### **Documentación del Proyecto "Mortal Kombat"**

Este documento ofrece una visión general del proyecto "Mortal Kombat", un juego de combate interactivo en consola desarrollado en Python, utilizando microservicios para la autenticación y la gestión de partidas.

#### **Descripción General del Proyecto**

"Mortal Kombat" es un juego de lucha interactivo para consola en el cual los jugadores pueden elegir entre cinco luchadores distintos, cada uno con habilidades únicas. Para acceder al juego y mantener un historial de combates y resultados, los jugadores deberán registrarse y autenticarse.

#### **Tecnologías y Librerías Recomendadas**

* **Python**: Todo el backend se desarrollará en Python.
* **Microservicios**:
  + **Servicio de Gestión de Combates**: Gestiona el registro y el login de los usuarios, así como el inicio de los combates.
  + **Servicio de Resultado de Combates**: Registra los resultados de los combates y la evolución de los personajes según vayan acumulando resultados.
* **RabbitMQ**: Se utiliza para la comunicación entre los microservicios, permitiendo el intercambio de mensajes de manera asincrónica.
* **SQLite o PSQL**: Bases de datos ligeras que se pueden usar para mantener un registro de los usuarios y los historiales de combate sin necesidad de un servidor de base de datos dedicado.

#### **Estructura de los Microservicios**

**Servicio de Gestión de Combates** (Terminal):

* **Register**: Permite a los usuarios registrarse.
* **Login**: Autentica a los usuarios.
* **start combat**: Inicia una nueva sesión de combate.
* **my heroes**: Conecta al queue de héroes para obtener esta información.
* **My history**: Muestra el resultado de los combates.

**Servicio de Resultado de Combates** (Queues):

* **heroes**: Devuelve las estadísticas y niveles actuales de tus héroes.
* **history**: Devuelve el historial de combates de un usuario.

#### 

#### 

#### **Reglas de Combate**

* **Inicio del Combate**: Cada combate comienza con ambos luchadores al 100% de salud. Los combates son por turnos, donde cada jugador tiene un tiempo limitado para elegir una acción.
* **Fin del Combate**: Un combate termina cuando la salud de uno de los luchadores llega a 0%. El jugador con salud restante es declarado ganador.
* **Probabilidades en Combate**: Cada movimiento tiene una probabilidad de acierto basada en las habilidades del luchador y las condiciones del combate. Los jugadores pueden esquivar o bloquear ataques basados en probabilidades determinadas por sus estadísticas y movimientos especiales.

#### **Personajes y Estadísticas**

**Scorpion**:

* **Salud**: 100
* **Fuerza**: 80
* **Agilidad**: 70
* **Movimientos**:
  + **Ataque Rápido**: 15 de daño, 70% de probabilidad de acierto.
  + **Lanza Infernal**: 30 de daño, 50% de probabilidad de acierto.
* **Movimiento Final (Fatality)**: 50 de daño, activación cuando el oponente tiene menos del 20% de salud.
* **Defensa**:
  + **Esquive Automático**: 10% de probabilidad.
  + **Escudo de Fuego**: Reduce el daño en un 25% cuando la salud está por debajo del 30%.

**Sub-Zero**:

* **Salud**: 100
* **Fuerza**: 75
* **Agilidad**: 75
* **Movimientos**:
  + **Puño de Hielo**: 20 de daño, 65% de probabilidad de acierto.
  + **Granizada Mortal**: 25 de daño, 55% de probabilidad de acierto.
* **Movimiento Final (Fatality)**: 45 de daño, activación cuando el oponente tiene menos del 25% de salud.
* **Defensa**:
  + **Esquive de Hielo**: 15% de probabilidad.
  + **Muro de Hielo**: Reduce el daño en un 30% cuando la salud está por debajo del 25%.

**Kitana**:

* **Salud**: 90
* **Fuerza**: 70
* **Agilidad**: 85
* **Movimientos**:
  + **Golpe de Abanico**: 18 de daño, 75% de probabilidad de acierto.
  + **Tornado de Viento**: 28 de daño, 45% de probabilidad de acierto.
* **Movimiento Final (Fatality)**: 55 de daño, activación cuando el oponente tiene menos del 15% de salud.
* **Defensa**:
  + **Esquive Aéreo**: 20% de probabilidad.
  + **Barrera de Viento**: Reduce el daño en un 20% cuando la salud está por debajo del 40%.

**Raiden**:

* **Salud**: 110
* **Fuerza**: 85
* **Agilidad**: 60
* **Movimientos**:
  + **Rayo Eléctrico**: 22 de daño, 60% de probabilidad de acierto.
  + **Tormenta Fulminante**: 35 de daño, 40% de probabilidad de acierto.
* **Movimiento Final (Fatality)**: 50 de daño, activación cuando el oponente tiene menos del 20% de salud.
* **Defensa**:
  + **Esquive Relámpago**: 5% de probabilidad.
  + **Campo Eléctrico**: Reduce el daño en un 35% cuando la salud está por debajo del 20%.

**Jade**:

* **Salud**: 95
* **Fuerza**: 65
* **Agilidad**: 90
* **Movimientos**:
  + **Golpe de Bastón**: 16 de daño, 80% de probabilidad de acierto.
  + **Lluvia de Shuriken**: 26 de daño, 60% de probabilidad de acierto.
* **Movimiento Final (Fatality)**: 48 de daño, activación cuando el oponente tiene menos del 30% de salud.
* **Defensa**:
  + **Esquive de Sombra**: 25% de probabilidad.
  + **Escudo de Jade**: Reduce el daño en un 15% cuando la salud está por debajo del 35%.

### **Influencia de las Estadísticas en el Juego**

**Fuerza y Agilidad de los Personajes**

Cada personaje en "Mortal Kombat" tiene dos estadísticas clave que influyen directamente en su rendimiento en combate: fuerza y agilidad. Estas estadísticas afectan tanto la capacidad de ataque como de defensa de los personajes de las siguientes maneras:

* **Fuerza**:
  + **Bonus de Daño de Ataque**: La fuerza de un personaje influye directamente en el daño que puede infligir durante un ataque. Cada punto de fuerza añade un bonus del 7% al daño de ataque base. Esto significa que un personaje con una alta fuerza será capaz de realizar ataques más devastadores.
* Ejemplo:
  + Si un personaje tiene una fuerza de 70 y un ataque base causa 20 puntos de daño, el daño total sería 20+(0.07×70)=24.9 puntos de daño.
* **Agilidad**:
  + **Probabilidad de Esquivar Ataques**: La agilidad de un personaje mejora su capacidad para esquivar ataques. Cada punto en agilidad aumenta la probabilidad de esquivar un ataque en un 4%. Un personaje ágil puede esquivar ataques más frecuentemente, lo que le permite evitar daños y tener una mejor oportunidad de contraatacar.
* Ejemplo:
  + Si un personaje tiene una agilidad de 50, y tiene 25% de probabilidad base de esquivar el ataque este aumentaría a 25% + (50\*0.08)% = 29% de probabilidad de esquivar el ataque

**Nota**: Recordar que la probabilidad de esquivar el ataque se da en cada turno, es una habilidad pasiva.

#### **Sistema de Niveles y Progresión del Personaje**

**Ganancia de Niveles:**

* Cada vez que un personaje gana un combate, subirá de nivel.
* Al subir de nivel, el jugador recibe un punto de estadística que puede asignar a fuerza, agilidad o salud.

**Asignación de Puntos de Estadística:**

* **Menú de Terminal**: Tras ganar un combate y subir de nivel, se presentará automáticamente un menú en la terminal que permite al jugador asignar el punto de estadística ganado.
* **Opciones de Asignación**:
  + **Fuerza**: Aumentar la fuerza mejora el daño base de los ataques del personaje.
  + **Agilidad**: Aumentar la agilidad mejora la probabilidad de esquivar ataques.
  + **Salud**: Aumentar la salud eleva los puntos de vida máximos del personaje.

#### 

#### **Implementación del Menú de Asignación de Puntos**

Cuando un personaje sube de nivel, el sistema deberá:

1. Notificar al jugador del aumento de nivel y mostrar el número total de puntos de estadística disponibles.
2. Presentar un menú en la terminal para la asignación de puntos.
3. Permitir al jugador asignar puntos y confirmar su selección.
4. Commits inter-diarios.

Sobre el puntaje de la evaluación:

* Debes tener unitTest.
* Se esperan algunos patrones de diseño y el uso de **SOLID**.
* Deberás crear una documentación técnica:
  + Manual para correr el software
  + Diagrama de Base de datos
* Eres libre de usar las librerías que mejor te permitan realizar este trabajo.

Puntos Extras Técnicos:

* Levantas todo o parcialmente con Docker Compose
* Al ingresar a la base de datos de usuario para saber su usuario y password estos deben estar encriptados.