



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2233 - Programación Avanzada  
1<sup>er</sup> semestre 2016

# Actividad 05

## Funcional 1

### Instrucciones

Eres un fanático de la música: Ves música, respiras música, consumes música, y obviamente, escuchas música. Tu pasión en el área, y además tus habilidades de programación te motivan a buscar trabajo en **Apple**, en particular, con el reproductor *iTunes*. Sin embargo, como ven que ni siquiera has completado el curso de Programación Avanzada, te echan sin siquiera cuestionárselo. Estás a punto de renunciar a tu sueño, hasta que ves que en el edificio de al lado se encuentra la compañía **Pear**, donde coincidentemente necesitan con urgencia programadores que puedan aportar a su nuevo reproductor, *yoMelodías*. Como al parecer no les importa tu experiencia previa en la Universidad, ves que la vida te da una segunda oportunidad y decides cumplir tu sueño (o bueno, algo parecido).



Figura 1: El logo de la original empresa y su reproductor.

En la compañía, te dicen que lo que buscan primordialmente, por ahora, es que puedas usar la información que tienen en su base de datos para modelarla con clases y responder algunas consultas que les podrían servir a futuro (algunas...). Para ello, te entregan los siguientes archivos:

- **Artistas.txt:** Archivo de texto que contiene la información de los artistas<sup>1</sup> dentro de la base de datos, de la forma: Nombre Artista|Género|Año formación
- **Discos.txt:** Archivo de texto que contiene la información de los discos pertenecientes a los artistas dentro de la base de datos, de la forma: Nombre Disco|Nombre Artista|Año Salida
- **Canciones.txt:** Archivo de texto que contiene la información de las canciones pertenecientes a los discos dentro de la base de datos, de la forma: Nombre Canción|Nombre Disco|Duración
- **Usuarios.txt:** Archivo de texto que contiene la información de los usuarios que tienen una cuenta, de la forma: Nombre Real Usuario|Username|Fecha de ingreso
- **Ratings.txt:** Archivo de texto que contiene la información de los ratings dados por usuarios a canciones de los artistas, de la forma: Username|Nombre Canción|Rating

## Requerimientos

A pesar de haberte contratado sin importarte tu experiencia, quieren comprobar que tienes lo que se necesita para ser un orgulloso programador de **Pear**, por lo que te piden lo siguiente, incluyendo ciertas restricciones especificadas en **Notas**.

- La lectura de archivos debe realizarse exclusivamente mediante el uso de generadores. La idea es no almacenar todo el archivo en memoria. **HINT:** al abrir un archivo en python, el objeto tipo “file” ya es un generador.
- Cada entidad debe tener un *id*, el que debe ser único para cada instancia. Estos identificadores deben ser creados exclusivamente mediante **generadores**.
- Todas las instancias creadas deben ser almacenadas en listas (lista de artistas, lista de discos, etc.) para su uso posterior.

---

<sup>1</sup> Artistas completamente originales y dedicados, para nada creados a través de la desconocida página <http://www.tunevault.com/band-name-generator/>

- El programa debe poder realizar, además, las siguientes consultas<sup>2</sup>:
  1. **Usuarios ordenados por vejez**<sup>3</sup>: Debes obtener la lista de todos los usuarios, pero estos ordenados por la fecha de ingreso, e imprimir: Posición del usuario, nombre del usuario, fecha de ingreso e id del usuario.
  2. **Canciones ordenadas por rating**: Debes obtener la lista de todas las canciones, pero estas ordenadas por su rating promedio, e imprimir: Posición de la canción, nombre de la canción, nombre del disco al que pertenece, rating promedio e id de la canción.
  3. **Discos ordenados por rating**: Debes obtener la lista de todos los discos, pero estos ordenados por el rating promedio<sup>4</sup>, e imprimir: Posición del disco, nombre del disco, nombre del artista al que pertenece, rating promedio e id del disco.
  4. **Artistas ordenados por rating**: Debes obtener la lista de todos los artistas, pero estos ordenados por el rating promedio<sup>5</sup>, e imprimir: Posición del artista, nombre del artista rating promedio e id del artista.

## Notas

- Los ciclos *for* y *while* solo son permitidos dentro de un **generador**, de una **lista de comprensión**, o para imprimir resultados. No los puedes utilizar bajo ningún otro contexto.
- Todas las consultas deben ser realizadas a través de **programación funcional**. No se permite solucionarlas con otras herramientas<sup>6</sup>.
- No todas las canciones tienen un rating asignado. **Recuerda considerar este hecho**.
- No hay ni bandas con el mismo nombre, ni discos con el mismo nombre, ni canciones con el mismo nombre, para que no te preocupes por eso<sup>7</sup>.
- El archivo **Resultados.txt** te mostrará los resultados que deberías obtener en base a tus consultas. Úsalo como referencia para ver si obtienes los resultados correctos.

<sup>2</sup> Para tu comodidad, al final del programa solo muestra los 10 primeros resultados por consulta.

<sup>3</sup> Esta consulta no le sirve a **Pear** para trabajar, pero te servirá a ti para practicar con algo sencillo :)

<sup>4</sup> No lo obtengas promediando el rating promedio de cada una de sus canciones. Esto añade sesgo y afecta los resultados.

<sup>5</sup> Al igual que con los discos, no promedies promedios (valga la redundancia).

<sup>6</sup> Si tienes dudas con respecto a las funcionalidades que puedes utilizar, pregúntale a alguno de los ayudantes antes de seguir con el desarrollo.

<sup>7</sup> Puede que existan bandas con el mismo nombre que canciones o discos, y lo mismo entre canciones y discos, pero esto es irrelevante al momento de crear las instancias de las clases.

## To - DO

- Definir cada clase correctamente con los atributos del archivo, y crear una lista con **todas** las instancias de esta luego de leer los archivos (0.20 pts por clase)
- Crear un identificador a través de un generador de la forma pedida (1.00 pts)
- Realizar la consulta 1 correctamente (1.00 pts)
- Realizar la consulta 2 correctamente (1.00 pts)
- Realizar la consulta 3 correctamente (1.00 pts)
- Realizar la consulta 4 correctamente (1.00 pts)

## Tips

- Para las fechas, se recomienda usar la librería de Python **datetime**<sup>8</sup>. Esta te ayudará a realizar la primer consulta de una forma más sencilla.
- Por simplicidad, puedes dejar el atributo de duración de una canción como string<sup>9</sup>.

## Entrega

- **Lugar:** GIT - Carpeta: Actividades/AC05
- **Hora:** 16:55

---

<sup>8</sup> Ver: <https://docs.python.org/2/library/datetime.html>

<sup>9</sup> Perdón por tantos pie de página. ¡Éxito!