

Aventuras

Proyecto de la convocatoria extraordinaria de Taller de Programación del curso 2023-2024

ETSI Sistemas Informáticos - Universidad Politécnica de Madrid

ÍNDICE:

<!-- TOC -->

* [Aventuras](#)

- [Objetivos del proyecto](#)
- [Introducción](#)
- [Componentes del sistema](#)
 - [Sala](#)
 - [Ítem](#)
 - [Monstruo](#)
 - [Trampa](#)
 - [Personaje](#)
 - [Motor del juego](#)
- [Funcionalidades del juego](#)
 - [Parámetros de ejecución](#)
 - [Inicialización del juego](#)
 - [Fichero de salas](#)
 - [Fichero de ítems](#)
 - [Fichero de monstruos](#)
 - [Fichero de trampas](#)
 - [Creación del personaje](#)
 - [Bucle de juego](#)
 - [Combatir con monstruos](#)
 - [Esquivar trampas](#)
 - [Recoger ítems](#)
 - [Moverse por las salas](#)
 - [Finalización del juego](#)

- [Gestión de puntuaciones](#)
- [Instrucciones para el desarrollo del Proyecto](#)
- [Normativa y evaluación](#)
- [Entrega del proyecto](#)

<!-- TOC -->

Objetivos del proyecto

El desarrollo de esta práctica pretende que el alumnado aplique la mayoría de los conceptos fundamentales de programación mediante la implementación de un proyecto de programación para la asignatura **Taller de Programación (TP)**. En concreto, el alumnado aplicará conceptos de:

- Manejo de Entrada y Salida
- Estructuras de control
- Manejo de Funciones
- Clases y Objetos
- Vectores y Matrices
- Manejo de Ficheros

Para ello, el alumnado desarrollará de forma grupal (por parejas) un proyecto de programación en lenguaje Java mediante el uso del entorno de programación *IntelliJ IDEA*, el cual proporciona las herramientas necesarias para el desarrollo, pruebas, documentación y depuración de código fuente en Java.

Introducción

Este proyecto consiste en la creación de un juego conversacional de aventuras en el que el jugador deberá superar una serie de habitaciones para poder avanzar en la historia. El juego se desarrollará en un entorno de consola y se podrá jugar en cualquier sistema operativo.

El objetivo del juego es encontrar la salida de un laberinto de habitaciones. Para ello, el jugador deberá recoger objetos para maximizar el valor de monedas con el que escapa.

Componentes del sistema

Los componentes que deberá tener el juego son los siguientes:

Sala

Una sala es un lugar en el que el personaje puede estar. Cada sala tiene un nombre y una descripción. Además, cada sala puede tener una serie de objetos que el jugador puede recoger. Los datos que definen una sala son los siguientes:

- **Descripción:** Texto que describe la sala.
- **Ítems:** Lista de objetos que contiene la sala.
- **Monstruos:** Lista de monstruos que contiene la sala.

- **Trampas:** Lista de trampas que contiene la sala.
- **Número máximo de ítems** que puede contener la sala
- **Número máximo de monstruos** que puede contener la sala
- **Número máximo de trampas** que puede contener la sala
- **Fila y columna** en la que se encuentra la sala en el laberinto.

Las salas están definidas en un fichero de texto que se cargará al inicio del juego. El fichero de texto contendrá la información de todas las salas del juego. Cada sala estará definida por un bloque de texto que contendrá la información de la sala.

Las salas pueden tener puertas en las direcciones norte, sur, este y oeste. Si una sala tiene una puerta en una dirección, el jugador podrá moverse a la sala que está al otro lado de la puerta.

Ítem

Un ítem es un objeto que el jugador puede recoger. Los datos que definen un ítem son los siguientes:

- **Descripción:** Texto que describe el ítem.
- **Valor:** Valor en monedas del ítem.
- **Peso:** Peso del ítem en kg.
- **Fila y columna** de la habitación en la que se encuentra el ítem.

Los ítems están definidos en un fichero de texto que se cargará al inicio del juego.

Monstruo

Un monstruo es un enemigo que el jugador puede encontrar en una sala. Los datos que definen un monstruo son los siguientes:

- **Nombre:** Nombre del monstruo.
- **Vida:** Puntos de vida del monstruo.
- **Ataque:** Puntos de ataque del monstruo.
- **Defensa:** Puntos de defensa del monstruo.
- **Fila y columna** de la habitación en la que se encuentra el monstruo.

Los monstruos están definidos en un fichero de texto que se cargará al inicio del juego.

Trampa

Una trampa es un obstáculo que el jugador puede encontrar en una sala. Los datos que definen una trampa son los siguientes:

- **Descripción:** Texto que describe la trampa.
- **Daño:** Puntos de daño que hace la trampa al jugador.
- **Fila y columna** de la habitación en la que se encuentra la trampa.

Las trampas están definidas en un fichero de texto que se cargará al inicio del juego.

Personaje

El personaje es el jugador que controla el usuario. Los datos que definen un personaje son los siguientes:

- **Nombre:** Nombre del personaje.
- **Vida:** Puntos de vida del personaje.
- **Ataque:** Puntos de ataque del personaje.
- **Defensa:** Puntos de defensa del personaje.
- **Destreza:** Puntos de destreza del personaje.
- **Inventario:** Lista de ítems que lleva el personaje.
- **Número máximo de ítems** que puede llevar el personaje.
- **Peso máximo de ítems** que puede llevar el personaje.

El personaje se creará al inicio del juego y se podrá mover por las salas del laberinto. El personaje podrá recoger ítems, luchar contra monstruos y esquivar trampas.

Motor del juego

El motor del juego es el encargado de gestionar el flujo del juego. El motor del juego se encargará de cargar las salas, los ítems, los monstruos y las trampas del juego. Además, el motor del juego se encargará de gestionar las acciones del jugador, como moverse por las salas, recoger ítems, luchar contra monstruos y esquivar trampas. Está definido por:

- **Mapa:** Matriz de salas que forman el laberinto.
- **Máximo número de ítems, monstruos y trampas** que puede haber en cada sala.

Funcionalidades del juego

El juego deberá tener las siguientes funcionalidades:

Parámetros de ejecución

El juego deberá poder ejecutarse con los siguientes parámetros de línea de comandos (en ese orden):

- **filas:** Número de filas del mapa.
- **columnas:** Número de columnas del mapa.
- **max_items:** Número máximo de ítems que puede haber en cada sala.
- **max_monstruos:** Número máximo de monstruos que puede haber en cada sala.
- **max_trampas:** Número máximo de trampas que puede haber en cada sala.
- **fichero_salas:** Fichero de texto con la información de las salas.
- **fichero_items:** Fichero de texto con la información de los ítems.
- **fichero_monstruos:** Fichero de texto con la información de los monstruos.

- **fichero_trampas**: Fichero de texto con la información de las trampas.
- **fichero_puntuaciones**: Fichero de texto con las puntuaciones de otras partidas.

Inicialización del juego

Al inicio del juego, el motor del juego cargará las salas, los ítems, los monstruos y las trampas del juego. Para ello, leerá los ficheros correspondientes y creará las estructuras de datos necesarias para representar el juego. A continuación se define el formato de los ficheros de texto que contienen la información de las salas, los ítems, los monstruos y las trampas.

Fichero de salas

El fichero de salas contendrá la información de todas las salas del juego. Cada sala estará definida por una línea de texto que contendrá los siguientes campos separados por el caracter **;**:

```
fila;columna;descripcion
```

Por ejemplo, el fichero de salas podría tener el siguiente contenido:

```
0;0;Habitación inicial
0;1;Bosque encantado
0;3;Guarida del dragón
1;1;Escondite del duende
1;2;Prado de las hadas
1;3;Cementerio encantado
2;1;Cabaña de la bruja
2;3;Torre del mago
3;1;Mina de los enanos
3;3;Pueblo elfo
4;1;Puente del minotauro
4;2;Río místico
4;3;Habitación de salida
```

Fichero de ítems

El fichero de ítems contendrá la información de todos los ítems del juego. Cada ítem estará definido por una línea de texto que contendrá los siguientes campos separados por el caracter **;**:

```
fila;columna;descripcion;valor;peso
```

Por ejemplo, el fichero de ítems podría tener el siguiente contenido:

```
0;0;Espada Mágica;1.5;100
0;3;Poción de Curación;0.5;50
0;3;Escudo de Oro;3.0;150
0;3;Daga de Plata;0.7;70
1;1;Armadura Encantada;5.0;200
1;1;Amuleto Místico;0.3;120
1;2;Sombrero de Mago;0.2;80
1;2;Oro del Goblin;0.5;50
3;1;Escala de Dragón;2.0;100
1;2;Ala de Hada;0.1;90
3;1;Martillo Enano;2.5;110
3;3;Arco Elfo;1.0;90
3;3;Club de Troll;3.0;80
```

```
3;1;Hacha de Orco;2.5;70
4;1;Bota de Gigante;4.0;60
4;1;Medalla de Héroe;0.3;200
4;2;Pergamino Antiguo;0.2;150
4;2;Orbe Místico;1.0;100
4;3;Anillo Encantado;0.1;120
4;3;Bola de Cristal;1.0;100
```

Fichero de monstruos

El fichero de monstruos contendrá la información de todos los monstruos del juego. Cada monstruo estará definido por una línea de texto que contendrá los siguientes campos separados por el caracter `;`:

```
fila;columna;nombre;vida;ataque;defensa
```

Por ejemplo, el fichero de monstruos podría tener el siguiente contenido:

```
0;0;Trasgo;20;5;2
0;1;Orco;30;10;4
0;1;Troll;50;15;6
0;3;Dragón;100;20;8
1;1;Araña Gigante;40;8;3
1;1;Duende;25;5;1
1;3;Vampiro;60;12;5
1;3;Hombre Lobo;80;16;7
1;3;Fantasma;45;9;3
2;1;Bruja;55;11;4
4;1;Minotauro;65;13;6
4;1;Arpía;35;7;2
4;3;Jefe Final;120;24;10
```

Fichero de trampas

El fichero de trampas contendrá la información de todas las trampas del juego. Cada trampa estará definida por una línea de texto que contendrá los siguientes campos separados por el caracter `;`:

```
fila;columna;descripcion;daño
```

Por ejemplo, el fichero de trampas podría tener el siguiente contenido:

```
0;1;Foso;10
0;3;Trampa de Flechas;5
2;3;Trampa de Fuego;15
2;3;Paredes Móviles;10
3;1;Trampa de Electricidad;15
3;3;Trampa de Hielo;15
```

Creación del personaje

Al inicio del juego, el jugador deberá crear un personaje proporcionando el nombre y la distribución de puntos de vida, ataque, defensa y destreza. La aplicación irá solicitando estos datos por pantalla, teniendo en cuenta que la suma de los puntos de vida, ataque, defensa y destreza no pueden superar los 250 puntos. Cabe destacar que los valores mínimos son 50 puntos de vida y 1 punto de ataque, defensa y destreza.

El número máximo de ítems que puede llevar el personaje y el peso máximo de los ítems que puede llevar el personaje se establecerán en función de los puntos de destreza y ataque del personaje. El número de ítems se corresponde con el valor entero de la división de los puntos de destreza entre 4, con un mínimo de 1 objeto. El peso máximo de los ítems se corresponde con el valor entero de la división de los puntos de ataque entre 2, con un mínimo de 1.0 kg.

Un ejemplo de traza de la creación del personaje sería el siguiente:

```
¿Cómo te llamas? Raul
¡Hola, Raul! Tienes 250 puntos para repartir entre vida, ataque, defensa y destreza.
¿Cuánta vida quieres tener? (50-247): 100
¿Cuánto ataque quieres tener? (1-148): 100
¿Cuánta defensa quieres tener? (1-49): 25
¿Cuánta destreza quieres tener? (1-25): 25
```

Es importante destacar que hay que controlar el rango de valores que puede introducir el usuario. Si el usuario introduce un valor fuera del rango permitido, se le deberá solicitar de nuevo.

Bucle de juego

Una vez creado el personaje, el juego comenzará. El jugador podrá moverse por las salas del mapa, recoger ítems, luchar contra monstruos y esquivar trampas. El juego terminará cuando el jugador encuentre la salida del mapa o cuando el jugador muera. Importante: la salida del laberinto estará en la sala que se encuentra en la última fila y última columna del mapa, mientras que la sala inicial estará en la primera fila y primera columna del mapa.

El bucle principal del juego deberá tener la siguiente estructura:

1. Mostrar la descripción de la sala en la que se encuentra el jugador.
2. Mientras haya monstruos en la sala, mostrar la lista de monstruos y pedir al jugador que elija uno para luchar.
3. Si hay trampas en la sala, decidir si el jugador esquiva la trampa o recibe daño.
4. Mientras haya objetos en la sala y el jugador no haya decidido irse, mostrar la lista de objetos de la sala y pedir al jugador que elija uno para recoger.
5. Mostrar el mapa y las opciones de movimiento al jugador y pedir al jugador que elija una dirección para moverse.

Combatir con monstruos

Cuando el jugador entra en una sala con monstruos, el jugador deberá luchar contra uno de los monstruos de la sala. El jugador podrá elegir uno de los monstruos de la sala para luchar. El jugador y el monstruo se atacarán mutuamente hasta que uno de los dos muera, teniendo en cuenta que el jugador siempre ataca primero.

El daño que hace el jugador al monstruo se corresponde con la diferencia entre el ataque del jugador y la defensa del monstruo. El daño que hace el monstruo al jugador se corresponde con la diferencia entre el ataque del monstruo y la defensa del jugador.

Un ejemplo de traza de un combate sería el siguiente:

```
Monstruos en la sala:
[ Orco (V: 30, A: 10, D: 4) ]
[ Troll (V: 50, A: 15, D: 6) ]
Escribe el nombre del monstruo que quieres atacar: Troll
{ Raul (V: 125, A: 100, D: 5) } ataca a [ Troll (V: 50, A: 15, D: 6) ] con 100 puntos
de daño
¡Has derrotado al monstruo!
Monstruos en la sala:
[ Orco (V: 30, A: 10, D: 4) ]
Escribe el nombre del monstruo que quieres atacar: Orco
{ Raul (V: 125, A: 100, D: 5) } ataca a [ Orco (V: 30, A: 10, D: 4) ] con 100 puntos
de daño
¡Has derrotado al monstruo!
```

Esquivar trampas

Cuando el jugador entra en una sala con trampas habrá que determinar si el jugador esquiva la trampa o recibe daño. Para ello, se generará un número aleatorio entre 1 y 50. Si el número aleatorio es menor o igual a la destreza del jugador, el jugador esquivará la trampa. En caso contrario, el jugador recibirá el daño de la trampa.

Un ejemplo de traza de procesar trampas sería el siguiente:

```
¡Has esquivado la trampa! Trampa de Fuego
¡Has caído en una trampa! Paredes Móviles
Te ha hecho 10 puntos de daño
```

Recoger ítems

Cuando el jugador entra en una sala con ítems, el jugador podrá recoger uno de los ítems de la sala. El jugador podrá llevar un número máximo de ítems y un peso máximo de ítems. Si el jugador intenta recoger un ítem y no puede llevarlo, se le mostrará un mensaje indicando que no puede llevar más ítems.

El jugador puede seguir recogiendo ítems mientras no haya alcanzado el número máximo de ítems o el peso máximo de ítems. Cuando el jugador haya decidido no recoger más ítems, lo indicará con la orden **NINGUNO**.

Cada vez que el jugador recoge un ítem, se le mostrará un mensaje indicando que ha recogido el ítem y se le mostrará el contenido de su mochila y el valor total de los ítems que lleva.

Un ejemplo de traza de recoger ítems sería el siguiente:

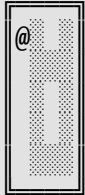
```
Ítems en la sala:
[ Espada Mágica (V: 1.5, P: 100) ]
[ Poción de Curación (V: 0.5, P: 50) ]
[ Escudo de Oro (V: 3.0, P: 150) ]
[ Daga de Plata (V: 0.7, P: 70) ]
Escribe la descripción del ítem que quieres coger (NINGUNO para cancelar): Espada
Mágica
¡Te guardas el objeto! | Espada Mágica (Peso: 1,5, Valor: 100,0) |
Mochila de Raul:
| Espada Mágica (Peso: 1,5, Valor: 100,0) |
```


Peso 1.5 kg (12.5)
Tu mochila vale 100.0 monedas

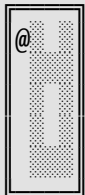
Moverse por las salas

Una vez que se han procesado los monstruos, las trampas y los ítems de la sala, el jugador podrá moverse por las salas del mapa. El jugador podrá moverse en las direcciones norte, sur, este y oeste. Para ello, se le mostrará al jugador el mapa y se le pedirá que elija una dirección para moverse.

Un ejemplo de traza de moverse por las salas sería el siguiente:



Introduce el movimiento (N, E, S, O): S
No puedes moverte al sur



Introduce el movimiento (N, E, S, O): E

Finalización del juego

El juego terminará cuando el jugador encuentre la salida del mapa o cuando el jugador muera. Si el jugador encuentra la salida del mapa, se le mostrará un mensaje indicando que ha ganado el juego y se le mostrará el valor total de los ítems que lleva en la mochila. Si el jugador muere, se le mostrará un mensaje indicando que ha perdido el juego y se le mostrará el valor total de los ítems que llevaba en la mochila.

Gestión de puntuaciones

El juego deberá llevar un registro de las puntuaciones de las partidas. Al inicio del juego, se cargarán las puntuaciones de las partidas anteriores desde un fichero de texto, cuya ruta se ha pasado por parámetro a la aplicación. Al finalizar el juego, se guardará la puntuación de la partida actual en el fichero de texto. Este fichero debe abrirse en modo de añadir información **sin sobrescribir**.

Cada partida se registra en una línea de texto con el siguiente formato:

fecha jugador, monedas

Por ejemplo, el fichero de puntuaciones podría tener el siguiente contenido:

```
2024-04-04 { Raul (V: -4, A: 50, D: 40, X: 20) }, 420.0 monedas
2024-04-06 { Edgar (V: 10, A: 30, D: 30, X:20) }, 620.0 monedas
2024-04-08 { Raul (V: 50, A: 100, D: 25, X: 25) }, 800.0 monedas
```

Instrucciones para el desarrollo del Proyecto

El proyecto de programación debe realizarse teniendo en cuenta las siguientes instrucciones:

- Se entrega un esqueleto de código (Skeleton.zip) incluyendo cada una de las clases obligatorias que deberá tener el proyecto. Estas clases contienen únicamente la componente pública no desarrollada (métodos públicos de cada clase). Se deberá definir tanto la componente privada como desarrollar todos los métodos definidos. **NO** está permitido alterar los nombres, parámetros ni tipo de retorno de ninguno de estos métodos. La alteración se evaluará con un 0 para el método alterado.
- Adicionalmente, la clase **Motor** define el atributo obligatorio **Sala[][] mapa**, el cual deberá ser usado para tratar la matriz de salas del mapa y no podrá ser alterado ni cambiado por otro atributo.
- Se permite el desarrollo tanto de clases como métodos adicionales a los proporcionados en el esqueleto de código, siempre y cuando estos no sean redundantes y sean invocados desde alguna de las funcionalidades definidas en el esqueleto de código.
- Es obligatorio usar las variables pasadas como argumento en los métodos. Su no uso implicará la calificación correspondiente a dicho método con un 0 (por ejemplo, si en la función **crearPersonaje** que recibe como parámetro un Scanner, se crea y usa otro objeto Scanner distinto).
- Todas las clases, atributos y métodos deben estar documentados utilizando Javadoc.

Normativa y evaluación

El proyecto de programación se evaluará teniendo en cuenta los siguientes aspectos:

- El proyecto se realizará de forma grupal (por parejas). Cada pareja deberá desarrollar de manera independiente su propio proyecto de programación y realizar su propia entrega.
- El plagio del Proyecto queda estrictamente prohibido. La detección de plagio supondrá una calificación de 0 en la convocatoria de la asignatura para todos los alumnos implicados, así como la posibilidad de apertura de expediente académico disciplinar.
- Para ser evaluado del proyecto es obligatorio entregarlo en plazo, habiendo realizado correctamente al menos una funcionalidad de las pedidas (tanto el método en la clase **Aventura** como sus clases asociadas para poder llevarla a cabo), y presentarse al examen final del proyecto. Si la calificación del examen es inferior a un 4, no se evaluará la entrega del proyecto. En este caso, se contabilizará la nota del examen, mientras que la nota de la entrega será 0.
- El caso de estudio de partida supone el enunciado base para la realización del proyecto de programación de la asignatura. Este enunciado podrá ser actualizado por el profesorado de la asignatura con fines aclaratorios y/o ejemplificadores.

Entrega del proyecto

El Proyecto de Programación de la asignatura Taller de Programación (TP) consta de una entrega. La fecha límite de entrega será anunciada en el Moodle de la asignatura.

La práctica se entregará a través del Moodle de la asignatura. Cada entrega debe contener al menos un proyecto de IntelliJ IDEA con todos los ficheros de configuración, documentación y código fuente de dicho proyecto (no es necesario entregar los ficheros de bytecodes).