

Parcial 1 - Algoritmos I Taller: Tema D

Debés entregar el código completo en los campos correspondientes de cada ejercicio del formulario en el que completaste tus datos personales. Este código debe poder ejecutarse en haskell sin errores. Te recomendamos para ello que pruebes con diferentes ejemplos antes de entregar.

En algunas apps de reproducción de música podemos escuchar lanzamientos musicales de los artistas. Cada lanzamiento puede ser un álbum o un sencillo (o single). También dichas apps disponen de funcionalidad para crear colas de reproducción.

Ejercicio 1:

a)

Definir el tipo Lanzamiento que consta de dos constructores Album y Sencillo con los siguientes parámetros:

- El constructor Album debe tomar como parámetros el nombre, el artista, la lista de nombres de temas y el año de estreno.
- El constructor Sencillo debe tomar como parámetros el nombre, el artista, la duración (en segundos) y el año de estreno.

b)

A partir del tipo definido en el punto anterior, definí los siguientes términos (lanzamientos)

```
thriller :: Lanzamiento  
thriller = <COMPLETAR>
```

correspondiente al álbum "Thriller" del artista "Michael Jackson", con lista de temas ["Wanna be starting something", "Baby me mine", "The girl is mine", "Thriller", "Beat it", "Billie Jean", "Human nature", "Pretty young thing", "The lady in my life"] y el año de estreno 1982.

```
yesterday :: Lanzamiento  
yesterday = <COMPLETAR>
```

correspondiente al sencillo "Yesterday" de "The Beatles" con duración 125 segundos y año de estreno 1965.

c)

Definir la función `esDiscoLargo :: Lanzamiento -> Bool` que dado un `Lanzamiento`, devuelve `True` si es un álbum con cantidad de tracks mayor o igual a 5, `False` en caso contrario.

d)

Definir la función `esSencilloActual :: Lanzamiento -> Bool` que dado un `Lanzamiento`, devuelve `True` si éste es un sencillo cuyo año de estreno es mayor o igual a 2020.

e)

Definir la función `tracksAlbumMasLargo :: [Lanzamiento] -> Int` que dada una lista de lanzamientos, devuelve la cantidad de tracks que tiene el álbum más largo que está en la cola de reproducción.

Ejercicio 2:

Dado el tipo recursivo `ColaLanzamiento` definido de la siguiente manera

```
data ColaLanzamiento = Vacia | Encolada Lanzamiento ColaLanzamiento
                      deriving Show
```

podemos definir, por ejemplo, una cola de reproducción de la siguiente manera

```
colaReproduccion :: ColaLanzamiento
```

```
colaReproduccion = Encolada thriller (Encolada yesterday Vacia))
```

Definir la función `albumsDelArtista :: ColaLanzamiento -> String -> ColaLanzamiento` que dada una cola de reproducción `q`, y el nombre de un artista `a`, devuelve la cola de reproducción que tiene solamente los álbumes del artista `a` (en el mismo orden que aparecen en `q`).