

THE KING



ÍNDICE

HOJA DE CONTROL DE VERSIONES.....	3
DESCRIPCIÓN.....	3
TEMA Y OBJETIVO.....	3
CONTEXTO.....	4
AUDIENCIA Y ESTILO DE JUEGO.....	4
REGLAS.....	4
MECÁNICAS Y JUGABILIDAD.....	5
ANÁLISIS DEL PROYECTO.....	5
CONTROL DE REQUISITOS.....	5
REQUISITOS FUNCIONALES.....	5
REQUISITOS NO FUNCIONALES.....	9
ESTRUCTURA DEL PROYECTO.....	10
DIAGRAMA MODULAR.....	11
Clases/Misiones/Misiones.java.....	12
Clases/Personajes/Personaje.java.....	13
Clases/Interfaz/Menus.java.....	14
Clases/Personajes/Personaje.java,Asesino.java,Luchador.java.....	15
Clases/Entrenamiento/Entrenamiento.java.....	15
Clases/Misiones/TablonMisiones.java.....	15
HERRAMIENTAS DE DESARROLLO.....	16
POSIBLES MEJORAS FUTURAS.....	16

HOJA DE CONTROL DE VERSIONES

VERSIÓN	FECHA	AUTOR	CAMBIOS
1.0	11/2024	IBG	Creación de la primera versión del juego
1.1	01/2025	IBG	Modularización del juego en métodos
1.3	01/2025	IBG	Actualización del documento
1.4	02/2025	IBG	Creación de clases y mejora de funcionalidad
1.6	03/2025	IBG	Implementación de Personajes extra y optimización general
1.7	03/2025	IBG	Implementación de la clase Partida



DESCRIPCIÓN

TEMA Y OBJETIVO

The King está ambientado en la época de la baja edad media cuando había feudos, castillos, caballeros... El juego es un RPG basado en consola donde los jugadores eligen un personaje único con estadísticas propias, realizan misiones, entrenan sus habilidades y compran objetos en una tienda. El objetivo principal es convertirse en el rey del reino o morir en el intento.

CONTEXTO

El juego sumerge al jugador en un mundo medieval donde debe completar misiones, entrenar habilidades y administrar recursos como dinero y salud. Cada personaje tiene un trasfondo único que se refleja en sus estadísticas iniciales, ofreciendo una experiencia personalizada para cada elección.

AUDIENCIA Y ESTILO DE JUEGO

- Audiencia: Ideal para jugadores interesados en experiencias de rol sencillas y sistemas de progresión. El juego es accesible para todas las edades, con énfasis en decisiones estratégicas.
- Estilo de Juego: El juego combina narrativa y estrategia con elementos de RPG. Los jugadores gestionan recursos como salud y dinero mientras completan misiones y mejoran estadísticas.

REGLAS

1. El jugador elige un personaje (Luchador, Asesino, Tanque, Mago), cada uno con estadísticas iniciales únicas.
2. Se pueden realizar cinco acciones principales:
 - Entrenar para mejorar estadísticas.
 - Comprar objetos en la tienda.



- Completar misiones para ganar recompensas.
 - Revisar estadísticas actuales.
 - Salir del juego.
3. El objetivo es alcanzar 250 unidades de dinero para ganar o evitar que la salud llegue a 0, lo que resulta en derrota.
 4. Las misiones tienen requisitos mínimos de estadísticas, y no cumplirlos puede impedir el éxito.
 5. Los objetos en la tienda pueden mejorar estadísticas específicas, pero algunos también disminuyen otras.

MECÁNICAS Y JUGABILIDAD

1. **Elección del Personaje:** Al iniciar, el jugador selecciona uno de los personajes disponibles, como el Luchador, con estadísticas iniciales asignadas.
2. **Entrenamiento:** Los jugadores pueden gastar puntos de experiencia ganados para mejorar habilidades como Vitalidad, Fuerza, Agilidad y Percepción Mágica.
3. **Tienda:** Los jugadores pueden comprar objetos que modifican sus estadísticas y requieren dinero acumulado en misiones.
4. **Misiones:** Cada misión tiene descripciones y desafíos específicos, como proteger a mercaderes o participar en asedios. Que se puedan hacer depende de las estadísticas del jugador.
5. **Condiciones de Victoria:** Se gana al alcanzar el objetivo de dinero; la derrota ocurre si la salud llega a cero.

Interacción del Usuario:

El juego utiliza entradas de texto para navegar por menús y tomar decisiones. Mensajes claros y uso de colores en consola (como ANSI) mejoran la experiencia visual y destacan información clave, como la pérdida de salud o la ganancia de dinero.

ANÁLISIS DEL PROYECTO

CONTROL DE REQUISITOS

REQUISITOS FUNCIONALES

1º. Caso de Uso: Elección de Personaje

Título Menu Personaje

Descripción

Mediante la introducción de la inicial del personaje podrás elegir el personaje con el que jugarás.

Precondiciones Ninguna

Entrada La inicial del personaje

Salida Las estadísticas del personaje escogido

Boceto Menú en el cual tienes varias opciones

Estado Implementado

ELIJA SU PERSONAJE ESCRIBIENDO LA INICIAL (o 's' para salir)

|| LUCHADOR || ASESINO || TANQUE || MAGO || SALIR ||

2º. Caso de Uso: Acción del Personaje

Título Menú de Acciones

Descripción

Mediante la introducción del número indicado podrás realizar una de estas 5 acciones

Precondiciones Haber elegido previamente un personaje

Entrada El número de la acción a realizar

Salida La acción y sus propiedades

Boceto Menú en el cual tienes varias opciones y una breve explicación de cada una de ellas

Estado Implementado



3º. Caso de Uso: Entrenamiento

Título Campo de Entrenamiento

Descripción

Al dar a la opción de entrenar en el menú de acciones, ganarás experiencia la cual se gastará en mejorar tus estadísticas.

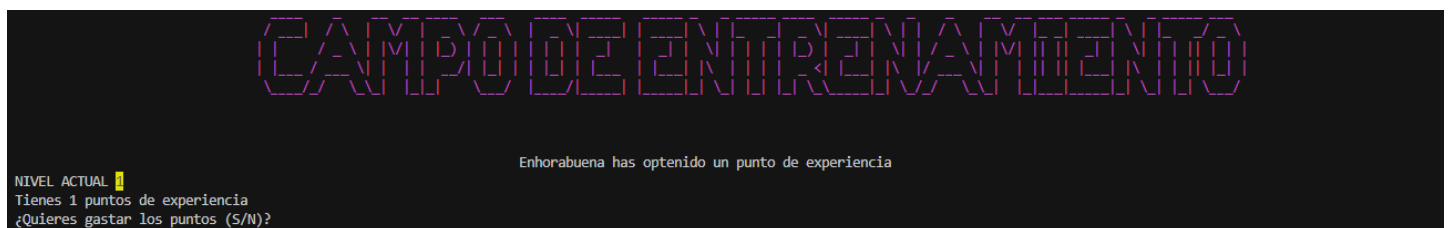
Precondiciones Haber escogido un personaje y la acción de entrenar

Entrada S/N en función de si quieres gastar la experiencia o no

Salida La estadística mejorada más la experiencia restante

Boceto Pregunta de SI o NO

Estado Implementado



4º. Caso de Uso: Misiones

Título Tablón Misiones

Descripción

Ganarás dinero a cambio de perder salud, pero necesitarás unas estadísticas necesarias propias de cada misión.

Precondiciones Tener las estadísticas necesarias

Entrada El número de la misión a realizar

Salida La misión y su descripción

Boceto Menú en el cual tienes varias opciones

Estado Implementado

TABLÓN DE MISIONES

1. Unirte a la leva local y ayudarles a tomar un castillo.
2. Unirte a la guarnición del castillo.
3. Acompañar y proteger a un mercader durante una ruta peligrosa.
4. Traer de vuelta al hijo del Duque de tu provincia.
5. Volver al Menú.

SALUD 5.0 DINERO 0

5º. Caso de Uso: Elección de Personaje

Título Tienda

Descripción

Puedes adquirir objetos para modificar estadísticas, requiriendo dinero acumulado de las misiones.

Precondiciones Tener Dinero para entrar y el suficiente para cada objeto

Entrada El número del objeto a comprar

Salida La modificación de estadísticas del objeto

Boceto Menú en el cual tienes varias opciones

Estado Implementado



5º. Caso de Uso: Enseñar Estadísticas

Título	Estadísticas
Descripción	Se muestran las estadísticas actuales del personaje
Precondiciones	Haber escogido un personaje
Entrada	Ninguna
Salida	Ninguna
Boceto	Título y texto que enseña una información
Estado	Implementado



REQUISITOS NO FUNCIONALES

1. Interfaz de Usuario (UX):

- Mensajes claros y uso de colores para mejorar la experiencia.
- Estado: Implementado.

2. Rendimiento:



- Respuesta eficiente a las entradas del usuario.
- Estado: Implementado.

3. **Portabilidad:**

- Compatible con sistemas Windows y Linux.
- Estado: Comprobado.

4. **Fiabilidad:**

- Validación adecuada de entradas del usuario mediante try catch y más.
- Estado: Implementado.

5. **Escalabilidad:**

- Preparado para añadir más personajes, misiones, mecánicas, objetos ...
- Estado: validada

ESTRUCTURA DEL PROYECTO

TheKingv2/: Carpeta principal del proyecto.

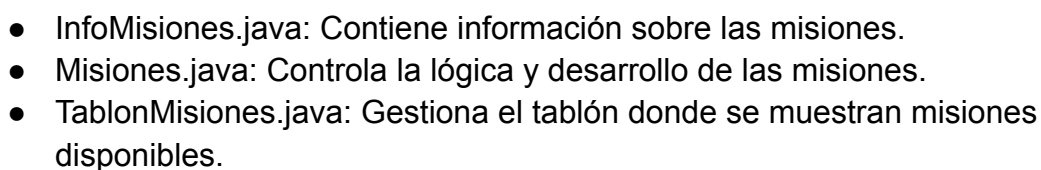
1. **README.md:** Archivo con una descripción del proyecto, instrucciones de instalación, ejecución y detalles adicionales.
2. **.vscode/:** Contiene configuraciones específicas del entorno de Visual Studio Code, como preferencias del editor y configuración del depurador.
 - **- settings.json:** Archivo de configuración del entorno en VS Code. En este proyecto no se han modificado, podría valorarse a futuro
3. **lib/:**
 - **Thekingv2.jar :** archivo .jar que se podrá ejecutar directamente desde la terminal pudiendolo enviar como forma de poder ejecutar tu programa sin tener el código fuente.
4. **src/:** Carpeta que contiene el código fuente.



- Clases
- Entrenamiento
- Entrenamiento.java
- Interfaz
- Menus.java
- Titulos.java
- Misiones
- InfoMisiones.java
- Misiones.java
- TablonMisiones.java
- Personajes
- AccionesPersonajes.java
- Asesino.java
- Luchador.java
- Mago.java
- Personajes.java
- Tanque.java
- Tienda
- ObjetoTienda.java
- Tienda.java
- Util
- Utilidades
- Videojuegov2.java**

DIAGRAMA MODULAR

Una serie de Métodos que se ejecutan en este orden menos los últimos tres(Que dependen de la elección del menú de acciones) para ejecutar el juego y poder continuar con el





5. src/clases/Personajes/

- AccionesPersonajes.java: Controla las acciones que los personajes pueden realizar.
- Asesino.java: Define las características del personaje "Asesino".
- Luchador.java: Define las características del personaje "Luchador".
- Mago.java: Define las características del personaje "Mago".
- Personaje.java: Clase base para todos los personajes.
- Tanque.java: Define las características del personaje "Tanque".

6. src/clases/Tienda/

- ObjetoTienda.java: Representa los objetos que se pueden comprar en la tienda.
- Tienda.java: Maneja la lógica de la tienda del juego.

7. src/util/

- Utilidades.java: Contiene funciones de utilidad generales para el juego.

Clases/Misiones/Misiones.java

- **estructuraMision(Misiones mision, Personaje personaje)**
 - **Objetivo:** Estructurar la misión, actualizar los atributos del personaje y mostrar los resultados en consola.
 - **Entradas:** Objeto de la clase Misiones con los datos de la misión actual y Objeto de la clase Personaje que realizará la misión.
 - **Salidas:** Ninguna.
- **getPagaMision()**
 - **Objetivo:** Retornar la paga de la misión.
 - **Entradas:** Ninguna.
 - **Salidas:** Cantidad de dinero que otorga la misión.
- **setPagaMision(int pagaMision)**
 - **Objetivo:** Actualizar la paga de la misión.



- **Entradas:** pagaMision: nuevo valor.
- **Salidas:** Ninguna.

- **getSaludPerdida()**

- **Objetivo:** Retornar la cantidad de salud perdida en la misión.
- **Entradas:** Ninguna.
- **Salidas:** Cantidad de salud que se pierde.

- **setSaludPerdida(double saludPerdida)**

- **Objetivo:** Actualizar la cantidad de salud perdida en la misión.
- **Entradas:** saludPerdida: nuevo valor.
- **Salidas:** Ninguna.

- **getDescripcion()**

- **Objetivo:** Retornar la descripción de la misión.
- **Entradas:** Ninguna.
- **Salidas:** Array de Strings con la descripción de la misión.

- **setDescripcion(String[] descripcion)**

- **Objetivo:** Actualizar la descripción de la misión.
- **Entradas:** descripcion: nuevo array de Strings con la descripción.
- **Salidas:** Ninguna.

Clases/Personajes/Personaje.java

- **enseñarEstadisticas()**

- **Objetivo:**Mostrar las estadísticas de cada personaje.
- **Entradas:** Ninguna
- **Salidas:**Ninguna.



- **introduccionPersonaje()**
 - **Objetivo:** Presentar al jugador y las estadísticas iniciales de su personaje elegido.
 - **Entradas:** Ninguna.
 - **Salidas:** Ninguna.
- **subirNivel()**
 - **Objetivo:** Verificar si el personaje ha subido de nivel y actualizarlo si es necesario.
 - **Entradas:** Ninguna.
 - **Salidas:** Ninguna.
- **condicionesVictoria()**
 - **Objetivo:** Comprobar si se cumplen las condiciones de victoria o de muerte y finalizar la partida si es necesario.
 - **Entradas:** Ninguna
 - **Salidas:** true si se cumple alguna condición de victoria o muerte, false en caso contrario.

Clases/Interfaz/Menus.java

- **menuPersonaje()**
 - **Objetivo:** Permitir al jugador seleccionar un personaje (Luchador, Asesino, Tanque, Mago) mediante un menú interactivo.
 - **Entradas:** Ninguna.
 - **Salidas:** La elección del usuario en forma de la inicial del personaje en minúsculas.
- **menuAcciones(Personaje personaje)**
 - **Objetivo:** Presentar al jugador las opciones principales del juego (entrenar, misiones, tienda, etc.).
 - **Entradas:** Objeto de la clase personaje para poder saber la salud y dinero actual del personaje.
 - **Salidas:** La opción de que acción hacer.



- **imprimirCuadroTexto(String[] texto)**
 - **Objetivo:** Encuadrar un texto para que se vea mejor en la consola
 - **Entradas:** Un array de String para poder encuadrarlo bien
 - **Salidas:** El texto encuadrado.

Clases/Personajes/Personaje.java,Asesino.java,Luchador.java...

- **introduccionLuchador/Asesino/Tanque/Mago()**
 - **Objetivo:** Enseña las estadísticas principales de cada personaje
 - **Entradas:** Se crea el método en la clase Personaje y se pasa mediante `super()` a las hijas, y añadiendo la estadística especial.
 - **Salidas:** Asignación de valores predeterminados de fuerza, vitalidad, agilidad y salud al personaje.
- **enseñarEstadisticasLuchador/Asesino/Tanque/Mago()**
 - **Objetivo:** Mostrar las estadísticas de cada personaje.
 - **Entradas:** el método `enseñarEstadísticas` de la clase Personaje se pasa mediante `super()` a las hijas, y añadiendo la estadística especial.
 - **Salidas:** Ninguna.

Clases/Entrenamiento/Entrenamiento.java

- **entrenamientoPersonaje(Personaje personaje)**
 - **Objetivo:** Permitir al jugador asignar puntos de experiencia para mejorar estadísticas específicas.
 - **Entradas:** Objeto de la clase personaje para poder usar sus métodos propios como los getters y setters.
 - **Salidas:** Actualización de la estadística elegida (fuerza, vitalidad, agilidad, percepción mágica).

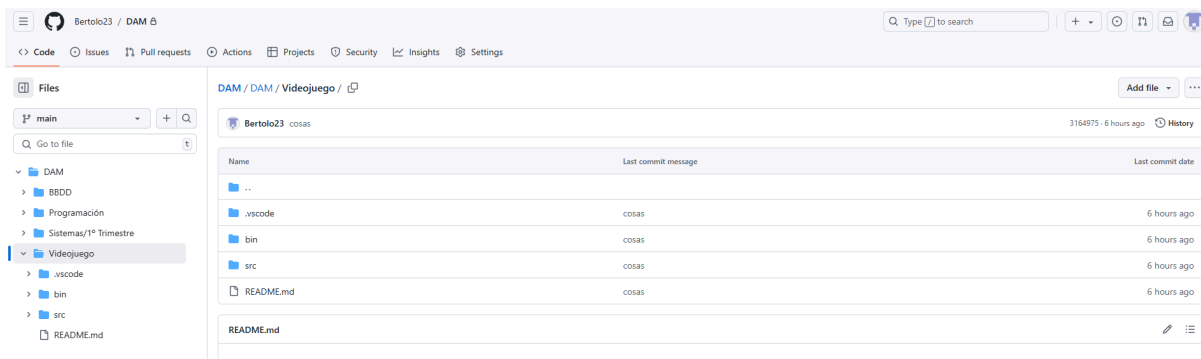
Clases/Misiones/TablonMisiones.java

- **tablonMisionesLuchador(Personaje personaje)**
 - **Objetivo:** Da todas las opciones de Misión que hay, tras terminarla mostrará la vida que se perdiera y dinero que se ganara por misión.

- **Entradas:** Objeto de la clase personaje para poder usar sus métodos propios como los getters y setters y también utilizo muchos métodos estáticos de otras clases;
- **Salidas:** Ninguna.

HERRAMIENTAS DE DESARROLLO

1. **JDK 21:** Para la compilación y ejecución del código.
2. **Visual Studio Code:** Como entorno de desarrollo integrado (IDE).
3. **GitHub:** Como almacenamiento en nube y capacidad de trabajar con el proyecto tanto en clase como en casa



4. **Chat GPT:** como apoyo para que me explicase cómo crear números aleatorios con `Math.random` y `.round`.

```
double saludPerdida = 0.25 + (Math.random() * (0.75 - 0.25));  
saludPerdida = Math.round(saludPerdida * 10) / 10.0;  
int pagaMision = (int)(Math.random() * (15 - 10 + 1)) + 10;
```

POSIBLES MEJORAS FUTURAS

1. **Sistema de Combate:** Agregar mecánicas de lucha contra enemigos.
2. **Interfaz Gráfica (GUI):** Migrar de consola a una experiencia visual más interactiva mediante Swing o JavaFX.



3. **Misiones Avanzadas:** Añadir mayor variedad de misiones y recompensas dinámicas.
4. **Mejores Objetos:** Añadir más objetos y la capacidad de equiparlos para darte la posibilidad de hacer misiones.
5. **Desbloqueo de Hechizos:** Al ganar percepción mágica poder desbloquear hechizos para por ejemplo curarte o poder ganar más dinero.
6. **Personalización:** Configurar colores y estilos antes de iniciar la partida.