# **Segundo Parcial 1c 2025 - Ingeniería de Software 1 - FIUBA**

Nuestro cliente quiere desarrollar un sistema de combate para un juego de rol. En este sistema, diferentes tipos de combatientes pueden realizar acciones durante el combate, pero las acciones disponibles y sus efectos dependen tanto del tipo de combatiente como del estado actual del combatiente. El sistema debe modelar diferentes tipos de acciones que un combatiente puede realizar:

* **Ataque físico**: al realizar esta acción sobre otro combatiente le inflige 25 puntos de daño físico. Es decir que quita 25 hit points (HP)
* **Hechizo de daño**: al realizar esta acción sobre otro combatiente le inflige 30 puntos de daño mágico.
* **Curación**: al realizar esta acción sobre otro combatiente le restaura 20 puntos de vida al objetivo
* **Defensa**: al realizar esta acción el combatiente se pone en postura defensiva, por lo que luego al recibir un ataque físico o un hechizo recibe 50% menos daño (luego vuelve a su estado normal)

Los combatientes pueden ser de diferentes tipos:

* **Guerrero**: Puede atacar físicamente y ponerse en una postura defensiva. No puede usar magia (ni hechizos ni curación). Comienza con 120 HP.
* **Mago**: Puede lanzar hechizos de daño y curación. No puede hacer ataques físicos ni defenderse. Comienza con 80 HP.
* **Paladín**: combatiente híbrido, puede atacar físicamente, defenderse y usar curación. No puede usar hechizos de daño. Comienza con 100 HP.

## **Interacciones Complejas**

Las acciones no siempre tienen el mismo efecto. Depende de varios factores:

### **Resistencias y Vulnerabilidades**

Diferentes tipos de combatientes tienen resistencias y vulnerabilidades específicas:

* **Mago vs Daño Físico**: los magos son vulnerables a ataques físicos, reciben 20% más daño físico
* **Guerrero vs Daño Mágico**: los guerreros son resistentes a la magia, reciben 30% menos daño de hechizos

### **Efectos Secundarios de las Acciones**

Algunas acciones pueden tener efectos adicionales además de su efecto principal:

* **Hechizo de Daño**: tiene 30% de probabilidad de envenenar al objetivo. Un combatiente envenenado pierde 2 puntos de vida al final de cada acción realizada.
* **Curación del Paladín**: tiene 40% de probabilidad de bendecir al objetivo. Todas las curaciones que realiza o recibe un combatiente bendecido son 50% más efectivas.

**Nota**: Esta primera versión del sistema intencionalmente no contempla ciertas interacciones entre estados (como restricciones para defenderse o incompatibilidades entre estados) que en un proyecto real deberían implementarse. Todas las bendiciones y envenenamientos duran para siempre y no se pueden remover. También se ignoran casos especiales donde un combatiente realiza acciones sobre sí mismo.

### **Ejemplos de Interacciones:**

* Un Guerrero que ataca físicamente a un Mago causa +20% de daño (el mago es vulnerable a ataques físicos)
* Un Mago que lanza hechizo de daño a un Guerrero causa -30% de daño (el guerrero resiste magia)
* Una curación realizada por o hacia un combatiente Bendecido se incrementa en 50%
* Un combatiente Defendiendo recibe 50% menos daño, pero solo del próximo ataque
* Un hechizo de daño puede envenenar al objetivo, causando daño adicional en acciones futuras

## **Reglas del Combate**

1. **Puntos de vida**: Cada combatiente comienza con cierta cantidad de HP según su tipo
2. **Estados temporales**: Algunos estados (como Defendiendo) duran solo hasta el próximo ataque recibido
3. **Estados persistentes**: Otros estados (como Envenenado) duran hasta ser curados
4. **Efectos post-acción**: Al finalizar una acción, se aplican efectos como el daño por veneno

## **Extensibilidad**

El sistema debe ser extensible para agregar:

* Nuevos tipos de combatientes con diferentes habilidades y resistencias
* Nuevas acciones con efectos únicos
* Nuevos estados que modifiquen las interacciones de formas diferentes
* Nuevas resistencias y vulnerabilidades entre tipos de combatientes

## **Modalidad de trabajo**

Desarrollar el sistema mediante TDD y siguiendo todas las heurísticas de diseño vistas durante la cursada. Quitar código repetido de los tests sólo suma puntos extra.

## **Entrega**

1. Entregar el fileout de la categoría de clase **2025-1C-Parcial-2** que debe incluir toda la solución (modelo y tests). El archivo de fileout se debe llamar: **2025-1C-Parcial-2.st**
2. Entregar también el archivo que se llama **CuisUniversity-nnnn.user.changes**
3. Probar que el archivo generado en 1) se cargue correctamente en una imagen “limpia” (o sea, sin la solución que crearon. Usen otra instalación de CuisUniversity/imagen si es necesario) y que todo funcione correctamente. Esto es fundamental para que no haya problemas de que falten clases/métodos/objetos en la entrega.
4. Deben realizar la entrega enviando mail a: **fiuba-ingsoft1-doc@googlegroups.com** con el **Subject: Padrón NNNNN - Solución 2do parcial 1c2025.**En caso de rebotar el envío, reintentar comprimiendo los adjuntos.
5. **RECOMENDACIÓN IMPORTANTE: Salvar la imagen de manera frecuente o con el autosave**
6. Se asume que a esta altura de la cursada saben trabajar con la imagen, recuperarla, recuperar código fuente, revertir cambios y demás incidencias que pudieran ocurrir durante el exámen.

**Revisen bien los puntos de arriba. Cualquier error en los nombres o formato podrían ser penalizados en la nota.**

**IMPORTANTE:** **No retirarse sin tener el ok de los docentes** de haber recibido la resolución por algún medio.

**CERRAR EL TRABAJO A LAS 21:45.**

**LAS ENTREGAS RECIBIDAS DESPUÉS DE HORA NO SERÁN TENIDAS EN CUENTA**