

Especificación de Requisitos según el estándar de IEEE 830

IEEE Std. 830-1998

Resumen

Este documento presenta, en castellano, el formato de Especificación de Requisitos Software (ERS) según la última versión del estándar IEEE 830. Según IEEE, un buen Documento de Requisitos, pese a no ser obligatorio que siga estrictamente la organización y el formato dados en el estándar 830, si debería incluir, de una forma o de otra, toda la información presentada en dicho estándar. El estándar de IEEE 830 no está libre de defectos ni de prejuicios, y por ello ha sido justamente criticado por múltiples autores y desde múltiples puntos de vista, llegando a cuestionar incluso si es realmente un estándar en el sentido habitual que tiene el término en otras ingenierías. El presente documento no pretende pronunciarse ni a favor ni en contra de unos u otros: tan solo reproduce, con propósitos fundamentalmente docentes, cómo se organizaría un Documento de Requisitos según el estándar IEEE 830.

Especificación de Requisitos de Software

Proyecto: Magical Creature

Valentina Cruz Rivera

Jhampier Santos Ortiz

Carol Nicole Marentes Torres

Índice

1. Introducción	3
1.1. Propósito	3
1.2. Ámbito del Sistema	3
1.3. Definiciones, Acronimos y Abreviaturas	3
1.4. Referencias	4
1.5. Visión General del Documento	5
2. Descripción General	5
2.1. Perspectiva del Producto	5
2.2. Funciones del Producto	5
2.3. Características de los Usuarios	6
2.4. Restricciones	6
2.5. Suposiciones y Dependencias	6
2.6. Requisitos Futuros	7
2.7. Objetivos Generales y Específicos	7
3. Recolección de datos	8
3.1. Historias de Usuario	8
3.2. Entrevista	9
3.3. Evaluación Heurística	9
4. Requisitos Específicos	10
4.1. Interfaces Externas	10
4.2. Funciones	10
4.3. Requisitos de Rendimiento	11
4.4. Restricciones de Diseño	11
4.5. Atributos del Sistema	12
4.6. Otros Requisitos	12
5. Apéndices	32

1. Introducción

La presente introducción tiene como finalidad definir los lineamientos iniciales para el desarrollo del juego digital Magical Creature propuesto en el marco del reto SenaSoft el juego consiste en una competencia de cartas en la que los jugadores buscan acumular la mayor cantidad de cartas mediante la comparación de atributos de modelos predefinidos.

El sistema se implementará utilizando PHP para la lógica de negocio y Css para la interfaz gráfica, asegurando una solución eficiente, moderna y atractiva para los participantes, Y de framework utilizaremos Bootstrap para simplificar y facilitar el desarrollo del juego de cartas

1.1. Propósito

Define de manera clara y detallada los requisitos funcionales y no funcionales del sistema Magical Creature un juego de cartas digital desarrollado en el marco del reto SenaSoft, este documento servirá como guía principal para el equipo de desarrollo asegurando que todos los miembros comprendan el alcance, los objetivos y las funcionalidades claves que debe cumplir el sistema.

1.2. Ámbito del Sistema

El sistema abarca las siguientes funcionalidades:

- Sala única de juego.
- Reparto aleatorio de cartas entre jugadores.
- Comparación de atributos por rondas.
- Determinación de ganadores.
- Visualización de cartas, atributos, turnos y puntajes.

Tecnologías utilizadas:

Backend: PHP

Frontend: CSS + Bootstrap

Base de datos: MySQL

1.3. Definiciones, Acronimos y Abreviaturas

SRS	Especificación de requisitos del software
-----	---

ADSO	Análisis y Desarrollo de Software
RF	Requisitos Funcionales
RFN	Requisitos no Funcionales
MER	Modelo Entidad Relación
GUI	Interfaz gráfica de usuario

1.4. Referencias

SERESMITOLOGICOS.(2018). Seres Mitológicos y Fantásticos.

<https://www.seresmitologicos.net/>

Mary Santos. (2018). Características de un usuario. Prezi.

<https://prezi.com/1hdbbej8zafu/caracteristicas-de-un-usuario/>

Blog de QAT Insights.(2025).Redacción de suposiciones y restricciones en una especificación de requisitos de software (SRS). QAT GLOBAL.

<https://qat.com/writing-assumptions-constraints-srs/>

María Alonso. (2024). Objetivos generales y específicos: Qué son y cómo redactarlos [Plantilla gratis]. Asana

<https://asana.com/es/resources/general-and-specific-objectives>

Alexander Menzinsky, Gertrudis López, Juan Palacio, Miguel Ángel Sobrino, Rubén Álvarez y Verónica Rivas. (2022). Historias de Usuario. Scrum Manager.

https://www.scrummanager.com/files/scrum_manager_historias_usuario.pdf

Santiago lopez. (2019). Formato requerimientos funcionales. Studocu.

<https://www.studocu.com/co/document/politecnico-grancolombiano/ingenieria-de-software-i/formato-requerimientos-funcionales/14047738>

Ayat Shukairy. (2025). Evaluación heurística: su guía completa para la evaluación heurística y la optimización de la conversión. Invesp.

<https://www.invespcro.com/blog/heuristic-evaluation-your-complete-guide>

1.5. Visión General del Documento

Se describen los requisitos necesarios para desarrollar el juego Magical Creature. Incluyendo una visión general del sistema, sus funciones, restricciones, características del usuario y requerimientos técnicos. Sirve como guía para el equipo de desarrollo y asegura que todos trabajen alineados bajo los mismos objetivos.

2. Descripción General

Magical Creature es un juego de cartas digital, ambientado en un universo mágico de criaturas del bosque. Cada jugador recibe 8 cartas con atributos numéricos como fuerza, velocidad, magia y resistencia. En cada ronda, se elige un atributo para competir, y quien tenga el valor más alto gana las cartas, es decir, gana los puntos de los demás y continúa la siguiente ronda.

El juego tiene como propósito ofrecer una experiencia dinámica, sencilla y entretenida, reforzando habilidades lógicas y de análisis comparativo.

2.1. Perspectiva del Producto

El juego será desarrollado como una aplicación web. Se Compondrá de dos módulos principales: el cliente (frontend) para interacción visual con las cartas y el servidor (backend) para manejar la lógica, turnos, reparto y resultados.

Se diseñará para ser autosuficiente, ejecutable para navegadores web.

2.2. Funciones del Producto

- Sala única y definir el número de jugadores.
- Repartir 8 cartas únicas por jugador.
- Visualizar cartas y atributos en pantalla.
- Permitir selección de atributos en cada ronda.
- Comparar valores y asignar cartas al ganador.
- Mostrar total cartas, turnos y resultados.
- Control general del tiempo para la partida.
- Finalizar la partida y declarar un ganador.

2.3. Características de los Usuarios

Rol del Usuario	Actividad
Usuario	Unirse a la sala única. Registrar nombre. Competir.

2.4. Restricciones

- El proyecto debe ser usado con acceso a internet
- El proyecto debe cumplir con las políticas de seguridad para proteger la información del usuario.
- La ejecución debe ser rápida y eficiente.
- El software debe ser intuitivo y fácil de usar.
- El proyecto debe ser compatible con diferentes navegadores web.

2.5. Suposiciones y Dependencias

- Se espera que los usuarios tengan acceso a internet y a algún navegador web.
- Se espera que el usuario tenga conocimientos básicos sobre la temática del juego.
- Se espera que el usuario tenga conocimientos básicos en tecnología.

2.6. Requisitos Futuros

- A largo plazo se incrementaran los niveles de dificultad
- A mediano plazo se tendrá la posibilidad de seleccionar el idioma de su preferencia.

2.7. Objetivos Generales y Específicos

Objetivo general: Diseñar e implementar el juego “Magical Creature”, un juego de cartas dinámico e interactivo que permita a varios jugadores competir en tiempo real mediante la comparación de atributos de cartas esto generando la necesidad de desarrollar una plataforma web que gestione las partidas, controle los turnos de manera de manecillas de reloj, garantizando una experiencia ágil y entretenida con un mínimo 2 jugadores y un máximo 7.

Objetivo específicos:

- Realizar el proceso de elicitación de requisitos.
- Hacer el análisis respectivo
- Diseñar la solución del producto
- Desarrollar la solución planteada
- Generar el proceso de implementación y evaluación.n

3. Recolección de datos

3.1. Historias de Usuario

Historia de Usuario #1:
Como: jugador nuevo Quiero: ver un tutorial con ejemplos de cartas y jugadas Para: entender rápidamente las reglas sin tener que leer demasiado.
Criterios de Aceptación: Se muestra una diapositiva o pantalla inicial con las reglas resumidas. Incluye imágenes de cartas de ejemplo. El jugador entiende en menos de 2 minutos cómo jugar.
Historia de Usuario #2:

<p>Como: jugador en medio de la partida</p> <p>Quiero: ver claramente cuántas cartas tiene cada participante</p> <p>Para:saber quién va ganando en todo momento.</p>
<p>Criterios de Aceptación:</p> <p>La interfaz muestra el conteo actualizado de cartas por jugador. El número cambia automáticamente cuando se gana o pierde una carta. Se resalta visualmente al jugador con más cartas.</p>

Historia de Usuario #3:
<p>Como: jugador competitivo</p> <p>Quiero: que el sistema me avise cuando cometo un error o jugada inválida</p> <p>Para:saber quién va ganando en todo momento.</p>
<p>Criterios de Aceptación:</p> <p>La interfaz muestra el conteo actualizado de cartas por jugador. El número cambia automáticamente cuando se gana o pierde una carta. Se resalta visualmente al jugador con más cartas.</p>

3.2. Entrevista

¿Cuál considera que es el principal objetivo del reto a enfrentar?

¿Qué competencias espera que desarrollemos con este tipo de actividades?

¿Qué tan importante es la organización del equipo para lograr buenos resultados en este reto?

¿De qué manera se evaluará nuestro desempeño durante el desarrollo del reto?

¿Qué errores comunes deberíamos evitar al enfrentar un reto como este?

Según su experiencia, ¿qué aspectos son fundamentales para que el juego sea atractivo y funcional?

¿Qué recomendación nos daría para la duración de la partida?

¿Qué alcance visualiza que tiene este proyecto?

¿Cómo cree que se debería mejorar la dinámica del juego?

Desde su experiencia, ¿qué recomendaciones nos ofrece para el desarrollo del proyecto?

3.3. Evaluación Heurística

- Principio: Visibilidad del estado del sistema

Heurística: El sistema debe mantener a los usuarios informados sobre lo que está pasando, a través de retroalimentación adecuada y de forma clara.

Evaluación: En el juego de cartas, es importante que el jugador siempre sepa en qué parte del juego está, es decir si es su turno, cuantas rondas quedan, cuántas cartas ha ganado, etc.

- Principio: Relación entre el sistema y el mundo real

Heurística: El juego debe usar conceptos familiares para el usuario y basarse en convenciones del mundo real, haciendo que este sea comprensible y accesible.

Evaluación: El uso de atributos de las criaturas místicas como fuerza, velocidad, magia, etc, se alinean con la temática del bosque donde los jugadores pueden identificar fácilmente las cartas y lo que representan.

- Principio: Control y libertad del usuario

Heurística: El usuario debe tener control sobre el sistema y ser capaz de

deshacer o corregir acciones para que este pueda evitar frustraciones.

Evaluación: En el juego actual el jugador no puede deshacer una carta si ya ha sido seleccionada para hacer la jugada, esto se hace para evitar que el jugador no se pueda arrepentir de su selección.

4. Requisitos Específicos

4.1. Interfaces Externas

Interfaz de Usuario (IU): Diseño web con CSS + Bootstrap.

Base de Datos: MySQL para persistencia de usuarios, partidas y estadísticas

API: Desarrollada en PHP para la lógica del juego.

4.2. Funciones

Aquí se detallan los requisitos funcionales del sistema:

- El sistema debe permitir a los usuarios registrarse con su nombre.
- Se crea solamente una sala para todo el juego.
- Se puede definir el número de jugadores permitidos en la sala (entre 2 y 7).
- El sistema debe notificar al jugador que no puede usar la sala cuando esta ya está en uso.
- Al inicio de una partida, el sistema debe repartir 8 cartas aleatorias y únicas a cada jugador.
- El total de cartas en el juego es de 56, si lo máximo son 7 jugadores.
- El sistema debe mostrar las cartas en la mano del jugador actual.
- El sistema debe mostrar los atributos de las cartas de forma clara.
- El sistema debe mostrar el número de cartas de cada participante
- El sistema debe indicar claramente de quién es el turno.
- El sistema debe asignar el primer turno de forma aleatoria.
- El jugador en turno debe poder seleccionar un atributo de su carta para competir.
- El sistema debe comparar el valor del atributo seleccionado entre todas las cartas jugadas en la ronda.
- El sistema debe asignar todas las cartas(puntos) jugadas en la ronda al

jugador que haya obtenido el atributo de mayor valor.

- La partida debe finalizar cuando un jugador ha acumulado todas las cartas(puntos) o cuando se cumpla un tiempo límite estimado.
- El sistema debe declarar al jugador con más cartas(puntos) al final de la partida como el ganador.
- El sistema debe mostrar un mensaje claro si el jugador intenta realizar una jugada inválida.

4.3. Requisitos de Rendimiento

- Tiempo de respuesta: Las interacciones del usuario (selección de cartas, actualización de puntajes) deben tener un tiempo de respuesta inferior a 2 segundos.
- Concurrencia: El sistema debe soportar al menos 7 partidas simultáneas con 7 jugadores cada una sin degradación significativa del rendimiento.
- Eficiencia en el reparto: El reparto aleatorio de cartas debe ser instantáneo al inicio de la partida.
- Escalabilidad: El diseño del *backend* debe permitir el escalado horizontal para soportar un mayor número de usuarios simultáneos en el futuro.
- Tecnología backend: El backend debe ser desarrollado utilizando PHP.
- Tecnología frontend: El *frontend* debe ser desarrollado utilizando CSS y Bootstrap.
- Base de datos: Se debe utilizar MySQL como sistema de gestión de bases de datos.
- Compatibilidad: La aplicación web debe ser compatible con las últimas versiones de los navegadores web principales (Chrome, Firefox, Edge, Safari).

4.4. Restricciones de Diseño

- Tecnología Backend: El backend debe ser desarrollado utilizando PHP.
- Tecnología Frontend: El frontend debe ser desarrollado utilizando CSS y Bootstrap.
- Base de Datos: Se debe utilizar MySQL como sistema de gestión de bases de datos.
- Compatibilidad: La aplicación web debe ser compatible con las últimas versiones de los navegadores web principales (Chrome, Firefox, Edge, Safari).

4.5. Atributos del Sistema

■ **Fiabilidad:** El sistema debe garantizar la persistencia de los datos de las partidas y perfiles de usuario, con un mínimo de pérdida de información en caso de fallos.

■ **Usabilidad:** La interfaz de usuario debe ser intuitiva y fácil de aprender, incluso para usuarios sin experiencia previa en juegos de cartas digitales.

■ **Mantenibilidad:** El código debe estar bien documentado, modularizado y seguir principios de diseño que faciliten futuras modificaciones y adiciones.

■ **Seguridad:** El sistema debe proteger la información del usuario y las partidas contra accesos no autorizados y manipulaciones.

4.6. Otros Requisitos

4.6.1. Requerimientos Funcionales

Identificador: RF 01		Nombre: Registro de Usuario	
Tipo: Usuario	Requerimiento que lo utiliza o especializa: N/A	¿Crítico? Si	
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup		
Entrada: - Nombre de Usuario		Salida: - Acceso al sistema o mensaje de error (credenciales incorrectas).	
Descripción: El sistema debe permitir a los usuarios registrarse con su nombre.			
Manejo de situaciones anormales Si el nombre de usuario ya está registrado, se muestra un mensaje de error.			

Criterios de aceptación

El usuario puede introducir su nombre de usuario en el formulario de registro.

Si las credenciales son inválidas, se muestra un mensaje de error adecuado.

Identificador: RF 02		Nombre: Sala de juego	
Tipo: Usuario	Requerimiento que lo utiliza o especializa: N/A	¿Crítico? Si	
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup		
Entrada: - Nombre de la sala, número de jugadores (entre 2 y 7)		Salida: - Sala de juego creada y visible para otros usuarios	
Descripción: El usuario puede unirse a la única sala existente y elegir el número de jugadores.			
Manejo de situaciones anormales Si el número de jugadores está fuera del rango permitido (2-7), el sistema debe mostrar un mensaje de error y no permitir la creación de la sala. Si no lo deja ingresar es porque la única sala está en uso y debe esperar a que se acabe la partida.			
Criterios de aceptación Se puede unir a una sola sala, donde se selecciona el número de jugadores que debe de ser entre 2 y 7. Si no puede entrar a la sala es porque está ocupada y debe de esperar a que esta acabe para poder iniciar una nueva ronda.			

Identificador: RF 03**Nombre:**
Notificación sala ocupada

Tipo: Sistema	Requerimiento que lo utiliza o especializa: N/A	¿Crítico? Si
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup	
Entrada: - Intento de jugar en la única sala		Salida: - Mensaje de notificación "Sala ocupada"
Descripción: El sistema debe notificar a los jugadores cuando la sala única está ocupada y no se puede unir.		
Manejo de situaciones anormales		
Criterios de aceptación Cuando un usuario intenta unirse a la sala que ya tiene su número de jugadores o ya ha empezado, se muestra un mensaje de "Sala ocupada" de forma prominente. El mensaje es claro y comprensible		

Identificador: RF 04		Nombre: Reparto aleatorio de cartas	
Tipo: Sistema	Requerimiento que lo utiliza o especializa: N/A	¿Crítico? Si	
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup		
Entrada: - Inicio de una partida		Salida: 8 cartas únicas repartidas a cada jugador	

Descripción: Al inicio de una partida, el sistema debe repartir 8 cartas aleatorias y únicas a cada jugador. El total de cartas en el juego es de 56 (máximo 7 jugadores)
Manejo de situaciones anormales <ul style="list-style-type: none"> Si no hay suficientes cartas para repartir 8 a cada jugador (ej. configuración incorrecta), el sistema debe generar un error y no iniciar la partida
Criterios de aceptación <ul style="list-style-type: none"> Se asegura que todas las cartas repartidas sean únicas. El número total de cartas en juego sea 56 si hay 7 jugadores, distribuidas

Identificador: RF 05		Nombre: Mostrar atributos de las cartas
Tipo: Sistema	Requerimiento que lo utiliza o especializa: N/A	¿Crítico? Si
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup	
Entrada: <ul style="list-style-type: none"> Carta seleccionada o en visualización. 		Salida: <ul style="list-style-type: none"> Valores de los atributos de la carta.
Descripción: El sistema debe mostrar los atributos (fuerza, velocidad, magia, resistencia) de las cartas de forma clara y legible.		
Manejo de situaciones anormales		
Criterios de aceptación Los atributos de cada carta se muestran de forma legible (con etiquetas y sus valores numéricos). Los atributos son fácilmente identificables y comparables.		

--

Identificador: RF 06		Nombre: Mostrar cartas en mano	
Tipo: Sistema	Requerimiento que lo utiliza o especializa: N/A	¿Crítico? Si	
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup		
Entrada: - Mano del jugador actual		Salida: - Representación visual de las cartas del jugador	
Descripción: El sistema debe mostrar las cartas que el jugador tiene en su mano de forma clara y accesible			
Manejo de situaciones anormales			
Criterios de aceptación <ul style="list-style-type: none">• Las cartas del jugador actual son visibles en su interfaz de juego• Cada carta se muestra con su diseño y atributos correspondientes• El tamaño y la disposición de las cartas permiten una fácil visualización y selección			

Identificador: RF 7		Nombre: Mostrar número de cartas por participante	
Tipo: Sistema	Requerimiento que lo utiliza o especializa: N/A	¿Crítico? Si	
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup		

Entrada: <ul style="list-style-type: none"> - Estado de las cartas de cada jugador. 	Salida: <ul style="list-style-type: none"> - Conteo de cartas por jugador visible en la interfaz.
Descripción: El sistema debe mostrar el número de cartas que cada participante tiene en su poder, incluyendo el propio jugador.	
Manejo de situaciones anormales	
Criterios de aceptación La interfaz de juego muestra el número actual de cartas de cada jugador.El conteo de cartas se actualiza en tiempo real después de cada ronda o acción. El número de cartas de cada jugador es visible.	

Identificador: RF 8		Nombre: Indicación de turno	
Tipo: Sistema	Requerimiento que lo utiliza o especializa: N/A	¿Crítico? Si	
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup		
Entrada: - Cambio de turno		Salida: - Resaltado visual o mensaje indicando el jugador en turno.	
Descripción: El sistema debe indicar claramente de quién es el turno de jugar.			
Manejo de situaciones anormales			

Criterios de aceptación

El jugador cuyo turno es se resalta visualmente en la interfaz. Se muestra un mensaje o ícono que indica "Es tu turno" o "Turno de (Nombre del jugador)". La indicación es persistente hasta que el turno cambia.

Identificador: RF 9		Nombre: Asignación aleatoria del primer turno	
Tipo: Sistema	Requerimiento que lo utiliza o especializa: N/A	¿Crítico? Si	
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup		
Entrada: - Inicio de partida		Salida: - Jugador inicial seleccionado aleatoriamente	
Descripción: El sistema debe asignar el primer turno de forma aleatoria al inicio de cada partida.			
Manejo de situaciones anormales			
Criterios de aceptación <ul style="list-style-type: none">● Al comenzar una partida, el sistema elige aleatoriamente a uno de los jugadores para iniciar.● El primer jugador en turno se indica claramente.● La aleatoriedad del turno inicial es consistente y no predecible.			

Identificador: RF 10**Nombre:**
Selección de atributo para competir

Tipo: Sistema	Requerimiento que lo utiliza o especializa: N/A	¿Crítico? Si
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup	
Entrada: <ul style="list-style-type: none"> - Carta en mano del jugador en turno, selección de atributo. 		Salida: <ul style="list-style-type: none"> - Atributo seleccionado para la ronda
Descripción: El jugador en turno debe poder seleccionar un atributo de una de sus cartas para competir en la ronda actual.		
Manejo de situaciones anormales Si el jugador intenta seleccionar una carta que no está en su mano o un atributo inválido, el sistema debe mostrar un mensaje de error.		
Criterios de aceptación <ul style="list-style-type: none"> • El jugador en turno puede seleccionar una carta de su mano. • Al seleccionar una carta, el jugador puede elegir uno de sus atributos para competir. • La selección del atributo se confirma antes de proceder con la comparación. 		

Identificador: RF 11		Nombre: Comparación de atributos
Tipo: Sistema	Requerimiento que lo utiliza o especializa: N/A	¿Crítico? Si
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup	

Entrada: <ul style="list-style-type: none"> - Atributo seleccionado y cartas jugadas por todos los participantes 	Salida: <ul style="list-style-type: none"> - Determinación del ganador de la ronda
Descripción: El sistema debe comparar el valor del atributo seleccionado entre todas las cartas jugadas en la ronda para determinar el ganador.	
Manejo de situaciones anormales	
Criterios de aceptación <ul style="list-style-type: none"> • El sistema compara los valores del atributo elegido de todas las cartas en juego para la ronda • Se identifica correctamente la carta con el valor más alto del atributo 	

Identificador: RF 12		Nombre: Asignación de cartas(puntos) al ganador de la ronda	
Tipo: Sistema	Requerimiento que lo utiliza o especializa: N/A	¿Crítico? Si	
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup		
Entrada: <ul style="list-style-type: none">- Resultado de la comparación de atributos.		Salida: <ul style="list-style-type: none">- Cartas(puntos) de la ronda transferidas al ganador.	

Descripción: El sistema debe asignar todas las cartas(puntos) jugadas en la ronda al jugador con el atributo de mayor valor.
Manejo de situaciones anormales
Criterios de aceptación <ul style="list-style-type: none"> El jugador ganador de la ronda recibe todas las cartas(puntos) que fueron acumuladas en esa ronda.

Identificador: RF 13		Nombre: Inicio de la siguiente ronda	
Tipo: Sistema	Requerimiento que lo utiliza o especializa: N/A		¿Crítico? Si
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup		
Entrada: <ul style="list-style-type: none">- El jugador gana la ronda		Salida: <ul style="list-style-type: none">- El sistema inicia la siguiente ronda, con el turno asignado al jugador siguiente de la ronda anterior.	
Descripción: El sistema debe pasar el turno en modo de manecillas de reloj cuando empieza la 2 ronda.			
Manejo de situaciones anormales			
Criterios de aceptación El turno se debe asignar al jugador siguiente como en las manecillas del reloj para dar inicio a la nueva ronda. La transición entre rondas es fluida y sin interrupciones.			

Identificador: RF 14		Nombre: Indicador de turno	
Tipo: Sistema	Requerimiento que lo utiliza o especializa: N/A		¿Crítico? Si
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup		
Entrada: - Cambio de turno		Salida: - Resaltado visual o mensaje indicando el jugador en turno.	
Descripción: El sistema debe indicar claramente de quién es el turno de jugar.			
Manejo de situaciones anormales			
Criterios de aceptación El jugador cuyo turno es se resalta visualmente en la interfaz. Se muestra un mensaje o ícono que indica "Es tu turno" o "Turno de (Nombre del jugador)". La indicación es persistente hasta que el turno cambia.			

Identificador: RF 15		Nombre: Finalización de partidas por cartas	
Tipo: Usuario	Requerimiento que lo utiliza o especializa: N/A		¿Crítico? Si
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup		

Entrada: <ul style="list-style-type: none"> - Un jugador acumula todas las cartas(puntos). 	Salida: <ul style="list-style-type: none"> - La partida finaliza.
Descripción: La partida debe finalizar cuando un jugador ha acumulado todas las cartas(puntos).	
Manejo de situaciones anormales	
Criterios de aceptación Cuando un jugador tiene la totalidad de las 56 cartas, el sistema debe reconocerlas automáticamente y finalizar la partida. Un mensaje claro de "¡Has ganado todas las cartas!" o similar debe ser mostrado al jugador ganador.-	

Identificador: RF 16		Nombre: Declaración de ganador	
Tipo: Sistema	Requerimiento que lo utiliza o especializa: N/A	¿Crítico? Si	
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup		
Entrada: <ul style="list-style-type: none">- Finalización de la partida (por cartas(puntos) o tiempo)		Salida: <ul style="list-style-type: none">- El jugador con más cartas es declarado ganador	
Descripción: El sistema debe declarar al jugador con más cartas(puntos) al final de la partida como el ganador.			
Manejo de situaciones anormales			

Criterios de aceptación

Al finalizar la partida, el sistema compara el número de cartas(puntos) de cada jugador. El jugador con el mayor número de cartas(puntos) es identificado y declarado como el ganador.

Identificador: RF 17		Nombre: Mensaje de jugada inválida	
Tipo: Sistema	Requerimiento que lo utiliza o especializa: N/A	¿Crítico? Si	
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup		
Entrada: - Intento de jugada inválida por parte del jugador		Salida: - Mensaje claro de error indicando la jugada inválida.	
Descripción: El sistema debe mostrar un mensaje claro si el jugador intenta realizar una jugada inválida (ej. seleccionar una carta que no está en su mano, intentar jugar fuera de turno, seleccionar un atributo no existente).			
Manejo de situaciones anormales			
Criterios de aceptación Cuando un jugador intenta una acción que no está permitida según las reglas del juego, se muestra instantáneamente un mensaje de error legible y comprensible. El mensaje debe indicar específicamente qué fue inválido y, si es posible, cómo corregirlo.			

4.6.2. Requerimientos No Funcionales

Identificador: RNF 1		Nombre: Tiempo de respuesta	
Tipo: Sistema	Requerimiento que lo utiliza o especializa: N/A	¿Crítico? Si	
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup		
Entrada: <ul style="list-style-type: none">- Interacciones del usuario (selección de cartas, actualización de puntajes)		Salida: <ul style="list-style-type: none">- Respuesta de la interacción	
Descripción: Las interacciones del usuario (selección de cartas, actualización de puntajes) deben tener un tiempo de respuesta inferior a 2 segundos.			
Manejo de situaciones anormales Si el tiempo de respuesta excede los 2 segundos, el sistema debe investigar y optimizar los procesos para garantizar la fluidez			
Criterios de aceptación Las interacciones del usuario deben completarse en menos de 2 segundos de forma consistente			

Identificador: RNF 2		Nombre: Concurrencia	
Tipo: Sistema	Requerimiento que lo utiliza o especializa: N/A	¿Crítico? Si	
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup		

Entrada: Múltiples partidas y jugadores simultáneos	Salida: Rendimiento estable del sistema
Descripción: El sistema debe soportar al menos 7 partidas simultáneas con 7 jugadores cada una sin degradación significativa del rendimiento	
Manejo de situaciones anormales Si se detecta una degradación del rendimiento con el número especificado de partidas y jugadores, se deben optimizar los recursos del servidor y la lógica del juego.	
Criterios de aceptación El sistema debe mantener un rendimiento óptimo (sin ralentizaciones o caídas) cuando se ejecutan 7 partidas con 7 jugadores cada una simultáneamente.	

Identificador: RNF 3		Nombre: Eficiencia en el reparto de cartas	
Tipo: Sistema	Requerimiento que lo utiliza o especializa: N/A	¿Crítico? Si	
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup		
Entrada: Inicio de una partida		Salida: Cartas repartidas instantáneamente a los jugadores	
Descripción: El reparto aleatorio de cartas debe ser instantáneo al inicio de la partida.			

<p>Manejo de situaciones anormales</p> <p>Si el reparto no es instantáneo, se deben revisar y optimizar los algoritmos de reparto y la carga de datos.</p>
<p>Criterios de aceptación</p> <p>Las 8 cartas únicas por jugador deben ser visibles en la interfaz del jugador inmediatamente después de que la partida comience.</p>

Identificador: RNF 4		Nombre: Escalabilidad	
Tipo: Sistema	Requerimiento que lo utiliza o especializa: N/A		¿Crítico? Si
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup		
Entrada: Futuro aumento de usuarios simultáneo		Salida: Capacidad de la aplicación para manejar más usuarios	
Descripción: El diseño del backend debe permitir el escalado horizontal para soportar un mayor número de usuarios simultáneos en el futuro.			
Manejo de situaciones anormales Si el sistema no puede escalar horizontalmente, se debe refactorizar la arquitectura del backend para permitir la adición de más recursos de servidor.			
Criterios de aceptación La arquitectura del backend debe ser modular y permitir la fácil adición de nuevas instancias de servidor para manejar un aumento de la demanda.			

Identificador: RNF 5	Nombre: Compatibilidad del navegador
-----------------------------	--

Tipo: Sistema	Requerimiento que lo utiliza o especializa: N/A	¿Crítico? Si
Prioridad de desarrollo: Alta	Documentos de visualización asociados: Mockup	
Entrada: Acceso a la aplicación web desde diferentes navegadores		Salida: Funcionamiento correcto en navegadores principales
Descripción: La aplicación web debe ser compatible con las últimas versiones de los navegadores web principales (Chrome, Firefox, Edge, Safari).		
Manejo de situaciones anormales Si la aplicación no funciona correctamente en alguno de los navegadores especificados, se deben realizar los ajustes necesarios en el código frontend.		
Criterios de aceptación La aplicación debe cargar y funcionar sin errores visuales o funcionales en las últimas versiones de Chrome, Firefox, Edge y Safari.		

5. Apéndices

<https://www.figma.com/design/0u7UkppVARrXQBZBYtlWB/Untitled?m=auto&t=d2bwGI5CkuyjKDwP-6>