{Ejercicio 4. Modelación con herencia}

# Análisis del Sistema (30 puntos)

## Requisitos funcionales del sistema (5 pts)

Escribir los puntos exactos que debe cumplir tu sistema.

Ejemplo: “El usuario puede crear una nueva solicitud de boletos”, etc.

- Se debe mantener un control de turnos y rondas.

- Se debe mantener un control de atributos de todos los combatientes.

- Se debe ser capaz de diferenciar entre roles para los jugadores y tipos monstruos para los enemigos.

- Se deben poder cambiar los atributos mediante objetos y ataques.

- Se debe poder escoger el enemigo que se quiere atacar.

## Clases necesarias y su propósito (5 pts)

|  |  |
| --- | --- |
| Clase | Propósito |
| Main | Clase de instancia principal. |
| Consola | Interacción con el usuario. |
| Game | Clase que contendrá la lógica y los elementos del juego |
| Combatant | Clase padre de jugador y enemigo que contendrá la información de atributos generales. |
| Player | Clase padre que contará con información general del jugador. |
| Warrior | Clase hija de jugador con información del rol guerrero. |
| Explorer | Clase hija de jugador con información del rol explorador. |
| Enemy | Clase padre que contará con información general de los enemigos. |
| Goblin | Clase hija de enemigo con general información del duende. |
| Skeleton | Clase hija de enemigo con general información del esqueleto. |
| GoblinNormal | Clase hija de Goblin con información de un goblin con estadísticas normales. |
| GoblinBoss | Clase hija de Goblin con información de un goblin con estadísticas mejoradas. |
| SkeletonNormal | Clase hija de Skeleton con información de un esqueleto con estadísticas normales. |
| SkeletonBoss | Clase hija de Skeleton con información de un esqueleto con estadísticas mejoradas. |
| Item | Clase con información de cantidad, espacio que ocupa y efectos de ítems. |
| Bag | Clase con información de los ítems de cada jugador. |
| Attack | Clase con información y efectos de los ataques. |
| Ability | Clase con información de efecto y disponibilidad de las habilidades. |
| Shop | Clase con información de todos los objetos. Se usará para asignar objetos al inicio de la partida. |

* 1. **Atributos de cada clase (10 pts)**

Clase:

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo de dato | Visibilidad | Propósito |
|  |  |  |  |

Clase:

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo de dato | Visibilidad | Propósito |
|  |  |  |  |

# Agregar las necesarias

## Métodos de cada clase (10 pts)

Clase:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Método | Parámetros : Tipo de dato | Visibilidad | Propósito |
|  |  |  |  |  |

Clase:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Método | Parámetros : Tipo de dato | Visibilidad | Propósito |
|  |  |  |  |  |

# Agregar las necesarias

1. **Diseño: Diagrama de Clases (30 puntos)**
   * Asegúrate de mostrar atributos y métodos con visibilidad (+, -).
   * Indica relaciones entre clases (asociación, agregación, etc.).
   * Incluye el driver program (Main).

Diagrama de clases aquí o adjunto en un archivo aparte.

# Programa (40 puntos)

En cada archivo `.java`, asegurarse de incluir:

* + Las clases necesarias.
  + Uso adecuado de objetos.

Menú que debe implementar el driver program:

* + 1. Nuevo comprador
    2. Nueva solicitud de boletos
    3. …
    4. …
    5. Salir

GitHub: colocar aquí la URL: <https://github.com/Hola2212/Ejercicio4_Modelacion_con_herencia.git>

# Checklist antes de entregar

⬛✓⬛ Está claro el análisis?

⬛✓⬛ El diagrama tiene los elementos UML correctamente?

⬛✓ Subiste tu código a GitHub con todo lo necesario?