



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Programa de Maestría y Doctorado en Música

Facultad de Música

Instituto de Ciencias Aplicadas y Tecnología

Instituto de Investigaciones Antropológicas

GUITARRA POSTONAL EN LÍNEA: UN SITIO WEB CON TRES APLICATIVOS  
AUXILIARES EN EL ESTUDIO Y LA IMPLEMENTACIÓN DE RECURSOS  
POSTONALES EN LA GUITARRA

TESIS / TESINA  
QUE, PARA OPTAR POR EL GRADO DE  
MAESTRO EN MÚSICA (TECNOLOGÍA MUSICAL)

PRESENTA  
LIC. HEGEL EMMANUEL PEDROZA VILLALOBOS

TUTOR  
HUGO SOLÍA GARCÍA (UAM LERMA)

CIUDAD DE MÉXICO. (JUNIO) 2023



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

*Declaro conocer el Código de Ética de la Universidad Nacional Autónoma de México, plasmado en la Legislación Universitaria. Con base en las definiciones de integridad y honestidad ahí especificadas, aseguro mediante mi firma al calce que el presente trabajo es original y enteramente de mi autoría. Todas las citas de obras elaboradas por otros autores, o sus referencias, aparecen aquí debida y adecuadamente señaladas, así como acreditadas mediante las convenciones editoriales correspondientes.*

# Agradecimientos

Este trabajo no hubiese sido posible sin el acompañamiento, cuidado e interés de las personas de mi círculo familiar cercano, Ivelin, Rulfo, Oliver, Lupe y René.

En el mismo sentido, agradezco el apoyo de amistades invaluable que en distintas conversaciones tuvieron la oportunidad de retroalimentarme en los procesos de la investigación, gracias Julián, Sergio, Gerardo, Mariana, Miguel y Enrique.

De igual forma, es necesario agradecer a mi profesor el Dr. Jorge David García y mi tutor el Dr. Hugo Solís García, sus atenciones resultaron fundamentales para dar forma y vida a la investigación, así como los comentarios hechos por el Dr. Roberto Velasco Segura, el Mtro. Diego Alberto Tinajero Islas, el Mtro. Eduardo Piastro, el Mtro. Aarón Flores Noval y el Mtro. Enrique García Alcalá que resultaron imprescindibles para corroborar la calidad y el alcancé del trabajo logrado.

Por último, muchas gracias a los músicos Ulises Peña, Josafath Larios y Rodrigo Cruz que me permitieron explorar creativamente las herramientas desarrolladas y de igual forma agradezco a todas las demás personas que me brindaron ayuda, directa o indirectamente, a lo largo de la maestría.



# Índice general

<b>Introducción</b>	<b>1</b>
<b>1. Fundamentos y Referentes Teóricos</b>	<b>7</b>
1.1. Allen Forte, una teoría transversal . . . . .	10
1.2. Miles Okazaki, diagramas de acordes . . . . .	17
1.3. Dariusz Terefenko, potencial armónico de los tricordes . . . . .	18
1.4. John O’Gallagher, estudio de horas . . . . .	21
1.5. Jonathan de Souza, coordenadas traste-cuerda . . . . .	23
<b>2. Revisión de Herramientas Similares</b>	<b>29</b>
2.1. Herramientas para calcular y generar diagramas de acordes. . . . .	31
2.2. Herramientas para la creación de partituras de acompañamiento. . . . .	35
2.3. Herramientas para la operación, análisis y visualización de material musical postonal . . . . .	38
<b>3. Implementación de Aplicativos y Tecnologías Desarrolladas</b>	<b>45</b>
3.1. Conceptualización y Diseño . . . . .	46
3.2. Aplicativos Implementados . . . . .	55
3.2.1. Calculador de Diagramas . . . . .	55
3.2.2. Generador de Acompañamientos . . . . .	60
3.2.3. Visualizador de Formas Primas . . . . .	62

3.3. Tecnologías Desarrolladas . . . . .	65
3.3.1. Interprete de Contextos Tonales y Formas Primas . . . . .	65
3.3.2. Descifrador de Diagramas . . . . .	72
3.3.3. Reproductor de Notas . . . . .	75
3.3.4. Operadores de Transposición, Inversión y Complemento . . . . .	76
3.3.5. Dibujador de Polígonos de Conjuntos . . . . .	83
3.3.6. Descifrador de Acompañamientos . . . . .	85
3.3.7. Partitura Interactiva . . . . .	88
3.3.8. Espacio Tridimensional . . . . .	91
<b>Conclusiones</b>	<b>97</b>
<b>Bibliografía</b>	<b>106</b>





# Introducción

Se presenta una tesina que describe detalladamente el desarrollo de tres aplicativos web integrados en un sitio web denominado “Guitarra Postonal en Línea”(<https://guitarrapostonalenlinea.netlify.app/>). Estos aplicativos funcionan como herramientas auxiliares para el estudio y la composición de material musical postonal en la guitarra. Su conceptualización, se basa en teorizaciones nacidas en el ámbito del jazz, que exploran el uso de procedimientos postonales en este género.

La motivación principal para esta investigación surge de mi formación como guitarrista eléctrico y mi interés personal en explorar lenguajes compositivos y procedimientos para la improvisación que no se basen en el sistema escala-acorde ampliamente aceptado en los espacios formativos del jazz. Además, en mi experiencia como docente, pude notar la necesidad de proporcionar a los estudiantes una herramienta tecnológica que les ayudara en el estudio y la implementación de material musical que no se rige por dicha relación.

Aunque los aplicativos desarrollados no fueron conceptualizados desde una dimensión pedagógica, facilitan el acceso a procesos comunes en la práctica profesional del guitarrista, como el cálculo de digitaciones y el desarrollo de acompañamientos. Además, ofrecen la posibilidad de analizar el material musical desde las perspectivas de la armonía del jazz y la teoría de conjuntos de tonos clase, lo que ayuda a comprender mejor los conjuntos de notas y sus relaciones tanto en contextos tonales como no tonales.

La conceptualización de los aplicativos se basa en trabajos previos de Miles Okazaki sobre el cálculo de diagramas de acordes de conjuntos de tres notas, el uso de los tricordes como elementos armónicos en posibles contextos tonales explorados por Dariusz Terefenko, y el estudio de series dodecafónicas formadas por un mismo tipo de tricorde desarrollado por John O’Gallagher. Estos trabajos se abordan de manera

transversal utilizando la teoría de conjuntos de tonos clase propuesta por Allen Forte en 1973 para el análisis y la composición de material musical postonal.

Los aplicativos desarrollados son: un “Calculador de Diagramas de Acordes” que permite calcular diagramas de acordes de guitarra a partir de un conjunto de notas seleccionado; un “Generador de Acompañamientos” que habilita al usuario a desarrollar una conducción de voces lógica a partir de una progresión armónica y una disposición de voces seleccionada; y un “Visualizador de Formas Primas” que ofrece una propuesta de visualización funcional interactiva, que permite explorar las relaciones de todas las formas primas posibles de conjuntos de entre tres y seis notas en un espacio tridimensional, configurado tomando en cuenta características de las formas primas, como su número de notas o la diversidad de su contenido interválico.

Para su elaboración, se requirió idear mecanismos para traducir en código las características físicas de la guitarra, especialmente en lo que se refiere al cálculo de digitaciones y la medición de la distancia entre notas dentro del diapason. Para este propósito, los trabajos de Jonathan de Souza resultaron fundamentales, ya que proponen una comprensión del espacio del diapason de un instrumento con trastes dentro de un sistema generalizado de intervalos expresados mediante coordenadas traste-cuerda.

En la implementación de los aplicativos, se emplearon tecnologías específicas para resolver problemáticas concretas, las cuales posteriormente se interconectaron para el funcionamiento de cada aplicación. Por tanto, es posible encontrar las mismas tecnologías utilizadas en más de un aplicativo.

En cuanto a la creación de las tecnologías, se emplearon técnicas comunes en el campo computacional, como el trabajo con listas y el cálculo de operaciones matemáticas para generar combinaciones de elementos y comparar similitudes entre conjuntos numéricos. Aunque estas técnicas resultaron útiles, su papel no fue fundamental en la conceptualización de los aplicativos, por lo que se configuran únicamente como medios tecnológicos para su desarrollo.

A lo largo de la investigación, se consultaron aplicativos similares que pudieran proporcionar información valiosa sobre interfaces, herramientas y mecanismos tecnológicos y metodológicos para la implementación de desarrollos de este tipo. En esta

búsqueda, se encontraron aplicaciones que cubren parcialmente algunas de las funcionalidades conceptualizadas, pero no se encontró un caso similar al propuesto.

La investigación se vio limitada en su capacidad de aprovechar elementos valiosos, tales como la retroalimentación documentada de comunidades artísticas y tecnológicas interesadas en las temáticas abordadas o la exploración creativa de las funcionalidades implementadas. En gran medida, esto se debió a que se tuvo que dedicar una cantidad significativa de tiempo a adquirir conocimientos y habilidades en programación web, por lo que estos elementos se consideran posibles incorporaciones futuras que no se reflejan en el momento del cierre de esta tesina.

La tesina consta de tres capítulos y un breve cierre configurado a manera de conclusiones, donde se exponen breves ejemplos de material musical creado usando los aplicativos.

Agrupados como “Fundamentos y Referentes Teóricos”, en el primer capítulo, se presentan las teorizaciones de Okazaki, Terefenko, O’Gallagher y De Souza, antecedidas de un breve contexto sobre el uso de la postonalidad en el jazz, que aporta un panorama general sobre guitarristas y exponentes del género que emplean recursos postonales en sus prácticas creativas.

En el capítulo dos, se hace una revisión de herramientas web similares, agrupándoles de acuerdo a con algunas funcionalidades de los aplicativos propuestos. En ese sentido, se estructuraron tres categorías:

1. Herramientas para calcular y generar diagramas de acordes.
2. Herramientas para la creación de partituras de acompañamiento.
3. Herramientas para la operación, análisis y visualización de material musical postonal.

Se abordaron tres casos diferentes para cada categoría, seleccionándolos en función de sus capacidades y objetos de estudio, tratando de demostrar cómo es que se relacionan con las funcionalidades implementadas en los aplicativos desarrollados.

Entrando en el cuerpo de la investigación, en el tercer capítulo, “Implementación de Aplicativos y Tecnologías Desarrolladas”, se aborda la lógica bajo la cual esta concebida la estructura de los aplicativos, qué tecnologías les integran y qué herramientas fueron empleadas para su desarrollo.

Este tercer capítulo se divide en dos secciones. En la primera, se describe la configuración de cada uno de los tres aplicativos, mostrando sus interfaces y diseño, así como una lista de las librerías de código utilizadas en su implementación. Y en la segunda, se abordan en detalle las tecnologías desarrolladas que hacen posible el funcionamiento de los aplicativos, explicando el orden lógico de sus algoritmos y los elementos ideados y empleados para su operación.

Debido a que se busca presentar la investigación de manera accesible para cualquier persona, no se ha incluido ninguna línea del código desarrollado (mismo que se puede consultar en el repositorio del sitio: <https://github.com/hegelespaul/Postonal-Guitar>), sin embargo, se hace un uso extenso de diagramas, figuras y tablas que ayudan a comprender de mejor manera la lógica de la programación de los algoritmos.

Cabe mencionar, que a lo largo de la investigación se llevó a cabo una exhaustiva búsqueda documental sobre programación y desarrollo de aplicativos web. Se consultaron numerosas fuentes, tales como artículos científicos, manuales técnicos, tutoriales y cursos en línea, entre otras. Todo ello con el objetivo de obtener una amplia comprensión sobre las herramientas y lenguajes de programación necesarios para la implementación de los aplicativos propuestos.

Asimismo, durante el proceso de investigación se contó con el apoyo y tutoría de expertos en programación, quienes brindaron valiosas orientaciones y asesoramiento en la selección de las herramientas y tecnologías más adecuadas para el desarrollo de los aplicativos. Gracias a ello, se pudo adquirir y consolidar habilidades en programación, tanto en el uso de librerías de código preexistentes como en la creación de algoritmos y funcionalidades específicas para cada aplicativo. Estas tutorías resultaron esenciales para el logro de los objetivos propuestos y para la consecución de aplicativos funcionales y eficientes.

Cabe mencionar, que en este trabajo se ha optado por limitar la bibliografía específica sobre programación, dejando de lado los manuales y guías introductorias a lengua-

jes de programación y enfoques teóricos generales sobre la materia. En cambio, se ha priorizado la inclusión de trabajos referenciados que sustenten los argumentos y teorizaciones expuestas, así como aquellos que presenten desarrollos tecnológicos similares a los aplicativos propuestos. Esto se debe a que la bibliografía empleada para programar fue sumamente amplia y diversa, y en muchos casos su valor radicó más en su utilidad práctica a la hora del uso de los lenguajes de programación que en su relevancia argumentativa.

Es importante destacar que los logros registrados en este informe corresponden a una primera aproximación de los aplicativos. Es necesario tener en cuenta que las conclusiones presentadas aquí son una instantánea del estado actual de la investigación, y se espera que surjan comentarios y propuestas adicionales que enriquezcan aún más los aplicativos desarrollados

Finalmente, es importante mencionar que gran parte de la bibliografía utilizada en este trabajo está en inglés. Por esta razón, se solicita amablemente a los lectores que tengan en cuenta que las citas y referencias bibliográficas en otros idiomas distintos al español son traducciones propias. Para facilitar la lectura del documento, se ha decidido no incluir las citas en su idioma original.

Sin más, deseando una amble lectura, se presenta el cuerpo del documento.



# 1 Fundamentos y Referentes Teóricos

Los recursos teóricos de los músicos de jazz son muy diversos<sup>1</sup>, hoy en día es posible encontrar algunos que recurren a organizaciones musicales fuera del sistema tonal para el desarrollo de sonoridades particulares.<sup>2</sup> Desde los años cincuenta, pero con mayor énfasis en los años sesenta, existieron muchos exponentes del jazz en Estados Unidos cuya obra cuestionó formalmente los límites de la tonalidad. La gran mayoría de los casos fueron músicos que buscaron generar piezas contemporáneas a sus tiempos y de ahí que existiese un cuestionamiento político y cultural al status quo.<sup>3</sup>

Eran tiempos difíciles los años 60. Teníamos el asunto de los derechos civiles, teníamos a (Martin Luther) King, teníamos a Malcolm (X), teníamos a los (Black) Panthers. Había tanta diversidad sucediendo. La gente gritaba por sus derechos y por querer ser iguales, ser libres. Y, naturalmente, la música refleja todo ese período... todo ese tiempo definitivamente influyó en la forma en que tocábamos. Creo que ahí es donde entró esa forma realmente libre. Todos querían alejarse de la cosa rígida, lejos de lo que estaba pasando antes; querían relacionarse con lo que estaba sucediendo y estoy seguro de que la música salió de todo eso.<sup>4</sup>

- Rashied Ali.<sup>5</sup>

---

<sup>1</sup>Véase Rawlins, Bahha, y Tagliarino (2005), Ligon (2001) o Levine (2011).

<sup>2</sup>Como ejemplo, el trabajo de Graaf (2017) aborda una colección muy representativa de distintos enfoques para ampliar la paleta de recursos de los músicos compositores e improvisadores en el jazz mediante la implementación de constructos teóricos provenientes de la música académica de principios del siglo XX.

<sup>3</sup>Existen muchos ensayos que problematizan dichas décadas, en caso de tener mayor interés, se recomienda las lecturas de Anderson (2006), Philippe Carles (1973) o Robinson (2005).

<sup>4</sup>Traducción propia de la cita extraída de (Shipton, 2007, p. 798).

<sup>5</sup>Baterista de Free Jazz, en su carrera musical sobresale el hecho de que llegó a formar parte

Gracias a sus prácticas se articularon los estilos musicales que hoy en día distinguimos y nombramos como Free Jazz y Avant-Jazz, este último, tal vez menos conocido, puede ser entendido como “música que es en gran parte improvisada utilizando muchas de las técnicas y dispositivos de la música contemporánea de las tradiciones clásicas europeas” (Parsons, 2017, p. 10). Músicos como John Coltrane, Alice Coltrane, Eric Dolphy, Anthony Braxton, Albert Ayler, Ornette Coleman, Cecyl Taylor, Yusef Lateef, Barbara Donald, George Ruseell o Miles Davis, por mencionar algunos, desde enfoques muy particulares buscaron ir más allá, reordenar o subvertir las lógicas de la tonalidad de distintas formas, ya sea eliminando armonías<sup>6</sup>, buscando disposiciones y clasificaciones diferentes de las notas y las escalas musicales<sup>7</sup> o incidiendo en prácticas que incorporasen notaciones gráficas o improvisaciones libres o dirigidas.<sup>8</sup> Muchos de sus procedimientos, innovadores en su momento, me atrevo a afirmar que hoy en día pueden llegar a ser considerados como técnicas o recursos del género.

En la inmensa discografía del jazz producida en la actualidad<sup>9</sup> podemos encontrar casos donde existen tratamientos armónicos, rítmicos o melódicos que hacen uso de recursos musicales que no obedecen a la relación tradicional del sistema escala-acorde<sup>10</sup> que opera dentro de la armonía de jazz. Música de exponentes como Ben Monder<sup>11</sup>,

---

de las agrupaciones de John Coltrane, grabando juntos “Interstellar Space”, un disco compuesto únicamente por duetos de saxofón y percusiones.

<sup>6</sup>Véanse las transcripciones de Ornette Coleman en Schuller (1961). “Forerunner”, fidedigna a la grabación, cuenta con indicaciones de compases de solos *ad libitum* sin ninguna armonía escrita. Un caso similar sucede con los temas de Albert Ayler en *Albert Ayler Sheet Music* (2011), en numerosas ocasiones los temas son únicamente melodías cortas sin ninguna armonía especificada.

<sup>7</sup>Los manuscritos de Cecil Taylor ofrecen una clara muestra del desarrollo de sistemas y notaciones no convencionales, esto se puede apreciar en el registro fotográfico de Meehan (2018). Asimismo, otro ejemplo notable es el manuscrito de la obra “A Love Supreme” de John Coltrane accesible en *John Coltrane Music Manuscript* (2005).

<sup>8</sup>El catálogo de obras de Anthony Braxton, disponible en Tri-Centric Foundation (2018), incluye varias partituras con notación gráfica. Por otro lado, la amplia producción del “Sun Ra Arkestra” es un ejemplo destacado de la exploración de diversas dinámicas de improvisación, tal y como se puede apreciar en su presentación en el Festival de Jazz de Montreal de 1976 *Sun Ra Arkestra 1976 Montreux, Switzerland TV* (2020).

<sup>9</sup>Tan solo la base de datos de Discogs (2022) arroja más de 9,680 lanzamientos a nivel global de álbumes de jazz en el 2022, reediciones incluidas.

<sup>10</sup>Un sistema de improvisación y análisis musical que consiste en vincular acordes con escalas dentro de un sistema tonal. Es “generalmente aceptado en el mundo del jazz actual” (Spitzer, 2001, p. 115) y existen un gran número publicaciones que lo abordan, entre ellas, se recomienda Nettles, Graf, y Berklee College of Music (1997) o Rawlins y cols. (2005).

<sup>11</sup>Feakes (2018) aborda un análisis fisicalista de varios temas de Ben Monder, entre ellos “Oceana”, del que el mismo Monder expresó en una entrevista en (Olson, 2006, para. 23) que “... el tema original que ocurre dos veces [...] Lo estoy dividiendo básicamente en dos temas, uno más escalar y otro más angular, y voy y vengo entre ellos, haciendo cosas diferentes con ellos. Así que ahí está la primera



Mary Halvorson<sup>12</sup>, Craig Taborn<sup>13</sup> o Rudresh Mahanthappa<sup>14</sup>, sucede en gran parte fuera de dicha relación.

La tarea de mapear todos los esquemas compositivos y de improvisación fuera del sistema tonal resulta ser enormemente compleja y supera los alcances de la presente investigación. Por ello, en lugar de intentar cubrir todo el espectro de enfoques y métodos, se optó por revisar teorizaciones documentadas en publicaciones actuales que exponen técnicas y métodos posttonales para la composición e improvisación en el jazz. Esto resultó ser una tarea útil y mucho menos compleja para encontrar puntos de partida o referentes teóricos para la conceptualización de un sitio web auxiliar en la composición y el estudio de material musical posttonal implementado en la guitarra.

Entre las propuestas teóricas seleccionadas se encuentran los diagramas de acordes de Okazaki (2015), la perspectiva del potencial armónico de los tricordes de Terefenko (2014) y el estudio de horas de O’Gallagher (2015). A lo largo de este capítulo se presentará una descripción general de estas propuestas, tratando brevemente cómo es que se incorporaron en los aplicativos del sitio.

Para evaluar transversalmente las teorizaciones recién mencionadas, se sirvió de la nomenclatura de conjuntos de tonos clase de Forte (1973). De igual forma, se expondrá brevemente en qué consiste y se explicará porque se decidió emplear en vez de otras sistematizaciones similares como las de Rahn (1987) o Yamaguchi y Lovell (2006).

En el desarrollo de los aplicativos, el uso de un esquema de coordenadas que sintetizara el espacio del diapasón de la guitarra fue fundamental para abstraer en código los procedimientos musicales tratados. Para el cálculo de diagramas y la generación de acompañamientos, se implementó una matriz de tres dimensiones [*cuern*, *traste*, *tono*], basándose en los trabajos de De Souza (2018, 2022). Estos trabajos conceptualizan el diapasón de un instrumento como un sistema generalizado de intervalos

---

declaración, que es bastante no tonal.”

<sup>12</sup>En *Mary Halvorson - Interval Training Exercises on the Guitar* (2020), Halvorson expone un recurso melódico que emplea para la improvisación que consiste en alejarse de los patrones tradicionales de escalas y pensar un solo intervalo como un modo.

<sup>13</sup>En la transcripción del tema “Neverland” de Taborn expuesta en Travaglini (2020), se puede constatar a lo largo de la pieza lo que auditivamente se infiere como el empleo de contrapuntos no tonales.

<sup>14</sup>En *JazzHeaven* (2021), expone un compendio de procedimientos que emplea en la improvisación melódica, entre los que destaca el uso de células de notas conceptualizadas a manera de conjuntos de tonos clase.

y transformaciones, representado por un plano cartesiano de dos dimensiones [*traste, cuerda*]. En el último apartado previo al cierre de este capítulo, se le revisará y se expondrá de manera clara y concisa en qué consiste y cómo es que se le tomó en consideración para la implementación de las funcionalidades de los aplicativos desarrollados.

Por último, es necesario apuntar que, aunque la gama de conceptos y técnicas que puede cobijar el término de postonalidad puede ser muy amplia en tanto que el prefijo *pos* explicita un *después de* la tonalidad, en el ámbito académico anglosajón desde hace ya varios años se puede encontrar una acotación del término a las teorías de conjuntos de tonos clases, la atonalidad y el serialismo, remitiendo así el concepto de postonalidad a los trabajos de músicos y autores como Allen Forte, Milton Babbitt, John Rahn o George Perle. Esto se puede constatar al revisar materiales considerados guías introductorias a los procedimientos de la música postonal, como las publicaciones de Roig-Francoli (2008) o de Straus (2005), que, sin necesariamente otorgar una definición clara y frontal de lo que se entiende por postonalidad, exponen procedimientos básicos ocurridos en las corrientes musicales recién mencionadas.

## 1.1. Allen Forte, una teoría transversal

Allen Forte fue profesor en la Universidad de Yale en Estados Unidos, donde enseñó durante más de 40 años y fue una figura influyente en el departamento de música. En 1973, con la publicación “The Structure of Atonal Music”, Forte daba a conocer un sistema de ordenamiento de estructuras dodecafónicas basado en una teoría de conjuntos en donde clasificaba grupos de notas y exponía posibles operaciones y relaciones de similitud. Su obra opera como un enfoque sistémico para abordar prácticas atonales y dodecafónicas y, “[...]aunque precipitó varias revisiones que fueron críticas con su sistema incluso en sus primeros años, su sistema (o un pariente cercano) ahora es un estándar. De hecho, hoy no es difícil encontrar ensayos que le utilicen sin citar su libro[...].” (Schiano, 1999, p. 1).

Forte no ha sido el único músico que ha propuesto un enfoque sistémico para abordar la música atonal, Arnold Schönberg en diversas oportunidades comentó sobre los procedimientos de su método de los doce tonos e incluso, Boulez (1963), mucho antes que Forte, ya había comentado sobre su perspectiva de la música arquitectónica en

su ensayo “Penser la musique aujourd’hui”, sin embargo, como bien apunta Célestin Deliège en el Primer Congreso sobre Análisis Musical en Colmar, Francia en 1989:

Forte merece el premio Nobel de Análisis Musical, si lo hubiera. En efecto, el problema del análisis de la música atonal radica en que, a diferencia de la música tonal, no existe una taxonomía análoga a las denominaciones y descripciones de acordes que fueron elaboradas y teorizadas a lo largo de más de dos siglos. El gran mérito de Forte es haber inventado para la música atonal una taxonomía que no es un simple injerto de categorías armónicas, sino que más bien busca su equivalente.” (Nattiez, 2008, p. 2).

Desarrollar a profundidad la teoría de conjuntos de tono clase es una tarea que involucra una publicación por si sola, en este apartado se limitará únicamente a exponer los recursos empleados en el sitio: forma prima, retrógrado, transposición, inversión, vector interválico y complemento.

En principio, el sistema de Forte evita el uso convencional de nombres de notas y les asigna un valor numérico donde “do” es representado por el número 0, “do sostenido” por 1 y así subsecuentemente hasta llegar a “si” que es representado por el número 11<sup>15</sup>. La tabla 1.1 hace cuenta de dicha asignación numérica, indicando el número de tono clase correspondiente a cada nota.

C	C#-Db	D	D#-Eb	E	F	F#-Gb	G	G#-Ab	A	A#-Bb	B
0	1	2	3	4	5	6	7	8	9	10	11

Tabla 1.1: Muestra los tonos clase correspondientes a cada nota en la escala cromática, donde los números del 0 al 11 representan cada una de las doce notas posibles.

Para clasificar y ordenar los posibles grupos de notas, Forte hace uso del concepto de forma prima. La forma prima es la versión más compacta para representar un conjunto

<sup>15</sup>Originalmente, para referirse a los números 10 y 11 se propuso el uso de letras T y E (Ten, Eleven), respectivamente. También es común encontrar el uso de las letras A y B para referirse dichos dígitos, sin embargo, se ha decidido emplear los números 10 y 11 ya que facilitan la tarea de cálculo en el ámbito computacional y siguen denotando con claridad suficiente a qué tipo de notas se refieren.

de notas, surge de analizar el conjunto desde cada uno de sus componentes tomando en consideración una lectura tanto ascendente como descendente de sus tonos clase. Forte clasificó cada forma prima posible bajo un nombre compuesto por dos dígitos separados por un guión conocido como número de Forte. El primer dígito indica la cantidad de notas de la forma prima y el segundo, el orden de aparición de la forma prima en función de un cálculo ascendente de la forma prima con los intervalos más reducidos a aquella con los intervalos más amplios; 3-1 es (0,1,2), 3-2 es (0,1,3) y así subsecuentemente hasta agotar todas las posibilidades, llegando a la última forma prima posible 12-1 que es equivalente a (0,1,2,3,4,5,6,7,8,9,10,11).

Por su parte, el procedimiento para el cálculo de un forma prima, aunque se abordará con mayor detalle en la sección 3.3.1 del documento, se puede exponer brevemente de la siguiente forma:

- En primer lugar, se evalúa la distancia del intervalo más amplio de las posibles permutaciones del grupo de notas.
- Luego, se analizan los intervalos internos, y la forma prima se identifica como la permutación que contiene los intervalos menores en relación al punto de partida.
- Finalmente, la forma prima se escribe entre paréntesis con los valores resultantes transpuestos a cero y fuera del paréntesis se indica la nota desde la cual se realiza el cálculo.

Es importante destacar que, si la forma prima se encuentra con sus componentes leídos de forma descendente, es decir, en el sentido del 11 al 0 en lugar del 0 al 11, se especifica que dicha forma prima se encuentra en retrógrado con una comilla antes del número que especifica el tono clase desde donde se evalúa. A continuación, a manera de ejemplo, la Figura 1.1 muestra el procedimiento para el cálculo de la forma prima de un conjunto de notas [0,1,6].

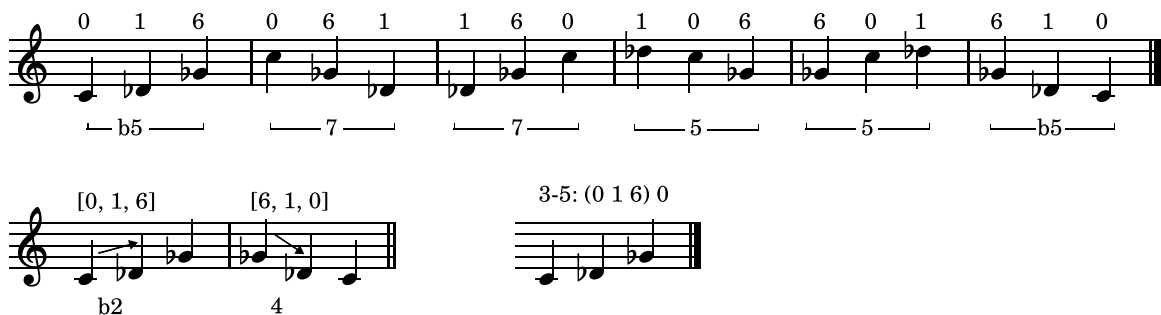


Figura 1.1: Muestra el procedimiento para calcular la forma prima del conjunto de notas [0,1,6]. En el primer sistema se muestran las permutaciones del conjunto, especificando la longitud de el intervalo externo de cada una de ellas. En el segundo sistema, se muestran aquellas permutaciones con el intervalo externo más reducido y la permutación seleccionada como forma prima debido a que posee el intervalo interno más reducido, en este caso, la permutación seleccionada es (0,1,6) a partir de cero, y de acuerdo a su orden de aparición, le corresponde el número de Forte 3-5.

En su publicación Forte incluye un listado de todas las formas primas posibles así como los número de Forte asignados, siendo hoy en día dicho listado fácilmente accesible a través de diversos sitios web, siendo tal vez su artículo en Wikipedia<sup>16</sup> uno de los más frecuentemente visitados.

En lo que respecta a los conjuntos, cada estructura de notas puede ser sujeta a diversas operaciones comúnmente empleadas para crear variaciones sobre un tema o crear o hilar bloques de sentido a nivel compositivo. Las más elementales son el retrógrado, la inversión y la transposición.

- La inversión, implica transformar una serie de notas de manera que su dirección melódica sea opuesta a la original. Por ejemplo, si se tiene una serie [0,6,1], que comienza en el tono cero, asciende seis semitonos y luego desciende cinco semitonos, su inversión comenzará en el tono cero, descenderá seis semitonos y luego ascenderá cinco semitonos, resultando en una nueva serie de notas [0,6,11].
- El retrógrado, se refiere a la lectura de las notas en orden inverso, por ejemplo, el retrógrado de la serie de notas [5,6,0] resulta en [0,6,5]; los elementos de la serie permanecen iguales, solo su orden de lectura se invierte.
- La transposición, se refiere a cambiar la altura de las notas por un intervalo determinado, por ejemplo, la interválica [0,1,6] descrita desde 5 resulta en los

<sup>16</sup>[https://en.wikipedia.org/wiki/List\\_of\\_set\\_classes](https://en.wikipedia.org/wiki/List_of_set_classes)

tonos clase  $[5,6,0]$ , mientras que, descrita desde cero, resulta en los tonos clase  $[0,1,6]$ .

La Figura 1.2 muestra las operaciones recién descritas aplicadas al conjunto de notas  $[0,1,6]$ .

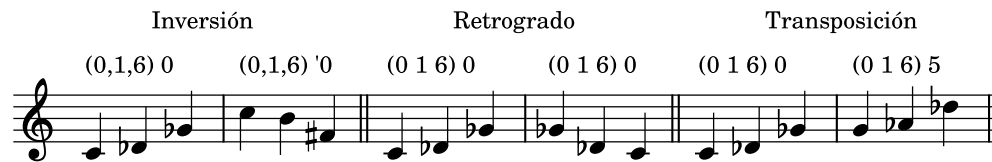


Figura 1.2: Se muestran las operaciones de inversión, retrógrado y transposición, aplicadas al conjunto de notas  $[0, 1, 6]$ . Al aplicar la inversión, se observa que la forma prima del conjunto se modifica, a diferencia del retrógrado donde las notas permanecen iguales después de la operación. En este caso, la forma prima se define desde el retrógrado de cero. En cuanto a la transposición, es importante destacar que al desplazar la interválica cinco semitonos hacia arriba, la nota de referencia para evaluar la forma prima cambia de cero a cinco, es decir, de “do” a “fa”. Este cambio es crucial para asegurarse de evaluar la forma prima desde el tono clase adecuado.

En lo que respecta al vector interválico, este consiste en una lista de seis elementos que sintetiza el contenido interválico de una agrupación de notas. Cada elemento del vector representa la cantidad de intervalos de un tipo específico que existen dentro de la estructura comenzando con el intervalo más pequeño (segunda menor) y avanzando hasta el tritono, que es el límite máximo de intervalo que se considera ya que los intervalos más grandes son considerados un reflejo de los más pequeños. Por ejemplo, en tanto que son inversiones de sí mismos, una sexta menor es un reflejo de una tercera mayor, y una quinta es un reflejo de una cuarta.

Para calcular el vector interválico, se mide la cantidad de intervalos posibles entre las distintas notas de la agrupación y se suman sus ocurrencias en la casilla del intervalo que les corresponde dentro del vector. Este procedimiento se ilustra en la Figura 1.3.



Figura 1.3: Muestra el vector interválico del tricorde  $[0,1,6]$  y el procedimiento para su cálculo. Consiste en medir todas las distancias entre sus componentes, en el ejemplo, las distancias ya medidas se especifican con paréntesis, siendo el vector resultante  $[1,0,0,0,1,1]$  que indica que el tricorde  $[0,1,6]$  tiene como contenido interválico una segunda menor, una quinta disminuida y una cuarta.

Por último, en lo que respecta al complemento, se refiere a las notas que quedan fuera de una selección de notas y que completan el total del espectro dodecafónico. Por ejemplo, cuando se elige una estructura  $[0,1,6]$ , las notas  $[2,3,4,5,7,8,9,10,11]$  quedan fuera de dicha selección y representan el complemento necesario para cubrir el total del espectro de las doce notas (Figura 1.4).



Figura 1.4: Muestra el complemento de la forma prima  $(0,1,6) 0$ , un conjunto de nueve notas que en forma prima se describe como  $(0,1,2,3,4,6,7,8,9)'11$ .

El sistema de Forte a lo largo del tiempo ha sido objeto de críticas y no es el único de su tipo. Rahn (1987) propuso un sistema similar con ligeras diferencias en la metodología para el cálculo de las formas primas, sin embargo, mientras que Forte dedica una gran parte de su publicación al estudio de las grandes estructuras de conjuntos intrincados que denomina “set-complex”, el sistema de Rahn no aborda estas estructuras de manera tan exhaustiva. Para una lectura detallada sobre diferentes propuestas para la sistematización de la música postonal que han tenido aceptación en el ámbito académico, se invita la consulta del trabajo de Schuijjer (2008). En él, es posible ver la evolución de los estudios de la postonalidad en el ámbito académico estadounidense y contrastar a profundidad las teorías de distintos exponentes, incluidas las de Forte y Rahn.

Otra propuesta reciente de sistematización, similar a la de Forte, es la presentada por Yamaguchi y Lovell (2006). En su publicación “Complete Thesaurus of Music”, evalúan una clasificación de los conjuntos de notas basada en las formas primas y sus retrógrados, describiendo cada conjunto según su interválica mediante una notación convencional de armonía de jazz (1-b2-2-b3-3-4-b5-5-b6-6-b7-7), para posteriormente clasificarlos según la cantidad de semitonos que poseen en su interválica interna.

Al ser una sistematización más reciente que la de Forte, no ha alcanzado la misma aceptación en los espacios académicos y no es tan ampliamente conocida<sup>17</sup>. Además, al examinar su lista de tricordes, es posible encontrar una inconsistencia notoria, ya que no incluye el tricorde 1-4-5 que corresponde al retrógrado de la forma prima [0,2,7]. Yamaguchi y Lovell presentan un total de 19 posibles tricordes en lugar de los 20 verdaderamente posibles. La tabla 1.2 muestra los tricordes del sistema de Yamaguchi y Lovell y su relación con las formas primas de Forte, evidenciando su falta de consideración del tricorde recién mencionado.

Tipos	Tricordes											
	[0,1,2]	[0,1,3]	[0,1,4]	[0,1,5]	[0,1,6]	[0,2,4]	[0,2,5]	[0,2,6]	[0,2,7]	[0,3,6]	[0,3,7]	[0,4,8]
Forte	[0,1,2]	[0,1,3]	[0,1,4]	[0,1,5]	[0,1,6]	[0,2,4]	[0,2,5]	[0,2,6]	[0,2,7]	[0,3,6]	[0,3,7]	[0,4,8]
Retrógrados		[0,3,2]	[0,3,4]	[0,4,5]	[0,5,6]		[0,3,5]	[0,4,6]	[0,5,7]		[0,4,7]	
Sin Semitonos						1-2-3	1-2-4	1-2-b5	1-2-5	1-b3-b5	1-b3-5	1-3-5
Retrógrados							1-b3-4	1-3-b5	###		1-3-5	
1 semitono		1-b2-b3	1-b2-3	1-b2-4	1-b2-b5							
Retrógrados		1-2-b3	1-b3-3	1-3-4	1-4-b5							
2 semitonos	1-b2-2											

Tabla 1.2: Compara los tricordes de Forte y sus retrógrados con los propuestos Yamaguchi en sus tres categorías (sin semitonos, 1 semitono, 2 semitonos). Es posible ver una correspondencia en casi todos los casos, excepto en el retrógrado del tricorde [0,2,7] marcado con símbolos de gato. Yamaguchi y Lovell no le toman en cuenta al considerar que no se encuentra en “Forma Normal”, sin embargo, dicha forma, que es entendida como la forma reducida de las notas, tampoco no se encuentra en los retrógrados de las formas primas de otros tricordes sí tomaron en consideración dentro de su sistema.

En síntesis, el sistema de Forte fue pionero al sistematizar de forma clara y consistente todas las posibilidades armónicas derivadas de la combinación total o parcial de las

<sup>17</sup>Esto se puede constatar haciendo una simple búsqueda de la cantidad de citas que recibe Yamaguchi en comparación con las de Forte.



doce notas y desde hace ya varios años goza de una amplia aceptación en diversos espacios académicos alrededor del mundo, razones por lo que se ha decidido, por encima de otro tipo de sistematizaciones similares, considerarle como la herramienta para abordar transversalmente las teorizaciones implementadas en la investigación, que, desde el propio ámbito del jazz, proponen el uso de recursos posttonales en el género.

## 1.2. Miles Okazaki, diagramas de acordes

Miles Okazaki es un destacado guitarrista de jazz y teórico musical con una amplia trayectoria en la escena jazzística neoyorquina. Además de ser docente de guitarra en el departamento de jazz de la Universidad de Princeton, Nueva Jersey, Okazaki es reconocido por su lenguaje particular en la guitarra y su capacidad para integrar la teoría musical en su proceso creativo. Cada una de sus composiciones se enmarca dentro de algún constructo o exploración teórica que idea previo a su desarrollo compositivo, lo que se refleja claramente en su discografía y sus partituras.

En su publicación del 2015, “Fundamentals of Guitar”, dedica un apartado al estudio de los diagramas de acorde de todos los posibles conjuntos de dos y tres notas, generando un mapeo metódico que excluye redundancias y repeticiones de diagramas, lo que resulta en una herramienta muy útil para guitarristas que buscan expandir su conocimiento de acordes y explorar nuevas sonoridades sin necesariamente partir desde un enfoque tonal para la evaluación de los acordes.

Las constantes en los diagramas es que son figuras únicas, no cuentan con cuerdas al aire ni notas repetidas y no tienen una extensión de más de cinco trastes. El proceso para calcularles consiste en agrupar las cuerdas en todas las combinaciones posibles, encontrar las relaciones interválicas de una misma forma prima en un espacio de 5 trastes para cada combinación, y filtrar las estructuras con intervalos repetidos. La Figura 1.5 ejemplifica el procedimiento recién descrito, mostrando en primera instancia la interválica de la estructura seleccionada, luego uno de los posibles grupo de cuerdas, y por último, los diagramas resultantes posibles en dicha selección de cuerdas.

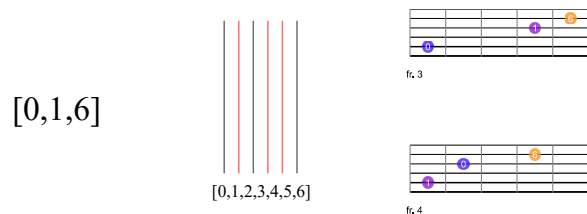


Figura 1.5: Muestra el procedimiento de Okazaki (de izquierda a derecha) para el cálculo de diagramas. Primero, la selección de la estructura a calcular. Segundo, la elección de cuerdas donde buscar los valores de notas que comprenden la estructura. Tercero, localizar a lo largo del diapasón las posibles maneras de digitar dichas notas en dichas cuerdas en un rango no mayor a cinco trastes. A diferencia de cómo están implementados en el sitio, los diagramas de Okazaki no tienen valores de notas y únicamente están marcados mediante círculos, ello pensando en la memorización de estructuras sin importar en qué parte del diapasón se interpreten ni que notas contengan, consideración dejada de lado pues se sugiere que, en los aplicativos, en todo momento las estructuras estén vinculadas directamente con el espacio de su ocurrencia en el diapasón y los valores de notas que les componen.

Los diagramas de acordes de Okazaki solo contemplan todas las agrupaciones de hasta tres notas, pero al aprovechar las seis cuerdas de la guitarra y las capacidades informáticas, se ha replicado su metodología para conjuntos de hasta seis notas, logrando generar una gran cantidad de acordes, algunos poco usuales en la música tonal.

Para evaluar estos acordes, se pueden emplear nomenclaturas conocidas como la de Forte (1973), pero para comprender mejor cómo utilizarlos como elementos compositivos o para la improvisación, es necesario analizarlos con detenimiento y encontrar sus posibles usos en contextos tonales. Esta tarea ha sido llevada a cabo por Terefenko (2014), quien propone vincular tricordes dodecafónicos a contextos tonales comúnmente empleados en el jazz.

### 1.3. Dariusz Terefenko, potencial armónico de los tricordes

Dariusz Terefenko es un reconocido pianista, profesor de jazz polaco-estadounidense. Actualmente es profesor asociado en la “Eastman School of Music” de Rochester, Nueva York donde forma parte del cuerpo docente de los departamentos de jazz y teoría musical. Ha publicado numerosos artículos y libros sobre teoría y análisis del

jazz, incluyendo su libro del 2014, “Jazz Theory: From Basic to Advanced Study” que se ha convertido en un texto ampliamente utilizado en escuelas de música y conservatorios alrededor del mundo.

En un capítulo dentro de dicha publicación, aborda las posibilidades del empleo de tricordes entendidos desde el sistema de Forte para el desarrollo de estructuras superiores de acordes en un contexto tonal. Su trabajo consiste en explorar conexiones entre las lógicas organizativas de la armonía de jazz y la teoría de conjuntos de tono clase para desarrollar recursos que permitan ampliar el lenguaje del músico en la composición y la improvisación en un contexto tonal.

Terefenko, propone que los doce tricordes en forma prima sean tomados en consideración no solo desde su forma prima sino también desde lo que él denomina sus “inversiones”, generando así veinte formas únicas (el término inversión que emplea hace alusión al retrogrado de las formas primas). Posteriormente, clasifica estas veinte formas únicas, nominadas como “complejos”, en cuatro categorías: diatónicas, de tonos enteros, triádicas y de semitonos y evalúa cada uno de los complejos en función de las distintas capacidades que tienen, en tanto componentes de acordes, para operar en contextos tonales armónicos mayores, menores, dominantes o disminuidos.

La Tabla 1.3 muestra cómo se agrupan los veinte posibles tricordes en las cuatro distintas categorías y la Figura 1.6 muestra como el complejo  $[0,1,6]$  puede ser empleado en los cuatro distintos contextos tonales.

Diatonicos		Tonos Enteros			Triadicos		De semitonos				
$[0,2,5]$	$[0,2,7]$	$[0,2,4]$	$[0,2,6]$	$[0,4,8]$	$[0,3,6]$	$[0,3,7]$	$[0,1,2]$	$[0,1,3]$	$[0,1,4]$	$[0,1,5]$	$[0,1,6]$
$[0,3,5]$	$[0,5,7]$		$[0,4,6]$			$[0,4,7]$		$[0,2,3]$	$[0,3,4]$	$[0,4,5]$	$[0,5,6]$

Tabla 1.3: Muestra las cuatro familias en las que se agrupan los veinte posibles tricordes. En la primer fila se muestran los doce tricordes en su expresión de forma prima y en la segunda, su retrógrado o “inversión”.

**[016] - COMPLEJO**
**MAYOR****MENOR**

G♭maj7#11 Amaj7#9 Emaj9#5 Dmaj7(♯7) E♭m13 Em13#5 C#m11(♯7)

**DOMINANTE****DIMINUIDO**

C7♭9#11 B9(♭9) B♭7#9#5 A13#9 G♭7#11 F7♭9#5 E13(♭13) E♭13#9 D7(♯7) G◦11(♯7) E◦9(♭13) C◦13(♭9)

Figura 1.6: Muestra cómo el tricorde [0,1,6] puede ser utilizado en los cuatro contextos tonales. Es importante destacar que los tricordes pueden generar tensiones en los acordes de maneras diversas, incluso utilizando tensiones poco comunes. Por ejemplo, en el segundo acorde del contexto mayor, se emplea el tricorde [0,1,6] para crear un acorde Amaj7(#9), que posiblemente haga referencia a una relación escala-acorde de la escala Lidio #9, es decir, el cuarto modo de la escala menor melódica. Este tipo de tensiones poco usuales pueden ser utilizadas para crear sonoridades diferentes a las relaciones escala-acorde comúnmente empleadas.

Existen procedimientos similares al recién expuesto, en tanto que conciben a los acordes como el resultado de estructuras organizadas bajo una lógica diferente que el tradicional apilamiento de notas a intervalos de terceras, como el concepto “Generic Modality Compression’ de Goodrick, Miller, y Feist (2012), las estructuras derivadas de la escala disminuida propuestas por Barry Harris expuestas en la publicación de Kingstone (2006) o el sistema “Minor Conversion” de Martino y Baruso (1989), por mencionar algunos; sin embargo, no toman en cuenta la generación del material armónico desde una perspectiva de la teoría de conjuntos de tonos clase, y, a pesar de existir sistematizaciones desde la postonalidad para su aplicación en el jazz como las de Yamaguchi y Lovell (2006) o O’Gallagher (2015), estas no necesariamente consideran el uso del material postonal en contextos tonales.

En los aplicativos, de igual forma se evalúan las estructuras armónicas desde la armonía del jazz y la teoría de los conjuntos de tonos clase, al relacionar, mediante la semejanza de sus componentes, los conjuntos de notas seleccionadas a casos de acor-

des comúnmente usados en la armonía del jazz. Aunque estas relaciones no operan de la misma manera que los procedimientos propuestos por Terefenko, que se basan en los cuatro contextos tonales mayor, menor, dominante y disminuido, ambas metodologías son similares en su objetivo de sugerir posibles lecturas y usos tonales del material postonal evaluado.

Desde otro nivel de análisis, también se pueden evaluar las relaciones entre distintos conjuntos de notas, un campo que ha sido explorado en los estudios de series de O’Gallagher. Este enfoque, sumado a lo previamente expuesto, amplía el panorama de los conjuntos de notas al considerarlos parte de un espectro dodecafónico, lo que permite generar estructuras de sentido más amplias que un solo conjunto de notas.

## 1.4. John O’Gallagher, estudio de horas

John O’Gallagher es un destacado saxofonista de jazz y compositor estadounidense que ha grabado varios álbumes como líder y ha colaborado con importantes músicos en la escena del jazz de Nueva York. Como profesor ha impartido diversas cátedras en varias universidades y conservatorios alrededor del mundo enfocadas principalmente en el estudio de elementos postonales en la composición e improvisación en el Jazz. Su libro del 2015, “Twelve Tone Improvisation: Method for using tone rows in Jazz”, explora el uso de series dodecafónicas compuestas por agrupaciones tricordes que comparten una misma forma prima.

Las series que aborda corresponden a las horas del sistema de organización musical “Tone Clock” de Schat (1993). Un sistema que a grandes rasgos puede ser entendido como una organización musical que consiste en doce series dodecafónicas sin notas repetidas, siendo cada una de ellas, el resultado de la combinación de tricordes que comparten una misma forma prima, a excepción de la hora “X” que la comprenden tres cuatricordes disminuidos, debido a que el tricorde disminuido  $[0,3,6]$  no permite abarcar por completo el espectro de las doce notas con cuatro tricordes únicos sin repetición de notas.

En su “zodiaco de horas” (Figura 1.7)<sup>18</sup>, Schat sintetiza visualmente su sistema.

---

<sup>18</sup>Consideración suficiente para entender la propuesta de O’Gallagher, sin embargo, un abordaje profundo del sistema “Tone Clock” implica el entendimiento de conceptos como “giros de volante”

O’Gallagher, basándose en dicha representación, explora posibles usos tonales de cada hora y propone permutaciones de las diferentes series, ofreciendo así un compendio de ejercicios para desarrollar la improvisación en cada una de las horas.

De O’Gallagher se retoma la idea de que una estructura de notas forma parte de una serie dodecafónica, implementado en el aplicativo dedicado al cálculo de diagramas, la posibilidad de visualizar todas las combinaciones posibles del complemento dodecafónico en agrupaciones de entre tres y seis notas, abriendo así la posibilidad de considerar series que nazcan de la combinación de conjuntos de diferentes cantidades de notas y distintas formas primas.

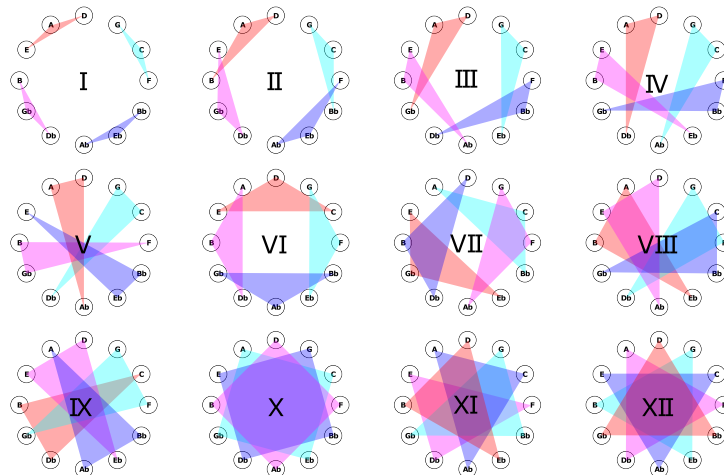


Figura 1.7: Muestra el “Zodiaco de Horas” teorizado por Schat. A diferencia de la disposición tradicional de un círculo de quintas, el orden de las series dentro del zodiaco de horas no obedece a una secuencia ascendente o descendente, sino a una lógica que se basa en la triangulación de notas que corresponden a un mismo tipo de tricorde, a excepción de la hora “X”, la cual tiene una disposición basada en un cuatricorde disminuido.

---

o las “leyes de la simetría cromática”, elementos ampliamente abordados por McLeod (1994).



Figura 1.8: Un extracto de la publicación de O’gallagher donde se muestran las doce horas de Schat. Nótese que para describir las horas emplea una nomenclatura basada en los intervalos internos de los tricordes que les componen. Por ejemplo, el tricorde  $[0,1,6]$  que origina la hora “X” es representado por “1 + 5”. Por tanto, a diferencia del “Zodiaco de Horas”, el orden de las series obedece al orden ascendente de los intervalos de los tricordes y no a las triangulaciones dentro del círculo de quintas expuesto por Schat.

Aunado a los recursos teóricos musicales recién expuestos, como elemento vinculante entre la teoría musical, las capacidades físicas del diapasón de la guitarra y su sistematización en algún tipo de esquema numérico empleable a nivel de código computacional, los trabajos de Jonathan De Souza, que a continuación se exponen, han resultado fundamentales para la configuración de una matriz con la que es posible evaluar las posiciones de las notas en el diapasón, así como las digitaciones de acordes y la conducción de voces dentro de progresiones armónicas.

## 1.5. Jonathan de Souza, coordenadas traste-cuerda

Jonathan De Souza es profesor asociado de la Facultad de Música Don Wright de la Universidad de Western de Ontario, Canadá. Se especializa en teorías del ritmo, la música popular y la semántica musical. Es autor del libro “Music at Hand:

Instruments, Bodies, and Cognition”, el cual explora la relación entre la música y el cuerpo humano. También ha publicado artículos en diversas revistas especializadas y ha presentado ponencias en conferencias internacionales de música.

Los trabajos de De Souza (2018, 2022), exploran el desarrollo de un conocimiento musical específico al que se accede en tanto la persona intérprete y/o creadora musical de un instrumento se enfrenta a las prestaciones <sup>19</sup> del mismo, un conocimiento que se sitúa en lo que Baily y Driver (1992) llamaron el “Pensamiento Espacio/Motor” y a lo que De Souza nomina como “Guitar Thinking”.

Inspirado en los “Sistemas Generalizados de Intervalos y Transformaciones” de Lewin (2007)<sup>20</sup>, De Souza desarrolla un modelo transformacional del diapasón inscrito en un plano cartesiano compuesto por coordenadas (*traste-cuerda*) (Figura 1.9). En dicho espacio se suscriben todas las posibles notas del diapasón de cualquier instrumento musical imaginado y pueden ser descritas diferentes transformaciones, desde movimientos sencillos de un punto a otro hasta movimientos complejos que impliquen una serie de pasos coordinados de más de un punto a la vez.

---

<sup>19</sup>El término empleado es “affordances” definido por primera vez en Gibson (1966). Es un concepto ampliamente empleado en el campo del diseño y la psicología cognitiva y comúnmente es entendido como las acciones posibles formadas por las relaciones entre un agente y su entorno.

<sup>20</sup>De manera sintética, un sistema generalizado de intervalos se puede definir como una tupla que se compone de un espacio definido, un grupo matemático de intervalos y una función que puede mapearse a cada miembro del grupo de intervalos, para mayor información sobre la definición de un sistema y sus condiciones se sugiere remitirse a la obra de Lewin (2007).



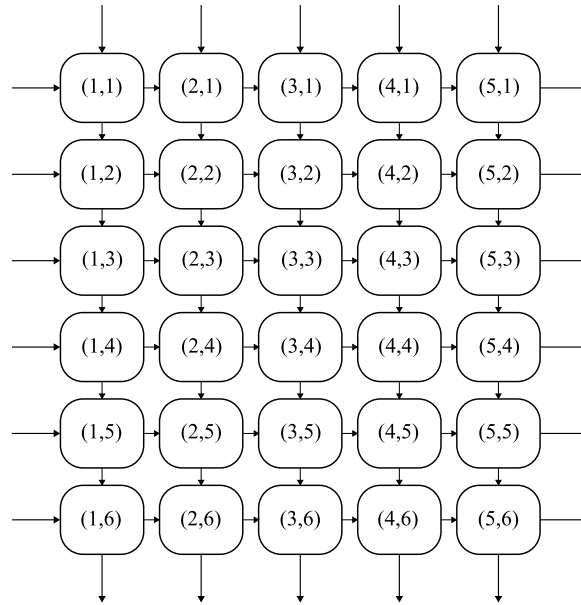


Figura 1.9: Desarrollado como una adaptación de un diagrama expuesto en De Souza (2018) que mapea parcialmente el espacio bidimensional infinito de coordenadas [*traste*, *cuerda*], la figura ejemplifica los primeros cinco trastes de un sistema de coordenadas correspondiente a una guitarra de seis cuerdas. Mediante flechas especifica el sentido de las coordenadas: de izquierda a derecha el número de trastes y de arriba hacia abajo el número de cuerdas, tal y como sucede en los diagramas de acordes.

Sus exploraciones del diapasón de la guitarra han servido para evidenciar procesos compositivos e interpretativos que nacen de un conocimiento guitarrístico (“Guitar Knowledge”) basado en posiciones de dedos y en sus transformaciones y no desde un conocimiento musical (“Musical Knowledge”) basado en elementos de teoría musical como armonía, conducción de voces o contrapunto. Ejemplos de este conocimiento guitarrístico plasmado en composiciones, pueden ser los estudios sencillos de Leo Brouwer analizados en De Souza (2022). En estos estudios, de los que el propio Brouwer apunta que “No es la música lo que es sencillo, siempre tengo una armonía complicada o significados o líneas o figuras rotas en la música que no son sencillas. Lo que es sencillo es la manera de tocarlos. Simplicidad para los dedos; complejidad para el oído.”<sup>21</sup>, es posible ver transformaciones sencillas en las digitaciones del diapasón que a nivel teórico exponen armonías complejas.

<sup>21</sup>Traducción propia de los comentarios de Leo Brouwer en Tonebase Guitar (2018).

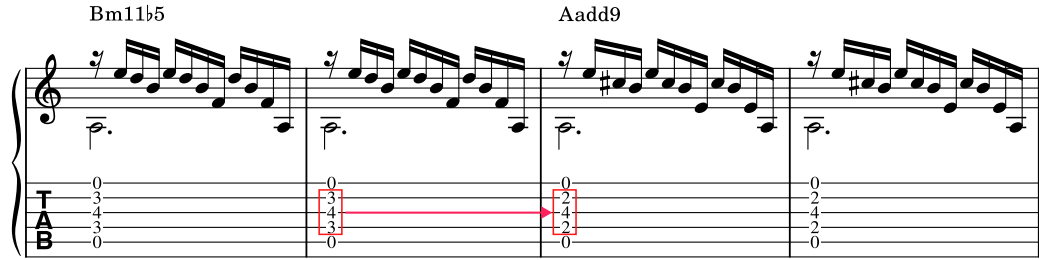


Figura 1.10: En la figura se muestra un pasaje del “Estudio Sencillo XI” de Leo Brouwer. En los compases siete al diez de este estudio, arriba mostrados, se puede apreciar cómo un simple movimiento de dos dedos a un traste por detrás conlleva una progresión armónica compleja, tal como se ilustra en la tablatura que muestra las posiciones de los dedos en cada compás

En los aplicativos, la posibilidad de describir del diapason mediante un conjunto de coordenadas, fue tomada en consideración para el desarrollo de una matriz  $[cuerda, traste, nota]$  que permitiese describir la posición de una nota en el diapason de la guitarra. La matriz no solo permite evaluar qué lugar ocupa la nota en el diapason, sino también considerar su dimensión de nota, abriendo la posibilidad de realizar cálculos y operaciones musicales difíciles de ejecutar en un sistema de coordenadas  $[traste, cuerda]$ . La figura 1.11 muestra como la representación gráfica del movimiento de un par de notas en un plano bidimensional  $[traste, cuerda]$  resulta ser muy diferente a la de un espacio tridimensional  $[cuerda, traste, nota]$ , exponiendo que un movimiento mínimo en el diapason de la guitarra, como lo es un salto de cuerda, musicalmente puede implicar un movimiento más amplio de intervalo de cuarta o tercera.

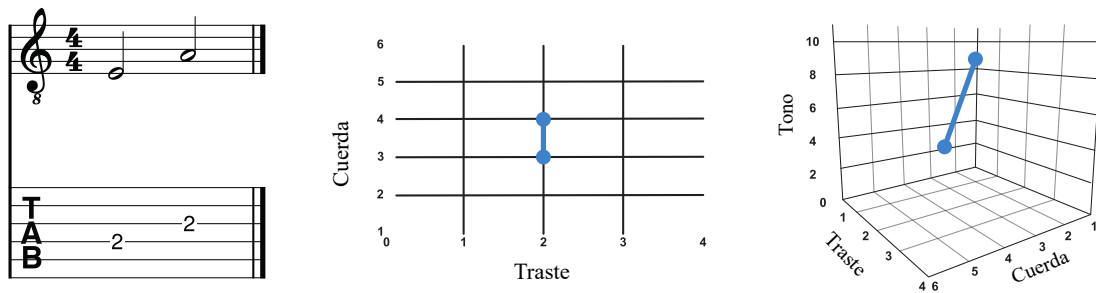


Figura 1.11: Ilustra un salto de cuerda en un espacio bidimensional  $[cuerda, traste]$  respecto a uno tridimensional  $[cuerda, traste, nota]$ . Es posible notar que al tomar en consideración una dimensión de valor de nota, un movimiento de salto de cuerda que parece cercano en tanto que implica un movimiento de un solo valor en el eje de las cuerdas, musicalmente supone un salto más amplio.

Los elementos tratados a lo largo de este capítulo han sido categorías, teorías y referentes fundamentales para la creación de los aplicativos desarrollados y, aunque a lo largo de la investigación en diferentes momentos se han relacionado de formas diferentes, desde un principio han servido como ejes para articular la investigación, Aunado a ello, el caso de De Souza ha servido como un punto de partida para poder empezar a materializar en código las ideaciones de herramientas auxiliares para el estudio y la composición de material musical postonal implementado en la guitarra.

A pesar de que los aplicativos desarrollados tienen dinámicas particulares que se basan en teorías musicales del jazz actual abordadas desde la teoría de Forte, se han encontrado herramientas similares que han servido para evaluar la viabilidad de los aplicativos y contrastar su enfoque basado en la fisicalidad de un instrumento y posibles cruces entre las teorías del jazz y la teoría de conjuntos de tonos clases. En el siguiente capítulo se abordan un compendio de estas herramientas agrupadas en tres categorías diferentes: herramientas para calcular y generar diagramas de acordes, herramientas para la creación de partituras de acompañamiento y herramientas para la operación, análisis y visualización de material musical postonal.



## 2 Revisión de Herramientas Similares

Hoy en día, el uso de herramientas tecnológicas auxiliares para el estudio de materiales musicales es relativamente común en las grandes ciudades, basta con ir a cualquier universidad o conservatorio para observar a algún estudiante usando algún dispositivo móvil para afinar sus instrumentos, leer partituras o para acceder a aplicativos de teoría musical y/o entrenamiento auditivo. Estadísticas como las de INEGI (2019); Kemp (2022) reportan que más del setenta por ciento de la población en México tiene acceso a internet y que la cantidad de conexiones de dispositivos móviles excede por cuatro por ciento a la población total del país.

Los guitarristas, desde hace ya varios años comparten material musical a través de foros o videos en internet<sup>1</sup>, lo que deja ver que es relativamente común que parte de la formación musical suceda no solo en centros de aprendizaje físicos sino de forma autodidacta a través del uso de la gran variedad de información dispuesta en internet, reforzando la idea de que “las redes sociales han dado lugar a nuevas oportunidades para aprender música. En entornos menos formales, se ha desarrollado un campo completamente nuevo para aprender música.” (Thorgersen y Zandén, 2014, p. 1).

Ante la enorme cantidad de grupos dedicados al estudio de la guitarra y las teorías musicales, sumados a la enorme cantidad de canales de YouTube<sup>2</sup> y la gran cantidad de aplicativos y sitios web dedicados al estudio de las teorías musicales y la guitarra<sup>3</sup>,

---

<sup>1</sup>Por ejemplo, “Guitarristas.info” (<https://www.guitarristas.info/>) desde el 2007 sirve como una comunidad virtual de guitarristas en habla hispana. Proporciona noticias, reseñas, entrevistas, tutoriales, foros de discusión y una amplia variedad de recursos relacionados con la guitarra, además de una sección de anuncios clasificados para comprar y vender instrumentos musicales.

<sup>2</sup>Véase School of Composition (2022). Un compendio de más de sesenta canales en YouTube dedicados al estudio de diversas teorías musicales.

<sup>3</sup>Véase Best-Colleges Team (2022). Un listado de más de cien sitios web con recursos en línea para el aprendizaje de la guitarra en distintos campos y niveles. Estos recursos pueden incluir tutoriales

el desarrollo de un sitio web para el aprendizaje de recursos musicales podría parecer una tarea que difícilmente abonaría algo nuevo al paradigma actual del acceso a materiales y herramientas auxiliares en línea para el estudio y la composición musical, sin embargo, a día de hoy no se ha encontrado algún aplicativo o un sitio web albergue aplicativos que funcionen como herramientas para:

1. Calcular y generar diagramas de acordes.
2. Crear, a partir de una conducción de voces automática, partituras con acordes que sirvan como guías de acompañamiento o puntos de partida para la composición y el desarrollo de estudios.
3. Analizar y visualizar el material musical desde las lógicas de la armonía de jazz y la teoría de conjuntos de tonos clase.

A continuación, se hará una breve revisión de aplicativos web similares que cumplen parcialmente las funcionalidades de los aplicativos desarrollados, agrupándoles en las siguientes tres categorías.

1. **Herramientas para el cálculo de diagramas de acordes:** Aquellas que permiten nominar acordes y calcular diagramas de acordes mediante una selección de valores de notas.
2. **Herramientas para la operación, análisis y visualización de material musical postonal:** Herramientas auxiliares para el desarrollo compositivo, la sistematización y el entendimiento de teorías musicales de orden postonal.
3. **Herramientas para la creación de partituras:** Aquellas que permiten al usuario el desarrollo de partituras con acordes.

Antes de abordar cada categoría, resulta pertinente mencionar que a lo largo de la investigación se entiende por aplicativo web a las “herramientas dinámicas que permiten a los usuarios realizar tareas” (Saia, Nelson, Young, Parham, y Vandegrift, 2022, en video, tablaturas, partituras, ejercicios de técnica, teoría musical y aplicativos auxiliares, entre otros.

p. 1), mediante “servicios interactivos que pueden ser accedidos a través de navegadores de internet” (Shklar y Rosen, 2003, p. 5). No es la finalidad de la investigación ahondar ni problematizar sobre la definición de aplicativo web, se sirve únicamente de la definición recién expuesta con la finalidad de contextualizar el proyecto y poder tener un terreno común de entendimiento para la explicación de las tecnologías desarrolladas. De igual forma, es pertinente mencionar que los casos a continuación expuestos, fueron tomados en cuenta por sus funcionalidades, porque están basados en web, y porque se accede a ellos empleando un navegador, evitando prestar atención a si se describen así mismas como aplicativos, páginas o plataformas web.

## 2.1. Herramientas para calcular y generar diagramas de acordes.

En “Guitarra Postonal en Línea” no existe un componente dedicado a describir el nombre de un acorde desde la armonía tonal, sino uno dedicado a otorgar posibles contextos tonales donde un grupo de notas, entendido desde la teoría de los conjuntos de tono clase, puede operar en función de las notas que le componen. Se ha decidido tomar en cuenta aplicativos web dedicados a nominar acordes a partir y al cálculo de posibles diagramas de acordes a partir de una selección de notas, debido a que son los casos más similares a la usabilidad del “Calculador de Diagramas” implementada y, aunque existe una amplia diversidad ellos, se limitará únicamente a considerar casos que trabajen con las características físicas de la guitarra<sup>4</sup>, dejando de lado aquellos que operan únicamente a un nivel teórico o están situados en otros instrumentos musicales como el ukelele o el piano, por mencionar algunos.<sup>5</sup>

“Scales-Chords”, “Guitar Chords” y “Oolimo”<sup>6</sup> son casos de sitios web que alojan aplicativos que le permiten al usuario calcular diagramas de acordes, escucharlos, nominar escalas y acordes, desarrollar progresiones armónicas, visualizar arpeggios de guitarra en el diapasón e incluso disponer de un metrónomo y un afinador digital. Aunque no se abordarán a detalle todas las funcionalidades de cada uno y se limitará

---

<sup>4</sup>Es decir aquellos casos que permitan el cálculo de acordes o diagramas tomando en cuenta la colocación de las notas en el diapasón u otras características físicas del instrumento.

<sup>5</sup>Por ejemplo: <https://ukebuddy.com/chord-namer> o <https://muted.io/chord-identifier/>.

<sup>6</sup><https://www.scales-chords.com/>, <https://www.all-guitar-chords.com/> y <https://www.oolimo.com/>, respectivamente.

únicamente a hablar de sus componentes dedicados al cálculo de los diagramas de acordes y su nominación, cabe mencionar que ninguno de los casos evalúa el material musical desde la teoría de los conjuntos de tonos clase y únicamente emplean la armonía de jazz para describir la identidad de los acordes.

En la interfaz gráfica de los tres casos se muestra el nombre de un acorde a partir de una selección de notas hecha mediante un diapasón de guitarra interactivo, sin embargo, sus resultados son diferentes debido a la manera en que fueron concebidos y las capacidades de sus algoritmos para trabajar con la disposición de las voces del acorde y casos de acordes poco comunes.

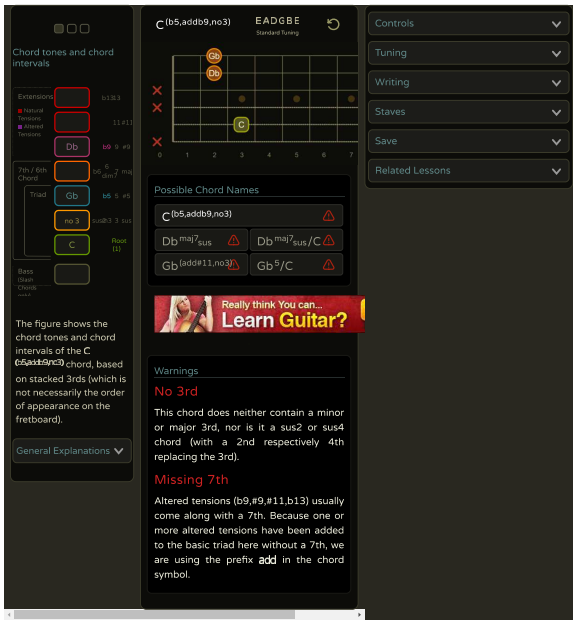
Tanto “Scales-Chords” como “Guitar Chords” trabajan con un listado de casos de cualidades acordes. Evalúan las notas ingresadas en función de esta lista y si ninguno de sus casos coincide con las notas ingresadas, no se muestra ningún resultado y se sugiere agregar otras notas para coincidir con alguna de las cualidades de acordes integradas en su lista. Aunque puedan parecer poco eficiente debido a la poca flexibilidad de sus algoritmos, en realidad el enfoque de trabajo con listas predefinidas es muy común en aplicaciones de teoría musical, e incluso es posible encontrarle en librerías de JavaScript especializadas en teoría musical como “Tonal”o “Music21”.<sup>7</sup>

Un caso contrario sucede en “Oolimo” ya que permite nominar cualquier combinación de notas, evaluándole desde cada uno de sus componentes y desde el acomodo de voces seleccionado al tomar en cuenta la voz más grave como la tónica o inversión del acorde más probable. La manera en que opera su algoritmo permite con amplia flexibilidad añadir especificaciones sobre tensiones o notas omitidas de cualquier cualidad de acorde, tomando en consideración las convenciones de la armonía de jazz para su nominación. La Figura 2.1, muestra como cada uno de los tres aplicativos reacciona a la selección de un acorde compuesto por las notas “do”, “re bemol” y “fa sostenido” [0,1,6].

---

<sup>7</sup><https://github.com/tonaljs/tonal> y <https://web.mit.edu/music21/music21j/doc/index.html>, respectivamente. Véanse los objetos de las librerías dedicados a la nominación de acordes, en ambos casos operan con una lista predefinida de cualidades de acordes.





All chords that have the notes: C C# F# with key note: C

No chord found with those notes  
Similar Chord Suggestions:

Chord	Chord Shape	Chord Sound
<b>C7b5</b> <i>original query without key note restrictions dropping the query note C#</i> C seventh flat fifth Notes: C E Gb Bb (occasionally used) Chord alternate symbols: <a href="#">C7(b5)</a> <a href="#">C7(-5)</a>		
<b>Cm7b5</b> <i>original query without key note restrictions dropping the query note C#</i> C minor seventh flat fifth Notes: C Eb Gb Bb (occasionally used) Chord alternate symbols: <a href="#">C7</a> <a href="#">C1/2dim</a> <a href="#">C1/2dim7</a> <a href="#">Cm7(b5)</a> <a href="#">Cm7(-5)</a>		

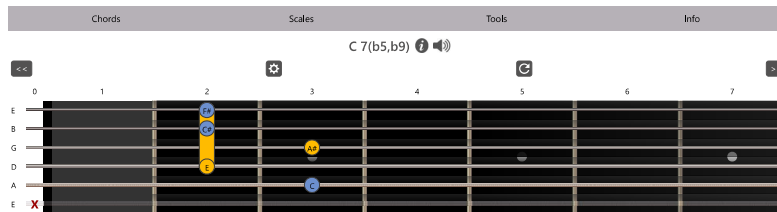
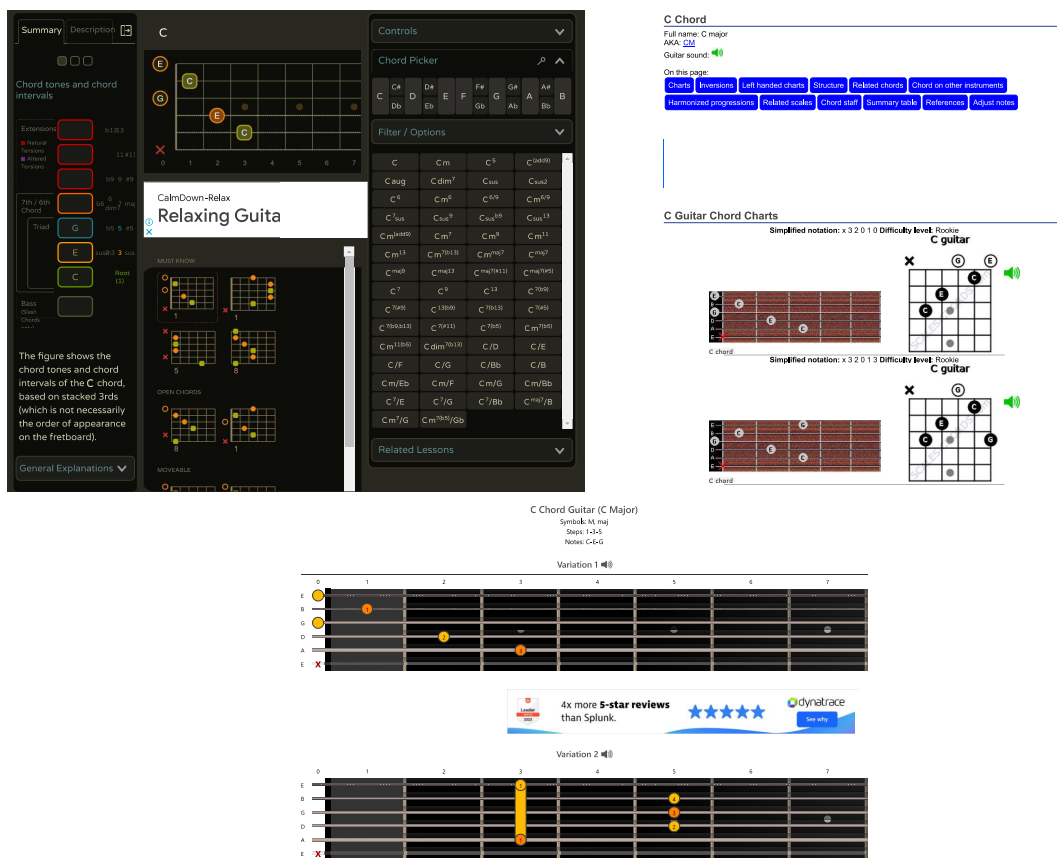


Figura 2.1: Muestra un extracto de los resultados que arrojan “Oolimo” (arriba izquierda), “Scales-Chords” (arriba derecha) y “All Guitar Chords” (abajo) al seleccionar las notas “do”, “re bemol” y “fa sostenido”, [0,1,6]. Nótese como “Scales-Chords” y “All Guitar Chords”, al no encontrar el acorde solicitado en su listado de casos, sugieren el uso de diferentes acordes. Por su parte “Oolimo” muestra seis posibles nombres, arrojando en cada uno de ellos advertencias sobre los elementos importantes del acorde que se omiten para ajustar su nombre de la forma más precisa posible de acuerdo a las notas seleccionadas.

Es importante señalar que en los tres aplicativos mencionados, los algoritmos no toman en cuenta si es posible o no tocar simultáneamente las notas seleccionadas en la guitarra. Sin embargo, estas herramientas pueden seguir siendo útiles para proporcionar posibles nombres de acordes a partir de un conjunto de notas dado, una tarea útil para compositores, arreglistas o músicos que buscan analizar o crear armonías a partir de un conjunto de notas. La presente propuesta profundiza en esta tarea al considerar la nominación de acordes también desde la teoría de los conjuntos tonos clase, lo que permite una clasificación sistemática que permite incluir lógicas pos-tonales en contextos tonales comunes.

En cuanto al cálculo de diagramas de acordes, cada uno de los sitios web ofrece aplicativos dedicadas a esta tarea<sup>8</sup>. Todos ellos utilizan una lista de casos y no tienen algoritmos que puedan calcular automáticamente las diversas formas anatómicamente posibles de tocar un acorde a lo largo del diapasón de la guitarra. La principal diferencia entre estos aplicativos, es la cantidad de casos que ofrecen, tanto de cualidades de acordes como de variaciones en la forma de tocar cada acorde. Por ejemplo, como se puede observar en la Figura 2.2, para un acorde de “Do mayor”, la cantidad de diagramas mostrados varía entre cinco en “Guitar Chord”, nueve en “Scales-Chords” y veinticuatro en “Oolimo”.



A pesar de que los tres aplicativos proporcionan diagramas de acordes, no está claro cuál es la lógica que utilizan para seleccionar y organizar los diferentes diagramas. Por ejemplo, algunos diagramas presentan notas repetidas, mientras que otros no, o algunas cualidades se exponen con diagramas de acordes que abarcan hasta quince trastes de la guitarra, mientras que otras no. En cuanto a la variación de los acordes, aunque “Oolimo” proporciona una taxonomía de diagramas clasificados en diferentes categorías, como “acordes que debes saber”, “acordes abiertos”, “acordes movibles” y “acordes con capotrasto”, al igual que el resto de los aplicativos, tampoco permite modificar las notas de las cualidades listadas.

Cabe mencionar que, aunque no han sido empleadas en los aplicativos recién expuestos, existen librerías en JavaScript dedicadas al dibujo de diagramas de acordes como “VexChords” “Rapahael.chord.js” “ChordJs” o “jTab”<sup>9</sup>, sin embargo ninguna de ellas trabaja con código generativo que evalúe los distintos acomodos de voces posibles, limitándose igualmente al trabajo con un listado predefinido de casos.

Por su parte, una librería que sí permite el reconocimiento y la generación dinámica de cualquier grupo de notas como diagramas de acordes para algún instrumento con trastes, es “Chordictionary”<sup>10</sup>. Sin embargo, a pesar de que permite realizar una búsqueda generativa de posibles diagramas de acordes a lo largo del diapason de la guitarra, como el “Calculador de Diagramas” propuesto, no se decidió incluir como una herramienta dedicada a calcular y generar diagramas debido a que opera únicamente a nivel de código, sin ofrecer ningún tipo de interfaz gráfica que le permita algún tipo de interactividad al usuario, imposibilitando así considerarle como un aplicativo web.

## 2.2. Herramientas para la creación de partituras de acompañamiento.

Existen múltiples sitios web con aplicativos dedicados al desarrollo de cifrados de acordes en línea, algunos permiten la creación de cifrado automático desde videos

---

<sup>9</sup><https://github.com/Oxfe/vexchords>, <https://justindarc.github.io/raphael.chord.js/>, <https://github.com/martijnversluis/ChordJS> y <https://jtab.tardate.com/>, respectivamente.

<sup>10</sup><https://github.com/greird/chordictionaryjs>

en línea<sup>11</sup>, otros crear anotaciones de acordes sobre letras de canciones<sup>12</sup> u otros crear y editar partituras formales de cualquier tipo a nivel multiplataforma mediante dispositivos móviles o cualquier equipo de cómputo.<sup>13</sup>

Con el objetivo de reducir la amplia variedad de opciones disponibles, se ha decidido limitar la búsqueda a aquellos sitios web que ofrezcan herramientas para crear progresiones armónicas con cifrado armónico, lo que implica enfocarse en aplicaciones cuyo resultado final sea una partitura y no una conducción de voces. Esto se debe a que, hasta el momento, no se han encontrado aplicaciones que permitan la generación automática de una conducción de voces a partir de una progresión dada, como lo hace el “Generador de Acompañamientos” de “Guitarra Postonal en Línea”.

“Chord Sheet Maker”, “Flat” y “Soundslice”<sup>14</sup>, aunque se diferencian por sus diferentes funcionalidades, en tanto que han sido concebidos para distintos usos, son casos a considerar debido a que se configuran como aplicativos web que permiten generar partituras de acordes con cifrado armónico.

- “Chord Sheet Maker”, funciona como un analizador sintáctico que mediante texto dispuesto en una gramática particular, puede generar partituras con cifrado de acordes, pero no permite el uso de tablaturas ni partituras ni diagramas de acordes, limitándose únicamente a cifrar cualidades de acordes que tenga listados como posibilidades válidas, dificultando así el cifrado de acordes complejos con notas omitidas o tensiones alteradas.
- “Flat”, permite la creación de partituras colaborativas para varios instrumentos, incluye la posibilidad de usar tablaturas y diagramas de acordes personalizados, aunque de igual forma, limita la nominación de acordes a un listado de casos, dificultando así el cifrado de de acordes complejos.

---

<sup>11</sup>Véase “Chodify” (<https://chordify.net>).

<sup>12</sup>Véase “Songcraft” (<https://songcraft.io>).

<sup>13</sup>Véase “Noteflight” (<https://www.noteflight.com>) o “ScoreCloud” (<https://scorecloud.com>).

<sup>14</sup><https://www.chordsheet.com>, <https://flat.io> y <https://www.soundslice.com>, respectivamente.

- “Soundslice”, opera como una red social para la compartición de interpretaciones musicales sincronizadas con videos, permitiendo a varios usuarios el desarrollo de contenidos multimedia de paga, como cursos o transcripciones de piezas musicales. Se incluye porque permite la creación de partituras con cifrados de acordes con una amplia flexibilidad, en tanto que permite cualquier tipo de texto para nominarles y cualquier tipo de diagrama para representar su digitación en el diapasón de la guitarra.

En lo que respecta de la generación de partituras con acordes, en primer instancia es sencillo discernir que “Chord Sheet Maker” únicamente trabaja con cifrado de acordes, sin partituras, tablatura ni diagramas, elementos que sí es posible emplear en “Flat” y “Soundslice”, aunque este último resulta ser el más flexible dado que una vez escrito un acorde en tablatura, muestra su diagrama correspondiente, sin embargo, de ser deseado así, permite al usuario ediar tanto su digitación como su nombre de la mejor manera que le parezca conveniente, siendo posible tener cualquier tipo de nombre y cualquier tipo de digitación para dicho acorde, independientemente de lo escrito en la partitura.

Por último, es importante mencionar que los tres aplicativos permiten imprimir las partituras (en “Soundslice” solo como usuario premium), descargarlas y tener un usuario para alojar partituras en la nube y a excepción de “Chord Sheet Maker”, que no tiene ningún motor de audio, permiten escuchar e interactuar con las partituras creadas. En el caso de “Guitarra Postonal en Línea” también se permiten las mismas funcionalidades, sin embargo, el poder alojar información en un servidor resulta una implementación a tratar en un futuro en tanto que no existe un servidor ni una infraestructura de usuarios, limitando únicamente al visitante a poder descargar el material generado para su impresión o edición en software de notación musical.

En la Figura 2.3, es posible ver un comparativo de una misma progresión armónica creada usando “Chord Sheet Maker”, “Flat” y “Soundslice”, resaltando elementos importantes en cada aplicativo, como el analizador sintáctico de “Chord Sheet Maker” y la notación personalizada de “Soundslice”.

The image displays three different musical notation styles for a chord progression. At the top left is the 'Chord Sheet Maker' interface, featuring a black text input area with white text: '(E7#5 Gmaj)13 Am13 Gmaj9' and '(E7sus2 Dm6 Cmaj7 F#7#5) Gmaj9'. To its right is the 'Flat' notation, showing a standard chord progression: E7#5 | GMA13 | Am13 | GMA9 | E7sus2 | Dm6 | CMA7 | F#7#5 | GMA9. At the bottom is the 'Soundslice' notation, which includes detailed guitar chord diagrams for each chord, such as (0,2,4,6)0, (0,1,3,5)7, (0,2,3,5)4, (0,1,3,5)6, Esus2#5, Dm6, Cmaj7, F#7#5, and Gmaj9, along with their corresponding musical staves.

Figura 2.3: Muestra un comparativo de una misma progresión armónica creada en “Chord Sheet Maker” (arriba, izquierda), “Flat” (arriba, derecha) y “Soundslice” (abajo). Nótese el recuadro negro con letras blancas de la esquina superior izquierda, ese es el analizador sintáctico de “Chord Sheet Maker” desde donde se escribe la progresión armónica deseada. De igual forma es importante resaltar como, a diferencia del resto de casos, “Soundslice” permite la nominación personalizada de los acordes, véase como se describen los acordes de los primeros cuatro compases de acuerdo a sus formas primas, justo por arriba de los diagramas, cuyas posiciones de dedos, son capaces de modificarse al gusto del usuario, independientemente de lo escrito en la partitura.

## 2.3. Herramientas para la operación, análisis y visualización de material musical postonal

En lo que respecta a herramientas auxiliares para la operación, análisis y visualización

de material musical postonal, se encontró que un gran número de ellas son aplicativos que funcionan para el cálculo de matrices dodecafónicas<sup>15</sup> o para ejercer operaciones de transposición, retrógrado, e inversión de un grupo de notas<sup>16</sup>. A la fecha, no se ha encontrado un solo aplicativo que funcione o trabaje tomando en consideración las características físicas de la guitarra, razón por la cual se plantea que la presente categoría opere a nivel teórico, dejando de lado la consideración de que el material musical postonal de alguna manera tenga que estar implementado en la guitarra. De igual forma, para delimitar la categoría, se tomó en cuenta la posibilidad de considerar aplicativos que trabajen bajo lógicas organizativas de cualquier teorización postonal y no necesariamente la de los conjuntos de tonos clase, logrando integrar así un abanico más amplio que considerara sistematizaciones diferentes a la de Forte. En este sentido, se evaluaron tres casos:

- “Tone Clock”:<sup>17</sup> Un sitio web dedicado a generar patrones visuales análogos al “Zodiaco de Horas” de Schat (1993). El usuario puede vincular distintas notas organizadas bajo un círculo de quintas y hacer su propia serie combinando visualizaciones de acordes y notas que se organizan como polígonos sobrepuestos. Contiene ya cargadas las visualizaciones de las horas de Schat y algunas escalas y modos de uso común. Su funcionamiento se explica en un video de youtube<sup>18</sup> que cubre todas sus funcionalidades, lamentablemente, no cuenta con una reproducción de audio, imposibilitando el poder escuchar las visualizaciones generadas.
- “Amazing Scale Finder”:<sup>19</sup> Un sitio web compuesto por una base de datos de diversas escalas, opera mediante la selección de un conjunto de notas y regresa al usuario páginas estáticas dedicadas a la descripción en profundidad de la escala seleccionada. Una vez dentro de cualquier página de cualquier escala, permite al

---

<sup>15</sup>Una técnica popular en el serialismo, basada en las matrices de invarianza descritas por Alphonse (1974), en sí, “expresado de manera abstracta, las matrices son arreglos bidimensionales que tabulan intervalos ordenados desde elementos en un segmento representado en el eje vertical (X) hasta uno representado en el eje horizontal (Y), donde los intervalos ordenados se obtienen aplicando alguna función que mapea X en Y.”(Morris, 1987, p. 187). Para mayor información sobre las matrices y sus usos en el ámbito dodecafónico, se recomienda la lectura de Morris (2007).

<sup>16</sup>Véase “PC Set Calculator” ([https://www.mta.ca/pc-set/calculator/pc\\_calculate.html](https://www.mta.ca/pc-set/calculator/pc_calculate.html)) o “Matrix Calculator” (<https://www.musictheory.net/calculators/matrix>).

<sup>17</sup><http://toneclock.org>

<sup>18</sup><https://youtu.be/NARPaH83dDc>

<sup>19</sup><https://ianring.com/musictheory/scales/finder.php>

usuario encontrar distintos componentes que arrojan información sobre el grupo seleccionado de notas, como su vector interválico, sus diagramas Tonnetz<sup>20</sup> y escalas vecinas, por ejemplo. Aunque no se describirá a detalle la cantidad de características que muestra el aplicativo de cada agrupación de notas debido a que implica profundizar en elementos que no conciernen a la presente investigación, a manera de síntesis, se puede decir que ejerce cruces de diversas teorías musicales y muestra distintos indicadores de características para el análisis de las notas seleccionadas.

- “Set Theory Relationship”:<sup>21</sup> Es un organizador de material musical postonal que evalúa conjuntos de notas desde la teoría de conjuntos de Forte. Permite generar un listado de hasta diez series o acordes que son evaluados desde su nominación en forma prima y, al ser seleccionados mediante un clic, muestran sus parentescos a las formas primas posibles de agrupaciones de entre tres y seis notas, es decir, muestran qué formas primas contienen o le contienen al conjunto de notas seleccionado. Aunque no permite la escucha de las formas primas, resulta útil para la creación de progresiones armónicas que tome en cuenta conjuntos y subconjuntos de notas para su desarrollo, practica análoga a lo expuesto por Forte (1973) en los “set-complex”.

Cada uno de los aplicativos son muy diferentes entre sí, sin embargo, los tres proponen maneras de organización y visualización de material postonal, permitiendo operaciones de transformación, ya sea desde la edición de las notas seleccionadas, hasta la manera en que el material es visualizado dentro de su interfaz gráfica.

A continuación, las Figuras 2.4, 2.5 y 2.6 muestran extractos de capturas de pantalla los casos recién listados, explicando brevemente en sus pies de fotos sus funcionalidades.

---

<sup>20</sup>El diagrama “...Tonnetz (en alemán, ‘red de tonos’) es un diagrama de reja conceptual que representa el espacio tonal, descrito por primera vez por Leonhard Euler en 1739.” (Euler, s.f., p. 147), se emplea generalmente como un elemento auxiliar para la composición y el estudio musical de acordes y escalas.

<sup>21</sup><http://strcalculator.com/>



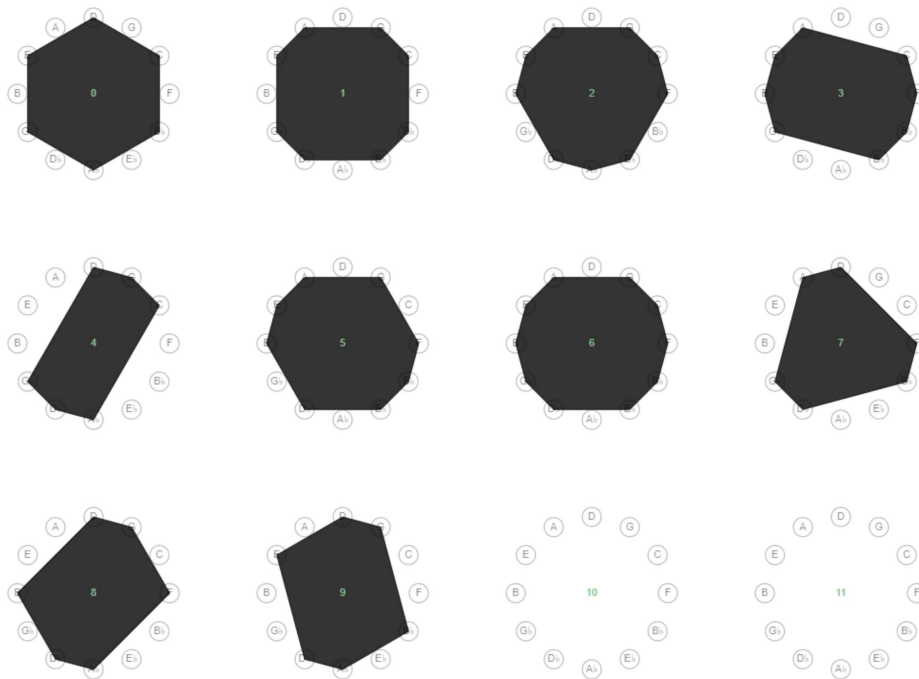


Figura 2.4: Muestra una plantilla realizada con “Tone Clock” compuesta por los polígonos de los modos de transposiciones limitadas de Messiaen (1966). Es uno de los preajustes incluidos dentro de la plataforma. El que viene por default, es un “Zodiaco de Horas” exactamente igual al de la Figura 1.6, expuesta en la sección 1.4.



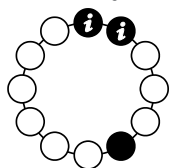
The Exciting Universe Of Music Theory  
presents

more than you ever wanted to know about...

## Scale 35: "Major Seventh Trichord"

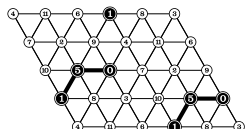


Bracelet Diagram



The bracelet shows tones that are in this scale, starting from the top (12 o'clock), going clockwise in ascending semitones. The "!" icon marks *imperfect* tones that do not have a tone a fifth above. Dotted lines indicate axes of symmetry.

Tonnetz Diagram



Tonnetz diagrams are popular in Neo-Riemannian theory. Notes are arranged in a lattice where perfect 5th intervals are from left to right, major third are northeast, and major 6th intervals are northwest. Other directions are inverse of their opposite. This diagram helps to visualize common triads (they're triangles) and circle-of-fifth relationships (horizontal lines).

### Nearby Scales:

These are other scales that are similar to this one, created by adding a tone, removing a tone, or moving one note up or down a semitone.

Scale <a href="#">31</a>		<a href="#">Honchushi</a>
Scale <a href="#">32</a>		<a href="#">Akhn 3 Tone</a>
Scale <a href="#">39</a>		<a href="#">AFLian</a>
Scale <a href="#">41</a>		<a href="#">AlEian</a>
Scale <a href="#">51</a>		<a href="#">ARFian</a>
Scale <a href="#">3</a>		<a href="#">Minor Second Ditone</a>
Scale <a href="#">19</a>		<a href="#">Hexatonic Trichord</a>

Figura 2.5: Muestra un extracto de las características que arroja "Amazing Scale Finder" una vez algún conjunto de notas es seleccionado. Es posible ver las notas seleccionadas en un diagrama de reloj donde cada tono clase representa una hora y un diagrama de Tonnetz, así como sugerencias de escalas cercanas a la selección de notas que, en este caso, son "do", "re bemol" y "fa". Cabe mencionar que el aplicativo forma parte de un sitio aún mayor llamado "A Study of Scales" que incluye artículos y guías teóricas para el estudio exhaustivo de las escalas musicales.

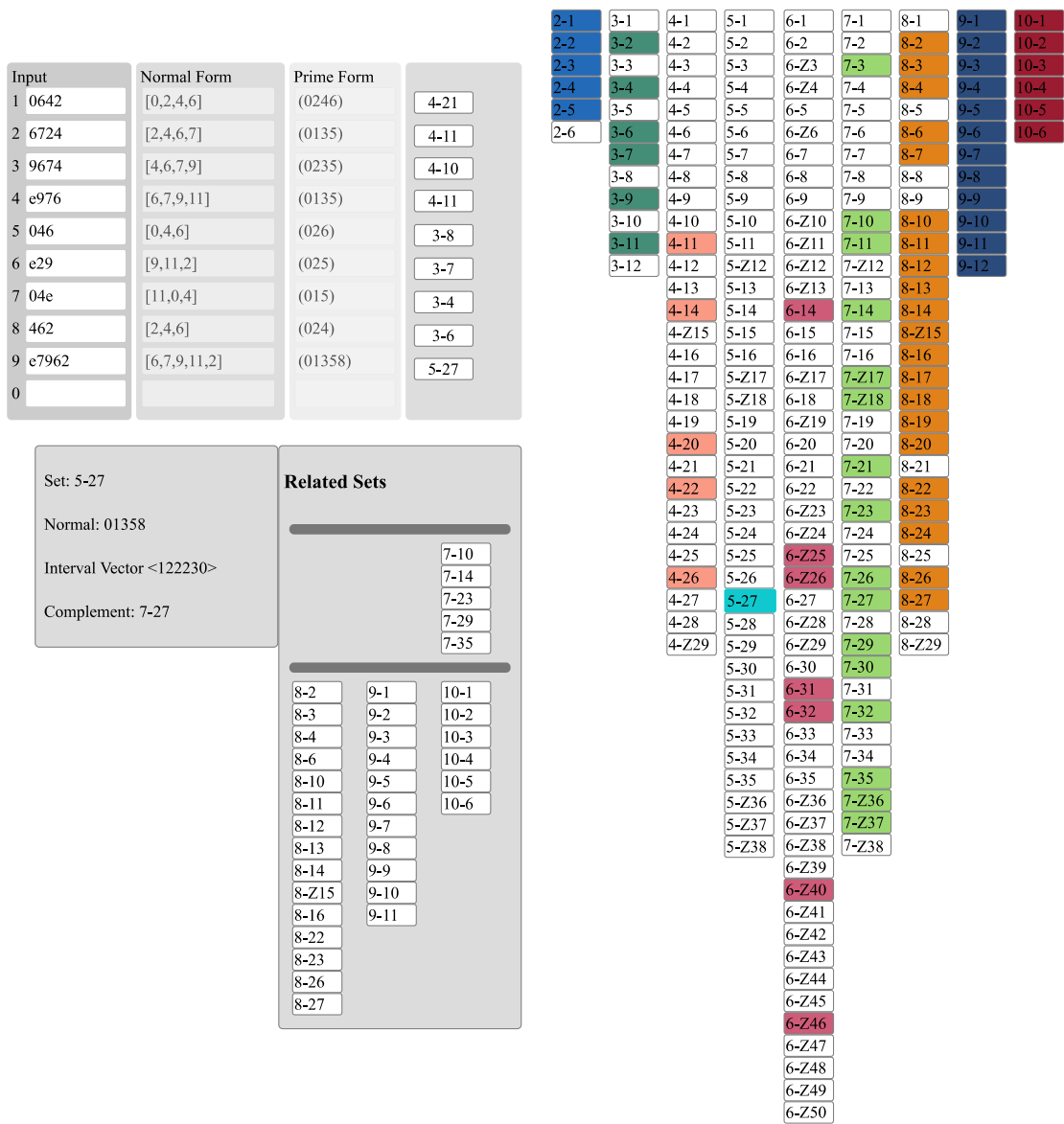


Figura 2.6: Muestra los componentes principales de la interfaz gráfica de “Set Theory Relationship”. En la parte superior izquierda están los conjuntos de notas que he seleccionado para visualizar parentescos (la misma progresión armónica mostrada en la Figura 2.3), y del lado derecho, muestra con color aquellos conjuntos que alberga y albergan el conjunto seleccionado, cuya información (vector interválico, forma normal y complemento) se visualiza en la parte inferior izquierda.

En lo que respecta a qué elementos de cada uno de los aplicativos son similares a los componentes o las lógicas organizativas de “Guitarra Postonal en Línea”, es posible afirmar que:

- De “Tone Clock”, comparte el uso una visualización de polígonos inscritos en un círculo de tonos, esto sucede en el aplicativo dedicado al cálculo de diagramas de acordes. A partir de una selección de notas, muestra el polígono correspondiente a la triangulación de dichas notas.
- De “Amazing Scale Finder”, comparte la evaluación de una estructura de notas desde las lógicas organizativas de más de una teoría musical, en particular, a lo largo de sus tres aplicativos “Guitarra Postonal en Línea” evalúa los grupos de notas desde la teoría de conjuntos de tonos clase y una adaptación de la teorización de Terefenko (2014) que evalúa el uso de tricordes en posibles contextos tonales.
- De “Set Theory Relationship” comparte el interés de ofrecer una representación visual funcional que exponga las relaciones entre las distintas formas primas, esto, plasmado en el “Visualizador de Formas Primas” que muestra un espacio tridimensional donde se inscriben las distintas formas primas de entre tres y seis notas, de acuerdo a características particulares representadas en cada eje.

En el siguiente capítulo, dedicado a la implementación de los aplicativos de “Guitarra Postonal en Línea”, se abordarán a detalle cada uno de los elementos recién mencionados, siendo aquí únicamente referidos para mostrar que las herramientas tomadas en consideración no son recursos inconexos, sino al contrario, abordan inquietudes similares a las de la presente investigación.

# 3 Implementación de Aplicativos y Tecnologías Desarrolladas

Este capítulo es el más importante del documento, en el, se aborda la lógica de los aplicativos implementados, las herramientas empleadas y las tecnologías desarrolladas, procurando hacer una descripción pormenorizada de dichos elementos sin tener que recurrir a la necesidad de mostrar el código creado. Para aquellas personas que deseen consultarlo, pueden hacerlo en el enlace del repositorio antes compartido (<https://github.com/hegelespaul/Postonal-Guitar>), aunado a que cada uno de los aplicativos y tecnologías presentadas, dispone un enlace a su código correspondiente, justo debajo de su título.

Primero, se presenta la conceptualización y diseño de los aplicativos, exponiendo una breve descripción sobre su origen y sus funcionalidades, las tecnologías que les conforman y las herramientas utilizadas para su desarrollo. Posteriormente, se aborda cada aplicativo, exponiendo su interfaz gráfica y sus funcionalidades y, por último, se pormenorizan las tecnologías desarrolladas, exponiendo las lógicas de su programación.

La decisión de organizar el documento de esta forma obedece a que algunas de las tecnologías desarrolladas se encuentran implementadas en más de un aplicativo, por lo que dividir la exposición de la información, primero tratando los aplicativos y luego sus componentes, ayuda a evitar redundancias y a centrar la atención al funcionamiento general de los aplicativos y a las lógicas de programación particulares de cada uno de sus componentes.

## 3.1. Conceptualización y Diseño

“Guitarra Postonal en Línea” es un sitio web compuesto por tres aplicativos donde se implementan cruces de teorizaciones postonales nacidas en el jazz, que atienden la necesidad del desarrollo de recursos musicales para la composición, el estudio y la improvisación en esquemas no tonales.

El primer paso para el desarrollo del sitio fue la creación de un prototipo de código que permitiese el cálculo de posibles diagramas de acorde de acuerdo a una selección de notas.<sup>1</sup> En él, se abordaron diversas tareas, tales como la traducción de las características físicas del diapasón de la guitarra al código, el cálculo de las posiciones de los dedos y la implementación de interactividad. Estas experiencias fueron valiosas para el desarrollo de habilidades de programación y sentaron las bases para la creación de los aplicativos.

Una vez terminado dicho borrador, se pensó que cada acorde seleccionado pudiese evaluarse desde las lógicas nominales de la armonía funcional moderna y desde la teoría de los conjuntos de tonos clase. Igualmente, se pensó que si se había desarrollado un algoritmo que descifraba diagramas de acordes, era conveniente pensar la posibilidad de desarrollar uno dedicado a descifrar conducciones de voces dentro de una progresión armónica.

Por último, al haber logrado dicha tarea, se pensó necesario elaborar un esquema que permitiese observar cómo los posibles acordes se vinculan entre sí. Para ello, se pensó el desarrollo de una representación basada en la similitud y el parentesco de distintos conjuntos de notas.

Estos logros terminaron plasmados en tres aplicativos: un “Calculador de Diagramas de Acordes” que permite calcular diagramas de acordes de guitarra a partir de un conjunto de notas seleccionado, también permite transponer e invertir el conjunto y visualizar posibles organizaciones de su complemento dodecafónico así como un polígono representativo de la interválica de su forma prima; un “Generador de Acom-

---

<sup>1</sup>Se encuentra alojado en el siguiente enlace: <https://github.com/hegelespaul/Postonal-Guitar-P5js-Sketch->. En este borrador, se hace uso de la librería dedicada al desarrollo de código creativo “P5js”, misma que se decidió no emplear en tanto que se encontraron otras herramientas más útiles y especializadas para las tareas a desarrollar.

pañamientos” que habilita al usuario a desarrollar una conducción de voces lógica a partir de una progresión armónica y una disposición de voces seleccionada; y un “Visualizador de Formas Primas” que ofrece una propuesta de visualización funcional interactiva, que permite explorar las relaciones de todas las formas primas posibles de conjuntos de entre tres y seis notas en un espacio tridimensional, configurado tomando en cuenta alguna de las características de las formas primas, como su número de notas o la diversidad de su contenido interválico.

Cada uno de los aplicativos posee componentes específicamente desarrollados para su funcionamiento. A continuación, con el propósito de ilustrar su estructura, la Tabla 3.1, expone las tecnologías que integra cada aplicativo para luego explicar de forma breve en qué consiste cada una de ellas.

<b>Componentes</b>	<b>Calculador de Diagramas</b>	<b>Generador de Acompañamientos</b>	<b>Visualizador de Formas Primas</b>
Intérprete de Contextos Tonales y Formas Primas	<b>X</b>	<b>X</b>	<b>X</b>
Dibujador de Polígonos de Conjuntos	<b>X</b>		
Descifrador de Diagramas	<b>X</b>	<b>X</b>	
Operadores de Transposición, Inversión y Complemento	<b>X</b>		
Partitura Interactiva		<b>X</b>	
Espacio Tridimensional			<b>X</b>
Reproductor de Notas	<b>X</b>	<b>X</b>	

Tabla 3.1: Muestra qué componentes tecnológicos integran cada aplicativo. Nótese como algunas tecnologías forman parte de más de un aplicativo, como es el caso del “Interprete de Contextos Tonales y Formas Primas”, que se halla presente en los tres.

- **Intérprete de Contextos Tonales y Formas Primas:** Se dedica a evaluar la forma prima de las notas seleccionadas y mostrar sus contextos tonales más cercanos.

- **Dibujador de Polígonos de Conjuntos:** Dibuja las formas resultantes al entender las notas seleccionadas como vértices de polígonos inscritos dentro de un reloj dodecafónico.
  
- **Descifrador de Diagramas:** Se dedica a evaluar los posibles diagramas de acordes de un conjunto de entre tres y seis notas.
  
- **Operadores de Transposición, Inversión y Complemento:** Forman parte de un mismo menú y se encargan de llevar a cabo las tareas de transponer e invertir el conjunto de notas seleccionado, así como de mostrar las posibles organizaciones en formas primas de su complemento dodecafónico.
  
- **Descifrador de Acompañamientos:** Calcula las distancias y similitudes entre las digitaciones de las posibles disposiciones de voces de los acordes que conforman una progresión armónica, para proponer una conducción de voces que tenga en cuenta, tanto la proximidad entre las voces superiores, como la similitud entre las digitaciones de sus acordes.
  
- **Partitura Interactiva:** Es la partitura donde queda plasmada la conducción de voces llevada a cabo por el “Descifrador de Acompañamientos”. Permite reproducir, imprimir o descargar la partitura con los acordes generados.
  
- **Espacio Tridimensional:** Proyecta un espacio euclídeo tridimensional interactivo, donde se sitúan las distintas formas primas posibles de calcular en el “Descifrador de Diagramas”. Los ejes de dicho espacio se estructuran de acuerdo a la cantidad de notas, la variabilidad del contenido interválico y el valor máximo de las formas primas.
  
- **Reproductor de Notas:** Es la herramienta dedicada a la reproducción de archivos de audio asociados a las notas musicales de los diagramas, partituras u acordes tratados en los diferentes aplicativos. Las notas son reproducidas de



acuerdo a una indexación cuerda-traste, lo que implica que los audios escuchados corresponden a una digitación específica de las notas en el espacio del diapasón de la guitarra.

Los aplicativos resultan ser el resultado de una interconexión de bloques de código casi modulares<sup>2</sup>, implementados en JavaScript, el lenguaje de programación más comúnmente empleado en el desarrollo web.<sup>3</sup>.

Aunque las lógicas de programación de los aplicativos son el resultado de implementaciones y concepciones de autoría propia, durante su desarrollo se hacen uso de librerías para diversas tareas, como el dibujo de figuras en la pantalla, la creación de partituras o la reproducción de archivos de audio. A continuación, se hace un listado de las librerías empleadas, exponiendo brevemente en qué consisten y para qué se han utilizado.

- **D3js**:<sup>4</sup>. (Data-Driven-Documents), es una librería dedicada a la creación de relaciones entre nodos y objetos de documentos HTML<sup>5</sup>. Funciona mediante la manipulación de los DOMs<sup>6</sup>, lo que le brinda todas las capacidades de los navegadores modernos para el diseño de interfaces visuales que incorporen gráficos y elementos como botones, campos de texto, reproductores de audio, etc.

Con “D3”, se pueden enlazar datos a elementos HTML y crear visualizaciones dinámicas, además de hacer uso de una amplia gama de herramientas como

---

<sup>2</sup>Se dicen casi modulares, en tanto que son bloques desarrollados como tecnologías independientes para lidiar con dificultades específicas, que posteriormente se interconectaron entre sí, sin embargo, sus desarrollos no permiten generar otro tipo de interconexiones fácilmente ni habilitan al usuario el desarrollo de interfaces basadas en una conexión de módulos, como sí lo habilitaría un entorno completamente modular

<sup>3</sup>Para mayor información sobre dicho lenguaje, se remite a la siguiente página web: <https://developer.mozilla.org/es/docs/Web/JavaScript>, en ella, se aborda JavaScript a profundidad, incluyendo materiales de consulta para su aprendizaje

<sup>4</sup><https://d3js.org/>

<sup>5</sup>HTML (HyperText Markup Language) es un lenguaje de marcado utilizado para crear páginas web. Es el lenguaje principal utilizado para definir la estructura y el contenido de un sitio web. Con HTML, se pueden crear y organizar diferentes elementos de una página web, como encabezados, párrafos, listas, imágenes, formularios, enlaces, entre otros

<sup>6</sup>“El Document Object Model (DOM) es una API de programación para documentos HTML y XML. Define la estructura lógica de los documentos y la forma en que se accede y manipula un documento.” Robie y Trexcel Research (1998).

generadores de formas, diagramas de datos, etc. La librería proporciona una amplia variedad de opciones para adaptar los elementos visuales, lo que facilita la creación de visualizaciones altamente personalizadas. Su uso fue fundamental para el desarrollo del aspecto visual y la interactividad del “Calculador de Diagramas” y el “Generador de Acompañamientos”, a excepción de la parte de la partitura interactiva donde se hizo uso de la librería “alphaTab”.

- **alphaTab:**<sup>7</sup> es una librería de JavaScript que permite integrar partituras musicales en aplicaciones web. Permite mostrar partituras en formato de tablatura, así como en notación musical estándar.

Además de incluir características como la posibilidad de reproducir la partitura con sonidos MIDI y una función de desplazamiento automático que permite al usuario seguir la partitura mientras toca su instrumento, “alphaTab”, posee “alphaTex”, un analizador sintáctico, que, mediante una serie de reglas gramaticales permite escribir partituras en muy pocas líneas de código. Dicho componente fue fundamental para poder escribir los acordes arrojados por el “Descifrador de Acompañamientos”.

Es importante destacar que, en cuanto a la reproducción de la partitura, se utilizó exclusivamente el reloj interno de “alphaTab” para sincronizar y reproducir grabaciones de las notas, pertenecientes a un prototipo de Pedroza, Meza, y Roman (2022), una librería de sonidos de guitarra procesados por doce efectos diferentes. Estas grabaciones, proporcionan un mayor realismo que los sonidos MIDI ofrecidos por la librería.

- **Web Audio API:**<sup>8</sup> es una interfaz de programación de aplicaciones (API) que permite crear y manipular audio en tiempo real. Es parte del conjunto de tecnologías web HTML5<sup>9</sup> y está disponible en la mayoría de los navegadores modernos.

---

<sup>7</sup><https://alphatab.net/>.

<sup>8</sup>[https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Audio\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API).

<sup>9</sup>Es la última versión de HTML, diseñada para mejorar la capacidad de la web para ofrecer contenido multimedia sin necesidad de tecnologías de terceros.

Con la “Web Audio API”, es posible crear, manipular y mezclar audio en tiempo real utilizando diferentes fuentes de sonido, efectos, filtros y dispositivos de entrada y salida, razones por las cuales, permite la creación de aplicaciones de audio avanzadas como sintetizadores o secuenciadores de sonido. Es una herramienta sumamente flexible y ha sido utilizada en una amplia variedad de aplicaciones, desde reproductores de música hasta juegos en línea y aplicaciones de música en vivo. Ha probado su suficiencia en distintos escenarios y representa casi un estándar para el manejo de audio en entornos web, por ello, todo aquello que suena en “Guitarra Postonal en Línea” ha sido programado mediante esta API.

- **Plotly:**<sup>10</sup> Es una biblioteca de visualización de datos interactiva, que permite crear gráficos y visualizaciones de alta calidad en diferentes lenguajes de programación, como Python, R o JavaScript. La biblioteca, ofrece una amplia gama de tipos de gráficos, incluidos gráficos interactivos y dinámicos que pueden ser altamente personalizados por los usuarios.

Tanto el aspecto visual, como la interactividad del “Visualizador de Formas Primas”, fueron desarrollados empleando la versión en JavaScript de “Plotly”. Específicamente, se usó su objeto de diagrama de dispersión 3D para configurar la proyección tridimensional del visulizador.

- **AmplifyJS:**<sup>11</sup> es una librería de JavaScript que proporciona una serie de utilidades y componentes que complementan a “jQuery”, una librería que ayuda a escribir código más limpio y conciso, simplificando la manera de interactuar con los documentos HTML, manipular los DOMs, manejar eventos y desarrollar animaciones en las páginas web.

En “Guitarra Postonal en Línea”, únicamente se empleó para el alojamiento de variables globales que pudiesen actualizarse a lo largo de las distintas páginas

---

<sup>10</sup><https://plotly.com/javascript/>.

<sup>11</sup><https://amplifyjs.com/>.

del sitio, si bien dicha funcionalidad puede ser llevada a cabo sin el empleo de “AmplifyJS”, el uso de la librería facilitó su programación, de forma tal, que fue posible ir incluyendo variables sin tener que nombrarlas o evaluarlas de nuevo en los distintos aplicativos, evitando así posibles conflictos en la definición de las variables o sus valores.

En cuanto al aspecto visual del sitio, se ha optado por un estilo minimalista que se enfoca principalmente en el desarrollo de los componentes necesarios para el funcionamiento de las aplicaciones. Se reconoce que tanto el aspecto visual como las interfaces pueden y deben mejorarse en versiones posteriores.

Esta versión actual, se considera un borrador o versión alfa, que priorizó el desarrollo de un prototipo funcional, en lugar de la mejora de aspectos visuales o la experiencia de usuario. Cabe mencionar que el sitio se encuentra implementado en un paradigma de desarrollo web tradicional, empleando archivos HTML, CSS<sup>12</sup> y JavaScript, sin el uso de ningún framework o preprocesador de código para el desarrollo de su aspecto visual o sus interfaces.<sup>13</sup>

Por último, es necesario apuntar que el sitio web es reactivo, es decir, sus elementos se adaptan visualmente a las características de las pantallas desde donde se acceda, facilitando así su navegación desde cualquier dispositivo móvil. En la presente tesina, únicamente se mostrarán las visualizaciones del sitio cuando es accedido a través de un equipo de cómputo, entendiendo el resto de las disposiciones visuales, como pequeñas variaciones de los contenidos aquí presentes.

A continuación, con el fin de ilustrar de manera visual lo descrito hasta este punto, en la Figura 3.1, se incluye una captura de pantalla de los aplicativos. Y posteriormente, en la Figura 3.2, es posible ver un mapa mental que sintetiza los aplicativos y sus componentes, permitiendo una navegación no lineal de los mismos, en tanto que cada nodo enlaza directamente a la sección del documento donde se le trata.

---

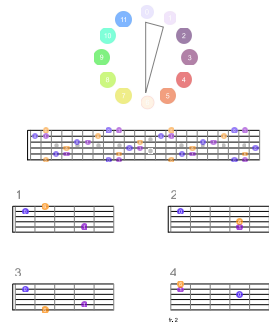
<sup>12</sup>CSS (Cascading Style Sheets), un lenguaje de diseño utilizado para describir la presentación visual de un documento HTML o XML.

<sup>13</sup>Un framework es un conjunto de herramientas, librerías, convenciones y patrones de diseño predefinidos que se utilizan para desarrollar aplicaciones web, móviles o de escritorio. Un preprocesador de código, es una herramienta que extiende la sintaxis básica de un lenguaje de marcado o de programación, agregando características adicionales y facilitando la programación.

### Calculador de Diagramas

3-5 | (0,1,6) 0 | C#mMaj7,F#sus2 Cdim

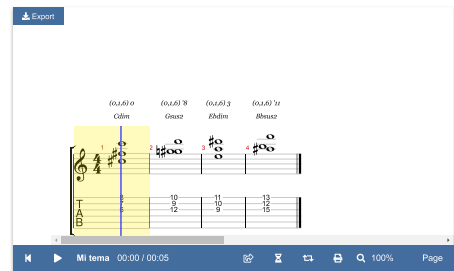
Complemento    Transponer    0    Inversión



### Generador de Acompañamientos

Nuevo Acorde    Borrar Último    Generar

- 3-5 | (0,1,6) 0 | C#mMaj7,F#sus2 Cdim
- 3-5 | (0,1,6) 8 | Ddim,Abmaj7#11 Gsus2
- 3-5 | (0,1,6) 3 | EmMaj7,Asus2 Ebdim
- 3-5 | (0,1,6) 11 | Fdim,Bmaj7#11 Bbsus2



### Visualizador de Formas Primas

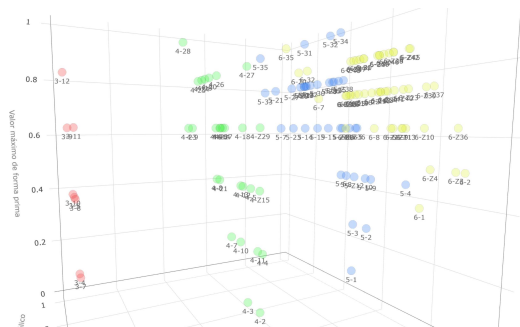


Figura 3.1: Muestra capturas de pantalla de los aplicativos de “Guitarra Postonal en Línea”. En la parte superior, el “Calculador de Diagramas”; al centro, el “Generador de Acompañamientos”; por debajo, el “Visualizador de Formas Primas”.

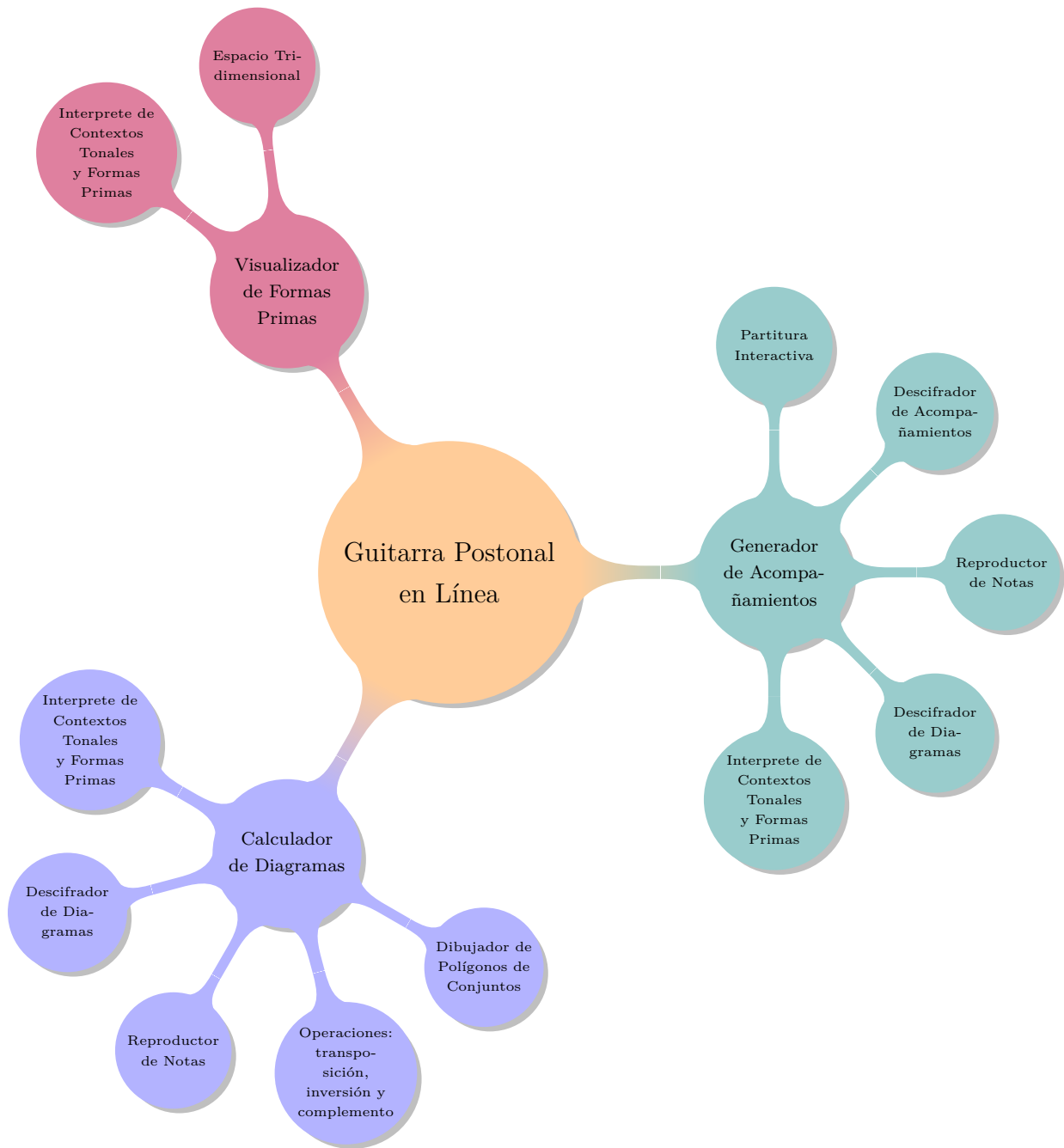


Figura 3.2: Mapa mental que sintetiza la arquitectura de “Guitarra Postonal en Línea”. Al dar clic a cada nodo, es posible ir al espacio del documento donde se le trata, y al dar clic al centro del mapa mental, se navega directamente al sitio web.

## 3.2. Aplicativos Implementados

A continuación, se describen los tres aplicativos, sirviéndose de capturas de pantalla para explicar sus funcionalidades y mostrar sus componentes. No se entrará a detalle sobre las lógicas de su programación, estas se discutirán a profundidad en la sección 3.3 donde se pormenorizan las tecnologías desarrolladas.

### 3.2.1. Calculador de Diagramas

[Enlace al código](#)<sup>14</sup>

Permite la creación de diagramas de acordes a partir de la selección de un conjunto de notas. Su algoritmo retoma los procedimientos descritos por Okazaki (2015) para el cálculo de diagramas de acordes. El aplicativo encuentra a lo largo de los 22 trastes de una guitarra en afinación estándar, todos los posibles diagramas de acordes de hasta seis notas que cumplan las limitantes de que no contengan notas repetidas, cuerdas al aire y una extensión superior a cinco trastes.<sup>15</sup>

Los diagramas de hasta cuatro notas que proporciona el aplicativo son fácilmente interpretables debido a que cumplen con los criterios descritos anteriormente, mismos que se relacionan con la anatomía de la mano que permite la digitación simultánea de cuatro dedos para tocar un acorde (excluyendo el pulgar) y una apertura relativamente cómoda en un espacio de cinco trastes.

Caso contrario, sucede con los diagramas de entre 5 y 6 notas, que operan no tanto como diagramas de acordes para la ejecución de notas simultáneas, sino como visualizaciones de digitaciones del conjunto de notas seleccionado. Aun así, la información dispuesta resulta valiosa en tanto que puede tratarse como elementos escalares para la improvisación, bloques armónicos para la composición o acordes para una ejecución “partida”, es decir, combinando las notas de un solo diagrama en disposiciones de

---

<sup>14</sup><https://github.com/hegelespaul/Postonal-Guitar/blob/master/scripts/drawer.js>

<sup>15</sup>Cabe aclarar que la manera en que fue programado permite, en una etapa posterior, implementar con facilidad afinaciones diferentes y modificar tanto el número de trastes como de cuerdas, permitiendo así adaptar sus procedimientos a instrumentos con características diferentes, como podrían ser un bajo, un banjo, una mandolina, una guitarra barítono, etc.

voces diferentes, evitando tocar todas las notas del diagrama de forma simultánea; una práctica usual en los acompañamientos de guitarra.<sup>16</sup>

“El Calculador de Diagramas”, también permite posibles modificaciones de las notas seleccionadas y los diagramas resultantes, habilitando el transponer e invertir el conjunto de notas seleccionado. Sumado a ello, su interfaz posee de un visualizador del polígono del conjunto de notas (similar al expuesto en el aplicativo “Tone Clock”), y un botón titulado “Complemento”, que da acceso a una página con un listado de las posibles organizaciones en formas primas, de entre tres y seis notas, de su complemento dodecafónico. Estos elementos, han sido integrados pensando en facilitarle al usuario herramientas para la evaluación y modificación del conjunto de notas seleccionado. A continuación, la Figura 3.3, expone una captura de pantalla del aplicativo, señalando cada uno de sus elementos.

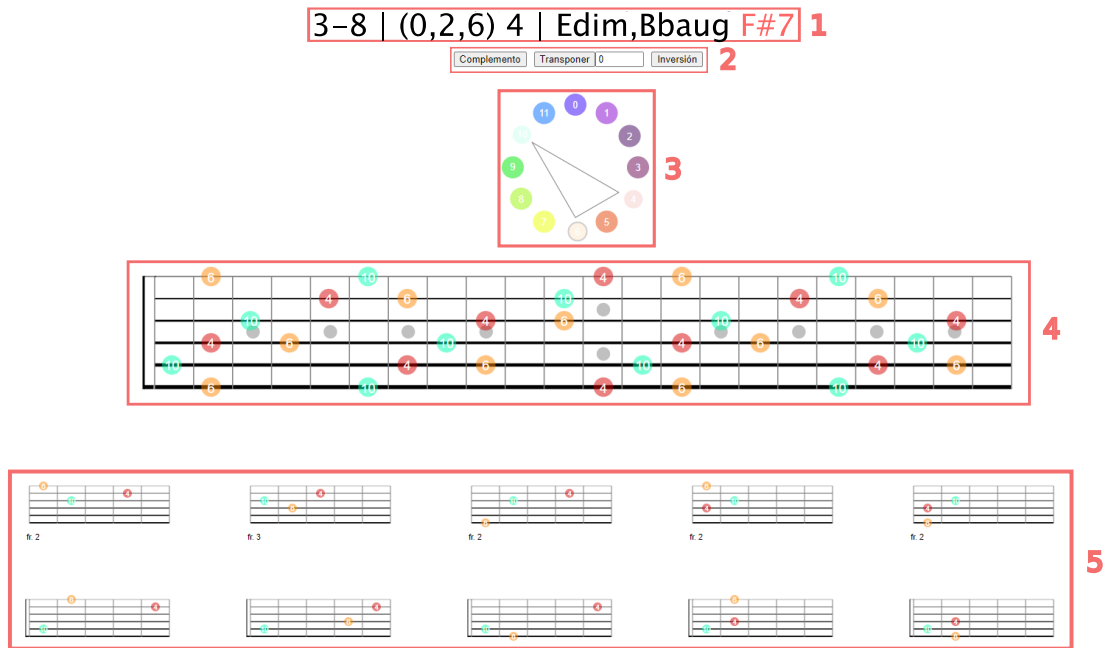


Figura 3.3: Muestra una captura de pantalla del “Calculador de Diagramas”. En ella se numeran las partes que le componen. **1** es el encabezado del calculador, **2** el menú de operaciones, **3** el dibujador de polígonos, **4** un diapasón que muestra cómo se digitan las notas seleccionadas y **5** el resultado visual de los diagramas calculados.

<sup>16</sup>Al respecto, Larsen (2022), muestra ejemplos de acompañamientos que parten de un diagrama de acordes para dividirlo y utilizar sus elementos de forma creativa, evitando su exposición simultánea todo el tiempo.



En el encabezado del calculador (1), se muestra el nombre, la forma prima y los contextos tonales más cercanos al conjunto de notas seleccionado, en este caso 3-8, (0,2,6) 4 y “Edim”, “BbAug” y “F#7” siendo éste último, escrito en rojo, el caso más cercano al conjunto de las notas “mi”, “fa sostenido” y “si bemol”.

Posteriormente, justo por debajo, en el menú de operaciones (2), es posible ver botones dedicados a ejercer tareas de transformación sobre la estructura seleccionada. Las operaciones habilitadas son:

1. **Transposición:** Traslada el conjunto de notas seleccionado a una cantidad específica de semitonos. En primera instancia, es necesario en el campo de número, es necesario seleccionar la cantidad específica de semitonos a la que se transpondrá el conjunto de notas. Es posible seleccionar tanto valores positivos como negativos, esto en función de si el intervalo a transponer es ascendente o descendente. Una vez especificado dicho intervalo, basta con dar clic al botón “Transponer”, para mover la estructura y los diagramas a la distancia del intervalo seleccionado. La Figura 3.4, muestra un ejemplo de esta operación, al exponer un conjunto de notas con forma prima (0,1,6) 0, transpuesto cinco semitonos hacia arriba.
2. **Inversión:** Hace referencia al retrógrado de la forma prima. Se decidió emplear el nombre de inversión y no de retrógrado, para evitar posibles confusiones con la palabra retrógrado, que, empleada como una operación a un conjunto de notas, se refiere únicamente a una lectura de derecha a izquierda de las notas de dicho conjunto. Con la palabra inversión, se pretende dejar en claro que la operación refiere al proceso de “espejo” donde la misma cantidad de intervalos que suceden ascendentemente desde un mismo punto de origen, se invierten en sentido descendente desde su mismo eje. La Figura 3.5, muestra a manera de ejemplo, el polígono de una estructura con forma prima (0,2,5) 6, sometida a este proceso.
3. **Complemento:** Es un botón que, al dar clic, abre una página externa con las posibles combinaciones de formas primas de entre tres y seis notas, del complemento dodecafónico del conjunto de notas seleccionado. Esta información,

se encuentra a manera de un listado de enlaces a los calculadores de diagramas de las formas primas de cada combinación.<sup>17</sup> La Figura 3.6, muestra la página externa del complemento dodecafónico (0,1,3,4,6,9) '11 de la forma prima (0,1,3,4,6,9) 0. Acompañando el texto de la página, se muestra por debajo su interpretación en partitura, ello con la finalidad de ejemplificar de forma más clara, cómo es que suceden las posibles organizaciones del complemento en formas primas de entre tres y seis notas.

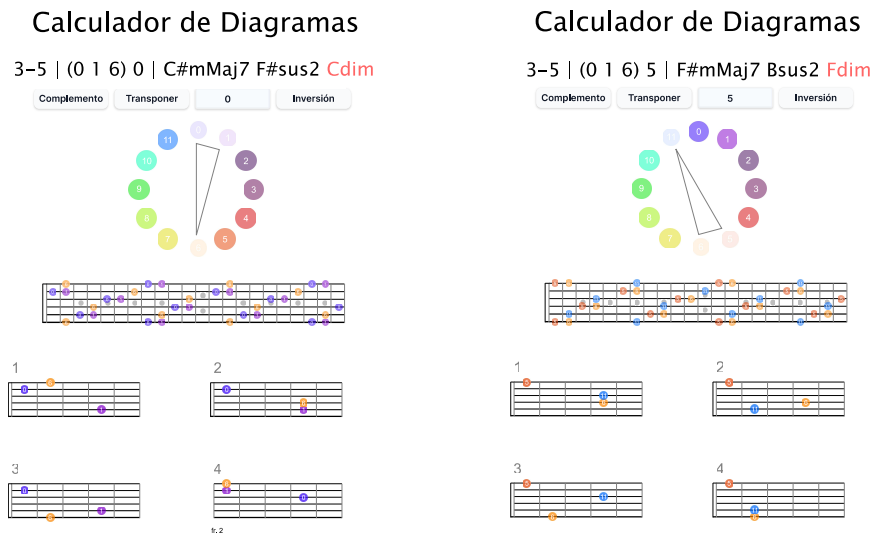


Figura 3.4: Muestra dos capturas de pantalla del “Generador de Diagramas”. La primera (izquierda) muestra los resultados arrojados a partir de una selección de notas [0,1,6]. La segunda, la transposición de dichos resultados 5 semitonos ascendentes, resultando así en la forma prima (0,1,6) 5.

<sup>17</sup>Si bien, el aspecto visual de la información mostrada puede resultar poco amigable, en tanto que se muestra como un acomodo en texto de muchos elementos, esta es una primera aproximación, posteriormente se prevé idear una interfaz visual más fácil de entender y más ilustrativa de cómo se puede organizar el complemento dodecafónico y cómo sus posibles organizaciones se vinculan entre sí.



Figura 3.5: Muestra a la izquierda la forma prima (0,2,5) 6 y a la derecha, la inversión de su interválica o el retrógrado de su forma prima; es decir, los intervalos [0,2,5] leídos de forma descendente a partir de seis. Véase cómo los polígonos que representan los conjuntos se reflejan, haciendo notar el efecto de “espejo” de la inversión.

## (0,1,3,4,6,9) '11

(0,2,5) '7	(0,1,3) '8	(0,1,6) '8	(0,3,6) 2	(0,3,7) '5	(0,2,5) 5	(0,3,7) 7	(0,2,6) 8	(0,2,5) '10	(0,1,3) 7
(0,1,3) '11	(0,1,4) 10	(0,1,6) '11	(0,1,4) '11	(0,1,4) 7	(0,3,6) '2	(0,3,6) '11	(0,2,6) 5	(0,3,7) '2	(0,3,6) 11

(0,1,3,4,6,9) '11

(0,2,5) '7	(0,1,3) '8	(0,1,6) '8	(0,3,6) 2	(0,3,7) '5	(0,2,5) 5	(0,3,7) 7	(0,2,6) 8	(0,2,5) '10	(0,1,3) 7
(0,1,3) '11	(0,1,4) 10	(0,1,6) '11	(0,1,4) '11	(0,1,4) 7	(0,3,6) '2	(0,3,6) '11	(0,2,6) 5	(0,3,7) '2	(0,3,6) 11

Figura 3.6: Muestra una captura de pantalla de la página del complemento de la forma prima (0,1,3,4,6,9) 0 y una representación en partitura de su información. Cada compás representa una posible organización del complemento dodecafónico, siendo la voz superior un conjunto de notas y la voz inferior otro. En este caso, sólo es posible dividir las seis notas resultantes en veinte combinaciones diferentes de dos formas primas de tres notas cada una.

Siguiendo en orden descendente, en **4**, es posible ver un diapasón de guitarra de 22 trastes en afinación estándar. En él, se muestra cómo es posible digitar las notas seleccionadas. Cada círculo contiene el nombre de cada nota y su color que obedece a un indicativo visual para especificar con relativa facilidad los distintos tonos clase. Este código de color es consistente a lo largo de los diferentes aplicativos.

Por último, en **5**, es posible ver una porción de los 91 diagramas generados para (0,2,6) 4. Cada diagrama cuenta con un indicativo de número para su identificación, así como la posibilidad de ser escuchado mediante un clic por parte del usuario. Con la

finalidad de ofrecer una representación sonora lo más precisa posible de cada acorde, los sonidos de los diagramas son generados a partir de las grabaciones de sus notas correspondientes, tocadas en las coordenadas [*traste*, *cuerda*] específicas de cada nota.

### 3.2.2. Generador de Acompañamientos

[Enlace al código](#)<sup>18</sup>

Permite desarrollar una conducción de voces que vincula un acorde a otro de acuerdo con la similitud de sus disposiciones de voces y la cercanía de sus voces superiores.<sup>19</sup>

El generador, funciona añadiendo acordes, seleccionando las notas correspondientes de cada acorde y eligiendo una disposición inicial de voces que servirá como punto de partida para la conducción de la progresión armónica seleccionada. A partir de ahí, evalúa la disposición de voces del primer acorde y le compara con las posibles disposiciones de voces del siguiente, un proceso que se repite en cada acorde hasta finalizar la progresión, misma que termina registrada en una partitura interactiva que se puede escuchar y descargar en formato de archivo “.pdf”y “.gp”.<sup>20</sup>

La partitura, queda registrada con el cifrado de acordes correspondiente a los nombres de los contextos tonales más cercanos y a la forma prima correspondiente de cada conjunto. Si bien, este cifrado resulta eficiente en tanto que permite mostrar la nominación de los acordes de acuerdo a las ópticas de las dos teorías musicales

---

<sup>18</sup><https://github.com/hegelespaul/Postonal-Guitar/blob/master/scripts/compjen.js>

<sup>19</sup>Willmott (1994), dedica una publicación entera a abordar este enfoque. Expone la conducción de voces entre acordes de acuerdo a la melodía generada por sus voces superiores. Expone diversos procedimientos, entre ellos, uno enfocado en mantener el mínimo movimiento posible en la voz superior a lo largo de una progresión armónica, utilizando una misma disposición de voces en todos los acordes.

<sup>20</sup>Un formato popular entre guitarristas, corresponde al software privativo “Guitar Pro”. En sí, es una carpeta comprimida en “.zip”con un archivo XML que con una cuidada edición, podría ser leído por distintos software de notación musical. Se exporta en formato “.gp” debido a que “alphaTab” aún no permite la exportación de las partituras en XML, sin embargo, se tiene contemplado que en aproximaciones futuras el archivo a descargar no necesite el uso de un software privativo para su lectura y edición. Si se necesita editar el archivo descargado y no se cuenta con alguna licencia de “Guitar Pro”, se recomienda acudir a “Soundslice”para cargar el archivo “.gp”y posteriormente exportarlo como “.xml”. Si bien, software no privativo como “Muse Score” puede leer archivos “.gp”, en su lectura a menudo deja de lado la carga de la información en tablatura, dificultando así entender la digitación de los acordes.

(la teoría de conjuntos de tonos clase de Forte y la armonía de jazz), en un futuro se tiene contemplado habilitar un botón que permita al usuario escoger el tipo de notación que desea para los acordes, ya sea a manera de forma prima o como cifrado convencional de acordes, permitiéndole elegir desde cuál de las notas del acorde sucede su nominación, ello con ánimos de dotar de flexibilidad a la herramienta y a que el usuario pueda plantear lógicas nominales altamente específicas.

A continuación, la Figura 3.7, expone una captura de pantalla del aplicativo, señalando cada uno de sus componentes.

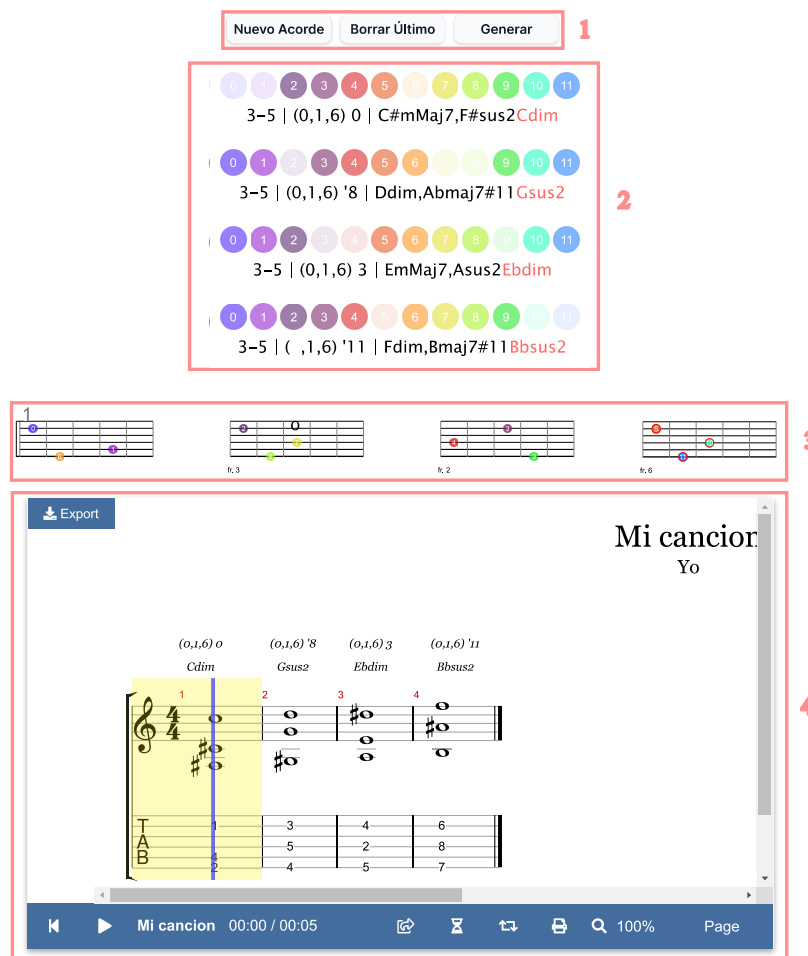


Figura 3.7: Muestra una captura de pantalla del “Generador de Acompañamientos”. En 1 es posible ver el menú principal desde donde se crean los acordes y la conducción de voces, en 2 se muestran los acordes de la progresión armónica, en 3 los diagramas de acordes resultantes de la conducción de voces y en 4 la partitura interactiva donde queda registrada la conducción de voces generada.

En la parte superior, **1**, se encuentra el menú principal diseñado para crear acordes, borrar el último de los acordes y generar la conducción de voces. Por debajo, en **2**, se muestran los acordes seleccionados<sup>21</sup>.

Una vez que se haya dado clic al botón “Generar” y se haya cumplido la condición de que cada uno de los acordes tengan por lo menos tres notas seleccionadas, el aplicativo provee un listado de posibles puntos de partida para el desarrollo de la conducción de voces, como el mostrado en la Figura 3.8. Una vez seleccionada la disposición donde se desea comenzar la conducción, el aplicativo muestra los acordes resultantes, tanto en el formato de diagramas (**3**), como dentro de una partitura interactiva (**4**).

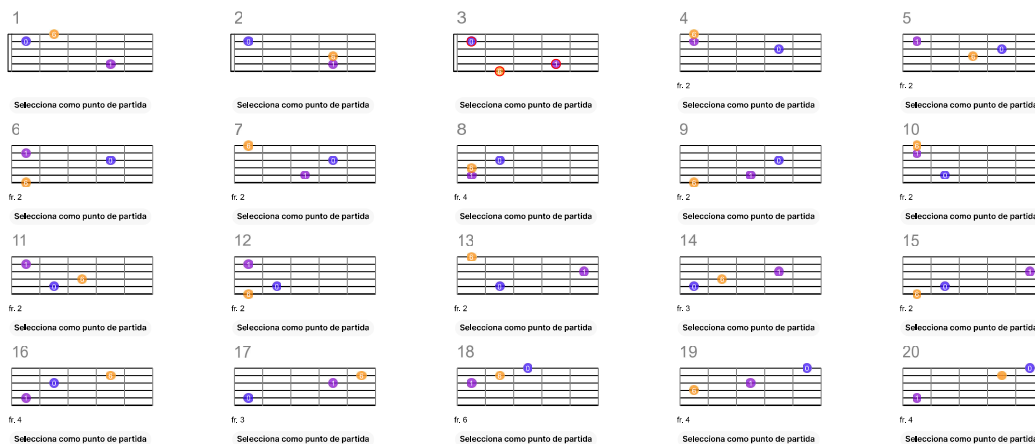


Figura 3.8: Muestra una captura de pantalla del listado de posibles puntos de partida para el desarrollo de una conducción de voces dentro del “Generador de Acompañamientos”. Cada uno de los diagramas mostrados, puede ser escuchado mediante un clic.

### 3.2.3. Visualizador de Formas Primas

[Enlace al código](#)<sup>22</sup>

<sup>21</sup>En un principio, el usuario sólo ve los círculos con los valores de notas, y una vez ha seleccionado más de tres notas, se muestra el nombre de la forma prima y el contexto tonal más cercano al conjunto de notas seleccionadas.

<sup>22</sup><https://github.com/hegelespaul/Postonal-Guitar/blob/master/scripts/analyser.js>

El visualizador, consiste en una proyección tridimensional interactiva donde se inscriben las posibles formas primas de entre tres y seis notas.<sup>23</sup> El eje “X” representa el número de notas, “Y” la desviación estándar de su vector interválico y “Z” la suma de sus intervalos en forma prima. Esta concepción de los ejes, permite mostrar una proyección organizada, que muestra cómo los distintos conjuntos de notas se relacionan entre sí y que tan similares o disímiles son unos de otros. Esta proyección se sugiere como una propuesta visual útil para organizar las formas primas de acuerdo a sus características particulares.

En cuanto a sus funcionalidades, el visualizador permite, al posicionar el ratón por encima de cada punto, observar, entendidos desde cero (“do”), la forma prima y los contextos tonales del conjunto de notas que representa. Aunado a ello, al dar clic a cada punto, se puede acceder al “Calculador de Diagramas” del conjunto de notas representado. A continuación, se muestra una captura de pantalla del “Visualizador de Formas Primas”, donde es posible apreciar los elementos que le componen.

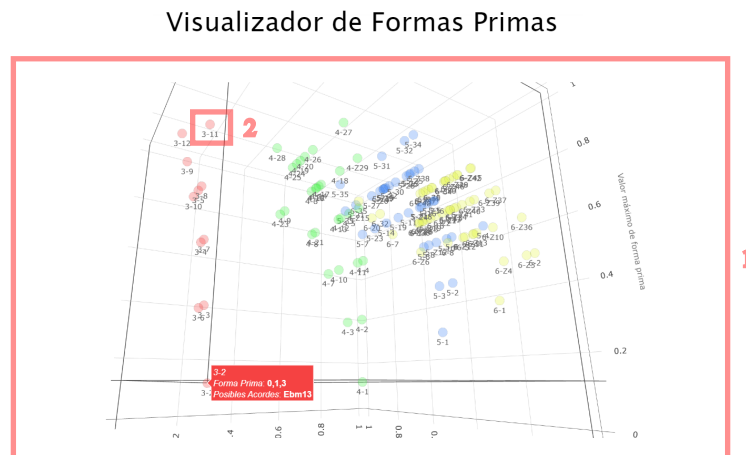


Figura 3.9: Muestra una captura de pantalla del “Visualizador de Formas Primas”. En **1** se muestra el espacio tridimensional interactivo y en **2**, la información mostrada cuando se posiciona el cursor sobre cualquiera de los puntos.

<sup>23</sup>En esta primera aproximación se muestran únicamente las formas primas posibles de conjuntos de entre tres y seis notas, en un futuro se prevé poder mostrar, cómo se inscriben dentro de este espacio, las formas primas de los acordes seleccionados en el “Generador de Acompañamientos”, a la par de habilitar posibles salidas visuales que orienten de mejor manera cómo es que las formas primas se vinculan entre sí, ya sea generando trayectos entre punto y punto o permitiendo generar triangulaciones entre más de dos puntos.

Al dar clic en cualquier espacio vacío y arrastrar el ratón, es posible navegar el espacio tridimensional interactivo (1), modificando el punto de vista desde donde se observa la visualización. A su vez, es posible acercarse o alejarse a los distintos puntos, mediante la gestualidad de zoom correspondiente a la interfaz desde donde se acceda, ya sea un “mousewheel” o un “trackpad”.

Hasta este punto, se han descrito las funcionalidades y diseños de los tres aplicativos. Antes de continuar con la siguiente sección del documento, se recomienda que se acceda a los aplicativos<sup>24</sup> e interactúe con ellos para experimentarlos de primera mano y comprender mejor su funcionamiento.

En ese sentido, de igual manera se hace una invitación a revisar la siguiente playlist de videos en donde se discuten y muestran los aplicativos (<https://youtube.com/playlist?list=PLpLKNYJFckJVeMiTJbDCmHhJoLDHV7HB0>). Los videos exponen sus funcionalidades, de forma breve, clara y amable para cualquier persona, sin ahondar en detalles sobre la lógica de su programación, sirviendo así como una guía introductoria que permite entender las funcionalidades y alcances de cada uno.

---

<sup>24</sup>Se pueden acceder a través de este enlace: <https://guitarrapostonalonlinea.netlify.app/>



### 3.3. Tecnologías Desarrolladas

A continuación, se abordarán a detalle el funcionamiento de las tecnologías desarrolladas. Se explicará el orden lógico de su programación, dotando de los elementos necesarios para entenderle. No se muestra en ningún momento el código generado, comprendiendo que la persona que desee verlo, lo puede hacer a través del repositorio de github del sitio web (<https://github.com/hegelespaul/Postonal-Guitar>) o mediante los enlaces dispuestos a lo largo del documento.

Es importante mencionar que el llamar “tecnologías” a los códigos generados, puede ocasionar cierta confusión, ya que comúnmente las funciones de código no se consideran tecnologías en sí mismas, sino componentes de una solución tecnológica más amplia. No obstante, considerarlas como tecnologías se debe a que se entienden como “[...]una aplicación sistemática del conocimiento científico (u otro conocimiento organizado)[...]” (Galbraith, 1984, p. 59-60) “[...]que permite describir, explicar, diseñar y aplicar soluciones técnicas de forma sistemática y racional” (Quintanilla, 1989, p. 153). Aunque este enfoque clásico de tecnología pertenece al campo de las ciencias políticas y sociales, puede ser aplicable a las funciones de código en tanto que utilizan conocimiento organizado para resolver problemas prácticos de manera racional.

Sin más, aclarado ese punto, se procede a detallar cada una de las tecnologías.

#### 3.3.1. Interprete de Contextos Tonales y Formas Primas

[Enlace al código](#)<sup>25</sup>

Se dedica a evaluar la forma prima de las notas seleccionadas y mostrar sus contextos tonales más cercanos. Al recibir un conjunto de notas, realiza dos tareas principales: la primera, consiste en calcular la forma prima del conjunto, y la segunda, implica evaluar la similitud de sus notas con relación a un listado de acordes (Figura 3.2), entendidos como contextos tonales comúnmente empleados.

---

<sup>25</sup><https://github.com/hegelespaul/Postonal-Guitar/blob/master/scripts/decifrador.js>

Acorde	Intervalos	Acorde	Intervalos
maj	[0, 4, 7]	add9	[0,2,4,7]
min	[0, 3, 7]	madd9	[0,2,3,7]
dim	[0, 3, 6]	add11	[0, 4, 5, 7]
aug	[0, 4, 8]	madd11	[0, 3, 5, 7]
sus2	[0, 2, 7]	6/9	[0, 2, 4, 9]
sus4	[0, 5, 7]	m6/9	[0, 2, 3, 4, 9]
6	[0, 4, 7, 9]	maj9	[0, 2, 4, 7, 11]
m6	[0, 3, 7, 9]	m9	[0, 2, 3, 7, 10]
maj7	[0, 4, 7, 11]	9	[0, 2, 4, 7, 10]
m7	[0, 3, 7, 10]	maj7#11	[0, 4, 6, 7, 11]
m7b5	[0, 3, 6, 10]	m11	[0, 3, 5, 7, 10]
mMaj7	[0, 3, 7, 11]	7#11	[0, 4, 6, 7, 10]
7	[0, 4, 7, 10]	7b13	[0, 4, 7, 8, 10]
7sus4	[0, 5, 7, 10]	maj13	[0, 4, 7, 9, 11]
7sus2	[0, 2, 7, 10]	m13	[0, 3, 7, 9, 10]
dim7	[0, 3, 6, 9]	13	[0, 4, 7, 9, 10]

Tabla 3.2: Muestra el listado de los contextos tonales considerados. Incluye acordes de hasta cinco notas de uso común. Se estructuró buscando plantear contextos tonales generales, para un posterior desarrollo añadir especificidad a la relación de estos con el conjunto de notas seleccionado y elaborar así un traductor literal del conjunto de notas a un cifrado de acorde específico.

Para llevar a cabo la primera tarea, se replica el procedimiento para el cálculo de formas primas de Forte expuesto en la Figura 1.1, haciendo a una transducción de sus procedimientos a una lógica de programación que consiste en los siguientes pasos:

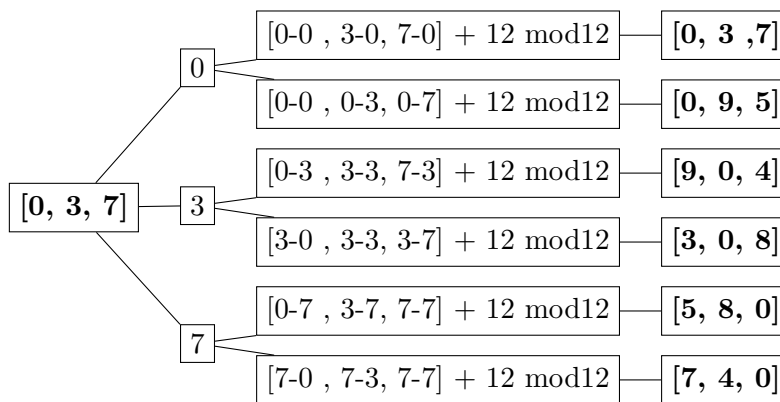
1. Se calcula, tanto en orden ascendente como en orden descendente, la distancia en semitonos de cada nota respecto a sí misma y al resto de notas del conjunto. Esto se logra al restar cada valor de nota a los valores del conjunto y viceversa, sumando doce y aplicando módulo doce<sup>26</sup> a cada valor obtenido, en vías de obtener solo valores positivos del cero al once que definan los intervalos a recorrer desde la nota de partida evaluada.

---

<sup>26</sup>El módulo doce es una operación matemática que consiste en tomar el resto de la división de un número entre doce. En la teoría musical, se utiliza para reducir los valores de las alturas de las notas a un rango de 0 a 11, el efecto del módulo facilita que cualquier valor positivo que sobrepase doce, vuelva a generar un recorrido de cero a doce, por ejemplo, el módulo doce de quince es tres y el de doce, cero. Para una explicación más detallada, se sugiere la consulta del primer capítulo de Straus (2005), en el, es posible encontrar varios ejemplos que abordan detalladamente este procedimiento.

2. Se ordenan ascendentemente los resultados, para sintetizar los distintos recorridos de intervalos, tomando como punto de partida cada nota del conjunto.
  
3. Se crea un listado con los conjuntos que tengan la menor distancia entre su primer y última nota, para posteriormente filtrarlos y determinar su forma prima al escoger el conjunto de dicho listado cuyo resultado de la suma de sus valores sea el menor, en tanto que la configuración de las formas primas, privilegia la permutación del conjunto de notas con los intervalos internos y externos más reducidos.

A continuación, la Figura 3.10, ilustra el proceso recién descrito aplicado a un conjunto de notas  $[0,3,7]$ .



<b>0</b>	<b>'0</b>	<b>3</b>	<b>'3</b>	<b>7</b>	<b>'7</b>
<b>[0 3 7]</b>	[0 5 9]	[0 4 9]	[0 3 8]	[0 5 8]	[0 4 7]

Figura 3.10: Ilustra el procedimiento implementado para el cálculo de la forma prima. Se muestra cómo cada nota del conjunto es restada a cada uno de los elementos del conjunto y viceversa, aplicando una suma de doce y un posterior módulo doce para la obtención de los intervalos como valores positivos del cero al once. Posteriormente, se disponen ascendentemente los elementos de cada conjunto para evaluar aquel que tenga la estructura mínima tanto en sus componentes internos como externos. En la figura dicho conjunto se resalta con color rojo y la columna donde pertenece, especifica el punto de partida y la dirección del trayecto de dichos intervalos, logrando interpretar así que la forma prima de un conjunto de notas  $[0,3,7]$  es **(0,3,7) 0**.

Es importante destacar que en algunos casos de formas primas con estructuras simétricas o palindrómicas, puede haber más de una definición válida de forma prima. Para estos casos, se ha decidido tomar en cuenta el orden en que aparece la primera forma prima posible. Esto se debe a que el algoritmo desarrollado aún no ha sido adaptado para reconocer conjuntos de notas con más de una forma prima válida posible, sin embargo, se contempla que en futuras mejoras se puedan reconocer todas las formas primas posibles para cada conjunto de notas.

En lo que respecta al cálculo de los contextos tonales más cercanos. El conjunto de notas seleccionado es sometido al primer punto del proceso recién listado, solo que no se calculan sus retrógrados, es decir, las restas de los elementos del conjunto al valor de nota. Esto, debido a que los casos de contextos tonales son comprendidos a partir de la exposición ascendente de los intervalos que les integran, siendo suficiente el ordenamiento ascendente de los elementos resultantes de la resta del conjunto al valor de nota, para obtener las versiones del conjunto que se comprenden como acordes ascendentes empezando desde cada una de sus notas (Figura 3.11).

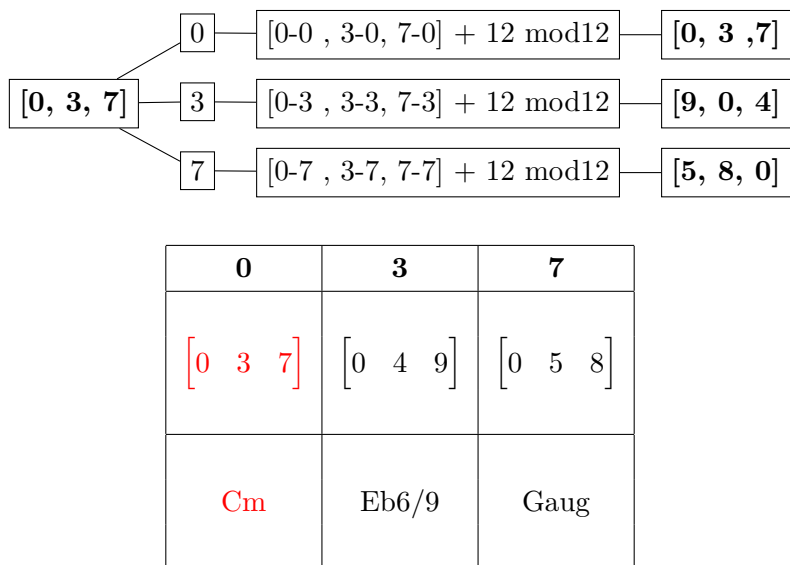


Figura 3.11: Muestra cómo se interpretan los intervalos desde cada nota para su posterior comparación con el listado de posibles contextos tonales. La columna de la tabla especifica desde que nota parte dicha intervállica. En este caso, la intervállica marcada con rojo, especifica que dicha versión del conjunto es la de mayor similitud coseno, siendo totalmente coincidente con el caso de contexto tonal “min”, que especifica la cualidad de un acorde menor, y como dicha versión fue evaluada desde cero, el contexto tonal más cercano resultante se comprende como “do menor”.

Para evaluar la cercanía de los conjuntos de notas con los casos de contextos tonales, se hace uso de la similitud coseno (Ecuación 3.1, una medida de la similitud existente entre dos vectores en un espacio de n dimensiones. Es una métrica comúnmente usada en la extracción y comparación de documentos de texto, “el ángulo entre dos vectores se usa como una medida de divergencia entre los vectores, y el coseno del ángulo se usa como su similitud numérica (ya que el coseno tiene la buena propiedad de que es 1,0 para vectores idénticos y 0,0 para vectores ortogonales).” Singhal (2001).

$$\cos(\mathbf{t}, \mathbf{e}) = \frac{\mathbf{t} \cdot \mathbf{e}}{\|\mathbf{t}\| \|\mathbf{e}\|} = \frac{\sum_{i=1}^n t_i e_i}{\sqrt{\sum_{i=1}^n (t_i)^2} \sqrt{\sum_{i=1}^n (e_i)^2}} \quad (3.1)$$

3.1: Fórmula matemática para el cálculo de la similitud coseno entre dos vectores en un espacio n-dimensional. La similitud coseno mide el grado de similitud entre dos vectores y se calcula dividiendo el producto punto de los vectores entre el producto de sus normas L-2. La norma L-2 es la raíz cuadrada de la suma de los valores absolutos de los cuadrados de los componentes de los vectores. En resumen, esta fórmula determina cuánto se parecen dos vectores en dirección y magnitud.

Para ilustrar de forma más clara<sup>27</sup>, sírvase de la Figura 3.12. En ella, se expone una representación visual de cómo opera el cálculo del ángulo entre dos vectores para la comparación de documentos. En el ejemplo, se comparan dos documentos, cada uno con un vector de dos dimensiones donde se hace un registro de cuántas veces aparecen dentro de cada documento las palabras “Hola”y “Mundo”.

---

<sup>27</sup>De existir más dudas se sugiere consultar los siguiente videos tutoriales que exponen con detenimiento la similitud coseno, sus posibles usos y el procedimiento para su cálculo: <https://www.youtube.com/watch?v=Ze6A08Pw5oE> y <https://www.youtube.com/watch?v=e9U0QAFbfLI&t=333s>.

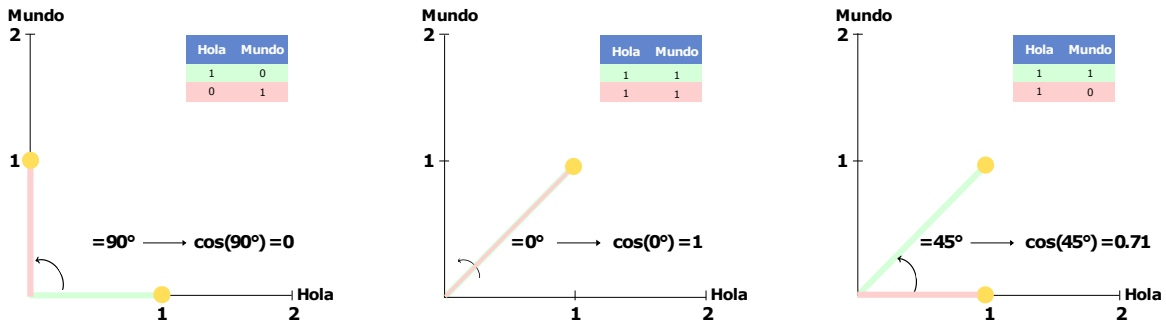


Figura 3.12: Muestra tres casos de cálculos de similitud coseno. El primero, un caso con vectores ortogonales (izquierda), otro con vectores idénticos (centro) y otro, con vectores que forman un ángulo de 45 grados, correspondiente a una similitud coseno de 0.71 (derecha).

En el caso de la comparación de la similitud de los conjuntos de notas con los casos de contextos tonales, se hace uso de una comprensión de su información en un vector de doce dimensiones, donde cada una de ellas, corresponde a los posibles intervalos que les componen.<sup>28</sup> Para evitar confusiones entre el valor cero, que es el que representa la nota como punto de partida desde donde se evalúan los intervalos, los elementos que no forman parte ni de los casos de acordes ni del conjunto de notas seleccionado, se representa con el valor de -1. Para ilustrar este proceso, la Figura 3.13, muestra el procedimiento del cálculo de la similitud coseno de un conjunto de notas con inter-vállica  $[0,1,6]$ , con respecto a un posible caso de contexto tonal de acorde dominante  $[0,4,7,10]$  y otro de acorde disminuido  $[0,3,6]$ .

<sup>28</sup>Para facilitar su cálculo, los intervalos de los conjuntos se acotan dentro de una octava, incluidas las tensiones, mismas que se transponen una octava por debajo para incorporarse dentro del espectro planteado.

$A = [0,1,6]$	$A = [0,1,-1,-1,-1,-1,6,-1,-1,-1,-1,-1]$	$\cos(A,B) = \frac{A \cdot B}{\ A\  \ B\ }$
$B = [0,4,7,10]$	$B = [0,-1,-1,-1,4,-1,-1,7,-1,-1,10,-1]$	
$A \cdot B = ((0 \times 0) + (1 \times 1) + (-1 \times -1) + (-1 \times -1) + (-1 \times -1) + (-1 \times 4) + (-1 \times -1) + (6 \times -1) + (-1 \times 7) + (-1 \times -1) + (-1 \times -1) + (-1 \times 10) + (-1 \times -1))$ $A \cdot B = (0 + (-1) + 1 + 1 + (-4) + 1 + (-6) + (-7) + 1 + 1 + (-10) + 1)$ $A \cdot B = -22$		$\cos(A,B) = \frac{-22}{(6.78)(13.15)}$
$\ A\  = ((0^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 6^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2)^{1/2})$ $\ A\  = \sqrt{46}$ $\ A\  = 6.78$		$\cos(A,B) = \frac{-22}{89.157}$
$\ B\  = ((0^2 + 1^2 + 1^2 + 1^2 + 4^2 + 1^2 + 1^2 + 7^2 + 1^2 + 1^2 + 10^2 + 1^2)^{1/2})$ $\ B\  = \sqrt{173}$ $\ B\  = 13.15$		<b><math>\cos(A,B) = -0.24</math></b>

$A = [0,1,6]$	$A = [0,1,-1,-1,-1,-1,6,-1,-1,-1,-1,-1]$	$\cos(A,B) = \frac{A \cdot B}{\ A\  \ B\ }$
$B = [0,3,6]$	$B = [0,-1,-1,3,-1,-1,6,-1,-1,-1,-1,-1]$	
$A \cdot B = ((0 \times 0) + (1 \times 1) + (-1 \times -1) + (-1 \times 3) + (-1 \times -1) + (-1 \times -1) + (6 \times 6) + (-1 \times -1) + (-1 \times -1) + (-1 \times -1) + (-1 \times -1) + (-1 \times -1))$ $A \cdot B = (0 + (-1) + 1 + (-3) + 1 + 1 + 36 + 1 + 1 + 1 + 1 + 1)$ $A \cdot B = 40$		$\cos(A,B) = \frac{40}{(6.78)(7.34)}$
$\ A\  = ((0^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 6^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2)^{1/2})$ $\ A\  = \sqrt{46}$ $\ A\  = 6.78$		$\cos(A,B) = \frac{40}{49.76}$
$\ B\  = ((0^2 + 1^2 + 1^2 + 3^2 + 1^2 + 1^2 + 6^2 + 1^2 + 1^2 + 1^2 + 1^2 + 1^2)^{1/2})$ $\ B\  = \sqrt{54}$ $\ B\  = 7.34$		<b><math>\cos(A,B) = 0.80</math></b>

Figura 3.13: Muestra el procedimiento del cálculo de la similitud coseno de un conjunto de notas [0,1,6], con respecto a un caso de posible contexto tonal de acorde de séptima dominante (arriba) y otro de acorde disminuido (abajo). Nótese cómo cada caso se define dentro de un vector de doce dimensiones y cómo las similitudes coseno entre los casos son muy diferentes. En el primer caso, se obtiene un resultado negativo, lo que indica que el ángulo entre los vectores sobrepasa los 90 grados y se encamina a la total diferencia, expresada por el valor -1.

Cabe mencionar, que el uso del valor -1 en la definición del vector, si bien puede que sea similar a escoger el valor positivo 1, favorece a que en la definición del vector exista una claridad de las posiciones dedos, y las cuerdas usadas y no usadas. El declarar valores para las cuerdas no usadas es necesario para poseer vectores con la misma dimensionalidad.

Una vez evaluada la similitud coseno de cada una de las versiones del conjunto en relación a cada uno de los posibles de contextos tonales, los casos por cada nota que arrojen la similitud coseno más cercana a uno, son mostrados como posibles contextos tonales. En la interfaz gráfica del calculador, para aclarar visualmente al usuario cuál de los posibles contextos tonales tiene la mayor similitud, se resalta en rojo el texto

de su cifrado. Hay veces, sobre todo en estructuras simétricas, que existe más de un caso de contexto tonal que comparte la mayor similitud coseno.

En esta tecnología existen tareas pendientes que no fueron posibles de completar en el marco del tiempo estipulado para la investigación, como lo son el desarrollo de una lógica más consistente para la selección de posibles casos de contextos tonales y una mayor especificidad en la nominación de los casos, buscando la congruencia total con el conjunto de notas seleccionado.

Esta primer aproximación, resulta valiosa en tanto que expone un camino para el empleo de la similitud coseno en la evaluación de un conjunto de notas, siendo las posibles implementaciones futuras, añadiduras a la lógica operativa implementada, misma que ha demostrado ser tan funcional como para considerar que en un futuro, esta tecnología deje de ser un intérprete de contextos tonales y pueda convertirse en un traductor literal de un conjunto de notas a un cifrado de acordes altamente específico, con una flexibilidad en su nominación similar a la vista en “Oolimo”.

### 3.3.2. Descifrador de Diagramas

[Enlace al código](#)<sup>29</sup>

El “Descifrador de Diagramas” es una herramienta que permite calcular las distintas formas en las que se puede tocar un grupo de notas en un rango de cinco trastes. Este componente consta de dos partes: una encargada de realizar el cálculo de las digitaciones y otra dedicada al dibujo de los diagramas de acordes.

Para llevar a cabo el cálculo de las digitaciones, se ha desarrollado un algoritmo que consta de los siguientes pasos:

1. Se calcula cómo es posible digitar el conjunto de notas seleccionado, a lo largo de los 22 trastes de una guitarra en afinación estándar, esto se logra al definir seis números correspondientes al valor de tono clase de cada una de las cuerdas

---

<sup>29</sup><https://www.bit.ly/3FYBjyQ>



y luego sumarle un valor por cada traste, aplicando posteriormente un módulo doce para determinar el tono clase de cada combinación cuerda-traste. Los resultados de este proceso se almacenan en un listado que contiene todas las coordenadas [*cuerda, traste, tono*] posibles, y de ahí se extraen aquellas que posean los tonos clase del conjunto de notas seleccionado.

2. Una vez obtenido el listado de las coordenadas, se procede a segmentarlo en los dieciocho posibles tramos de cinco trastes, comenzando desde el traste 1 y avanzando de manera continua hasta el traste 22. Es decir, se toman en cuenta los tramos del 1 al 5, del 2 al 6, del 3 al 7 y así sucesivamente, hasta llegar al tramo del 18 al 22.
  
3. Después de segmentar el listado en los distintos tramos de cinco trastes, se procede a crear listados separados para cada una de las notas del conjunto seleccionado que aparecen dentro de cada tramo. Esto significa que, si la estructura consta  $n$  notas, se crean  $n$  listados distintos, uno por cada nota. En ellos, se almacenan las coordenadas [*cuerda, traste, tono*] donde dicha nota aparece dentro de cada tramo. Logrando así sintetizar en qué posición se encuentra cada tipo de nota en cada tramo, facilitando el proceso de construcción de los diagramas correspondientes a cada tramo.
  
4. Para finalizar, se realiza una operación de producto cartesiano<sup>30</sup> con cada uno de los listados correspondientes a cada tipo de nota dentro de cada tramo. Esto consiste en combinar los elementos de los listados en cada tramo para generar todas sus posibles combinaciones. Posteriormente, se descartan aquellas combinaciones que contienen notas que se tocan en la misma cuerda, por ser físicamente imposibles de tocar de forma simultánea, y luego, se eliminan los elementos duplicados de la lista que surgen debido a que una misma combinación de notas puede aparecer en más de un tramo.

---

<sup>30</sup>“Un producto cartesiano se define como un conjunto ordenado de conjuntos. Es el conjunto de todas las combinaciones ordenadas posibles que contienen un miembro de cada uno de esos conjuntos.” (Johnston, 2014, p. 36).

A continuación, la Figura 3.14, ilustra el proceso del cálculo de digitaciones aplicado al primer tramo de un conjunto de notas [0,1,6].

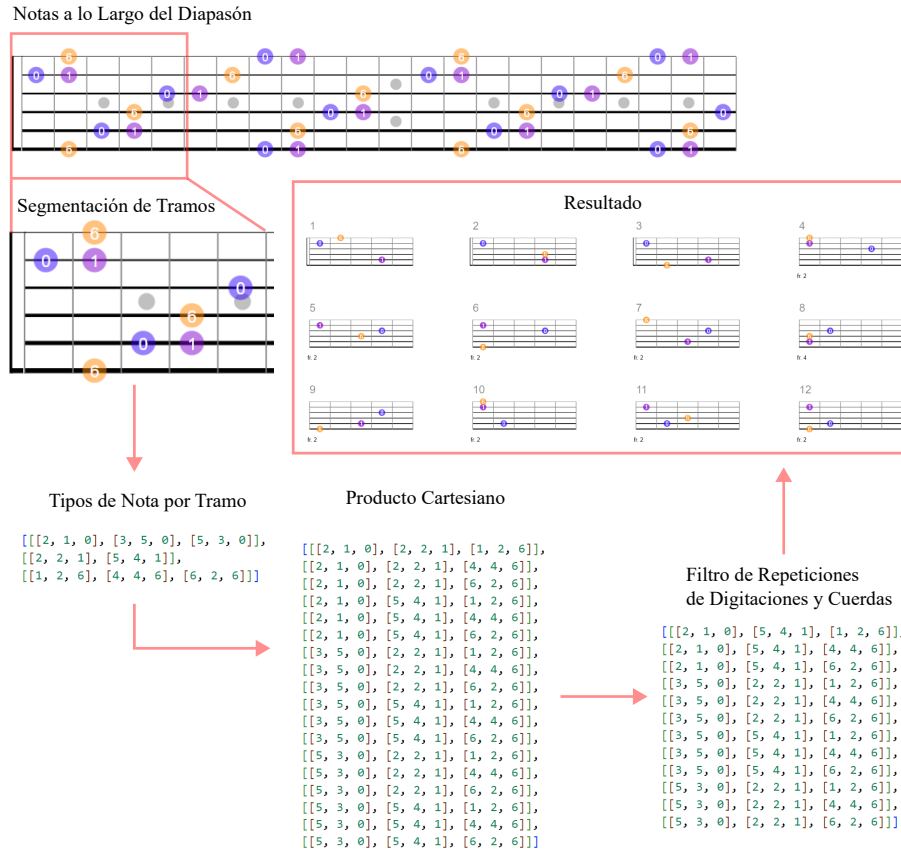


Figura 3.14: Ilustra el procedimiento para el cálculo de digitaciones aplicado a un conjunto de notas [0,1,6], mostrando únicamente los diagramas correspondientes al primer tramo de cinco trastes.

En lo que respecta al dibujador, se utiliza un mismo bloque de código para cada diagrama, en el, se define el dibujo de seis líneas verticales como los trastes y seis líneas horizontales como las cuerdas de la guitarra y se especifica que cada nota se represente mediante un círculo cuyo centro está en la distancia media entre las líneas verticales que representan los trastes. Además, incluye la definición de un número de traste para indicar la posición del diagrama en el diapasón y un número para especificar la ubicación de cada diagrama en la lista de diagramas generada por el descifrador. La Figura 3.15, muestra los componentes de cada diagrama.

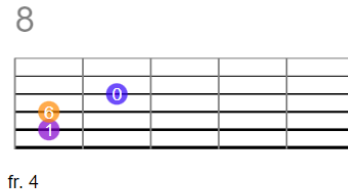


Figura 3.15: Muestra la estructura de los diagramas generados. Cada diagrama está numerado en la parte superior para facilitar su ubicación. Presenta seis cuerdas y cinco trastes, y por debajo, un número de traste que facilita la identificación de las notas y su posición en el diapasón de la guitarra.

### 3.3.3. Reproductor de Notas

[Enlace al código](#)<sup>31</sup>

Consiste en una función que llama la reproducción específica de cada nota de acuerdo a sus coordenadas [*traste-cuerda*].

Los sonidos están alojados en una subcarpeta del sitio y corresponden a archivos “.mp3” de todas las notas posibles de una guitarra Yamaha PF-300 grabadas mediante una interfaz de audio Audient iD14. Cada grabación dura cinco segundos y fue hecha con la pastilla del brazo, las perillas de volumen y tono al máximo y con cuerdas de 0.10mm a 0.56mm punteadas por una plumilla de plástico Dunlop de 2mm de grosor.



Figura 3.16: Muestra el equipo usado para la grabación de los sonidos de guitarra eléctrica. De izquierda a derecha, la guitarra eléctrica Yamaha PF300, la plumilla Dunlop de 2.00mm de grosor y la interfaz de audio Audient iD14.

<sup>31</sup><https://www.bit.ly/3TU50Me>

Para poder hacer uso de los archivos de sonido, fue necesario no solo llamarlos, sino pedir su carga previa y alojarlos en un buffer (espacio de memoria para alojar la información del audio). Una vez cargados, se llama a la reproducción simultánea de los archivos y, haciendo uso de los objetos de la “Web Audio API”, se modifica su ganancia, para que, de acuerdo a la cantidad de archivos de audio se reduzca su amplitud, evitando así la saturación de la señal. Finalmente, a cada archivo se le agrega una envolvente de sonido para que solo tenga una duración de dos segundos, en vez de los cinco segundos que originalmente duran, evitando así la superposición de sonidos a la hora de dar clic a más de un diagrama de acorde en un corto espacio de tiempo.

### 3.3.4. Operadores de Transposición, Inversión y Complemento

[Enlace al código de operadores](#)<sup>32</sup> [Enlace al código de la página del complemento](#)<sup>33</sup>

Los operadores son elementos que permiten modificar las notas seleccionadas o trabajar con el espectro de notas que han quedado fuera de dicha selección, se exponen en conjunto ya que forman parte de un mismo menú desde donde llevan a cabo sus tareas. A continuación, se describe su funcionamiento y sus lógicas operativas.

#### Transposición

Es el más sencillo de los operadores, para utilizarlo, primero se debe seleccionar el índice de transposición deseado que indica la cantidad de semitonos que la estructura de notas será transportada, para luego activar mediante un clic el botón “Transponer”. El índice puede tener un valor entre -6 y 6 semitonos, si el valor es negativo, la transposición será descendente, mientras que, si es positivo, será ascendente.

---

<sup>32</sup><https://www.bit.ly/3G0sUuF>

<sup>33</sup><https://github.com/hegelespaul/Postonal-Guitar/blob/master/scripts/complemento.js>

El algoritmo funciona de la siguiente manera: primero, toma en cuenta las notas que el usuario ha seleccionado, sumando a cada nota el índice de transposición, para luego realizar un cálculo de módulo doce y obtener los tonos clase resultantes de la suma. Una vez obtenidos los nuevos tonos, el algoritmo deselectiona las notas que había elegido el usuario y selecciona automáticamente las notas resultantes de la transposición. Logrando así transponer todos los valores del aplicativo, como la nominación del conjunto de notas seleccionado o sus diagramas resultantes. Con fines de clarificar su procedimiento, La Figura 3.17, muestra cómo sucede a nivel aritmético la transposición de un conjunto de notas  $[0,1,6]$ , desplazándolo cinco semitonos ascendente y descendente.

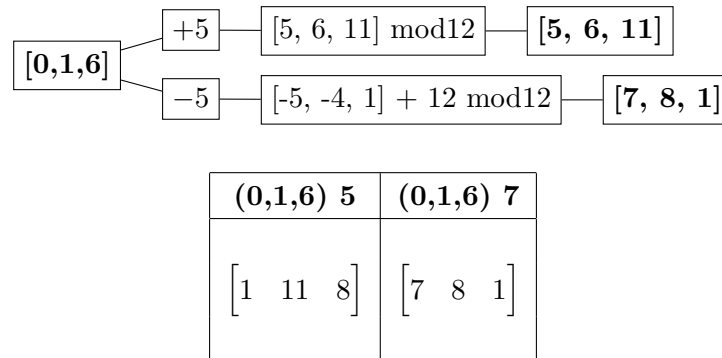


Figura 3.17: Muestra el proceso implementado para transponer un conjunto de notas  $n$  intervalos ascendente o descendente. En este caso, el conjunto de notas  $[0,1,6]$  se transpone cinco intervalos ascendente y descendente, terminando en los conjuntos  $[5,6,0]$  y  $[7,8,1]$ , respectivamente.

## Inversión

La inversión, opera tomando en consideración la nota desde la cual se evalúa la forma prima del conjunto de notas seleccionado, ello, como punto de partida para desarrollar en dirección contraria los intervalos que le componen. Es decir, la inversión crea una nueva estructura que es una versión especular de la original, basada en un eje de simetría determinado por la misma nota de partida.

El algoritmo desarrollado, evalúa la nomenclatura de la forma prima de las notas seleccionadas y sus intervalos. En primer lugar, detecta si el nombre de la forma

prima incluye una comilla que indique si es una forma prima en retrógrado o no. Y luego, utiliza esta información para estructurar pasos condicionales y continuar con el proceso de evaluación.

Si la forma prima no está en retrogrado, el algoritmo invierte las notas, multiplicando los intervalos por -1 y sumando la nota de partida a cada una de ellas. Luego, se suma 12 a cada nota resultante para asegurarse de que sean valores positivos y se aplica módulo doce para obtener los tonos clase correspondientes. En caso de que la forma prima esté en retrogrado, únicamente se suma la nota de partida a los intervalos que componen la estructura y se aplica módulo doce para obtener los tonos clase correspondientes.

La Figura 3.18 ejemplifica el procedimiento recién expuesto aplicado a dos formas primas, una en retrógrado y la otra no.

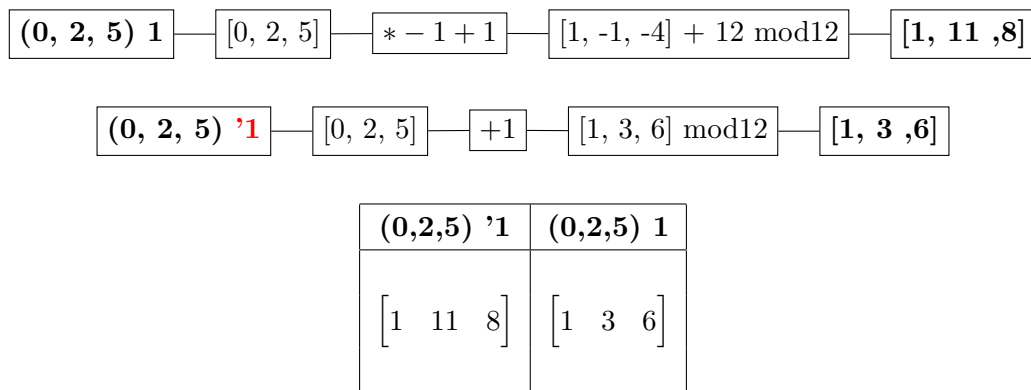


Figura 3.18: Muestra el proceso implementado para invertir un conjunto de notas. Se muestra el proceso de inversión para un conjunto en retrógrado (señalado con rojo) y otro en forma normal. La tabla de dos columnas de la parte inferior muestra las inversiones resultantes, teniendo en el encabezado sus formas primas y por debajo los componentes de cada conjunto.

## Complemento

El operador de complemento muestra las diferentes maneras en que las notas no seleccionadas se pueden agrupar en formar primas de entre tres y seis notas.

El algoritmo funciona de la siguiente manera: en primer lugar, se obtiene el complemento evaluando qué notas del espectro dodecafónico no han sido seleccionadas por el usuario. A continuación, se aplica el algoritmo de permutación de Heap<sup>34</sup> para generar todas las posibles combinaciones de las notas del complemento. Luego, estas permutaciones se dividen en conjuntos de una cantidad determinada de elementos y se ordenan ascendentemente tanto los elementos internos de los conjuntos, como los conjuntos mismos con el fin de identificar y eliminar elementos duplicados. Por último, se identifican las formas primas de los subconjuntos resultantes y se arrojan sus resultados a manera de listado.

Aunque el algoritmo del operador de complemento puede explicarse de manera relativamente sencilla, hay varios elementos a considerar para poder comprenderlo en profundidad, como lo son el funcionamiento del algoritmo de Heap para el cálculo de permutaciones, la lógica detrás de la división en subconjuntos de las permutaciones y la importancia de su ordenamiento ascendente para la eliminación de elementos duplicados.

En lo que respecta al funcionamiento del algoritmo de permutaciones de Heap, para una secuencia de entrada de  $n$  elementos, se mantiene fijo el elemento en la última posición y se construyen todas las permutaciones para el resto de los elementos. Después de eso, se intercambia el elemento en la última posición con uno del resto y se repite el proceso. La descripción original del algoritmo expuesta en la publicación de Heap (1963), es la siguiente:

1. Aplica este algoritmo a los elementos en las posiciones 1 a  $n-1$ .
2. Repite los siguientes pasos  $n-1$  veces, incrementando un contador  $i$  en 1 después de cada iteración:
  - a) Intercambia el elemento en la posición  $n$  con otro elemento en la lista de la siguiente manera:
    - 1) Si  $n$  es impar, intercambia los elementos en las posiciones 1 y  $n$ .

---

<sup>34</sup>Aunque no se describirá a detalle, el algoritmo de permutaciones de Heap es una técnica eficiente para generar todas las permutaciones posibles de un conjunto de elementos. Para una exposición detallada de su funcionamiento, se remite a Heap (1963), Sedgewick (1977) o GTProyectosMX (2021).

- 2) Si  $n$  es par, intercambia los elementos en las posiciones  $i$  y  $n$ .
- b) Permuta los elementos en las posiciones 1 a  $n-1$  aplicando este mismo algoritmo.

Este algoritmo se utiliza comúnmente para permutar una lista mediante intercambios sucesivos entre sus elementos. Utiliza un índice para evaluar el orden de dichos intercambios, lo que lo hace altamente eficiente ya que genera permutaciones de forma incremental y no requiere almacenar las permutaciones en la memoria al mismo tiempo.

El algoritmo de Heap tiene diversas implementaciones en código, algunas de las cuales mantienen inmutable el primer elemento y otras el último. Además, hay implementaciones que utilizan funciones recursivas<sup>35</sup> en lugar de estructuras condicionales, modificando así el planteamiento original del algoritmo.

En el presente desarrollo, por la facilidad de tomar una implementación en JavaScript ya elaborada, se optó por una versión recursiva basada en el algoritmo propuesto por el programador David Ernst en su sitio web personal: <http://dsernst.com/2014/12/14/heaps-permutation-algorithm-in-javascript/>, haciendo una ligera modificación para que las transformaciones del conjunto de notas evaluado, ocurran en una copia del conjunto para evitar modificarlo directamente.

La Figura 3.19, ilustra en forma de árbol de nodos, el resultado combinatorial llevado a cabo por el algoritmo de Heap para el cálculo de todas las permutaciones posibles de un conjunto de cuatro elementos “A, B, C, D”.

---

<sup>35</sup>La recursividad en la programación es una técnica muy útil que implica que una función se llame a sí misma para resolver un problema. Como se muestra en Cormen, Leiserson, Rivest, y Stein (2009), en una función recursiva, la función sigue llamándose a sí misma con una entrada modificada hasta que llega a un caso base que detiene la recursión.



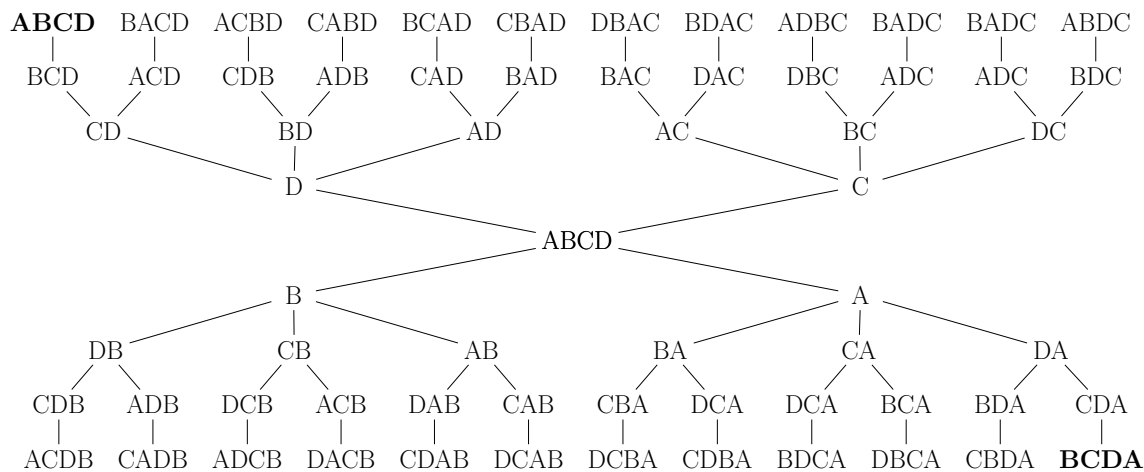


Figura 3.19: Se muestran, en forma de diagrama de árbol, las permutaciones posibles de un conjunto de elementos “A,B,C,D” obtenidas mediante el algoritmo de permutaciones de Heap. El orden de las permutaciones se lee de izquierda a derecha y de arriba abajo, es decir, empezando por  $ABCD$  y terminando por  $BCDA$ . Como tal, el diagrama no ilustra los pasos del algoritmo descrito, sino el orden de las permutaciones resultantes, clarificando cómo es que suceden las permutaciones manteniendo en la última posición cada uno de los distintos elementos  $D$ ,  $C$ ,  $B$ ,  $A$ .

Respecto a la lógica de la división de las permutaciones en subconjuntos, esta se basa en calcular cuántos grupos compuestos por conjuntos de entre tres y seis elementos se pueden formar con las notas del complemento. Existen proporciones que determinan la forma en que se dividen las permutaciones en subconjuntos y permiten evaluar todas las posibles combinaciones. Por ejemplo, un complemento de seis notas solo se puede dividir en dos subconjuntos de tres elementos, mientras que un complemento de nueve se puede dividir en tres subconjuntos de tres elementos, un subconjunto de cuatro y otro de cinco y en un subconjunto de seis y otro de tres. A continuación, la Tabla 3.3 muestra las posibles divisiones de los complementos en subconjuntos de acuerdo a su cantidad de elementos. Las permutaciones generadas por el algoritmo de Heap, se subdividen respecto a sus proporciones correspondientes. En algunos casos, como en los complementos de ocho o nueve notas, es necesario utilizar la lista de permutaciones más de una vez debido a que existen varias posibilidades de dividir el complemento en conjuntos de entre tres y seis notas.

<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
3 + 3	3 + 4	3 + 5	3 + 3 + 3
		4 + 4	4 + 5
			6 + 3

Tabla 3.3: Muestra cómo se subdividen los complementos de entre seis y nueve notas. En el encabezado, se muestran la cantidad de notas del conjunto y por debajo las maneras posibles de dividirse en subconjuntos de entre tres y seis notas.

Por último, el ordenamiento ascendente es crucial para eliminar permutaciones con los mismos elementos. Al dividir las permutaciones en grupos de subconjuntos, es posible que más de un grupo contenga subconjuntos con los mismos elementos, pero en un orden diferente. Al ordenar los elementos de los subconjuntos en orden ascendente, es posible identificar repeticiones de grupos con subconjuntos que contengan los mismos elementos, para así solo considerar casos únicos para el cálculo de las formas primas de cada subconjunto. La Figura 3.20 ilustra cómo funciona el ordenamiento ascendente para la identificación y eliminación de casos de subconjuntos con elementos repetidos.

$$\begin{array}{cccccc}
\begin{bmatrix} 0 & 1 & 2 \end{bmatrix} & \begin{bmatrix} 0 & 2 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 2 \end{bmatrix} & \begin{bmatrix} 1 & 2 & 0 \end{bmatrix} & \begin{bmatrix} 2 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 2 & 1 & 0 \end{bmatrix} \\
\begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \end{bmatrix} & \begin{bmatrix} 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} & \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 2 \end{bmatrix} & \begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix} & \begin{bmatrix} 2 & 0 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} & \begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix} \\
\begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \end{bmatrix} & \begin{bmatrix} 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} & & \begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix} & & 
\end{array}$$

Figura 3.20: Muestra cómo el ordenamiento ascendente de subconjuntos de las permutaciones del conjunto  $[0,1,2]$ , sirve para la eliminación de elementos duplicados. En la primera fila, se muestran todas las permutaciones posibles del conjunto  $[0,1,2]$ . En la segunda fila, se muestran los grupos de subconjuntos creados a partir de la división de las permutaciones en un subconjunto de dos elementos y otro de uno. En la tercera fila, se muestran los grupos de subconjuntos con elementos únicos. En la ilustración se identificando con colores aquellos subconjuntos que se consideran duplicaciones en tanto que contienen los mismos elementos solo que en distinto orden, algo que. Rojo para los grupos con subconjuntos  $[0,1][2]$ , verde para los grupos con subconjuntos  $[0,2][1]$  y azul para los grupos con subconjuntos  $[1,2][0]$ .

### 3.3.5. Dibujador de Polígonos de Conjuntos

[Enlace al código](#)<sup>36</sup>

Es una herramienta visual que permite representar la triangulación de notas inscritas dentro de un círculo que sigue la lógica de un reloj dodecafónico.

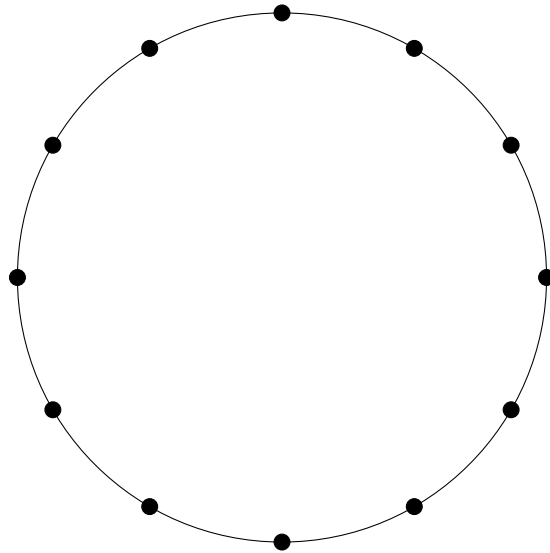
Para situar los elementos del reloj, primero se crearon doce botones circulares interactivos que sirven para seleccionar las notas de las que el usuario desea calcular sus diagramas de acordes y luego, se inscribieron en un círculo dividido en 12 secciones iguales.

Para generar las secciones del reloj, se calculó el ángulo que cada sección debe abarcar, dividiendo 360 grados (el ángulo completo de un círculo) entre 12 (el número de secciones). Luego, se le asignó a cada botón las coordenadas polares “X” y “Y”, obtenidas mediante el cálculo de las funciones seno y coseno de su ángulo en radianes (Ecuación 3.2).

---

<sup>36</sup><https://www.bit.ly/31M64A3>

$$x_i = \sin\left(\frac{360i}{12} \cdot \frac{\pi}{180}\right) \quad y_i = \cos\left(\frac{360i}{12} \cdot \frac{\pi}{180}\right) \quad (3.2)$$



3.2: La fórmula genera pares de coordenadas  $(x, y)$  que representan puntos equidistantes en la circunferencia de un círculo unitario. La coordenada  $x$  se calcula como el seno del ángulo correspondiente, y la coordenada  $y$  se calcula como el coseno del mismo ángulo. El ángulo se expresa en grados y se convierte a radianes multiplicando por  $\pi/180$ . La fórmula se utiliza comúnmente en geometría y trigonometría para representar la posición de puntos en un círculo. En la fórmula dada, la variable  $i$  es un índice que se utiliza para referirse a los diferentes puntos en la circunferencia del círculo unitario. La fórmula calcula las coordenadas  $(x, y)$  del punto  $i$ -ésimo en la circunferencia, donde  $i$  varía de 0 a 11 para generar un total de 12 puntos equidistantes.

Una vez obtenidas las coordenadas, para dibujar los polígonos, se utilizó el objeto “polygon ” de la librería “D3js”, el cual permite dibujar cualquier polígono mediante la especificación de las coordenadas polares de sus vértices. El lienzo donde se dibujan los polígonos se situó debajo de los botones del reloj, y a los vértices se les asignó un margen igual al radio de los botones para evitar que las líneas del polígono les atravesasen. A manera de ejemplo, la Figura 3.21 muestra una recopilación de los doce polígonos que corresponden a las doce formas primas posibles de conjuntos de tres notas, evaluadas desde cero.

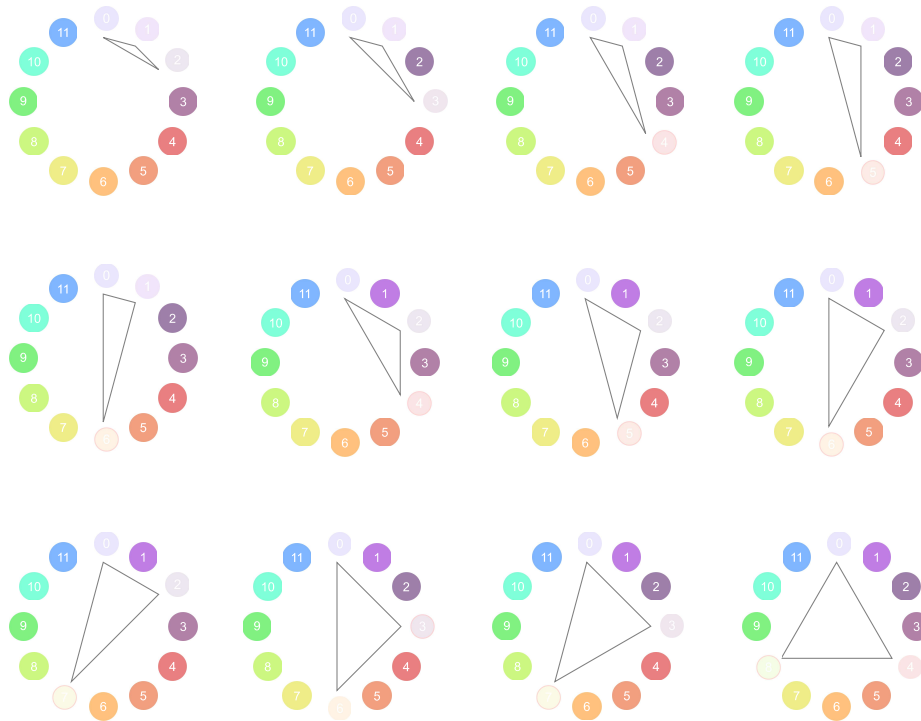


Figura 3.21: Muestra los polígonos de las doce posibles formas primas de conjuntos de tres notas, evaluadas desde cero. Ilustra cómo las notas seleccionadas pueden funcionar como vértices de polígonos que ayudan a sintetizar visualmente la estructura interválica de un conjunto de notas.

### 3.3.6. Descifrador de Acompañamientos

[Enlace al código](#)<sup>37</sup>

El descifrador de acompañamientos permite generar una conducción de voces de una progresión armónica. Para su funcionamiento, se debe seleccionar una progresión armónica y una disposición de voces del primer acorde como punto de partida. Una vez seleccionados, el descifrador se encarga de generar una conducción basada en la cercanía de las voces superiores y la semejanza de las digitaciones entre las disposiciones de voces de los acordes.

En primer lugar, se obtienen las coordenadas [*cuerda*, *traste*, *tono*] de la voz superior del primer acorde seleccionado. Luego, se comparan estas coordenadas con las de las

<sup>37</sup><https://www.bit.ly/3Khmxpz>

notas superiores de las posibles disposiciones de voces del siguiente acorde, midiendo la distancia euclidiana (Ecuación tal) de sus coordenadas tridimensionales.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (3.3)$$

3.3: La fórmula matemática permite calcular la distancia entre dos puntos en un espacio euclídeo de tres dimensiones. Estos puntos se representan mediante coordenadas cartesianas como  $(x_1, y_1, z_1)$  y  $(x_2, y_2, z_2)$ , y la distancia entre ellos se obtiene mediante la raíz cuadrada de la suma de los cuadrados de las diferencias de cada una de las coordenadas. La distancia euclidiana resultante es una medida de la distancia más corta que un objeto recorrería si se desplazara desde un punto al otro en línea recta dentro de un espacio tridimensional. La distancia euclidiana es la medida de distancia más comúnmente utilizada en la geometría euclidiana, y se aplica en diversas áreas de estudio, incluyendo las matemáticas y la física, entre otras disciplinas.

Luego, se identifican aquellas posibles disposiciones de voces del siguiente acorde, cuya distancia euclidiana de su voz superior sea la menor posible en comparación con la voz superior del primer acorde. Posteriormente, se evalúa la similitud de la digitación de cada caso en relación con la digitación del primer acorde, para finalmente, seleccionar como siguiente acorde, de entre la lista de posibles casos previamente identificados, aquel cuya digitación sea la más similar al primer acorde. Este proceso se repite para cada acorde de la progresión, hasta completar la misma.

Para evaluar la similitud de las digitaciones, se evalúa la similitud coseno (Ecuación 3.1) entre los vectores de diapasón del primer acorde y el siguiente. El vector de diapasón es una representación propuesta de un vector de  $n$  dimensiones que resume qué trastes y cuerdas son utilizados en la digitación de notas en instrumento con trastes. Su cálculo funciona de la siguiente manera:

Primero, se determina la cantidad de cuerdas del instrumento empleado y se crea una lista por cuerda. A continuación, se identifican qué cuerdas son utilizadas en la digitación que se desea representar y se anotan dos valores dentro de cada uno de los listados correspondientes de las cuerdas empleadas, el primero, representa el número de cuerda y el segundo, el número de traste. Finalmente, al resto de listados de cuerdas no empleadas se le agregan dos valores de -1, que especifica que ni la cuerda ni el traste han sido empleados. Esto permite que todos los arreglos mantengan las mismas dimensiones, independientemente del número de notas utilizadas en cada

digitación. Posteriormente, los listados resultantes se integran a un solo listado que muestra las coordenadas [*cuerda-traste*] de cada cuerda en orden ascendente. Este paso es necesario para poder integrar todo en un solo vector de una dimensión. A continuación, la Figura 3.22 muestra a manera de ejemplo, los vectores de diapasón de una progresión armónica de cuatro acordes, en ella se evidencia cómo es posible compactar cualquier digitación de una guitarra de seis cuerdas, en un vector de doce dimensiones.

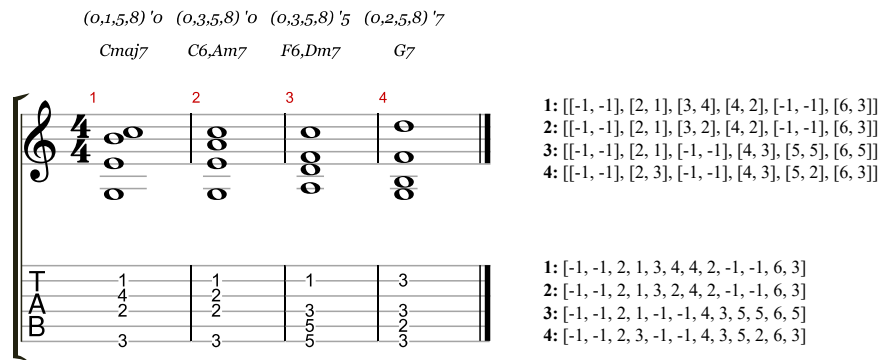


Figura 3.22: Muestra una progresión armónica de cuatro acordes y su proceso para extraer el vector de diapasón de cada uno. El primer compendio de cuatro conjuntos (arriba, derecha) muestra por cada acorde, un conjunto con un subconjunto por cuerda que comprende sus coordenadas [*cuerda, traste*] correspondientes. El segundo compendio de conjuntos (abajo, derecha) muestra como una vez “desempaquetadas” dichas coordenadas, cada acorde queda comprendido en un vector de doce dimensiones. Al igual que en el vector implementado en el “Interprete de Contextos Toniales” (Figura 3.13) al definir con el valor de -1 elementos no empleados, a la hora de la evaluación coseno se obtiene una misma dimensionalidad en todos los vectores y una claridad sobre las cuerdas usadas y no usadas.

Como se ha visto, el “Descifrador de Acompañamientos” resulta una herramienta útil para generar una conducción de voces de una progresión armónica. Su funcionamiento se basa en la cercanía de las voces superiores y la semejanza de las digitaciones de los acordes. Sin embargo, los elementos desarrollados para su funcionamiento abren la posibilidad de pensar en otras posibles lógicas para la conducción de voces. Por ejemplo, una lógica que evalúe la similitud coseno de todas las notas de los acordes sin tener en cuenta la distancia euclidiana de las notas superiores. Otra lógica que conecte los acordes según las notas inferiores y superiores más cercanas entre las digitaciones. Y una tercera lógica que pueda considerar el movimiento mínimo de las voces internas de los acordes, por mencionar algunos casos. Sin embargo, estos desarrollos quedan fuera de esta primera versión y se consideran posibles implementaciones futuras. Por

último, con ánimo de ilustrar el funcionamiento del descifrador, la Figura 3.23 expone el orden lógico de su algoritmo, haciendo énfasis en mostrar el razonamiento que subyace en la conducción de voces entre acordes.

(0,2,5,7) '5 F7sus4 (0,1,5,6) '2 Dmaj7#11

The diagram illustrates the voice leading process. At the top, the initial chord F7sus4 is shown with its notes (F, C, G, Bb) and a guitar diagram with fingerings (T: 8, A: 8, B: 6). Below this, a sequence of 20 possible next chords is shown, each with its notes and guitar diagram. A red box highlights the chosen path, which starts with the chord (F, C, G, Bb) and moves to (F, C, G, B) and finally to (F, C, G, Bb). The final chord diagram shows the notes (D, F, A, C) and fingerings (T: 8, A: 6, B: 6) for Dmaj7#11.

Figura 3.23: Ilustra la conducción de voces partiendo de una disposición de voces de un primer conjunto de notas (0,2,5,7) '5 F7sus4, para buscar la siguiente disposición de voces de un acorde (0,1,5,6) '2 Dmaj7#11. Primero, se evalúan diversas disposiciones de voces posibles del siguiente acorde que comparten la misma voz superior y luego se selecciona el siguiente acorde basándose en la similitud de las digitaciones de ambos acordes. Nótese cómo la disposición elegida (marcada en rojo) utiliza las mismas cuerdas que el primer acorde, realizando pequeños movimientos de trastes entre sus elementos.

### 3.3.7. Partitura Interactiva

Enlace al código<sup>38</sup>

<sup>38</sup><https://www.bit.ly/3Zphnfx>



Es el componente donde queda plasmada la conducción de voces elaborada en el “Generador de Acompañamientos”. Permite visualizar en partitura y tablatura los acordes, reproducir la partitura, definir bucles en la reproducción e imprimir la partitura y descargarla en formato “.gp”.

La Partitura Interactiva se compone de varios elementos de la librería “alphaTab”. La implementación se basa en un tutorial publicado en la página oficial de la librería (<https://alphatab.net/docs/tutorial-web>). En él, se explica cómo generar, a nivel de código, la interfaz de la partitura interactiva y sus funcionalidades. A continuación, se resumen los pasos necesarios para implementar la “Partitura Interactiva”, mismos que se tratan a detalle en el tutorial mencionado.

1. Preparar el proyecto al definir el lugar donde se incluirá la partitura dentro de su archivo HTML
2. Determinar dónde se integra cada uno de sus elementos (cabecera, ventana principal, reproductor de audio, etc.).
3. Inicializar la librería mediante la declaración de su función en JavaScript.
4. Crear la interfaz visual de un indicador de carga para señalar al usuario el proceso de la generación de la partitura.
5. Desarrollar un selector de pistas.
6. Desarrollar los controles de reproducción y la salida gráfica de las funcionalidades declaradas.

Además de prescindir del selector de pista, dado que las partituras generadas son para un instrumento solista, a la implementación de la partitura mostrada en el tutorial se le realizaron dos ajustes: el primero, consistió en optar por utilizar grabaciones propias para reproducir las notas de las partituras, en lugar de los sonidos MIDI de “alphaTab”. El segundo, fue el empleo del componente “alphaTex” para la creación de las partituras, en vez de un archivo XML.

Con el objetivo de simplificar la generación de la partitura, se estableció que cada acorde se escriba en un solo compás de cuatro cuartos, con una duración equivalente a una nota redonda. Esto permite una mayor facilidad en la lectura y escritura de la partitura, ya que se evita la necesidad de utilizar diferentes figuras rítmicas y compases complicados para representar los acordes, aunando a que sirve para establecer un punto de partida flexible para que el usuario pueda descargar la partitura y modificarla de acuerdo a sus necesidades.

En cuanto a la reproducción de sonido, se hizo uso del reloj interno de la librería, el cual permite llevar un seguimiento preciso de la reproducción de la partitura. Se calculó el momento de inicio de cada compás y, como se expuso en el apartado del “Reproductor de Notas”, se detonaron las grabaciones correspondientes a las coordenadas cuerda-traste de las notas de los acordes, logrando así una experiencia de audio más realista y personalizada que con los sonidos MIDI de la librería.

A pesar de que la partitura interactiva cuenta con múltiples funcionalidades, actualmente el archivo descargado es de un software privativo. Sin embargo, se espera poder implementar la descarga de un archivo XML que pueda ser utilizado por distintos software libres. Esta funcionalidad se prevé relativamente sencilla, ya que el archivo “.gp” descargable, es en realidad una carpeta ZIP renombrada que contiene un archivo XML.<sup>39</sup>

Por último, es importante mencionar que en la partitura de los acordes, se optó por dejar un cifrado que muestre la forma prima y el contexto tonal más cercano de cada acorde. La Figura 3.24, muestra un ejemplo de una partitura generada con dicho cifrado, mostrando a su derecha el texto procesado por “alphaTex” para su creación.

---

<sup>39</sup>Aunado al cambio de extensión, hace falta un proceso de edición del XML donde se traduzcan sus elementos a un esquema de archivo de partitura convencional, ya que “Guitar Pro”, editó el nombre de varias etiquetas y agregó información especial para su software.

# Mi tema

Yo

(0,1,5) 9 (0,2,5,7) 0 (0,1,3,5) 7 (0,2,4,6,9) '9  
Bbmaj7 G7sus4 C7b13 F9

```
\title "Mi tema"
\subtitle "Yo"
\tempo 180
\instrument 25
.
(5.1 5.5 6.6 ).1{ch"(0,1,5) 9 Bbmaj7"} |
(3.1 1.6 5.5 1.2 ).1{ch"(0,2,5,7) 0 G7sus4"} |
(4.6 3.1 1.5 1.2 ).1{ch"(0,1,3,5) 7 C7b13"} |
(3.1 4.2 5.6 3.4 3.5 ).1{ch"(0,2,4,6,9) '9 F9"}
```

Figura 3.24: Se muestra una progresión armónica de cuatro acordes generada con el “Generador de Acompañamientos”, acompañada del texto en “alphaTex” que le genera. Es interesante notar lo sencillo que resulta entender la morfología del lenguaje utilizado. En primer lugar, se escriben las coordenadas traste-cuerda de cada nota del acorde entre paréntesis, seguidas por un número que indica la duración de las notas (en este caso, 1, que equivale a la duración de un compás). Después, se utiliza el comando “ch” dentro de corchetes para indicar que las notas corresponden a un acorde, colocando su cifrado entre comillas.

## 3.3.8. Espacio Tridimensional

[Enlace al código](#)<sup>40</sup>

Es el componente principal del “Visualizador de Formas Primas”. Consiste en una visualización interactiva de un plano tridimensional que alberga puntos que representan las distintas formas primas de conjuntos de entre tres y seis notas.

El espacio tridimensional, está programado mediante la función del gráfico de dispersión de “Plotly” que permite crear puntos en un plano tridimensional especificando sus coordenadas “X”, “Y” y “Z”. Esta función, también permite agregar nombres y

<sup>40</sup><https://github.com/hegelespaul/Postonal-Guitar/blob/master/scripts/analyser.js>

etiquetas con valores de texto a cada punto y programar interactividad con los puntos, ya que al mover el puntero del ratón sobre o al dar clic sobre ellos, es posible activar diferentes funciones o elementos.

En el espacio tridimensional, cada punto representa una forma prima y tiene una etiqueta con su nombre. Al mover el ratón sobre el punto, se muestra la estructura de la forma prima y sus posibles contextos tonales evaluados desde cero (transpuestos a “do”). Y al hacer clic, se puede navegar a la página del “Calculador de Diagramas” correspondiente a esa forma prima. Los ejes del espacio tridimensional están estructurados de la siguiente manera:

1. X: Número de notas que contiene la forma prima.
2. Y: La desviación estándar del vector interválico definido en semitonos.
3. Z: El valor máximo de la forma prima.

El eje “X” es autoexplicativo, ya que señala el número de notas que tiene cada forma prima. En este sentido, las formas primas solo pueden tener cuatro valores diferentes: 3, 4, 5 y 6, debido a que solo se están considerando conjuntos que contengan entre tres y seis notas.

El eje “Y”, representa la desviación estándar del vector interválico de las formas primas, expresado por su cantidad en semitonos, o, dicho de otra forma, expresado por el resultado de la multiplicación de la cantidad de los intervalos de cada elemento del vector por la cantidad de semitonos que contiene el intervalo que representa.

La desviación estándar, es una medida estadística que indica la cantidad de dispersión o variabilidad de un conjunto de datos con respecto a su media aritmética. En general, cuanto mayor sea la desviación estándar, mayor será la variabilidad de los datos. Esta medida se expresa como la raíz cuadrada de la varianza de los datos y se representa por la letra griega sigma. Su fórmula matemática es la siguiente:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (3.4)$$

3.4:

$\sigma$  es la desviación estándar,  $n$  es el número de elementos,  $x_i$  es cada uno de los elementos,  $\bar{x}$  es la media de los elementos.

Este cálculo se realiza en varios pasos. Primero, se resta la media ( $\bar{x}$ ) de cada observación ( $x_i$ ) y se eleva al cuadrado la diferencia. Luego, se suman estos resultados y se divide por el número de observaciones. Finalmente, se toma la raíz cuadrada del resultado para obtener la desviación estándar. La Figura 3.25, ejemplifica la obtención de la desviación estándar del vector interválico (expresado por su cantidad en semitonos), de dos formas primas, (0,1,2) y (0,2,4,6,8,10).

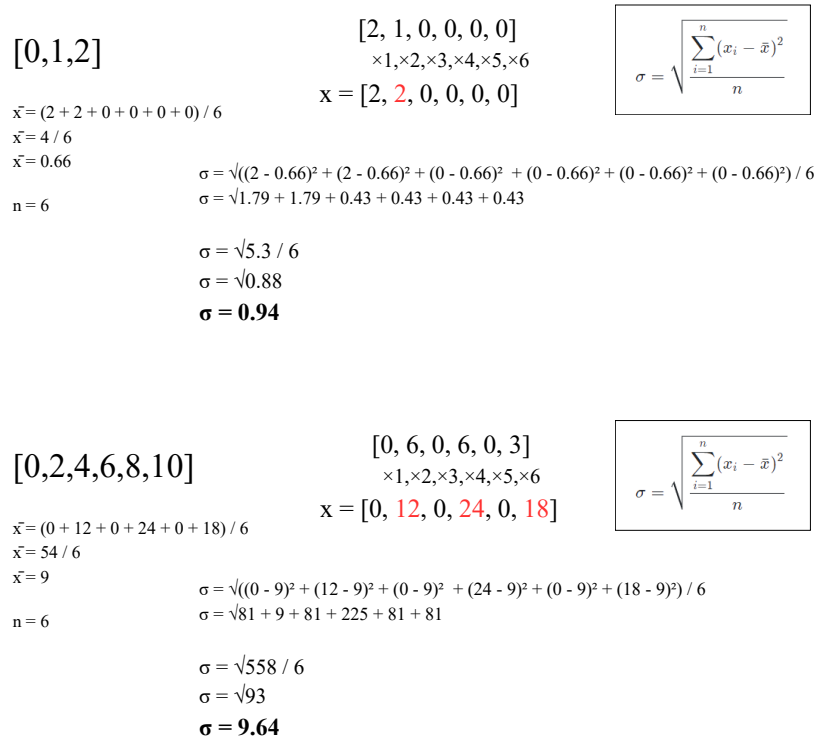


Figura 3.25: Muestra el cálculo de la desviación estándar del vector interválico definido por su cantidad de semitonos de los conjuntos de notas [0,1,2] y [0,2,4,6,8,10]. Nótese como cada elemento de los vectores interválicos se multiplica por la cantidad de semitonos del intervalo que representa, siendo esto un paso crucial para determinar la desviación estándar, en tanto que muestra de mejor manera los semitonos que involucra cada intervalo representado.

Por último, en lo que respecta al eje “Z”, este se estructura en base al valor máximo de las formas primas. Permite agrupar en una misma coordenada unidimensional, aquellas formas primas que compartan un valor máximo sin importar su contenido interno. Por ejemplo, las formas primas (0, 1, 4), (0, 2, 4) y (0, 1, 2, 3, 4), se agrupan en el mismo valor cuatro, independientemente de sus valores internos o la cantidad de notas que contengan.

Cabe mencionar, que cada uno de los ejes se halla normalizado entre valores del 0 al 1, esto sucede para permitir una proyección de puntos equitativa a lo largo de los ejes, ya que los valores de cada eje pueden variar mucho uno con otro y generar proyecciones irregulares. Para normalizar sus datos, primero se obtuvo el valor mínimo y el valor máximo del conjunto de datos. Una vez que se obtuvieron estos valores, se sustrae el valor mínimo al valor de cada punto y se divide entre la resta del valor mínimo al valor máximo ( Ecuación 3.5).

$$x_{norm} = (x - \min(x)) / (\max(x) - \min(x)) \quad (3.5)$$

3.5: El valor original es definido por  $x$  y  $x_{norm}$  es el valor normalizado. Esta fórmula escala los valores originales de manera que el valor mínimo del conjunto de datos se convierte en 0 y el valor máximo se convierte en 1, escalando proporcionalmente el resto de los valores.

Es importante tener en cuenta que la normalización de los datos no cambia su forma general de distribución, sino que simplemente los ajusta para que estén en una escala común.

La estructuración del espacio tridimensional, mediante estos tres ejes, permite observar relaciones de semejanza entre las distintas formas primas, teniendo en cuenta aspectos como la cantidad de notas, la apertura de sus formas primas (representada por su valor máximo) y la varianza de su contenido interválico. Aunque esta propuesta es sencilla en comparación con técnicas de reducción de dimensionalidad más avanzadas, como PCA o t-SNE<sup>41</sup>, ofrece una mayor claridad en la estructuración de cada eje

---

<sup>41</sup>El análisis de componentes principales (PCA, por sus siglas en inglés), es una técnica de reducción dimensional comúnmente utilizada para analizar datos. Se enfoca en la cantidad máxima de varianza con el menor número de componentes principales. Por su parte, t-SNE, es un algoritmo de reducción de dimensionalidad que preserva las relaciones entre los datos en un espacio de baja dimensión, lo

y permite una mejor diferenciación de los componentes en función de características fundamentales para su comparación.

Los ejes propuesta resultan efectivos a la hora de proponer una visualización funcional para visualizar relaciones de similitud entre las distintas formas primas, en tanto que permite fácilmente ver, por ejemplo, aquellas formas primas que poseen un mismo número de notas y un mismo valor máximo de apertura pero una mayor o menor diversidad en su contenido interválico, o aquellas que tienen una diversidad similar pero diferente número de notas, como 3-11 y 6-Z36. A continuación, en la Figura 3.26 se muestra una captura de pantalla del “Espacio Tridimensional”.

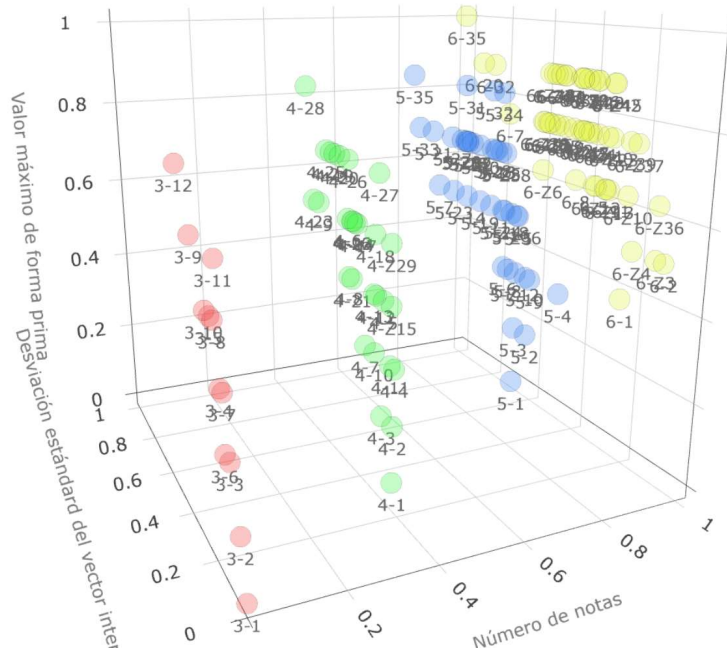


Figura 3.26: Muestra el espacio tridimensional generado. Nótese como la conceptualización presentada de los ejes permite un orden de las formas primas que permiten encontrar coincidencias entre las distintas cardinalidades de los puntos, a tal grado que los extremos de la proyección son 3-1 (0,1,2) y 6-35 (0,2,4,6,8,10), demostrando que un ordenamiento basado en la cantidad de notas, la desviación estándar del vector interválico y el valor máximo de forma prima son congruentes con las características de las formas primas, ya que 3-1 es la forma prima más simple y aparece como la primera en el listado de Forte y la forma 6-35 es la única forma prima de seis notas con un intervalo máximo de diez semitonos y con tres tritonos dentro de su vector interválico.

que lo hace ideal para descubrir patrones y estructuras complejas en datos no lineales. Para mayor información sobre técnicas comunes de reducción de dimensionalidad de datos, se recomienda el trabajo de Sorzano, Vargas, y Montano (2014) que expone un compendio de ellas.

Aunque el visualizador resulta ser una proyección funcional de las formas primas que permite evidenciar relaciones entre ellas, en un futuro, no se descarta explorar implementaciones que ocupen métricas diferentes para configurar los ejes, como la variación del espectro de las distribuciones circulares de los tonos clases expuestas en Krantz y Douthett (2005) o la propiedad de Myhill descrita en Clough y Myerson (1985), por mencionar algunas.

Se han presentado los tres aplicativos creados y las tecnologías que los integran, junto con la explicación detallada de sus lógicas de programación. Estos aplicativos funcionales son la prueba fehaciente de que los algoritmos diseñados son efectivos para llevar a cabo las tareas descritas. Aunque el código desarrollado no ha sido presentado en la tesina, este posee comentarios a lo largo de sus líneas para facilitar su comprensión. Se espera que aquellos interesados en revisarlo posean las habilidades necesarias para interpretarlo adecuadamente y apreciar el trabajo realizado.

Se espera que lo expuesto hasta este punto haya aclarado el trabajo desarrollado durante la maestría. Sin más preámbulos, a continuación, se presentan las conclusiones del trabajo, en ellas se incluyen breves ejemplos de elementos musicales creados empleando los aplicativos desarrollados.



# Conclusiones

La tesina se ha enfocado en la descripción de los tres aplicativos desarrollados, siendo su implementación el eje central de la investigación. No obstante, es posible profundizar aún más en cada una de estas tecnologías mediante un enfoque crítico que analice detalladamente sus procedimientos y conexiones técnicas y teóricas, para poder generar en un futuro nuevas perspectivas de estudio o desarrollos tecnológicos. Aunque esta investigación continuará más allá de lo aquí presentado, este documento ha marcado el cierre del trabajo realizado durante los dos años de duración de la maestría, y sirve como evidencia de los logros alcanzados en este período.

A lo largo del documento, se explicó cómo se desarrolló “Guitarra Postonal en Línea”, un sitio web que aloja tres aplicativos auxiliares para el estudio y la composición de material musical postonal implementado en la guitarra.

Se presentaron los referentes teóricos que sirvieron para la conceptualización de los aplicativos y también, se mencionaron casos de aplicativos similares que cubren parcialmente las funcionalidades implementadas, debido a que no se encontraron casos que realicen tal cual las mismas tareas.

Tanto el sitio como los aplicativos presentados, se consideran una primera aproximación para ofrecer tecnologías que cubran tres necesidades comunes en la práctica de guitarristas y compositores: el cálculo de digitaciones de acordes o escalas, la creación de conducción de voces entre acordes y el análisis armónico de conjuntos de notas.

De igual forma, los aplicativos sirven como un recurso tecnológico para facilitar el análisis, el estudio y la implementación de material musical postonal en contextos tonales, en tanto que se ejercen cruces entre la teoría de conjuntos de tonos clase de Allen Forte y la armonía de jazz.

Aunque se hace un recuento detallado de las tecnologías desarrolladas para su funcionamiento, exponiendo las lógicas de sus algoritmos y cómo se vinculan entre sí, se observan al menos dos tareas pendientes que podrían agregar un gran valor significativo a la investigación presentada. En primer lugar, la retroalimentación de comunidades artísticas y tecnológicas derivada de la interacción con los aplicativos. En segundo lugar, un enfoque creativo de los aplicativos, es decir, la creación de material musical usando estas herramientas.

En cuanto al primer caso, se espera establecer un diálogo con cualquier persona que lea el presente documento, incluyendo el sínodo que evaluará la investigación, recibiendo de la mejor manera los posibles comentarios sobre los aplicativos desarrollados. Algunos de estos, se espera, que se encuentren recogidos en el cuerpo del documento donde se mencionan posibles implementaciones futuras, como lo son:

1. Una mejora en la visualización de la organización del complemento, evitando el uso de un listado en texto ya que resulta ser poco intuitivo y sobresaturado de información.
2. Añadir la posibilidad de que el usuario elija el cifrado de acorde de forma específica, seleccionado desde cuál de sus componentes se evalúa armónicamente el conjunto de notas seleccionado.
3. La exportación de la partitura generada en algún formato libre para permitir su edición mediante el uso de software de notación musical no privativo.
4. La visualización de los acordes seleccionados en el “Generador de Acompañamientos” dentro del espacio tridimensional propuesto, para evidenciar relaciones entre los acordes.
5. Una mejora en el algoritmo del cálculo de formas primas para la detección de todas las posibles formas primas que se pueden alojar en conjuntos con estructuras simétricas o palindrómicas.
6. Una mayor especificidad de los contextos tonales, logrando una traducción literal de la forma prima a acorde con especificaciones sobre sus tensiones, alteraciones y omisiones de notas.

7. El desarrollo de otras lógicas en la conducción de voces entre acordes.
8. Explorar otras métricas para configurar otra visualización del espacio tridimensional que tome en cuenta características escalares y de conjunto de datos de igual forma valiosos para el planteamiento de visualizaciones funcionales.

Aunado a esto, se espera poder continuar compartiendo los aplicativos en distintos ámbitos, tanto musicales como pedagógicos, para poder recoger impresiones, opiniones y comentarios de una comunidad amplia y diversa que pueda estar interesada en las funcionalidades de los aplicativos. Incluso, se vislumbra la posibilidad de implementar un formulario dentro del sitio para que cualquier persona pueda dar retroalimentación y enriquecer así el proyecto.

En cuanto al segundo punto, un enfoque creativo podría constituir una investigación por sí misma. Debido a que a lo largo de la maestría se ha dedicado un tiempo significativo a la adquisición de conocimientos tecnológicos para desarrollar los aplicativos, solo se han realizado ligeras exploraciones en este sentido. En la sección “Ejemplos de Uso” (<https://guitarrapostonalenlinea.netlify.app/ejemplosdeuso>), se presenta un breve compendio de ejercicios musicales (en audio y partitura) creados con la ayuda de los aplicativos. Estos ejemplos se dividen en dos categorías: estudios y composiciones musicales. Aunque puede ser difícil revisar cada ejemplo a detalle dentro de las conclusiones, es esencial, antes de terminar la tesina, presentar algunos de estos ejemplos como la culminación musical de las funcionalidades implementadas en los aplicativos. A continuación, sólo con el propósito de ilustrar los resultados obtenidos con el uso de las herramientas desarrolladas, se presentan dos ejemplos de canciones.

La primera, titulada Ivelin, nace de pensar en el uso de acordes totalmente diatónicos a “Sol mayor” planteando en un primer momento el uso de conjuntos de cuatro notas y luego de tres, para un cierre con un conjunto de cinco notas. Al seleccionar un punto de partidas e hilar la conducción de voces entre los acordes, se respetó casi por completo el resultado dado, únicamente se incorporó una presentación de los acordes a manera de arpeggio y se añadieron ligeras modificaciones en el segundo y el último acorde. Una vez teniendo la disposición de cada acorde, se procedió a dotar una estructura añadiendo barras de repetición y un bajo ostinato que se haya presente a lo largo del tema. La Figura 3.27 muestra la partitura final y por debajo, el acompañamiento generado que sirvió como punto de partida para su composición.

# Ivelin

Hegel Pedroza

Intro

3

5 x2

7 x2

12 Solos

(0,2,4,6) 0 (0,1,3,5) 7 (0,2,3,5) 4 (0,1,3,5) 6 (0,2,6) 6 (0,2,5) 9 (0,1,5) 11 (0,2,4) 2 (0,1,3,5,8) 6  
 D9 Gmaj13 Am13 B7b13 F#m7b5 Dm6 Cmaj7 F#7b13 Gmaj9

T	5	5	5	7	7	5	7	10	10
A	7	7	5	7	7	5	7	10	7
B	3	5	5	10	7	5	7	9	10

Figura 3.27: Muestra la partitura de la pieza titulada “Ivelin” y la progresión armónica creada en el “Generador de Acompañamientos” que sirvió de base para su creación.

En lo que respecta al segundo ejemplo, “Aidee”. Su concepción nace de un juego sencillo de proporciones entre conjuntos de notas. Primero, decidiendo seleccionar un conjunto formado por las notas [0,1,2] y luego otro con sus extremos opuestos [9,10,11], para finalmente llenar el espectro dodecafónico con los conjuntos [3,4,5] y [6,7,8]. Los tres conjuntos están conformador por la misma forma prima (0,1,2) y de acuerdo con Schat, pertenecerían a una configuración de la hora “IV” o 1 + 1 según O’Gallagher.

Posteriormente, se pensó en la selección de otros cinco acordes de tres notas que que compartiesen entre ellos un mismo elemento, proponiendo el uso de una relación que permitiese que el elemento en común permaneciese estático mientras los otros dos elementos se moviesen un semitono de distancia entre cada acorde, uno de forma ascendente y otro descendente. Configurando así los conjuntos [4,5,6], [4,6,8], [3,6,9], [2,6,10] y [1,6,11].

Por último, se pensaron otros cuatro acordes a manera de cierre en donde se combinasen conjuntos de entre tres y cuatro notas. Para plantear dicho cierre, se pensó en que la nota “do”, a propósito no expuesta más que al inicio de la progresión, fuera un componente del último acorde. Derivando así en cuatro acordes donde los primeros tres cubren casi el total del espectro dodecafónico, cerrando con un cuarto acorde que presenta la nota “do ”y que configura con el resto de sus elementos una simetría de dos tritonos, en tanto que comprende el tritono de “do”, “fa sostenido”, y un semitono por debajo, otro tritono compuesto por las notas “fa”y “si”. Esta progresión de cierre terminó por tener los conjunto de notas [5,6,7],[2,3,4,5], [8,9,10,11] y [0,5,6,11].

Para poder ilustrar de mejor manera los conjuntos de notas empleados, se muestra en la Figura 3.28 el menú de botones de la progresión desarrollada, donde se pueden observar con facilidad las notas de cada conjunto seleccionado. Posteriormente, al igual que el ejemplo pasado, en la Figura 3.29, se muestra la partitura final de la pieza en contraste con la partitura hecha en el “Generador de Acompañamientos” que sirvió de base para su desarrollo.

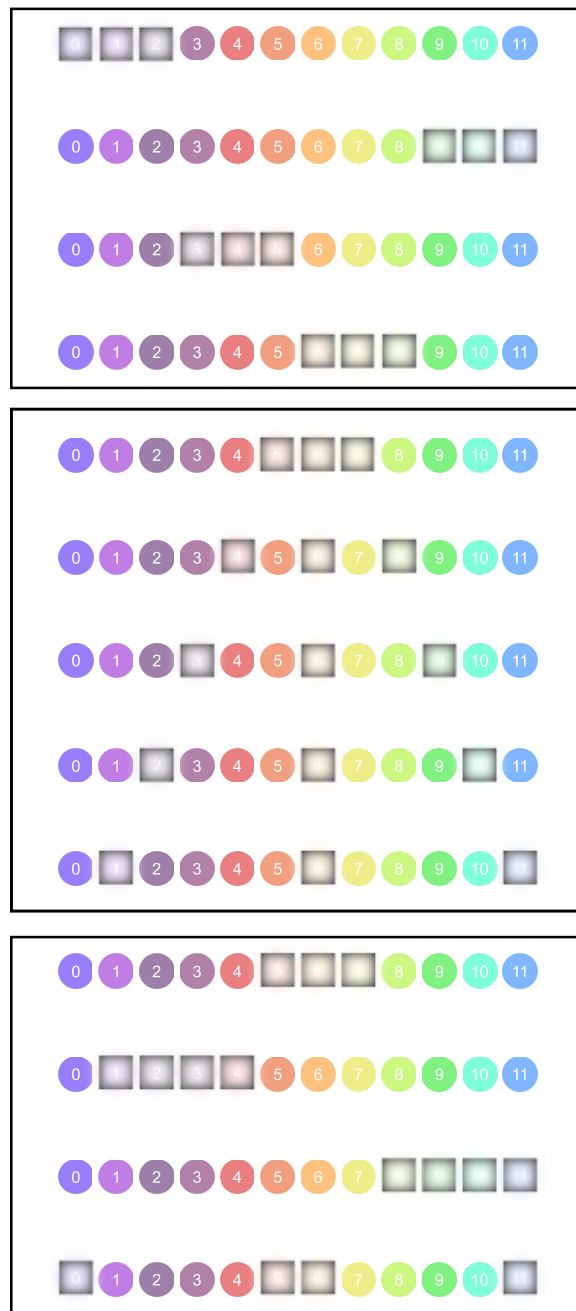


Figura 3.28: Se muestran, con cuadrados, las notas de los conjuntos de notas seleccionados para la creación de la conducción de voces de la pieza titula “Aidee”. Mediante recuadros, se dividen los tres momentos descritos de la progresión.



$(0,1,2) 0$     $(0,1,2) 9$     $(0,1,2) 3$     $(0,1,2) 6$     $(0,1,2) 5$     $(0,2,4) 4$     $(0,3,6) 3$     $(0,4,8) 2$     $(0,2,7) 1$     $(0,1,2) 5$     $(0,1,2,3) 1$   
*C#mMaj7*   *BbmMaj7*   *EmMaj7*   *GmMaj7*   *F#mMaj7*   *Ab7b13*   *Ebdim* *Daug*, *F#aug*, *Bbdim*, *F#sus4*, *Bsus2*   *F#mMaj7*   *Dmaj9*

$(0,1,2,3) 8$     $(0,1,6,7) 0$   
*Amaj9* *Cmaj7#11*, *F#maj7#11*

Figura 3.29: Muestra la partitura de la pieza titulada “Aidee” y la progresión armónica creada en el “Generador de Acompañamientos” que sirvió de base para su composición

Es posible ver que la partitura final tiene varias adaptaciones sobre la conducción de voces generada, incluyendo no solo tiempos armónicos diferentes sino también acordes de paso y ajustes en las disposiciones de voces, así como en su exposición, sin embargo, la identidad sonora permanece inalterable ya que se han respetado los elementos seleccionados, lo que permite una relación auditiva entre las sonoridades de la partitura final y las generadas en el aplicativo.

Ambos ejemplos han sido grabados en formato de trio de jazz, compuesto por guitarra eléctrica, contrabajo y batería, incluyendo pasajes de solos improvisados con una estructura tradicional donde se cierran las piezas con una reexposición del tema principal.

Se han mencionado solo dos de los ejemplos mostrados en el enlace compartido, debido a que representan composiciones musicales completas que muestran un notable contraste en su tratamiento musical. Una de ellas es más atonal, mientras que la otra es diatónica, aunque ambas fueron creadas utilizando el “Generador de Acompañamiento” y se basaron en premisas básicas para su concepción.



Este contraste, entre el material musical que sigue un tratamiento armónico propio de la tonalidad y otro que se deriva de una selección de notas basada en proporciones, y no en términos tonales, refleja de cierta forma un enfoque central de esta investigación, que es el uso de elementos posttonales en la guitarra y su análisis para su posible uso en contextos tonales comúnmente empleados. Los cruces entre ambas perspectivas de organización musical, resultan ser un área muy fructífera que puede dar lugar a otras investigaciones y a la creación de teorizaciones y material musical como el mostrado en el primer capítulo de la tesina.

La idea inicial de la investigación fue desarrollar una herramienta que permitiera el estudio de teorizaciones posttonales del ámbito del jazz en la guitarra eléctrica. A partir de ahí, se lograron materializar tres aplicativos que resultan auxiliares en dicha tarea, y permiten, además, el análisis y el cálculo de digitaciones para fines de composición y estudio musical, siendo así útiles no solo para guitarristas, sino también para compositores que deseen utilizarlos dentro de su actividad artística, como lo dejan ver los ejemplos musicales expuestos.

Sumado a ello, es posible afirmar que en los algoritmos ideados para la implementación de los aplicativos:

1. Se extendieron los procedimientos de Okazaki para el cálculo de diagramas de acordes a conjuntos de voces de hasta seis notas.
2. Se llevó a cabo una comparativa de material posttonal con el objetivo de evaluar su posible uso en contextos tonales comúnmente empleados. Este procedimiento fue similar a los utilizados por Terefenko, pero se amplió el uso de tricordes hasta conjuntos de seis notas y se usaron un mayor número de contextos tonales con ánimos de sentar las bases para la creación de un traductor literal de un conjunto de notas a un cifrado de acordes altamente específico.
3. Se llevó a cabo un ordenamiento de todo el espectro dodecafónico en subconjuntos posibles de formas primas que van desde tres hasta seis notas. Este procedimiento fue similar a las horas de Schat estudiadas por O’Gallagher, con la diferencia de que el total de las doce tonas se dividió por el total de sus

posibles agrupaciones a partir de una forma prima inicial, sin la condición de que el resto de los conjuntos compartiesen la misma forma prima, agotando así todas las posibles combinaciones.

4. Se expandió la concepción de De Souza del espacio del diapasón descrito mediante coordenadas bidimensionales traste-cuerda a un espacio de tres dimensiones descrito por coordenadas [*cuerda, traste, tono*], con el fin de realizar los cálculos musicales necesarios para el funcionamiento de los aplicativos. Un espacio que tenga en cuenta la dimensión de tono no solo considera el espacio físico del diapasón, sino también la información musical contenida en él.

De igual forma, resulta valioso resaltar que en el desarrollo de las tecnologías se expusieron procedimientos de ideación propia para el cálculo de digitaciones y la conducción de voces entre acordes, mismos que incluyen implementaciones de algoritmos y fórmulas comúnmente empleadas en el ámbito computacional para el trabajo con listas o la comparación de documentos. Aunque no son innovadores en el campo técnico de la programación, las estrategias elaboradas representan enfoques originales para el uso de estos recursos en el desarrollo de aplicaciones enfocadas al campo musical.

Se espera que el trabajo realizado tenga un impacto significativo en las comunidades interesadas en teorizaciones postonales, el desarrollo de aplicaciones y la abstracción de las características físicas de un instrumento como la guitarra, en porciones de código que permitan realizar cálculos musicales. Por último, se expresa un profundo agradecimiento a todas las personas que colaboraron en este proyecto, ya sea de forma directa o indirecta, por su inestimable contribución y apoyo durante todo el proceso. Sin su colaboración, este trabajo no habría sido posible. Se les desea todo el éxito en sus futuros proyectos y emprendimientos, y se espera seguir colaborando en un futuro cercano.

# Bibliografía

*Albert Ayler Sheet Music.* (2011). Descargado de [http://www.ayler.co.uk/html/sheet\\_music.html](http://www.ayler.co.uk/html/sheet_music.html)

Alphonse, B. (1974). *The Invariance Matrix.* Yale University, 1974. Descargado de <https://books.google.com.mx/books?id=jco5AQAAIAAJ>

Anderson, I. (2006). *This Is Our Music: Free Jazz, the Sixties, and American Culture.* University of Pennsylvania Press. Descargado de <http://gen.lib.rus.ec/book/index.php?md5=4e804e90bde46f053c08efb848790ccf>

Baily, J., y Driver, P. (1992). Spatio-Motor Thinking in Playing Folk Blues Guitar. *The World of Music*, 34(3), 57–71. Descargado 2022-11-29, de <http://www.jstor.org/stable/43563264>

Best-Colleges Team. (2022, 5). *100 Free Resources For Teaching Yourself Guitar.* Descargado de <https://best-universities.net/resources/free-guitar-lessons/>

Boulez, P. (1963). *Penser la Musique Aujourd'hui.* Gallimard. Descargado de <http://gen.lib.rus.ec/book/index.php?md5=60388a4eabbe7501efea972adeed463b>

Clough, J., y Myerson, G. (1985). Variety and Multiplicity in Diatonic Systems. *Journal of Music Theory*, 29(2), 249–270. Descargado 2023-03-21, de <http://www.jstor.org/stable/843615>

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., y Stein, C. (2009). *Introduction to Algorithms*. MIT Press.
- De Souza, J. (2018, abril). Fretboard Transformations. *Journal of Music Theory*, 62(1), 1–39. Descargado 2022-06-16, de <https://read.dukeupress.edu/journal-of-music-theory/article/62/1/1/133949/Fretboard-Transformations> doi: 10.1215/00222909-4450624
- De Souza, J. (2022, enero). Guitar Thinking. *Soundboard Scholar*, 7(1). Descargado de <https://digitalcommons.du.edu/sbs/vol7/iss1/1>
- Discogs. (2022). *Discogs, Jazz Releases in 2022*. Descargado de <https://www.discogs.com/search/?type=release>
- Euler, L. (s.f.). *Tentamen Novae Theoriae Musicae 1739*. Descargado de [https://books.google.com.mx/books?id=N\\_E2zwEACAAJ](https://books.google.com.mx/books?id=N_E2zwEACAAJ)
- Feakes, J. (2018). *The Guitar Music of Ben Monder: A Physicalistic Analysis*. University of Western Australia.
- Forte, A. (1973). *The Structure of Atonal Music*. New Haven: Yale University Press.
- Galbraith, J. K. (1984). *The Anatomy of Power*. Houghton Mifflin.
- Gibson, J. (1966). *The Senses Considered as Perceptual Systems*. Boston: Houghton Mifflin.
- Goodrick, M., Miller, T., y Feist, J. (2012). *Creative Chordal Harmony for Guitar: Using Generic Modality Compression*. Boston, MA : Milwaukee, Wis: Berklee Press ; Distributed by Hal Leonard. (OCLC: ocn791641665)
- Graaf, D. P. d. (2017). *Beyond Borders : Broadening the Artistic Palette of (Composing) Improvisers in Jazz* (Tesis Doctoral, Leiden University). Descargado 2022-11-09, de <https://hdl.handle.net/1887/57415>

- GTProyectosMX. (2021). *Algoritmo de Heap*. Descargado 2023-03-21, de <https://www.youtube.com/watch?v=7cISHX4kbMY>
- Heap, B. (1963). Permutations by interchanges. *The Computer Journal*, 6(3), 293–298.
- INEGI. (2019). *Encuesta Nacional Sobre Disponibilidad y Uso de Tecnologías de la Información en los Hogares (ENDUTIH) 2019*. Descargado de [https://www.gob.mx/cms/uploads/attachment/file/534997/INEGI\\_SCT\\_IFT\\_ENDUTIH\\_2019.pdf](https://www.gob.mx/cms/uploads/attachment/file/534997/INEGI_SCT_IFT_ENDUTIH_2019.pdf)
- JazzHeaven. (2021). *Rudresh Mahanthappa: Intervalic Pitch-Cell Approaches (Edge)*. Descargado de <https://www.jazzheaven.com/yes/courses/rudresh-mahanthappa-intervalic-pitch-cell-approaches-edge/>
- John Coltrane Music Manuscript*. (2005). Descargado de <https://sova.si.edu/record/NMAH.AC.0903>
- Johnston, T. (2014). Chapter 3 - The Relational Paradigm: Mathematics. En T. Johnston (Ed.), *Bitemporal Data* (p. 35-41). Boston: Morgan Kaufmann. Descargado de <https://www.sciencedirect.com/science/article/pii/B9780124080676000036> doi: <https://doi.org/10.1016/B978-0-12-408067-6.00003-6>
- Kemp, S. (2022). *Digital 2022: Mexico - DataReportal Global Digital Insights*. Descargado de <https://datareportal.com/reports/digital-2022-mexico>
- Kingstone, A. (2006). *The Barry Harris Harmonic Method for Guitar*. Jazzworkshop Productions.
- Krantz, R., y Douthett, J. (2005, 01). Circular Distributions and Spectra Variations in Music: How Even Is Even?.
- Larsen, J. (2022). *Jazz Chords and the Best way to Think About Them*. Descargado

- 2023-03-17, de <https://www.youtube.com/watch?v=NlYS3TlIq-U>
- Levine, M. (2011). *The Jazz Theory Book*. Sher Music. Descargado de <https://books.google.com.mx/books?id=iyNQpJ4oaMcC>
- Lewin, D. (2007). *Generalized Musical Intervals and Transformations*. New York: Oxford University Press. Descargado de <http://www.ingentaconnect.com/content/oso/5481464/2007/00000001/00000001/art00000> doi: 10.1093/acprof:oso/9780195317138.001.0001
- Ligon, B. (2001). *Jazz Theory Resources: Tonal, Harmonic, Melodic, & Rhythmic Organization of Jazz* (n.º v. 2). Houston Pub. Descargado de <https://books.google.com.mx/books?id=7fzZzAEACAAJ>
- Martino, P., y Baruso, T. (1989). *Linear Expressions*. Place of publication not identified: Reh Book.
- Mary Halvorson - Interval Training Exercises on the Guitar*. (2020). Descargado de <https://www.soundslice.com/slices/tzZVc/>
- McLeod, J. (1994). *Tone Clock Theory Expanded: Chromatic Maps I & II*. Descargado 2023-03-24, de <http://archive.org/details/tonelock-B01095DS>
- Meehan, R. (2018, 4). *Drafts of Poetry and Scores at the 2016 Cecil Taylor Retrospective, 2016*. Descargado de <https://twitter.com/cryinmeehan/status/982133964218314753/photo/2>
- Messiaen, O. (1966). *Technique de mon Langage Musical*. A. Leduc. Descargado de <http://gen.lib.rus.ec/book/index.php?md5=7af0f53c36278391691ae7921ef1e829>
- Morris, R. (1987). *Composition with pitch-classes*. Yale University.
- Morris, R. (2007). *Mathematics and the Twelve-Tone System: Past, Present, and*

- Future. *Perspectives of New Music*, 45(2), 76–107. Descargado 2023-02-22, de <http://www.jstor.org/stable/25164658>
- Nattiez, J.-J. (2008, 01). Allen Forte's Set Theory, Neutral Level Analysis and Poietics.
- Nettles, B., Graf, R., y Berklee College of Music. (1997). *The Chord Scale Theory & Jazz Harmony*. Advance Music. Descargado de <https://books.google.com.mx/books?id=OpMyAAAACAAJ>
- O'Gallagher, J. (2015). *Twelve-Tone Improvisation: A Method for Using Tone Rows in Jazz*. Mel Bay Publications.
- Okazaki, M. (2015). *Fundamental of Guitar: A Workbook for Beginning, Intermediate, or Advanced Students*. Mel Bay Publications.
- Olson, P. (2006, 2). Ben Monder: Surprise from Cohesion. Descargado de <https://www.allaboutjazz.com/ben-monder-surprise-from-cohesion-ben-monder-by-paul-olson>
- Parsons, L. (2017). *The Avant-garde in Jazz as Representative of Late 20th Century American Art Music*. University of Florida.
- Pedroza, H. E., Meza, G., y Roman, I. R. (2022, diciembre). Egfxset: Electric Guitar Tones Processed Through Real Effects of Distortion, Modulation, Delay and Reverb. En *Proceedings of the First MiniCon Conference*. Descargado 2023-01-16, de [https://ismir2022program.ismir.net/lbd\\_367.html](https://ismir2022program.ismir.net/lbd_367.html)
- Philippe Carles, J.-L. C. (1973). *Free Jazz. Black Power*. Einaudi. Descargado de <http://gen.lib.rus.ec/book/index.php?md5=946afaabaf4c76f372eaa720b7ac3fb4>
- Quintanilla, M. . (1989). *Sistemas de información para la gestión empresarial*. McGraw-Hill Interamericana.

- Rahn, J. (1987). *Basic Atonal Theory*. Schirmer Books. Descargado de <https://books.google.com.mx/books?id=ZJEYAQAAIAAJ>
- Rawlins, R., Bahha, N., y Tagliarino, B. (2005). *Jazzology: The Encyclopedia of Jazz Theory for All Musicians*. Hal Leonard. Descargado de <https://books.google.com.mx/books?id=WWhjprIWnwoC>
- Robie, J., y Trexcel Research. (1998). *What is the Document Object Model?* Descargado 2023-02-23, de [https://www.w3.org/TR/W3-DOM/introduction.html#:~:text=The%20Document%20Object%20Model%20\(DOM,document%20is%20accessed%20and%20manipulated](https://www.w3.org/TR/W3-DOM/introduction.html#:~:text=The%20Document%20Object%20Model%20(DOM,document%20is%20accessed%20and%20manipulated).
- Robinson, J. (2005, 9). The Challenge of the Changing Same: The Jazz Avant-Garde of the 1960s, the Black Aesthetic and the Black Arts Movement. *Critical Studies in Improvisation / Études Critiques en Improvisation*, 1(2). Descargado de <http://dx.doi.org/10.21083/csieci.v1i2.17> doi: 10.21083/csieci.v1i2.17
- Roig-Francoli, M. A. (2008). *Understanding Post-Tonal Music*. Boston: McGraw-Hill.
- Saia, S. M., Nelson, N. G., Young, S. N., Parham, S., y Vandegrift, M. (2022). Ten Simple Rules for Researchers Who Want to Develop Web Apps. *PLOS Computational Biology*, 18(1), e1009663. Descargado 2022-11-30, de <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1009663> doi: 10.1371/journal.pcbi.1009663
- Schat, P. (1993). *The Tone Clock* (n.º v. 7). Chur, Switzerland ; Langhorne, Pa: Harwood Academic Publishers.
- Schiano, M. J. (1999). The Role of Musical Set Theory in Undergraduate Education. *College Music Symposium*, 39, 64–86. Descargado 2023-02-13, de <http://www.jstor.org/stable/40374570>



- School of Composition. (2022, 10). *60+ Best Music Theory Composition YouTube Channels*. Descargado de <https://www.schoolofcomposition.com/music-theory-composition-youtube-channels/>
- Schuijjer, M. (2008). *Analyzing Atonal Music: Pitch-Class Set Theory and Its Contexts* (NED - New edition ed.). Boydell Brewer. Descargado 2023-02-13, de <http://www.jstor.org/stable/10.7722/j.ctt14brsj5>
- Schuller, G. (1961). *A Collection of the Compositions of Ornette Coleman*. New York: MJQ Music Inc.
- Sedgewick, R. (1977). Permutation Generation Methods. *ACM Computing Surveys (CSUR)*, 9(2), 137–164. doi: 10.1145/356689.356692
- Shipton, A. (2007). *A New History of Jazz*. Continuum. Descargado de <https://books.google.com.mx/books?id=XyYJAQAAMAAJ>
- Shklar, L. A., y Rosen, R. (2003). Web application architecture: Principles, protocols and practices..
- Singhal, A. (2001). Modern Information Retrieval: A Brief Overview. *IEEE Data Eng. Bull.*, 24(4), 35-43. Descargado de <http://dblp.uni-trier.de/db/journals/debu/debu24.html#Singhal01>
- Sorzano, C., Vargas, J., y Montano, A. (2014). A Survey of Dimensionality Reduction Techniques.
- Spitzer, P. (2001). *Jazz Theory Handbook*. Mel Bay. Descargado de <https://books.google.com.mx/books?id=rbC8oiKLY-oC>
- Straus, J. N. (2005). *Introduction to Post-Tonal Theory* (3rd ed ed.). Upper Saddle River, N.J: Prentice Hall.
- Sun Ra Arkestra 1976 Montreux, Switzerland TV*. (2020, 10). Descargado de

[https://www.youtube.com/watch?v=Y\\_6h6LNxen4](https://www.youtube.com/watch?v=Y_6h6LNxen4)

Terefenko, D. (2014). *Jazz Theory: from Basic to Advanced Study*. New York, NY ; London: Routledge.

Thorgersen, K., y Zandén, O. (2014). The Internet as Teacher. *Journal of Music, Technology and Education*, 7(2), 233. Descargado 2022-11-30, de [https://www.academia.edu/11366807/The\\_Internet\\_as\\_Teacher](https://www.academia.edu/11366807/The_Internet_as_Teacher)

Tonebase Guitar. (2018, 12). *Leo Brouwer Teaches Estudios Sencillos*. Descargado de <https://www.youtube.com/watch?v=2IWN8NWf1ZA>

Travaglini, S. (2020). *Neverland 6, Avening Angel - 2011*. Descargado de <https://www.stefanotravaglini.com/wp-content/uploads/2020/03/neverland.pdf>

Tri-Centric Foundation. (2018). *Anthony Braxton Catalog*. Descargado de <https://tricentricfoundation.org/scores>

Willmott, B. (1994). *Mel Bay's Complete Book of Harmony, Theory & Voicing*. Pacific, MO: Mel Bay.

Yamaguchi, M., y Lovell, J. (2006). *The Complete Thesaurus of Musical Scales*. (OCLC: 71295781)