

# Tierra de Bárbaros

Rechapos! - En la tierra de Udrogoth los reyes bárbaros formaron un imperio de guerreros, dragones y magia. Ahora marcharon a la guerra contra el mal y dejaron a sus herederos a cargo.... y bueno, hay que conformarse.

Sin embargo, nos encargaron confeccionar un programa en Haskell para manejar los asuntos del reino, y evitar así el tener que gobernar.

Consideraciones:

- Escribir el tipo de **todas** las funciones principales
- Emplear sinónimos de tipo cuando sea posible.
- No se permite usar recursividad salvo que se indique lo contrario
- Definir las funciones en estilo point-free cuando sea posible



## Punto 1

Se sabe que los bárbaros tienen nombre, fuerza, habilidades y objetos, que los ayudarán más adelante en su lucha contra el mal. Por ejemplo:

```
dave = Barbaro "Dave" 100 ["tejer","escribirPoesia"] [ardilla, varitasDefectuosas]
```

Se pide definir los siguientes objetos y definir algunos bárbaros de ejemplo

1. Las espadas aumentan la fuerza de los bárbaros en 2 unidades por cada kilogramo de peso.
2. Los amuletosMisticos puerco-marranos otorgan una habilidad dada a un bárbaro.
3. Las varitasDefectuosas, añaden la habilidad de hacer magia, pero desaparecen todos los demás objetos del bárbaro.
4. Una ardilla, que no hace nada.
5. Una cuerda, que combina dos objetos distintos, obteniendo uno que realiza las transformaciones de los otros dos.

## Punto 2

El megafono es un objeto que potencia al bárbaro, concatenando sus habilidades y poniéndolas en mayúsculas<sup>1</sup>.



```
*Main> megafono dave  
Barbaro "Dave" 100 ["TEJERESCRIBIRPOESIA"] [<function>,<function>]
```

Sabiendo esto, definir al `megafono`, y al objeto `megafonoBarbarico`, que está formado por una cuerda, una ardilla y un megáfono.

<sup>1</sup> Recuerden que esta la función `toUpper` que toma una Char y devuelve un Char pero en mayúscula

Main> toUpper 'a'

'A'

### Punto 3 - Aventuras

Los bárbaros suelen ir de aventuras por el reino luchando contra las fuerzas del mal, pero ahora que tienen nuestra ayuda, quieren que se les diga si un grupo de bárbaros puede sobrevivir a cierta aventura. Una aventura se compone de uno o más eventos, por ejemplo:

1. invasionDeSuciosDuendes: Un bárbaro sobrevive si sabe “Escribir Poesía Atroz”
2. cremalleraDelTiempo: Un bárbaro sobrevive si no tiene pulgares. Los bárbaros llamados Faffy y Astro no tienen pulgares, los demás sí.
3. ritualDeFechorias: Un bárbaro puede sobrevivir si pasa una o más pruebas como las siguientes:
  - a. saqueo: El bárbaro debe tener la habilidad de robar y tener más de 80 de fuerza.
  - b. gritoDeGuerra: El bárbaro debe tener un poder de grito de guerra igual a la cantidad de letras de sus habilidades. El poder necesario para aprobar es 4 veces la cantidad de objetos del bárbaro.
  - c. caligrafia: El bárbaro tiene caligrafía perfecta (para el estándar barbárico de la época) si sus habilidades contienen más de 3 vocales y comienzan con mayúscula.

Sabiendo esto, se pide:

- Definir los eventos, modelar las aventuras y dar un ejemplo.
- Definir la función sobrevivientes que tome una lista de bárbaros y una aventura, y diga cuáles bárbaros la sobreviven (es decir, pasan todas las pruebas)

### Punto 4 - Dinastía

**A** - Los bárbaros se marean cuando tienen varias habilidades iguales. Por todo esto, nos piden desarrollar una función que elimine los elementos repetidos de una lista (sin utilizar nub ni nubBy)

```
> sinRepetidos [1,2,3,4,4,5,5,6,7]
[1,2,3,4,5,6,7]
```

**Nota:** Puede usarse recursividad para este punto.

**B** - Los bárbaros son una raza muy orgullosa, tanto que quieren saber cómo van a ser sus descendientes y asegurarse de que los mismos reciban su legado.

El descendiente de un bárbaro comparte su nombre, y un asterisco por cada generación. Por ejemplo "Dave\*", "Dave\*\*", "Dave\*\*\*", etc.

Además, tienen en principio su mismo poder, habilidades **sin repetidos**, y los objetos de su padre, pero antes de pasar a la siguiente generación, utilizan (aplican sobre sí mismos) los objetos. Por ejemplo, el hijo de Dave será equivalente a:

```
(ardilla.varitasDefectuosas) (Barbaro "Dave*" 100 ["tejer","escribirPoesia"])
[ardilla, varitasDefectuosas])
```

Definir la función descendientes, que dado un bárbaro nos de sus infinitos descendientes.

**C** - Pregunta: ¿Se podría aplicar sinRepetidos sobre la lista de objetos? ¿Y sobre el nombre de un bárbaro? ¿Por qué?