

# Taller 4

Estructura de Datos y Laboratorio

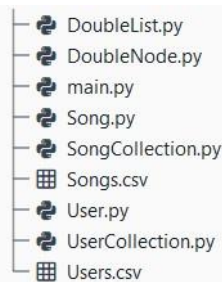
Docente: Ricardo Franco Ceballos

Instituto Tecnológico Metropolitano

**Instrucciones:** Lea atentamente las instrucciones. El correcto seguimiento de éstas también hace parte de la calificación.

- Si el código presenta **errores de ejecución**, automáticamente la nota será de **0,0**.
- Igualmente, se penalizará la **copia** de códigos. Sean honestos con ustedes mismos.
- **TODOS** los códigos utilizados deben ir debidamente marcados en comentarios en la parte superior. Indicando los integrantes y el grupo al cual pertenecen.
- Envíe los códigos utilizados en un **comprimido** al correo [ricardofranco9295@correo.itm.edu.co](mailto:ricardofranco9295@correo.itm.edu.co). Con el asunto taller 4 – Grupo X, indique en el cuerpo del correo los nombres de todos los integrantes.
- El límite para la recepción de la solución es el 11 de mayo de 2022 a las 23:59.
- Los métodos implementados únicamente pueden hacer uso de Listas Dobles. **NO** utilizar: arreglos, listas de Python, listas simples, etc. El uso de otra estructura le otorgará una nota de 0.0

**Ayuda: Deben quedarle estos archivos.**



Implemente los códigos teniendo en cuenta los siguientes diagramas de clase y en módulos separados.

| Song  |                                |
|---|--------------------------------|
| - code:String<br>- name:String<br>- artist:String   | - album: String<br>- year: int |
| + Song(String code, String name, String artist, String album, int year)<br>+ getCode():String<br>+ getName():String<br>+ getArtist():String<br>+ getAlbum():String<br>+ getYear():int<br>+ setCode(String c)<br>+ setName(String n)<br>+ setArtist(String a)<br>+ setAlbum(String a)<br>+ setYear(int y)<br>+ toString():String |                                |

| User   |
|--|
| - username:String<br>- userSongList:DoubleList   |
| + User(username)<br>+ getUsername():String<br>+ setUsername(String u)<br>+ addSong(Song s)<br>+ findSong(String name) : DoubleNode<br>+ removeSong(String name)<br>+ playList()<br>+ toString():String |

| SongCollection  |
|---|
| - songs:DoubleList  |
| + SongCollection()<br>+ add(Song s)<br>+ find(String name):DoubleNode<br>+ remove(String name, UserCollection userList)<br>+ play() |

| UserCollection   |
|--|
| - users:DoubleList   |
| + UserCollection()<br>+ getUsers():DoubleList<br>+ add(User u)<br>+ find(String username): DoubleNode<br>+ remove(String username): User<br>+ addUserSong(String username, String songName, SongCollection songList)<br>+ playUserList(String username)<br>+ print() |

El código del main se encuentra en la parte inferior, **NO** lo modifique. Adicionalmente, se tiene la salida que debe arrojar por consola.

#### **Punto 1. Clase UserCollection (1.8)**

- `add(User u)`: método que recibe un objeto de la clase `User` y lo almacena en la lista doble `users`
- `find()`: método que busca un usuario en `users` por su `username` y retorna el nodo que lo contiene
- `remove()`: método que elimina un usuario de `users` por su `username`
- `print()`: método que imprime todos los usuarios almacenados en `users`
- `addUserSong(String username, String songName, SongCollection songList)`: Método que recibe un `username` y el título de una canción. Si el usuario existe y si la canción existe se agrega la canción a la `userSongList` del usuario
- `playUserList(username)`: Método que recibe un `username` y lo busca. Si lo encuentra, reproduce la `userSongList` del usuario.

#### **Punto 2. Clase SongCollection (1.8)**

- `add(Song s)`: método que recibe un objeto de la clase `Song` y lo almacena en la lista doble `songs`
- `find()`: método que busca una canción en `songs` por su nombre y retorna el nodo que lo contiene
- `remove(String name, UserCollection userList)`: Método que intenta eliminar una canción de la lista `songs`. Si la elimina, debe ir a la `userSongList` de cada usuario y eliminarla también de esta lista.
- `play()`: método que reproduce todas las canciones almacenadas en `songs` (Imprime mensajes con la información de las canciones que está reproduciendo).

#### **Punto 3. Clase User (1.4)**

- `addSong(Song s)`: Método que recibe un objeto de tipo `Song` y lo agrega a la lista doble `userSongList`
- `findSong(String name)`: Método que busca una canción, por nombre, en la `userSongList` y retorna el nodo que la contiene.
- `removeSong(String name)`: Método que elimina una canción de la `userSongList`
- `playList()`: Método que reproduce todas las canciones almacenadas en `userSongList`.

|         |
|---------|
| Main.py |
|---------|

---

```
import Song as s import
User as u import
SongCollection as sc
import UserCollection as uc

try:
    # Apertura Archivos
    songs=open("Songs.csv","r")
    users=open("Users.csv","r")

    # Creación de Objetos SongCollection y UserCollection
    songList=sc.SongCollection()
    userList=uc.UserCollection()

    # Lectura de lineas archivos Songs.csv
```

```

songLine=songs.readline()  songLine=songs.readline()

while songLine!="":
    data=songLine.split(";")
    # Creando objetos de tipo Song
    song=s.Song(data[0], data[1], data[2], data[3], int(data[4][:1]))    songList.add(song)
songLine=songs.readline()

# Lectura de lineas archivos Users.csv  userLine=users.readline()  userLine=users.readline()

while userLine!="":
    # Creando objetos de tipo User    user=u.User(userLine[:-1])    userList.add(user)
    userLine=users.readline()

# Cierre de Archivos
songs.close()  users.close() except:
print("Error")

print("\n\nLista de Canciones\n")
songList.play()

print("\n\nLista de Usuarios\n")
userList.print()

print("\n\nEliminacion de Usuarios\n")
print(userList.remove("Bruise"))
print(userList.remove("Kenny"))

userList.addUserSong("Lynch", "La bamba", songList) userList.addUserSong("Lynch", "Jump", songList)
userList.addUserSong("Aspect", "Jump", songList) userList.addUserSong("Aspect", "Beat It", songList)
userList.addUserSong("Kraken", "Beat It", songList)
userList.addUserSong("Kraken", "Stairway To Heaven", songList) userList.addUserSong("Psycho", "Hotel
California", songList)

print("\n\nAgregando Canciones a Un Usuario Eliminado\n") userList.addUserSong("Bruise", "Hey Jude",
songList)

print("\n\nAgregando Canciones que no Existen\n") userList.addUserSong("Kraken", "One", songList)

print("\n\nReproduciendo Lista de Canciones de Usuario\n") userList.playUserList("Kraken")

```

```
print("\n\nEliminando Cancion\n") songList.remove("Beat  
It", userList)  
  
print("\n\nReproduciendo Lista de Canciones Luego de Eliminación\n") userList.playUserList("Kraken")
```

## Output

Lista de Canciones

Reproduciendo: (I Can't Get No) Satisfaction - The Rolling Stones  
Reproduciendo: Respect - Aretha Franklin  
Reproduciendo: Stairway To Heaven - Led Zeppelin  
Reproduciendo: Like a Rolling Stone - Bob Dylan  
Reproduciendo: Born to Run - Bruce Springsteen  
Reproduciendo: Hotel California - Eagles  
:  
Reproduciendo: Thunder Road - Bruce Springsteen  
Reproduciendo: No Woman, No Cry - Bob Marley & The Wailers  
Reproduciendo: La bamba - Ritchie Valens  
Reproduciendo: We've Only Just Begun - The Carpenters  
Reproduciendo: Papa Was a Rollin' Stone - The Temptations

Lista de Usuarios

Aspect  
Kraken  
Bender  
Lynch  
Big Papa  
Mad Dog  
Bowser  
Bruise  
Psycho  
Cannon

Eliminacion de Usuarios

Bruise  
None

Agregando Canciones a Un Usuario Eliminado

Error: El usuario no existe

Agregando Canciones que no Existen

Error: La canción no existe

Reproduciendo Lista de Canciones de Usuario

Kraken está reproduciendo: Beat It

Kraken está reproduciendo: Stairway To Heaven

Eliminando Cancion

Eliminada: 52-Beat It- Michael Jackson-Thriller-1983

Reproduciendo Lista de Canciones Luego de Eliminación

Kraken está reproduciendo: Stairway To Heaven