



Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

Título:

“Automatización de contrapunto musical utilizando programación con restricciones”

Tesis presentada por:

Carlos Augusto Reyes Hernández

Como requisito para obtener el título de Maestría en Ciencias de la Computación

Asesores:

Dr. José Alejandro Rangel Huerta

Dr. Guillermo De Ita Luna

Puebla Pue., Noviembre de 2022

Índice

1	Resumen	3
2	Introducción.....	4
3	Objetivo General.....	6
3.1	Objetivos específicos.....	6
4.	Resumen de teoría musical	7
4.1	Definiciones en la música.....	8
4.2	Parámetros del sonido	8
4.3	El tiempo musical.....	10
4.4	Las notas musicales.....	10
4.5	La altura musical.....	11
4.6	Equivalencia entre las figuras musicales	12
4.7	El símbolo de ligadura de unión	14
4.8	La partitura	15
4.9	El intervalo	16
5	El contrapunto.....	19
5.1	Especies de contrapunto	19
5.2	Las reglas de Contrapunto.....	21
6	La música como un problema combinatorio	22
6.1	¿Cuántas combinaciones de notas musicales hay en un compás?.....	23
6.2	¿Cuántas soluciones permiten las reglas de contrapunto?	25
7	Metodología.....	28
8	Codificación de la partitura	30
8.1	Codificación en letras y números	30
8.2	Prolog y la codificación de partituras musicales	32
8.3	Implementación de las reglas en Prolog.....	35
9	Unión entre Python y Prolog	40
9.1	La librería Pyswip	41
9.2	La librería Music21	42
10	Conclusión	45
11	Bibliografía.....	47

1 Resumen

Esta tesis tiene como objetivo ser la presentación del trabajo desarrollado para obtener el grado de Maestría en Ciencias de la Computación dentro de la Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla.

El proyecto consiste en diseñar un sistema que automatice la composición musical basada en contrapunto.

Para lograr esto se plantea la técnica de contrapunto como un problema de satisfacción de restricciones.

Dada una melodía, (Cantus Firmus) el sistema aplicará las reglas de contrapunto musical para generar las posibles armonías correspondientes a cada nota.

Para la implementación de las reglas se usará programación declarativa.

Se realiza el cálculo de cuantas posibles combinaciones se pueden crear en un compás musical y cuantas posibles soluciones existen aplicando las reglas de contrapunto.

Se utilizará un conjunto de 9 reglas obtenidas del trabajo de Giovanni Pierluigi de Palestrina y Joseph Fux, en la época Renacentista, asociadas a ese estilo de composición.

Además, se muestra un modelo para representar partituras en una lista de letras y números, y otro modelo para representar una partitura en una lista de números.

2 Introducción

¿Cuál es la limitante de la inteligencia artificial (I.A.)?, es decir, ¿existe un área del conocimiento humano que la máquina no pueda lograr?.

Los actuales avances en inteligencia artificial dejan un espacio muy pequeño donde el ser humano pueda destacar en contra de la computadora. Un ejemplo de esto es Alphacode, un sistema creado por la empresa DeepMind, que genera soluciones a problemas de programación competitiva, lo que de cierta manera, podría significar que la I.A. puede reemplazar incluso a los creadores de la propia I.A.

Este hecho es interesante, ya que este avance podría significar que las áreas de conocimiento humano, en las cuales se busca simplificar el resultado a dos posibles respuestas, (correcto o incorrecto) están al alcance de los avances computacionales.

Suponiendo que las ciencias están cubiertas en mayor o menor medida por los avances computacionales, ¿existen otras áreas de conocimiento disponibles?

El arte en general, y en específico la música, más que un problema lógico, está catalogado como un medio de comunicación, en donde un emisor (el artista), crea un mensaje (obra artística) y por medio de un canal (pintura, escultura, música, cine, etc.) hace llegar este mensaje a un receptor (espectador) [1].

Pero cada artista es distinto, cada obra es distinta y la codificación y decodificación del mensaje es igual o más amplia que la misma capacidad del emisor y del receptor para codificarlo y decodificarlo. Y más importante aún, cada proceso de codificado depende de cada individuo distinto. El generalizar en un si o un no el arte sería generalizar cada una de las posibles opiniones que las personas puedan tener de cada una de las distintas obras.

Si bien no es posible crear un modelo general del arte o de la música, si se pueden crear modelos limitando el alcance de estos, específicamente en la música existen trabajos previos y posteriores a la invención de la computadora moderna, que intentan formalizar la creatividad humana.

Un ejemplo de esto es el Musikalisches Würfelspiel, inventado en Berlín en 1792 [2], el cual es un sistema que con el uso de dados selecciona de manera aleatoria breves pasajes de música, que son puestos uno tras otro para crear una pieza musical, sin necesidad de tener conocimientos musicales.

En 1959 Hiller e Issacson fueron probablemente los primeros que utilizaron un modelo computacional utilizando generadores de números aleatorios y cadenas de Markov para la composición algorítmica.

Luego, en 1999, Papadopoulos y Wiggins acuñan la definición de la composición algorítmica, que es un área de investigación centrada en la composición musical automática.

La composición musical automática se ha abordado usando varias técnicas, como son: la lógica difusa , sistemas expertos y lógica probabilística [3].

La composición automática de contrapunto es una dirección popular de investigación en composición algorítmica, debido a las reglas estrictamente definidas que deben obedecerse. Esto facilita las implementaciones de sistemas de composición automática.

Más actualmente, podemos encontrar trabajos realizado en 2014 sobre la automatización de contrapunto de primera especie.

El contrapunto de primera especie solo evalúa la relación entre dos notas con la misma figura musical, el problema se vuelve más complicado cuando se agregan todas las figuras musicales [4].

La intención de este trabajo es crear un sistema que evalúe la relación de las 5 especies de contrapunto, es decir, cualquier relación entre notas y utilizando cualquier figura musical.

Para abordar este problema, primero se presentará la composición musical como un problema combinatorio. Se mostrará el espacio de soluciones posible después de aplicar las restricciones de la técnica de contrapunto.

Seguido de esto, se hará un resumen de la teoría musical necesaria para entender los conceptos relacionados con la técnica de contrapunto.

Después se implementará en un lenguaje declarativo las reglas de contrapunto.

Para poder realizar la evaluación, se creará una codificación de la partitura que permita representar todas las figuras musicales, primero utilizando letras y números, y luego solo una lista de números.

Por último se creará una interfaz visual y auditiva que sirva como herramienta al usuario al momento de componer en base a esta técnica de contrapunto.

3 Objetivo General

- Construir un prototipo de software que logre la automatización de la evaluación de las reglas utilizadas en la técnica de composición por contrapunto.
- Diseñar e implementar una interfaz que permita al usuario interactuar con el programa de manera visual y auditiva.

3.1 Objetivos específicos

- Definir los límites y criterios implicados en el proceso de evaluación de la técnica de contrapunto.
- Investigar y comprender el uso del software SWI-Prolog para resolver un problema utilizando el paradigma de programación de restricciones.
- Diseñar e implementar las restricciones del modelo.
- Analizar resultados obtenidos.
- Diseñar e implementar una interfaz visual y auditiva para la interacción con el programa.

4. Resumen de teoría musical

La música se define como el arte de crear y organizar sonidos y silencios respetando los principios fundamentales de la melodía, la armonía y el ritmo, mediante la intervención de complejos procesos psicoanímicos. Desde hace varias décadas se ha vuelto más compleja la definición de qué es y qué no es la música, ya que destacados compositores en el marco de diversas experiencias artísticas fronterizas han realizado obras que, si bien podrían considerarse musicales, expanden los límites de la definición de este Arte [5].

La música, como toda manifestación artística, es un producto cultural con múltiples finalidades, entre otras, la de suscitar una experiencia estética en el oyente, la de expresar sentimientos, emociones, circunstancias, pensamientos o ideas, y cada vez más, cumplir una importante función terapéutica a través de la musicoterapia.

La música cumple una función de vital importancia en el desarrollo cognitivo del ser humano. Está relacionada con el pensamiento lógico matemático, la adquisición del lenguaje, el desarrollo psicomotriz, las relaciones interpersonales, el aprendizaje de lenguas no nativas y a potenciar la inteligencia emocional, entre otros. Por este motivo, la música debe estar presente en cualquier plan educativo moderno y ser reconocida como una disciplina imprescindible dentro de la enseñanza obligatoria [6].

La música es un estímulo sonoro que afecta al campo perceptivo del individuo; así, el flujo sonoro puede cumplir variadas funciones (entretenimiento, comunicación, ambientación, diversión, etc.). En muchas culturas, la música es una parte importante del modo de vida de la gente, ya que desempeña un papel fundamental en rituales religiosos, en las ceremonias de rito de paso (por ejemplo, la graduación y el matrimonio), en las actividades sociales (por ejemplo, en el baile) y en las actividades culturales que van desde el canto aficionado en el karaoke hasta tocar en una banda amateur de funk o cantar en un coro comunitario.

La gente puede hacer música por afición, como por ejemplo, un adolescente que toca el violonchelo en una orquesta juvenil, o trabajar como músico o cantante profesional. La industria musical incluye a las personas que crean nuevas canciones y piezas musicales (como los cantautores y los compositores), a las personas que interpretan música (que incluyen a los músicos de orquesta, de bandas de jazz y de bandas de rock, a los cantantes y a los directores de orquesta), a las personas que graban música (productores musicales e ingenieros de sonido), a las personas que organizan giras de conciertos y a las personas que venden grabaciones y partituras a los clientes. Incluso una vez que se ha interpretado una canción o pieza, la crítica musical, el periodismo musical y la musicología pueden valorar y evaluar la pieza y su interpretación.

4.1 Definiciones en la música

Las definiciones sobre la música parten desde el seno de las culturas, y así, el sentido de las expresiones musicales se ve afectadas por cuestiones psicológicas, sociales, culturales e históricas. De esta forma, surgen múltiples y diversas definiciones que pueden ser válidas en el momento de expresar qué se entiende por música. Es importante aclarar que ninguna, en el sentido extenso de la palabra puede considerarse como una definición perfecta o absoluta.

Una definición bastante amplia determina que la música es sonoridad organizada, donde es posible encontrar patrones perceptibles por entes consientes. Esta definición parte de que en aquello a lo que consensualmente se puede denominar música, se le pueden percibir ciertos patrones del flujo sonoro en función de cómo las propiedades del sonido son aprendidas y procesadas por los humanos, e incluso, hay quienes consideran que también por los animales [2].

Hoy en día es frecuente trabajar con un concepto de música basado en tres atributos esenciales: que utiliza sonidos, que es un producto humano, tanto natural como “artificial”, y que predomina la función estética. Si tomamos en cuenta solo los dos primeros elementos de la definición, nada diferencia a la música del lenguaje.

4.2 Parámetros del sonido

El sonido es la sensación percibida por el oído al recibir las variaciones de presión generadas por el movimiento vibratorio de los cuerpos sonoros. Se transmite por el medio que los envuelve, que generalmente es el aire de la atmósfera. La ausencia perceptible de sonido es el silencio, aunque es una sensación relativa, ya que el silencio absoluto no se da en la naturaleza.

El sonido tiene cuatro parámetros fundamentales:

1) La altura es el resultado de la frecuencia que produce un cuerpo sonoro; es decir, de la cantidad de ciclos de las vibraciones por segundo de hercios (Hz) que se emiten. De acuerdo con esto se pueden definir los sonidos como graves y agudos. Cuanto mayor sea la frecuencia, más agudo (o alto) será el sonido. La longitud de onda es la distancia medida en la dirección de propagación de la onda, entre dos puntos cuyo estado de movimiento es idéntico; es decir, que alcanzan sus máximos y mínimos en el mismo instante.

2) La duración corresponde al tiempo que duran las vibraciones que producen un sonido. La duración del sonido está relacionada con el ritmo. La duración viene representada en la onda por los segundos que esta contenga.

3) La intensidad es la fuerza con la que se produce un sonido; depende de la energía. La intensidad viene representada en una onda por la amplitud.

4) El timbre es la cualidad que permite distinguir los diferentes instrumentos o voces a pesar de que estén produciendo sonidos con la misma altura, duración e intensidad. El timbre depende de la cantidad de armónicos o de la forma de la onda que tenga un sonido y de la intensidad de cada uno de ellos, a lo cual se le denomina espectro. El timbre se representa en una onda por el dibujo. Un sonido puro, como la frecuencia fundamental, o cada sobretono, se representa con una onda sinusoidal, mientras que un sonido complejo es la suma de ondas senoidales puras.

Los sonidos que escuchamos son complejos; es decir, son el resultado de un conjunto de sonidos simultáneos (tonos, sobre tonos y armónicos), pero que nosotros percibimos como uno (sonido fundamental). El espectro es una sucesión de barras verticales repartidas a lo largo de un eje de frecuencia y que representan a cada una de las senoides correspondientes a cada sobretono, y su altura indica la cantidad que aporta cada una al sonido resultante.

4.3 El tiempo musical

El tiempo musical se define como la duración de un sonido o un silencio, y se representa por símbolos llamados “figuras musicales”. Cada figura representa un tiempo en específico, como se muestra a continuación:











Nombre	Figura	Duración	Silencio
Redonda		4 pulsos	
Blanca		2 pulsos	
Negra		1 pulso	
Corchea		$\frac{1}{2}$ (medio) pulso	
Semicorchea		$\frac{1}{4}$ (cuarto) pulso	

Figura 1: Nombre de figuras música , símbolos y tiempos

4.4 La notas musicales

Las notas musicales son Do, Do#/Reb, Re, Re#/Mib, Mi, Mi#/Fab, Fa, Sol, Sol#/Lab, La, La#/Sib y Si.

Otra forma de representar las notas es por letras, las cuales son C, C#/Db, D, D#/Eb, E, E#/Fb, F, G, G#/Ab, A, A#/Bb, B.

Las notas están relacionadas con la frecuencia de un sonido, por ejemplo, el Do central, el cual se usa para afinar instrumentos, tiene una frecuencia de 261. 6 Hertz [7].

Como un dato adicional, que una misma nota suene distinta entre un instrumento y otro, por ejemplo un Do tocado en un piano y un Do tocado en una flauta, no es debido a la frecuencia, sino el timbre de cada instrumento, esta característica depende de la cantidad de armónicos que tenga un sonido y las distintas intensidades de estos armónicos.

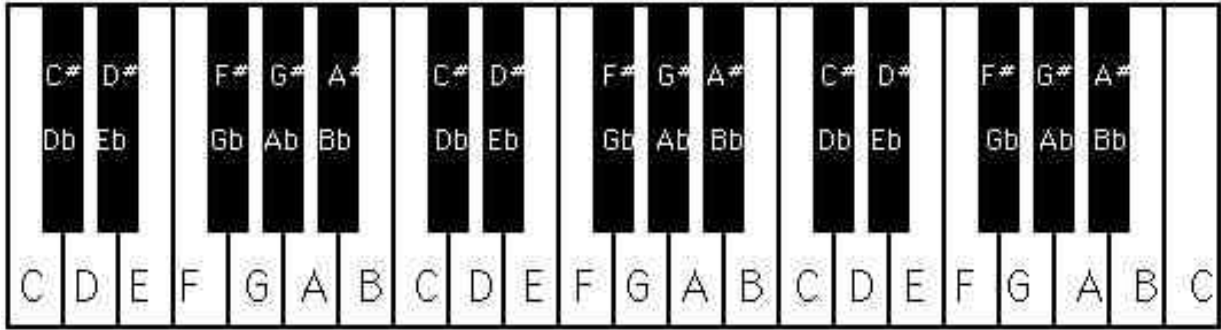


Figura 2: las diferentes notas musicales y sus respectivos sostenidos y be moles en un teclado de un piano.

4.5 La altura musical

La altura indica que tan grave o agudo es un sonido. En las partituras esto es representado tomando 12 diferentes notas y separándolas en octavas. Así, la diferencia entre un C4 y un C6, será que C6 está 2 octavas más arriba con respecto de C4.

La partitura utiliza claves para separar las octavas. Las claves más utilizadas son la de Sol, Fa y Do, respectivamente.

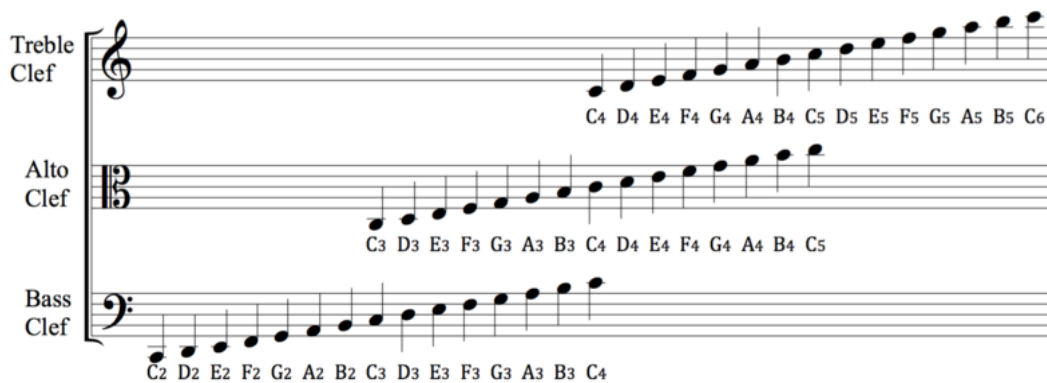


Figura 3: Las claves de Sol, Do y Fa y las respectivas notas con su octava

En la siguiente tabla se muestra el espectro de sonido en frecuencias y como se divide este en notas musicales y octavas.

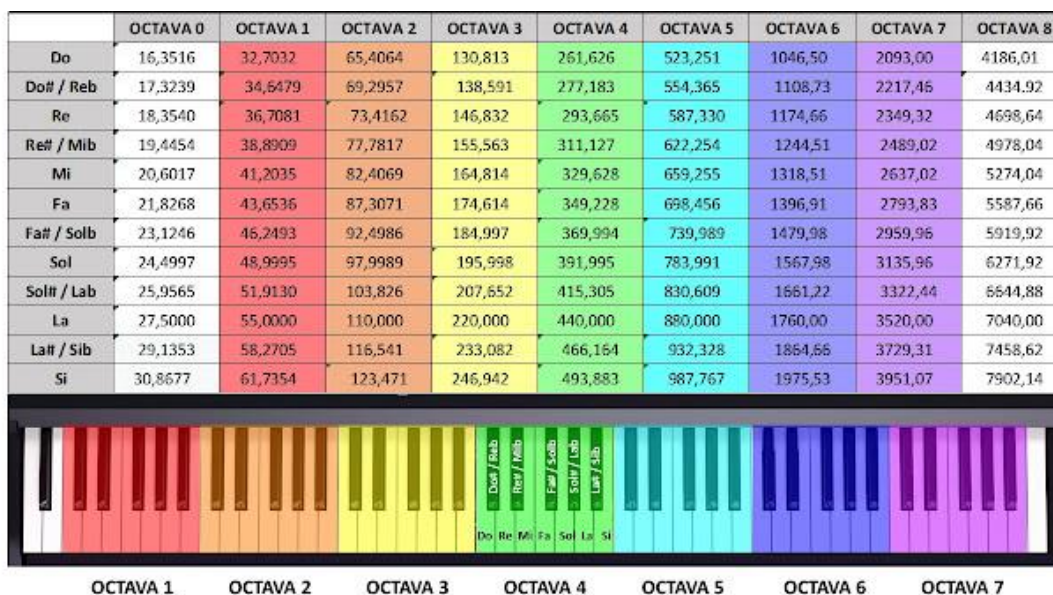


Figura 4: Espectro de sonido, su separación por notas y octavas. Y su respectiva posición en las teclas de un piano.

4.6 Equivalencia entre las figuras musicales

En la escritura musical se manejan equivalencias entre los símbolos, esto se hace para que la lectura sea más sencilla para el intérprete. En específico para la codificación tomaremos en cuenta dos equivalencias, la equivalencia entre las figuras musicales y el uso de la ligadura.

Las figuras musicales representan tiempos. Para referirse a duraciones menores a un tiempo determinado, es posible usar fracciones.

Por ejemplo, si una redonda dura 4 tiempos y una negra dura 1 tiempo, entonces podemos decir que 4 negras equivalen a 1 redonda, o que una negra es igual a $\frac{1}{4}$ de redonda.

Otro ejemplo: una negra tiene el valor de 1 tiempo y una corchea tiene el valor de $\frac{1}{2}$ tiempo. Esto significa que dos corcheas equivalen a 1 negra.

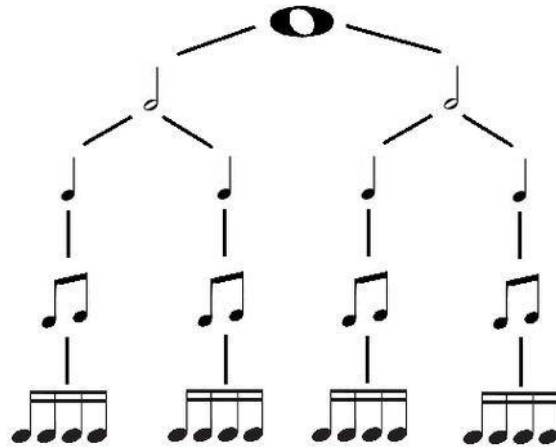


Figura 5: Equivalencias de figuras musicales

Es importante aclarar que, si bien por ejemplo, una redonda y dos blancas tendrán el mismo tiempo de duración, la percepción auditiva es distinta. Es decir, una redonda sonará como un sonido continuo de una determinada duración y dos blancas sonarán como 2 sonidos cada uno con la mitad de duración que la redonda.

Para poder representar una redonda con dos blancas y que el sonido sea idéntico se usa el símbolo de ligadura de unión.

4.7 El símbolo de ligadura de unión

La ligadura de unión conecta dos notas y las convierte en una sola nota que tendrá un valor igual a la suma de las dos notas unidas. El símbolo de ligadura se ocupa cuando las notas conectadas no pueden escribirse con una de mayor valor.

Esta característica permite representar una nota musical como una lista de números repetidos

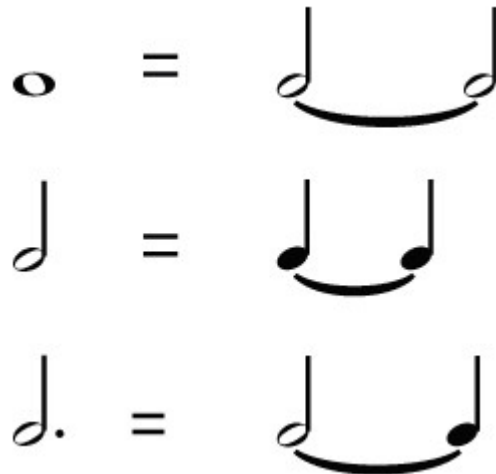


Figura 6: Equivalencias de la ligadura de unión

4.8 La partitura

Una partitura es un documento manuscrito o impreso que representa e indica cómo debe interpretarse una composición musical, mediante un lenguaje propio formado por signos musicales y el llamado sistema de notación. Como sus análogos los libros, los folletos, etc., el medio de la partitura generalmente es el papel o, en épocas anteriores, el pergamino [7]. Aunque el acceso a la notación musical en los últimos años incluye también la presentación en tabletas, teléfonos inteligentes y computadoras.

En música orquestal, se denomina partitura al documento que utiliza exclusivamente el director de orquesta y que contiene la obra que se ejecutará con todas sus indicaciones. Esta se desglosa en partes: nombre dado a cada uno de esos desgloses o particiones instrumentales que tienen los intérpretes de los diferentes instrumentos, las cuales pueden incluir cada una dos o más instrumentos similares, como un piano y una flauta, un oboe y un corno inglés, etc., y ser ejecutadas por un mismo intérprete [8].

La partitura se puede utilizar como un registro, una guía o un medio para interpretar una pieza de música. Aunque no sustituye al sonido de la ejecución musical, la partitura se puede estudiar para construir la interpretación y para dilucidar los aspectos de la música que pueden no ser evidentes a partir de la simple audición. Se puede obtener información fidedigna sobre una pieza musical mediante el estudio de los bocetos y las primeras versiones escritas de las obras que el compositor pudo haber conservado, así como la partitura final autógrafa y las anotaciones personales hechas en borradores y partituras impresas.

La comprensión de las partituras requiere una forma especial de alfabetización, la capacidad de leer notación musical. No obstante, la capacidad de leer o escribir música no es un requisito para componer música. Muchos compositores han sido capaces de crear música en formato impreso, sin la habilidad de leer o escribir en notación musical, siempre y cuando un copista de algún tipo estuviese disponible. Por ejemplo, el compositor ciego del siglo xviii d. C. John Stanley así como los compositores y letristas del siglo xx d. C. Lionel Bart, Irving Berlin y Paul McCartney [10]

Una partitura es un documento manuscrito o impreso que representa e indica cómo debe interpretarse una composición musical, mediante un lenguaje propio formado por signos musicales y el llamado sistema de notación musical.



Figura 7: Ejemplo de una partitura

4.9 El intervalo

El intervalo es la distancia (en términos de altura) entre dos notas musicales o entre dos sonidos. En general, suelen medirse según las notas de una escala incluyendo los extremos (por ejemplo, entre RE y FA, si se cuenta de manera ascendente, hay una tercera), o bien por la diferencia de tonos y semitonos, o solamente de semitonos. Por ejemplo, una quinta justa es un intervalo de 7 semitonos que se miden en un teclado.

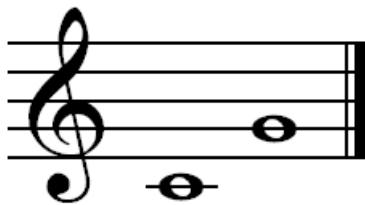


Figura 8. Intervalo melódico de quinta justa ascendente

La figura anterior, recibe el nombre de quinta porque hay una distancia de cinco grados entre las notas que lo forman (do y sol).

Recibe el apelativo de justa porque hay una distancia de tres tonos y un semitono entre los sonidos que lo forman.

Los intervalos tonales (a veces llamados consonancias perfectas) pueden ser justos; los modales (a veces llamados consonancias imperfectas y disonancias) pueden ser mayores o menores. Además todos los tipos de intervalos pueden ser aumentados o disminuidos.

Se denomina armónico al intervalo cuyos sonidos suenan simultáneamente y melódico a aquel cuyos sonidos suenan sucesivamente.



Figura 9. Intervalos simples

La teoría musical considera tonales los intervalos de primera, unísono, cuarta, quinta y octava y modales los dos intervalos tonales (a veces llamados consonancias perfectas) pueden ser justos; los modales (a veces llamados consonancias imperfectas y disonancias) pueden ser mayores o menores. Además todos los tipos de intervalos pueden ser aumentados o disminuidos [8]

Se denomina armónico al intervalo cuyos sonidos suenan simultáneamente y melódico a aquel cuyos sonidos suenan sucesivamente.

Los intervalos melódicos pueden ser, además, ascendentes o descendentes.

Se consideran simples los intervalos no mayores que una octava y compuestos a los que la exceden. Los intervalos compuestos son análogos a los intervalos simples correspondientes. Así, una novena es una segunda a la octava y puede ser mayor o menor; una duodécima es análoga a una quinta y puede ser justa, aumentada o disminuida.

Se llaman complementarios los intervalos que, sumados, conforman una octava: una cuarta y una quinta son complementarias. Nótese que la suma de los cuatro grados de la cuarta y los cinco grados de la quinta se resuelve en ocho grados, no nueve, porque el cuarto grado de la cuarta es a la vez el primer grado de la quinta, segunda, tercera, sexta y séptima [10]

Dependiendo de la distancia entre semitonos, cada intervalo lleva un nombre en particular. Los nombres de cada intervalo se muestran a continuación:

Nombre del intervalo	Distancia en semitonos	Valor codificado
Unísono	0 semitonos	0
Segunda menor	1 semitono	1
Segunda mayor	2 semitonos	2
Tercera menor	3 semitonos	3
Tercera mayor	4 semitonos	4
Cuarta justa	5 semitonos	5
Cuarta aumentada	6 semitonos	6
Quinta justa	7 semitonos	7
Sexta menor	8 semitonos	8
Sexta mayor	9 semitonos	9
Séptima menor	10 semitonos	10
Séptima mayor	11 semitonos	11
Octava justa	12 semitonos	12

Figura 10: Intervalos musicales y su respectiva codificación

5 El contrapunto

El contrapunto (del latín *punctus contra punctum*, «nota contra nota») es una técnica de improvisación y composición musical que evalúa la relación existente entre dos o más voces independientes con la finalidad de obtener cierto equilibrio armónico. Casi la totalidad de la música compuesta en Occidente es resultado de algún proceso contrapuntístico. Esta práctica, que surgió en el siglo XIV, alcanzó un alto grado de desarrollo en el Renacimiento y el periodo de la práctica común, especialmente en la música del Barroco, y se ha mantenido hasta nuestros días.

En su aspecto más general, el contrapunto implica la creación de líneas musicales que se mueven independientemente unas de otras pero guardan una relación armoniosa entre ellas. En cada época, la escritura de música organizada contrapuntísticamente ha estado sujeta a reglas, en ocasiones estrictas. Por definición, los acordes se producen cuando diversas notas suenan al mismo tiempo. Sin embargo, los rasgos armónicos verticales se consideran secundarios e incidentales cuando el contrapunto es el elemento textural predominante. El contrapunto se centra en la interacción melódica y solo en segundo lugar en las armonías producidas por esa interacción.

“Es difícil escribir una canción hermosa. Es más difícil escribir varias canciones hermosas de forma individual que, cuando se cantan al mismo tiempo, suenan en su conjunto polifónico aún más bello. Las estructuras internas que crea cada una de las voces por separado deben contribuir a la estructura emergente de la polifonía, que a su vez debe reforzar y desarrollar las estructuras de las voces individuales. La forma en que se lleva a cabo en detalle es... “el contrapunto” [11]

5.1 Especies de contrapunto

En general, las especies de contrapunto ofrecen menos libertad para el compositor que otros tipos de contrapunto, por lo que es conocido como contrapunto riguroso o estricto. Las especies de contrapunto se han desarrollado como una herramienta pedagógica, en la que un estudiante avanza a través de varias «especies» de complejidad creciente, con una parte muy simple que no cambia, conocido como el *cantus firmus* (en latín significa «melodía fija»). El estudiante va adquiriendo gradualmente la capacidad de escribir contrapunto libre, que es el contrapunto menos rigurosamente limitado, por lo general sin *cantus firmus*, conforme a las reglas del momento.

El pedagogo más famoso que utilizó el término y que lo hizo famoso fue Johann Joseph Fux. En 1725 publicó *Gradus ad Parnassum* (Pasos al Parnaso), una obra para ayudar en la enseñanza de la composición mediante contrapunto a los estudiantes. En concreto, se centraba en el estilo contrapuntístico practicado por Palestrina a finales del siglo XVI como la principal técnica. Como base de su simplificada y a menudo excesivamente restrictiva codificación de la práctica de Palestrina (ver «Notas generales» abajo), Fux describió cinco especies:

1. Nota contra nota.
2. Dos notas contra una.
3. Cuatro (extendido por otros para incluir tres o seis, etc.) notas contra una.
4. Notas descolocadas contrapuestas entre sí (como suspensiones).
5. Todas las primeras cuatro especies juntas, como contrapunto florido.



Figura 11. Ejemplo de contrapunto musical, dos melodías con notas relacionadas paralelamente

5.2 Las reglas de Contrapunto

Para este trabajo, se seleccionaron 9 reglas de contrapunto que serán implementadas usando el lenguaje de programación Prolog. Además, se creó un sistema dirigido al usuario para evaluar el sonido y la composición de las reglas de contrapunto. Las nueve reglas, descritas en lenguaje natural, son las siguientes:

Regla 1: Comenzar y terminar en unísono, octava o quinta, a menos que la voz añadida sea inferior, en cuyo caso se comienza y termina solamente en unísono u octava.

Regla 2: No utilizar unísono excepto al principio o al final.

Regla 3: Evitar las quintas u octavas paralelas entre dos voces cualesquiera y evitar las quintas u octavas paralelas «ocultas». Es decir, solo se harán movimientos contrarios hacia una quinta u octava perfectas, a menos que una parte se mueva por grados conjuntos.

Regla 4: No moverse sobre cuartas paralelas.

Regla 5: No moverse sobre terceras o sextas paralelas durante mucho tiempo.

Regla 6: Tratar de mantener dos voces adyacentes dentro de una distancia de una décima entre ellas.

Regla 7: Evitar que las dos partes se muevan en la misma dirección por el salto.

Regla 8: Tratar de introducir tanto movimiento contrario como sea posible.

Regla 9: Evitar los intervalos disonantes entre dos voces cualquiera: las segundas mayores y menores, las séptimas menores, cualquier intervalo aumentado o disminuido, así como las cuartas perfectas (en muchos contextos).

6 La música como un problema combinatorio

¿Cuánta música existe?, antes de poder encontrar una respuesta a esta pregunta es necesario poner limitantes al problema, ya que en un tiempo infinito, las posibles combinaciones de notas son claramente infinitas.

Esto podría parecer algo obvio, pero al tratarse la música de una expresión artística, muchas veces la lógica es insuficiente.

Un ejemplo de esto es la interpretación de Organ/ASLSP (As SLOW as Possible), creada en 1987 por el compositor estadounidense John Cage, esta obra comenzó a tocarse el 5 de septiembre de 2001 por un órgano construido especialmente para interpretar esta obra, y terminará de interpretarla en el año 2640.



Figura 12: Fotografía del órgano donde se interpreta la obra ‘Organ/ASLSP’ que dura 639 años

Las melodías, y las canciones en general se dividen en compases. Un compás es una entidad métrica musical compuesta por varias unidades de tiempo (figuras musicales) que se organizan en grupos.

En la teoría musical el compás se escribe con 2 números. El primero indica cuantas notas se pueden agregar en un compás y el segundo número indica de que tipo de nota se habla. En la práctica, la división más comúnmente usada es la de 4-4, donde el primer cuatro hace referencia a que habrá 4 figuras musicales (o sus equivalentes) y el segundo cuatro significa que estas cuatro notas deben ser figuras de “negra” o sus equivalentes.

Existen otros compases, como 4-8, que son cuatro corcheas por compás o sus equivalentes o también 2-2 para representar que por cada compás habrá dos blancas o sus equivalentes.



Figura 13: Ejemplos de melodías todas en compás 4-4

6.1 ¿Cuántas combinaciones de notas musicales hay en un compás?

A continuación, se hará el cálculo de cuántas posibles combinaciones de notas se pueden crear en un compás comenzando con un caso particular y aumentando hasta cubrir todas las posibilidades.

Como se mencionó en el capítulo 4, una figura musical, puede representar doce notas distintas, que a su vez se repiten en diferentes octavas.

Podemos ver el conjunto de las diferentes notas y octavas como una incógnita, "x".

Ahora, supongamos que se tiene un compás, donde solo se permite agregar una nota de un tiempo, la figura redonda.

Para este caso, el número de posibles combinaciones diferentes será:

$$x = (12 \text{ notas} * \text{No. de octavas})$$

Para el segundo caso, permitiremos que se agregue la figura musical de $\frac{1}{2}$ de tiempo, la blanca.

Entonces, las posibles combinaciones en el compás serán:

$$\text{dos blancas} = x^2$$

ó

$$\text{una redonda} = x$$

Es decir, $x+x^2$ diferentes posibles combinaciones

El siguiente caso será cuando se permite agregar la figura musical de $\frac{1}{3}$ de tiempo. La negra con puntillo.

Las posibles combinaciones serán:

tres negras con puntillo	ó	= x^3
una blanca y una negra con puntillo	ó	= x^2
una negra con puntillo y una blanca	ó	= x^2
una redonda		= x

es decir $x^3 + 2x^2 + x$ posibles combinaciones.

Resumiendo, lo anterior, permitiendo solamente usar la figura de un tiempo, la redonda, tenemos: x posibles combinaciones

Usando la figura de un tiempo (redonda) y la figura de $\frac{1}{2}$ tiempo, la blanca, tenemos: $(x)(x+1) = x^2 + x$ posibles combinaciones.

Usando la figura de redonda, la de $\frac{1}{2}$ tiempo, la blanca y la de $\frac{1}{3}$ de tiempo, la negra con puntillo. Tenemos $(x)(x+1)^2 = (x^2+x)(x+1) = x^3 + 2x^2 + x$ posibles combinaciones.

Luego, usando la figura de redonda, la de $\frac{1}{2}$ tiempo, la blanca, la de $\frac{1}{3}$ de tiempo, negra con puntillo, y la figura negra que equivale a $\frac{1}{4}$ de tiempo, podemos obtener:
 $(x)(x+1)^3 = (x^3 + 2x^2 + x)(x+1) = x^4 + 3x^3 + 3x^2 + x$ posibles combinaciones.

Hasta este punto, podemos ver, que el agregar una figura musical es como multiplicar el valor anterior de las combinaciones por $(x+1)$, y es necesario preguntar, ¿cuál es la división más pequeña que se puede considerar?. En la teoría musical no hay un límite para subdividir el valor de una figura musical, pero, en la práctica la figura más pequeña generalmente utilizada es la fusa que tiene un tiempo de $\frac{1}{32}$.

Con esto en mente, para un compás en el que se permite el uso de todas las figuras musicales, desde la redonda hasta la fusa, y siguiendo las fórmulas anteriores, llegamos a la conclusión que se pueden obtener:

$$(x)(x+1)^{31} = x^{32} + 31x^{31} + 465x^{30} + 4495x^{29} + 31465x^{28} + 169911x^{27} + 736281x^{26} + 2629575x^{25} + 7888725x^{24} + 20160075x^{23} + 44352165x^{22} + 84672315x^{21} + 141120525x^{20} + 206253075x^{19} + 265182525x^{18} + 300540195x^{17} + 300540195x^{16} + 265182525x^{15} + 206253075x^{14} + 141120525x^{13} + 84672315x^{12} + 44352165x^{11} + 20160075x^{10} + 7888725x^9 + 2629575x^8 + 736281x^7 + 169911x^6 + 31465x^5 + 4495x^4 + 465x^3 + 31x^2 + x$$

posibles combinaciones distintas.

Si consideramos únicamente una octava y las 12 notas musicales, el valor de la fórmula anterior es:

$(12)(12+1)^{31} = 4.0871932e+35$ posibles melodías distintas en un compás.

Para poner en contexto este número, se calcula que hay $56e+18$ granos de arena en el planeta, es decir la mitad de las melodías distintas mencionadas anteriormente, es decir, un número muy grande.

6.2 ¿Cuántas soluciones permiten las reglas de contrapunto?

El cálculo anterior es sin aplicar ningún tipo de restricción a que notas o a que intervalo entre notas utilizar. La pregunta siguiente por responder es: ¿Las reglas de contrapunto reducen considerablemente las posibles combinaciones?

Analicemos el conjunto de las 9 reglas mencionadas en el capítulo 8.

Regla 1:

Comenzar y terminar en unísono, octava o quinta, a menos que la voz añadida sea inferior, en cuyo caso se comienza y termina solamente en unísono u octava.

Dis_1, Dis_n pertenece a $\{[CF_{1,n}, CF_{1,n}+12, CF_{1,n}+CF_{1,n}+7] \text{ or } [CF_{1,n}, CF_{1,n}+12]\}$

Es decir, que para la primera nota y la última, solo se podrá elegir a lo más entre un conjunto de 3 notas distintas.

$Dis_1 = 3, Dis_2 = x_2, Dis_3 = x_3, \dots, Dis_n = 3$

Regla 2:

No utilizar unísono excepto al principio o al finalidad

$Dis_2, Dis_3, \dots, Dis_{n-1}$ No pertenece a $\{CF_2, CF_3, \dots, CF_{n-1}\}$

$Dis_2 = x-1, Dis_3 = x-1 \dots, Dis_{n-1} = x-1$

Regla 3, 4, 5 :

No moverse en quintas, octavas, cuartas, terceras o sextas paralelas.

$Dis_2, Dis_3, \dots, Dis_{n-1}$ No pertenece a $\{ [CF_2+7, CF_2+12, CF_2+5, CF_2+3, CF_2+8], [CF_3+7, CF_3+12, CF_3+5, CF_3+3, CF_3+8], \dots \}$

$$[CF_{n-1} + 7, CF_{n-1} + 12, CF_{n-1} + 5, CF_{n-1} + 3, CF_{n-1} + 8]$$

$$Dis_2 = x-6, Dis_3 = x-6 \dots Dis_{n-1} = x-6$$

La regla 6:

Mantener dos voces adyacentes dentro de una distancia de décima entre ellas.

$$Dis_2, Dis_3, \dots Dis_{n-1} \text{ pertenece a } \left\{ \begin{array}{l} [CF_2 - 15 < CF_2 < CF_2 + 15], \\ [CF_3 - 15 < CF_3 < CF_3 + 15], \\ \vdots \\ [CF_{n-1} - 15 < CF_{n-1} < CF_{n-1} + 15] \end{array} \right\}$$

$$Dis_2 = (31-6)n, Dis_3 = (31-6)n \dots Dis_{n-1} = (31-6)n$$

$$Dis_2 = 25n, Dis_3 = 25n \dots Dis_{n-1} = 25n$$

Regla 7 y 8:

Evitar que las partes se muevan en la misma dirección y agregar tanto movimiento contrario como sea posible.

$$Dis_2, Dis_3, \dots Dis_{n-1} \text{ pertenece a } \left\{ \begin{array}{l} [CF_2 - 12 < CF_2] \text{ or } [CF_2 < CF_2 + 12], \\ [CF_3 - 12 < CF_3] \text{ or } [CF_3 < CF_3 + 12], \\ \vdots \\ [CF_{n-1} - 12 < CF_{n-1}] \vee [CF_{n-1} < CF_{n-1} + 12] \end{array} \right\}$$

$$Dis_2 = 13n, Dis_3 = 13n \dots Dis_{n-1} = 13n$$

Regla 9:

Evitar los intervalos disonantes entre dos voces cualquiera: las segundas mayores y menores, las séptimas menores, cualquier intervalo aumentado o disminuido así como las cuartas perfectas.

$Dis_2, Dis_3, \dots, Dis_{n-1}$ no pertenece a $\{ [CF_2+1, CF_2+2, CF_2+7, CF_2+5, CF_2+6],$
 $[CF_3+1, CF_3+2, CF_3+7, CF_3+5, CF_3+6],$
 \vdots
 $[CF_{n-1}+1, CF_{n-1}+2, CF_{n-1}+7, CF_{n-1}+5, CF_{n-1}+6]\}$

$Dis_2 = (12-5)n, Dis_3 = (12-5)n \dots Dis_{n-1} = (12-5)n$
 $Dis_2 = 7n, Dis_3 = 7n \dots Dis_{n-1} = 7n$

Del análisis anterior, podemos concluir que para la segunda melodía a crear para acompañar a la primera, tendremos para la primera y última nota a lo más 3 diferentes notas a elegir y para la segunda, tercera hasta la penúltima nota tendremos 8 diferentes notas a elegir.

Para un compás donde se permita utilizar todas las figuras musicales tendremos,

$(3)(7)^{30}(3) = 2.0285406e+26$ posibles combinaciones de notas distintas.

7 Metodología

La programación declarativa es un paradigma de programación basado en condiciones, proposiciones, afirmaciones y restricciones que describen el problema y detallan su solución.

Debido a estas características resulta conveniente la utilización de la programación descriptiva como propuesta de solución de la técnica de contrapunto.

El lenguaje de programación declarativa Prolog, es altamente usado en el campo de la Inteligencia Artificial ya que es en sí mismo un problema de satisfacción de restricciones, es decir, trata de buscar valores para las variables que sea compatibles con las restricciones impuestas por las cláusulas. Su nombre proviene de la abreviatura de PROgrammation en LOGique [13].

Un programa escrito en PROLOG consiste en un conjunto de cláusulas positivas. Las cláusulas pueden ser de dos tipos: reglas y hechos. Una cláusula puede contener un encabezado y un cuerpo cuando son reglas y únicamente un encabezado cuando son hechos. En PROLOG no existen instrucciones de control, ya que la ejecución de un programa está basada en la unificación y en el backtracking.

Además, Prolog facilita las tareas de análisis gracias a su estructura de formalización de conocimiento en hechos y reglas.

Por lo tanto, para resolver la técnica de contrapunto utilizando el paradigma de satisfacción de restricciones, resulta conveniente utilizar el lenguaje de programación Prolog.

Para lograr el alcance del objetivo se propone realizar una abstracción de los conceptos generales en la técnica de contrapunto y con esto definir reglas para poder representar la información en un grafo con restricciones.

Seguido de esto, profundizar en el paradigma de programación de satisfacción de restricciones para presentar un algoritmo que resuelva el problema y su respectiva implementación en software.

Aprovechando la experiencia previa en la programación declarativa, la propuesta de herramienta de software fue SWI-Prolog.

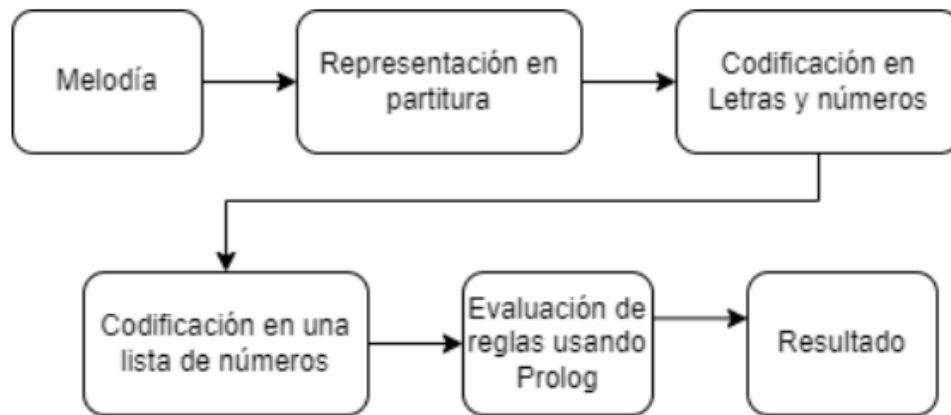


Figura 14: Diagrama de flujo de la metodología a utilizar

8 Codificación de la partitura

Hasta este punto, como hemos podido ver, la música se puede representar de manera visual con una partitura, además se ha visto que todos los conceptos previos están incluidos en esta representación.

Pero, para que una computadora pueda procesar esta información es necesario desarrollar un lenguaje entendible para la máquina, y aún más, un lenguaje que sea posible procesar con los lenguajes de programación elegidos.

A continuación se muestra el trabajo desarrollado para poder generar una codificación que abarque de manera extensiva todas las características involucradas en la composición musical por contrapunto.

8.1 Codificación en letras y números

Para codificar una partitura se utilizarán arreglos de 3 elementos cada uno. La primera letra representa la figura musical, la segunda tabla la nota y la tercer tabla el número de la octava, todas estas abreviadas de la siguiente forma:

Abreviación de figuras musicales	
Redonda	R
Blanca	B
Negra	N
Corchea	C
Semicorchea	S
Fusa	F

Figura 15: Abreviación de las figuras musicales

Abreviación de notas musicales	
Do	C
Do# / Reb	C# / Db
Re	D
Re# / Mib	D# / Eb
Mi	E
Fa	F
Fa# / Solb	F# / Gb
Sol	G
Sol # / Lab	G # / Ab
La	A
La# / Sib	A # / Bb
Si	B

Abreviación de Octavas	
Primera octava	1
Segunda octava	2
Tercera octava	3
Cuarta octava	4
Quinta octava	5
Sexta octava	6
Séptima octava	7
Octava octava	8
Novena octava	9

Figura 16 y Figura 17: Abreviación de notas y octavas musicales

Utilizando las tablas anteriores, podemos codificar una partitura musical, en un conjunto de letras y números, como se muestra en la siguiente imagen:

RBb4, BF5, BC5, ND5, NF5, NA5, CF5, CG5, RA5

Figura 18: Ejemplo de representación de una partitura en un arreglo de letras y números

8.2 Prolog y la codificación de partituras musicales

PROLOG es un lenguaje soportado por el área de programación lógica. PROLOG es también uno de los lenguajes más utilizado en el área de inteligencia artificial [14]. PROLOG surgió de un proyecto enfocado en el procesamiento del lenguaje natural, concretamente a principios de los años 70's en Francia, se presentó por vez primera, una versión preliminar de PROLOG. Su nombre proviene de la abreviatura de PROgrammation en LOGique [13].

Uno de los primeros objetivos al desarrollar PROLOG fue el realizar deducciones basadas en textos escritos en lenguaje natural, por sus inicios, fueron deducciones basadas en el idioma Francés para ser precisos [13]. Posteriormente, PROLOG fue ampliamente utilizado para escribir sistemas expertos en diversos ámbitos [14].

Un programa escrito en PROLOG consiste en la escritura de reglas lógicas, también conocidas como cláusulas positivas. Las cláusulas pueden ser de dos tipos: reglas y hechos. Una cláusula puede contener un encabezado y un cuerpo cuando son reglas y únicamente un encabezado cuando son hechos. En PROLOG no existen instrucciones de control, ya que la ejecución de un programa está basada en la unificación y en el backtracking.

El propósito de PROLOG no es generar todas las respuestas posibles a una meta dada, antes de pasar a la siguiente meta, en vez de ello, PROLOG genera una posible respuesta y la evalúa para la meta solicitada, con la expectativa de obtener la respuesta completa cuando ésta haya sido explorada hasta el final. Para la búsqueda de respuestas en PROLOG, se aplica la búsqueda primero en profundidad. Esto sirve como punto de elección, al cual regresar si al finalizar la búsqueda en profundidad no ha encontrado una solución que satisfaga las restricciones de la meta, entonces se elige otra rama para explorar dentro de la misma búsqueda en profundidad en búsqueda de una solución [14].

Hay algunos intentos del uso de PROLOG para representar y ejecutar melodías, veáse por ejemplo [17]. En [17], el autor propone confiar en el poder semántico de los números para describir hechos armónicos y conceptos tales como consonancia – disonancia. También confía en el poder de la recursión en PROLOG, para que en los programas pueda resintonizar una partitura bajo una entonación definida, de forma que también permita el generar una armonización automática de una canción dando como entrada la lista de nombres de acordes, y también para producir música ex-novo.

En nuestro proyecto de investigación, para poder realizar la evaluación de las reglas en Prolog, es necesario convertir toda la información anterior a una lista de números. El proceso para esta codificación es el siguiente:

Paso 1:

Se crearán 3 listas distintas, la primera lista representará a las figuras musicales. La segunda lista a las notas musicales y la tercer lista a la escala de cada una de las notas.

Paso 2:

Se crea una cuarta lista, que será el resultado de multiplicar cada uno de los elementos de la lista 3 (lista de octavas) por 12. Después se sumará cada elemento de la lista 2 (lista de notas) generando así una lista que represente tanto la nota y su octava juntas.

Paso 3:

Por último, cada elemento de la Lista 4 se multiplica n veces según cada elemento de la lista 1 Creando la lista final. Este paso se justifica con las equivalencias de notación musical mencionadas anteriormente.

Ejemplo:

Se tiene la lista: **RBb4,BF5,BC5,ND5,NF5,NA5,CF5,CG5,RA5**

Paso 1:

Crear 3 listas independientes, utilizando las abreviaturas de las tablas 1, 2, y 3. (figuras musicales, notas musicales y octavas respectivamente)

Lista1=[32, 16, 16, 8, 8, 8, 4, 4, 32]

Lista2=[11, 6, 1, 3, 6, 10, 6, 8, 10]

Lista3=[4, 5, 5, 5, 5, 5, 5, 5, 5]

8.3 Implementación de las reglas en Prolog

A continuación, se muestran las reglas implementadas en lenguaje Prolog:

%PARA TODAS LAS REGLAS "X" SE REFIERE A LA PRIMERA MELODÍA, LLAMADA CANTUS FIRMUS
O C.F. DE IGUAL FORMA "Y" SE REFIERE A LA SEGUNDA MELODÍA, LLAMADA DISCANTUS

%Calcula el intervalo entre dos notas

intervalo(X,Y,Z) :- Z is abs(X-Y).

%Devuelve el último elemento de una lista

ultimoElemen([Y], Y).

ultimoElemen(_|Xs, Y):- ultimoElemen(Xs, Y),!.

% Calcula el intervalo con signo. Valor positivo intervalo ascendente,

% valor negativo, intervalo descendente.

intervaloSigno(X,Y,Z) :- Z is (Y-X).

%Predicado nand

nand(A,B) :- not(and(A,B)).

%"X" se refiere al CF y "Y" al Discantus

% Regla 1:

%Comenzar y terminar en unisono, octava o quinta, a menos que

% la voz añadida sea inferior, en cuyo caso se comienza y se termina

% solamente en unisono u octava.

% Evalúa si el intervalo paralelo entre las dos primeras notas cumple

% la condición

regla1_1([],[]).

regla1_1([X_|_],[Y_|_]):- intervalo(X,Y,I), member(I,[0,12,7]),!.

```

% Evalúa si el intervalo paralelo entre las dos últimas notas cumple la
% condición
regla1_2(X,Y) :- ultimoElemen(X,U), ultimoElemen(Y,U2), intervalo(U,U2,I), member(I,[0,12,7]),!.

%Evalúa las dos condiciones anteriores al mismo tiempo.
regla1(X,Y) :- regla1_1(X,Y), regla1_2(X,Y),!.

%Regla 2:
%No utilizar unisono excepto al principio o al final.
% Ignora el primer elemento y evalúa el intervalo paralelo entre las
% siguientes 2 notas
regla2([],[]).
regla2([_,X|Xs],[_,Y|Ys]):- intervalo(X,Y,I),I\=0, regla2([X|Xs],[Y|Ys]),!.

% Evalúa el intervalo paralelo entre las 2 penúltimas notas, e ignora
% las ultimas 2 notas
regla2([X,_|[]],[Y,_|[]]):- intervalo(X,Y,I),I\=0,!.

%Regla3:
% Evitar las quintas u octavas paralelas entre dos voces cualesquiera y
% evitar las quintas u octavas paralelas «ocultas».
regla3([],[]).

%Evalúa que no haya intervalos paralelos de quinta u octava.
regla3([X|[X2|Xs]],[Y|[Y2|Ys]]):- intervalo(X,Y,I), intervalo(X2,Y2,I2),
not(member(I,[7,12])),not(member(I2,[7,12])), regla3([X2|Xs],[Y2|Ys]),!.

% Si hay intervalos de quinta u octava. que solo sea el segundo

```

% intervalo y se llegue por movimiento contrario.

*regla3([X|[X2|Xs]], [Y|[Y2|Ys]]):- intervaloSigno(X,X2,I), intervaloSigno(Y,Y2,I2), intervalo(X,Y,I3), ((I*I2)<0),*

not(member(I3,[-7,-12,7,12])), regla3([X2|Xs],[Y2|Ys]), !.

%Condición cuando solo queda un intervalo paralelo.

regla3([_|_], [_|_]).

%Regla4:

%No moverse en cuartas paralelas.

regla4([], []).

%Evalúa que no haya intervalos paralelos

regla4([X|[X2|Xs]], [Y|[Y2|Ys]]):- intervalo(X,Y,I), intervalo(X2,Y2,I2), I=I2, regla4([X2|Xs],[Y2|Ys]), !.

%Si hay intervalos paralelos que no sean de quinta.

regla4([X|[X2|Xs]], [Y|[Y2|Ys]]):- intervalo(X,Y,I), intervalo(X2,Y2,I2), I=I2, I=5, regla4([X2|Xs],[Y2|Ys]), !.

%Condición cuando solo queda un intervalo paralelo.

regla4([_|_], [_|_]).

%Regla 5:

%No moverse en terceras o sextas paralelas durante mucho tiempo.

%(Mucho tiempo se consideran mas de tres veces seguidas)

regla5([], []).

regla5([X|[X2|[X3|[X4|[X5|Xs]]]]], [Y|[Y2|[Y3|[Y4|[Y5|Ys]]]]]) :- intervalo(X,Y,I),

intervalo(X2,Y2,I2),

intervalo(X3,Y3,I3),

intervalo(X4,Y4,I4),

intervalo(X5,Y5,I5),

not((I=3,I2=3,I3=3,I4=3,I5=3)),

not((I=6,I2=6,I3=6,I4=6,I5=6)),

regla5([X2|[X3|[X4|[X5|Xs]]], [Y2|[Y3|[Y4|[Y5|Ys]]]), !.

%Regla 9:

% Evitar los intervalos disonantes entre dos voces cualquiera: las

% segundas mayores y menores, las séptimas menores, cualquier intervalo

% aumentado o disminuido así como las cuartas perfectas (en muchos contextos).

regla9([],[]).

regla9([X|Xs],[Y|Ys]):- intervalo(X,Y,R), not(member(R,[1,2,7,5,6])), regla9(Xs,Ys),!.

regla9([X|_],[Y|_]):- intervalo(X,Y,R), not(member(R,[1,2,7,5,6])),!.

9 Unión entre Python y Prolog

Si bien para realizar la evaluación de las reglas es bastante útil el lenguaje Prolog. Para realizar la interfaz gráfica no lo es tanto.

Por esta razón, el preprocesamiento, la reproducción del sonido y la interfaz gráfica están escritas en lenguaje Python.

Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos, entre ellos por ejemplo, la programación orientada a objetos, la programación imperativa y la programación funcional. Otros paradigmas están soportados mediante el uso de extensiones.

Python usa tipado dinámico y conteo de referencias para la gestión de memoria.[15]

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable.

Aunque la programación en Python podría considerarse en algunas situaciones hostil a la programación funcional tradicional del Lisp, existen bastantes analogías entre Python y los lenguajes minimalistas de la familia de Lisp como puede ser Scheme. [16]

9.1 La librería Pyswip

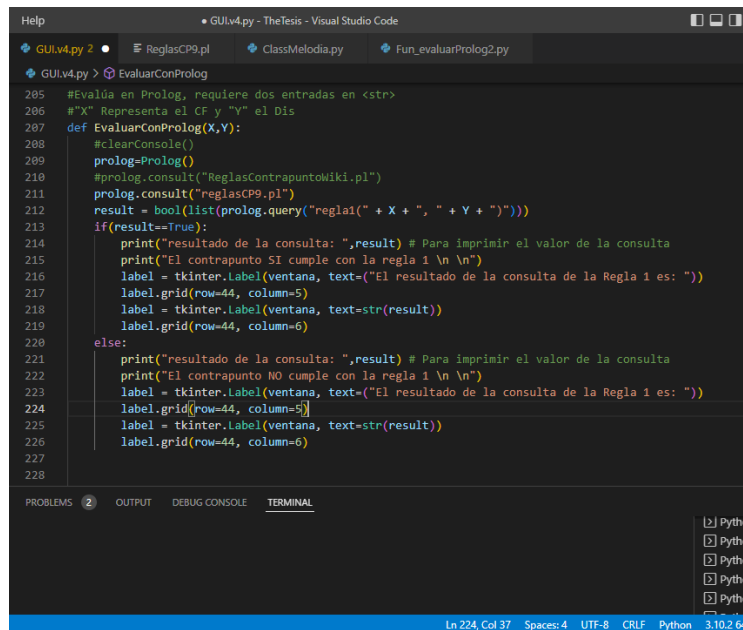
PySWIP es un puente entre Python y SWI-Prolog que permite consultar SWI-Prolog en los programas de Python.

A continuación se describirá el sistema de interfaz entre el módulo implementado en Prolog, y la interfaz gráfica de comunicación con el usuario.

En el programa principal se crea una función la cual recibe dos parámetros de entrada, “X” y “Y” que corresponden a las entradas codificadas para el Cantus Firmus y el Discantus, respectivamente.

Seguido de esto, la función evalúa independientemente cada una de las reglas y devuelve un valor de “True” cuando se cumple la regla, y “False”, cuando no se cumple.

A continuación se muestra un ejemplo para la evaluación de la Regla 1:



```
Help
• GUI.v4.py - TheTesis - Visual Studio Code
  • GUI.v4.py 2
  • ReglasCP9.pl
  • ClassMelodia.py
  • Fun_evaluarProlog2.py
  • GUI.v4.py > EvaluacionConProlog
    #Evalúa en Prolog, requiere dos entradas en <str>
    #“X” Representa el CF y “Y” el Dis
    def EvaluacionConProlog(X,Y):
        #clearConsole()
        prolog=Prolog()
        #prolog.consult("ReglasContrapuntowiki.pl")
        prolog.consult("reglasCP9.pl")
        result = bool(list(prolog.query("regla1(" + X + ", " + Y + ")")))
        if(result==True):
            print("resultado de la consulta: ",result) # Para imprimir el valor de la consulta
            print("El contrapunto SI cumple con la regla 1 \n \n")
            label = tkinter.Label(ventana, text=("El resultado de la consulta de la Regla 1 es: "))
            label.grid(row=44, column=5)
            label = tkinter.Label(ventana, text=str(result))
            label.grid(row=44, column=6)
        else:
            print("resultado de la consulta: ",result) # Para imprimir el valor de la consulta
            print("El contrapunto NO cumple con la regla 1 \n \n")
            label = tkinter.Label(ventana, text=("El resultado de la consulta de la Regla 1 es: "))
            label.grid(row=44, column=5)
            label = tkinter.Label(ventana, text=str(result))
            label.grid(row=44, column=6)
    205
    206
    207
    208
    209
    210
    211
    212
    213
    214
    215
    216
    217
    218
    219
    220
    221
    222
    223
    224
    225
    226
    227
    228
  PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL
  Python
  Python
  Python
  Python
  Python
  Python
  Ln 224, Col 37 Spaces: 4 UTF-8 CRLF Python 3.10.2 64-
```

Figura 19: Función para realizar consultas en Prolog desde Python

Una vez que se ha introducido las entradas X y Y, se muestra un módulo gráfico de interfaz con el usuario, que permite visualizar los resultados de las 9 reglas del contrapunto. Aparte de que tiene botones gráficos que permite la reproducción musical, tanto de las entradas como de las salidas.

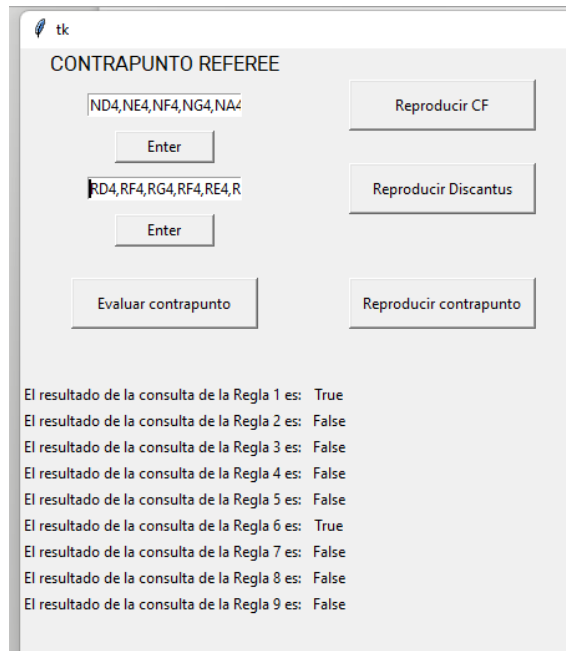


Figura 20: Interfaz gráfica del usuario

9.2 La librería Music21

Para lograr reproducir las melodías ingresadas se usó la librería Music21.

Music21 es un conjunto de herramientas basadas en la programación orientada a objetos, para analizar, buscar, y transformar música.

Para esto, se toma la lista de letras y números ingresada en la interfaz gráfica, y se procesa elemento por elemento.

La primera letra que está relacionada con la figura musical marcará el tiempo de dicho sonido.

Para una redonda, la letra “R” reproduce un sonido de 2 segundos, para una blanca “B” se reproducirá un sonido de un segundo de duración, para una negra “N” se reproducirá un sonido de medio segundo de duración, y así sucesivamente.

La nota y la altura que corresponde a la segunda y tercera parte de los elementos de la melodía se ingresan directamente como parámetros a la función ReproducirNota().

```
24
25 #Función para reproducir una sola nota
26 #Convierte la abreviación (letra inicial) de la figura en valor numérico
27 #quarterLength=4 es Redonda 'R' en 4-4,#quarterLength=2 es Blanca 'B' en 4-4
28 #quarterLength=1 es Negra 'N' en 4-4,#quarterLength=.5 es Corchea 'C' en 4-4
29 #quarterLength=.25 es Semiconchea 'S' en 4-4,#quarterLength=.125 es Fusa 'F' en 4-4
30 def ReproducirNota(Simbolo):
31 >     if Simbolo[0]=='R':...
33 >     elif Simbolo[0]=='B':...
35     elif Simbolo[0]=='N':
36         figura=1
37     elif Simbolo[0]=='C':
38         figura=.5
39     elif Simbolo[0]=='S':
40         figura=.25
41     elif Simbolo[0]=='F':
42         figura=.125
43     nota=Simbolo[1:]
44     return(note.Note(nota,quarterLength=int(figura)))
45
```

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL
resultado de la consulta: False
El contrapunto NO cumple con la regla 8

resultado de la consulta: False
El contrapunto NO cumple con la regla 9
```

Python Python Python Python Python Python Python Python Python Python Python Python

Ln 29, Col 52 Spaces: 4 UTF-8 CRLF Python 3.10.2 64-bit kite: ready

Figura 21: La función ReproducirNota(), regresa una melodía codificada que la librería Music21 puede reproducir.

Tanto el Cantus Firmus como el Discantus se pueden reproducir independientemente, o los dos al mismo tiempo, esto sirve al usuario como una referencia auditiva de la composición.

Que una melodía cumpla con todas las reglas del contrapunto, no significa que la melodía resultante sea agradable al escucharla. Ya que el gusto musical, como se mencionó en la introducción, está relacionado, individualmente con cada persona y de una manera más general, con la cultura, historia, región geográfica, etcétera.

```

43 nota=Simbolo[1:]
44 return(note.Note(nota,quarterLength=int(figura)))
45
46 #Reproduce CF
47 def ReproducirCF():
48     mel1.show('midi')
49
50 #Reproduce Discantus
51 def ReproducirDis():
52     mel2.show('midi')
53
54 #Toma el texto de la caja 1 <str> y lo convierte a <list>
55 > def EnterCF():...
56
57 #Toma el texto de la caja 2 <str> y lo convierte a <list>
58 > def EnterDis():...
59
60 #Combina el CF y el Discantus y lo reproduce
61 def ReprContrapunto():
62     cp = stream.Score()
63     cp.insert(0,mel1)
64     cp.insert(0,mel2)
65     cp.show('midi')
66
67 #Para limpiar la consola

```

Figura 22: Las funciones, reproducirCF() y reproducirDis(). Toman la melodía codificada y la convierten en un archivo MIDI que puede reproducir el reproductor de audio de la computadora

La función reproducir contrapunto toma las melodías creadas anteriormente y las coloca cada una en un canal del protocolo MIDI para poder reproducirlas simultáneamente.

```

77
78 #Combina el CF y el Discantus y lo reproduce
79 def ReprContrapunto():
80     cp = stream.Score()
81     cp.insert(0,mel1)
82     cp.insert(0,mel2)
83     cp.show('midi')
84
85 #Para limpiar la consola

```

Figura 23: La función reproducirContrapunto() reproduce dos melodías simultáneamente.

10 Conclusión

En este trabajo se muestra una implementación basada en PROLOG y Phyton para representar y ejecutar partituras musicales que han sido previamente codificadas mediante lista de números. Se utilizó el ambiente de SWI-Prolog debido a su potencialidad recursiva, lo que ayuda a que a través de reglas y hechos intentemos hacer de forma automática el contrapunto musical de una melodía.

Si bien para expresar las reglas de contrapunto es bastante útil el lenguaje Prolog. Para realizar la interfaz gráfica no lo es tanto. Por esta razón, el preprocesamiento, la reproducción del sonido y la interfaz gráfica están escritas en lenguaje Python.

Para lograr reproducir las melodías ingresadas se usó la librería Music21. Music21 es un conjunto de herramientas basadas en la programación orientada a objetos, para analizar, buscar, y transformar música.

El sistema de contrapunto musical parte de realizar una codificación funcional para convertir las partituras a una lista de números y letras, y también a una lista de solo números. Como se mencionó anteriormente en el documento, se consideran tres características de la música, la altura, la nota y la escala.

Estas características son representadas bajo símbolos y posiciones en un pentagrama dentro de la partitura, para poder procesar la información por la computadora, cada uno de estos elementos primero se transforma a letras y números y luego a listas de números. Esta lista de números y letras serán la entrada a nuestro sistema de revisión de reglas de contrapunto.

También se muestra que el número de posibles combinaciones para un compás es un número de una magnitud de 10^{35} , y que restringiendo este número de posibilidades con las reglas de contrapunto, el espacio se reduce a un número de magnitud 10^{18} , lo cual sigue siendo un espacio de soluciones muy grande.

Si bien es posible utilizar un algoritmo determinista para conocer todas las posibles soluciones al problema de contrapunto, en la práctica no resulta viable, ya que como se mencionó en el capítulo 3, el número de soluciones es muy amplio, y la música al tratarse de una expresión

artística, el verdadero valor de una respuesta correcta o incorrecta dependen del compositor y del oyente, y no de si se cumplen correctamente las reglas mencionadas.

La sugerencia que se hace para trabajos futuros es utilizar un algoritmo heurístico para poder encontrar una solución de forma automática, también se sugiere separar el espacio de soluciones utilizando probabilidades, o algún proceso que permita separar el espacio de soluciones en subconjuntos, con la idea de reducir las evaluaciones necesarias.

Si se evalúa auditivamente un resultado y se sabe que para este resultado hay un conjunto de posibles soluciones parecidas, será más fácil tomar la decisión de seguir buscando en ese conjunto, o bien cambiar completamente a otro conjunto con características distintas.

11 Bibliografía

- [1] Pecino, I. (2014). Teoría de Sistemas Musicales 2ª Parte: Fractalidad en la Serie Armónica. Revista online de música y arte sonoro.
- [2] Nierhaus, G. (2009). Algorithmic composition: paradigms of automated music generation. Springer Science & Business Media.
- [3] Dean, R. T. (2018). The Oxford handbook of algorithmic music. Oxford University Press.
- [4] Romero N.C.G. (2014). Resolución de contrapunto de primera especie utilizando programación con restricciones. Pontificia Universidad Católica de Valparaíso.
- [5] Kennedy, M., & Kennedy, J. (2013). The Oxford dictionary of music. Oxford University Press.
- [6] Small, C. (1996). Music, society, education. Wesleyan University Press.
- [7] De Candé, R. (2002). Nuevo diccionario de la música (Vol. 2). Grasindo.
- [8] Dionisio Garma, F. P. (2013). Evaluación de la consistencia epistemológica y el nivel de aplicabilidad de las propuestas pedagógicas en los proyectos educativos institucionales, en las instituciones educativas estatales del nivel secundario, Ugel N° 6, Vitarte.
- [9] Randel, D. M. (Ed.). (2014). The Harvard dictionary of music. Harvard University Press.
- [10] Michels, U., Vogel, G., Mames, L., & Marcos, J. J. R. (1994). Atlas de música. Alianza.
- [11] Piston, W., & Westergaard, P. (1968). Conversation with Walter Piston. Perspectives of New Music, 3-17.
- [12] Zoreda, M. L., & Lima, J. V. (2006). El texto literario adaptado como andamiaje para el desarrollo de competencias lingüísticas e interculturales en inglés como lengua extranjera. Estudios de Lingüística Aplicada, (43), 31-54.

- [13] Alain Colmerauer and Philippe Roussel. (1996). The birth of Prolog. History of programming languages---II. Association for Computing Machinery, New York, NY, USA, 331–367. <https://doi.org/10.1145/234286.1057820>
- [14] Russel, S.J. Norvig, P.. (2004). Inteligencia Artificial, un enfoque moderno. Madrid: Pearson Educación, S.A.
- [15] Lutz, M. (2013). Learning python: Powerful object-oriented programming. " O'Reilly Media, Inc."
- [16] Matthes, E. (2019). Python crash course: A hands-on, project-based introduction to programming.
- [17] Luca Perotti (2021), Music in Prolog (LucaX8), Página Web: <https://swi-prolog.discourse.group/t/music-in-prolog/3466>