**Pruebas de la Aplicación “Batalla Pokémon”**

**1. Descripción General**

El presente documento describe los casos de prueba para verificar el correcto funcionamiento del programa “Batalla Pokémon”, desarrollado en Java. El objetivo es comprobar que se cumplen los requisitos: selección de Pokémon, ataques con polimorfismo según tipo, probabilidad de fallo, uso de excepciones personalizadas y reporte de estadísticas finales.

**Casos de Prueba**

**Caso 1 – Selección válida de Pokémon**

Objetivo: Verificar que el usuario puede elegir correctamente un Pokémon inicial.

Entradas: Número 2 (Charmander).

Pasos:

* - Ejecutar la aplicación.
* - Ingresar nombre “Luis Osorio”.
* - Ingresar “2” cuando se pidan los Pokémon.

Resultado Esperado: Se muestra que el jugador seleccionó Charmander. La CPU elige aleatoriamente otro Pokémon distinto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.Resultado Obtenido:

**Caso 2 – Entrada inválida al elegir Pokémon**

Objetivo: Comprobar manejo de InvalidChoiceException.

Entradas: Texto “abc”.

Pasos:

* - Ejecutar la aplicación.
* - Ingresar nombre cualquiera.
* - Escribir “abc” cuando se pida el número.

Resultado Esperado: Se imprime mensaje “Debes escribir un número”. Se vuelve a solicitar la elección.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.Resultado Obtenido:

**Caso 3 – Ataque con ventaja de tipo (polimorfismo)**

Objetivo: Verificar que el cálculo de daño cambia según el tipo.

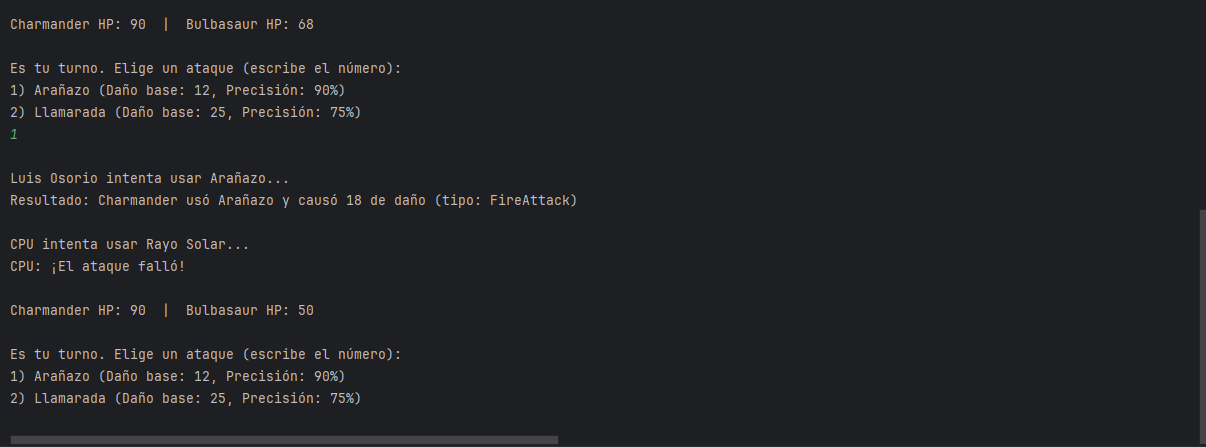
Entradas: Jugador: Charmander, CPU: Bulbasaur.

Pasos:

* - Elegir Charmander.
* - Atacar con “Llamarada”.

Resultado Esperado: El daño mostrado es aproximadamente (25 → ~37). Mensaje “causó X de daño” refleja ese incremento.

Texto

El contenido generado por IA puede ser incorrecto.Resultado Obtenido:

**Caso 4 – Ataque con desventaja de tipo**

Objetivo: Confirmar reducción de daño.

Entradas: Jugador: Squirtle, CPU: Bulbasaur

Pasos:

* - Elegir Squirtle
* -Usar Hidrobomba

Resultado Esperado: Daño reducido ( , 26 → ~18).

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.Resultado Obtenido:

**Caso 5 – Probabilidad de fallo**

Objetivo: Verificar que un ataque puede fallar.

Entradas: Cualquier Pokémon con ataque de precisión < 100 %.

Pasos:

* - Usar varias veces un ataque de precisión 75 %.
* - Observar que en algún turno se muestre “¡Tu ataque falló!”.

Resultado Esperado: Aparece mensaje “falló” y no se resta HP en ese turno.

Texto

El contenido generado por IA puede ser incorrecto.Resultado Obtenido:

**Caso 6 – Turno CPU y fallo**

Objetivo: Verificar que la CPU también falla.

Entradas: Cualquiera.

Pasos:

* - Jugar varios turnos.

Resultado Esperado: Algún turno muestra “CPU … falló con …”.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.Resultado Obtenido: Misma imagen que la adjunta en el caso “4”

**Caso 7 – Final de batalla y estadísticas**

Objetivo: Confirmar impresión de log y estadísticas.

Entradas: Jugar hasta que uno quede con 0 HP.

Pasos:

* - Ejecutar la aplicación hasta el final.

Resultado Esperado: Muestra ganador. Lista de eventos. Total de fallos, top 3 daños, promedio, conteo por actor.

Texto

El contenido generado por IA puede ser incorrecto.Resultado Obtenido:   
Texto

El contenido generado por IA puede ser incorrecto.