

Expertos en maquinitas

Queremos desarrollar un programa en Haskell para modelar los juegos de un casino y la posibilidad de los jugadores de ganar o salir con los bolsillos vacíos.

La información que disponemos de las personas es su nombre, la cantidad de dinero que tienen, su suerte y una lista de factores que pueden ayudarles a ganar distintos juegos, como ser su inteligencia, algún amuleto, etc. De cada factor tenemos un valor numérico que nos indica qué tan valioso es.



```
data Persona = UnaPersona String Float Int [(String,Int)] deriving(Show)
```

Tenemos un par de personas de ejemplo para poder probar nuestro programa:

```
nico = (UnaPersona "Nico" 100.0 30 [("amuleto", 3), ("manos magicas",100)])
maiu = (UnaPersona "Maiu" 100.0 42 [("inteligencia",55), ("paciencia",50)])
```

1. Conocer la suerte total de una persona. Si no tiene un amuleto, es su suerte normal, si tiene uno, su suerte se multiplica por el valor de ese amuleto.

En general, sólo se considera que una persona tiene un factor si el valor del mismo es mayor a cero.

Tener un amuleto de valor 0 es lo mismo que no tenerlo en absoluto.

```
> suerteTotal nico
90
> suerteTotal maiu
42
```

2. Desarrollar el data para el tipo Juego y las funciones ruleta y maquina sabiendo que un juego se compone por un **nombre**, una función que determina **cuánto dinero se ganaría a partir de un monto apostado** y una **serie de criterios** determinantes para ganar. Para modelar el dinero usar el tipo Float.

a- La **ruleta** que se gana 37 veces lo apostado. Para ganar la persona debe tener una suerte total mayor a 80.

b- La **maquina** que se basa en un jackpot y lo que se gana es la apuesta más el jackpot. Para ganar se deben cumplir dos condiciones: que la persona tenga una suerte total mayor a 95 y además que tenga paciencia.

3. Saber si un jugador puede ganar un juego, lo cual sucede si cumple todas las condiciones para ganar ese juego.

4. Dado un jugador, una apuesta inicial y una lista de juegos, obtener la cantidad total de dinero que puede conseguir esa persona con ese monto si apuesta en cada juego lo conseguido en el juego anterior, evitando los juegos en los cuales no pueda ganar.

Si no puede ganar en ningún juego, el resultado sería la apuesta inicial, ya que no apostaría en ninguno. Resolver de dos formas:

a- con funciones de **orden superior, aplicación parcial y composición**

b- con **recursividad**

Mostrar un ejemplo de uso de esta función con las personas maiu y nico y una lista de juegos que incluya una ruleta y una maquinita.

5. Dada una lista de jugadores y una lista de juegos, retornar los nombres de los jugadores que no pueden ganar a ningún juego.

6. Hacer que un jugador apueste una cantidad en un juego, que implica que la persona baje su saldo esa cantidad y luego juegue al juego. Si puede ganar en ese juego retorna a la persona con su saldo incrementado en lo que gana en el juego, de lo contrario retorna a la persona sin ganar nada.

7. Inferir el tipo de la siguiente función:

```
elCocoEstaEnLaCasa x y z = all ((>z).(+42)).foldl (\a (b,c) -> y c ++ b a) (snd x)
```


