

Recuperatorio: Tema D - Haskell

Ejercicio 1

a) Definir el tipo `ArtistaMusical` que consta de dos constructores `Cantante` e `Instrumentista`, que toman los siguiente parámetros:

- Constructor `Cantante`: Tiene tres parámetros, el primero de tipo `Registro`, el segundo de tipo `TipoCanto` y el tercero de tipo `Trayecto`
- Constructor `Instrumentista`: Tiene dos parámetros, el primero de tipo `Instrumento` y el segundo de tipo `Trayecto`.

Además:

- El tipo `Registro` tiene constructores `Soprano`, `Contralto`, `Tenor`, `Baritono` y `Bajo`, todos sin parámetros. El tipo **no debe** estar en la clase `Eq`.
- El tipo `TipoCanto` tiene constructores `Solista` y `Coral`, ambos sin parámetros. El tipo **no debe** estar en la clase `Eq`.
- El tipo `Instrumento` tiene constructores `Violin`, `Clarinete`, `Trompeta` y `Timbales`, todos sin parámetros.
- El tipo `Trayecto` debe ser un sinónimo de `Int` que indica la cantidad de años de trayectoria..

b) Programar la función

```
trayectoria_musical :: ArtistaMusical -> Trayecto
```

que devuelve la cantidad de años de trayectoria de un/una artista musical.

c) Incluir el tipo `ArtistaMusical` en la clase `Ord` de manera tal que un/una artista se considere menor o igual según su trayectoria.

Ejercicio 2

a) Programar de manera recursiva la función

```
buscarSolistas :: [ArtistaMusical] -> Registro -> [ArtistaMusical]
```

que dada una lista de artistas `as` y un registro `r` devuelve una lista con los elementos de `as`, que son cantantes solistas con registro `r`.

b) Escribir una lista de artistas con **al menos tres elementos** que incluyan al menos un cantante y al menos un instrumentista.

c) Escribir el resultado de `buscarSolistas` para la lista del punto **b)**

Ejercicio 3

Basados en el tipo `ListaAsoc` del *Proyecto 2*, programar recursivamente la función:

```
la_algun_multiplo :: ListaAsoc a b -> b -> Bool
```

que dada una lista de asociaciones `la` y un dato `x` devuelve `True` si alguno de los datos de `la` es múltiplo de `x`, en caso contrario devuelve `False`. Completar el tipado de la función para incluir los *type classes* necesarios para programarla.