

Eliminatorias

Se está organizando un torneo en el cual muchos participantes compiten en diversos desafíos, volviéndose más experimentados luego de cada desafío ganado.

De cada participante nos interesará su nombre, la cantidad de experiencia que tiene, su nivel de inteligencia y destreza física, y el rol que desempeña. Un participante podría ser más apto para un rol que para otro, con lo cual será un factor importante elegir el rol más adecuado para encarar cada desafío.



Ocasionalmente podrían tener que usar armas, para las cuales nos interesa su valor de combate y la experiencia mínima que debe tener el participante para usarla apropiadamente. La potencia que tendrá el arma para un participante dependerá de si dicho participante alcanza la experiencia mínima indicada: en caso de tener experiencia suficiente, la potencia será el valor de combate del arma, si no será la mitad.

Algunos roles que necesitamos representar inicialmente son los siguientes:

- Indeterminado: la aptitud de un participante para este rol equivale a la suma entre su inteligencia y su destreza física.
- Soporte: la aptitud de un participante para este rol equivale a su inteligencia multiplicada por 7 sumada a su experiencia.
- Primera línea: para este rol se indicará también el arma que deberá manejar el participante; la aptitud de un participante para este rol equivale a la suma de su destreza física y la potencia del arma, todo eso multiplicado por la experiencia del participante dividida por 100.

Resolver los siguientes requerimientos de modo que se aproveche todo lo posible del uso de **composición**, **aplicación parcial** y **orden superior**. **Explicitar los tipos** de todas las funciones desarrolladas. **Evitar repetir lógica y soluciones poco declarativas**.

1. A partir de la información del dominio previamente indicada:

- Desarrollar todo lo necesario para representar a los participantes, armas y roles explicados, considerando que sea posible incorporar nuevos roles fácilmente.
- Declarar una constante de tipo Participante que tenga 1000 puntos de experiencia, inteligencia igual a 20, destreza física igual a 12 y el rol indeterminado.
- Desarrollar una función para conocer el poder de un participante, que se calcula como su experiencia multiplicada por la aptitud que tiene ese participante para el rol que desempeña.
*Ejemplo: para el participante del punto anterior, su poder debería ser $1000 * (20 + 12)$.*

Con este modelo en mente, y sabiendo que a su vez disponemos de las siguientes funciones para usar:

```
maximoSegun :: Ord a => (b -> a) -> [b] -> b
maximoSegun f = foldl1 (mayorSegun f)

mayorSegun :: Ord a => (p -> a) -> p -> p -> p
mayorSegun f a b
  | f a > f b = a
  | otherwise = b
```

- a. Desarrollar una función para que un participante elija un nuevo rol a partir de un conjunto de roles disponibles, cambiando su rol actual por el elegido. El rol a elegir para el participante debería ser aquel que maximice su aptitud para dicho rol.
- b. Mostrar una consulta que permita conocer el poder del participante del punto 1b luego de elegir un rol, incluyendo entre los roles disponibles el indeterminado, el de soporte y el de primera línea con un arma de valor de combate 20 y experiencia mínima de 750.
- c. Incorporar un nuevo rol: maestro de armas. Se debe poder indicar una lista de armas disponibles para que el participante use al elegir dicho rol, y la aptitud del mismo debe calcularse como la suma de potencia de las primeras 3 armas que pueda usar apropiadamente (por alcanzar la experiencia mínima necesaria).
- d. ¿Sería posible determinar la aptitud de un participante para el rol de maestro de armas si la lista de armas que se pueden usar fuera infinita? Justificar.

Si no puedes usar ninguna => Que de 0

3. A continuación queremos prepararnos para resolver los desafíos que incorporaremos más adelante. Para ello se pide desarrollar funciones para...

- a. Saber si un participante se encuentra entre un grupo de participantes. Debería cumplirse si entre esos participantes hay alguno que tenga el mismo nombre que el que nos interesa.
- b. Calcular la **recompensa** de vencer a otros participantes. Para ello se proveerá una lista con todos los participantes de un desafío y una lista con aquellos que consideramos ganadores. La recompensa equivaldrá a la **experiencia** del participante que tenga el nivel más alto entre su inteligencia y destreza física, de entre aquellos participantes **que no se encuentren entre los ganadores**.

Ejemplo: si el participante del punto 1b es uno de los participantes que no están entre los ganadores, y los demás que no ganaron no superan los 20 puntos de inteligencia o destreza física (que es el nivel más alto de ese participante de referencia), la recompensa será 1000.

4. Necesitamos desarrollar lo necesario para que un conjunto de participantes encaren un desafío. Para ello incorporaremos a los desafíos a nuestro modelo, de esta forma:

```
data Desafio = Desafio {
  rolesDisponibles :: [Rol],
  pruebaASuperar :: Participante -> Bool }
```

Resolver la lógica para que un conjunto de participantes encaren un desafío, donde lo esperado es que primero **elijan el rol para el cual sean más aptos** a partir de aquellos que están disponibles para el desafío, luego **traten de superar la prueba** de dicho desafío (quedando sólo aquellos que lo logran) y finalmente se **incremente la experiencia de todos los ganadores** (quienes pasaron la prueba) en el valor de la **recompensa** por haber vencido a los demás.

Nota: se espera que el rol que tengan los ganadores luego del desafío sea el elegido para el mismo, no el que tenían antes de encarar el desafío.

5. Finalmente necesitamos hacer que un conjunto de participantes compitan en un torneo. Sabemos que un torneo se compone por muchos desafíos, a partir de los cuales esos participantes irán quedando eliminados progresivamente. Necesitamos conocer cómo quedarán aquellos participantes que logren terminar el torneo, luego de encarar los desafíos uno tras otro.