



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Facultad de Ingeniería en Electricidad y Computación

“COMPOSICIÓN E INTERPRETACIÓN CONCURRENTES
DE MELODÍAS MUSICALES EN TIEMPO REAL
SINTETIZADAS EN BASE A EMOCIONES HUMANAS
UTILIZANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL”

TESIS DE GRADO

Previo a la obtención del Título de:

**INGENIERO EN CIENCIAS COMPUTACIONALES
ORIENTACIÓN SISTEMAS MULTIMEDIA**

**INGENIERO EN CIENCIAS COMPUTACIONALES
ORIENTACIÓN SISTEMAS TECNOLÓGICOS**

Presentada por:

Efraín José Astudillo Vargas

Pedro Pablo Lucas Bravo

GUAYAQUIL - ECUADOR

2015

AGRADECIMIENTO

*Agradezco eternamente a Dios por
permitirnos alcanzar este objetivo.*

*A mis padres, familiares y amigos que fueron
de ayuda y soporte en mi etapa de estudiante universitario.*

*A Roberto Bernous y Enrique Bernous por habernos
brindado sus conocimientos musicales requeridos en este trabajo.*

*Al Ph.D. Enrique Peláez, nuestro director de tesis,
por su paciencia y guía.*

*Un agradecimiento muy especial a todos mis profesores
y compañeros de mi carrera que han sido de inspiración
y de ejemplo para mi vida profesional.*

Efraín Astudillo V.

*Al Arquitecto del Universo, por ser el soporte espiritual
para superar las dificultades encontradas en el camino.*

*A mis padres, por su apoyo incondicional en esta etapa de mi
vida y sacrificarse fervorosamente para alcanzar este logro.*

*A mis hermanas, a quiénes a pesar de la distancia
las tengo siempre presentes.*

*A mi compañero de tesis, por el trabajo conjunto
desempeñado en el proceso.*

*A mis compañeros de Freaky Creations, por su inspiración
para el tema escogido y comprensión en el desarrollo de este
trabajo, en especial Roberto Bernous y Enrique Bernous
por el aporte con su conocimiento musical.*

*A mis familiares y amigos, los cuales han sido parte
importante en mi desarrollo emocional y profesional.*

*A nuestros profesores, por su dedicación y empeño
en formarnos para la sociedad del conocimiento.*

Pedro Lucas B.

DEDICATORIA

*Al Creador del Universo que nos ha dado
la fortaleza necesaria para terminar este trabajo de tesis.
A mis padres y familiares por su preocupación y apoyo durante
toda esta etapa de constante aprendizaje.
A Pedro Lucas, mi compañero de tesis, por compartir este
tiempo de dedicación a este trabajo.
Y en general a todos aquellos que de alguna u otra manera
fueron parte fundamental y elemental en mi proceso de aprendizaje
como ser humano y como profesional.*

Efraín Astudillo V.

*Al Arquitecto del Universo, cuya orquestación
del mundo hizo posible el conocimiento
generado en este trabajo.*

*A mi mamá, por su preocupación diaria
en mi bienestar y su apoyo incondicional.*

*A mi papá, por sus consejos y sacrificio
durante todas las etapas de mi vida.*

*A mis hermanas, a quiénes deseo inspirar
para que sean excelentes profesionales.*

*A mis compañeros de Freaky Creations,
por el soporte de ideas innovadoras.*

*A mis familiares y amigos, por su
consideración en mi accionar.*

*A todos aquellos que desean un mundo
donde el arte, la filosofía y la tecnología
permitan la trascendencia del ser humano.*

Pedro Lucas B.

TRIBUNAL DE SUSTENTACIÓN



MSc. Sara Ríos O.

PRESIDENTA



Ph.D. Enrique Peláez J.

DIRECTOR DE TESIS



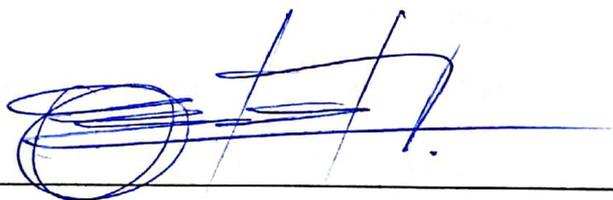
Ph.D. Sixto García A.

MIEMBRO PRINCIPAL

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesis de Grado, nos corresponde exclusivamente; y el patrimonio intelectual del mismo a la Escuela Superior Politécnica del Litoral.”

(Reglamento de exámenes y títulos profesionales de la ESPOL)



Efraín José Astudillo Vargas



Pedro Pablo Lucas Bravo

RESUMEN

Este trabajo explora la aplicación de la Inteligencia Artificial con respecto a la música. El alcance implica únicamente *melodías musicales* que son secuencias de notas ejecutadas en el tiempo y las técnicas de Inteligencia Artificial que soportan la generación de esas secuencias.

Se detalla un modelo que representa el conocimiento musical con respecto a melodías, basado en la revisión del estado del arte y en el criterio de artistas practicantes de la música experimental. También el diseño y la implementación de un sistema que soporta este modelo, el cuál procesa la entrada de eventos MIDI de un teclado para componer e interpretar melodías de manera concurrente en tiempo real; es decir, acompaña al músico o músicos que aportan con la entrada. Además, este sistema está guiado por la intención emocional del músico intérprete de tal manera que genere sentimientos en la audiencia comparable a como un artista humano lo haría.

La arquitectura del modelo propuesto y el sistema prototipo fueron sometidos a una prueba de estética, prueba de Turing, prueba de percepción emocional con respecto a oyentes, y pruebas de rendimiento. Estas pruebas son analizadas para evaluar el desempeño del enfoque basado en técnicas de

Inteligencia Artificial para la solución del problema, en el que se implementaron dos componentes del modelo que fueron: *Cadenas de Markov*, por su uso para secuencia de elementos y *Lógica Difusa*, por su utilización en situaciones inciertas, para este caso en el uso de emociones humanas.

ÍNDICE GENERAL

AGRADECIMIENTO	ii
DEDICATORIA	iv
TRIBUNAL DE SUSTENTACIÓN	vi
DECLARACIÓN EXPRESA	vii
RESUMEN	viii
ÍNDICE GENERAL	x
ÍNDICE DE FIGURAS	xviii
ÍNDICE DE TABLAS	xxv
INTRODUCCIÓN	xxvii
1. GENERALIDADES	1
1.1. Planteamiento y Justificación del Proyecto de Investigación	1
1.1.1. Problema	1
1.1.2. Justificación	3
1.1.3. Solución Propuesta	4
1.2. Objetivos de la Tesis	5

1.2.1. Objetivo General	5
1.2.2. Objetivos Específicos	5
1.3. Metodología	6
1.4. Estructura de Documento	9
2. MARCO TEÓRICO	11
2.1. Teoría Musical Básica	11
2.1.1. Notas Musicales y Escalas	12
2.1.2. Ritmo, Tempo, y Longitud de Notas	15
2.1.3. Métrica	17
2.1.4. Acordes y Armonías	18
2.1.5. Melodías y Motivos	21
2.1.6. Composición Musical	22
2.1.7. Interpretación Musical	24
2.2. Síntesis de Sonido	25
2.2.1. Definición	25
2.2.2. Música Electroacústica	26
2.2.3. Componentes Básicos de Síntesis	27
2.2.3.1. Osciladores y Wavetables	27
2.2.3.2. Filtros	29
2.2.3.3. Envolventes y Enrutamiento	31
2.2.4. El Sintetizador	33

2.2.5. Modelamiento de Sonido	35
2.2.6. Síntesis mediante software	38
2.3. MIDI (Musical Instrument Digital Interface)	39
2.3.1. Definición	39
2.3.2. Componentes de Hardware	41
2.3.3. Componentes de Software	44
2.4. Lenguajes de Programación de Síntesis en Tiempo Real	48
2.4.1. Descripción	48
2.4.2. El lenguaje SuperCollider	49
2.5. Sistemas Concurrentes en Tiempo Real	52
2.5.1. Definición	52
2.5.2. Características Principales	52
2.5.3. Consideraciones al momento de desarrollar este tipo de sistemas	55
2.5.4. Uso del lenguaje de programación C++ para estos sistemas	57
2.6. Inteligencia Artificial aplicada a la música	59
2.6.1. Cognición Musical	59
2.6.1.1. Conocimientos de Composición e Interpretación musical	60
2.6.1.2. Estudio de las emociones producidas por la música	64

2.6.1.3. Percepción de emociones mediante sonidos	
específicos	69
2.6.2. Técnicas de Inteligencia Artificial para Composición	
Musical	74
2.6.2.1. Representación del Conocimiento Musical y	
Emocional	75
2.6.2.2. Estructuras que soportan las representaciones	101
2.6.2.3. Algoritmos de Composición Musical	106
2.6.2.4. Estrategias de asociación entre emociones y	
color del sonido	117
2.6.3. Manejo de datos relacionados a la evaluación estética de	
la música	121
2.6.3.1. Parámetros a recopilar con respecto a	
composición y sentimientos	121
2.6.3.2. ¿Cómo interpretar y analizar los datos	
recopilados?	125
2.6.3.3. ¿Cómo establecer y comunicar los resultados	
obtenidos?	126
3. ANÁLISIS Y SELECCIÓN DE LAS TÉCNICAS DE INTELIGENCIA	
ARTIFICIAL	128
3.1. Composición Musical	131

3.1.1. Representación del Conocimiento	131
3.1.2. Algoritmos a aplicar	149
3.2. Síntesis de Sonido basado en emociones	162
3.2.1. Representación del Conocimiento	162
3.2.2. Algoritmos a aplicar	166
3.3. Integración e interacción entre las estrategias seleccionadas	167
4. DISEÑO DE LA INFRAESTRUCTURA TECNOLÓGICA DEL SISTEMA	170
4.1. Diseño del componente de Hardware	172
4.1.1. Elementos físicos de la infraestructura	172
4.1.1.1. Controladores MIDI	172
4.1.1.2. Interfaces de Audio	173
4.1.1.3. Parlantes y otros dispositivos de salida	174
4.1.2. Ensamblaje de los elementos hacia el computador	175
4.2. Diseño del componente de Software	176
4.2.1. Adquisición de datos	176
4.2.2. Almacenamiento del Conocimiento	178
4.2.3. Distribución de entradas a procesamientos concurrentes	180
4.2.4. Procesamiento concurrente temporizado	182
4.2.5. Estrategias de composición y síntesis de audio aplicadas en tiempo real	184
4.2.6. Módulo de entrenamiento del sistema	187

4.2.7. Módulo de producción del sistema	194
4.2.8. Salidas producidas por el sistema	201
4.2.9. Distribución de las salidas hacia un lenguaje de programación de síntesis	204
4.2.10. Módulo de síntesis de audio y la reproducción del resultado al exterior	206
4.2.11. Módulo de grabación de los resultados emitidos	209
4.3. Diseño de Pruebas	210
5. IMPLEMENTACIÓN, ENTRENAMIENTO Y EJECUCIÓN DEL SISTEMA	218
5.1. Consideraciones de implementación	219
5.1.1. Herramientas utilizadas	219
5.1.2. Restricciones del ambiente	222
5.1.3. Optimizaciones	223
5.2. Entrenamiento del sistema	224
5.2.1. Características de los expertos que aportarán al entrenamiento	224
5.2.2. Facilidades de interacción con el sistema para los expertos	225
5.2.3. Plan de escenarios de entrenamiento	226
5.3. Ejecución del sistema	228

5.3.1. Interpretación musical en tiempo real de melodías por	
parte de los expertos y el sistema	228
5.3.2. Recolección de información para evaluar la estética de la	
composición	229
5.3.2.1. Características de los escuchas	229
5.3.2.2. Exposición de la música generada ante los	
escuchas	229
5.3.2.3. Recolección de información de la experiencia	
auditiva de los escuchas en términos de	
composición, sonido y emociones	230
5.3.3. Recolección de datos acerca del rendimiento del sistema	231
6. ANÁLISIS DE LOS RESULTADOS	233
6.1. Presentación y Análisis de los Resultados de la composición	
musical	233
6.2. Presentación y Análisis de los Resultados de la síntesis de	
sonido y su evocación de emociones	237
6.3. Presentación y Análisis de los Resultados del Rendimiento del	
Sistema	249
6.4. Análisis del proceso ingenieril aplicado a la música	258
CONCLUSIONES Y RECOMENDACIONES	263
Conclusiones	263

Recomendaciones	267
A. Lineamientos Obtenidos de la Entrevista No Estructurada Aplicada a los	
Músicos Experimentales	270
Bibliografía	281

ÍNDICE DE FIGURAS

2.1. Notas musicales para las teclas blancas de un teclado musical	12
2.2. Notas musicales para las teclas negras de un teclado musical	13
2.3. Notas musicales y sus octavas	14
2.4. Tiempos para notas y silencios	17
2.5. Uso de la línea de Bar en una partitura	18
2.6. Intervalos de notas musicales	19
2.7. Acordes tónico, dominante y subdominante en la clave C mayor	
(C)	20
2.8. Ejes de tiempo (time) y tono (pitch) para una melodía	21
2.9. Ondas seno, cuadrada, triangular y diente de sierra	28
2.10. Estructura básica de un Wavetable	29
2.11. Filtros: (a) Paso-Bajo, (b) Paso-Alto, (c) Paso-Banda, (d)	
Rechazo-Banda	31
2.12. Envolvente ADSR y sus segmentos	32
2.13. Sintetizador Moog	34
2.14. Modelo de interacción entre un usuario y un sintetizador	35
2.15. Proceso de síntesis sustractiva	36
2.16. Proceso de síntesis aditiva	37

2.17. Plugin de síntesis de sonido Hybrid	39
2.18. Puertos MIDI THRU, MIDI OUT y MIDI IN	42
2.19. Controlador MIDI en forma de teclado musical	43
2.20. Controlador MIDI Linnstrument	43
2.21. Ejemplo de un mensaje MIDI	45
2.22. Notas musicales, números MIDI y frecuencias	46
2.23. MPC Essentials: Secuenciador MIDI de Akai	47
2.24. IDE de programación para Supercollider con código de ejemplo	51
2.25. Sistema en Tiempo Real	53
2.26. Modelo de conmoción que relaciona al intérprete (Actor/musician) con el oyente (Listener)	65
2.27. Versión modificada del modelo Brunswik's Lens para la comunicación de emociones en la música	66
2.28. Grupos de emociones distribuidos en ocho clases distintas	68
2.29. Escala Self Assessment Manikin (SAM)	73
2.30. Línea de tiempo de un secuenciador MIDI	79
2.31. Métodos de Inteligencia Artificial relacionados a la composición musical	82
2.32. Conjuntos difusos para la variable lingüística <i>size</i>	84
2.33. Cadena de Markov representada por una máquina de estado finito	89

2.34. Grafo y matriz de transiciones para una cadena de Markov de primer orden	91
2.35. Cadena oculta de Markov para el problema del lanzamiento de monedas.	92
2.36. Derivación de una gramática de estructura de frase	95
2.37. Derivación de una simple representación de un L-System	99
2.38. Red de transiciones aumentada (ATN), para la producción de melodías	100
2.39. Manipulación de arreglos para aplicar la operación <i>crossover</i>	102
2.40. (a) Grafo no dirigido. (b) Grafo dirigido. (c) Grafo ponderado	104
2.41. (a) Grafo dirigido. (b) Matriz de adyacencia. (c) Lista de adyacencia	105
2.42. Recombinación de dos cadenas de bits de longitud 8. # puede ser 0 o 1	113
2.43. Componentes de síntesis de sonidos para evaluar respuesta emocional	119
2.44. Modelo de generación de sonido a través de emociones	120
2.45. La Prueba de Turing	123
3.1. Arquitectura de Representación del Conocimiento para Composición Musical	132

3.2. Matriz de transiciones de notas de la Tocata y Fuga en re menor, BWV 565 de Johann Sebastian Bach	136
3.3. Funciones de Membresía para patrones melódicos con respecto a emociones	148
3.4. Modelo de Metrónomo para la Adquisición de Datos y la Ejecución de la Composición	154
3.5. Ejemplo de Árboles para Emociones	157
3.6. Ejemplo de la Estructura de un Nodo para un Árbol de Emoción	157
3.7. Claves más cercanas a 10 para un Árbol de Felicidad	158
3.8. Arquitectura de Representación del Conocimiento para Síntesis de Sonido Basado en Emociones	164
3.9. Arquitectura Completa de Representación del Conocimiento Musical	168
4.1. Diagrama de Bloques del Sistema	171
4.2. Infraestructura de Hardware del Sistema	176
4.3. Módulo de Adquisición de Datos del Sistema	177
4.4. Modelo de Almacenamiento de Datos del Sistema	178
4.5. Diagrama de Actividades de la Distribución Concurrente de Entradas al Sistema	181
4.6. Ejecución del Sistema restringida a un Metrónomo	183
4.7. Hilos que representan las melodías producidas por el sistema	184

4.8. (a) Diagrama de Actividades para el Proceso de Grabación del Sistema. (b) Diagrama de Actividades para el Proceso de Entrenamiento del Sistema	188
4.9. Diagrama de Clases para el Módulo de Entrenamiento	191
4.10. Diagrama de Secuencia para el Módulo de Entrenamiento	193
4.11. Diagrama de Actividades para el Módulo de Producción	196
4.12. Diagrama de Clases para el Módulo de Producción	198
4.13. Diagrama de Secuencia para el Módulo de Producción	202
4.14. Ejemplo de un objeto <code>CompositionOutput</code>	203
4.15. Arquitectura de Módulo de Síntesis para el Sistema	207
4.16. Arquitectura del Módulo de Grabación de apoyo al Sistema	209
5.1. Arquitectura de Representación del Conocimiento y Producción de Material Musical para la Implementación	219
6.1. Diagrama de cajas para los grupos de melodías evaluados	235
6.2. Porcentajes de aciertos para la prueba de Turing	237
6.3. Intención Emocional y Percepciones para la Pieza Musical 1	241
6.4. Intención Emocional y Percepciones para la Pieza Musical 2	241
6.5. Intención Emocional y Percepciones para la Pieza Musical 3	242
6.6. Intención Emocional y Percepciones para la Pieza Musical 4	242
6.7. Intención Emocional y Percepciones para la Pieza Musical 5	243
6.8. Intención Emocional y Percepciones para la Pieza Musical 6	243

6.9. Intención Emocional y Percepciones para la Pieza Musical 7	. . . 244
6.10. Intención Emocional y Percepciones para la Pieza Musical 8	. . . 244
6.11. Intención Emocional y Percepciones para la Pieza Musical 9	. . . 245
6.12. Intención Emocional y Percepciones para la Pieza Musical 10	. . . 245
6.13. Intención Emocional y Percepciones para la Pieza Musical 11	. . . 246
6.14. Intención Emocional y Percepciones para la Pieza Musical 12	. . . 246
6.15. Intención Emocional y Percepciones para la Pieza Musical 13	. . . 247
6.16. Intención Emocional y Percepciones para la Pieza Musical 14	. . . 247
6.17. Intención Emocional y Percepciones para la Pieza Musical 15	. . . 248
6.18. Tiempos Promedio de Procesamiento 251
6.19. Porcentaje de Utilización de CPU con 1 Hilo 252
6.20. Porcentaje de Utilización de CPU con 2 Hilos 253
6.21. Porcentaje de Utilización de CPU con 3 Hilos 253
6.22. Porcentaje de Utilización de CPU con 5 Hilos 253
6.23. Porcentaje de Utilización de CPU con 8 Hilos 253
6.24. Porcentaje de Utilización de CPU con 13 Hilos 254
6.25. Porcentaje de Utilización de CPU con 21 Hilos 254
6.26. Porcentaje de Utilización de CPU con 34 Hilos 254
6.27. Porcentaje de Utilización de CPU con 55 Hilos 254
6.28. Consumo de memoria por Hilos 255
6.29. Utilización de Memoria con 1 Hilo 256

6.30. Utilización de Memoria con 2 Hilos	256
6.31. Utilización de Memoria con 3 Hilos	256
6.32. Utilización de Memoria con 5 Hilos	257
6.33. Utilización de Memoria con 8 Hilos	257
6.34. Utilización de Memoria con 13 Hilos	257
6.35. Utilización de Memoria con 21 Hilos	257
6.36. Utilización de Memoria con 34 Hilos	257
6.37. Utilización de Memoria con 55 Hilos	258

ÍNDICE DE TABLAS

2.1. Notas musicales y sus números MIDI	46
2.2. Escala Cromática enumerada	76
2.3. Escala Cromática enumerada para obtener la escala F Mayor	77
3.1. Partes y emociones de la Tocata y fuga en re menor, BWV 565 de Johann Sebastian Bach	134
4.1. Especificación de Diseño de Prueba: Adquisición de Datos	211
4.2. Especificación de Diseño de Prueba: Manipulación de la Base de Conocimientos	212
4.3. Especificación de Diseño de Prueba: Sincronización de Tiempo	213
4.4. Especificación de Diseño de Prueba: Integración con el Motor de Síntesis	214
4.5. Especificación de Diseño de Prueba: Rendimiento	215
4.6. Especificación de Diseño de Prueba: Composición e Interpretación de Melodías Musicales	216
6.1. Resultados de la prueba de Shapiro-Wilk	234
6.2. Estadística descriptiva de los dos grupos	235
6.3. Resultados de la prueba de Shapiro-Wilk	236

6.4. Prueba de Levene para las varianzas de las emociones en cada	
pieza musical	240
6.5. Estadística descriptiva de los tiempos de ejecución en	
milisegundos por número de hilos	251

INTRODUCCIÓN

Las investigaciones relacionadas con la reproducción musical basada en representaciones abstractas, utilizando técnicas de Inteligencia Artificial, se han enfocado en tres aspectos básicos: composición, improvisación e interpretación expresiva, [1] y [2]. De estos, el aspecto más explorado ha sido la composición musical, como un proceso de “armar” o ensamblar varias partes del sonido de un mismo instrumento para lograr música polifónica, o de varios sonidos distintos para crear obras musicales; como lo define el Diccionario de Música Harvard [3].

Establecer normas generales válidas para la creación musical, aplicadas a cualquier ámbito, es prácticamente imposible; por esta razón, existen estrategias de composición para ámbitos específicos, que dependiendo del entorno sugieren normas de composición, permitiendo la aplicación de técnicas definidas en la Ingeniería de Conocimiento para estructurar estas normas y poder realizar sistemas basados en ese conocimiento adquirido.

Desde el punto de vista de la música, existen estándares para la transmisión de información musical. En *Lilienfeld1962* se toman en cuenta dos aspectos importantes sobre el conocimiento musical, la teoría musical y el análisis

musical, cuyas interacciones a lo largo del tiempo han generado el conocimiento que hoy es impartido en la academia. Tomando en cuenta otras áreas, Christopher Alexander [4], quien realizó un estudio exhaustivo de los procesos de diseño en las construcciones arquitectónicas, establece la utilización de patrones y un lenguaje que permite la interacción con esos patrones para alcanzar lo que él denomina la *cualidad sin nombre*, que en el ámbito musical sería una característica que los músicos buscan en sus creaciones y que en ciertos casos es difícil de comunicar más que en sus obras.

Las investigaciones relacionadas a composición musical [2] en su gran mayoría están enfocadas a la música occidental cuyo énfasis se centra en la música clásica y géneros estándares como el Jazz o el Blues, pero poco en la música experimental, donde según Holmes [5] existen innovadoras formas de expresar nuevas ideas para la composición produciéndose un proceso de exploración; donde más que comenzar con una partitura, se comienza con escuchar. El tipo de música experimental toma en consideración no sólo criterios elementales de armonía, melodía y ritmo, sino el tipo de sonido que se desee crear.

Las estrategias utilizadas en el ámbito de composición musical algorítmica han sido de gran interés para varias investigaciones como se detalla en [2] y [6], las cuáles son basadas en técnicas de Inteligencia Artificial

considerando que algunas de ellas utilizan la teoría lingüística para representar reglas de composición musical, e identificar patrones; tales como, frases melódicas, transiciones, arreglos de armonías, entre otros [2], de esta forma se brinda variabilidad a la salida del algoritmo de composición, y meta-conocimiento como criterio de evaluación para “guiar la composición” más que la “composición” en sí.

El modelo para la adquisición del conocimiento, desarrollado en esta investigación, permite crear una base de conocimientos que toma en consideración la experiencia humana de dos compositores músicos entrevistados, relacionada a la composición de música experimental y las técnicas de inteligencia artificial aplicadas al proceso creativo de los artistas, bajo el mismo esquema que utilizarían cuando crean este tipo de música. Sobre este modelo se aplicaron estrategias para poder componer e interpretar en tiempo real melodías musicales en un proceso que denominamos *composición Humano-Máquina* donde el músico real interpreta una base armónica que es seguida por el agente artificial mediante la reproducción de melodías musicales.

CAPÍTULO 1

1. GENERALIDADES

1.1 Planteamiento y Justificación del Proyecto de Investigación

1.1.1 Problema

La *música* es considerada un arte en la sociedad humana que ha sido desarrollada a lo largo de los siglos, la cual es un medio para comunicar emociones y provocar, algunas veces de manera intencional, reacciones profundas a una variedad de sentimientos, reforzando además eventos importantes en la vida de las personas. Con el advenimiento de la tecnología, los músicos buscan innovar en procesos de composición y en las obras que

se pudieran producir, como se lo intenta lograr por medio de la música experimental [5], de tal manera que el mensaje emocional detrás de cada pieza sea percibido con intensidad por parte de los oyentes.

Esta forma de arte ha sido estudiada desde el punto de vista cognitivo con énfasis en identificar modelos y procesos que aporten a la habilidad de composición para los artistas de esta rama. Es por ello que el problema presentado en este trabajo es acerca de la manera para representar el conocimiento musical y utilizarlo con el fin de que una plataforma tecnológica componga e interprete obras musicales en vivo. El alcance de la solución se limita únicamente a *melodías musicales* debido a la complejidad que demandaría incluir otros elementos tales como *armonías, ritmos y timbre*. Cabe mencionar que el tipo de música a tratar es la occidental, la cual es familiar en nuestra cultura.

Por la naturaleza de este problema, es adecuado un enfoque basado en técnicas de Inteligencia Artificial debido a que en el ámbito de melodías musicales podemos tener una explosión combinatorial con respecto a la mezcla de notas y tiempos musicales en secuencias que podrían no poseer una buena estética y por ende no producir emociones en los oyentes, es por ello que con estrategias adecuadas para este problema no convencional podemos lograr melodías con sentido y sin combinaciones innecesarias de

notas.

Por otro lado, el proyecto de investigación ANDAMIOS, que ejecuta el Centro de Tecnologías de Información, busca aportar con tecnologías al desarrollo más eficiente de los estudiantes de ingeniería ecuatorianos, para que puedan competir en la sociedad del conocimiento. Es así como la solución planteada aporta a la educación de los estudiantes de ingeniería con respecto a la interacción entre el área técnica y el área artística para la producción de elementos por parte de ingenieros hacia artistas musicales.

1.1.2 Justificación

Con el presente trabajo se espera entregar un nuevo aporte al área de investigación de la inteligencia artificial con respecto a la reproducción de actividades consideradas como creativas, en este caso la generación de la música.

Los músicos pueden hacer uso de un sistema implementado con estos lineamientos con el objetivo de incrementar su abanico de posibilidades e inspiración para la composición de nuevas piezas de su autoría considerando que son ellos mismos los generadores de esas ideas pero con otras perspectivas. La inmediata aplicación de un sistema como éste para los

músicos sería realizar interpretaciones en vivo donde el sistema los acompañe en la ejecución de una pieza improvisada.

Se busca contribuir en el proyecto ANDAMIOS con una perspectiva distinta para el desarrollo de soluciones tecnológicas que permitan mejorar las limitaciones existentes en el desarrollo de habilidades de los estudiantes de ingeniería, habilidades que sean aplicadas en este caso particular para tratar con problemas relacionados a áreas artísticas y humanísticas como la música. Estos productos podrán ser replicados en las instituciones de educación superior ecuatoriana que forman ingenieros a fin de preparar profesionales en capacidad de competir en la sociedad del conocimiento.

1.1.3 Solución Propuesta

La solución al problema planteado consiste en establecer representaciones y estrategias para la composición musical basadas en técnicas de inteligencia artificial. Estos elementos serán plasmados en la implementación de un piloto cuyo resultado debe ser aceptable estéticamente; y para ello, se someterá a evaluaciones con personas reales. El programa piloto que se desarrolle será un sistema concurrente en tiempo real que reproduzca melodías en conjunto con músicos humanos en base a las abstracciones y algoritmos propuestos, por lo tanto se tomará en cuenta su rendimiento para este proceso.

1.2 Objetivos de la Tesis

1.2.1 Objetivo General

Desarrollar un modelo de representación del conocimiento musical y soportarlo mediante una infraestructura tecnológica, que comprenda hardware y software para la solución del problema planteado, y así poder sintetizar de manera concurrente melodías musicales en tiempo real en conjunto con músicos humanos, cuya implementación utilice técnicas de inteligencia artificial y considere el carácter emocional que se desee transmitir a una audiencia, para finalmente evaluar la calidad estética producida por el sistema.

1.2.2 Objetivos Específicos

- Estudiar el estado del arte de las técnicas de inteligencia artificial aplicadas a la identificación de emociones de tal manera que se escojan las adecuadas para el desarrollo de la tecnología que soportará lo postulado en el presente trabajo, estas estrategias deben soportar mayoritariamente a las propiedades que puedan encontrarse en la composición e interpretación de las melodías musicales tanto como los

sentimientos que desean transmitir.

- Proponer las estructuras de datos que soportarán la representación del conocimiento a manipular por las técnicas de AI, cuyas implementaciones permitan construir funcionalidades altamente eficientes.
- Aplicar las tecnologías desarrolladas y sus estructuras propuestas en la implementación de un piloto que reciba las entradas de los músicos y demás parámetros que este software requiera para una alta efectividad.
- Evaluar la calidad de composición y transmisión de emociones a través de las melodías y sonidos generados por el sistema mediante lineamientos detallados de la experiencia auditiva que éste brinde a músicos y personas que no necesariamente son músicos.
- Proponer directrices para desarrollar soluciones que involucren el descubrimiento o reforzamiento de habilidades creativas en un estudiante de Ingeniería según los objetivos del proyecto ANDAMIOS.

1.3 Metodología

Inicialmente se hará una revisión de literatura que involucre los procesos mentales relacionados con la composición musical además de las normas

metódicas que rigen la teoría musical, luego se relacionará con las estrategias de inteligencia artificial utilizadas en el ámbito de la música para llegar a una decisión final acerca de cuál o cuáles técnicas serán utilizadas para desarrollar el software de la infraestructura tecnológica descrita en este trabajo. Luego se explorarán las estrategias que permitan considerar la representación de emociones con respecto a sonidos percibidos con el objetivo de sintetizarlos en las melodías generadas.

Una vez establecidas las bases teóricas se procederá a evaluar la utilización de cada una de las herramientas y tecnologías orientadas al desarrollo del software que soportará las técnicas de AI considerando un nivel alto de efectividad y eficiencia para la construcción de la aplicación en tiempo real, luego se procederá a la construcción de este sistema conjuntamente con la integración de un motor de síntesis de audio que permita una flexibilidad comparable a la de un lenguaje de programación, en este caso se hará uso de SuperCollider [7]. Finalmente se procederá a armar toda la infraestructura tecnológica que involucra hardware y software.

Para el ensamble de la infraestructura se utilizará el formato estándar de comunicación de dispositivos musicales MIDI para interactuar entre todas las partes desde las entradas hasta las salidas.

Luego, el sistema será entrenado por músicos profesionales principalmente

aquellos orientados a música experimental [8] cuyo aporte brindará un enfoque musical innovador a la base de conocimientos del sistema; además, de un criterio valioso para la síntesis de los sonidos por la naturaleza de este tipo de música.

Después de ser alimentado, el sistema podrá realizar interpretaciones musicales en tiempo real acompañando a músicos que entonen piezas distintas a las que se entrenó el sistema, de tal manera que se aprecie la generación de las melodías de manera concurrente con respecto a una temática musical que al o a los artistas se les ocurra en ese instante.

Cuando se habla de melodías producidas concurrentemente lo que sucedería es que al mismo tiempo se reproducirían dos o más secuencias de notas con distintos motivos musicales, es decir, sonidos sintetizados de manera distinta.

Las piezas musicales que genere el sistema serán evaluadas en su calidad estética e impacto emocional y creativo. Considerando que estos parámetros son subjetivos, antes de realizar estas evaluaciones se deben escoger los métodos adecuados para la recolección de este tipo de información.

Cuando se obtengan los resultados de las evaluaciones de las piezas musicales, éstos serán interpretados para concluir si la metodología utilizada en el desarrollo de la aplicación se acerca a la reproducción de esta actividad

creativa, como lo harían artistas reales; o si el resultado de la producción del sistema carece de naturalidad y de carácter humano.

Finalmente, se evaluará el proceso para el desarrollo del presente trabajo y se detallarán los lineamientos relevantes que estudiantes de ingeniería puedan incluir en su conjunto de habilidades para expandir su aporte a la sociedad del conocimiento aplicando destrezas en el área técnica sobre ramas artísticas y humanísticas. Estos lineamientos serán parte de las consideraciones para el desarrollo de las soluciones específicas en las que el proyecto ANDAMIOS se enfoca.

1.4 Estructura de Documento

En el capítulo dos se detallan las bases teóricas de este trabajo considerando fundamentos de música y la revisión de literatura relacionadas a las técnicas de Inteligencia Artificial utilizadas en el ámbito musical. En el capítulo tres se realiza un análisis, en base a la revisión hecha y a los criterios de músicos entrevistados, para establecer el modelo de representación del conocimiento musical y las estrategias a aplicar para la generación de melodías. En el capítulo cuatro se detalla el diseño de la arquitectura propuesta para la implementación de un sistema concurrente en tiempo real, que soporta el

modelo. En el capítulo cinco se presentan las consideraciones para la implementación del sistema diseñado, así como el entrenamiento y su ejecución. En el capítulo seis se muestran y se analizan los resultados de las pruebas realizadas con músicos y oyentes para la evaluación del modelo, así como pruebas de rendimiento con respecto al sistema. Finalmente, se muestran las conclusiones y recomendaciones del enfoque escogido para solucionar el problema propuesto.

CAPÍTULO 2

2. MARCO TEÓRICO

2.1 Teoría Musical Básica

A lo largo del tiempo se ha registrado el conocimiento musical que rige la cultura occidental de tal manera que actualmente se mantienen estándares, métodos de representación simbólica, estrategias de composición y demás elementos que son utilizados para generar material creativo con sonidos que es a lo que denominamos **música** cuya definición según [9] trata de la agrupación *rítmica, melódica, o armónica* de sonidos que son especialmente *compuestos* y que forman una unidad con el objetivo de transmitir un mensaje para comunicarlo, o para entretener.

Para comprender a plenitud la propuesta de este trabajo se han definido puntos claves de la teoría musical que son consideradas en el contexto de las soluciones establecidas con respecto a esta área artística, principalmente desde el punto de vista de la composición musical.

2.1.1 Notas Musicales y Escalas

Nota musical es la denominación que se le da al *tono* o *frecuencia fundamental* específica para representar sonidos en términos musicales. Las notas musicales son representadas por las siete primeras letras del alfabeto (A, B, C, D, E, F, G) cuyo orden es establecido desde *C* y pueden ser identificadas en un teclado musical como lo ilustra la figura 2.1, las cuales son asignadas a las teclas blancas [10].

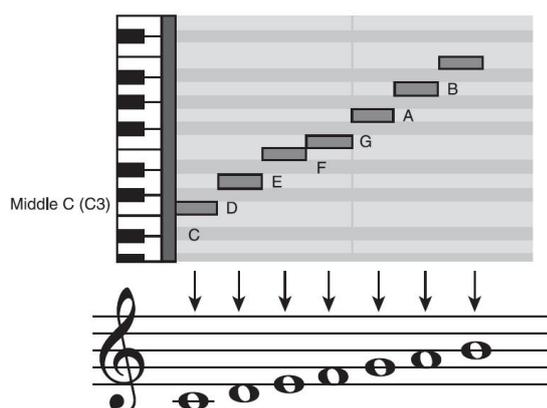


Figura 2.1: Notas musicales para las teclas blancas de un teclado musical

Fuente: Music Theory for Computer Musicians [10]

A estas notas musicales elementales es posible agregar alteraciones

conocidas como *bemoles* o *sostenidos*. Estas alteraciones son las teclas negras de un teclado musical cuya posición es establecida como lo muestra la figura 2.2. Estas alteraciones son denotadas por la nota elemental más el símbolo de la alteración; en el caso de los bemoles el símbolo es \flat , y para los sostenidos es \sharp . Las notas bemoles son las que se encuentran un semitono bajo una nota fundamental en particular, las notas sostenidas en cambio son las que se encuentran un semitono sobre la nota en mención; esta característica la podemos notar en la figura 2.2, si tomamos por ejemplo la nota D, que posee las alteraciones $D\flat$ y $D\sharp$. Nótese que existen notas con una misma posición pero con nombres distintos, por ejemplo $D\flat = C\sharp$, en ese caso se dice que son *equivalentes enarmónicos* [10].

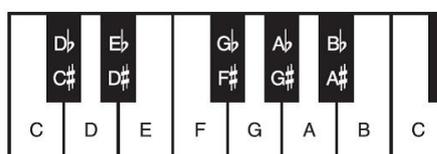


Figura 2.2: Notas musicales para las teclas negras de un teclado musical
Fuente: Music Theory for Computer Musicians [10]

Las notas musicales descritas no son restringidas a un tono fijo ya que el mismo conjunto puede ser asociado con sonidos más agudos o más graves; es decir, que la nota C puede tener distintas frecuencias dependiendo de la **octava** en que se encuentre. Una *octava* es un grupo compuesto por un intervalo de ocho notas fundamentales que se diferencia de otro grupo gracias a que los sonidos producidos son más agudos o más graves. Se dice

que cuando nos encontramos en una octava en particular, estamos en un *registro* específico. Las octavas son distribuidas en un teclado musical como se muestra en la figura 2.3. En ese caso si se quiere referir a una nota específica podemos utilizar una notación de letra, número y símbolo de alteración si es necesario. Por ejemplo el *C* central (middle C) es $C3$ [10].

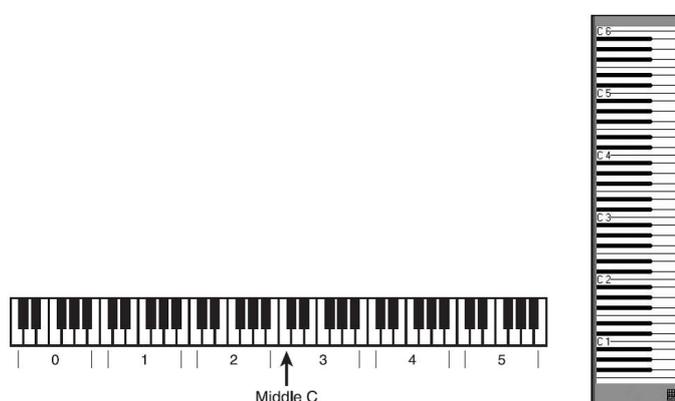


Figura 2.3: Notas musicales y sus octavas
Fuente: Music Theory for Computer Musicians [10]

Una **escala** es la ejecución de una serie de notas musicales, generalmente dentro del rango de una octava y repetido en las otras octavas. Puede ser utilizado en el contexto de *melodías* (ejecución sucesiva de notas que siguen una progresión lineal) o de *armonías* (ejecución simultánea de notas) [10].

Existe un gran número de escalas definidas, la más importante es la *escala cromática* que prácticamente se compone del conjunto de notas completo discutido iniciando en *C* de esta manera:

C - C# - D - D# - E - F - F# - G - G# - A - A# - B

A partir de ella pueden construirse otras escalas tales como: *major*, *minor*, *pentatonic*, *modal*, *octatonic*, *hexatonic* entre otras que pueden ser encontradas en [11]. La escala *C major* es la más conocida compuesta por las notas $C - D - E - F - G - A - B$. La construcción de escalas a nivel general es explicada en la sección 2.6.2.1.

Las escalas son denominadas dependiendo de su naturaleza (major, minor, etc) y su **nota clave** (key) que es la nota base para la construcción de una escala; es decir, su nota de inicio. Por ejemplo, ya conocemos que la escala *C major* esta compuesta por C, D, E, F, G, A, B y por ende su clave es *C*; entonces, según la explicación en 2.6.2.1, la escala *F major* sería $F_3, G_3, A_3, A_3\#, C_4, D_4, E_4$ si iniciamos en la octava tres. El conocimiento de estas escalas permite a los compositores crear música que tenga un claro sentido de armonía y coherencia [11].

2.1.2 Ritmo, Tempo, y Longitud de Notas

La **duración** es un parámetro del sonido musical. Cada sonido en composición musical tiene una particular duración o **longitud de nota**. Esta duración representa su valor en la dimensión del **tiempo musical**. La

sucesión de tales valores o eventos que ocurren a través del tiempo musical es llamado **ritmo** [10]. En este sentido, el parámetro de duración es de gran importancia porque provee los bloques de construcción para el ritmo. El ritmo es y siempre ha sido el ingrediente esencial de todo tipo de música. De hecho, sin ritmo, probablemente no habría ningún tipo de música. El ritmo también puede ser definido como un patrón de sonidos y silencios [10].

Un entendimiento de la naturaleza del ritmo comienza con la observación del mundo y nosotros mismos como parte de ello. Es inmediatamente obvio que el ritmo juega una parte vital en el mundo. Lo podemos encontrar en el palpitar del corazón, en la respiración, en nuestro caminar, etc.

El *tiempo musical* tiende a estar dividido en compases regulares de una duración consistente para producir algún ritmo en particular. Estos compases pueden ser relativamente lentos o rápidos, esta velocidad es conocida como **tempo**. El tempo es determinado por la duración del *compás*; es decir, el número de tiempos que ocurren por minuto, donde los tiempos son denominados *beats* y por ende la unidad para representarlos son los *beats por minuto (BPM)* [10].

La *longitud de la nota* determina cuan rápido o lento la música es percibida. También indica cuántos tiempos deben cubrir cada nota en el compás. De manera análoga a las notas, para cada tiempo existen los *silencios* que

también son denominados *descansos* los cuales son muy importantes porque representan los espacios entre eventos que dan forma a la composición [10]. En la figura 2.4 se muestran los tiempos correspondientes para notas y silencios en notación musical.

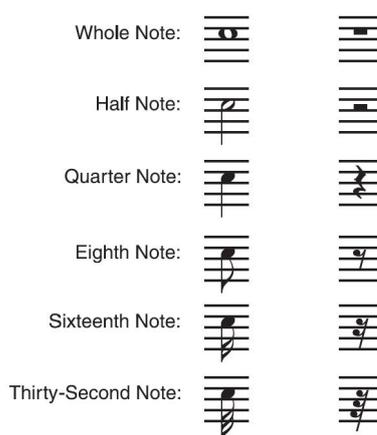


Figura 2.4: Tiempos para notas y silencios
Fuente: Music Theory for Computer Musicians [10]

2.1.3 Métrica

La **métrica** se preocupa por la manera en que los tiempos son combinados y estructurados dentro de grandes unidades o grupos con los cuales se puede generar música agradable al oído [10].

Estas unidades también conocidas como *ciclo métrico o medida*; están divididas o separadas por una línea vertical en la partitura llamada bar como se muestra en la figura 2.5. La métrica juega un rol importante para el ritmo

en la música, como una escala musical es a la melodía y armonía [10].



Figura 2.5: Uso de la línea de Bar en una partitura

Fuente: Music Theory for Computer Musicians [10]

Existen tres métricas básicas o ciclos métricos llamados: *compases binarios*, *compases ternarios* y *compases cuaternarios*. El ciclo métrico usado en una composición es indicado por un número fraccionario, por lo general va al inicio de la partitura. El numerador indica cuántos tiempos van en cada ciclo métrico o medida y el denominador indica la duración de la nota usada para cuantificar estos tiempos. Un ejemplo para un tiempo de 4/2 es el mostrado en la figura 2.5.

2.1.4 Acordes y Armonías

Un **acorde** es un elemento que proviene de la ejecución simultánea de un conjunto de notas [10]. Para poder referirnos a ellos a mayor profundidad es necesario conocer que es un *intervalo*.

Un **intervalo** es una relación entre dos notas en términos de su frecuencia fundamental de tal manera que el oído y el cerebro establecen asociaciones

congruentes [10]. La construcción y denominación de los intervalos es mostrado en la figura 2.6, cuyos elementos son intervalos de: *primera*, *segunda*, *tercera*, *cuarta*, *quinta*, *sexta*, *séptima* y *octava*.

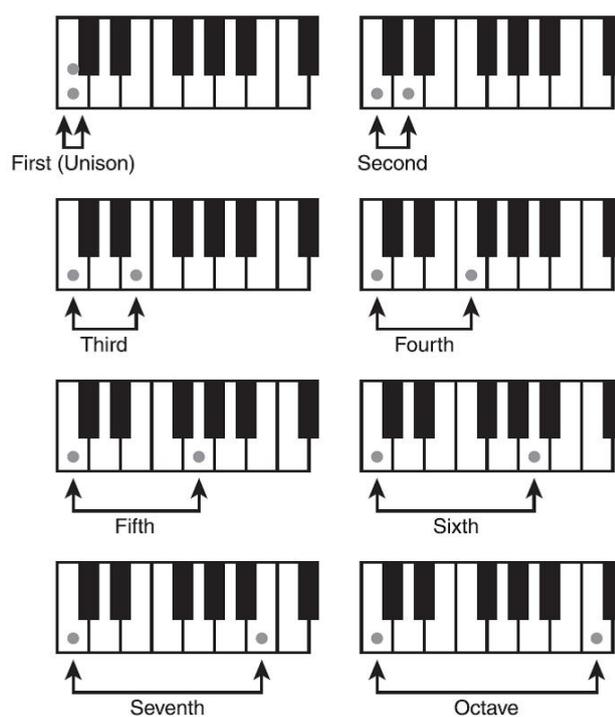


Figura 2.6: Intervalos de notas musicales
Fuente: Music Theory for Computer Musicians [10]

Si estos intervalos son ejecutados al mismo tiempo y la percepción de ellos es armoniosa, entonces son considerados *consonantes*; en cambio, si expresan cierta tensión se denominan *disonante* [10].

La **armonía** tiene que ver con estos elementos descritos que pueden ser combinados de tal manera que resulten en obras expresivas basadas en ejecuciones secuenciales de elementos que en sí pueden ser simultáneos; es

decir, una *progresión de acordes* [10].

El principio primordial para una buena progresión de acordes que tenga sentido para el oído es la **tonalidad** que es la cualidad que sobresale de la progresión gracias a un acorde llamado **tónico** que es el acorde raíz que le da el carácter a la progresión y es considerado como su *clave* [10]. El acorde tónico es respaldado por el acorde **dominante** que ocurre en el quinto grado de la escala. La transición del acorde dominante al tónico se denomina *cadencia* [10]. El acorde **subdominante** es el que se forma en la cuarto grado de la escala y se considera un contrabalance del dominante y un refuerzo para el tónico [10]. En la figura 2.7 se muestran los acordes: tónico, dominante y subdominante en la clave de C mayor (C).

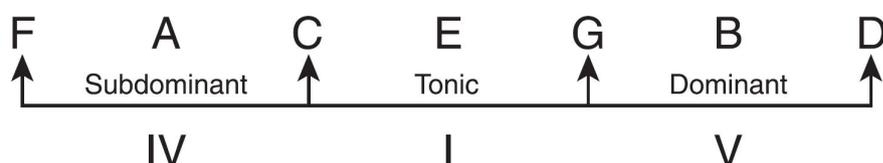


Figura 2.7: Acordes tónico, dominante y subdominante en la clave C mayor (C)
Fuente: Music Theory for Computer Musicians [10]

Distintas combinaciones de acordes relacionados a una clave pueden ser utilizados para una composición musical; por ejemplo, para C mayor (C) se podrían utilizar: D menor (Dm), E menor (Em), F mayor (F), G mayor (G), A menor (Am) o B disminuido (Bo) [10].

2.1.5 Melodías y Motivos

Una **melodía** es una secuencia lineal de notas que cambian en el tiempo y que son ajustadas a un ritmo de tal manera que se organizan para lograr una forma musical percibida como una unidad en la mente [10]. En una obra, la melodía puede ser dividida en *líneas melódicas* que pueden ser vistas como dos ejes distintos, tal como se ilustra en la figura 2.8. El eje vertical es el eje del tono (pitch) de la nota y el eje horizontal representa el tiempo durante el cual se presentan los cambios de tono.

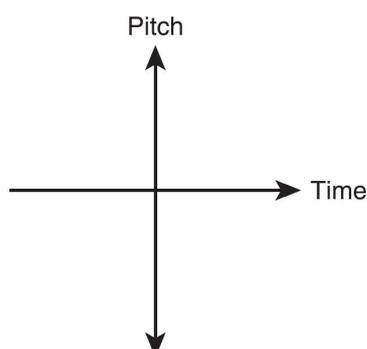


Figura 2.8: Ejes de tiempo (time) y tono (pitch) para una melodía
Fuente: Music Theory for Computer Musicians [10]

Esta sucesión de notas en el tiempo debe ser seleccionada por el compositor para un propósito expresivo, por lo tanto la construcción de ella no es solo una tarea aleatoria, y para ello las escalas pueden ser utilizadas como restrictoras en la combinación, de tal manera que haya congruencia con alguna base armónica que soporte la melodía [10].

Los **motivos** son fragmentos cortos de melodía que son altamente reconocibles debido a su ritmo inconfundible [10]. Si cualquiera de estos fragmentos es memorable para las personas, es porque ha sido construido a base a motivos. Un conjunto de motivos que constituye una línea melódica que es percibida como una unidad se denomina **frase** [10]. Por consiguiente, las frases melódicas influyen significativamente en el oyente de tal manera que perdure en su memoria y genere sensaciones únicas [10].

2.1.6 Composición Musical

En términos de composición musical, el artista debe considerar como bases principales los conceptos de *armonías, melodías y ritmos* para producir una obra con calidad estética, lo que implica que la pieza debe ser concebida como una *forma organizada*; es decir, los elementos involucrados deben funcionar como aquellos de los que un *organismo vivo* se constituye [12].

Sin una organización la música carece de propósito, es por eso que la composición debe ser lógica y coherente. La presentación, desarrollo e interconexión de las ideas deben estar basadas en relaciones y diferenciadas de acuerdo a su importancia y función.

El compositor construye la obra basado en una visión espontánea de toda la

composición; sin embargo, a pesar de ellos necesita considerar bloques musicales para su creación. Estos bloques musicales tales como motivos, frases melódicas, progresiones de acordes, etc., proveen el material para construir bloques más grandes de tal manera que desarrolle una estructura completa que cumpla con la visión espontánea idealizada [12].

Conocer principios perceptivos con respecto a los elementos estudiados es importante para llegar a deleitar al oyente. Por ejemplo, en el ámbito de motivos, una recomendación proveída en [10] es que, para crear un buen motivo hay que trabajar primero en el ritmo y luego en los intervalos de tal manera que se origine un contexto emocional que impacte en el oyente para que perdure en su memoria.

Existen métodos y estrategias de composición provistos por músicos a lo largo del tiempo que han servido de guías para nuevas generaciones de artistas. La particularidad esencial en estos casos es que estos músicos han logrado articular su conocimiento en procedimientos concretos basados en su experiencia [12]. Sin embargo, se podría decir que han habido músicos que han estado tan inmiscuidos en su proceso que les resulta complicado transmitir su conocimiento más que lo que reflejan sus obras. En ese caso es importante desarrollar mecanismos que puedan extraer ese conocimiento para que sea compartido por las nuevas generaciones y evitar que se

desvanezca en el tiempo [12].

2.1.7 Interpretación Musical

El producto de la composición en su forma tradicional es una **partitura** donde se reflejan las armonías, melodías y ritmos generados por el compositor. Este documento en sí es considerado otro lenguaje así como el lenguaje natural que utilizamos para comunicarnos diariamente [12].

La acción de ejecutar lo que describe una partitura sobre un instrumento musical es la **interpretación musical**, la cual está relacionada con la partitura en sí y el músico que la interpreta. La partitura provee un mínima guía acerca de las matices de expresividad que son necesarias para que la música cobre vida. Es allí donde son importantes las habilidades del intérprete para desenvolverse de la mejor manera en la ejecución [13].

La interpretación requiere del músico habilidades motrices excepcionales y un vasto conocimiento de la estructura musical y las tradiciones para lograr esa interpretación [13]; para ello, debe entrenar en el uso de temporización (aceleración y desaceleración), intensidad (acentos y crescendi), entonación y articulación [13].

En un proceso que demanda mayor carga cognitiva llamado **improvisación**,

el músico debe componer e interpretar en tiempo real, lo cual requiere de un cumplimiento inmediato de la visión espontánea que se menciona en [12] y una construcción de los bloques con la menor cantidad de errores posibles sin alejarse de esa visión.

2.2 Síntesis de Sonido

2.2.1 Definición

La síntesis de sonido es el proceso de producir sonido. Este proceso puede reutilizar sonidos existentes procesándolos, o puede generar sonidos electrónicamente o mecánicamente. Utiliza matemáticas, física o incluso biología; y junta arte con ciencia en una mezcla de habilidades musicales y técnicas [14].

Con el uso adecuado de la síntesis se pueden generar interpretaciones musicales con fuerte contenido emocional si se producen uno o varios *timbres* que refuercen esas emociones [14]. El **timbre** se conoce en términos musicales como el **color del sonido** o color del tono [10], que es la propiedad que permite al oído distinguir entre sonidos que ejecutan una misma nota; es decir, la nota *C* en un piano clásico se percibirá de manera distinta que en una flauta; esa diferencia es el timbre.

Por lo tanto, con los principios de síntesis de sonidos y sus elementos, es posible establecer una infinidad de timbres para reforzar las composiciones musicales.

2.2.2 Música Electroacústica

El estudio de las conversiones entre la energía eléctrica y la energía acústica (energía del sonido) es llamado **electroacústica**; por consiguiente, la música que es realizada por medio de instrumentos musicales electrónicos y otras técnicas electrónicas es denominada **música electroacústica** [14].

Este tipo de música implica un gran abanico de posibilidades creativas muy distintas a las que se podrían conseguir con los instrumentos tradicionales [5]. Es prácticamente componer no solo considerando los aspectos de armonía, melodía y ritmo; sino haciéndolo desde el punto de vista del timbre, el cual es generado por instrumentos que pueden *crear* timbre (sonido); para este caso, son claves los instrumentos eléctricos, electrónicos, y en esta nueva era, digitales, tales como los **sintetizadores** [5], de los cuáles se discutirá más adelante.

Debido a su vasta flexibilidad, la música electroacústica es ligada con la **música experimental** ya que ella provee innovadoras formas para expresar

nuevas ideas con respecto a composición que es en sí el proceso de experimentación en donde más que comenzar con una partitura, se inicia con *escuchar* [5]. Esta música llamada experimental ha sido posible y está siendo posible gracias al desarrollo de la tecnología con respecto a la síntesis.

2.2.3 Componentes Básicos de Síntesis

Los siguientes componentes descritos son las abstracciones básicas para la generación de sonidos que los sintetizadores utilizan, ya sea que su implementación sea mediante hardware o software. Estos componentes son utilizados como bases en las técnicas de síntesis que un músico podría ejecutar.

2.2.3.1 Osciladores y Wavetables

Un **oscilador** es una forma de onda repetida con una frecuencia fundamental y una amplitud de pico; es considerado como la base de las técnicas más populares de síntesis; además, La característica más importante es la *forma* [15]. En la figura 2.9 se encuentran las formas de ondas comunes en el dominio del tiempo tales como: *seno*, *cuadrada*, *triangular* y *diente de sierra*.

Generalmente, los osciladores son controlados por teclados musicales

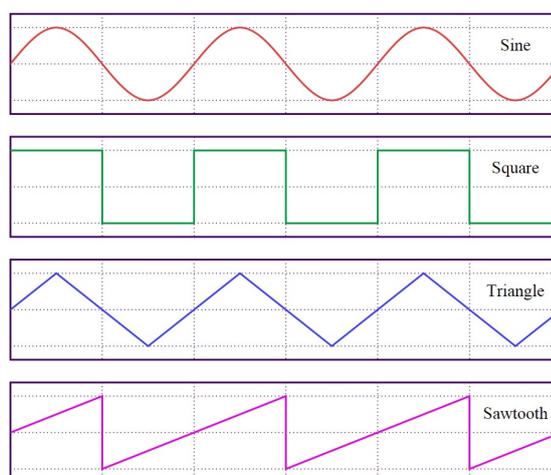


Figura 2.9: Ondas seno, cuadrada, triangular y diente de sierra
Fuente: Sound Synthesis Theory/Oscillators and Wavetables [15]

siguiendo el estándar MIDI (ver sección 2.3), donde cada nota es convertida a un valor de frecuencia dado en hertz (Hz) que afectará el período de la onda la cuál puede ser modificada por otros componentes para un sonido más enriquecedor.

Los **wavetables** o *tablas de onda*, son fundamentalmente arreglos de N valores cuyo rango de 1 a N representa un ciclo de un oscilador; es decir, con herramientas gráficas es posible crear una forma de onda que será repetida a lo largo del tiempo como en el caso de los osciladores discutidos anteriormente [15]. En la figura 2.10 se puede apreciar una onda personalizada sobre un wavetable; aquí podemos notar la existencia de un puntero llamado *read pointer*, el cual recorre el arreglo para obtener la salida de cada índice dependiendo de una velocidad dada. Cuando el puntero llega

al final el ciclo se reinicia [15].

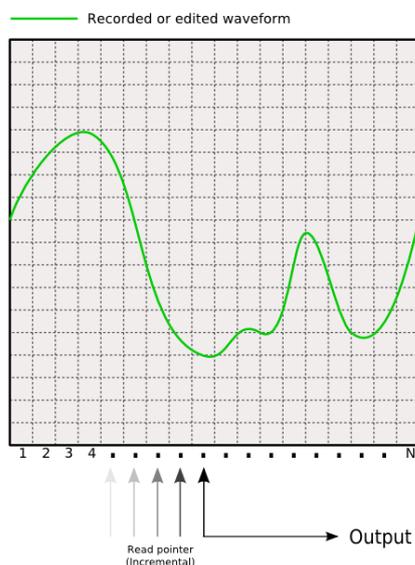


Figura 2.10: Estructura básica de un Wavetable

Fuente: Sound Synthesis Theory/Oscillators and Wavetables [15]

Los wavetables tienen ventaja sobre los osciladores en términos de eficiencia ya que es menos costoso recorrer un arreglo de elementos para obtener cada muestra de la señal, que aplicar la fórmula matemática para un oscilador básico tal como el sinusoidal, debido al costo de funciones senos y cosenos [15]. Además, son una herramienta poderosa en términos creativos por su flexibilidad de cambiar directamente la onda básica sonora.

2.2.3.2 Filtros

Los filtros son elementos que permiten alterar la señal producida por los osciladores o cualquier onda sonido sobre la cual actúen. Como su nombre lo

indica, ellos *filtran* el sonido de tal manera que alteran su naturaleza [16]. Esto sucede de manera común con alteraciones en rangos de frecuencia cuyo control principal es la *frecuencia de corte* (cutoff), el cual es un indicador del punto donde el filtro debe actuar. Los tipos comunes de filtros son [16]:

Low-Pass: Filtro Paso-Bajo, permite el paso de todas las frecuencias que se encuentran bajo la frecuencia de corte y atenúa las que están sobre ella.

High-Pass: Filtro Paso-Alto, permite el paso de todas las frecuencias que se encuentran sobre la frecuencia de corte y atenúa las que están bajo ella.

Band-Pass: Filtro Paso-Banda, permite el paso de todas las frecuencias que se encuentran en un rango que toma de referencia el centro de todo el dominio de frecuencias y atenúa las que están fuera de ese rango.

Band-Notch/Reject: Filtro Rechazo-Banda, permite que todas las frecuencias que se encuentran en un rango que toma de referencia el centro de todo el dominio de frecuencias sea atenuado, y deja pasar las que están fuera de ese rango.

En la figura 2.11 se ilustra como trabajan estos filtros de manera gráfica sobre las frecuencias de una señal de audio. Una característica importante de este tipo de filtros es la *resonancia*, la cual impulsa la frecuencia en la que el valor de frecuencia de corte está establecido, y por ende permite que alrededor de ella se impulsen un conjunto de frecuencias en el filtrado que puede ser mayor mientras la resonancia aumenta [17].

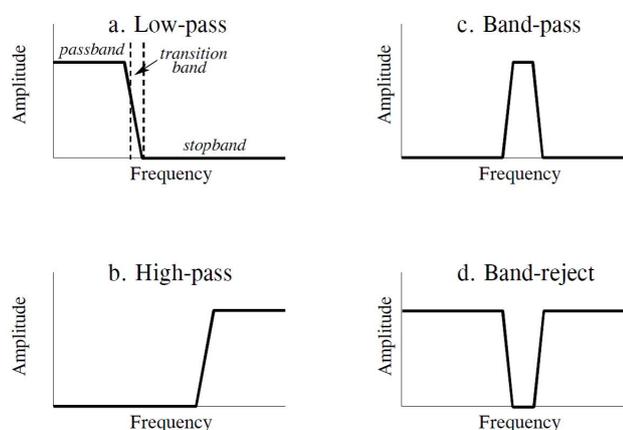


Figura 2.11: Filtros: (a) Paso-Bajo, (b) Paso-Alto, (c) Paso-Banda, (d) Rechazo-Banda

Fuente: The scientist and engineer's guide to digital signal processing [17]

2.2.3.3 Envoltentes y Enrutamiento

Un **envolvente** (envelope) es un elemento que brinda la "forma" del volumen de un sonido a través del tiempo; es decir, modifica la amplitud de la señal de audio en un período determinado siguiendo una función [14].

Los envoltentes son divididos generalmente en cuatro partes como se ilustra

en la figura 2.12. El **ataque** (attack) que va del tiempo que proviene del silencio al punto inicial más alto, el **decaimiento** (decay) es el tiempo en que el envolvente decrece hasta llegar a un valor estable, el **sostenimiento** (sustain) es el tiempo en el que se mantiene un valor estable proveniente del decaimiento, y finalmente la **relajación** (release) que es el tiempo en el que el envolvente decae al silencio [14].

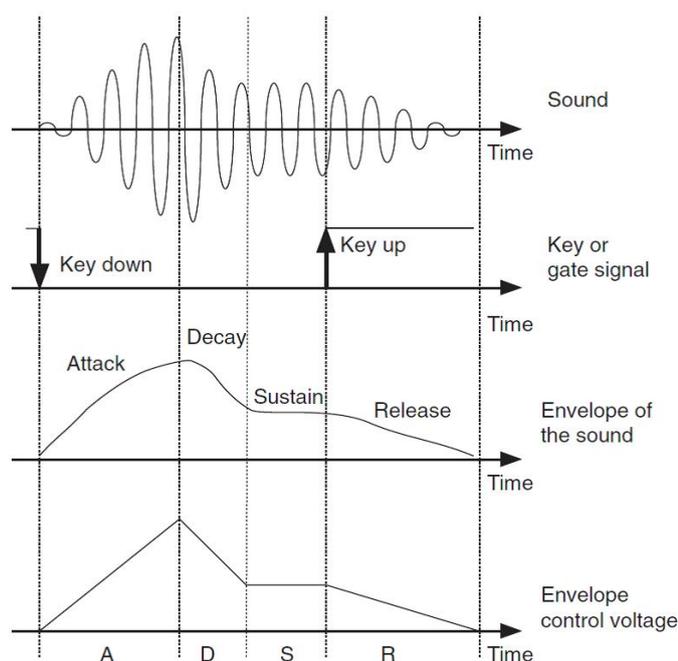


Figura 2.12: Envolvente ADSR y sus segmentos

Fuente: Sound Synthesis and Sampling [14]

Los envolventes que poseen las cuatro partes mencionadas se denominan *envolventes ADSR* por las iniciales en inglés de cada parte; en base a ello, existen otras combinaciones más sencillas como los AR (attack release), AD (attack decay), ADR (attack decay release) y ADS (attack decay sustain) [14].

Los envolventes no necesariamente son utilizados para control de volumen, también pueden controlar la frecuencia de corte de un filtro o la frecuencia de un oscilador. Son parte importante de un proceso de conexión llamado **enrutamiento** (routing), en el cual se conectan varios elementos como osciladores, filtros y envolventes, para obtener un determinado tipo de señal que puede ser controlada por esos envolventes en procesos intermedios de síntesis o en la salida final [18].

2.2.4 El Sintetizador

En términos de sonido, un sintetizador es conocido como un instrumento electrónico capaz de producir una amplia variedad de sonidos [14]. En él se encuentran implementado los elementos que discutidos anteriormente tales como osciladores, filtros y envolventes, además de otros componentes que permiten un interacción flexible y fluida con el artista.

Existe una gran variedad de sintetizadores, desde equipos sencillos hasta complejas estaciones de producción sonora como el equipo mostrado en la figura 2.13. A pesar de ello es posible clasificarlos según su construcción como: *sintetizadores de producción* y *sintetizadores modulares*. Los *sintetizadores de producción* poseen conexiones estándares y un flujo establecido de componentes, pero su flexibilidad es limitada; son los que

ocupan el mayor espacio en el mercado. Los *sintetizadores modulares* en cambio no tienen conexiones fijas de componentes y por ende brindan mayor flexibilidad en las combinaciones; sin embargo poseen mayor complejidad en la utilización [14]. El sintetizador presentado en la figura 2.13 es un sintetizador modular, los cuales son fácilmente identificables por conexiones manuales mediante cables colgantes entre los módulos del equipo.



Figura 2.13: Sintetizador Moog

Fuente: Moog Synthesizer - Viewing Gallery [19]

Los sintetizadores son generalmente controlados por un teclado musical; sin embargo no es un dispositivo estricto ya que otros controladores como perillas, botones, touchpads u otras interfaces pueden ser utilizadas como controladores [14]. Además, es necesario reducir en lo posible la complejidad de la presentación de los componentes hacia el usuario, de tal manera que su uso sea lo más sencillo posible; es por ello que las interfaces entre el usuario

y el sintetizador utilizan modelos basados en abstracciones que utilizan metáforas para que exista una interacción fluida [14], como se muestra en 2.14.

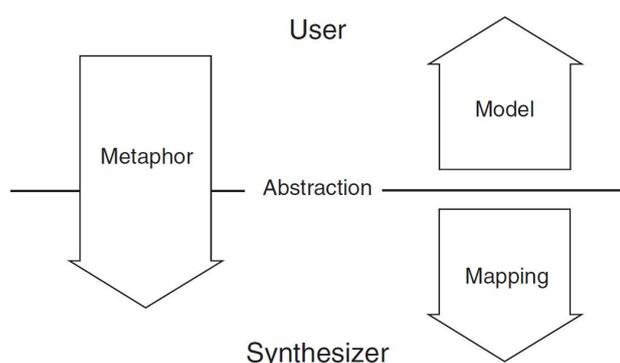


Figura 2.14: Modelo de interacción entre un usuario y un sintetizador

Fuente: Sound Synthesis and Sampling [14]

Estos equipos desarrollados a través de casi un siglo son representados en esta nueva era no solo de manera física, si no por medio de software de computadora, permitiendo la exploración de nuevas posibilidades creativas con elementos de mayor complejidad.

2.2.5 Modelamiento de Sonido

La cantidad de posibilidades de generación de sonidos en un sintetizador es sumamente grande considerando la utilización de los elementos atómicos que podrían interactuar (osciladores, filtros, envolventes, etc) [14]; por ello, se han desarrollado métodos para modelar los sonidos de una manera organizada y

coherente.

Los métodos básicos de modelamiento de sonido son: la **síntesis sustractiva** y la **síntesis aditiva**. La síntesis sustractiva toma como materia prima sonidos ricos en contenido armónicos que generalmente están dados por los osciladores básicos de onda seno, coseno, cuadrada, triangular y diente de sierra. Estos sonidos entran a un proceso de filtrado de frecuencias para luego ser manipulados por procesos de modulación (alguna otra señal que afecte a la producida) o efectos tales como la utilización de envolventes y finalmente producir la salida [14]. En la figura 2.15 se muestra de manera gráfica este proceso.

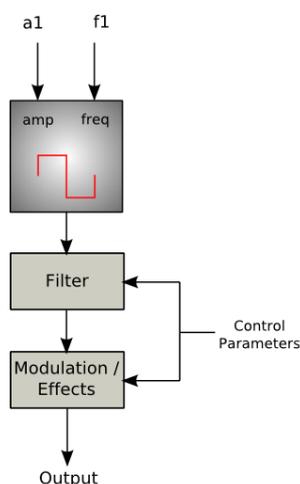


Figura 2.15: Proceso de síntesis sustractiva

Fuente: Sound Synthesis Theory/Subtractive Synthesis [20]

La síntesis aditiva permite generar sonidos mediante la suma de un gran número de ondas seno con diferentes frecuencias de tal manera que se

produzca un timbre final [14]. El inconveniente con este método es el grado de complejidad para controlar todas las ondas. Un esquema de este proceso se muestra en 2.16.

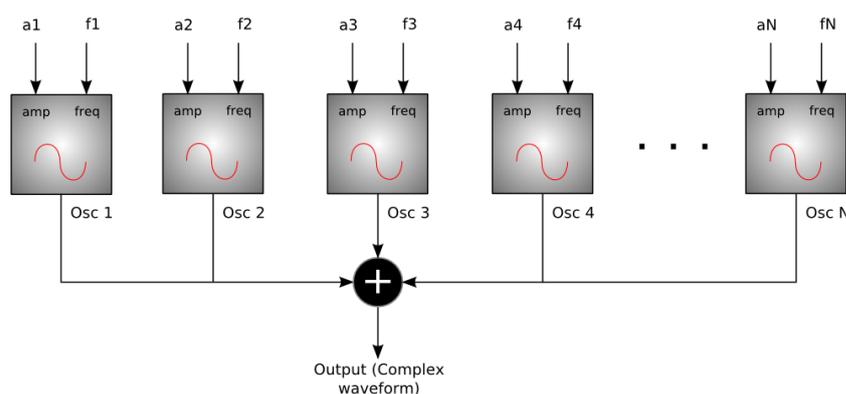


Figura 2.16: Proceso de síntesis aditiva

Fuente: Sound Synthesis Theory/Additive Synthesis [21]

En la sección 2.2.3.1 se presentó al *wavetable*, el cual puede ser utilizado como una extensión de la síntesis sustractiva para generar ondas sobre las cuales aplicar el método.

Un método importante de modelamiento de sonido en el ámbito de los sintetizadores es el **modelamiento físico**, que utiliza ecuaciones matemáticas que describen cómo un instrumento musical físico funciona [14]. En este método la característica más relevante es la manera en la cual el modelo responde en comparación al instrumento real y por lo tanto es importante considerar las técnicas de utilización de estos instrumentos por los intérpretes [14].

2.2.6 Síntesis mediante software

Los elementos, principios y métodos explicados han sido plasmados a lo largo del tiempo en equipos físicos haciendo uso de la electrónica analógica. Sin embargo, con la venida del computador de propósito general y las estrategias de procesamiento digital de señales, ha sido posible extrapolar la esencia de los sintetizadores análogos a sus homólogos digitales mediante software de computadora.

Ahora es posible instalar ambientes de producción con sintetizadores digitales a un menor costo y con mayor cantidad de funciones de tal manera que se pueden tener interfaces que permiten una gran flexibilidad a la hora de manejar sonidos [14]. Un ejemplo es la interfaz del plugin *Hybrid* ilustrado en la figura 2.17, en la cual podemos apreciar a simple vista que poseen tres osciladores configurables, filtros, envolventes para filtros, y otras funcionalidades para la generación de sonido [22].

Una característica destacable de este tipo de enfoque mediante software es la posibilidad de almacenar las configuraciones que permitieron generar un sonido en particular, de tal manera que ese sonido puede ser cargado nuevamente cuando se lo necesite [14]. Esta funcionalidad no es posible manejarla de una manera sencilla en un sintetizador analógico ya que



Figura 2.17: Plugin de síntesis de sonido Hybrid
Fuente: AIR Music Technology - Hybrid 3 [22]

depende de anotaciones de las posiciones de perillas, botones y demás controles la cuál no es precisa, y debido a la sensibilidad de algunos controles es casi imposible volver a generar con exactitud el mismo sonido [14]. Esa es la razón por la cual los artistas que utilizan estos equipos no generan exactamente el mismo sonido en vivo comparado con sus grabaciones si utilizan sintetizadores análogos.

2.3 MIDI (Musical Instrument Digital Interface)

2.3.1 Definición

MIDI es un acrónimo de **Musical Instrument Digital Interface** (*Interface Digital para Instrumentos Musicales*). Es un estándar que permite la

comunicación entre instrumentos musicales electrónicos y computadores mediante el envío y recepción de instrucciones de manera serial [23]. Por lo tanto, no se debe confundir al MIDI con los sonidos que viajan a través de cables MIDI.

El estándar MIDI permite realizar las siguientes acciones:

- El usuario puede utilizar el instrumento MIDI que mejor se ajuste a sus necesidades para generar sonidos desde otros dispositivos MIDI, incluyendo software para síntesis.
- Crear ricas texturas musicales por medio de la utilización de capas de sonidos provenientes de varios dispositivos MIDI, o asignar sonidos distintos con frecuencias distintas.
- Utilizar secuenciadores MIDI, los cuales capturan los datos de instrumentos MIDI para grabar la interpretación de músicos. Además, es posible modificar esos datos con respecto a tempo, nota, volumen, sonidos; es decir, que la interpretación pueda repararse en el caso de haber errores.
- Los bancos de sonidos MIDI permiten reproducir música pre-grabada de manera instantánea en varios dispositivos MIDI y en una computadora.

- Un dispositivo MIDI puede controlar a otro, permitiendo flexibilidad en tareas de producción musical. Por ejemplo, operar grabadores de audio desde un controlador MIDI compuesto de perillas y deslizadores, o también *sintetizadores* que acepten el control por estos tipos de dispositivos.

2.3.2 Componentes de Hardware

En esencia, los dispositivos físicos que se ajustan al estándar MIDI envían y/o reciben información [23]. Estos dispositivos pueden realizar distintas tareas dependiendo de su función. Para que esta comunicación sea posible desde el punto de vista físico, es necesario establecer conexiones entre los dispositivos; por lo cual, existen cables especiales (DIN de 5 pines) para estas conexiones que son conectados a los equipos mediante puertos como los mostrados en la figura 2.18, y son de tres tipos:

MIDI IN, el cual recibe datos de otro dispositivo MIDI. **MIDI OUT**, que envía datos MIDI producidos por el dispositivo hacia otro dispositivo MIDI. **MIDI THRU**, que envía datos MIDI que el mismo dispositivo recibe por el puerto MIDI IN hacia el exterior como lo hace MIDI OUT, solo que en este caso los datos no son alterados por el dispositivo [23].

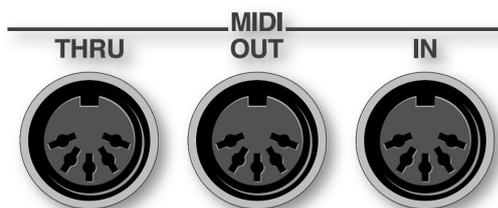


Figura 2.18: Puertos MIDI THRU, MIDI OUT y MIDI IN

Fuente: An Introduction to MIDI [23]

A pesar de la existencia de estos puertos, es posible encontrar dispositivos MIDI que se comuniquen a través de puertos USB, en ese caso pueden enviar y recibir datos por el mismo cable; generalmente su interacción se da en conjunto con un computador [23].

Nótese que gracias a los puertos explicados anteriormente es posible realizar cadenas de dispositivos conectados entre sí. Sin embargo, por el hecho de que la transmisión de datos es serial, pueden existir retardos que afecten al proceso creativo de un artista debido al bajo procesamiento de los dispositivos o a cadenas extensas que degradan la señal a medida que viajan por cada uno de ellos [23].

Dentro de la variedad de dispositivos MIDI se encuentran los **controladores MIDI**, que son equipos que ejercen control sobre otros equipos de hardware o software a través de mensajes MIDI [23]. El controlador mayormente utilizado es el teclado musical como el mostrado en la figura 2.19. Sin embargo, existen otros tipos de controladores basados en instrumentos musicales

tradicionales tales como guitarras, flautas, instrumentos de percusión, etc. y otros que son dispositivos únicos que brindan experiencias totalmente distintas para el usuario, como por ejemplo el *Linnstrument*, ilustrado en la figura 2.20, que se compone de un panel con una matriz de superficies táctiles interactivas (touchpads) con la posibilidad de controlar timbre, tono y velocidad de nota [24].



Figura 2.19: Controlador MIDI en forma de teclado musical
Fuente: MAX49 : Akai Professional Keyboard MIDI Controller [25]

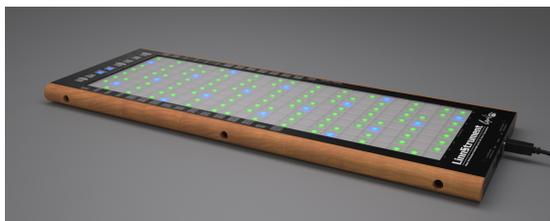


Figura 2.20: Controlador MIDI Linnstrument
Fuente: Linnstrument [24]

Siempre y cuando se siga el estándar MIDI de comunicación, cualquier dispositivo físico puede ser candidato a ser un controlador MIDI, sobretodo gracias a la posibilidad de usar puertos de comunicación comunes como el USB; por ejemplo, un joystick de una consola de videojuegos puede ser

programado como dispositivo MIDI de tal manera que las palancas de movimiento (sticks) controlen la frecuencia de un sonido en particular y los botones ritmos o notas musicales puntuales.

2.3.3 Componentes de Software

A nivel abstracto, los dispositivos MIDI se comunican a través de la transmisión de datos en forma serial; es decir, uno después de otro. Los tramos de datos son dominados **mensajes MIDI** (MIDI messages) y se componen de bytes que pueden ser de dos tipos: *bytes de estado* (status byte) y *bytes de datos* (data bytes) [26]. Los bytes de estado informan el tipo de mensaje que se va a enviar y su bit más significativo (primer bit de izquierda a derecha) inicia con uno (1), mientras que los bytes de datos inician con cero (0) y representan valores de 0 a 127.

Existen dos tipos de mensajes MIDI; **mensajes de canal** (channel messages) y **mensajes del sistema** (system messages). Los mensajes de canal son los encargados en transportar datos con respecto a la manipulación de controles en un dispositivo; es decir, cuando una tecla en un piano MIDI es presionada, se envía un mensaje MIDI que se compone de tres bytes como se ilustra en la figura 2.21, donde de izquierda a derecha, el primer byte es un byte de estado que informa que el mensaje es `note on` (nota presionada) dado por

los 4 bits más significativos seguido por `MIDI CH` que es el código del canal MIDI utilizado¹; los siguientes dos bytes son bytes de datos que poseen el número de la nota (C3 por ejemplo) y la velocidad con la que es ejecutada, respectivamente [26].

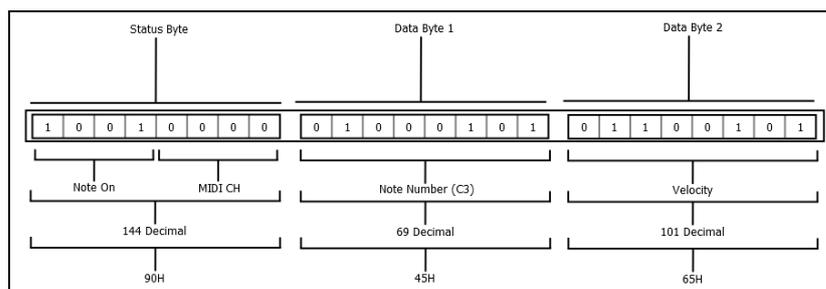


Figura 2.21: Ejemplo de un mensaje MIDI
Fuente: MIDI-protocol - 03-MIDI Message [26]

Los mensajes del sistema transportan datos relacionados a actividades de control entre los dispositivos como por ejemplo la sincronización de las señales en el proceso de comunicación [26].

Dentro de este esquema es necesario conocer que existe una asociación estándar entre las notas musicales y su representación en datos MIDI. Esta asociación se da mediante números enteros como lo muestra la tabla 2.1, en la cual se puede apreciar la organización de estos números con respecto a notas y sus octavas. Adicionalmente, en la figura 2.22 se muestra un modelo completo que describe en un teclado musical la asociación de notas, números

¹El canal MIDI de un dispositivo es aquel que determina a qué parte del dispositivo corresponde el control del cuál se está haciendo referencia en un mensaje MIDI; por ejemplo, las teclas de un piano tienen un canal distinto a los botones adicionales que podrían haber en el mismo dispositivo. También esta denominación es utilizada para diferenciar entre varios dispositivos de una cadena de conexión.

MIDI y frecuencias dadas en Hertz (Hz).

Octavas	Números de Notas											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
-1	0	1	2	3	4	5	6	7	8	9	10	11
0	12	13	14	15	16	17	18	19	20	21	22	23
1	24	25	26	27	28	29	30	31	32	33	34	35
2	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59
4	60	61	62	63	64	65	66	67	68	69	70	71
5	72	73	74	75	76	77	78	79	80	81	82	83
6	84	85	86	87	88	89	90	91	92	93	94	95
7	96	97	98	99	100	101	102	103	104	105	106	107
8	108	109	110	111	112	113	114	115	116	117	118	119
9	120	121	122	123	124	125	126	127				

Tabla 2.1: Notas musicales y sus números MIDI

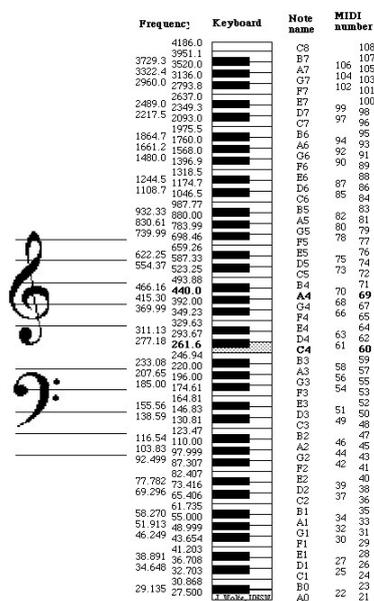


Figura 2.22: Notas musicales, números MIDI y frecuencias
Fuente: Note names, MIDI numbers and frequencies [27]

El tipo de software utilizado que permite gestionar estos mensajes MIDI para producción de audio es llamado **secuenciador**. Un secuenciador MIDI

permite capturar los mensajes MIDI de tal manera que es posible grabarlos y estructurarlos en abstracciones basadas en la interpretación de música en una línea temporal; además es posible la reproducción de esos mensajes en tiempo real; es decir que una interpretación musical en un teclado sería posible escucharla mientras se la está ejecutando [23]. En la figura 2.23 se muestra un ejemplo de un secuenciador con una vista de piano-roll donde es posible apreciar que a lo largo de la línea de tiempo se han ejecutado notas correspondientes a ciertas teclas.

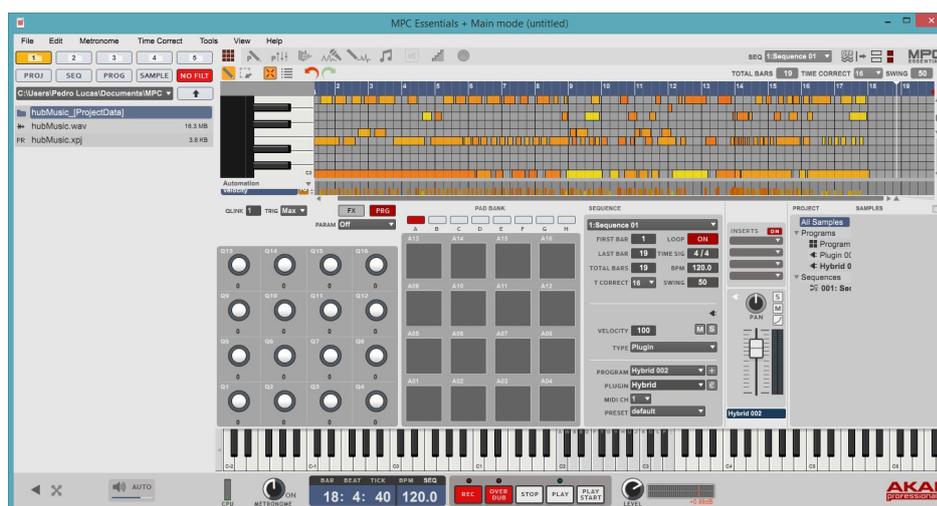


Figura 2.23: MPC Essentials: Secuenciador MIDI de Akai
Fuente: MPC Software : Akai Professional MIDI Secuencer [28]

La variedad en software basado en el estándar MIDI no se limita solo a secuenciadores, debido a que es posible utilizar APIs que gestionen mensajes MIDI en lenguajes de programación para realizar aplicaciones que requieran la interacción de equipos musicales.

2.4 Lenguajes de Programación de Síntesis en Tiempo Real

2.4.1 Descripción

Los lenguajes de programación de síntesis en tiempo real son un tipo de lenguaje de programación dinámico; es decir, un lenguaje de alto nivel que ejecuta en tiempo real tareas que los lenguajes estáticos desarrollan en tiempo de compilación [29]. Suelen ser interpretados y poseen en su núcleo un **motor de síntesis** el cual permite la interacción con el hardware de audio de la máquina y las operaciones de procesamiento de señales necesarias para generar sonidos.

Las características importantes de estos tipos de lenguajes son las relacionadas al ámbito de producción de música y audio en general; es decir, la reproducción de eventos musicales en tiempos específicos. Es por ello que su mayor preocupación es el control sobre el tiempo de los eventos, además de parámetros tales como frecuencias, amplitud, propiedades de filtros, entre otros [30]. La manera en que trabajan generalmente es a partir de módulos especializados que reciben los parámetros brindados por el programador para calcular el audio.

Además de nociones tales como el tiempo, la concurrencia es un factor importante ya que la música u otros ámbitos sonoros a menudo son producidos por eventos paralelos y por consiguiente estos lenguajes establecen paradigmas para programar de manera concurrente sin utilizar estrategias complejas de sincronización tales como mutexes, semáforos, variables de condición y canales [30].

Existen varios de estos lenguajes con enfoque gráfico como *Pure Data* y *MAX*; o de texto tales como *Supercollider* y *Chuck*. En este trabajo nos enfocaremos en la utilización de **Supercollider**.

2.4.2 El lenguaje SuperCollider

Supercollider es un lenguaje de programación para síntesis de audio en tiempo real que es multiplataforma (OS X/Linux/Windows) y se comunica vía *Open Sound Control* (OSC)² con uno o más servidores de síntesis [7]. La estructura del lenguaje es producto de una combinación de la estructura orientada a objetos de *Smalltalk* y características de lenguajes funcionales con sintaxis de la familia de *C*.

Está concebido como una combinación de dos componentes principales: un

²Open Sound Control (OSC) es un protocolo de comunicación entre computadores, sintetizadores de sonido, y otros dispositivos multimedia que es optimizado para la tecnología de redes moderna [31].

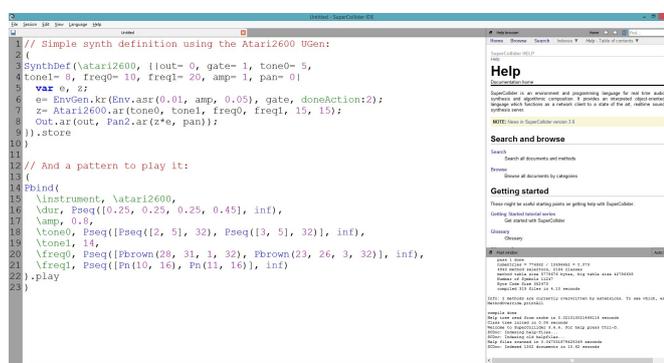
servidor (*scsynth*) y un *cliente* (*sclang*) que se comunican a través de mensajes OSC [7].

El componente **scsynth** es el motor de síntesis que provee configuraciones multicanales de audio, ya sean entradas o salidas; organiza los módulos de síntesis en una estructura de árbol la cuál puede ser modificada para alterar ese orden dependiendo de qué modulo se desee ejecutar con respecto a otros; permite reestructuración dinámica del flujo de sonido mediante buses lógicos del sistema; brinda buffers de audio para lectura y escritura; calcula a diferentes velocidades dependiendo de la necesidad para regular el procesamiento de la CPU. Además, posee varios *Unit Generators*, que son módulos considerados como bloques de construcción para el diseño de síntesis y de algoritmos de procesamiento de señales [7].

El componente **sclang** es el lenguaje de programación en sí, que es dinámicamente tipado, utiliza garbage collection, y posee un enfoque orientado a objetos desde el punto de vista de *funciones* que son implementadas utilizando un paradigma de los lenguajes funcionales; y por ende, estas funciones pueden ser compuestas por otras funciones. Brinda co-rutinas, listas, utilización de patrones mediante arreglos y la posibilidad de extender el lenguaje mediante plugins implementados en C++ [7].

El IDE predeterminado utilizado por Supercollider es el mostrado en la figura

2.24. Además, se encuentra un ejemplo de un código que define un módulo de síntesis (`SynthDef`) que produce un sonido semejante a los encontrados en juegos antiguos de la consola Atari, cuya composición es basada en funciones de envolventes (`Env`) y de panoramización (`Pan2`); para luego reproducir ese sonido sobre un patrón de números que brindan las notas (en formato MIDI) y los tiempos para generar una melodía con el sonido definido **7**.



```

1 // Simple synth definition using the Atari2600 UGen:
2 {
3   SynthDef(\atari2600, {out= 0, gate= 1, tone0= 5,
4     tone1= 8, freq0= 10, freq1= 20, amp= 1, pan= 0 |
5     var e, z;
6     e= EnvGen.kr(Env.asr(0.01, amp, 0.05), gate, doneAction:2);
7     z= Atari2600.ar(tone0, tone1, freq0, freq1, 15, 15);
8     Out.ar(out, Pan2.ar(z*e, pan));
9   }.store
10 }
11
12 // And a pattern to play it:
13 {
14   Pbind(
15     \instrument, \atari2600,
16     \dur, Pseq([0.25, 0.25, 0.25, 0.45], inf),
17     \amp, 0.8,
18     \tone0, Pseq([Pseq([2, 5], 32), Pseq([3, 5], 32)], inf),
19     \tone1, 14,
20     \freq0, Pseq([Pbrown(28, 31, 1, 32), Pbrown(23, 26, 3, 32)], inf),
21     \freq1, Pseq([Pn(10, 16), Pn(11, 16)], inf)
22   ).play
23 }

```

Figura 2.24: IDE de programación para SuperCollider con código de ejemplo
Fuente: SuperCollider **7**

Este lenguaje fue escogido para este trabajo debido a su estabilidad y robustez en el ámbito de generación de sonido, ha sido revisado y actualizado desde 1996 y utilizado en varias investigaciones con respecto a las áreas de sonido, composición musical y programación de audio; además, existe documentación de soporte, libros y una comunidad que discute en foros los problemas y soluciones con respecto a otros lenguajes de síntesis; además, es un lenguaje de código abierto permitiendo una mayor flexibilidad para la integración con otras plataformas o para la modificación del mismo acorde a las necesidades requeridas **7**.

2.5 Sistemas Concurrentes en Tiempo Real

2.5.1 Definición

Un sistema en tiempo real es un sistema de computadora donde la exactitud del *comportamiento del sistema* depende no solo de los resultados lógicos calculados, sino también del *tiempo físico* en el cual estos resultados son producidos [32]. *Comportamiento del sistema* se refiere en este caso a las secuencia de salidas en el tiempo por parte del sistema.

Desde el punto de vista de la concurrencia, estos tipos de sistema producen los resultados mencionados de manera paralela dependiendo de las actividades a realizar y deben enfrentar los problemas tradicionales que presenta los sistemas que utilizan paralelismo tales como sincronización o memoria compartida [33].

2.5.2 Características Principales

En un sistema en tiempo real, el flujo de tiempo es modelado por una línea de tiempo que va desde el pasado hacia el futuro. Un punto específico en esta línea de tiempo es denominada *instante*. Una ocurrencia ideal que suceda

en un instante es denominada *evento*, cuya información que lo describa se llama *información de evento* [32]. Existe un evento especial llamado *actual*, el cual separa el pasado del futuro. Un *intervalo* en la línea de tiempo, llamado *duración*, es definido por dos eventos, el *evento de inicio* y el *evento finalizador* del intervalo.

Un sistema en tiempo real cambia en función del tiempo físico y es posible descomponerlo en un conjunto de subsistemas autocontenidos llamados *clusters*, como se muestra en la figura 2.25. El *cluster controlado* (controlled cluster), es el ente, físico o virtual, que será controlado; el *cluster computacional* (computational cluster) que el sistema computacional en tiempo real que hace las veces de controlador; y el *cluster del operador* (operator cluster), que es básicamente el subsistema que representa al usuario que va a interactuar con el sistema.

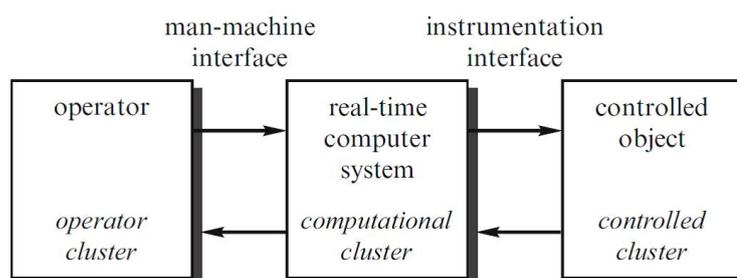


Figura 2.25: Sistema en Tiempo Real

Fuente: Real-Time Systems: Design Principles for Distributed Embedded Applications [32]

La comunicación entre los clusters es dada mediante interfaces. La interface

man-machine permite la interacción entre el usuario y el sistema computacional en tiempo real; la interface *intrumentation* es capaz de enlazar el sistema en tiempo real con el objeto controlado.

Un sistema en tiempo real debe reaccionar a los estímulos que el ambiente ejerce sobre él en intervalos de tiempo dictaminados por el ambiente en sí. El instante en el cual un resultado debe ser producido se denomina *deadline*. Si un resultado aún es válido después de que el *deadline* ha ocurrido, entonces es clasificado como *soft*, caso contrario es llamado *firm*.

Considerando la importancia del tiempo en estos tipos de sistemas, una característica que resalta es la existencia de *ciclos de control* (control loops) que son procedimientos que se están ejecutando constantemente dentro de un determinado tiempo. En estos procedimientos se sensea el estado de las variables y reaccionan de acuerdo a sus cambios de tal manera que las salidas se reflejen en el tiempo esperado que en muchos casos tiende a ser cero considerando que se desean resultados instantáneos. Además, estos procedimientos pueden estar ejecutándose de manera concurrente dependiendo de la aplicación con las respectivas medidas del caso para su interacción.

2.5.3 Consideraciones al momento de desarrollar este tipo de sistemas

Se espera que estos tipos de sistemas sean eficientes (mínimo consumo de recursos computacionales y tiempo) en la realización de sus tareas. Por lo tanto, estrategias que consuman una gran cantidad de ciclos de CPU producirán *latencia* en la respuesta a las entradas por parte del usuario [32]. Por ejemplo, en el caso de dispositivos MIDI, un sistema en tiempo real está sensando la recepción de mensajes en un computador; si la implementación de este sistema procesa los mensajes de una manera ineficiente o el hardware no es tan poderoso para soportar una respuesta que tienda a ser instantánea, entonces se percibirá un retraso en la respuesta de la máquina siendo algo no adecuado para interpretaciones musicales en tiempo real porque es prácticamente imposible llevar el ritmo musical si no se perciben las salidas en los tiempos adecuados.

Otras consideraciones son relacionadas al ámbito de la concurrencia. Los procesos concurrentes enfrentan retos tales como la *sincronización* y *memoria compartida* [32].

La *sincronización* trata de resolver los problemas relacionados a las *condiciones de carreras* que se dan cuando varias tareas concurrentes tratan

de acceder y manipular los mismos datos, y el resultado final depende del orden en que los accesos se ejecutaron. En estos casos se aplican estrategias que permitan coordinar estos accesos de tal manera que los resultados sean coherentes y que ingrese una tarea a la vez al bloque en cuestión [32].

La *memoria compartida* es la memoria que puede ser accedida concurrentemente por varias tareas de tal manera que ellas se puedan comunicar eficientemente o evitar la redundancia de datos [32]. Es importante establecer estrategias de utilización de la memoria ya que se debe restringir la utilización de espacios exclusivos de una tarea en particular, para no afectar la integridad de los datos y del proceso que se está realizando.

La selección de las correctas estructuras de datos para la resolución de problemas en este tipo de sistemas, es un factor determinante de la eficiencia del mismo, y por ende dependiendo del caso, se deben definir las abstracciones adecuadas tratando de evitar carga estructural si la eficiencia se vuelve un factor altamente crítico [32].

2.5.4 Uso del lenguaje de programación C++ para estos sistemas

C++ es un lenguaje de programación imperativo fundado a partir del lenguaje estructural C. Utiliza paradigmas de programación orientada a objetos y programación genérica [34].

La gestión de concurrencia en C++ incluye: manejo de threads, sincronización de operaciones entre threads, y operaciones atómicas³ de bajo nivel [34].

Esta librería de threads está diseñada de tal manera que exista la mayor eficiencia posible para no construir soluciones a bajo nivel, ya que esto incurre en un costo alto en mantenimiento a cambio de eficiencia. En el caso en que esta librería no cumpla con la eficiencia requerida, es necesario utilizar soluciones específicas de la plataforma en la cual se ejecutará la aplicación.

Las especificaciones acerca de las instrucciones y estrategias para realizar programas concurrentes están fuera del ámbito de este trabajo, sin embargo pueden ser encontrados en [34], de donde se tomó un ejemplo sencillo para un código que utiliza threads mostrado en el algoritmo 2.1.

C++ fue el lenguaje seleccionado para construir la solución tecnológica

³Las operaciones atómicas son aquellas que solo pueden ser ejecutadas por un solo proceso, es decir que otro proceso paralelo no la puede efectuar al mismo tiempo.

Algoritmo 2.1: Programa concurrente “Hello World” en C++11 [34]

```
#include <iostream>
#include <thread>

void hello()
{
    std::cout<<"Hello_Concurrent_World\n";
}

int main()
{
    std::thread t(hello);
    t.join();
}
```

debido a que es ampliamente utilizado en aplicaciones en tiempo real que interactúan muy cerca con el hardware gracias a la posibilidad de manejar parámetros a bajo nivel como el caso de la memoria y de brindar al mismo tiempo formas abstractas de representación basadas en el paradigma orientado a objetos, programación estructurada, y programación genérica; lo que disminuye el grado de complejidad en la manipulación de datos manteniendo la eficiencia de los procesos que se realicen [34]. Para el manejo de concurrencia, desde la definición del estándar C++11, se utiliza la librería de **Hilos** (Threads) para la ejecución de tareas simultáneas.

2.6 Inteligencia Artificial aplicada a la música

2.6.1 Cognición Musical

La ciencia cognitiva es un campo de investigación que considera dos puntos de vista, el cerebro como una máquina de procesamiento de información y la mente como un ente consecuente que surge de ese proceso para producir el razonamiento; por lo que los científicos cognitivos son capaces de explicar las operaciones de ambos basados en modelos y abstracciones que los representen [35]. En el ámbito de la cognición musical normalmente se involucran procesos de escucha, interpretación y un área fuertemente explorada como la composición [36], donde se desea ir más allá de los lineamientos impuesto por la teoría musical con el objetivo de expandir el conocimiento y los procesos detrás de este arte.

El aporte de la ciencia cognitiva al campo de la inteligencia artificial es muy importante ya que brinda puntos de partida para el desarrollo de modelos computacionales que ponen a prueba las abstracciones del cerebro y sobretodo de la mente para producir resultados similares a los obtenidos por humanos; en el caso de la música, hay particularidades que deben ser tomadas en cuenta para las operaciones sobre el conocimiento de donde se

basaría un ente artificial que genere música.

2.6.1.1 Conocimientos de Composición e Interpretación musical

- **Conocimiento de Composición:** La composición musical demanda un gran esfuerzo cognitivo al ser el núcleo de la producción musical de un ser humano y posee gran complejidad, tanto así que no existe un modelo general que pueda capturar con exactitud el proceso subyacente que ocurre detrás de esta actividad. Es por ello que la ciencia cognitiva ha propuesto abstracciones que puedan servir de apoyo para la explicación de los resultados de este proceso, donde algunas han sido llevadas al ámbito computacional a diferencia de otras que solo se enfocan en un carácter descriptivo sin posibilidad a un enfoque computacional [35].

En [2], [1] y [35] se detallan abstracciones para la representación del conocimiento musical y las estrategias para manipularlas con el objetivo componer o analizar piezas musicales. A nivel macro es posible identificar en estos trabajos que las representaciones son basadas en gramáticas, métodos estocásticos, redes neuronales, sistemas basados en reglas, programación por restricción y algoritmos evolutivos. Literalmente se hace uso de la teoría lingüística para representar reglas

de composición donde los elementos vienen a ser patrones tales como frases melódicas, transiciones, arreglos de armonías, etc. Con el objetivo de brindar variabilidad al proceso artificial de composición, se utilizan probabilidades basadas en el conocimiento de un compositor, en otras ocasiones se toma el riesgo de la innovación en composición sin un conocimiento exhaustivo que refuerza la estrategia como en el caso de los algoritmos evolutivos que producen resultados interesantes a pesar de su naturaleza, otros casos utilizan híbridos entre estrategias de distintas categorías para enriquecer la composición [2].

Consecuentemente, los modelos aplicados en las investigaciones relacionadas a la composición, cumplen objetivos específicos dependiendo de su morfología. Es decir, si uno de ellos es excelente en un contexto particular, no lo será en otro. Adicionalmente, cabe recalcar, que luego de ser creada toda la pieza musical a través del proceso de composición, es analizada a través del proceso de interpretación [1]. En caso de que la pieza sea interpretada mientras está siendo compuesta nos lleva al ámbito de la improvisación que será explicado más adelante.

- **Conocimiento de Interpretación:** Cuando se habla de interpretación musical no solo se hace referencia al hecho de generar los sonidos para que fluya la composición, detrás de ello es la expresividad humana que

implica gestos y particularidades en la ejecución de una pieza, más que nada es el “toque de distinción” del músico que transmite la composición [1]. Es por eso que se ha hecho énfasis en que la interpretación musical posee un proceso cognitivo que es transmitido a través del plano físico. Es así como las abstracciones consideran elementos tales como el tempo de la pieza, articulación de notas, movimientos corporales, expresiones faciales, etc.

En [1] se mencionan trabajos en el ámbito de la interpretación expresiva donde esta característica es representada por ejemplos brindados por intérpretes humanos para aplicar la estrategia de *Razonamiento Basado en Casos* (Case-Base Reasoning), este enfoque es más efectivo que hacer explícito este conocimiento por medio de reglas debido a la dificultad de la captura de las variaciones presentes en una interpretación musical.

- **Composición e Interpretación en tiempo real (Improvisación):** Los dos puntos anteriores pueden fusionarse en otro proceso creativo con respecto a la música, la *improvisación*, que básicamente es la composición y la interpretación restringida a una producción en tiempo real lo que implica la siguiente prescripción sugerida en [37]:

1. El aprendizaje debe ser incremental y rápido con el objetivo de que

sea compatible con la interacción en tiempo real.

2. Las unidades musicales que se generen deben ser sincronizadas con un planificador en tiempo real, básicamente restringidas a un metrónomo.
3. Para brindar variabilidad a la improvisación se deben mantener varios modelos y cambiar entre ellos en el tiempo de generación.
4. Los modelos que se utilicen deben ser de varios atributos de tal manera de que no solo se alimenten de notas, sino de otros parámetros que mejoren el proceso de improvisación.

Es posible utilizar los modelos de composición y de interpretación como abstracciones para la representación del conocimiento utilizado en la improvisación como es notable en [1]; sin embargo, se debe considerar la eficiencia detrás del uso de esas representaciones. Computacionalmente la composición no debe ocupar un gran tiempo o el resultado de la interpretación perdería sincronía con el tempo de la pieza produciendo resultados que técnica y estéticamente no son aceptables, es por ellos que es importante un estudio del problema a resolver para escoger con criterio la representación más significativa al problema y la más eficiente.

2.6.1.2 Estudio de las emociones producidas por la música

Existe una estrecha relación entre la música y las emociones humanas; al menos en nuestra cultura occidental es un hecho que se considera obvio y que varios investigadores han explorado como lo mencionan los trabajos a los que hace referencia esta sección.

El contenido emocional de una interpretación musical no solo depende de la composición, también están involucrados el estado emocional y experiencia del intérprete u/y oyente, por consiguiente, una misma pieza musical puede poseer sentimientos distintos desde “felicidad” hasta “tristeza” dependiendo de ciertas variables en la interpretación como tempo, dinámica y articulación [38]. La psicología cognitiva es la encargada de asociar emociones a estructuras musicales específicas y afirma que las emociones son expresadas a través de la música, la misma que es percibida por los oyentes; sin embargo, este punto de vista de los “cognitivistas” es cuestionado por los “emotivistas” quienes declaran que la música es capaz de producir emociones, esto implica que no solo un oyente puede identificar la emoción, sino que puede llegar a realmente experimentarla [39].

La relación intérprete - oyente es un factor importante para lograr hacia el segundo la intención emocional del primero. En [39] se muestra cómo

interactúan varios elementos alrededor del intérprete (actor / musician) y el oyente (listener) en un modelo de conmoción como lo muestra la figura 2.26.

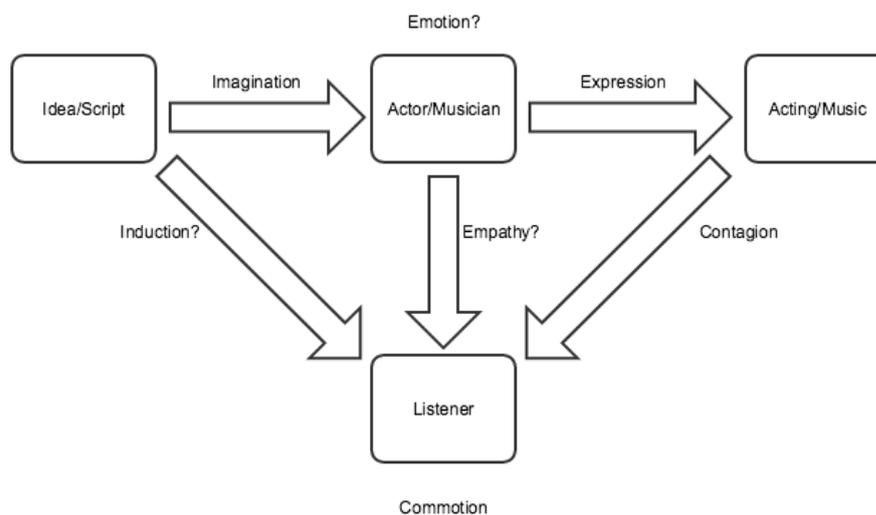


Figura 2.26: Modelo de conmoción que relaciona al intérprete (Actor/musician) con el oyente (Listener)

Fuente: Emotional effects of music: Production rules [39]

La explicación detrás del mecanismo presentado en la figura 2.26 es la siguiente: La relación entre el intérprete y el escucha nace de una idea o un evento en la memoria del músico que estimula su imaginación de tal manera que puede expresar esa emoción de manera física y congruente con la composición; y por lo tanto, inducir esa emoción al oyente por medio de la empatía que se llegue a sentir entre los dos, proveniente del contagio producido por la expresión. Este modelo se muestra superficialmente como una estructura subjetiva de la producción de emoción hacia el oyente; sin embargo, es posible combinar parámetros objetivos con subjetivos como se lo

realiza en [40] cuya intención fue la evaluación de la expresividad transmitida por un intérprete considerando características objetivas de la música tales como tempo, nivel de intensidad del sonido, articulación y timbre, la cuales se relacionaron psicológicamente con las impresiones subjetivas del oyente; para ello se utilizó una modificación del modelo Brunswik's Lens que se detalla en la figura 2.27.

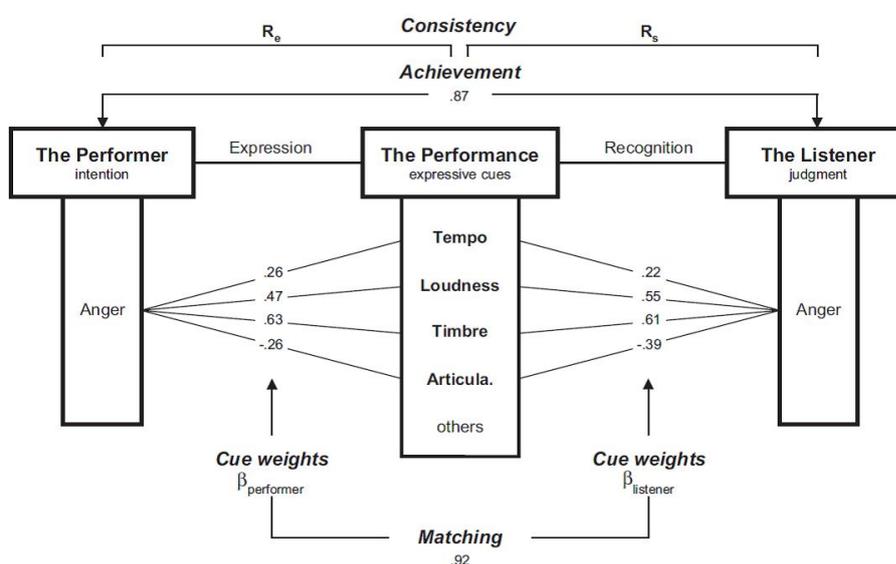


Figura 2.27: Versión modificada del modelo Brunswik's Lens para la comunicación de emociones en la música

Fuente: Play it again with feeling: computer feedback in musical communication of emotions [40]

El modelo Brunswik's Lens [40] se basa en la retroalimentación cognitiva (Cognitive Feedback) que permite comparar la emoción que transmite el músico por medio de su interpretación con respecto a lo que percibe el oyente tomando en cuenta los parámetros de expresividad (expressive cues) cuyo peso en la composición con respecto al carácter emocional es asignado por el

intérprete e idealmente esos pesos deberían ser los mismos que percibe el oyente; para evaluar esa comparación se correlacionaron estadísticamente estos pesos entre intérprete y oyente en [40].

En [41] se expresa la dificultad de la extracción de emociones de la música por ser una cualidad subjetiva relacionada a la cultura, este criterio también es compartido por [42], el cual depone el proceso de composición en un segundo plano con el objetivo de dar paso a las emociones que un sistema de composición musical podría producir siendo este argumento la base del diseño de WOLFGANG, un sistema basado en la cultura de la música occidental del siglo XIX y en etiquetas de emociones brindadas por humanos con el fin de interactuar con lo que ellos llaman una gramática cultural que es un manera de guiar la composición utilizando la teoría musical de una cierta cultura como la mencionada anteriormente. Un factor adicional que tienen en común [41] y [42] es que las emociones se clasifican por medio de etiquetas; un modelo de la clasificación de emociones basados en etiquetas es detallado en [43] el cual se muestra en la figura 2.28 donde es posible identificar las clases para los grupos de emociones; este modelo fue utilizado conjuntamente con algoritmos de clasificación en [41] asumiendo que las emociones dependen de los acordes y del timbre de una pieza de música de carácter occidental.

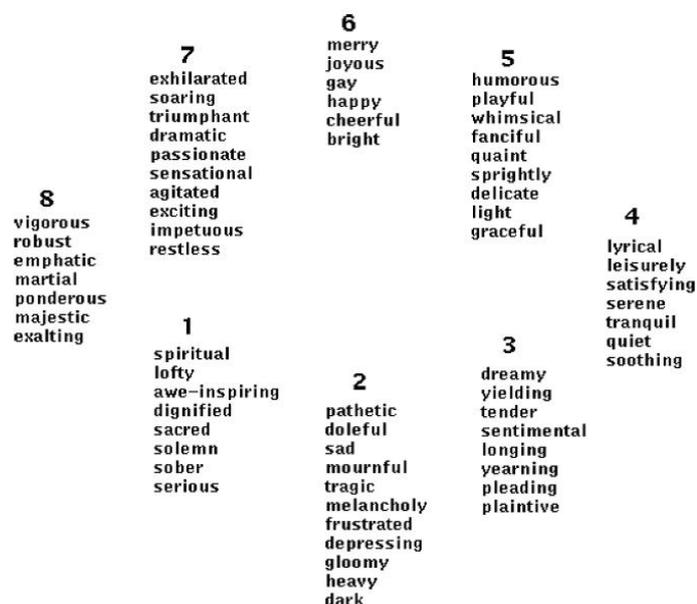


Figura 2.28: Grupos de emociones distribuidos en ocho clases distintas
Fuente: Experimental Studies of the Elements of Expression in Music [43]

Nótese que entre los parámetros a los que hacen alusión las investigaciones detalladas anteriormente se encuentra el timbre, que se relaciona directamente con la estructura del sonido que se quiere sintetizar en base a las emociones que se deseen transmitir y que no ha sido explorado exhaustivamente como otras partes relacionadas a la música; sin embargo, de aquellos pocos trabajos podemos destacar el de [44], donde se fusiona parte de la teoría de la síntesis de audio para producir elementos sonoros que tengan un efecto emocional. En la siguiente sección se presenta la influencia de los sonidos en la relación a la percepción de emociones.

2.6.1.3 Percepción de emociones mediante sonidos específicos

En [45], se detalla un estudio psicológico exhaustivo acerca de la percepción musical y las emociones con respecto a la música, relacionado con la percepción de emociones con respecto al sonido fundamentado en la perspectiva egocéntrica.

De acuerdo con Vickhoff [45], la perspectiva egocéntrica se basa en la fuente de sonido, así como la identificación de esa fuente, la dirección a la fuente y el movimiento de la fuente, lo cual se refiere a los aspectos físicos relacionados al ámbito de la acústica, así como también al área de la acústica que especifica la respuesta humana a los sonidos, de manera que esta perspectiva puede ser enfocada a la percepción del sonido como ente que posee un significado implícito y que puede afectarnos emocionalmente.

El sonido en el aspecto musical tiene relación directa con el timbre y es el que le da color a toda la experiencia musical, siendo parte del impacto emocional sobre la audiencia. Para que pueda haber tal impacto emocional, el oyente debe reconocer y diferenciar entre fuentes sonoras lo cual se logra a través de la impresión de estímulos (stimulus imprinting) que ha aportado a la evolución humana en el ámbito del reconocimiento de fuentes de sonido; por ejemplo, los seres humanos son excelentes en reconocer las voces de las personas [45].

La impresión de estímulos es la razón por la cual la música Pop es de un buen gusto para una gran cantidad de personas ya que confía en el timbre como se lo expone en el estudio en mención; esto es debido, no a un virtuosismo particular, sino a los sonidos que componen el timbre que permanece en la memoria de los oyentes [45]. La consideración de la memoria es de gran significancia ya que es parte importante de la percepción y permite la identificación de estilos musicales solo con el hecho de detectar el sonido que un instrumento musical emite como por ejemplo; una guitarra española es relacionada con el flamenco [45], esto es debido a los efectos de unificación que el cerebro humano realiza a causa de la habilidad que posee para completar patrones, pero no solo completarlos, sino unificarlos; es decir, los seres humanos perciben timbres más que las frecuencias que los componen, acordes que los tonos separados que forman los acordes, sonidos orquestales que los instrumentos separados produciendo el sonido, y melodías más que tonos aislados.

Este mecanismo de percepción se fusiona con la parte emotiva considerando que se producen efectos como parte de un estímulo sonoro. Un ejemplo de estos efectos es que sonidos suficientemente elevados en volumen son aterradores. Pero, generalmente los efectos de mayor relevancia con respecto a emociones son los de disonancia y consonancia.

La disonancia es asociada con sonidos desagradables y la consonancia con sonidos agradables a nivel perceptual; sin embargo, desde el punto de vista musical la disonancia se da en estructuras no congruentes en un contexto dado, mientras que en la consonancia hay congruencia, por ejemplo el acorde de C Major es compuesto por las notas C, E, y G y se puede decir que es consonante porque perceptualmente es puro, claro y agradable, pero si a esas tres notas agregamos C#, entonces tendremos un intruso que volverá al acorde impuro, no entendible y desagradable y por lo tanto disonante.

La consonancia y disonancia difieren en ciertos casos entre la técnica musical y la percepción del oyente; es decir que si la técnica musical considera un elemento disonante, no necesariamente es considerado de esa manera por el oyente y eso es notable principalmente en un pieza donde hay elementos disonantes que, más que contaminar la obra, la enriquecen, pero eso es debido a que estos elementos están dentro de un contexto particular que es la obra musical completa; por ejemplo, los acordes mayores son relacionas a emociones de felicidad, mientras que los menores a tristeza; se sugiere que esto es debido a que los acordes menores tienen un grado de disonancia perceptual; sin embargo, la emoción que estos acordes pueden expresar se da en el contexto de la obra musical que los posea y es dictaminado por la clave musical en la que estén; es decir, si hay una pieza en clave C major, dentro de sus acordes puede poseer D minor, pero eso no lo hace una pieza

que evoque tristeza a diferencia de una pieza que esté en clave D menor que sí podría sugerir ese tipo de emoción [45].

La propuesta de Vickhoff [45] nos lleva a pensar en pre-concepciones del timbre musical que pueden llegar a estimular emocionalmente a un ser humano; tomando esto en cuenta se podría decir superficialmente que sonidos extraños jamás escuchados no producirían tal efecto; sin embargo, eso no ha detenido a las artes sonoras para poder innovar en las matices de timbre que una obra podría tener, es allí donde entra uno de los elementos de mayor influencia en la música experimental que es el sintetizador [5].

Existe un vasto campo de estudio en el tema de la síntesis de sonido desde el punto de vista técnico teniendo como ente físico clave al sintetizador [5]. Hay una amplia variedad de elementos que son parte de la generación de sonido como los vistos en la sección de síntesis de sonido de este documento, el número de combinaciones que se pueden lograr entre elementos y sus configuraciones es prácticamente infinita desde el ámbito perceptual humano, lo que lleva a no tener límites en el proceso creativo que utiliza síntesis, debido a esa complejidad se limitan los elementos que se deben utilizar [44].

Existen pocos estudios que relacionan el sonido con su impacto emocional debido a que el interés es basado en la composición e interpretación más que en el timbre. Uno de ellos es el de Le Groux et al. [44], en donde se limitan los

elementos de síntesis para una evaluación subjetiva, la cual fue realizada utilizando una teoría tridimensional de emociones que considera: valencia (valence), excitación (arousal) y dominancia (dominance), ésta teoría se basa en la escala Self Assessment Manikin (SAM) presentada en la figura 2.29.

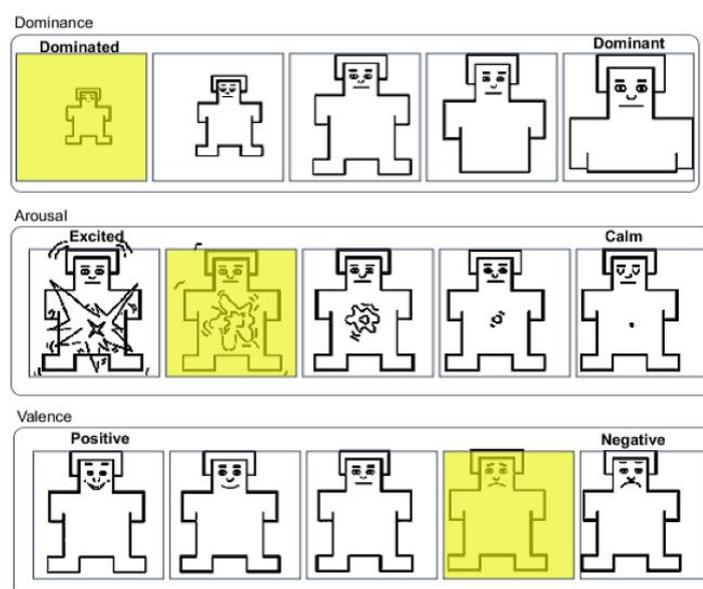


Figura 2.29: Escala Self Assessment Manikin (SAM)

Fuente: Emotional Responses to the Perceptual Dimensions of Timbre: A Pilot Study Using Physically Informed Sound Synthesis [44]

Los resultados obtenidos en [44] basados en la dependencia de la configuración de los descriptores de timbre y su relación con la escala de SAM permiten clasificar esos descriptores con respecto a emociones sirviendo de pautas para la producción de sistemas musicales dirigidos por emociones.

Es importante tomar en cuenta que, considerando la influencia de la memoria

en la percepción emocional, hay que evaluar en primera instancia la reacción de los oyentes ante fuentes sonoras no comunes como las generadas por sintetizadores, pero además se debe considerar la opinión de los músicos por ser el medio de transmisión y empatía hacia los oyentes para producir algún efecto emocional como así lo considera el modelo Brunswik's Lens del que se habló en la sección anterior [40].

2.6.2 Técnicas de Inteligencia Artificial para Composición Musical

La Inteligencia Artificial es una rama de las ciencias computacionales enfocada en la creación de elementos de hardware y software que puedan realizar actividades como percepción, razonamiento y comportamiento inteligente que posee como objetivo a largo plazo alcanzar un nivel de "inteligencia" comparable al humano e incluso superarlo, y a corto plazo desarrollar sistemas que cumplan con el objetivo anterior pero para tareas específicas; por ello se busca no "simular" la inteligencia humana, más bien "reproducir" los efectos de un comportamiento inteligente [46].

Parte de las investigaciones en el área de Inteligencia Artificial toman como referencias áreas humanísticas y los trabajos relacionados a la música no son la excepción considerando que hay técnicas basadas en modelos y métodos

formales provenientes de estudios de la cognición humana como los mencionados en la sección anterior, es por ello que se mostrarán a continuación representaciones y estrategias de Inteligencia Artificial basadas en el conocimiento musical y emocional.

2.6.2.1 Representación del Conocimiento Musical y Emocional

Para poder describir representaciones que nos ayuden a estructurar el conocimiento sobre el cual pueden operar distintas estrategias de Inteligencia Artificial, debemos considerar los elementos atómicos que actuarán en la formación de esas representaciones los cuales provienen de la teoría musical. Recordemos que el enfoque de este trabajo está dado hacia la música occidental que es parte de la cultura por nuestra situación geográfica y por ende utilizaremos su sistema de doce notas con sus frecuencias y respectivas variaciones de acuerdo a las diferentes octavas.

C - C# - D - D# - E - F - F# - G - G# - A - A# - B

Como parte de este sistema de la música occidental, existen manipulaciones con estas notas como doblar frecuencia o reducir a la mitad la frecuencia (subir o bajar octavas) y sus combinaciones puntuales de secuencias (melodías) y simultáneas (armonías). Estos son los elementos atómicos para

la música que rige en nuestra cultura y nuestro modelo mental a menos que hayamos experimentado música con otro tipo de sistema como la hindú o la china donde las octavas no son divididas como la música occidental [47].

Las notas presentadas anteriormente pertenecen a una escala musical llamada la *escala cromática* que es considerada la base para generar otras escalas y acordes. Con esto en mente podemos generar representaciones generales que pueden ser almacenadas en una base de conocimientos, en el siguiente ejemplo mostraremos como lograrlas.

En esta representación tomamos en cuenta tres enfoques, **escalas, acordes y tiempos**. De manera similar como en [48] y [49] podemos asignar a la escala cromática un orden como el mostrado en la tabla 2.2.

C	C#	D	D#	E	F	F#	G	G#	A	A#	B
0	1	2	3	4	5	6	7	8	9	10	11

Tabla 2.2: Escala Cromática enumerada

Existen varias combinaciones estándares de escalas y acordes como lo podemos encontrar en [11]. Estas combinaciones pueden ser patrones que pueden representarse basándose en la *nota clave* que representan, como por ejemplo:

La escala C Mayor que es muy conocida está compuesta por las notas:

C Major: (C, D, E, F, G, A, B)

La nota clave de esta escala es C y tomando en cuenta el orden asignado a la escala cromática podemos representar C Major como lo muestra la ecuación [2.1](#), asumiendo que C es X .

$$\mathbf{X Major:} (X, X + 2, X + 4, X + 5, X + 7, X + 9, X + 11) \quad (2.1)$$

Por lo tanto con esta representación podríamos obtener otras escalas dependiendo de su clave, pero hay que considerar la octava a las que pertenecerían el conjunto de notas a obtener y para ello podemos utilizar la notación de nombre de nota de la figura [2.22](#). En base a este sistema de notaciones y a la ecuación [2.1](#), podemos por ejemplo, obtener la escala F Major en la octava 3, la cuál tendría como ente generador la escala cromática con los órdenes reasignados como lo muestra la tabla [2.3](#).

F3	F3#	G3	G3#	A3	A3#	B3	C4	C4#	D4	D4#	E4
0	1	2	3	4	5	6	7	8	9	10	11

Tabla 2.3: Escala Cromática enumerada para obtener la escala F Major

F3 Major: (F3, G3, A3, A3#, C4, D4, E4)

Como podemos observar, se puede obtener un patrón de melodía bajo una clave musical X , que puede ser representado mediante una ecuación, tal como

la mostrada en [2.2](#).

Patrón de melodía X: $(X, X + a, X + b, X + c, \dots X + n)$ tal que $a, b, c, n \in \mathbb{Z}$

(2.2)

También, podemos extrapolar esta forma de representación para acordes; tomando en cuenta que son notas ejecutadas simultáneamente, por lo tanto podemos representarlos mediante la ecuación [2.3](#).

Patrón de acorde X: $\begin{pmatrix} X \\ X + a \\ X + b \\ X + c \\ \dots \\ X + n \end{pmatrix}$ tal que $a, b, c, n \in \mathbb{Z}$

(2.3)

Para el caso de los tiempos podemos representar una ejecución E como se la muestra en la ecuación [2.4](#).

E: $(nota, te, tp)$

(2.4)

Donde *nota* es la nota musical que es ejecutada, *te* el tiempo de ejecución (por ejemplo tiempo que se mantiene una tecla presionada) y *tp* el tiempo del silencio antes de ejecutar una siguiente nota.

Los tiempos son relativos al tempo general de una pieza que es dado en **BPM** (beats por minuto). Un ejemplo gráfico de esta última representación se da en la línea de tiempo de los secuenciadores como se muestra en la figura [2.30](#).

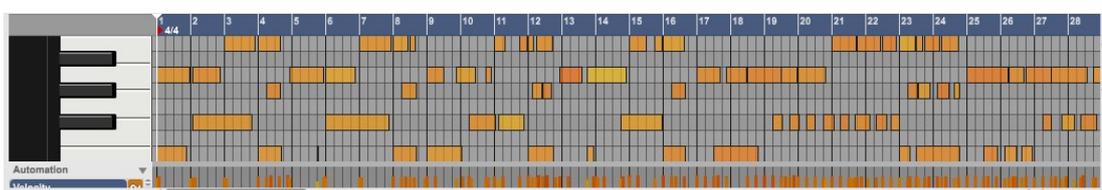


Figura 2.30: Línea de tiempo de un secuenciador MIDI
Fuente: MPC Software : Akai Professional MIDI Secuencer [\[28\]](#)

Para mayor facilidad en la manipulación matemática es adecuado utilizar la notación MIDI como se la presenta en la figura [2.22](#) de la sección [2.3.3](#), además por su amplia utilización a lo largo de las tecnologías de producción de audio tanto en hardware como en software.

Los elementos presentados anteriormente (escalas, acordes y tiempos) forman parte de un nivel de abstracción superior a las notas musicales pero inferior a las representaciones utilizadas para la tarea de composición. Estas representaciones poseen mayor complejidad y están basadas en procesos de composición musical explícitos, análisis de las obras musicales para encontrar procesos implícitos o estrategias computacionales que no tienen que ver

directamente con la música pero aún así han sido explorados en este campo [2] [1].

Para llegar en ciertos casos a concluir el tipo de representaciones a utilizar, es necesario no solo explorar en la teoría, sino considerar el conocimiento específico de los compositores musicales ya que ellos intentan innovar en sus obras y por consiguiente pueden existir lineamientos que no están reflejados en la teoría musical. En esos casos es importante la aplicación de los principios de la **Ingeniería del Conocimiento** para estructurar aquellos lineamientos.

Según [50], la Ingeniería del Conocimiento comprende un conjunto de actividades que pueden desglosarse de la siguiente manera:

1. Adquisición del conocimiento
2. Validación del conocimiento
3. Representación del conocimiento
4. Inferencia
5. Explicación y justificación

Generalmente el conocimiento es obtenido de un experto en el área, ya sea por medio de una entrevista con él o una recolección de datos acerca de sus

actividades, los cuales deben ser procesados y llevados a nivel de conocimiento. Por ejemplo en [49], se realizó este proceso en base a patrones de melodías para la música carnática hindú. Si es obtenido mediante una charla con el experto es sumamente importante que la representación resultante sea entendible tanto para el ingeniero del conocimiento como para el humano experto y el programador del sistema basado en conocimiento.

Existe una variedad de estudios que relacionan el área de composición musical y las ciencias computacionales específicamente en el área de la Inteligencia Artificial donde el objetivo, en la mayoría de casos, es lograr que una máquina sea capaz de generar obras musicales congruentes con las emociones humanas; para ello, se han basado en abstracciones de la cognición humana y en las que permiten resolver problemas no convencionales en el área de la computación. En [6] podemos encontrar una recopilación profunda de algunos enfoques con relación a la composición algorítmica y su relación con la Inteligencia Artificial donde expone las ramas concernientes a la materia como se muestra en la figura 2.31.

A continuación se describen las representaciones utilizadas en la solución propuesta en este trabajo las cuales son parte de la clasificación mostrada en la figura 2.31. La argumentación de la selección de las estrategias se detallará a profundidad en el capítulo 3.

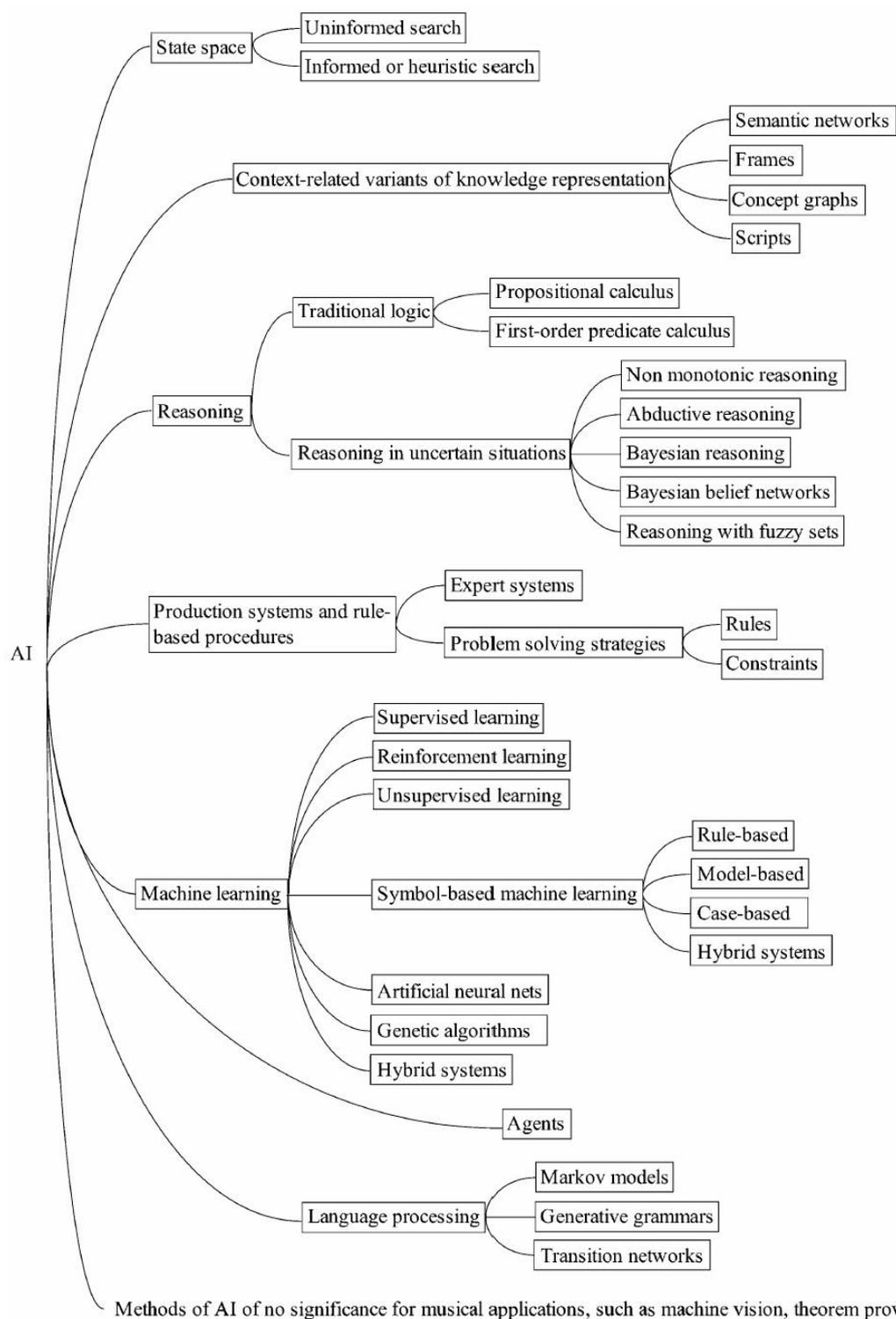


Figura 2.31: Métodos de Inteligencia Artificial relacionados a la composición musical

Fuente: Algorithmic Composition: Paradigms of Automatic Music Generation [6]

•**Lógica Difusa:** La **lógica difusa** (*fuzzy logic*) es una forma de representación basada en situaciones inciertas, donde el valor de verdad no se limita a una dicotomía de verdadero y falso (0 y 1), sino a un conjunto de valores entre 0 y 1 que cataloga que tan verdadero o falso es un ente. En lógica difusa existe el concepto de los conjuntos difusos (*fuzzy sets*) cuyos elementos poseen un grado de pertenencia (*membership degree*) en un intervalo de los número reales $[0, 1]$ que es obtenido mediante una función de membresía (*membership function*) la cuál representa el conocimiento con respecto a un contexto en particular. Estos conjuntos difusos pueden estar organizados en categorías etiquetadas por variables lingüísticas para representar un parámetro en particular, por ejemplo si tenemos un parámetro *tamaño*, éste puede ser representado por una variable lingüística llamada *size* cuyos valores pueden ser pequeño, mediano y grande (*small, medium, large*) y cada uno de ellos es un conjunto difuso. La figura [2.32](#) ilustra gráficamente este ejemplo donde cada etiqueta esta sobre la función de membresía que le corresponde.

En el ámbito de composición, la lógica difusa permite que la definición de áreas imprecisas de parámetros musicales sea posible tales como las emociones que los músicos pretenden transmitir a una audiencia; por otro lado, la aplicación de reglas podrían ser variablemente ponderada

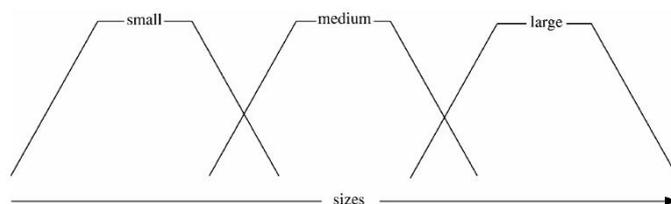


Figura 2.32: Conjuntos difusos para la variable lingüística *size*
Fuente: Algorithmic Composition: Paradigms of Automatic Music
 Generation [6]

por medio de conjuntos difusos que representen diferentes soluciones musicales [6].

En [51], Suiter propone un método para representar una gran serie de parámetros para una composición musical expresiva y estéticamente aceptable donde establece clases importantes a considerar en una composición, además de un ejemplo de los conjuntos difusos que se podrían utilizar con respecto al *timbre*. También se sugiere la utilización de lógica difusa para sistemas de composición algorítmica debido a la imprecisión de los parámetros a los que hace referencia, esta visión es compartida con la que se mencionó en el párrafo anterior encontrada en [6]. La lógica difusa en el ámbito de la música no solo se podría enfocar en composición, más bien en formar representaciones del conocimiento necesitado para esta tarea como lo hicieron en [49], donde un clasificador difuso fue parte de un sistema de adquisición de conocimiento para generar patrones melódicos de la música carnática

hindú.

•**Algoritmos Genéticos:** Los **algoritmos genéticos** (*Genetic Algorithms*) se fundamentan en la idea de formar una población de individuos a través de la supervivencia de los miembros más prometedores; así como la teoría la evolución, es un proceso de selección donde se introduce variaciones a los individuos iniciales a lo largo de generaciones sucesivas donde se van eliminando a los individuos menos prometedores por los más prometedores quienes se adaptan para generar nuevas poblaciones donde emerge diversidad con respecto a las anteriores iteraciones [46]. Por analogía, los individuos son soluciones a un problema determinado que se van alterando para conseguir mejores soluciones. El proceso de resolución de un problema se establece en tres escenarios: en el primero, las soluciones potenciales individuales son codificadas en representaciones que soporten las modificaciones respectivas y los operadores de selección; en el segundo, algoritmos análogos a la actividad reproductiva de un ser vivo, produce nuevos individuos cuyas características son recombinadas con las de sus padres; y el tercero una *función de aptitud* (fitness function) establece cuales son los mejores individuos apropiados para solucionar el problema [46].

Los elementos básicos que están presentes en este proceso son:

Genotype: El genotipo, que es el constituyente genético de un organismo vivo.

Phenotype: El fenotipo, las características observables de un organismo.

Fitness: Aptitud, el grado de aptitud que un individuo posee para continuar.

Chromosome: El cromosoma, la representación del genotipo a través de una cadena; por ejemplo una cadena de bits, donde los elementos atómicos son los **genes**.

Para que sea posible la aplicación de esta estrategia, los datos para la población inicial deben ser codificados de tal manera que puedan ser manipulados. Comúnmente una representación puede ser dada por una cadena de bits; es decir, secuencias de unos y ceros; también una cadena de números enteros, números reales, o caracteres para situaciones de mayor complejidad. En todos estos casos la posición de un dígito representa el valor de algún aspecto de la solución y la definición de las operaciones a aplicar son más sencillas; por ejemplo,

podemos cambiar unos por ceros, tomar un número de manera aleatoria y transformarlo, cambiar un caracter por otro, etc. Otra estrategia de representación de mayor complejidad es construir árboles que soporten las soluciones como estructuras de datos cuyas operaciones pueden involucrar la manipulación de nodos o sub árboles [46].

En [1], se reportan trabajos con respecto a algoritmos genéticos que son catalogados como interesantes y algunos aceptables estéticamente. Además, por la naturaleza de la estrategia, estos resultados en algunos casos se consideran innovadores ya que el propio algoritmo se encarga de generar los nuevos individuos sin una guía heurística muy profunda más que la función de aptitud la cual es complicada de encontrar si se desea una salida con mínima disonancia musical.

Algunas representaciones de los individuos que serían manipulados por una estrategia de algoritmos genéticos son descritas en [6]. Para la generación de frases melódicas se ha optado por iniciar el proceso con un patrón inicial de notas musicales a las cuales se les aplica operaciones de recombinación (crossover) y mutación (mutation) que serán explicados en la sección 2.6.2.3; la función de aptitud se basa en una comparación de los patrones generados con otros patrones de referencia basados en la tonalidad.

El enfoque de algoritmos genéticos puede ser comparado con una estrategia de composición utilizada por los músicos, la cual consiste en tomar una idea inicial existente y cambiarla de alguna manera o más bien *mutarla* de tal manera que se genere una nueva idea [6]. En esta técnica los individuos son representados como notas con las parámetros de tono, duración y velocidad; la población es generada considerando la nota clave y patrones de referencia; las operaciones a aplicar son de espejo (mirror), inversión (invert), reordenamiento (rearrange) y mutación (mutation). La función de aptitud es una comparación entre los patrones generados y los de referencia que es aplicada en cada nota donde el resultado es uno (1) si la nota de un conjunto corresponde al otro; la aptitud resultante será la suma de las aptitudes de cada nota.

La evaluación de aptitud de una población también puede ser realizada por un experto y es referida como una función de aptitud humana (human fitness function); sin embargo, existen inconvenientes con este enfoque ya que el evaluador puede llegar a fatigarse y sus preferencias pueden cambiar a través del tiempo provocando un cuello de botella en el proceso. Soluciones a este problema van desde mantener una población baja o combinar este método con estrategias algorítmicas.

● **Modelos de Markov:** Los **modelos de Markov** (*Markov Models*), son

una representación del procesamiento del lenguaje que son presentados como máquinas de estado finito de manera gráfica como se lo puede apreciar en la figura 2.33. A este tipo de representación también se la denomina *cadena de Markov* y es una especialización de las máquinas de estado probabilísticas; es decir que cada estado representa un ente particular del problema a representar y las transiciones son evaluadas de tal manera que la probabilidad de pasar de un estado o una secuencia de estados encadenado hacia otro estado, es almacenada [46].

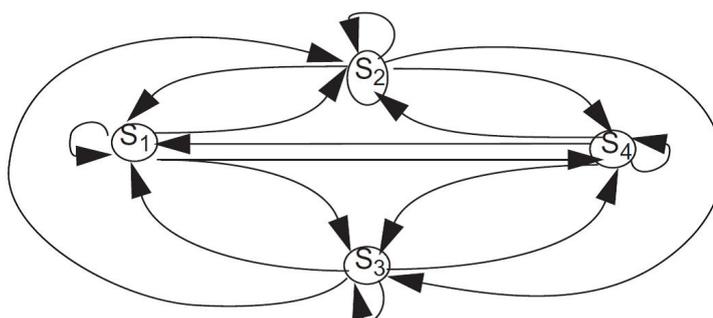


Figura 2.33: Cadena de Markov representada por una máquina de estado finito
Fuente: Artificial Intelligence: Structures and Strategies for Complex Problem Solving [46]

Formalmente podemos denotar el conjunto de estados como S cuyos elementos son $s_1, s_2, s_3, \dots, s_n$; se realizan cambios de estados en tiempos discretos regulares, con la posibilidad de transicionar en el propio estado. Los intervalos de tiempo discretos están dados por el conjunto T que posee los elementos $t_1, t_2, t_3, \dots, t_t$. El cambio de estados

se da de acuerdo a la distribución de probabilidades de cada estado. Considerando que el estado actual en el tiempo t se denota como σ_t , la descripción probabilística requiere la descripción del estado actual σ_t en términos de sus predecesores resultando en una probabilidad que puede ser obtenida para cualquier estado por la ecuación [2.5](#) [\[46\]](#).

$$p(\sigma_t) = p(\sigma_t | \sigma_{t-1}, \sigma_{t-2}, \sigma_{t-3}, \dots) \quad (2.5)$$

La probabilidad obtenida por la ecuación [2.5](#) es un modelo general para la representación de una cadena de Markov de orden n ; es decir, el orden de la cadena depende de el número de estados que le precede a σ_t . Este modelo es llamado **cadena de Markov observable** ya que cada estado corresponde a un evento físico observable.

Una cadena de Markov puede ser representada también como una *matriz de transiciones* como se lo muestra en la figura [2.34](#) con un ejemplo para una cadena de Markov de primer orden relacionado al clima. Una particularidad de esta representación es que la suma de todas las probabilidades de cada estado debe ser igual a 1

Una variación de los modelos de Markov son los **modelos ocultos de**

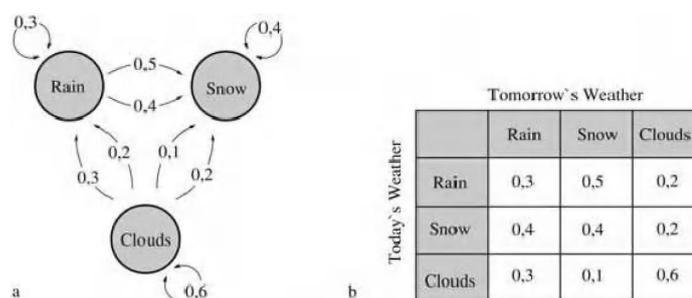


Figura 2.34: Grafo y matriz de transiciones para una cadena de Markov de primer orden

Fuente: Algorithmic Composition: Paradigms of Automatic Music Generation [6]

Markov (*hidden Markov models*). En esta representación la secuencia de salidas observables del modelo de Markov son visibles, pero los estados y las transiciones no; por lo que estos estados no observables producen *probabilidades de emisión* (emission probabilities) las cuales generan las salidas, que en el caso del material musical serían notas con tonos y longitudes [6]. Un ejemplo de un modelo oculto de Markov es el que se puede apreciar en la figura 2.35 donde cada estado representa una función probabilística del lanzamiento de una moneda para obtener cara (head o H) o sello (tail o T) dados por los sesgos b_1 y b_2 que vienen a componer las probabilidades de emisión del evento que no se puede observar de tal manera que se produzcan salidas observables con secuencias de H y/o T .

En el ámbito de composición musical, los modelos de Markov son utilizados para la representación de parámetros musicales de estructura

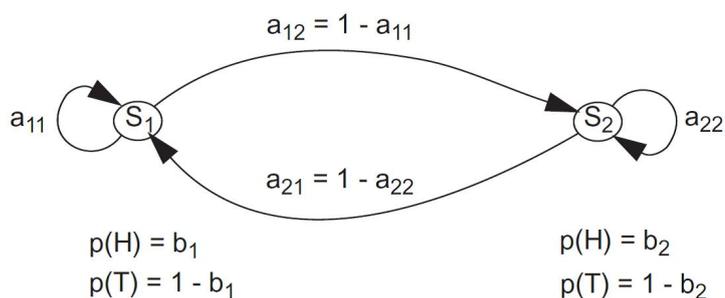


Figura 2.35: Cadena oculta de Markov para el problema del lanzamiento de monedas.

Fuente: Artificial Intelligence: Structures and Strategies for Complex Problem Solving [46]

individual que pueden ser organizados en secuencias tales como las melodías, que son un secuencia de notas; o armonías, una secuencia de acordes; además pueden ser útiles para la imitación de estilos mediante el análisis del cuerpo de una pieza [6]. En [1] se menciona que las cadenas de Markov son utilizadas muy frecuentemente por los investigadores donde se han obtenido resultados poco favorables estéticamente si los modelos son de orden pequeño; sin embargo, los resultados mejoran con órdenes más altos pero sacrificando eficiencia computacional, es por ello que en su utilización se recomienda un balance dependiendo del problema a resolver.

Moorer establece en [52] que las cadenas de Markov no son de alta calidad para oyentes, por lo que su criterio es acorde al que se encuentra en [1]; sin embargo las piezas son razonables por su congruencia con la experiencia pasada en el estilo de la música que los

oyentes evaluadores habían experimentado.

•**Gramáticas Generativas:** Las **gramáticas generativas** (*generative grammars*), propuestas como entes que estructuran una pieza musical en este trabajo, son consideradas una herramienta con un aporte considerable para composición algorítmica y análisis musical [6]. El principio básico consiste en generar construcciones de un contexto particular utilizando *reglas de re-escritura* (rewriting rules). Este principio es basado en la teoría de sintaxis para la construcción de un lenguaje.

La teoría de sintaxis tiene que ver con la estructura formal de sentencias compuestas. La sintaxis tiene como objetivo representar los principios y estructuras para la formación de esas sentencias para un lenguaje. Considerando este contexto, un *lenguaje* consiste de expresiones denotadas por símbolos cuyas estructuras siguen ciertas reglas; es allí donde la sintaxis verifica si la expresión de un lenguaje está acorde con aquellas reglas donde de ser cierto se dice que la expresión es *sintácticamente correcta*. Es posible que dentro de las segmentaciones que se pueden construir mediante la sintaxis exista composición de expresiones que pueden ser estructurados mediante elementos en ordenes distintos pero que cumplen con las reglas semánticas; estas expresiones no son equivalentes y en ese caso son necesarias

transformaciones gramaticales [6].

La generación de nuevas secuencias en una gramática generativa ocurre por medio de reglas de re-escritura como se lo mencionó anteriormente, donde los símbolos del lado izquierdo de una expresión son re-escritos por símbolos del lado derecho; estos símbolos pueden seguir siendo re-escritos en un proceso recursivo si son *símbolos no terminales* (non-terminal symbols); y no re-escritos si son *símbolos terminales* (terminal symbols). Un ejemplo de este proceso se da en las gramáticas de estructuras de frase como el ejemplo encontrado en [6]. En este ejemplo las abreviaciones significan lo siguiente: S (sentence), NP (nominal phrase), VP (verbal phrase), V (verb), PP (prepositional phrase), AP (adjective phrase), Adv (adverb), A (adjective), P (preposition), DET (article o también determiner), y N (noun). La estructura se da a continuación:

$$\begin{array}{l}
 S \rightarrow NP \quad VP \\
 VP \rightarrow V \quad (NP) \quad PP \\
 AP \rightarrow (Adv) \quad A \quad PP \\
 PP \rightarrow P \quad NP \\
 NP \rightarrow (DET) \quad (AP) \quad N \quad (PP)
 \end{array}$$

Los símbolos considerados terminales se dan por la siguiente asignación:

S → man girl John

DET → a the

V → met saw

A → nice good quick

Adv → very extremely

P → in for to

Con esta representación y los datos presentados, un resultado que se podría obtener, es el de la figura 2.36. Una particularidad de esta representación es la posibilidad de representarla gráficamente como un árbol que despliega el proceso para llegar a los resultados de la generación.

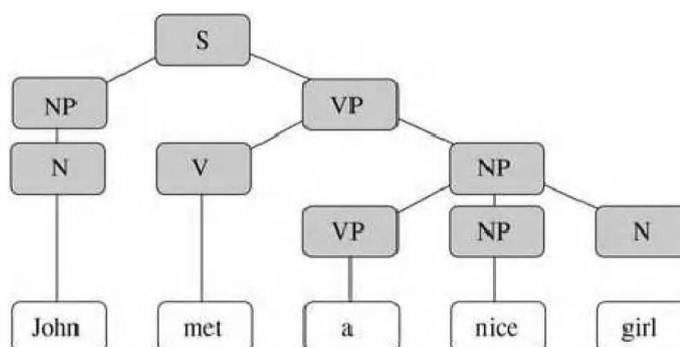


Figura 2.36: Derivación de una gramática de estructura de frase
Fuente: Algorithmic Composition: Paradigms of Automatic Music
 Generation [6]

Para composición musical, las gramáticas generativas son buenas candidatas para representar de manera natural conceptos que permitan

una división jerárquica del material musical y trabajen con la sustitución de símbolos [6]. Es posible llevar a cabo el proceso de formulación de estas gramáticas con dos enfoques; el primero es mediante reglas formuladas de manera explícita o también llamado un *enfoque basado en conocimiento* que podría seguir los lineamientos de construcción de sistemas expertos; el segundo es cuando las reglas son generadas del análisis del cuerpo de piezas musicales, en ese caso es un *enfoque no basado en conocimiento* que también es denominado como **inferencia de gramática**.

Los símbolos terminales para una gramática que represente un tipo de material musical pueden ser por ejemplo acordes, movimientos armónicos, fragmentos de melodías, figuras rítmicas o técnicas de interpretación para algún instrumento en particular. Una estructura que utilice alguno de los ejemplos mencionados anteriormente no necesariamente cumple con las reglas implícitas dictaminadas por la teoría musical; es decir, estructuras de composición que puedan ser aceptables para generar una parte de una pieza puede que sean disonantes para otras partes de la misma pieza u otras. Para sobrellevar este problema se recomienda un análisis del material base para la construcción de la gramática, sobre todo para enfoques que utilicen inferencia de gramática [6].

En [52], Moorer establece estrategias de composición basadas en gramáticas generativas con el objetivo de destilar estilos por medio de inferencia de gramática como también es considerado en los estudios de música e inteligencia artificial en [1]. Conseguir coherencia musical por medio de reglas establecidas manualmente es complicado, por ello se recomienda hacer uso de la inferencia para aumentar la coherencia y asemejar a un estilo [1].

Un caso especial de la utilización de gramáticas aplicadas a la composición musical son los **Lindenmayer Systems** o conocidos también como **L-Systems**. Los L-Systems son una variante de gramática formal cuya característica que sobresale es la *re-escritura paralela* (parallel rewriting) donde las reglas de re-escritura son ejecutadas al mismo tiempo. Estas representaciones han sido aplicadas exitosamente a escenarios de auto-similitud jerárquica para organismos vivos [2]. Para composición musical los elementos del L-System podrían ser colecciones de fragmentos de material musical pre-generados o cualquier otro objeto musical y definir estrategias algorítmicas que interpreten las secuencias generadas en una composición musical estructurada.

Un ejemplo sencillo de un L-System podemos encontrarlo en [53], donde

en base a las producciones de una gramática de contexto libre de la forma: $predecesor \rightarrow sucesor$ tenemos que: El predecesor es un símbolo del alfabeto V , y el sucesor es una (posiblemente vacía) palabra sobre V con lo cual podemos tener las siguientes representaciones:

Alfabeto:

$$V : AB$$

Reglas de producción:

$$P1 : A \rightarrow AB$$

$$P2 : B \rightarrow A$$

Axioma:

$$\omega : B$$

Lo cual produce lo siguiente para cada paso de derivación n cuya representación gráfica se puede apreciar en la figura [2.37](#):

$$n = 0 : B$$

$$n = 1 : B$$

$$n = 2 : AB$$

$$n = 3 : ABA$$

$$n = 4 : ABAAB$$

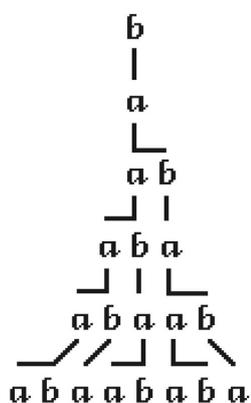
$$n = 5 : ABAABABA$$


Figura 2.37: Derivación de una simple representación de un L-System

Fuente: Musical L-Systems [53]

Podemos destacar otra representación abstracta basada en el procesamiento del lenguaje llamada **redes de transiciones** (*transition networks*). Las redes de transiciones son un conjunto de máquinas de estado finito y son representadas como un grafo, ellas pueden ser extrapoladas para una utilización basada en gramáticas. Una extensión de una red de transiciones es la **red de transiciones aumentada** (*augmented transition network* o ATN), la cual permite instrucciones específicas, saltos condicionales entre estados u otras redes adjuntadas

a los arcos. En la figura [2.38](#) se muestra una red de transiciones aumentada sencilla que permite producir frases melódicas e incluye un comando `jump` que permite la omisión del nodo de ser necesario.

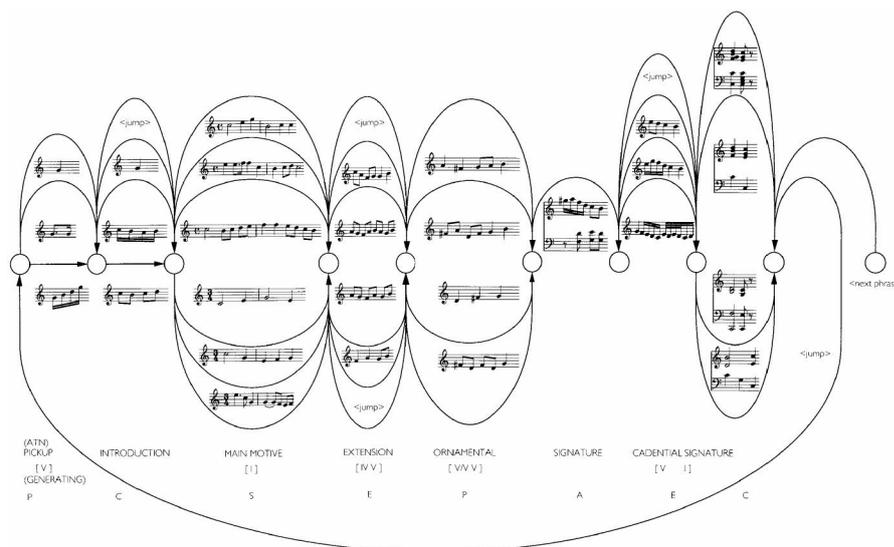


Figura 2.38: Red de transiciones aumentada (ATN), para la producción de melodías

Fuente: Algorithmic Composition: Paradigms of Automatic Music Generation [\[6\]](#)

Un experimento que utiliza ATNs, es el realizado por David Cope cuyo trabajo, “Experiments in Musical Intelligence” (EMI), es analizado por [\[1\]](#), [\[2\]](#) y [\[6\]](#). EMI aplica algoritmos de coincidencia de patrones (pattern-matching), para extraer secuencias musicales cortas que caractericen el estilo del conjunto de piezas analizadas, determinando cómo y cuándo utilizar estos fragmentos para nuevas composiciones. Una vez hecho este análisis, la composición es sintetizada siguiendo las especificaciones provistas por una ATN inferida produciendo resultados

satisfactorios [2].

Cabe destacar que las representaciones descritas no necesariamente son utilizadas individualmente, más bien existen aplicaciones que combinan las estrategias basadas en estas representaciones y otras no discutidas en este documento que permiten plantear hipótesis acerca de los resultados que podrían obtener mediante una configuración u otra. En [2] podemos encontrar una análisis conciso del estado del arte con respecto a la composición musical por medio de la inteligencia artificial en dónde se hace énfasis a modelos híbridos de varias representaciones.

Las representaciones utilizadas en el modelo propuesto son: **Cadenas de Markov, Algoritmos Genéticos , Lógica Difusa y Gramáticas de Contexto Libre**. Los criterios de selección y el funcionamiento de la solución a partir de estas estrategias son tópicos abordados en el capítulo [3].

2.6.2.2 Estructuras que soportan las representaciones

Las representaciones del conocimiento documentadas en la sección [2.6.1.1] tienen potencial para ser implementadas en programas de computadora, para ello es necesario conocer el tipo de estructuras abstractas y concretas que se ajusten mejor para estas representaciones en una máquina, de tal manera que

soporten los algoritmos que pudieran operar sobre ellas.

Una estructura básica es el **arreglo** (*array*), el cual consiste en una colección de elementos que generalmente están ordenados de manera contigua y pueden ser accedidos aleatoriamente por identificadores llamados *índices* [54]. Esta estructura es utilizada de base para representar otras más complejas como lo estudiaremos más adelante.

Una aplicación directa de los arreglos es su utilidad para representar los individuos que participan de los procesos que se desarrollan en los *algoritmos genéticos*, cuyos elementos pueden ser bits, enteros, flotantes u otros que permitan la manipulación de ellos mediante operadores genéticos. Un ejemplo de esto se ilustra en la figura 2.39 donde se aplica la operación *crossover* para manipular una cadena de bits que pudiera ser representada por un arreglo cuya manipulación se daría por medio de aritmética de índices.

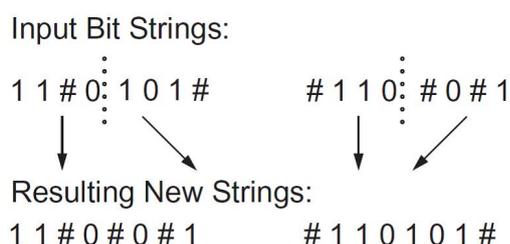


Figura 2.39: Manipulación de arreglos para aplicar la operación *crossover*
Fuente: Artificial Intelligence: Structures and Strategies for Complex Problem Solving [46]

De las representaciones estudiadas, los **grafos** (**graphs**) son parte de las

estructuras que son mayormente utilizadas [46]. Los grafos están presentes en *State Spaces*, *State Machines*, *Semantic Networks*, *Scripts*, *Concept Graphs*, *Neural Networks*, *Markov Models* y *Generative Grammars* [46].

Formalmente un grafo se define como lo muestra la ecuación 2.6, donde V es un conjunto de vértices y E un conjunto de arcos, tal que cada arco en E es una conexión entre un par de vértices V [55].

$$\mathbf{G} = (\mathbf{V}, \mathbf{E}) \quad (2.6)$$

Un grafo puede ser *dirigido* (directed) si los arcos son conectados de un vértice al otro en una sola dirección; en contraste, un grafo cuyos arcos no son conectados en una sola dirección, más bien es de doble vía, es llamado *no dirigido* (undirected). Es posible asociar cada arco con un costo (weight), en cuyo caso el grafo es considerado *ponderado* weighted [55]. La figura 2.40 muestra estos tres tipos de grafos a los que se hace referencia.

Existen dos maneras comunes de implementar un grafo. La primera consiste en utilizar una matriz de adyacencia como lo muestra la figura 2.41(b). La matriz de adyacencia es un arreglo $|V| \times |V|$, donde $|V| = n$ y n es la cantidad de vértices en el grafo, para considerar una conexión entre un vértice y otro

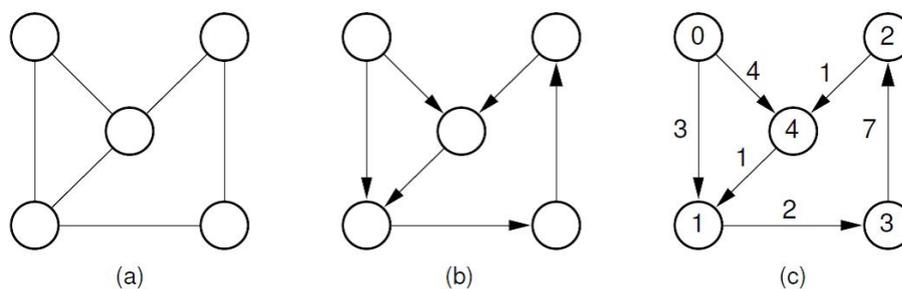


Figura 2.40: (a) Grafo no dirigido. (b) Grafo dirigido. (c) Grafo ponderado
Fuente: Data structures and algorithm analysis in C++ [55]

se asocia un 1 con la casilla que interseca a esos vértices; si necesitamos que el grafo sea ponderado, el valor del costo sería el que ocupe la casilla. Ya hemos mostrado un ejemplo para una de las representaciones en la figura 2.34 referente a los modelos de Markov [46].

La segunda forma de implementar un grafo es utilizando una lista de adyacencia, ilustrada en la figura figura 2.41(c). La lista de adyacencia es un arreglo de listas enlazadas⁴ de longitud $|V|$ donde cada una de las listas representan los arcos que existen por cada vértice adyacente al vértice que la posee, cada uno de esto arcos puede poseer información sobre la conexión como por ejemplo el costo del arco [55].

Las representaciones estudiadas pueden ser modeladas mediante el **diseño orientado a objetos** para ajustarse lo mejor posible a su abstracción [46]. El diseño orientado objetos resulta de gran utilidad si se lo utiliza adecuadamente

⁴Las listas enlazadas son elementos secuenciales que utilizan asignación dinámica de memoria, es decir que nuevos elementos pueden ser agregados por demanda.

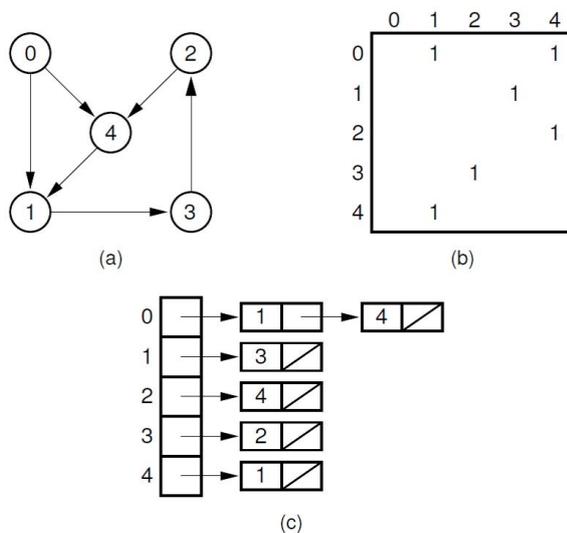


Figura 2.41: (a) Grafo dirigido. (b) Matriz de adyacencia. (c) Lista de adyacencia
Fuente: Data structures and algorithm analysis in C++ [55]

para representar correctamente los datos que se desean manipular; es decir, aplicar correctamente los conceptos de tal manera que se obtenga un sistema robusto y fácilmente mantenible a lo largo del tiempo donde el conocimiento de buenos diseños tomados de experiencias pasadas pueden ser extrapolados a nuevas aplicaciones; para este caso la utilización de patrones de diseño toma una vital importancia como conjunto de herramientas a la hora de modelar un sistema [56].

Sin embargo, no hay que pensar que todo problema debe ser resuelto mediante el diseño orientado a objetos, ya que existen restricciones que pueden limitar este tipo de modelamiento por la sobrecarga estructural que podría afectar a un sistema; es por ello que para un software que se enfoque en ambientes distribuidos, en tiempo real o aplicaciones de dominio

específico, posiblemente no es el enfoque correcto [56].

Las estructuras presentadas son las más comunes para el modelamiento e implementación de las representaciones del conocimiento explicadas; sin embargo, dependiendo de las restricciones que se puedan presentar al desarrollar un sistema que utilice estas representaciones junto con sus algoritmos, se debe tener en cuenta la posibilidad de favorecer al parámetro que lo necesita más; es decir, si la aplicación que se esté desarrollando necesita de un nivel crítico de optimización en CPU, debemos reducir la sobrecarga de procesamiento al máximo en donde podrían encajar otros paradigmas como el **diseño orientado a datos** (*data-oriented design*) donde se reduce la complejidad de las abstracciones a arreglos básicos y se utiliza una filosofía de “transformación de datos” a favor de la eficiencia [57].

2.6.2.3 Algoritmos de Composición Musical

En esta sección se plantean las estrategias más importantes utilizadas con las representaciones del conocimiento y sus estructuras descritas en las secciones anteriores. El enfoque será exclusivo para composición musical; sin embargo, cabe recalcar que los algoritmos presentados no solo tratan con tareas de composición, más bien algunos de ellos son inspirados en otras aplicaciones que hacen uso de técnicas de inteligencia artificial.

Para composición musical, *heuristic search* (búsqueda heurística) es el método utilizado mayoritariamente para dirigir la búsqueda de soluciones en un espacio representado por elementos musicales, sobre todo cuando se trata de aplicaciones que necesiten reducir la sobrecarga de procesamiento en un gran espacio [6]. Existen algoritmos basados en búsquedas heurística tales como **backtrack**, **hill climbing**, **simulated annealing** and **dynamic programming**, cuyo modelo que guía la búsqueda es fijo [46]. Sin embargo, un algoritmo que permite mayor flexibilidad es el **best-first search** cuyo pseudocódigo es presentado en el algoritmo 2.2.

El núcleo de un algoritmo basado en heurística es su *función heurística* la cual está basada en la experiencia humana con respecto a un problema en particular y es la que dirige la búsqueda. Dependiendo de la aplicación en composición musical que se esté realizando, la función heurística puede cambiar; es así como pueden haber funciones que comparen patrones de melodías dependiendo de su estructura en un espacio cuyos nodos representan precisamente esos patrones, o también se pueden extraer reglas de armonía que representen las funciones heurísticas para generar nuevas armonizaciones [6]. En [58] se propone una comparación entre la interpretación de un ente humano y partituras con notas que se esperan ejecutar para acompañar al humano; esta comparación es la función heurística para encontrar la mejor coincidencia entre notas entre los dos entes

Algoritmo 2.2: Best-First Search [46]

```

1 function best_first_search()
2
3   begin
4     open:= [Start ]           // initialize
5     closed:= []
6     while open  $\neq$  [] do           // states remain
7       remove the leftmost state from open, call it X
8       if X = goal then
9         return the path from Start to X
10      else
11        generate children of X
12        for each child of X do
13          switch child do
14            case the child is not on open or closed
15              assign the child a heuristic value
16              add the child to open
17            end
18            case the child is already on open
19              if the child was reached by a shorter path then
20                give the state on open the shorter path
21              end
22            end
23            case the child is already on closed
24              if the child was reached by a shorter path then
25                remove the state from closed
26                add the child to open
27              end
28            end
29          endsw
30          put X on closed
31          re-order states on open by heuristic merit (best leftmost)
32        end
33      end
34    end
35    return FAIL           // open is empty
36  end

```

comparados y posteriormente utilizar este enfoque para el diseño de un acompañamiento en tiempo real.

En el caso del **razonamiento en situaciones inciertas** (*reasoning in uncertain situations*), la utilización de **lógica difusa** es adecuada para parámetros imprecisos y una de las metodologías para la aplicación de esta representación se define en [51]. Esta metodología inicia de la pre-definición de fuentes de sonido; por ejemplo, un timbre inicial del cuál posteriormente se pueden dar dos tipos de tratamiento. El conjunto de decisiones para el primer tratamiento está relacionado a la segmentación del timbre en sí, a través del tiempo (por ejemplo, creando ritmos) y a través del espacio o registro de frecuencias (por ejemplo, únicamente utilizando las octavas medias del piano o un filtro paso-alto). El conjunto de decisiones para el segundo tratamiento se relaciona a los cambios externos impuestos en el timbre; es decir, los que no se originan del timbre en sí, lo que incluye cambios en la densidad del sonido, en los niveles de amplitud, y en la locación de fuente de sonido dada en la interpretación. Por implicación, los factores mencionados describen en conjunto la *esencia* de la música y como un todo también determinará la *forma*. Este enfoque puede ser extrapolado a otros parámetros a más del timbre, tales como frecuencia, tiempos, o patrones más complejos [51].

El proceso para la utilización de lógica difusa es dado por los siguientes pasos:

1. Se definen los objetivos y el criterio a utilizar de tal manera que se tenga pleno conocimiento del problema a resolver, las salidas esperadas y posibles fallas.
2. Se determinan las relaciones entre las entradas y las salidas, además se definen las variables difusas que actuarán en el sistema.
3. El problema es descompuesto en una serie de reglas `IF X AND Y, THEN Z` que definen las repuestas para obtener las salidas del sistema deseadas con respecto a las condiciones de entrada.
4. Crear funciones de membresía que definan el significado o los valores de las entradas y salidas utilizadas en las reglas.
5. Ya en funcionamiento, las reglas son aplicadas dependiendo de los valores recibidos en las entradas generando grados de membresía para la obtención del resultado final por medio de las operaciones sobre conjuntos difusos.
6. Los resultados del paso anterior entran a un proceso llamado *defuzzifying* para obtener el resultado exacto de todo el proceso.

En la sección [2.6.1.1](#) presentamos de manera general la representación de los **algoritmos genéticos** y la descripción de sus elementos fundamentales.

Recordemos que esta estrategia utiliza una *función de aptitud* (fitness function) para evaluar las soluciones de tal manera que se seleccione a los mejores candidatos para producir la próxima generación de soluciones. Es así como, a través de operaciones análogas a la transferencia de genes en la reproducción sexual, el algoritmo crea una nueva población de soluciones [46]. Ahora, consideremos $P(t)$ definido en la ecuación 2.7 como la población de soluciones candidatas X_i^t en el tiempo t . Con ella podemos presentar la forma general del algoritmo genético ilustrado en el algoritmo 2.3.

$$P(t) = \{X_1^t, X_2^t, \dots, X_n^t\} \quad (2.7)$$

Algoritmo 2.3: Genetic Algorithm [46]

```

1 procedure genetic_algorithm()
2
3   begin
4     set time t:= 0
5     initialize the population P (t)
6     while the termination condition is not met do
7       evaluate fitness of each member of the population P (t)
8       select members from population P (t) based on fitness
9       produce the offspring of these pairs using genetic operators
10      replace, based on fitness, candidates of P (t) , with these
        offspring
11      set time t:= t +1
12    end
13  end

```

En esencia, un algoritmo genético es estructurado de la siguiente manera [46]:

1. Generar aleatoriamente una población de n individuos.
2. Calcular la aptitud de cada individuo.
3. Aplicar operadores genéticos tales como recombinación (*crossover*), mutación (*mutation*), inversión (*inversion*), etc.
4. Reemplazar la población anterior con los nuevos candidatos de acuerdo a la aptitud.
5. Repetir el paso 2 con la nueva generación.

Parte importante de este proceso son los operadores genéticos aplicados para producir las nuevas generaciones. La **recombinación** (*crossover*) toma dos soluciones candidatas y las divide intercambiando componentes de ella para producir otra dos nuevas candidatas. Un ejemplo de este proceso se da en la figura 2.42, donde se produce un *crossover* sobre una cadena de bits de longitud 8. El punto de división es arbitrariamente escogido, sin embargo, este punto puede ser cambiado en cualquier momento del proceso y seleccionado aleatoriamente si se lo desea [46].

La **mutación** (*mutation*) toma un único candidato y cambia aleatoriamente algunos aspectos de él. Por ejemplo, si tomamos una de la cadenas de la figura 2.42, podemos escoger uno de sus bits y reemplazarlo por 1, 0 o #. La

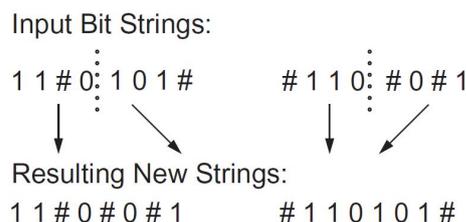


Figura 2.42: Recombinación de dos cadenas de bits de longitud 8. # puede ser 0 o 1

Fuente: Artificial Intelligence: Structures and Strategies for Complex Problem Solving [46]

inversión (*inversion*) permite invertir el orden de los elementos en una cadena, por ejemplo, para la cadena de bits 101010, la aplicación del operador de inversión produciría la cadena 010101. El **intercambio** (*exchange*) permite intercambiar dos elementos distintos en una cadena, por lo que si consideramos una vez más al individuo 101010, la operación de intercambio entre el bit cero y el bit cinco resultaría en 001011. Estos operadores son los más utilizados, sin embargo existen otros que pueden ser considerados dependiendo de la aplicación, tales como: **espejo** (*mirror*) o **reordenamiento** (*rearrange*) [46].

La función de aptitud (fitness function) es clave para la selección de los individuos. En el ámbito de composición musical es posible definir distintas funciones para los parámetros que quisiéramos utilizar en alguna aplicación y podrían provenir también del criterio directo de un experto. Ejemplos de funciones de aptitud pueden ser encontradas en [6], y algunas de ellas son

presentadas a continuación:

- Variedad de Tonalidad: $\frac{\sum \text{tonos distintos}}{\sum \text{todos los tonos}}$
- Intervalos de disonancia: $\frac{\sum \text{intervalos de disonancia}}{\sum \text{intervalos}}$
- Rango rítmico : $\frac{\sum(\text{duración máxima de nota}) - (\text{duración mínima de nota})}{\sum \text{valores rítmicos}}$

Estas funciones de aptitud no son aplicables a toda problema; es decir que el problema de composición a resolver debe ser estudiado de tal manera que se defina la función que mejor se ajuste a él o en todo caso se puede evaluar una función propuesta por medio de experimentación [6].

Para los **modelos de Markov** (*Markov models*) el interés principal en composición musical es la generación de secuencias a partir del conocimiento representado en las probabilidades de transición de un elemento a otro. Para la generación de secuencias en representaciones basadas en modelos de Markov se utilizan criterios heurísticos para dirigir la navegación por el grafo de estados sobre algoritmos comunes que obtienen probabilidades de secuencias. Algunos de estos algoritmos son mencionados en [6] particularmente para **modelos ocultos de Markov** (*hidden Markov models*) donde además de las probabilidades de transición existen probabilidades de *emisión* que dictaminan el grado de ocurrencia de una salida para un estado

particular. Estos algoritmos son: **forward-backward algorithm**, **Viterbi algorithm** y **Baum–Welch algorithm**.

Forward-backward algorithm calcula las probabilidades de las secuencias más recurrentes hacia adelante (forward) y luego hacia atrás (backward) donde simultáneamente se van obteniendo los *smoothed values* que son probabilidades obtenidas por la multiplicación de las probabilidades de forward y backward para obtener los estados más posibles en un tiempo dado [59]. Esta estrategia es presentada en el algoritmo [2.4].

Algoritmo 2.4: Forward-Backward Algorithm [59]

```

1 function ForwardBackward (guessState, sequenceIndex)
2
3   begin
4     if sequenceIndex is past the end of the sequence then
5       return 1
6     end
7     if (guessState, sequenceIndex) has been seen before then
8       return saved result
9     end
10    for each neighboring state n do
11      result:= result + (transition probability from guessState to n
        given observation element at sequenceIndex) *
        ForwardBackward (n, sequenceIndex + 1)
12    end
13    save result for (guessState, sequenceIndex)
14    return result
15  end

```

Viterbi algorithm se encarga de encontrar las probabilidades de la ruta más prometedora para cada estado del grafo con la posibilidad de recuperar esa ruta por medio de un proceso de rastreo al finalizar el algoritmo. Esta

estrategia puede ser implementada de manera similar al algoritmo [2.4] cambiando el proceso suma y multiplicación dentro de la iteración por la búsqueda del `sequenceIndex` que posea la *máxima probabilidad* de transición en el estado analizado [6].

En el caso del **Baum–Welch algorithm** podemos utilizar un cuerpo como el del algoritmo [2.4] así como *Viterbi*, pero la operación a realizar es la búsqueda de los parámetros de una cadena oculta de Markov de tal manera que la probabilidad para un estado en particular sea la máxima. Estos parámetros son: *las probabilidades de transición, las probabilidades de emisión, los estados del grafo*. La estrategia en cuestión debe encontrar los entes que permitan producir aquel máximo local [6].

Finalmente, para las **gramáticas generativas y redes de transiciones aumentadas** se aplican estrategias de producción de elementos siguiendo la sintaxis especificada por las reglas de las gramáticas propuestas; estas estrategias se basan simplemente en reemplazos sucesivos de símbolos no terminales hasta poder obtener una estructura compuesta por símbolos terminales. Lo importante se encuentra en cómo dirigir la generación de los elementos; por ejemplo, en [6] se hace referencia al trabajo de David Cope que se mencionó al final de la sección [2.6.1.1] con respecto a EMI. Para este caso en particular, además de la guía sintáctica por parte de las gramáticas,

se aplicaron criterios de “semántica” tales como *tensión* y *relajación* reflejados en la forma de la estructura de los patrones musicales básicos para la generación. Así también, dependiendo de la aplicación, se pueden definir criterios que permitan producir material musical a partir de estructuras lingüísticas.

Los algoritmos presentados en esta sección representan únicamente guías para poder generar composiciones musicales que tienden a producir emociones, no deben ser vistos como estrategias absolutas. Ellos podrían ser utilizados en procesos que combinen más de un procedimiento, lo cual dependerá de lo que se desee obtener como salida. Es importante tomar en cuenta que la composición musical tiene un componente subjetivo fuerte y que las heurísticas son elementos importantes en cualquier estrategia o estrategias que se deseen aplicar [6].

2.6.2.4 Estrategias de asociación entre emociones y color del sonido

Recordemos que el *color del sonido* se relaciona al **timbre** que es lo que diferencia un sonido de otro a pesar de que en términos musicales se encuentre en la misma tonalidad (frecuencia fundamental) [10]. En esta sección se exponen estrategias de asociación entre emociones y timbre, las cuales podrían ser parte de una plataforma de composición musical como

ente adicional para incrementar la expresividad en una interpretación.

En la sección [2.6.1.3](#), se hizo referencia a la *síntesis de sonido* como productora del timbre en composiciones musicales que pudieran generar emociones; principalmente se hizo mención del trabajo de Le Groux et al. [\[44\]](#) que plantea una estrategia para poder encontrar asociaciones entre emociones y el timbre (color del sonido), donde se sigue un enfoque basado en modelamiento físico de sonidos de impacto y se proponen parámetros acústicos que vienen a ser *descriptores perceptuales de timbre* tales como *log-attack time*, *spectral centroid* y *spectral flux*; la descripción de cada uno se encuentra en el artículo de investigación.

Se diseñó un modelo de síntesis considerando a los descriptores perceptuales con el objetivo de evaluar la respuesta emocional que la manipulación de ellos pudiera producir [\[44\]](#). Este modelo consta de un generador de impulso (Impulse Generator), filtros resonantes (Resonators) y Envoltentes (ADSR) como se muestra en la figura [2.43](#). Nótese que existe un bajo grado de complejidad en la estructuración del modelo y la cantidad de elementos de síntesis, sin embargo los parámetros modificables en los descriptores pueden llevar una gran cantidad de combinaciones para obtención de sonidos, es por ello que en el estudio se limitó a la evaluación de valores fijos para los parámetros en cuestión.

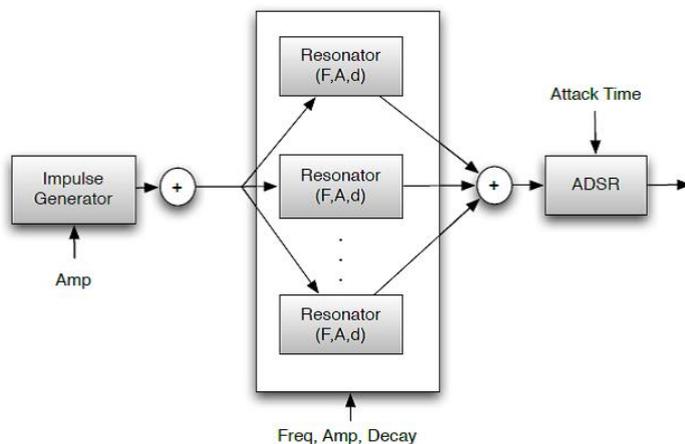


Figura 2.43: Componentes de síntesis de sonidos para evaluar respuesta emocional

Fuente: Emotional Responses to the Perceptual Dimensions of Timbre: A Pilot Study Using Physically Informed Sound Synthesis [44]

En [60], se presenta un modelo para control de sonido por medio de emociones las cuales son reflejadas indirectamente a través de parámetros físicos del intérprete con respecto a la audiencia. Este modelo se presenta en la figura 2.44. El control de la generación del sonido se daría por medio de la asociaciones entre emociones y parámetros de síntesis para modelamiento físico de instrumentos. En este trabajo se sugiere que la adquisición de datos se la realice por medio de: gestos de las extremidades, actividad cardíaca, conductividad de la piel, temperatura, movimiento ocular, gestos faciales e incluso actividad cerebral y que esos parámetros sean asociados con elementos de síntesis. No se especifican los elementos que se podrían utilizar para la generación de sonido en particular, solo se nombran parámetros de modelos físicos que podrían ser asociados a este método tales como los

correspondientes a: flautas, clarinetes e instrumentos de viento metálicos; a pesar de ello, el enfoque para adquirir descriptores de emociones sería adecuado para incrementar la expresividad de una interpretación.

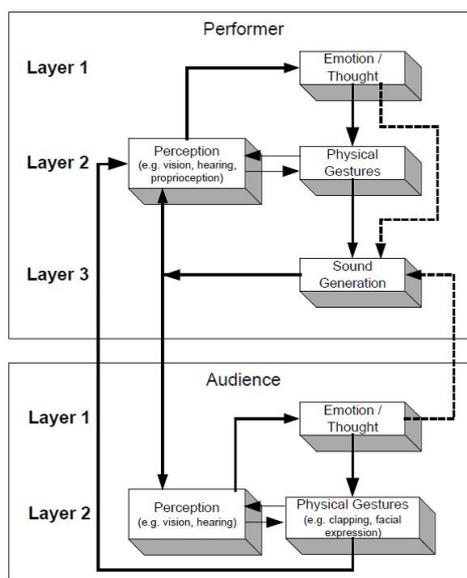


Figura 2.44: Modelo de generación de sonido a través de emociones
Fuente: The Integral Music Controller : Introducing a Direct Emotional Interface To Gestural Control of Sound Synthesis [60]

Estos trabajos presentan dos escenarios; el primero consiste en identificar descriptores que permiten establecer emociones, el segundo es la asociación de estos descriptores a parámetros de síntesis para obtener timbres congruentes con las emociones. Basado en este modelo general y utilizando lo discutido en las secciones 2.6.1.2 y 2.6.1.3 se podrían formular descriptores de timbre basados en un listado de emociones y utilizar estrategias de clasificación para asignarlos a parámetros de síntesis y a material musical que un sistema de composición pudiera generar.

2.6.3 Manejo de datos relacionados a la evaluación estética de la música

Teniendo en mente que la composición musical y su influencia en los seres humanos es subjetiva, las evaluaciones de piezas musicales vienen a ser un indicador de la intención pretendida por el intérprete de la obra musical evaluada y su relación con los oyentes como se explicó de manera detallada en la sección [2.6.1.2](#). A continuación se explican consideraciones que han sido propuestas para realizar este tipo de evaluación sobre métodos y sistemas de composición musical.

2.6.3.1 Parámetros a recopilar con respecto a composición y sentimientos

Básicamente una evaluación de sentimientos con respecto a piezas musicales se basa en asociar las emociones con la pieza o el fragmento de pieza musical que se está escuchando. Esto nos lleva a considerar modelos que relacionen estos dos factores cuyos actores son los intérpretes (músicos) y los oyentes (audiencia); en el caso de los sistemas de composición musical el intérprete sería la máquina. Las figuras [2.26](#) y [2.27](#) presentadas en la sección [2.6.1.2](#) muestran modelos que consideran el proceso por el que la música fluye entre

los dos actores.

En medio de la relación entre el intérprete y el oyente se encuentran los parámetros definidos cuyos valores representan un cierto grado de expresividad que debería ser congruente entre los dos actores como lo presente el modelo modificado *Brunswik's Lens* de la figura [2.27](#). La utilización de esta metodología para evaluar la calidad estética de una composición resulta un tanto compleja para oyentes con pocas bases musicales. El objetivo primordial es que el oyente llegue a comunicar las emociones que las obras transmiten, por lo tanto otras estrategias de evaluación se basan en la clasificación de emociones por medio de *etiquetas* para las piezas que representan un sentimiento en particular; [41](#) y [42](#).

Las emociones pueden ser establecidas de manera puntual y seguir clasificaciones tales como la propuesta por [43](#) presentada en la figura [2.28](#) de la sección [2.6.1.2](#), o también la escala SAM utilizada por [44](#) para evaluar emociones con respecto al timbre. Pero a pesar de un intento por establecer esas clasificaciones, los sentimientos producidos pueden no ser los mismos de persona a persona para una misma pieza si hablamos de una clasificación sumamente específica, es por ello que las emociones propuestas para una evaluación en ciertos casos ocupan áreas más generales como las descritas en [38](#), las cuáles toman en cuenta: ira, tristeza, felicidad y miedo; algunas

veces complementadas con solemnidad y ternura.

Por lo tanto, la evaluación de emociones puede ser enfocada tanto a intérpretes humanos como máquinas. Sin embargo, una prueba importante con relación a sistemas basados en inteligencia artificial es la **prueba de Turing** (*Turing Test*). Como se explica en [46], el matemático británico Alan Turing propuso una prueba empírica basada en el cuestionamiento de la capacidad de inteligencia que podría poseer una máquina. Esta prueba fue denominada por Turing como el *juego de la imitación* (imitation game), el cual consiste en colocar una máquina y un humano en lugares separados de un segundo humano que sería el interrogador para ambos entes como lo muestra la figura 2.45. El interrogador no debe tener contacto con los sujetos más que de manera indirecta para evitar un sesgo en su juicio. Si el interrogador no logra distinguir entre el humano y la máquina en base a sus preguntas, entonces se considera a la máquina como un ente “inteligente” [46].

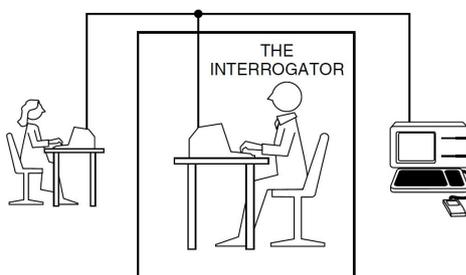


Figura 2.45: La Prueba de Turing

Fuente: Artificial Intelligence: Structures and Strategies for Complex Problem Solving [46]

Para el caso de las obras musicales generadas por entes artificiales, se podría adaptar la prueba de Turing de tal manera que a expertos musicales se les requiera clasificar las piezas producidas por humanos y las producidas por máquinas de un único grupo de piezas ordenadas al azar. Sin embargo, según Wiggins et al. [35], éste método sufre de tres grandes dificultades descritas a continuación:

- Se puede producir un sesgo por las pre-concepciones acerca de música por computadora.
- Permite juicios mal formados.
- Falla en examinar el criterio que es utilizado para juzgar las composiciones.

Wiggins et al. [35] expone un método de evaluación llamado **consensual assessment technique (CAT)** que es un método empírico para la evaluación de creatividad que resuelve los tres problemas de la prueba de Turing expuestos anteriormente. La prueba es realizada por jueces que deben seguir las siguientes indicaciones con respecto al producto a evaluar:

1. Tener experiencia en el dominio.
2. Realizar evaluaciones independientes.

3. Evaluar otros aspectos del producto tales como realización técnica, atractivo estético u originalidad.
4. Realizar juicios relativos de cada producto en relación con el resto del estímulo (conjunto de productos).
5. Proveer calificaciones en órdenes aleatorios diferentes para cada juez.

Dependiendo del interés de una investigación, estos métodos de evaluación pueden ser acogidos en su totalidad o modificados según las necesidades del investigador el cual debe de elevar al máximo el grado de confiabilidad de las pruebas a realizar.

2.6.3.2 ¿Cómo interpretar y analizar los datos recopilados?

En el ámbito de evaluación emocional los datos recopilados se pueden interpretar como un conjunto de asociaciones entre las piezas musicales generadas y las emociones asignadas. En este caso se realizaría una comparación entre la intención emocional original de la pieza generada y la emoción real producida en el oyente. Esta comparación puede realizarse cuantitativamente contabilizando aciertos y fallas en las asociaciones obteniendo datos estadísticos para realizar pruebas de hipótesis que permitan determinar diferencias significativas entre parámetros definidos, como los

experimentos realizados en [40].

Para pruebas relacionadas al enfoque de Turing o CAT también se puede hacer uso de la estadística para el análisis de datos estableciendo características cualitativas dadas por criterios estéticos, o cuantitativas como calificaciones para las piezas, o aciertos y fallas en clasificar las obras como productos de máquinas o de humanos, además se puede utilizar la retroalimentación de los evaluadores para expandir los criterios que puedan explicar los resultados como se lo realiza en [35].

2.6.3.3 ¿Cómo establecer y comunicar los resultados obtenidos?

Los resultados serán las tendencias descubiertas en los grupos evaluados que sean obtenidas por el análisis estadístico. La retroalimentación de estos resultados más los comentarios de los evaluadores son las guías para llegar a conclusiones exactas con respecto a los experimentos realizados.

Estos resultados no solo pueden estar plasmados en números o características puntuales. Ellos pueden ser organizados como un “todo” explicativo para establecer fallas y fortalezas del enfoque en composición utilizado; de esta manera puede iniciar una nueva iteración sobre las estrategias utilizadas en la aplicación. En [35] se maneja detalladamente este

enfoque y lo utilizan precisamente para realizar mejoras en los métodos de composición algorítmicos.

Finalmente, los resultados pueden ir en este caso en particular a dos principales audiencias: científicos computacionales y músicos; es decir, audiencia técnica y audiencia artística. Dependiendo del caso, los resultados pueden ser condensados para el área de interés a la cual se pretenda apuntar. En un caso es adecuado la descripción de procedimientos, representaciones, tecnologías y demás materias que interesen al científico computacional obviando la discusión filosófica detrás de la generación de ese conocimiento, lo cual sería importante para el otro bando; el de los músicos, quienes buscan la fuente y el método como guía para su inspiración y proceso creativo. La interacción armoniosa entre estas dos áreas es importante para el bien común, que es apoyar a los artistas en sus procedimientos mediante herramientas tecnológicas inteligentes para lo cual se pretende que este trabajo de investigación sea parte de los cimientos que permitan lograr este objetivo.

CAPÍTULO 3

3. ANÁLISIS Y SELECCIÓN DE LAS TÉCNICAS DE INTELIGENCIA ARTIFICIAL

Para la selección de las estrategias de inteligencia artificial adecuadas a resolver este tipo de problemas, se optó por establecer una guía basada en la revisión del estado del arte en el área de inteligencia artificial aplicada a la música, que se encuentra en la sección [2.6](#); por lo cuál, se realizó inicialmente un proceso de *Ingeniería de Conocimiento* a dos músicos con bases académicas y practicantes de la música experimental; proceso que se convertiría en nuestra primera estrategia a aplicar.

Para este proceso se siguieron los procedimientos encontrados en [\[50\]](#). Por lo

cual, inicialmente se realizó una **entrevista no estructurada**¹ estableciendo lineamientos fundamentados en las experiencias e idiosincrasias de los entrevistados, dando como resultado una serie de puntos con respecto a su proceso de composición y las bases; técnicas y filosóficas, de las cuales dependen. Estos lineamientos pueden ser encontrados en la sección de **Anexos** (ver Anexo **A**).

La entrevista no estructurada realizada a los músicos nos brinda parte del conocimiento de ellos a nivel macro. Decimos que es parte del conocimiento macro por dos razones: la primera es que su proceso creativo está en constante evolución generando nuevo conocimiento que influencia a la producción de su música y la segunda es que en el proceso de la entrevista; por su naturaleza, se pudieron haber obviado puntos que deriven a una mayor cantidad de información.

El enfoque en este tipo de casos conlleva a realizar una segunda entrevista llamada **entrevista estructurada**² que se enfoca en el *por qué* de ciertas acciones para obtener y entender el *cómo* se toman esas decisiones. Debido a la naturaleza del dominio, se consideró que no es posible realizar una

¹Una entrevista no estructurada en el ámbito de ingeniería del conocimiento es una conversación fluida con preguntas espontáneas para tener una visión general del dominio del conocimiento del experto [50].

²Una entrevista estructurada en el ámbito de ingeniería del conocimiento permite formular preguntas específicas acerca de las reglas que rigen el dominio, de tal manera que puedan ser plasmadas en un sistema experto. Estas preguntas están guiadas por la entrevista no estructurada realizada con anterioridad [50].

entrevista estructurada, ya que para aquellos músicos era complicado mencionar su proceso de toma de decisiones a nivel de ejecución (notas, acordes, ritmos) lo que se cataloga como algo normal si analizamos las investigaciones previas donde músicos interactúan directamente con un sistema de adquisición de datos con el objetivo de estructurar su conocimiento [49].

Por tal motivo, se realizó otro enfoque basado en los lineamientos de los músicos en conjunto con la revisión de literatura realizada, de tal manera que se obtuvieron metodologías abstractas para la extracción del conocimiento que ellos no pudieron articular, y así ser representado para posteriormente definir algoritmos con la finalidad de manipular este conocimiento.

En este capítulo se expondrá un análisis para la selección de estos elementos (representaciones y algoritmos) para la estructuración del enfoque de composición musical mediante inteligencia artificial.

3.1 Composición Musical

3.1.1 Representación del Conocimiento

En primera instancia, recordemos que el tipo de música en que se fundamenta este trabajo es de carácter occidental. Por lo tanto, para la representación de los elementos atómicos de este tipo de música tales como escalas, acordes y tiempos, se utilizarán las abstracciones propuestas al inicio de la sección [2.6.2.1](#), cuyo fundamento son las notas musicales de la escala cromática y que se describen en las ecuaciones [2.1](#), [2.2](#) y [2.3](#). Estos elementos serán los que las representaciones más complejas manipularán.

Ascendiendo en el nivel de abstracción, proponemos una arquitectura de composición para adquisición y representación del conocimiento, ilustrada en la figura [3.1](#).

Esta arquitectura trata directamente con el conocimiento implícito que un músico refleja en sus obras, el cual es adquirido y representado en base a los lineamientos de los músicos entrevistados y las estrategias de composición algorítmica expuestas en el marco teórico. Como se podrá notar, la arquitectura está dividida en módulos especializados que aportan de manera incremental a la producción de soluciones. Cabe destacar que los elementos

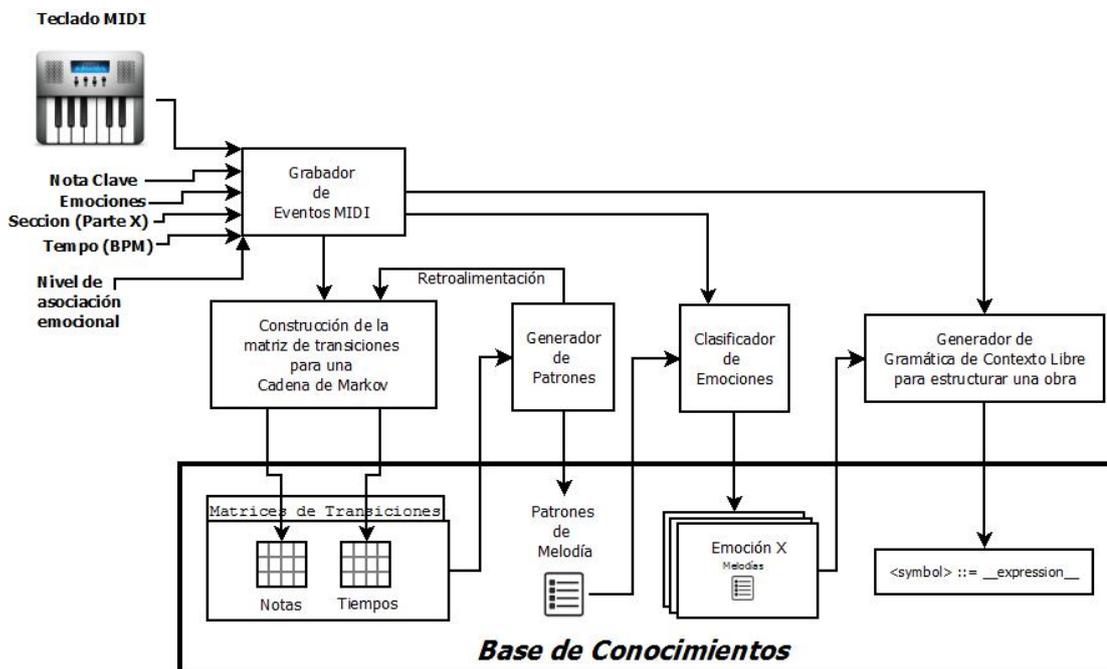


Figura 3.1: Arquitectura de Representación del Conocimiento para Composición Musical

esenciales inherentes a esta representación son **melodías**, por consiguiente, no está considerado un modelo que involucre armonías debido a que el enfoque es dirigido finalmente a la producción de melodías concurrentes que acompañan a músicos reales, por lo tanto estos músicos son los que ejecutan las bases armónicas. Estas consideraciones serán explicadas en secciones posteriores.

A continuación se describen los componentes de esta arquitectura tomando de ejemplo una obra de uno de los compositores con gran influencia en los músicos entrevistados, consideraremos la *Tocata y fuga en re menor, BWV 565* de *Johann Sebastian Bach* que es muy conocida mundialmente y controversial

por sus componentes únicos para la época:

● **Entradas:** Tal como se muestra en la figura 3.1, esta arquitectura necesita los siguientes datos de entrada:

- La información de la ejecución de la composición por los músicos; es decir, una secuencia de eventos compuestos por tres elementos que son: nota musical que se está ejecutando (0 a 127), velocidad (0 a 100) y duración (BPM beats por minutos). Estos son los datos adquiridos de un teclado musical siguiendo el estándar MIDI.
- La nota clave (*key*) que rige la interpretación del músico, relativa a los patrones que se van a extraer.
- La emoción que expresa el sentimiento de lo que se está interpretando en ese momento (alegría, nostalgia, ira, etc), y que forman parte de la base de conocimientos.
- La sección o tramo de la interpretación, tal como fue asignado por el autor a su composición; es decir, la parte A, B, C, etc., de la composición las cuales cada una podría tener su conjunto de emociones en particular. Y,
- El tempo o la restricción de tiempo dada en BPM (beats por minuto),

guiado por un metrónomo de sincronización entre la interpretación y la grabación que el sistema realiza.

Considerando la obra de Bach, podemos estructurar un conjunto hipotético de entradas basado en el criterio de los músicos de esta manera:

Teclado MIDI: A, G, A, G, A, E, D, C#, D..... (Cada una debe ser acompañada por su velocidad y tiempo)

Nota Clave: D menor

Tempo: 80 BPM

Emociones y Partes: Mostrados en la tabla [3.1](#)

Intervalo de tiempo (mm:ss)	Parte	Emociones
[0 : 00, 00 : 24)	A	Ira
...
[02 : 48, 04 : 05)	F	Melancolía, Nostalgia, tristeza
...
[08 : 11, 09 : 10)	J	Sensación de libertad

Tabla 3.1: Partes y emociones de la Tocata y fuga en re menor, BWV 565 de Johann Sebastian Bach

- **Grabador de Eventos MIDI:** Este módulo recolecta las entradas y las procesa de acuerdo al requerimiento de los otros módulos del sistema; también, almacena la interpretación combinada con el resto de entradas, de tal manera que puedan reproducirse después. La utilización de aplicaciones de grabación MIDI reduce la complejidad de

la implementación.

•**Matriz de transiciones para una Cadena de Markov:** Las Cadenas de Markov, al ser estructuras de secuencias basadas en probabilidades, cuentan con un gran potencial para la representación de frases melódicas donde los estados de dicha cadena pueden ser dados por valores de notas MIDI, frecuencias, tiempos, o duraciones, entre otros. En base a ello y al conocimiento adquirido de los músicos entrevistados, en este módulo se generan dos modelos de cadenas Markov mediante matrices, una para las transiciones entre notas musicales (números MIDI); y, otra para las transiciones entre tiempos y duraciones, representados por (te, tp) . Esta representación nos permite a futuro generar composiciones más ricas, combinando patrones de tiempo con patrones de secuencias de notas que no necesariamente utilizaron esos patrones de tiempo y generar así conocimiento nuevo, que no fue adquirido directamente de los expertos.

Por ejemplo, las matrices de transiciones de notas y duraciones para la Toccata y Fuga de Bach presentan una particularidad en sus melodías, donde algunas de ellas no se encuentran dentro de la clave D menor, y las duraciones se pueden encontrar en su mayoría representadas por semicorcheas (Sixteenth Notes); es decir, si tenemos un tempo de 80

BPM, considerando que un beat es una corchea (Quarter Note), entonces las duraciones de las notas en su mayoría serían de 0.375 segundos. En la figura 3.2 se presenta la matriz de transiciones de notas para esta obra.

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
C	0.06	0.01	0.19	0.04	0.07	0.09	0.02	0.08	0.00	0.11	0.27	0.05
C#	0.02	0.04	0.36	0.00	0.24	0.02	0.00	0.02	0.00	0.15	0.13	0.03
D	0.11	0.08	0.21	0.05	0.12	0.08	0.02	0.10	0.01	0.15	0.06	0.01
D#	0.15	0.02	0.42	0.02	0.00	0.23	0.02	0.08	0.00	0.02	0.02	0.02
E	0.05	0.13	0.19	0.00	0.05	0.23	0.01	0.16	0.00	0.15	0.02	0.00
F	0.04	0.02	0.17	0.05	0.19	0.04	0.00	0.23	0.01	0.17	0.06	0.01
F#	0.02	0.01	0.22	0.09	0.03	0.04	0.01	0.35	0.01	0.18	0.05	0.00
G	0.08	0.01	0.09	0.03	0.14	0.17	0.04	0.05	0.01	0.23	0.11	0.04
G#	0.00	0.00	0.17	0.00	0.08	0.06	0.04	0.17	0.12	0.25	0.00	0.12
A	0.03	0.08	0.17	0.00	0.08	0.12	0.02	0.19	0.00	0.11	0.17	0.04
A#	0.11	0.10	0.10	0.02	0.05	0.02	0.01	0.20	0.01	0.34	0.03	0.01
B	0.24	0.22	0.09	0.01	0.01	0.04	0.00	0.17	0.01	0.17	0.00	0.04

Figura 3.2: Matriz de transiciones de notas de la Toccata y Fuga en re menor, BWV 565 de Johann Sebastian Bach

Se puede apreciar en esta matriz que apenas 18 transiciones de las 144 posibles tienen probabilidad cero lo cual nos da un indicio de que las líneas melódicas no se restringen de una manera estricta a D menor.

•**Generador de Patrones:** En este módulo de la arquitectura propuesta, se crean los patrones melódicos y rítmicos usando un corpus de pequeñas piezas melódicas. Cada pieza es una limitada secuencia de notas con sus respectivas duraciones generadas por las cadenas de Markov que han sido previamente entrenadas con melodías creadas por los músicos.

Cada pieza es representado por la ecuación 3.1. Donde M es la pieza

melódica obtenida del proceso de Markov, y cada elemento S_i , contiene $(nota_i, te_i)$, donde $nota_i$ es el valor de la nota MIDI y te_i el valor de la duración de la $nota_i$.

$$\text{Patrón de Melodía M: } (S_1, S_2, S_3, \dots, S_n) \quad (3.1)$$

El proceso para generar estos patrones consiste en combinar N diferentes piezas melódicas del corpus, creando nuevas secuencias. Debido a que el corpus puede ser vasto, debemos realizar una búsqueda informada entre las diferentes soluciones. Los resultados de esta búsqueda son las piezas melódicas que estén basadas en las características fundamentales y estructurales de una composición musical.

En el proceso de composición existe una retroalimentación al módulo de construcción de matrices para facilitar el proceso iterativo que ocurre durante la composición, y así generar más patrones que son el resultado del autoaprendizaje del sistema. Algunos de esos patrones no necesariamente son congruentes con las emociones que se desean transmitir, de manera que este modelo concibe un módulo clasificador de emociones, para clasificar estos patrones.

Por ejemplo, en la obra de Bach se generan secuencias de notas con largas duraciones y esa secuencia es similar a las notas musicales de una parte que evocaba esperanza, pero con duraciones cortas, posiblemente, el resultado emocional sería más sublime apuntando a sensaciones positivas.

•**Clasificador de Emociones a través de Lógica Difusa:** Una de las entradas al sistema son las etiquetas de emociones que el músico coloca a sus composiciones. Este módulo permite asignar esas etiquetas a los patrones generados por el sistema, dependiendo de su nivel de relación con patrones ya etiquetados; es decir, si son patrones nuevos provenientes del generador de patrones discutido en el punto anterior, y deseamos saber si pertenecen a la emoción “alegría”, basta con comparar con los patrones almacenados de alegría, y determinar su grado de cercanía a través de *lógica difusa*; esta comparación también nos permite determinar el grado de membresía que podría tener con respecto a otras emociones considerando que una pieza musical puede provocar más de una emoción con distintos niveles como lo expusieron los músicos en la entrevista aplicada.

En el ámbito de Lógica Difusa, para construir el conocimiento requerido, tomamos en cuenta un proceso de **fusificación** a partir de los patrones

de melodías generados cuya representación es dada por la ecuación [3.2](#), donde *note* es un entero entre 0 - 127 que representa un número MIDI para la nota musical, y *duration* que es un tiempo relativo basado en el *tempo* (beats per minute, BPM) el cual marca el ritmo del patrón de melodía; por ejemplo, si un *duration* es igual a 1 y el *tempo* para un patrón de melodía es 120 BMP, entonces el tiempo absoluto *t* para una nota musical será de 0.5 segundos basado en la ecuación , la cual convierte el tiempo relativo *duration* a un tiempo absoluto *t*.

$$\begin{aligned} \text{Melody Pattern: } & (note_0, duration_0), & (3.2) \\ & (note_1, duration_1), \\ & \dots, (note_n, duration_n) \end{aligned}$$

$$t = \frac{60 \cdot duration}{tempo} \quad (3.3)$$

Las duraciones relativas descritas anteriormente son fijas y representadas por este arreglo de números decimales [0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3, 3.25, 3.5, 3.75, 4, 4.25, 4.5, 4.75],

los cuales son representaciones de tiempos musicales como la teoría musical lo describe [10], donde 1 es una *negra* cuyo símbolo es ♩, y el resto son duraciones específicas (por ejemplo, 0.5 es una corchea con el símbolo ♪) o combinaciones (por ejemplo, 0.75 es una corchea + semicorchea que en notación musical es ♪.).

EL tamaño n o *número de patrones* que se encuentra en la ecuación 3.2 es un parámetro de entrada que controla el número de pares generados (note, duration) que compondrán el patrón de melodía. *Los silencios musicales* [10] no son considerados en un patrón de melodía debido a que son fusionados con su nota previa inmediata con el objetivo de reducir la complejidad de la representación del patrón.

Las matrices de transiciones son construidas a partir del conjunto de melodías interpretadas por el compositor humano quien provee las emociones con su respectivo *nivel de asociación emocional* que vienen a ser la *intención emocional* del compositor hacia su audiencia, estas matrices no utilizan esas intenciones en el proceso de generación de nuevos patrones de melodía, pero son utilizadas en el proceso de clasificación como fuente para etiquetar aquellos nuevos patrones de tamaño n , con las intenciones dadas por los músicos. Este proceso de clasificación es descrito a continuación:

1. El proceso es aplicado a las notas y las duraciones de manera independiente, así que un patrón de melodía es dividido en dos arreglos; uno llamado *notes* y otro *durations* para notas y duraciones respectivamente. Un patrón de melodía x generado de las Cadenas de Markov será llamado MM_x (Melody Machine), cuyos arreglos correspondientes son $MMnotes_x$ y $MMdurations_x$

2. Las melodía grabadas de los humanos, como *material musical*, tienen un tamaño de m , y serán conocidas como MH_y (Melody Human) entonces, los dos arreglos que les corresponden serán divididos como en el punto anterior, por lo tanto tenemos $MHnotes_y$ y $MHdurations_y$. El próximo paso obtendrá la distancia entre MM_x y MH_y la cual será utilizada en el etiquetado de emociones para MM_x .

3. Para calcular la diferencia entre cada elemento de los vectores $MMnotes_x$ y $MHnotes_y$, los cuales contienen números MIDI entre 0 y 127, utilizaremos la siguiente ecuación:

$$\Delta note_{ij} = |MHnotes_y[i] - MMnotes_x[j]| \text{ mód } 12 \quad (3.4)$$

Debido a que tratamos con notas musicales, las cuales son

representaciones lineales compuestas esencialmente de 12 elementos distribuidos en varias octavas [10], tenemos que esta ecuación considera que las octavas no son relevantes; por ejemplo, si tenemos una nota C3 la cual es 48 en MIDI, y otra C4, con nota MIDI 60, entonces la diferencia es 12, y por tanto con $\text{mód}12$ obtenemos cero y básicamente no hay diferencias entre estas dos notas.

4. La ecuación [3.4] se aplica para cada elemento de $MMnotes_x$ y $MHnotes_y$ para calcular la distancia entre $MMnotes_x$ y un segmento $S_y[k]$. El segmento $S_y[k]$ es un subconjunto de tamaño n donde k es un entero positivo entre 0 y $m - n$, y está contenido en $MHnotes_y$. Si $m \geq n$, habrá $m - n + 1$ segmentos contenidos en una melodía creada por un humano, pero si $m < n$, entonces habrá únicamente solo un segmento y las operaciones no considerarán todo el arreglo $MMnotes_x$. Las ecuaciones [3.5] y [3.6] ilustran el cálculo para la distancia entre $MMnotes_x$ y un segmento $S_y[k]$.

$$\Delta note_{ik} = |MHnotes_y[i + k] - MMnotes_x[i]| \text{ mód } 12 \quad (3.5)$$

$$d(MMnotes_x, S_y[k]) = ds[k] = \frac{\sum_{i=0}^{\min(m,n)-1} \Delta note_{ik}}{\min(m,n)} \quad (3.6)$$

Esencialmente, este modelo considera un promedio entre los valores $\Delta note_{ik}$ en un segmento $S_y[k]$ cuyo resultado siempre será un número entre 0 y 11, debido a la operación mód.

5. El número de distancias obtenidas por cada segmento, entre $MMnotes_x$ y $MHnotes_y$ es de $m - n + 1$; por consiguiente, para determinar la distancia mínima entre todos los segmentos en $MHnotes_y$ y $MMnotes_x$, utilizamos la siguiente ecuación:

Para $m > n$,

$$d(MMnotes_x, MHnotes_y) = dnotes_{xy} = \min\{ds[k] : k \in [0, m - n + 1]\} \quad (3.7)$$

Para $m < n$ habrá solo una distancia, la cual es:

$$d(MMnotes_x, MHnotes_y) = ds[0].$$

6. Este cálculo de distancia entre una melodía de máquina $MMnotes_x$ y una de humano $MHnotes_y$, tiene que ser aplicado para todas las melodías de la base de conocimientos; por lo tanto, si p es el número

de melodías de humano en la base de conocimientos, entonces la *melodía humana más cercana*, en términos de distancia y del patrón relacionado, a $MMnotes_x$ es dada por las ecuaciones [3.8](#) y [3.9](#).

$$DNotesMin_x = \text{mín}\{dnotes_{xy} : y \in [0, p - 1]\} \quad (3.8)$$

$$MHnotes_{min} = MHnotes_y, \text{ tal que} \quad (3.9)$$

$$\text{mín}\{dnotes_{xy} : y \in [0, p - 1]\}$$

7. La melodía de humano **notes** $MHnotes_{min}$ es relacionada con una melodía completa (notas y duraciones) que llamaremos **Notes** MH_{min} que fueron una de las melodías etiquetadas previamente por un músico quien estableció las emociones y sus pesos (nivel de asociación emocional). A este conjunto de emociones lo llamaremos E cuyo tamaño lo denotaremos como ne y una emoción específica será conocida como E_r , tal que r es un entero en el intervalo $[0, ne - 1]$. Para cada melodía de humano MH , hay

un conjunto de emociones E con sus correspondientes pesos w_r . Por lo tanto, las emociones y sus pesos para **Notes** MH_{min} son utilizadas para etiquetar el nuevo patrón generado $MMnotes_x$ como lo describe la ecuación [3.10](#).

$$EMMnotesX_r = EMHnotesMin_r \left(1 - \frac{DNotesMin_x}{11}\right) \quad (3.10)$$

Utilizamos 11 para normalizar $DNotesMin_x$ para cada patrón debido a que este valor está en un rango entre 0 y 11, tal como se explica en el punto 4.

Lo que la ecuación [3.10](#) hace es asignar pesos para cada emoción $EMMnotesX_r$ en el patrón de la melodía de máquina $MMnotes_x$ utilizando los pesos de las emociones que ponderan a la *melodía de humano más cercana a $MMnotes_x$* la cual es **Notes** MH_{min} . Los pesos de estas emociones son denotados como $EMHnotesMin_r$ y son ponderados por $DNotesMin_x$ como la ecuación lo describe. Por ejemplo, si un patrón generado ($MMnotes_x$) tiene una distancia de 3.5 ($DNotesMin_x$) con respecto a su melodía de humano más cercana (**Notes** MH_{min}), y las emociones proveídas por el músico a esa melodía son *felicidad(10)*,

tristeza(90) y *melancolía(75)* (en este caso $ne = 3$ y r está en $[0, 2]$), entonces el patrón generado será ponderado utilizando $EMHnotesMin_r(1 - \frac{3,5}{11})$, por lo tanto los resultados son *felicidad(6.82)*, *tristeza(61.36)* y *melancolía(51.14)* para ese patrón generado ($MMnotes_x$).

8. Este proceso es extrapolado para duraciones, por lo tanto los mismos principios son aplicados al arreglo $MMdurs_x$, pero considerando: Primero, la distancia entre los elementos $MMdurs_x$ y $MHdurs_y$ estará dada por la ecuación **3.11**.

$$\Delta dur_{ij} = \text{mín}(|MHdurs_y[i] - MMdurs_x[j]|, 4) \quad (3.11)$$

Debido a que las duraciones son relativas al tempo *tempo* (BPM), el valor es limitado a 4 beats, lo que nos permite tener una referencia numérica para el factor de normalización cuando los pesos de las emociones son calculadas. Se escogió 4 porque una *medida completa de ritmo* (bar) puede ser marcada por 4 beats como lo hace un metrónomo **[10]**; y, segundo, el factor de normalización de 4, como se describió anteriormente.

9. Finalmente, el proceso tiene que ser aplicado a todos los patrones

MM ; es decir, para todos los patrones construidos por el generador de patrones. Por lo tanto, la base de conocimientos tendría dos conjuntos de patrones ponderados con emociones, uno para las notas $MMnotes$ y otro para las duraciones $MMdurs$. Debido a que estos conjuntos no son combinados en un solo MM con ponderaciones, una melodía puede ser construida con notas y duraciones que provengan de diferentes patrones generados cuando se aplique un **proceso de defusificación**, lo que proveería una mayor flexibilidad en el proceso de composición musical.

Por ejemplo, la figura 3.3 muestra posibles funciones de membresía correspondientes a patrones melódicos con respecto al ejemplo de emociones identificadas en la tabla 3.1 para la obra de Bach. Nótese que en el eje X (patrones melódicos), los valores P_i serán discretos y representados por arreglos de patrones, pues se debe considerar la posibilidad de que existan elementos con el mismo grado de membresía.

•**Generador de Gramática de Contexto Libre para Estructurar una Obra:**

A través de este módulo se logra estructurar la obra como un todo a partir de la definición de las secciones (partes) y los patrones de melodías. El producto de este módulo es el conjunto de reglas que

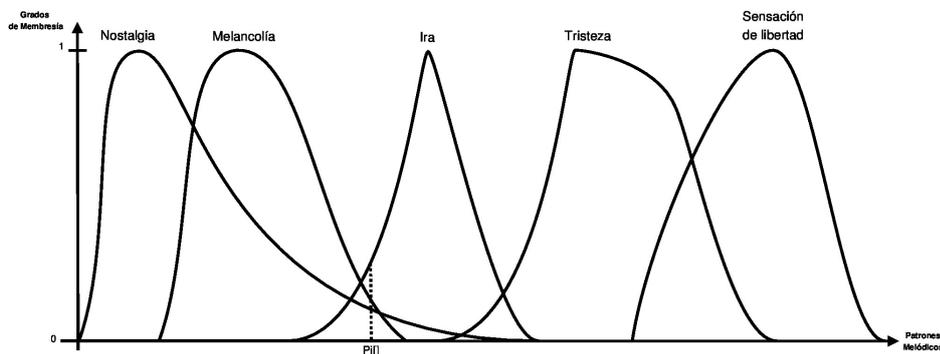


Figura 3.3: Funciones de Membresía para patrones melódicos con respecto a emociones

representan la gramática de contexto libre, a través de la cual se pueden generar las estructuras de la obra.

Para obtener esta gramática se combinan dos estrategias: Obtener reglas de los expertos, que permitan clasificar las partes, y a través de un mecanismo de *inferencia de gramática* obtener las reglas desde las composiciones.

Por ejemplo, para una gramática de la estructura de Johann Sebastian Bach, considerando el criterio de los músicos y patrones genéricos sería:

$\langle \text{Parte A} \rangle ::= \text{'Patrón 1'} \mid \text{'Patrón 2'} \dots$

$\langle \text{Parte C} \rangle ::= \text{'Patrón 9'} \mid \text{'Patrón 3'} \mid \text{'Patrón 10'} \mid \dots$

$\langle \text{Parte F} \rangle ::= \langle \text{Parte A} \rangle \mid \text{'Patrón 5'} \mid \langle \text{Parte B} \rangle \langle \text{Parte h} \rangle \mid \dots$

...

<Parte I> ::= <Parte D> | <Parte B> 'Patrón 6' | ...

Como se discutió en el marco teórico, existen varias abstracciones para la representación del conocimiento musical; sin embargo las presentadas en esta sección fueron escogidas de tal manera que exista una coherencia en el proceso de composición desde el punto de vista de melodías y fundamentada en la entrevista de los músicos, por los que podemos resumir este proceso en: extracción de patrones de secuencias (estilo de composición de melodías de músicos reales), generación de nuevos patrones (nuevo conocimiento generado para innovar en composición), asignación de emociones a los patrones bases y a los nuevos patrones (expresión de material musical con sentimientos particulares) y finalmente la organización de la composición (estructura de composición coherente para un oyente).

3.1.2 Algoritmos a aplicar

A continuación se describen las estrategias a ser utilizadas en la implementación de sistemas de composición e interpretación musical que utilicen la arquitectura propuesta.

Para generar las secuencias de notas utilizando las matrices de transiciones de Markov se puede utilizar el algoritmo descrito en [3.1](#). Existen otros

algoritmos como **Fordward-Backward**, **Viterbi** y **Baum–Welch**, estudiados anteriormente, que también podrían ser usados con pequeñas variaciones para ser adaptados a las cadenas de Markov de primer orden ya que su principal enfoque son los Modelos Ocultos de Markov.

Algoritmo 3.1: Algoritmo para generar el Siguiente estado

Input: Actual elemento de la secuencia

Output: Siguiente elemento de la secuencia

```

1 function GenerarSiguienteEstado (estadoActual)
2
3   guessState ← ObtenerNumeroRandom
4   LimiteMaximo ← 0,0
5   begin
6     for each neighboring state n of estadoActual do
7       LimiteMaximo ← LimiteMaximo + ( transition probability
8         from estadoActual to n )
9       if guessState is less or equal than LimiteMaximo then
10        return n
11      end
12    end

```

Adicionalmente, podrían aplicarse otras estrategias, recordemos que la matriz de transiciones puede ser trasladada a una representación de grafo dirigido y por tanto podemos aplicar otros algoritmos que permitan extraer secuencias, si los modificamos adecuadamente, según el problema que necesitemos resolver, ellos pueden ser **Best-First**, **Simulated Annealing** u otros que permitan realizar búsquedas heurísticas ya que los lineamientos dados por otros músicos podrían darnos pautas para generar funciones heurísticas sobre esta base de conocimientos con el fin de variar el estilo de composición

que surge de la combinación de los patrones.

Más allá de la utilización del conocimiento generado, podríamos también experimentar con la composición si cambiamos los patrones desde sus entes atómicos; es decir, notas y tiempos. Los músicos entrevistados componen mediante un proceso iterativo que lo consideran al mismo tiempo como un *proceso evolutivo* que mejora a medida que progresan en sus exploraciones; por lo tanto, podemos hacer uso de esta analogía con los **algoritmos genéticos**, y así lograr una composición e interpretación distinta e innovadora por parte de la máquina.

Para este caso nuestros individuos serían las secuencias, los genes estarán ligados a las notas musicales y duraciones producidas por el proceso Markoviano previamente descrito. Cada cromosoma es representado por cada una de estas secuencias ya sea de duraciones o de notas. Para las secuencias de notas, se utiliza una función de aptitud que permita clasificar y seleccionar los mejores cromosomas que contengan patrones similares a las melodías provistas por los músicos. Recordando, estas melodías fueron las mismas para generar las matrices de transiciones de las cadenas de Markov, que a su vez también pueden ser utilizadas para el entrenamiento de la estrategia que permita la clasificación.

Para las secuencias de duraciones la función de aptitud estaría limitada a

evaluar cada cromosoma sumando sus elementos o genes y verificando que sea igual a cierto valor previamente fijado, el cual da el ritmo de las melodías que están siendo evaluadas. Por ejemplo, si una melodía es compuesta a un ritmo de 4/4, lo que significa que en cada bar de la partitura debe existir 4 notas de 1/4 (Negra) cada una de ellas. Si la secuencia de duraciones consta de 5 bares al mismo ritmo, esto equivale a tener 20 notas de 1/4 (Negra) en total. Algo que es importante recalcar, es que las dimensiones de los vectores tanto de notas como duraciones deben coincidir.

En el ámbito de emociones humanas el panorama es complejo por el nivel de subjetividad en las personas; sin embargo, esta incertidumbre nos indica que la obra puede caer en diversos tipos de emociones como lo mencionaron los músicos. Esta idea es representada en la clasificación de los patrones.

Como hemos visto, una obra no puede pertenecer a una emoción en particular, la **lógica difusa** puede ser de gran ayuda para implementar sistemas difusos que generen composiciones con el carácter sentimental que deseemos y aplicar lo discutido anteriormente para lograr una composición más cercana al ser humano con el objetivo de que pueda proyectarse en ellas.

Para la utilización de las funciones de membresía obtenidas por el método mostrado en la sección anterior, se propone un procedimiento de *defusificación* para generar composiciones basadas en ese conocimiento emocional.

El proceso propuesto está enmarcado en el ámbito de una *composición en tiempo real* (improvisación) [10]. Para este caso en particular le llamaremos una **Composición Musical Humano-Máquina** ya que un músico interpreta acordes estableciendo la base armónica que el sistema utiliza para realizar un acompañamiento con melodías seleccionadas del conjunto de soluciones construido con el proceso de fusificación explicado anteriormente. Las entradas involucradas en este proceso son las siguientes:

Start Input Antes de iniciar el agente inteligente, el músico humano debe proveer los pesos de las emociones que desea transmitir, lo cual será posteriormente considerado para la búsqueda de los patrones melódicos. Estos valores representan la intención emocional que el compositor humano desea transmitir a una audiencia. Además, el tempo (BPM) y la nota clave (tonalidad) deben ser dadas. Por ejemplo, un músico podría desear interpretar una pieza con las siguientes emociones y sus intenciones (nivel de asociación emocional): *felicidad(10)*, *tristeza(90)* y *melancolía(75)*, tempo *80 BPM*, y utilizar la nota clave *C mayor*.

Real-Time Input Mientras el músico esta componiendo e interpretando, el agente artificial recibe las notas musicales generadas por el artista (*acordes*), y produce nuevas melodías en tiempo real, utilizando la

información los datos de esas notas conjuntamente con la base de conocimientos.

Para el proceso de composición, un metrónomo guía al compositor con el tempo correcto. En cada beat que el metrónomo marca, el sistema produce un patrón de melodía que es el resultado del proceso de composición. Este proceso utiliza las notas adquiridas del período de tiempo anterior entre el beat actual y su predecesor como se muestra en la figura 3.4.

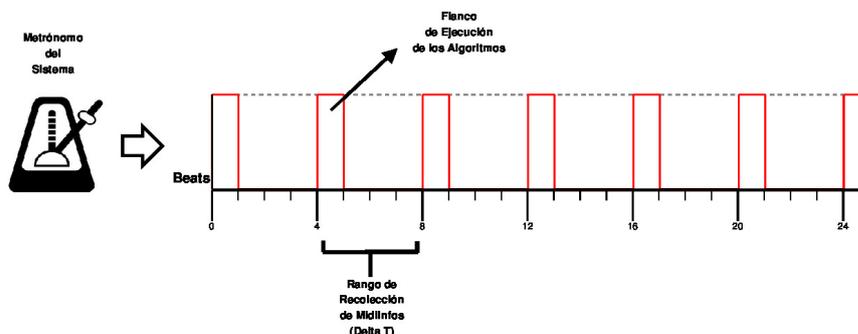


Figura 3.4: Modelo de Metrónomo para la Adquisición de Datos y la Ejecución de la Composición

Como se discutió anteriormente, no todas las notas de un patrón de melodía generado de tamaño n son ejecutadas, debido a la superposición de notas en cada beat. Esta superposición producirá disonancia si las notas reproducidas están todavía siendo interpretadas y el próximo acorde no es congruente con ese conjunto de notas [10]. Para resolver este problema, el patrón de melodía es únicamente ejecutado en un rango entre 1 y el *tamaño de entrada*. Por ejemplo, si tenemos un patrón de melodía con $n = 20$ y el sistema recibe una

entrada como esta: (48, 50, 52, 55), que traducida del MIDI sería (C3, D3, E3, G3), entonces el sistema dividiría ese patrón en 1, 2, 3 o 4 notas sin tener que reproducir las 20. Sin embargo, si el compositor interpreta muchas notas, todas las notas será reproducidas. En este caso la superposición no puede ser controlada; pero a niveles bajos de superposición, este comportamiento produce un comportamiento interesante que hace que el sistema genere armonías más que melodías en la composición; un efecto que no es disonante.

Para seleccionar el patrón de melodía que mejor se ajuste en cada beat, seguimos el procedimiento de *defusificación* propuesto a continuación:

1. Debido a que la base de conocimientos puede ser muy extensa, necesitamos una estrategia para encontrar la mejor solución acorde a las entradas. Por consiguiente, los patrones de melodías generados son organizados en *árboles binarios balanceados*.
2. En este enfoque, la base de conocimientos es estructurada en ne árboles binarios balanceados, tal que ne es el número de emociones involucradas para todos los patrones generados, donde todos estos patrones tienen un cierto grado de membresía para cada emoción. Por lo tanto, cada árbol representará una emoción donde las **claves** son los *niveles de asociación emocional* (pesos o grados de membresía) y los **valores** son los *patrones*

de melodías asociados con esa emoción. Aunque esta representación requiere algo de memoria extra debido a las claves, es una estrategia relevante debido a que ayuda a reducir el espacio de soluciones para encontrar los patrones correctos y reproducirlos, como lo ilustraremos más adelante.

Por ejemplo, si la base de conocimientos ha sido entrenada con tres emociones *felicidad*, *tristeza*, y *melancolía*, entonces tendremos tres árboles binarios balanceados como se ilustra en la figura 3.5. El correspondiente algoritmo de inserción toma el peso emocional por cada patrón en cada emoción y coloca el patrón de melodía en cada árbol, dependiendo de esos pesos, lo que significa que, si un peso para felicidad es 20.0, entonces un nuevo nodo es creado en el *Árbol de Felicidad* con una *clave* = 20,0, y, si ya existe un nodo con esa clave, entonces el patrón *P* es asociado a un conjunto de ellos con el objetivo de compartir el mismo nodo como se puede apreciar en la figura 3.6. Dado que la base de conocimientos tiene dos tipos de patrones de melodías ponderados con emociones, uno para *notas* y otra para *duraciones*, entonces hay que considerar la creación de seis árboles binarios balanceados.

3. Como se mencionó anteriormente, un operación importante es el proceso

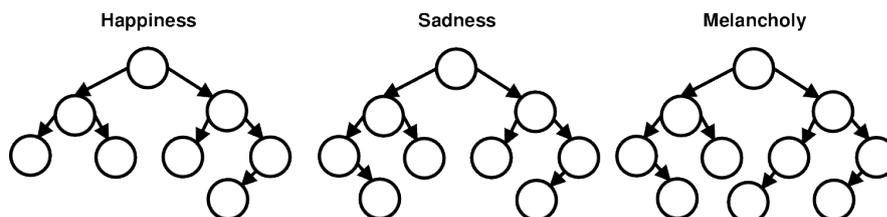


Figura 3.5: Ejemplo de Árboles para Emociones

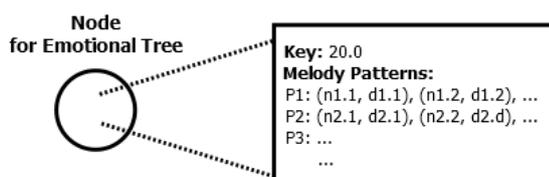


Figura 3.6: Ejemplo de la Estructura de un Nodo para un Árbol de Emoción

de **búsqueda**; principalmente debido a que este proceso definirá cuán rápido es alcanzado el objetivo.

Los pesos para las emociones que fueron provistos por los músicos en el *Start Input* podrían no estar registradas como claves en los árboles debido a que no apareció previamente en el entrenamiento, por ejemplo, si el músico ingresa *felicidad(10)*, *tristeza(90)* y *melancolía(75)*, y no hay una clave 10 en el árbol de felicidad, entonces tenemos que considerar las *claves más cercanas* como se muestra en la figura [3.7](#).

Las **claves más cercanas** representan los nodos que deben ajustarse a los siguiente requerimientos:

- Si el peso emocional es encontrado, entonces se tomarán todos los

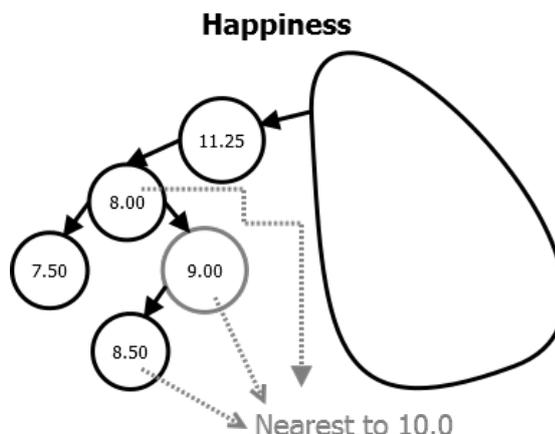


Figura 3.7: Claves más cercanas a 10 para un Árbol de Felicidad

patrones de melodías asociados con este nodo objetivo y sus nodos adyacentes, es decir, su padre y sus hijos.

- Si el objetivo no es encontrado, navegaremos el árbol hasta que una hoja nula (NULL) sea encontrada, subiremos a su padre y así como en el punto anterior, tomaremos todos los patrones de melodía de este nodo y de sus nodos adyacentes.

El objetivo es tener un espacio de soluciones reducido con el número de patrones relevantes a cada árbol (emoción) de manera independiente, los cuales son los más cercanos con respecto a los pesos que los músicos proveen en el *Start Input*. Este procedimiento puede ser ejecutado en el momento de iniciar el proceso de composición.

4. Cuando el proceso de composición está en marcha y el compositor

humano está interpretando, se reciben las notas musicales de un teclado MIDI(Real-Time Input). Estas entradas más los árboles con el espacio de soluciones reducido, son utilizados para ejecutar una melodía a través de los parlantes. Para obtener el patrón de melodía, el sistema itera sobre el espacio de soluciones reducido buscando patrones cuyas *notas* cumplan los siguientes dos criterios:

- El conjunto de emociones ponderadas Eh de tamaño ne las cuales son proveídas por el compositor humano al iniciar, es comparado con todas las emociones ponderadas Em para cada patrón que se encuentra en el espacio de soluciones reducido utilizando la ecuación [3.12](#) la cual es una *Distancia Euclidiana* para tomar en cuenta todas las emociones que son utilizadas.

$$Dhm = \sqrt{\sum_{r=1}^{ne} (Em_r - Eh_r)^2} \quad (3.12)$$

La estrategia de los árboles binarios nos brinda soluciones para cada emoción de una manera independiente, pero con las distancia entre todas las emociones, podemos decir que todas son utilizadas para el cálculo, lo que es un criterio que se debe cumplir en la Lógica Difusa ya que todos los conjuntos difusos participan en el

proceso con un cierto grado de membresía para los elementos en cuestión, en este caso los pesos para el conjunto difuso de emociones sobre los elementos que son los patrones melódicos.

El objetivo de este punto es encontrar el patrón de melodía que tenga la distancia emocional mínima y además que sea *consistente musicalmente* con las notas de entrada, como se explica a continuación:

- El patrón de melodía a escoger debe ser *musicalmente consistente* con la armonía (acordes) que el compositor humano está interpretando. Por lo tanto, utilizamos el siguiente criterio: *Si la primera nota en el patrón de melodía candidato es parte de la entrada que el compositor humano provee desde el teclado musical, entonces aquel patrón de melodía es consistente con la armonía recibida.* Podrían haber otros criterios heurísticos, sin embargo no queremos tener fuertes restricciones que inhiban la creatividad artificial de la solución.

Estos dos criterios son combinados por medio de una operación de conjunción *and*(\wedge) para formar una sola expresión y obtener el patrón de melodía objetivo basado únicamente en las *notas*. Para *duraciones* solo necesitamos aplicar el primer criterio.

5. Finalmente, el *patrón de melodía para notas* seleccionado y *patrón de melodía para duraciones* son fusionados para generar una nueva melodía que es convertida en sonidos a través un motor de síntesis para ser reproducida a través de los parlantes. Por lo tanto, del conjunto difuso de *Emociones*, se obtiene un valor puntual que es nuestro nuevo patrón de melodía.

Para la estructuración de la obra hemos propuesto utilizar gramáticas de contexto libre ya que a la composición final hay que darle un sentido y un modelamiento ordenado con el objetivo de establecer un flujo poético en la obra así como los músicos lo mencionaron en la entrevista. Para lograr esto las *etiquetas de sección* deben ser significativas, por ejemplo: preludio, interludio, etc. No necesariamente pueden fundamentarse en la teoría musical, lo importante es que se entienda su intención dentro de la estructura de la obra.

Un enfoque para la utilización de las reglas de la gramática podría ser los **L-Systems**, donde las iteraciones de los reemplazos en la gramática dictan la forma de la composición y pueden llegar a formar fractales enfocados a una ejecución paralela de melodías donde cada una de ellas posea su estructura, pero hay también un conjunto de las mismas que forma la estructura al ejecutarse concurrentemente, lo cual generaría una obra interesante basada

en el principio del *contrapunto*³ explicado por los músicos y descubierto por Johann Sebastian Bach, además esto se acerca a su modelo de composición por capas para formar la obra final.

3.2 Síntesis de Sonido basado en emociones

3.2.1 Representación del Conocimiento

Una característica sumamente importante de la música experimental es la composición de sonidos en las piezas creadas por los músicos. Es allí donde la aplicación de la *síntesis de sonido* desempeña un papel primordial en la generación del timbre musical.

Los equipos para lograr generar sonidos se denominan *sintetizadores*, cuyos componentes básicos son osciladores, filtros y envolventes. La construcción de un sonido depende de la configuración de los elementos basados en estos componentes y de las estrategias de síntesis a aplicar tales como síntesis aditiva, sustractiva o por wavetables como se discutió en el marco teórico.

Estas configuraciones pueden llegar a ser muy complejas para ser analizadas desde sus elementos básicos ya que los valores de regulación son

³Ejecución de melodías distintas que forman una armonía al combinarse.

parámetros continuos que poseen una alta influencia en el timbre del sonido por más mínimo que sea el cambio; es decir, si una perilla que controla un filtro Paso-Bajo se mueve unos pocos milímetros, el sonido puede cambiar significativamente.

Para reducir la complejidad de la representación de un modelo que considere la síntesis de sonido basado en emociones, tomamos en cuenta las **configuraciones de síntesis**, más que cada elemento que participe en la generación de un sonido. Esto significa de manera concreta que, para un sonido particular, existirá un archivo que represente la configuración de un sintetizador que lo concibe. Este archivo tendrá la información de los valores numéricos asignados a los componentes (osciladores, filtros y envolventes) para ese sonido.

Una vez seleccionado el elemento atómico que será utilizado en la representación general (archivos de configuraciones de síntesis), podemos establecer la arquitectura para la síntesis de sonido basado en emociones ilustrado en la figura [3.8](#).

Los componentes de esta arquitectura son explicados a continuación:

- **Entradas:** Las entradas de este modelo constan de:

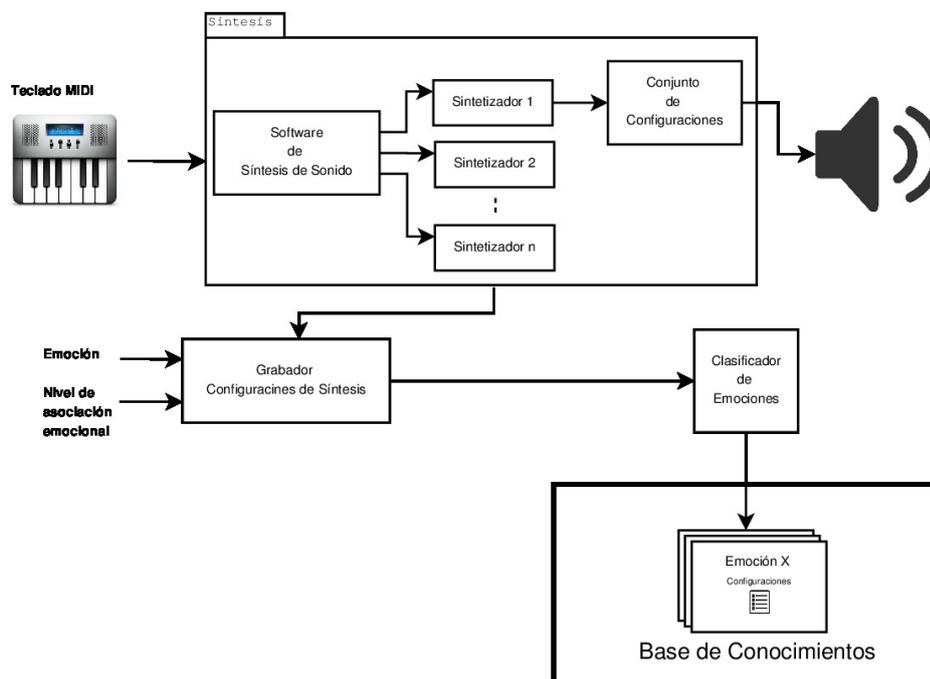


Figura 3.8: Arquitectura de Representación del Conocimiento para Síntesis de Sonido Basado en Emociones

- Eventos MIDI desde un teclado musical, el cual puede constar de notas musicales o datos de controles tales como perillas o deslizadores.
- Emoción, que es una etiqueta para los sonidos generados por las configuraciones establecidas en los sintetizadores de software.
- Nivel de asociación emocional, que representa un porcentaje de correspondencia entre la emoción proporcionada y el sonido que se desea etiquetar; por ejemplo, un músico puede catalogar un sonido con un 25 % de melancolía y un 75 % de nostalgia.

•**Módulo de Síntesis:** Este módulo es el componente de la arquitectura que recibe las entradas del teclado MIDI para poder generar sonidos como resultado de los procesos de síntesis que ocurren en su interior. El núcleo de su infraestructura es el *software de síntesis de sonido* el cuál posee el motor de síntesis para la producción de sonidos y las interfaces que se comunican con el usuario que lo utiliza. Estas interfaces vienen a ser abstracciones de *sintetizadores* conocidas por los artistas que practican música electro-acústica y sirven para manipular directamente los componentes de síntesis que construyen el timbre. El software de síntesis puede estar compuesto de N sintetizadores, los cuales pueden estructurar archivos de configuraciones para almacenar los parámetros de creación de un sonido. Estas configuraciones poseen la información necesaria para que el motor de síntesis construya el sonido y lo envíe a un dispositivo de salida, en este caso, parlantes.

•**Grabador de Configuraciones de Síntesis:** Este componente recibe tres tipos de entradas; las configuraciones de los sintetizadores provenientes del módulo de síntesis, las emociones que corresponden a esas configuraciones (sonidos), y el nivel de asociación entre esas emociones y los sonidos. Básicamente este módulo almacena la relación entre emociones y sonidos con el objetivo de alimentar al clasificador de emociones que será explicado a continuación.

- **Clasificador de Emociones a través de Lógica Difusa:** Este módulo actúa de manera similar al clasificador de composiciones musicales discutido en la sección anterior. Recibe las relaciones entre emociones y sonidos, además de los niveles de asociación; de tal manera que construye funciones de membresía para representar el conocimiento.

Nótese que este modelo considera la parte de síntesis únicamente a nivel de software. Esto es por la dificultad de registrar las configuraciones de sintetizadores físicos y posteriormente utilizarlas para que un software de composición musical manipule estos entes.

3.2.2 Algoritmos a aplicar

En este caso se ha optado nuevamente por un enfoque de **lógica difusa**. Los niveles de asociación entre emociones y sonidos son los indicadores que nos permiten representar el rango de emociones explicados por los músicos, pero en este caso para los sonidos.

Las estrategias a aplicar son las relacionadas a conjuntos difusos siguiendo los pasos para el *razonamiento en situaciones inciertas* exclusivo para sistemas basados en lógica difusa descrito en la sección [2.6.2.3](#).

Cabe destacar que este enfoque de configuraciones de síntesis puede ser

modificado de tal manera que se aumente el grado de complejidad en el análisis; es decir, considerar los componentes estipulados en las configuraciones más que las configuraciones completas en sí. En ese caso se pueden aplicar algoritmos que realicen combinaciones entre los distintos componentes de diferentes configuraciones y así obtener sonidos generados por la máquina mediante agentes inteligentes, a diferencia del enfoque propuesto, que simplemente organiza y clasifica los sonidos, pero no los construye debido a que el enfoque principal de este trabajo se centra en composición musical.

3.3 Integración e interacción entre las estrategias seleccionadas

Los modelos de composición musical y de síntesis basados en emociones, pueden ser combinados de tal manera que se estructure una arquitectura semejante a la mostrada en la figura [3.9](#), la cual establece la abstracción completa para la adquisición y representación del conocimiento implícito en las obras de los músicos experimentales.

En este modelo general se fusionan dos de los módulos que fueron propuestos independientemente para cada caso; el grabador y el clasificador

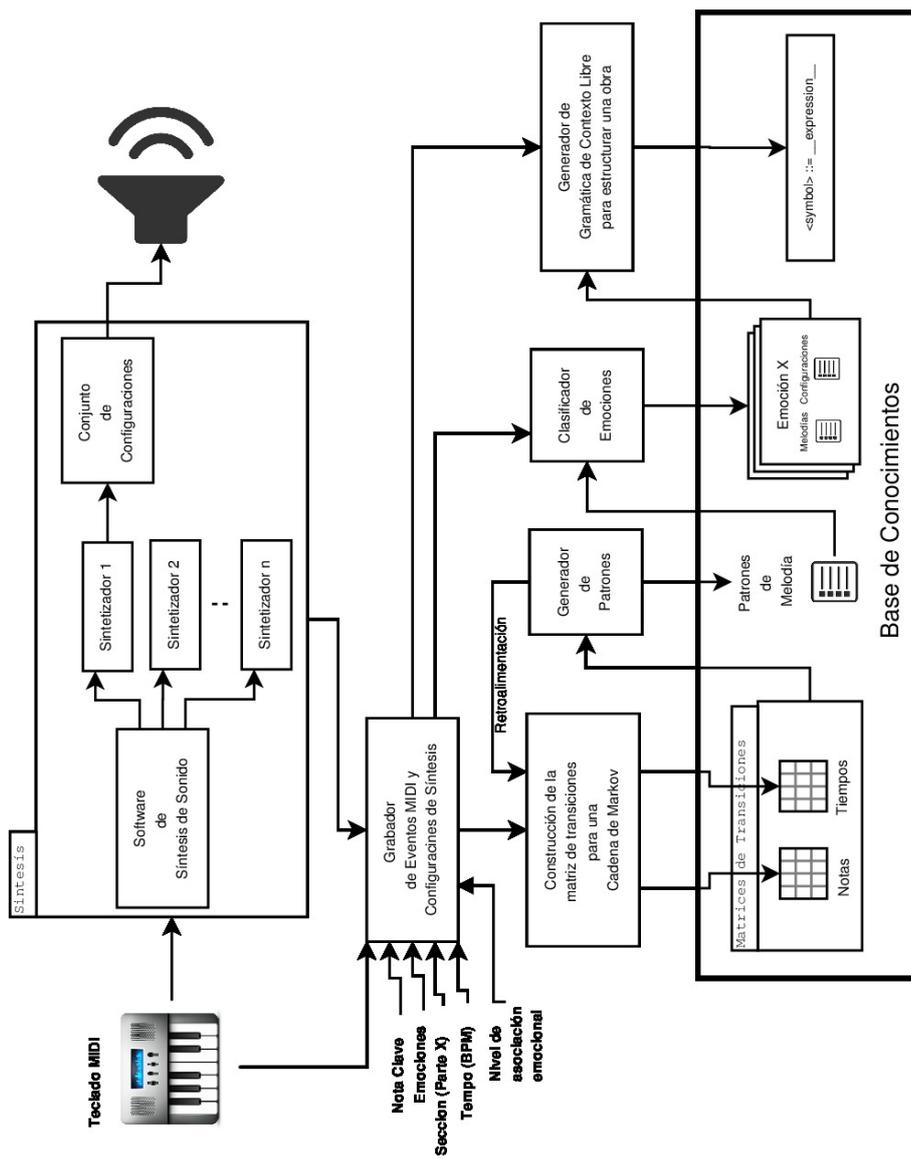


Figura 3.9: Arquitectura Completa de Representación del Conocimiento Musical

de emociones. Ahora, el grabador considera los eventos MIDI para la composición y las configuraciones de síntesis para alimentar a toda la base de conocimientos. El clasificador de emociones permite organizar tanto los patrones melódicos como las configuraciones de síntesis en grupos de emociones de tal manera que se asignen los sonidos adecuados a las composiciones de melodías para emociones particulares o rango de emociones dependiendo del razonamiento sobre los conjuntos difusos.

Parte de esta arquitectura será la base para el piloto propuesto en este trabajo y soportará modificaciones que permitan su utilización en un ambiente en tiempo real. El diseño de este sistema se detallará en el siguiente capítulo.

CAPÍTULO 4

4. DISEÑO DE LA INFRAESTRUCTURA TECNOLÓGICA DEL SISTEMA

El sistema que permite probar el modelo completo propuesto en la figura 3.9 de la sección 3.3, es un **sistema concurrente en tiempo real** que compone melodías de manera paralela considerando la *base de conocimientos* previamente entrenada y fundamentada en las abstracciones discutidas en la sección anterior. La tarea esencial de este sistema es de **acompañamiento musical**, ya que es guiado por la *interpretación* de uno o varios músicos que provean, en vivo y de manera simultánea, entradas al sistema; de tal manera que éste produzca melodías paralelas acordes a la guía proporcionada que, además de la interpretación, recibiría la *intención emocional* del músico (indicación de la emoción que se desea transmitir en ese instante) para

ejecutar los timbres de sonidos adecuados y así producir la obra musical. En la figura 4.1 se muestra un diagrama de bloques que describe de manera fundamental el flujo del sistema basado en las consideraciones mencionadas.



Figura 4.1: Diagrama de Bloques del Sistema

Nótese que la obra musical es el producto de los dos entes; humano y máquina, donde la máquina depende del músico para producir su aporte en la obra. El diseño del sistema en sus dos componentes esenciales; hardware y software, será explicado en detalle en este capítulo junto con el conjunto de pruebas a realizar sobre él; esto con el fin de evaluar parámetros importantes tales como *rendimiento* y *efectividad* para las tareas que realiza.

4.1 Diseño del componente de Hardware

4.1.1 Elementos físicos de la infraestructura

4.1.1.1 Controladores MIDI

Los controladores MIDI a utilizar son **teclados musicales**, ya que proveen una mejor interacción para la transmisión de información musical por su topografía que asocia directamente el ente físico con las notas musicales; además, es el equipo utilizado por excelencia en el ámbito de síntesis de sonido y control MIDI.

Lo ideal es que el teclado tenga un número de octavas aceptables para una interpretación musical cómoda. Un teclado de cinco octavas puede cumplir esta característica [14]; sin embargo, por el hecho de que el sistema recibe más de una entrada MIDI, es posible conectar otros dispositivos con menor número de teclas para tareas específicas tales como melodías cortas, acordes puntuales, o ejecución de efectos específicos.

La propuesta del teclado no es estricta, ya que el sistema recibe mensajes MIDI independientemente del equipo utilizado, pero es importante conocer que los mensajes que envíe el dispositivo a utilizar sean notas musicales, ya

que existen otros controles; como perillas y deslizadores, que poseen una codificación distinta en los bytes que llevan el mensaje. Estos controles en cambio son útiles para la configuración de los valores de componentes en los sintetizadores de software, ya que ellos permiten un mapeo entre el ente virtual y el físico, o en todo caso para entradas adicionales al sistema, como por ejemplo la selección de la intención emocional del músico que lo manipula.

4.1.1.2 Interfaces de Audio

Las interfaces de audio son elementos de hardware que permiten la comunicación entre equipos de audio y un computador. Permiten la conversión de las señales de audio a datos digitales para que puedan ser manipulados por una máquina [61]. Esto significa que son básicamente un *tarjeta de sonido* con mayor capacidad de procesamiento y opciones de interacción entre equipos.

Las interfaces de audio que reciben señales MIDI son útiles en el caso de necesitarse la conexión de múltiples dispositivos. Sin embargo, los controladores MIDI actuales incluyen conectores USB en vez de los terminales estándar discutidos en el marco teórico, lo cual permite obviar la utilización de una interfaz de audio debido a la posibilidad de conectar

controladores MIDI directamente a la máquina, en un número similar a la cantidad de puertos USB que ésta posea. A pesar de ello, la capacidad de procesamiento de una interfaz permite que las señales viajen a una velocidad aceptable con el objetivo de evitar retardos que afecten una interpretación musical, aunque una excelente tarjeta de sonido puede realizar este trabajo.

En el sistema se incluyen un interfaz de audio como medio entre los controladores MIDI de teclado y la máquina en donde correría la aplicación, o en su defecto, si la máquina es lo suficientemente eficiente en procesamiento de audio, puede ser obviado y realizar las conexiones directas.

4.1.1.3 Parlantes y otros dispositivos de salida

Para poder apreciar el resultado de la composición necesitamos *escucharlo*, por este motivo un equipo esencial es el parlante.

El módulo de síntesis explicado posteriormente en la sección [4.2.10](#) es el encargado de gestionar el hardware al cuál se dirigen las señales producidas por los algoritmos aplicados por el motor de síntesis. El hardware gestionado son los **parlantes** del computador o en todo caso interfaces de audio de salida que permitan tomar las señales digitales y traducirlas a señales analógicas distribuidas a un sistema de parlantes, si se requiere de mayor

complejidad.

La señal de salida del sistema no necesariamente se puede limitar a sonido; es decir que es posible utilizarla con otros dispositivos de características visuales que permitan asociar el sonido con efectos gráficos con el objetivo de brindar un espectáculo sonoro-visual a la audiencia que está escuchando la obra, por ejemplo luces cuyos colores reaccionen con las frecuencias de la señal de audio.

4.1.2 Ensamblaje de los elementos hacia el computador

Finalmente, los elementos físicos interactúan con un máquina capaz de ejecutar el software del sistema, de esta manera se puede obtener el diseño de la infraestructura de hardware mostrada en la figura [4.2](#) de la cual el sistema depende para poder producir los resultados.

En esta infraestructura se conectan a la interfaz de audio N controladores MIDI que son utilizados por los músicos que guían el sistema, esas señales son manipuladas por el computador que ejecuta los algoritmos de composición para finalmente obtener el sonido de la obra mediante un sistema de parlantes que puede ser tan complejo como se requiera. Nótese que a diferencia del diagrama de bloques de la figura [4.1](#) presentado al inicio del capítulo, no existe

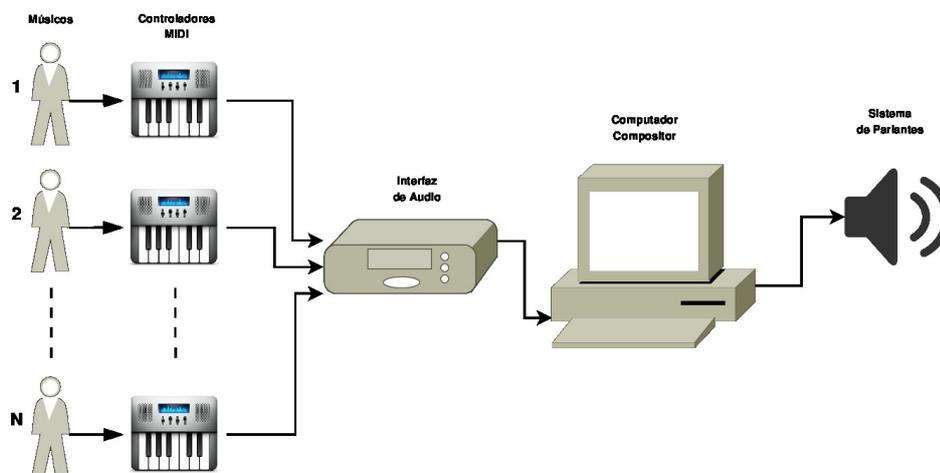


Figura 4.2: Infraestructura de Hardware del Sistema

una comunicación directa entre la interpretación en los teclados con respecto al sistema de parlantes, esto es debido a que el computador reproduciría no solo el resultado de la composición de la máquina, sino también la interpretación de los músicos porque todo el proceso de síntesis es mediante software, a menos que los controladores produzcan sonidos por ellos mismos mientras los mensajes MIDI son enviados.

4.2 Diseño del componente de Software

4.2.1 Adquisición de datos

El sistema recolecta los datos brindados por los controladores MIDI de manera serial siguiendo el estándar de mensajes expuesto en la sección [2.3.3](#), además

de entradas adicionales provistas por los músicos, ya sea que el sistema se encuentre en *modo de entrenamiento* o en *modo de producción*.

La figura 4.3 muestra el módulo de adquisición de datos, donde se pueden apreciar los dos tipos de entradas que provienen del exterior; los mensajes MIDI y los datos adicionales proveídos por el usuario. Las salidas se componen de abstracciones que son utilizadas por el sistema para la aplicación de los procesos de composición y síntesis.



Figura 4.3: Módulo de Adquisición de Datos del Sistema

Este módulo se comunica directamente con los controladores MIDI con el objetivo de extraer datos relevantes de la comunicación tales como: la cantidad de dispositivos conectados, la identificación de cada dispositivo (nombres o códigos) y el mensaje MIDI. El mensaje MIDI es representado por una colección de tres bytes que al llegar al módulo es almacenado en una estructura de datos intuitiva que almacena la nota y velocidad por cada colección para su posterior uso, la cual denominaremos `MidiInfo`. A partir de esta estructura se inicia un proceso concurrente de distribución de entradas para alimentar al sistema, el cual es explicado posteriormente en la sección 4.2.3.

4.2.2 Almacenamiento del Conocimiento

Para el almacenamiento de datos en el disco, el sistema sigue el modelo establecido en la figura 4.4. La gestión del almacenamiento ocurre en dos procesos distintos. El primero es el *entrenamiento*, en el cual los datos que contribuirían a la base de conocimientos son guardados desde la memoria al disco al finalizar el proceso. El segundo es la *producción*, donde entra en funcionamiento el sistema de composición en tiempo real que arrancaría después de cargar en memoria la base de conocimientos. Cabe destacar que en *producción* es posible guardar datos si el sistema implementa un módulo de entrenamiento mientras el proceso de composición algorítmico está siendo ejecutado.

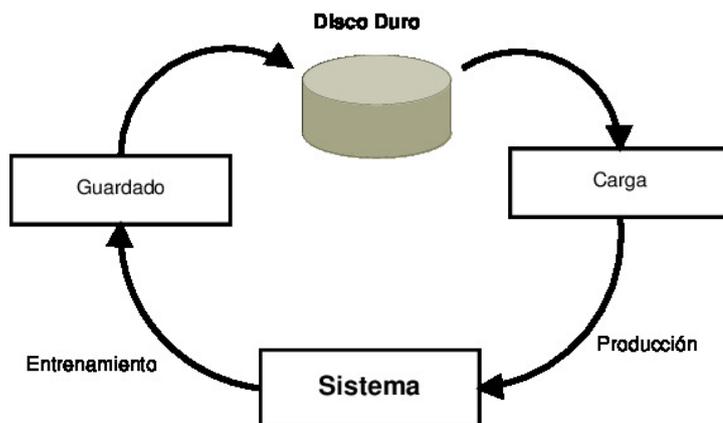


Figura 4.4: Modelo de Almacenamiento de Datos del Sistema

El modelo abstracto de representación de la base de conocimientos se encuentra en la figura 3.9 de la sección 3.3. Básicamente consta de cuatro

secciones independientes que son: las matrices de transiciones, un listado de los patrones de melodías, un conjunto de emociones donde cada emoción tiene su propia lista de patrones de melodía con configuraciones de síntesis, y expresiones de gramática BNF.

Las matrices y los listados son almacenados en archivos de texto plano por su simplicidad estructural. El conjunto de emociones es más complejo; su almacenamiento es dado por dos tipos de archivos, el primero es un listado de patrones de melodías que están etiquetadas con emociones y su nivel de asociación en la misma línea de texto, el segundo es también un listado con las mismas características, pero en este caso los elementos principales son referencias (direcciones en el computador) a los archivos de configuraciones de síntesis. Para el caso de gramáticas se hacen uso de tablas de símbolos almacenadas también en archivos.

Estos archivos son procesados en conjunto para ser almacenados en memoria, debido a que los tiempos de acceso son mínimos en comparación a las peticiones de disco; pero incurre en un costo en RAM que puede ser no aceptable si existe una gran cantidad de datos. En esos casos es posible utilizar una estrategia de carga selectiva que permita recuperar los datos de disco en momentos oportunos durante la interpretación.

4.2.3 Distribución de entradas a procesamientos concurrentes

Como se discutió anteriormente, el módulo de adquisición de datos se encarga de recibir los mensajes MIDI y transformarlos en estructuras de tipo `MidiInfo`. Durante ese proceso de adquisición, se verifica por cada colección de bytes MIDI que los mensajes pertenezcan a notas musicales y no a otros controles como perillas o deslizadores; además, cada vez que estas colecciones llegan al módulo, son almacenadas en un contenedor al cual llamaremos `MidiLine`. Pueden haber tantos `MidiLines` como dispositivos conectados.

Considerando que este sistema actúa en tiempo real, un nuevo `MidiLine` es construido por un período de tiempo dado; es decir, si el sistema está configurado para ejecutar un proceso cada cuatro beats (un bar); entonces, antes de que se cumpla ese tiempo se habrán recolectado M cantidad de `MidiInfos` para construir un `MidiLine`, y en el momento del flanco que anuncia el cuarto beat, se ejecutan los algoritmos de composición para generar las salidas en base al `MidiLine` de ese período. Luego, ese `MidiLine` se elimina para darle paso a uno nuevo en el siguiente ciclo.

El proceso descrito anteriormente es extrapolado para un sistema de N dispositivos MIDI que generen entradas, de tal manera que, por cada dispositivo haya un `MidiLine`. En ese caso los contenedores son

construidos de manera concurrente; es decir, por cada dispositivo existe un *hilo* (thread) que almacena `MidiInfos` y por ende surgirán N `MidiLines`. Para que los algoritmos se ejecuten, todos los `MidiLines` deben estar listos, por lo tanto, es necesario *sincronizar* estos procesos de llenado de tal manera que el flujo del sistema continúe adecuadamente. Este proceso es ilustrado gráficamente en el diagrama de actividades de la figura 4.5.

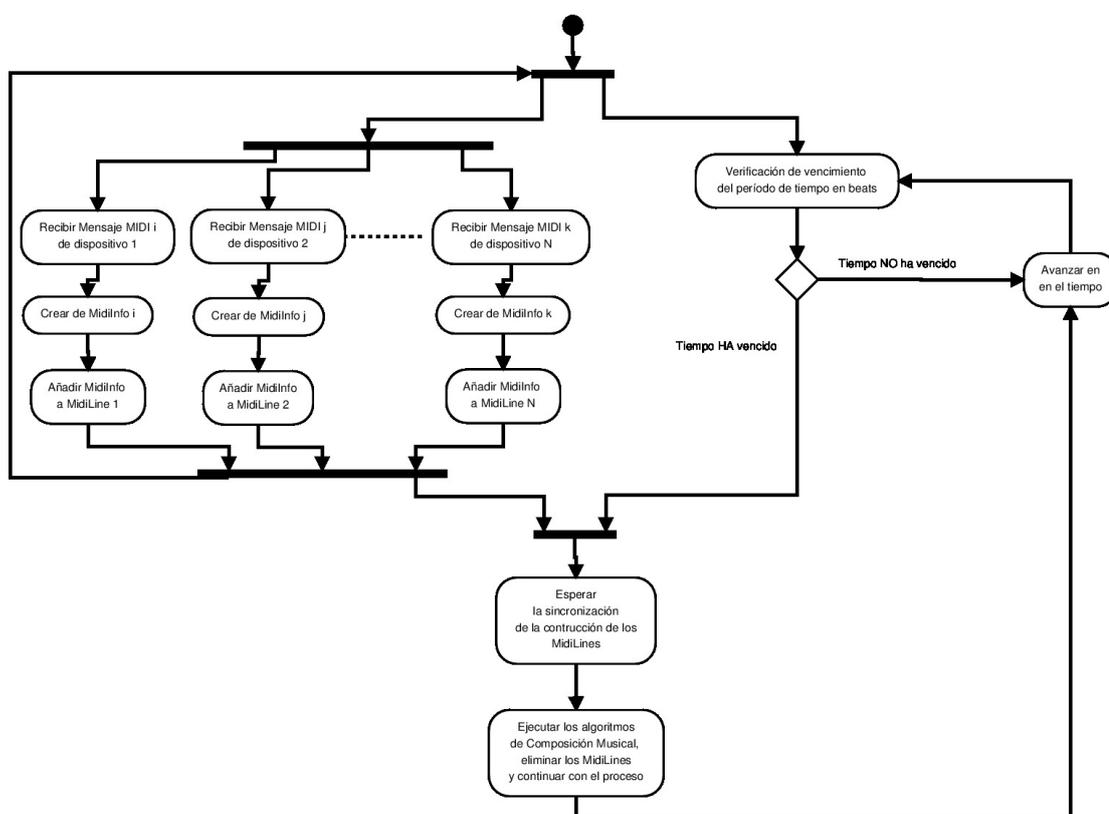


Figura 4.5: Diagrama de Actividades de la Distribución Concurrente de Entradas al Sistema

4.2.4 Procesamiento concurrente temporizado

Este componente de software sigue los lineamientos de un sistema concurrente en tiempo real, cuya restricción de tiempo es dada por un **metrónomo** que marca el *tempo* que el músico ajusta para su obra musical dado en beats por minuto (BPM).

Los resultados producidos por este sistema (melodías concurrentes) son sincronizados con el metrónomo para que la obra esté ajustada en el tiempo musical adecuado; el sonido producido por el metrónomo es escuchado por los músicos de tal manera que también ellos se sincronicen con la pieza y así evitar fallas rítmicas.

La ejecución de los algoritmos de composición musical sobre la base de conocimientos se da cada vez que se cumple un tiempo ΔT que podría estar ajustado con un intervalo de cada beat o cada cuatro beats (un bar) que son medidas naturales para que ingrese una frase melódica en la composición, aunque el número de beats puede cambiar dependiendo de la estrategia de composición a aplicar. En la figura [4.6](#) se muestra un ejemplo de la ejecución del sistema basado en el metrónomo a través del tiempo. En este caso los algoritmos se ejecutan cada cuatro beats y por lo tanto ese es el rango de tiempo ΔT que el sistema aprovecha para construir los `MidiLines` en base a

los MidiInfos recolectados.

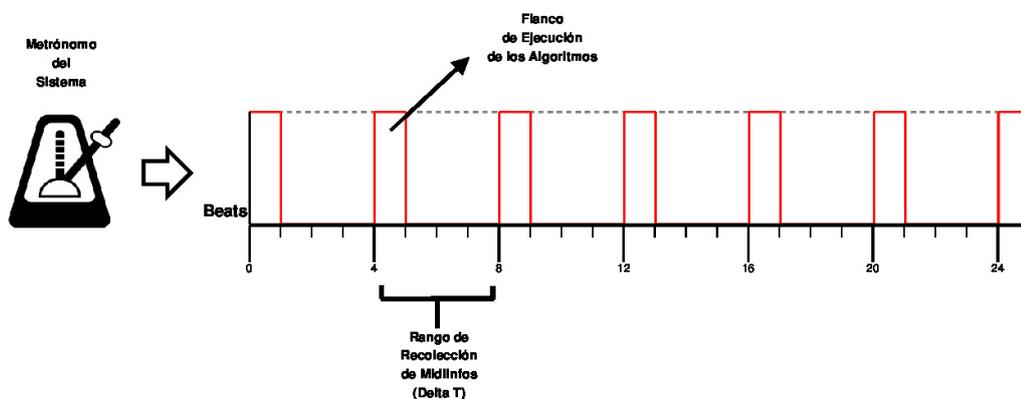


Figura 4.6: Ejecución del Sistema restringida a un Metrónomo

En el ámbito concurrente, considerando que las salidas de los algoritmos son no determinísticas, cada hilo puede calcular una melodía utilizando todas las estrategias con diferentes sonidos para que puedan ser identificadas, estos hilos vendrían a ser **capas** de música que se agregan a la obra y son denominadas en el ámbito de producción musical como **pistas**(tracks). También, se pueden reproducir estas pistas concurrentes asignando una estrategia distinta para evaluar la aportación independiente de los módulos de la base de conocimientos; por ejemplo, a más de la pista que el músico real interpreta en ese instante, otra pista puede calcular una melodía basada solo en Cadenas de Markov, mientras que una adicional puede utilizar Algoritmos Genéticos más Cadenas de Markov; es decir que se puede realizar un proceso incremental de producción de material musical dependiendo de las combinaciones de las estrategias sobre la base de conocimientos. Una representación gráfica de las salidas concurrentes del sistema se puede

apreciar en la figura [4.7](#).

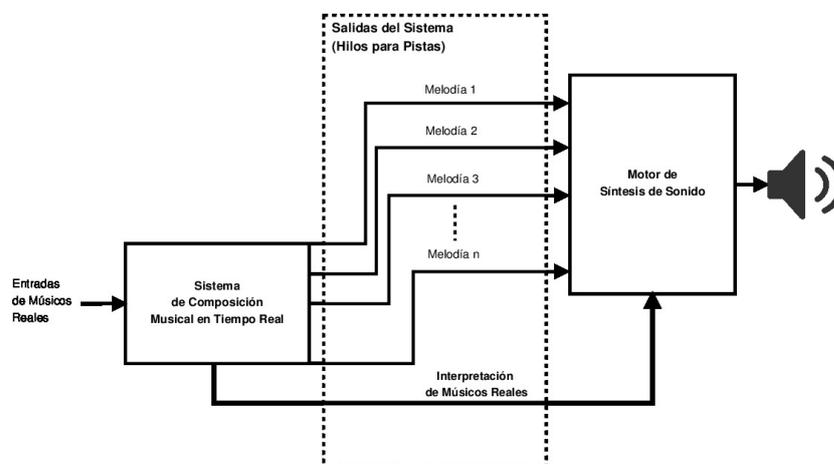


Figura 4.7: Hilos que representan las melodías producidas por el sistema

4.2.5 Estrategias de composición y síntesis de audio aplicadas en tiempo real

Con estrategias nos referimos a *algoritmos* que participarán en el proceso de composición y síntesis. Estos algoritmos ya fueron discutidos de manera general tanto en el marco teórico como en las secciones [3.1.2](#) y [3.2.2](#) para las representaciones del conocimiento seleccionadas.

La aplicación de estas estrategias es dada en tiempo real, es decir que serán ejecutadas constantemente cada ΔT período de tiempo, por lo que surgen ciertas restricciones que se deben cumplir para el correcto funcionamiento del sistema.

La primera restricción es el **tiempo de ejecución**, el cuál debe ser el mínimo para evitar retardos en la salida de los resultados, ya que como conocemos, el sistema está regido por un tiempo musical dado por un metrónomo virtual y la reproducción debe ser sincronizada con la mayor exactitud posible. Si las salidas se reproducen en un tiempo distinto al dictaminado por el músico, no habrá coherencia en el ritmo, y por lo tanto se perderá totalmente la estética de la obra.

Para enfrentar este problema se consideran varias situaciones que ayudan a resolverlo. En primer lugar, las entradas de los algoritmos son las notas que el músico interpreta en el rango de tiempo ΔT , por lo tanto la cantidad de datos de entrada no es abrumante como si se tratara de toda una pieza, a menos que exista un gran número de controladores MIDI que proporcionen datos al mismo tiempo. En segundo lugar, el *motor de síntesis* está separado de la aplicación, se ejecuta en segundo plano (background), lleva su propio control del tiempo dado por el metrónomo, recibe únicamente una cadena de caracteres que describe la melodía a reproducir, y produce la melodía en el tiempo exacto; bajo estas condiciones, el sistema de composición puede retrasarse en producir esa cadena de caracteres que será enviada al motor debido a que las melodías compuestas serán reproducidas en el tiempo correcto; pero, puede que haya un desfase de b número de beats que haga que la melodía no sea un tanto coherente con la base armónica que los

músicos están interpretando; en consecuencia el objetivo es tratar de que b tienda a cero.

La segunda restricción es la **efectividad del sistema con respecto al número de datos de entrada**; es decir, en cada período de tiempo se recibe una M cantidad de notas donde M puede llegar a tener valores muy pequeños, e incluso llegar a cero, lo cual deja sin entradas a los algoritmos para ese período de tiempo. En esos casos se puede llevar un historial de la interpretación musical que permita decidir la producción de las salidas, por ejemplo, se podría definir una regla indicando que, si no hay datos de entrada en el período t , entonces componer en base a las entradas del período $t - 1$, y así sucesivamente hasta encontrar datos. En el caso de no haberlos, no ejecutar las estrategias.

Otro inconveniente son los **cálculos exhaustivos** de los algoritmos, por ejemplo, aplicar el algoritmo genético puede llegar a sobrecargar el procesamiento del sistema si lo configuramos para que se ejecute un gran número de generaciones. Una solución a esto podría ser el *pre-cálculo* de soluciones que aporten a la composición. Considerando el mismo ejemplo del algoritmo genético, podemos generar en el modo de entrenamiento los patrones melódicos para que simplemente sean escogidos en tiempo real dependiendo de la intención emocional y las notas de entrada, lo que libera

significativamente la carga en el procesamiento.

En el caso de la síntesis de sonido, el motor de síntesis ya es en sí un sistema en tiempo real que recibe las configuraciones de sonidos y las melodías a reproducir del proceso de composición. En este caso, es importante que el tiempo del motor esté en sincronía con el del sistema para evitar desfases no deseados.

4.2.6 Módulo de entrenamiento del sistema

Este módulo permite construir la base de conocimientos a partir de las interpretaciones melódicas de músicos reales. Recopila los datos de los controladores MIDI y demás entradas de composición, para estructurarlos en las abstracciones plasmadas en la figura 3.9 de la sección 3.3. Esta arquitectura se convertiría en el diagrama de bloques básico para este módulo.

Aunque en la arquitectura se muestra el entrenamiento como un único proceso y se da a entender que el sistema se lo hace mientras el músico interpreta la obra; no necesariamente debe ser así en esencia. Esto significa que este módulo es dividido en dos partes; la **grabación** y el **entrenamiento**. El criterio para dividirlo de esa manera radica en el hecho de que estas

actividades pueden hacerse en dos ocasiones distintas; es decir, un músico puede grabar la obra en archivos de formato MIDI junto con las configuraciones de síntesis de la pieza y posteriormente en otro instante se puede entregar ese material al sistema para que alimente su base de conocimientos. Por lo tanto, las actividades de este módulo del sistema considerando esta particularidad, se muestran en las figura 4.8(a) y 4.8(b) para grabación y entrenamiento respectivamente.

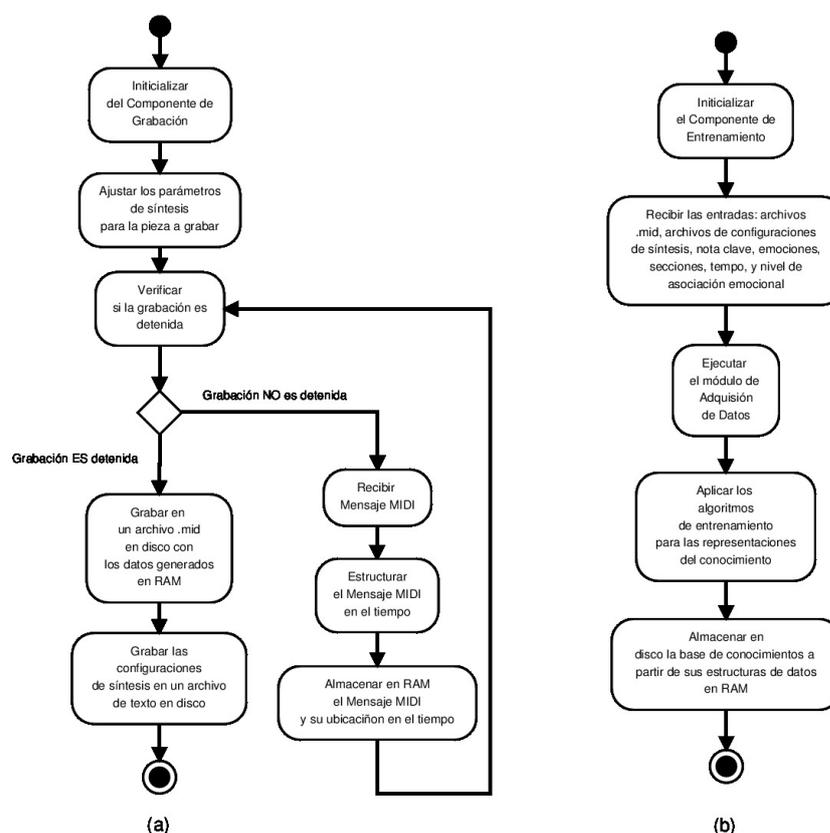


Figura 4.8: (a) Diagrama de Actividades para el Proceso de Grabación del Sistema. (b) Diagrama de Actividades para el Proceso de Entrenamiento del Sistema

Nótese que el módulo de adquisición de datos en la figura 4.8(b) no actúa

directamente sobre controladores MIDI, ya que como únicamente se encarga de los mensajes, entonces puede obtenerlos de archivos MIDI de extensión .mid que pueden ser reproducidos en cualquier máquina debido a que son parte del estándar MIDI.

El modelo de entrenamiento propuesto no es un sistema concurrente en tiempo real de manera estricta, ya que el componente de grabación es el único que procesa en tiempo real, pero el componente de entrenamiento procesa de manera lineal. Esto es debido a que la tarea consiste en únicamente alimentar a la base de conocimientos, lo cual se lo puede hacer de manera serial. Sin embargo, en el caso en que se requiera un entrenamiento donde varios controladores MIDI son utilizados al mismo tiempo, entonces el componente de grabación debe ser capaz de construir *pistas* (tracks) para que se fusionen en un solo archivo .mid al final del proceso; luego, en el componente de entrenamiento se separan esos tracks y se los procesa como si fueran piezas distintas, por lo tanto sigue siendo innecesario que el componente de entrenamiento sea concurrente en tiempo real.

El componente de grabación es fundamentalmente un *secuenciador MIDI* (ver sección [2.3.3](#)) y por lo tanto el diseño del sistema no lo considera a profundidad.

Conociendo hasta ahora los componentes que interactúan en el diseño del sistema completo, el diagrama de clases del módulo de entrenamiento se ilustra en la figura [4.9](#).

De este diagrama podemos destacar las siguientes clases:

MidInfo: Permite almacenar los datos obtenidos de los bytes MIDI (`note` y `velocity`) además de su duración en la línea de tiempo (`duration`).

KnowledgeRepresentation: Es un clase abstracta cuyos hijos son especializados en las cuatro representaciones propuestas en la base de conocimientos: cadenas de Markov (`MarkovKnowledgeRepresentation`), algoritmos genéticos (`GeneticKnowledgeRepresentation`), lógica difusa (`FuzzyKnowledgeRepresentation`), y gramáticas de contexto libre (`FuzzyKnowledgeRepresentation`). Cada una con sus estructuras de datos que permiten utilizarlas en el sistema, como por ejemplo, matrices de transiciones para cadenas de Markov.

AIMusicianTraining: Es una clase abstracta cuya estructura está basada en una combinación de los patrones de diseño *Strategy* y *Composite*, dado que es un conjunto de estrategias de entrenamiento para la base de conocimiento y esas estrategias pueden utilizar otras de

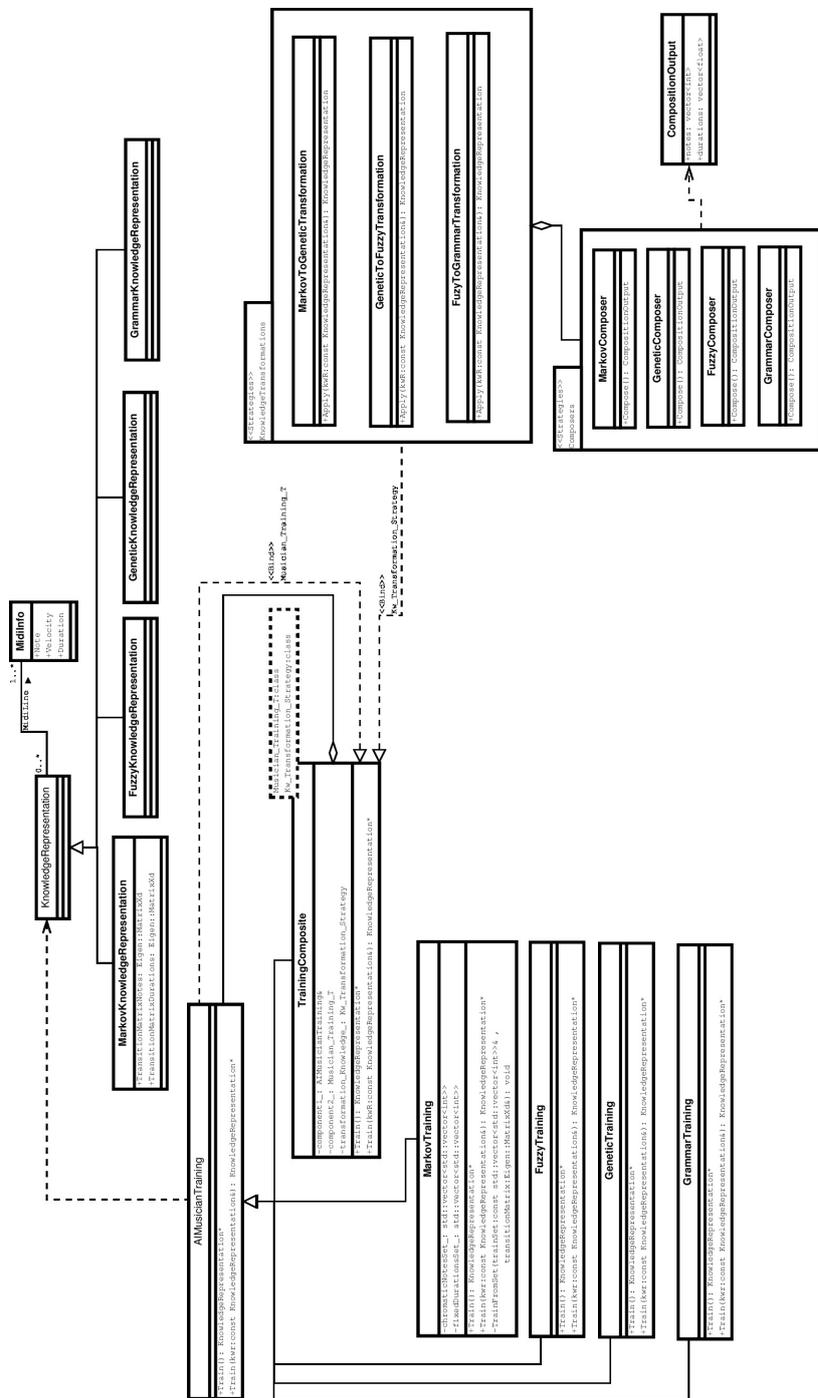


Figura 4.9: Diagrama de Clases para el Módulo de Entrenamiento

su misma naturaleza en una cadena de entrenamiento que para obtener una representación del conocimiento final (`KnowledgeRepresentation`).

KnowledgeTransformations: Un `AIMusicianTraining` necesita transformar una representación del conocimiento en otro que sea entendible para el siguiente proceso de entrenamiento en una cadena. Por lo tanto es necesario *transformar* las abstracciones de una representación a otra para que haya compatibilidad en el procedimiento. Es por ello que existen una serie de estrategias para realizar esas transformaciones del tipo `XtoYTransformation` que son utilizadas en `TrainingComposite` como *Templates*.

Composers: Los *composers* son estrategias de composición musical que dependen de las representaciones del conocimiento y sus transformaciones para generar la composición (`CompositionOutput`). Esta parte del diagrama corresponde al *módulo de producción* que será explicado más adelante, sin embargo es mostrado en este contexto para ilustrar el aporte del módulo de entrenamiento al módulo de producción.

El proceso de entrenamiento basado en los objetos producidos por las clases descritas se muestra en el diagrama de secuencias de la figura [4.10](#), en la que se puede apreciar un solo objeto de transformación, debido a que este gráfico

explica la esencia del funcionamiento del módulo.

4.2.7 Módulo de producción del sistema

Este módulo es el encargado de generar la composición musical y las síntesis de sonido utilizando la base de conocimientos y la guía de músicos reales. Es la parte del sistema que funciona de manera concurrente y en tiempo real, ya que debe realizar las siguientes tareas:

- Recibir las entradas de inicialización del sistema tales como: tempo(BPM), nota clave, intención emocional y configuración de síntesis para la base armónica que el músico real va a interpretar.
- Iniciar el motor de síntesis considerando el sonido que va a reproducir (dado por la configuración de síntesis de entrada) cada vez que el músico ejecute una nota.
- Cargar en memoria la base de conocimientos en las estructuras de datos adecuadas y únicamente con información relevante.
- Iniciar la reproducción del metrónomo en base a los BPM ingresados para el sistema y para el motor de síntesis.

- Sensar el ingreso de mensajes MIDI provenientes de los controladores constantemente para adquirir los datos y transformarlos en objetos `MidiInfo` y `MidiLine`.

En cada b número de beats:

- Aplicar los algoritmos de composición musical utilizando los `MidiLines` y la base de conocimientos para generar un patrón de melodía (notas y tiempos) a reproducir y una configuración de síntesis a utilizar en esa reproducción.
- Enviar el patrón de melodía y la configuración de síntesis al motor de síntesis para ejecutar las salidas por los parlantes.

El flujo de estas tareas es descrito en el diagrama de actividades presentado en la figura [4.11](#), además de su interacción con el motor de síntesis.

Cabe recalcar que la síntesis de sonido no se da solo para la composición del sistema, sino para la base armónica que está interpretando el o los músicos en ese instante; por ejemplo, el músico puede ajustar un sintetizador para producir un sonido de un órgano de aire y cada vez que ejecuta una nota, el motor de síntesis reproducirá ese sonido en conjunto con los sonidos producidos por la composición que la máquina haya generado y que pueden tener un

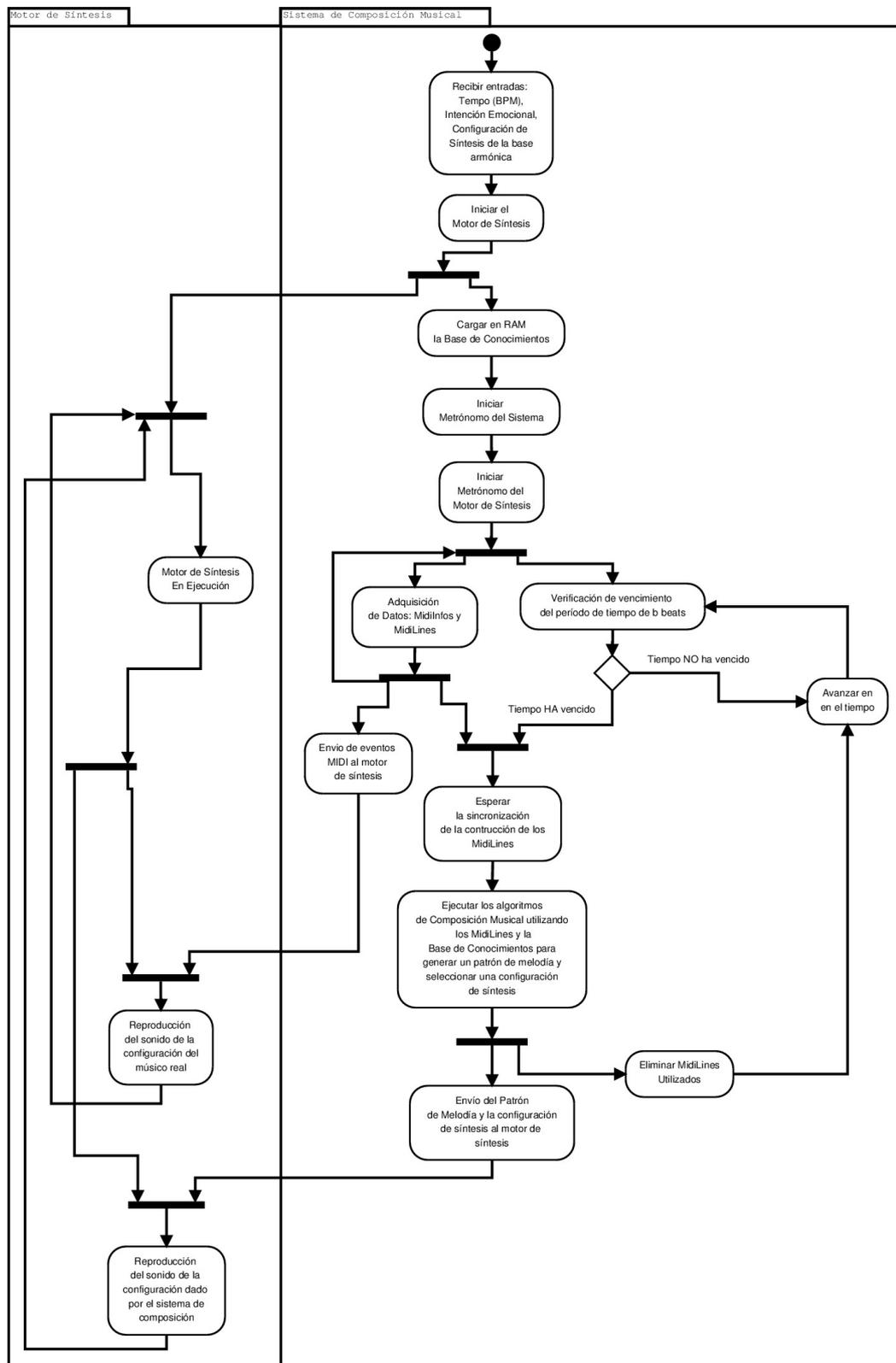


Figura 4.11: Diagrama de Actividades para el Módulo de Producción

timbre distinto al sonido del órgano. Sin embargo, si el controlador MIDI tiene la facultad de reproducir sonidos independientemente de la máquina, o está conectado a un sintetizador, ya sea de hardware o software, entonces no es necesario que el motor de síntesis genere sonidos para él.

En base a las tareas descritas, el diagrama de clases que refleja el diseño del sistema bajo el paradigma orientado a objetos se muestra en la figura [4.12](#).

Las clases propuestas son descritas a continuación:

RealTimeController: Es el objeto que gestiona todos los procesos que ocurren en la ejecución del sistema; es decir, conduce los procesos de síntesis y composición musical delegando tareas a el resto de objetos que interactúan con el directamente. Se encarga de iniciar o detener el motor de síntesis, iniciar el metrónomo, cargar la base de conocimientos en memoria y alimentar al improvisador (`MusicImproviser`) con los datos que requiera para la composición en tiempo real.

Metronome: Esta clase representa a un metrónomo virtual que se ajustará al tempo dado en las entradas. El `RealTimeController` se encargará de iniciarlo tanto en la aplicación como en el motor de síntesis para que sea escuchado por el músico, y sirva de guía en la composición que realiza el sistema. Cada b cantidad de beats se cumplirá un tiempo ΔT , en ese instante se ejecutará un callback

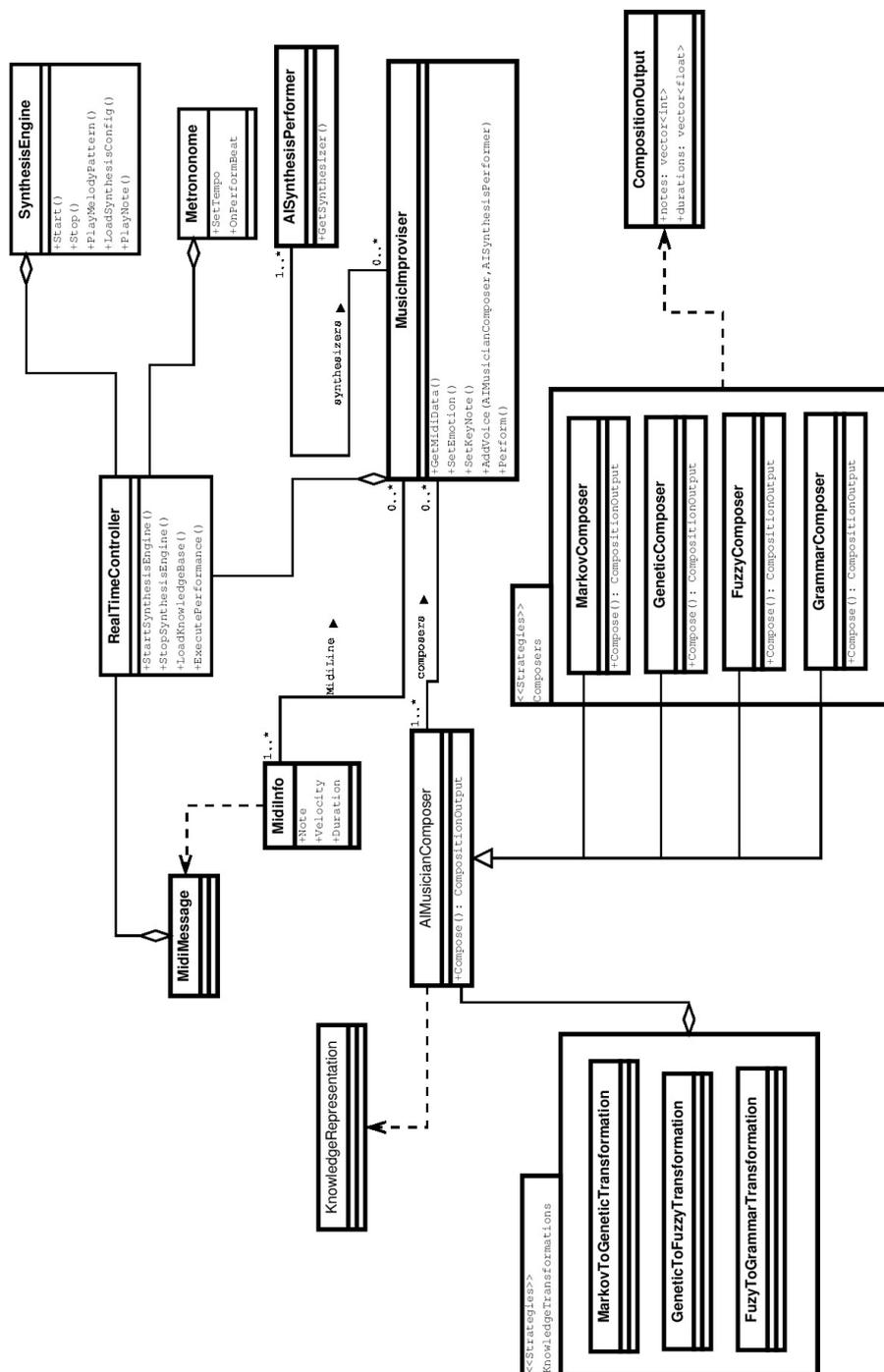


Figura 4.12: Diagrama de Clases para el Módulo de Producción

llamado `OnPerformBeat()` para aplicar los algoritmos de composición. De manera predeterminada b será igual a 4, ya que cada cuatro beats se cumple un bar en el tiempo musical y es más natural que ingresen melodías en ese instante, pero ese valor podría cambiar si se lo desea e incluso ser parte de las entradas del sistema.

SynthesisEngine: Es la clase que encapsula al motor de síntesis de audio. Debido al nivel de complejidad de este software, es necesario utilizar una interface que lo abstraiga en un modelo más sencillo de utilizar. Esta clase sería un *Facade* que permite la interacción con el motor.

AIMusicianComposer: Esta clase abstracta permite que sus hijos implementen, según su función, los algoritmos de inteligencia artificial para la composición musical. Estas clases concretas son un conjunto de estrategias que utilizan objetos de tipo `KnowledgeRepresentation` para generar las salidas (objetos `CompositionOutput`) que luego son dirigidas al motor de síntesis. Además de utilizar la base de conocimientos, son alimentados con los datos recolectados de los músicos que estén interpretando una obra en ese instante.

CompositionOutput: Es la clase que representa la salida producida por los algoritmos de composición. Consta de *notas* y *duraciones* las cuales

son transformadas en patrones de melodía según los requerimientos del formato que recibe el motor de síntesis. Este formato especial será detallado en la sección [4.2.8](#).

SynthesisPerformer: Esta clase se encarga de traducir los datos de configuraciones de síntesis a abstracciones que sean atendibles por el motor de síntesis. Básicamente son sintetizadores virtuales con parámetros específicos para producir el sonido requerido.

MusicImproviser: Esta clase es muy importante, ya que crea un hilo (`thread`) por cada pista que se desee reproducir; es decir que construye las melodías de manera concurrente para luego enviarlas al motor de síntesis y reproducirlas. Para lograr esto, se crean dos arreglos de datos, uno de objetos `AIMusicianComposer`, y otro de objetos `SynthesisPerformer`. El índice de acceso a los arreglos i , es el dato común entre un par de objetos $(AIMusicianComposer_i, SynthesisPerformer_i)$, donde `AIMusicianComposer_i` ejecutará su algoritmo de composición en el hilo i . Todos los hilos deben ser sincronizados de tal manera que terminen su tarea al mismo tiempo. El resultado de los algoritmos es otro arreglo con objetos `CompositionOutput`, el cual se asociará por medio de su índice con su correspondiente `SynthesisPerformer` de tal manera que se se

produzcan dos elementos que serán enviados al motor de síntesis: el patrón de melodía por cada `CompositionOutput`, y el tipo de sonido a reproducir dado por el `SynthesisPerformer` asociado. De esa manera se producirán melodías concurrentes por cada tiempo ΔT según la sincronización con el metrónomo cuando se ejecute `OnPerformBeat()`.

El proceso de producción de las salidas con los objetos descritos, es explicado gráficamente en el diagrama de secuencia que se encuentra en la figura [4.13](#). Nótese que el proceso concurrente en tiempo real es representado en la esquina inferior derecha.

4.2.8 Salidas producidas por el sistema

Las salidas producidas por el sistema son de dos tipos: *el patrón de melodía a reproducir*, y la *configuración de síntesis* del sonido a generar.

Como conocemos hasta ahora, el sistema produce objetos de tipo `CompositionOutput` que poseen *notas* y *duraciones* para cada nota. En esencia, `CompositionOutput` posee dos arreglos para representar cada ente y son asociados uno con otro por medio de un índice que comparten las dos colecciones. La descripción gráfica de esta representación se ilustra con

un ejemplo en la figura [4.14](#).

En este caso el objeto `CompositionOutput` es de tamaño ocho. Las notas musicales están en números MIDI, según la tabla [2.1](#), las notas de estos números serían: C3, E3, G3, A3, D4, A2, F4, C3. Las duraciones son relativas a los BPM ingresados al sistema, por ejemplo, si la entrada fue 120 BPM, entonces un beat se ejecuta cada 0.5 segundos; por lo tanto, las duraciones en segundos quedarían así: 0.25, 0.125, 0.5, 0.75, 1, 0.5, 0.5, 2.

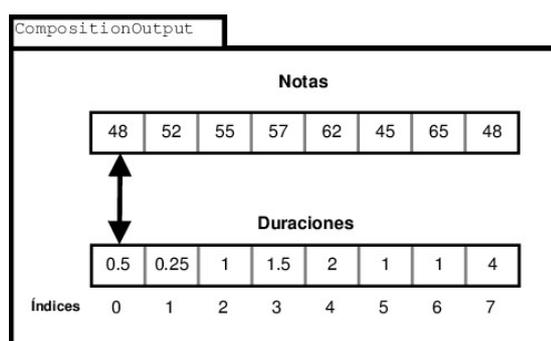


Figura 4.14: Ejemplo de un objeto `CompositionOutput`

Estas salidas son producidas con esta representación debido al formato requerido por el motor de síntesis, que se discutirá en la sección [4.2.9](#).

Para la *configuración de síntesis* el sistema produce objetos de tipo `SynthesisPerformer` que representan básicamente a un sintetizador con parámetros específicos. Estos parámetros son enviados al motor de síntesis para la construcción y reproducción del sonido; por ejemplo, uno de estos

sintetizadores virtuales puede estar compuesto por un oscilador sinusoidal, un filtro Paso-Bajo con la frecuencia de corte a 2000 Hz, y un envolvente ADSR con un attack de 0.12 seg, un decay de 0.1 seg, sustain de 0.25 y release de 0.35.

4.2.9 Distribución de las salidas hacia un lenguaje de programación de síntesis

Generalmente, un lenguaje de programación de síntesis posee objetos llamados `Patterns`, que son fundamentalmente arreglos de elementos a reproducir secuencialmente en el motor de síntesis. Un uso para estos `Patterns` es la generación de melodías.

El motor de síntesis posee procedimientos para reproducir estas secuencias y transformarlas en sonido. La función que realiza esto recibe como parámetros: la definición de síntesis del sonido a generar, un `Pattern` notas, y un `Pattern` de duraciones para esas notas; en consecuencia, estos dos patrones son del mismo tamaño. Las notas pueden estar dadas en valores de frecuencias o en número de notas MIDI; las duraciones están representadas por valores relativos al tempo general asignado al motor de síntesis.

Nótese que las salidas estudiadas en la sección [4.2.8](#) cumplen con las

condiciones de los patrones de un lenguaje de síntesis y por ende es posible asociarlas directamente con estos `Patterns`. Sin embargo, dependiendo de la forma de comunicación entre el sistema de composición y el motor de síntesis, esta representación puede cambiar desde el punto de vista de su formato.

Comúnmente, la comunicación de un motor de síntesis con otras aplicaciones se da por medio de los mensajes OSC (véase la sección [2.4.2](#)). Para este caso, se hace uso de este protocolo para poder enviar el `CompositionOutput` al motor de síntesis. Por lo cual, el objeto `CompositionOutput` es convertido a dos cadenas de caracteres, una para las notas, y otra para los tiempos, de tal manera que el mensaje OSC enviaría estas dos cadenas al motor para que sean procesadas como `Patterns`. Por ejemplo, en el lenguaje *Supercollider* explicado en la sección [2.4.2](#), los `Patterns` correspondientes a los arreglos de la figura [4.14](#) para notas y duraciones respectivamente, serían: “[48,52,55,57,62,45,65,48]”, y “[0.5,0.25,1,1.5,2,1,1,4]”.

Debido a que es posible enviar valores al motor de síntesis en variables numéricas además de cadenas de caracteres; los valores de los parámetros de los sintetizadores virtuales pueden hacer uso de esta ventaja y enviar por OSC las configuraciones para que sean asignadas a definiciones de síntesis;

por ejemplo, en Supercollider existe el objeto `SynthDef` que permite definir la señal de audio por medio de parámetros de síntesis, estos parámetros pueden ser reemplazados por las variables que el mensaje OSC recibe en el motor y así construir la definición, luego esa definición es utilizada como parámetro en la función que reproduce los `Patterns` para finalmente generar el sonido.

4.2.10 Módulo de síntesis de audio y la reproducción del resultado al exterior

La manera en que opera el motor de síntesis a escoger debe ser considerada en el diseño del sistema y encapsulado de tal manera que pueda ser reemplazado de ser necesario. Para ello, el motor de síntesis es representado en el sistema por la clase `SynthesisEngine`, que abstrae las funcionalidades necesarias del sistema en una sola interfaz para que sea utilizada por el objeto `RealTimeController`.

En la figura [4.15](#) se muestra la arquitectura del módulo de síntesis para el sistema, en la que se puede apreciar la interacción del objeto `SynthesisEngine`, que se encuentra del lado del sistema de composición, con los componentes del lado del motor de síntesis. Este arquitectura esta basada en los elementos abstractos que posee *Supercollider* para gestionar

la generación de sonidos, y por ende podría cambiar ligeramente dependiendo del motor de síntesis a utilizar.

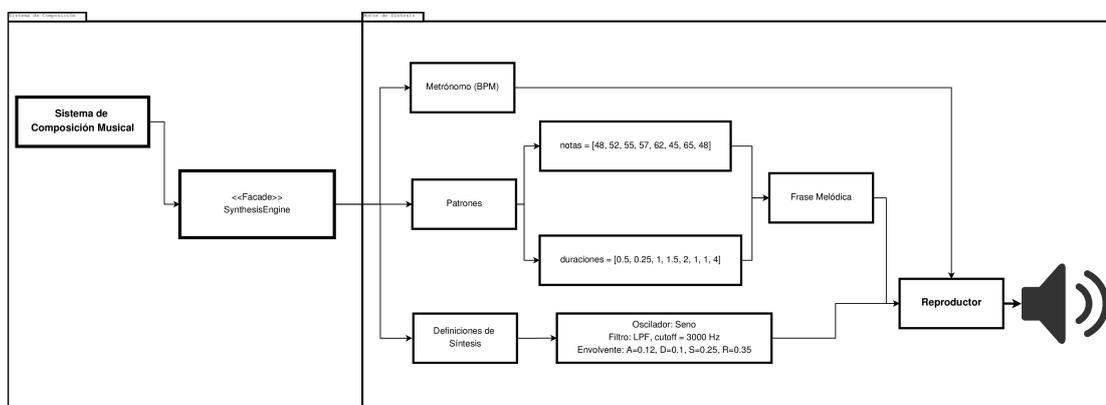


Figura 4.15: Arquitectura de Módulo de Síntesis para el Sistema

Los nuevos componentes presentados en este módulo son descritos a continuación:

Metrónomo(BPM): Este metrónomo es el que rige la reproducción de secuencias musicales (*Patterns*) dado un *tempo* en BPM; es decir, las duraciones serán transformadas a tiempos considerando este valor debido a que ellas estarán en unidades relativas a cualquier BPM ingresado. A través del *SynthesisEngine*, el sistema envía por un mensaje OSC el valor a utilizar.

Patrones: Son los objetos *Patterns* que en esencia representan colecciones de datos que componen una frase melódica. En este caso los patrones a utilizar son notas musicales en números MIDI, y

duraciones relativas a los BPM como muestra la arquitectura. El sistema de composición transmite vía OSC estos patrones en cadenas de caracteres como se explica en la sección [4.2.9](#).

Frases Melódicas: Estos elementos están compuestos por un par de `Patterns` (notas y duraciones) de tal manera que son utilizados para la reproducción de una melodía ajustada en el tempo del metrónomo.

Definiciones de Síntesis: Son un conjunto de configuraciones para generar un sonido en particular cuyas variables se alimentan de los mensajes OSC que recibe del sistema de composición. Estas configuraciones preparan los objetos necesarios en el motor de síntesis para hacer el cálculo de la señal de audio.

Reproductor: El reproductor es un elemento que toma las frases melódicas y las definiciones de síntesis para ejecutar una melodía y generar un sonido para esa melodía, con el fin de producir señales de audio que pasarán a los parlantes y serán escuchadas posteriormente.

Es ideal que para la manipulación del módulo de síntesis por parte de un usuario músico, se construya una interfaz gráfica que utilice las metáforas de manipulación de sonido, donde el mejor enfoque es la realización de un diseño basado en un sintetizador físico, debido a que los artistas están

acostumbrados a este tipo de modelos gráficos.

4.2.11 Módulo de grabación de los resultados emitidos

Se pretende que el sistema propuesto genere salidas en vivo como se ha venido mencionando a lo largo del documento; si queremos conservar esos resultados para escucharlos posteriormente es necesario *registrarlos*; además, es imperativo que exista una funcionalidad que permita esta tarea dado que las pruebas al sistema desde el punto de vista estético serían más sencillas de ejecutar porque en ese caso no habría necesidad de reunir a los sujetos de prueba y a los músicos que utilicen el sistema al mismo tiempo y en un mismo lugar.

Debido a las razones explicadas en el párrafo anterior, en la figura [4.16](#) se muestra una arquitectura básica para un módulo de grabación de los resultados emitidos por el sistema.

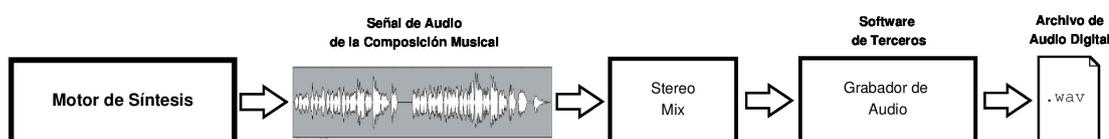


Figura 4.16: Arquitectura del Módulo de Grabación de apoyo al Sistema

Este módulo no es una parte a desarrollar en el sistema debido a que puede ser concebido con herramientas de terceros siguiendo las siguientes

indicaciones:

El motor de síntesis genera una señal de audio que es dirigida a los parlantes; sin embargo, esa señal puede ser enviada a un componente llamado *Stereo Mix*. Un *Stereo Mix* es un elemento virtual de un computador que recibe las señales de audio que son transmitidas por las salidas analógicas del mismo; es decir, lo que se escucha por los parlantes es lo que este componente recibe. El sistema operativo en conjunto con el driver de la tarjeta de audio brindan la posibilidad de utilizar este elemento. Luego de esto, se utiliza un software de grabación de audio de un tercero; generalmente están embebidos en editores de audio tales como *Adobe Audition*, *Audacity*, *Pro Tools*, etc. Este grabador debe ser configurado para que pueda recibir la entrada del Stereo Mix. Al final, el resultado es un archivo de audio digital de extensión .wav.

4.3 Diseño de Pruebas

Para definir las especificaciones del diseño de pruebas del sistema, utilizamos el estándar **IEEE 829-1998** [62], tomando en cuenta la sección de **Test Design Specification**. Por lo tanto, cada *Especificación de Diseño de Prueba* es detallada en las tablas que van de la 4.1 a la 4.6. Se presentarán solo las más relevantes.

Ítem	Descripción
Identificador:	DA.TD.01
Funcionalidad(es) a ser evaluada(s):	<ul style="list-style-type: none"> ●Conexión a controladores MIDI estándar. ●Recepción de mensajes MIDI. ●Abstracciones de los mensajes MIDI.
Refinamientos del Enfoque:	<p>La prueba de adquisición de datos consta desde el hardware que proporciona la entrada, hasta el conjunto de abstracciones que representan a los datos adquiridos que serán utilizados en las estrategias posteriores. Para evaluar las funcionalidades se conectarían varios dispositivos, verificando su funcionamiento en algún software receptor de mensajes MIDI comprobando que el hardware para utilizar en la prueba envía los mensajes correctos, luego contrastar si esos mismos mensajes son los que recibe la aplicación, verificar que los mensajes correspondan al dispositivo que los emite, implementar un algoritmo que transforme las abstracciones a sus originales mensajes MIDI para compararlos con los utilizados que dieron lugar a esas abstracciones.</p>
Identificación de Casos de Prueba:	<p>DA.TC.001 - Conexión de N controladores MIDI a la aplicación, tal que N es el número de puertos (terminales que acepten MIDI) disponibles en la máquina de prueba. Asegurarse que $N \geq 2$.</p> <p>DA.TC.002 - Envío desde los dispositivos de una serie de mensajes compuestos por las doce notas desde $middleC(C3)$ hasta C4 para verificar mensajes.</p> <p>DA.TC.003 - Los mensajes corresponden al dispositivo que los envía.</p> <p>DA.TC.004 - Abstracciones de datos corresponden a los mensajes MIDI recibidos.</p>
Criterio de aprobación/fallo:	<ul style="list-style-type: none"> ●Debe soportar dos o más controladores MIDI al mismo tiempo. ●Todos los mensajes enviados deben corresponder exactamente a los recibidos.

Tabla 4.1: Especificación de Diseño de Prueba: Adquisición de Datos

Ítem	Descripción
Identificador:	KB.TD.02
Funcionalidad(es) a ser evaluada(s):	<ul style="list-style-type: none"> ●Guardado hacia la base de conocimientos en disco. ●Carga desde la base de conocimientos en disco.
Refinamientos del Enfoque:	<p>Cuando las estructuras de datos en memoria están listas con la información generada por el entrenamiento, éstas se almacenarán en la base de conocimientos en disco pero aún no serán liberadas con el objetivo de compararlas con nuevos objetos en memoria que carguen lo que se guardó en disco y verificar la consistencia de esa información.</p>
Identificación de Casos de Prueba:	<p>KB.TC.001 - Guardado de la base de conocimientos en disco efectuado (sin verificación de consistencia) para cada estructura de datos utilizada.</p> <p>KB.TC.002 - Carga de la base de conocimientos en disco efectuada (sin verificación de consistencia) para cada estructura de datos utilizada.</p> <p>KB.TC.003 - Verificación de consistencia entre las estructuras de datos que se guardaron y las que se cargaron.</p>
Criterio de aprobación/fallo:	<ul style="list-style-type: none"> ●La tasa de fallos con respecto a las operaciones de disco debe tender a cero. ●La información de las estructuras que sirvieron para guardar los datos debe ser idéntica a las cargadas posteriormente.

Tabla 4.2: Especificación de Diseño de Prueba: Manipulación de la Base de Conocimientos

Ítem	Descripción
Identificador:	TS.TD.03
Funcionalidad(es) a ser evaluada(s):	<ul style="list-style-type: none"> •Exactitud del metrónomo del sistema.
Refinamientos del Enfoque:	El metrónomo del sistema será comparado con otro metrónomo ajeno a la aplicación de tal manera que los dos inicien al mismo tiempo y se compruebe que no se desfasan significativamente durante un período determinado.
Identificación de Casos de Prueba:	<p>TS.TC.001 - Ejecución de los dos metrónomos con un tempo de 20 BPM por un lapso de una hora.</p> <p>TS.TC.002 - Ejecución de los dos metrónomos con un tempo de 300 BPM por un lapso de quince minutos.</p>
Criterio de aprobación/fallo:	<ul style="list-style-type: none"> •El desfase entre los dos metrónomos debe ser menor o igual a 100 milisegundos.

Tabla 4.3: Especificación de Diseño de Prueba: Sincronización de Tiempo

Ítem	Descripción
Identificador:	SEI.TD.04
Funcionalidad(es) a ser evaluada(s):	<ul style="list-style-type: none"> ●Ejecución correcta del envío de datos desde el sistema y recepción del motor. ●Sincronización de tiempo entre los dos entes ●Consistencia de los datos enviados con los recibidos
Refinamientos del Enfoque:	El enfoque es similar a la especificación DA.TD.01 y TS.TD.03. Los datos enviados por cada hilo deben ser recibidos en igual número por el motor de síntesis y deben corresponder exactamente a cada uno de esos hilos. Luego, estos datos deben ser consistentes uno con otro, para ello se utilizarán notas musicales, de la cuales se mostrará lo que envió el sistema y lo que recibió el motor de síntesis además de emitir los respectivos sonidos para verificar que todo el flujo es correcto. Adicionalmente, se comprobará que no haya desfase de tiempo entre el metrónomo del sistema y el del motor de síntesis.
Identificación de Casos de Prueba:	<p>SEI.TC.001 - Ejecución de 9 hilos, donde cada uno soportará la ejecución de las 12 notas musicales desde la octava cero a la ocho (un hilo por octava), y enviarlos al motor de síntesis.</p> <p>SEI.TC.002 - Ejecución de los dos metrónomos con un tempo de 20 BPM por un lapso de una hora.</p> <p>SEI.TC.002 - Ejecución de los dos metrónomos con un tempo de 300 BPM por un lapso de quince minutos.</p>
Criterio de aprobación/fallo:	<ul style="list-style-type: none"> ●Los datos enviados deben coincidir exactamente con los recibidos. ●El desfase entre los dos metrónomos debe ser menor o igual a 100 milisegundos.

Tabla 4.4: Especificación de Diseño de Prueba: Integración con el Motor de Síntesis

Ítem	Descripción
Identificador:	PF.TD.05
Funcionalidad(es) a ser evaluada(s):	<ul style="list-style-type: none"> ●Concurrencia. ●Utilización de memoria. ●Procesamiento. ●Sincronización entre entradas y salidas.
Refinamientos del Enfoque:	Se ejecutarán un número determinado de procesos concurrentes que pueda soportar el sistema, de los cuales se tomarán datos relevantes acerca de la memoria utilizada y tiempos de ejecución.
Identificación de Casos de Prueba:	<p>PF.TC.001 - Ejecución de 8 hilos que simulen recibir entradas de controladores MIDI considerando que es el máximo de entradas paralelas de un interfaz MIDI en el mercado.</p> <p>PF.TC.002 - Ejecución de la escala de Fibonacci (para obtener resultados a lo largo de un crecimiento rápido de hilos) 1, 2, 3, 5, 8, 13, 21, 34 y 55 hilos para las salidas generadas por todas las estrategias, salidas que son enviadas al motor de síntesis.</p>
Criterio de aprobación/fallo:	<ul style="list-style-type: none"> ●Debe soportar al menos el 50% de la carga concurrente. ●El desfase entre el inicio del procesamiento con las estrategias de inteligencia artificial y la reproducción debe de ser de mínimo un beat. ●El desfase entre las entradas y salidas debe ser menor o igual a 200 milisegundos considerando dentro de este tiempo el procesamiento de las estrategias. ●La memoria utilizada únicamente para el procesamiento debe ser menor a 20 MB entre la utilización del sistema y del motor de síntesis.

Tabla 4.5: Especificación de Diseño de Prueba: Rendimiento

Ítem	Descripción
Identificador:	MC.TD.06
Funcionalidad(es) a ser evaluada(s):	<ul style="list-style-type: none"> ●Naturalidad de la composición del sistema (no distinción entre si es un humano o una máquina). ●Estética de la composición. ●Asociación entre la intención emocional del músico-sistema¹ y la percepción emocional del oyente.
Refinamientos del Enfoque:	<p>Para estas pruebas intervienen personas que soportarán las evaluaciones. Serán de dos tipos, músicos y no músicos escogidos al azar. Los músicos evaluarán con criterios puntuales lo siguiente: la composición de melodías de la máquina sin la guía de un intérprete a nivel estético, clasificación de un grupo de melodías que contiene frases compuestas por la máquina y otras por músicos reales. Los oyentes no músicos brindarán su percepción de las emociones generadas por una interpretación del músico-sistema. Para una guía mas concreta con respecto a estas pruebas, ver la sección 2.6.3.</p>

Tabla 4.6: Especificación de Diseño de Prueba: Composición e Interpretación de Melodías Musicales

<p>Identificación de Casos de Prueba:</p>	<p>MC.TC.001 - Músicos bajo el mismo estilo componen 30 melodías, 15 para entrenamiento y 15 para pruebas con un promedio de 30 segundos cada una. El sistema genera 30 melodías utilizando los módulos de la base de conocimientos. Se entrega un total de 45 melodías a los músicos evaluadores para que califiquen estéticamente con un valor de 0 a 10 y clasifiquen en dos grupos; melodías del humano y melodías de la máquina. Hacer uso de la <i>Prueba de Turing</i> mejorada (CAT) (ver sección 2.6.3.1).</p> <p>MC.TC.002 - Niveles de percepción de emociones por parte de oyentes con respecto a la intención emocional de 15 piezas de 60 segundos cada una grabadas de una interpretación músico-sistema en tiempo real. Los oyentes tienen que clasificar a qué emoción o emociones pertenecen y con cuanto porcentaje (Ej. tristeza con 60%, melancolía con 40% para la pieza 5). La comparación entre intención y percepción puede abarcar más parámetros como los mostrados en la figura 2.27.</p>
<p>Criterio de aprobación/fallo:</p>	<ul style="list-style-type: none"> ●No debe haber diferencia significativa entre las calificaciones asignadas a las composiciones del humano con respecto a la de la máquina. ●Más del 50% de los evaluadores debe confundir el producto del sistema con el de músicos reales. ●No debe haber una diferencia significativa entre la intención emocional y la percepción emocional de las piezas músico-sistema. ●Los oyentes deben reportar que percibieron emociones en las piezas a pesar de que exista o no diferencia entre la intención y la percepción emocional.

Tabla 4.6: Especificación de Diseño de Prueba: Composición e Interpretación de Melodías Musicales

CAPÍTULO 5

5. IMPLEMENTACIÓN, ENTRENAMIENTO Y EJECUCIÓN DEL SISTEMA

Para este trabajo se implementó un sistema piloto para la arquitectura propuesta en el capítulo 3 e ilustrada en la figura 3.9 bajo las especificaciones del diseño de un sistema que la soporte como se lo expone en el capítulo 4. La arquitectura completa del sistema desarrollado se muestra en la figura 5.1 donde es posible apreciar la interacción entre las partes con el fin de ejecutar la producción de material musical (melodías concurrentes) a partir del conocimiento adquirido en el entrenamiento.

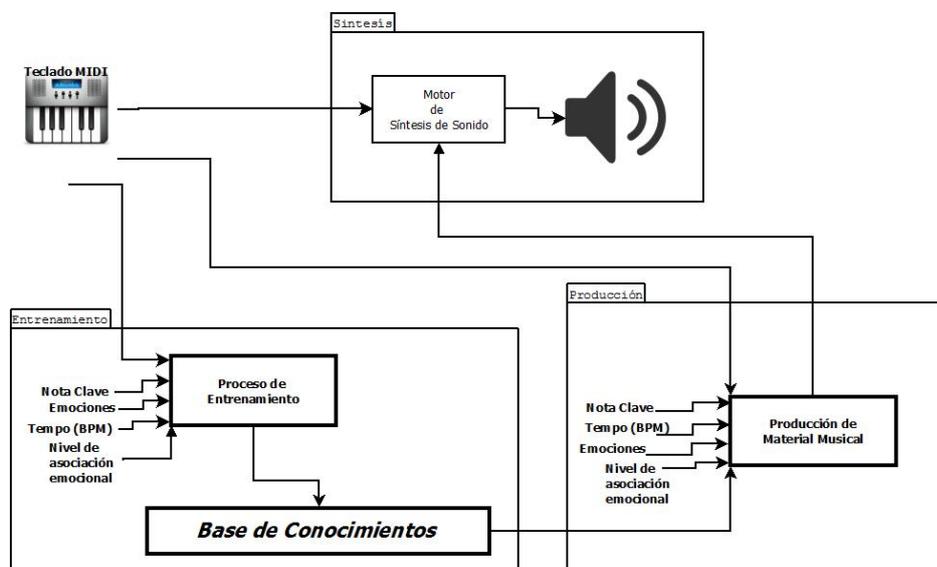


Figura 5.1: Arquitectura de Representación del Conocimiento y Producción de Material Musical para la Implementación

5.1 Consideraciones de implementación

5.1.1 Herramientas utilizadas

Las siguientes herramientas de hardware y software fueron utilizadas para la implementación del piloto:

Controladores MIDI con forma de teclado musical: Dispositivos de hardware que fueron conectados al sistema vía USB con la posibilidad de enviar y recibir mensajes MIDI. Únicamente la funcionalidad de envío de mensajes es necesaria para el sistema. Un teclado musical permite una asociación más clara de las notas musicales con un ente

físico.

Tarjeta de sonido interna: Debido a la capacidad de la máquina utilizada no fue necesario utilizar interfaces de audio adicionales más que la tarjeta de sonido interna ya que no se perciben retardos significativos que afecten la interpretación de un músico.

Secuenciador MIDI: Para entrenar al sistema no es necesario interactuar con él en tiempo real como se lo explicó en la sección [4.2.6](#)). Así que se le entrega un archivo .mid que es grabado de un secuenciador MIDI (ver sección [2.3.3](#)) para que pueda extraer los mensajes MIDI y realizar el entrenamiento. Se utilizó en específico el software *MPC Essentials* que opera en conjunto con controladores MIDI de Akai, pero que también no se restringe a otros controladores que cumplan el estándar.

Lenguaje de Programación C++: Para la implementación de los algoritmos que rigen al núcleo del sistema (entrenamiento y producción) se utilizó C++ en su versión C11 por su alto nivel de abstracción y acercamiento al hardware.

Librería RtMidi: La librería RtMidi [\[63\]](#) está compuesta de un conjunto de clases implementadas en C++ para la simplificación de los procesos

de interacción con dispositivos y programas que utilicen el estándar MIDI. Con esta librería se gestionaron y se extrajeron los mensajes MIDI de los dispositivos físicos.

Librería `MidiFile`: Las clases en C++ de la librería `MidiFile` [64] permiten leer y acceder a los atributos de archivos MIDI así como también escribir archivos de esta naturaleza. Con esta librería se extrajeron los mensajes MIDI de los archivos `.mid` para el entrenamiento.

Librería `oscpack`: Librería `oscpack` [65] permite el envío y recepción de mensajes OSC mediante el API escrito en C++ que ofrece. Esta librería fue utilizada para el envío de los resultados de los algoritmos de composición al motor de síntesis así como las notas que los músicos usuarios están interpretando en ese instante.

Lenguaje y Motor de Síntesis `SuperCollider`: `SuperCollider` (ver sección 2.4.2) fue la herramienta de síntesis de sonido con la cuál se pueden apreciar los resultados concretos de la producción del sistema.

5.1.2 Restricciones del ambiente

El sistema es flexible en su implementación en hardware; es decir, es posible recibir N controladores MIDI hacia el computador. Sin embargo, el acondicionamiento del ambiente en el cual se ejecute es importante tanto por el equipamiento como por los sujetos que utilicen el sistema.

Es necesario en lo posible que se determine una configuración del lugar de tal manera que no afecte la interpretación de los músicos cuando se entrene el sistema o se utilice en el ámbito de producción de melodías. La importancia de esto radica en el efecto que puede provocar en los procesos creativos que suceda en el cerebro de los músicos en ese instante. Si las condiciones los perjudican puede que la intención que ellos deseen plasmar sea modificada negativamente y produzca un sesgo en los datos recolectados e inconsistencia entre las melodías generadas y su interpretación ya en la puesta en marcha del sistema en tiempo real. Por lo tanto, se debe intentar aislar de ruidos externos e incomodidades tanto del ambiente como del sistema para lograr una interpretación lo más pura posible para que refleje la intención emocional del músico.

5.1.3 Optimizaciones

Dado que el proceso en esencia permite la generación de melodías, entonces se va a lidiar netamente con *secuencias* de elementos, tales como notas y tiempos. Por lo tanto, para optimizar las operaciones que realice el sistema es necesario ajustarse a esta particularidad de las melodías; es por ello que lo mejor es utilizar estructuras de datos secuenciales cuyos elementos estén organizados en la memoria de manera contigua. En C++ lo adecuado sería la utilización de arreglos estáticos (`[]`) y dinámicos (`vector`).

También, el *pre-cálculo* de datos requeridos en el funcionamiento del sistema es una estrategia importante para la optimización del procesamiento. Por ejemplo, la mayor eficiencia es requerida para el sistema en tiempo real y por lo tanto resultaría costoso ejecutar un algoritmo genético durante un gran número de generaciones, por lo cual resulta necesario que los resultados sean calculados previos a la ejecución; es decir, en la etapa de entrenamiento de tal manera que sean guardados y recuperados posteriormente para ser seleccionados en la etapa de producción en tiempo real y así reducir el tiempo de procesamiento en composición.

Reducir la *carga estructural* también provee beneficios. Esto significa que en el diseño orientado a objetos se reduzca en lo posible jerarquías de clases

muy profundas. Esto lo maneja de una manera excelente el lenguaje C++ a través de sus `templates` los cuales permiten definir tipos de datos en tiempo de compilación y así establecer y agregar funcionalidad además de evitar redundancia sin necesidad de jerarquizar el diseño de manera intensa.

5.2 Entrenamiento del sistema

5.2.1 Características de los expertos que aportarán al entrenamiento

Considerando la especificación de diseño de prueba **MC.TD.06** de la tabla 4.6, para el entrenamiento del piloto se consideraron a los dos músicos entrevistados para la guía de selección de técnicas de inteligencia artificial a aplicar explicada en el capítulo 3.

Estos dos músicos han estudiado en la *academia* pero han bifurcado ligeramente esa línea de conocimientos para orientarla a la práctica de la *música experimental*, que implica no solo innovación en el proceso de composición, sino en la generación de sonidos mediante sintetizadores que refuerzan esa composición (Ver Anexo A).

La base de conocimientos refleja parte de su proceso creativo implícito que es

plasmado en sus obras y por lo tanto refuerza la arquitectura generada por el conocimiento explícito que ellos brindaron en conjunto con las estrategias de inteligencia artificial exploradas en este trabajo.

5.2.2 Facilidades de interacción con el sistema para los expertos

En esencia, los expertos deben simplemente interpretar melodías y definir los siguientes parámetros para cada pieza:

Intención Emocional: Definir las emociones que participan en la pieza y su nivel de asociación con ella. Por ejemplo tristeza con 60 %, melancolía con 40 %.

Tempo: El valor en BMP para el tempo en el cuál el músico va interpretar.

Nota Clave: La nota clave que rige la composición de tal manera que se apliquen operaciones de transposición en base a ella si se utilizan estos datos para una nota clave distinta.

Por lo cual, la interacción con el sistema se basa en proporcionar esos parámetros, dar inicio a la grabación y detener la grabación cuando se cumpla el tiempo establecido para continuar con la siguiente pieza hasta cumplir el

número de muestras requeridas.

5.2.3 Plan de escenarios de entrenamiento

Los músicos expertos tienen sesiones en las que exploran temáticas particulares y que pueden abarcar un rango de emociones que refuerzan esa temática. El enfoque puede ser de una sola emoción, pero ellos son conscientes de que la subjetividad de sus obras puede lograr que surjan variaciones de su intención emocional.

En base a esto, el enfoque de entrenamiento fue dirigido a un grupo de emociones sobre una clase en particular de grupo. En la sección [2.6.1.2](#) se hace referencia a ocho clases de grupos de emociones mostrados en la figura [2.28](#). Las emociones escogidas son parte de estas clases y en las que los músicos propusieron, de tal manera que se ajustan más al trabajo de ese instante de ellos para formar un solo grupo de emociones concernientes a ese trabajo; por lo tanto, el grupo seleccionado está conformado por las siguientes emociones cuyas definiciones fueron consultadas en un diccionario y modificadas por ellos:

Alegría: Sentimiento grato y vivo producido por un motivo placentero.

Serenidad: Tranquilidad, calma, apacibilidad, relajación

Tristeza: Sentimiento de estar afligido, apenado, oscuro, apagado.

Nostalgia: Sentimiento de pena que causa recordar personas o cosas del pasado.

Apasionamiento: Grado de energía o fuerza hacia acciones con un alto entusiasmo, puede ser considerado de cierta manera una contra parte de “serenidad” pero no del todo.

Con respecto a este grupo, los escenarios de entrenamiento a aplicar fueron los siguientes:

1. Asignar los valores de tempo, nota clave, y configuración del sonido que se desea. Luego interpretar la pieza.
2. Después de la interpretación, ponderar con un porcentaje entre 0 y 100 % cada emoción mencionada en el listado anterior para la pieza.

En una primera evaluación, para probar el componente de Cadenas de Markov, se aplicó el primer escenario para 30 piezas musicales de un promedio de 30 segundos cada una, de donde 15 se escogieron para el entrenamiento y el resto para las pruebas como está plasmado en la especificación de diseño de prueba **MC.TD.06** de la tabla [4.6](#).

Para una segunda evaluación, se consideró el componente de Cadenas de Markov, el Generador de Patrones basado en el algoritmo [3.1](#) para la generación de notas y duraciones con respecto a las matrices de transiciones, y el Clasificador de Emociones; de tal manera que comparó la intención emocional y la percepción por parte de los oyentes como se explica en **MC.TD.06**.

5.3 Ejecución del sistema

5.3.1 Interpretación musical en tiempo real de melodías por parte de los expertos y el sistema

Una vez que el sistema está entrenado, se realiza la ejecución del mismo tomando en cuenta que los expertos lo utilizarán en un ambiente en tiempo real, lo cual quiere decir que las piezas producidas son un producto de una *improvisación* músico-sistema. Para las pruebas la restricción de duración de las piezas fue de aproximadamente 60 segundos.

5.3.2 Recolección de información para evaluar la estética de la composición

5.3.2.1 Características de los escuchas

Tomando en cuenta la especificación de diseño de prueba **MC.TD.06** de la tabla [4.6](#). Los *escuchas* o también denominados *oyentes*, son de dos tipos:

Músicos: Personas que se encuentran bajo una línea similar a los expertos de los cuales se basó la construcción y entrenamiento del sistema. Es decir, músicos experimentales cuyos esfuerzos son enfocados en innovación a nivel de composición y sonido.

No Músicos: Personas escogidas al azar con edades entre 20 y 30 años cuyas inclinaciones en la música que escuchan involucran lineamientos similares a la música que producen los artistas experimentales en conjunto con sonidos que las refuercen.

5.3.2.2 Exposición de la música generada ante los escuchas

Para una mayor flexibilidad de las pruebas es necesario que las piezas que se generen en el modo de ejecución sean grabadas para que posteriormente se

las envíe a los escuchas con el objetivo de que realicen las respectivas evaluaciones. En ese caso es necesario considerar dentro de la implementación el uso del módulo de grabación explicado en la sección [4.2.11](#).

Esto es debido a que se requiere que el oyente encuentre las condiciones adecuadas que le permiten analizar con calma el material musical generado para evitar un sesgo en la recopilación de datos.

5.3.2.3 Recolección de información de la experiencia auditiva de los escuchas en términos de composición, sonido y emociones

En una primera evaluación al sistema, se consideró producir composiciones generadas netamente por la máquina con el objetivo de aplicar una *Prueba de Turing* modificada con el método de evaluación *consensual assessment technique (CAT)* explicado en la sección [2.6.3.1](#). Por lo tanto, los datos para esta primera evaluación son:

Calificaciones de estética: Los músicos evaluadores deben proporcionar calificaciones a las melodías generadas por la máquina entre 0 y 10 considerando criterios estéticos con respecto a la composición.

Clasificación de la naturaleza de la pieza: Los músicos evaluadores deben clasificar, de entre un conjunto ordenado al azar de melodías producidas por el software de inteligencia artificial y por músicos reales, cuales son compuestas por la máquina y cuales por el humano.

La segunda evaluación hace uso de piezas musicales completas generadas por los músicos y el sistema en tiempo real; es decir, no solo son melodías, sino pequeñas obras que también poseen una base armónica. Esta prueba fue realizada en base a las emociones estipuladas en la sección **5.2.3**. El escenario en el que se desenvuelve esta prueba considera emociones ponderadas por: *el intérprete* (músico-sistema) quien propone una intención emocional, y *el escucha*, quien percibe las emociones. Por lo tanto, los datos a recopilar son los valores de las ponderaciones basadas en porcentajes por cada emoción transmitida y percibida de tal manera que se verifique si lo pretendido es lo percibido por los oyentes que participen.

5.3.3 Recolección de datos acerca del rendimiento del sistema

Una parámetro importante con respecto a la ejecución del sistema es su *rendimiento*. Por lo cual, siguiendo la especificación de diseño de prueba **PF.TD.05** de la tabla **4.5**, los casos **PF.TC.001** y **PF.TC.002** son considerados para este tipo de pruebas; sin embargo, es importante la opinión

independiente de usuarios del sistema con respecto a los retardos debidos al procesamiento y la comunicación entre componentes ya que 200 milisegundos puede no ser adecuado para algunos músicos y tal vez se necesite un límite menor.

CAPÍTULO 6

6. ANÁLISIS DE LOS RESULTADOS

6.1 Presentación y Análisis de los Resultados de la composición musical

Para la prueba de composición musical considerando el componente de Cadenas de Markov, se pidió a cinco músicos compositores con formación académica evaluar melodías generadas por el sistema y otras producidas por los músicos entrevistados, calificándolas en un rango de 0 a 10, siendo el valor más alto la mejor calidad de la melodía según el criterio de cada músico evaluador.

El banco de melodías compuestas por los músicos experimentales fueron 30;

15 de ellas fueron usadas para el entrenamiento del sistema, y las otras 15 fueron utilizadas para evaluar el sistema. Las 15 melodías de entrenamiento tuvieron un total de 845 notas con un promedio de 56.33 notas para cada secuencia de melodía; lo mismo fue aplicado para las duraciones de cada nota, lo que en conjunto alimentó las matrices de transiciones. También, se generaron 30 melodías por el sistema, luego de ser entrenado, y a través de la estrategia de generación de secuencias para las Cadenas de Markov. En total se entregaron 45 melodías a los músicos evaluadores, ordenadas al azar.

Para establecer el tipo de prueba estadística a aplicar, fue necesario verificar la normalidad de las muestras. En la tabla [6.1](#) se presentan los resultados obtenidos por el test de normalidad de *Shapiro-Wilk*.

Compositor	Valor p	W
Humano	0.003841	0.9479
Máquina	3.895e-07	0.924

Tabla 6.1: Resultados de la prueba de Shapiro-Wilk

Podemos observar que los datos no siguen una distribución normal; por lo tanto, se optó por la prueba *Wilcoxon signed-rank*. *Wilcoxon signed-rank* es una prueba no paramétrica que no asume la normalidad de los datos. Es usada para comparar dos grupos que vengan de los mismos participantes, con la finalidad de observar el cambio en los sujetos de un tiempo a otro, o cuando

los individuos son sometidos a diferentes condiciones y analizar los efectos causados en los individuos. Para la prueba se asume que los dos tipos de melodías siguen el mismo estilo de los músicos que las generaron.

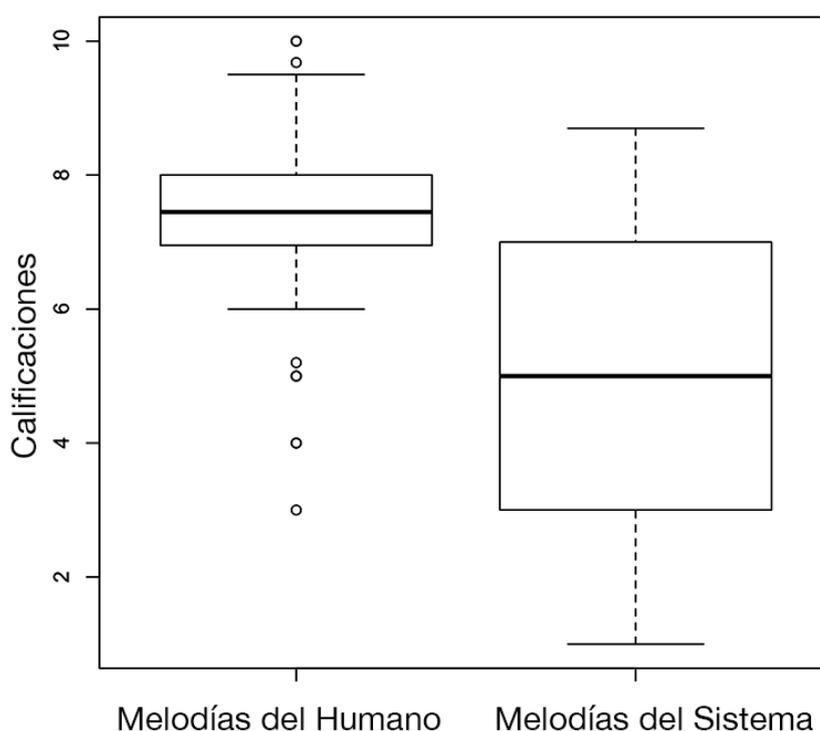


Figura 6.1: Diagrama de cajas para los grupos de melodías evaluados

Compositor	N	Mín.	Máx.	Media	Mediana	Desv. Stand.
Humano	75	3.00	10.00	7.343	7.450	1.457167
Máquina	150	1.00	8.70	4.902	5.00	2.109666

Tabla 6.2: Estadística descriptiva de los dos grupos

En la figura [6.1](#) y la tabla [6.2](#) se muestra un diagrama de cajas y la estadística descriptiva entre los dos grupos de melodías respectivamente, se puede

observar que las medianas difieren significativamente, lo cual fue comprobado por la prueba de *Wilcoxon signed-rank*, como se muestra en la tabla 6.3, de manera que podemos verificar que las melodías generadas por el sistema no siguen los mismos patrones que se le administró para el entrenamiento con el 95 % de confiabilidad.

Valor p	W
4.733e-16	9355

Tabla 6.3: Resultados de la prueba de Shapiro-Wilk

La hipótesis nula en la prueba *Wilcoxon signed-rank* define que las diferencias de medianas entre pares de observaciones es igual a cero, por consiguiente el valor p, obtenido en la investigación muestra que existe una diferencia estadística significativa de las medianas de estos dos grupos de melodías.

Con respecto al tipo de prueba de Turing realizada, se tomaron en cuenta los aciertos con respecto a la clasificación en cada grupo (humano o sistema) como lo muestra la figura 6.2.

Nótese que dos de los evaluadores (el 2 y el 5) lograron clasificar el material correctamente, el resto posee un porcentaje de fallo que indica que hubo melodías que no pudieron ser distinguidas para el grupo al que pertenecían. Además, los comentarios con respecto las pruebas reportan que las melodías

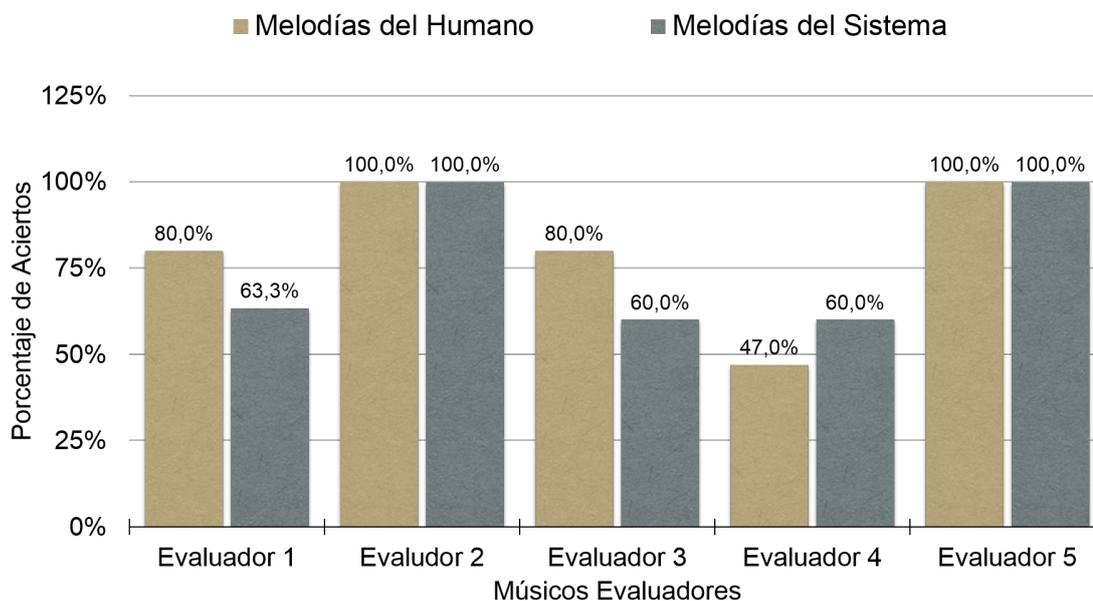


Figura 6.2: Porcentajes de aciertos para la prueba de Turing

que ellos creían que eran generadas por humanos tenían estructura y coherencia de inicio a fin, mientras que las demás se sentían más aleatorias y con una estructura difusa, adicionalmente mencionaron que la calidad pudiera aumentar si existiese una base armónica que soportase cada melodía.

6.2 Presentación y Análisis de los Resultados de la síntesis de sonido y su evocación de emociones

Esta prueba considera los componentes de Cadenas de Markov, Generador de Patrones (utilizando el algoritmo sobre las matrices de transiciones) y el Clasificador de Emociones.

El sistema fue entrenado con 15 melodías de un promedio de 30.0 segundos por cada una. Las emociones seleccionadas por el músico que proporcionó el entrenamiento fueron: *felicidad, serenidad, tristeza, nostalgia, apasionamiento* las cuales fueron ponderadas con un nivel emocional entre 0 y 100 para cada melodía, dependiendo de la intención emocional, además la *nota clave* y el *tempo* fueron proporcionados durante el entrenamiento. El sistema generó 30 patrones de melodías utilizando el enfoque de Cadenas de Markov, los cuales fueron ponderados utilizando el proceso de fusificación explicado en la sección [3.1.1](#).

El compositor humano interpretó la base armónica (acordes) para 15 piezas musicales de una duración de 60 segundos aproximadamente para cada una, además se proporcionó la intención emocional para estas piezas, de tal manera que el sistema interpretó melodías consistentes con la armonía e intenciones emocionales dadas utilizando el proceso de defusificación que se explica en [3.1.2](#).

Estas 15 piezas musicales fueron enviadas a personas (oyentes) familiarizadas con la música occidental con el objetivo de que ellos ponderen cada canción con su nivel de percepción emocional según lo que sintieron, 30 personas contribuyeron a la evaluación.

Los resultados son presentados desde la figura [6.3](#) a la [6.17](#), como diagramas

de cajas que muestran la intención emocional del músico y las emociones percibidas por los oyentes.

Considerando una de las imágenes, en la figura 6.3, la intención emocional del músico difiere de la tendencia de la percepción emocional del oyente; sin embargo, esta percepción se inclina de manera similar a la intención; por ejemplo, en la figura 6.3 hay más *serenidad* y *nostalgia* que *felicidad* y *apasionamiento* en las percepciones así como en las intenciones, pero la *tristeza* no se ajusta a este comportamiento. Este fenómeno es similar en el resto de diagramas donde todas las emociones se comportan de manera parecida o con un comportamiento distinto al explicado con *tristeza*.

Otras emociones que fueron sentidas por los oyentes fueron: melancolía, reminiscencia, calma, relajación, depresión, esperanza y ansiedad. Algunas de ellas son relacionadas de alguna manera con las emociones pretendidas por el músico; sin embargo, dada la subjetividad del contexto, las personas podrían tener diferentes percepciones acerca de las emociones relacionadas como *tristeza*, *nostalgia* y *melancolía*. En la tabla 6.4, presentamos los resultados para una prueba de Levene que es utilizada para evaluar la similitud de varianzas para una variable, calculada para dos o más grupos. En nuestro caso, la prueba es aplicada a cada canción y los grupos por canción son las emociones (*felicidad*, *serenidad*, *tristeza*, *nostalgia*, *apasionamiento*).

Esta prueba fue aplicada con 95% de confiabilidad y nos dice que la variabilidad para cada canción, con respecto a las emociones, no difiere significativamente excepto por la *Pieza Musical 5* y *Pieza Musical 6*, lo que significa que los oyentes percibieron cada canción con un mismo grado de incertidumbre.

Canciones	W	Valor p
<i>Pieza Musical 1</i>	1.731	0.146
<i>Pieza Musical 2</i>	0.377	0.824
<i>Pieza Musical 3</i>	1.833	0.125
<i>Pieza Musical 4</i>	1.353	0.253
<i>Pieza Musical 5</i>	3.289	0.0129
<i>Pieza Musical 6</i>	2.816	0.0274
<i>Pieza Musical 7</i>	1.952	0.104
<i>Pieza Musical 8</i>	0.753	0.557
<i>Pieza Musical 9</i>	1.120	0.349
<i>Pieza Musical 10</i>	0.124	0.973
<i>Pieza Musical 11</i>	1.460	0.217
<i>Pieza Musical 12</i>	0.968	0.427
<i>Pieza Musical 13</i>	0.412	0.799
<i>Pieza Musical 14</i>	0.978	0.421
<i>Pieza Musical 15</i>	1.599	0.177

Tabla 6.4: Prueba de Levene para las varianzas de las emociones en cada pieza musical

Finalmente, a pesar de estos resultados, los oyentes no reportaron comentario alguno que sugiriese una composición aleatoria de melodías a diferencia de los resultados presentados en la sección [6.1](#). Sin embargo, ellos sintieron algunas piezas con patrones melódicos similares; es decir, se repetían ciertas partes en distintas piezas.

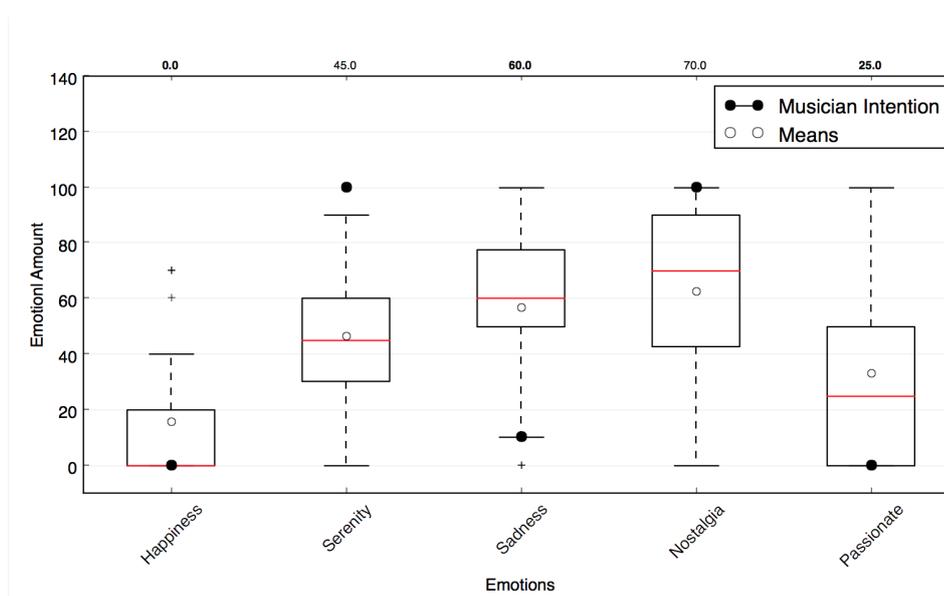


Figura 6.3: Intención Emocional y Percepciones para la Pieza Musical 1

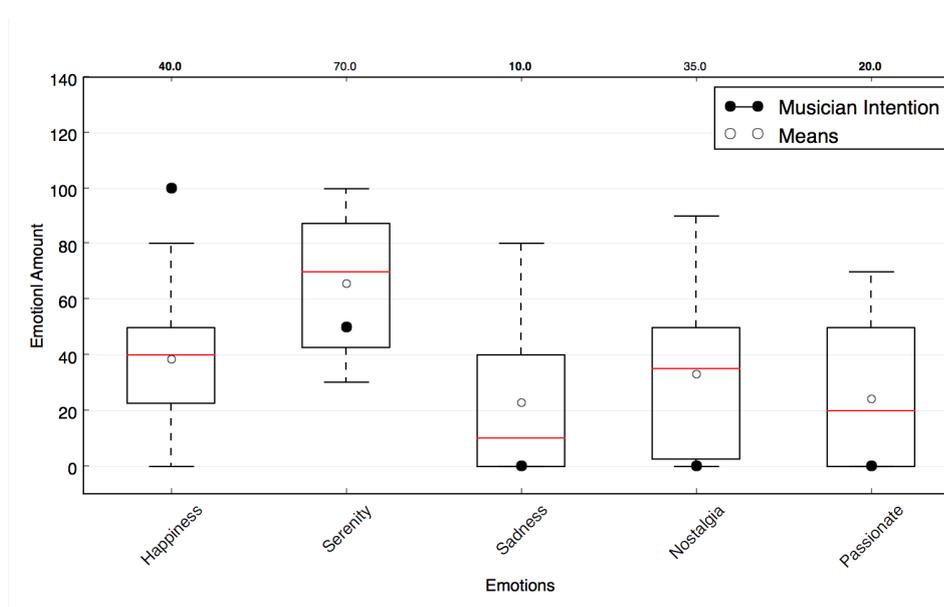


Figura 6.4: Intención Emocional y Percepciones para la Pieza Musical 2

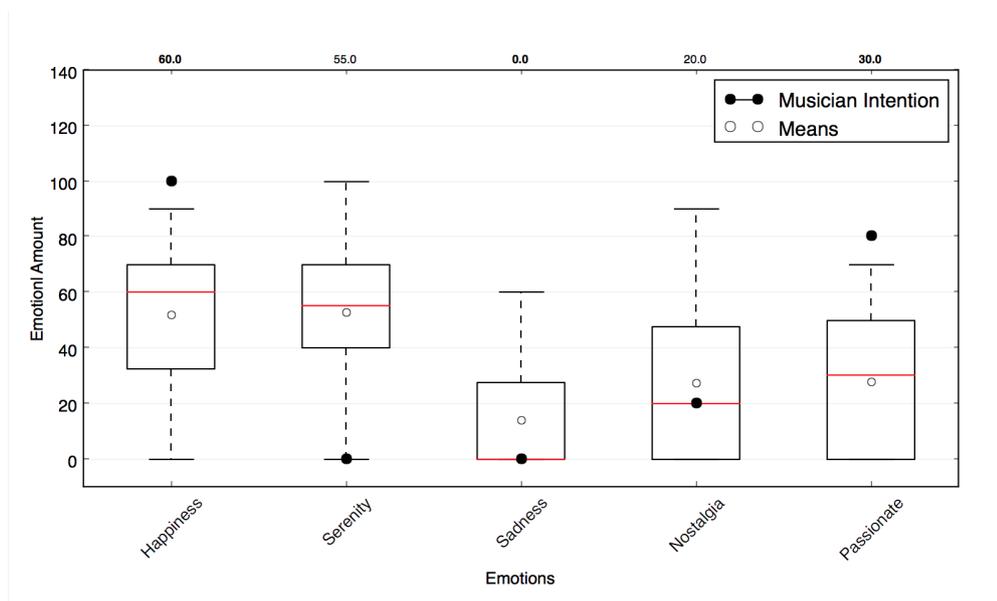


Figura 6.5: Intención Emocional y Percepciones para la Pieza Musical 3

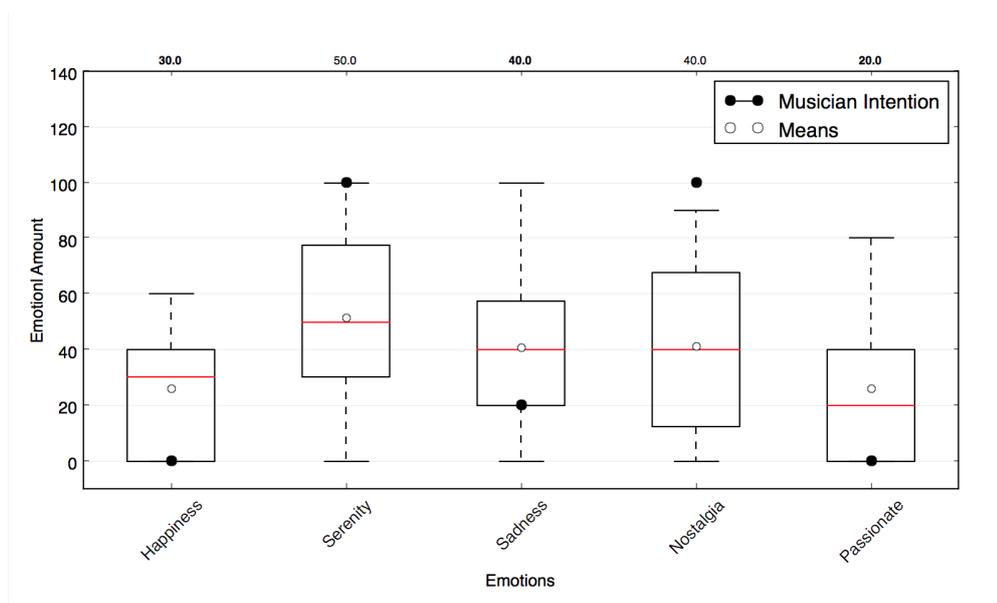


Figura 6.6: Intención Emocional y Percepciones para la Pieza Musical 4

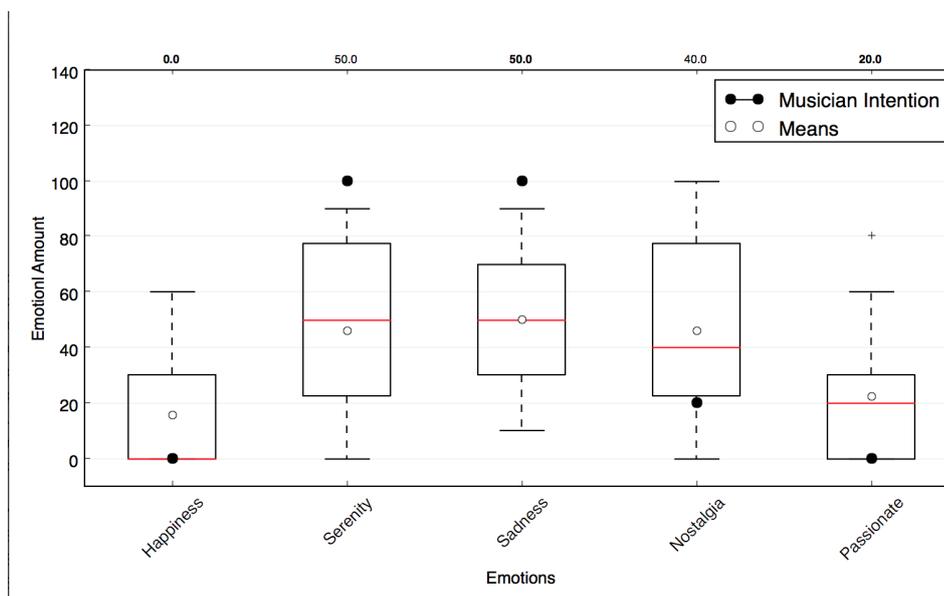


Figura 6.7: Intención Emocional y Percepciones para la Pieza Musical 5

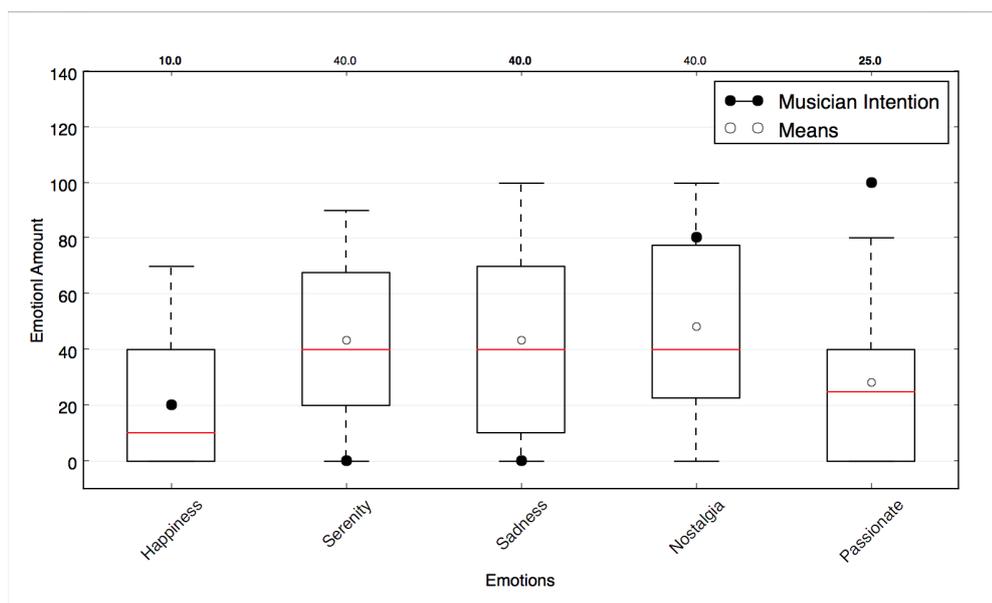


Figura 6.8: Intención Emocional y Percepciones para la Pieza Musical 6

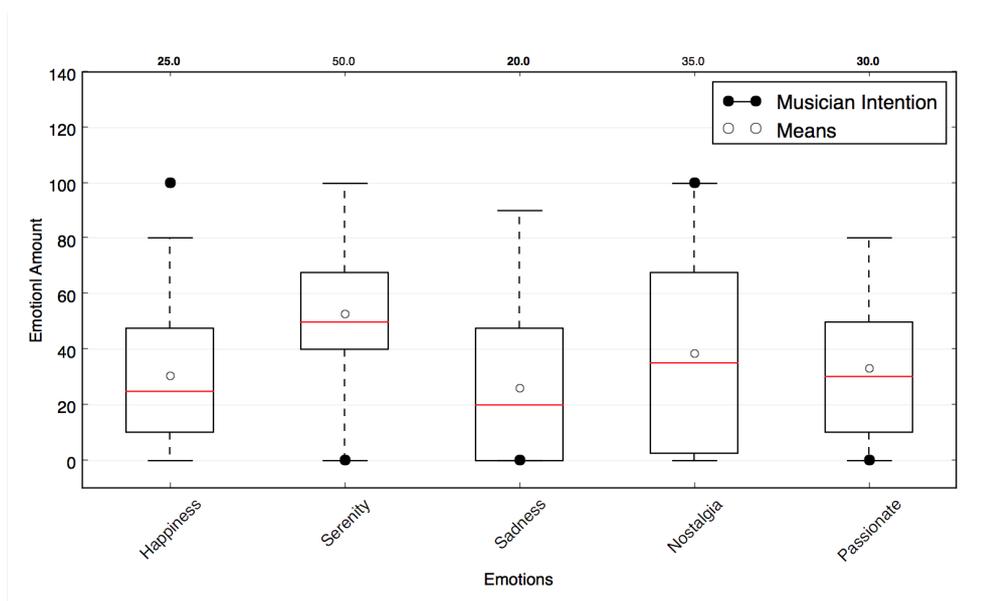


Figura 6.9: Intención Emocional y Percepciones para la Pieza Musical 7

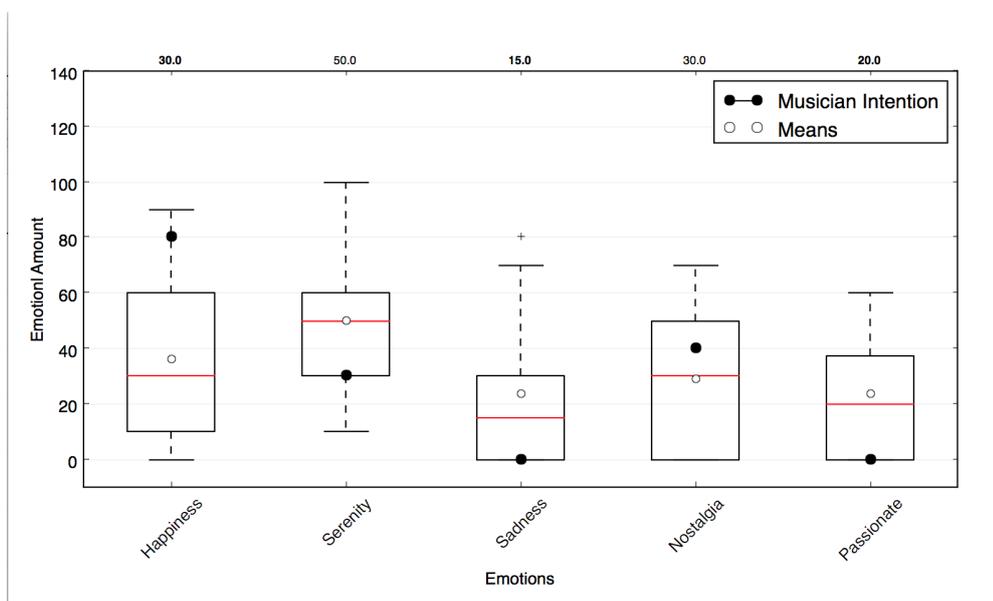


Figura 6.10: Intención Emocional y Percepciones para la Pieza Musical 8

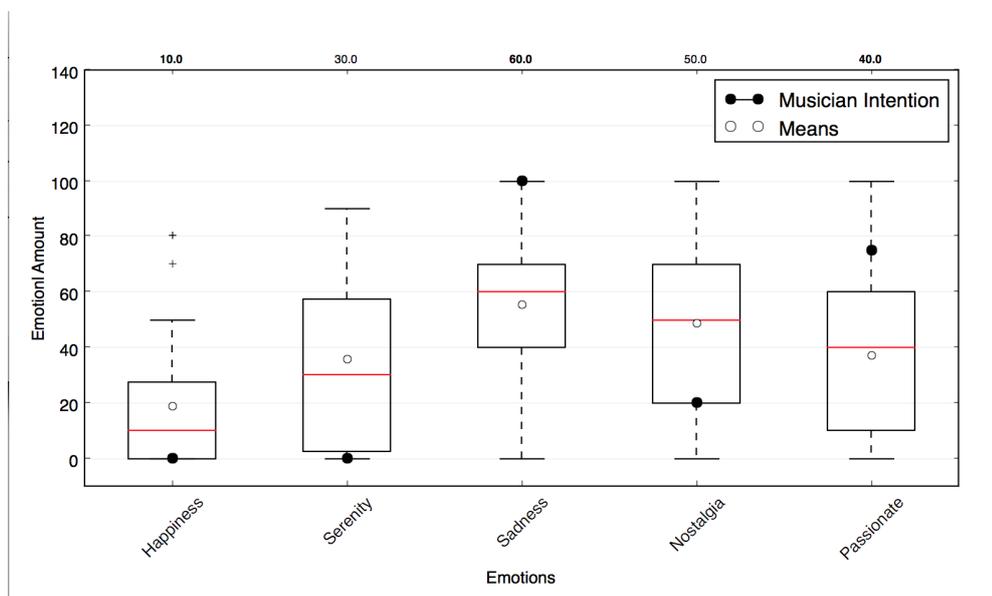


Figura 6.11: Intención Emocional y Percepciones para la Pieza Musical 9

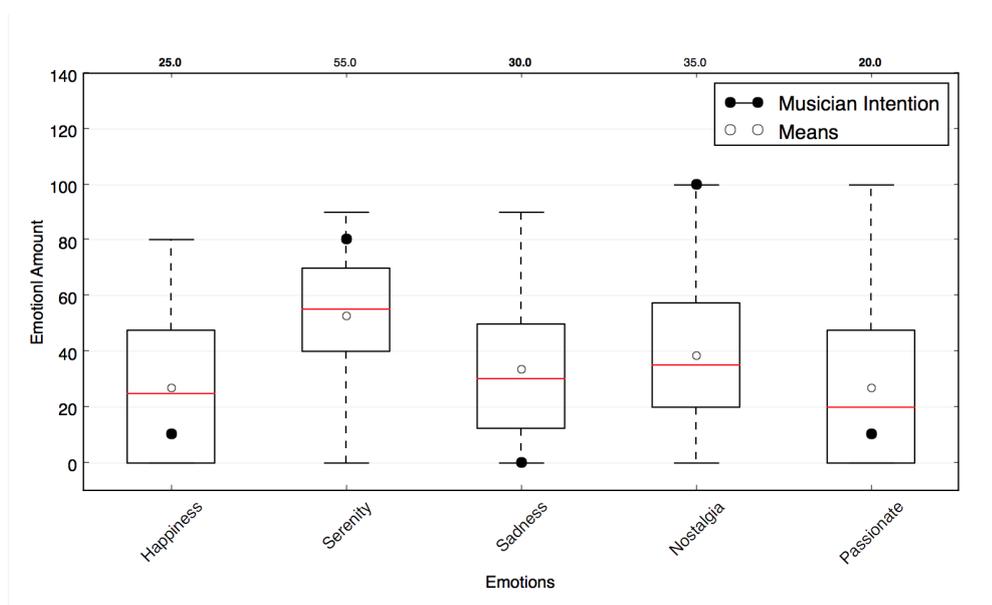


Figura 6.12: Intención Emocional y Percepciones para la Pieza Musical 10

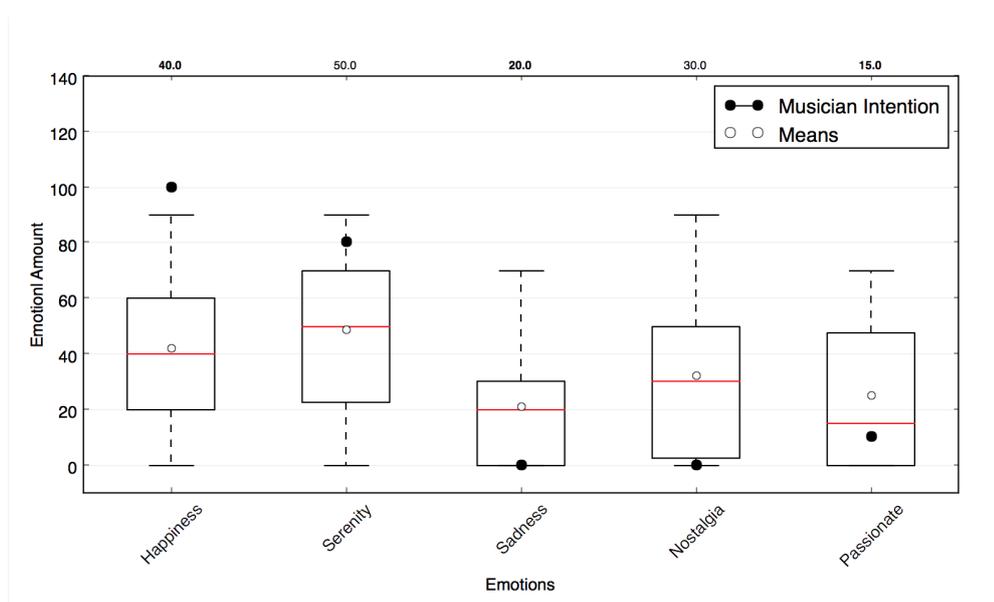


Figura 6.13: Intención Emocional y Percepciones para la Pieza Musical 11

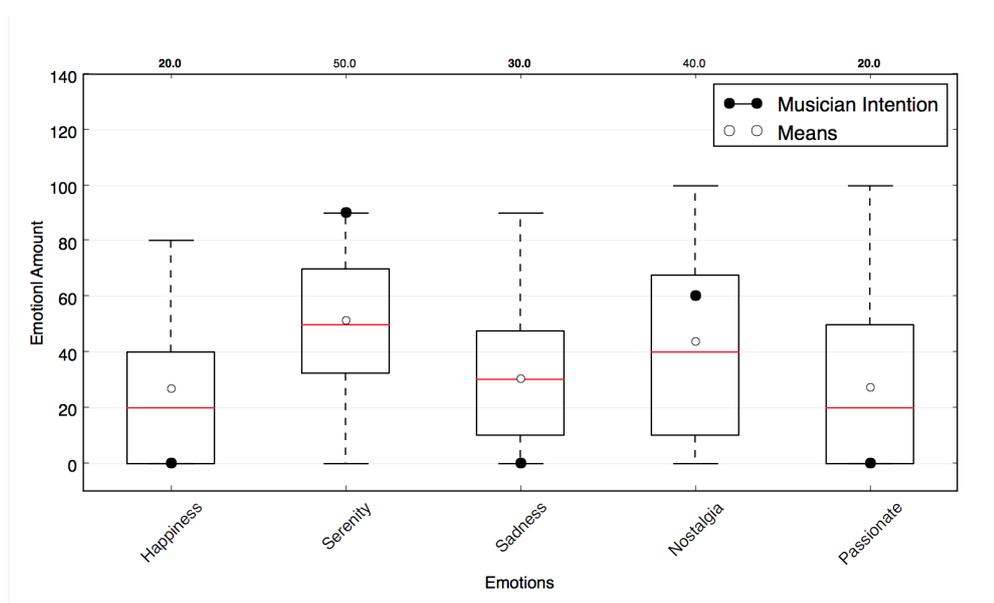


Figura 6.14: Intención Emocional y Percepciones para la Pieza Musical 12

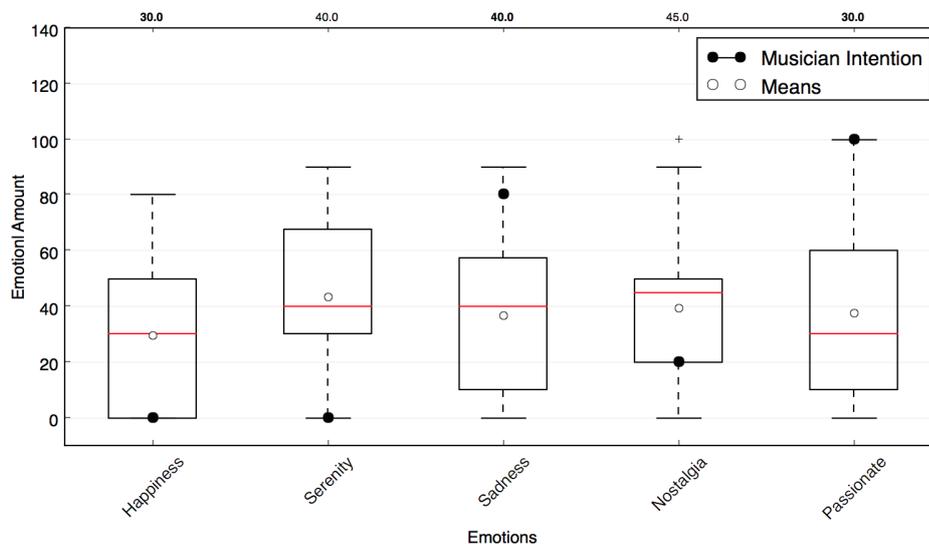


Figura 6.15: Intención Emocional y Percepciones para la Pieza Musical 13

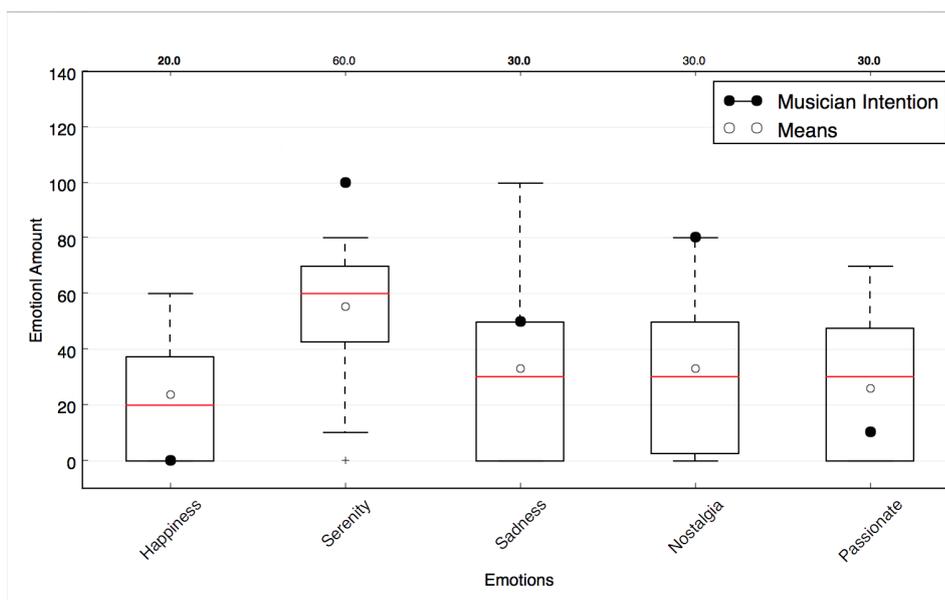


Figura 6.16: Intención Emocional y Percepciones para la Pieza Musical 14

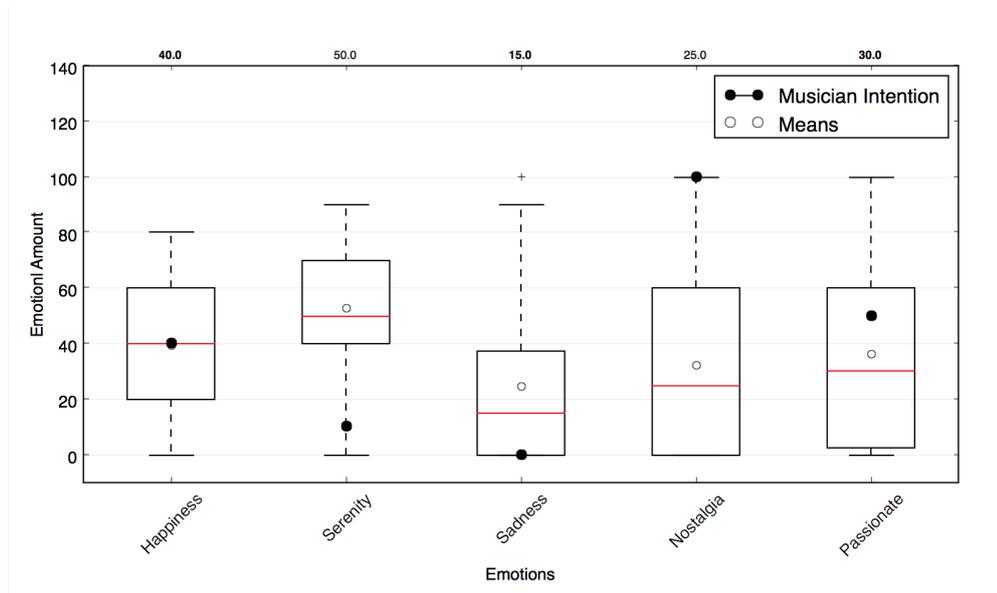


Figura 6.17: Intención Emocional y Percepciones para la Pieza Musical 15

6.3 Presentación y Análisis de los Resultados del Rendimiento del Sistema

Estas pruebas fueron ejecutadas en el contexto de la composición e interpretación de una pieza musical en tiempo real. Para ello, el sistema fue puesto en marcha en un computador con procesador *Intel Core I7* de 2.20 Ghz, 6 Gb de memoria RAM y sistema operativo Windows 8.1 de 64 bits. Se utilizó un teclado *Yamaha PSR-E413* de cinco octavas como controlador MIDI vía USB.

Considerando que es posible conectar hasta 8 entradas MIDI mediante una interfaz de hardware hacia un computador [66], se simuló la entrada de 8 hilos, los cuales copiaron los datos de los eventos MIDI desde el controlador hacia la máquina, lo que significa que ocho instrumentos musicales son manipulados por músicos reales en el exterior para alimentar el sistema en tiempo real.

Tomando en cuenta este número de entradas, se produjeron salidas del sistema a través de motor de síntesis con varias cantidades de hilos donde cada uno representa la composición en interpretación de la máquina para un instrumento musical que acompaña a los 8 hilos de entrada. Los números de hilos escogidos para realizar las pruebas siguen una escala de Fibonacci desde 1 a 55 hilos de salida (ocho números que son: 1, 2, 3, 5, 8, 13, 21, 34,

55). Se escogió esta escala para observar lo que sucede en las primeras ejecuciones con pocos hilos y para avanzar más rápidamente en cantidades altas.

Por consiguiente, se realizaron 8 procesos de toma de datos para cada número de la escala con una pieza musical de tempo 80 BPM, nota clave C Major o A minor y ejecutando el algoritmo basado en lógica difusa para la composición musical de la máquina para cada hilo de salida. Por cada proceso se tomaron 100 muestras; es decir, como las estrategias de composición se ejecutan cada beat, entonces el tiempo de duración de la toma de datos para cada proceso fue de 100 beats.

En la tabla [6.5](#), se detalla la estadística descriptiva para los **tiempos de procesamiento** cuando un flanco de tiempo (beat) ocurre considerando el número de hilos de salida. En la figura [6.18](#) se puede observar el crecimiento del tiempo de procesamiento con respecto al promedio. Según [\[67\]](#), la retroalimentación hacia un usuario debe ser menor o igual a 200 ms para que pueda ser percibida de forma inmediata, y, considerando que el músico debe percibir una respuesta inmediata en la improvisación (composición e interpretación) podemos observar que con menos de 34 hilos se está llegando a ese límite, más de ello resultaría en desincronización del tiempo musical y por ende una interpretación no agradable e imposible de seguir por

el músico real.

Hilos	N	Mín.	Máx.	Media	Mediana	Desv. Stand.
1	100	9.000	16.000	11.730	11.500	1.188
2	100	12.000	32.000	17.640	17.000	3.221
3	100	16.000	41.000	24.660	25.000	4.729
5	100	23.000	63.000	37.950	39.000	8.051
8	100	34.000	75.000	56.850	60.000	11.308
13	100	51.000	139.000	95.230	100.000	22.473
21	100	42.000	193.000	116.38	101.000	44.011
34	100	73.000	383.000	219.09	204.000	69.761
55	100	128.000	632.09	380.080	374.000	6137.281

Tabla 6.5: Estadística descriptiva de los tiempos de ejecución en milisegundos por número de hilos



Figura 6.18: Tiempos Promedio de Procesamiento

Cabe destacar que con 13 hilos de ejecución los sonidos producidos por el motor de síntesis comienzan a deteriorarse, es decir que se perciben interferencias y ruidos no deseados por la carga de procesamiento que se está soportando. Por lo tanto, el sistema con menos de 13 hilos responde adecuadamente en tiempo real con respecto al procesamiento en CPU, por lo

que el sistema puede soportar aproximadamente 21 instrumentos musicales ejecutándose al mismo tiempo si consideramos también las 8 entradas; sin embargo esto no significa que se obtendrá una buena estética ya que esto depende de los algoritmos que se ejecuten.

Conjuntamente con estos resultados, las figuras de la [6.19](#) a la [6.27](#) muestran el **porcentaje de utilización de CPU**. Nótese que mientras hay más hilos encontramos picos más pronunciados, esto es debido a que cada uno de ellos se da cuando entra un beat dado que el procesamiento es más costoso en ese lapso de tiempo. El porcentaje de procesamiento va desde valores menores al 20% para 1 hilo, hasta oscilaciones cercanas al 30% para 55 hilos. Nótese además que al inicio hay muy poco procesamiento, esto es porque el sistema es arrancado en un proceso manual donde se inicia primero al motor de síntesis para luego activar el procesamiento, lo que es reflejado en ese tiempo de baja utilización.

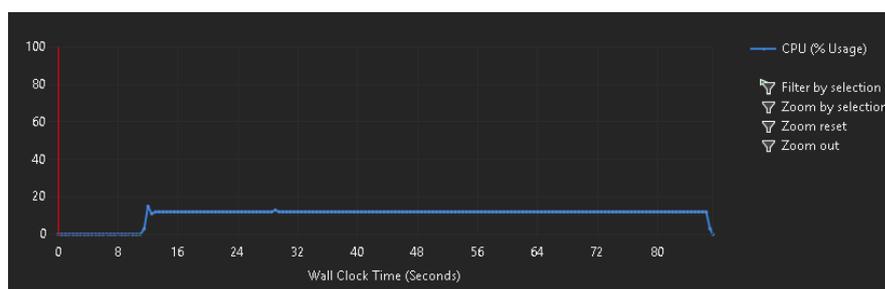


Figura 6.19: Porcentaje de Utilización de CPU con 1 Hilo

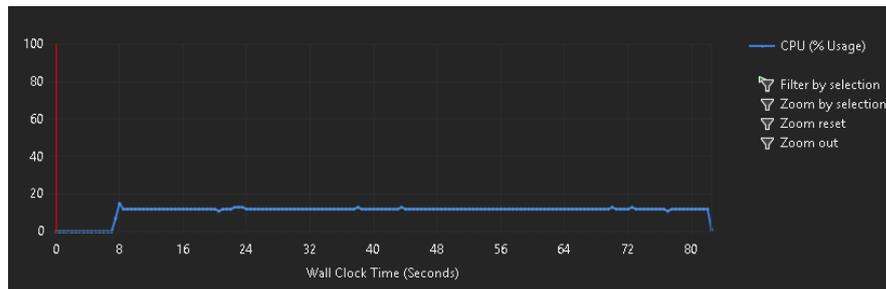


Figura 6.20: Porcentaje de Utilización de CPU con 2 Hilos

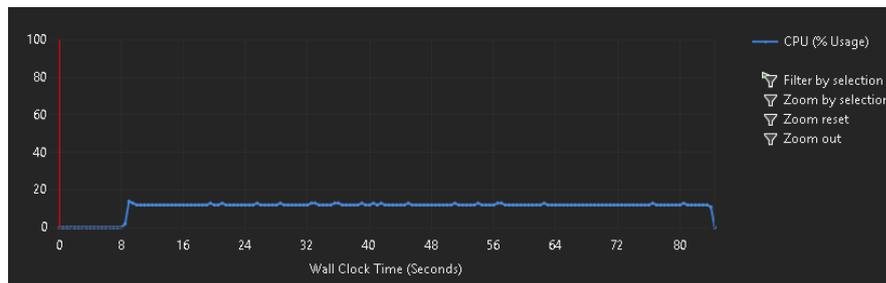


Figura 6.21: Porcentaje de Utilización de CPU con 3 Hilos

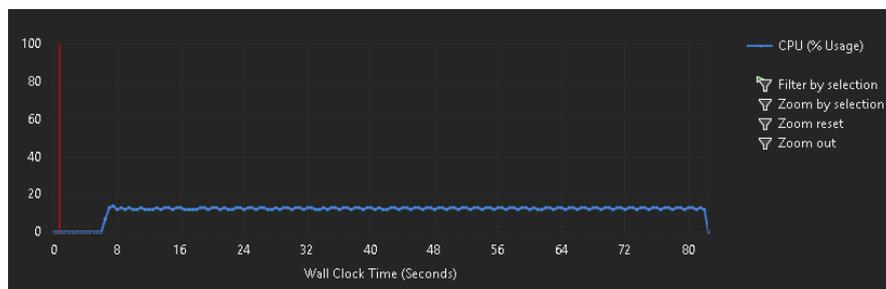


Figura 6.22: Porcentaje de Utilización de CPU con 5 Hilos

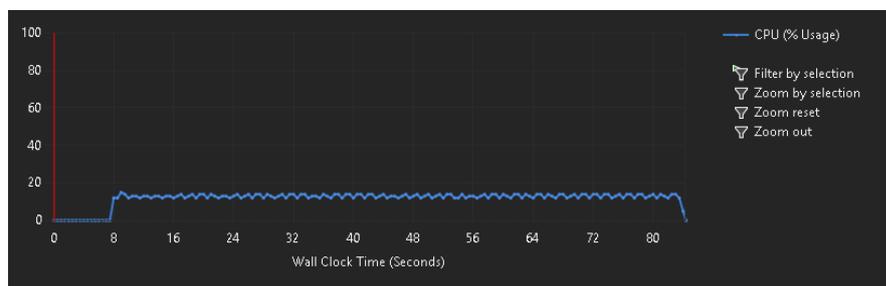


Figura 6.23: Porcentaje de Utilización de CPU con 8 Hilos

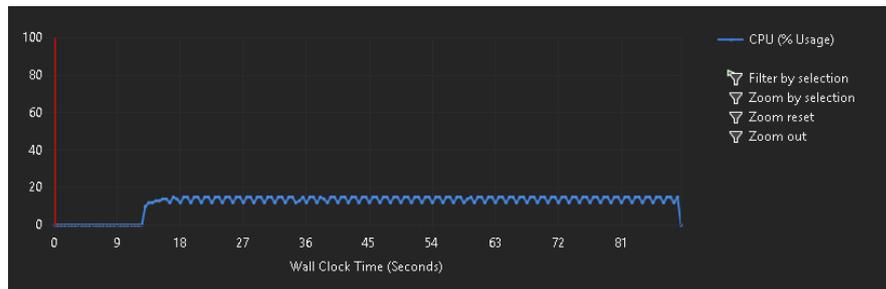


Figura 6.24: Porcentaje de Utilización de CPU con 13 Hilos

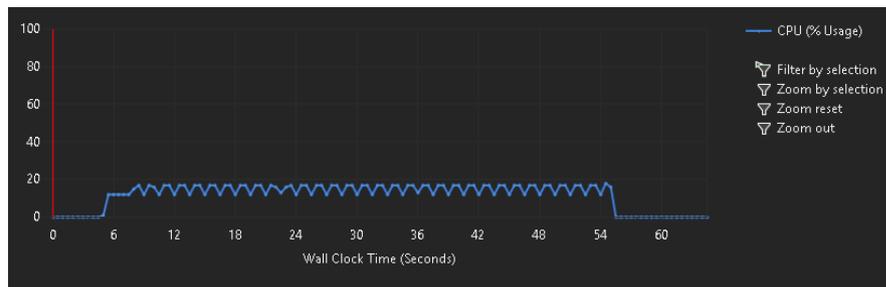


Figura 6.25: Porcentaje de Utilización de CPU con 21 Hilos

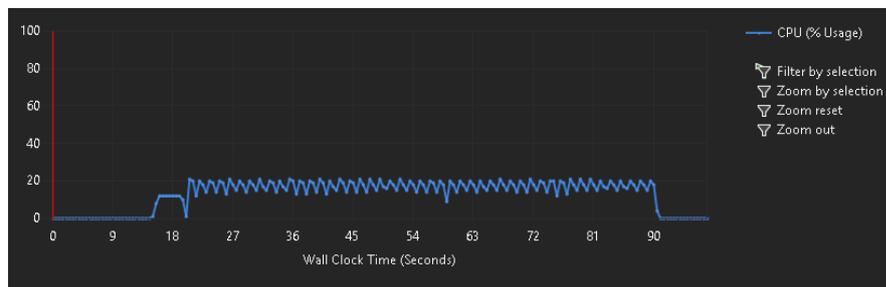


Figura 6.26: Porcentaje de Utilización de CPU con 34 Hilos

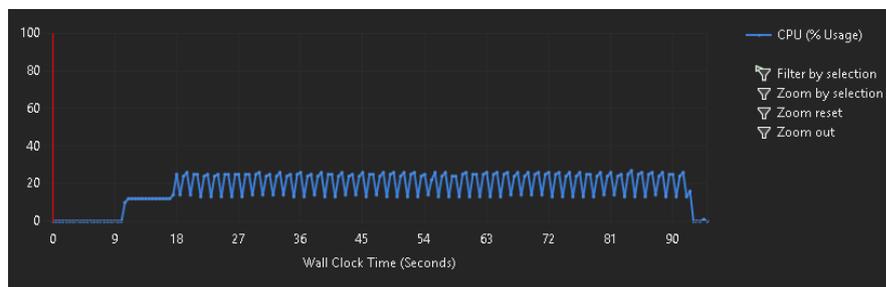


Figura 6.27: Porcentaje de Utilización de CPU con 55 Hilos

Con respecto a la *utilización de memoria*, el comportamiento de crecimiento con respecto a la cantidad de hilos de salida es mostrado en la figura [6.28](#). Estos valores dados en megabytes (Mb) fueron muestras tomadas de los resultados obtenidos del profiler de la herramienta *Visual Studio 2015* que pueden ser apreciados desde la figura [6.29](#) a la [6.37](#). Esta cantidad de memoria utilizada es debido a que cada hilo carga una copia de la base de conocimientos con el objetivo de evitar accesos continuos a disco y así ahorrar tiempo de procesamiento, además para evitar la complejidad de la gestión de recursos compartidos entre procesos concurrentes.

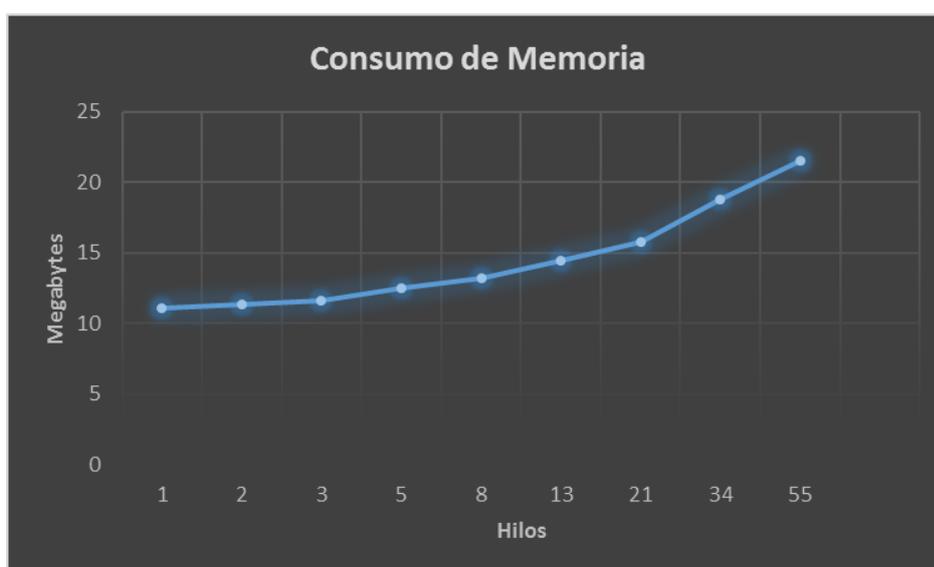


Figura 6.28: Consumo de memoria por Hilos

Nótese que desde la figura [6.29](#) a la [6.37](#) existe un bajo consumo de memoria que sube posteriormente; esto es debido a lo que se explicó anteriormente, es decir, el inicio manual del motor de síntesis y la posterior ejecución del

proceso de composición. Además, es posible observar que la memoria crece de manera lineal a lo largo del tiempo, esto es debido a que el sistema graba la interpretación musical para luego almacenarla en un archivo MIDI (.mid) si se lo requiere. Otra particularidad son los picos producidos en los gráficos asociados con una mayor cantidad de hilos, esto es debido a que la cantidad de datos asociados con las salidas crece con el número de hilos y como el procesamiento para esa salida se da en un ámbito local, entonces la memoria oscila entre crecimiento y disminución en el lapso de tiempo que dura el procesamiento.

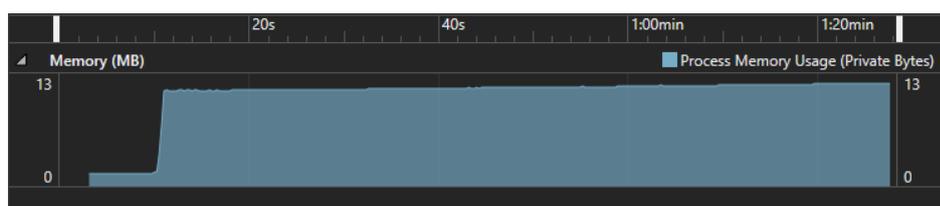


Figura 6.29: Utilización de Memoria con 1 Hilo

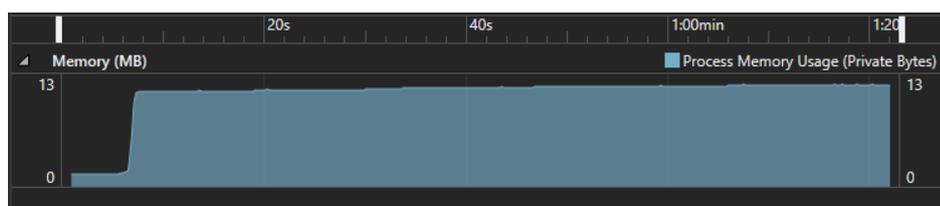


Figura 6.30: Utilización de Memoria con 2 Hilos

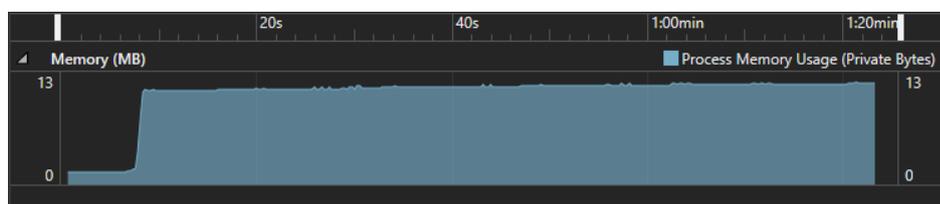


Figura 6.31: Utilización de Memoria con 3 Hilos

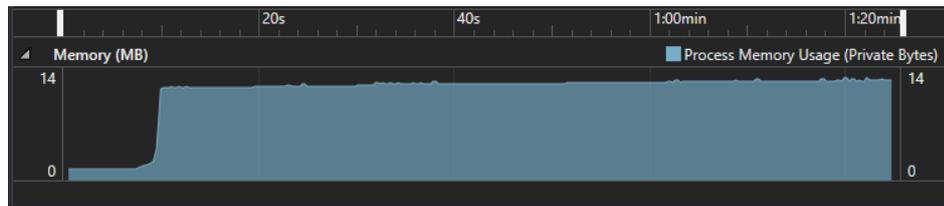


Figura 6.32: Utilización de Memoria con 5 Hilos

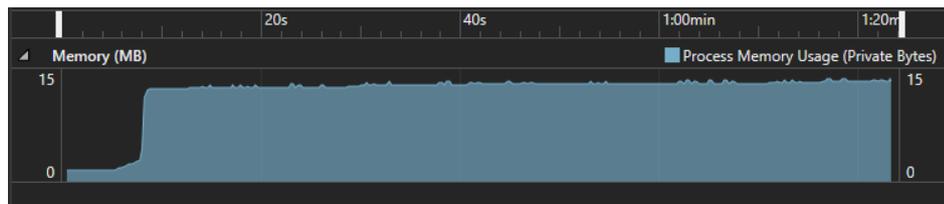


Figura 6.33: Utilización de Memoria con 8 Hilos

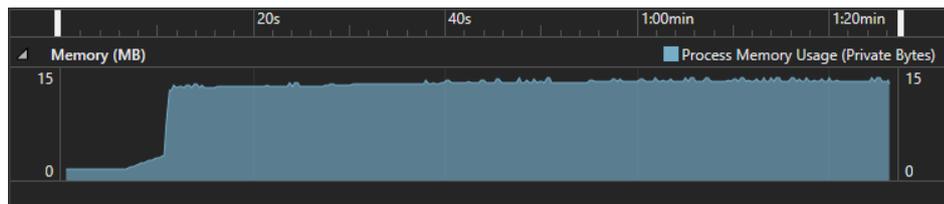


Figura 6.34: Utilización de Memoria con 13 Hilos

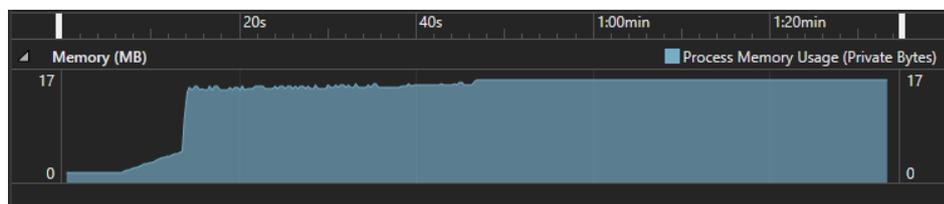


Figura 6.35: Utilización de Memoria con 21 Hilos

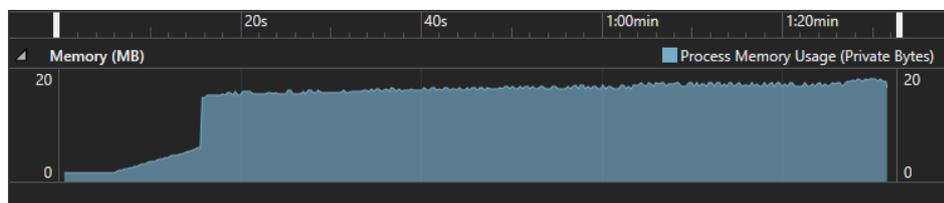


Figura 6.36: Utilización de Memoria con 34 Hilos

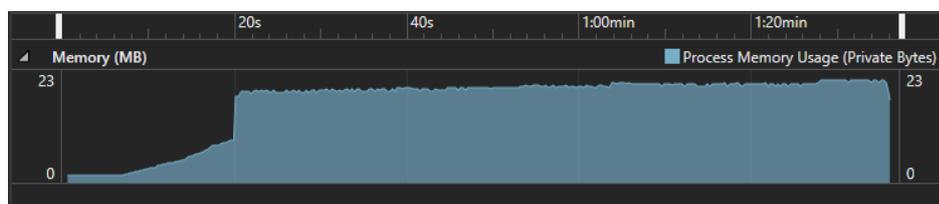


Figura 6.37: Utilización de Memoria con 55 Hilos

6.4 Análisis del proceso ingenieril aplicado a la música

Desde los lineamientos para la obtención del conocimiento relacionado a melodías musicales, hasta el desarrollo de un sistema que permita componerlas e interpretarlas en tiempo real; existió un proceso que involucró no solo la parte técnica, si no la parte artística relacionada a la música, por consiguiente, podemos establecer en base a esta experiencia, los siguientes puntos para una interacción entre ambas:

Adquirir bases elementales técnicas y artísticas: Se revisó la literatura con respecto al estado del arte en el ámbito de la Inteligencia Artificial en relación a la música, pero además se revisó la teoría musical en complemento a conocimientos empíricos que los autores poseen desde el punto de vista artístico con respecto a la música.

Establecer una guía basada en el criterio de músicos: Considerando que se posee una base elemental como se lo indicó en el punto anterior, se puede establecer una comunicación de mayor fluidez evitando la

explicación de términos o sensaciones que son familiares tanto para la persona que no es netamente músico como para el músico experto en una conversación que busca extraer lineamientos del proceso de composición musical de tal manera que sea relacionado con la revisión de literatura previa o con conceptos técnicos que puedan ser extrapolados para soportar estos procesos creativos y guiar a la concepción de un modelo cognitivo y una solución tecnológica.

Modelar el conocimiento de composición musical específico: Con los criterios de los músicos expertos, más la base elemental adquirida, se obtiene una guía para la definición de un modelo que soporte el conocimiento implícito en las obras de aquellos músicos. En el caso de este trabajo, se establecieron módulos para la representación del conocimiento musical con respecto a melodías de tal manera que el producto de esta representación es una base de conocimientos utilizada para una composición por parte de un software.

Adquirir el conocimiento de los músicos: Este es un proceso donde se definen los algoritmos que transforman los datos extraídos de los expertos al modelo de representación del conocimiento definido. En esencia es un *entrenamiento* que se realiza para nutrir la base de conocimientos. Esta acción requiere una nueva interacción con los

músicos quienes necesitan estar bajo las condiciones adecuadas (las cuales dependerá de ellos) para poder componer y evitar inhibiciones en su creatividad.

Definir las estrategias de composición musical: Sobre la base de conocimientos pueden establecerse operaciones que la manipularán de tal manera que se obtenga una interpretación musical por parte de un software hacia el exterior por medio de sonidos. Estas estrategias son algoritmos que obtienen secuencias de notas y tiempos en base al conocimiento almacenado para el caso de melodías musicales.

Definir las estrategias de interpretación musical: El resultado de la composición musical debe ser reproducido hacia el exterior, para ello se definen los métodos de interpretación que tienen como resultado la producción de sonidos basados en la composición generada. Para esto, es posible utilizar desde sonidos fijos como los proporcionados en bancos MIDI o sonidos obtenidos de un sintetizador que pueden llegar a ser complejos en su construcción.

Interacción del ente artificial con otros sujetos: El ente artificial modelado con el proceso descrito puede interactuar con otros entes, ya sean humanos o artificiales, para ello las estrategias de composición deben considerar el flujo de datos entre estos entes con el fin de

aprovechar ese aporte para el proceso de composición. En el caso de nuestro trabajo, se toman las notas de un músico real para acompañarlo con melodías generadas por el sistema, por lo que es una interacción la cual llamamos *Humano-Máquina*.

Implementación de un soporte tecnológico: En base al proceso descrito, se desarrolla un sistema utilizando los criterios y técnicas que nos brindan las ciencias computacionales asegurando la calidad del producto y una utilización acorde a los usuarios que la usarían, que en este caso son artistas, tomando siempre en cuenta que para ellos es un aporte a su proceso creativo, mas no un reemplazo de ellos.

Cabe destacar que la experiencia de la interacción entre personas del área técnica con personas del área artística es enriquecedora para ambas ramas. Los artistas aportan con ideas para mejorar su proceso creativo las cuales son retos tecnológicos interesantes para un ingeniero y que pueden ser concretadas siguiendo una metodología iterativa que permita probar constantemente las soluciones que podrían desarrollarse para ellos.

Una acotación importante cuando se trata con personas del área musical es que ellos no pueden estructurar estrictamente su proceso creativo, sobretodo cuando se trata de música experimental donde el objetivo está en innovar, es por ello que en el proceso ingenieril descrito está considerada la extracción

del conocimiento implícito a través de sus obras, pero además, algo que no fue realizado en este trabajo, fue la observación constante de su comportamiento a la hora de componer, la obtención de las fuentes de inspiración y demás material que aporta a la composición; es decir, un seguimiento estricto del músico. Esta idea del *seguimiento estricto* fue mencionada por los músicos entrevistados con la observación de que ellos están sumergidos tan profundamente en su proceso que otras personas deberían ser las encargadas de este seguimiento para evitar la inhibición de la creatividad. El aporte de este seguimiento tendría una gran significancia en las mejoras para el ámbito de composición.

CONCLUSIONES Y RECOMENDACIONES

En contraste con los objetivos propuestos en el presente trabajo, podemos establecer las siguientes conclusiones y recomendaciones.

Conclusiones

1. Se levantó el estado del arte con respecto a las técnicas de inteligencia artificial relacionadas al ámbito musical, de donde se tomaron criterios de ingeniería del conocimiento aplicados a artistas practicantes de la música experimental, para guiar a la construcción de una solución tecnológica que componga e interprete melodías musicales de manera concurrente en tiempo real con carácter emocional. Por lo tanto, considerando que los resultados en las pruebas realizadas fueron satisfactorios, el enfoque escogido fue adecuado para el desarrollo de esta solución específica.
2. Las estructuras de datos utilizadas en la implementación tales como arreglos dinámicos, árboles balanceados, matrices, clases personalizadas y `templates` fueron determinantes en el rendimiento del sistema lo que permitió que se procesen 8 entradas concurrentes

para producir hasta 13 hilos de ejecución para la composición musical en cada beat del metrónomo con un aproximado de 100 milisegundos de procesamiento sin deteriorar el sonido y sin producir retardos que influyan en la desincronización del tiempo musical de la pieza. Cabe destacar que estas pruebas no consideraron estética y su objetivo fue la identificación de los límites del sistema.

3. Se utilizaron varias herramientas para la implementación, donde se consideró hardware y software para el desarrollo del sistema, por lo que este sistema es un híbrido entre varias tecnologías desde la adquisición de datos de un controlador MIDI hasta la reproducción de los resultados por medio de parlantes. Sistemas como estos demandan un grado de complejidad en la integración de estas herramientas y en la implementación del núcleo que las sostiene.
4. La estrategia de generación de melodías aplicada a las matrices de transiciones basadas en Cadenas de Markov promete ser un enfoque adecuado cuando se provee una base armónica sobre la cuál sea reproducida una melodía en particular como se lo realizó al agregar el componente de lógica difusa. Además, considerando la versión de la prueba de Turing realizada, los resultados de la composición de este enfoque son naturales para varios músicos como si se tratara de un

artista real.

5. La ejecución del sistema en tiempos precisos y casi inmediatos permitió una sincronización adecuada entre el músico real y el músico artificial, dando lugar a una interacción similar a dos músicos reales con una conexión emocional en relación a la obra que se interpreta.
6. En la interpretación en tiempo real para la realización de las pruebas, se reportó que algunas veces la intención emocional del músico real cambiaba un poco para ser consistente con lo que el sistema estaba reproduciendo; sin embargo, esta situación no afectó significativamente a la composición, es por ello que el sistema no se restringe a lo que se requiere que produzca, sino que contribuye al proceso de composición con su propio *estilo*.
7. Este inesperado cambio mencionado en el punto anterior, podría haber causado que la percepción emocional evaluada en los oyentes no fuera similar de una manera significativa con la intención emocional del músico real, como lo sugieren los resultados. Además, este experimento no controló el estado emocional de cada oyente, por lo que esto pudo haber influenciado en las respuestas; sin embargo, la variabilidad con respecto a estas respuestas es similar en cada una de las 15 piezas musicales de la prueba, lo que significa que existe un grado de

subjetividad a ser considerado cuando las personas escuchan una canción que evita tener una tendencia marcada.

8. A pesar de la conclusión del punto anterior, los oyentes reportaron que sintieron algunas de las emociones establecidas para la prueba (*felicidad, serenidad, tristeza, nostalgia, apasionamiento*), e incluso otras tales como melancolía, reminiscencia, calma, relajación, depresión, esperanza y ansiedad. Estos resultados demuestran que el enfoque propuesto influencia sentimentalmente en las personas familiarizadas con la música occidental.
9. Las estrategias de representación del conocimiento para músicos experimentales artificiales, deben tomar en cuenta el proceso creativo tanto en lo explícito como lo implícito; es decir, argumentar las representaciones en base a la identificación de sus ideologías, influencias, actividades e incluso equipos como los sintetizadores, lo cual sería lo explícito, y en lo implícito extraer ese conocimiento directamente de sus creaciones.
10. La implementación de sistemas de composición e interpretación, que asistan a músicos en su proceso creativo, debe considerar los elementos de la obra que ellos quisieran mejorar estéticamente, esto implica en primera instancia un entendimiento del proceso de

composición; como el realizado en este trabajo, con el objetivo de generar recomendaciones por parte del sistema; sin embargo, existen límites impuestos por los artistas, como el no aceptar que la máquina sea un músico más, y por tanto el enfoque de este tipo de sistemas debe ser de colaboración, con el objetivo de cortar camino para llegar a componer obras intensamente emocionales y condensadas, tal como lo mencionaron los músicos entrevistados.

11. El proceso ingenieril que se desarrolló en este trabajo fue descrito para que sea considerado como un conjunto de directrices que aporten al grupo de conocimientos adquiridos por un estudiante de ingeniería según los objetivos del proyecto ANDAMIOS y fomentar la interacción entre el área técnica y el área artística en el ámbito de la música.

Recomendaciones

1. Explorar otras áreas en el ámbito de la música para utilizar un enfoque similar que implique la utilización de técnicas de inteligencia artificial. Este trabajo se basó fundamentalmente en *melodías*, sin embargo, se sugiere la exploración de *armonías* y *ritmos*, así como la expresividad en la *interpretación musical* que implica el ámbito de los movimientos

físicos que un intérprete musical puede realizar en un instrumento.

2. Extender el aporte de la síntesis de sonido con respecto a las emociones producidas por una obra musical, lo que implica estudiar la influencia de los sonidos sobre las personas y registrar las configuraciones de los elementos que componen esos sonidos para que pueden ser modelados y utilizados en aplicaciones con entes artificiales y así complementar el ámbito de composición musical.
3. Realizar un mejor control de los experimentos relacionados a parámetros subjetivos como las emociones humanas, lo que implica un mayor esfuerzo en tiempo y recursos, pero brindará un mayor beneficio al generar resultados que tengan la menor cantidad de sesgo posible.
4. Proponer y probar modelos perceptivos de emoción humana para los oyentes de tal manera que se aumente la objetividad en la evaluación de pruebas ya que la comparación sería en contra del modelo. Esta investigación es adecuada para el área de la psicología cognitiva.
5. Generar nuevas soluciones en base a los lineamientos (requerimientos) proveídos por los músicos, soluciones que pudieran o no requerir la utilización de técnicas de inteligencia artificial de tal manera que se soporte el proceso creativo de los músicos para elevar la calidad de las

obras en tiempos más cortos.

6. Utilizar los modelos y tecnologías presentados en este trabajo para aplicaciones multimedia en tiempo real tales como juegos de video o experiencias interactivas que requieran diseño de sonido dinámico en el ámbito musical.
7. Impartir el proceso realizado en este trabajo a estudiantes de ingeniería para expandir su abanico de habilidades apuntando a áreas artísticas, en este caso, la música.

APÉNDICE A

LINEAMIENTOS OBTENIDOS DE LA ENTREVISTA NO ESTRUCTURADA APLICADA A LOS MÚSICOS EXPERIMENTALES

1. La música es para ellos es su vida, un movimiento de notas y acordes que va de acuerdo a lo que el alma o estado anímico desea comunicar, de manera técnica es matemática, a nivel de síntesis es la oportunidad de diseñar sus propios sonidos y producir esa música desde cero.
2. Su música la comparan con una fotografía porque para lograrla existen varios filtros; la lente, la cámara, el ojo humano. Estos filtros, son capas que pueden cambiar la perspectiva de esa fotografía dependiendo del estado en que se encuentren, es una estructura totalmente variable donde los más minúsculos cambios podrían lograr algo considerablemente distinto en el resultado final.
3. Los sentimientos tienen muchas matices, es decir que por ejemplo no se puede catalogar completamente una composición con un carácter triste porque para unas personas será triste o melancólica, pero para otras

actuará como calmante o relajante. Desde ese punto de vista ellos expresan sorpresa por el hecho de que su música puede alcanzar un rango de sensaciones en vez de una sola; lo cual los alimenta para lograr esta variabilidad en su composición. Generalmente ellos se enfocan en un sentimiento, pero al final no conocen el sentimiento en su totalidad, simplemente promueven una tendencia.

4. Al momento de componer ellos realizan todo de manera granulada; el todo lo seccionan en pequeñas partes minimalistas. Por ejemplo con 2 o 3 notas para una melodía que podría expresar algo de un todo significativo. Atrás de esas melodías se desarrolla todo un mundo (fondo de armonización), es decir existe un cierto grado de complejidad alta comparado con las melodías que están adelante.
5. Para lograr esas pequeñas piezas existen elementos intrínsecos en ellos que no pudieron explicarlos los cuales son también granulados por su asociación hacia la producción de esas partes minimalistas , como las 2 o 3 notas mencionadas anteriormente, existió algo interno que llevó a su composición.
6. Este conjunto de elementos granulados permite que el sentimiento al cuál se quiere aludir (recordemos que a pesar de que existe un rango se busca explorar un sentimiento particular) sea estructurado en varias

piezas donde cada uno es sonido y melodía, es algo minimalista que se entiende a primera instancia.

7. Las típicas notas y acordes como los utilizados por ejemplo en los años 60 ya no funcionan para ellos, los clichés y la introducción de la voz humana con un estilo pop no son utilizados en su música.
8. Patrones estándares de las masas como por ejemplo que los acordes menores son conocidos como acordes de tristeza son considerados lineales. Por lo tanto la música que posee estos patrones también es lineal. Sin embargo, cuando la música es vista por ellos por este compendio de elementos granulados ya no es lineal, es algo que se contempla hacia arriba, con esto ellos tratan de decir que existe evolución en la música y que no debe quedar solo en la tradicional.
9. Como se mencionó anteriormente, estos elementos granulados que pueden ser asociados a elementos musicales básicos como notas en una melodía son difíciles de definir, aunque están influenciados por otras piezas musicales, libros, las experiencias del día a día, etc.
10. Eso que es tan difícil de definir ellos lo denominan la cualidad intangible, la cual a ellos les gustaría entender, pero es casi imposible. Esta cualidad intangible es en esencia la cualidad sin nombre, explicada en la

introducción.

11. Ellos piensan que esta cualidad no se puede explicar y que solo se puede demostrar por hechos, para este caso, en su música. Es como la existencia de una planta la cual no se puede explicar y que para los humanos solo queda disfrutarla.
12. Las personas dicen que la música es un lenguaje, existe una fascinación en el mundo por verla de esa manera, resulta obvio como cuando afirmamos que el agua nos hidrata, pero ellos sostienen que el lenguaje natural (el que utilizamos para comunicarnos) está en decadencia y por ello en cierta forma es una ofensa decir que la música es un lenguaje, más bien la música tiene un lenguaje cuya evolución comprende desde el hecho del entendimiento, luego la estructura para transmitir ese entendimiento y finalmente la poesía que es la utilización sublime del lenguaje para al final resolverse en música complementada con la técnica.
13. Con respecto a la integración de músicos con sus mismo lineamientos, ellos no desean enseñar, más bien la integración de otras personas debería ser análogo a las antiguas logias donde más que impartir enseñanza para ser como ellos les gustaría nuevas ideas para enriquecer su conocimiento y complementar sus obras. Estas personas

deben ser innovadoras, violentas, osadas.

14. Otro punto con respecto a enseñanza es que ellos no lo pudieran hacer como la academia lo realiza, porque su proceso es exploratorio, se consideran un laboratorio y donde además se debe aprender a explorar, contemplar el entorno y experimentar, tal como pasó con la música de Johann Sebastian Bach y su descubrimiento del contrapunto (melodías distintas tocadas simultáneamente de tal manera que armonizan como un todo) lo cual era algo totalmente radical para la época.
15. Con respecto a su proceso creativo, ellos pueden describir como sería el proceso de composición para por ejemplo el Jazz, sin embargo el proceso para obtener su música no pueden estructurarlo tan rígidamente por el hecho de que ella es experimental.
16. Para tener constancia de las partes granuladas compuestas, ellos registran diariamente su trabajo por medio de grabadoras de audio y anotaciones de la configuración de los sintetizadores y pedales para la obtención de sonidos particulares.
17. Aun así con el registro de esta información, es difícil replicar esa información a todas las instancias; es decir, si tengo un registro completo de la configuración de un sintetizador analógico para un

sonido y ritmo particular, probablemente no sea igual en otro equipo debido a su interacción con otros equipos predeterminados en el lugar o por otros factores externos ambientales como el clima.

18. Con respecto a la síntesis de sonido, ella no es puntual y es el vivo ejemplo de la minuciosidad en su música ya que el movimiento de milímetros para una perilla en un sintetizador puede cambiar totalmente la textura del sonido generado, por eso ellos graban una gran cantidad de estos sonidos por el hecho de que muy probablemente se perdería para siempre si no se lo registra.

19. Para estructurar una obra final se basan en influencias de bandas musicales acorde a lo que están creando, su proceso de creación podría ser infinito pero existe un punto en donde ellos simplemente dicen hasta aquí. La estructura se deriva en varias secciones que las suelen llamar parte A, parte B, parte C y así sucesivamente, existen puentes entre esas partes y cada una de ellas tiene un matiz peculiar que en sí sola desea comunicar algo y al mismo tiempo es congruente con el todo, por ejemplo de manera básica se puede decir que para una canción X, su parte A es fuerte, la parte B es mucho más fuerte, C es sutil, y D puede estar en medio de los matices A y C. Todo esto viene a ser un grupo de patrones que permiten estructurar el producto final.

20. Los patrones que surgen tienen que ver con lo que se desea expresar y al mismo tiempo con lo que no se quiere expresar; es decir, que existe una intención, pero intrínsecamente está el recuerdo de lo que somos como humanos y nuestras experiencias lo cual es aquello que no se expresa intencionalmente y depende del estado actual en que ellos se encuentren.
21. El sentimiento nace con aquel estado actual y es llevado a producción, luego lo que se obtiene alimenta otra visión del sentimiento u otros sentimientos, después eso es utilizado para seguir produciendo y así ellos pueden estar inmersos en este proceso iterativo que logra expandir sus esfuerzos a una composición rica.
22. Es posible quedar atrapado en un círculo infinito, pero llega un punto en donde hay que cortar. Ellos creen que son muy jóvenes aún para lograr condensar sus obras de tal manera que las sensaciones sean fuertes con un tamaño de la obra calculada lo más exacta posible por ellos, para lograrlo es cuestión de evolución y exploración constante.
23. Su influencia no solo viene de músicos, sino de otras artes como la visual y la escrita, pero en realidad ellos consideran que todo es influencia, el mudo es su inspiración. Además consideran muy importante la retroalimentación de otros músicos que no están inmiscuidos con ellos en su proceso pero son cercanos.

24. Para ellos no hay un momento adecuado para componer (haciendo alusión a estados mentales y emocionales) pero si una hora adecuada que es mayoritariamente en la noche y un entorno que no limite su fluidez en la composición.
25. La academia los ha ayudado a ganar disciplina, sin embargo mencionan que aprendieron la técnica para destruirla en su totalidad, ellos consideran la academia como una zona de confort de la cual hay que salir para observar otros horizontes.
26. Cuando ellos no pueden fluir en la composición en algo particular simplemente se lo deja ir y se inicia de nuevo porque ese es un indicador de que lo producido no es lo suficientemente maduro con respecto a lo que se quiere transmitir. Desde ese punto de vista ellos consideran que no hay bloqueos creativos mentales, más bien las condiciones ambientales y la falta de equipos es lo que impedirían que desarrollen su música.
27. Las melodías no las consideran como escalas o patrones únicamente, más bien ellas son historias, las cuales siguen un trayecto, pasan por una etapa problemática, existe un clímax, y poseen una resolución. Dependiendo del estado en que los músicos se encuentran se define cual es el punto hacia donde esa historia va a llegar.

28. La parte de armonías refuerzan esas historias rompiendo ciertas normas y produciendo todo un mundo tratando de evitar caer en el cliché.
29. Ellos no creen que su música sea eterna porque consideran que esa característica la tiene la música popular debido a que su música no vibra con el sentimiento común de las masas, por lo tanto su audiencia debe estar fuera de esa masa, aun así ellos no han presentado sus obras a una gran cantidad de personas para evaluar el impacto de ellas.
30. La síntesis es la nueva forma de hacer música, el renacimiento de ella en el siglo 21, con una amplitud inigualable, microscópica y expansiva al mismo tiempo. Un sintetizador es la expresión física de eso, es como un ser vivo que no es comparable a un instrumento tradicional como una guitarra.
31. Existen métodos para poder manipular los sintetizadores, pero queda en el artista hacer uso de esos métodos para combinarlos y obtener un resultado innovador haciendo uso de armonías, melodías y ritmos que se pueden producir en estos instrumentos. Se vuelve una tarea minuciosa y experimental donde se quiere que los sonidos en sí entre en el rango de emociones que se desea transmitir.
32. Los distintos tipos de sintetizadores pueden colocar distintos

sentimientos y estados a la obra, no existe un equipo “todólogo” ya que ellos ven a un sintetizador como un panel con pinturas, donde las pinturas representan la calidad del sintetizador, estas pinturas determinan el grado de eficacia en la obra sobre todo con los matices pequeños.

33. Ellos están conforme con una pieza cuando los hace estremecer aun siendo escuchada en promedio diez veces, sin embargo luego de un tiempo (generalmente unos 3 meses) sienten que hay algo más que falta, que es imperfecta, lo cual señala que ellos están progresando en su nivel de composición y que existe evolución en su conocimiento.
34. Con respecto a una herramienta tecnológica que componga música creen que lo mejor sería una máquina de asesoría que se encuentre alineada con su proceso de composición de tal manera que sugiera ideas y corrija elementos puntuales, por ejemplo recomendar la frecuencia correcta para ecualizar un bajo dependiendo de lo que se desea transmitir en la obra, etc. El objetivo de aquella máquina sería acortar el camino para llegar más rápido a concretar un conocimiento trascendental en la música que actualmente poco a poco están cultivando en su experimentación, abrir un abanico de posibilidades con el fin de encontrar ese algo que a veces falta. Para cada sugerencia esta

máquina explicaría qué llevó a tomar una decisión y por qué el músico está fallando cuando ese sea el caso. Esta máquina tendría una experiencia condensada donde está concentrado el conocimiento útil para las sugerencias con el fin de no desviar el camino de la obra.

35. Ellos no aceptan la máquina como un músico más que haga totalmente todo, no es natural para ellos, sin embargo para reducir espacios sería una herramienta de ensueño.

BIBLIOGRAFÍA

- [1] Lopez de Mantaras, R., Making Music with AI: Some examples, *In Proceedings of the 2006 conference on Rob Milne: A Tribute to a Pioneering AI Scientist, Entrepreneur and Mountaineer*, páginas 90–100, 2006.

- [2] Fernández, J. y Vico, F., AI methods in algorithmic composition: a comprehensive survey, *Journal of Artificial Intelligence Research* 48, vol. 48, páginas 513–582, 2013.

- [3] Willi Apel, *Harvard Dictionary of Music*, 2nd ed. Cambridge, Massachusetts: The Belknap Press of Harvard University Press, 1974.

- [4] Alexander, C., *The Timeless Way of Building*. Oxford University Press, 1979.

- [5] Holmes, T., *Electronic and Experimental Music: Pioneers in Technology and Composition*, 3rd ed. London and New York: Routledge, 2008.

- [6] Nierhaus, G., *Algorithmic Composition: Paradigms of Automatic Music Generation*. Graz: Springer-VErlag/Wien, 2009.

- [7] Community, S., SuperCollider. <http://supercollider.sourceforge.net/>, fecha de consulta 2014-01-01
- [8] Wikipedia, Experimental music. http://en.wikipedia.org/wiki/Experimental_music
- [9] Dictionary, O., Music. <http://dictionary.onmusic.org/terms/2278-music>, fecha de consulta 2015-04-11
- [10] Hewitt, M., Music Theory for Computer Musicians. Boston, USA: Course Technology PTR, 2008.
- [11] Davies, S., Musical Chords and Scales. http://www.cs.cmu.edu/~scottd/chords_and_scales/music.html, fecha de consulta 2015-02-15
- [12] Newlin, D., Schoenberg, A., y Strang, G., Fundamentals of Musical Composition, 1968, vol. 55.
- [13] Thompson, W. F., Bella, S. D., y Keller, P. E., Music Performance, en *Advances in Cognitive Psychology*, vol. 2, 2006, páginas 99–102.
- [14] Russ, M., Sound Synthesis and Sampling. Taylor & Francis, 2009.
- [15] Wikibooks, Sound Synthesis Theory/Oscillators and Wave-tables. http://en.wikibooks.org/wiki/Sound_Synthesis_Theory/Oscillators
-

[and_Wavetables](#), fecha de consulta 2015-04-14

[16] InnovativeSynthesis, Basic Synthesis: Part 2 - Filters. <http://www.innovativesynthesis.com/basic-synthesis-part-2-%E2%80%93-filters/>, fecha de consulta 2015-04-14

[17] Smith, S. W. S., The scientist and engineer's guide to digital signal processing, Smith, S. W., Ed. California Technical Publishing, 1997, vol. 3, no. 3.

[18] Synth School, Part 2: Resonance, Envelopes & Routing. http://www.soundonsound.com/sos/1997_articles/aug97/synthschool2.html, fecha de consulta 2015-04-14

[19] Galleryhip, Moog Synthesizer-Viewing Gallery. <http://galleryhip.com/moog-synthesizer.html>, fecha de consulta 2015-04-14

[20] Wikibooks, Sound Synthesis Theory/Subtractive Synthesis. http://en.wikibooks.org/wiki/Sound_Synthesis_Theory/Subtractive_Synthesis, fecha de consulta 2015-04-14

[21] Wikibook, Sound Synthesis Theory/Additive Synthesis. http://en.wikibooks.org/wiki/Sound_Synthesis_Theory/Additive_Synthesis, fecha de consulta 2015-04-14

- [22] AIR Music Technology, Hybrid 3. <http://www.airmusictech.com/product/hybrid-3#.VS2bJfnF-FI>, fecha de consulta 2015-04-14
- [23] Roland Corporation, An Introduction to MIDI, 2009. <http://www.midi.org/aboutmidi/intromidi.pdf>
- [24] Rogerlinn, LinnStrument. <http://www.rogerlinndesign.com/linnstrument.html>, fecha de consulta 2015-04-16
- [25] Akai, MAX49 : Akai Professional Keyboard MIDI Controller. <http://www.akaipro.com/product/max49>, fecha de consulta 2015-04-16
- [26] Wikispaces, MIDI-Protocol - 03-MIDI Message. <http://midi-protocol.wikispaces.com/03-MIDI+Message>, fecha de consulta 2015-04-16
- [27] Wolfe, J., Note names, MIDI numbers and frequencies. <http://newt.phys.unsw.edu.au/jw/notes.html>, fecha de consulta 2015-02-15
- [28] Akai, MPC Software : Akai Professional MIDI Secuencer. <http://www.akaipro.com/misc/mpc-software/essentials>, fecha de consulta 2015-04-16
- [29] PcMag, Dynamic Programming Language. <http://www.pcmag.com/encyclopedia/term/57998/dynamic-programming-language>, fecha de consulta 2015-04-16

- [30] Wang, G. y Cook, P., ChuckK : A Programming Language for On-the-fly , Real-time Audio Synthesis and Multimedia, *12th annual ACM international conference on Multimedia*, páginas 812–815, 2004.
- [31] CNMAT, Introduction to OSC. <http://opensoundcontrol.org/introduction-osc>, fecha de consulta 2015-04-16
- [32] Kopetz, H., Real-Time Systems: Design Principles for Distributed Embedded Applications, 1997.
- [33] Herlihy, M. y Shavit, N., The Art of Multiprocessor Programming. Burlington, USA: Elsevier Inc., 2008.
- [34] Williams, A., C++ Concurrency in Action. Manning Publications Co., 2012.
- [35] Wiggins, G., Pearce, M., y Müllensiefen, D., Computational modeling of music cognition and musical creativity, 2009.
- [36] Temperley, D., The Cognition of Basic Musical Structures. The MIT Press, 2005, vol. 23.
- [37] Assayag, G. y Dubnov, S., Using factor oracles for machine improvisation, *Soft Computing*, vol. 8, páginas 604–610, 2004.
- [38] Bezirganyan, A. H., Analysis and Estimation of Emotionally Colouring of

- Music Performance, *Proceedings of the 8th International Conference on Music Perception and Cognition*, páginas 710–712, 2004.
- [39] Scherer, K. R. y Zentner, M. R., Emotional effects of music: Production rules. *Music and emotion: Theory and research*, páginas 361–392, 2001.
- [40] Juslin, P. N., Karlsson, J., Lindström, E., Friberg, A., y Schoonderwaldt, E., Play it again with feeling: computer feedback in musical communication of emotions. *Journal of experimental psychology. Applied*, vol. 12, no. 2, páginas 79–95, 2006.
- [41] Wieczorkowska, A., Synak, P., Lewis, R., y Ras, Z., Extracting emotions from music data, *Foundations of Intelligent Systems*, páginas 456–465, 2005.
- [42] Riecken, D. y Hill, M., WOLFGANG : Emotions and Architecture which Enable Learning to Compose Music, *Proceedings of the International Conference of the Society for Adaptive Behavior*, 1998.
- [43] Hevner, K., Experimental Studies of the Elements of Expression in Music, *The American Journal of Psychology*, vol. 48, no. 2, páginas 246–268, 1936.
- [44] Le Groux, S. y Verschure, P. F., Emotional Responses to the Perceptual Di-

- mensions of Timbre: A Pilot Study Using Physically Informed Sound Synthesis, *In Proceedings of the 7th International Symposium on Computer Music Modeling*, páginas 1–15, 2010.
- [45] Vickhoff, B., *A Perspective Theory of Music Perception and Emotion*, 1st ed. Gothenburg: Intellecta DocuSys, Västra Frölunda, 2008, no. 90.
- [46] Luger, G. F., *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, 2005, vol. 5th.
- [47] Váczy, T., The Creation of the Musical Scale. <http://vaczy.dk/htm/scales2.htm>, fecha de consulta 2015-02-14
- [48] Ulrich, J., The analysis and synthesis of jazz by computer, *Fifth International Joint Conference on Artificial Intelligence*, páginas 865–872, 1977.
- [49] Ganesh Ram, S., Palaniappan, C. T., Ramakrishnan, M. S., y Devanathan, R., Knowledge Engineering of Creative Musical Expressions using Carnatic Music Ideology, *International Florida Artificial Intelligence Research Society Conference*, 2004.
- [50] Kendal, S. y Creen, M., *An Introduction to Knowledge Engineering*. Springer-Verlag New York, 2007.

- [51] Suiter, W., Toward Algorithmic Composition of Expression in Music Using Fuzzy Logic, *Proceedings of the 2010 International Conference on New Interfaces for Musical Expression*, páginas 319–322, 2010.
- [52] Moorer, J., Music and computer composition, *Communications of the ACM*, vol. 15, no. 2, páginas 104–113, febrero 1972.
- [53] Manousakis, S., Musical L-Systems, Disertación para Ph.D. , 2006.
- [54] NIST, Dictionary of Algorithms and Data Structures: Arrays. <http://xlinux.nist.gov/dads//HTML/array.html>, fecha de consulta 2015-04-04
- [55] Shaffer, C., Data structures and algorithm analysis in C++. Dover Publications, 2011, vol. 3.
- [56] Gamma, E., Helm, R., Johnson, R., y Vlissides, J., Design Patterns: Elements of Reusable Object-Oriented Software, 2007.
- [57] CppCon, CppCon 2014: Mike Acton - Data-Oriented Design and C++. <https://www.youtube.com/watch?v=rX0ltVEVjHc>, fecha de consulta 2015-04-04
- [58] Dannenberg, R. B., An On-Line Algorithm for Real-Time Accompaniment, *Proceedings of the 1984 International Computer Music Conference*, pági-

nas 193–198, 1985.

- [59] Russell, S. y Norvig, P., *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, New Jersey: Pearson Education/Prentice-Hall, 2009.
- [60] Knapp, R. B. y Cook, P. R., *The Integral Music Controller : Introducing a Direct Emotional Interface To Gestural Control of Sound Synthesis*, *Proceedings of the International Computer Music Conference ICMC*, páginas 4–9, 2005.
- [61] BHRE, Audio Interfaces. <http://www.basic-home-recording-studio.com/audio-interfaces.html>, fecha de consulta 2015-04-18
- [62] IEEE, IEEE Standard for Software Test Documentation, *IEEE Std 829-1998*, página i, 1998.
- [63] McGill, The RtMidi Tutorial. <http://www.music.mcgill.ca/~gary/rtmidi/index.html#intro>, fecha de consulta 2015-04-24
- [64] MidiFile, C++ library for parsing Standard MIDI Files. <http://midifile.sapp.org/>, fecha de consulta 2015-04-24
- [65] CNMAT, Oscpack. <http://opensoundcontrol.org/oscpack>, fecha de consulta 2015-04-24

[66] MOTU, MIDI Express 128 - MIDI Interface. <http://www.motu.com/products/>

[midi/128](#), fecha de consulta 2015-06-29

[67] Schell, J., The Art of Game Design: A Book of Lenses, ser. Morgan Kaufmann. Morgan Kaufmann, 2008, vol. 54.