

# **ERS (Especificación de Requisitos de Software): "La Rata"**

**Versión:** 1.0 (Sprint 1)

**Fecha:** 21 de octubre de 2025

## **1. Introducción**

### **1.1 Propósito**

Este documento define la Especificación de Requisitos de Software (ERS) para el proyecto "La Rata". El objetivo es crear una aplicación web multijugador que implemente las reglas del juego de cartas "La Rata". Este documento servirá como base para el diseño, desarrollo y pruebas del prototipo inicial.

### **1.2 Alcance del Producto**

El sistema permitirá a los usuarios registrarse, iniciar sesión, crear y unirse a partidas, jugar al juego "La Rata" siguiendo las reglas establecidas , y consultar un ranking de jugadores. El sistema también contará con un rol de Administrador para la gestión de usuarios y partidas.

### **1.3 Definiciones, Acrónimos y Abbreviations**

- **ERS:** Especificación de Requisitos de Software.
- **RF:** Requisito Funcional.
- **RNF:** Requisito No Funcional.
- **Sistema:** La aplicación web "La Rata".
- **Jugador:** Usuario registrado en el sistema.
- **Administrador (Admin):** Usuario con permisos para gestionar el sistema.
- **Ronda:** Un ciclo completo de apuestas y juego (ej. ronda de 5 cartas).
- **Mini ronda:** Una única jugada dentro de una ronda donde cada jugador tira una carta.

## **2. Descripción General**

### **2.1 Perspectiva del Producto**

El producto será una aplicación web *standalone* desarrollada con HTML, Javascript y CSS. Se comunicará con un servidor (no definido en este documento) para gestionar la lógica de juego multijugador y la persistencia de datos (Ajax).

### **2.2 Funciones del Producto**

- Gestión de usuarios (Registro, Login).
- Gestión de partidas (Crear, Unirse, Jugar).
- Implementación de la lógica del juego (Apuestas, Rondas, Penalizaciones).
- Visualización de historial de partidas y rankings.
- Administración del sistema.

### **2.3 Características del Usuario**

Existen dos tipos de usuarios:

- **Jugador:** Puede registrarse, iniciar sesión, unirse a partidas y jugar.
- **Administrador:** Puede iniciar sesión y tiene acceso a paneles para preparar torneos , gestionar usuarios y ver reglas.

### **2.4 Restricciones Generales**

- El desarrollo debe seguir la metodología SCRUM.
- El proyecto debe gestionarse mediante Git y GitHub.
- El prototipo de interfaz debe usar HTML/JS.
- El sistema debe ser compatible con los navegadores modernos.

### **2.5 Suposiciones y Dependencias**

- Las reglas definidas en Reglamento\_La\_Rata.pdf son finales.
- Los jugadores disponen de una conexión a internet estable.
- El sistema de "código de sala" es suficiente para que los jugadores se unan (no se requiere un lobby público).

### **3. Requerimientos Específicos**

#### **3.1 Requisitos Funcionales (RF)**

##### **Gestión de Usuarios y Roles**

- **RF-01:** El sistema debe permitir a un nuevo usuario registrarse proporcionando un nombre de usuario, email y contraseña.
- **RF-02:** El sistema debe permitir a un usuario (Jugador o Admin) iniciar sesión con sus credenciales.
- **RF-03:** El sistema debe diferenciar los permisos entre Jugadores y Administradores.
- **RF-04:** El Administrador debe poder consultar una lista de todos los usuarios (Jugadores) registrados.
- **RF-05:** El Administrador debe poder consultar el historial de partidas jugadas en el sistema.

##### **Gestión de Partida (Sala)**

- **RF-06:** Un Jugador debe poder "Crear una Partida", lo cual generará un código de sala único.
- **RF-07:** Un Jugador debe poder "Unirse a una Partida" introduciendo un código de sala.
- **RF-08:** El anfitrión (quien creó la partida) debe poder "Iniciar la Partida" cuando haya entre 3 y 6 jugadores en la sala.
- **RF-09:** Al iniciar la partida, el sistema debe asignar 5 vidas a cada jugador.
- **RF-10:** El sistema debe usar una baraja de póker virtual de 52 cartas (sin comodines).

##### **Lógica de Juego (Basado en Reglamento\_La\_Rata.pdf)**

- **RF-11:** El sistema debe gestionar el número de cartas repartidas en cada ronda, siguiendo la secuencia: 5, 4, 3, 2, 1, 2, 3, 4, 5, y repitiendo el ciclo 5-1-5.
- **RF-12:** El sistema debe solicitar las apuestas (victorias esperadas) a cada jugador en orden, comenzando por el jugador "primero".
- **RF-13:** El sistema debe validar la apuesta del último jugador, impidiendo que la suma total de apuestas sea igual al número de cartas repartidas en la ronda.

- **RF-14:** (Excepción RF-13) La regla de desempate de apuestas no se aplicará en la ronda de 1 carta.
- **RF-15:** En la ronda de 1 carta, la carta de un jugador no debe ser visible para él mismo.
- **RF-16:** En cada minironda, el sistema debe permitir al jugador "primero" jugar una carta , seguido por los demás en orden.
- **RF-17:** El sistema debe determinar al ganador de la minironda como el jugador que tiró la carta más alta (A > K > Q > J > 10... > 2).
- **RF-18:** En caso de empate en el valor de la carta (RF-17), el ganador de la minironda será el jugador que jugó la carta primero.
- **RF-19:** El jugador que inicia la primera minironda debe iniciar todas las minirondas de esa ronda.
- **RF-20:** Al final de la ronda, el sistema debe calcular la diferencia absoluta entre la apuesta de cada jugador y sus victorias. Si la diferencia es mayor que 0, restará esa cantidad de las vidas del jugador.
- **RF-21:** Si un jugador acierta exactamente su apuesta (diferencia 0), no perderá vidas.
- **RF-22:** El rol de "primero" (para apostar y jugar) debe rotar al jugador de la derecha en la siguiente ronda.
- **RF-23:** El sistema debe eliminar de la partida a cualquier jugador que llegue a 0 vidas.
- **RF-24:** El sistema debe declarar ganador al último jugador que quede con vidas.

#### **Historial y Rankings (Basado en L3\_Lab\_proyecto (1).pdf)**

- **RF-25:** El sistema debe almacenar un registro de todas las partidas jugadas , incluyendo los movimientos y el estado en cada momento.
- **RF-26:** El sistema debe mostrar una pantalla de "Ranking" donde los jugadores puedan ver las partidas ganadas y perdidas.
- **RF-27:** El sistema debe permitir que una partida inacabada pueda continuarse en un futuro, siempre que los participantes sean los mismos.

#### **3.2 Requisitos No Funcionales (RNF)**

- **RNF-01 (Compatibilidad):** El sistema será compatible con las versiones estables más recientes de navegadores Chrome, Firefox y Safari.

- **RNF-02 (Seguridad):** Las contraseñas de los usuarios deben almacenarse en la base de datos utilizando hashing SHA-256 con una sal (salt) única por usuario.
- **RNF-03 (Tecnología):** El prototipo de la interfaz de usuario debe desarrollarse utilizando HTML, CSS y Javascript.
- **RNF-04 (Tecnología):** Se recomienda el uso del framework Bootstrap para la maquetación del front-end.
- **RNF-05 (Usabilidad):** La interfaz de juego debe mostrar en todo momento: (a) las vidas de todos los jugadores, (b) la apuesta de la ronda actual de cada jugador, (c) las victorias actuales de cada jugador en la ronda, y (d) un indicador visual claro de quién es el turno.
- **RNF-06 (Rendimiento):** Las acciones del juego (ej. jugar una carta, hacer una apuesta) deben reflejarse en las pantallas de todos los demás jugadores en menos de 2 segundos bajo condiciones de red normales.