Proyecto Final Computación

# Proyecto de Programación: Juego de Rol por Turnos en Python

En este proyecto, deberán desarrollar un juego de rol por turnos (RPG) que se ejecutará en la consola. Los jugadores controlarán un grupo de héroes que deberán enfrentarse a enemigos generados aleatoriamente en diversas mazmorras. El objetivo del juego es derrotar a todos los enemigos, mejorar las habilidades de los personajes a lo largo de las batallas, y avanzar a través de una serie de mazmorras hasta llegar al jefe final.

## Requerimientos mínimos del Proyecto:

1. **Generación de Personajes:**

* El programa debe generar 3 personajes principales, cada uno con atributos específicos como salud, fuerza, defensa, y nivel.
* Cada personaje puede pertenecer a una clase específica (por ejemplo, Guerrero, Mago, Arquero) con atributos base diferentes.

1. **Generación de Enemigos:**

* Los enemigos deben ser generados de manera aleatoria cada vez que comienza una nueva batalla.
* Los enemigos también deben tener atributos como salud, fuerza, y defensa que varían de manera aleatoria.

1. **Sistema de Combate por Turnos:**

* Los jugadores deben seleccionar qué personaje atacará y a qué enemigo se enfrentará en cada turno.
* Después de cada turno de ataque del jugador, los enemigos deben tener su turno de ataque (automático).

1. **Sistema de Nivel y Experiencia:**

* Todos los personajes que sobrevivan a una batalla recibirán puntos de experiencia.
* Al alcanzar cierta cantidad de experiencia, los personajes subirán de nivel, lo que mejorará sus atributos.

1. **Sistema de Ítems:**

* Implementar un sistema de ítems que los enemigos pueden dejar caer tras ser derrotados.
* Crear una lista de al menos 5 ítems, donde cada uno pueda mejorar los atributos de los personajes o restaurar salud.
* Algunos ítems pueden tener un efecto permanente en los atributos de los personajes, mientras que otros pueden ser consumibles.

1. **Avance por Mazmorras:**

* El juego debe incluir varias mazmorras (al menos 3), cada una con su propia serie de enemigos y desafíos.
* Los jugadores deben avanzar a través de las mazmorras, enfrentándose a enemigos y resolviendo posibles acertijos o desafíos.
* Al final de cada mazmorra, los jugadores deben enfrentarse a un jefe, un enemigo más poderoso que los normales.

1. **Narrativa del Juego (opcional):**

* Incluir una narrativa básica que explique el contexto de la aventura y motive a los jugadores a seguir avanzando.
* Cada mazmorra puede tener una historia o misión específica que los héroes deben completar.
* La narrativa debe ser presentada en texto durante el juego, antes y después de las batallas o al avanzar de una mazmorra a otra.

1. **Condiciones de Victoria y Derrota:**

* El jugador debe completar todas las mazmorras y derrotar al jefe final para ganar el juego.
* El juego solo puede ser completado si al menos uno de los héroes sigue vivo al final de la última batalla.
* Si todos los héroes son derrotados en cualquier momento, el juego se reinicia desde el principio.

## Herramientas y Requerimientos Adicionales:

1. **Uso de Git:**

* Deben utilizar Git para gestionar el control de versiones de su proyecto.
* Se debe crear un repositorio en una plataforma como GitHub, donde se realizarán los commits y se documentará el progreso del proyecto.

1. **Diagramas UML:**

* Deben entregar un diagrama de clases UML que represente la estructura del código.
* El diagrama debe incluir las clases principales del juego, sus atributos, métodos, y las relaciones entre ellas.

1. **Entregables:**

* Código fuente: Deben entregar el código del juego en Python, asegurándose de que esté bien documentado y organizado.
* Repositorio Git: Deben proporcionar un enlace al repositorio Git donde se ha gestionado el proyecto.
* Informe: Se debe incluir un breve informe explicando el diseño del juego, la estructura del código y las decisiones técnicas tomadas.
* Diagramas UML: Los diagramas deben reflejar la estructura y diseño del código de manera clara y precisa.

1. **Criterios de Evaluación:**

* **Funcionalidad:** El programa cumple con todos los requisitos establecidos.
* **Calidad del código:** Se evaluará la claridad, la organización y la documentación del código.
* **Gestión del Proyecto con Git:** Se evaluará el uso adecuado de Git, incluyendo la frecuencia y calidad de los commits, así como el manejo de ramas y merges si es aplicable.
* **Diagramas UML:** Se evaluará la precisión y claridad del diagrama de clases UML entregado.
* **Narrativa y Diseño de Mazmorras (suma puntos):** Se valorará la creatividad y coherencia en la narrativa del juego y el diseño de las mazmorras.
* **Robustez:** El código debe manejar adecuadamente los errores y las entradas no válidas