

Documento de Requerimientos Funcionales: "Cripto-Caballo" - Chess Arcade

Versión: 1.0

Fecha: Noviembre 2025

Proyecto: Plataforma Educativa y Lúdica de Ajedrez

1. Introducción

1.1 Propósito

El objetivo de este documento es definir los requerimientos funcionales y no funcionales para el desarrollo de la aplicación "Cripto-Caballo", un juego educativo de ajedrez que combina la lógica del movimiento del caballo con la criptografía (mensajes ocultos). La aplicación se dividirá en dos módulos principales: un **Back-office (Generador)** para el administrador y un **Front-end (Juego)** para los usuarios finales.

1.2 Alcance

El sistema permitirá al administrador crear rompecabezas personalizados basados en frases ocultas en tableros de ajedrez de diversos tamaños. Los usuarios finales podrán acceder a estos rompecabezas, intentar resolverlos interactuando con el tablero y recibir retroalimentación inmediata.

2. Descripción General del Sistema

El sistema se compone de dos módulos interconectados mediante una base de datos centralizada (Supabase).

2.1 Módulo Administrador (Generador)

Herramienta de uso exclusivo para el creador del contenido. Permite:

- Generar frases manuales o mediante IA.
- Configurar el tablero y la dificultad.
- Visualizar la solución.
- Guardar el nivel en la base de datos para su publicación.

2.2 Módulo Jugador (Cliente)

Interfaz pública donde los usuarios juegan. Permite:

- Seleccionar niveles disponibles.
- Interactuar con el tablero para resolver el criptograma.
- Recibir pistas visuales según la dificultad configurada.

3. Requerimientos Funcionales

Módulo 1: Administrador (Generador)

RF-01: Generación de Frases

- RF-01.1: El sistema debe permitir ingresar un mensaje manualmente.
- RF-01.2: El sistema debe integrar IA (Gemini) para generar frases temáticas (ajedrez, filosofía) o célebres automáticamente.
- RF-01.3: El sistema debe ofrecer una función de "Autocompletado IA" para rellenar frases cortas hasta alcanzar la longitud exacta del tablero.

RF-02: Configuración del Tablero

- RF-02.1: El administrador debe poder seleccionar dimensiones de tablero (3x4, 4x5, 5x5, 5x6, 6x7, 8x8).
- RF-02.2: El sistema debe validar que la longitud del mensaje no exceda la capacidad del tablero.
- RF-02.3: El sistema debe permitir configurar el relleno de casillas sobrantes (letras al azar, bloque sólido, X, vacío).

RF-03: Gestión de Niveles y Dificultad (Persistencia)

- RF-03.1: El sistema debe permitir catalogar el nivel (Fácil, Medio, Difícil, Experto).
- RF-03.2: El administrador debe poder configurar pistas iniciales:
 - *Checkbox*: Marcar casilla inicial (revelada).
 - *Checkbox*: Marcar casilla final (revelada).
 - *Selector*: Revelar N letras intermedias aleatorias.
- RF-03.3: El sistema debe permitir guardar el nivel generado en la base de datos (Supabase). Los datos a guardar incluyen:
 - ID del nivel.
 - Frase original.
 - Configuración del tablero (filas, columnas).
 - Ruta de solución (array de coordenadas).
 - Configuración de pistas.
 - Fecha de creación.

Módulo 2: Jugador (Cliente)

RF-04: Selección de Niveles

- RF-04.1: El usuario debe ver un listado o mapa de niveles disponibles, filtrables por dificultad.
- RF-04.2: El sistema debe indicar qué niveles ya han sido resueltos por el usuario (si hay sistema de cuentas o localStorage).

RF-05: Mecánica de Juego

- RF-05.1: El usuario debe poder seleccionar una casilla de inicio (si no está predefinida).
- RF-05.2: Al seleccionar una casilla, el sistema debe resaltar los movimientos legales de caballo posibles.

- **RF-05.3:** El usuario debe poder deshacer movimientos haciendo clic en la última casilla seleccionada.
- **RF-05.4:** El mensaje se debe ir construyendo visualmente a medida que el usuario avanza.

RF-06: Validación y Victoria

- **RF-06.1:** El sistema debe detectar automáticamente cuando el usuario completa la frase correcta.
- **RF-06.2:** Al ganar, se debe mostrar una animación de celebración (confeti, luces) y el mensaje completo formateado.
- **RF-06.3:** El sistema debe permitir al usuario ver el mensaje original con espacios y puntuación tras ganar.

4. Requerimientos No Funcionales

- **RNF-01: Responsividad:** La aplicación debe ser completamente funcional y estéticamente agradable en dispositivos móviles (Portrait/Landscape), Tablets y Escritorio. El tablero debe ajustarse automáticamente al viewport.
- **RNF-02: Performance:** La generación de rutas (algoritmo de Warnsdorff) debe ejecutarse en menos de 1 segundo para tableros estándar (hasta 8x8).
- **RNF-03: Usabilidad:** Las interacciones táctiles en móviles deben ser precisas y no interferir con el scroll de la página.
- **RNF-04: Seguridad:** La API Key de la IA no debe exponerse en el cliente final (Módulo Jugador); el Módulo Jugador solo debe consumir los datos ya generados de la base de datos.

5. Stack Tecnológico Sugerido

- **Frontend:** HTML5, CSS3 (Tailwind CSS), JavaScript (Vanilla o framework ligero).
- **Backend / Base de Datos:** Supabase (PostgreSQL + Realtime).
- **IA:** Google Gemini API (para generación de contenido en el módulo administrador).
- **Despliegue:** Vercel o Netlify.