

Trabajo Práctico Integrador: "La Arena del Gladiador"



1. Descripción del Escenario

Vas a desarrollar un simulador de batalla por turnos en Python. El programa enfrentará a un usuario (Gladiador) contra un oponente controlado por la computadora (Enemigo). El objetivo es reducir los puntos de vida del oponente a cero antes de que él lo haga contigo.

Este ejercicio evalúa el uso de variables (`int`, `float`, `string`, `boolean`), estructuras de control (`if/elif/else`), ciclos (`while` y `for`) y validación de datos estricta.

2. Requerimientos Técnicos

A. Tipos de Datos

Debes utilizar obligatoriamente los siguientes tipos de datos para las variables del juego:

- **String:** Para el nombre del jugador.
- **Int:** Para los Puntos de Vida (HP) y cantidad de paciones.
- **Float:** Para el cálculo del daño (ej: un golpe crítico multiplica el ataque por 1.5).
- **Boolean:** Para controlar si el juego sigue activo o quién tiene el turno.

B. Reglas de Validación (¡Importante!)

- **No está permitido** usar bloques `try / except`.
- Para validar texto, debes usar el método `.isalpha()` dentro de un ciclo `while`.
- Para validar números, debes usar el método `.isdigit()` dentro de un ciclo `while`.

3. Flujo del Programa

Paso 1: Configuración del Personaje

El programa inicia pidiendo el nombre del Gladiador.

- **Validación:** El nombre solo puede contener letras. Si el usuario ingresa números, símbolos o lo deja vacío, el programa debe decir "Error: Solo se permiten letras" y volver a preguntar hasta que sea válido.

Paso 2: Inicialización de Estadísticas

El programa debe definir las variables iniciales (sin preguntar al usuario):

- Vida del Gladiador: 100 (int)
- Vida del Enemigo: 100 (int)
- Pociones de Vida: 3 (int)
- Daño base "Ataque Pesado": 15 (int)
- Daño base del enemigo: 12 (int)
- Turno Gladiador : True (booleano)

Paso 3: El Ciclo de Combate

El juego entra en un ciclo que se repite **mientras** ambos combatientes tengan más de 0 puntos de vida.

Turno del Jugador:

Muestra la vida actual de ambos y las pociones restantes. Luego, ofrece un menú con 3 opciones:

1. **Ataque Pesado**
 2. **Ráfaga Veloz** (Requiere uso de `for`)
 3. **Curar**
- **Validación del Menú:** El programa debe pedir la opción al usuario.
 1. Verificar que lo ingresado sea un número (`.isdigit()`).
 2. Verificar que el número sea **1, 2 o 3**.
 - Si falla alguna validación, mostrar mensaje de error y volver a pedir.

Lógica de las Acciones:

Acción A: Ataque Pesado (Opción 1)

- Calcula el daño final. Si la vida del enemigo es menor a 20 puntos, el jugador realiza un "Golpe Crítico" multiplicando su daño base por `1.5` (resultado float).
- Resta el daño a la vida del enemigo.
- Muestra un mensaje: "*¡Atacaste al enemigo por X puntos de daño!*"

Acción B: Ráfaga Veloz (Opción 2)

- Esta acción realiza una serie de golpes rápidos. **Debes implementar un bucle `for`.**
- El bucle debe repetirse **3 veces** (usando `range`).
- Dentro del bucle, en cada repetición:
 1. Resta **5 puntos** de daño a la vida del enemigo.
 2. Muestra el mensaje: "*> Golpe conectado por 5 de daño*".

Acción C: Curar (Opción 3)

-
- Si tienes pociones (> 0): Suma 30 puntos a tu vida y resta 1 poción.
 - Si **NO** tienes pociones: Muestra "*¡No quedan pociones!*" y pierdes el turno (el enemigo ataca igual).
-

Turno del Enemigo:

Justo después de tu acción, el enemigo ataca automáticamente.

- Resta el daño base del enemigo (12) a tu vida.
- Muestra un mensaje: "*¡El enemigo te atacó por 12 puntos de daño!*"

Paso 4: Fin del Juego

Cuando el ciclo termine (porque la vida de alguno llegó a 0 o menos), debes evaluar:

-
- Si `vida_jugador > 0`: Mostrar "**VICTORIA!** [Nombre] ha ganado la batalla."
 - Si `vida_jugador <= 0`: Mostrar "**DERROTA.** Has caído en combate."
-

4. Ejemplo de Ejecución (Consola)

Plaintext

```
--- BIENVENIDO A LA ARENA ---
Nombre del Gladiador: Leonidas1
Error: Solo se permiten letras.
Nombre del Gladiador: Leonidas

==== INICIO DEL COMBATE ====
Leonidas (HP: 100) vs Enemigo (HP: 100) | Pociones: 3

Elige acción:
1. Ataque Pesado
2. Ráfaga Veloz
3. Curar
Opción: A
Error: Ingrese un número válido.
Opción: 2

>> ¡Inicias una ráfaga de golpes!
> Golpe conectado por 5 de daño
> Golpe conectado por 5 de daño
> Golpe conectado por 5 de daño

>> ¡El enemigo contraataca por 12 puntos!

==== NUEVO TURNO ====
Leonidas (HP: 88) vs Enemigo (HP: 85) | Pociones: 3
...
```

Criterios de Evaluación

1. **Funcionalidad:** El juego no se rompe si ingreso letras en el menú o números en el nombre.
2. **Uso de Estructuras:** Se implementa correctamente el ciclo `while` para el juego y el ciclo `for` para la "Ráfaga Veloz".
3. **Código Limpio:** Uso correcto de identación y nombres de variables descriptivos (ej: `vida_jugador`, no `v1`).
4. **Lógica:** El ciclo termina correctamente cuando la vida llega a cero.