

# THE KING





# ÍNDICE

<b>HOJA DE CONTROL DE VERSIONES.....</b>	<b>3</b>
<b>DESCRIPCIÓN.....</b>	<b>3</b>
TEMA Y OBJETIVO.....	3
CONTEXTO.....	3
AUDIENCIA Y ESTILO DE JUEGO.....	3
REGLAS.....	4
MECÁNICAS Y JUGABILIDAD.....	4
<b>ANÁLISIS DEL PROYECTO.....</b>	<b>5</b>
CONTROL DE REQUISITOS.....	5
REQUISITOS FUNCIONALES.....	5
REQUISITOS NO FUNCIONALES.....	8
ESTRUCTURA DEL PROYECTO.....	9
DIAGRAMA MODULAR.....	10
HERRAMIENTAS DE DESARROLLO.....	11
<b>POSIBLES MEJORAS FUTURAS.....</b>	<b>12</b>



## 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 funcinabilidad

## 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.

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.

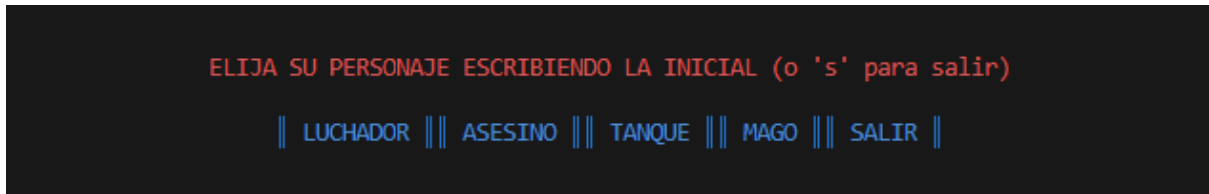
El juego utiliza entradas de texto para navegar por menús y tomar decisiones. Mensajes claros y uso de colores en consola (como ANSI) mejoran la experiencia visual y destacan información clave, como la pérdida de salud o la ganancia de dinero.

## REQUISITOS FUNCIONALES

Salida	Las estadísticas del personaje escogido
--------	---

**Boceto** Menú en el cual tienes varias opciones

**Estado** Implementado



## 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

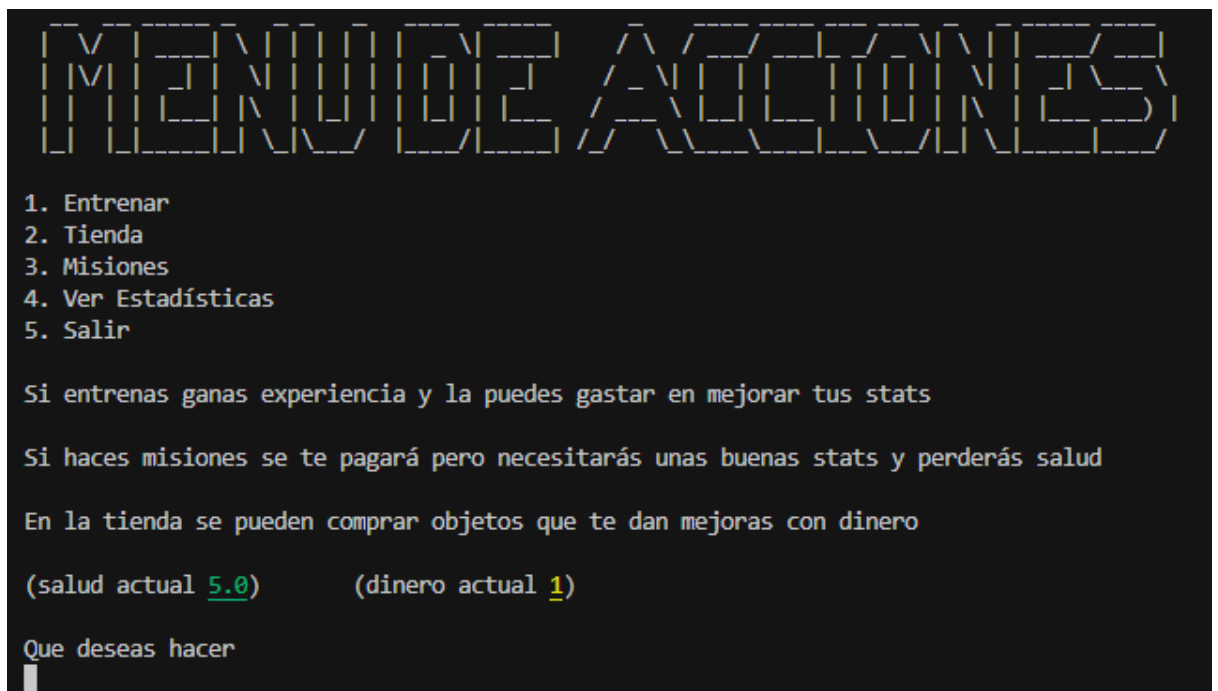
**Precondiciones** Haber elegido previamente un personaje

**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 cuál se gastará en mejorar tus estadísticas.

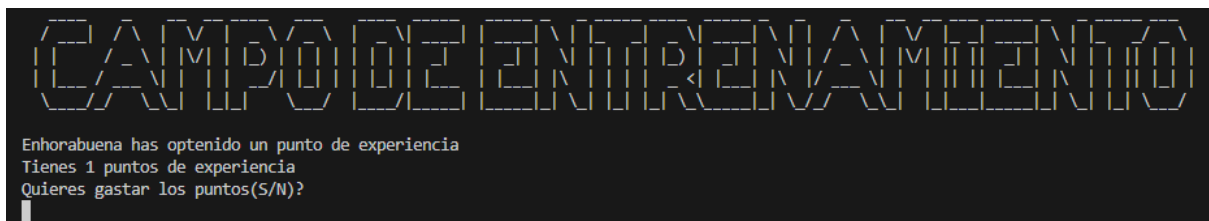
**Precondiciones** Haber escogido un personaje y la acción de entrenar

**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.

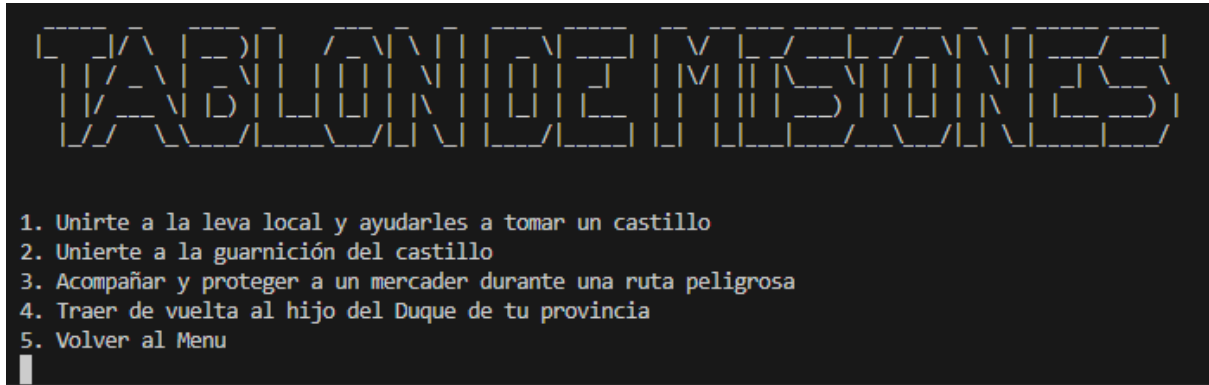
**Precondiciones** Tener las estadísticas necesarias

**Entrada** El número de la misión a realizar

**Salida** La misión y su descripción

**Boceto** Menú en el cuál tienes varias opciones

**Estado** Implementado



## 5º. Caso de Uso: Elección de Personaje

**Título** Tienda

### Descripción

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

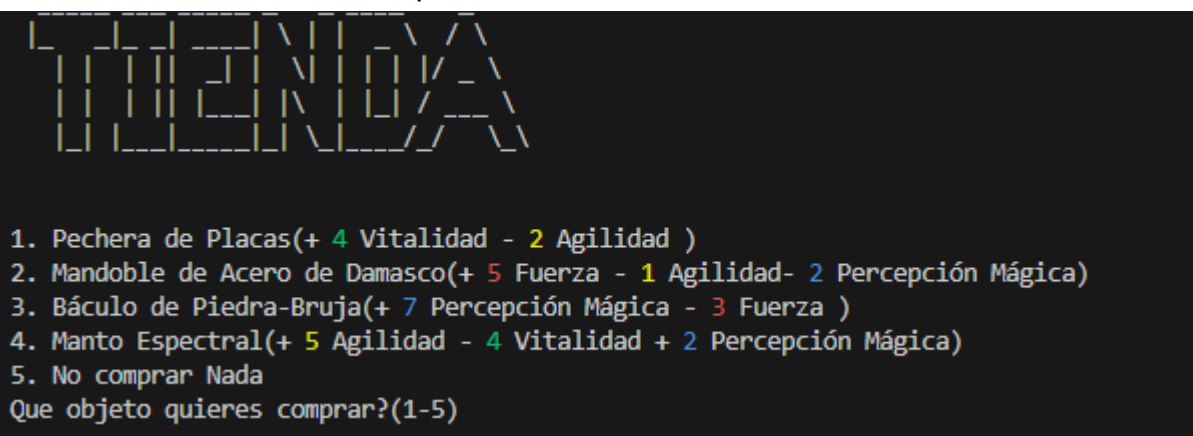
**Precondiciones** Tener Dinero para entrar y el suficiente para cada objeto

**Entrada** El número del objeto a comprar

**Salida** La modificación de estadísticas del objeto

**Boceto** Menú en el cual tienes varias opciones

**Estado** Implementado



## 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: validado

# 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/:** (Vacío en este caso) Es el lugar destinado a incluir bibliotecas externas o archivos JAR que el proyecto podría necesitar. Actualmente no se han necesitado
4. **src/:** Carpeta que contiene el código fuente.
  - **TheKingv2.java:** Versión inicial del código del juego.

## 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

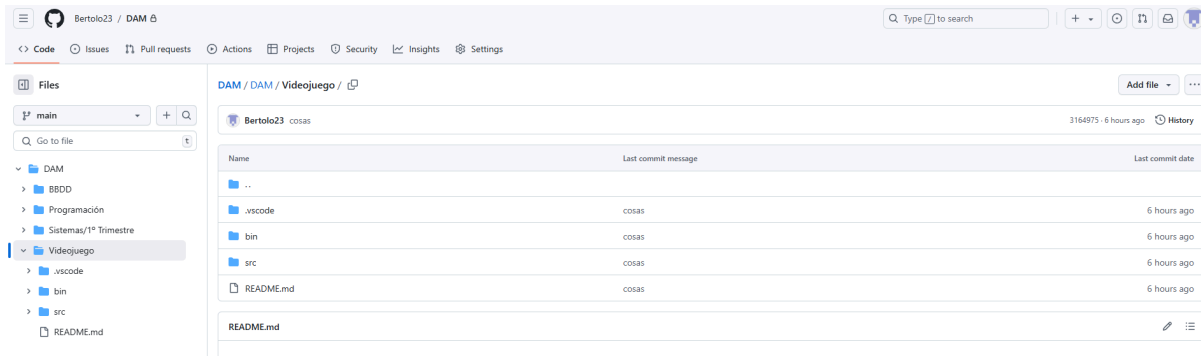
- **enseñarEstadisticas(Titulo titulo)**
  - **Objetivo:**Mostrar las estadísticas de cada personaje.
  - **Entradas:** El objeto titulo de la clase Titulo que servira para sacar por pantalla el titulo de las estadísticas.
  - **Salidas:**Ninguna.
- **menuPersonaje(BufferedReader LECTOR, String opcion)**
  - **Objetivo:** Permitir al jugador seleccionar un personaje (Luchador, Asesino, Tanque, Mago) mediante un menú interactivo.
  - **Entradas:** Un lector de entradas (BufferedReader) y una opción seleccionada por el usuario.
  - **Salidas:** El nombre del personaje seleccionado y configuración de su rol en el juego.
- **introduccionPersonaje()**
  - **Objetivo:** Presentar al jugador y las estadísticas iniciales de su personaje elegido.
  - **Entradas:** Ninguna.
  - **Salidas:** Ninguna.
- **introduccionLuchador()**
  - **Objetivo:**Enseña las estadísticas principales de la clase luchador
  - **Entradas:** el método de la clase madre introduccionPersonaje() mediante super.



- **Salidas:** Asignación de valores predeterminados de fuerza, vitalidad, agilidad y salud al personaje.
  
- **menuAcciones(BufferedReader lector, int eleccionAccion, Titulos titulos, Personaje personaje)**
  - **Objetivo:** Presentar al jugador las opciones principales del juego (entrenar, misiones, tienda, etc.).
  - **Entradas:** Un lector de entradas (BufferedReader), la opción seleccionada por el jugador, el objeto titulo y uno de personaje este último para poder saber la salud y dinero actual del personaje.
  - **Salidas:** Acción correspondiente según la opción elegida.
  
- **menuEntrenamiento(BufferedReader LECTOR, int opcionMejora)**
  - **Objetivo:** Permitir al jugador asignar puntos de experiencia para mejorar estadísticas específicas.
  - **Entradas:** Un lector de entradas (BufferedReader) y la opción seleccionada para mejorar una estadística.
  - **Salidas:** Actualización de la estadística elegida (fuerza, vitalidad, agilidad, percepción mágica).
  
- **imprimirCuadroTexto(String[] texto)**
  - **Objetivo:** Encuadrar un texto para que se vea mejor en la consola
  - **Entradas:** Un array the String para poder encuadrarlo bien
  - **Salidas:** El texto encuadrado.
  
- **tablonMisionesLuchador(BufferedReader lector, Titulos titulo, Menus menu, Luchador luchador, InfoMisiones info)**
  - **Objetivo:** Da todas las opciones de Misión que hay, más la vida que se perderá y dinero que se ganara por misión.
  - **Entradas:** Un lector de entradas (BufferedReader), el objeto titulo para imprimir el título del tablon, un objeto menu para poder enseñar las diferentes misiones al jugador, un objeto luchador para modificar la salud y dinero y el objeto info que te da los datos de cada misión.
  - **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. **Ampliación de Personajes:** Introducir más clases con habilidades únicas.
2. **Sistema de Combate:** Agregar mecánicas de lucha contra enemigos.
3. **Interfaz Gráfica (GUI):** Migrar de consola a una experiencia visual más interactiva mediante Swing o JavaFX.
4. **Misiones Avanzadas:** Añadir mayor variedad de misiones y recompensas dinámicas.



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