DSLMusic

Dimitry Castex Marcelo Valdivia

4 de octubre de 2013

Resumen

Así como las personas desarrollan la capacidad de redactar sin la necesidad de leer en voz alta lo que escriben, un compositor por medio de la experiencia logra desarrollar la habilidad de imaginar el sonido y crear música sin necesidad escucharla.

Así nace la idea de crear DSLMusic, una herramienta de apoyo a músicos que permite componer melodías de nivel básico y medio que pueden ser reproducidas y exportadas a partituras de forma rápida y limpia, de manera tal que exista una compatibilidad entre lo que el compositor escribe y quién interpreta la música usando algún instrumento.

El desarrollo de este software es posible gracias a ANTLR, una herramienta que proporciona un marco para la construcción de reconocedores, intérpretes, compiladores y traductores de lenguajes a partir de descripciones gramaticales. Usar ANTLR permitirá determinar si una sentencia (para el caso de este software, adición de figuras musicales y notas) pertenece o no a un lenguaje en específico.

Un estudio en profundidad acerca del funcionamiento de ANTLR y otras herramientas necesarias para la creación del producto de software quedará representado dentro de este documento, tanto en su desarrollo como en la sección de anexos.

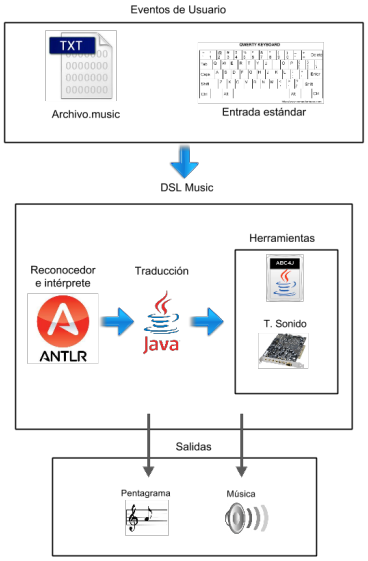
1. Introducción

DSLMusic, es un software que mediante un reconocedor, interprete y traductor de lenguaje, crea música a partir de sentencias gramaticales escritas por los usuarios. Este software posee los elementos básicos necesarios para crear melodías de nivel básico y medio, pues posee un panel de control que permite manipular la duración de un tono, modificar los intervalos entre sonidos (octavas), ingresar notas y silencios de acuerdo a las figuras musicales existentes (redonda, negra, blanca, corchea, etc.) [1] y además, incluye un módulo que se encarga de exportar una melodía a pentagrama, dando la posibilidad de que un usuario externo pueda interpretar una composición usando el lenguaje universal de la música.

2. Modelo

La base de DSLMusic se centra en el reconocimiento, análisis y traducción de distintas sentencias gramaticales ingresadas por el de usuario, de manera tal, que si éstas están correctamente escritas, una composición musical será reproducida y a su vez, un pentagrama gráfico sea generado y mostrado en pantalla.

El flujo de datos queda representado según el siguiente diagrama general:



2.1. Eventos de usuario

Los eventos de usuario son generados a partir de dos entradas, una por archivo de texto de extensión ".musicz otra mediante la entrada estándar de tecla-

[1] figura musical `http://es.wikipedia.org/wiki/Figura_musical`

do. Se requiere que la entrada corresponda a un conjunto de sentencias gramaticales definidas para el DSL construído.

2.2. DSLMusic

DSLMusic una vez que escucha la entrada de usuario, procede a utilizar una serie de herramientas antes de generar las salidas esperadas.

2.2.1. Reconocedor e intérprete

DSL Music para reconocer e interpretar las sentencias gramaticales de las entradas de usuario, usa ANTLR. Con esta herramienta podemos parsear una serie de sentencias ingresadas por el usuario para luego interpretarlas y generar al mismo tiempo alguna acción[2], en este caso, componer melodías.

2.2.2. Traducción

Una vez que es comprobado que la entrada de usuario fue correcta mediante el uso de ANTLR, será necesario crear un parseador interno en Java que permita por un lado, traducir una sentencia del tipo "(NOTAOCTAVA,FIGURAMUSICAL)" a una frecuencia medida en hertz y una duración en milisegundos. Por otro lado, fue necesario generar un segundo parseador para traducir la misma sentencia descrita anteriormente a una aceptada por la librería ABC4J, la cual permite generar pentagramas en forma gráfica [3].

2.2.3. Herramientas

ABC4J ABC4J es una librería para el lenguaje de programación Java que permite generar pentagramas en forma gráfica a partir de un string, el cual debe ser formateado de tal forma que coincida con la notación aceptada por la librería.

Tarjeta de sonido Una vez que el parseador interno traduce una sentencia a frecuencia y duración, esta debe ser interpretada y luego enviada a la tarjeta de sonido como un buffer de salida de bytes. De esta forma, es comprobado que DSLMusic no posee sonidos por defecto, sino que los genera convirtiendo una frecuencia a arreglos de bytes que luego son enviados a un buffer de salida y emitidos por la tarjeta de sonido.

2.2.4. Salidas

Música Mediante una función en Java, los arreglos de bytes que representan una frecuencia medida en Hertz, son expulsados como sonidos, los cuales no sufren variaciones al momento de ejecutar la aplicación en distintos dispositivos, por lo cual, el resultado siempre será el mismo.

Pentagrama Mediante Java y ABC4J, se crea un pentagrama en forma gráfica que representa de forma idéntica la melodía que se ha compuesto.

3. Implicaciones del estudio

La propuesta se inserta dentro del perfil profesional que un Ingeniero en Computación de la Universidad de La Serena desarrolla durante su etapa de formación, específicamente bajo las características de un individuo capaz de adaptarse a la evolución de las tecnologías de la información [4] , pues para llevar a cabo la realización del producto será necesario investigar acerca de librerías acordes al contexto de nuestro proyecto y usar técnicas de programación de acuerdo a los actuales estándares de la ingeniería de software.

Cabe la posibilidad que para los alumnos de este curso, el uso de una herramienta como ANTLR haya sido un tanto desligada de lo que se acostumbra a utilizar. Desde un comienzo se le ha dado énfasis al desarrollo de software acoplado solamente con el uso de base de datos y redes para algunos casos. Debido a esta inclinación generada hacia el desarrollo de software con las características antes mencionadas, se pasan por alto la realización de proyectos para la manipulación de lenguaje de datos, desarrollo de lenguajes de consulta para datos astronómicos, creación de compiladores, etc., proyectos que solo llegan a conocerse al momento de rendir el curso de Teoría de Autómatas y Lenguajes Formales.

Si bien ya nos encontramos en una etapa donde el curso se ha familiarizado con ANTLR y hasta se han creado aplicaciones Java que se acoplan perfectamente con las gramáticas generadas, el contexto de las aplicaciones construidas no se escapa de lo común y es un punto el cual hemos querido atacar.

Más que ir por el contrario a los proyectos de la Universidad de Columbia, la motivación por crear un software que tuviese relación con el arte gráfico o la música era una idea que venía desde mucho antes y se tomó esta oportunidad para combinar una serie de contenidos dentro de un todo, hacer uso de aspectos de la ingeniería de Software, software libre, herramientas y librerías externas e incluirlos bajo un contexto fuera de lo común, para este caso en particular, la música.

Se espera que el desarrollar una aplicación de esta índole pueda ser el impulso inicial para que muchos es-

[2]ANTLR, web oficial `http://www.antlr.org/`

[3]abc4j, `libreriajavaparanotacionmusicalhttps://code.google.com/p/abc4j/`

[4]Perfil profesional de un ingeniero en computacion ULS: `http://www.userena.cl/admision.carrera.html?carrera=3`

tudiantes a futuro, puedan generar nuevas ideas de proyectos tomando en cuenta éste y/o desarrollar nuevas versiones, tomando como base los aspectos de gestión de la configuración de la Ingeniería de Software

Junto a lo mencionado, también se espera que se logre crear un impacto en la sociedad con la generación de aplicaciones ligadas al arte en general, fomentando así el desarrollo de la cultura.

4. Contexto

DSLMusic puede usarse por una persona o músico que necesite elaborar una partitura que él conoce, de forma rápida y limpia, la cual tenga directa relación con una obra vocal o instrumento solista (el software se limita a trabajar con un instrumento a la vez). [5]

La ventaja principal que posee este software frente otros más sofisticados como "Finale" (software líder en el área de las aplicaciones para generar partituras) [6] , es que para obtener una simple partitura, requiere una menor curva de aprendizaje de los usuarios sobre el uso de la herramienta, pues no generaría mayor complicación a personas con conocimientos básicos de composición y un nivel novato en cuanto al uso de computadoras y software.

En cuanto a usuarios menos experimentados en el tema de la música, DSLMusic ofrece un panel de ayuda que permite escuchar notas musicales antes de ser agregadas a una partitura, permitiendo la creación de melodías sin la necesidad de ser un experto en el tema.

5. Ciclo de Vida Adoptado

El ciclo de vida elegido para este proyecto es Incremental - Evolutivo debido a las características del problema:

- Los requisitos cambian conforme al desarrollo del producto final.
- Debido a las fechas impuestas por término de semestre, es necesario introducir una versión limitada y funcional del producto.
- Los requisitos centrales están bien definidos y las extensiones del producto aún no son totalmente definidas.
- Para obtener un producto final, es necesario un mayor número de iteraciones, motivación principal para quienes rindan el curso a futuro y puedan generar nuevas versiones del software y satisfacer nuevos requisitos.
- La cantidad de desarrolladores es baja, no es suficiente para hacer un software sofisticado en el tiempo dado, se prioriza los requerimientos que permiten desarrollar el núcleo del software.

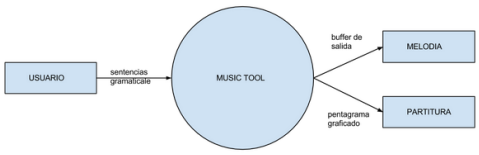
6. Modelamiento

La mejor forma de obtener un producto que cumpla en su plenitud tanto los requerimientos funcionales como los no funcionales, es enfocándose en realizar un buen diseño de software. Para esta etapa, es necesario usar técnicas UML para modelar los flujos y procesos involucrados en la aplicación de manera que en la etapa de implementación del software, los desarrolladores tengan alguna herramienta de comunicación y puedan desarrollar el producto final sin errores de mayor índole.

Para este proyecto se decidió utilizar el modelamiento por diagrama de flujo de datos, el cual es una representación gráfica del flujo de datos a través de un sistema informático. Un diagrama de flujo de datos también se puede utilizar para la visualización de procesamiento de datos (diseño estructurado). [7] Es una práctica común para un diseñador dibujar un contexto a nivel de DFD que primero muestra la interacción entre el sistema y las entidades externas.

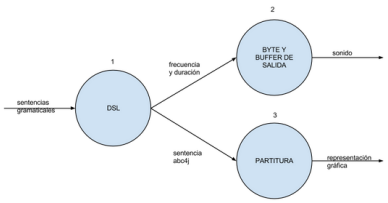
Los elementos que componen un diagrama de flujo de datos son cuatro, de los cuales tres son los ocupados para representar DSL Music: Proceso (se representa mediante un círculo), Entidad Externa (corresponde a un rectángulo) y el Flujo de Datos (representado con una flecha).

6.1. Diagrama de contexto (Nivel0)

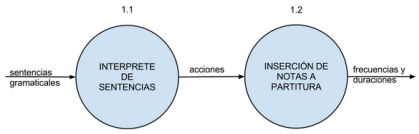


[5] monofonía: http://es.wikipedia.org/wiki/Monofon%C3%ADa
[6] Finale, estandar mundial en notacion musical http://www.finalemusic.com/
[7] Data flow diagram: http://en.wikipedia.org/wiki/Data_flow_diagram

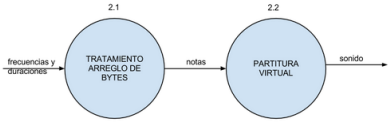
6.2. Diagrama de Nivel Superior (Nivel 1)



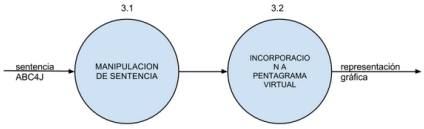
6.3. Diagrama de detalle DSL (nivel 2)



6.4. Diagrama de detalle: byte y buffer de salida(Nivel 2)



6.5. Diagrama de detalle: pentagrama(Nivel 2)



7. Generación del sonido: cuestionamiento central

El sonido se produce cuando hay una variación en la presión dentro de un medio. Esta variación es audible cuando el movimiento se produce dentro de un rango de velocidad determinada. Por ejemplo, una señal continua, aunque existe "presión", no es audible por no existir fluctuaciones. Por ello, cuando se muestrea y se visualiza un sonido suele ser una señal oscilante.

Cuando esta oscilación se realiza a una frecuencia determinada constante, no solo se emite un sonido, si no que se emite un tono. La variación de la frecuencia de la señal emitida provoca la modificación del tono. Para la creación de notas musicales se han estipulado frecuencias concretas siendo la referencia la nota LA (A) pudiendose obtener a partir de la siguiente fórmula $55Hz * 2^{octava}$: $55Hz$, $110Hz$, $220Hz$, $440Hz$, $880Hz$, etc (contando las octavas en base 0) [8].

El resto de notas siguen una función exponencial con la fórmula $frecuencia = 55Hz \cdot 2^{octava+(nota/12)}$, igualmente asumiendo que la numeración de las octavas y notas están en base 0, es decir la octava 1 es la 0 y la nota A es la 0.

En cuanto al muestreo del sonido hay que tener en cuenta diferentes parámetros de configuración. Uno de estos parámetros es la cantidad de canales que se desea muestrear concurrentemente. Normalmente suelen utilizarse dos modos: 1 canal (mono) y 2 canales (estéreo). Otro parámetro a configurar es el número de bits que representa la precisión de cada muestra En la actualidad lo habitual es 8bits, 16bits y 32bits. Finalmente, la frecuencia de muestreo representa cada cuanto tiempo tomaremos una muestra. Lo habitual es 11.025Hz, 22.050Hz y 44.100Hz [9] . La configuración preferencial para crear un teremin virtual será con un muestreo en mono (solo 1 canal), 8bits que se reflejará en el tipo de variable a utilizar (byte) y 22.000Hz que afectará a la cantidad de bytes que se introducirán en el buffer de salida en 1 milisegundo.

8. DSLMusic: Funcionamiento y tutorial del lenguaje

DSLMusic posee un lenguaje propio pero con un toque de algunos lenguajes como los vistos a lo largo de nuestra formación en la Universidad como lo son Pascal, C, C♯ y el más usado Java, pues las sentencias gramaticales que dan vida a las melodías son similares a la declaración de funciones o más bien llamados métodos, los cuales reciben por parámetro una serie de valores y Strings. Además, cada sentencia del tipo mencionado, debe ser finalizada por medio de la puntuación coma ",".

DSLMusic posee dos formas de composición de música, una mediante sentencias gramaticales en formato musical estándar, el cual recibe en sus parámetros una nota musical, octava y figura musical. En cuanto al otro método de composición, solo es necesario incluir una frecuencia en hertz y una duración en milisegundos.

[8] Nota musical LA: `http://en.wikipedia.org/wiki/Data_flow_diagram`
[9] Frecuencias de muestreo: `http://es.wikipedia.org/wiki/Audio_digital`

8.1. Parámetros de nota, octava y figura musical

Para generar una melodía a través de este método, es necesario hacer una composición de la siguiente manera:

```
UNIDAD=1000MS;
NOTA(LA4,REDONDA);
NOTA(SOLS4,REDONDA);
SILENCIO(NEGRA);
END
```

Primero, con UNIDAD se declara la unidad de tiempo en segundos o milisegundos para la duración de una figura musical redonda". A modo de ejemplo, si una figura musical vale 1000MS, una figura blanca durará 500MS y una negra 250MS.

Para añadir notas o silencios a una partitura, se hace la declaración en este caso NOTA(LA4,REDONDA); en donde, primero se especifica al comienzo de la sentencia si nos encontramos con una nota o un silencio, seguido de una entrada de parámetros donde se recibe la nota seguido del valor de la octava (intervalo para separar sonidos) y luego en el siguiente parámetro, se especifica la figura musical, para este caso una redonda.

Para finalizar la melodía, se agrega la sentencia END y si no hay errores de sintaxis, la composición estaría lista para ser reproducida.

8.2. Parámetros de frecuencia y duracion

Para generar una melodía a través de este método, es necesario hacer una composición de la siguiente manera:

```
SONIDO(440HZ,1000MS);
SILENCIO(10MS);
END
```

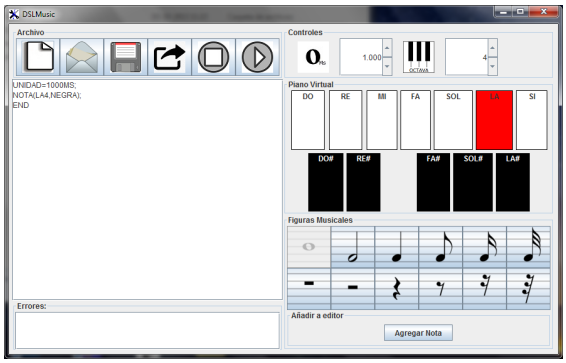
A diferencia del método anterior, aquí no es necesario declarar la unidad de tiempo al comienzo. Para agregar un sonido (como se ingresa en las frecuencias que uno desee, ya no los consideramos como notas) , tomando como ejemplo la sintaxis SONIDO(440HZ,1000MS); se aprecia que primeramente se hace la distinción entre un silencio y un sonido. Luego, como primer parámetro se espera una frecuencia seguido de su medida en HZ. El último parámetro recibe la duración del sonido, expresado en milisegundos. Para finalizar la composición, se añade la sentencia END y ya puede ser reproducida.

Se debe resaltar que para generar un pentagrama con figuras musicales, este método de composición no es el indicado, puesto que no se hace representación de notas exactas y menos de figuras musicales. Para exportar a representación gráfica es estrictamente necesario componer basándose en el primer método mencionado.

9. Inicio de la aplicación

Paso 1: el usuario debe hacer doble clic sobre la aplicación llamada DSLMusic.jar. Si el usuario posee la máquina virtual de Java en su equipo, la aplicación comenzará su ejecución, de modo contrario, se debe descargar desde la página oficial de Java.

Una vez ejecutada la aplicación, se presenta la interfaz gráfica de usuario.



Paso 2: la composición se realiza mediantes dos entradas de usuario, una por archivo y otra por teclado. Ambas entradas escriben sentencias gramaticales dentro del panel de composición de la aplicación.

Para guardar una composición realizada, se debe presionar el botón de diskette. Para abrir una ya realizada, se debe presionar el boton con ícono de carta.

Paso 3: Para aquellos usuarios menos experimentados con la aplicación, se creó el panel de ayuda al lado derecho, en donde, es posible escuchar las notas musicales antes de añadirlas a una composición. Una vez que el usuario esté decidido, presionando el botón "agregar nota", la aplicación toma la nota musical marcada en rojo desde el piano virtual, la figura musical presionada y la octava, y agrega una sentencia gramatical a la composición con el siguiente formato (DOS4,REDONDA);.

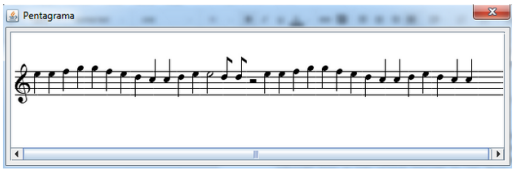
Paso 4: Cada vez que el usuario ingrese sentencias incorrectas, el panel de errores dará aviso al usuario y le dirá la o las líneas donde se encuentran errores léxicos.

Paso 5: En el panel superior de los controles se encuentran dos herramientas, una para asignar tiempo en milisegundos a una figura musical REDONDA y con esto, asignar los tiempos al resto de las figuras musicales. A modo de ejemplo, si asignamos un valor de REDONDA=1000MS, lla correspondencia sería BLANCA=500MS, NEGRA=250MS, CORCHEA=125MS, SEMICORCHEA=62.5MS y FUSA=31.25MS.

El control para "corchea" permite definir los intervalos entre notas, haciendo que una de éstas en octava 4 emita un sonido más grave que una en octava 6.

Paso 6: Una vez terminada una composición y esta esté libre de errores, para iniciar la reproducción de la melodía basta con presionar el botón "Play".

Paso 7: Si la composición está escrita correctamente y con el método de nota, octava y figura musical, es posible hacer la exportación a pentagrama presionando el botón de exportación (cuarto botón en el panel de archivo). La generación es la siguiente:



Paso 8: Aquí se deja la composición de la primera parte del himno de la alegría.

```
UNIDAD=300MS;
NOTA(MI4,NEGRA);
NOTA(MI4,NEGRA);
NOTA(FA4,NEGRA);
NOTA(SOL4,NEGRA);
NOTA(SOL4,NEGRA);
NOTA(FA4,NEGRA);
NOTA(MI4,NEGRA);
NOTA(RE4,NEGRA);
NOTA(DO4,NEGRA);
NOTA(DO4,NEGRA);
NOTA(RE4,NEGRA);
NOTA(MI4,NEGRA);
NOTA(MI4,BLANCA);
NOTA(RE4,CORCHEA);
NOTA(RE4,CORCHEA);
SILENCIO(BLANCA);
NOTA(MI4,NEGRA);
NOTA(MI4,NEGRA);
NOTA(FA4,NEGRA);
NOTA(SOL4,NEGRA);
NOTA(SOL4,NEGRA);
NOTA(FA4,NEGRA);
NOTA(MI4,NEGRA);
NOTA(RE4,NEGRA);
NOTA(DO4,NEGRA);
NOTA(DO4,NEGRA);
NOTA(RE4,NEGRA);
NOTA(MI4,NEGRA);
NOTA(RE4,NEGRA);
NOTA(DO4,NEGRA);
NOTA(DO4,NEGRA);
END
```

La composición puede ser copiada y pegada en el panel de composición de DSLMusic para luego comenzar con la emisión del sonido y a su vez, mostrar la partitura asociada.

10. Conclusión

Este trabajo ha permitido incursionar en materias desconocidas tanto por el lado de la Ingeniería de Software como por el lado artístico.

Aveces creemos que por el hecho de estudiar una ingeniería somos superiores a otros profesionales que se desempeñan en otras áreas, específicamente para este caso, la música. Fue bastante complicado entender ciertos conceptos de la teoría musical, específicamente en el área de escritura de figuras musicales en pentagrama. Creímos que sería una tarea fácil, pues asimilamos que los conceptos necesarios para llevar a cabo el proyecto tendrían directa relación con los conocimientos que en algún instante adquirimos durante la educación básica y media.

La teoría musical es un tema bastante complejo y por lo mismo, el proyecto tuvo que ser acotado de acuerdo al tiempo disponible para llegar a un producto final. Hubo conceptos que quedaron pendientes y que por razones de tiempo no pudieron ser implementados, temas relacionados con compases, metrónomos, BPM, unión de figuras musicales, entre otros. [10] pesar de esto, estamos bastante conformes con el resultado, pues se ha logrado crear una aplicación con un lenguaje propio para la composición de melodías de nivel básico y medio, siendo probada y validada por dos alumnos de pedagogía en música de la Universidad de La Serena, Adolfo Ramírez y Variniha Badilla.

En cuanto al lenguaje de dominio específico creado y a su implementación, queremos hacer referencia a ANTLR y al curso de Teoría de Autómatas y Lenguajes Formales. De no ser por la parte teórica del curso rendida en este semestre, no podríamos haber hecho el lazo con la parte práctica. En un comienzo no podíamos distinguir la diferencia entre usar ANTLR o realizar la misma tarea con algún tipo de código Java.

[10] Notacion ABC nivel avanzado `http://abcnotation.com/`

Mientras pasó el tiempo, nos dimos cuenta de lo poderosa que es esta herramienta y más aún, el cuanto facilita el trabajo que se deseaba desarrollar. Llegamos a un punto donde cada tipo de reconocimiento, análisis y traducción de sentencias gramaticales fue enlazada casi automáticamente al uso de ANTLR.

Estamos seguros que el producto final generado da para más, tenemos la esperanza de que se puede llegar a crear un software más sofisticado tomando éste como base, incluyendo aquellos aspectos de la teoría musical que quedaron inconclusos y más aun, considerando que se logró un desarrollo modular en la implementación, proporciona una ventaja para aquellos futuros ingenieros que deseen incursionar en áreas poco incursionadas por los alumnos de la carrera de ingeniería en computación. La idea fue crear una aplicación novedosa, enfocada al lado de las artes y capaz de realizar alguna acción donde pudiesen ser integrados los conceptos más fuertes del curso, estamos conformes de haber podido encontrar el lazo entre la teoría y la práctica.