

# Composición Musical

Dayner Umaña y Esteban Noguera  
CI-1441 Paradigmas computacionales  
Escuela de Ciencias de la Computación e Informática  
Facultad de Ingeniería  
Universidad de Costa Rica  
[darafael.5959@gmail.com](mailto:darafael.5959@gmail.com), [noguera.esteban@gmail.com](mailto:noguera.esteban@gmail.com)

Julio de 2015

## Resumen

En el siglo XV surgió una técnica que busca mantener la armonía entre dos voces independientes entre sí, llamada **contrapunto**. Dicha técnica consiste en que a partir de una melodía inicial (*cantus firmus*), se le debe aplicar una serie de reglas, para crear una segunda voz melódica.

El contrapunto se divide en 5 partes o fases llamadas especies, las especies son numeradas del 1 a al 5. Esta numeración indica la complejidad de las reglas de cada especie, siendo la quinta especie la de mayor complejidad debido a lo complejo de sus reglas.

El Proyecto utiliza como base las reglas del contrapunto del siglo XV, para las cuales se implementa un Sistema utilizando un lenguaje de programación, en este caso JAVA, que sea capaz de interpretar dichas reglas, que aplicadas a un cantus firmus dado genere una segunda voz melódica.

**Palabras clave:** contrapunto, paradigmas, melodía, composición, reglas.

## 1. Introducción

La composición musical consiste en la creación de un conjunto de notas que unidas, forman una melodía. El problema que surge es como hacer para que una máquina sepa que notas debe utilizar para crear una melodía.

El contrapunto necesita de una melodía inicial, pero que a partir de esta se puede generar **una segunda voz melódica**. Las cuales son posibles de encontrar

aplicando las reglas de las 5 especies que definen el contrapunto.

El sistema implementado es capaz de crear una segunda voz y esto es logrado gracias desarrollando un sistema basado en reglas implementado en el lenguaje de programación JAVA.

## 2. Marco Teórico

En la búsqueda de información sobre trabajos hechos para las especies de contrapunto no se encontró algún trabajo que fuese un referente para los objetivos propuestos para este trabajo. Básicamente lo que se hizo fue utilizar toda la teoría referente a las especies de contrapunto, tomadas del libro Ejercicios preliminares de contrapunto (Schoenberg, 1963). Para este caso en particular se utilizó toda la teoría en relación al contrapunto en primera especie. Las especies de contrapunto son una metodología pedagógica escrita por Johann Joseph Fux, en 1725, expuestas en su tratado Gradus ad Parnassum.

## 3. El problema

Crear una composición musical contrapuntística a partir de una melodía inicial (*cantus firmus*), aplicando las reglas del contrapunto del siglo XV. En el caso del contrapunto en primera especie se pretende que el alumno comprenda el manejo melódico y armónico de las notas e intervalos consonantes, a saber: tercera mayor y menor, cuarta justa, quinta justa, octava justa y décima.

## 4. Objetivos

Objetivo general:

1. Crear una aplicación, en un lenguaje de programación capaz de ejecutar reglas del contrapunto, por especies.

Objetivos Específicos:

1. Crear una composición musical, a partir de una melodía inicial (*cantus firmus*), en formato MIDI.
2. Aplicar al *cantus firmus* dado, las reglas del contrapunto del siglo XV.
3. Utilizar el paradigma de sistema basado en reglas, como medio para obtener una solución.
4. Utilizar un lenguaje de programación para la implementación del sistema basado en reglas.

## 5. La propuesta de solución

Se decide crear un sistema basado en reglas utilizando JAVA como lenguaje de programación. Principalmente por su facilidad para añadirle otras funcionalidades, tales como la manipulación de archivos MIDI. Permitiendo pasar de un archivo MIDI, aun conjunto de símbolos que se pueden interpretar fácilmente en JAVA. Dicha funcionalidad es importante debido a que el *cantus firmus*, utilizado por el sistema, es un archivo de este tipo.

En JAVA se debe utilizar el encadenamiento hacia adelante para interpretar las reglas, esto debido a que las decisiones que tomará el sistema deben ser de los hechos construidos hasta el momento.

## 6. Desarrollo, prueba y validación

En primera instancia se intentó implementar un sistema basado en reglas usando Prolog como máquina de inferencia de reglas de composición para crear melodías contrapuntísticas, y posteriormente usando la aplicación de Java para la apertura de archivos MIDI, y enviando la información a Prolog para la derivación de la melodía a partir del *cantus firmus*.

Como no fue posible la implementación y unificación de ambas aplicaciones, se tomó la decisión de realizar la aplicación en su totalidad en Java, logrando terminar con éxito la primera de las cinco especies.

La forma de implementación fue utilizar el algoritmo de las 8 reinas, que es un algoritmo recursivo de búsqueda exhaustiva de una solución, creando un árbol

en la pila de la recursividad no un árbol físico, en cada iteración el algoritmo prueba cada uno de los intervalos que son permitidos en la primera especie, así para cada una de las notas del *cantus firmus*, que es semejante a los hechos dados en Prolog, que son la premisa que se utilizan para el encadenamiento hacia adelante.

Cada vez que prueba cada uno de los intervalos permitido según el compás, hace un incorporación a la memoria de trabajo(*assert*), y posteriormente pasa al siguiente compas tratando de insertar un intervalo permitido y así recursivamente.

Es importante mencionar que la primera especie tiene como premisa realizar todo el movimiento contrario según la naturaleza del *cantus firmus* lo permita.

Como herramienta para la manipulación de archivos MIDI, se utilizó la biblioteca de JAVA, llamada *JFugue*, con la cual es posible convertir un pentagrama a símbolos que pueden ser interpretados fácilmente por JAVA, y de esta manera poder utilizar las notas musicales como variables en funciones, como parámetros, entre otros.

## 7. Experimentación y análisis

Para este experimento en particular se creó un archivo en MIDI llamado cf6.midi que es abierto desde el programa principal, y mapeado según ciertas reglas que van a permitir ejecutar las reglas de contrapunto, una vez que se abre la ventana de la aplicación, seleccionamos el archivo, y posteriormente se da click en *play cantus firmus* para escuchar solamente en *cantus firmus*, además se puede dar click sobre *play composición*, para escuchar la melodía creada. Dentro de esta ventana de usuario se puede ver adicionalmente los dos sistemas que corresponden al *cantus firmus* y a la melodía derivada, para ver las notas correspondientes en cada compás y así verificar el trabajo realizado por la aplicación.

La siguiente imagen muestra el *cantus firmus* dado en el archivo cf6.midi el cual permite ver y analizar la curva melódica correspondiente.

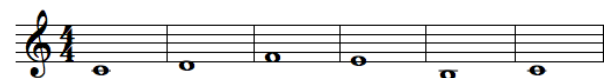
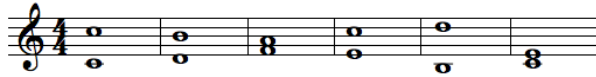


Ilustración 1-Cantus Firmus

Luego de seleccionado el archivo, se puede apreciar en la siguiente imagen el tipo de movimiento creado por la aplicación para la voz derivada. Como se puede apreciar en cada uno de los compases se realiza un

estricto movimiento contrario para cada nota del *cantus firmus*. Este es, como se menciona anteriormente es uno de los propósitos principales del contrapunto en primera especie, ya que logra que las voces creadas se muevan con un grado mayor de independencia, y de hecho al escuchar las melodías juntas es bastante sencillo de apreciar el direccionamiento de cada una de las voces, que a diferencia de los movimientos paralelos u oblicuos son más difíciles de percibir.



**Ilustración 2- Composición Creada**

## 8. Recomendaciones y consideraciones a futuro

En el futuro se desea modificar la aplicación, para que no solo aplique las reglas de la primera especie como actualmente lo hace, sino además poder programar las reglas de las 5 especies que actualmente existen.

Se desea como un extra de la aplicación, que no solo presente en pantalla las notas que conforman la composición creada, como actualmente se hace, sino que integrando un módulo de Matlab a la aplicación de JAVA, se presente en pantalla un pentagrama con la solución obtenida, y de esta manera poder apreciar de una mejor manera la altura de las notas. Esto ya que el pentagrama es la forma por excelencia de entender la música.

## 9. Problemas abiertos y problemas futuros

Inicialmente se propuso utilizar Prolog como lenguaje de programación para implementar un sistema basado en reglas. Pero debido a la poca bibliografía existente sobre el encadenamiento hacia adelante en Prolog, se dificultó el uso de este.

Debido al tiempo tomado para buscar una solución para este problema, se hizo difícil la implementación de reglas del contrapunto de las cinco especies existentes.

## 10. Bibliografía

- Bratko. (s.f.). *UNSW Australia*. Recuperado el 15 de Julio de 2015, de <http://www.cse.unsw.edu.au/~billw/cs9414/notes/kr/rules/forward.pro>
- Bratko, I. (2001). *Programming for artificial intelligence*. Addison-Wesley.
- Schoenberg, A. (1963). *Ejercicios preliminares de contrapunto*. Cooper: SpanPress.

## 11. Agradecimientos

Deseamos agradecer a las siguientes personas:

Edmundo Nuñez Incer, por haber propuesto valiosas ideas para la implementación del proyecto.

Profesor Álvaro de la Ossa por sus aportes y su disposición para colaborar en este proyecto.

Alonso Torres, compañero y profesor de la escuela de Artes Musicales de la UCR, por su aporte en la bibliografía utilizada para el proyecto.