INF-253 Lenguajes de Programación
Tarea 3: Java

Profesor: José Luis Martí Lara, Roberto Diaz Urra
Ayudantes Cátedra: Gabriela Acuña Benito, Hugo Sepúlveda Arriaza,
Lucio Fondón Rebolledo
Ayudantes Tareas: Gabriel Carmona Tabja, Domingo Benoit Cea,
Héctor Larrañaga Jorquera, Ignacio Ulloa Huenul,
Joaquín Gatica Hernández, Javier Pérez Riveros,
José Runín Basáez, Rafael Aros Soto
Rodrigo Pérez Jamett

26 de octubre de 2021

1. Vamos a una aventura mágica

Después de tener el cerebro fundido al hacer la simulación y ver como los animales se mataron entre ellos, provocando una tristeza intensa en el informático anónimo. Este decidió buscar otras pasiones para pasar las penas de su simulación.

Se encontró con juegos de rol de texto como Zork, esto provocó que el informático anónimo se obsesionará mucho con hacer un juego de este estilo. Por lo que en una noche creo las reglas de un mundo mágico e intrigante. El informático anónimo les pide ayuda a ustedes para poder ayudarlo a construir este mundo.

2. The legend of Morio

The legend of Morio es el mundo creado por el informático anónimo, en este mundo habrán dos entes: el Jugador y el dungeon master.

El Dungeon Master (DM) será el encargado de crear el mundo, este mundo constará de solo una dimensión, o sea que se puede ir de izquierda a derecha y es un mundo cíclico, además cada casilla constará con una especificación de tipo de Tierra (Montaña, Planicie, Bosque). El DM también será encargado de situar AL PRINCIPIO del programa NPCs (Buenos, Malos y Neutros) y Enemigos (Monstruos y Jefe Final) en cada una de estas Tierras.

El Jugador será el encargado de controlar el personaje y sus acciones, además de interactuar con los elementos del mundo. Este personaje se podrá mover solo de izquierda a derecha por el mundo y dependiendo de lo que haya en casa casilla tendrá un suceso distinto.

2.1. Misiones

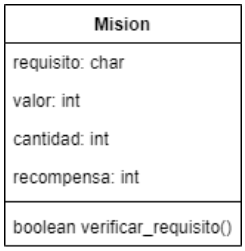
Las misiones serán una clase que constará de los siguiente atributos:

- **requisito**: un carácter que corresponderá al objetivo de la misión. Donde será 'v' si es que debe llegar a una casilla especifica o 'm' si es que debe matar a una cantidad de Monstruos en especificó

- **valor**: coordenada de la casilla que debe llegar o la cantidad de Monstruos que debe matar dependiendo del **requisito**
- **cantidad**: la posición actual del personaje o la cantidad de Monstruos que lleva el Jugador matados
- **recompensa**: la cantidad de **xp** que gana el Jugador por cumplir la misión

Y deberá implementar el siguiente método:

- **verificar_requisito**: debe retornar verdadero o falso dependiendo si la misión ya cumplió su **valor** o no



2.2. Jugador

Un Jugador es una clase abstracta, el cual tendrá los siguientes atributos:

- **nombre**: el **nombre** del Jugador
- **vida**: la **vida** total del Jugador
- **xp**: la experiencia total acumulada del Jugador
- **fuerza**: la **fuerza** del Jugador
- **inteligencia**: la **inteligencia** del Jugador
- **energia**: la **energía** física total del Jugador
- **mana**: el **maná** total del Jugador
- **lista_misiones**: la lista de misiones del Jugador

Todas las clases que hereden a Jugador tendrán que partir con un **nombre** dado por consola y su **xp** en 0. Además cuenta con los siguientes métodos abstractos que tendrá que ser implementada en cada clase hijo:

- **ataque**: debe retornar el daño que haría un ataque del Jugador y deberá a restar una cantidad a su **energía** dependiendo de cada clase, si la **energía** es 0 el daño INDEPENDIENTE DE LA CLASE será 0.
- **hechizo**: debe retornar el daño mágico que haría un hechizo del Jugador y deberá a restar una cantidad a su **maná** dependiendo de cada clase, si el **maná** es 0 el daño INDEPENDIENTE DE LA CLASE será 0.
- **subir_experiencia**: debe recibir la cantidad de experiencia a sumar y debe mejorar los atributos dependiendo de la clase en caso de que suba de nivel. Los niveles en este juego serán los siguientes:

- 1° nivel es cuando uno tiene menos de 10 de **xp**
- 2° nivel es cuando uno tiene menos de 25 de **xp**
- 3° nivel es cuando uno tiene menos de 50 de **xp**
- 4° nivel es cuando uno tiene menos de 100 de **xp**
- 7° nivel es cuando uno tiene menos de 200 de **xp**
- 8° nivel es cuando uno tiene menos de 350 de **xp**
- 9° nivel es cuando uno tiene menos de 600 de **xp**
- 10° nivel es cuando uno tiene menos de 900 de **xp**

Cuando se llegue al 10° nivel no podrá seguir subiendo de nivel.

Esta clase es padre de otras tres clases: mago, guerrero y druida.

- Mago
  - Atributos:
    - La **vida** máxima e inicial es 10
    - La **fuerza** inicial es 3
    - La **inteligencia** inicial es 10
    - La **energía** máxima e inicial es 6
    - El **maná** máxima e inicial es 15
  - **ataque**: el daño realizado por el Jugador mago será

    $$\textbf{fuerza} \cdot \frac{\textbf{inteligencia}}{4} + \textbf{energia}$$

    Además, se le debe restar a la **energía** actual 3.
  - **hechizo**: el daño mágico realizado por el Jugador mago será

    $$\textbf{inteligencia} \cdot \frac{\textbf{fuerza}}{4} + \textbf{mana}$$

    Además, se le debe restar a el **maná** actual 5.
  - **subir_experiencia**: En caso de que suba de nivel se mejorarán de esta forma los atributos:
    - La **vida** máxima se aumenta en $2 \cdot nivel$
    - La **fuerza** se aumenta en 1
    - La **inteligencia** se aumenta en $3 \cdot nivel$
    - La **energía** máxima se aumenta en 1
    - El **maná** máxima se aumenta en $3 \cdot nivel$
- Guerrero
  - Atributos:
    - La **vida** máxima e inicial es 20
    - La **fuerza** inicial es 9
    - La **inteligencia** inicial es 1
    - La **energía** máxima e inicial es 10
    - El **maná** máxima e inicial es 2

- **ataque**: el daño realizado por el Jugador guerrero será:

$$\textbf{fuerza} \cdot 2 + \textbf{energia}$$

  Además, se le debe restar a la **energía** actual 5.

- **hechizo**: el daño mágico realizado por el Jugador guerrero será

$$\textbf{inteligencia} \cdot \frac{\textbf{fuerza}}{4} + \textbf{mana}$$

  Además, se le debe restar a el **maná** actual 3.

- **subir_experiencia**: En caso de que suba de nivel se mejorarán de esta forma los atributos:
  - La **vida** máxima se aumenta en $3 \cdot nivel$
  - La **fuerza** se aumenta en $5 \cdot nivel$
  - La **inteligencia** se aumenta en 1
  - La **energía** máxima se aumenta en $2 \cdot nivel$
  - El **maná** máxima se aumenta en 1

- Druida
  - Atributos:
    - La **vida** máxima e inicial es 15
    - La **fuerza** inicial es 5
    - La **inteligencia** inicial es 5
    - La **energía** máxima e inicial es 5
    - El **maná** máxima e inicial es 5
  - **ataque**: el daño realizado por el Jugador druida será

$$\frac{\textbf{fuerza} + \textbf{inteligencia}}{3} \cdot max(\frac{\textbf{energia}}{3}, \frac{\textbf{mana}}{2})$$

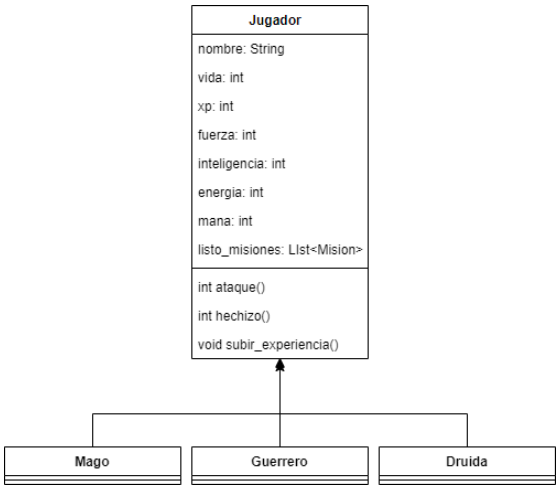
    Además, se le debe restar a la **energía** actual 3.

  - **hechizo**: el daño mágico realizado por el Jugador druida será

$$\frac{\textbf{fuerza} + \textbf{inteligencia}}{3} \cdot max(\frac{\textbf{energia}}{2}, \frac{\textbf{mana}}{3})$$

    Además, se le debe restar al **maná** actual 3.

  - **subir_experiencia**: En caso de que suba de nivel se mejorarán de esta forma los atributos:
    - La **vida** máxima se aumenta en *nivel*
    - La **fuerza** se aumenta en *nivel*
    - La **inteligencia** se aumenta en *nivel*
    - La **energía** máxima se aumenta en *nivel*
    - El **maná** máxima se aumenta en *nivel*



2.3. NPC

Los NPC serán una clase abstracta que tendrá el atributo **nombre** y el método **interacción** en la cual dependiendo de si es Bueno, Malo o Neutro hará algo distinto. El método interacción siempre recibirá un jugador.

- Los NPC buenos tendrán la característica que siempre aumentarán algún atributo del Jugador, para ello tendrá los siguientes atributos:

  - **atributo**: un string que corresponderá al **nombre** del atributo que será aumentado, donde solo podrá ser: **vida**, **xp**, **energía** y **maná**
  - **cantidad**: la cantidad en que será aumentado el atributo.

  El método **interacción** deberá mostrar por pantalla:

  "< **nombre** >: Creo que necesitas un poco de ayuda te subire < **cantidad** > a tu < **atributo** >!!!!"

  Donde **nombre**, **cantidad** y **atributo** deben ser reemplazados por los valores que estos contengan y sumarle a ese **atributo** la **cantidad** del NPC bueno. Si la **energía** o **maná** del Jugador quedan mayores que la cantidad máxima en ese momento, debe ser reducida hasta su máximo posible.

- Los NPC malos tendrán la característica que intentarán empeorar la **energía** y **maná** del Jugador, para ello tendrá los siguientes atributos:

  - **cantidad_energia**: la cantidad de **energia** que disminuirá.
  - **cantidad_mana**: la cantidad de **mana** que disminuirá.

  El método **interacción** deberá mostrar por pantalla:

  "< **nombre** >: SOY MALO TE VOY A DISMINUIR TU ENERGIA y MANA EN < **cantidad_energia** > y < **cantidad_mana** >"

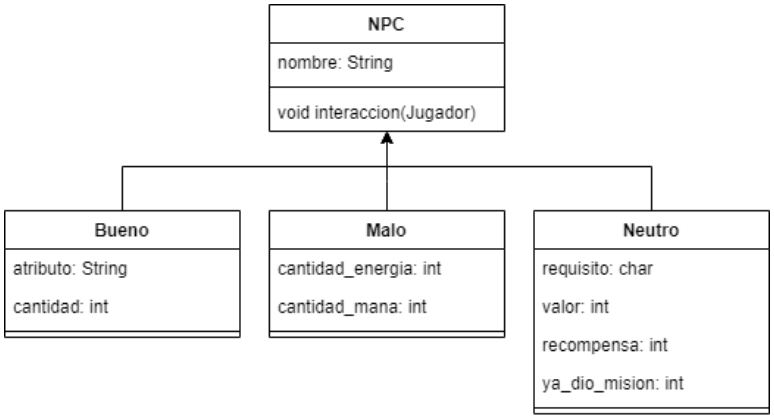
  Donde **nombre**, **cantidad_energia** y cantidad_**mana** deben ser reemplazados por los valores que estos contengan. Finalmente deberá tirar un número al azar entre 1 y 10 si el número es menor o igual a 6 se disminuye la **energía** del Jugador en **cantidad_energia**, luego debe repetir este proceso para el **maná** del Jugador.

- Los NPC neutros tendrán la característica que otorgarán una misión al Jugador a cambio de una **recompensa**, para ello tendrá los siguientes atributos:

- **requisito**: el **requisito** de la misión
- **valor**: el **valor** que debe cumplir el Jugador en la misión
- **recompensa**: la **recompensa** que recibiría el Jugador al cumplir la misión
- **ya_dio_mision**: un booleano que sea verdadero o falso dependiendo si ya dio la misión o no

El método **interacción** deberá hacer una acción dependiendo de **ya_dio_mision**:

- Si **ya_dio_mision** es falso deberá mostrar por pantalla
  "< **nombre** >: hola, ¿quieres cumplir esta mision? Deberas < **(llegar a/matar)** > < **valor** > < **(del mundo/de Monstruos)** > y recibirás < **recompensa** > de xp"
  Donde **nombre**, **valor** y recompensa deben ser reemplazados por los valores que estos contengan, además de usar la palabras según el **requisito**: **llegar a** y **del mundo** para 'v'; **matar** y **de Monstruos** para 'm'), el Jugador debe poder rechazar o aceptar la misión, en caso de aceptar se agrega la misión a la lista de misiones del Jugador y cambiar el valor de **ya_dio_mision** a verdadero.
- Si **ya_dio_mision** es verdadero deberá mostrar por pantalla "**nombre**: ya te di mision, saludos."



2.4. Enemigos

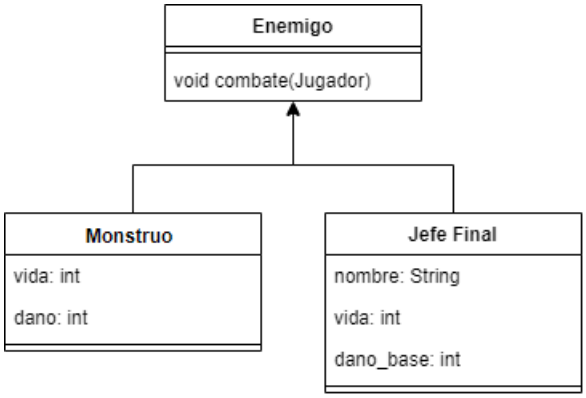
Los Enemigos serán una interfaz que tendrá el método combate que recibirá un Jugador, en la cual dependiendo de si es Monstruo o Jefe Final tendrá una implementación distinta, pero SIEMPRE será un combate por turno en el que partirá el Jugador, luego le seguirá el Enemigo y deberá repetir estos turnos hasta que alguno de los dos tenga **vida** menor o igual a 0.

- Los Monstruos serán simples y tendrán los siguientes atributos:
  - **vida**: la cantidad de **vida** del Monstruo
  - **dano**: el daño que realizará el Monstruo

  En el combate, el Jugador debe decidir si hacer un ataque o un hechizo y dependiendo de esa elección hacerle daño al Monstruo, luego es turno del Monstruo donde SIEMRPE hará el mismo daño.
- El Jefe Final consistirá en un Enemigo que tendrá 2 fases y tendrá los siguientes atributos:
  - **nombre**: el **nombre** del Jefe Final

- **vida**: la cantidad de **vida** del Jefe Final
- **dano_base**: el daño que hará el Jefe Final

El combate parte con el Jefe Final en fase 1. El Jugador debe decidir si hacer un ataque o un hechizo y dependiendo de esa elección hacerle daño al Jefe Final, luego es turno del Monstruo que hará **dano_base** $+ 2 \cdot fase\_actual$, si la **vida** del Jefe Final es menor o igual a la mitad de su **vida** inicial está en la fase 1, si no esta en la fase 2.



2.5. Tierra

La tierra será una clase abstracta que tendrá los siguientes atributos:

- **probabilidad_enemigo**: la probabilidad de que cuando llegue el Jugador aparezca un Enemigo (0,0 hasta 1,0)
- **enemigo**: el tipo de Enemigo que podría aparecer, es una instanciación de Monstruo o Jefe Final (con atributos inicializados desde el inicio del programa)
- **npc**: el NPC que está en esta tierra, es una instanciación de bueno, malo o neutro (con atributos inicializados desde el inicio del programa)

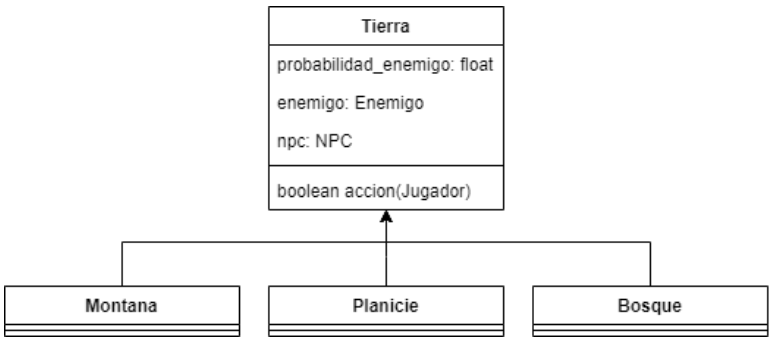
Y tendrá el siguiente método abstracto a implementar:

- **accion**: recibirá un Jugador, deberá realizar ver si el Jugador puede llegar dependiendo de la clase que lo implemente, en caso de que pueda llegar se de proceder con lo siguiente, el Jugador deberá interactuar con el NPC (si es que hay) y finalmente ver si aparece un Enemigo basado en la probabilidad. En el caso que aparezca un Enemigo el Jugador debe combatir con ese Enemigo. Debe retornar verdadero si es que pudo llegar, hacer las acciones y quedar vivo después del combate, si no cumple alguna de las condiciones debe retornar falso.

Esta clase heredará a las clases Montaña, Planifica y Bosque. Y esto tendrá que implementar cada una:

- Montaña
  - **accion**: si la **energía** del Jugador es 0 significa que no puede llegar a la montaña. Si su **energía** es mayor a 0, al llegar el Jugador debe restarle 3 a su **energía**, si la **energía** del Jugador es menor a 3 debe restarle a su **vida** lo que falte. Si es que sigue vivo deberá hacer todo lo descrito anteriormente. es que sigue vivo deberá hacer todo lo descrito anteriormente.

- Planicie
  - **accion**: el Jugador siempre puede llegar y no pierde nada por llegar. Luego el Jugador deberá hacer todo lo descrito anteriormente.
- Bosque
  - **accion**: si el **maná** del Jugador es 0 significa que no puede llegar a la montaña. Si su **maná** es mayor a 0, al llegar el Jugador debe restarle 3 a su **maná**, si el **maná** del Jugador es menor a 3 debe restarle a su **vida** lo que falte. Si es que sigue vivo deberá hacer todo lo descrito anteriormente.



2.6. Main

Al principio debe permitir que el DM ingrese el tamaño del mundo, luego irá casilla por casilla rellenándola de la siguiente forma:

1. Qué tipo de tierra es.
2. La probabilidad de que un Enemigo aparezca
3. Qué tipo de Enemigo podría aparecer
4. Los atributos del Enemigo
5. Si es que habrá un NPC
6. Si la respuesta es que sí, deberá pedir:
   - Qué tipo de NPC es
   - Los atributos del NPC

Después de rellenar todas las casillas el DM debe ingresar la posición inicial del Jugador. Finalmente debe dejarle crear su personaje al Jugador, donde este deberá ingresar su nombre y que clase quiere ser.

Al crear su personaje se da comienzo al juego, donde debe existir un menú que permita mover al Jugador (izquierda o derecha) y poder ver cada atributo del Jugador con sus detalles. Cada vez que se mueva un Jugador debe llamarse al método **accion** de esa Tierra, en caso de que esa función retorna verdadero se continua el juego, si retorna falso debe verificar que la vida del Jugador no haya alcanzado el valor 0. Si alcanzó el valor 0 el juego termina, sino significa que el Jugador no se pudo mover, por ende se devuelve a su casilla donde estaba originalmente. Si el Jugador entró a una Tierra y tuvo un combate con un Jefe Final, el juego termina y el Jugador gano.

Todos los ingresos al programa debe ser por la entrada estándar.

3. Datos de Vital Importancia

- Si alguna división no da exacta se debe redondear hacia abajo
- Luego de subir de nivel deberá recuperar la **vida**, la **energía** y el **maná** a su máximo actual
- Si en algún punto del programa la **energía** o **maná** sean menores o iguales que 0 se deben dejar en 0
- Si en algún punto del programa la **vida** es menor o igual que 0 se debe dejar en 0
- Por cada acción, situación o cosa que sucede se debe mostrar a través de la salida estándar indicando claramente que esta pasando. Por ejemplo, al moverme de casilla, se deberá mostrar por pantalla si es que pude llegar, si que había NPC, que paso con el NPC, si es que aparece enemigo y el combate con el enemigo.
- Cada atributo debe tener un método para obtenerlo (getter) y un método para modificarlo (setter). Esto quiere decir que todos los atributos a usar deberán ser privados.
- Las misiones deben partir siempre con la **cantidad** en 0. Luego cada vez que sea necesario debe modificar ese valor e ir verificando si cumplió la misión o no. En caso de que la misión se cumple, debe darle la recompensa al jugador y eliminarla de la lista de misiones del jugador.
- Se pueden agregar atributos, métodos y clases según lo estime conveniente, pero aquello que esta presente en el enunciado debe ir sí o sí.

4. A entregar

- Main.java
- Mision.java
- Jugador.java, Mago.java, Guerrero.java y Druida.java
- NPC.java, Bueno.java, Malo.java y Neutro.java
- Enemigos.java, Monstruos.java y Jefe_Final.java
- Tierra.java, Montana.java, Planicie.java y Bosque.java
- makefile
- readme.txt

5. Sobre Entrega

- Se deberá ocupar el compilador javac, con versión 13.0.7 o mayor
- El código debe venir ordenado
- Los casos de borde pueden ser manejados según lo considere adecuado, indicando claramente su acercamiento en el README y cuidando de no contradecir el resto del enunciado
- Se darán puntos por creatividad :D, máximo 10 puntos extras, si su nota supera el 100 quedará como 100.

- Cada método DEBE llevar arriba un comentario que indique nombre de la función, parámetros que recibe, una breve descripción de lo hace y lo que retorna
- Debe estar presente el archivo **makefile** para que se efectúe la revisión, este debe compilar **TODOS** los archivos. Si su código viene sin makefile (o el makefile no funciona), no se revisará su tarea
- El trabajo es individual
- La entrega debe realizarse en tar.gz y debe llevar el nombre: Tarea3LP_RolIntegrante.tar.gz
- El archivo README.txt debe contener nombre y rol del alumno e instrucciones detalladas para la compilación, utilización de su programa y cualquier detalle extra que se relevante. SE DESCONTARÁ SI FALTA ALGO DE LO SEÑALADO
- La entrega será vía aula y el plazo máximo de entrega es hasta el **12 de noviembre a las 23:55 hora aula**
- Por cada día de atraso se descontarán 20 puntos
- Las copias serán evaluadas con nota 0 y se informarán a las respectivas autoridades

6. Calificación

- Jugador (30 pts):
  - Mago (10 pts)
  - Guerrero (10 pts)
  - Druida (10 pts)
- NPC (30 pts):
  - Bueno (10 pts)
  - Malo (10 pts)
  - Neutro (10 pts)
- Tierra (30 pts)
  - Monstaña (10 pts)
  - Planicie (10 pts)
  - Bosque (10 pts)
- Misiones (5 pts)
- Main (5 pts)
- Descuentos:
  - Warning (-5 pts c/u, max -30pts)
  - getter o setter faltante (-5 pts c/u)
  - No compila (-100pts)
  - Reglas de entrega (max -30pts)