***Poker Royal: Juego de Poker Multijugador en Línea***

**Autor:** Agustín Álvarez Fijo  
**Tutor:** Pablo Rodriguez  
**Fecha de presentación:** Lunes 2 de Junio de 2025

**2. Resumen (Abstract)**

Poker Royal es una plataforma de póker en línea que permite partidas en tiempo real entre dos jugadores mediante WebSockets y Socket.io. El proyecto integra un frontend en React y un backend en Node.js (Express), con autenticación segura basada en JWT y almacenamiento de datos en MongoDB. Los usuarios pueden registrarse, activar su cuenta por correo electrónico y unirse a partidas mediante matchmaking o códigos de sala. Esta arquitectura garantiza baja latencia y robustez en la comunicación, así como una experiencia de usuario fluida. Se implementaron pruebas unitarias e integrales para verificar la lógica de negocio y la estabilidad de las conexiones, y se generó una demo desplegada en Netlify para demostrar la funcionalidad.

**3. Introducción**

**Contexto y motivación:**

• Creciente demanda de aplicaciones interactivas en tiempo real (juegos, chat, colaboración).

• Póker como juego de mesa popular, con gran comunidad online y necesidad de soluciones ligeras de emparejamiento.

**Objetivos:**

• Diseñar e implementar un sistema cliente-servidor que soporte partidas de póker 1vs1 en tiempo real.

• Garantizar seguridad en el proceso de autenticación y gestión de sesiones.

• Ofrecer una interfaz intuitiva y responsive.

**Estructura del documento:**

• Matriz DAFO

• Análisis de mercado

• Diseño y arquitectura

• Desarrollo y codificación

• Pruebas y validación

• Conclusiones y mejoras futuras

**4. Matriz DAFO**

|  |  |
| --- | --- |
| Fortalezas | Oportunidades |
| Comunicación en tiempo real con baja latencia. | Crecimiento de la industria de eSports y juegos web. |
| Autenticación segura con JWT y encriptación de contraseñas. | Integración futura con más plataformas (móvil, desktop). |
| Despliegue sencillo en Netlify y Heroku/MongoDB Atlas. | Extensión a torneos y ranking global. |

|  |  |
| --- | --- |
| Debilidades | Amenazas |
| Escalabilidad limitada a dos jugadores por partida. | Competencia de plataformas consolidadas (PokerStars). |
| Dependencia de un solo canal de correo para verificación. | Legislación sobre juegos de azar online. |
| Poca personalización de la interfaz. | Cambios en policies de Netlify/servicios en la nube. |

**5. Análisis de Mercado**

Existen múltiples soluciones de póker online, desde grandes plataformas (PokerStars, 888poker) hasta aplicaciones más ligeras. La propuesta de valor de Poker Royal radica en la simplicidad de emparejamiento 1vs1 y la agilidad de desarrollo con tecnologías JavaScript full-stack.

**6. Análisis y Diseño del Software**

6.1 Casos de uso clave:

Ver diagramas PlantUML en carpeta docs: casos de uso, clases y secuencia.

**7. Cronograma**

|  |  |  |  |
| --- | --- | --- | --- |
| Fase | Duración | Fecha inicio | Fecha fin |
| Planificación | 2 semanas | 01/12/2024 | 20/12/2024 |
| Diseño | 3 semanas | 15/02/2025 | 07/03/2025 |
| Desarrollo Frontend | 4 semanas | 08/03/2025 | 04/04/2025 |
| Desarrollo Backend | 4 semanas | 05/04/2025 | 02/05/2025 |
| Pruebas y depuración | 3 semanas | 03/05/2025 | 23/05/2025 |
| Despliegue | 1 semana | 24/05/2025 | 30/05/2025 |

**8. Problemas y Soluciones**

• Conexiones interrumpidas en Socket.io: reconexión automática y manejo de disconnect.

• Errores CORS: configuración de CORS en Express para aceptar peticiones desde el cliente.

• Entrega de correos fallida: cambio a proveedor SMTP fiable y manejo de errores.

**9. Codificación**

• Frontend: React con Create React App, componentes funcionales y hooks.

• Backend: Node.js (Express), rutas REST, modelos Mongoose para MongoDB.

• Comunicación: Socket.io (Server y Cliente).

• Gestión de estado: Context API y hooks personalizados.

**10. Interfaces de Usuario**

• Pantalla de login/registro

• Lobby con matchmaking y salas

• Tabla de juego con cartas SVG importadas

• Notificaciones en tiempo real

**11. Pruebas**

• Unitarias: Jest y Supertest para APIs.

• Integración: Simulación de partidas automatizadas.

• Manuales: Pruebas de usuario en demo desplegada.

**12. Futuras mejoras**

• Soporte para más de 2 jugadores.

• Torneos y clasificación mundial.

• Aplicación móvil con React Native.

• Chat integrado y emojis en tiempo real.

**13. Bibliografía**

• Documentación oficial de React.js.

• Docs de Socket.io y Express.

• Guías de MongoDB Atlas y Mongoose.

**14. Anexos**

• Código fuente en GitHub: <https://github.com/agustin-alvarez-fijo/poker-royal>

• Diagramas UML en carpeta docs/