



Laboratorio #3

Objetivo

El objetivo de este laboratorio es que el alumno entienda y aplique todos los conceptos aprendidos de Kotlin. Este es un laboratorio que sólo se enfoca en el lenguaje de programación, y no en cómo aplicarlo en Android

Requisitos

Este laboratorio se enfocará, básicamente, en el uso de **Data Classes**, **Extension Functions** y **Constructores**. El programa consiste en una mini base de datos de música.

Typealiases

Necesito que creen, por lo menos, este typealias. Si ustedes desean crear más, bienvenidos.

- Nombre: **"Duration"**
- Tipo: **Long**

Extension functions

Necesito, por lo menos, que creen estas 2 extension functions:

1. **getStringDuration()**, utilizada en objetos de tipo **Duration**: se va a encargar de pasar una duración de canción (que siempre será dada en milisegundos) a un string de tipo "mm:ss". Ejemplo:
 - a. 540000L -> 9:00 (OJO, los segundos siempre se mostrarán en 2 dígitos)
 - b. 255000L -> 4:15
2. **getFirstTwoSongs()**, utilizada en objetos de tipo **List<Song>**: este deberá retornar un objeto de tipo **Pair**, con las 2 primeras canciones de la lista. Para definir cuáles son las primeras 2 canciones, se utilizará el atributo **trackNumber**. Entonces, si tenemos 3 canciones (trackNumber=4, trackNumber=1, trackNumber=10) debería retornar de primero la de trackNumber=1 y luego trackNumber=4

Clases a crear

Deben tener cuidado cómo crean estas clases, ya que es crucial usar los atributos correctos y crear los constructores necesarios para que, al correr el programa de evaluación, funcione.

- **Song**

- Atributos que vendrán en el constructor principal de la clase
 - **name** de tipo **String**, no nullable
 - **duration** de tipo **Duration**, no nullable
- Atributos que serán parte de la clase, pero no del constructor principal
 - **trackNumber** de tipo **Int**, este podrá ser null
- La clase deberá tener un constructor secundario en donde se puedan mandar los 3 atributos anteriores.
- Deberán sobre escribir la función **toString**, con el objetivo de que cada vez que impriman esta clase, pueda retornar algo así:
 - Cuando la clase no tiene un trackNumber:
 - "The Unforgiven - 2:15"
 - Cuando la clase sí tiene trackNumber
 - "1. Hit the lights - 2:33"
 - "20. La canción - 10:22"

- **Disc**

- Tendrá únicamente 2 atributos, y ambos vendrán en el constructor principal
 - **name** de tipo **String**, no nullable
 - **songs** de tipo **List<Song>**, no nullable
- Deberán sobre escribir la función **toString**, con el objetivo de que cada vez que impriman esta clase, pueda retornar algo así:
 - "Nombre del disco, 10 songs on it"
 - "The disc name, 1 songs on it"

- **Artist**

- Atributos que vendrán en el constructor principal de la clase
 - **name** de tipo **String**, no nullable
 - **country** de tipo **String**, no nullable
- Atributos que serán parte de la clase, pero no del constructor principal
 - **discography** de tipo **List<Disc>**, no nullable
 - **type** de tipo **String**, no nullable
 - **singles** de tipo **List<Song>**, no nullable y por defecto, una lista vacía
- La clase deberá tener un constructor secundario en donde se puedan mandar los atributos **name**, **country**, **discography** y **type**.
- Deberán sobre escribir la función **toString**, con el objetivo de que cada vez que impriman esta clase, pueda retornar algo así:
 - Si **singles** está vacío
 - "name is a type from country. Type has discography.size discs and no singles"
 - Si **singles** tiene al menos un elemento

- “**name** is a **type** from **country**. **Type** has **discography.size** discs and **singles.size** singles”

Criterio de evaluación y entregables

La forma en la que se realizará la calificación es la siguiente:

- Deberá entregar el link de kotlin playground.
- El código deberá tener las 3 clases, las 2 extension functions y como mínimo un el typealias descrito.
- La calificación se basará en el resultado de 7 unit tests que ya están desarrollados y que pueda correr el programa inicializando las clases con los datos que yo quiera.
 - Test del 1 al 7: 10 puntos cada uno
 - Correr el programa con datos propios: 30 puntos

Recursos adicionales

- Playground a utilizar con unos ejemplos: <https://pl.kotl.in/utXH1InA6>