

## **Análisis de requerimientos funcionales:**

### **1. Jugar ajedrez en modo un jugador:**

- **Descripción:**
- **Prioridad (método MoSCoW): Could Have**
- **Tiempo estimado:** 8 días
- **Dependencias:**
  - Motor de ajedrez **StockFish**.

### **2. Emparejamiento en línea:**

- **Descripción:** Implementar el emparejamiento automático entre jugadores.
- **Prioridad (método MoSCoW): Must Have**
- **Tiempo estimado:** 3 días
- **Dependencias:**
  - Algoritmo para buscar jugadores con un rango de ELO similar.
  - Gestión de jugadores en cola.
  - Almacenamiento de jugadores disponibles en tiempo real con su ID, ELO y modo de juego en una base de datos.
  - Uso de WebSocket para mantener la conexión activa mientras los jugadores están en cola de espera.

### **3. Reglas y validación del juego:**

- **Descripción:** Asegurar que todos los movimientos sigan las reglas oficiales del ajedrez.
- **Prioridad (método MoSCoW): Must Have**
- **Tiempo estimado:** 1 día
- **Dependencias:**
  - Uso de la librería de JavaScript “chess.js” para validar y verificar las reglas del ajedrez.
  - Uso de la librería de JavaScript “chessboard.js” para graficar el tablero de ajedrez.

### **4. Historial de partidas:**

- **Descripción:** Registrar y mostrar las partidas jugadas por los usuarios.
- **Prioridad (método MoSCoW): Should Have**
- **Tiempo estimado:** 8 días
- **Dependencias:**
  - Base de datos de partidas en programas como MySQL o PostgreSQL.

### **5. Modos de juego:**

- **Descripción:** Distintas modalidades y estilos de juego en el ajedrez.
- **Prioridad (método MoSCoW): Could Have**
- **Tiempo estimado:** 8 días
- **Dependencias:**
  - Librería “Chess.js” para aplicar lógica correspondiente al modo de juego.

#### **6. Sistema de clasificación:**

- **Descripción:** Jerarquización de jugadores en base a las partidas ganadas.
- **Prioridad (método MoSCoW): Should Have**
- **Tiempo estimado:** 1 día
- **Dependencias:**
  - Sistema de cálculo ELO para ajustar el ranking de los jugadores después de cada partida almacenado en base de datos relacional como PostgreSQL o MySQL.

#### **7. Chat en línea:**

- **Descripción:** Facilitar la comunicación entre jugadores en las partidas.
- **Prioridad (método MoSCoW): Should Have**
- **Tiempo estimado:** 3 días
- **Dependencias:**
  - Uso de WebSocket para garantizar un chat fluido y a tiempo real.
  - Uso de una base de datos relacional para almacenar mensajes en tiempo real.

#### **8. Registro de usuario e inicio de sesión:**

- **Descripción:** Permitir a los usuarios registrarse y autenticarse de forma segura.
- **Prioridad (método MoSCoW): Should Have**
- **Tiempo estimado:** 5 días.
- **Dependencias:**
  - Base de datos de usuarios en programas como PostgreSQL o MySQL.
  - Inicio de sesión usando autenticación compatible con CloudFlare integrando APIs de terceros como Google, Facebook entre otros.

### **Análisis de requerimientos No funcionales:**

#### **9. Rendimiento y tiempo de respuesta del servidor:**

- **Descripción:**
  - Acceso a las partidas y registros de estas de forma inmediata.
  - Respuesta rápida del servidor a los usuarios.
- **Prioridad (método MoSCoW): Must Have**
- **Tiempo estimado:** 5 días
- **Dependencias:**
  - Herramientas de prueba de carga como Apache JMeter o k6.
  - Código base funcional.

#### **10. Interfaz de usuario intuitiva:**

- **Descripción:** Diseño atractivo y fácil de usar.
- **Prioridad (método MoSCoW): Must Have**
- **Tiempo estimado:** 5 días
- **Dependencias:**
  - Diseño responsivo completado.
  - Herramientas de diseño como Figma o Adobe.

### 11. Compatibilidad:

- **Descripción:** Soporte para diferentes navegadores.
- **Prioridad (método MoSCoW): Could Have**
- **Tiempo estimado:** 5 días
- **Dependencias:**
  - Diseño responsivo.
  - Herramientas de pruebas de compatibilidad como BrowserStack.

### 12. Tolerancia a fallos:

- **Descripción:** Recuperar automáticamente partidas interrumpidas.
- **Prioridad (método MoSCoW): Should Have**
- **Tiempo estimado:** 3 días
- **Dependencias:**
  - Base de datos transaccional con persistencia configurada.

### 13. Soporte multiplataforma:

- **Descripción:** Asegurar la compatibilidad con dispositivos móviles y menos comunes.
- **Prioridad (método MoSCoW): Won't Have**
- **Tiempo estimado:** 8 días
- **Dependencias:**
  - Uso de framework o librerías CSS como **Bootstrap** o **Tailwind CSS** para crear interfaces adaptables a diferentes tamaños de pantalla.
  - Frameworks de frontend como React.