

# Documento de Especificación de Requisitos Software (ERS)

**Proyecto:** Juego de Cartas — Ronda Marroquí

**Versión del Documento:** 3.0 (Sprint 3 - Implementación Técnica)

**Fecha:** 19 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 3.0 es documentar la arquitectura técnica final y los requisitos implementados durante el Sprint 3. Este sprint se centró en la implementación profunda de la base de datos y el backend para soportar la lógica compleja de partidas multijugador y el sistema de torneos.

### 1.2. Alcance del Sprint 3

El trabajo realizado abarca:

- **Base de Datos:** Implementación del esquema relacional completo (MySQL) para soportar partidas persistentes y torneos.
- **Backend:** Desarrollo de controladores (torneoController, partidaController) y rutas API para gestionar la lógica de negocio.
- **Frontend:** Implementación de las interfaces de sala-espera.html y torneos.html.
- **Integración:** Conexión del motor de juego RondaGame con la base de datos mediante serialización JSON.

## 3. Requisitos Específicos (Implementados en el Sprint 3)

### 3.1. Requisitos Funcionales (RF)

ID	Requisito	Descripción	Estado	Componente Técnico
RF-14	Gestión de Torneos	El sistema permite crear	Completado	torneoController.js

		torneos (eliminatorios/todos contra todos) y gestionar su estado.		
<b>RF-15</b>	Inscripción y Clasificación	Los usuarios pueden inscribirse en torneos y el sistema calcula sus puntos automáticamente.	Completado	Tabla Clasificacion_Torneo
<b>RF-16</b>	Sala de Espera Multijugador	Los usuarios pueden unirse a una partida en estado 'esperando' hasta llenar la capacidad.	Completado	sala-espera.html
<b>RF-17</b>	Persistencia de Partida	El estado exacto del juego (mazo, manos, turno) se guarda en la BD tras cada movimiento.	Completado	Campo estado_juego_json
<b>RF-18</b>	Historial de Movimientos	Se registra cada acción (jugar/robar) para auditoría y repetición.	Completado	Tabla Movimiento
<b>RF-19</b>	Navegación Integrada	Acceso fluido entre Perfil, Torneos y Sala de Espera.	Completado	Botones de navegación

### 3.2. Requisitos No Funcionales (RNF)

ID	Requisito	Descripción	Solución Técnica
RNF-08	Integridad Transaccional	Evitar sobrecapacidad en torneos cuando muchos usuarios se inscriben a la vez.	Uso de transacciones SQL con FOR UPDATE.
RNF-09	Serialización de Estado	Guardar objetos complejos de juego de forma eficiente.	Uso de columnas JSON/TEXT en MySQL.
RNF-10	Validación de Reglas	Las reglas de torneo (potencias de 2 para eliminatorios) deben validarse en el servidor.	Lógica en torneoController.js.

## 4. Especificaciones Técnicas del Sprint 3

### 4.1. Modelo de Base de Datos (Implementación Final)

El esquema de base de datos se amplió y migró (sprint3\_torneos.sql) para incluir:

- Tabla Partida (Modificada):**
  - estado: ENUM ('esperando\_jugadores', 'en\_curso', 'terminada', 'guardada').
  - estado\_juego\_json: TEXT (Almacena el objeto RondaGame serializado).
  - max\_jugadores: INT (Control de capacidad de la sala).
- Tabla Torneo (Nueva):**
  - Soporta configuración de tipo (eliminatorio/liga), cartas\_iniciales y max\_participantes.
- Tabla Clasificacion\_Torneo (Nueva):**
  - Tabla puente que vincula Usuario con Torneo y almacena los puntos.
- Tabla Movimiento:**
  - Registro inmutable de cada jugada (tipo\_movimiento, carta\_jugada, timestamp).

### 4.2. Arquitectura del Backend

Se añadieron nuevos módulos al servidor Express:

- **torneoController.js (504 líneas):** Contiene 9 funciones críticas, incluyendo crearTorneo, inscribirUsuario y generarClasificacion.
- **torneoRoutes.js:** Define 7 endpoints REST (ej. POST /api/torneos/:id/inscribir) protegidos por middleware de autenticación.
- **Lógica de Auto-Inicio:** El sistema detecta automáticamente cuando una sala de espera está llena para instanciar RondaGame y cambiar el estado a 'en\_curso'.

### 4.3. Interfaces de Usuario

- **torneos.html:** Panel de administración para crear torneos y vista de usuario para inscribirse y ver tablas de clasificación.
- **sala-espera.html:** Interfaz reactiva que muestra los jugadores conectados a una partida antes de su inicio.