

All the Night Club

Se cuenta con personas que asisten a una discoteca, entre los que contamos con personas tranquilas y agitadores modeladas de la siguiente manera:

```
data Persona = Tranqui String
              | Agitador Int Int String
              deriving (Show, Eq)
```

```
fer    = Tranqui "fer"
flor   = Tranqui "flor"
rodri  = Agitador 1670 2000 "rodri"
deby   = Agitador 7000 1500 "deby"
```

Las personas tranquilas definen

- Cómo se llama la persona

Las que agitan definen

- La cantidad de levante que tiene
- El nivel de aguante de alcohol
- Cómo se llama la persona

Por otra parte se tiene la lista de personas en una disco:

```
type Disco = [ (Persona, [ (String, Int) ]) ]
```

```
laDisco :: Disco
laDisco = [
    (fer, [("Coca cola", 1), ("Sprite Zero", 1)]),
    (rodri, [("Cerveza", 2)]),
    (deby, [("Grog XD", 25), ("Cerveza", 1)]),
    (flor, [("Grapa", 1)])
]
```

Modelados como una tupla con:

- Persona
- Bebidas que tomó, modeladas como una lista de tuplas con:
 - Nombre de la bebida
 - Cantidad de vasos

Y por último, información sobre los tragos:

```
tragos = [("Coca cola", 0), ("Grog XD", 350),
          ("Sprite Zero", 0), ("Cerveza", 10), ("Grapa", 40)]
```

El formato que sigue la tupla es:

- Nombre de la bebida
- Graduación alcohólica de cada vaso s

Condiciones de aprobación

- Como en cualquier parcial de Paradigmas, no se debe repetir código, se deben usar las abstracciones adecuadas, y se deben favorecer la expresividad y la declaratividad del código.
- De Funcional, se deben respetar las ideas del paradigma, y además usar donde sea más conveniente composición, aplicación parcial, y funciones de orden superior.
- Se debe declarar el tipo de las funciones principales, usando los alias definidos.

Requerimientos

1. Control de escabio

- a. Saber las **bebidas** que tomó una persona en *laDisco*.
- b. Obtener la **graduación alcohólica** de un trago a partir de su nombre.
- c. Calcular la cantidad de **alcohol en sangre** que tiene una persona.

2. Lo borrachos

- a. Saber si una persona **está borracha**:
 - para una persona tranquila, no puede tener nada de alcohol en sangre.
 - para un agitador, si el aguante es menor que el alcohol en sangre.
- b. Dada una lista de personas, obtener las personas **impresentables**, que son aquellas que están borrachas o pidieron más de 5 tragos.
Por cada una retornar una tupla con el nombre de la persona y su alcohol en sangre.

3. A festejar!

- a. Armar las siguientes funciones que modifican una Disco
 - **entrar**: agrega una persona a la lista sin haber pedido ningún trago.
 - **pedir bebida**: cuando una persona pide una bebida hay que incorporarla a la lista de tragos que bebió (asumir 1 vaso, no importa que se repita la misma bebida en la lista).
 - **descontrolarse**: si la persona es un agitador, se pide tantos “Grog XD” como la mitad de su aguante. Las personas tranquilas no se puede descontrolar.
- b. Las discotecas tienen planes predefinidos para eventos especiales. Un plan está declarado como
`type Plan = [Disco -> Disco]`
Hacer la función **planDeCumple** que dada una persona retorna un plan para su cumpleaños. Dicho plan incluye, en orden: *entrar*, *pedir un “Fernet”*, *pedir un “Campari”*, *descontrolarse* y *pedir un “Agua”*.
- c. Hacer la función **festejarCumple** que, dada una persona, ejecute su plan de cumpleaños en *laDisco*. El ejecutar un plan se ejecutan todas las acciones definidas en *laDisco* con un máximo de 15 acciones¹.
 - ¿Qué pasa si *fer* quisiera festejar su cumple?
- d. Hacer la función **salidaSana** que dada una persona devuelva un plan en donde primero entre y luego pida toda la cantidad de “Agua” que se pueda.

¹ Tener en cuenta que si se evalúa la función **take** con una cantidad mayor al número de elementos de la lista, retorna la lista completa.

- ¿Qué pasa si quiere saber el largo de dicha lista? ¿Por qué?
- ¿Se puede ejecutar dicha salida? ¿Por qué?