

Carreras de Chocobos

Parcial de Paradigmas de Programación - Paradigma Funcional - Jueves Noche - 2020

Nos piden desarrollar un sistema para administrar las carreras de chocobos que se realizan semanalmente en las pistas localizadas en las inmediaciones del complejo *Gold Saucer*, con el fin de simular y predecir los resultados de las carreras y asistir a los apostadores.

Para eso necesitamos modelar el perfil de cada ave que participa, así como también las características de las diversas pistas en las que corren y el modo en que estas afectan los resultados.

De cada **Chocobo** nos interesa registrar su color, el tiempo que lleva corriendo desde que empieza una carrera y la velocidad a la que se desplaza.

También sabemos que las **Pista** en las que se realizan las carreras se componen de **Tramos** de distinta longitud, cada uno de los cuales atraviesa un tipo diferente de **Terreno**. El terreno del tramo influencia en gran medida la forma en que cada ave puede recorrerlo: No es lo mismo correr en una pradera que esquivando ramas en la selva! Cada terreno afecta la velocidad a la que un chocobo puede atravesar el tramo y el tiempo que este le requiere.

Para representar estos datos, elegimos utilizar las siguientes estructuras:

```
data Chocobo = Chocobo {  
    tiempo :: Int,  
    velocidad :: Int,  
    color :: String  
} deriving (Eq, Show)  
  
type Pista = [Tramo]  
  
data Tramo = Tramo {  
    longitud :: Int,  
    terreno :: Terreno  
}
```



Teniendo en cuenta lo descrito anteriormente se pide modelar los siguientes puntos **explicitando el tipo de cada función y utilizando los conceptos aprendidos del Paradigma Funcional**, poniendo especial énfasis en el uso de **Composición, Aplicación Parcial y Orden Superior**.

1. Modelar una función para **bajar en cierta cantidad la velocidad** a la que un Chocobo corre, y otra para **hacer que corra una distancia dada**, considerando que recorrer una distancia, le lleva al Chocobo un tiempo equivalente a su velocidad dividida dicha distancia.
2. Definir el tipo **Terreno**, de forma tal que resulte adecuado para representar los diversos terrenos que los tramos de las pistas atraviesan y...

- a. Modelar los siguientes terrenos:

asfalto: Este terreno es ideal para correr ya que no impacta de ninguna forma a los Chocobos.

bosque: Los múltiples obstáculos y peligros de este terreno obligan a los Chocobos a reducir su velocidad una cuarta parte.

agua: Los tramos sumergidos en agua pueden impactar a las aves de diversas maneras. Si la profundidad del agua es menos de 50 cm, los pájaros son capaces de atravesarla perdiendo dos puntos de velocidad por cada cm de profundidad. Por otro lado, si el agua es más profunda que eso, los Chocobos no se animan a cruzarla y, en cambio, pierden 5 minutos bordeando por el costado. Cabe destacar que estas penalidades no afectan a los Chocobos Azules, que son naturalmente buenos nadadores y cruzan los tramos de agua como si nada.

pantano: Los pantanos son lugares horribles que mezclan todos los peores aspectos de los bosques y las zonas inundadas. Cuando un Chocobo cruza un pantano lo impacta igual que cruzar un bosque y 20 cm de agua (en ese orden). Además de eso, pierde un punto extra de velocidad, sólo por el cagazo que le da el lugar.

- b. Modelar una función **correrTramo** que, dado un Tramo y un Chocobo haga que el ave corra la distancia del tramo luego de ser afectada por el terreno.
 - c. Dar un ejemplo de pista que contenga tramos con todos los terrenos descritos anteriormente **y uno más, inventado por vos**, con un comentario que describa qué cambio hace.
3. Definir funciones que, dada una lista de Chocobos:

- a. Indique cuál es el mayor tiempo de los Chocobos que todavía están corriendo (es decir, aquellos cuya velocidad es > 0).
- b. Dada una lista de colores, indique si todos los Chocobos que llevan más de treinta minutos corriendo son de un color de la lista.
- c. Dado un tramo y un Chocobo campeón, indique cuántos de los Chocobos de la lista pueden correr dicho tramo más rápido que el campeón.

4. Utilizando la siguiente función **ordenar**:

```
ordenar _ [] = []  
ordenar f (x:xs) = (ordenar f . filter ((f x >=).f)) xs ++  
                  [x] ++  
                  (ordenar f . filter ((f x <).f)) xs
```

Modelar la función **correrCarrera** :: **[Chocobo] -> Pista -> [Chocobo]**, que reciba una lista de participantes y una pista y retorne el podio resultante de la carrera, es decir, los tres Chocobos que recorrieron todos los tramos de la pista (en orden) en el menor tiempo, ordenados de menor a mayor tiempo.