

# **EL USO DE ONTOLOGÍAS COMO TECNOLOGÍA PARA EL ANÁLISIS DE COMPOSICIONES MUSICALES EN FORMATO MUSICXML.**

**JUAN DIEGO ESTRADA PÉREZ**

Proyecto de grado para optar al título de  
Ingeniero de Sistemas

## **Asesor**

Juan Guillermo Lalinde Pulido  
Doctor Ingeniero en Telecomunicaciones

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS  
ESCUELA DE INGENIERÍA  
UNIVERSIDAD EAFIT  
MEDELLÍN  
2.011

Nota de Aceptación

---

---

---

---

Juan Guillermo Lalinde Pulido

---

Doctor Fernando Gil Araque

---

Ingeniero Diego Fernando Gómez

Medellín, 25 de Noviembre de 2011

## **AGRADECIMIENTOS**

A mis padres Alfonso Estrada Araque y Ximena Pérez Murillo y mis hermanos Andrés Estrada Pérez y Carolina Estrada Pérez por su apoyo incondicional. Siempre confiaron en mis capacidades y aptitudes para abordar este proyecto. A mi novia Luisa Fernanda Pérez, muchas gracias por su gran aporte en la elaboración de la base de conocimiento de aspectos teóricos musicales, al Doctor Juan Guillermo Lalinde Pulido por saber encaminar el proyecto. A todos mis amigos y familiares que de alguna manera aportaron a este gran logro.

## TABLA DE CONTENIDO

AGRADECIMIENTOS .....	4
TABLA DE CONTENIDO .....	5
TABLA DE FIGURAS.....	8
GLOSARIO .....	9
RESUMEN .....	10
INTRODUCCIÓN .....	11
1. ALCANCE.....	13
2. OBJETIVOS .....	14
2.1 OBJETIVO GENERAL.....	14
2.2 OBJETIVOS ESPECÍFICOS.....	14
3. ESTADO DEL ARTE .....	15
4. MARCO TEÓRICO.....	23
4.1 ONTOLOGÍA.....	23
4.1.1 Definición de ontología.....	23
4.1.2 Tipos de ontologías .....	25
4.1.3 Criterios de diseño .....	26
4.2 WEB SEMÁNTICA.....	27
4.2.1 Definición de Web semántica.....	27
4.2.2 Componentes de la Web semántica .....	28
4.3 LAS NOTAS.....	35
4.3.1 Definición de Sonido .....	35
4.3.2 Distancia entre dos notas.....	36
4.3.3 Tipos de Intervalos .....	37
4.3.4 Alteraciones y semitonos .....	38
4.4 NOTACIÓN DE LAS ALTURAS.....	39

4.4.1	El pentagrama .....	39
4.4.2	La clave Sol .....	39
4.4.3	La clave Fa .....	40
4.5	LAS TONALIDADES .....	40
4.5.1	Modulación .....	41
4.5.2	Funciones Toniales .....	42
4.6	LAS ESCALAS .....	42
4.6.1	La escala diatónica mayor .....	43
4.8	MUSICXML .....	44
4.8.1	Formas de estructurar archivos MusicXML .....	44
4.8.2	“Hola Mundo” en MusicXML .....	46
5.	TECNOLOGÍAS .....	52
5.1	PROTÉGÉ .....	52
5.2	ECLIPSE .....	52
5.3	OWL API .....	52
5.4	HERMIT .....	53
5.5	PROXYMUSIC .....	53
6.	ONTOLOGÍA MUSICAL .....	54
6.1	DEFINICIÓN DE CLASES .....	54
6.1.1	Distance .....	54
6.1.2	Interval .....	56
6.1.3	MComposition .....	57
6.1.4	Nota .....	58
6.2	PROPIEDADES .....	58
6.2.1	hasDistance .....	58
6.2.2	hasInterval .....	59
6.2.3	hasNote .....	59
6.2.4	hasNoteOne .....	59
6.2.5	hasNoteTwo .....	59
6.2.6	hasTonic .....	59

6.2.7 isIntervalOf.....	59
6.2.8 isNoteOf.....	60
6.2.9 isNoteOneOf.....	60
6.2.10 isNoteTwoOf.....	60
7. PRUEBA DE CONCEPTO .....	61
8. CONCLUSIÓN.....	65
9. TRABAJO FUTURO.....	66
BIBLIOGRAFÍA.....	67
CIBERGRAFÍA.....	68

## TABLA DE FIGURAS

Figura 1. Push Button Bertha. ....	16
Figura 2. Estructura conceptual de las Capas de MX.....	21
Figura 3. Representación y clasificación de los recursos musicales en la ontología. .....	22
Figura 4. Representación del dominio del vino mediante clases, instancias y relaciones entre ellas. ....	24
Figura 5. La Pila de la Web semántica. ....	29
Figura 6. Un Grafo RDF Describiendo a Eric Miller .....	31
Figura 7. Distancia entre dos notas. ....	37
Figura 8 Intervalos (música).....	38
Figura 9. Alteraciones y semitonos. ....	38
Figura 10. El pentagrama. ....	39
Figura 11. Clave de sol. ....	40
Figura 12. Clave de fa. ....	40
Figura 13. Las Tonalidades. ....	41
Figura 14. Escala diatónica mayor.....	43
Figura 15. Escala de Sol mayor.....	44
Figura 16. Do en la escala de Do mayor. ....	46
Figura 17. Representación en MusicXML de la nota Do en la escala de Do mayor. .....	47
Figura 18. Definición de la clase Distance.....	54
Figura 19. Individuo C-D. Representa la distancia entre .....	56
las notas Do y Re en la escala Do mayor.....	56
Figura 20. Representación de una composición en Do Mayor mediante la ontología. ....	58
Figura 21. Ejecutando la prueba de concepto desde consola.....	61
Figura 22. Resultado de la prueba de concepto desde consola.....	63
Figura 23. Partitura de la composición Arroz con leche. ....	64

## GLOSARIO

**Modelo de datos jerárquico:** Es un modelo de datos en el cual éstos son organizados en una estructura parecida a la de un árbol. Esta estructura permite representar información usando la relación padre/hijo: cada padre puede tener muchos hijos, pero cada hijo tiene un solo padre.

**Modelo de datos relacional:** Es un modelo de datos basado en la lógica de predicados y en la teoría de conjuntos. Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente.

**Código abierto:** Software de computación disponible gratuitamente y que puede ser redistribuido con o sin modificación.

**Tesauro:** Listado de palabras o términos empleados para representar conceptos.

**Hipertexto:** Texto que contiene elementos a partir de los cuales se puede acceder a otra información.

**Sonología:** Es la relación entre las funciones de los sonidos en las composiciones y su representación acústica.

**Interfaz de línea de comandos:** Es un mecanismo para interactuar con un sistema operativo o software tecleando comandos para ejecutar tareas específicas.

**Descriptores lógicos:** Es una familia de lenguajes formales de representación de conocimiento. Es una herramienta de formalismo lógico utilizado en ontologías y la Web Semántica.

## **RESUMEN**

Este trabajo presenta la propuesta del uso de ontologías como herramienta para el análisis de la tonalidad musical de canciones infantiles, escritas en tonalidades mayores y con un carácter melódico, mediante el desarrollo de una ontología básica de prueba en OWL.

## INTRODUCCIÓN

Hace ya varias décadas el aporte de la Inteligencia artificial a la música se ha manifestado en las tareas interpretación, composición, teoría musical y procesamiento de sonido digital mediante el uso de computadores<sup>1</sup>.

En un principio, se utilizaban para contar el número de ocurrencias de un intervalo o una nota musical en determinadas composiciones, pero pronto se dieron cuenta de las limitaciones de este tipo de programas y de las posibilidades que podrían abrirse si se desarrollaban sistemas basados en el conocimiento, que permitieran no sólo identificar patrones de datos sino también su análisis<sup>2</sup>.

Este trabajo de grado titulado “El uso de ontologías como tecnología para el análisis de composiciones musicales en formato MusicXML”, integra la ingeniería de Sistemas con el análisis musical, mediante el desarrollo de una ontología básica de prueba, que a futuro pueda ser ampliada y usada por los músicos como una herramienta para el análisis armónico, melódico y rítmico de diferentes tipos de composiciones.

Para lograr este proyecto se desarrolló una ontología musical en OWL (Ontology Web Language) que pudiera inferir la tonalidad de piezas musicales infantiles en formato MusicXML.

En la primera parte se revisan los principales trabajos a partir de la unión entre inteligencia artificial y música. En la segunda parte, se definen los conceptos relativos a las ontologías y a la Web semántica, así como los conceptos musicales de sonido, intervalos, alteraciones, notación musical, tonalidad, modulación,

---

<sup>1</sup> Curtis, Roads. Research in Music and Artificial Intelligence. En: Computing Surveys, Junio de 1985, Vol. 17, No. 2, p. 163. Disponible en: <http://students.cs.byu.edu/~cs470ta/articles/p163-roads.pdf> Consultado: Marzo 2011

<sup>2</sup> Ibíd., p. 163.

funciones tonales y escalas. A continuación, se explica qué es MusicXML y su estructura; se mencionan las tecnologías involucradas en el proyecto, y finalmente, se presenta la ontología musical y la prueba de concepto realizada.

## 1. ALCANCE

Las ontologías permiten la formalización de un dominio de interés mediante la definición de clases, propiedades y las relaciones entre las clases. El desarrollo más reciente en lenguajes de ontologías estándares es OWL, del *World Wide Web Consortium (W3C)*, el cual permite no sólo definir, sino publicar y compartir ontologías en la Web. Este lenguaje ha sido diseñado para capturar el conocimiento, de manera que permita a las aplicaciones entender mejor los recursos de acceso vía Web, y que los utilicen de forma más inteligente.

*MusicXML* permite la representación en formato XML de composiciones musicales y en la actualidad existen nuevas tecnologías que pueden ser utilizadas para realizar análisis de composiciones musicales bajo *MusicXML*, por lo que en esta propuesta se plantea el análisis musical por medio del uso de ontologías. Para demostrar la anterior afirmación se pretende desarrollar una ontología básica de prueba en OWL, que permita analizar composiciones musicales infiriendo su tonalidad bajo unos criterios de teoría musical seleccionados. Las piezas musicales que serán tenidas en cuenta para el análisis son canciones infantiles, compuestas en tonalidades mayores, escritas en un solo pentagrama y con carácter melódico únicamente.

Por último, va a ser desarrollada una prueba de concepto, por medio del programa que utilizará la ontología para realizar el análisis musical.

## **2. OBJETIVOS**

### **2.1 OBJETIVO GENERAL**

Demostrar que las ontologías sirven como herramienta para el análisis de melodías de canciones infantiles en formato *MusicXML*.

### **2.2 OBJETIVOS ESPECÍFICOS**

- Construir una ontología en OWL para representar conocimiento en el área de la música, en los aspectos de tonalidad, escalas e intervalos.
- Desarrollar una prueba de concepto que utilice la ontología musical.

### 3. ESTADO DEL ARTE

Desde hace ya varias décadas, la música y la tecnología han estado relacionadas, especialmente con el uso de dispositivos electrónicos y software que facilitan la reproducción, grabación, composición, almacenamiento y rendimiento. Un ejemplo de esto, para George Yúdice es que:

Las nuevas tecnologías facilitan varios *modus operandi* de grabación y producción musical que prescinden de los grandes estudios fonográficos: desde las prácticas “hazlo tu mismo” (*do it yourself* o DIY) y “cortar y pegar” (*cut’n paste*) – características del punk, del bricolaje vernáculo y de movimientos artísticos previos que usaron el *collage* – hasta los estudios de grabación caseros, cada vez más sofisticados<sup>3</sup>.

Además, la inteligencia artificial ha realizado un gran aporte a la música en las últimas décadas: tareas musicales que anteriormente eran llevadas a cabo por personas (especialmente músicos), son realizadas por “sistemas musicalmente inteligentes”. Las investigaciones más recientes incluyen composición musical, interpretación, teoría musical y procesamiento de sonido digital<sup>4</sup>.

Por solo abordar algunos de los primeros ejemplos de la composición asistida por computador, se puede mencionar el programa para componer melodías “Tin Pan Allen”, creado por la Compañía de Martin Klein y Douglas Bolitho Burroughs, llamado DATATRON. Este programa funcionaba a partir de unas notas aleatorias, ingresadas por medio de dígitos, que luego generaban una muestra mayor que representaba las ocho notas de la escala diatónica; y de ahí, el programa recogía

---

<sup>3</sup> Yúdice, George. Nuevas tecnologías, música y experiencia. Gedisa Editorial, Barcelona, 2007. Consultado: Junio 2011

<sup>4</sup> Curtis, Roads. Research in Music and Artificial Intelligence. En: Computing Surveys, Junio de 1985, Vol. 17, No. 2, p. 163. Disponible en: <http://students.cs.byu.edu/~cs470ta/articles/p163-roads.pdf> Consultado: Marzo 2011

unas notas al azar, creando una composición musical<sup>5</sup>.

Uno de los resultados de esta investigación fue la melodía *Push Button Bertha*, con texto del compositor Jack Owens.

ElectroData Division of Burroughs

## "PUSH BUTTON BERTHA"

LYRIC BY JACK OWENS MUSIC BY DATATRON  
A.S.C.A.P. MATHEMATICIANS  
DR. MARTIN KLEIN  
DR. DOUGLAS BOLITHO

2 2 2 2 2 2 2 2 2 2  
 0 0 0 0 7 1 9 2 7  
 RANDOM NUMBERS  
 USED FOR MUSIC.

MODERATE BRIGHT BRONZE

SHE'S PUSH - BUT-TON BER-THA - SWEET MA-CHINE - WHAT A QUEEN  
 CAL-CU - LAT - IN' DAL - DI - TA - TIN' CHICK - WITH A  
 CLICK - MY PUSH - BUT-TON BER-THA - NOT TOO LARGE - WHAT A CHARGE  
 E - LEC - TRON - IC SU - PER - SON - IC FRIEND - THE  
 END - ONCE SHE'S OP - ER - A - TIN' WATCH HER ROCK AND ROLL -  
 BER-THAS NOT DE - MAND - ING - NEV - ER WANTS YOUR DOUGH -  
 COOL AND CAL - CU - LA - TIN' THIS GAL HAS NO HEART OR SOUL - SHE'S  
 AL - WAYS UN - DER - STAND - ING - JUST FLIP A SWITCH AND SHE'LL GO -  
 PUSH - BUT-TON BER-THA - AU - TO - MA - TION DI - VINE  
 NOW HEAR THIS - SHE CAN'T KISS  
 PAY THE LIGHT BILL AND YOU'RE RIGHT SHE'S MINE - ALL MINE -  
 TEN WEIGHT OIL - MAKES HER LOY - AL DREAM - MA - CHINE -  
©1956 BY OWENS-KEMP MUSIC CO. - (ASCAP) 516 AMALFEI DR. PARIS PAUSADES, S.B.V.E.

Figura 1. Push Button Bertha. <sup>6</sup>

Otra aplicación en esta vertiente fue la realizada por Lejaren Hiller y Leonard Issacson, pioneros en trabajar con algoritmos musicales, en la Universidad de Illinois, en 1956, bajo la premisa de que la música era un proceso algorítmico. El

<sup>5</sup> Ames, Charles, "Automated composition in retrospect: 1956-1986, *Leonard. edición especial: Artes visuales, sonido, música y tecnología*, Vol. 20, No. 2, Massachusetts Institute of Technology (MIT), Massachusetts, 1987, p. 170. Disponible en: <http://www.jstor.org/stable/3679501> Consultado: Octubre 2011

<sup>6</sup> Push Button Bertha, Musica por Datatron, Letra de Jack Owens. Dominio público. Disponible en: <http://datatron.blogspot.com/2009/03/bertha-preserves-datatron.html> Consultado: Octubre 2011

resultado fue la presentación en concierto de algunas obras y la posterior aparición de la *Suite Illiac*, para cuarteto de cuerdas<sup>7</sup>.

En cuanto a la vertiente del sonido digital, los ingenieros de sonido se han visto asistidos por herramientas y programas de computador que les permiten ver las ondas gráficamente, manipularlas, e inclusive pegar fragmentos de otras ondas en ellas. También, la edición digital inteligente permite el reconocimiento automático de la estructura musical, el análisis de líneas melódicas individuales en una textura polifónica y el cambio de notas, duración, amplitud, articulación y el timbre, etc.

A continuación van a presentarse los problemas en el campo de la teoría musical, campo en el cual se apoya este proyecto, y que han sido susceptibles a soluciones ofrecidas por el campo de inteligencia artificial.

Alrededor de la década de los setenta, muchos de los musicólogos comenzaron a utilizar los computadores para sus trabajos teóricos. Además, en sus investigaciones predominaban técnicas estadísticas para realizar análisis de aspectos teóricos en composiciones musicales. Por ejemplo, algunos estudios contaban el número de *Do* sostenidos en una composición o cada ocurrencia de un *intervalo de tercera menor*, con el fin de caracterizar su estilo musical. Pero se dieron cuenta que estos programas eran muy limitados desde el punto de vista de la construcción teórica y que podría ser más útil si el computador no sólo encontraba instancias de estructuras musicales, sino que también pudiera realizar las conexiones entre éstas, para construir estructuras internas de conocimiento que permitieran entender la estructura musical<sup>8</sup>.

Es por este motivo que se comenzaron a desarrollar sistemas basados en el conocimiento, en los cuales la búsqueda de patrones de los datos de entrada era

---

<sup>7</sup> *Ibíd.*, p. 170

<sup>8</sup> *Ibíd.*, p. 168.

asistida por un dominio específico de conocimiento, que era usado para organizar la búsqueda, analizarla y realizar procesos de inferencia<sup>9</sup>.

En este campo han sido de vital importancia los aportes de Terry Winograd, quien ha asumido el lenguaje como un conjunto (finito o infinito) de frases, cada una de las cuales es finita y construida a partir de un número finito de elementos<sup>10</sup>.

Esta definición, entonces, le ha servido a Winograd para buscar la relación entre música y lenguaje, relación que ha sido objeto de discusión de varios teóricos, y sobre la que se han realizado algunos estudios que comienzan a usar ideas básicas de la lingüística para expresar la estructura musical.

El autor, en base a los teóricos musicales, establece que la teoría musical contiene una serie de reglas estructurales específicas (sintaxis) que rigen una gran cantidad de tipos de música, y una “gramática generativa” puede proporcionar una manera útil y ordenada de expresar dichas reglas<sup>11</sup>. Un ejemplo de este tipo fueron los programas que realizaban análisis armónicos, como el desarrollado en la década de los ochenta, por Terry Winograd, quien adaptó libremente la gramática musical, de manera tal que pudiera adecuarse a peculiaridades de la estructura armónica y la dividió en tres rangos simplificados: tonalidad, acordes y notas. Luego pasó a desarrollar el algoritmo de análisis en LISP, un lenguaje de programación que se compone de varias funciones individuales, en la que se pueden ingresar una serie de entradas para calcular valores<sup>12</sup>. El resultado de este programa fue un excelente desempeño en composiciones simples, y uno relativamente bueno en composiciones más complejas<sup>13</sup>.

---

<sup>9</sup> Ibid., p. 168.

<sup>10</sup> Winograd, Terry, “Linguistics and the Computer Analysis of Tonal Harmony”, *Journal of Music Theory*, Vol. 12, No. 1, Departamento de Música de la Universidad de Yale, New Haven, 1968, p. 3. Disponible en: <http://www.jstor.org/stable/842885> Consultado: Octubre 2011

<sup>11</sup> Ibid., p. 4.

<sup>12</sup> Ibid., p. 33.

<sup>13</sup> Ibid., p. 36.

Entre otros programas de este tipo de análisis, se destacan el SLAM (a Simple Language for the Analysis of Music), desarrollado por T. G. Whitney en 1975, y que permitía al estudiante obtener análisis de composiciones que habían sido previamente codificadas y almacenadas en el computador<sup>14</sup>; el de Phelps y W. Poland, que identificaba notas cromáticas y modulaciones en música tonal; y el de Fred T. Hofstetter, que catalogaba patrones melódicos y armónicos<sup>15</sup>.

En la década de 1970, O. Laske comenzó a trabajar en una teoría de la cognición musical basada en la psicología de procesamiento de información y modelos de gramática generativa de A. Newell, H. Simon, G. Miller y N. Chomsky. Entre los años de 1972-1973, Laske llegó a la conclusión de que un programa de reconocimiento musical tenía que articularse desde el proceso de la escucha. El diseño incluía un bosquejo de un analizador acústico que extraía un conjunto de características para luego analizar desde la sonología<sup>16</sup> y determinar las posibles funciones musicales del objeto de sonido extraído<sup>17</sup>.

En la década de 1980, en el Centro de Experimentación de Música de la Universidad de California (San Diego), un nuevo esfuerzo se dirigió a un enfoque cognitivo para determinadas áreas de la música. Pueden mencionarse dos proyectos: el llevado a cabo por D. Deutsch, mediante el uso de un equipo de síntesis de sonido en VAX (sistema informático), y que gira en torno a la capacidad de memoria de los oyentes para tonos de complejidad espectral compositiva; el segundo proyecto, el de los científicos D. Norman y D. Rumelhart,

---

<sup>14</sup> Lamb, Martin Robert, *The computer as musicianship teaching aid*, Tesis doctoral en Ingeniería Eléctrica de la Universidad de Canterbury, Nueva Zelanda, 1979, p. 16. Disponible en: [http://ir.canterbury.ac.nz/bitstream/10092/3997/1/Thesis\\_fulltext.pdf](http://ir.canterbury.ac.nz/bitstream/10092/3997/1/Thesis_fulltext.pdf) Consultado: Octubre 2011

<sup>15</sup> *Ibíd.*, p. 16.

<sup>16</sup> La Sonología es la relación entre las funciones de los sonidos en las composiciones y su representación acústica.

<sup>17</sup> Roads, C., "Artificial intelligence and music", *Computer Music Journal*, Vol. 4, No. 2, Massachusetts Institute of Technology (MIT), Massachusetts, 1980, P. 17. Disponible en: <http://www.jstor.org/stable/3680079> Consultado: Octubre 2011.

que se basaba en la visión de la música como un dominio de vasto conocimiento<sup>18</sup>.

En el texto *Automating Motivic Analysis through the Application of Perceptual Rules*<sup>19</sup>, de O. Lartillot y E. Saint-James, se propuso un nuevo enfoque para el descubrimiento de patrones musicales, basado en un modelo de los mecanismos de percepción musical, desarrollado en una librería *OpenMusic*, llamada OMkanthus, que analiza archivos MIDI. Básicamente, los resultados que encontraron fue que la estructura permitía reconocer patrones básicos que eran también reconocidos por los oyentes, pero que con algunas mejoras en el algoritmo, se podrían descubrir patrones más sutiles que no eran fácilmente discriminados por los oyentes<sup>20</sup>.

Por otro lado, en el Laboratorio de Informática Musical, de la Universidad degli Studi di Milano (Milán, Italia), el grupo *XML Music Application*, desarrolló una instancia de XML que integrara diferentes aspectos de la representación musical, mediante un lenguaje que fuera una meta representación de la información musical (propiedades básicas de las notas tales como duración, tono, timbre, etc.) para describir y procesarla en un ambiente multicapas (ver figura 2)<sup>21</sup>.

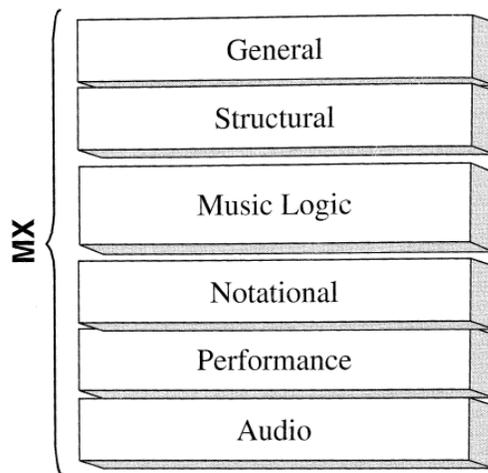
---

<sup>18</sup> *Ibíd.*, p. 20.

<sup>19</sup> Lartillot, Olivier y Saint-James, Emmanuel, "Automating Motivic Analysis through the Application of Perceptual Rules", *Computing in Musicology*, Vol. 13, Massachusetts Institute of Technology (MIT), Massachusetts, 2004, pp. 73-92. Disponible en: [http://jyu.academia.edu/OlivierLartillot/Papers/299275/Automating\\_Motivic\\_Analysis\\_Through\\_the\\_Application\\_of\\_Perceptual\\_Rules](http://jyu.academia.edu/OlivierLartillot/Papers/299275/Automating_Motivic_Analysis_Through_the_Application_of_Perceptual_Rules) Consultado: Octubre 2011.

<sup>20</sup> *Ibíd.*, p. 86

<sup>21</sup> Haus, Goffredo y Longari, Maurizio, "A Multi-Layered, Time-Based Music Description Approach Based on XML", *Computer Music Journal*, Vol. 29, No. 1, Massachusetts Institute of Technology (MIT), Massachusetts, 2005, pp. 70-85, Disponible en: <http://www.jstor.org/stable/3681878> Consultado: Octubre 2011.



**Figura 2. Estructura conceptual de las Capas de MX<sup>22</sup>**

En este mismo sentido, Alfio Ferrara, Luca A. Ludovico, Stefano Montanelli, Silvana Castano y Goffredo Haus, también de la Universidad degli Studi di Milano, trabajaron, desde el 2006, en una descripción multidimensional de recursos musicales, en el sentido semántico, sobre la noción de contexto y género musical.

El propósito era definir una ontología para describir metadatos musicales, que cumpliera con los siguientes requisitos: 1) separar la información sobre el contexto y la clasificación de género; 2) expresar adecuadamente las relaciones entre las características musicales; 3) modelar adecuadamente la clasificación de género<sup>23</sup>.

La siguiente imagen muestra la representación y clasificación de los recursos musicales en esta ontología.

<sup>22</sup> *Ibíd.*, p. 73.

<sup>23</sup> Ferrara, Alfio; Ludovico, Luca A.; Montanelli, Stefano; Castano Silvana y Haus, Goffredo, "A Semantic Web Ontology for Context-based Classification and Retrieval of Music Resources", *ACM Transactions on Multimedia Computing, Communications and Applications*, Vol. V, abril de 2006, pp. 1–25. Consultado: Octubre 2011

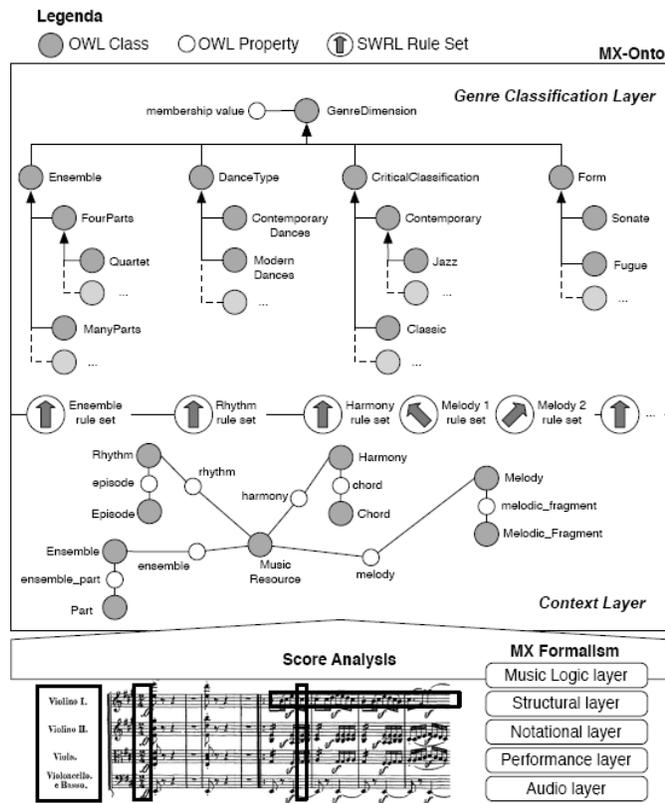


Figura 3. Representación y clasificación de los recursos musicales en la ontología<sup>24</sup>.

Como se ha visto tras la revisión de la literatura escrita sobre inteligencia artificial y análisis musical, existen diversas posibilidades de apoyo para la música desde la ingeniería de sistemas, además de que se pueden integrar otros sectores de la música, como la composición y la interpretación.

La ventaja de trabajar en *MusicXML* es la portabilidad, debido a que representa la notación musical tradicional de occidente desde el S. XVII y que permite la distribución de partituras en línea. Además, es de distribución gratuita y soportado por más de 150 aplicaciones, haciéndolo de más fácil acceso para desarrolladores de aplicaciones y para músicos.

<sup>24</sup> Ibíd. p. 8.

## 4. MARCO TEÓRICO

### 4.1 ONTOLOGÍA

#### 4.1.1 Definición de ontología<sup>25</sup>

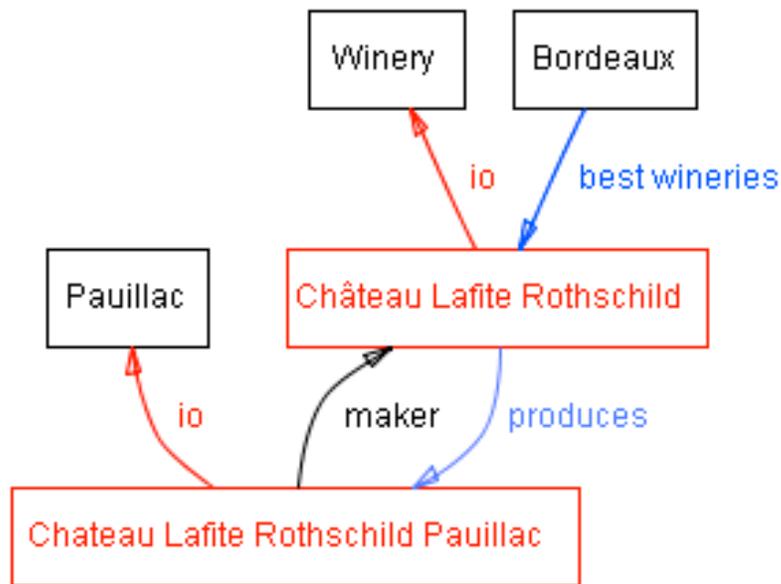
Según Tom Gruber una ontología, en ciencias de la computación e información, define un conjunto de primitivas de representación con el cual se modela un dominio de conocimiento, real o imaginado. En otras palabras, es un término técnico que denota un artefacto diseñado para la modelación de conocimientos sobre determinado dominio. Además, debe cumplir el siguiente conjunto de primitivas de representación:

- Clases o conjuntos. Los miembros (instancias) de la clase comparten algunas propiedades o características.
- Atributos o propiedades que cada miembro de la clase pueda tener.
- Relaciones entre los miembros de las clases.

A continuación se ilustran algunas clases, instancias y las relaciones entre éstas, bajo el dominio del Vino. Las clases están en color negro y las instancias en rojo. Los enlaces directos representan las propiedades entre las clases y las que tienen la etiqueta “*io*” (*instance of*) define las instancias.

---

<sup>25</sup> Gruber, Tom. “Ontology”, En: Ling Liu and M. Tamer Özsu (Eds.), Encyclopedia of Database Systems, Springer-Verlag, 2009. Disponible en: <http://tomgruber.org/writing/ontology-definition-2007.htm> Consultado: Enero 2011.



**Figura 4. Representación del dominio del vino mediante clases, instancias y relaciones entre ellas.<sup>26</sup>**

Las ontologías se expresan usualmente en un lenguaje basado en la lógica, de manera que se puedan establecer diferencias detalladas, precisas, coherentes, racionales, y sobre todo, significativas, entre las clases, propiedades y relaciones<sup>27</sup>.

Una ontología también puede ser vista como un nivel de abstracción de un modelo de datos, análogo a los modelos jerárquico y relacional, pero con el propósito de modelar conocimiento sobre individuos, sus atributos y las relaciones con otros individuos.

<sup>26</sup> Noy, Natalya; McGuinness, Deborah. "Ontology Development 101: A Guide to Creating Your First Ontology", Stanford University. Disponible en: [http://protege.stanford.edu/publications/ontology\\_development/ontology101.pdf](http://protege.stanford.edu/publications/ontology_development/ontology101.pdf) Consultado: Noviembre 2010.

<sup>27</sup> "OWL Web Ontology Language Use Cases and Requirements", W3C Recommendation. Disponible en: <http://www.w3.org/TR/2004/REC-webont-req-20040210/> Consultado: Agosto 2010.

Las ontologías son normalmente especificadas en lenguajes que permiten abstracción independiente de la estructura de datos e implementación. En la práctica, los lenguajes de ontologías están cerca, en su poder expresivo, a la lógica de primer orden, como se dijo previamente. Por esta razón, se dice que las ontologías están en el nivel “semántico”, mientras que por ejemplo, el esquema de una base de datos, son modelos de datos a nivel “lógico” o “físico”. Debido a su independencia de los niveles inferiores, las ontologías se utilizan para la integración de bases de datos heterogéneas, permitiendo interoperabilidad entre sistemas dispares, y la especificación de interfaces de servicios independientes, basadas en el conocimiento.

#### 4.1.2 Tipos de ontologías<sup>28</sup>

Gruber clasifica las Ontologías en las siguientes categorías, de acuerdo al nivel de dependencia a un dominio o tarea en particular:

- **Ontologías de alto nivel:** Describen conceptos muy generales, que son independientes de un dominio o problema en particular.
- **Ontologías de Dominio:** Describen el vocabulario relacionado a un dominio genérico (como la medicina o los automóviles).
- **Ontologías de tareas:** Describen el vocabulario relacionado a una tarea o actividad genérica, por medio de la especialización de términos introducidos por una ontología de alto nivel.

---

<sup>28</sup> Guariola, Nicola. “Formal Ontology and Information Systems”, National Research Council, LADSEB-CNR. Disponible en: <http://www.iao-cnr.it/Papers/FOIS98.pdf> Consultado: Noviembre 2010.

- **Ontologías de Aplicación:** Describen conceptos que dependen de un dominio o tarea en particular. Estos conceptos a menudo corresponden a roles ejecutados por entidades del dominio mientras realizan cierta actividad.

#### 4.1.3 Criterios de diseño<sup>29</sup>

Al preguntarse cómo representar algún dominio en una ontología, se están tomando decisiones de diseño. Para orientar y evaluar el diseño, se necesitan criterios objetivos basados en los efectos del artefacto resultante, en lugar de basarse en alguna noción de verdad. A continuación se describen los criterios de diseño para ontologías, según Gruber:

- **Claridad:** Una ontología debe comunicar efectivamente el significado de los términos definidos. Las definiciones deben ser objetivas e independientes de un contexto social o computacional. En lo posible, una definición debe ser expresada en axiomas lógicos y además, deben estar documentadas en lenguaje natural.
- **Coherencia:** Esto significa que las inferencias a partir de las definiciones deben ser consistentes.
- **Extensible:** La ontología debe estar diseñada de tal forma que permita definir nuevos términos basados en el vocabulario existente.

---

<sup>29</sup> Gruber, Thomas. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing", Stanford Knowledge Systems Laboratory, p. 12. Disponible en <http://tomgruber.org/writing/onto-design.pdf> Consultado: Agosto 2010.

- **Sesgo mínimo de codificación (Minimal encoding bias):** Las conceptualizaciones deben ser especificadas a nivel de conocimiento sin depender de la representación y la notación a nivel de símbolo.
- **Mínimo compromiso ontológico (Minimal ontological commitment):** Se deben hacer el mínimo de conceptualizaciones posibles sobre el mundo a ser modelado, permitiendo a los agentes extender las ontologías según propósitos particulares.

## 4.2 WEB SEMÁNTICA<sup>30</sup>

El objetivo de la Web Semántica es transformar la Web, de un repositorio de documentos enlazados a una base de conocimiento distribuido y una plataforma de aplicaciones, permitiendo que la amplia gama de información y servicios disponibles en la Web sean explotados eficientemente.

### 4.2.1 Definición de Web semántica<sup>31</sup>

Actualmente, el contenido Web consiste principalmente de hipertexto distribuido, el cual se accede a través de una combinación de palabras clave de búsqueda (por medio de un motor de búsqueda) y navegación por enlaces. Esta simplicidad ha sido una de las grandes fortalezas de la Web, y un factor importante en su popularidad y crecimiento.

Debido al crecimiento en la variedad y cantidad del contenido Web, han ido surgiendo algunas deficiencias graves en el paradigma hipertextual. En primer

---

<sup>30</sup> Horrocks, Ian. "Semantic Web: The Story So Far", *ACM W4A '07 Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A)*, Mayo 2007, pp. 120 - 124. Consultado: Marzo 2011

<sup>31</sup> *Ibíd.*

lugar, cada vez es más difícil de localizar el contenido requerido bajo el paradigma de búsqueda y navegación. Encontrar información sobre personas que tengan nombres en común, por ejemplo, puede ser una experiencia frustrante. Preguntas más complejas puede ser aún más problemáticas. Un ejemplo de ello puede ser la búsqueda de animales que vuelen, que no sean mamíferos ni ovíparos, debido a que podría retornar resultados irrelevantes relacionados con animales de ambas categorías (debido a que el motor de búsqueda no entiende la negación), ó podría fallar al retornar muchos resultados relevantes (debido a que la mayoría de páginas Web relevantes mencionan también animales ovíparos y mamíferos). Ahora, si usuarios humanos tienen dificultad accediendo a contenido Web, el problema es más aún delicado para procesos automatizados. Esto se debe a que el contenido Web esta principalmente destinado para presentación y consumo de usuarios humanos.

La Web Semántica tiene como objetivo superar el problema planteado, haciendo el contenido Web más accesible a procesos automatizados (programas, agentes informáticos). El objetivo final es transformar la Web actual en un conjunto de aplicaciones conectadas que formen una Web lógicamente consistente de datos. Esto ha de lograrse agregando anotaciones semánticas al contenido Web, es decir, anotaciones que describan el significado del contenido. De esta manera, la Web semántica va a permitir que procesos automatizados entiendan y respondan a solicitudes complejas, ya sean por humanos u otros programas informáticos, en función de su significado.

#### **4.2.2 Componentes de la Web semántica**

La Web semántica comprende estándares y herramientas de XML, XML Schema, RDF, RDF Schema y OWL, que según Tim Berners Lee se organizan en la Pila de la Web Semántica de la siguiente manera:

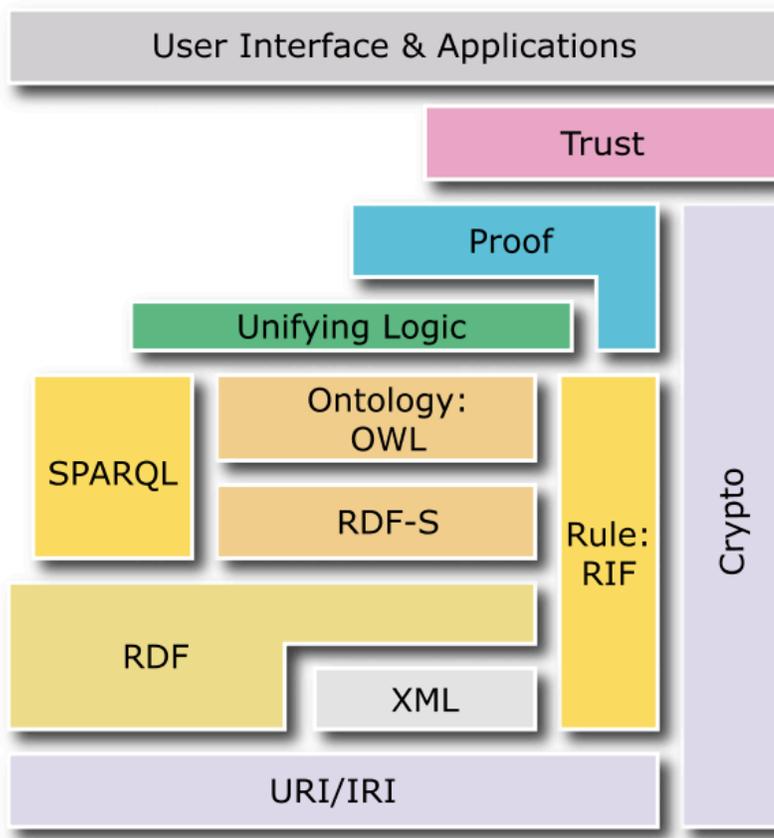


Figura 5. La Pila de la Web semántica.<sup>32</sup>

#### 4.2.2.1 XML<sup>33</sup>

Extensible Markup Language (XML) es un formato de texto simple y muy flexible, derivado de SGML (Standard Generalized Markup Language ISO 8879), y desarrollado, en 1996, por un Grupo de Trabajo sobre XML formado bajo el patrocinio del World Wide Web Consortium (W3C). Originalmente fue diseñado para afrontar los retos de publicación electrónica a gran escala, pero también está desempeñando un papel muy importante en el intercambio de una amplia variedad de datos en la Web y otros lugares.

<sup>32</sup> Bratt, Steve. "Semantic Web, and Other Technologies to Watch". Disponible en: [http://www.w3.org/2007/Talks/0130-sb-W3CTechSemWeb/#\(24\)](http://www.w3.org/2007/Talks/0130-sb-W3CTechSemWeb/#(24)) Consultado: Marzo 2011.

<sup>33</sup> "Extensible Markup Language (XML) 1.0 (Fifth Edition)", W3C Recommendation. Disponible en: <http://www.w3.org/TR/xml/> Consultado: Marzo 2011

Los objetivos de diseño para XML fueron:

- Debe ser directamente utilizable en Internet.
- Debe soportar una amplia variedad de aplicaciones.
- Debe ser compatible con SGML.
- Será fácil escribir programas que procesen documentos XML.
- El número de características opcionales en XML es que se le mantenga al mínimo absoluto, idealmente cero.
- Los documentos XML deben ser legibles por humanos y razonablemente claros.
- El diseño de XML debe ser preparado rápidamente.
- El diseño de XML debe ser formal y conciso.
- Los documentos XML deberán ser fáciles de crear.
- Mínima importancia en la concisión de las marcas XML

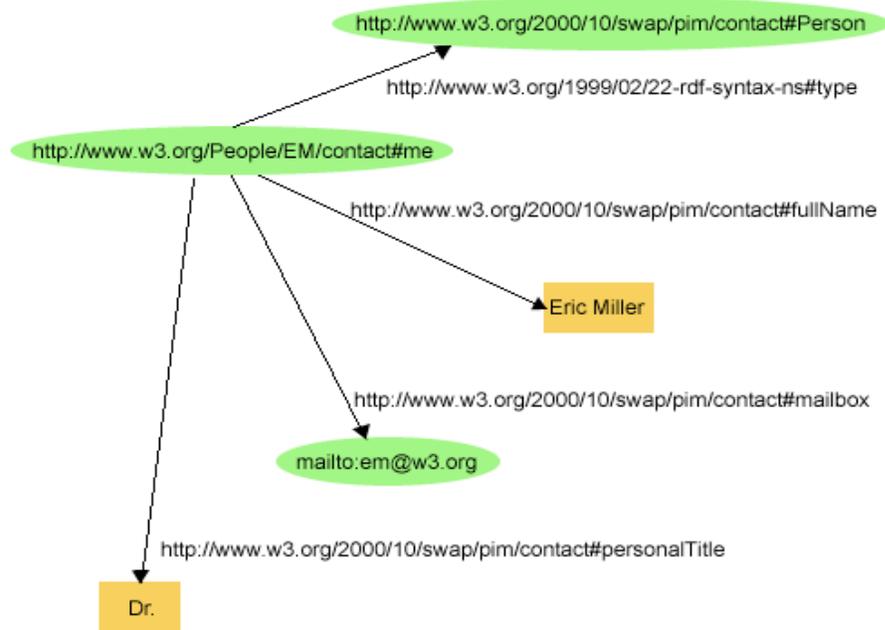
#### **4.2.2.2 RDF y RDFS**

Resource Description Framework (RDF) es un lenguaje para representar información sobre recursos en la World Wide Web. Busca representar metadatos sobre recursos Web, como el título, autor y fecha de modificación de una página Web, derechos de autor e información de la licencia de un documento Web, o la disponibilidad de horario de algún recurso compartido.

RDF está diseñado para situaciones en las que esta información debe ser procesada por aplicaciones, en lugar de sólo ser mostrada a la gente. RDF proporciona un marco común de expresión de esta información para que pueda ser intercambiada entre aplicaciones sin pérdida de significado.

RDF se basa en la idea de identificar recursos por medio de identificadores Web, llamados Uniform Resource Identifiers (URI), y la descripción de recursos en

términos de propiedades simples y valores. Esto le permite representar afirmaciones simples sobre recursos como un grafo de nodos y arcos, representando los recursos, propiedades y valores, como se ilustra a continuación:



**Figura 6. Un Grafo RDF Describiendo a Eric Miller** <sup>34</sup>

RDF, sin embargo, no proporciona mecanismos para la descripción de estas propiedades, ni proporciona ningún mecanismo para describir las relaciones entre estas propiedades y otros recursos. Ese es el papel del lenguaje de descripción de RDF Schema (vocabulario RDF), el cual define las clases y propiedades que pueden utilizarse para describir clases, propiedades y otros recursos<sup>35</sup>.

<sup>34</sup> "RDF Primer". Disponible en: <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/> Consultado: Marzo 2011.

<sup>35</sup> "RDF Vocabulary Description Language 1.0: RDF Schema". Disponible en: <http://www.w3.org/TR/rdf-schema/> Consultado: Marzo 2010

#### 4.2.2.3 OWL

Es un lenguaje de marcado semántico para definir, publicar y compartir ontologías en la Web, desarrollado por *The World Wide Web Consortium (W3C)*. OWL es desarrollado como una extensión del vocabulario de RDF (Resource Description Framework) y es derivado del lenguaje DAML + OIL. Está diseñado para ser utilizado por las aplicaciones que necesitan procesar el contenido de la información en lugar de sólo ilustrarlo a los humanos. Permite la formalización de un dominio mediante la definición de clases, propiedades y las relaciones entre las clases<sup>36</sup>.

OWL provee tres sub-lenguajes, cada uno con un nivel de poder expresivo diferente, diseñados para el uso de comunidades específicas de desarrolladores o usuarios, según sea el uso y el nivel de expresividad requerido. Estos tres sub-lenguajes son<sup>37</sup>:

- **OWL Lite:** Es para aquellos usuarios que necesitan una clasificación de jerarquía y restricciones con características simples. Por ejemplo, en el caso de restricciones de cardinalidad, éste solo permite valores cardinales de 0 ó 1. Es más fácil proveer herramientas de soporte para este lenguaje debido a su simpleza, además de ofrecer una ruta de migración rápida para tesauros y otras taxonomías.
- **OWL DL:** Es para aquellos usuarios que necesitan el máximo de expresividad sin perder integridad de computo y resolubilidad en los sistemas de razonamiento, es decir, para todos los cálculos se garantiza que sean computables y que terminen en un tiempo finito. Se llama OWL

---

<sup>36</sup> "OWL Web Ontology Language Reference". Disponible en: <http://www.w3.org/TR/2004/REC-owl-ref-20040210/> Consultado: Agosto 2010.

<sup>37</sup> "OWL Web Ontology Language Guide". Disponible en: <http://www.w3.org/TR/2004/REC-owl-guide-20040210/> Consultado: Febrero 2011.

DL en correspondencia a *Description Logics*, un campo de investigación que ha estudiado la lógica para representación formal del conocimiento en OWL.

OWL DL tiene las propiedades computacionales deseables para sistemas de razonamiento.

- **OWL Full:** Es para aquellos usuarios que necesitan el máximo de expresividad y libertad en la sintaxis, sin garantizar computacionalidad. Por ejemplo, una clase puede ser tratada como una colección de individuos y como un individuo simultáneamente, por derecho propio. Es poco probable que cualquier software de razonamiento sea capaz de soportar todas las características de OWL Full.

La mayor parte de los elementos en una ontología desarrollada en OWL, son las clases, propiedades, instancias de clases y relaciones entre estas instancias. A continuación se presenta una descripción de cada uno de estos elementos:

- **Clases:** Una clase define un grupo de individuos que comparten algunas propiedades. Por ejemplo, Deborah y Frank son miembros de la clase Persona. Las clases pueden ser organizadas por jerarquías según su especialización mediante el uso de subclases.
- **Propiedades:** Las propiedades pueden ser usadas para declarar relaciones entre individuos o de individuos con datos. Algunos ejemplos de propiedades son *tieneHijo*, *tieneFamiliar*, *tieneHermano* y *tieneEdad*. Las primeras tres pueden ser usadas para relacionar una instancia de la clase Persona con otra instancia de la clase *Persona*. A este tipo de propiedad se le llama *ObjectProperty*. La última propiedad (*tieneEdad*) puede ser usada

para relacionar una instancia de la clase *Persona* con una instancia del tipo de dato entero. A este tipo de propiedad se le llama *DatatypeProperty*.

- **Dominio:** Un dominio de una propiedad limita los individuos para los cuales la propiedad puede ser aplicada. Si una propiedad relaciona el individuo A con otro individuo B, y la propiedad tiene una clase como uno de sus dominios, entonces el individuo A debe pertenecer a esa clase. Por ejemplo, podría establecerse que la propiedad *tieneHijo* tiene como dominio la clase *Mamífero*.

De lo anterior, un razonador podría deducir que si Frank *tieneHijo* Anna, entonces Frank debe ser un *Mamífero*.

- **Rango:** El rango de una propiedad limita los individuos que la propiedad puede tener como valor. Si una propiedad relaciona un individuo A con otro individuo B, y la propiedad tiene una clase como su rango, entonces el individuo B debe pertenecer al rango de la clase. Por ejemplo, podría establecerse que la propiedad *tieneHijo* tiene como rango la clase *Mamífero*. De lo anterior, un razonador podría deducir que si Lousie está relacionada con Deborah por la propiedad *tieneHijo* (i.e., Deborah es hija de Louise), entonces Deborah es *Mamífero*.

Las propiedades a su vez tienen ciertas características, que denotan sus restricciones globales de cardinalidad: funcional, Inversamente funcional, transitiva y simétrica.

- Una propiedad es *funcional*, si es una propiedad que puede tener solamente un único valor y para cada instancia x. Esto quiere decir, que no pueden haber dos valores diferentes  $y_1$  y  $y_2$  de tal forma que

los pares  $(x, y_1)$  y  $(x, y_2)$  sean ambos instancias de esta propiedad.

- Una propiedad  $P$  es *inversamente funcional*, si un valor  $y$  puede ser solamente el valor de  $P$  para una única instancia de  $x$ . Esto quiere decir, que no pueden haber dos instancias diferentes  $x_1$  y  $x_2$  de tal forma que los pares  $(x_1, y)$  y  $(x_2, y)$  sean ambos instancias de  $P$ .
- Al afirmar que una propiedad  $P$  es *transitiva*, significa que si un par  $(x, y)$  es una instancia de  $P$ , y el par  $(y, z)$  es también una instancia de  $P$ , entonces se puede inferir que el par  $(x, z)$  es también una instancia de  $P$ .
- Una propiedad *simétrica* es una propiedad para la cual dice que si el par  $(x, y)$  es una instancia de  $P$ , entonces el par  $(y, x)$  es también una instancia de  $P$ .
- **Individuos:** Los individuos son instancias de clases, y las propiedades pueden ser usadas para relacionar diferentes individuos entre sí. Por ejemplo, un individuo llamado *Deborah* puede describirse como una instancia de la clase *Persona* y la propiedad *tieneEmpleo* puede ser usada para relacionar el individuo *Deborah* con el individuo *StandfordUniversity*.

## 4.3 LAS NOTAS

### 4.3.1 Definición de Sonido

Desde el punto de vista de la física, la altura de un sonido corresponde a un determinado número de vibraciones por segundo llamado frecuencia; a mayor

número de vibraciones por segundo más agudo es el sonido, y a menor número de vibraciones más grave es el sonido<sup>38</sup>.

Las diferentes alturas de los sonidos se representan en música mediante las notas musicales. En los sistemas español, francés e italiano, de grave a agudo, los nombres de las notas son: *Do, Re, Mi, Fa, Sol, La, Si*. Esta serie se repite de manera infinita y sucesiva: en ascenso, comenzando *Do, Re, Mi...* hasta llegar a *Si*.

#### 4.3.2 Distancia entre dos notas

La distancia entre dos sonidos es lo que se conoce como *intervalo musical*.

En la música occidental, la distancia más corta entre dos notas contiguas es el semitono y la máxima, el tono<sup>39</sup>. En relación con las notas musicales, el semitono se encuentra entre las notas *Mi* y *Fa*, y entre *Si* y *Do*; y el tono se encuentra entre *Do* y *Re*, por ejemplo.

La mejor representación de estas dos distancias en el teclado del piano es la distancia entre las teclas blancas y negras: entre una tecla blanca y una negra, la distancia es un semitono; y entre dos teclas blancas, la distancia es un tono. Las únicas excepciones a esta relación, son las teclas de *Mi - Fa* y *Si - Do*, entre las cuales si bien existen dos teclas blancas contiguas, la distancia sigue siendo un semitono.

---

<sup>38</sup> "Altura (música)". Disponible en: [http://es.wikipedia.org/wiki/Altura\\_\(música\)](http://es.wikipedia.org/wiki/Altura_(música)) Consultado: Enero 2011.

<sup>39</sup> A partir de este momento, se representarán mediante las siglas 1/2T y T, respectivamente.

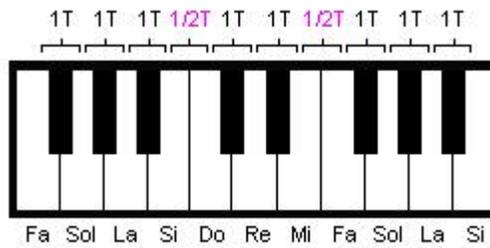


Figura 7. Distancia entre dos notas.<sup>40</sup>

### 4.3.3 Tipos de Intervalos

Los intervalos son importantes porque son la base de la armonía musical, siendo las distancias entre las notas las que sirven para la construcción de los acordes, estructuras complejas que dan mayor riqueza a la música.

Existen distintos tipos de intervalos de acuerdo a su cualidad: **justa, mayor, menor, disminuido o aumentado**. La tabla a continuación muestra la clasificación de los intervalos de acuerdo a la distancia de tono-semitono de sus notas.

2 <sup>a</sup> menor	½ tono
2 <sup>a</sup> mayor	1 tono
3 <sup>a</sup> menor	1 tono y ½
3 <sup>a</sup> mayor	2 tonos
4 <sup>a</sup> justa	2 tonos y ½
4 <sup>a</sup> aumentada	3 tonos
5 <sup>a</sup> disminuida	3 tonos
5 <sup>a</sup> justa	3 tonos y ½

<sup>40</sup> “Blog del Curso de Solfeo para Adultos de L’Elia”. Disponible en: <http://iniciamusica.blogspot.com/> Consultado: Enero 2011.

5ª aumentada	4 tonos
6ª menor	4 tonos
6ª mayor	4 tonos y ½
7ª menor	5 tonos
7ª mayor	5 tonos y ½
8ª justa	6 tonos

**Figura 8. Intervalos (música).**

#### 4.3.4 Alteraciones y semitonos

Las siete notas musicales pueden modificarse por medio de las alteraciones. Una alteración actúa sobre una nota agregándole o quitándole un semitono.

El sostenido y el bemol alteran las notas un semitono, el primero de manera ascendente y el segundo, de manera descendente. El doble sostenido y el doble bemol alteran las notas de la misma manera, pero a un tono de la nota. El becuadro cancela una alteración realizada previamente. Los símbolos y la distancia de cada alteración se representan en el siguiente cuadro.

Alteración	Símbolo	Cambio de altura
Sostenido	#	Medio tono más agudo
Bemol	b	Medio tono más grave
Doble sostenido	##	Un tono más agudo
Doble bemol	bb	Un tono más grave
Becuadro	□	Anula la alteración previa

**Figura 9. Alteraciones y semitonos.**

Por regla general, estas alteraciones sólo tienen validez por un compás.

## 4.4 NOTACIÓN DE LAS ALTURAS

### 4.4.1 El pentagrama

Es el lugar donde se escriben las notas musicales y tiene las siguientes características:

- Está formado por cinco líneas horizontales y cuatro espacios.
- Las cinco líneas y los cuatro espacios se cuentan de abajo hacia arriba.
- La parte baja del pentagrama corresponde a los sonidos graves y la parte alta a los agudos.
- La lectura o desplazamiento de izquierda a derecha representa el avance en el tiempo.

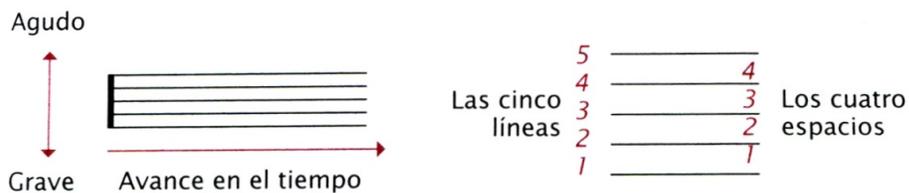


Figura 10. El pentagrama.<sup>41</sup>

### 4.4.2 La clave Sol

Indica que la segunda línea del pentagrama corresponde al *sol*/3. Las demás notas se leen tomando este *sol* como referencia. Se utiliza para los registros medio y agudo.

<sup>41</sup> Abromont, Claude; De Montalembert, Eugene. TEORÍA DE LA MÚSICA; UNA GUÍA, Fondo de Cultura Económica USA, 2005, p. 35.



Figura 11. Clave de *sol*.<sup>42</sup>

#### 4.4.3 La clave Fa

Indica que la cuarta línea del pentagrama corresponde a un *fa*2. Las demás notas se leen o acomodan tomando este *fa* como referencia. Se utiliza para los registros medio y grave.

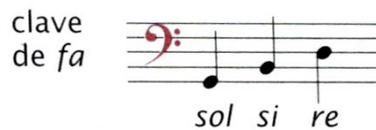


Figura 12. Clave de *fa*.<sup>43</sup>

### 4.5 LAS TONALIDADES

La tonalidad es un grupo de sonidos que forman un sistema y están regidos por una nota central o fundamental llamada tónica. La tonalidad se define en una pieza musical, a partir de la escala y los acordes que se utilicen y se identifica mediante las notas que se encuentren alteradas a lo largo de la pieza o por la armadura al comienzo, que es el grupo de alteraciones que acompañan a la clave al principio de la pieza musical.

La figura a continuación muestra todas las armaduras y la tonalidad a la que corresponde cada una.

---

<sup>42</sup> *Ibíd.*, p. 39.

<sup>43</sup> *Ibíd.*, p. 39.

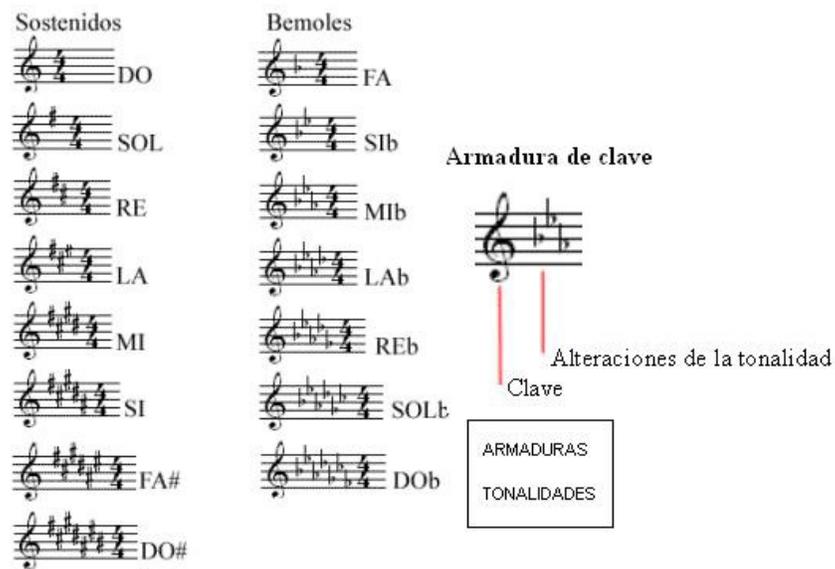


Figura 13. Las Tonalidades.<sup>44</sup>

#### 4.5.1 Modulaci3n

Modular es cambiar la tonalidad de una obra en el transcurso de 3sta, sin necesidad de tener que indicarlo mediante una doble barra, siempre y cuando, se sigan determinadas reglas de armon3a.

Las modulaciones cuentan con una cadencia propia que reafirma la nueva tonalidad y por lo general se realizan a tonalidades vecinas, es decir, aquellas que son cercanas a la tonalidad principal y para las que son necesarias menor n3mero de alteraciones: tonalidad del relativo, tonalidad de la dominante y su relativo, tonalidad de la subdominante y su relativo.

Por ejemplo, si la tonalidad es Do mayor, las tonalidades vecinas a las que generalmente se puede modular son: La menor (relativo menor), Sol mayor

<sup>44</sup> Las Tonalidades. Disponible en: <http://bach2411111.blogcindario.com/2006/04/00130-teoria-musical-armaduras.html> Consultado: Enero 2011.

(dominante), Mi menor (relativo menor del dominante), Fa mayor (subdominante) y Re menor (relativo menor de la subdominante).

#### **4.5.2 Funciones Tonales**

La música del periodo entre los S. XVII y XIX, o música de la práctica común, se fundamenta en ciertas relaciones especiales entre los grados de las escalas. Dichas relaciones, llamadas funciones tonales, tienen que ver con el nivel de reposo o inestabilidad de cada una de las notas y su aporte al discurso musical.

Existen tres funciones que rigen no sólo los acordes sino también las regiones tonales de una obra, que son:

La función de tónica, caracterizada por su estabilidad, se representa con el I grado o tónica; la función de dominante, caracterizada por su inestabilidad, se representa con el V grado, aunque el VII también se emplea para esta función; y la función de subdominante, que posee las cualidades de las dos anteriores, por lo que es utilizada para desplazarse de una función a otra. El grado representativo de esta región es el IV.

En el conjunto de piezas que se tuvieron en cuenta para la ontología, la última nota corresponde al I grado, es decir, que termina en una función de tónica.

#### **4.6 LAS ESCALAS**

Una escala es la sucesión ascendente o descendente, ordenada consecutivamente, de todas las notas de un sistema sonoro particular.

Existen diferentes tipos de escalas, como la escala pentatónica o pentáfona (de cinco sonidos), la escala de tonos enteros (de seis sonidos separados entre sí por un tono), la escala cromática (de doce sonidos separados entre sí por medio tono) y la escala diatónica (de siete sonidos, conformada por intervalos de segunda consecutivos, que se divide en dos variantes o modos: mayor y menor), por sólo mencionar algunas.

Para este proyecto se hace necesario profundizar en esta última escala, por ser ésta la de mayor predominancia en las canciones infantiles.

#### 4.6.1 La escala diatónica mayor

En la teoría musical, una escala mayor es aquella cuyo tercer grado se encuentra a un intervalo de tercera mayor sobre la tónica. Las distancias entre los tonos de una escala mayor es T-T-1/2T-T-T-T-1/2T.



Figura 14. Escala diatónica mayor.<sup>45</sup>

Las escalas mayores pueden comenzarse con cualquier nota, pero siempre deben acomodarse al modelo de tonos y medios tonos de la Escala Mayor<sup>46</sup>.

<sup>45</sup> Escala diatónica mayor. Disponible en: <http://musica-bella.blogspot.com/2009/07/las-escalas-diat-mayor-y-menor.html> Consultado: Enero 2011.

<sup>46</sup> Brimhall, John: Cuaderno de Teoría. 3 Libros en 1, p. 25.



Figura 15. Escala de Sol mayor.<sup>47</sup>

## 4.8 MUSICXML<sup>48</sup>

Es una tecnología desarrollada por Recordare LLC, para representar en formato XML la notación musical occidental a partir del siglo XVII. Está diseñada como formato de intercambio para la notación, análisis, recuperación y rendimiento de las aplicaciones y se basa principalmente en dos formatos musicales académicos:

- El formato **MuseData**, desarrollado por Walter Hewlett del Center for Computer Assisted Research in the Humanities (CCARH), localizado en la Universidad de Standford.
- El formato **Humdrum**, desarrollado por David Huronm del Ohio State University.

### 4.8.1 Formas de estructurar archivos MusicXML<sup>49</sup>

- **Partwise:** Contiene una o más partes, cada una de las cuales contiene una o más medidas y cada medida tiene datos musicales.

Se define de la siguiente manera:

<sup>47</sup> Escala sol mayor. Disponible en: <http://musica-bella.blogspot.com/2009/07/las-escalas-diat-mayor-y-menor.html> Consultado: Enero 2011.

<sup>48</sup> "Recordare. Partwise DTD". Disponible en: <http://www.recordare.com/musicxml/dtd/partwise-dtd> Consultado: Enero 2011

<sup>49</sup> "Recordare, MusicXML 2.0 Tutorial". Disponible en: <http://www.recordare.com/sites/default/files/musicxml-tutorial.pdf> Consultado: Enero 2011

```

<![ %partwise; [
<!ELEMENT score-partwise (%score-header;, part+)>
<!ELEMENT part (measure+)>
<!ELEMENT measure (%music-data;)>
]]>

```

- **Timewise:** Contiene una o más medidas, cada una de las cuales contiene una o más partes y cada parte tiene datos musicales.

Se define de la siguiente manera:

```

<![ %timewise; [
<!ELEMENT score-timewise (%score-header;, measure+)>
<!ELEMENT measure (part+)>
<!ELEMENT part (%music-data;)>
]]>

```

En ambos casos, los elementos de nivel inferior tiene las entidades de tipo *music-data*, la cual está definida de la siguiente manera:

```

<!ENTITY % music-data
"(note | backup | forward | direction | attributes |
harmony | figured-bass | print | s grouping | link |
bookmark) *">

```

#### 4.8.2 “Hola Mundo” en MusicXML<sup>50</sup>

A continuación se ilustra la representación de la nota musical Do en formato MusicXML y se explica en detalle cada una de sus partes.



Figura 16. Do en la escala de Do mayor.

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<!DOCTYPE score-partwise PUBLIC
    "-//Recordare//DTD MusicXML 2.0 Partwise//EN"
    "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="2.0">
  <part-list>
    <score-part id="P1">
      <part-name>Music</part-name>
    </score-part>
  </part-list>
  <part id="P1">
    <measure number="1">
      <attributes>
        <divisions>1</divisions>
        <key>
          <fifths>0</fifths>
```

<sup>50</sup> “Hello World in MusicXML” Disponible en: <http://www.recordare.com/musicxml/tutorial/hello-world>  
Consultado: Enero 2011

```

    </key>
    <time>
      <beats>4</beats>
      <beat-type>4</beat-type>
    </time>
    <clef>
      <sign>G</sign>
      <line>2</line>
    </clef>
  </attributes>
  <note>
    <pitch>
      <step>C</step>
      <octave>4</octave>
    </pitch>
    <duration>4</duration>
    <type>whole</type>
  </note>
</measure>
</part>
</score-partwise>

```

**Figura 17. Representación en MusicXML de la nota Do en la escala de Do mayor.**

```
<score-partwise version="2.0">
```

Aquí se define el tipo de archivo MusicXML; *partwise* o *timewise*. El elemento `<score-partwise>` está compuesto por partes, cada una de las cuales está compuesta por medidas.

```
<part-list>
  <score-part id="P1">
    <part-name>Part 1</part-name>
  </score-part>
</part-list>
```

Un archivo MusicXML comienza con un encabezado que lista las diferentes partes musicales de la partitura, en este caso solo una.

```
<part id="P1">
```

Aquí comienza la descripción de la primera y única parte en el documento. El atributo *id* debe hacer referencia a un atributo *id* del *score-part* en la cabecera.

```
<measure number="1">
```

Aquí se comienza con la primera medida en la primera parte.

```
<attributes>
```

Los atributos contienen información clave necesaria para interpretar las notas y datos musicales que siguen en esta parte.

```
<divisions>1</divisions>
```

Cada nota en MusicXML tiene una duración. Los elementos *divisions* provee la unidad de medida para la duración en términos de divisiones por cuarto de nota.

Las duraciones musicales son típicamente expresadas en fracciones, como cuartos y octavos de notas. Las duraciones en MusicXML son fracciones también.

Dado que el denominador rara vez tiene que cambiar, es representado por separado en el elemento de *divisions*, de modo que sólo el numerador debe ser asociado con cada nota individualmente.

```
<key>
  <fifths>0</fifths>
</key>
```

El elemento *key* es usado para representar la clave. En este caso la clave es *Do* mayor, sin bemoles ni sostenidos, por tanto el elemento *fifths* es 0. Si estuviera en la clave de *Re* mayor con dos sostenidos, el elemento *fifths* sería 2. Y si estuviera en la clave de *Fa* mayor con un bemol, el elemento *fifths* sería -1.

```
<time>
  <beats>4</beats>
  <beat-type>4</beat-type>
</time>
```

El elemento *time* representa un compás. Y son dos los elementos que lo componen, *beat* y *beat-type*: el numerador y denominador del compás respectivamente.

```
<clef>
  <sign>G</sign>
  <line>2</line>
</clef>
```

Aquí, es representada la clave de Sol situada en la segunda línea del pentagrama.

```
</attributes>
```

```
<note>
```

Se ha finalizado con los atributos, y se comienza con la primera nota.

```
<pitch>  
  <step>C</step>  
  <octave>4</octave>  
</pitch>
```

El *pitch* representa el tono y debe tener los elementos *step* y *octave*. Opcionalmente puede haber un elemento *alter*, si hay un bemol o un sostenido involucrado. En este caso no se tiene alteración. El tono es C y el elemento *octave* de valor 4 indica que la octava comienza en C central. Por tanto la nota es un *do* central.

```
<duration>4</duration>
```

Es la duración de la nota musical.

```
<type>whole</type>
```

El elemento *type* dice que la nota se simboliza con una redonda.

```
</note>
```

Se ha finalizado con la nota.

```
</measure>
```

Se ha finalizado con la medida.

</part>

Se ha finalizado con la parte.

</score-partwise>

Y por último se ha finalizado con la partitura.

## 5. TECNOLOGÍAS

### 5.1 PROTÉGÉ

Protégé es un editor de ontologías que está basado en Java, de código abierto<sup>51</sup> y que es extensible en la medida que provee un entorno plug & play, lo que permite una base flexible para desarrollo de aplicaciones. Fue desarrollado por la Universidad de Standford en colaboración con la Universidad de Manchester y es soportado por una comunidad de desarrolladores, académicos, usuarios gubernamentales y corporativos.

### 5.2 ECLIPSE

Eclipse es un ambiente integrado de desarrollo de software. Tiene un sistema extensible de plugins que permite desarrollar en diferentes lenguajes de programación, tales como: Java, Ada, C, C++, COBOL, Perl, PHP, Python, Ruby, Scala, entre otros.

### 5.3 OWL API

OWL API es una API de Java e implementación de referencia para crear, manipular y serializar Ontologías en OWL.

OWL API es de código abierto y está disponible bajo la licencia LGPL. OWL API se integra fácilmente los siguientes motores de inferencia: como FaCT++, Hermit, Pellet and Racer. En el proyecto se utiliza Hermit.

---

<sup>51</sup> Código abierto o en Ingles Open source, es el término con el que se conoce al software distribuido y desarrollado libremente.

## **5.4 HERMIT**

Hermit es un razonador para ontologías en OWL. Dado una ontología, Hermit puede determinar si es consistente, inferir relaciones entre clases y mucho más.

## **5.5 PROXYMUSIC**

Es una librería para Java que permite leer y escribir composiciones musicales en formato MusicXML.

## 6. ONTOLOGÍA MUSICAL

La ontología desarrollada para este proyecto ha sido implementada en OWL-DL, un sub-lenguaje de OWL, que permite la representación de conocimiento del dominio de interés, de una forma estructurada y formalmente bien comprendida, mediante la definición de clases, propiedades, instancias de clases, relaciones entre estas instancias y el uso de descriptores lógicos (description logics – DL).

### 6.1 DEFINICIÓN DE CLASES

#### 6.1.1 Distance

**Descripción:** Esta clase representa las distancias entre las notas musicales, para las escalas: CMajor, DMajor, EMajor, FMajor, GMajor, AMajor, BMajor.

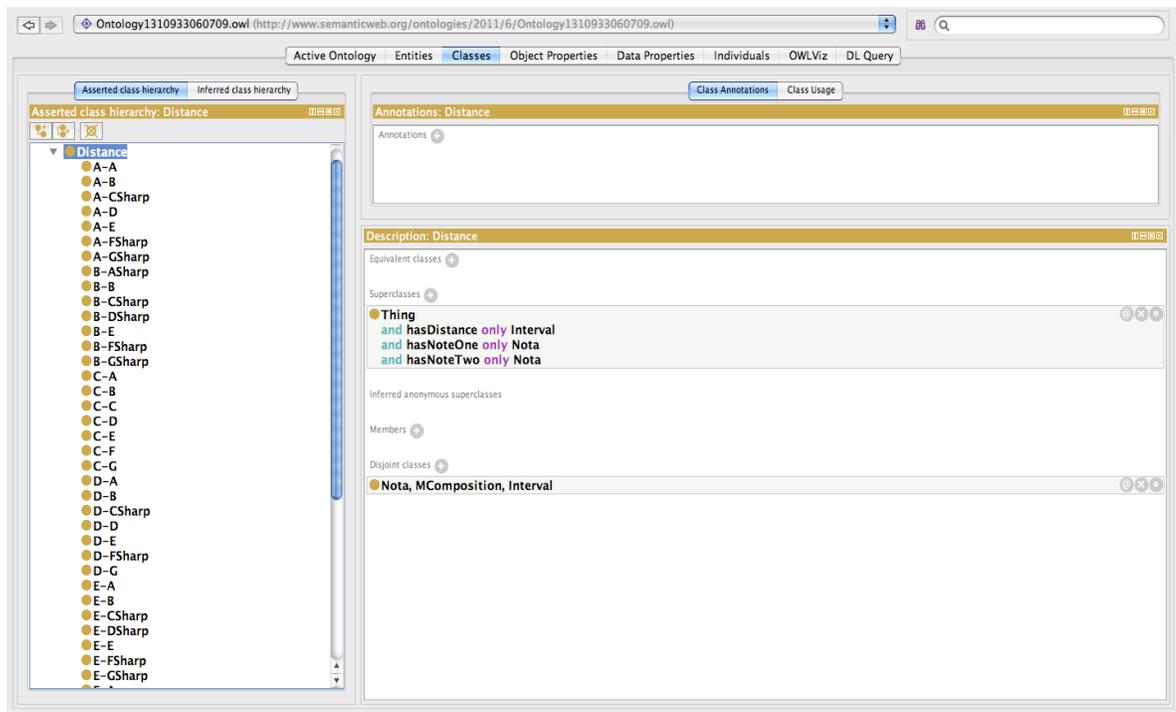


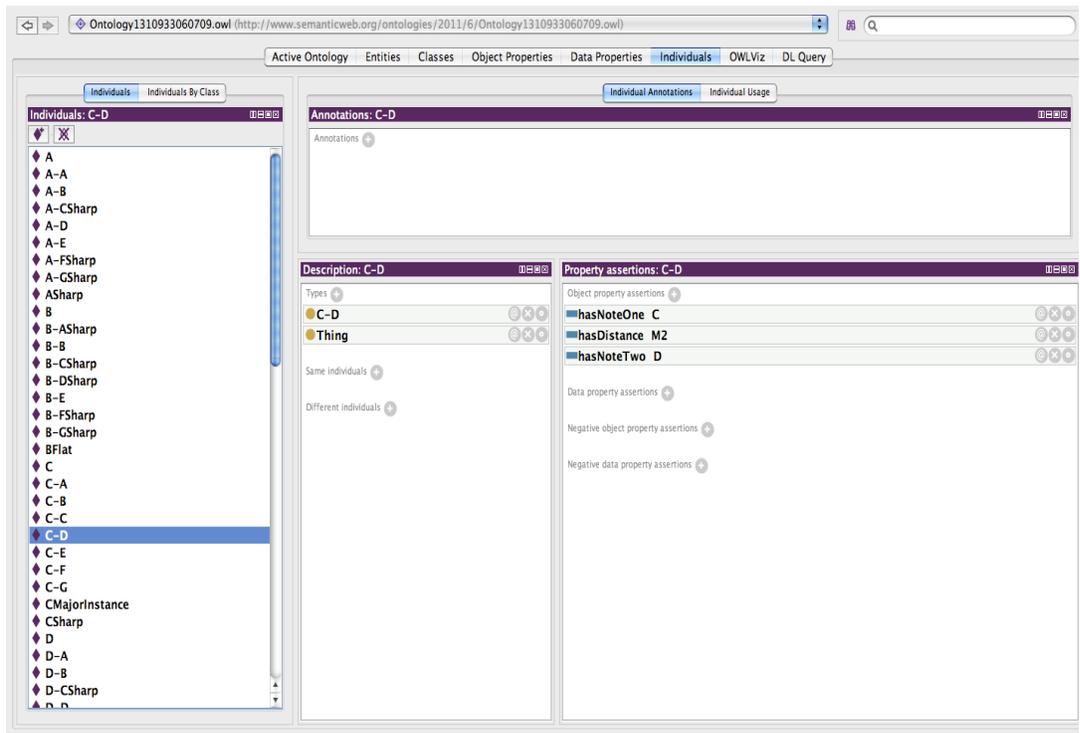
Figura 18. Definición de la clase Distance.

Cada instancia de la clase tiene las siguientes propiedades:

- *hasNoteOne*:  
Dominio: Distance.  
Rango: Nota.
  
- *hasNoteTwo*:  
Dominio: Distance.  
Rango: Nota.
  
- *hasDistance*:  
Dominio: Distance.  
Rango: Interval.

**Ej.** El individuo **C-D** representa la distancia entre las notas *Do* y *Re* en la escala *Do mayor*, mediante las siguientes afirmaciones:

- **C-D** *hasNoteOne* **C**
- **C-D** *hasNoteTwo* **D**
- **C-D** *hasDistance* **M2**



**Figura 19. Individuo C-D. Representa la distancia entre las notas *Do* y *Re* en la escala *Do mayor*.**

### 6.1.2 Interval

**Descripción:** Esta clase agrupa los distintos tipos de intervalos que se encuentran en una escala de una octava, de acuerdo a su cualidad: Mayor, Menor, Justa, Disminuida o Aumentada.

**Miembros de la clase:** P1, m1, m2, m3, P4, A4, d5, P5, m6, M6, m7, M7.

### 6.1.3 MComposition

**Descripción:** Representa el conjunto de composiciones musicales en tonalidad mayor. Tiene unas subclases que especifican cual es la tonalidad: CMajor, DMajor, EMajor, FMajor, GMajor, AMajor, BMajor.

Para designar las tonalidades se establecieron las siguientes condiciones:

1. Debe tener sólo las notas pertenecientes a la respectiva escala mayor. Ej. Para Do Mayor debe tener solamente las notas: *Do, Re, Mi, Fa, Sol, La, Si*.
2. Debe tener al menos las notas correspondientes a los grados I, III, VII y además, la última nota debe ser la correspondiente al grado I, es decir, la tónica. Ej. Para Do Mayor debe tener al menos las notas: *Do, Mi y Si*. Y terminar en *Do*.

La siguiente imagen muestra la representación de una composición musical en Do Mayor mediante la ontología:

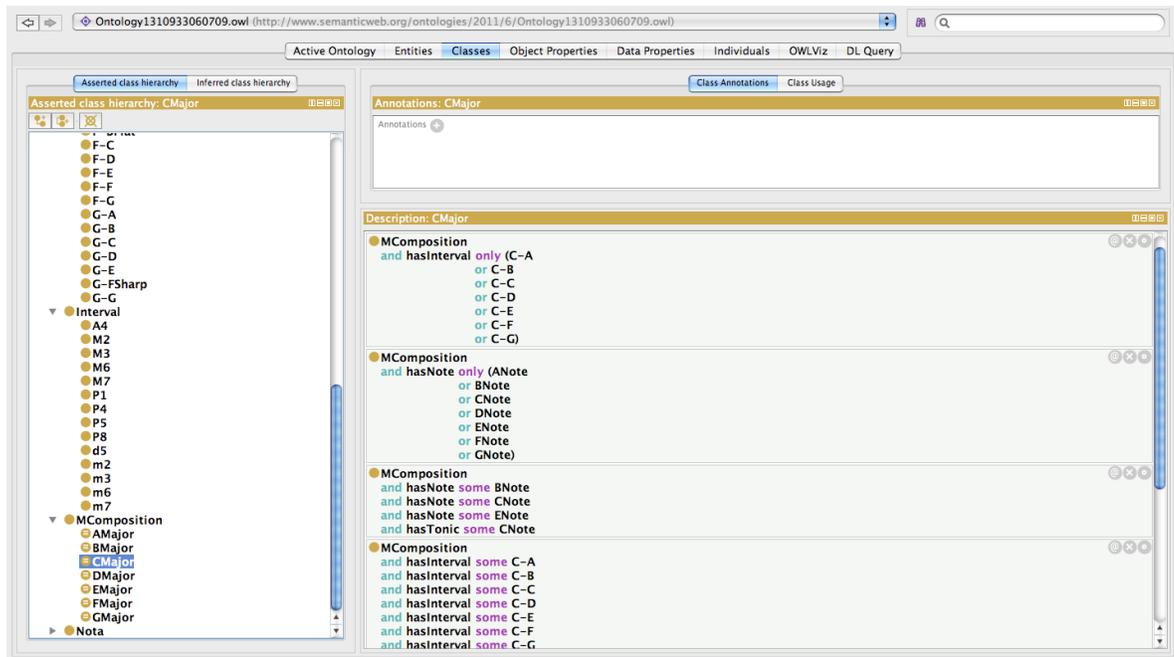


Figura 20. Representación de una composición en Do Mayor mediante la ontología.

#### 6.1.4 Nota

**Descripción:** Representa el conjunto de notas musicales, presentes en las escalas de Do Mayor, Re Mayor, Mi Mayor, Fa Mayor, Sol Mayor, La Mayor y Si Mayor.

**Miembros de la clase:** C, C#, D, DSharp, E, F, FSharp, G, GSharp, A, ASharp, B, BFlat.

## 6.2 PROPIEDADES

### 6.2.1 hasDistance

Dominio: Distance.

Rango: Interval.

Característica: Funcional.

### **6.2.2 hasInterval**

Dominio: MComposition.

Rango: Distance.

Característica: Inversamente funcional.

### **6.2.3 hasNote**

Dominio: MComposition.

Rango: Nota.

Propiedades inversas: isNoteOf.

### **6.2.4 hasNoteOne**

Dominio: Distance.

Rango: Nota.

Característica: Funcional.

Propiedades inversas: isNoteOneOf.

### **6.2.5 hasNoteTwo**

Dominio: Distance.

Rango: Nota.

Característica: Funcional.

Propiedades inversas: isNoteTwoOf.

### **6.2.6 hasTonic**

Dominio: MComposition.

Rango: Nota.

Característica: Funcional, Inversamente funcional.

### **6.2.7 isIntervalOf**

Dominio: Distance.

Rango: MComposition.

Característica: Funcional.

Propiedades inversas: hasInterval

### **6.2.8 isNoteOf**

Dominio: Nota.

Rango: MComposition.

Propiedades inversas: hasNote.

### **6.2.9 isNoteOneOf**

Dominio: Nota.

Rango: Distancia.

Propiedades inversas: hasNoteOne.

### **6.2.10 isNoteTwoOf**

Dominio: Nota.

Rango: Distance.

Propiedades inversas: hasNoteTwo.

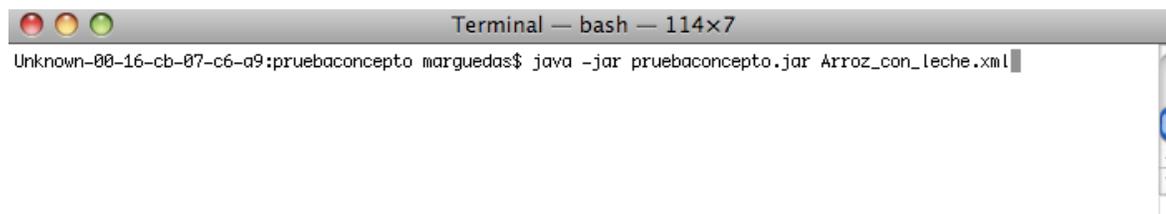
## 7. PRUEBA DE CONCEPTO

La prueba de concepto es el programa que utiliza la ontología para inferir la tonalidad de una composición musical en formato MusicXML.

Consta de un archivo llamado `pruebaconcepto.jar`, el cual contiene los ejecutables del programa, y dos directorios: `composicionesMusicales` y `baseDeConocimiento`. En el primer directorio se encuentran las composiciones musicales que el programa va a analizar: Arroz con leche, Fray Santiago, Mambrú se fue a la Guerra, Pinpón. Y en el segundo directorio, la ontología.

El programa es ejecutado desde cualquier interfaz de línea de comandos, ya sea en Windows, Linux o Mac OS, de la siguiente manera:

```
java -jar pruebaconcepto [ArchivoComposicionMusical]
```



**Figura 21. Ejecutando la prueba de concepto desde consola.**

Al ejecutar el programa se obtienen las notas de la composición musical en XML, y son cargadas en la ontología. Luego es creado el individuo `Song`, que sería la representación de la composición musical en la ontología, y las relaciones de cada nota con este individuo, mediante la propiedad `hasNote`. A su vez se obtiene la última nota de la composición musical y se relaciona con el individuo `Song` mediante la propiedad `hasTonic`.

Finalmente, se utiliza el razonador para inferir de qué tipo (tonalidad) es el individuo `Song`, es decir, a cuál clase pertenece.

Si el programa se ejecutó de manera exitosa se imprime la siguiente información:

- La última nota de la composición musical.
- Las notas de la composición musical.
- La consistencia de la ontología luego de haber realizado operaciones con el razonador.
- Los valores de la propiedad `hasNote` para el individuo `Song`, que vienen siendo las notas musicales.
- El valor de la propiedad `hasTonic` para el individuo `Song`, que viene siendo la última nota de la composición musical.
- El tipo de individuo `Song`.

```
Terminal — bash — 108x43
marco-arguedass-macbook-pro-225:pruebaconcepto marguedas$ java -jar pruebaconcepto.jar Arroz_con_leche.xml
The last note (tonic) is: F
The notes in the composition are:
C
F
A
F
F
C
F
F
A
G
A
BFlat
A
G
F
E
E
C
D
E
E
F
Consistent: true

The hasNote property values for Song are:
<http://www.semanticweb.org/ontologies/2011/6/Ontology1310933060709.owl#E>
<http://www.semanticweb.org/ontologies/2011/6/Ontology1310933060709.owl#G>
<http://www.semanticweb.org/ontologies/2011/6/Ontology1310933060709.owl#D>
<http://www.semanticweb.org/ontologies/2011/6/Ontology1310933060709.owl#F>
<http://www.semanticweb.org/ontologies/2011/6/Ontology1310933060709.owl#A>
<http://www.semanticweb.org/ontologies/2011/6/Ontology1310933060709.owl#C>
<http://www.semanticweb.org/ontologies/2011/6/Ontology1310933060709.owl#BFlat>
The hasTonic property values for Song are:
<http://www.semanticweb.org/ontologies/2011/6/Ontology1310933060709.owl#F>
Song Type:
<http://www.semanticweb.org/ontologies/2011/6/Ontology1310933060709.owl#FMajor>

marco-arguedass-macbook-pro-225:pruebaconcepto marguedas$
```

**Figura 22. Resultado de la prueba de concepto desde consola.**

La imagen anterior muestra que la tonalidad de la composición musical `Arroz_con_leche.xml` es Fa Mayor, ya que el individuo `Song`, que representa dicha composición en la ontología, pertenece a la clase `FMajor`.



## 8. CONCLUSIÓN

*“Una ontología se utiliza típicamente para permitir a los computadores dar sentido a los datos que están manipulando, como, por ejemplo, deducir nuevo conocimiento a partir de los datos dispuestos gracias a un motor de inferencia simple”<sup>52</sup>.*

Si bien la teoría musical plantea reglas teóricas que pueden ser trasladadas a una ontología, existe un amplio rango de composiciones musicales en las que se requiere un análisis diferente al que provee el formalismo musical. Es por esto que las ontologías siguen siendo una herramienta limitada a la hora de analizar ciertas obras y sigue siendo indispensable el criterio de un músico; es por esto que pensar en el análisis musical solamente realizado por un computador sigue siendo una utopía.

---

<sup>52</sup> Sebastien, Véronique; Sebastien, Didier; Conruyt Noël, "An Ontology for Musical Performances Analysis: Application to a collaborative platform dedicated to instrumental practice", 2010 Fifth International Conference on Internet and Web Applications and Services, mayo de 2011, p. 538. Consultado: Octubre 2011

## 9. TRABAJO FUTURO

- Es importante que se piense en enriquecer la ontología con nuevas clases y propiedades que capturen otros elementos de la teoría musical, para que se puedan realizar a futuro análisis melódicos, rítmicos y armónicos más complejos de piezas tonales. Así como pensar, que tal vez exista la manera de posibilitar los análisis de piezas atonales.
- También, se podría pensar en utilizar la ontología en programas que sirvan como asistentes de enseñanza de la teoría musical, en la que los estudiantes puedan resolver ejercicios y a partir de la ontología decir cuáles son los errores.

## BIBLIOGRAFÍA

Abromont, Claude; De Montalembert, Eugene. TEORÍA DE LA MÚSICA; UNA GUÍA, Fondo de Cultura Económica USA, 2005, p. 35.

Brimhall, John: Cuaderno de Teoría. 3 Libros en 1, p. 25.

Ferrara, Alfio; Ludovico, Luca A.; Montanelli, Stefano; Castano Silvana y Haus, Goffredo, "A Semantic Web Ontology for Context-based Classification and Retrieval of Music Resources", *ACM Transactions on Multimedia Computing, Communications and Applications*, Vol. V, abril de 2006, pp. 1–25. Consultado: Octubre 2011

Yúdice, George. Nuevas tecnologías, música y experiencia. Gedisa Editorial, Barcelona, 2007.

## CIBERGRAFÍA

“Altura (música)”. Disponible en: [http://es.wikipedia.org/wiki/Altura\\_\(música\)](http://es.wikipedia.org/wiki/Altura_(música))  
Consultado: Enero 2011.

Ames, Charles, “Automated composition in retrospect: 1956-1986, *Leonard. Edición especial: Artes visuales, sonido, música y tecnología*, Vol. 20, No. 2, Massachusetts Institute of Technology (MIT), Massachusetts, 1987, p. 170.  
Disponible en: <http://www.jstor.org/stable/3679501> Consultado: Octubre de 2011

“Blog del Curso de Solfeo para Adultos de L’Eliana”. Disponible en: <http://iniciamusica.blogspot.com/> Consultado: Enero 2011.

Bratt, Steve. “Semantic Web, and Other Technologies to Watch”. Disponible en: [http://www.w3.org/2007/Talks/0130-sb-W3CTechSemWeb/#\(24\)](http://www.w3.org/2007/Talks/0130-sb-W3CTechSemWeb/#(24)) Consultado: Marzo 2011.

Curtis, Roads. Research in Music and Artificial Intelligence. En: Computing Surveys, Junio de 1985, Vol. 17, No. 2, p. 163. Disponible en: <http://students.cs.byu.edu/~cs470ta/articles/p163-roads.pdf> Consultado: Marzo 2011

Escala sol mayor. Disponible en: <http://musica-bella.blogspot.com/2009/07/las-escalas-diat-mayor-y-menor.html> Consultado: Enero 2011.

“Extensible Markup Language (XML) 1.0 (Fifth Edition)”, W3C Recommendation. Disponible en: <http://www.w3.org/TR/xml/>

Gruber, Thomas. "Ontology", En: Ling Liu and M. Tamer Özsu (Eds.), *Encyclopedia of Database Systems*, Springer-Verlag, 2009. Disponible en: <http://tomgruber.org/writing/ontology-definition-2007.htm> Consultado: Enero 2011.

\_\_\_\_\_. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing", Stanford Knowledge Systems Laboratory, p. 12. Disponible en <http://tomgruber.org/writing/onto-design.pdf> Consultado: Agosto 2010.

Guariola, Nicola. "Formal Ontology and Information Systems", National Research Council, LADSEB-CNR. Disponible en: <http://www.loa-cnr.it/Papers/FOIS98.pdf> Consultado: Noviembre 2010.

Haus, Goffredo y Longari, Maurizio, "A Multi-Layered, Time-Based Music Description Approach Based on XML", *Computer Music Journal*, Vol, 29, No. 1, Massachusetts Institute of Technology (MIT), Massachusetts, 2005, pp. 70-85, Disponible en: <http://www.jstor.org/stable/3681878> Consultado: Octubre 2011.

"*Hello World* in MusicXML". Disponible en: <http://www.recordare.com/musicxml/tutorial/hello-world> Consultado: Enero 2011

Horrocks, Ian. "Semantic Web: The Story So Far", *ACM W4A '07 Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A)*, Mayo 2007, pp. 120 - 124. Consultado: Marzo 2011

Lamb, Martin Robert, *The computer as musicianship teaching aid*, Tesis doctoral en Ingeniería Eléctrica de la Universidad de Canterbury, Nueva Zelanda, 1979, p. 16. Disponible en: [http://ir.canterbury.ac.nz/bitstream/10092/3997/1/Thesis\\_fulltext.pdf](http://ir.canterbury.ac.nz/bitstream/10092/3997/1/Thesis_fulltext.pdf) Consultado: Octubre 2011

Lartillot, Olivier y Saint-James, Emmanuel, "Automating Motivic Analysis through the Application of Perceptual Rules", *Computing in Musicology*, Vol. 13, Massachusetts Institute of Technology (MIT), Massachusetts, 2004, pp. 73-92. Disponible en: [http://jyu.academia.edu/OlivierLartillot/Papers/299275/Automating\\_Motivic\\_Analysis\\_Through\\_the\\_Application\\_of\\_Perceptual\\_Rules](http://jyu.academia.edu/OlivierLartillot/Papers/299275/Automating_Motivic_Analysis_Through_the_Application_of_Perceptual_Rules) Consultado: Octubre 2011.

"Las Tonalidades". Blog Lo mejor del mundo: La música. Disponible en: <http://bach2411111.blogcindario.com/2006/04/00130-teoria-musical-armaduras.html> Consultado: Enero 2011.

Noy, Natalya; McGuinness, Deborah. "Ontology Development 101: A Guide to Creating Your First Ontology", Stanford University. Disponible en: [http://protege.stanford.edu/publications/ontology\\_development/ontology101.pdf](http://protege.stanford.edu/publications/ontology_development/ontology101.pdf) Consultado: Noviembre 2010.

"OWL Web Ontology Language. Use Cases and Requirements", W3C Recommendation. Disponible en: <http://www.w3.org/TR/2004/REC-webont-req-20040210/> Consultado: Agosto 2010.

"OWL Web Ontology Language Reference". Disponible en: <http://www.w3.org/TR/2004/REC-owl-ref-20040210/> Consultado: Agosto 2010.

"OWL Web Ontology Language Guide". Disponible en: <http://www.w3.org/TR/2004/REC-owl-guide-20040210/> Consultado: Febrero 2011.

Push Button Bertha, Musica por Datatron, Letra de Jack Owens. Dominio público. Disponible en: <http://datatron.blogspot.com/2009/03/bertha-preserves-datatron.html> Consultado: Octubre 2011

“RDF Primer”. Disponible en: <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/> Consultado: Marzo 2011.

“RDF Vocabulary Description Language 1.0: RDF Schema”. Disponible en: <http://www.w3.org/TR/rdf-schema/>

“Recordare. Partwise DTD”. Disponible en: <http://www.recordare.com/musicxml/dtd/partwise-dtd> Consultado: Enero 2011

“Recordare, MusicXML 2.0 Tutorial”. Disponible en: <http://www.recordare.com/sites/default/files/musicxml-tutorial.pdf> Consultado: Enero 2011

Roads, C., “Artificial intelligence and music”, *Computer Music Journal*, Vol. 4, No. 2, Massachusetts Institute of Technology (MIT), Massachusetts, 1980, P. 17. Disponible en: <http://www.jstor.org/stable/3680079> Consultado: Octubre 2011.

Sebastien, Véronique; Sebastien, Didier; Conruyt Noël, "An Ontology for Musical Performances Analysis: Application to a collaborative platform dedicated to instrumental practice", 2010 Fifth International Conference on Internet and Web Applications and Services, mayo de 2011, p. 538. Consultado: Octubre 2011.

Winograd, Terry, “Linguistics and the Computer Analysis of Tonal Harmony”, *Journal of Music Theory*, Vol. 12, No. 1, Departamento de Música de la Universidad de Yale, New Haven, 1968, p. 3. Disponible en: <http://www.jstor.org/stable/842885> Consultado: Octubre 2011