

Algo de música

Parcial de Paradigmas de Programación - Paradigma Funcional - Jueves Noche - 2023

Nos piden desarrollar un sistema para administrar las escuchas de canciones que se realizan semanalmente de las playlists más exitosas del momento.

Para eso necesitamos modelar el perfil de cada oyente que participa, así como también las características de las diversas categorías y el modo en que estas afectan los sentimientos de las personas.

De cada **Persona** nos interesa registrar de que artista es fanático, el tiempo libre que tiene y su estado de ánimo.

También sabemos que las **playlist** se componen de **canciones** de distinta duración, cada una de las cuales pertenece a distintas **categorías**. La categoría de la canción influencia en gran medida la forma en que cada persona puede escucharla.

Para representar estos datos, elegimos utilizar las siguientes estructuras:

```
data Persona = Persona {  
  tiempoLibre :: Int,  
  estadoDeAnimo :: Int,  
  fanDe :: [Artista],  
} deriving (Eq, Show)
```

```
type Artista = String
```

```
type Playlist = [Cancion]
```

```
data Cancion = Cancion {  
  duracion :: Int,  
  categoria :: Categoria  
}
```



Teniendo en cuenta lo descrito anteriormente se pide modelar los siguientes puntos **explicitando el tipo de cada función y utilizando los conceptos aprendidos del Paradigma Funcional**, poniendo especial énfasis en el uso de **Composición, Aplicación Parcial y Orden Superior**.

1. Modelar una función para que una persona **disminuya en minutos su tiempo libre**, y otra para que, dada una función $Int \rightarrow Int$ que representa un cambio en el estado de ánimo, **actualice el estado de la persona con el resultado de aplicar el cambio al estado de ánimo actual**.
2. Definir el tipo **Categoría**, de forma tal que resulte adecuado para representar los diversos estados de ánimo que las personas atraviesan al escuchar canciones...

a. Modelar las siguientes categorías:

musicaClásica: Esta categoría es ideal para escuchar ya que mantiene serenas a las personas, y no las impacta al ánimo de ninguna forma.

pop: Los ritmos enérgicos y pegadizos de este género obligan a las personas a subir su estado de ánimo un 25%.

desamor: Puede afectar a las personas de diversas formas. Si la intensidad de la categoría es menor de 50 puntos, las personas son capaces de escucharla perdiendo dos puntos de estado de ánimo por cada punto de intensidad. Por otro lado, si la intensidad es mayor, los oyentes no se animan a escucharla y, en cambio, pierden 5 minutos de tiempo libre buscando otra cosa para escuchar. Cabe destacar que estas penalidades no afectan a las personas felices, que pueden escuchar cualquier tipo de canción sin sufrir. Una persona feliz es aquella que tiene más de 100 puntos de estado de ánimo.

taylorSwift: Mezclan todos los aspectos de las categorías de desamor y las pop. Lo impacta igual que la categoría pop y la de desamor con 20 puntos de intensidad (en ese orden). Además de eso, gana un punto extra de estado de ánimo, sólo porque esta categoría es muy conmovedora.

- b. Modelar una función **escucharCancion** que, dado una canción y una Persona, haga que la persona disminuya su tiempo libre escuchando la canción, luego de ser afectada por la categoría de la misma.
- c. Dar un ejemplo de playlist que contenga canciones con todas las categorías descritas anteriormente **y una más, inventada por vos**, con un comentario que describa qué cambios significa.

3. Definir funciones que, dada una lista de Personas:

- a. Indique cuál es el mayor tiempo libre de las personas que todavía pueden escuchar canciones (es decir, aquellas cuyo estado de ánimo es > 0).
 - b. Dada una lista de artistas, indique si todas las personas que tienen todavía más de treinta minutos libres son fanáticos de algún artista de la lista.
 - c. Dada una canción y una persona ociosa, indique cuántas de las personas de la lista pueden escuchar dicha canción y terminar con más tiempo libre que dicha persona
4. Modelar la función `escucharPlaylist :: [Persona] -> Playlist -> [Persona]`, que reciba una lista de personas y una playlist y retorne el podio resultante de la escucha, es decir, los tres oyentes que escucharon todas las canciones de la playlist (en orden) y tienen el mayor estado de ánimo.
- Nota:** Se puede utilizar la función `sortOn :: Ord a => (b -> a) -> [b] -> [b]` que dado una función cuantificadora y una lista, retorna la lista ordenada de menor a mayor resultado de aplicar la función cuantificadora a cada elemento.