

Documento de Especificación de Requisitos Software (ERS)

Proyecto: Juego de Cartas - Ronda Marroqui

Versión del Documento: 2.0 (Sprint 2)

Fecha: 01 de noviembre de 2025

Autores: Yahya Aboulafiya - Adrián Hoyos Sánchez - Souhail Batah - Carlos Robledo Badía

1. Introducción

1.1. Propósito

El propósito de esta versión del documento es **actualizar el ERS** con los requisitos funcionales y no funcionales del **Sprint 2**. Este Sprint se centra en el diseño e implementación de la persistencia de datos (Base de Datos) y la lógica de servidor (Backend) que soportará la aplicación.

1.2. Alcance del Proyecto

(El alcance general del proyecto no ha cambiado)

El proyecto consiste en el desarrollo de una aplicación web que permita a entre 3 y 6 jugadores jugar partidas del juego Ronda Marroqui. El sistema gestionará el reparto de cartas, la lógica de los turnos, las reglas especiales y determinará al ganador y al "perdedor". El sistema también deberá incluir un sistema de usuarios (jugadores y administradores) y un ranking de partidas.

1.3. Glosario de Términos

Término	Definición
Ronda Marroqui	Nombre del juego de cartas a desarrollar.
Perdedor	Término para el último jugador que se queda con cartas.
Palo	Se refiere a los cuatro tipos de cartas de la baraja española (oros, copas, espadas, bastos).

Mazo de Robo	Pila de cartas de donde los jugadores roban cuando no pueden jugar.
CRUD	(Nuevo) Acrónimo de C rear, R eer (Leer), U pdate (Actualizar), D
E/R	(Nuevo) Modelo E ntidad/ R elación. El plano de la base de datos.
UML	(Nuevo) U nified M odeling L anguage. El plano del código de la aplicación.
Torneo	(Nuevo) Un evento de competición que agrupa varias partidas.
Clasificación	(Nuevo) Un ranking de puntos específico de un torneo.

2. Descripción General

2.1. Perspectiva del Producto

La aplicación será un sistema de juego de cartas online. Los jugadores interactuarán a través de una interfaz web. El backend se desarrollará para gestionar la lógica del servidor y la base de datos en **MySQL**.

2.2. Características de los Usuarios

- **Jugador:** Usuario que puede registrarse, iniciar sesión, unirse a partidas, jugar sus cartas y ver los rankings.
- **Administrador:** Usuario con privilegios para preparar torneos, definir reglas y gestionar el sistema.

2.3. Restricciones Generales

- El prototipo inicial se desarrollará utilizando HTML y JavaScript.
- Se utilizará la baraja española de 40 cartas.
- El proyecto debe estar gestionado a través de un repositorio en GitHub (doc/, src/, lib/).
- **(Nuevo)** El backend se desarrollará con JavaScript del lado del servidor.
- **(Nuevo)** La base de datos será MySQL.

3. Requisitos Específicos

3.1. Requisitos Funcionales (RF)

(Se mantienen los requisitos del Sprint 1 y se añaden los del Sprint 2)

ID	Requisito	Descripción	Sprint
RF-01	Gestión de Usuarios	El sistema debe permitir el registro y autenticación de jugadores, y contar con un rol de Administrador.	Futuro
RF-02	Creación de Partidas	El sistema debe permitir crear una partida para 3 a 6 jugadores, barajando y repartiendo las cartas.	S1
RF-03	Lógica del Juego	El sistema debe validar los turnos, permitir jugar o robar cartas según las reglas, y gestionar el mazo.	Futuro
RF-04	Cartas Especiales	El sistema debe aplicar correctamente los efectos de las cartas 1, 2, 4 y 7, incluyendo la	Futuro

		acumulación.	
RF-05	Fin de la Partida	El sistema debe detectar al ganador (primer jugador sin cartas) y al "perdedor".	Futuro
RF-06	Ranking Global	El sistema debe almacenar y mostrar un ranking global de partidas ganadas/perdidas por cada jugador.	Futuro
RF-07	Registro de Usuario (CRUD - Create)	(Nuevo) El sistema debe permitir a un nuevo usuario registrarse (crear) en la base de datos.	S2
RF-08	Autenticación (CRUD - Read)	(Nuevo) El sistema debe permitir a un usuario existente iniciar sesión (leer) validando sus credenciales contra la base de datos.	S2
RF-09	Persistencia de Partidas	(Nuevo) El sistema debe guardar el resultado de cada partida (quién jugó, ganador, perdedor) en la base de datos.	S2
RF-10	Actualización de Ranking (CRUD - Update)	(Nuevo) Al finalizar una partida, el sistema debe actualizar (update) las estadísticas	S2

		(partidas_ganadas, partidas_perdidas) del Usuario.	
RF-11	Gestión de Torneos	(Nuevo) El sistema debe permitir a un usuario Administrador crear y gestionar Torneos en la base de datos.	S2
RF-12	Clasificación de Torneos	(Nuevo) El sistema debe almacenar una Clasificación de puntos para los jugadores dentro de un Torneo.	S2
RF-13	Historial de Movimientos	(Nuevo) El sistema debe registrar cada Movimiento (jugar, robar) de una partida en la base de datos para su futura consulta.	S2

3.2. Requisitos No Funcionales (RNF)

ID	Requisito	Descripción	Sprint
RNF-01	Usabilidad	La interfaz debe ser intuitiva, con acciones claras y accesibles (jugar/robar con un clic).	Todos
RNF-02	Fiabilidad	El sistema debe ser estable y mantener el estado de la partida de forma	S2 en adelante

		consistente.	
RNF-03	Rendimiento	Las acciones del juego deben reflejarse en la interfaz de todos los jugadores en menos de 1 segundo.	Futuro
RNF-04	Seguridad de Datos	(Nuevo) Las contraseñas de los usuarios deben almacenarse en la base de datos usando un hash (p.ej. bcrypt) y nunca en texto plano.	S2
RNF-05	Integridad de Datos	(Nuevo) La base de datos debe usar Claves Foráneas para asegurar la integridad referencial entre las tablas.	S2

3.3. Requisitos de la Interfaz de Usuario (UI)

(No hay cambios en esta sección para el Sprint 2)

- **UI-1 (Lobby):** Pantalla para crear o unirse a partidas.
- **UI-2 (Tablero de Juego):**
 - Área central para la carta jugada.
 - Zona inferior para la mano del jugador activo.
 - Indicadores para los oponentes (nombre y número de cartas).
 - Representación visual del mazo de robo.
- **UI-3 (Modal de Selección):** Ventana para elegir palo al jugar un "7".
- **UI-4 (Pantalla Final):** Muestra al ganador y al "perdedor".

4. (Nuevo) Modelo del Sistema (Diseño del Sprint 2)

Esta sección define la arquitectura técnica acordada durante la planificación del Sprint 2.

4.1. Modelo de la Base de Datos (E/R)

El sistema utilizará una base de datos relacional (MySQL) que sigue el modelo Entidad/Relación definido en el artefacto de diseño (diagrama_bd.plantuml).

Las entidades principales son:

- **Usuario:** Almacena jugadores y administradores, así como el ranking global.
- **Partida:** Almacena información de una sesión de juego, su estado y resultado.
- **Partida_Jugador:** Tabla puente que conecta Usuario y Partida (relación N:M).
- **Movimiento:** Almacena el historial de cada jugada de una partida.
- **Torneo:** Entidad para que los administradores creen eventos.
- **Clasificacion_Torneo:** Tabla que almacena los puntos de los jugadores en un torneo.

4.2. Diagrama de Clases (UML)

La lógica del servidor (backend) se estructurará siguiendo el diagrama de clases UML definido en el artefacto (diagrama_clases.plantuml).

Las clases principales son:

- **Partida:** Gestiona el estado de una sesión de juego.
- **RondaGame:** El "motor" de reglas (iniciado en Sprint 1).
- **DBControlador:** Clase estática que maneja todas las operaciones CRUD con la base de datos.
- **Usuario (Abstracta):** Clase base para los usuarios.
- **Jugador (Hereda de Usuario):** Representa a un jugador en el juego.
- **Administrador (Hereda de Usuario):** Representa a un administrador.
- **Torneo:** Clase para gestionar la lógica de los torneos.