

Parcial 1 - Algoritmos I Taller: Tema B

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):

```
odisea2001 :: Video
odisea2001 = <COMPLETAR>
```

correspondiente a la película "2001: Odisea del espacio" con director "Stanley Kubrick", duración 142 minutos y año de estreno 1968.

```
theOfficeS02E05 :: Video
theOfficeS02E05 = <COMPLETAR>
```

correspondiente al capítulo de serie "The Office", con capítulo número 5, temporada 2 y año de estreno de temporada 2005.

c)

Definir la función `esDelDirector :: Video -> String -> Bool` que dado un `Video` y el nombre de un director, devuelve `True` si el video es una película del director dado.

d)

Definir la función `esActual :: Video -> Bool` que dado un `Video`, devuelve `True` si el video es un capítulo de serie cuyo año de estreno de temporada es mayor o igual a 2020 y `Falso` en caso contrario.

e)

Definir la función `tiempoPelis :: [Video] -> Int -> Int` que dada una lista de videos y un año dado, devuelve la suma de la duración de todas las películas de ese año que están en la cola.

Ejercicio 2:

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

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

podemos definir `colaReproducción` de la siguiente manera

```
colaReproduccion :: ColaVideo
```

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

Definir la función `seriesActuales :: ColaVideo -> ColaVideo` que dada una cola de reproducción de videos `q`, devuelve la cola de reproducción que tiene solamente los capítulos de serie actuales (en el mismo orden que aparecen en `q`).