa es la cantidad de tiempos que la melodía comenzará antes que la armonía. Por ejemplo, si la anacrusa es 2, en un compás de 4/4, la melodía comenzará 2/4 antes que la armonía. Su **valor por defecto es 0**.
 - k. **Octava de la armonía:** Número de octava en que comenzarán los acordes de la armonía. Su **valor por defecto es 3**.
 - l. **Octava de la melodía:** Número de octava en que comenzará la melodía. **Por defecto es 4**.
 - m. **Tempo:** Cantidad de *beats por minuto* que tendrá la pieza. Su **valor por defecto es 100**.
 - n. **Afinación:** El valor de frecuencia asignado a la nota A4, **por defecto es 440**.
2. **Versos personalizados:** Bajo este tipo de entrada, el usuario tiene la posibilidad de utilizar sus propias oraciones, de manera que estas serán musicalizadas a lo largo del algoritmo. Se utilizarán igualmente los parámetros anteriores, todos excepto **a. Cantidad de frases en la melodía** y **d. Patrón de cada frase**.

Las entradas no son la única manera en que el usuario puede ajustar el resultado de la pieza musical. También posee parámetros para los distintos módulos del algoritmo.

1. **Preferencias de ritmo:** Estos parámetros permiten ajustar las elecciones en las duraciones de las notas en la melodía, configurando lo siguiente:
 - a. **Recocido simulado (Temperatura inicial):** Parámetro del algoritmo de *Recocido Simulado*, este así como el resto de los parámetros de algoritmos metaheurísticos no es necesario que se modifiquen **su valor por defecto es 100**.
 - b. **Recocido simulado (Tasa de enfriamiento):** Qué tanto se reduce la temperatura en cada iteración del Recocido. Su **valor por defecto es 0.98** para un enfriamiento lento.
 - c. **Recocido simulado (Iteraciones):** Cantidad máxima de iteraciones en el algoritmo.
 - d. **Sesgo de selección:** Su valor va entre 1 y 10, y mientras más grande sea, más preferencia tendrá el algoritmo por figuras de duración larga. Si se quiere que se tengan más figuras rápidas, el valor apropiado sería 1, pero si se quiere mayormente figuras largas: 10. Su **valor por defecto es 5**.
 - e. **Probabilidad de punto:** Esta es una probabilidad entre 0 y 100 de que aparezcan puntos de aumento de duración en las figuras de la melodía. Su **valor por defecto es 20%**.
 - f. **Importancia de tamaño perfecto:** Valor entre -100 y 100 que implica cuánta importancia se quiere en que la frase dure exactamente la cantidad de compases disponibles. Un valor de -100 implica que se quiere todo lo contrario a que ajuste su tamaño; un valor de 0 implica que el

parámetro no tendrá importancia y un valor de 100 implica que se tiene absoluta importancia a que el tamaño se ajuste correctamente. Su **valor por defecto es 100**.

- g. **Importancia de tónicas en tiempos fuertes:** Entre -100 y 100, implica qué tanto se quiere que la melodía respete las reglas de la prosodia de ajustar las sílabas tónicas a los tiempos fuertes. Un valor de -100 implica ignorar totalmente la prosodia, un valor de 0 implica que este parámetro no tendrá impacto y un valor de 100 implica total importancia. Su **valor por defecto es 100**.
- h. **Importancia de existencia de silencio inicial:** Este parámetro entre -100 y 100 implica qué tanto se valoraría que las frases tengan un silencio antes de comenzar a sonar. Con -100 se espera que no haya silencio, con 0 no implica nada y con 100 implica que se prefieren silencios largos. Su **valor por defecto es 10**.
- i. **Penalización de silencio inicial con anacrusa:** Este parámetro entre -100 y 100 es un valor de penalización en el que, ya teniendo una anacrusa en la pieza, aun así se coloque un silencio inicial. Un valor de -100 evitará que se usen silencios al iniciar la melodía si ya existe una anacrusa. Un valor de 0 no implicará nada y un valor de 100 implicará que aún con anacrusa se colocarán silencios. Si no se tiene anacrusa, este parámetro no se aplicará. Su **valor por defecto es -100**.
- j. **Importancia de silencio al final:** Este parámetro entre -100 y 100 permitirá favorecer que las frases tengan un silencio de descanso al final. Un valor de -100 evitará silencios al final de cada frase, un valor de 0 no afectará y un valor de 100 maximizará el tamaño de los silencios al final de cada frase. Su **valor por defecto es 20**.
- k. **Importancia de notas finales largas:** Este parámetro entre -100 y 100 maximizará que la nota final de cada frase sea larga, en un intento por imitar la forma en que se suelen alargar en la música popular. Un valor de -100 evitará que sean notas largas, un valor de 0 no hará diferencia y un valor de 100 maximizará la duración de la nota final de cada frase. Su **valor por defecto es 40**.
- l. **Probabilidad de encontrar silencio inicial:** Entre 0 y 100, implica qué tan probable es que la frase tenga silencios al inicio. Su **valor por defecto es 20**.
- m. **Probabilidad de encontrar silencio final:** Entre 0 y 100. Qué tan probable es que se usen silencios al final de cada frase. El **valor por defecto es 20**.
- n. **Uniformidad:** Valor entre 0 y 100 que implica qué tan iguales serán las duraciones de las notas en la frase. Si la uniformidad es 100, se favorecerán ritmos repetitivos, y si es 0, no se propiciarán patrones de similitud en las duraciones de las notas de las frases (al menos no intencionalmente).

2. Preferencias de armonía: Estos parámetros permiten ajustar las preferencias en la armonía.

- a. **Algoritmo Genético (Tamaño de población):** Este parámetro es específico del *Algoritmo Genético*, implica cuál será la cantidad de progresiones de acordes a generar para el proceso evolutivo. Mientras más grande sea, más lento será el algoritmo. Un tamaño **por defecto es 50**. Este valor es recomendable para no sobre explotar las reglas de composición.
- b. **Algoritmo Genético (Generaciones por acorde):** Este parámetro indica la cantidad de iteraciones generacionales en el método del Algoritmo Genético; está ajustado de manera que la cantidad de iteraciones dependa de la cantidad de acordes (elementos en la secuencia de cada individuo). Su **valor por defecto es 12**, lo que implica que si una progresión tiene 2 acordes, entonces hará 24 iteraciones; si tiene 4 acordes, entonces 48 iteraciones. Mientras más grande sea este valor, los resultados serán siempre más similares, se sobreexplotarán las reglas de composición en armonía y se utilizará más cómputo (tardará más).
- c. **Algoritmo Genético (Generaciones por acorde en “no reinicio”):** Una funcionalidad del Algoritmo Genético que compone armonías se implementó para que, en caso de que se quiera crear la variación de una progresión ya existente y dada por el mismo Algoritmo Genético, es decir, que la población no se reinicie, sino que se siga iterando sobre la población ya antes iterada. Este valor de generaciones se aplicará cuando se quiera hacer este no-reinicio, por lo

que si se quiere crear una variación de una progresión ya creada por un Genético, entonces una buena manera es aplicar iteraciones (menos que las de antes) para no cambiar tanto la progresión. Su **valor por defecto es 8**.

- d. **Algoritmo Genético (Probabilidad de mutación):** Entre 0 y 100. Implica qué tan recurrentes serán las mutaciones en el Algoritmo Genético. Un valor alto implicará que se explorará mucho el espacio de posibles armonías, de manera que en cada iteración cambiarán mucho las soluciones. Un valor bajo en cambio hará que no se explore mucho y que se exploten más las buenas soluciones encontradas, por lo que en las iteraciones cambiarán menos las soluciones. Su **valor por defecto es 10%**.
- e. **Algoritmo Genético (Tamaño de torneo):** Como se revisará en el módulo de Algoritmo Genético, se utilizó un método de selección de procreadores llamado “Torneo”, y este parámetro indica la cantidad de miembros en cada torneo. Su **valor por defecto es 5**.
- f. **Algoritmo Genético (Tamaño de elitismo):** La cantidad de los mejores individuos de cada generación en el AG que se conservará para pasarse a la nueva generación. El **valor por defecto es 3**.
- g. **Complejidad de acordes:** Valor entre 0 y 100, mientras más alto sea, mayor probabilidad habrá de tener acordes extendidos (es decir, de más de 3 notas), por lo que se verán más acordes complejos. Este parámetro podría ajustar el género de la pieza a algo más parecido al Jazz, que posee alta complejidad en sus armonías. Su **valor por defecto es 20**.
- h. **Preferencias de puntuación:** Todos estos parámetros son números de puntos que calificarán como mejores o peores a las armonías. Aunque su rango no está limitado, sería recomendable valores entre 0 y 100. Valores negativos harán que la característica se penalice.
 - i. **El primer acorde es Tónica:** Puntos que se darán si el primer acorde en la progresión es el acorde de tónica. **Por defecto: 2**.
 - ii. **El primer acorde no es Tónica (penalización):** Este parámetro si es positivo, será una penalización en que el primer acorde sea tónica. **Por defecto: 0**.
 - iii. **El último acorde es Dominante:** Si el último acorde es el Dominante de la tonalidad. **Por defecto: 0**.
 - iv. **El primer acorde es Tónica y el último es Dominante:** Si ambas condiciones se tienen. **Por defecto: 5**.
 - v. **El último acorde es Tónica:** Si el último acorde de la progresión es Tónica. **Por defecto: 2**.
 - vi. **Puntos por cada Dominante Secundaria:** Se premiará por cada Dominante Secundaria que se logre. **Por defecto: 10**.
 - vii. **Cadencia completa o Cadencia Plagal:** Si existe la presencia de alguna cadencia completa o alguna cadencia plagal, se premiará. **Por defecto: 30**.
 - viii. **Pertenencia a la escala:** Por cada acorde en la progresión que sí pertenezca a la escala, se premiará. **Por defecto: 50**.
 - ix. **Puntaje por diversidad:** Si este valor es alto, se propiciará que no haya mucha repetición de acordes. Si es bajo, no se perjudicará la repetición. **Por defecto: 40**.
 - x. **Puntaje por direccionamiento de voces:** Por cada indicio de direccionamiento de voces en las notas de los acordes, se premiará. **Por defecto: 10**.
 - xi. **Puntaje por resolución suspendida:** Se premiará específicamente cuando se encuentre una secuencia de dos acordes en la que el primer acorde es suspendido (*sus2* o *sus4*) y el siguiente es un acorde mayor con la misma nota de raíz. Por ejemplo, *Dsus4* antes de un *D*). **Por defecto: 10**.
 - xii. **No resolución suspendida:** Este valor, de ser positivo representa la penalización que se hará si se tiene un acorde suspendido (*sus2* o *sus4*) y no se resuelve a un acorde mayor de su misma raíz.
 - xiii. **Penalizador de complejidad:** Un parámetro extra para penalizar (si el valor es positivo) la presencia de acordes con muchas notas. **Por defecto: 17**.
 - xiv. **Tensión – Resolución:** Premio cuando hay *Tensión-Resolución*. **Por defecto: 20**.

- xv. **Resolución de séptima (leading tone):** Premio a la presencia de resolución utilizando el grado VII de la escala. Es decir, VII y luego I. **Por defecto: 5.**
- xvi. **Presencia de Cadencia Engañosa:** **Por defecto: 17.**
- xvii. **Presencia de Acorde 6/4 cadencial:** **Por defecto: 10.**
- xviii. **Presencia de Semi-cadencia:** **Por defecto: 4.**
- xix. **Presencia de Tercera de Picardía:** **Por defecto: 6.**

9.1.b Módulo de Frases Melódicas: Gramáticas Generativas

Este módulo toma como entrada la cantidad de frases a generar, además de los parámetros para las frases y del *Recocido Simulado* (9.1.a Entradas y parámetros de preferencias). El módulo genera aleatoriamente una oración del tipo especificado utilizando la *Gramática Generativa* respectiva para cada tipo de frase y arroja cada una en su formato de *Categorías gramaticales*, es decir, las Gramáticas generan una oración, pero en lugar de utilizar palabras, utiliza el tipo de palabra (Categoría Gramatical) y posteriormente reemplaza cada tipo de palabra con una palabra real del diccionario, generando así una oración aleatoria.

Este módulo utiliza un diccionario de palabras tomado de Carlos Fenollosa (2019). Este autor recopiló una considerable cantidad de palabras del español (1,110,478 palabras y 678,831 únicas) y las clasificó según su tipo, incluyendo también para cada palabra, su separación silábica y su número de sílaba tónica. Este diccionario se modificó para corregir errores en el número de tónica y se separó en archivos individuales para cada categoría de palabra, de modo que al reemplazar cada palabra en el resultado de la Gramática Generativa se consigue una oración aleatoria con una sintaxis definida, así como la separación de sílabas y la distribución de tónicas y átonas.

Las reglas de producción de las Gramáticas Generativas se construyeron sin consideraciones formales de alguna fuente que estudie la sintaxis española, sino como un mero análisis de las estructuras en diversas oraciones simples de cada tipo. Las reglas de producción se definen a continuación.

Para oraciones simples declarativas:

- S → Sujeto Predicado.
- Sujeto → Det Sus | PronPer | PronDem Sus.
- Predicado → Verbo | Verbo Objeto.
- Verbo → VTran | VIntra | VConj | VCop.
- VTran → “V,singular,regular,transitivo_intransitivo,conjugado,presente_indicativo,él”.
- VTran → “V,plural,irregular,transitivo_intransitivo_pronominal,conjugado,pretérito_indicativo,yo”.
- VIntra → “V,singular,regular,transitivo_intransitivo,conjugado,presente_indicativo,tú”.
- VIntra → “V,plural,irregular,transitivo_intransitivo,conjugado,pretérito_indicativo,ustedes”.
- VConj → “V,singular,regular,transitivo_intransitivo,conjugado,presente_indicativo,vos”.
- VConj → “V,plural,irregular,transitivo_intransitivo,conjugado,presente_indicativo,nosotros”.
- VCop → “V,singular,regular,transitivo_intransitivo,conjugado,presente_indicativo,él”.
- VCop → “V,plural,irregular,transitivo_intransitivo,conjugado,pretérito_indicativo,ellos”.
- Objeto → Det Sus | Sus.
- Det → “Det”.
- Sus → “Sus”.
- PronPer → “PronPer”.
- PronDem → “PronDem”.

Para las oraciones simples interrogativas:

- S → PartInterr Sujeto Verbo | PartInterr Verbo.
- PartInterr → PronIntExc | AdvInt.
- Sujeto → PronPer | PronDem | Sus.
- Verbo → VTran | VIntra | VCop | VConj.

- VTran → “V,singular,regular,transitivo_intransitivo,conjugado,presente_indicativo,él”.
- VTran → “V,plural,irregular,transitivo_intransitivo_pronominal,conjugado,pretérito_indicativo,yo”.
- VIntra → “V,singular,regular,transitivo_intransitivo,conjugado,presente_indicativo,tú”.
- VIntra → “V,plural,irregular,transitivo_intransitivo,conjugado,pretérito_indicativo,ustedes”.
- VCop → “V,singular,regular,transitivo_intransitivo,conjugado,presente_indicativo,él”.
- VCop → “V,plural,irregular,transitivo_intransitivo,conjugado,pretérito_indicativo,ellos”.
- VConj → “V,singular,regular,transitivo_intransitivo,conjugado,presente_indicativo,vos”.
- VConj → “V,plural,irregular,transitivo_intransitivo,conjugado,presente_indicativo,nosotros”.
- PronIntExc → “PronIntExc”.
- AdvInt → “PronIntExc”.
- Det → “Det”.
- Sus → “Sus”.
- PronPer → “PronPer”.
- PronDem → “PronDem”.

Para oraciones simples exclamativas:

- S → Int Sujeto Verbo | Int Verbo.
- Int → “Int”.
- Sujeto → Det Sus | PronPer | PronDem Sus.
- Verbo → VTran | VIntra | VCop | VConj.
- VTran → “V,singular,regular,transitivo_intransitivo,conjugado,presente_indicativo,él”.
- VTran → “V,plural,irregular,transitivo_intransitivo_pronominal,conjugado,pretérito_indicativo,yo”.
- VIntra → “V,singular,regular,transitivo_intransitivo,conjugado,presente_indicativo,tú”.
- VIntra → “V,plural,irregular,transitivo_intransitivo,conjugado,pretérito_indicativo,ustedes”.
- VCop → “V,singular,regular,transitivo_intransitivo,conjugado,presente_indicativo,él”.
- VCop → “V,plural,irregular,transitivo_intransitivo,conjugado,pretérito_indicativo,ellos”.
- VConj → “V,singular,regular,transitivo_intransitivo,conjugado,presente_indicativo,vos”.
- VConj → “V,plural,irregular,transitivo_intransitivo,conjugado,presente_indicativo,nosotros”.
- Det → “Det”.
- Sus → “Sus”.
- PronPer → “PronPer”.
- PronDem → “PronDem”.

Para oraciones simples imperativas:

- S → VerboImp | VerboImp Objeto.
- VerboImp → VImp.
- VImp → “V,singular,regular,neutro,conjugado,imperativo,tú”.
- VImp → “V,singular,irregular,neutro,conjugado,imperativo,tú”.
- VImp → “V,plural,irregular,neutro,conjugado,imperativo,ustedes”.
- VImp → “V,plural,regular,neutro,conjugado,imperativo,ustedes”.
- Objeto → Det Sus | Sus.
- Det → “Det”.
- Sus → “Sus”.

Ya implementada la lógica del módulo en código de programación, junto al método que sustituye la oración de categorías gramaticales por palabras reales del español, se pueden extraer resultados como los siguientes.

Para oraciones simples declarativas:

- “*Cuántos taloneras cande estocásticas*”.
- “*Vosotras cotizáremos*”.

- *"Nosotras segmenta mis matadas"*.
- *"Tú ejabriremos"*.
- *"Cuántas polvoraduques trascole nuestras segundas"*.
- *"Ella engavillase"*.
- *"Esas empalmaduras rehogar"*.
- *"Ese cordera astilla sobresaliente"*.
- *"Él enrobina"*.
- *"Esos tractos desacatara"*.

Para oraciones simples interrogativas:

- *"Cuánta placeres soliloquiaren"*.
- *"Cuáles aquellos escaramuzarían"*.
- *"Cuántas nosotras bribarás"*.
- *"Cuáles emparienten"*.
- *"Cuál yo trisquemos"*.
- *"Cuáles desmandases"*.
- *"Qué percutía"*.
- *"Cuántos atajaste"*.
- *"Cuáles engrasaras"*.
- *"Quién empique"*.

Para oraciones simples exclamativas:

- *"Puf ociarían"*.
- *"Vaya zorreasen"*.
- *"Anda manares"*.
- *"Uy aquellas huroneras chingares"*.
- *"Vaya yo cerraría"*.
- *"Cielos intributen"*.
- *"Olé buzoneá"*.
- *"Anda esos vedados desmechaseis"*.
- *"Caramba arreglen"*.
- *"Uf una roble enconchases"*.

Para oraciones simples imperativas:

- *"Amesurases"*.
- *"Calcaréis la países"*.
- *"Respingaron"*.
- *"Automedicaba nuestra área"*.
- *"Caldearen"*.
- *"Politiquearen"*.
- *"Enzurronasen tus bujedos"*.
- *"Coopere altoparlantes"*.
- *"Manufactura todos aglomeramientos"*.
- *"Ponerían vuestra emergente"*.

Debido a que no se hizo un estudio riguroso de las reglas de producción para este tipo de oraciones en el español, los resultados pueden resultar incoherentes. Sin embargo, satisfacen el propósito de este trabajo, pues el objetivo es generar una frase finita de palabras y con una distribución única de sílabas tónicas, átonas y espacios entre las palabras. Por ejemplo; obteniendo *"Vaya yo cerraría"* como oración y separándola por sus sílabas tónicas y átonas como: *"Va*-ya yo* ce-rra-rí*-a"*.

9.1.c Módulo de Ritmo: Recocido Simulado

Ya teniendo una oración, su separación silábica y su identificación de sílabas tónicas y átonas, el módulo de Recocido Simulado ejecuta la Metaheurística tomando también como entrada el número correspondiente de “Duración de compases de cada frase”, de manera que este cumple con el objetivo de asignar a cada sílaba una figura de duración (consultar **Figura 2**), obteniendo una frase melódica (de cuyas notas sólo se define su duración y no su tono) del tamaño esperado de compases y cuyas notas (sílabas) maximizan una función objetivo que favorece en mayor o menor medida las reglas de prosodia especificadas (consultar **6.2.b.2 Prosodia**).

Al comenzar el proceso, así como el algoritmo de Recocido Simulado (consultar **6.2.d.1 Recocido simulado**), este módulo necesita crear una “solución inicial”. El paso de inicialización consiste en asignar a cada sílaba una duración basada en las figuras rítmicas disponibles:

1. Redonda.
2. Blanca.
3. Negra.
4. Corchea.
5. Semi-corchea.
6. Triplete de negra²¹.
7. Triplete de blanca.
8. Fusa.
9. Triplete de corchea.
10. Semi-fusa.

El proceso de asignar una figura de tiempo a cada sílaba se realiza de manera aleatoria utilizando un algoritmo de selección similar a LWRS llamado “Selección Aleatoria Ponderada Exponencialmente” (“Exponential Random Weighted Selection”: EWRS), autoría de Ramos Herrera (2025b). Este algoritmo, al igual que LWRS, asigna mayor probabilidad de selección a los elementos disponibles, conforme a su orden en la lista, pero a diferencia de este, EWRS aplica un sesgo mayor o menor con respecto a un parámetro “base”, que su autor define entre 1 y 10. Con un valor “base” de 1, la diferencia de probabilidad entre los elementos se distingue, pero no en la misma magnitud que para un valor de 10, donde el sesgo basado en el orden es mucho más marcado.

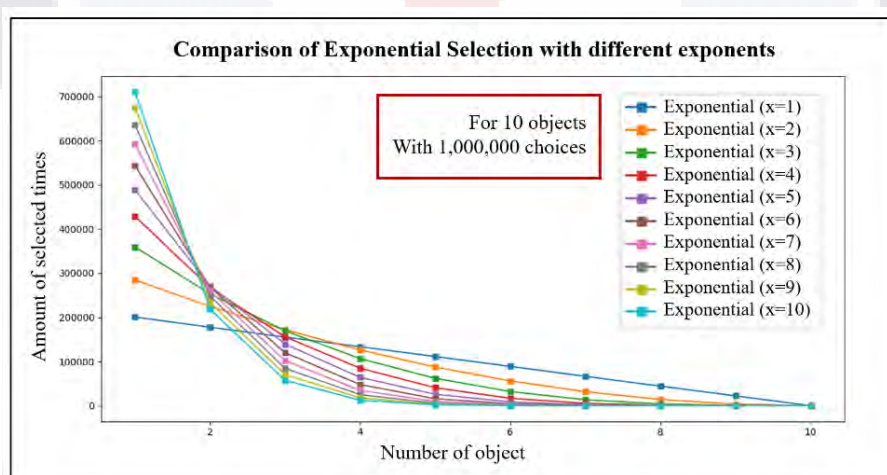


Figura 26. Preferencias de selección en el algoritmo EWRS con valores x entre 1 y 10, para 10 objetos.

Nota: Ilustración extraída del artículo de Ramos Herrera (2025b).

²¹ Los tripletes son estructuras en las que se determina que tres notas tendrán la duración de dos figuras de duración mayor inmediata. Por ejemplo, un triplete de corchea repartirá el espacio de dos notas negras (siendo la negra la nota de mayor duración más inmediata) entre tres tiempos, de manera que al tener una división ternaria, se obtienen ritmos más variados que los posibles con las figuras disponibles descritas en **6.1.a Conceptos básicos**.

La **Figura 26** muestra el nivel de sesgo en selecciones de prueba con una lista de 10 elementos y para los valores de *base* (“*x*”) entre 1 y 10. Se distingue que, por ejemplo, con un valor de $x = 1$, el primer elemento de la lista recibe aproximadamente el 20% de las selecciones (200,000 de 1,000,000) y así como el segundo elemento un 17%, el tercero un 15%, el cuarto un 13%, etc. En contraposición, con un valor de $x = 10$, la probabilidad para el primer elemento es alrededor del 71%, del segundo 22%, del tercero 5%, etc.

EWRS garantiza que en este primer paso de creación de una solución inicial, se dará preferencia a las figuras con respecto al orden en que se acomodan, siendo el orden por defecto el antes descrito en este apartado, en donde las notas se ordenan dando preferencia conforme a más largas sean estas²². Esta selección ponderada parte de la hipótesis del autor de que, en la música común suele ser más común el uso de notas largas y muy raro el uso de notas rápidas y cortas. A pesar del sesgo en la velocidad de las notas, más parámetros como el *tempo* pueden definir el carácter rítmico de la pieza, además, siendo que esta es una solución inicial, no se cae en el riesgo de limitar la pieza al uso de figuras de larga duración, pues aún se aplicará un proceso iterativo que permitirá la aparición de más figuras y más posibles resultados de ritmo. El parámetro descrito en **9.1.a Entradas y parámetros de preferencias**, “*Sesgo de selección*”, permite al usuario definir dicho sesgo.

El proceso inicial de asignar figuras de duración a cada sílaba de la oración también contempla tres pasos previos: 1. Aplicar un silencio antes de comenzar la frase (que depende del parámetro “*Probabilidad de encontrar silencio inicial*”), 2. Aplicar un silencio al terminar la frase (que depende del parámetro “*Probabilidad de encontrar silencio final*”) y 3. Aplicar o no un punto de aumento de duración a cada nota (dependiente de “*Probabilidad de punto*”). Estas decisiones se toman de manera aleatoria respetando la probabilidad elegida por el usuario para cada caso, creando en conjunto con lo anterior, una solución inicial conformada de una serie de figuras rítmicas elegidas para los silencios inicial y final (si existen) y para las sílabas de la oración de entrada que se convertirá a frase melódica. Esta frase inicial generada posee la característica importante de su duración (la suma de las duraciones de todos sus elementos), siendo este un valor que se debe ajustar al tamaño de compases definido, maximizando las reglas definidas en la función objetivo.

A través del proceso iterativo, la función objetivo computa un puntaje para cada solución. Dicha función objetivo puede definirse de la siguiente manera; siendo:

- $D = \{d_0, d_1, \dots, d_n, d_{n+1}\}$: la secuencia de duraciones (en unidad de Tiempos), de cada nota sílaba, siendo d_0 la duración del silencio inicial, d_{n+1} la duración del silencio final y $d_1 \dots d_n$ las duraciones de cada nota sílaba. Cada d_i contiene la duración extra si la nota posee punto.
- T : el espacio total disponible según la cantidad de compases.
- $S = \sum_{i=0}^{n+1} d_i$: la suma de todas las duraciones en notas.
- τ : siendo el conjunto donde cada $t \in \{1, \dots, n\}$ es una sílaba tónica.
- Para cada sílaba tónica t , siendo $r_t \in [0, 1]$ el *factor de alineación al pulso* (siendo 1 si está perfectamente sobre un tiempo fuerte y decayendo linealmente hasta 0 con un límite de tolerancia, y siendo $b_t \in \{10, 1, 0\}$ un multiplicador que es 010 cuando es un tiempo *fuerte*, 1 si está en un tiempo *semi-fuerte* y 0 en otro caso).
- Siendo $U = \sum_{i=1}^{n-1} [1d_i = d_{i+1} - 1d_i \neq d_{i+1}]$ el conteo de uniformidad (+1 por cada figura vecino-repetida, y -1 en otro caso).
- Siendo d_{max} la duración máxima posible permitida en unidades de tiempo (96, para una redonda de 64 tiempos con punto, es decir con 32 tiempos extra).
- Siendo d_{last} la duración de la última sílaba.
- Teniendo además el vector de pesos de importancias definidos en los parámetros: $p = (p_{fit}, p_{beat}, p_{anac}, p_{init}, p_{final}, p_{last})$, siendo estas las importancias de; el correcto ajuste de las sílabas en el tamaño, tónicas en tiempos fuertes, importancia de silencios iniciales en anacrusis,

²² Con algunas excepciones en las que se elevan en la lista, figuras más comunes como el Triplete de negra y de corchea.

importancia de silencios inicial y final y la importancia de una última sílaba larga, siendo cada $p \in [-100, 100]$, y siendo el Score final definido como:

$$Score(D) = -|S - T| * \frac{p_{fit}}{100} + \sum_{t \in \tau} (r_t b_t) * \frac{p_{beat}}{100} - \underbrace{\frac{d_0}{d_{max}}}_{\text{Silencio inicial}} * \frac{p_{anac}}{100} + \underbrace{\frac{d_0}{d_{max}}}_{\text{Silencio inicial fraccional}} * p_{init} \\ + \underbrace{\frac{d_{n+1}}{d_{max}}}_{\text{Silencio final fraccional}} * p_{final} + U * \frac{100 - p_{uni}}{100} + \underbrace{\frac{d_{last}}{d_{max}}}_{\text{Nota final fraccional}} * p_{last} \quad (8)$$

En resumen, la función objetivo del Recocido Simulado computa una suma cuyos términos evalúan cada uno a un factor estético en el ritmo de las sílabas, conforme a los parámetros p , cada término de estos está multiplicado por un factor de importancia entre -100 y 100, que premia o castiga el resultado de puntaje, según como le configure el usuario. Recordando nuevamente, los términos de dicha suma son: 1. Importancia de ajuste correcto, 2. Cantidad de sílabas tónicas en tiempos fuertes, 3. Penalización por silencios en anacrusas, 4. Duración del silencio inicial, 5. Duración del silencio final, 6. Duración de la última nota sílaba.

El proceso de encontrar una solución vecina es simple. En cada iteración se toma una sílaba al azar y se modifica su duración, cambiando la figura rítmica y/o el punto en la duración, de manera que estas pequeñas modificaciones propician la exploración, sobre la cuál, después de cierta cantidad de iteraciones, y de su respectivo enfriamiento, se consigue una combinación de figuras rítmicas que favorezcan puntajes altos según la configuración de función objetivo deseada por el usuario. De esta manera, si la entrada del módulo es una oración separada por sílabas tónicas y átonas, por ejemplo: “Va*-ya yo* ce-rra-rí*-a” (“Vaya yo cerraría”), y otra entrada es una cantidad compases en los que ajustar la oración, por ejemplo, dos compases de 4/4, la salida será una configuración de duración de sílabas que, (usando como ejemplo los parámetros por defecto de las preferencias de ritmo) se justarán a los compases maximizando sobre reglas de prosodia y ritmo.

Una salida posible del módulo se ilustra en la **Figura 27**. Dada la frase de ejemplo (“Vaya yo cerraría”), se muestra una posible configuración de figuras rítmicas para cada sílaba, que ajusta de manera que las tónicas caen en tiempos fuertes del compás, ajustando sus duraciones para caber en dos compases de 4/4, sin silencio inicial, con una nota final larga y un pequeño silencio al terminar la frase.



Figura 27. Posible salida de oración convertida en frase rítmica mediante Recocido Simulado.

Adicional a lo anterior descrito, el módulo se complementa con dos funciones cuyo fin es mejorar los resultados de ajuste rítmico. En primer lugar, se implementó un algoritmo de *Greedy Coin-change* (“Algoritmo glotón para cambio en monedas”)²³ para, en casos en los cuales incluso la solución final exceda el tamaño al que debe ajustarse. Este algoritmo reduce la duración del último sonido para no sobrepasar el límite disponible de compases y reparte el silencio sobrante en figuras, para forzar el ajuste del tamaño de las sílabas. En segundo lugar, se utiliza un parámetro extra “Mejor ritmo entre”. Este parámetro indica la cantidad de Recocidos Simulados distintos e independientes que se aplicarán para la misma oración, de manera que se elige el resultado con mejor puntaje entre los resultados finales de cada Recocido. Dado que esta Metaheurística es eficiente en

²³ Entiéndase simplemente como un algoritmo que calcula la cantidad mínima de monedas para ajustar una cantidad determinada (cambio), aplicada a figuras de duración musical.

tiempo de ejecución, es posible ejecutar varios de estos sin que esto represente un aumento drástico en el tiempo de generación. Un **valor por defecto** para “Mejor ritmo entre” puede ser **20**.

9.1.d Módulo de Armonía: Algoritmo Genético

El módulo de Algoritmo Genético puede ejecutarse en paralelo al Recocido Simulado, antes o después del mismo, pues sus procesos son independientes entre sí. Para la arquitectura *Harmfirst a1* la melodía se construye con base en los acordes, pero sólo los tonos en la melodía, no las duraciones de sus notas.

La entrada de este módulo son los parámetros de configuración y las entradas del algoritmo. Específicamente se necesita: *Cantidad de frases en la melodía*, *Acordes por cada frase* y *Duración de compases de cada frase*. Sabiendo la cantidad de frases en la melodía y la cantidad de acordes para cada frase, se puede saber la cantidad de acordes que tendrá la Progresión en la Idea Musical a generar. Por ejemplo, queriendo cuatro frases y dos acordes por cada frase, se sabe que la pieza tendrá ocho acordes. Sabiendo además la cantidad de compases de cada frase se pueden decidir las duraciones de cada acorde. Se utiliza un Recocido Simulado para saber qué duración tendrá cada acorde según los compases de sus respectivas frases, con diferencia que este Recocido tiene todos sus parámetros ajustados a un valor por defecto para garantizar alta uniformidad y el ajuste perfecto de sus tiempos, de modo que, si se tiene una progresión de ocho acordes, se construye una oración de ocho palabras (a fines prácticos no importa la palabra, sólo que esté separada por espacios) y se considera que cada palabra es de una sílaba tónica. Por ejemplo: “A* A* A* A* A* A* A* A*”. Los parámetros de Recocido para este escenario de acordes son por defecto:

- **Sesgo de selección:** 2.
- **Importancia de tamaño perfecto:** 100.
- **Importancia de tónicas en tiempos fuertes:** 100.
- **Importancia de existencia de silencio inicial:** -100.
- **Penalización de silencio inicial con anacrusa:** -100.
- **Importancia de silencio al final:** -100.
- **Importancia de notas finales largas:** 0.
- **Probabilidad de encontrar silencio inicial:** 0.
- **Probabilidad de encontrar silencio final:** 0.
- **Uniformidad:** 80.

Estos ajustes garantizan una armonía que no sea compleja en su ritmo, sino que simplemente rellena el espacio de manera mayormente uniforme. Además, el sesgo de selección es bajo, para que, aun teniendo alta uniformidad exista espacio a ligeras variaciones en las figuras seleccionadas para la armonía.

Si bien esta sección utiliza el Módulo de Recocido Simulado, el fuerte de su funcionamiento es independiente de cualquier otro módulo. El centro es el Algoritmo Genético, que toma como entrada la cantidad de acordes que conformarán la armonía y construye una de estas de manera aleatoria y única. Recordando el proceso del Algoritmo Genético (consultar **6.2.c.1 Algoritmo Genético**), el primer paso es construir una población inicial de soluciones, completamente aleatoria. Esta población es una lista de progresiones generadas aleatoriamente, y si bien sería factible generar progresiones totalmente al azar, para esta implementación se utiliza un sesgo inicial para favorecer la aparición de ciertos acordes específicos en la creación de la población inicial. Para la creación de cada individuo en la población inicial se utiliza el algoritmo LWRS con la siguiente lógica:

1. Se construye una lista de selección que contiene 12 elementos, de los cuales los 7 primeros simbolizan alguno de los acordes dentro de la tonalidad de la pieza y los últimos 5 representan acordes fuera de la tonalidad. La lista luce algo así: [“in”, “in”, “in”, “in”, “in”, “in”, “in”, “out”, “out”, “out”, “out”, “out”].
2. Para cada acorde de los n acordes a generar, y para cada individuo, se utiliza el algoritmo LWRS para seleccionar si el acorde será o no de la tonalidad. Utilizando la ecuación (7) sabe que la probabilidad

de seleccionar un acorde “in” es de 81% en este escenario, de manera que un acorde “out” tiene probabilidad de 19%²⁴.

3. Si se ha decidido que el acorde será de la escala, entonces se selecciona al azar alguno de los 7 acordes de su escala (Consultar **6.1.c.2 Círculo de quintas**); este será forzosamente una triada (mayor, menor, aumentado o disminuido).
4. Si se ha decidido que el acorde no será de la escala, entonces se elige uno al azar entre todos los posibles acordes mayores y menores con las doce notas, exceptuando aquellos que ya estén dentro de los acordes en la tonalidad/escala.
5. Ya habiendo decidido el acorde de triada, se decide si se aumentará su complejidad. Es decir, si el acorde permanecerá en su categoría de mayor/menor/aumentado/disminuido o si se tomará uno de su mismo tipo pero con mayor cantidad de notas.
 - a. Sabiendo la categoría del acorde, se genera una lista con todos los posibles tipos de acorde de su misma categoría, entre los posibles acordes a usar (consultar **Tabla 4**).
 - b. Teniendo los acordes candidatos y estando estos ordenados comenzando por menor complejidad, se decide el tipo mediante el algoritmo CWRS (“Complexity Weighted Random Selector: “Selector Aleatorio Ponderado por Complejidad”), un nuevo algoritmo de selección creado para fines de este proyecto. Este selector (explicado más a detalle a continuación) utiliza como parámetro el valor de “complejidad” (entre 0 y 100) dado por el usuario. Si la complejidad es cercana a 0, la preferencia de selección estará dada casi totalmente hacia los primeros elementos (como un sesgo en EWRS con alto valor de *base*), y mientras más cercano a 100 sea la complejidad, la probabilidad de selección de todos los elementos será casi igual. Es decir, a baja complejidad la preferencia está a los primeros elementos, a alta complejidad la probabilidad de selección es igual para todos los elementos.
 - c. Ya sabiendo el acorde y su tipo, se decide si tendrá una inversión, utilizando nuevamente LWRS con una lista de todas las posibles inversiones ordenadas de 0 a n (según su cantidad de notas). De manera que se dará mayor preferencia a la inversión 0 (estado fundamental).

La raíz del acorde (nombre de la nota base), el tipo de acorde y el número de inversión son los tres datos esenciales para modelar cada elemento en una lista de progresión, de manera que un individuo generado en la población del Algoritmo Genético se construye de producir mediante el proceso anterior, la cantidad necesaria de acordes, y una población se generará construyendo una p cantidad de progresiones. Los tipos de acordes que se seleccionaron para el algoritmo MIA se enlistan en la **Tabla 4**.

Tabla 4. Acordes disponibles para el algoritmo MIA.

Tipo de acorde	Categoría	Descripción
Mayor	Mayor	Triada mayor
m	Menor	Triada menor
7	Mayor	Mayor con 7ma menor
maj7	Mayor	Mayor con 7ma mayor
m7	Menor	Menor con 7ma menor
mmaj7	Menor	Menor con 7ma mayor
aug	Aumentado	Triada aumentada: quinta aumentada

²⁴ Resultaría innecesario utilizar LWRS en este paso de selección, pues las probabilidades siempre serán de 81% y 19%, sin embargo, se implementó de esta manera para que en futuras modificaciones del modelo, en donde se utilicen escalas con no sólo 7 notas, se pueda crear probabilidades variables para la selección, con base en la cantidad de notas en la escala y la cantidad de notas fuera de la escala.

dim	Disminuido	Triada disminuida: quinta disminuida
sus2	Suspendido	Triada suspendida con 2da mayor
sus4	Suspendido	Triada suspendida con 4ta justa
9	Mayor	Mayor con 7ma menor y 9na mayor
maj9	Mayor	Mayor con 7ma mayor y 9na mayor
m9	Menor	Menor con 7ma menor y 9na mayor
11	Mayor	Mayor con 7ma menor, 9na mayor y 11na justa
maj11	Mayor	Mayor con 7ma mayor, 9na mayor y 11na justa
m11	Menor	Menor con 7ma menor, 9na mayor y 11na justa
13	Mayor	Mayor con 7ma menor, 9na mayor, 11na justa y 13na mayor
maj13	Mayor	Mayor con 7ma mayor, 9na mayor, 11na justa y 13na mayor
m13	Menor	Menor con 7ma menor, 9na mayor, 11na justa y 13na mayor
dim7	Disminuido	Disminuido con 7ma disminuida
m7b5	Disminuido	Semi-disminuido: triada disminuida con 7ma menor
add9	Mayor	Mayor con 9na mayor
madd9	Menor	Menor con 9na mayor
6	Mayor	Mayor con 6ta mayor
m6	Menor	Menor con 6ta mayor
aug7	Aumentado	Aumentado con 7ma menor
7b9	Mayor	Mayor con 7ma menor y 9na bemol
7#9	Mayor	Mayor con 7ma menor y 9na sostenida
7b5	Mayor	Mayor con 7ma menor y quinta bemol
7#5	Mayor	Mayor con 7ma menor y quinta sostenida
9#11	Mayor	Mayor con 7ma menor, 9na mayor y 11na sostenida

Mientras más notas posee un acorde, más alta es su complejidad. Acordes como los **maj13** utilizan incluso 7 notas. La **Tabla 4** contiene los acordes ordenados por dicha complejidad, de manera que el sistema MIA utiliza esta misma tabla cuando con CWRS necesita decidir si hará más complejo o no un acorde base, por ejemplo, Cm (Do menor). El método descrito anteriormente consiste en tomar todos los acordes con a misma categoría (Menor) y pasarlos (ya en su orden) como lista de selección a CWRS junto al parámetro “*complejidad*”. Mientras más cercana sea la complejidad a 0, más seguro será que el acorde siga siendo sólo Menor, y mientras más cercana a 100, la probabilidad de tener cualquier acorde de categoría Menor será prácticamente la misma.

El algoritmo CWRS sigue los siguientes pasos:

Algoritmo 1. Complexity Weighted Random Selector

Entradas: *elementos* (lista), *complejidad* (número entre 0 y 100)

Salida: *elemento*

```
1   $n \leftarrow \text{longitud}(\text{elementos})$ 
2   $c \leftarrow \text{complejidad} / 100$ 
3
4  Si  $c = 0$ 
5       $\text{exponente} \leftarrow n$ 
6  Sino
7       $s \leftarrow (1 - c^2) / c$ 
8
9  Para  $i$  desde 0 hasta  $n-1$ 
10      $\text{pesos}[i] \leftarrow (i + 1)^{-s}$ 
11
12   $\text{seleccionado} \leftarrow \text{aleatorio}(\text{elementos}, \text{pesos})$ 
13  Devolver  $\text{elementos}[\text{seleccionado}]$ 
```

El **Algoritmo 1** comienza transformando el valor de complejidad en un factor c normalizado entre 0 y 1, y a partir de él calcula un exponente s que controla qué tan “pendiente” es la ley de potencias usada para generar pesos $\propto (i + 1)^{-s}$. Cuando la complejidad es casi 0, s toma un valor muy alto (igual al número de elementos), de modo que el primer elemento acapara casi toda la probabilidad. A medida que la complejidad sube hacia 100, s tiende a cero y todos los pesos se igualan, resultando en una distribución prácticamente uniforme. Finalmente, se escoge un elemento al azar usando estos pesos sin necesidad de normalizarlos explícitamente. La **Figura 28** muestra las probabilidades que tendrían los elementos (en un ejemplo práctico con 10 opciones) al ser seleccionados con el algoritmo CWRS, para tres posibles valores de complejidad: 0, 20, 40, 60, 80 y 100.

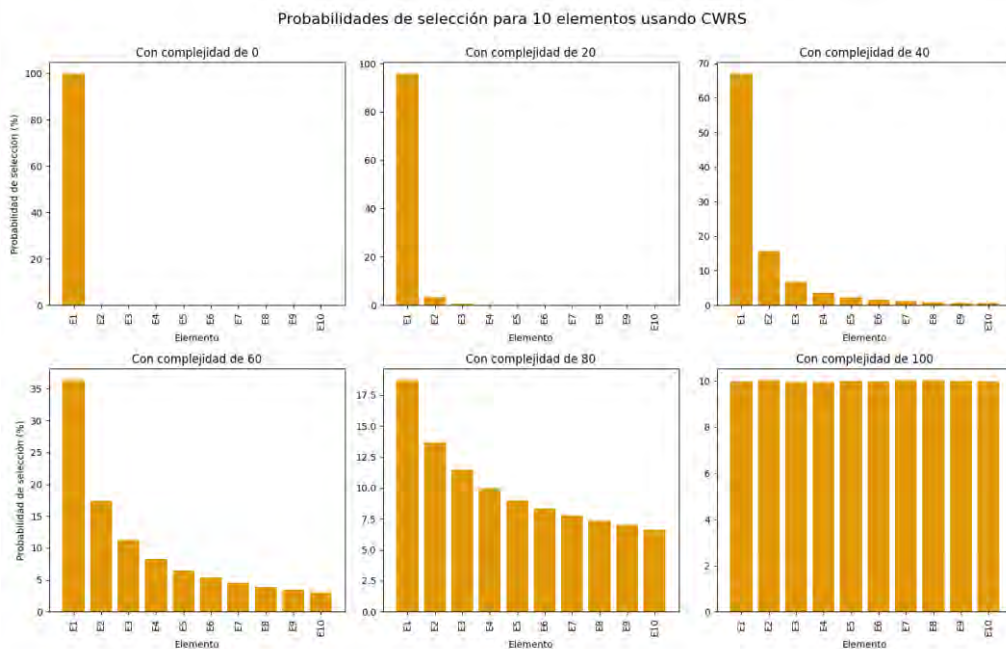


Figura 28. Probabilidades de selección para 10 elementos usando CWRS, para complejidades de 0, 20, 40, 60, 80 y 100.

Al aplicarse CWRS a la selección del tipo de acordes se logra que, al seleccionar complejidades bajas y cercanas a 0, prácticamente el primer tipo de acorde (acorde de triada) será usado y se obtendrán armonías de baja complejidad (en cantidad de notas por cada acorde), y al utilizar valores cercanos a 100, las armonías tendrán mayor cantidad de notas y por ende mayor complejidad. Este parámetro puede dirigir fuertemente el estilo y género de la pieza a generar, siendo que ciertos estilos como el Jazz utilizan altas armonías complejas.

Ya habiendo generado una población inicial, en la que cada miembro es una progresión de acordes, se siguen los pasos clásicos del Algoritmo Genético de Holland. Para el proceso de Selección se implementó el método “*Selección por Torneo*”, descrito por Goldberg et al. (1989), que consiste en definir un “*tamaño de torneo*” (con un valor por defecto de 5, según lo descrito en **9.1.a Entradas y parámetros de preferencias – Preferencias de armonía**) y separar grupos de miembros de dicha cantidad (la cantidad de veces necesarias para formar la nueva población tras el cruzamiento, y tomando aleatoriamente miembros en cada caso), con miembros de la población, y seleccionar el mejor de ellos según su puntaje de la función objetivo²⁵. Cuando se necesite aplicar el operador de mutación, se toma un acorde al azar en la solución a mutar y se reemplaza totalmente por uno nuevo, cuyas características como nota raíz, tipo de acorde e inversión se seleccionan aleatoriamente sin ningún sesgo ni distinción; es decir, la mutación introduce con la misma probabilidad cualquier acorde posible sin preferencia alguna. Finalmente el método de *cruzamiento* selecciona un punto aleatorio en las progresiones a cruzar y se mezclan nuevas progresiones con base en este punto de división.

La efectividad detrás de la creación de buenas progresiones de acordes está dada por la *función objetivo*, que se configuró considerando las reglas de composición para armonía descritas en **6.1.d.1 Para armonía**. Es función de *fitness* se calcula como una simple suma de las “*reglas musicales*” evaluadas individualmente, cada una ponderada por su parámetro de preferencia correspondiente. En concreto, se define que:

- $A = (a_1, a_2, \dots, a_n)$ es una progresión de acordes, siendo una secuencia de n acordes.
- $w = (w_1, w_2, \dots, w_k)$ siendo el vector de todos los parámetros de preferencia de puntuación, es decir, las bonificaciones y penalizaciones.
- Y siendo $\{f_j(A)\}_{j=1}^k$ el conjunto de funciones de valor real que miden cada regla en A , como por ejemplo, “*el primer acorde es tónica*”, “*el último acorde es dominante*”, “*el acorde es miembro de la escala*”, etc. (consultar **9.1.a Entradas y parámetros de preferencias – Preferencias de armonía - h. Preferencias de puntuación**).

De esta manera, la función objetivo se define como:

$$Fitness(A) = \sum_{j=1}^k w_j f_j(A) \quad (9)$$

Donde cada $f_j(A)$ es calculada de tres posibles maneras:

1. **Reglas de presencia binaria:** Por ejemplo, si el primer acorde es o no Tónica, si el último acorde es o no Dominante, si hay presencia de *Tercera de Picardía*, es decir, las reglas que se definen con la presencia o la ausencia de un componente en la progresión. Si la condición se cumple (o no) se aplica la bonificación (o penalización):

$$f_j(A) \in \{+1, -1, 0\} \quad (10)$$

2. **Reglas basadas en conteo o suma:** Por ejemplo, el número de *Dominantes secundarios*, cantidad de cadencias, cantidad resoluciones de suspendidos, etc. Se aplican la bonificación o penalización:

²⁵ Se utilizó este Selección por Torneo por ser un método popular, mas no por alguna razón medida específica.

$$f_j(A) = \sum_{i=1}^{n-1} 1_{\text{evento } j \text{ entre los acordes } i \rightarrow i+1} \quad (11)$$

3. **Medidas de proporción o promedio:** Por ejemplo, la proporción de acordes que pertenecen a la tonalidad, proporción de diversidad de acordes, el direccionamiento de voces, entre otras:

$$f_j(A) = \begin{cases} \frac{1}{n} \sum_i \frac{\# \text{tonos que son de la tonalidad}}{\# \text{tonos en el acorde}_i}, & (\text{pertenencia a la tonalidad}) \\ \vdots \\ \frac{\# \text{acordes únicos}}{n}, & (\text{diversidad}) \\ \vdots \\ \max\left(0, 1 - \frac{\text{promedio de movim de semitonos}}{6}\right), & (\text{direccionamiento de voces}) \\ \vdots \end{cases} \quad (12)$$

Colocando todas las funciones individuales, se tienen las siguientes veintiuno:

- $f_1(A) = 1_{\text{primer acorde es Tónica}}$
- $f_2(A) = -1_{\text{primer acorde no es Tónica}}$
- $f_3(A) = 1_{\text{último acorde es Dominante}}$
- $f_4(A) = -1_{\text{último acorde no es Dominante}}$
- $f_5(A) = 1_{\text{primer acorde Tónica y último Dominante}}$
- $f_6(A) = 1_{\text{último acorde es Tónica}}$
- $f_7(A) = \sum 1_{\text{dominante secundario}}$
- $f_8(A) = \sum 1_{\text{cadencia V} \rightarrow \text{I or IV} \rightarrow \text{I}}$
- $f_9(A) = \frac{1}{n} \sum \frac{\# \text{tonos que son de la tonalidad}}{\# \text{tonos en el acorde}_i}$
- $f_{10}(A) = \frac{\# \text{acordes únicos}}{n}$
- $f_{11}(A) = \max\left(0, 1 - \frac{\text{promedio de movim de semitonos}}{6}\right)$
- $f_{12}(A) = \sum_{i=1}^n \begin{cases} -\max(0, n_i - 3) \frac{50-C}{50} F, & C < 50, \\ \max(0, n_i - 3) \frac{C-50}{50} F, & C > 50, \\ 0, & C = 0. \end{cases}^{26}$
- $f_{13}(A) = \sum 1_{\text{sus} \rightarrow \text{mayor}}$
- $f_{14}(A) = -\sum 1_{\text{sus sin resolución}}$
- $f_{15}(A) = \sum 1_{\text{Dominante secundario}}$
- $f_{16}(A) = \sum 1_{\text{tensión} \rightarrow \text{resolución}}$
- $f_{17}(A) = \sum 1_{\text{vij}^\circ \rightarrow \text{I}}$
- $f_{18}(A) = \sum 1_{\text{cadencia engañosa}}$
- $f_{19}(A) = \sum 1_{\text{acorde 6/4 cadencial}}$
- $f_{20}(A) = \sum 1_{\text{semi-cadencia}}$
- $f_{21}(A) = 1_{\text{tercera de picardía}}$

²⁶ La función f_{12} es un penalizador por dificultad que responde al parámetro del mismo nombre. La expresión $\max(0, n_i - 3)$ es la cuenta de tonos extra a la triada en un acorde, C es la complejidad y F es el factor de penalización por complejidad. Esta función se encarga de penalizar la progresión si los acordes poseen más de tres notas y la complejidad es menor a 50 o de premiar si sucede todo lo contrario.

Usando puntualmente las funciones individuales y haciendo uso de los pesos w_j , la función objetivo final del Algoritmo Genético para la evaluación de progresiones de acordes es:

$$Fitness(A) = \sum_{j=1}^{21} w_j f_j(A) \quad (13)$$

Una vez completado el ciclo generacional, que comprende la evaluación de cada solución mediante la función objetivo, la selección de progenitores por torneo, la creación de una nueva población con sus descendientes y la aplicación sucesiva de mutación y elitismo (manteniendo el número de candidatos definido por el parámetro correspondiente), se extrae el mejor individuo (la mejor progresión de acordes) y se arroja como salida del módulo después de asignarle a cada acorde la duración respectiva dada por el proceso inicial de Recocido Simulado, de manera que se obtiene una armonía completa ajustada al tamaño de las frases melódicas próximas a ser musicalizadas.

9.1.e Módulo de Melodía: Algoritmo LWRS

El Módulo LWRS utiliza el algoritmo del mismo nombre. Su entrada es, tanto la armonía (acordes con sus respectivas duraciones) y melodía rítmica (melodía cuyas notas sólo poseen duración y no tono). Teniendo la entrada en este punto de la arquitectura, es posible determinar qué acorde de la armonía “*corresponde*”²⁷ cada nota de la armonía; se desarrolló una función que mediante las duraciones de los acordes y de las notas de la melodía, determina eso. Ya sabiendo qué nota de la melodía corresponde a cada acorde, la lógica del módulo consiste en; para cada caso, construir una lista los posibles tonos a seleccionar, ordenados por importancia e implementando la selección LWRS para dar mayor probabilidad según dicha importancia.

Se desarrollaron tres variantes distintas de este módulo, pues las dos primeras no cumplieron el requerimiento de la **Fase 4** de la metodología (“*Evaluación inicial del algoritmo*”), lo que les requirió modificaciones. A continuación se detallan las variantes.

9.1.e.1 Primera versión

Todas las variantes de este módulo siguen la misma lógica general: para cada frase en la melodía se inicia un proceso iterativo para asignar tono a cada nota; para cada nota a procesar se crea una lista con los posibles tonos candidatos, ordenados por su preferencia según reglas de música tonal, de modo que ya construida la lista se utiliza el algoritmo LWRS para seleccionar el tono. Este proceso se repite para todas de cada frase, siendo el mecanismo de ordenamiento y selección distinto dependiendo de si la nota es la primera de cada frase. Para la primer nota de cada frase el proceso se reinicia, consiguiendo que la selección del primer tono no depende de ninguna nota, mientras que el resto de las notas en la frase sí dependen las unas de las otras; concretamente para esta primer variante, cada tono a asignar depende del tono anterior.

Esta primer variante de módulo sigue una lógica sencilla: para cada lista de selección se crean tres subgrupos de tonos candidatos: *tonos del acorde al que pertenece la nota, tonos de la escala/tonalidad, el resto de los tonos entre los doce posibles*. Cada subgrupo ordena sus tonos con respecto a la cercanía a su nota anterior (en el caso de no ser la primer nota de la frase). Además, para decidir el tono en cada caso no sólo se utiliza una selección de sus tonos candidatos, también se decide la dirección de la nota a elegir; siendo posible ir hacia “*arriba*” y hacia “*abajo*”²⁸. Al iniciar el proceso de cada frase se arroja una selección aleatoria entre “*arriba*” y “*abajo*” (50% de probabilidad igual para cada uno); esta dirección decidida definirá mayoritariamente la dirección de la frase. Pues al seleccionar cada uno de los tonos candidatos (exceptuando la primer nota de la frase) se hará una selección LWRS en donde la dirección seleccionada estará en primer lugar y después la otra dirección, de manera que al seleccionar la dirección de cada nota el algoritmo dará preferencia a la dirección general de la frase, pero dejando

²⁷ Se dice que una nota de la melodía “*pertenece*” a determinado acorde, si dicha nota suena mientras el acorde suena.

²⁸ Si la nota de referencia fuera; por ejemplo: D4, las notas hacia “*arriba*” serían D#4, E4, F4, F#4, G4, etc, y hacia “*abajo*” serían Db4, C4, B3, Bb3, A3, etc.

posibilidad a seleccionar notas en la otra dirección. En la práctica las probabilidades para cada nota serían: 63% para la dirección global elegida y 37% para la dirección contraria. Al obtener el resultado de selección de LWRS y saber qué dirección tomará el tono a elegir, los tonos candidatos serán aquellos en la dirección elegida y a partir de la nota anterior. Por ejemplo, si en la nota anterior se seleccionó un F4 y la dirección del nuevo tono es hacia abajo, entonces los candidatos serán F4, E4, Eb4, D4, ..., F3²⁹.

Los subgrupos para la lista de selección son los siguientes:

1. **Tonos del acorde:** Este subgrupo consisten en los tonos que son del acorde al que pertenece la nota a musicalizar.
 - a. **Si la nota es la primera de una frase:** Para este caso se ordenan los tonos comenzando por el más agudo hasta el más grave. Por ejemplo, si se está musicalizando la primer nota de una frase y el acorde correspondiente es Do mayor en segunda inversión ($G \rightarrow C \rightarrow E$), entonces este subgrupo ordenado será [E, C, G]. Al seleccionar este primer tono de cada frase, la octava se asigna según el parámetro del usuario de la octava de la melodía. Es decir, si los candidatos de este subgrupo son [E, C, G] y el usuario definió que la octava para la melodía es la 4, entonces los candidatos son [E4, C4, G4].
 - b. **Si la nota no es la primera de una frase:** En este caso el orden de los tonos dependerá del tono seleccionado para la anterior nota, concretamente en la cercanía hacia la nota anterior con respecto a la dirección seleccionada por la nota. Por ejemplo, si el acorde actual que suena es Do mayor en segunda inversión ($G \rightarrow C \rightarrow E$), el tono anterior seleccionado fue un A4, y la dirección decidida es hacia arriba, entonces, ordenando los candidatos tenemos que el más cercano al A4 (hacia arriba) es C, concretamente con un C5, después E5 y finalmente G5, teniendo entonces como subgrupo ordenado: [C5, E5, G5]. Este orden basado en la cercanía se implementa con el fin de dar preferencia a los tonos más cercanos a la nota anterior, para minimizar la aparición de saltos abruptos en las notas y favorecer más a pasos pequeños y fluidos en las notas de la melodía.
2. **Tonos de la escala:** Este subgrupo se compone de los tonos que pertenecen a la escala/tonalidad de la pieza. Sin considerar aquellos que ya están en el primer subgrupo (los tonos del acorde)³⁰.
 - a. **Si la nota es la primera de una frase:** En este caso, el subgrupo se ordena con base a los tonos de la escala en orden hacia la dirección elegida; comenzando por la tónica y hasta la séptima. Por ejemplo, si la tonalidad es Do mayor y la dirección es hacia arriba, el orden de los tonos es: $C \rightarrow D \rightarrow E \rightarrow F \rightarrow G \rightarrow A \rightarrow B \rightarrow C$. Este subgrupo considera también la tónica de una octava superior (o inferior según la dirección). Es decir, si la octava parámetro del usuario es 4, se considera tanto el C4 como el C5 (o C3 si la dirección es hacia abajo), para permitir la posibilidad de saltos de octava en la melodía (siendo este el salto máximo posible entre dos notas de la misma frase). Por ejemplo, si el acorde que suena es Do mayor en segunda inversión ($G \rightarrow C \rightarrow E$), entonces este subgrupo serán las notas de la escala sin contar las que ya suenan en el acorde (que están en el primer subgrupo); es decir: [D4, F4, A4, B4, C5]; la tónica de octava superior (C5) sí se considera.
 - b. **Si la nota no es la primera de una frase:** En este caso, el subgrupo se ordena con la misma lógica que el primer subgrupo: la cercanía respecto a la nota anterior. Por ejemplo, si el acorde actual es Do mayor en segunda inversión, la nota anterior fue un F4 y la dirección es hacia arriba, el subgrupo final ya ordenado mediante cercanías sería: [F4, A4, B4, D5, F5]; es decir, todas las notas de la escala de Do mayor hacia arriba, desde F4 hasta F5 y sin contar las notas ya presentes en el acorde. Este subgrupo se ilustra en la **Figura 29** con los tonos candidatos señalados en color verde, partiendo desde F4 hasta F5.

²⁹ Los candidatos posibles se toman desde el tono origen hasta el tono de mismo nombre a una octava (arriba o abajo), propiciando que el salto máximo entre dos tonos de una melodía (si son notas de la misma frase) será de una octava.

³⁰ Para esta primer variante fallida del módulo se probó tanto que este grupo contenga los tonos que ya se colocaron en el primer subgrupo, como que se ignoren los que ya están en él; pero ninguna alternativa dio buenos resultados.

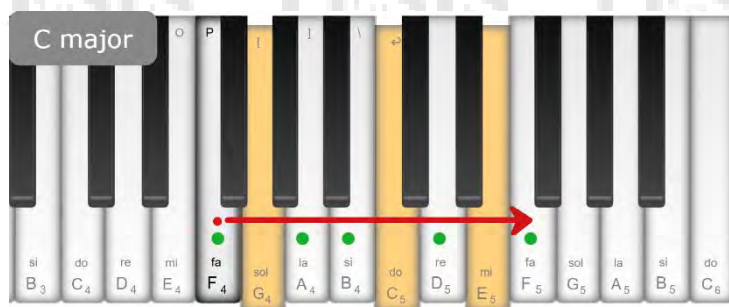


Figura 29. Escala de Do mayor hacia arriba ignorando tonos del acorde Do mayor en segunda inversión.

Nota: Ilustración generada desde una captura del piano digital en Apronus.com.

3. **Tonos restantes cromáticos:** Este subgrupo se compone del resto de las doce notas, que no estuvo presente en los grupos anteriores. La lógica de orden es la misma que para el subgrupo y a grandes rasgos: si es la primer nota de una frase, se toman en el orden natural, en la escala, si no, se toman con distancia de la nota anterior. Para el mismo ejemplo de acorde de Do mayor en segunda inversión, con F4 como nota anterior y con dirección hacia arriba, el resto de los tonos restantes sería [F#4, G#4, A#4, C#5, D#5]. La **Figura 30** muestra las notas de este subgrupo señaladas con cuadrados de color amarillo.

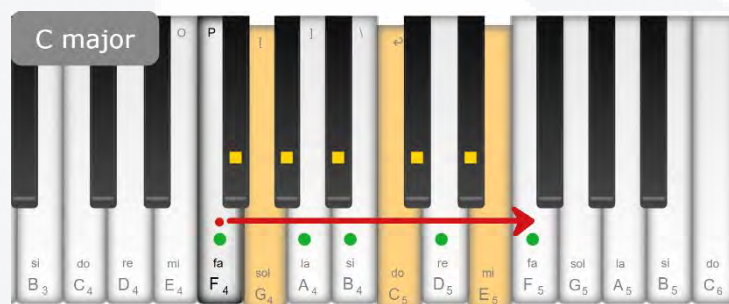


Figura 30. Tonos restantes cromáticos en una escala de Do mayor.

Nota: Ilustración generada desde una captura del piano digital en Apronus.com.

Bajo el ejemplo práctico anterior descrito, para el cuál se va a decidir tono teniendo un acorde sonando de Do mayor en segunda inversión, siendo la nota anterior un F4 y con dirección seleccionada hacia arriba, los tres subgrupos serán:

1. **Notas del acorde ordenadas por cercanía a F4:** [G4, C5, E5].
2. **Notas de la escala/tonalidad por cercanía a F4:** [F4, A4, B4, D5, F5].
3. **Notas cromáticas restantes por cercanía a F4:** [F#4, G#4, A#4, C#5, D#5].

En conjunto la lista de decisión será: [G4, C5, E5, F4, A4, B4, D5, F5, F#4, G#4, A#4, C#5, D#5], para este escenario, siguiendo exactamente la misma lógica que definía Ravera sobre la jerarquía de tonos en la música tonal, pero agregando el factor de favorecer pasos pequeños para producir melodías fluidas, todo implementando un algoritmo que les selecciona dando preferencia pero sin dejar de considerar cualquier posibilidad de nota. Aplicando la ecuación (7), las probabilidades de selección para cada nota en este escenario serían aproximadamente:

- **G4:** 26.26%.
- **C5:** 15.36%.
- **E5:** 10.9%.
- **F4:** 8.45%.
- **A4:** 6.9%.
- **B4:** 5.84%.

- **D5:** 5%.
- **F5:** 4.46%.
- **F#4:** 4%.
- **G#4:** 3.61%.
- **A#4:** 3.3%.
- **C#5:** 3%.
- **D#5:** 2.81%.

Acumulando los porcentajes, tenemos que la probabilidad de elegir cualquier tono del acorde es de 52.52%, mientras que para tonos de la escala es de 30.65% y 16.72% para el resto de los tonos en la escala cromática.

En resumen, esta variante, crea para cada nota a elegir, una lista con los candidatos de tonos, colocando primero los que son del acorde, después los que son de la escala y finalmente los demás tonos entre los doce posibles, ordenando cada subgrupo conforme a la cercanía a la nota anterior elegida con base en una dirección (excepto cuando es la primer nota de cada frase), después se utiliza el algoritmo LWRS para seleccionar aleatoriamente y se asigna tono a la nota. Se repite el proceso hasta musicalizar cada nota de la melodía.

Los resultados de esta variante no fueron lo suficientemente buenos según el criterio de la fase 4 de la metodología. Las melodías generadas no parecían tener coherencia en medida que se presentaban muchos saltos drásticos entre tonos, que daban la sensación de ser aleatorios y sin sentido, además de que se presentaban muchas disonancias gracias a tonos del tercer subgrupo apareciendo. Los saltos abruptos generaban una sensación de improvisación sin fluidez ni intencionalidad. Se distinguen dos razones a las cuales atribuir el fallo de esta lógica: lo primero es que, los tonos elegidos para las notas dependen enteramente de la anterior y no del resto de notas de cada frase; al hacerse decisiones sólo basadas en esto no se da cabida a que las frase por sí mismas tengan un contorno, figura o en general una intención colectiva que haga que el resto de las notas tenga relación las unas con las otras. Como segunda razón, se sabe que absolutamente todas las notas que no son la primera de cada frase utilizaban exactamente el mismo método de generación, sin dar cabida a que las notas tengan jerarquías según factores como su duración, es por ello que se presentaban escenarios en los que notas largas y críticas estructuralmente elegían tonos que disuenan fuertemente con el acorde.

Esta primer versión, en conjunto con el resto de los módulos del algoritmo, compuso al primer intento lo que se podría decir la primer obra musical compuesta por el algoritmo MIA. Utilizando como entrada las cuatro primeras líneas del tema de 1968 “Hey, Jude” de Paul McCartney: “Hey, Jude”, “Don’t make it bad”, “Take a sad song”, “And make it better”.

Se utilizaron estas frases como entrada del algoritmo para musicalizarlas y se definieron parámetros muy similares a los que se describieron como parámetros por defecto, dándole a cada frase un compás de 4/4, asignando anacrusa de un compás y componiendo en Do mayor. Dando como resultado la pieza mostrada en la **Figura 31**.

Melodic Jude

Letra de Paul McCartney Primera composición de MIA Harmfirst a1 ARH MIA

♩ = 90

Oboe

Hey jude Don't make it bad Take a sad song and make it better Hey

Piano

Cm G Dm G7 C

Figura 31. "Melodic Jude", primer tema compuesto por el algoritmo MIA.

9.1.e.2 Segunda versión

La segunda variante es una leve modificación de la primer variante. Esta utiliza exactamente la misma lógica salvo por un nuevo subgrupo: *tonos comunes de los acordes adyacentes*. Este subgrupo se compone de tonos que tienen en común el acorde actual y los acordes adyacentes: anterior y/o siguiente, y se construye en validando cualquiera de los dos siguientes casos:

1. **Si la nota es de frontera:** Se le llama nota de frontera cuando esta es la primera que cae al iniciar un nuevo acorde. Una nota frontera es pues la primera nota de la melodía que se tiene al cambiar de acorde. Este es un primer acercamiento en darle “jerarquía” y distinción a las notas. Al ser una nota de frontera, el subgrupo se forma con dos conjuntos:
 - a. **Tonos comunes del acorde actual con el acorde siguiente:** Estos son los tonos que comparte el acorde actual y al acorde al que se quiere llegar enseguida. Por ejemplo; si el acorde actual es un Sol mayor ($G \rightarrow D \rightarrow B$), y el siguiente acorde es un Mi menor ($E \rightarrow G \rightarrow B$), estos dos acordes poseen en común G y B, por lo que ambas notas forman parte del primer conjunto y estas se ordenan conforme a la cercanía con la nota anterior; si es la primer nota de la frase entonces se toman en el orden en que estén: [G, B].
 - b. **Tonos comunes del acorde actual con el acorde anterior:** Estos son los tonos que comparten el acorde actual con el anterior. Por ejemplo, si el acorde actual es Sol mayor ($G \rightarrow D \rightarrow B$) y el anterior fue Do menor ($C \rightarrow E_b \rightarrow G$), en común se tienen [G]. La misma lógica de orden del conjunto anterior se aplica a este conjunto.
2. **Si la nota no es de frontera:** Este subgrupo se formará únicamente de las notas comunes del acorde actual y el acorde anterior.

Si la nota es de frontera, ambos conjuntos generados se unen tomando primero el grupo de tonos comunes con el acorde siguiente, y si hay tonos repetidos en ambos conjuntos, se omiten los del segundo conjunto que ya estén en el primero. Si la nota no es de frontera se toma el subgrupo respectivo generado. Este subgrupo se coloca en segundo lugar respecto a la jerarquía de la variación primera de este módulo. Es decir, que la lista de selección se construiría: 1. Tonos del acorde, 2. Tonos comunes de los acordes adyacentes, 3. Tonos de la escala, 4. Tonos restantes cromáticos, e igualmente se selecciona tono utilizando LWRS.

La lógica de utilizar tonos en común entre el acorde actual y el siguiente o el anterior parte del concepto que desarrolla George Russell en “*The Lydian Chromatic Concept of Tonal Organization for Improvisation*” (“*El concepto cromático lidio de organización tonal para la improvisación*”) de 1959. Russell aborda la idea de que cada acorde tiene una escala matriz asociada, y que un músico (haciendo improvisación) debe pensar siempre tanto el acorde actual como el siguiente para crear líneas melódicas coherentes y expresivas.

Este enfoque pareció traer buenos resultados, con melodías con mejor calidad a la escucha y se consideró temporalmente como la versión definitiva. Inclusive se compuso una pieza incluida en los experimentos finales, que surgió de un primer intento al introducirse versos de un poema de Ruth Serafio Méndez (tema “*Desaparecer*”³¹, consúltese 10 Experimentos – 10.1 Planeación). La pieza compuesta (previo a la incorporación de la mecánica de múltiples Recocidos Simulados) se muestra en la **Figura 32**. Para esta se tomaron cuatro frases con una duración de dos compases por cada frase, con un acorde por cada frase excepto en la tercer frase, en una tonalidad de Sol menor, sin anacrusa y con parámetros muy similares a los descritos como parámetros por defecto. Los cuatro versos son: “*Pero algo dentro de mí*”, “*Decía que tú eras la razón y*”, “*Volviste a dar un desdén*” y finalmente “*Decirle que no al amor*” Serafio Méndez (2025a).

³¹ El poema no posee una fuente pública, pues es un trabajo en edición. Los versos se compartieron de manera privada al autor del presente trabajo.

Desaparecer

Versos de
Ruth Serafio

Versos musicalizados

ARH MIA

Ob. $\text{♩} = 90$

Pero al godentro de mí De cí a que tue ras la razón y Vol vis teadar

F D7 Gm

Piano

Ob. 7

un des dén De cí le que no al a mor

Dm G#dim

Pno.

Figura 32. "Desaparecer": tema compuesto con la segunda variante del módulo LWRS, con versos de un poema.³²

A pesar de ser una buena variante para el módulo, pues compuso melodías buenas según el criterio de la fase 4 de la metodología, se decidió reescribir la lógica debido a que este módulo limitaba fuertemente la variedad en sus resultados, a causa de propiciar muchas repeticiones en los mismos tonos elegidos. Al construir una armonía con reglas tonales, se garantiza que buena parte (o todos) los acordes de la misma serán propios de la escala/tonalidad, por lo que todos estos estarán formados por las mismas siete notas; este principio, sumado a la introducción de un subgrupo que repite tonos ya presentes en el resto de subgrupos, pero considerando tonos en común entre acordes adyacentes, hace que se propicie mayor repetición de las mismas notas, al estar varias veces en las listas de selección para LWRS. Esto genera buenos resultados, pero limita la exploración de contornos más interesantes en las melodías.

9.1.e.3 Variante funcional

La versión final implementada contempla solucionar dos errores existentes en las dos versiones anteriores: en primer lugar, propiciar verdaderos contornos melódicos³³, para mostrar intencionalidad en las melodías, y en segundo lugar, establecer jerarquías más sólidas para las notas según su criticidad dentro de la melodía. Otro cambio relevante es la variabilidad de las listas de selección en cada caso, que ahora no son siempre la misma estructura de lista, sino que varían según la jerarquía de cada nota.

Las jerarquías de nota funcionan como la mejora más relevante en esta variante del módulo, pues mejoran la variedad de listas de decisión para cada caso específico, en medida que introducen una mayor cantidad de reglas lógicas basadas en el contexto específico de cada nota a musicalizar. Previo a la iteración entre notas para construir listas de decisión, se categoriza cada nota en una de cuatro jerarquías: "estructural", "principal", "importante" y "de paso", y estas jerarquías se asignan respondiendo las siguientes preguntas:

³² Las duraciones de los acordes se aplazaron para fines visuales en el pentagrama, sin embargo originalmente tienen el tamaño máximo de redonda con puntillo, pero no se modificaron los inicios de cada acorde.

³³ Entiéndase como "contorno" en la melodía cuando los saltos en los tonos de sus notas, en conjunto sugieren figuras intencionadas, como arcos, valles, escaladas, descensos, etc.

1. **¿La nota es fuerte?:** Se entiende como “fuerte” si una nota inicia justo cuando ocurre un cambio de acorde o si es la primer nota inmediatamente después de un segmento de anacrusa. Esta fuerza se define como el acento que aporta dentro de una melodía.
2. **¿La nota es larga?:** Se define a una nota larga si ocupa por lo menos el 50% del tiempo de un compás. Las notas largas tienen una necesaria jerarquía en medida que deberán evitar producir tonos altamente disonantes, y preferiblemente deben favorecer tonos que al ser reproducidos por largo tiempo, suenen agradables y en sintonía con la armonía.
3. **¿Es la primer nota de una frase?:** Si es la nota con la que una frase comienza, entonces debe tener una especial selección de tonos.
4. **¿Es la última nota de una frase?:** Tanto si es la primera, como si fuera la última, estas notas tienen un peso estructural en la melodía, e indirectamente denotan dirección; pues una define el punto de inicio de la frase y la otra el punto final.

Contestando a estas preguntas y para cada nota, es que se pueden asignar las siguientes jerarquías:

1. **Estructural:** Notas más relevantes en la estructura de una melodía. La nota es estructural si cumple alguna de las tres combinaciones de condiciones:
 - a. Es fuerte y es larga.
 - b. Es fuerte y es la primera de la frase.
 - c. Es larga y es la última de la frase.
2. **Principal:** Indican que la nota coincide con el cambio de un acorde, pero no cumple los requisitos de duración o de ser la primera de una frase. Se marca si no fue estructural y si en cambio cumple alguna de las siguientes combinaciones de condiciones:
 - a. Es fuerte.
 - b. Es fuerte y es la última de la frase.
3. **Importante:** Notas que son importantes por su duración o por marcar el inicio/final de una frase, aunque no coincidan con el punto más fuerte y acentuado armónicamente. Se marcan así si no son estructurales o principales y si cumplen alguna de las siguientes características:
 - a. Es larga.
 - b. Es la primera de la frase.
 - c. Es la última de la frase.
4. **De paso:** Notas que no entran en ninguna de las categorías anteriores. Estas notas son de transición entre las que sí son importantes en el contexto melódico, por lo que permitirían utilizar tonos disonantes sin representar problemas a la escucha.

Si bien esta variante sigue considerando crear listas de selección e implementar LWRS, la lógica cambia según la jerarquía de nota, siendo primeramente que las notas de categoría *estructural* serán “*anclas*” y sus tonos se elegirán en la primera “*pasada*” del algoritmo; y el resto se resolverán en una segunda pasada. La decisión de tonos en estas “*anclas*” permitirán aproximar el contorno de las frases. Los contornos los decide el usuario para cada frase, entre cuatro posibles alternativas³⁴:

- **Arco:** Frase cuyas notas (mayoritariamente, pero no para todas las notas) suben hasta un “*clímax*” ancla (aproximadamente a la mitad de la frase) y después descienden.
- **Valle:** Frases cuyas notas descienden hasta un clímax y después ascienden. Exactamente la lógica de un “*arco*”, pero de manera invertida.
- **Subida:** Frases cuyas notas mayormente ascienden.
- **Bajada:** Frases cuyas notas mayormente descienden.

Las notas estructurales se definen construyendo una lista de selección que consiste en únicamente los tonos del acorde. A diferencia de versiones pasadas del módulo, acá se utiliza este subgrupo para las notas estructurales,

³⁴ El usuario también puede seleccionar “aleatorio” como contorno de frase, y en ese caso el contorno se elegirá mediante azar y sin distinción de probabilidad entre las opciones.

de manera que estas siempre serán sí o sí tonos del acorde que suena. Sin embargo, el tono del acorde a utilizar se define mediante el cálculo de una altura “ h ” que permite considerar a estas notas como ancla en la creación de los contornos melódicos definidos. La interpretación geométrica de cada contorno es la siguiente:

- **Arco:** Para cada posición relativa $p_{relativa} = \frac{r}{(M-1)}$ donde M es la cantidad de notas estructurales y r va desde 0 a $M - 1$, se le asigna una altura: $h = 4p_{relativa} * (1 - p_{relativa})$, que genera una parábola con un máximo en el medio (si $M > 1$).
- **Valle:** Se invierte la parábola de arco como $h = 1 - 4p_{relativa} * (1 - p_{relativa})$ con un mínimo en el medio (para $M > 1$).
- **Subida:** Se asignan linealmente $h = p_{relativa}$, de modo que la primer estructural ancla queden en la posición de altura mínima (0) y la última en la máxima (1).
- **Bajada:** Si invierte el valor de subida como $h = 1 - p_{relativa}$.

Esta lógica de valores de altura “ h ” se utilizan para decidir qué tono de la lista de tonos del acorde asignar a cada ancla estructural de la frase, de modo que dichas notas formen un conjunto predefinido (arco, valle, subida o bajada) en lugar de elegir siempre el mismo tono del acorde. Este h está entre 0 y 1 según la posición relativa de esa ancla entre todas las notas estructurales.

- Si $h = 0$ se toma la nota más grave del acorde.
- Si $h = 1$ se toma la nota más aguda del acorde.
- Si h está entre medio (por ejemplo, 0.5 en un arco), se elige el tono intermedio³⁵.

De esta manera, en la primer pasada, las notas estructurales no se colocan al azar, sino que quedan distribuidas a lo largo del acorde que suena, lo que garantiza un “*esqueleto melódico*” coherente antes de rellenar el resto de las notas.

En la segunda pasada, y una vez que todas las notas estructurales han quedado fijas como anclas con base al acorde correspondiente, siguiendo el contorno elegido, se procede a asignar tonos al resto de notas de la frase, que fueron clasificadas como “*principal*”, “*importante*” o “*de paso*”. El objetivo es que estas notas intermedias fluyan de manera natural hacia la siguiente ancla, manteniendo coherencia melódica y evitando saltos bruscos.

Para cada nota que aún no tenga asignado un tono, el primer paso consiste en la determinación de su dirección hacia la siguiente ancla. Para cada tramo entre dos anclas se calcula la diferencia en semitonos entre el tono ya asignado a la nota anterior y el tono proyectado para la próxima ancla, y a partir de esa diferencia se determina cuál es la dirección preferida; “*hacia arriba*” si la ancla tiene una frecuencia más alta (es más aguda, por ende está arriba), o “*hacia abajo*” en caso contrario. Para introducir cierta flexibilidad sin perder la tendencia, se utiliza el algoritmo EWRS, a diferencia de la primer versión del módulo, que utilizaba LWRS para determinar la dirección de una próxima nota, dando preferencia a la dirección preferida. EWRS aumenta el sesgo ligeramente utilizando el parámetro de base 2, es decir, un 66.7% de probabilidad para la dirección ya establecida y 33.3% para la dirección contraria³⁶. Esto asegura que, aunque los tonos tiendan a dirigirse hacia la ancla, se decidan ocasionalmente desvíos melódicos que aportan variedad.

Posterior a seleccionar la dirección del tono en la próxima nota, e idénticamente a las dos versiones anteriores al módulo, se construyen listas de selección con los posibles tonos candidatos, para ser seleccionado uno mediante LWRS, aunque en este caso la categoría de nota cambiará los candidatos en la lista, teniendo los siguientes casos, y para cada uno, los respectivos subgrupos que le conforman:

1. **Para notas de tipo “Principal”:** Tonos del acorde + Tonos de la escala (sin repetir los del acorde) + Tonos del acorde (nuevamente); Aquí es necesaria una nota: originalmente se pensaba que este grupo

³⁵ Para el caso en que exista sólo una nota estructural ($M=1$), se fuerza que $h=0$.

³⁶ Una ventaja del uso de EWRS en lugar de LWRS para la selección de dirección, es que el parámetro “base” puede ajustarse para maximizar el direccionamiento.

fuera únicamente Tonos del acorde + Tonos de la escala incluyendo los que ya estén en el acorde, sin embargo se desconoce porqué se codificó como se describe, pues no hay recuerdo claro de la razón. Es probable que se haya hecho de esta manera debido al propósito de sí conservar los tonos del acorde pero pasarlos a un último subgrupo y así darle más peso al conjunto de los tonos de la escala separados de los tonos del acorde, pero se cometió el error de pensar que estos tonos del acorde podrían ni siquiera estar en la escala. También es probable que se haya hecho con la intención de maximizar aún más el uso de tonos del acorde.

2. **Para notas del tipo “Importante”:** Tonos del acorde + Tonos de la escala sin contar los del acorde.
3. **Para notas del tipo “De paso”:** Tonos del acorde + Tonos de la escala sin contar los del acorde + Tonos cromáticos restantes. Acá hay otra inconsistencia desconocida respecto a la planeación original “*en papel*”, pues se figuraba ya no incluir el subgrupo de los tonos del acorde, sólo los tonos de la escala y los tonos cromáticos, sin embargo hay posibilidad de que el cambio respecto a la codificación se deba a una necesidad de maximizar aún más la presencia de tonos del acorde.

Previo a utilizar la lista en la selección con LWRS se realiza un filtro con el que se descartan de la lista aquellos tonos que estén a una distancia en semitonos mayor a un parámetro cuyo nombre es “*salto máximo de semitonos*”. Este parámetro tiene un **valor por defecto** en **12** semitonos (es decir, una octava) y delimita que la melodía no tendrá saltos entre notas de una misma frase, mayores a dicha distancia. Después del filtro, entonces se utiliza LWRS para seleccionar el tono entre los candidatos, y se repite el proceso antes descrito con el resto de las notas no estructurales de la melodía.

De forma resumida, durante la segunda pasada, cada nota intermedia avanza así:

1. Se identifica la próxima ancla y se calcula la dirección preferida.
2. Con EWRS se decide, con aproximadamente 66% de probabilidad, si la nota debe moverse en la dirección deseada o bien, con 33% de probabilidad, si se opta por la dirección opuesta.
3. Se construye la lista de candidatos en tres subgrupos de tonos (del acorde, de la escala, cromáticos restantes) de acuerdo con la categoría (principal, importante o de paso).
4. Se descartan los candidatos cuya diferencia en semitonos supere el salto máximo permitido.
5. Se elige un tono con LWRS, que privilegia mayormente los primeros elementos de la lista para reforzar la consonancia, pero sin eliminar la posibilidad de grados menos convencionales.

El resultado del módulo es una melodía que trata de imprimir intencionalidad mediante contornos melódicos (en la medida que sea posible) y que al jerarquizar sus notas, no sólo propicia variedad en la lógica de decisión (al menos con respecto a las versiones anteriores del código), sino que incluso introduce valoradores basados en el ritmo (duración y fuerza en las notas), de manera que se garanticen resultados más consonantes al oído. La salida es entonces, la melodía rítmica recibida pero con tonos adaptados a la armonía. El resultado final esperado y el último componente necesario para concretar una “*idea musical*”.

Con esta versión se compusieron el resto de las piezas musicales utilizadas en los experimentos, además de pruebas interesantes como la mostrada en la **Figura 33**, que se realizó utilizando los versos del poema “*Aceptación*”, compartidos en privado por su autor, Abel Soto Valdez: “*Tus dotes a la sazón*”, “*Hacen sentir a la razón*”, “*De ahí que el intricado*”, “*Se vea ahora en un altercado*”. La idea musical compuesta en la tonalidad de Re mayor (D), con dos compases de 4/4 para cada frase, con anacrusa de 1 tiempo y con los contornos de “*arco*”, “*valle*”, “*subida*”, “*bajada*”³⁷, se ilustra.

³⁷ Se evidencia que no en todos los casos se consiguen los contornos esperados, al depender estos sólo de notas estructurales; sin embargo, es una característica que se puede mejorar.

Aceptación

Versos de
Abel Soto

MIA Harmfirst a1 versión final

ARH MIA

$\text{♩} = 100$ VERSION 2_a1

Oboe

Tus do tes a la sazón Ha cen sen tir a la razón Dea

Piano

Bm F#m

6

Ob.

hí quel in tri ca do Se vea ho ra en un al ter ca do Tus

Pno.

C#dim G D

Figura 33. "Aceptación 2a1": Pieza generada por la versión final de MIA Harmfirst a1.

De la misma manera se compuso una segunda obra con los mismos versos y las mismas características; ilustrada en la **Figura 34** y siendo esta la octava pieza generada (siendo la anterior la primera y seleccionando las dos mejores entre estas 8 generaciones de piezas).

Aceptación

Versos de
Abel Soto

MIA Harmfirst a1 versión final

ARH MIA

$\text{♩} = 100$ VERSIÓN 4_a8

Oboe

Tus do tes a la sa zón Ha cen sen tir a la razón Dea

Piano

G D

6

Ob.

hí quel in tri ca do Se vea ho ra en un al ter ca do Tus

Pno.

A7 A7 D

Figura 34. "Aceptación 4a8": Octava pieza generada por la versión final de MIA Harmfirst a1.

9.1.f Módulo MIDI

El módulo MIDI se encarga de transformar las representaciones computacionales internas de los elementos de melodía y armonía en un archivo. El MIDI es un formato de archivo que contiene información sobre notas y otros datos musicales asociados, en lugar del sonido real. No es un archivo de audio como MP3 o WAV, sino una secuencia de instrucciones que un dispositivo MIDI puede interpretar para generar sonido. El estándar MIDI es uno de los más populares en el contexto musical; lo producen como salida tanto los instrumentos digitales (véase pianos y teclados electrónicos, baterías eléctricas, etc.), como los software más populares de notación musical, como *Muscore*, *Sibelius*, *Finale*, *Dorico*, entre otros. Además, cualquier software de grabación y producción (DAW: “*Digital Audio Workstation*”) trabaja con MIDIs y los interpreta visualmente como “*piano roll*”; véase *Ableton*, *FL Studio*, *Pro Tools*, *Logic Pro X*, *Cubase*, y *Studio One*, *Reaper*, etc. En la práctica, cualquier software popular de herramienta para músicas utiliza MIDI como un formato estándar para la representación digital de música. La **Figura 35** muestra el llamado “*piano roll*” que cualquier DAW visualiza al trabajar con archivos MIDI, en este caso se coloca el tema de la **Figura 34** en el software *Reaper*, mostrando en color azul las notas de la melodía y en color rojo las notas de los acordes en la armonía.

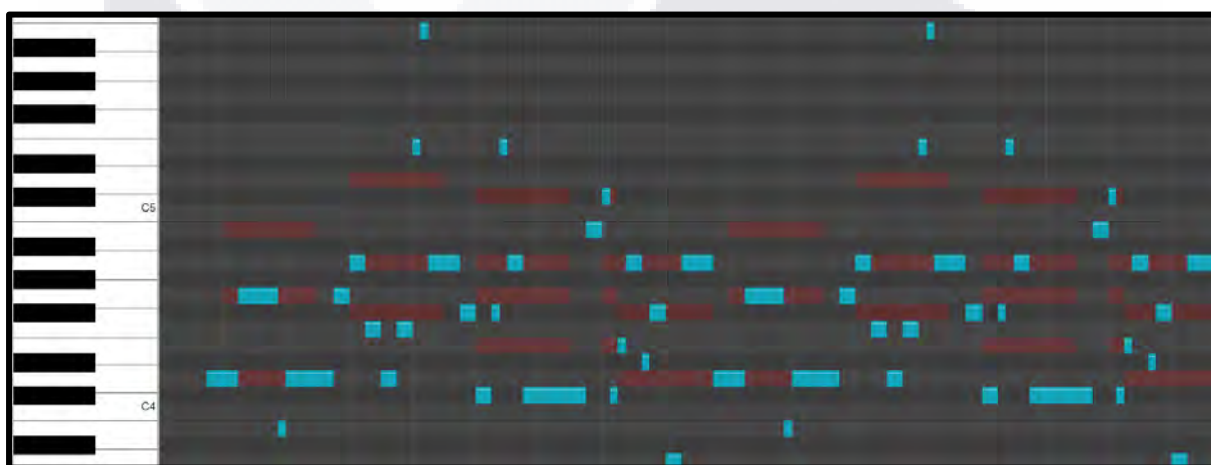


Figura 35. Piano roll para la pieza "Aceptación 4a8" visto en el DAW *Reaper*.

La salida MIDI del algoritmo resulta perfecta para cualquier músico, pues de abrirla con su herramienta de notación, verá la partitura (como las mostradas en las **Figuras 32, 33 y 34**) para ser modificada o reproducida, de abrirla con su herramienta DAW, podrá introducirle instrumentos virtuales, modificarla, aplicarle efectos y reproducirle. Reproductores de audio populares en computadoras con sistema operativo Windows, como *VLC* incluso permiten reproducirlos aplicando sonido según los instrumentos codificados. La salida generada por el algoritmo MIA está, de manera simple, hecha para tomarse y comenzarse a trabajar directamente.

9.1.g Salidas

El sistema completo guarda en el computador que le ejecuta un archivo MIDI con la pieza generada, además de un archivo de texto (TXT) con la información y representación interna de toda la información de la pieza. Como ejemplo de cómo luce la representación en texto plano de una pieza generada por MIA, el archivo TXT de la pieza en la **Figura 33**, contiene lo siguiente:

MIA Harmfirst a1 COMPOSITION
Date: 11/04/2025 at 12:40:39

Title: Tus dotes
Composer: ARH MIA

Melody(

```

signature=(4, 4),
key=('D', 'major'),
tuning=A4[440Hz],
upbeat=1,
anacrusis=True,
space=512.0,
bars_amount=8.0,
phrases=[
    Phrase(
        space=128.0,
        notes=[
            Note(note=D, octave=4, frequency=293.6648Hz, time=4, name=Quarter
dot=False space=16, tuning=A4[440Hz]),
            Note(note=D, octave=4, frequency=293.6648Hz, time=4, name=Quarter
dot=True space=24.0, tuning=A4[440Hz]),
            Note(note=A, octave=3, frequency=220.0Hz, time=8, name=Eighth dot=False
space=8, tuning=A4[440Hz]),
            Note(note=C#, octave=4, frequency=277.1826Hz, time=2, name=Half
dot=False space=32, tuning=A4[440Hz]),
            Note(note=C#, octave=4, frequency=277.1826Hz, time=8, name=Eighth
dot=False space=8, tuning=A4[440Hz]),
            Note(note=C#, octave=4, frequency=277.1826Hz, time=8, name=Eighth
dot=False space=8, tuning=A4[440Hz]),
            Note(note=D, octave=4, frequency=293.6648Hz, time=2, name=Half dot=False
space=32, tuning=A4[440Hz])
        ]
    ),
    Phrase(
        space=128,
        notes=[
            Note(note=D, octave=4, frequency=293.6648Hz, time=4, name=Quarter
dot=False space=16, tuning=A4[440Hz]),
            Note(note=G, octave=4, frequency=391.9954Hz, time=4, name=Quarter
dot=False space=16, tuning=A4[440Hz]),
            Note(note=A, octave=4, frequency=440.0Hz, time=4, name=Quarter dot=False
space=16, tuning=A4[440Hz]),
            Note(note=A, octave=4, frequency=440.0Hz, time=4, name=Quarter dot=False
space=16, tuning=A4[440Hz]),
            Note(note=C, octave=5, frequency=523.2511Hz, time=4, name=Quarter
dot=False space=16, tuning=A4[440Hz]),
            Note(note=G, octave=4, frequency=391.9954Hz, time=8, name=Eighth
dot=False space=8, tuning=A4[440Hz]),
            Note(note=G, octave=5, frequency=783.9909Hz, time=8, name=Eighth
dot=False space=8, tuning=A4[440Hz]),
            Note(note=A, octave=4, frequency=440.0Hz, time=2, name=Half dot=False
space=32, tuning=A4[440Hz])
        ]
    ),
    Phrase(
        space=128,
        notes=[
            Note(note=A, octave=4, frequency=440.0Hz, time=4, name=Quarter dot=False
space=16, tuning=A4[440Hz]),
            Note(note=F#, octave=4, frequency=369.9944Hz, time=4, name=Quarter
dot=False space=16, tuning=A4[440Hz]),
            Note(note=F, octave=5, frequency=698.4565Hz, time=4, name=Quarter
dot=False space=16, tuning=A4[440Hz]),
            Note(note=F#, octave=4, frequency=369.9944Hz, time=4, name=Quarter
dot=False space=16, tuning=A4[440Hz]),
            Note(note=E, octave=4, frequency=329.6276Hz, time=4, name=Quarter
dot=False space=16, tuning=A4[440Hz]),
            Note(note=E, octave=4, frequency=329.6276Hz, time=4, name=Quarter
dot=False space=16, tuning=A4[440Hz]),

```



```

    Note(note=C#, octave=4, frequency=277.1826Hz, time=2, name=Half
dot=False space=32, tuning=A4[440Hz])
    ]
    ),
    Phrase(
        space=128,
        notes=[
            Note(note=E, octave=4, frequency=329.6276Hz, time=4, name=Quarter
dot=False space=16, tuning=A4[440Hz]),
            Note(note=E, octave=4, frequency=329.6276Hz, time=4, name=Quarter
dot=False space=16, tuning=A4[440Hz]),
            Note(note=E, octave=4, frequency=329.6276Hz, time=8, name=Eighth
dot=False space=8, tuning=A4[440Hz]),
            Note(note=F#, octave=4, frequency=369.9944Hz, time=8, name=Eighth
dot=False space=8, tuning=A4[440Hz]),
            Note(note=A, octave=4, frequency=440.0Hz, time=8, name=Eighth dot=False
space=8, tuning=A4[440Hz]),
            Note(note=F#, octave=5, frequency=739.9888Hz, time=8, name=Eighth
dot=False space=8, tuning=A4[440Hz]),
            Note(note=F#, octave=5, frequency=739.9888Hz, time=4, name=Quarter
dot=False space=16, tuning=A4[440Hz]),
            Note(note=C, octave=5, frequency=523.2511Hz, time=4, name=Quarter
dot=False space=16, tuning=A4[440Hz]),
            Note(note=A, octave=4, frequency=440.0Hz, time=2, name=Half dot=False
space=32, tuning=A4[440Hz])
        ]
    )
]
)
Harmony(
    signature=(4, 4),
    key=('D', 'major'),
    tuning=A4[440Hz],
    upbeat=1,
    anacrusis=True,
    space=528.0,
    bars_amount=8.25,
    chords=[
        Chord(name=X, ctype=None, inversion=None, octave=None, degree=, time=4, space=16,
dot=False notes=[Note(note=X, octave=0, frequency=0Hz, time=4, name=Quarter dot=False
space=16, tuning=A4[440Hz])], notes_str=[]),

        Chord(name=B, ctype=m, inversion=1, octave=3, degree=vi, time=1, space=96.0,
dot=True notes=[Note(note=D, octave=4, frequency=293.6648Hz, time=1, name=Whole dot=True
space=96.0, tuning=A4[440Hz]), Note(note=F#, octave=4, frequency=369.9944Hz, time=1,
name=Whole dot=True space=96.0, tuning=A4[440Hz]), Note(note=B, octave=4,
frequency=493.8833Hz, time=1, name=Whole dot=True space=96.0, tuning=A4[440Hz])],
notes_str=['D4', 'F#4', 'B4']),

        Chord(name=X, ctype=None, inversion=None, octave=None, degree=, time=2, space=32,
dot=False notes=[Note(note=X, octave=0, frequency=0Hz, time=2, name=Half dot=False
space=32, tuning=A4[440Hz])], notes_str=[]),

        Chord(name=F#, ctype=m, inversion=2, octave=3, degree=iii, time=1, space=96.0,
dot=True notes=[Note(note=C#, octave=4, frequency=277.1826Hz, time=1, name=Whole dot=True
space=96.0, tuning=A4[440Hz]), Note(note=F#, octave=4, frequency=369.9944Hz, time=1,
name=Whole dot=True space=96.0, tuning=A4[440Hz]), Note(note=A, octave=4,
frequency=440.0Hz, time=1, name=Whole dot=True space=96.0, tuning=A4[440Hz])],
notes_str=['C#4', 'F#4', 'A4']),

        Chord(name=X, ctype=None, inversion=None, octave=None, degree=, time=2, space=32,
dot=False notes=[Note(note=X, octave=0, frequency=0Hz, time=2, name=Half dot=False
space=32, tuning=A4[440Hz])], notes_str=[]),
    ]
)

```

```

Chord(name=C#, ctype=dim, inversion=1, octave=4, degree=VII, time=1, space=96.0,
dot=True notes=[Note(note=E, octave=4, frequency=329.6276Hz, time=1, name=Whole dot=True
space=96.0, tuning=A4[440Hz]), Note(note=G, octave=4, frequency=391.9954Hz, time=1,
name=Whole dot=True space=96.0, tuning=A4[440Hz]), Note(note=C#, octave=5,
frequency=554.3653Hz, time=1, name=Whole dot=True space=96.0, tuning=A4[440Hz])],
notes_str=['E4', 'G4', 'C#5']),

Chord(name=X, ctype=None, inversion=None, octave=None, degree=, time=2, space=32,
dot=False notes=[Note(note=X, octave=0, frequency=0Hz, time=2, name=Half dot=False
space=32, tuning=A4[440Hz])], notes_str=[]),

Chord(name=G, ctype=, inversion=0, octave=4, degree=IV, time=4, space=16, dot=False
notes=[Note(note=G, octave=4, frequency=391.9954Hz, time=4, name=Quarter dot=False
space=16, tuning=A4[440Hz]), Note(note=B, octave=4, frequency=493.8833Hz, time=4,
name=Quarter dot=False space=16, tuning=A4[440Hz]), Note(note=D, octave=5,
frequency=587.3295Hz, time=4, name=Quarter dot=False space=16, tuning=A4[440Hz])],
notes_str=['G4', 'B4', 'D5']),

Chord(name=D, ctype=, inversion=1, octave=4, degree=I, time=1, space=96.0, dot=True
notes=[Note(note=F#, octave=4, frequency=369.9944Hz, time=1, name=Whole dot=True
space=96.0, tuning=A4[440Hz]), Note(note=A, octave=4, frequency=440.0Hz, time=1, name=Whole
dot=True space=96.0, tuning=A4[440Hz]), Note(note=D, octave=5, frequency=587.3295Hz,
time=1, name=Whole dot=True space=96.0, tuning=A4[440Hz])], notes_str=['F#4', 'A4', 'D5']),

Chord(name=X, ctype=None, inversion=None, octave=None, degree=, time=4, space=16,
dot=False notes=[Note(note=X, octave=0, frequency=0Hz, time=4, name=Quarter dot=False
space=16, tuning=A4[440Hz])], notes_str=[])
]
)

```

LOGS:

SENTENCES:

[1]: Tus dotes a la sazón

- ['Tus*', ' ', 'do*', 'tes', ' ', 'a*', ' ', 'la*', ' ', 'sa', 'zón*']
- Bars: 2
- Chords: 1
- Shape: arch

[2]: Hacen sentir a la razón

- ['Ha*', 'cen', ' ', 'sen', 'tir*', ' ', 'a*', ' ', 'la*', ' ', 'ra', 'zón*']
- Bars: 2
- Chords: 1
- Shape: valley

[3]: De ahí que el intricado

- ['De*', ' ', 'a', 'hí*', ' ', 'que*', ' ', 'el*', ' ', 'in', 'tri', 'ca*', 'do']
- Bars: 2
- Chords: 1
- Shape: rise

[4]: Se vea ahora en un altercado

- ['Se*', ' ', 've*', 'a', ' ', 'aho*', 'ra', ' ', 'en*', ' ', 'un*', ' ', 'al', 'ter', 'ca*', 'do']
- Bars: 2
- Chords: 2
- Shape: fall

9.2 Implementación del algoritmo

El sistema se programó en el lenguaje de programación Python, con su versión 3.11. Se utilizan como dependencias las librerías: *pandas* 2.2.2 para la lectura de los diccionarios de palabras, *nltk* 3.8.1 para la generación de palabras mediante la definición de la Gramática Generativa y *midituil* 1.2.1 para la creación de archivos MIDI con la información de las notas generadas en una idea musical.

Se crearon clases de Python para representar cada aspecto básico de las piezas, como la *Nota* siendo la clase de más baja representación y con sus características de *nombre*, *octava*, *frecuencia*, *tiempo* (un indicador numérico del tipo de figura de duración), *nombre* (con el nombre en texto de la figura de duración), *punto*, que indica con *verdadero* o *falso* si la nota contiene punto, *espacio* (que es una representación numérica de la duración, y cuya unidad más baja es 1, de la figura *semifusa*), y *afinación* (número que determina qué frecuencia de afinación contiene la nota central A4, que por defecto es 440). La clase *Acorde* contiene en cambio *nombre*, *tipo*, *inversión*, *octava*, *grado* (que implica en notación de números romanos a qué grado en la escala equivale el acorde), *tiempo*, *espacio*, *punto* y *notas* (que equivale a una lista de los objetos *Nota* que le componen).

Otras clases importantes son *Staff*, que posee información como *indicador de compás*, *tonalidad*, *afinación*, *anacrusa*, *espacio* (que guarda el tamaño acumulado por sus elementos), *cantidad de compases* y *contenido*. La variable “*contenido*” es una lista de elementos, dado que *Staff* hereda a dos clases nuevas: *Phrase* y *Harmony*, siendo que *Phrase* es un objeto con las mismas características de *Staff* y cuyo contenido es una lista de objetos *Nota*. El objeto *Melody* es una lista de objetos *Phrase*. En cambio *Harmony*, que también hereda características de *Staff*, tiene como contenido una lista de acordes.

El código contiene sus respectivos archivos con los datos necesarios para: *reglas de las gramáticas generativas*, *diccionario de palabras*, *acordes disponibles*, *escalas disponibles* y *figuras rítmicas disponibles*. Estos datos se consultan siempre que es necesario en el código, y de modificarse directamente cambiarán los resultados de las piezas generadas a gusto de las modificaciones realizadas.

El proyecto es de código abierto y se encuentra de manera pública y gratuita como repositorio en la plataforma GitHub (Ramos Herrera, 2025a). Es accesible desde el url: <https://github.com/arhocoder/MIA>. El repositorio contiene toda la historia de versiones y de código agregado (junto a sus fechas y horas) además de piezas musicales generadas a lo largo del proceso y bastante material extra para encontrar pormenores del proyecto. Es posible especialmente revisar las versiones no utilizadas del módulo LWRs, siendo:

1. **Primer versión del módulo LWRs:** *commit* [f9fe2197b82696db7d809f85318d4ac3f325e22f](https://github.com/arhocoder/MIA/blob/f9fe2197b82696db7d809f85318d4ac3f325e22f/src/Melodic.py): <https://github.com/arhocoder/MIA/blob/f9fe2197b82696db7d809f85318d4ac3f325e22f/src/Melodic.py>.
2. **Segunda versión del módulo LWRs:** *commit* [a1907354059d3d6d722365bdd8927a14de3785d9](https://github.com/arhocoder/MIA/blob/a1907354059d3d6d722365bdd8927a14de3785d9/src/Melodic.py): <https://github.com/arhocoder/MIA/blob/a1907354059d3d6d722365bdd8927a14de3785d9/src/Melodic.py>.
3. **Versión final del módulo LWRs:** *commit* [916c3509826d2664c641029463ede1e1bfc0203a](https://github.com/arhocoder/MIA/blob/916c3509826d2664c641029463ede1e1bfc0203a/src/Melodic.py): <https://github.com/arhocoder/MIA/blob/916c3509826d2664c641029463ede1e1bfc0203a/src/Melodic.py>.

Cualquier contribución es bienvenida.

9.2.a Estructura TEXTO

La estructura “*TEXTO*” se crea deshabilitando el módulo de Gramática Generativa y dando en cambio como entrada directamente las frases separadas por sílabas. Actualmente no se posee un método que separe una oración en sus componentes silábicos, aun si es factible con los componentes actuales, pues que se posee un banco con 678,831 palabras únicas (Fenollosa, 2019). La entrada actual entonces es directamente la separación manual de las sílabas y su indicación de tónicas, de manera que la salida del algoritmo es una *idea musical* con una musicalización perfectamente cantable de la letra introducida; perfecta para tanto para músicos, pero en especial para no-músicos quienes quieren comenzar a componer teniendo letras originales escritas.

9.2.b Estructura abab’

Esta estructura recibe como entrada las características exclusivamente dos frases melódicas. El algoritmo compondrá dos frases y una progresión de acordes para las mismas, posteriormente tomará como tercera frase una repetición de la primera junto a sus acordes asociados, y creará una variación (ya sea rítmica o de oración) de la segunda frase, creándola como cuarta frase y tomando sus acordes asociados y modificándolos, tratando de conseguir una variación. La variación de acordes se consigue con la funcionalidad del módulo de Algoritmo Genético que permite tomar la población ya creada y optimizada y seguir optimizando una cantidad definida de iteraciones extra. Estas iteraciones extra podrán producir en algunos casos exactamente la misma progresión

original (lo que significa que no se consiguió una progresión mejor) y en otros casos se obtendrá una progresión ligeramente distinta a la original; perfecta para implementar los acordes respectivos de la cuarta frase.

Esta función introduce la posibilidad (potencial para una implementación futura de interfaz de usuario) de reutilizar en una misma idea musical, estructuras ya creadas, como oraciones, frases, ritmos, acordes, etc., para que una idea musical no posea enteramente contenido aleatorio nuevo, sino que sea posible repetir estos elementos para propiciar la aparición de *motifs* o para hacer memorable a una pieza.

9.2.c Estructura libre

Esta conviene en la arquitectura clásica de MIA Harmfirst, que compone todas las oraciones, ritmos y acordes de manera libre, sin restricciones de repetición de ningún tipo.

9.3 Métrica de Estética Musical Perceptual

El marco de construcción de una métrica para las evaluaciones de piezas generadas por el algoritmo MIA así como para piezas de la música tonal contemporánea, conviene a los siguientes intereses:

- **La métrica debe ser perceptual:** Resulta complejo definir y cuantificar criterios objetivos de medición que puedan asignar un valor único de calidad, a una pieza musical, es decir, definir el cómputo que lleve la obtención de dicha medida mediante únicamente la pieza y la métrica. Sin embargo, para el escenario de este trabajo conviene más factible depender de un tercer elemento: *el escucha*; la desventaja de ello radica en que una métrica basada en la percepción propiciaría múltiples valores de calidad para una misma pieza, pues estos dependerían de quien escucha la pieza, en cambio, una ventaja es que esta permitiría detallar más libremente criterios objetivos de evaluación, que no estén sujetos a una verdad absoluta, sino que puedan valorarse desde el criterio del escucha. Un ejemplo de métrica no perceptual son las funciones objetivo en los módulos de Recocido Simulado y Algoritmo Genético; pues estas cuantifican de forma automática la calidad de los elementos en la pieza (ritmo y armonía), asignando un valor único a cada uno. Si bien las funciones objetivo dirigen a una pieza a encontrar criterios estéticos definidos, lo cierto es que sólo involucran un grupo selecto y finito de evaluadores de importancia, mismos que no representan en su completitud la variedad de estéticas en las piezas musicales, de ahí la necesidad de involucrar la percepción, que introduce de primera mano el criterio de escucha del individuo evaluador, y no se obliga a puntualizar todos los posibles componentes estéticos a evaluar en razón de construir una métrica automática.
- **La métrica debe estar enfocada a escuchas estudiados:** Parte de la intención de reflejar objetividad radica en considerar aspectos técnicos que sólo un estudioso en el análisis musical pueda distinguir y valorar, por lo que resulta coherente dirigir el mecanismo hacia individuos con conocimientos sólidos en la evaluación de calidad musical, en medida en que se han formado académicamente para ello.
- **Valoraciones esperadas:** La métrica debe considerar aquellos elementos que se espera que estén presentes en una obra generada por MIA, de la misma manera en que están presentes en la música que escuchamos. De lograrse incluir elementos que se demuestre que están presentes en piezas valoradas popularmente como de buena calidad, entonces debería ser esperable verlos presentes en resultados de la música del algoritmo.
- **Separación e integración:** La métrica debe permitir entender el desempeño de los componentes que integran una pieza, de manera individual, para señalar deficiencias o aciertos, del mismo modo que debe valorar los componentes de manera integrada.
- **Enfocar elementos básicos y universales:** Se deben separar aquellos elementos que son universales en piezas de la música tonal, en contraposición con componentes variables según el género u estilo, todo desde el análisis puro de la composición musical. Por ejemplo, la *instrumentación* no podría ser un elemento universal, pues una misma pieza compuesta puede interpretarse con distinta variedad de instrumentos musicales, y si se busca evaluar la composición en sí misma, el instrumento que le ejecute es indiferente a la calidad de la misma. La *interpretación* es otro componente que no puede ser universal bajo este contexto, debido a que, al igual que la instrumentación, el músico que ejecuta

la obra imprime mayor o menor expresividad, siendo variable e indistinto de la pieza musical en sí³⁸. Los componentes universales por valorar son *Melodía*, *Ritmo* y *Armonía*, pues son elementos que, indistintamente del género o estilo, son parte esencial en la construcción de la música tonal contemporánea. Otros factores importantes derivados de ellos pueden ser la *Intencionalidad* y la *Emotividad*, mismos que se pueden percibir en mayor o menor medida sólo a través de los componentes de melodía, ritmo y armonía, y que distinguen una obra bien trabajada, de algo generado por un algoritmo sin verdadero propósito de transmitir emotividad.

Bajo estos principios se decidió que la métrica sea un cuestionario, que a través de preguntas con valoraciones numéricas de escala entre 1 y 5, permitirían cuantificar la percepción del escucha con base en componentes clave en la estructura de las piezas. De esta manera, las preguntas, separadas en secciones de interés, permitirán valorar numéricamente las percepciones de músicos experimentados, permitiendo que, si bien la percepción puede generar múltiples valores de calidad a una misma pieza, con el uso de un número considerable de evaluaciones, se pueda aproximar un valor único obteniendo promedios. Además, el utilizar una escala permitirá no solamente colocar cada percepción en un número, sino que facilitará generar puntajes para hacer recuentos estadísticos para la identificación de deficiencias y aciertos.

A continuación se describen las cinco secciones de evaluación en la métrica, cada una con tres preguntas individuales con una escala del 1 al 5.

9.3.a Coherencia Estructural

Esta sección evalúa la unidad, el propósito y el equilibrio formal de la pieza. El objetivo es medir en qué grado la composición presenta un motivo reconocible, evidencia intencionalidad en la disposición melódica y armónica, y refleja un equilibrio formal que genere sensación de continuidad y coherencia. Para ello, se incluyen las siguientes preguntas:

1. **¿La melodía tiene una idea motivica clara (un tema reconocible)?**: Un motivo musical bien definido (**Consultar 6.1.b.2 Motivo**) aporta identidad y memoria a la pieza, según describía Bribitzer-Stull (2015). Un tema reconocible facilita la percepción de cohesión interna, permitiendo al escucha distinguir una “*huella*” melódica que se repetirá o variará a lo largo de la obra. En música tonal contemporánea, un motivo sólido suele ser la base para desarrollos temáticos y variaciones, por lo que su ausencia podría percibirse como azar o falta de dirección. Por lo tanto, al esperarse que las obras generadas por MIA eviten la percepción de aleatoriedad, es preciso medir en qué magnitud lo hacen, desde el punto de vista de los motivos. **La escala es:**
 - **Con un valor de 1:** Sin motivo discernible.
 - **Con un valor de 5:** El motivo es claro y creativo.
2. **¿Se percibe intencionalidad en la composición de melodía y armonía?**: La sensación de propósito en las relaciones entre melodía y armonía permite al escucha identificar una lógica interna. Si la sucesión de eventos (notas, acordes) parece arbitraria, se pierde la conexión emocional e intelectual. En contraste, la intencionalidad (que cada elección melódica o armónica aporte a un sentido arquitectónico) refuerza la idea de obra coherente y no de simple concatenación de sonidos. **La escala de puntajes se define entre:**
 - **Con un valor de 1:** Mucha aleatoriedad en la composición.
 - **Con un valor de 5:** Se percibe intencionalidad en la composición.
3. **¿Existe una sensación de equilibrio formal (simetría de la frase, repetición/contraste)?**: El equilibrio formal alude a la organización de las frases, la alternancia entre secciones y la distribución de tensiones y resoluciones en el tiempo. Una composición equilibrada suele emplear esquemas simétricos o segmentos contrastantes que garanticen variedad y unidad a la vez. Desde la perspectiva perceptual, el escucha valora esa armonía interna: si las frases parecen caóticas o disparejas (por

³⁸ La interpretación sí es un componente que se puede dejar impreso en una obra en partitura, en cuestiones como fraseos, dinámicas en la intensidad de ejecución, indicaciones de expresividad, etc. Sin embargo, para fines de evaluar a la composición desde el punto de vista de su estructura, se deja de lado la interpretación.

ejemplo, frases demasiado largas sin estructura reconocible), la pieza pierde la sensación de “forma” que caracteriza la música tonal. **La escala es:**

- **Con un valor de 1:** *Caótico o desequilibrado.*
- **Con un valor de 5:** *Equilibrado e intencional.*

En conjunto, estas tres preguntas permiten identificar si la obra posee un esqueleto temático sólido (motivo), si existe lógica interna (intencionalidad) y si esa lógica está organizada de manera equilibrada entre repetición y contraste, o si en contraste se percibe alta aleatoriedad y caos.

9.3.b Integración Armónico-Melódica

En esta sección se mide la coherencia entre la línea melódica y el acompañamiento armónico, así como el grado de funcionamiento tonal en la pieza. La armonía y la melodía son dos elementos inseparables en la percepción de la música tonal, pero ambas están relacionadas dentro de una pieza. Si el algoritmo genera cada una mediante un método distinto, es preciso que se logre una integración entre ambas que sea agradable a la escucha. Las preguntas incluidas buscan valorar esa interacción:

1. **¿Los acordes parecen tener un propósito en relación con la melodía?:** Cuando la armonía está al servicio de la melodía, cada acorde refuerza el contorno melódico, enfatiza momentos de tensión o resolución, y guía hacia una sensación de dirección tonal. Si, en cambio, los acordes se perciben meramente superpuestos, sin conexión aparente, la pieza puede resultar desconectada o carente de sentido armónico. **La escala es:**
 - **Con un valor de 1:** *Acordes aleatorios / no relacionados.*
 - **Con un valor de 5:** *Los acordes complementan la melodía muy bien.*
2. **¿La melodía en suma a la armonía se resuelven de una manera apropiada?:** La *resolución* es un elemento crucial en la música tonal (consultar **6.1.b.5 Reposo** y **6.1.b.6 Tensión**); las disonancias deben resolverse de forma que el escucha perciba una cadencia o un regreso a un centro tonal. Si surgieran disonancias sin resolución o con transiciones abruptas, la sensación podría ser de inestabilidad permanente. Evaluar este punto permite verificar si el algoritmo MIA respeta las convenciones de resolución, o si, intencionalmente, ofrece ambigüedad como recurso creativo. **La escala de valores y su significados:**
 - **Con un valor de 1:** *Hay disonancias no resueltas.*
 - **Con un valor de 5:** *La melodía se resuelve de manera elegante.*
3. **¿La armonía crea un centro tonal claro o alguna ambigüedad intencional?:** Un centro tonal definido brinda al escucha un punto de referencia estable y es el elemento más importante que define a una pieza musical de la música tonal; si se tiene intención de generar ejemplos de música tonal, el centro tonal es esencial. **La escala es:**
 - **Con un valor de 1:** *Tonalmente confuso.*
 - **Con un valor de 5:** *Tonalmente intencional o creativamente ambiguo.*

En su conjunto, la sección de Integración armónico-melódica evalúa qué tan correctamente se complementan e integran la melodía y la armonía, tanto en su consonancia como en su seguimiento de reglas de la música tonal.

9.3.c Fluidez Rítmica

El ritmo está llevado a cabo mediante el Recocado Simulado, por lo que la percepción de calidad reflejada en este apartado permitirá entender el desempeño del módulo en la calidad final de la pieza. Evalúa:

1. **¿El ritmo de la melodía se siente natural y con un propósito?:** Una melodía con ritmo inconexo, saltos bruscos o acentos inesperados puede resultar confusa para el oyente y propiciar a menor calidad estética. Un ritmo natural en cambio mantiene coherencia y constancia en los acentos según las reglas de acentuación (Consultar **6.1.a Conceptos básicos – tiempos fuertes y débiles**). Con esta pregunta se busca medir si el fraseo melódico respeta reglas básicas de métrica y al mismo tiempo crea interés. **La escala de valores es:**
 - **Con un valor de 1:** *El ritmo es inconexo.*
 - **Con un valor de 5:** *El ritmo fluye naturalmente.*

2. **¿El ritmo armónico se alinea con el fraseo de la melodía?:** Cuando los cambios de acordes caen en momentos que refuerzan el sentido de fraseo, la sensación de coherencia aumenta. Por ejemplo, un acorde que cambia en medio de una frase melódica sin justificación puede interrumpir la línea. Al alinear la cadencia armónica con el fraseo melódico, la pieza adquiere fluidez y sentido de dirección.

Su escala de valores es:

- **Con un valor de 1:** *Los acordes cambian arbitrariamente.*
 - **Con un valor de 5:** *El ritmo armónico mejora el fraseo.*
3. **¿Se identifican patrones rítmicos distintivos?:** Más allá de la fluidez básica, la presencia de patrones rítmicos bien definidos (por ejemplo, una célula rítmica que se repite con variaciones) contribuye a generar identidad musical. Este evaluador busca identificar si MIA produce ritmos coherentes que, aunque simples y aleatorios, sean reconocibles y contribuyan a la memorabilidad de la pieza. **Su escala de valores es:**
- **Con un valor de 1:** *Ritmo sin patrones claros.*
 - **Con un valor de 5:** *Ritmo coherente.*

La evaluación de la fluidez rítmica refiere a la percepción temporal: un algoritmo que crea ritmos arbitrarios o desconectados genera sensaciones de caos o fragmentación. De esta manera, medir la naturalidad, la sincronía con los cambios armónicos y la existencia de patrones, garantiza que MIA proporcione una base rítmica sólida; más específicamente el módulo de Recocido Simulado.

9.3.d Impacto Emocional y Expresivo

En esta sección se valora la capacidad de la pieza para evocar emociones, articular frases de modo perceptible y generar memorabilidad. Aun si la estructura y la coherencia técnica son esenciales, un gran factor diferenciador es la expresividad. Si bien esta sección es subjetiva, como se analizó anteriormente en **6.1.a Conceptos básicos – acordes**, la combinación de sonidos suele estar vinculada a ciertas emociones; como los acordes mayores y menores asociados a la alegría y la tristeza respectivamente. De este modo se sabe que la estructura melódico-armónica puede propiciar algún impacto emocional, en medida que se aprovechen correctamente los componentes compositivos para explotar una u otra emoción. Se evalúa:

1. **¿La pieza evoca un estado de ánimo o emoción específicos?:** Una obra musical, aun siendo breve, debe sugerir al escucha un clima afectivo: tristeza, alegría, melancolía, tensión. Esta pregunta permite medir si MIA consigue transmitir alguna cualidad afectiva reconocible, o si el resultado se percibe neutro e indiferente. **La escala es:**
- **Con un valor de 1:** *Sin carácter emocional.*
 - **Con un valor de 5:** *La emoción es vívida y consistente.*
2. **¿Se identifica la separación de las frases y/o la articulación de ideas en la melodía?:** La articulación; es decir, el modo en que las frases se diferencian entre sí, mediante pausas, acentos o respiraciones imaginarias, contribuye al carácter expresivo. Cuando el oyente percibe con claridad dónde acaban y empiezan las ideas musicales, se propicia a que exista una expresividad bien diferenciada. **Su escala es:**
- **Con un valor de 1:** *No se identifica la separación de frases.*
 - **Con un valor de 5:** *Se siente la separación de frases.*
3. **¿La música se siente emocionalmente atractiva o memorable?:** A Más allá de identificar una emoción puntual, se busca determinar si la pieza deja una huella en la memoria del evaluador. Una obra con alto nivel de memorabilidad suele caracterizarse por “*motivicismo*” eficaz, progresiones armónicas sugestivas o ritmos marcados. **Su escala es:**
- **Con un valor de 1:** *Olvidable.*
 - **Con un valor de 5:** *Muy atractiva/memorable.*

Con estos criterios y sin dejar de ser perceptuales, se valora el impacto que la música genera en un escucha formado, equiparando la calidad estética con la capacidad de transmitir emocionalmente. Este aspecto, a pesar de interpretarse como subjetivo, evalúa de manera objetiva la utilización de recursos musicales para propiciar emocionalidad e impacto.

9.3.e Elementos individuales

Esta última sección desglosa los tres componentes principales (armonía, melodía y ritmo) para evaluarlos por separado, sin considerar su interacción. El propósito es detectar fortalezas o debilidades en cada elemento básico, de manera que, si la pieza falla en integración pero destaca en uno de ellos por separado, se detecte claramente. Esto facilita la identificación de áreas de mejora para el algoritmo o respondería a las preguntas de la posibilidad de implementar los distintos tipos de algoritmos en los distintos módulos de MIA. Se evalúa:

1. **¿La progresión de acordes es buena por sí misma (sin considerar la melodía)?:** Independientemente de la melodía, la progresión de acordes debe destacar por sí misma. **La escala:**
 - **Con un valor de 1:** *Progresión sin identidad propia.*
 - **Con un valor de 5:** *Progresión buena y satisfactoria.*
2. **¿La melodía es buena por sí misma (sin considerar los acordes)?:** De igual manera, la melodía debe brillar individualmente. **La escala es:**
 - **Con un valor de 1:** *La melodía no es buena por sí misma".*
 - **Con un valor de 5:** *La melodía es buena y satisfactoria.*
3. **¿El ritmo de las notas se adecúan bien al tipo de compás?:** Debe haber consistencia entre el tipo de compás elegido y los acentos rítmicos. **La escala es:**
 - **Con un valor de 1:** *El ritmo no se adecúa bien al tipo de compás.*
 - **Con un valor de 5:** *El ritmo distingue bien el compás.*

Con los resultados de esta sección es posible identificar los módulos mejor configurados, así como las áreas de mejora para futuras modificaciones de la arquitectura MIA Harmfirst a1.

9.3.f Cómputo de puntuación

Definiendo $r_{p,s}$ como la calificación (entero entre 1 y 5) que un evaluador le otorga a la pregunta p de la sección s , con: $s = 1, 2, 3, 4, 5$, $p = 1, 2, 3$, entonces se obtiene:

1. **Puntaje por sección:** Para cada sección s , se suma la calificación de sus tres preguntas:

$$Score_{sec\ p} = \sum_{s=1}^3 r_{p,s} \Rightarrow Score_{sec\ p} \in [3, 15] \quad (14)$$

2. **Puntaje total:** El puntaje total de una pieza musical (para un evaluador se obtiene sumando los cinco puntajes de sección:

$$TotalScore = \sum_{p=1}^5 Score_{sec\ p} = \sum_{p=1}^5 \sum_{s=1}^3 r_{p,s} \Rightarrow TotalScore \in [5 * 3, 5 * 15] = [15, 75] \quad (15)$$

3. **Puntaje porcentual:** Definiendo el máximo valor posible como $MaxTotal = 5(\text{secciones}) * 3(\text{preguntas}) * 5(\text{máximo por pregunta}) = 75$ y $MinTotal = 15$, el puntaje porcentual se calcula como:

$$TotalScore\% = \frac{TotalScore - MinTotal}{MaxTotal - MinTotal} * 100 \quad (16)$$

De esta manera:

- Si $TotalScore = 15$, entonces $TotalScore\% = 0\%$.
- Si $TotalScore = 75$, entonces $TotalScore\% = 100\%$.
- Para cada sección s el porcentaje dentro de ese bloque se puede calcular igualando 3 como 0% y 15 como 100%:

$$Score\%_{sec\ p} = \frac{Score_{sec\ p} - 3}{15 - 3} * 100 \quad (17)$$

La fórmula porcentual de puntaje global en la ecuación 16, así como la ecuación 17 de puntaje porcentual por cada sección, serán los cálculos principales para reportar en la sección de resultados, pues reflejarán un promedio de la opinión en los evaluadores.

9.3.g Cuestionamientos extra

Adicional a los cuestionamientos técnicos separados en secciones, se hicieron dos cuestionamientos extra:

1. **¿Del 0 al 10 qué calificación le das a la pieza?:** Esta pregunta permitirá conocer el gusto personal (y puramente subjetivo) del individuo respecto a la obra evaluada. Su valor es una escala entre 0 y 10, donde 0 equivale a total disgusto por la pieza y 10 un gusto total.
2. **¿Tiene potencial para convertirse en una canción/obra musical completa?:** Esta pregunta intenta saber si las piezas generadas por el algoritmo pueden tener el potencial para inspirar una obra completa; siendo este el objetivo del proyecto completo.

9.4 Implementación de la métrica

Para realizar los experimentos de evaluación se construyó una página *web* que permite a cada usuario evaluador interactuar, tanto con videos de las piezas a evaluar, como con las preguntas que conforman la *Métrica de Estética Musical Perceptual*. Esta herramienta *web* facilita la recolección de resultados en las evaluaciones, de manera que basta únicamente una computadora con audio y la colección de videos a cargar con las piezas musicales a evaluar. La sencilla interacción con las preguntas y su fácil respuesta da comodidad al usuario evaluador, para evaluar, cambiar su respuesta, avanzar en las preguntas, retroceder, etc.

El repositorio de código de la herramienta *web* se encuentra disponible en la siguiente dirección:

- GitHub: <https://github.com/arhcoder/MIA-Experiments> (Ramos Herrera, 2025c).

Adicional al código, actualmente se encuentra desplegada una versión en línea de la herramienta:

- Despliegue: <https://arhcoder.github.io/MIA-Experiments> (Ramos Herrera, 2025d).

La implementación en línea contiene como inicio un formulario de datos generales para almacenar la información del evaluador. La **Figura 36** muestra esta interfaz de inicio, sobre la que se necesitan llenar los datos de “Nombre”, “Género”, “Edad”, y “Semestre” (siendo este dato solicitado especialmente para la intención de evaluar estudiantes de la licenciatura en música, pero no necesario en otros casos).

Figura 36. Cuestionario inicial de la interfaz *web* para la Métrica de Estética Musical Perceptual.

El formulario inicial muestra además una opción para “*Seleccionar carpeta songs*”, que ejecutará un explorador de archivos en la computadora para seleccionar una carpeta que contenga todos los videos (en formato MP4) de las piezas musicales que se pretenden evaluar. Finalmente el botón “*Iniciar Evaluación*” introduce la sección de preguntas aplicada a cada pieza musical a evaluar.

La **Figura 37** muestra la interfaz principal de evaluación para la métrica. Se contiene primeramente el título de la pieza a evaluar (en este caso “*MIA P01*”), junto a un contador de la cantidad de piezas evaluadas con respecto a la cantidad a evaluar (“*1 / 14*”). Sobre el centro se contiene un video reproductor con sus controladores para reproducir, pausar, reiniciar, subir volumen, colocar pantalla completa, etc., al igual que un barra de progreso para colocar partes específicas de la reproducción. Finalmente en la parte inferior de la interfaz se contiene una caja con las preguntas a evaluar, primeramente mostrando el título de la sección (“*COHERENCIA ESTRUCTURAL*”) así como el número de sección (“*1 / 5*”). Se presenta la pregunta y además una serie de botones de escala entre 1 y 5 con sus respectivos textos de complemento. Al presionarse un botón de la escala, se habilita la opción de avanzar de pregunta, a la par que se señala de un color más oscuro el botón seleccionado (en este caso “*4*”). Es posible retroceder entre preguntas mientras existan.

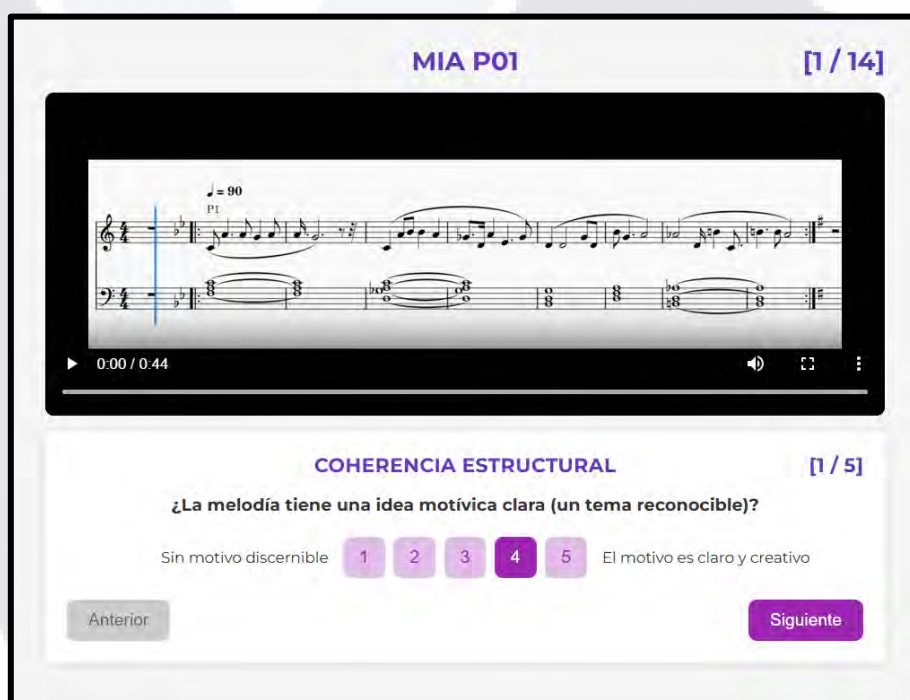


Figura 37. Interfaz principal del instrumento web de evaluación de la Métrica de Estética Musical Perceptual.

Esta herramienta en línea tiene mecanismos constantes para el almacenamiento de la información de respuestas. Cada que se termina con una pieza musical, el sitio web arroja un archivo del formato JSON que almacena las respuestas del evaluador, como medida de precaución ante posibles errores en la herramienta durante el transcurso de una pieza en el proceso. Adicional a ello, al finalizar con las evaluaciones se almacena un nuevo archivo JSON con la recopilación completa de respuestas, al igual que la información del usuario evaluador. Este almacenamiento se da mediante la descarga del archivo en la computadora que ejecuta la herramienta, por lo que no existe ningún almacenamiento en servidores de internet.

La **Figura 38** muestra los bloques para las preguntas extra a la métrica, tanto para la evaluación subjetiva entre 0 y 10 sobre el gusto en la pieza musical, como para la pregunta sobre el potencial inspiracional en la pieza para convertirse en una obra musical completa.

PREGUNTAS EXTRA [6 / 5]

¿Del 0 al 10 qué calificación le das a la pieza?

0 1 2 3 4 5 6 7 8 9 10

Anterior Siguiente

PREGUNTAS EXTRA [6 / 5]

¿Tiene potencial para convertirse en una canción/obra musical completa?

Sí No

Anterior Siguiente

Figura 38. Bloques con las preguntas extra en la herramienta web de evaluación.

10 Experimentos

Ya superadas las fases 4 (*Evaluación inicial del algoritmo*) y 5 (*Formulación de la métrica*), la fase 6 (*Evaluación Experimental del Algoritmo*) contempla utilizar dos elementos: un conjunto de piezas generadas por MIA Harmfirst a1 y uno de piezas musicales reales; con base en estos e implementando la *Métrica de Estética Musical Perceptual* en su herramienta web, se pretende que un grupo de músicos evaluadores con formación académica en la música, ofrezcan valoraciones sobre todas las piezas. El método implica seleccionar a un grupo de sujetos que no conozcan ninguna de las piezas reales ya existentes, de modo que pueda indicárseles que todas las piezas son generadas por el algoritmo, buscando evitar sesgos de percepción producto de los orígenes artificial y natural de música. Aun si éstos llegasen a reconocer alguna pieza, la evaluación seguiría siendo válida.

10.1 Planeación

Respecto a la recolección de piezas musicales, se decidió conformar los grupos de piezas con siete ejemplos cada uno. Siete piezas de algoritmo y siete piezas de compositores humanos, conformando un conjunto de catorce piezas. Se decidieron utilizar siete piezas de cada tipo como una forma de permitir crear subgrupos dentro de cada grupo, para explorar distintas características y sus resultados de calidad.

10.1.a Piezas musicales

Para la fase experimental se contempla crear grupos de piezas de los distintos orígenes, pero cada uno con un subgrupo específico para comprender el impacto de las distintas estructuras de algoritmo. Para el conjunto de piezas del algoritmo MIA se pretenden crear piezas para las estructuras “*TEXTO*”, “*abab*” y “*libre*” (consultar secciones 9.2.a, 9.2.b y 9.2.c), con dos, dos y tres piezas para cada uno respectivamente.

10.1.a.1 Piezas de MIA Harmfirst

Las piezas se crearon con los parámetros por defecto descritos en 9.1.a **Entradas y parámetros de preferencias**³⁹, con piezas seleccionadas como mejores entre (aproximadamente) ocho versiones generadas; es decir, que cada pieza elegida finalmente es la mejor entre ocho generadas por el algoritmo⁴⁰. Las siete piezas son:

1. **P01: “Desaparecer” [texto]:** Esta pieza se mostró anteriormente en la **Figura 32**. Se creó utilizando versos de Serafio Méndez (2025a). Se eligió la tonalidad de Sol menor (Gm), sin anacrusa, y con dos compases para cada frase. Se generó con la versión segunda del módulo LWRS (sin contorno de frase). Se utilizaron cuatro oraciones, cada una con un acorde exceptuando la tercera, con dos acordes.

³⁹ No exactamente todos los parámetros idénticos, pero con ligeras variaciones no registradas.

⁴⁰ “La mejor entre ocho” aplica para las piezas excepto P01.

2. **P02:** “¿Cuántas pasquinase?” [abab’]: Se utilizó la estructura “abab’”, es decir, dos frases libres y después la repetición de los mismos, con variación de la última, al igual que para la progresión de acordes. Se indicó una anacrusa de 1 tiempo en compás de 4/4. Se utilizó la tonalidad de Sol mayor (G) usando dos compases y dos acordes para cada oración. Se usaron contornos de “subida” para las oraciones. Se muestra en la **Figura 39**.

¿Cuántas pasquinase?
P02

ARH MIA

$\text{♩} = 94$

Oboe

Piano

D7 G D7#5 G D7 G D7#5 Gsus4 G

Figura 39. Pieza de algoritmo P02: “¿Cuántas pasquinase?”

3. **P04:** “Mi dulce Tierra” [texto]: Se utilizaron tres versos de Serafio Méndez (2025b): “Mi dulce Tierra”, “Respiro en ti”, “Queriendo cuidar los recuerdos de un ayer”, con los contornos “subida”, “bajada”, “valle” respectivamente. Dando un compás y un acorde a las dos primeras frases y dos compases y acordes a la última. Se usó tonalidad de Fa mayor (F) con anacrusa de 1 tiempo en 4/4. La pieza generada se muestra en la **Figura 40**.

Mi dulce Tierra
P04

ARH MIA

Versos de Ruth Serafio

$\text{♩} = 74$

Oboe

Piano

Bb7 F C7 F

Figura 40. Pieza de algoritmo P04: “Mi dulce Tierra”.

4. **P06:** “¿Cuáles esos morigeran?” [libre]: Compuesta con estructura libre en la tonalidad de Do mayor (C), sin anacrusa y con cuatro frases, cada una con dos acordes y compases. Se indicó que la frase 3 fuera repetición (sólo de oración) de la frase 1, y que la frase 4 sea repetición (oración y ritmo) de la segunda oración. No se registraron los contornos utilizados. Se ilustra en la **Figura 41**.

¿Cuáles esos morigeran?
P06

ARH MIA

$\text{♩} = 74$

Oboe

Piano

C G7 C Em Bdim Dm F7 C

Figura 41. Pieza de algoritmo P06: “¿Cuáles esos morigeran?”

5. **P09: “Ajuntases” [abab’]:** Se eligió la estructura “abab’”, además, se probó utilizar una escala modal: Re Lidio. Sin anacrusa y con un compás de 3/4, tres compases por cada frase y tres acordes para las primeras dos frases (con otros tres acordes de variación para las últimas dos frases). Se utilizaron contornos de “subida” para todas las frases. Para esta pieza se buscó no sólo explorar el uso del compás 3/4, sino además de una escala modal; además, se implementó que la última frase fuese una variación de la segunda, pero de forma manual en su escritura, como experimento de la posibilidad de crear variaciones de oraciones y no sólo de ritmo. Se ilustra en la **Figura 42**.

Ajuntases
P09

ARH MIA

Oboe

Piano

E A7 D E A7 D

Figura 42. Pieza de algoritmo P09: "Ajuntases".

6. **P10: “¿Quién refrigerador enzurdecieres?” [libre]:** Nuevamente un compás 4/4, en tonalidad de Do mayor (C), sin anacrusas, con cuatro oraciones: dos compases por frase y dos acordes por cada una, exceptuando la última, con cuatro acordes. No se registraron los contornos musicales, pero se configuró para que las dos primeras frases fuesen originales, la tercera una copia sólo de la oración primera y la cuarta siendo una repetición de oración y ritmo de la segunda. Se ilustra en la **Figura 43**.

¿Quién refrigerador enzurdecieres?
P10

ARH MIA

Oboe

Piano

Em B7 C G Dm Am G7 Aug Gm C

Figura 43. Pieza de algoritmo P10: "¿Quién refrigerador enzurdecieres?".

7. **P14: “Me acompañó” [texto]:** Compuesta utilizando cuatro versos del poema “Tres Jacarandas” (Serafio Méndez, 2023): “Me acompañó hasta el río”, “Y mirando el infinito”, “Caí en llanto”, “Sin poder soltar mi esperanza”. Se utilizó como tonalidad Si bemol dórico y un compás de 4/4. Las frases tienen los contornos respectivos de “arco”, “arco”, “bajada”, “bajada”. Ilustrada en la **Figura 44**.

Me acompañó
P14

ARH MIA

Versos de
Ruth Serafio

Oboe

Piano

Cm Gdim G# Eb7 Bbm

Figura 44. Pieza de algoritmo P14: "Me acompañó".

10.1.a.2 Piezas de compositores reales

Para seguir la misma lógica de tres subgrupos (respecto a las composiciones del algoritmo), se decidió seleccionar tres compositores distintos para tomar sus piezas como parte del conjunto de música real. Se utilizaron principalmente a los compositores Nobuo Uematsu y Toby Fox, siendo estos ampliamente galardonados pues sus obras han inspirado fuertemente a la cultura popular. Al necesitar siete piezas (cantidad impar), se utilizó además una pieza extra compuesta por el autor de este trabajo (Ramos Herrera).

Nobuo Uematsu es un compositor japonés nacido en 1959; famoso por haber creado música para trece de los dieciséis títulos de la exitosa saga de videojuegos “*Final Fantasy*” entre 1987 y 2016 (Guinness World Records, 2016). Su música es mayormente de nicho, lo que quiere decir que si bien es ampliamente conocida, esto es sólo para quienes están adentrados en el género jugabilístico “*RPG japonés*” (al que pertenecen los juegos de la saga). Algunos de sus galardones como compositor son:

- **Premio “Disco de Oro de Japón” (1999):** Su canción “*Eyes on me*”, para el videojuego “*Final Fantasy VIII*”, ganó “*la Canción del Año – Música Occidental*” en los 14º Premios Disco de Oro de Japón, siendo la primera vez que una canción de videojuego logró este honor (Low, 2019).
- **Revista Time – Time 100: “The Next Wave – Music” (2001):** Reconocido como uno de los principales innovadores musicales a nivel mundial por unir los videojuegos y la música orquestal/pop (Low, 2019).
- **Salón de la Fama de Classic FM (2017):** Su “*Tema de Aerith*” (“*Final Fantasy VII*”) se convirtió en la primera pieza de videojuego en ingresar al Salón de la Fama anual (en el puesto 16º en 2012), seguido por una posición máxima en el 3º en 2013, y se mantuvo entre los 20 primeros al menos hasta 2016 (Atlantic Award Group, 2017).
- **Récord Mundial Guinness (2016):** Obtuvo el título de “*Compositor con la música de videojuegos más interpretada*” en la historia (Guinness World Records, 2016).
- **Nominación al premio BAFTA de videojuegos (2007):** Co-nominado a *mejor banda sonora original* junto con Hitoshi Sakimoto por “*Final Fantasy XII*” (BAFTA, 2007).
- **Premio Golden Joystick a la trayectoria:** Reconocido con este premio, que honra su impacto duradero en la música de los videojuegos (Low, 2019).
- **Iniciativa Premio Atlántico (2017):** Galardonado con un premio por contribución especial en el concierto *Distant Worlds* en Newark, Nueva Jersey, en reconocimiento a sus extraordinarias contribuciones a la industria (Atlantic Award Group, 2017).

Bajo este panorama de la calidad de su trabajo, se seleccionaron tres piezas de la banda sonora del videojuego “*Final Fantasy VI*” de 1994. Este juego fue anterior a la inmensa fama de la saga en occidente, pues fue apenas el tercer título de la franquicia en ser publicado en occidente (Estados Unidos); su banda sonora es, en opinión del autor, una obra maestra que demuestra la gran capacidad de su autor para crear melodías memorables y con fuerte carga emocional, y está compuesta con principios “*motívicos*” para personajes de la obra. Se seleccionaron “*Edgar & Sabin’s Theme*”, “*Terra’s Theme*” y “*Celes’s Theme*” (Uematsu, 1994). Se trasladaron sus melodías y armonías al formato de una partitura con la melodía interpretada por oboe y armonía interpretada por piano.

La pieza “*Edgar & Sabin’s Theme*” está en la tonalidad de Si bemol mayor (Bb), y contiene una estructura “*abab*”, es decir, sus dos primeras frases melódicas contienen una progresión de acordes definida, después estas se repiten variando la última frase y sus acordes. La pieza “*Terra’s Theme*” a pesar de contener las mismas notas en la escala que el tema anterior, su tonalidad es Sol menor (su relativo menor; es decir, Gm); esta tiene una distribución más compleja en sus ocho frases, que son repeticiones muy fieles entre sí. En cambio, “*Celes’s Theme*” figura más como una composición de frases libre, con una larga progresión de acordes que acompaña a sus seis frases melódicas. Está compuesta en la tonalidad de Re mayor, con una anacrusa de 1/4 al igual que el tema de Terra. La **Figura 45** muestra la partitura con las tres piezas elegidas.

Obras seleccionadas de Nobuo Uematsu

Del videojuego Final Fantasy VI

Nobuo Uematsu

The image displays three musical staves, each representing a different theme from the video game Final Fantasy VI, composed by Nobuo Uematsu. Each staff includes a melody line for Oboe (Ob.) and a piano accompaniment line (Pno.).

- P05 Edgar & Sabin's Theme:** The tempo is marked as $J = 92$. The piano accompaniment features a sequence of chords: Cm, Cm, Cm, Ab, Bb, Eb, F6, Cm, Cm, Cm, Ab, Bb, and Csus4 C.
- P07 Terra's Theme:** The tempo is marked as $J = 80$. The piano accompaniment features a sequence of chords: Gm, Dm, Gm, Dm, Bb, F, Eb, Cm, and Gm.
- P12 Celes's Theme:** The tempo is marked as $J = 60$. The piano accompaniment features a sequence of chords: D, F#m, G, F#m, Em, D, C#, and F# A7.

Figura 45. Piezas musicales elegidas, del compositor Nobuo Uematsu, del videojuego Final Fantasy VI.

Nota: Transcripción realizada con base en la banda sonora original (Uematsu, 1994).

Para el compositor americano Toby Fox, se eligió la banda sonora de su ópera prima: “Undertale”. Fox figura como un compositor renombrado en la industria de los videojuegos independientes, dado su gran reconocimiento en la comunidad de jugadores, a pesar de haber desarrollado (además de su proyecto actual en desarrollo: “Deltarune”) solamente un videojuego. Este artista funge no sólo como compositor, sino como escritor, diseñador y programador de su obra. Aun si no lo ha mencionado expresamente el autor, se sabe popularmente que la inspiración para *Undertale* (tanto en la historia como en la música) son los RPG japoneses. Incluso un tema musical en su banda sonora, “Oh! One True Love”, es una referencia bastante evidente del tema “Aria di Mezzo Carattere” del título “Final Fantasy VI”, siendo este último una versión extendida de “Celes’s Theme”; Toby no solo toma los instrumentos digitales originales del juego japonés, sino que recrea una parodia de la escena en que suena la pieza mencionada. Toby Fox, nacido en 1991, con apenas 24 años publica *Undertale*, y se hace acreedor de múltiples galardones, y si bien, la mayoría de estos refieren a distintas partes de su creación, algunos cuantos se enfocan específicamente en su música, por ejemplo:

- **Mejor banda sonora – Premios Vidya Gaem (2015):** Undertale ganó el 1er lugar a *Mejor banda sonora* (Vidya Gaem Awards, 2015).
- **Premios de videojuegos Wizard Dojo – Mejor música (2016):** Celebrada a principios de 2016, reconocieron a Undertale como la banda sonora destacada de 2015 (Wizard Dojo, 2017).
- **Premios Steam 2017 – “Mejor banda sonora”; Categoría Comunidad (2017):** Undertale fue nominado a la categoría de *Mejor Banda Sonora* creada por la comunidad.

La música de Fox posee más galardones de comunidades de internet, y esta se ha interpretado en giras e internacionales de conciertos sinfónicos. GamesRadar clasificó a Undertale entre las “25 mejores bandas sonoras de todos los tiempos”, siendo la segunda mejor (Bentley, 2025).

Se eligieron las piezas “It’s Showtime!”, “Determination” y “Battle Against a True Hero” (Fox, 2015). Se muestran sus transcripciones en la **Figura 46**.

Obras seleccionadas de Toby Fox

Del videojuego Undertale

Toby Fox

The image displays three musical staves for the video game Undertale, each featuring an Oboe (Ob.) and Piano (Pno) part. The first piece, 'P03 It's Showtime!', is in 4/4 time with a tempo of 126 bpm. The piano accompaniment consists of chords: Dm, G, C, A7, Dm, G, C, C. The second piece, 'P11 Determination', is in 4/4 time with a tempo of 115 bpm. The piano accompaniment consists of chords: Ebm, Ab, B Db, Ebm, Ebm, Ab, B Db, Ebm. The third piece, 'P13 Battle Against a True Hero', is in 4/4 time with a tempo of 150 bpm. The piano accompaniment consists of chords: Db, Eb, Eb, Fm, Db, Eb, Fsus4, F, Db, Eb, Eb, Fm, Db, Eb, Fsus4, F.

Figura 46. Piezas musicales elegidas, del compositor Toby Fox, del videojuego Undertale.

Nota: Transcripción realizada con base en la banda sonora original (Fox, 2015).

La pieza “It’s Showtime!” está escrita en la tonalidad de Do mayor (C) y contiene una estructura “*abab*”, es decir, sus dos primeras frases se acompañan de una armonía (Dm – G – C – A7) y posteriormente se tiene una repetición de la primer frase y una variación de la segunda, teniendo una armonía ligeramente variada (Dm – G – C – C). “Determination” contiene la misma estructura, siendo una pieza en la tonalidad de Si bemol menor (Ebm). El tema “Battle Against a True Hero” no sólo destaca por su alta velocidad (un tiempo de negra igual a 150 bpm), sino a que utiliza una doble estructura “*abab*”: dos frases melódicas acompañadas por sus acordes (Db – Eb y Eb – Fm), después una repetición de la primer frase con mismos acordes, una variación de la segunda, con variación de acordes (Fsus4 – F en lugar de Eb – F); finalmente, remata con una repetición de la estructura “*abab*” antes descrita, variando nuevamente la última frase, aunque conservando los acordes antes variados (Fsus – F).

Finalmente, se tomó el tema principal de una pieza compuesta por el autor de este trabajo, que no se encuentra pública en ninguna fuente pública; el tema “Sandy”, que se encuentra ilustrado en la **Figura 47**. Este tema posee una estructura “*abab*”, con dos frases melódicas y su progresión, repetidas variando la última frase y sus últimos acordes, con una resolución de Sus4 a mayor.

Sandy

P08

Alejandro Ramos
@arhocoder

The image displays the musical score for the piece 'Sandy'. It features an Oboe (Ob.) and Piano (Pno) part. The tempo is 60 bpm. The piano accompaniment consists of chords: A, A, G, G, A, A, G, D, Fmaj7, Esus4 E.

Figura 47. Pieza musical humana extra P08: “Sandy”.

De esta manera se constituyen las catorce piezas musicales necesarias para realizar evaluaciones experimentales a través de la *Métrica de Estética Musical Perceptual*, para así obtener estadísticas de calidad comparables entre música automática de MIA y música de artistas galardonados y reconocidos, tomando piezas humanas, que a pesar de ser populares, son conocidas entre un nicho de personas, lo que facilita encontrar músicos evaluadores que no les conozcan, para así indicárseles que son realizadas por el algoritmo MIA.

10.1.a.3 Conjunto de piezas de evaluación

Los nombres identificadores “P01”, “P02”, ..., “P14” están dados de forma aleatoria. La lista de temas originales se reordenó de esta manera, para ser indistinto el orden en que el usuario evaluador los escucha. La **Tabla 5** contiene una referencia de los temas a evaluar y sus características.

Tabla 5. Conjunto completo de piezas musicales a evaluar.

Identificador	Compositor	Título	Estructura
P01	 MIA	<i>Desaparecer</i>	Texto
P02	 MIA	<i>¿Cuántas pasquinase?</i>	abab'
P03	 Toby Fox	<i>It's Showtime!</i>	abab'
P04	 MIA	<i>Mi dulce Tierra</i>	Texto
P05	 Nobuo Uematsu	<i>Edgar & Sabin's Theme</i>	abab'
P06	 MIA	<i>¿Cuáles esos morigeran?</i>	Libre
P07	 Nobuo Uematsu	<i>Terra's Theme</i>	Libre
P08	 Alejandro Ramos	<i>Sandy</i>	abab'
P09	 MIA	<i>Ajuntases</i>	abab'
P10	 MIA	<i>¿Quién refrigerador enzurdecieres?</i>	Libre
P11	 Toby Fox	<i>Determination</i>	abab'
P12	 Nobuo Uematsu	<i>Celes's Theme</i>	Libre
P13	 Toby Fox	<i>Battle Against a True Hero</i>	abab' aba'c
P14	 MIA	<i>Me acompañó</i>	Texto

Ya constituida la lista de piezas a evaluar, así como su orden, se crearon videos con las partituras de cada una (removiendo los textos de títulos y/o sílabas), utilizando el software *Musescore 4* para reproducirlas, así como un capturador de pantalla de *Windows*, incorporando en un editor de video la reproducción visual capturada, coordinada con la pista de audio exportada en formato *WAV* directamente del software *Musescore*. Con los videos separados y ordenados en una carpeta, y teniendo ya lista la herramienta *web* para la métrica de evaluación, únicamente se necesitan los sujetos evaluadores.

10.1.b Sujetos evaluadores

Se decidió contactar estudiantes de la Licenciatura en Música de la Universidad Autónoma de Aguascalientes; concretamente del octavo semestre. Un estudiante de octavo semestre de la carrera habrá cursado para entonces (en el caso de no tener adeudo de materias; cosa que no aplica para ninguno de los sujetos seleccionados), las siguientes materias de interés, según el plan de estudios de la UAA (2017):

- *Armonía elemental.*
- *Teoría tonal y análisis I.*
- *Teoría tonal y análisis II.*
- *Armonía y análisis del siglo XIX.*
- *Modernismo: técnicas, análisis y contexto.*
- *Análisis musical: percepción e interpretación.*

La materia “*Análisis musical: percepción e interpretación*” es cursada durante el octavo semestre, por lo que este es un conocimiento adquirido durante el preciso período de experimentos (semanas finales del semestre en curso). Aunado a los ocho semestres de formación académica en la música, estas asignaturas anterior descritas nos garantizan que los estudiantes han pasado a través del aprendizaje formal del análisis musical, lo que hace valiosa su percepción en el proceso de experimentación.

Se asistió a la unidad de estudio de los estudiantes: el *Departamento de Música*, ubicado en *Gral. Álvaro Obregón 419, Zona Centro, 20000 Aguascalientes, Ags., México*, de forma que se tuvo la posibilidad de hablar con los estudiantes de octavo semestre y hacer el ofrecimiento de participar como evaluadores bajo las

características de los experimentos ya descritos. Hubo seis interesados, a los que se les preguntó si alguna vez habían escuchado la música de algunos de los videojuegos de la saga *Final Fantasy* o de *Undertale*. Según lo respondido, se descubrió que seis de los sujetos desconocían el rubro de la música de las obras en cuestión, así como uno de ellos mencionó conocer no sólo el género y las obras, sino ser admirador de las mismas. Este último sujeto se descartó y se agendó cita para realizar la evaluación con el resto de los cinco.

10.2 Ejecución

Gracias a la autorización del actual Jefe del Departamento de Música, el Mtro. en A.C. César Alfonso Guzmán Portal, se tuvo la oportunidad de utilizar las instalaciones (las aulas de práctica con asilamiento sonoro), para realizar las evaluaciones. Se utilizó una laptop con la herramienta de evaluación web y se les asignó a los evaluadores un par de auriculares *in-ear Huawei Freebuds SE 2 ULC-CT010* (desinfectados con alcohol antes de cada evaluación) para escuchar las piezas y contestar sus evaluaciones de forma individual y privada.

Antes de comenzar cada evaluación, se les hizo firmar un documento de consentimiento informado respecto a su participación en el experimento. Las firmas se encuentran presentes en la sección **Anexos – B Cartas de consentimiento de participación**; estas cartas confirman la autorización de participación, y, además informan si el sujeto desea o no ser mencionado en el presente documento, o si en cambio prefiere permanecer anónimo. Posterior a la firma de consentimiento, se les indicaron instrucciones con el siguiente mensaje aproximado:

“Estás a punto de escuchar una serie de catorce piezas musicales. Son piezas cortas, que consisten en una melodía de una sola voz interpretada por oboe y una armonía definida como una serie de acordes tocados por piano. Debes evaluar la calidad estética de las piezas considerando únicamente su composición, es decir, sin considerar valoraciones de interpretación, sino únicamente la unión de melodía y armonía. Para cada pieza tendrás que responder una serie de preguntas divididas en secciones que corresponden a distintos criterios estructurales de la pieza, procurando hacerlo de la manera más objetiva posible, escogiendo para cada pregunta un valor entre la escala del 1 al 5, donde un 1 representa la ausencia de dicha característica preguntada y un 5 la presencia total de la misma. Adicionalmente posees un par de preguntas extra, una respecto a tu valoración de gusto subjetivo en una escala entre 0 y 10, en la que debes responder con respecto a qué tanto te gustó la pieza, y otra respecto al potencial de la pieza para convertirse en una obra completa. Si necesitas descansar en cualquier momento, tanto para revisar tu teléfono, responder mensajes, hacer alguna llamada o en general para despejar la mente, eres libre de hacerlo, de la misma manera en que puedes hacer cualquier pregunta que tengas durante el proceso; estaré sentado por aquí [a una distancia aproximada de 2 metros y detrás del sujeto]. Es probable que durante el proceso tome un par de fotografías como evidencia; si prefieres que esto no sea así, puedes decírmelo. Para controlar el volumen del equipo puedes usar estos botones [teclas para subir y bajar volumen]. Si en cualquier momento decides terminar, eres libre de hacerlo, puedes tomarte tu tiempo. Estimo que la prueba tarde entre cuarenta minutos y una hora. Buena suerte y agradecemos por tu participación”.

Ya dadas las instrucciones para iniciar, los cinco sujetos citados realizaron sus evaluaciones respectivas. Los resultados y observaciones de cada sujeto se listan a continuación.

10.2.a Sujeto 01

El sujeto y su evaluación poseen las siguientes características:

- **Nombre del sujeto:** Óscar Rodríguez Frago.
- **Edad:** 22 años.
- **Instrumento principal:** Guitarra clásica.
- **Duración de evaluación:** 64 minutos sin considerar las pausas.
- **Fecha de evaluación:** Miércoles 14 de mayo de 2025 a las 11:00am.
- **Aula utilizada:** A-3.
- **Observaciones:**
 - ❖ No conocía las piezas de compositores reales, sin embargo mencionó que algunas de ellas tenían que serlo. Mencionó que había contraste entre piezas que sí podrían ser de algoritmo y otras que con seguridad no eran de algoritmo, por ser tan buenas.

- ❖ Dudó para la pieza P01 “Desaparecer” al responder la pregunta “¿La armonía crea un centro tonal claro o alguna ambigüedad intencional?”, pues no entendía si la pieza podía ser “creativamente ambigua” (parte de la descripción en el valor 5 de la escala) tonalmente hablando, se le pidió que se limitara a si la obra crea o no un centro tonal, o si en cambio cree que no lo hace, pero se siente intencionado.
- ❖ No comprendió la pregunta final referente al potencial de una pieza a ser una obra más grande; el sujeto preguntó si esto era tomando la pieza sin cambios o si había posibilidad de hacer modificaciones en los instrumentos, los acordes, las notas, etc., en este proceso de convertirla en una obra completa. Se le explicó que respondiera pensando en poder hacer ligeras modificaciones en la estructura o agregando instrumentación y componentes musicales extra.
- ❖ Mención el término “semilla” para referirse a que las piezas sí podrían ser la inspiración para crear algo más grande.
- ❖ Mencionó que para su perspectiva, dos o tres piezas son de compositores humanos.
- ❖ Tarareó algunas de las piezas, quizás inconscientemente, durante sus evaluaciones.
- ❖ Descansó aproximadamente seis minutos después de la séptima pieza evaluada. Durante ese tiempo hizo preguntas respecto al desarrollo del sistema y la intención del proyecto. Se le respondió con honestidad a sus planteamientos.
- ❖ Mencionó que el ejercicio, más que pesado fue divertido.
- ❖ Se mostró bastante interesado e impresionado por el proyecto, y al final de sus evaluaciones hizo bastantes preguntas sobre el proceso en todo. Mencionó además sorprenderle que el autor tuviese conocimientos tan sólidos de teoría musical.

- **Fotografía de evidencia:** La **Figura 48** contiene al Sujeto 01 realizando evaluación.

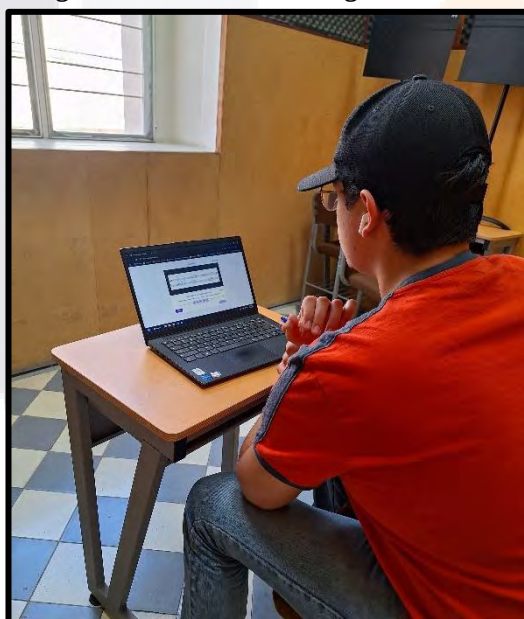


Figura 48. Sujeto 01 realizando evaluación.

- **Resultados de evaluación:** La **Tabla 6** contiene las respuestas del Sujeto 01.

Tabla 6. Resultados de evaluación del sujeto 01.

Pieza	Sección 1			Sección 2			Sección 3			Sección 4			Sección 5			Extra	
	p1	p2	p3	p1	p2	p3	p1	p2	p3	p1	p2	p3	p1	p2	p3	p1	p2
P01	4	4	3	5	2	4	2	5	2	3	2	1	2	3	3	4	Sí
P02	5	4	5	3	4	5	4	4	5	4	4	4	2	4	5	7	Sí
P03	4	3	4	4	3	4	4	4	5	5	5	3	3	4	5	8	Sí
P04	2	2	2	2	2	2	2	2	2	4	2	2	2	2	4	2	No

P05	5	5	5	5	5	5	5	5	5	5	5	4	4	4	5	9	Sí
P06	3	3	3	2	2	2	2	3	4	2	3	2	2	2	3	3	No
P07	5	5	5	5	4	5	5	5	5	5	4	4	4	4	5	9	Sí
P08	5	5	5	5	5	5	5	4	5	5	5	5	4	5	5	8	Sí
P09	3	3	4	3	3	2	2	3	3	2	2	2	3	2	4	2	No
P10	1	2	2	2	2	1	1	2	2	2	3	1	2	1	3	1	No
P11	5	5	5	5	4	5	5	5	5	5	5	5	4	5	5	9	Sí
P12	5	4	4	4	4	4	4	4	4	4	4	4	3	4	5	7	Sí
P13	5	5	5	5	5	5	5	5	5	5	5	4	4	5	4	8	Sí
P14	3	3	3	3	3	3	3	2	2	2	2	2	2	2	3	4	No

10.2.b Sujeto 02

El sujeto y su evaluación poseen las siguientes características:

- **Nombre del sujeto:** Anónimo a decisión del evaluador.
- **Edad:** 22 años.
- **Instrumento principal:** Viola.
- **Duración de evaluación:** 41 minutos sin considerar las pausas.
- **Fecha de evaluación:** Miércoles 14 de mayo de 2025 a la 1:00pm.
- **Aula utilizada:** A-1.
- **Observaciones:**
 - ❖ Escucha con calma cada pieza y la repite varias veces.
 - ❖ Durante la escucha de la pieza *P01 “Desaparecer”* marca el pulso con una mano, probablemente tratando de descifrar alguna rareza en el ritmo.
 - ❖ Descansó aproximadamente tres minutos después de la pieza *P12*, lo que aprovechó para ir al inodoro y despejar la cabeza.
 - ❖ Mencionó que la pieza *P13 “Battle Against a True Hero”* ya la había escuchado en alguna parte, aunque no recordaba de dónde.
 - ❖ Descubrió que había piezas que no eran generadas por algoritmos y que eran muy bonitas.
 - ❖ Describió que aquellas piezas que sentía que sí eran de algoritmo, le gustaron por utilizar frecuentemente alteraciones modales en su escala habitual. El sujeto mencionó que estas pequeñas alteraciones a la escala le daban un aire “curioso e innovador”.
 - ❖ Mencionó que algunas de las piezas se sentían como si fueran de videojuegos.
- **Fotografía de evidencia:** Debido al anonimato deseado por el sujeto, no se adjuntarán fotografías de su proceso de evaluación.
- **Resultados de evaluación:** La **Tabla 7** contiene las respuestas del Sujeto 02.

Tabla 7. Resultados de evaluación del sujeto 02.

Pieza	Sección 1			Sección 2			Sección 3			Sección 4			Sección 5			Extra	
	p1	p2	p3	p1	p2	p3	p1	p2	p3	p1	p2	p3	p1	p2	p3	p1	p2
P01	2	3	3	5	4	4	3	5	3	4	4	3	4	3	4	7	Sí
P02	3	3	4	3	3	3	4	4	4	4	3	3	2	3	4	6	No
P03	5	5	4	5	5	5	4	5	5	5	5	5	4	5	5	9	Sí
P04	2	3	3	4	5	3	3	5	3	3	3	3	3	3	3	5	No
P05	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	10	Sí
P06	1	2	1	1	2	2	1	2	2	2	1	1	2	1	2	3	No
P07	5	5	5	5	5	5	4	4	5	5	5	5	4	5	5	10	Sí
P08	4	4	4	5	5	4	3	5	4	5	5	4	4	3	3	8	Sí
P09	2	3	2	3	3	2	2	3	2	2	2	2	3	2	2	4	No
P10	3	3	4	5	4	5	5	4	4	4	4	3	3	4	4	8	Sí

P11	4	4	4	4	5	5	5	5	5	4	5	4	4	4	5	8	Sí
P12	5	5	4	5	4	5	5	5	5	4	4	4	4	4	4	8	Sí
P13	5	5	5	5	5	5	5	4	5	5	5	5	5	5	5	10	Sí
P14	3	3	3	2	3	3	3	3	2	2	2	3	1	3	4	4	No

10.2.c Sujeto 03

El sujeto y su evaluación poseen las siguientes características:

- **Nombre del sujeto:** Ángela María Salazar Córdova.
- **Edad:** 21 años.
- **Instrumento principal:** Percusiones.
- **Duración de evaluación:** 37 minutos.
- **Fecha de evaluación:** Miércoles 14 de mayo de 2025 a las 2:30pm.
- **Aula utilizada:** A-1.
- **Observaciones:**
 - ❖ No hizo comentarios, ni demostró mucha actividad durante la evaluación.
 - ❖ Mencionó haber reconocido tres piezas, pero sólo de saber que las había escuchado, no de conocer su origen.
 - ❖ Mencionó que algunas de las piezas eran “*bastante bonitas*”.
 - ❖ Su participación fue discreta, sin hacer muchas preguntas ni comentarios y prácticamente sin actividad física notable durante el proceso.
- **Fotografía de evidencia:** La **Figura 49** contiene al Sujeto 03 realizando evaluación.



Figura 49. Sujeto 03 realizando evaluación.

- **Resultados de evaluación:** La **Tabla 8** contiene las respuestas del Sujeto 03.

Tabla 8. Resultados de evaluación del sujeto 03.

Pieza	Sección 1			Sección 2			Sección 3			Sección 4			Sección 5			Extra	
	p1	p2	p3	p1	p2	p3	p1	p2	p3	p1	p2	p3	p1	p2	p3	p1	p2
P01	4	3	4	5	4	5	3	5	2	4	4	2	5	2	3	7	Sí
P02	3	4	5	5	5	5	5	5	5	4	4	4	5	5	3	7	Si
P03	5	5	5	5	5	5	5	4	5	5	5	5	3	5	4	9	Sí
P04	4	5	5	5	4	4	4	4	3	3	5	2	4	2	3	6	No

P05	5	5	5	3	4	4	5	3	5	5	5	5	3	5	5	10	Sí
P06	3	5	2	2	2	1	5	2	2	4	2	4	1	5	2	7	No
P07	5	4	5	5	5	4	5	5	5	5	5	5	5	5	5	10	Sí
P08	5	5	5	5	3	4	5	5	5	4	5	4	3	5	3	8	Sí
P09	3	4	3	2	5	2	5	5	5	2	5	2	2	4	5	7	Sí
P10	1	1	1	1	1	1	2	2	3	2	3	3	2	4	4	5	No
P11	5	5	5	5	5	5	5	4	5	5	5	5	5	5	5	10	Sí
P12	5	5	4	5	5	3	5	5	3	5	4	5	3	5	5	8	Sí
P13	5	5	5	5	4	5	5	5	5	4	5	4	4	5	2	9	Sí
P14	3	2	1	1	3	1	3	2	3	1	1	1	1	1	2	3	No

10.2.d Sujeto 04

El sujeto y su evaluación poseen las siguientes características:

- **Nombre del sujeto:** Ana Alejandra Celaya Marín.
- **Edad:** 21 años.
- **Instrumento principal:** Violín.
- **Duración de evaluación:** 25 minutos.
- **Fecha de evaluación:** Viernes 16 de mayo de 2025 a la 1:00pm.
- **Aula utilizada:** A-1.
- **Observaciones:**
 - ❖ Identificó que sí había piezas que son de compositores y otras que resultaban parecer muy “*hechas por computadora*”, aun si no conocía las piezas reales.
 - ❖ Mencionó que había piezas musicales que sí demostraban algún sentimiento y lo explotaban en su desarrollo, por ejemplo, melancolía, aventura, euforia, etc. El sujeto mencionó que este detalle hizo la diferenciación entre las piezas que se sentían “*muy aleatorias*”.
 - ❖ Mencionó que aquellas piezas que se sentían “*de computadora*” resultaban muy “*impredecibles*”, es decir, que “*dejan muy expectante de qué pasaría enseguida*”, en comparación a las que pensó que son de compositores humanos, que resuelven adecuadamente y de una manera predecible para el oído.
- **Fotografía de evidencia:** La **Figura 50** muestra el Sujeto 04 realizando su proceso de evaluación.

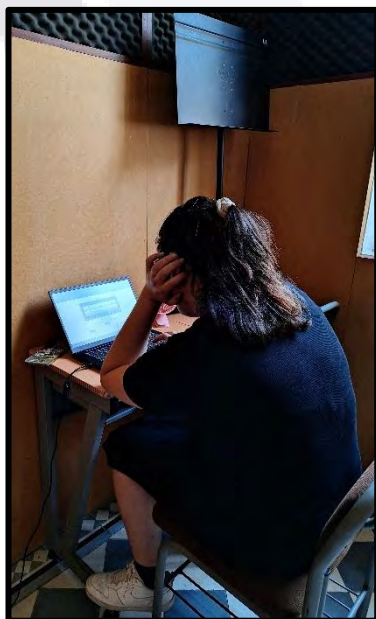


Figura 50. Sujeto 04 realizando evaluación.

- **Resultados de evaluación:** La **Tabla 9** contiene las respuestas del Sujeto 04.

Tabla 9. Resultados de evaluación del sujeto 04.

Pieza	Sección 1			Sección 2			Sección 3			Sección 4			Sección 5			Extra	
	p1	p2	p3	p1	p2	p3	p1	p2	p3	p1	p2	p3	p1	p2	p3	p1	p2
P01	3	3	3	4	5	5	5	4	5	4	5	3	5	3	4	6	Sí
P02	2	2	5	2	2	3	5	5	4	1	3	1	3	4	5	4	No
P03	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	10	Sí
P04	5	5	5	4	4	5	5	5	5	4	5	3	5	3	4	7	No
P05	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	10	Sí
P06	3	1	3	4	4	3	4	2	1	2	2	3	4	3	2	3	No
P07	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	10	Sí
P08	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	10	Sí
P09	2	2	1	3	3	3	3	5	4	3	3	4	4	2	5	5	No
P10	3	3	3	2	2	3	5	1	4	5	4	4	3	5	5	8	Sí
P11	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	10	Sí
P12	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	10	Sí
P13	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	10	Sí
P14	3	3	3	3	3	3	3	4	2	3	2	1	3	2	2	5	Sí

10.2.e Sujeto 05

El sujeto y su evaluación poseen las siguientes características:

- **Nombre del sujeto:** Jorge Leonardo Díaz Morones.
- **Edad:** 26 años.
- **Instrumento principal:** Contrabajo y bajo eléctrico.
- **Duración de evaluación:** 58 minutos.
- **Fecha de evaluación:** Miércoles 21 de mayo de 2025 a las 11:00am.
- **Aula utilizada:** A-1.
- **Observaciones:**
 - ❖ El sujeto no hizo muchos comentarios adicionales.
 - ❖ No reconocía ninguna de las piezas, pero mencionó que muchas le recordaban al estilo de música de videojuegos.
 - ❖ Utilizaba mucho tiempo de reflexión incluso al terminar de responder una pregunta, por lo que en ocasiones cambiaba sus respuestas.
 - ❖ Mucha actividad física con el pie, que se detenía brevemente al avanzar de pregunta.
 - ❖ A este sujeto, al igual que al 01 se le hizo una explicación extra sobre la pregunta final “¿Tiene potencial para convertirse en una canción/obra musical completa?”, pues se agregó a la explicación original que el potencial implicaría si el músico toma y puede hacer cambios mínimos a la pieza, como cambiar un par de notas, acordes, agregar instrumentos, etc. La prueba del Sujeto 05 evidenció una falta grave con los sujetos 02, 03 y 04, pues a estos no se les hizo dicha explicación, tal como al Sujeto 01 sí, por su confusión con la pregunta.
 - ❖ Mencionó que había temas que eran muy buenos.
- **Fotografía de evidencia:** La **Figura 51** muestra al Sujeto 05 realizando el proceso de evaluación en el Aula A-1, misma que los sujetos 02, 03 y 04.



Figura 51. Sujeto 05 realizando evaluación.

- **Resultados de evaluación:** La **Tabla 10** contiene las respuestas del Sujeto 05.

Tabla 10. Resultados de evaluación del sujeto 05.

Pieza	Sección 1			Sección 2			Sección 3			Sección 4			Sección 5			Extra	
	p1	p2	p3	p1	p2	p3	p1	p2	p3	p1	p2	p3	p1	p2	p3	p1	p2
P01	3	4	2	4	3	4	2	5	2	2	2	2	5	2	3	5	Sí
P02	5	4	3	4	3	2	5	5	5	4	5	4	4	5	5	7	Sí
P03	4	4	4	5	3	4	4	5	4	2	4	3	4	3	4	5	Sí
P04	1	2	1	3	2	3	1	4	2	3	2	1	2	2	3	3	Sí
P05	5	4	4	4	4	4	4	4	5	4	5	4	4	4	4	7	Sí
P06	3	2	2	3	2	3	1	4	2	1	2	2	3	2	4	2	Sí
P07	5	5	4	5	4	4	4	3	5	3	4	4	3	4	5	8	Sí
P08	5	5	5	5	5	5	5	5	4	4	4	4	5	4	5	9	Sí
P09	4	3	4	2	2	2	4	4	4	2	4	2	3	2	3	4	Sí
P10	3	2	3	2	2	2	3	4	3	2	2	2	1	1	4	2	Sí
P11	5	5	5	5	5	5	5	5	5	4	5	5	5	5	5	10	Sí
P12	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	9	Sí
P13	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	10	Sí
P14	3	2	2	2	1	2	4	4	3	2	3	1	2	2	4	3	Sí

Las respectivas aulas utilizadas A-1 y A-3 poseían un aislamiento sonoro proporcionado por muros extra de madera sobre las paredes y bloques de espuma acústica. Este aislamiento es necesario, pues las aulas son para practicar instrumentos musicales, y esto requiere aislamiento sonoro del resto de aulas. La principal entrada de sonido son las ventanas en ambos salones; se dejó ligeramente abierta, y no provocó molestias de ruido durante los procesos. La temperatura aproximada de los salones rondaba entre los 24 y 26 grados centígrados, al hacerse en una temporada calurosa del año. Las respectivas aulas se muestran en la **Figura 52**.

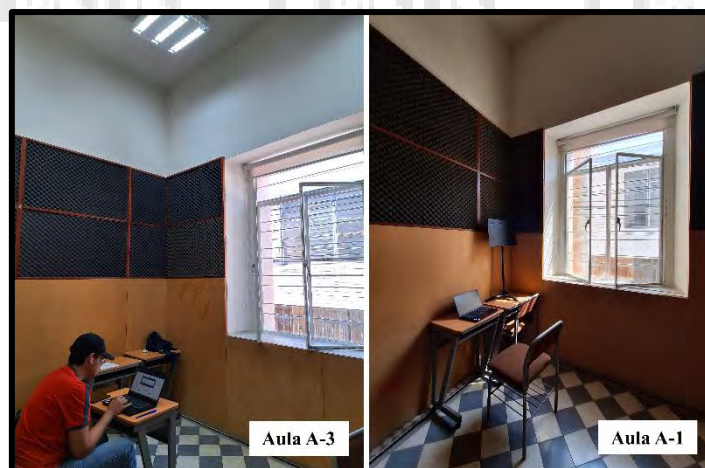


Figura 52. Aulas de evaluación A-3 y A-1.

Terminada cada evaluación se les explicó que ciertas piezas entre las evaluadas no eran generadas por algoritmo, sino por compositores humanos, detalle que no causó sorpresa, pues prácticamente todos los evaluadores notaron la diferencia de calidad entre las piezas, dejando claro el contraste entre las piezas humanas con el resto, sin saber siquiera que existía dicho grupo. Se les solicitó a los evaluadores no divulgar este último hecho entre el resto de los participantes. Se compensó el tiempo tomado de los músicos, con cien pesos mexicanos, como estaba previamente acordado e incluso firmado como consentimiento de compensación.

11 Resultados

11.1 Resultados objetivos globales

Se concentraron todos los resultados de evaluación (Tabla 6, 7, 8, 9, 10) e implementando la fórmula (16) de 9.3.f **Cómputo de puntuación**, se obtuvieron los valores porcentuales que describen la calidad estética de las piezas evaluadas, donde un valor de 0% representa que una pieza careció totalmente de calidad y un 100% representa que la pieza fue perfecta a oído de los músicos evaluadores.

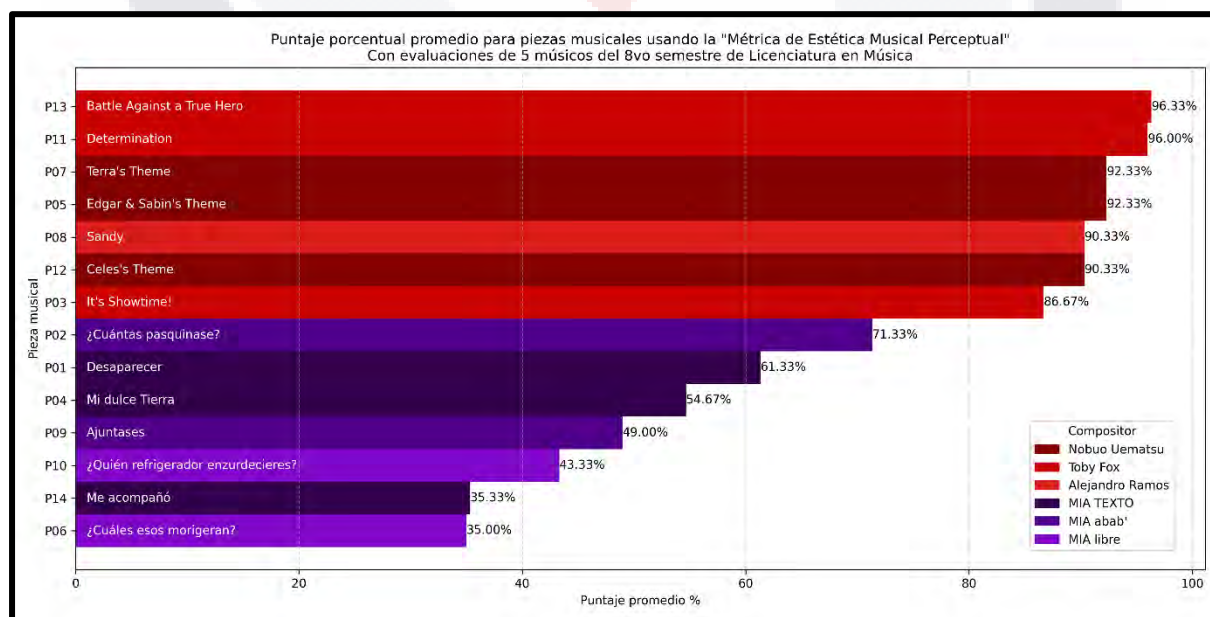


Figura 53. Resultados objetivos globales de evaluación: MIA vs compositores.

La **Figura 53** muestra un gráfico de barras con el recuento de las catorce piezas evaluadas. Las barras de color rojo representan las piezas hechas por compositores humanos, con un color rojo oscuro para las piezas de Nobuo Uematsu, un rojo ligeramente más claro para Toby Fox, y un rojo muy claro para la pieza de Alejandro Ramos. Las barras moradas en cambio representan a las creaciones del algoritmo MIA, usando un morado oscuro para las piezas de estructura *TEXTO*, un morado más claro para las piezas de estructura *abab'* y un morado claro para las composiciones de estructura *libre*. Mientras más larga es una barra, mayor es su calidad promediada, y mientras más arriba está entre las barras, mejor fue con respecto a las demás, pues se ordenaron comenzando con la mejor hasta la peor⁴¹, siendo que "*Battle Against a Real Hero*" fue la pieza mejor evaluada con la métrica objetiva, con un 96.33% de calidad estética, mientras que "*¿Cuáles esos morigeran*" fue la peor evaluada, con un 35% de calidad estética.

El gráfico en la **Figura 53** engloba una visión general del rendimiento de la música de algoritmos con respecto a la música de compositores galardonados. Tal como describieron todos los sujetos evaluadores, es evidente un contraste grande entre las piezas artificiales y las reales, siendo que cada grupo está separado del otro, en medida que no hay piezas artificiales que al ordenarse estén entre las piezas humanas. Esta diferencia se nota aún más cuando se calcula el promedio y la desviación estándar de los puntajes de calidad en todas las piezas de cada grupo, obteniendo lo siguiente:

- **Para piezas reales:**
 - ❖ **Promedio:** 92%.
 - ❖ **Desviación estándar:** 3.39%.
- **Para piezas artificiales:**
 - ❖ **Promedio:** 50%.
 - ❖ **Desviación estándar:** 13.48%.

Existe un 42% de margen entre la calidad de las piezas artificiales y las reales, demostrando un gran espacio de oportunidad para mejorar la calidad en las composiciones de MIA. Sin embargo, la desviación estándar da pista sobre el desempeño del algoritmo. Una desviación estándar de 13.48% refleja una alta variabilidad entre la calidad de las piezas; es decir, que la calidad en las obras generadas puede variar hasta en un 13% según la aleatoriedad en la generación. Esta alta desviación estándar es clara cuando se analiza que la pieza real peor evaluada tiene una diferencia de 15.34% con la pieza mejor evaluada del algoritmo MIA. Las piezas *P02* "*¿Cuántas pasquinase?*" y *P01* "*Desaparecer*" son las mejor evaluadas de origen artificial, con 71.33% y 61.33% de calidad respectivamente; estos resultados son buenos desde la perspectiva de una generación totalmente aleatoria y desde cero, pero demuestran una muy alta variabilidad, tomando como ejemplo a *P06* "*¿Cuáles esos morigeran?*" y *P14* "*Me acompañó*" con 35% y 35.33% respectivamente.

La desviación estándar de 3.29% para las piezas compuestas por humanos refleja dos conclusiones: en primer lugar, la métrica resulta efectiva en medir la calidad musical de manera mayormente objetiva, puesto que se esperaba una baja variabilidad y altos porcentajes en la música humana elegida, dada la calidad de las piezas; en segundo lugar, denota que se eligieron bien las piezas de compositores humanos, dada las calificaciones obtenidas. Dado que las piezas musicales humanas seleccionadas de compositores son renombradas y aclamadas por su calidad, se espera que la evaluación demuestre justamente esto; una evaluación objetiva reflejará que el componente de calidad está presente en las piezas. Mientras menos objetiva sea la métrica, mayor será la variabilidad en sus resultados, pues significaría que esta no logra enfocar los componentes que la hacen buena.

Estos resultados demuestran un panorama de los resultados obtenidos, pues, por un lado, se consiguió crear una métrica mayormente objetiva que permite reflejar la calidad estética de piezas musicales, con base en la percepción de músicos estudiados, y por otro lado, se logró construir un algoritmo que genera composiciones musicales desde cero y sin ningún entrenamiento con piezas de ejemplo.

⁴¹ Para las piezas con empate, se utilizó el promedio porcentual subjetivo, es decir, para la pregunta de gusto entre el 0 y el 10, de manera que los empates entre *P05* y *P07* y entre *P08* y *P12* se definieron por esta medida, permitiendo ordenar las piezas de la mejor a la peor.

11.2 Resultados subjetivos globales

Atendiendo específicamente a la pregunta “¿Del 0 al 10 qué calificación le das a la pieza?”, se promediaron las respuestas de los evaluadores (aunque en una escala entre 0 y 100 en lugar de 0 y 10) para cada pieza, para así obtener una gráfica similar a la anterior mostrada, pero graficando el gusto subjetivo hacia las piezas.

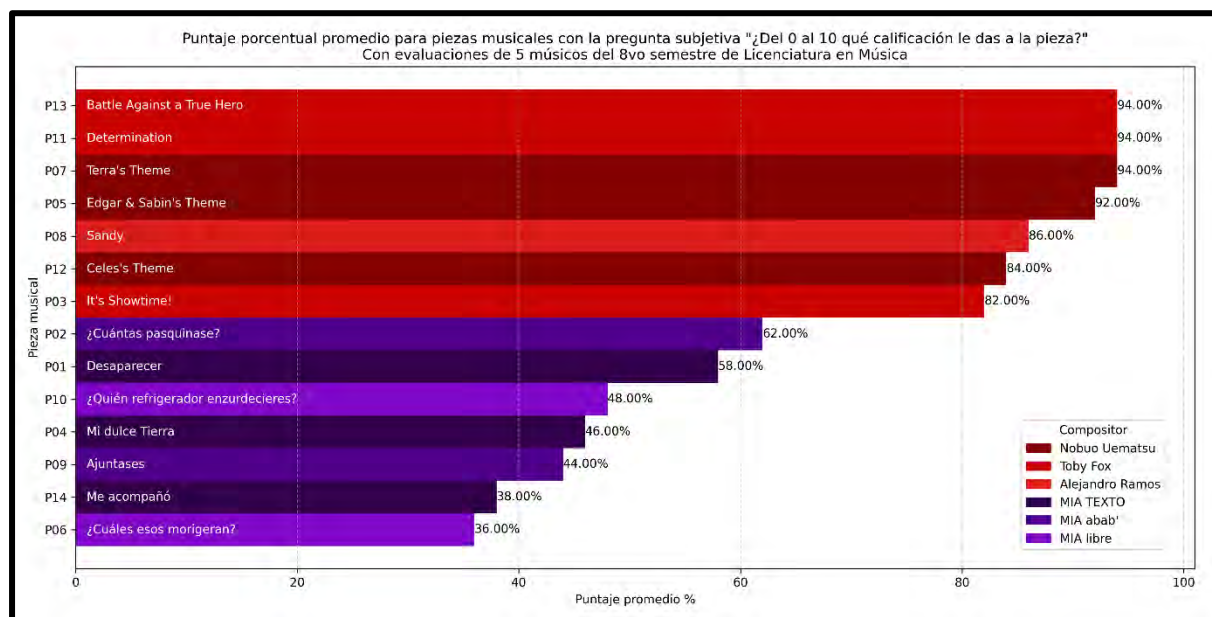


Figura 54. Resultados subjetivos globales de evaluación: MIA vs compositores.

La **Figura 54** muestra los resultados de la evaluación subjetiva, con el mismo formato de colores que la **Figura 53**. Se observa el mismo fenómeno de separación entre calidad de piezas del algoritmo y piezas humanas. Para las piezas musicales automáticas se obtuvo un 47.4% de calidad, con 9.6% de desviación estándar y para piezas humanas 89.4% de calidad y 5.2% de desviación estándar. Tal como se esperaría, la desviación es más alta para la evaluación subjetiva en piezas humanas, mostrando mayor variabilidad en sólo describir entre el 0 y 10 la calidad de una pieza. La desviación estándar en piezas automáticas se redujo casi 4%, lo que demostraría que en cuestión de *gusto*, hay más acuerdo entre los evaluadores.

Las evaluaciones subjetivas redujeron los porcentajes de calidad para ambos conjuntos de piezas, demostrando que a los músicos las piezas les gustaron menos de lo que reflejaba la métrica, aunque sólo es alrededor de un 2% la diferencia. El orden en las piezas se mantuvo para todo el conjunto de piezas humanas y para las primeras dos y últimas dos del conjunto de composiciones del algoritmo, en cambio hubo cambios en el orden entre las piezas P04 “Mi Dulce tierra”, P09 “Ajuntases” y P10 “¿Quién refrigerador enzurdecieres?”.

Es posible que exista un sesgo en la evaluación subjetiva, producto de responder a la pregunta después de haber respondido el resto del cuestionario. Este sesgo podría intensificar la similitud entre las métricas objetivas y subjetivas, pues el evaluar el gusto de una pieza posterior a haberla analizado estructuralmente puede propiciar al evaluador a reflejar subjetivamente lo mismo que objetivamente ya analizó.

11.3 Resultados por sección evaluada

Se calcularon los puntajes promedio para todas las piezas generadas por el algoritmo, con especial énfasis en los puntajes de cada sección evaluada por la métrica, con el objetivo de identificar fortalezas y falencias en las partes específicas que componen la pieza. La **Figura 55** muestra las estadísticas porcentuales por sección, para todas las piezas generadas por el algoritmo⁴².

⁴² Las estadísticas y conclusiones deben tomarse con cautela, especialmente para las secciones evaluadas para estructuras de algoritmo específicas; esto debido a la pequeña cantidad de muestras en cada caso. Un ejercicio con un número mayor de piezas podría revelar estadísticas más contundentes.

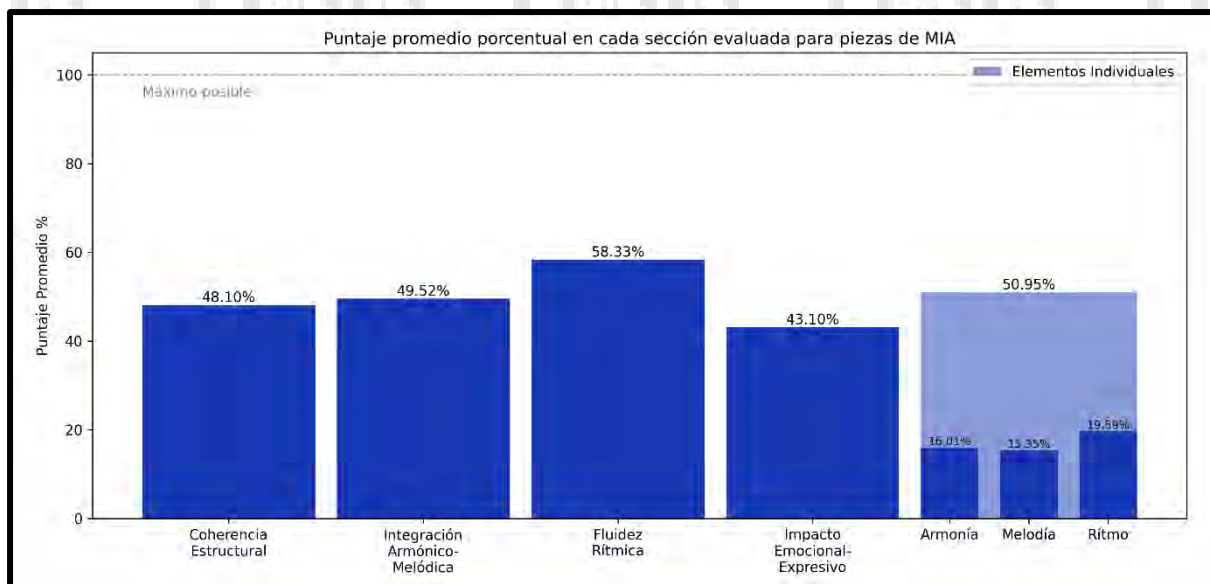


Figura 55. Puntaje porcentual promedio en cada sección evaluada para piezas de MIA.

De la **Figura 55** se pueden extraer las siguientes conclusiones:

- El componente mejor evaluado en las piezas de MIA es el ritmo, con un 58.33% de calidad en este rubro individual. El ritmo está a cargo del Recocido Simulado, por lo que este componente tuvo el mejor desempeño la arquitectura del algoritmo MIA Harmfirst a1.
- El componente peor evaluado es el “*Impacto Emocional-Expresivo*”, naturalmente, pues la arquitectura no incorpora elementos que permitan guiar la generación automática para explorar sentimientos basados en los sonidos.
- Entre los componentes individuales, el que peor se desempeñó fue la *melodía*, a cargo del Módulo LWRS. Sin embargo, la armonía estuvo cerca en su desempeño, estando a cargo del módulo de Algoritmo Genético.
- Con respecto al módulo de Gramática Generativa no es factible aun extraer características de su desempeño, sin embargo, será posible al diferenciar las estadísticas entre una estructura que use estas gramáticas (por ejemplo, la *Estructura libre*) con una estructura que no la utilice (*Estructura TEXTO*).
- No hay una diferencia notable entre los desempeños de todas las secciones, lo que implica que no hay diferencias marcadas entre la capacidad de los módulos del algoritmo, sin embargo, se nota que aún hay un gran espacio de mejora para alcanzar el máximo en los 100%.
- Los siguientes criterios mejor valorados son *Elementos individuales* e *Integración Armónico-Melódica*, donde el primero de estos evalúa cómo se desempeñan la *Armonía*, *Melodía* y *Ritmo* individualmente y sin depender del resto de elementos, y el segundo evalúa qué tal se complementan. Esto sugiere que los componentes básicos de las piezas musicales generadas por el algoritmo MIA funcionan tanto por separado como incorporados, aunque en ambos casos sus desempeños están a mitad de camino de una calidad perfecta.

En resumen, el análisis individual por secciones permite entender qué módulos proporcionan mejor rendimiento, de manera general. Se concluye que el Recocido Simulado cumple un papel bueno en adaptar sílabas en oraciones, a frases rítmicas interesantes, aun si el módulo LWRS le hace perder calidad a la potencial melodía. El Algoritmo Genético por su parte, puede mejorarse aún más, pues la calidad de este propicia igualmente la calidad en la melodía final, así como en la armonía.

11.3.a Estructura TEXTO

La **Figura 56** contiene las estadísticas por sección para piezas específicamente de la *Estructura TEXTO*, es decir, para piezas que en lugar de utilizar el módulo de Gramática Generativa, utilizan como entrada versos escritos, que, como se mencionó anteriormente, son autoría de Ruth (Serafio Méndez, 2023, 2025b, 2025a).

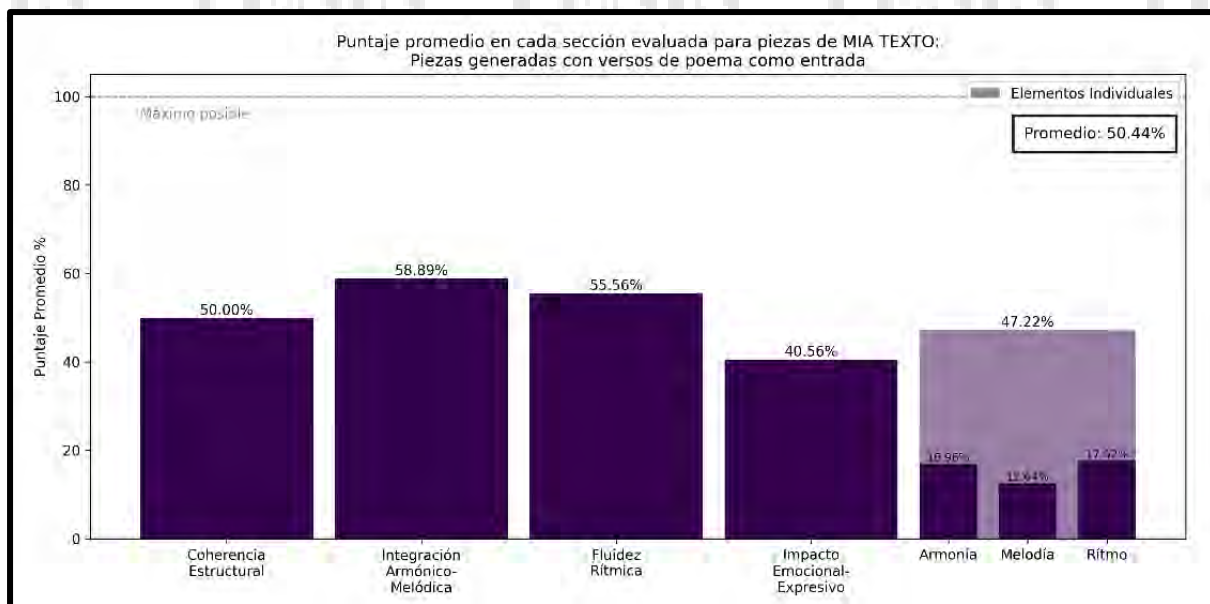


Figura 56. Puntaje promedio en cada sección evaluada para piezas de MIA TEXTO.

Con base en la **Figura 56** se pueden extraer las siguientes conclusiones:

- La calidad en la *Fluidez Rítmica* para piezas de *MIA TEXTO* es de 55.56%, es decir 2.77% menos que el promedio para piezas del algoritmo. Esto implica que, contrario a lo que se esperaba, el uso de versos humanos escritos no necesariamente mejoró el resultado final. Sin embargo, se necesita una comparación con la *Estructura libre* para saber si los resultados empeoraron.
- El promedio fue equiparable al promedio global de piezas del algoritmo; es decir, 50%.
- La *Coherencia Estructural* en la pieza es mejor que el promedio.

11.3.b Estructura abab'

La **Figura 57** contiene las estadísticas de calidad para las piezas del algoritmo con la estructura *abab'*, es decir, las piezas con una repetición de las dos primeras frases y sus acordes específicos, con una variación de la segunda frase como cuarta frase y con una variación (o no) de sus acordes.

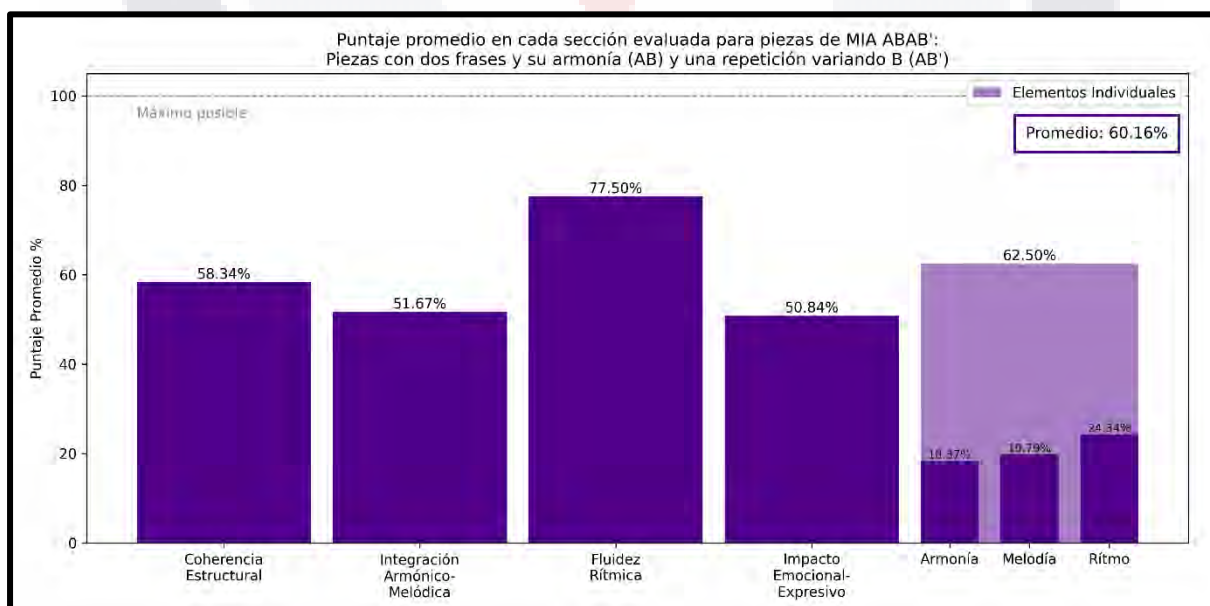


Figura 57. Puntaje promedio en cada sección evaluada para piezas de MIA abab'.

Con base en la **Figura 57** se obtienen las siguientes conclusiones:

- La *Estructura abab'* propició una mejora bastante considerable, pues el desempeño de las piezas con esta estructura es de 60.16%, comparado al 50% en las piezas del algoritmo. Esto debe tomarse con cautela, debido la existencia de sólo dos ejemplos para esta estructura, aunado a la alta variabilidad en la calidad. Esta mejora podría estar potenciada por aleatoriedad de generar alguna pieza radicalmente buena, como la *P02* “¿Cuántas pasquinase?”.
- La estructura mejoró significativamente la calidad del ritmo, con un 77.5% de calidad en comparación al 58.33% promedio, con un 19.17% de diferencia. **Esto confirma que la repetición (parte esencial de la estructura abab') es una decisión que favorece mucho a la calidad de una pieza, justo como el indica el principio básico del motivo musical (consultar 6.1.b.2 Motivo)⁴³.**
- La *Coherencia Estructural* también mejoró considerablemente, con un 58.34% de calidad, en comparación al 48.10% del promedio, siendo 10.24% el incremento. Nuevamente, el factor de la repetición se muestra indispensable en la mejora de calidad en una pieza.
- Se generaron 7.74% extra de calidad para el rubro *Impacto Emocional-Expresivo*, sugiriendo que la repetición también propicia a la sensación de expresividad.
- La *Integración Armónico-Melódica* no es significativamente superior con respecto al promedio.
- Los componentes individuales *Armonía*, *Melodía* y *Ritmo* son superiores en un 11.55% respecto al promedio, sugiriendo que la repetición de componentes integrales en la melodía y armonía propician a una mejor valoración para estos.

Los resultados de esta sección sugieren que un espacio de mejora está en, teniendo ya los módulos generadores, **de conseguir una manera en la que se generen componentes y/o partes más pequeñas, que se integren manualmente, propiciando a la repetición y reutilización de elementos**, se podría conseguir un sistema que dé más control al usuario músico para conseguir mejores resultados de calidad estética.

11.3.c Estructura libre

La **Figura 58** contiene los resultados específicos para la sección *Estructura libre*, es decir, para aquellas piezas que se generaron creando componentes únicos para cada caso, de forma libre y sin estructura clara.

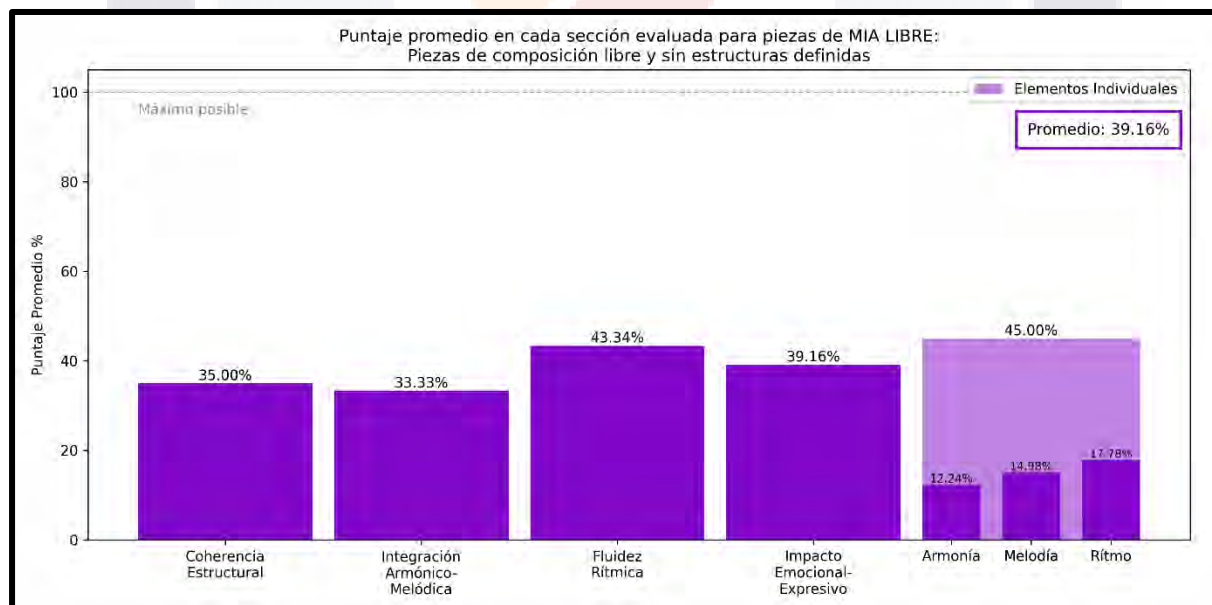


Figura 58. Puntaje promedio en cada sección evaluada para piezas de MIA libre.

⁴³ Considérese además que el *Recocido Simulado* utiliza para cada generación de solución, múltiples variantes (veinte según el parámetro por defecto descrito en **9.1.a Entradas y parámetros de preferencias**), de modo que para obtener una solución, se elige al mejor *Recocido* entre veinte.

De la **Figura 58** y con base en el resto de los resultados analizados, se pueden extraer las conclusiones:

- La composición libre provocó el principal descenso en la calidad promedio de las piezas generadas por algoritmo, al estar 10.84% puntos por debajo. Esto refuerza la idea de que la repetición de componentes generados sí propicia la calidad en una pieza, dado que esta es la única estructura que no repite ni reutiliza nada en su proceso de generación.
- El uso de versos para la estructura *TEXTO* sí puede propiciar mayor calidad, pues si una serie de oraciones tiene una distribución repetitiva de sílabas tónicas y átonas, llevaría al Recocido Simulado a tomar decisiones muy similares de asignación de ritmo, de la misma manera que esto llevaría a tomar decisiones muy similares en la decisión de tonos para las notas con base en su duración, llevado a cabo por el Módulo LWRS. Esto propiciaría la aparición de patrones repetitivos, dando mayor sensación de presencia de *motivos*.
- Siendo esta la versión básica de la arquitectura del algoritmo MIA, se puede ver que grandes rasgos, la armonía es la que tiene peor desempeño (Algoritmo Genético), sin embargo, es necesario tomarse con cautela dicha afirmación. Un melodía es buena no sólo por los tonos elegidos por el Módulo LWRS, sino también por el ritmo en sus notas; de forma que el desempeño del ritmo propicia el desempeño de la melodía, del mismo modo que esta depende de la armonía. La armonía carga con el peso completo del desempeño de la obra, pues si esta no es lo suficientemente buena, entonces la melodía tampoco lo será.
- A pesar de depender del Recocido Simulado como del Módulo LWRS, la melodía mostró superioridad sobre la armonía, de modo que el Módulo LWRS también es responsable de buena parte de la calidad en una pieza generada.
- Se confirma que la falta de repetición y reutilización de componentes genera mayor variabilidad en los resultados, así como menor calidad en los mismos. El promedio de 39.16% de calidad refleja que estas piezas son las principales responsables de la alta desviación estándar en la música de MIA.
- La sección mejor evaluada es la de *Componentes individuales*, lo que sugiere que **la generación individual de componentes de una pieza musical, con la arquitectura de MIA Harmfirst funciona de buena manera, por lo que, de utilizarse para estructurar mejor las piezas, se podría mejorar la calidad en los resultados.**

Dados los resultados, resulta entonces lógica la afirmación de Arnold Schoenberg en “*Fundamentos de la Composición Musical*”; que traducida al español sería la siguiente:

“La inteligibilidad en la música parece imposible sin la repetición. Mientras que la repetición sin variación puede fácilmente producir monotonía, y la yuxtaposición de elementos distantes puede fácilmente degenerar en falta de sentido” (Schoenberg, 1967).

11.4 Resultados de utilidad

Retomando la pregunta “¿*Tiene potencial para convertirse en una canción/obra musical completa?*” que es parte de la métrica, se debe hacer una aclaración sobre los resultados. Tal como se mencionó en la sección **10 Experimentos – 10.2 Ejecución**, se cometió un fallo metodológico, cuando no se explicó completamente el objetivo de la pregunta mencionada. Se olvidó agregar el detalle de que, el “*potencial para convertirse en una obra musical completa*” refiere a si el músico toma la pieza y hace modificaciones menores (pues al fin y al cabo las ideas generadas sirven de inspiración). Este detalle sólo se explicó a los sujetos 01 y 05.

Tomando las respuestas de los sujetos 01 y 05, y considerando cada “*Sí*” y cada “*No*” de respuesta a la pregunta, como un voto, se recuenta que las piezas de compositores humanos recibieron 14 votos “*Sí*” (es decir, para siete piezas y dos sujetos, el 100% de los votos es 14), de modo que se confirma que el 100% de las piezas humanas son perfectamente adaptables para ser una obra musical completa (esperable, pues en efecto, lo son). Para el caso de las piezas automáticas, son 9 votos a “*Sí*” y 5 votos a “*No*”, de manera que **se concluye que el 64% de las piezas generadas por MIA Harmfirst a1 puede ser perfectamente la inspiración para una obra musical completa**, siendo este el objetivo principal de la creación del algoritmo.

Si se calcula la estadística de aceptación con los sujetos a quienes no se les explicó la pregunta, se obtiene 100% votos “Sí” para piezas humanas y 33% votos “Sí” para piezas del algoritmo. Demostrando una diferencia entre explicar o no la pregunta. De otra manera, haciendo la estadística con todos los sujetos, se obtiene un 100% para las piezas humanas y un 45.7% para las piezas artificiales.

11.5 Recuento de estadísticas

La **Tabla 11** enlista todas las estadísticas antes descritas para cada pieza musical en la lista de experimentos según la “*Métrica de Estética Musical Perceptual*”, recopilando el puntaje porcentual de cada pieza, tanto de manera global, como de manera enfocada en secciones. El color de las celdas numéricas marca una escala, donde mientras más verde es, más calidad refleja, y mientras más blanco, menor calidad refleja. Se diferencian además las piezas compuestas por humanos y por el algoritmo artificial.

Tabla 11. Puntajes promedio para cada pieza musical y sus secciones.

Pieza	Compositor	% Global	Coherencia Estructural	Integración Armónico-Melódica	Fluidez Rítmica	Impacto Emocional-Expresivo	Elementos Individuales			
							Total	Armonía	Melodía	Ritmo
P01	Artificial	61.33	55.00	80.00	61.67	50.00	60.00	24.71	15.29	20.00
P02	Artificial	71.33	70.00	61.67	90.00	61.67	73.33	19.89	26.10	27.34
P03	Humano	86.67	86.67	88.33	90.00	86.67	81.67	24.24	28.07	29.35
P04	Artificial	54.67	53.33	61.67	58.33	50.00	50.00	17.78	13.33	18.89
P05	Humano	92.33	96.67	88.33	93.33	95.00	88.33	27.28	29.88	31.18
P06	Artificial	35.00	36.67	33.33	36.67	30.00	38.33	12.11	13.11	13.11
P07	Humano	92.33	96.67	93.33	91.67	90.00	90.00	27.39	30.00	32.61
P08	Humano	90.33	95.00	93.33	91.67	90.00	81.67	26.80	28.07	26.80
P09	Artificial	49.00	46.67	41.67	65.00	40.00	51.67	16.85	13.48	21.34
P10	Artificial	43.33	33.33	33.33	50.00	48.33	51.67	12.36	16.85	22.46
P11	Humano	96.00	95.00	96.67	98.33	95.00	95.00	30.35	31.67	32.99
P12	Humano	90.33	93.33	90.00	91.67	88.33	88.33	25.98	29.88	32.48
P13	Humano	96.33	100.00	98.33	98.33	95.00	90.00	30.00	32.61	27.39
P14	Artificial	35.33	41.67	35.00	46.67	21.67	31.67	8.38	9.31	13.97

12 Conclusiones

12.1 Objetivos

Retomando los objetivos planteados, se tiene que, como objetivo general se planteó:

“Diseñar y desarrollar un algoritmo autónomo de composición musical que, sin entrenamiento previo con música real, genere piezas equiparables a ejemplos de la música tonal contemporánea y que incorpore una métrica validada mediante evaluación de músicos expertos para medir su calidad estética”.

Se concluye que el objetivo general fue completado. Se logró diseñar y construir una arquitectura de algoritmo generador de piezas musicales comparables con piezas musicales reales, todo sin depender del entrenamiento de modelos generativos mediante música de ejemplo. Además, se planteó una métrica que permitió medir la calidad de piezas generadas por la arquitectura de algoritmo, con músicos expertos como propiciadores.

Como objetivos particulares, se definió:

1. **Explorar e incorporar reglas de composición musical, técnicas de Procesamiento de Lenguaje Natural, Cómputo Evolutivo y Metaheurísticas, a la composición autónoma de piezas musicales originales:** Se lograron incorporar las técnicas, pues la arquitectura del algoritmo desarrollado utiliza

Gramáticas Generativas (Procesamiento de Lenguaje Natural), *Algoritmo Genético* (Cómputo Evolutivo y Metaheurística) y *Recocido Simulado* (Metaheurística), de modo que cada componente funge como pieza fundamental en la composición de piezas musicales artificiales.

2. **Crear una métrica de puntos basada en la percepción de escucha, para evaluar las cualidades estéticas en los componentes estructurales de composiciones musicales:** Se desarrolló la “*Métrica de Estética Musical Perceptual*”, que a través de preguntas cuyas respuestas son valores numéricos entre 1 y 5, permite hacer conteo de puntos para determinar la calidad en distintas cualidades estéticas en composiciones musicales.
3. **Recopilar una selección de obras de la música tonal contemporánea y una selección de piezas generadas por el algoritmo diseñado, para evaluar su calidad estética con un grupo de músicos estudiados en el análisis musical:** Durante la fase de experimentación se crearon conjuntos con piezas musicales de tres compositores humanos contemporáneos y se sometieron a evaluación junto a piezas generadas por el algoritmo desarrollado, de modo que con la percepción de cinco músicos del octavo semestre de la *Licenciatura en Música* se pudo analizar la calidad del algoritmo.
4. **Analizar los resultados de evaluación de las piezas para entender fortalezas y deficiencias en la arquitectura del algoritmo diseñado:** Se desarrolló un análisis de resultados que enlistó fortalezas y deficiencias en el algoritmo, de modo que se facilita abordar el problema para mejorar futuros resultados sobre la misma línea de investigación.

12.2 Preguntas de investigación

Se enlistaron las siguientes preguntas a responder:

- **¿Qué arquitectura de algoritmo computacional permitiría la composición musical autónoma, sin que dependa de entrenamiento con música real?:** Se diseñó una arquitectura llamada *MIA Harmfirst a1*, que logró crear piezas musicales sin depender de entrenamiento. Esta arquitectura responde directamente a la pregunta.
- **¿De qué manera es posible adaptar técnicas de Procesamiento de Lenguaje Natural y Metaheurísticas, a la composición autónoma de melodías musicales?:** Mediante módulos que se centren en cuestiones específicas de la composición; en este caso las *Gramáticas Generativas* para construir una distribución finita y aleatoria de sílabas con sus distintas características (para crear una melodía) y la metaheurística de *Recocido Simulado* para darle ritmo a las sílabas generadas, con una función objetivo que, gracias a una parametrización del usuario, favorece en mayor o menor medida a las soluciones, con base en reglas de la *prosodia*.
- **¿De qué manera el Cómputo Evolutivo puede ser parte del proceso de composición de la armonía de una pieza musical?:** Con el *Algoritmo Genético* enfocado en optimizar cadenas de acordes con respecto a definiciones de preferencia basadas en reglas de composición armónica.
- **¿Qué elementos cualitativos podrían evaluar la calidad estética de una composición musical?:** La coherencia estructural en una obra, la correcta integración de su melodía y armonía, la fluidez en su ritmo, el impacto emocional de la pieza y el desempeño individual e independiente de sus componentes: armonía, melodía y ritmo.

12.3 Contribuciones

Consúltese 7.5 Oportunidades de contribución.

1. Creación de un algoritmo generador de música artificial que se construyó sin el uso de piezas musicales reales para su entrenamiento, llamado “*MIA Harmfirst a1*”.
2. Creación de un algoritmo generador de música artificial que no se limita a ningún género musical, estilo o imitación de compositor específico.
3. Creación del mecanismo que permite componer piezas musicales cantables a partir de una letra escrita por el usuario (tanto músico como no-músico).

4. Incorporación de mecanismos para evitar la *aleatoriedad* y favorecer la *coherencia* en la colocación de notas musicales de una melodía, mediante el uso de *Gramáticas Generativas* que crean conjuntos finitos de sílabas (notas) con distribuciones únicas y coherentes de sílabas tónicas y átonas.
5. Exploración de las técnicas de *Algoritmo Genético* (para creación de armonía) y *Recocido Simulado* (para asignación de ritmo a sílabas en una oración) aplicadas en la composición musical. Con las mediciones hechas en la configuración específica de estos componentes, es posible no sólo comprender el alcance y potencial de las mismas, sino conocer qué tan bien se desempeñaron.
6. Creación de una métrica objetiva llamada "*Métrica de Estética Musical Perceptual*", que permite medir numéricamente la calidad estética de una "*Idea musical*" constituida por una melodía y una armonía, que se mide con base en la percepción de un músico educado académicamente. Esta métrica puede ser un mecanismo experimental útil para ofrecer valores medibles y comparables sobre la calidad de piezas generadas por computadora.
7. Implementación de "*co-creatividad*" mediante la definición de parámetros con los cuales el usuario generador puede direccionar el resultado de la *Idea musical*. Además, las entradas basadas en texto y las definiciones de repetición de frases, facilitan la creación de piezas musicales mejor estructuradas según el criterio creativo del usuario.
8. Creación de un algoritmo original para la ponderación en los elementos de una lista de selección, mediante aleatoriedad y con un componente de *complejidad* que sesga en mayor o menor medida la preferencia de los elementos en el tope de la lista. Este algoritmo es llamado *CWRS (Complexity Weighted Random Selector)*.
9. Creación de *Funciones objetivo* para medir: 1. La calidad en una progresión de acordes y 2. Medir la calidad del ritmo en la dicción una oración.
10. Mecanismo de inicialización en el *Algoritmo Genético*, que al crear una progresión da preferencia a acordes de la tonalidad usando *LWRS*, y que selecciona acordes según su complejidad usando *CWRS*.
11. Aplicación de un algoritmo totalmente original, de autoría propia y no implementado antes: *LWRS*, que en conjunto con una lógica de jerarquización de tonos de la teoría tonal (igualmente original), permite componer melodías únicas, usando como base una progresión de acordes.

12.4 Trabajo futuro

Sobre esta misma línea de investigación, las siguientes áreas de exploración resultan interesantes:

1. **Mecanismos de repetición:** Según los resultados del presente trabajo, una oportunidad para mejorar la calidad en las composiciones radica en implementar mecanismos que, ya sea en el desarrollo de melodías, armonías o ritmos, o bien en la estructuración de frases para la constitución de *Ideas musicales*, mejoren la calidad de resultados ofreciendo repetición y variación se componentes dentro de una misma pieza musical.
2. **Mecanismos de reparación:** Las piezas musicales elegidas de compositores humanos no son el primer resultado creado por los mismos en cada caso de composición, pues estas estuvieron sujetas a un proceso iterativo de mejora, que finalmente da como resultado una pieza musical sólida en su estética. El algoritmo en cambio se evalúa con la idea inicial que genera. Bajo esta comparación, se concibe factible que un resultado generado por *MIA Harmfirst a1* pueda ser el paso inicial en un proceso iterativo que "*repare*" la obra para mejorarla, tanto si esto es automático como si depende de interacciones expresas del usuario para direccionar las preferencias.
3. **Mecanismos de agrupamiento:** Dado que se concluyó que el algoritmo puede generar componentes individuales valiosos en estética, pero que en conjunto pueden estar mal estructurados por la composición *libre* de *MIA*, se concibe que una implementación más compleja (preferiblemente de interfaz visual) podría permitir al usuario generar componentes individuales para mezclarlos, rehacerlos, variarlos y estructurarlos para conseguir resultados más "*co-creativos*". Esta función debería permitir incluso colocar manualmente constantes en los resultados, como oraciones en específico, acordes y progresiones de elección del usuario o notas específicas de la melodía, decididas por el usuario.

4. **Modelo “*Melfirst a1*”:** Tal como “*Harmfirst*” compone inicialmente una armonía y con base en esta musicaliza la melodía, podría existir un modelo que genere inicialmente una melodía y construya su armonía con base en esta; es decir: “*Melodyfirst*”, “*Melodía primero*”.

12.4 Reflexiones finales

A lo largo de este trabajo se exploraron técnicas computacionales que no necesitaron de datos de entrenamiento en el diseño y construcción de una arquitectura de algoritmo de composición musical autónoma. Aún se tiene mucho trabajo por realizar; las evaluaciones experimentales aquí mostradas no son suficientes para hacer conclusiones factuales sobre el desempeño de las técnicas implementadas, pero sí para ofrecer visibilidad sobre su efectividad y potencial. Existen infinitas ideas para mejorar lo logrado aquí. Queda a consideración de quien lee esto, contribuir con cualquier observación que ofrezca una nueva perspectiva sobre el proyecto MIA, el cuál es libre y de código abierto. Cualquier participación será bien recibida.

Iván Alejandro Ramos Herrera

arhcoder@gmail.com.

Bibliografía

- Ackerman, M., & Loker, D. (2016). *Algorithmic Songwriting with ALYSIA*.
- Ames, C. (1987). Automated Composition in Retrospect: 1956-1986. *Leonardo*, 20(2), 169–185.
<https://doi.org/10.2307/1578334>
- Ames, C. (1990). Statistics and Compositional Balance. *Perspectives of New Music*, 28(1), 80–111.
<https://doi.org/10.2307/833345>
- Atlantic Award Group. (2017, febrero 2). *Award to Nobuo Uematsu*. <https://atlanticaward.com/news/1757>.
- BAFTA. (2007). *Original Score - Bafta*. <https://www.bafta.org/awards/games/original-score-games>.
- Bartneck, C., Lütge, C., Wagner, A., & Welsh, S. (2021). What Is AI? En C. Bartneck, C. Lütge, A. Wagner, & S. Welsh (Eds.), *An Introduction to Ethics in Robotics and AI* (pp. 5–16). Springer International Publishing. https://doi.org/10.1007/978-3-030-51110-4_2
- Benjamin, T., Horvit, M., Nelson, R., & Koozin, T. (2014). *Techniques and Materials of Music: From the Common Practice Period Through the Twentieth Century, Enhanced Edition*. Cengage Learning.
- Bentley, J. (2025, mayo 2). *The 25 best video game soundtracks of all time*. <https://www.gamesradar.com/best-gaming-soundtracks/>.
- Bickerman, G., Bosley, S., Swire, P., & Keller, R. (2010). Learning to Create Jazz Melodies Using Deep Belief Nets. *Proceedings of the International Conference on Computational Creativity, ICC-C-10*.
- Bosque, I. (2015). *Las categorías gramaticales: relaciones y diferencias*. Síntesis.
- Bosque, I., & Demonte, V. (1999). *Gramática descriptiva de la lengua española: Entre la oración y el discurso. Morfología*. Espasa.
- Bosque, I., & Gutiérrez-Rexach, J. (2020). *Fundamentos de sintaxis formal*. Akal.
- Bretan, M., Weinberg, G., & Heck, L. (2016). *A Unit Selection Methodology for Music Generation Using Deep Neural Networks*.
- Bribitzer-Stull, M. (2015). Motive, phrase, melody, and theme. En M. Bribitzer-Stull (Ed.), *Understanding the Leitmotif: From Wagner to Hollywood Film Music* (pp. 33–55). Cambridge University Press.
<https://doi.org/DOI: 10.1017/CBO9781316161678.003>
- Bryson, A. E. (1961). A gradient method for optimizing multi-stage allocation processes. *Proc. Harvard Univ. Symposium on digital computers and their applications*, 72, 22.
- C chords*. (s/f). <https://www.pianochord.org/c.html>.
- Carvajal Ramírez, J. C., & Giraldo Giraldo, F. A. (2012). Composición musical usando algoritmos genéticos. *Tecnura*, 16(33), 145–157.

- Centro Virtual Cervantes. (s/f). CVC. *Diccionario de términos clave de ELE. Prosodia*.
https://cvc.cervantes.es/ensenanza/biblioteca_ele/diccio_ele/diccionario/prosodia.htm.
- Choi, K., Fazekas, G., & Sandler, M. (2016). *Text-based LSTM networks for Automatic Music Composition*.
- Chomsky, N. (1956). Three models for the description of language. *IEEE Transactions on Information Theory*, 2(3), 113–124. <https://doi.org/10.1109/TIT.1956.1056813>
- Chomsky, N. (1957). *Syntactic Structures*. Mouton.
- Chu, H., Urtasun, R., & Fidler, S. (2016). *Song From PI: A Musically Plausible Network for Pop Music Generation*.
- Cox, G. (2010). *On the Relationship Between Entropy and Meaning in Music: An Exploration with Recurrent Neural Networks*.
- Dagdia, Z. C., & Mirchev, M. (2020). When Evolutionary Computing Meets Astro- and Geoinformatics. *Knowledge Discovery in Big Data from Astronomy and Earth Observation: Astrogeoinformatics*, 283–306. <https://doi.org/10.1016/B978-0-12-819154-5.00026-6>
- Danhauser, A. (2008). *TEORIA DE LA MUSICA: Vol. I* (1a ed.). Melos.
- De la Herrán, J. (2007). *Física y música: Vol. I* (1a ed.). ADN Editores, Consejo Nacional para la Cultura y las Artes.
- De la Motte, D. (2007). *ARMONÍA: Vol. I* (1a ed.). MUNDIMÚSICA EDICIONES S.L.
- Dines, L. L. (1926). On Positive Solutions of a System of Linear Equations. *Annals of Mathematics*, 28(1/4), 386–392. <https://doi.org/10.2307/1968384>
- Dionisio de Pedro. (2008a). *Teoría Completa de la Música 1* (3ra ed., Vol. 1). Real Musical.
- Dionisio de Pedro. (2008b). *Teoría Completa de la Música 2* (3ra ed., Vol. 2). Real Musical.
- Dostert, L. E. (1955). The georgetown-ibm experiment. *Machine translation of languages*, 8, 129ff.
- Dreyfus, S. (1962). The numerical solution of variational problems. *Journal of Mathematical Analysis and Applications*, 5(1), 30–45. [https://doi.org/10.1016/0022-247X\(62\)90004-5](https://doi.org/10.1016/0022-247X(62)90004-5)
- Dreyfus, S. (1973). The computational solution of optimal control problems with time lag. *IEEE Transactions on Automatic Control*, 18(4), 383–385. <https://doi.org/10.1109/TAC.1973.1100330>
- Eigenfeldt, A., & Pasquier, P. (2012). Creative Agents, Curatorial Agents, and Human-Agent Interaction in Coming Together. *Proceedings of the 9th Sound and Music Computing conference (SMC 2012)*, 181–186.
- Fanani, A. Z., Syarif, A. M., Shidik, G. F., & Marjuni, A. (2024). Expressing and Developing Melodic Phrases in Gamelan Skeletal Melody Generation Using Genetic Algorithm. *IEEE Access*, 12, 130512–130523. <https://doi.org/10.1109/ACCESS.2024.3457880>

- Fenollosa, C. (2019, agosto 23). *Diccionario libre en español - Free Spanish dictionary*.
https://cfenollosa.com/blog/tag_nlp.html.
- Fernández, J. P. (2021, diciembre 16). *Síncopa en música: definición, aplicaciones y ejemplos*.
<https://cresciente.net/ritmo/sincopa-en-musica-definicion-ejemplos>.
- Fox, T. (2015, septiembre 15). *Undertale OST (CD Quality)*. <https://archive.org/details/Undertale-OST-CD>.
- Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4), 193–202.
<https://doi.org/10.1007/BF00344251>
- Galanter, P. (2012). Computational Aesthetic Evaluation: Past and Future. En J. McCormack & M. d’Inverno (Eds.), *Computers and Creativity* (pp. 255–293). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-31727-9_10
- Gao, W. (2018). A comprehensive review on identification of the geomaterial constitutive model using the computational intelligence method. *Advanced Engineering Informatics*, 38, 420–440.
<https://doi.org/10.1016/j.aei.2018.08.021>
- Gioti, A.-M. (s/f). A Compositional Exploration of Computational Aesthetic Evaluation and AI Bias. *NIME 2021*. <https://doi.org/10.21428/92fbeb44.de74b046>
- Gioti, A.-M. (2021). Artificial Intelligence for Music Composition. En E. R. Miranda (Ed.), *Handbook of Artificial Intelligence for Music: Foundations, Advanced Approaches, and Developments for Creativity* (pp. 53–73). Springer International Publishing. https://doi.org/10.1007/978-3-030-72116-9_3
- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5), 533–549. [https://doi.org/10.1016/0305-0548\(86\)90048-1](https://doi.org/10.1016/0305-0548(86)90048-1)
- Goetze, T. S. (2024). AI Art is Theft: Labour, Extraction, and Exploitation: Or, On the Dangers of Stochastic Pollocks. *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, 186–196. <https://doi.org/10.1145/3630106.3658898>
- Goldberg, D. E., Korb, B., & Deb, K. (1989). Messy genetic algorithms: Motivation, analysis, and first results. *Complex systems*, 3(5), 493–530.
- González Mares, M. (2019). Hernández-Sampieri, R. & Mendoza, C (2018). Metodología de la investigación. Las rutas cuantitativa, cualitativa y mixta. *Revista Universitaria Digital de Ciencias Sociales (RUDICS)*, 10(18), 92–95. <https://doi.org/10.22201/fesc.20072236e.2019.10.18.6>
- Grachten, M., & Chacón, C. E. C. (2017). *Strategies for Conceptual Change in Convolutional Neural Networks*.
- Guinness World Records. (2016, febrero 27). *Most performed composer at videogame concerts*.
<https://www.guinnessworldrecords.com/world-records/395815-most-performed-composer-at-videogame-concerts>.
- Hadjeres, G., Pachet, F., & Nielsen, F. (2016). *DeepBach: a Steerable Model for Bach Chorales Generation*.

- Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7), 1527–1554. <https://doi.org/10.1162/neco.2006.18.7.1527>
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Holland, J. H. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press.
- Holland, J. H. (1984). Genetic Algorithms and Adaptation. En O. G. Selfridge, E. L. Rissland, & M. A. Arbib (Eds.), *Adaptive Control of Ill-Defined Systems* (pp. 317–333). Springer US. https://doi.org/10.1007/978-1-4684-8941-5_21
- Holland, J. H. (1992a). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press.
- Holland, J. H. (1992b). Genetic Algorithms. *Scientific American*, 267(1), 66–73. <http://www.jstor.org/stable/24939139>
- Holland, J. H. (1992c). Genetic Algorithms. *Scientific American*, 267(1), 66–73. <http://www.jstor.org/stable/24939139>
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8), 2554–2558. <https://doi.org/10.1073/pnas.79.8.2554>
- Huang, A., & Wu, R. (2016). *Deep Learning for Music*.
- ISO. (s/f). *What is artificial intelligence (AI)?* <https://www.iso.org/artificial-intelligence/what-is-ai>.
- ISO. (2021). *Information technology — Artificial intelligence (AI) — Use cases Technologies de l'information — Intelligence artificielle (IA) — Cas pratiques*. <https://standards.iteh.ai/catalog/standards/sist/91e6c088-f9bb-46f8-851f-1dbc1de09078/iso-iec-tr-24030-2021>
- Kaliakatsos-Papakostas, M. A., Epitropakis, M. G., & Vrahatis, M. N. (2011). Weighted Markov Chain Model for Musical Composer Identification. En C. Di Chio, A. Brabazon, G. A. Di Caro, R. Drechsler, M. Farooq, J. Grahl, G. Greenfield, C. Prins, J. Romero, G. Squillero, E. Tarantino, A. G. B. Tettamanzi, N. Urquhart, & A. Ş. Uyar (Eds.), *Applications of Evolutionary Computation* (pp. 334–343). Springer Berlin Heidelberg.
- Kaliakatsos-Papakostas, M., & Cambouropoulos, E. (2014). *Probabilistic harmonization with fixed intermediate chord constraints*. <https://doi.org/10.13140/2.1.3079.5526>
- Kelley, H. J. (1960). Gradient Theory of Optimal Flight Paths. *ARS Journal*, 30(10), 947–954. <https://doi.org/10.2514/8.5282>
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(4598), 671–680. <https://doi.org/10.1126/science.220.4598.671>

- Krumhansl, C., & Kessler, E. (1982). Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychological review*, 89, 334–368. <https://doi.org/10.1037/0033-295X.89.4.334>
- Lawrence Livermore National Laboratory. (s/f). *The birth of Artificial Intelligence (AI) research*. <https://st.llnl.gov/news/look-back/birth-artificial-intelligence-ai-research>.
- Legg, S., & Hutter, M. (2007). *Universal Intelligence: A Definition of Machine Intelligence*.
- Lerdahl, F., & Jackendoff, R. (1985). *A Generative Theory of Tonal Music*. The MIT Press.
- Linnainmaa, S. (1976a). Taylor expansion of the accumulated rounding error. *BIT*, 16(2), 146–160. <https://doi.org/10.1007/BF01931367>
- Linnainmaa, S. (1976b). Taylor expansion of the accumulated rounding error. *BIT*, 16(2), 146–160. <https://doi.org/10.1007/BF01931367>
- Lopez-Rincon, O., Starostenko, O., & Martín, G. A.-S. (2018). Algorithmic music composition based on artificial intelligence: A survey. *2018 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, 187–193. <https://doi.org/10.1109/CONIELECOMP.2018.8327197>
- Low, C. (2019). *Awards - Nobuo Uematsu*. <https://nobuouematsu.weebly.com/awards.html>.
- Macdonald, N. (1954). Language translation by machine-a report of the first successful trial. *Computers and automation*, 3(2), 6–10.
- Maher, P. (1998). From Al-Jabr to Algebra. *Mathematics in School*, 27(4), 14–15. <http://www.jstor.org/stable/30211868>
- Markoff, A. (1910). Recherches sur un cas remarquable d'épreuves dépendantes. *Acta Mathematica*, 33(0), 87–104. <https://doi.org/10.1007/BF02393213>
- Martindale, C. (1998). Biological Bases of Creativity. En R. J. Sternberg (Ed.), *Handbook of Creativity* (pp. 137–152). Cambridge University Press. <https://doi.org/DOI: 10.1017/CBO9780511807916.009>
- Matzmatzmatzmatz. (2020). *Minimalist frequency chart for tuning your drums*. https://www.reddit.com/r/edmproduction/comments/gcxzwu/minimalist_frequency_chart_for_tuning_your_drums/.
- McCarthy, J., Minsky, M., Rochester, N., & Shannon, C. E. (2006). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955. *AI Mag.*, 27(4), 12–14. <https://doi.org/10.1609/AIMAG.V27I4.1904>
- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115–133. <https://doi.org/10.1007/BF02478259>

- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6), 1087–1092. <https://doi.org/10.1063/1.1699114>
- Meyer, L. B. (1956). Emotion and meaning in music. En *Emotion and meaning in music*. Chicago.
- Mitchell, T. (1997). *Machine learning* (Vol. 1). McGraw-hill New York.
- Moldaver, E., Merlino, H., & Fernández, E. (2014a). Composición Musical a través del uso de Algoritmos Genéticos. *Archivo de la Revista Latinoamericana de Ingeniería de Software*, 2(3), 149–156.
- Moldaver, E., Merlino, H., & Fernández, E. (2014b). *Composición Musical a Través del Uso de Algoritmos Genéticos Revista Latinoamericana de Ingeniería de Software*. 2(3).
- Moncada García, F. (1966). *Las más Sencilla, Útil y Práctica TEORÍA DE LA MÚSICA* (1a ed.). Ediciones Framong.
- Montesinos López, O. A., Montesinos López, A., & Crossa, J. (2022). Fundamentals of Artificial Neural Networks and Deep Learning. En O. A. Montesinos López, A. Montesinos López, & J. Crossa (Eds.), *Multivariate Statistical Machine Learning Methods for Genomic Prediction* (pp. 379–425). Springer International Publishing. https://doi.org/10.1007/978-3-030-89010-0_10
- Musicca. (s/f). *Círculo de quintas*. <https://www.musicca.com/es/circulo-de-quintas>.
- Myers, D. S. (2024). *Data Structures and Algorithms in Java: A Project-Based Approach*. Cambridge University Press. <https://doi.org/DOI: 10.1017/9781009260350>
- Navarro, M., Corchado Rodríguez, J., & Demazeau, Y. (2014). *A Musical Composition Application Based on a Multiagent System to Assist Novel Composers*.
- Palmer, W. A., Manus, M., & Lethco, A. V. (1994). *The Complete Book of Scales, Chords, Arpeggios & Cadences: Includes All the Major, Minor (natural, Harmonic, Melodic) & Chromatic Scales Plus Additional Instructions on Music Fundamentals*. Alfred Music.
- Papadopoulos, A., Roy, P., & Pachet, F. (2016). Assisted Lead Sheet Composition Using FlowComposer. En M. Rueher (Ed.), *Principles and Practice of Constraint Programming* (pp. 769–785). Springer International Publishing.
- Pattison, P. (1991). *Songwriting essential guide to lyric form and structure*. Hal Leonard Corporation.
- Pattison, P., & Welch, G. (2009). *Writing Better Lyrics: The Essential Guide to Powerful Songwriting*. Writer's Digest Books.
- PianoChordsPlays. (2023, agosto 6). *C major Chord Inversions on Piano*. <https://www.deviantart.com/pianochordsplays/art/C-major-Chord-Inversions-on-Piano-975675700>.
- Pilhofer, M., & Day, H. (2011). *Music Theory For Dummies*. Wiley.
- Piston, W., Devoto, M., & Jannery, A. (1987). *Harmony*. W.W. Norton & Company.

Ramos Herrera, I. A. (2025a). *arhcoder/MIA* (Harmfirst_a1_v.1.0.0). GitHub. <https://github.com/arhcoder/MIA>

Ramos Herrera, I. A. (2025b). Logarithmic Weighted Random Selector Algorithm: A Novel Approach for Biasing Selection Based on Positional Order Without Hyperparameters. En L. Martínez-Villaseñor, G. Ochoa-Ruiz, M. Montes Rivera, M. L. Barrón-Estrada, & H. G. Acosta-Mesa (Eds.), *Advances in Computational Intelligence. MICAI 2024 International Workshops* (pp. 43–54). Springer Nature Switzerland.

Ramos Herrera, I. A. (2025c, marzo 5). *arhcoder/MIA-Experiments*. <https://github.com/arhcoder/MIA-Experiments>.

Ramos Herrera, I. A. (2025d, marzo 5). *MIA Harmfirst: Evaluación de Ideas Musicales*. <https://arhcoder.github.io/MIA-Experiments/>.

Ravera, L. (2022). *Tonal and modal music: what is the difference?* <https://www.italianpiano.com/music-lessons/tonal-music-modal-music-difference/>

Read, G. (1974). *Music notation : a manual of modern practice*. Gollancz. <https://cir.nii.ac.jp/crid/1130000795861984768.bib?lang=ja>

Real Academia Española. (2001). *Nuevo Diccionario De La Lengua Española* (22a ed., Vols. 1, 2). Espasa-Calpe.

Robbins, H., & Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3), 400–407. <https://doi.org/10.1214/aoms/1177729586>

Rosen, K. H. (2002). *Discrete Mathematics and Its Applications* (5th ed.). McGraw-Hill Higher Education.

Rosenblatt, F. (1957). *The perceptron: A perceiving and recognizing automaton* (Números 85-460–1). Project PARA, Cornell Aeronautical Laboratory.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408. <https://doi.org/10.1037/h0042519>

Rosenblatt, F. (1961). *PRINCIPLES OF NEURODYNAMICS. PERCEPTRONS AND THE THEORY OF BRAIN MECHANISMS*. <https://doi.org/10.21236/AD0256582>

Rumsey, F., & McCormick, T. (2004). *Sonido y grabación: Introducción a las técnicas sonoras*. Instituto Oficial de Radio y Televisión.

Russell, G. (1959). *The Lydian Chromatic Concept of Tonal Organization for Improvisation*. Concept Publishing Company.

Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3), 210–229. <https://doi.org/10.1147/rd.33.0210>

Sandred, Ö., Laurson, M., & Kuuskankare, M. (2009). Revisiting the Illiac Suite - A rule-based approach to stochastic processes. *Sonic Ideas/Ideas Sonicas*, 2, 42–46.

- Schoenberg, A. (1911). *Harmonielehre*. Faber & Faber.
- Schoenberg, A. (1967). *Fundamentals of Musical Composition* (1st ed.). Faber and Faber Limited.
- Schoenberg, A. (1983). *Theory of Harmony*. University of California Press.
- Scholes, P. A. ., & Ward, J. Owen. (1977). *The concise Oxford dictionary of music*. Oxford University Press.
- Scirea, M., & Barros, G. (2015). *SMUG: Scientific Music Generator*.
- Serafio Méndez, R. M. (2023). *Tres Jacarandas*.
- Serafio Méndez, R. M. (2025a). *Desaparecer*.
- Serafio Méndez, R. M. (2025b). *Mi dulce Madre Tierra*.
- Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3), 379–423. <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>
- Shoemaker, E. (2024). Is AI Art Theft? The Moral Foundations of Copyright Law in the Context of AI Image Generation. *Philosophy & Technology*, 37(3), 114. <https://doi.org/10.1007/s13347-024-00797-x>
- Soto Valdez, A. (2021). *Aceptación*.
- Sr. Imposible. (2024, enero 29). *Las figuras musicales*. <https://hacercanciones.com/tutorial/las-figuras-musicales/>.
- Steinke, G. A., & Harder, P. O. (2009). *Harmonic Materials in Tonal Music: A programed Course Part 1* (10th ed., Vol. 1). Prentice Hall.
- Stevenson, A. (2024, julio). *ML Algorithm vs. ML Model*. <https://community.cisco.com/t5/devnet-general-knowledge-base/ml-algorithm-vs-ml-model/ta-p/5120750>.
- Tenney, J. (1988). *Stochastic string quartet* (Número partes 1-4). Smith Publications.
- Teoría Musical. (2021). *Clase 9: Acentuación (tiempos fuertes y débiles)*. <https://www.youtube.com/watch?v=llr6yFhAi5k>.
- The Editors of Encyclopaedia Britannica. (2025). *motive*. <https://www.britannica.com/art/motive-music>.
- Tsai, C.-W., & Chiang, M.-C. (2023). Introduction. *Handbook of Metaheuristic Algorithms*, 3–11. <https://doi.org/10.1016/B978-0-44-319108-4.00014-9>
- Tsiros, A. (2017). *Sonic Xplorer: A Machine Learning Approach for Parametric Exploration of Sound*. <https://doi.org/10.14236/ewic/EVA2017.34>
- Turing, A. M. (1950). COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236), 433–460. <https://doi.org/10.1093/mind/LIX.236.433>
- UAA. (2017). *CENTRO DE LAS ARTES Y CULTURA: LICENCIATURA EN MÚSICA: PLAN CARRERA 2017*. https://dgdg.uaa.mx/catalogo/artes_y_cultura/lic_en_musica.pdf.

Uematsu, N. (1994, marzo 25). *Final Fantasy VI Original Soundtrack*. <https://archive.org/details/final-fantasy-vi-original-soundtrack>.

Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 12.

Vidya Gaem Awards. (2015). *Winners – Vidya Gaem Awards*. <https://2015.vidyagaemawards.com/winners>.

Wang, F. (2024). Design and Implementation of Automatic Music Composition System Based on Deep Learning. *Journal of Electrical Systems*, 20(3s), 355–373.

Wizard Dojo. (2017, febrero 9). *Where My 2016 Awards At? (Also Something About Undertale)*. <https://wizarddojo.com/2017/02/09/where-my-2016-awards-at-also-something-about-undertale/>.

Woolhouse, M., & Cross, I. (2008). *Using Kolmogorov-Smirnov to Determine the Effect of Interval Cycles on the Formation of Krumhansl & Kessler's (1982) Tonal Hierarchies*.

Xu, D., & Xu, H. (2024). Application of genetic algorithm in model music composition innovation. *Applied Mathematics and Nonlinear Sciences*, 9(1). <https://doi.org/doi:10.2478/amns.2023.2.00070>

Zhang, Z., Yu, Y., & Takasu, A. (2024). Controllable Syllable-Level Lyrics Generation from Melody with Prior Attention. *IEEE Transactions on Multimedia*, 1–11. <https://doi.org/10.1109/TMM.2024.3443664>

Anexos

A.1 Calendario de actividades

La siguiente ilustración muestra el calendario de actividades para el desarrollo del proyecto MIA, entre agosto de 2024 y junio de 2025. A continuación de la imagen se describen los sucesos.

Proyecto MIA

Calendario de actividades (Ago 2024 - Junio 2025)

Fase	Ago	Sep	Oct	Nov	Dic	Ene	Feb	Mar	Abr	May	Jun
1. Estudio del problema											
2. Formulación del algoritmo											
3. Construcción del algoritmo											
4. Evaluación inicial del algoritmo											
5. Formulación de métrica											
6. Evaluación experimental del algoritmo											
7. Documentación de resultados											
Documentación de tesina											
Discusión y presentación											

Según el calendario, cada fase se realizó de la siguiente manera:

- [1] **Estudio del problema:** Entre el 14 de agosto y el 20 de diciembre de 2024 (128 días).
- [2] **Formulación del algoritmo:** Entre el 16 de octubre y el 24 de diciembre (69 días) y de nuevo retomada en la última semana de marzo de 2025 y la segunda semana de abril de 2025, sumando estas dos nuevas repeticiones 5 días más.
- [3] **Construcción del algoritmo:** Entre el 6 de enero de 2025 y el 14 de abril de 2025.
 - Esta fase se terminó inicialmente el 24 de marzo de 2025, pero al no obtener resultados aceptables, se repitió la fase 2 nuevamente, reformulando nuevamente el algoritmo.
 - Entre las últimas semanas de marzo y las primeras de abril se volvió a terminar el desarrollo, pero nuevamente el algoritmo no cumplía con lo necesitado, por lo que se repitió nuevamente la fase 2 y se construyó otra implementación del algoritmo.
 - Se registraron las horas puntuales utilizadas exclusivamente en la escritura de código, concluyendo el proceso con **166 horas de programación** repartidas a lo largo de **31 días** puntuales de trabajo; es decir, en promedio 5.3 horas por día.
- [4] **Evaluación inicial del algoritmo:** Se realizó en tres ocasiones puntuales, entre dos semanas de marzo y abril. Las dos primeras ocasiones la evaluación demostró que el algoritmo necesitaba reformulaciones y la tercer ocasión permitió consolidar la arquitectura *Harmfirst a1*.
- [5] **Formulación de la métrica:** Se realizó desde inicios de febrero a mediados de marzo, en donde además de la métrica se construyó una interfaz web para que los evaluadores interactúen con sus elementos durante la fase experimental 6.
- [6] **Evaluación experimental del algoritmo:** Presencialmente en las instalaciones del *Departamento de Música* de la *Universidad Autónoma de Aguascalientes*, entre la segunda y la tercer semana de mayo.
- [7] **Documentación de resultados:** Realizada en las dos últimas semanas de mayo.
- [8] **Documentación de tesina:** Comenzada entre el 23 de septiembre de 2024 y el 16 de junio de 2025 con períodos fluctuantes. Con un total de **824 horas**.
- [9] **Discusión y presentación:** Entre la segunda y tercer semana de junio, concluyendo la *defensa* presencial ante un comité tutorial.

A.2 Cartas de consentimiento de participación

A.2.1 Carta de consentimiento del Sujeto 01

Carta de Consentimiento Informado para Participación en Evaluación de Composiciones Musicales Generadas por Algoritmo

A quien corresponda:

Yo, Oscar Rodríguez Fragoso, estudiante de la Licenciatura en Música de la Universidad Autónoma de Aguascalientes, con número de matrícula 261921, actualmente cursando el semestre número 8, por medio de la presente manifiesto mi consentimiento para participar en el proceso de evaluación de obras musicales generadas por el proyecto de investigación titulado: "**Algoritmo de Composición Musical Autónoma sin Entrenamiento aplicando Procesamiento de Lenguaje Natural, Cómputo Evolutivo y Metaheurísticas**", desarrollado por el estudiante Iván Alejandro Ramos Herrera, de la carrera de Ingeniería en Computación Inteligente de esta misma institución.

Declaro haber sido informado/a que mi participación consiste en escuchar una serie de piezas musicales generadas por el algoritmo del proyecto, y en evaluarlas de forma individual mediante un cuestionario denominado "**Métrica de Estética Musical Perceptual**", el cual está diseñado para calificar distintos criterios estructurales de las composiciones presentadas.

Asimismo, declaro que:

- Participo libremente y con una única remuneración de cien pesos mexicanos.
- Emitiré mis valoraciones con objetividad técnica, de acuerdo con los conocimientos adquiridos en mi formación académica.
- Autorizo que las respuestas que proporcione sean utilizadas con fines estadísticos y analíticos dentro del marco del mencionado proyecto, con la finalidad de contribuir al análisis y validación de los resultados obtenidos.
- Autorizo la toma de fotografías durante el proceso de evaluación, así como su posible publicación únicamente con fines ilustrativos y académicos dentro del trabajo de investigación.

He sido informado/a de que puedo desistirme de participar sin que esto represente consecuencia alguna para mi situación académica.

Departamento de Música de la Universidad Autónoma de Aguascalientes,

a 14 de mayo de 2025.

Firma del participante: _____

Nombre completo del participante: Oscar Rodríguez Fragoso.

Autorizo que se mencione mi nombre e identidad en publicaciones dentro del marco del proyecto.



Prefiero que mi identidad se mantenga anónima en publicaciones dentro del marco del proyecto.



**Carta de Consentimiento Informado para Participación en Evaluación de
Composiciones Musicales Generadas por Algoritmo**

A quien corresponda:

Yo, [REDACTED], estudiante de la Licenciatura en Música de la Universidad Autónoma de Aguascalientes, con número de matrícula [REDACTED], actualmente cursando el semestre número 8, por medio de la presente manifiesto mi consentimiento para participar en el proceso de evaluación de obras musicales generadas por el proyecto de investigación titulado: "Algoritmo de Composición Musical Autónoma sin Entrenamiento aplicando Procesamiento de Lenguaje Natural, Cómputo Evolutivo y Metaheurísticas", desarrollado por el estudiante Iván Alejandro Ramos Herrera, de la carrera de Ingeniería en Computación Inteligente de esta misma institución.

Declaro haber sido informado/a que mi participación consiste en escuchar una serie de piezas musicales generadas por el algoritmo del proyecto, y en evaluarlas de forma individual mediante un cuestionario denominado "Métrica de Estética Musical Perceptual", el cual está diseñado para calificar distintos criterios estructurales de las composiciones presentadas.

Asimismo, declaro que:

- Participo libremente y con una única remuneración de cien pesos mexicanos.
- Emitiré mis valoraciones con objetividad técnica, de acuerdo con los conocimientos adquiridos en mi formación académica.
- Autorizo que las respuestas que proporcione sean utilizadas con fines estadísticos y analíticos dentro del marco del mencionado proyecto, con la finalidad de contribuir al análisis y validación de los resultados obtenidos.
- Autorizo la toma de fotografías durante el proceso de evaluación, así como su posible publicación únicamente con fines ilustrativos y académicos dentro del trabajo de investigación.

He sido informado/a de que puedo desistirme de participar sin que esto represente consecuencia alguna para mi situación académica.

Departamento de Música de la Universidad Autónoma de Aguascalientes,

a 14 de Mayo de 2025.

Firma del participante: [REDACTED]

Nombre completo del participante: [REDACTED]

*Autorizo que se mencione mi nombre e identidad
en publicaciones dentro del marco del proyecto.*



*Prefiero que mi identidad se mantenga anónima
en publicaciones dentro del marco del proyecto.*



**Carta de Consentimiento Informado para Participación en Evaluación de
Composiciones Musicales Generadas por Algoritmo**

A quien corresponda:

Yo, Angela María Salazar Córdova, estudiante de la Licenciatura en Música de la Universidad Autónoma de Aguascalientes, con número de matrícula 2620 78, actualmente cursando el semestre número 8, por medio de la presente manifiesto mi consentimiento para participar en el proceso de evaluación de obras musicales generadas por el proyecto de investigación titulado: "Algoritmo de Composición Musical Autónoma sin Entrenamiento aplicando Procesamiento de Lenguaje Natural, Cómputo Evolutivo y Metaheurísticas", desarrollado por el estudiante Iván Alejandro Ramos Herrera, de la carrera de Ingeniería en Computación Inteligente de esta misma institución.

Declaro haber sido informado/a que mi participación consiste en escuchar una serie de piezas musicales generadas por el algoritmo del proyecto, y en evaluarlas de forma individual mediante un cuestionario denominado "Métrica de Estética Musical Perceptual", el cual está diseñado para calificar distintos criterios estructurales de las composiciones presentadas.

Asimismo, declaro que:

- Participo libremente y con una única remuneración de cien pesos mexicanos.
- Emitiré mis valoraciones con objetividad técnica, de acuerdo con los conocimientos adquiridos en mi formación académica.
- Autorizo que las respuestas que proporcione sean utilizadas con fines estadísticos y analíticos dentro del marco del mencionado proyecto, con la finalidad de contribuir al análisis y validación de los resultados obtenidos.
- Autorizo la toma de fotografías durante el proceso de evaluación, así como su posible publicación únicamente con fines ilustrativos y académicos dentro del trabajo de investigación.

He sido informado/a de que puedo desistirme de participar sin que esto represente consecuencia alguna para mi situación académica.

Departamento de Música de la Universidad Autónoma de Aguascalientes,

a 14 de mayo de 2025.

Firma del participante: .

Nombre completo del participante: Angela María Salazar Córdova

Autorizo que se mencione mi nombre e identidad en publicaciones dentro del marco del proyecto.



Prefiero que mi identidad se mantenga anónima en publicaciones dentro del marco del proyecto.



**Carta de Consentimiento Informado para Participación en Evaluación de
Composiciones Musicales Generadas por Algoritmo**

A quien corresponda:

Yo, Ana Alejandra Celaya Marín, estudiante de la Licenciatura en Música de la Universidad Autónoma de Aguascalientes, con número de matrícula 233300, actualmente cursando el semestre número 8vo por medio de la presente manifiesto mi consentimiento para participar en el proceso de evaluación de obras musicales generadas por el proyecto de investigación titulado: "Algoritmo de Composición Musical Autónoma sin Entrenamiento aplicando Procesamiento de Lenguaje Natural, Cómputo Evolutivo y Metaheurísticas", desarrollado por el estudiante Iván Alejandro Ramos Herrera, de la carrera de Ingeniería en Computación Inteligente de esta misma institución.

Declaro haber sido informado/a que mi participación consiste en escuchar una serie de piezas musicales generadas por el algoritmo del proyecto, y en evaluarlas de forma individual mediante un cuestionario denominado "Métrica de Estética Musical Perceptual", el cual está diseñado para calificar distintos criterios estructurales de las composiciones presentadas.

Asimismo, declaro que:

- Participo libremente y con una única remuneración de cien pesos mexicanos.
- Emitiré mis valoraciones con objetividad técnica, de acuerdo con los conocimientos adquiridos en mi formación académica.
- Autorizo que las respuestas que proporcione sean utilizadas con fines estadísticos y analíticos dentro del marco del mencionado proyecto, con la finalidad de contribuir al análisis y validación de los resultados obtenidos.
- Autorizo la toma de fotografías durante el proceso de evaluación, así como su posible publicación únicamente con fines ilustrativos y académicos dentro del trabajo de investigación.

He sido informado/a de que puedo desistirme de participar sin que esto represente consecuencia alguna para mi situación académica.

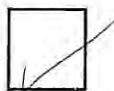
Departamento de Música de la Universidad Autónoma de Aguascalientes,

a 16 de mayo de 2025.

Firma del participante: [Firma]

Nombre completo del participante: Ana Alejandra Celaya Marín

Autorizo que se mencione mi nombre e identidad
en publicaciones dentro del marco del proyecto.



Prefiero que mi identidad se mantenga anónima
en publicaciones dentro del marco del proyecto.



**Carta de Consentimiento Informado para Participación en Evaluación de
Composiciones Musicales Generadas por Algoritmo**

A quien corresponda:

Yo, Jorge Leonardo Díaz Morales, estudiante de la Licenciatura en Música de la Universidad Autónoma de Aguascalientes, con número de matrícula 189374, actualmente cursando el semestre número 8, por medio de la presente manifiesto mi consentimiento para participar en el proceso de evaluación de obras musicales generadas por el proyecto de investigación titulado: "Algoritmo de Composición Musical Autónoma sin Entrenamiento aplicando Procesamiento de Lenguaje Natural, Cómputo Evolutivo y Metaheurísticas", desarrollado por el estudiante Iván Alejandro Ramos Herrera, de la carrera de Ingeniería en Computación Inteligente de esta misma institución.

Declaro haber sido informado/a que mi participación consiste en escuchar una serie de piezas musicales generadas por el algoritmo del proyecto, y en evaluarlas de forma individual mediante un cuestionario denominado "Métrica de Estética Musical Perceptual", el cual está diseñado para calificar distintos criterios estructurales de las composiciones presentadas.

Asimismo, declaro que:

- Participo libremente y con una única remuneración de cien pesos mexicanos.
- Emitiré mis valoraciones con objetividad técnica, de acuerdo con los conocimientos adquiridos en mi formación académica.
- Autorizo que las respuestas que proporcione sean utilizadas con fines estadísticos y analíticos dentro del marco del mencionado proyecto, con la finalidad de contribuir al análisis y validación de los resultados obtenidos.
- Autorizo la toma de fotografías durante el proceso de evaluación, así como su posible publicación únicamente con fines ilustrativos y académicos dentro del trabajo de investigación.

He sido informado/a de que puedo desistirme de participar sin que esto represente consecuencia alguna para mi situación académica.

Departamento de Música de la Universidad Autónoma de Aguascalientes,

a 21 de Novio de 2025.

Firma del participante:

Nombre completo del participante: Jorge Leonardo Díaz Morales

Autorizo que se mencione mi nombre e identidad
en publicaciones dentro del marco del proyecto.



Prefiero que mi identidad se mantenga anónima
en publicaciones dentro del marco del proyecto.



TESIS TESIS TESIS TESIS TESIS



@arhcoder

TESIS TESIS TESIS TESIS TESIS