THE QUESTING KNIGHT



ÍNDICE

[**HOJA DE CONTROL DE VERSIONES 3**](#_heading=h.gjdgxs)

[**DESCRIPCIÓN 3**](#_heading=h.30j0zll)

[TEMA Y OBJETIVO 3](#_heading=h.1fob9te)

[CONTEXTO 4](#_heading=h.3znysh7)

[AUDIENCIA Y ESTILO DE JUEGO 4](#_heading=h.2et92p0)

[REGLAS 4](#_heading=h.tyjcwt)

[MECÁNICAS Y JUGABILIDAD 5](#_heading=h.3dy6vkm)

[**ANÁLISIS DEL PROYECTO 5**](#_heading=h.1t3h5sf)

[CONTROL DE REQUISITOS 5](#_heading=h.4d34og8)

[REQUISITOS FUNCIONALES 5](#_heading=h.2s8eyo1)

[REQUISITOS NO FUNCIONALES 9](#_heading=h.17dp8vu)

[ESTRUCTURA DEL PROYECTO 10](#_heading=h.3rdcrjn)

[DIAGRAMA MODULAR 11](#_heading=h.lnxbz9)

[Clases/Misiones/Misiones.java 12](#_heading=h.6sed54v6he9d)

[Clases/Personajes/Personaje.java 13](#_heading=h.iakgzgc49z5i)

[Clases/Interfaz/Menus.java 14](#_heading=h.w7vzgl47xfg7)

[Clases/Personajes/Personaje.java,Asesino.java,Luchador.java… 15](#_heading=h.55cnqfoskgwf)

[Clases/Entrenamiento/Entrenamiento.java 15](#_heading=h.zi6iaiz63aw1)

[Clases/Misiones/TablonMisiones.java 15](#_heading=h.uw3yue9ti7h6)

[HERRAMIENTAS DE DESARROLLO 16](#_heading=h.35nkun2)

[**POSIBLES MEJORAS FUTURAS 16**](#_heading=h.1ksv4uv)

# 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 |
| FINAL | 04/2025 | IBG | Implementación de Sistema de pelea, exportación de partidas a .txt y BBDD |

# 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

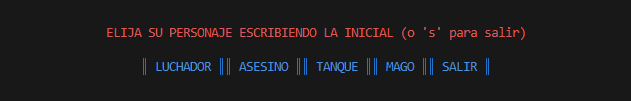
**Descripción**

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

**Entrada** La inicial del personaje

**Salida** Las estadísticas del personaje escogido

**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

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

**Salida** La acción y sus propiedades

**Estado** Implementado



**3º. Caso de Uso: Entrenamiento**

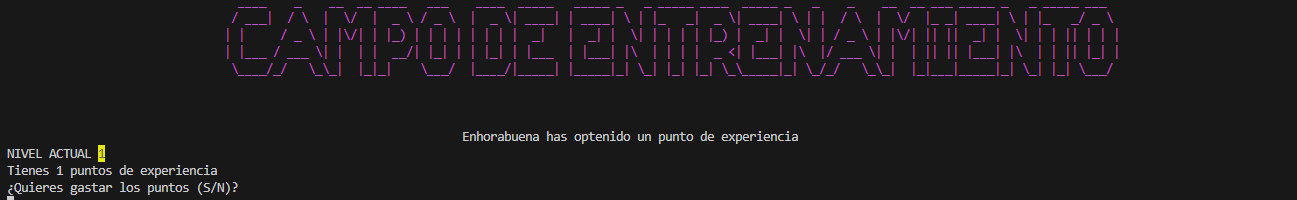
**Título**  Campo de Entrenamiento

**Descripción**

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

**Entrada** 1

**Salida** La estadística mejorada más la experiencia restante

**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.

**Entrada** 3

**Salida** La misión y su descripción

**Estado** Implementado



**5º. Caso de Uso: Compra de Objetos**

**Título**  Tienda

**Descripción**

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

**Entrada** 2

**Salida** La modificación de estadísticas del objeto

**Estado** Implementado



**6º. Caso de Uso: Compra de Objetos sin Dinero**

**Título**  Tienda

**Descripción** Te avisa de que sin dinero no puedes comprar nada

**Entrada** 2

**Salida** La modificación de estadísticas del objeto

**Estado** Implementado



**7º. Caso de Uso: Enseñar Estadísticas**

**Título**  Estadísticas

**Descripción** Se muestran las estadísticas actuales del personaje

**Entrada** 2

**Salida** Mensaje de que no tienes dinero

**Estado** Implementado



**8º. Caso de Uso: Historial de Partidas**

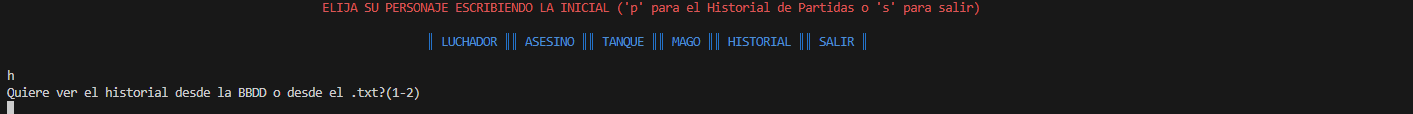
**Título**  Historial

**Descripción** Se muestran las partidas jugadas previamente desde un .txt o desde la base de datos

**Entrada** h

**Salida** las partidas jugadas previamente

**Estado** Implementado



**9º. Caso de Uso: Partidas desde la BBDD**

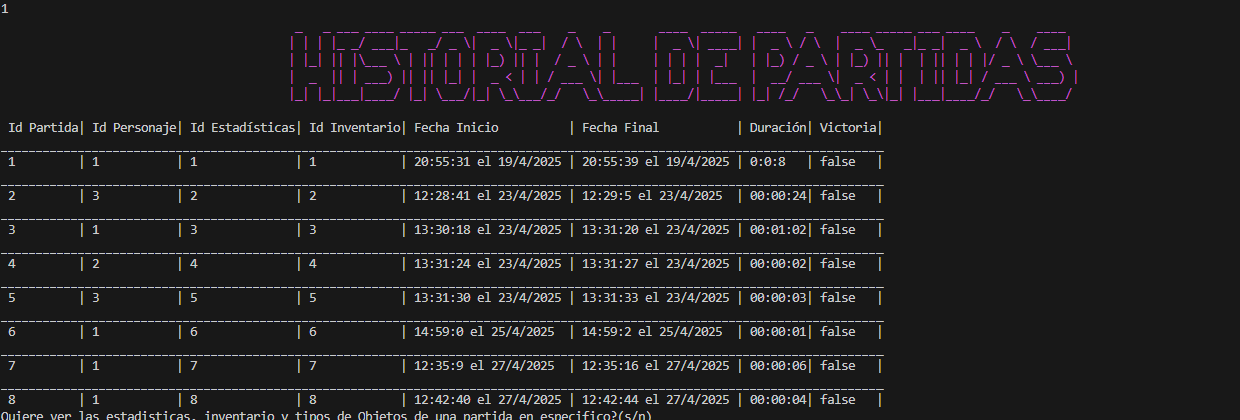
**Título**  Partidas BBDD

**Descripción** Se muestran las partidas jugadas previamente desde la base de datos con toda su información

**Entrada** 1

**Salida** las partidas jugadas previamente

**Estado** Implementado



**10º. Caso de Uso: Información complementaria de Partidas**

**Título**  Info Partidas

**Descripción** Se muestras las demas tablas dando una información complementaria a la de Partida

**Entrada** Si y id de la partida

**Salida** Info complementaria de las Partidas

**Estado** Implementado

### 

**11º. Caso de Uso: Partidas .txt**

**Título**  Info Partidas

**Descripción** Se muestras las sesiones de juego y las partidas jugadas

**Entrada** 2

**Salida** Info de las Partidas jugadas

**Estado** Implementado



**12º. Caso de Uso: Mostrar Inventario Vacío**

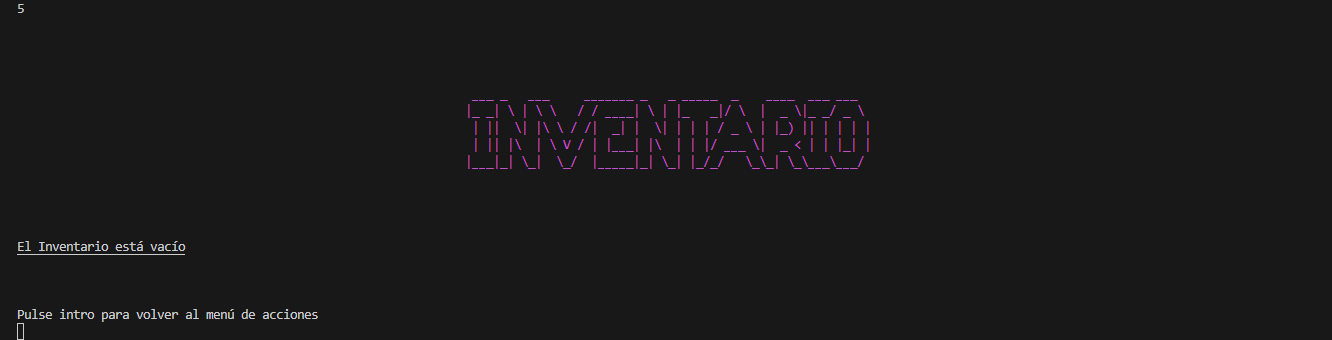
**Título**  Inventario Vacío

**Descripción** Se muestra el inventario del jugador que en este caso está vacío porque no ha comprado ningún arma

**Entrada** 5

**Salida** Mensaje Inventario vacío

**Estado** Implementado



**13º. Caso de Uso: Mostrar Inventario**

**Título**  Inventario

**Descripción** Se muestra el inventario, los objetos que ha comprado y tiene equipados

**Entrada** 5

**Salida** Tabla del inventario

**Estado** Implementado



**14º. Caso de Uso: Subir de Nivel**

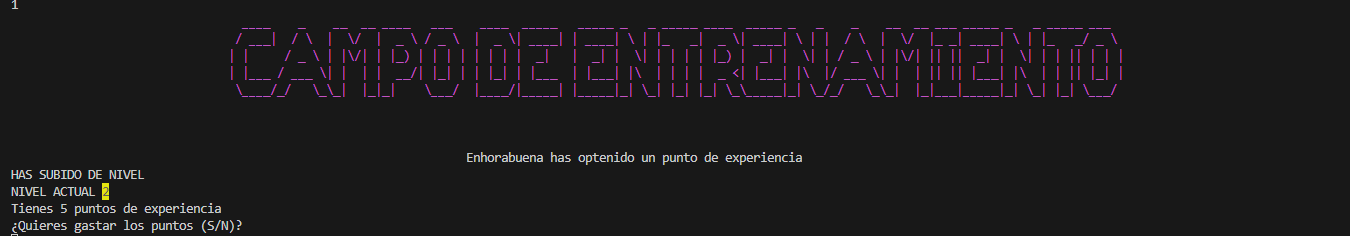
**Título**  Subir Nivel

**Descripción** si acumulas 5 puntos de experiencia y no los gastas subirás un nivel asi progresivamente

**Entrada** 1 y No a gastar los puntos

**Salida** Mensaje de subida de Nivel

**Estado** Implementado



**15º. Caso de Uso: No cumplir Requisitos de Misión**

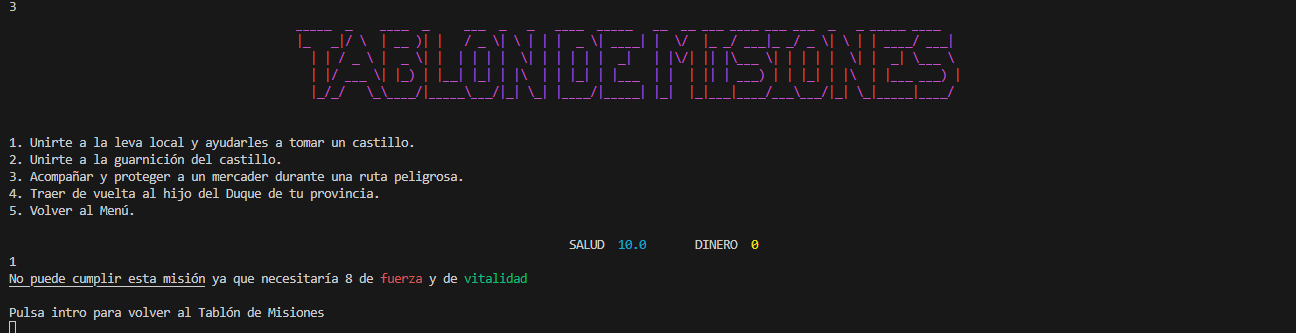
**Título**  No cumplir Requisitos

**Descripción** si no tienes las estadisticas y objetos necesarios no podras acceder a la misión

**Entrada** cualquier misiom

**Salida** Mensaje de falta de requisitos

**Estado** Implementado



**16º. Caso de Uso: Lucha en Mision**

**Título**  Sistema de Pelea

**Descripción** cuando empiezas una misión te llevara a un sitema de lucha en que elegiras 1(atacar) o 2(Defenderte)

**Entrada** c

**Salida** Sitema de pelea

**Estado** Implementado

### 

**17º. Caso de Uso: Atacar**

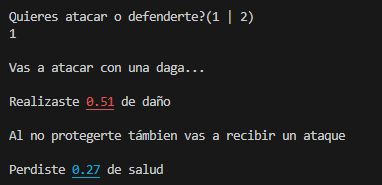
**Título**  Atacar

**Descripción** atacaras, si tienes la espada equipada realizaras más daño y también escala en función de la fuerza del personaje y disminuye con la vitalidad del enemigo

**Entrada** 1

**Salida** Mensajes de realización de daño y perdida de salud

**Estado** Implementado



**18º. Caso de Uso: Defenderte**

**Título**  Defenderte de ataque

**Descripción** te defenderas, si tienes el escudo equipado tendrás más posibilidades de parar el ataque enemigo y también escala en función de la agilidad del personaje y disminuye con la agilidad del enemigo

**Entrada** s

**Salida** Mensajes porcentaje de protegerte y o perdida de salud o exito a la hora de protegerte

### **Estado** Implementado

**19º. Caso de Uso: Comprar Objeto sin tener dinero suficiente**

**Título**  No tener dinero suficiente

**Descripción** intentar comprar un objeto y no tener dinero suficiente

**Entrada** s

**Salida** Mensajes de falta de dinero

**Estado** Implementado

### 

### 

### REQUISITOS NO FUNCIONALES

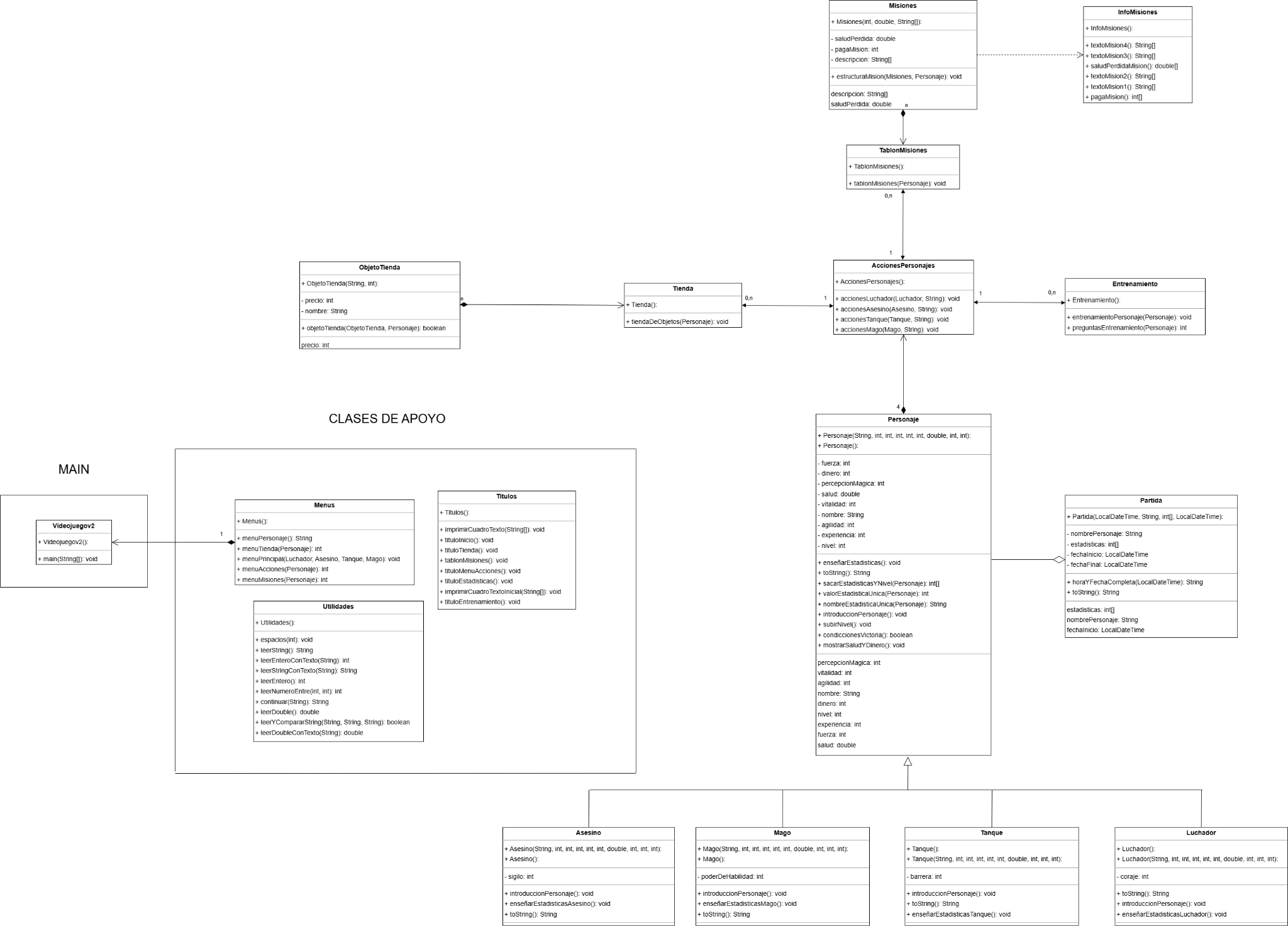
1. **Interfaz de Usuario (UX)**:
   * Mensajes claros y uso de colores para mejorar la experiencia.
   * Estado: Implementado.
2. **Rendimiento**:
   * Respuesta eficiente a las entradas del usuario.
   * Estado: Implementado.
3. **Portabilidad**:
   * Compatible con sistemas Windows y Linux.
   * Estado: 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

## 

## 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



**src/** **Videojuegov2.java** → Archivo principal que ejecuta el videojuego.

* iniciarJuego() : void
* terminarJuego() : void

**src/bbdd/** **ConexionBBDD.java** → Gestiona la conexión con la base de datos.

* conectar() : void
* desconectar() : void

**CRUD.java** → Maneja operaciones de Crear, Leer, Actualizar y Borrar en la base de datos.

* insertar(Object obj) : boolean
* leer(int id) : Object

**src/clases/entrenamiento/** **Entrenamiento.java** → Lógica de entrenamiento de personajes.

* entrenar(Jugador jugador) : void
* mejorarHabilidad(Jugador jugador) : void

**src/clases/estadísticas/** **GestionEstadisticas.java** → Maneja las estadísticas de personajes o partidas.

* calcularEstadisticas(Personaje personaje) : void
* mostrarEstadisticas() : String

**src/clases/interfaz/** **Menus.java** → Menús interactivos del juego.

* mostrarMenuPrincipal() : void
* mostrarMenuOpciones() : void

**Titulos.java** → Títulos y encabezados del juego.

* mostrarTituloJuego() : void

**src/clases/misiones/** **InfoMisiones.java** → Información sobre las misiones.

* obtenerInfoMision(int id) : String

**Misiones.java** → Lógica y desarrollo de las misiones.

* iniciarMision(Jugador jugador) : boolean

**Pelea.java** → Control de los combates dentro de las misiones.

* iniciarPelea(Jugador jugador, Enemigo enemigo) : void

**TablonMisiones.java** → Tablón de misiones disponibles.

* mostrarMisionesDisponibles() : void

**src/clases/partida/** **Partida.java** → Gestión del avance de una partida.

* guardarPartida(String nombreArchivo) : void
* cargarPartida(String nombreArchivo) : Partida

**src/clases/personajes/** **AccionesPersonajes.java** → Acciones que pueden realizar los personajes.

* atacar(Personaje objetivo) : void
* defender() : void

**Asesino.java** → Personaje tipo asesino.

**CreacionPersonajes.java** → Lógica de creación de personajes.

* crearNuevoPersonaje(String tipo) : Personaje

**Enemigo.java** → Representa enemigos en el juego.

**Luchador.java** → Personaje tipo luchador.

**Mago.java** → Personaje tipo mago.

**Personaje.java** → Clase base de todos los personajes.

* subirNivel() : void

**Tanque.java** → Personaje tipo tanque.

**src/clases/tienda/** **ObjetoTienda.java** → Representa objetos disponibles en la tienda.

* getPrecio() : int
* aplicarEfecto(Jugador jugador) : void

**Tienda.java** → Gestión de compras de objetos y mejoras.

* mostrarCatalogo() : void
* comprarObjeto(int idObjeto) : boolean

**src/ficheros/** **GestionFicheros.java** → Manejo de archivos (guardar/cargar partidas, etc).

* guardarDatos(String ruta) : void
* cargarDatos(String ruta) : void

**src/util/** **Utilidades.java** → Funciones de apoyo y utilidades generales.

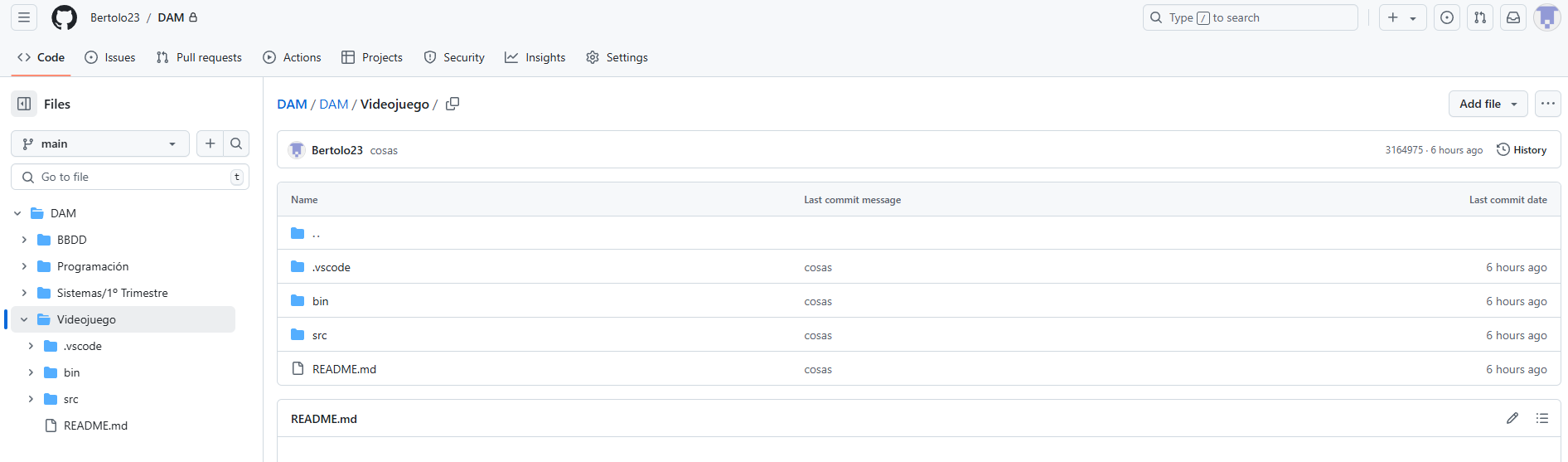
* generarNumeroAleatorio(int min, int max) : int
* formatearTexto(String texto) : String

# 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.



# POSIBLES MEJORAS FUTURAS

1. **Interfaz Gráfica (GUI)**: Migrar de consola a una experiencia visual más interactiva mediante Swing o JavaFX.
2. **Misiones Avanzadas**: Añadir mayor variedad de misiones y recompensas dinámicas.
3. **Mejores Objetos**: Añadir más objetos y la capacidad de equiparlos para darte la posibilidad de hacer misiones.
4. **Desbloqueo** **de Hechizos**: Al ganar percepción mágica poder desbloquear hechizos para por ejemplo curarte o poder ganar más dinero.
5. **Personalización**: Configurar colores y estilos antes de iniciar la partida.