



Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI-2691 - Laboratorio de Algoritmos y Estructuras I
Trimestre Abril-Julio 2018

Proyecto: Compositor musical

El objetivo de este proyecto es la implementación de un programa no trivial, haciendo uso del análisis descendente y de las técnicas vistas a lo largo del curso de teoría y práctica de Algoritmos y Estructuras I

Planteamiento del Problema

Existen muchas herramientas para representar, crear y analizar música en el computador. Una de estas herramientas es music21, una biblioteca en Python de representación y análisis musical. Esta biblioteca fue creada para musicólogos, pero cuenta con muchas funciones que permiten representar, manipular y hasta crear música de manera sencilla.

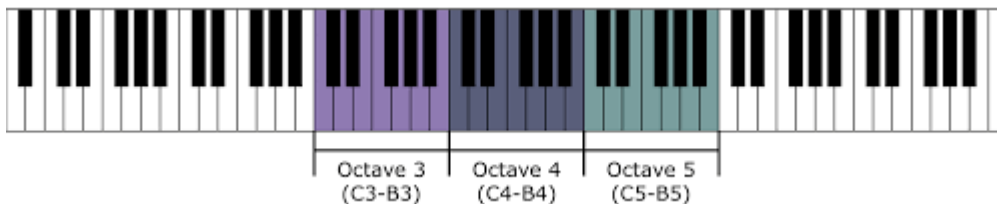
El primer paso para poder utilizar music21 es instalar la biblioteca (ver instrucciones en la sección “Instalación de music21”). Una vez instalada, para comenzar a usarla podemos escribir en un intérprete de Python:

```
>>> from music21 import *
```

En music21, las notas se representan mediante el tipo de datos note. Por ejemplo, si ejecutamos la siguiente línea, asignamos la nota F5 a la variable f

```
>>> f = note.Note('F5')
```

Aquí, F se refiere a la nota (Fa), mientras que 5 se refiere a la octava, es decir, qué tan grave o agudo es el Fa. En la siguiente figura se muestran todas las octavas de un piano, y podemos ver que la octava central es la octava 4.



En este momento, la nota que creamos está guardada en la variable f. Podemos verificarlo escribiendo f en el intérprete:

```
>>> f
<music21.note.Note F>
```

Podemos cerciorarnos de que sea un Fa y en la octava 5 revisando los atributos `.name` y `.octave` de la nota:

```
>>> f.name
'F'
>>> f.octave
5
```

Además de notas, es posible representar melodías en music21. Una melodía puede expresarse simplemente como una lista de notas, utilizando la estructura de datos lista que ya conocemos. Supongamos que tenemos dos notas, un Do y un Fa# en la octava central:

```
>>> nota1 = note.Note('C4')
>>> nota2 = note.Note('F#4')
```

Podemos crear una melodía formada por esas dos notas de la siguiente manera:

```
>>> listaDeNotas = [nota1, nota2]
```

Podemos verificar que `listaDeNotas` contiene nuestra melodía imprimiéndola:

```
>>> print(noteList)
[<music21.note.Note C>, <music21.note.Note F#>]
```

La mayoría de la música que conocemos no está formada por una sola melodía, sino por varias melodías que suenan simultáneamente. En music21, es posible crear una composición que tenga varias partes, donde cada parte es una melodía que suena al mismo tiempo que las demás. En el ejemplo que sigue, se crea una composición de dos partes, la primera contiene la nota C y la segunda, la nota G.

```
>>> cancion = stream.Score()
>>> p0 = stream.Part()
>>> p1 = stream.Part()
>>> p0.append(note.Note('C', type="whole"))
>>> p1.append(note.Note('G', type="whole"))
>>> cancion.insert(0, p0)
>>> cancion.insert(0, p1)
>>> cancion.show('text')
{0.0} <music21.stream.Part 0x2d88bb0>
      {0.0} <music21.note.Note C>
{0.0} <music21.stream.Part 0x416f650>
      {0.0} <music21.note.Note G>
```

Hasta ahora hemos visto como imprimir por consola las notas, melodías y composiciones. Pero en music21 también es posible reproducir el sonido de todos estos elementos. Para escuchar una nota, podemos usar la función `show('midi')`:

```
>>> f = note.Note('F5')

>>> f.show('midi')
```

Esto crea un archivo MIDI que se puede escuchar en cualquier reproductor de archivos multimedia. También es posible reproducir las composiciones directamente desde la consola, por ejemplo:

```
>>> sp = midi.realtime.StreamPlayer(p0)
>>> sp.play()
>>> sp = midi.realtime.StreamPlayer(cancion)
>>> sp.play()
```

Una ventaja de music21 es que permite cargar melodías almacenadas en un archivo. En Moodle, junto con el enunciado de este proyecto, puede encontrar un archivo llamado `cancion.tinynotation` que contiene una melodía. Para escucharlo, ejecute:

```
>>> s = converter.parse('cancion.tinynotation')
>>> sp = midi.realtime.StreamPlayer(s)
>>> sp.play()
```

Un concepto muy utilizado en música es el concepto de intervalo. Un intervalo es la distancia entre dos notas. Si las dos notas son la misma, es decir, tienen la misma altura, el intervalo entre ellas es un unísono (representado en music21 como el intervalo “P1”). Los intervalos más comunes que se utilizan en teoría musical se muestran en la siguiente tabla.

Intervalo	Representación en music21
Unísono	“P1”
Segunda menor	“m2”
Segunda mayor	“M2”
Tercera menor	“m3”
Tercera mayor	“M3”
Cuarta justa	“P4”
Quinta justa	“P5”
Sexta menor	“m6”
Sexta mayor	“M6”
Séptima menor	“m7”
Séptima mayor	“M7”
Octava justa	“P8”

Transportar una melodía significa sumarle un intervalo ascendente o descendente a todas sus notas. Al transportar una melodía, sonará más aguda o más grave, pero la melodía no cambia. En otras palabras, cambian las alturas de las notas pero no cambia el intervalo entre ellas, por lo tanto la melodía se percibe igual.

En music21 es posible transportar tanto las melodías como las notas individuales, utilizando el método `.transpose()`, que recibe como parámetro un intervalo. Este método transporta la melodía por el intervalo indicado. Por ejemplo:

```
>>> s.transpose("M3")
```

Un arpeggio es una sucesión de notas separadas todas por el mismo intervalo. Un arpeggio puede ser ascendente o descendente. En este ejemplo, se construye un arpeggio ascendente de ocho notas separadas por una tercera menor (debe haber instalado la librería pygame previamente, ver sección “Instalación de librerías”):

```
arpeggio = stream.Part()
nota = note.Note("C4")
for i in range(0, 8):
    arpeggio.append(nota)
    nota = nota.transpose("m3")
sp = midi.realtime.StreamPlayer(arpeggio)
sp.play()
```

Requerimientos del proyecto

En este proyecto, se requiere que usted desarrolle un programa que permita representar, crear, manipular y escuchar una composición de hasta cuatro partes. El programa se podrá utilizar mediante un menú interactivo con las siguientes opciones:

1	Parte 1
2	Parte 2
3	Parte 3
4	Parte 4
5	Escuchar toda la composición
6	Salir

Las opciones 1 a la 4 llevarán al submenú de cada parte, y este submenú tendrá las siguientes opciones:

1. Cargar de un archivo
2. Generar arpeggio
3. Transportar
4. Escuchar parte
5. Borrar parte
6. Volver al menú anterior

La primera opción permitirá cargar una melodía de un archivo, que se asignará a la parte seleccionada.

La segunda opción permitirá generar un arpeggio a partir de un intervalo indicado por el usuario. El arpeggio se asignará a la parte seleccionada.

La tercera opción permitirá transportar la parte por el intervalo indicado.

La cuarta opción permitirá reproducir la parte individualmente, sin escuchar las otras partes.

La quinta opción borrará la parte seleccionada.

La sexta opción permitirá regresar al menú principal.

Instalación de librerías

music21 (o pygame)

Para instalar music21 (o pygame) puede hacerlo mediante el programa pip, el cual es un sistema de gestión de paquetes utilizado para instalar y administrar paquetes de software escritos en Python.

El programa pip viene instalado por defecto en Python 2.7.9 y versiones más nuevas. Para el proyecto se recomienda Python 3 o superior.

Una vez que se asegure que tiene conexión a internet. Abra un terminal en modo administrador.

Por ejemplo, en el caso de Windows 7 puede seleccionar en Inicio, luego Accesorios y luego la consola ("Símbolo del Sistema") (esta última opción con clic derecho del mouse) y seleccione "Ejecutar como administrador". Seguidamente ejecute en la consola:

```
pip install music21 (pip install pygame)
```

En Linux, en caso de no abrir una consola como root, ejecute:

```
sudo pip install music21 (sudo pip install pygame)
```

Una vez instalada la librería music21 y para comenzar a probarla, ejecute desde Python el siguiente comando:

```
>>> from music21 import *
```

Estructura general del programa

A continuación, se sugiere una estructura bastante general para dividir la funcionalidad de su programa:

Programa principal Menú. Desde este programa principal se hace la llamada a los siguientes subprogramas:

- Subprograma 1: Configurar Parte
- Subprograma 2: Escuchar Composición

A su vez el subprograma 1 hace la llamada a los siguientes subprogramas:

- Subprograma 1.1: Cargar Archivo
- Subprograma 1.2: Generar Arpeggio
- Subprograma 1.3: Transportar
- Subprograma 1.4: Escuchar Parte

Recuerde que la estructura anterior es de referencia. Sea que el equipo siga esta estructura o plantee la suya propia debe seguir las pautas del análisis descendente del curso de teoría y el de práctica.

Especificaciones generales de la interfaz

- El requerimiento es programar un menú tipo consola, es decir, sin elementos gráficos como botones u otros, donde la interacción es vía teclado.
- Si se programan los menús con elementos gráficos y se puede interactuar con el mouse, esto será un plus en el proyecto y será reconocido con hasta 6% adicional a la nota total del proyecto. Se sugiere el uso de la librería Tkinter.

Condiciones de la entrega

El proyecto se realizará por equipos de dos personas pertenecientes a la misma o a diferentes secciones. El proyecto consta de dos entregas:

Entrega 1 (10%)

Implementación de los siguientes requerimientos:

- El menú principal con la siguiente funcionalidad:
 - Al seleccionar las opciones 1 a la 4 se muestra el submenú correspondiente.
 - La opción 5 mostrará un mensaje y permitirá devolverse al menú principal.
 - La opción 6 permitirá salirse del programa principal.
- El submenú para las opciones 1 a la 4 se mostrará con la siguiente funcionalidad:
 - Al seleccionar la opción 1 se debe cargar un archivo y reproducir la melodía cargada.
 - Las opciones 2, 3, 4 y 5 deben mostrar un mensaje y permitir regresar al submenú.
 - La opción 6 debe permitir volver al menú principal.

La primera entrega debe ser subida al aula virtual en un archivo llamado Entrega1ci2691AbrJul118_X_Y.zip, siendo X y Y los carnés de los integrantes del equipo, a más tardar para el miércoles 27 de junio a las 4:30 pm. Debe subirse al espacio del aula virtual de cada integrante. Además, deben entregar impresa la “Declaración de Autenticidad para Entregas” firmada por los integrantes del equipo (puede entregarse en el casillero de uno de los profesores).

Entrega 2 (25%)

Implementación del proyecto completo. La segunda entrega debe ser subida al aula virtual en un archivo llamado Entrega2ci2691AbrJul18_X_Y.zip **(18%)**, siendo X e Y los carnés de los integrantes del equipo, a más tardar para el miércoles 11 de julio a las 4:30pm. Debe subirse al espacio del aula virtual de cada integrante.

Además, deben entregar:

- Impresa la “Declaración de Autenticidad para Entregas” firmada por los integrantes del equipo (puede entregarse en el casillero de uno de los profesores).
- Un informe que deben subir a su espacio de aula virtual y que tenga la siguiente estructura **(7%)**:
 - Portada: Debe incluir los nombres y los números de carnés de los integrantes del equipo, así como el nombre del profesor encargado de cada integrante.
 - Introducción: Breve descripción del problema resuelto, cuáles son los objetivos planteados y el alcance de la solución. Debe indicar cuál es el contenido del informe.
 - Diseño: Explicación del análisis descendente hecho para resolver el problema. Descripción de las estructuras de datos utilizadas. Explicación de los subproblemas claves del programa.
 - Estado actual del proyecto: Se debe indicar el grado de operatividad del programa, es decir, si funciona perfectamente o no; en caso negativo indique cuáles son los errores.
 - Conclusiones: Resultados obtenidos, dificultades presentadas, su visión sobre la experiencia del desarrollo del proyecto.
 - Bibliografía.

Consideraciones finales

- Es obligatorio que el equipo trabaje de forma coordinada. El profesor podrá hacer un interrogatorio individual a cada integrante del equipo, y el no responder adecuadamente a las preguntas puede tener como consecuencias penalizaciones en la nota del proyecto.
- Cualquier error que sea hallado en este enunciado, así como cualquier tipo de observación adicional sobre el proyecto, serán publicadas como fe de erratas en la página web del curso.
- No debe haber copia, ni intercambio de información específica, ni ayuda detallada entre los equipos. El incurrir en cualquiera de las acciones descritas anteriormente tendrá como consecuencia sanciones severas.
- Si el código de alguna entrega no puede ser ejecutado por el interpretador, entonces la entrega será calificada con cero.
- El no cumplimiento de todos los requerimientos podrá resultar en el rechazo de su entrega.