# Identificación del problema y análisis de requerimientos

## Caso de Estudio :

|  |  |
| --- | --- |
| Cliente |  |
| Usuario |  |
| Contexto del problema | *El sistema PokeCollector necesita gestionar una colección de cartas de Pokémon, permitiendo registrar, modificar, consultar y eliminar cartas. Cada carta debe almacenar información relevante como un ID único, nombre, puntos de vida, tipo y rareza. Además, es esencial que las cartas se puedan ordenar por puntos de vida de mayor a menor para facilitar la visualización y gestión de las cartas por parte del usuario.* |
| Requerimientos funcionales | RF1 = Registrar cartas con todos sus atributos  RF2=Ordenar las cartas de mayor a menor según sus puntos de vida |
| Requerimientos no funcionales | 1. El sistema debe ser capaz de manejar hasta 50 cartas.  2. La interfaz debe ser intuitiva y fácil de usar  3. Debe garantizar la persistencia de datos durante la sesión. |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | RF1 = Registrar cartas con todos sus atributos | | | |
| Resumen | *El sistema permite al usuario registrar una carta ingresando los atributos requeridos (ID, nombre, puntos de vida, tipo, rareza). El ID debe ser único; en caso contrario, se solicitará uno nuevo. Posteriormente, las cartas pueden ser ordenadas por puntos de vida en orden descendente.* | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| ID | String | | *No puede ser nulo y debe ser único* |
| Nombre | String | | *No puede ser nulo* |
| Puntos de vida | int | | *Debe ser un número positivo* |
| Tipo | String | | *Debe pertenecer a una lista de tipos predefinidos* |
| Rareza | String | | Debe ser uno de los valores: básico, raro, mítico, legendario |
| Resultado o Postcondición | La carta es registrada en el sistema y se garantiza que el ID es único. En caso de que el ID no sea único, se requiere un nuevo ingreso del ID o muestra un mensaje de error | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
| Mensaje de confirmación | | String | *"Carta registrada exitosamente"* |
| Lista de cartas | | String | “Formato: "ID - Nombre - Puntos de Vida - Tipo - Rareza" |
|  | Error | | String | *"El ID ya existe, digite otro"* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador y nombre | RF2=Ordenar las cartas de mayor a menor según sus puntos de vida | | | |
| Resumen | *El sistema permite al usuario ordenar las cartas registradas por puntos de vida en orden descendente y mostrar la lista resultante* | | | |
| Entradas | **Nombre entrada** | **Tipo de dato** | | **Condición valores válidos** |
| N/A | N/A | | N/A |
| Resultado o Postcondición | Las cartas se ordenan y se muestra la lista de cartas en orden de puntos de vida de mayor a menor o muestra un mensaje de error | | | |
| Salidas | **Nombre salida** | | **Tipo de dato** | **Formato** |
| Lista de cartas ordenadas | | String | “Formato: "ID - Nombre - Puntos de Vida - Tipo - Rareza" |
|  | Error | | String | *"Error, no se pudo ordenar"* |