Parcial 1 - Algoritmos I Taller: Tema A

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.

Descripción del problema

En las páginas de visualización de streaming, normalmente encontramos películas y series. Los videos que se reproducen pueden corresponder a películas o capítulos de series. En algunas plataformas también podríamos tener cola de reproducción de videos, de manera que al terminar de ver uno, comienza el otro.

Ejercicio 1:

a)

Definir el tipo Video que consta de dos constructores `Pelicula` y `CapSerie` con los siguientes parámetros:

- El constructor `Pelicula` debe tomar como parámetros el nombre, el director, la duración (en minutos) y el año de estreno.
- El constructor `CapSerie` debe tomar como parámetros el nombre de la serie, el nro de capítulo, la temporada, el año de estreno de la temporada.

b)

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

```
elPadrino :: Video
elPadrino = <COMPLETAR>
```

correspondiente a la película “El Padrino” con director “Francis Ford Coppola”, duración 177 minutos y año de estreno 1972.

```
breakingBadS01E01 :: Video
breakingBadS01E01 = <COMPLETAR>
```

correspondiente al capítulo de serie “Breaking Bad”, con capítulo número 1, temporada 1 y año de estreno de temporada 2008.

c)

Definir la función `esPrimerCapitulo :: Video -> Bool` que dado un Video, devuelve `True` si el video es el primer capítulo de la primera temporada de una serie, False caso contrario.

d)

Definir la función `esEstreno :: Video -> Bool` que dado un `Video`, devuelve `True` si el video es una película cuyo año de estreno es igual a 2024.

e)

Definir la función `duracionPeliMasLarga :: [Video] -> Int` que dada una lista de de videos, devuelve la duración de la película más larga. En caso que no haya películas devuelve 0.

Ejercicio 2:

Dado el tipo recursivo ColaVideo definido de la siguiente manera

```
data ColaVideo = Vacia | Encolada Video ColaVideo deriving Show
```

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

```
colaReproduccion :: ColaVideo

colaReproduccion = Encolada elPadrino (Encolada breakingBadS01E01
Vacia))
```

Definir la función `pelisDelDirector :: ColaVideo -> String -> ColaVideo` que dada una cola de reproducción de videos q, y el nombre de un director d, devuelve la cola de reproducción que tiene solamente las películas del director d (en el mismo orden que aparecen en q).