Parcial 1 - Algoritmos I Taller: Tema B

Debés entregar el código completo en los campos correspondientes de cada ejercicio del formulario de entrega y subir el archivo .hs correspondiente. Este código debe poder ejecutarse en haskell sin errores. Te recomendamos para ello que pruebes con diferentes ejemplos antes de entregar.

Descripción del problema

Silksong es un videojuego de acción y aventura en 2D. El jugador controla a Hornet, una princesa guerrera, mientras explora el misterioso reino de Pharloom, una tierra desconocida y repleta de peligros.

Ejercicio 1:

Definir el tipo `EnemigoSilksong` que consta de dos constructores `Comun` y `Jefe` con los siguientes parámetros:

- El constructor `Comun` debe tomar como parámetros el nombre, cantidad de vida, cantidad daño y cantidad de rosarios
- El constructor `Jefe` debe tomar como parámetros el nombre, cantidad de vida, cantidad de fases, cantidad de daño y el nombre de la ubicación donde se encuentra

Ejercicio 2:

A partir del tipo definido en el punto anterior, definí los siguientes partidos:

```
pilgrimHiker :: EnemigoSilksong
pilgrimHiker = <COMPLETAR>
```

correspondiente al enemigo comun Pilgrim Hiker, que tiene 50 puntos de vida, 1 punto de daño y 14 rosarios.

```
lace :: EnemigoSilksong
lace = <COMPLETAR>
```

correspondiente al jefe final Lace, que tiene 250 puntos de vida, 3 fases, infligen 2 puntos de daño y se encuentra en Deep Docks.

Ejercicio 3:

Definir la función `jefeDerrotado :: EnemigoSilksong -> Bool` que dado un `enemigo`, devuelve `True` si es un jefe con cantidad de puntos de vida igual a 0, `False` caso contrario.

Definir la función `esPeligroso :: EnemigoSilksong -> Bool` que dado un `enemigo`, devuelve `True` si el enemigo inflige 2 o mas puntos de daño. `False` caso contrario.

Ejercicio 4:

Definir la función `jefesPeligrosos :: [EnemigoSilksong] -> Int` que dada una lista de enemigos, devuelve la cantidad de jefes con cantidad de fases mayor a 2.

Ejercicio 5:

Dado el tipo recursivo `ColaCombate` definido de la siguiente manera

```
data ColaCombate = Vacia | Encolada EnemigoSilksong ColaCombate deriving Show
```

Definir la función `colaDeJefes :: ColaCombate -> String -> ColaCombate` que dada una cola de enemigos, y el nombre de una ubicacion, devuelve la cola de jefes que se encuentran en esa ubicación.