

Área de Conocimiento de Tecnología de la Información y Comunicación

"SisteMa Web de gestión de torneos de ajedrez para la Academia "Estrellas del Ajedrez"

Trabajo Monográfico para optar al título de Ingeniero en Computación

Elaborado por:

Sr. Engel Antonio
Largaespada Vargas
Carnet: 2019-0328U

Sr. Laisha Nazaret
Acevedo García
Carnet: 2019-1161U

Tutor:

Msc. Luis Eduardo
Chávez Mairena



Secretaría Académica
DACTIC

SECRETARIA DE ÁREA ACADÉMICA

F-8: CARTA DE FINALIZADO PLAN DE ASIGNATURA

El Suscrito Secretario del **ÁREA DE CONOCIMIENTO DE TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIÓN** hace constar que:

LARGAESPADA VARGAS ENGEL ANTONIO

Carné: **2019-0328U** Turno: **Diurno** Plan de Asignatura: **2015** de conformidad con el Reglamento Académico vigente en la Universidad, ha aprobado todas las asignaturas correspondientes a la carrera de **INGENIERÍA EN COMPUTACIÓN**, en el año 2023 y solo tiene pendiente la realización de una de las formas de culminación de estudio.

Se extiende la presente **CARTA DE FINALIZADO PLAN DE ASIGNATURA**, a solicitud del interesado en la ciudad de Managua, a los veinte y nueve días del mes de octubre del año dos mil veinte y cinco.

Atentamente,

MSc. Luisa Massiel Mercado Gutiérrez
SECRETARIO DE ÁREA ACADÉMICA



📞 Móvil: (505) 83803517

📍 Recinto Universitario Simón Bolívar
Avenida Universitaria.
Managua, Nicaragua.
Apdo: 5595

IMPRESO POR SISTEMA DE REGISTRO ACADÉMICO EL 29-oct-2022 hazzely.orozco@dactic.uni.edu.ni
 www.uni.edu.ni



Área de Conocimiento de
Tecnología de la Información
y Comunicación

SECRETARIA DE ÁREA ACADÉMICA

F-8: CARTA DE FINALIZADO PLAN DE ASIGNATURA

El Suscrito Secretario del ÁREA DE CONOCIMIENTO DE
TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIÓN hace constar que:

ACEVEDO GARCÍA LAISHA NAZARET

Carné: 2019-1161U Turno: Diurno Plan de Asignatura: 2015 de
conformidad con el Reglamento Académico vigente en la
Universidad, ha aprobado todas las asignaturas correspondientes
a la carrera de **INGENIERÍA EN COMPUTACIÓN**, en el año 2024 y solo
tiene pendiente la realización de una de las formas de
culminación de estudio.

Se extiende la presente **CARTA DE FINALIZADO PLAN DE
ASIGNATURA**, a solicitud del interesado en la ciudad de Managua,
a los catorce días del mes de mayo del año dos mil veinte y
cinco.

Atentamente,



HAZZELY DEL CARMEN OROZCO MIRANDA
SECRETARIO DE ÁREA ACADÉMICA



Móvil: (505) 8588 8333



Recinto Universitario Simón Bolívar
Avenida Universitaria.
Managua, Nicaragua.
Apdo: 5595



www.uni.edu.ni



Programa Académico
de Ingeniería en
Computación

Managua, 03 de noviembre 2025

MSc. Claudia Lucía Benavidez Rugama.
Directora DACTIC.

Sus manos.

Estimada MSc. Benavidez.

Reciba cordiales saludos y deseos de éxito en el desempeño de sus funciones.

Por este medio, nos permitimos solicitar la autorización para realizar la defensa del trabajo monográfico titulado:

“Sistema web de gestión de torneos de ajedrez para la academia Estrellas del Ajedrez”, trabajado por los bachilleres:

- Laisha Nazaret Acevedo García con número de carnet 2019-1161U
- Engel Antonio Largaespada Vargas con número de carnet 2019-0328U

Agradezco, su pronta gestión de asignar local, hora y fecha de la defensa, me despido deseándole logros en el desempeño de sus funciones.

Atentamente,

MSc. Luis Eduardo Chavez Mairena
Docente Titular Dactic.
Tutor Monográfico.

Cc: Archivo



Área de Conocimiento de Tecnología de la Información y Comunicación

Managua, 28 de noviembre 2024

Bachilleres

Br. Engel Antonio Largaespada Vargas 2019-0328U

Br. Laisha Nazaret Acevedo García 2019-1161U

Egresados Programa académico Ingeniería en Computación
Sus manos.

Estimados Bachilleres:

Reciban cordiales saludos de mi parte y deseándole el mejor de los éxitos en sus actividades diarias.

Por medio de la presente, les comunico la aprobación e inscripción del Protocolo de trabajo monográfico, titulado: **“Sistema web de gestión de torneos de ajedrez para la Academia “Estrellas del ajedrez”**, el cual cumple con los requisitos establecidos en el capítulo II de la normativa para los trabajos Monográficos de la Universidad Nacional de Ingeniería (UNI) como forma de culminación de estudios.

No omito manifestar, que el Maestro **MSc. Luis Eduardo Chávez Mairena**, es el docente encargado de acompañarlos con responsabilidad y compromiso como tutor en el proceso de desarrollo de la monografía.

Así mismo, en correspondencia con la Normativa para los Trabajos Monográficos, a partir de la fecha de aprobación e inscripción tendrán un máximo de 12 meses para la ejecución y entrega de los ejemplares para la organización de la pre-defensa y defensa del mismo.

Sin más a que hacer referencia, les deseo el mejor de los éxitos en la culminación de esta etapa, les saludo.

Atentamente,

Claudia Benavides
MSc. Claudia Benavides Rugama
Directora Área de Conocimiento de
Tecnología de la Información y Comunicación



CC: MSc. Luis Eduardo Chávez Mairena – Tutor
MSc. Cedrick DallaTorre Parrales – Secretario Académico
Archivo DACTIC 2024

Móvil: (505) 8588 8333

Recinto Universitario Simón Bolívar
Avenida Universitaria.
Managua, Nicaragua.
Apdo: 5595

www.uni.edu.ni

DEDICATORIA

Dedicamos este trabajo a todas aquellas personas que fueron parte fundamental de nuestro caminar académico y trayectoria personal.

Engel Antonio Largaespada Vargas: A mis padres, por ser el pilar de mi vida y mi apoyo incondicional; a mi coach, Escarleth Rivas, por guiarme y motivarme en todo momento; a mis amigos, por apoyarme y acompañarme en esta travesía; y en especial a la memoria de mi fiel compañera Lucy.

Laisha Nazaret Acevedo García: A Dios todopoderoso, quien me fortaleció en toda mi carrera y siempre recordó que en Josué 1:9 dice: "Mira que te mando que te esfuerces y seas valiente; no temas ni desmayes, porque Jehová tu Dios estará contigo dondequiera que vayas". A mi madre, por brindarme su amor incondicional, a mis amigos por apoyarme en este camino; a mi tía Luisa, que fue parte fundamental en mi crecimiento; y a la memoria de mi tía Amelia, quien también fue mi madre, mi maestra y dejó huellas imborrables en mi vida.

De la misma manera, reconocemos con gratitud a nuestro tutor, **Luis Eduardo Chávez Mairena**, quien guió, aconsejó y compartió sus valiosos conocimientos con nosotros. Rasgos que permitieron la culminación de este trabajo monográfico. Sin su orientación y su entrega, el presente trabajo no habría sido posible.

A todos ustedes, nuestro más sincero agradecimiento.

RESUMEN

Este trabajo monográfico presenta el desarrollo de un Sistema Web de Gestión de Torneos de Ajedrez desarrollado para modernizar y optimizar las operaciones de la Academia Estrellas del Ajedrez. El proyecto nace de la necesidad de superar los métodos tradicionales de organización, proponiendo una solución integral y tecnológica que centraliza en una sola plataforma la gestión de torneos, el registro de jugadores y el análisis del rendimiento.

El sistema se construyó utilizando un stack tecnológico moderno y robusto (Laravel 12, PHP 8.2+ y MySQL), priorizando la escalabilidad, la seguridad y una experiencia de usuario intuitiva. Para garantizar un acceso seguro y personalizado, se implementó un módulo de autenticación que incluye roles de usuario y recuperación de contraseña, simplificando así el primer contacto con la plataforma.

El sistema web fue desarrollado pensando en las necesidades reales de una academia. Permite gestionar de manera eficiente el perfil de cada ajedrecista, almacenando su información y su evolución durante el desarrollo de los torneos, lo que facilita un seguimiento personalizado de su trayectoria.

El corazón del sistema reside en su capacidad para organizar torneos de manera flexible y automática. Soporta los formatos más populares (suizo, Round-Robin, Eliminación Directa y por equipos), y se encarga de manera inteligente de los complejos emparejamientos, equilibrando colores y evitando enfrentamientos repetitivos, lo que libera a los organizadores de cálculos manuales propensos a errores.

Más allá de la organización, el sistema web ofrece valor añadido al integrar un módulo de análisis de partidas mediante archivos PGN. Este módulo transforma los datos de cada juego en métricas comprensibles, identificando aciertos, errores y oportunidades de aprendizaje, lo que lo convierte en una herramienta pedagógica poderosa para instructores y alumnos.

La interfaz, desarrollada con Blade, Bootstrap y Tailwind CSS, es responsive y ofrece dashboards claros con la información más relevante disponible de un vistazo.

Finalmente, este sistema web no es solo una herramienta de software, sino una plataforma estratégica que permite a academias y clubes de ajedrez profesionalizar su gestión, enriquecer la experiencia formativa de sus miembros y potenciar el desarrollo del ajedrez competitivo mediante la tecnología.

ÍNDICE

I. INTRODUCCIÓN	1
II. ANTECEDENTES	2
III. JUSTIFICACIÓN	4
IV. OBJETIVOS	6
4.1. Objetivos generales	6
4.2. Objetivos específicos	6
V. MARCO TEÓRICO	7
5.1. Ajedrez	7
5.2. Notación algebraica del ajedrez	8
5.3. Sistemas de competencias del ajedrez	9
5.3.1. Por eliminación o KO	9
5.3.2. Round-Robin	11
5.3.3. Sistema suizo	11
5.4. Sistemas de desempate	12
5.4.1. Partidas de desempate (Play-offs)	12
5.4.2. Valoración promedio de los oponentes	13
5.4.3. Sistema Buchholz	13
5.4.4. Resultado particular o confrontación directa	14
5.4.5. Sistema Koya	14
5.4.6. Número de partidas con las piezas negras	14
5.4.7. Sonnerborn-Berger	14

5.5.	ELO	14
5.6.	Análisis de partidas.....	15
5.7.	Herramientas de desarrollo.....	15
5.7.1.	Visual studio code.....	15
5.7.2.	PhpMyAdmin	15
5.7.3.	HTML.....	16
5.7.4.	CSS	16
5.7.5.	Javascript	16
5.7.6.	PHP	16
5.7.7.	Laravel.....	16
5.8.	Metodología en Cascada	17
VI. ANÁLISIS Y PRESENTACIÓN DE LOS RESULTADOS		18
6.1.	Diseño metodológico	18
6.1.1.	Requisitos.....	18
6.1.2.	Diseño	19
6.1.3.	Implementación	20
6.1.4.	Verificación	21
6.1.5.	Mantenimiento	22
6.2.	Fase de requisitos	22
6.2.1.	Requerimientos funcionales.....	23
6.2.2.	Requerimientos no funcionales.....	36

6.3. Fase de diseño	37
6.3.1. Arquitectura del sistema – Clean arquitecture.....	37
6.4. Fase de desarrollo	78
6.4.1. Preparación del entorno de desarrollo	78
6.4.2. Implementación de los Módulos.....	79
6.4.3. Arquitectura del Sistema.....	100
6.4.4. Optimizaciones y Rendimiento.....	105
6.4.5. Integración y control de versiones.....	109
6.5. Fase de Pruebas y Validación	112
6.5.1. Objetivos de la Fase de Pruebas	114
6.5.2. Estrategia de Pruebas	114
6.5.3. Casos de Prueba Implementados	114
6.5.4. Resultados de las Pruebas	122
6.5.5. Validación de Requisitos.....	124
6.5.6. Certificación de Calidad	125
VII. CONCLUSIÓN	126
VIII. RECOMENDACIONES	127
IX. BIBLIOGRAFÍA	128
X. ANEXOS	131
10.1. Diccionario de datos	131
10.2. Matriz de pruebas de caja negra.....	144

10.3. Resultados de encuesta de satisfacción de usuario.....	148
10.4. Estudio de Factibilidad.....	156
10.4.1. Factibilidad Técnica	156
10.4.2. Factibilidad Económica	156
10.4.3. Factibilidad Operativa	161
10.4.4. Estudio de Factibilidad Legal	162

ÍNDICE DE FIGURAS

Figura 1: Colocación inicial de las piezas de ajedrez	7
Figura 2: Numeración Algebraica para las 64 casillas	8
Figura 3: Notación de las piezas	9
Figura 4: Ejemplo de eliminación directa	10
Figura 5: Ejemplo de eliminación doble	11
Figura 6: Ejemplo de sistema suizo	12
Figura 7: Gráfico de las fases de la Metodología de desarrollo Cascada	18
Figura 8: Arquitectura patrón en capas	19
Figura 9: Arquitectura general del sistema	37
Figura 10: Diagrama de clases del sistema	39
Figura 11: Diagrama de caso de uso No. 1	46
Figura 12: Diagrama de caso de uso No. 2	47
Figura 13: Diagrama de caso de uso No. 3	49
Figura 14: Diagrama de caso de uso No. 4	50
Figura 15: Diagrama de caso de uso No. 5	51
Figura 16: Diagrama de caso de uso No. 6	52
Figura 17: Diagrama de caso de uso No. 7	53
Figura 18: Diagrama de actividades No. 1	54
Figura 19: Diagrama de actividades No. 2	55
Figura 20: Diagrama de actividades No. 3	56
Figura 21: Diagrama de actividades No. 4	57
Figura 22: Diagrama de actividades No. 5	58
Figura 23: Diagrama de actividades No. 6	59
Figura 24: Diagrama de secuencia No. 1	60
Figura 25: Diagrama de secuencia No. 2	61

Figura 26: Diagrama de secuencia No. 3	62
Figura 27: Diagrama de secuencia No. 4	63
Figura 28: Diagrama de secuencia No. 5	64
Figura 29: Diagrama de secuencia No. 6	65
Figura 30: Diagrama de secuencia No. 7	66
Figura 31: Diagrama de secuencia No. 8	67
Figura 32: Diagrama de secuencia No. 9	68
Figura 33: Diagrama de secuencia No. 10	69
Figura 34: Diagrama de Interfaces de Repositorio	72
Figura 35: Diagrama de Implementaciones de Repositorio	73
Figura 36: Diagrama de Arquitectura de Software.....	74
Figura 37: Diagrama de Base de Datos	75
Figura 38: Wireframe de visualización de registros	77
Figura 39: Interfaz de inicio de sesión con credenciales incorrectas	80
Figura 40: Interfaz de recuperación de contraseña	81
Figura 41: Correo para el restablecimiento de la contraseña	82
Figura 42: Interfaz del módulo de gestión de usuarios	83
Figura 43: Formulario del módulo de gestión de usuarios	84
Figura 44: Formulario del módulo de gestión de permisos	85
Figura 45: Interfaz del módulo de gestión de torneos.....	86
Figura 46: Formulario para la gestión de torneos	87
Figura 47: Interfaz del módulo para la gestión de participantes en un torneo	88
Figura 48: Interfaz del módulo para la gestión de partidas	89
Figura 49: Interfaz del módulo de gestión de miembros	90
Figura 50: Formulario del módulo de gestión de miembros	91
Figura 51: Interfaz del módulo de gestión de academias	93

Figura 52: Formulario del módulo de gestión de academias	94
Figura 53: Interfaz del módulo de participantes con la función de Generar Emparejamiento	95
Figura 54: Interfaz del módulo de auditorías	96
Figura 55: Dashboard principal del sistema	98
Figura 56: Gráficos de desempeño y actividad competitiva.....	99
Figura 57: Patrón MVC del sistema.....	100
Figura 58: Servicios principales del negocio/	102
Figura 59: Middleware implementado en el sistema.....	103
Figura 60: Helpers implementados.....	104
Figura 61: Protección Automática contra Inyección SQL.....	109
Figura 62: Enviroment del sistema desplegado desde Laravel cloud.	111
Figura 63: Variables de entorno personalizadas.	112
Figura 64: Base de datos con las tablas utilizadas para las pruebas.....	113
Figura 65: Tablas con torneos creados	115
Figura 66: Formulario de edición de torneo.....	116
Figura 67: Torneo cancelado	117
Figura 68: Modal de confirmación para eliminación de torneo.....	117
Figura 69: Modal de incorporación de participantes	118
Figura 70:Clasificación de participantes	118
Figura 71: Resultados obtenidos en un torneo	119
Figura 72: Emparejamiento de una ronda	120
Figura 73: Interfaz de análisis de una partida.....	120
Figura 74: Resultado de rendimiento	121
Figura 75: Pruebas de rendimiento terminal.....	121
Figura 76: Pruebas de rendimiento en el sistema desplegado.	122
Figura 77: Tiempo de respuesta por consulta	122

Figura 78: Cadenas maliciosas utilizadas para el test de inyección SQL	123
Figura 79: Fragmento de código para las pruebas de inyección SQL	123
Figura 80: Resultados de pruebas de inyección SQL.....	124
Figura 81: Diccionario de datos del sistema 1	131
Figura 82: Diccionario de datos del sistema 2.....	132
Figura 83: Diccionario de datos del sistema 3.....	133
Figura 84: Diccionario de datos del sistema 4	134
Figura 85: Diccionario de datos del sistema 5.....	135
Figura 86: Diccionario de datos del sistema 6	136
Figura 87: Diccionario de datos del sistema 7	137
Figura 88: Diccionario de datos del sistema 8	138
Figura 89: Diccionario de datos del sistema 9	139
Figura 90: Diccionario de datos del sistema 10	140
Figura 91: Diccionario de datos del sistema 11	141
Figura 92: Diccionario de datos del sistema 12	142
Figura 93: Diccionario de datos del sistema 13	143
Figura 94: Matriz de pruebas de caja negra implementadas 1	144
Figura 95: Matriz de pruebas de caja negra implementadas 2	145
Figura 96: Matriz de pruebas de caja negra implementadas 3	146
Figura 97: Matriz de pruebas de caja negra implementadas 4	147
Figura 98: Encuesta de satisfacción de usuario	148
Figura 99: Resultados de encuesta de satisfacción 1.....	148
Figura 100: Resultados de encuesta de satisfacción 2.....	149
Figura 101: Resultados de encuesta de satisfacción 3.....	149
Figura 102: Resultados de encuesta de satisfacción 4.....	150
Figura 103: Resultados de encuesta de satisfacción 5.....	150

Figura 104: Resultados de encuesta de satisfacción 6.....	151
Figura 105: Resultados de encuesta de satisfacción 7.....	151
Figura 106: Resultados de encuesta de satisfacción 8.....	152
Figura 107: Resultados de encuesta de satisfacción 9.....	152
Figura 108: Resultados de encuesta de satisfacción 10.....	153
Figura 109: Resultados de encuesta de satisfacción 11.....	153
Figura 110: Resultados de encuesta de satisfacción 12.....	154
Figura 111: Resultados de encuesta de satisfacción 13.....	154
Figura 112: Resultados de encuesta de satisfacción 14.....	155

ÍNDICE DE TABLAS

Tabla 1: Requerimiento funcional No. 1	23
Tabla 2: Requerimiento funcional No. 2	24
Tabla 3: Requerimiento funcional No. 3	25
Tabla 4: Requerimiento funcional No. 4	26
Tabla 5: Requerimiento funcional No. 5	27
Tabla 6: Requerimiento funcional No. 6	28
Tabla 7: Requerimiento funcional No. 7	29
Tabla 8: Requerimiento funcional No. 8	31
Tabla 9: Requerimiento funcional No. 9	32
Tabla 10: Requerimiento funcional No. 10	33
Tabla 11: Requerimiento funcional No. 11	34
Tabla 12: Requerimiento funcional No. 12	35
Tabla 13: Servicios de mantenimiento e infraestructura	157
Tabla 14: Costos de desarrollo	158
Tabla 15: Costos totales	159

I. INTRODUCCIÓN

El ajedrez es una disciplina milenaria que combina estrategia, lógica y concentración. En Nicaragua, su práctica se remonta a mediados del siglo XX, cuando comenzaron a surgir clubes y torneos en ciudades como Managua y León. Desde entonces, el ajedrez ha ganado terreno en el ámbito educativo y competitivo, impulsado por instituciones como la Federación Nicaragüense de Ajedrez (FENICA), fundada en 1972.

Actualmente, se estima que en el país se realizan más de 200 torneos de ajedrez al año, entre eventos escolares, universitarios, abiertos nacionales y competencias federadas. Solo en Managua se organizan más de 50 torneos formales anualmente, con un promedio de 30 a 40 participantes por evento. A nivel nacional, existen aproximadamente 20 academias activas, entre ellas la Academia Carlos Fonseca, la Julio Ramírez de Arellano y Estrellas del Ajedrez.

En el escenario nacional del ajedrez, la academia Estrellas del Ajedrez destaca como una iniciativa formativa comprometida con el desarrollo del talento juvenil. Fundada por el Maestro FIDE (MF) y FIDE Instructor (FI) Steven Sánchez, la academia ha logrado posicionarse como un espacio de aprendizaje y competencia en Ciudad Sandino, promoviendo la participación activa de sus estudiantes en torneos locales y nacionales. Su enfoque combina la enseñanza técnica con la experiencia práctica, fomentando el crecimiento integral de los ajedrecistas desde etapas iniciales. Esta trayectoria convierte a Estrellas del Ajedrez en un referente dentro del ecosistema ajedrecístico nicaragüense y en un caso representativo para el análisis y aplicación de soluciones tecnológicas orientadas a la gestión de torneos en contextos formativos.

Este documento está estructurado en tres secciones principales: la introducción, donde se presenta el contexto general del ajedrez en Nicaragua y el problema de gestión de torneos; los antecedentes, que describen la situación actual de la academia Estrellas del Ajedrez y las limitaciones de las herramientas existentes; y la justificación, donde se detallan los beneficios del sistema web propuesto.

II. ANTECEDENTES

La academia Estrellas del Ajedrez pretende inspirar y promover el desarrollo del talento ajedrecístico nicaragüense, siendo el puente que convierte en realidad los sueños y las metas de los ajedrecistas nacionales, la academia está compuesta por MF y FI Steven Sánchez ubicada en Ciudad Sandino Oro verde, del restaurante TipTop 1 Cuadra al norte 7 Cuadras al Este 75 varas al Norte con número de teléfono: 82861396.

Actualmente, la academia no cuenta con un sistema interno que permita gestionar de manera eficiente la información de sus deportistas ni los torneos que organiza. La administración se realiza de forma manual, utilizando hojas de cálculo para registrar datos personales, estadísticas de rendimiento y resultados de competencias. Este método, aunque funcional en etapas iniciales, se vuelve ineficiente ante el crecimiento sostenido de su comunidad, que mantiene un promedio de 15 a 20 competidores activos. La falta de una plataforma digital limita la capacidad de análisis, seguimiento y planificación estratégica de la academia.

Existen herramientas como Swiss-Manager, un software especializado en la administración y emparejamiento de torneos de ajedrez. Este programa tiene un costo único, sin pagos mensuales ni anuales. La versión light cuesta 75€ y permite hasta 60 participantes en formato Round Robin, con limitaciones en la modalidad por equipos. La versión completa cuesta 150€ y permite hasta 900 jugadores y 46 rondas. Sin embargo, su uso está restringido a árbitros avalados por la FIDE, lo que limita su implementación en academias formativas que no cuentan con personal certificado. (herzog, 2014).

Estas capacidades exceden considerablemente las necesidades de una academia de escala local, haciendo que la inversión no sea proporcional al uso requerido para la gestión de torneos internos y formativos.

Por otra parte, Chess-result, que es un poderoso servidor dedicado exclusivamente a resultados de ajedrez; dentro de ella se pueden buscar torneos, jugadores y partidas (Herzog, 2002). Este servidor trabaja de la mano de Swiss Manager, por ende, se pueden ver/exportar/imprimir información detallada del torneo. Aunque es una herramienta poderosa para la difusión de resultados, su uso también está restringido a árbitros FIDE, y solo permite la publicación de torneos nacionales e internacionales oficialmente avalados. Esta condición excluye a academias como Estrellas del Ajedrez, donde la mayoría de los alumnos son principiantes sin rating FIDE y los torneos son de carácter local o formativo.

En la universidad católica del Perú se realizó el análisis, diseño e implementación de un sistema administrador de torneos de ajedrez que se propuso como solución a la falta de una herramienta adecuada que administre de forma eficiente este tipo de eventos. Cabe indicar que es de uso privado y se optó por implementar un procedimiento de emparejamiento de jugadores a lo largo de las rondas de un torneo en el sistema suizo que mejora el tiempo de respuesta en la generación de las parejas según los procedimientos establecidos por los reglamentos de la FIDE. (Colonia Machado, 2009) Sin embargo, estas soluciones no están adaptadas a la escala y necesidades específicas de academias nicaragüenses como Estrellas del Ajedrez, que requieren gestionar eficientemente sus grupos de alumnos en torneos internos y locales.

En Nicaragua, el desarrollo de herramientas tecnológicas aplicadas al ajedrez ha sido limitado, especialmente en el ámbito de la gestión de torneos. En la Universidad Nacional de Ingeniería (UNI) se identificó una aplicación móvil orientado a la enseñanza del ajedrez, enfocado en el fortalecimiento de habilidades técnicas y pedagógicas del deporte (Granados, 2020). No obstante, dicho trabajo no aborda la problemática de la administración de torneos. A pesar de haberse realizado una revisión bibliográfica en dicha institución, no se encontró evidencia de propuestas que respondan a las necesidades de gestión interna en academias locales.

III. JUSTIFICACIÓN

La gestión eficiente de torneos de ajedrez representa un desafío constante para academias formativas como Estrellas del Ajedrez, especialmente en contextos donde los recursos tecnológicos son limitados. Aunque actualmente se utilizan hojas de cálculo para registrar información de jugadores y resultados, este método presenta dificultades para mantener la coherencia, la actualización y el análisis de datos en tiempo real, especialmente cuando se organizan múltiples eventos en distintas modalidades y ubicaciones.

La ausencia de una plataforma digital especializada no solo ralentiza los procesos administrativos, sino que también limita la capacidad de generar informes, evaluar el rendimiento de los jugadores y planificar estratégicamente futuras competencias. En academias donde el enfoque está en la formación continua, contar con herramientas que faciliten el seguimiento del progreso individual y colectivo es fundamental para mejorar la calidad del entrenamiento y la toma de decisiones.

El sistema web propuesto busca responder directamente a estas necesidades, ofreciendo una solución integral que automatiza y centraliza la gestión de torneos. Entre sus principales beneficios se encuentran:

- Registro automatizado de participantes y equipos, con perfiles individuales que permiten visualizar el historial competitivo de cada jugador.
- Gestión flexible de modalidades de juego y horarios, adaptada a torneos suizos, Round Robin, por equipos y otras variantes.
- Historial de resultados y generación de informes detallados, que facilita el análisis estadístico y la retroalimentación técnica.

- Análisis de partidas y cálculo automatizado de sistemas de desempate, conforme a los reglamentos de la FIDE, lo que garantiza transparencia y equidad en la competencia.

Además de optimizar los procesos internos, este sistema representa una herramienta accesible y escalable para academias que no cuentan con árbitros certificados ni con los recursos para adquirir software especializado. Su implementación permitirá profesionalizar la organización de torneos, reducir errores humanos, y fortalecer el desarrollo del ajedrez en Nicaragua desde una perspectiva formativa, tecnológica y comunitaria.

IV. OBJETIVOS

4.1. Objetivos generales

- Desarrollar un sistema web de gestión de torneos de ajedrez para la academia "Estrellas del ajedrez".

4.2. Objetivos específicos

- Definir las necesidades, funcionalidades y requerimientos esenciales del sistema web de gestión de torneos de ajedrez, en colaboración con la Academia "Estrellas del Ajedrez".
- Estructurar la arquitectura del sistema mediante la identificación y organización de los módulos de gestión, torneos, tablas y resultados, garantizando así el funcionamiento eficiente y coherente del sistema.
- Codificar el sistema web utilizando las tecnologías HTML5, CSS3, JavaScript, PHP 8.2, MySQL, Laravel 12, Bootstrap y Tailwind CSS, para su desarrollo según los requerimientos establecidos.
- Implementar pruebas de caja negra en el sistema web, garantizando su funcionalidad, eficacia y calidad.

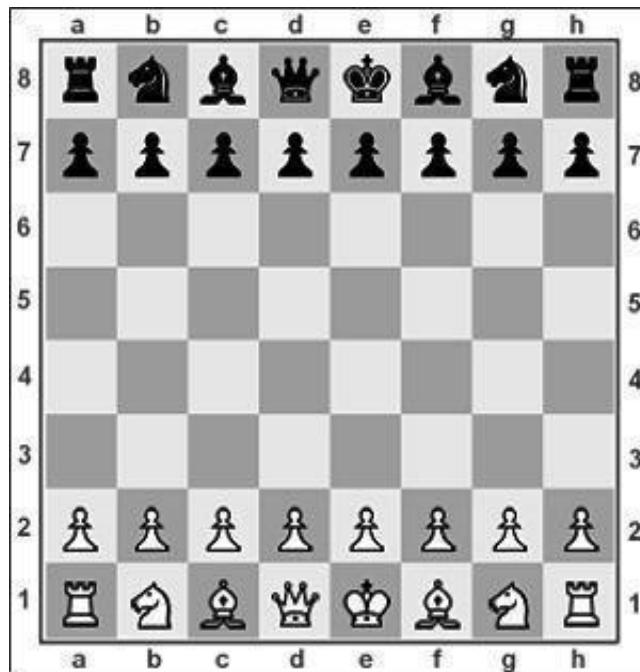
V. MARCO TEÓRICO

A continuación, se profundizará en conceptos fundamentales que permitirán una mejor comprensión al momento del desarrollo del sistema, donde se abordarán definiciones del ajedrez y las herramientas que se utilizarán para llevar a cabo este proyecto.

5.1. Ajedrez

El Ajedrez es un juego de mesa para dos jugadores, se utiliza un tablero de juego compuesto de un cuadrado formado por 64 casillas en 8 filas y 8 columnas, las casillas son de dos colores y están alternadas por color, cada jugador al inicio del juego cuenta con 16 piezas. (Ubalgo, 2018)

Figura 1: Colocación inicial de las piezas de ajedrez



Cada bando se compone de un rey, una dama o reina, dos alfiles, dos caballos, dos torres y ocho peones. Gana el jugador que primero consigue atacar y acorralar al rey del adversario sin que éste tenga escapatoria, lo que se conoce como jaque mate.

5.2. Notación algebraica del ajedrez

La notación algebraica es el lenguaje que hace posible expresar cualquiera de las jugadas reglamentarias de una partida de ajedrez.

Basado en un sistema desarrollado por Philipp Stamma, el sistema de **notación Algebraica** usa las coordenadas de la cuadrícula 8x8 del tablero, se ha convertido en la norma usada por la Federación Internacional de Ajedrez y es usado por la mayoría de los jugadores. Cada columna tiene asignada una letra minúscula de izquierda a derecha: **[a b c d e f g h]** y cada fila es nombrada de abajo hacia arriba con los números: **[1 2 3 4 5 6 7 8]**. Las blancas se colocan en las filas 1 y 2 y las negras en la 7 y 8. (123ajedrez.com, s.f.)

Figura 2: Numeración Algebraica para las 64 casillas

	a	b	c	d	e	f	g	h	
8	a8	b8	c8	d8	e8	f8	g8	h8	8
7	a7	b7	c7	d7	e7	f7	g7	h7	7
6	a6	b6	c6	d6	e6	f6	g6	h6	6
5	a5	b5	c5	d5	e5	f5	g5	h5	5
4	a4	b4	c4	d4	e4	f4	g4	h4	4
3	a3	b3	c3	d3	e3	f3	g3	h3	3
2	a2	b2	c2	d2	e2	f2	g2	h2	2
1	a1	b1	c1	d1	e1	f1	g1	h1	1

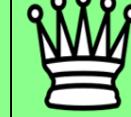
Notación algebraica corta

Esta es la forma más popular del sistema algebraico porque prescinde de la información superflua.

- No se escribe la casilla de inicio de la pieza que se mueve, solamente se escribe el nombre de la pieza y la casilla de destino.
- Las letras con que se nombran las piezas de ajedrez son siempre mayúsculas y las letras de las coordenadas del tablero son minúsculas.
- La letra 'P' de Peón nunca se usa y únicamente se anotan las coordenadas de la columna y fila final.

Los nombres de las piezas son:

Figura 3: Notación de las piezas

INICIALES PARA NOMENCLATURA EN ALGEBRAICO						
Pieza	Rey	Dama	Torre	Alfíl	Caballo	Peón
Español	R	D	T	A	C	p
Inglés	K	Q	R	B	N	p
Dibujo						

5.3. Sistemas de competencias del ajedrez

La mayoría de los torneos de ajedrez están organizados y regulados por los reglamentos de la FIDE, los formatos más comunes utilizados en la realización de torneos son los siguientes:

Por eliminación o KO, Round-Robin, sistema suizo.

5.3.1. Por eliminación o KO

Este es el sistema más sencillo y corto, consiste en que el perdedor de un encuentro es inmediatamente eliminado. Se van jugando rondas y en cada una de ellas se elimina cierto

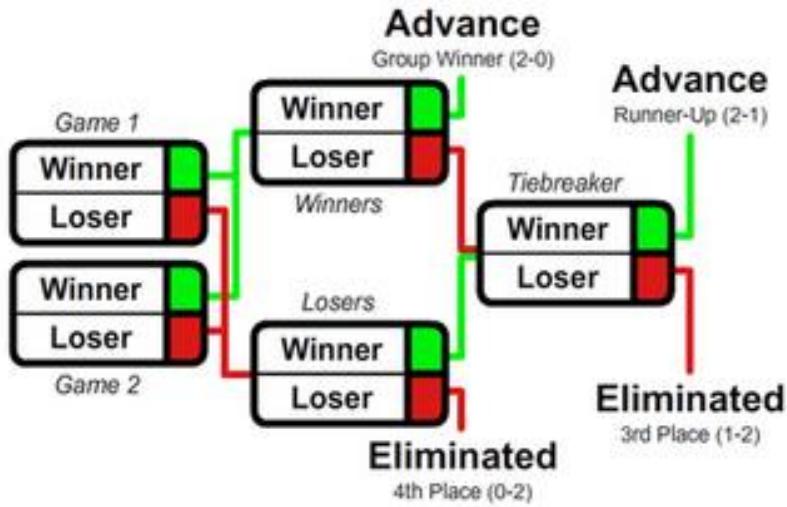
número de participantes hasta dejar un único competidor que se corona como campeón.
(Anónimo, 2018)

Figura 4: Ejemplo de eliminación directa



Otra versión de este tipo de formato es el **sistema de eliminación doble**. Mediante este sistema se busca dar mayor oportunidad a los participantes, los cuales al perder el primer partido no quedan excluidos automáticamente del certamen si no que tienen una segunda opción de ocupar los primeros puestos jugando por el lado de perdedores. Para la ubicación de los participantes en la gráfica se conservan los mismos principios de eliminación sencilla. (ecured, 2018)

Figura 5: Ejemplo de eliminación doble



5.3.2. Round-Robin

Es el típico sistema todos contra todos. El número de rondas de juego es igual al número de participantes menos uno, teniendo en cuenta que si son impares habrá que inscribir a un jugador fantasma para que sean pares (al que le toque jugar contra él libra el turno). En cada ronda se van alternando los colores blanco y negro. (123ajedrez.com, s.f.)

Para determinar el número total de partidos se utiliza la siguiente fórmula: $n= N (N-1)$ dividido entre 2, donde $n= \#$ de confrontaciones $N= \#$ de participantes $n= N-1$ (número de confrontaciones), $tp=$ total participantes. (ecured, 2018)

Otra versión del Round Robin es el Round-Robin de dos vueltas. Cada jugador jugará con las piezas blancas y negras con cada uno de los otros participantes. De esta forma nadie obtiene ninguna ventaja.

5.3.3. Sistema suizo

Es la mejor combinación de los anteriores sistemas, no hay eliminación y los jugadores con el mismo número de puntos se van enfrentando entre sí. La distribución de puntos tiene forma de campana de Gauss.

Figura 6: Ejemplo de sistema suizo

Jugadores	Mínimo de rondas
3-4	2
5-8	3
9-16	4
17-32	5
33-64	6
64-128	7
129-256	8

El número de rondas mínimo depende del número de jugadores y la principal ventaja es que cuanto más avanza el torneo más se iguala la fuerza de los ajedrecistas en los emparejamientos y los emparejamientos se deciden al final de cada ronda y para ello se suele utilizar un programa de ordenador, algunos de ellos gratuitos como el **CHESSP**, **VEGA** o **SEVILLA**, aunque hay más. (123ajedrez.com, s.f.)

5.4. Sistemas de desempate

Todo torneo de ajedrez está regido por “Reglas y Leyes del Ajedrez” impuestas por la Federación Internacional de Ajedrez. Dentro de estas reglas están determinados los sistemas de desempate, estas sirven para determinar la clasificación de los jugadores con los mismos puntos. Todos los sistemas de desempate que sean usados deben seguir un orden y ser anunciados antes del inicio del torneo, constando en el reglamento específico de la competición.

5.4.1. Partidas de desempate (Play-offs)

Es considerado el sistema de desempate más justo por definir la clasificación en una disputa de partidas entre los jugadores empatados con el mismo número de puntos en el torneo, generalmente, se realiza con un ritmo de juego de menor tiempo, como partidas en las

modalidades de ajedrez rápido o blitz. Los jugadores empatados juegan entre sí, todos contra todos o en sistema de eliminatoria (ejemplo semifinal y final). Los puestos de cada uno en el torneo serán definidos por el resultado del desempate. (Daniel Munoz, s.f.)

5.4.2. *Valoración promedio de los oponentes*

La Federación Internacional de Ajedrez tiene su sistema de puntuación internacional, llamado **rating FIDE** o popularmente conocido como “Elo” en el ambiente ajedrecístico por haber sido creado por el matemático húngaro Arpad Elo. El objetivo es medir y determinar el nivel de juego de cada jugador por medio de un cálculo estadístico, considerando los resultados de las partidas. El criterio de desempate “**Valoración promedio de los oponentes**” es la suma del rating FIDE de cada uno de los rivales de un jugador durante el torneo, dividido por la cantidad de partidas jugadas. (Daniel Munoz, s.f.)

5.4.3. *Sistema Buchholz*

El **sistema Buchholz** es un sistema de clasificación o puntuación en torneos de ajedrez desarrollado por Bruno Buchholz. Es el criterio más utilizado en los torneos suizos, consiste en sumar los puntos obtenidos en el torneo de cada uno de los oponentes de un jugador. Este sistema también puede ser aplicado de las siguientes maneras:

- **Buchholz Medio o mediano:** No considera en la suma al oponente que obtuvo la puntuación más alta ni al de más baja. Puede aplicarse también descartando a los dos oponentes con más puntos y los dos con menos.
- **Buchholz con corte:** También llamado de “Buchholz Top”, descarta de la suma al oponente de menor puntuación. También pueden desconsiderarse las dos puntuaciones más bajas (Daniel Munoz, s.f.)

5.4.4. Resultado particular o confrontación directa

Puede aplicarse en caso de que todos los jugadores empatados en puntos se hayan enfrentado, sumando los puntos de las partidas entre ellos, generando así, una clasificación entre los empatados. (Daniel Munoz, s.f.)

5.4.5. Sistema Koya

Es utilizado en los torneos de todos contra todos, también conocidos como round-Robin o schuring, se suman los puntos logrados contra todos los rivales que alcanzaron la mitad o más de los puntos totales del torneo. (Daniel Munoz, s.f.)

5.4.6. Número de partidas con las piezas negras

El nombre del criterio lo define fácilmente, también puede aplicarse considerando la cantidad de partidas ganadas con las piezas negras. Las partidas no jugadas por causa de ausencia se cuentan como si fueran con las piezas blancas. (Daniel Munoz, s.f.)

5.4.7. Sonnerborn-Berger

Es el sistema más utilizado en torneos Round-Robin, donde todos los jugadores se enfrentan entre sí. El cálculo es la suma de los puntos totales de los oponentes que ganaron y la mitad de los puntos contra quienes empataron. No se suma nada de las partidas perdidas.

5.5. ELO

El ELO es un sistema de puntuación numérico que intenta determinar la fuerza teórica de cada jugador de ajedrez en una escala ascendente. Esto quiere decir que si un jugador posee un número mayor a otro significa que teóricamente es más fuerte.

Existen dos tipos, el ELO ficticio donde básicamente es el que se encuentra en páginas webs dedicadas al ajedrez. Simplemente es un ELO online que permite realizar una clasificación de jugadores por orden de fuerza ajedrecística dentro de la página y, así, emplear un sistema de

emparejamiento que facilite al usuario encontrar jugadores cercanos a su nivel, para una experiencia más agradable y productiva. (Herrera, 2018)

El ELO real es el que rige las competiciones oficiales. Para poder acceder a la obtención de este ELO se necesita estar federados y pagar una licencia federativa (la que exija en este caso FENANIC).

5.6. Análisis de partidas

Profundizar en el análisis de partidas de ajedrez implica una revisión exhaustiva para detectar fallos, comprender movimientos esenciales y potenciar tus destrezas, siendo un paso imprescindible en la evolución ajedrecística. Para que este análisis sea realmente útil, se debe examinar las acciones de ambos contendientes, valorar las distintas etapas del juego (apertura, medio juego y final), señala las coyunturas decisivas y se recurre a un motor de análisis para contrastar tus conclusiones, aunque lo más aconsejable es que el jugador se esfuerce por descubrir las mejores opciones por el mismo antes de pedir ayuda a alguna herramienta de análisis.

5.7. Herramientas de desarrollo

5.7.1. *Visual studio code*

Es un editor de código fuente independiente que se ejecuta en Windows, macOS y Linux. La elección principal para desarrolladores web y JavaScript, con extensiones para admitir casi cualquier lenguaje de programación. (microsoft, 2022)

5.7.2. *PhpMyAdmin*

PhpMyAdmin es una aplicación web que sirve para administrar bases de datos MySQL de forma sencilla y con una interfaz amistosa. Se trata de un software muy popular basado en PHP. La ventaja de usar una aplicación web es que nos permite conectarnos con servidores

remotos, a los cuales no siempre se puede acceder usando programas de interfaz gráfica (Arsys, s.f.).

5.7.3. *HTML*

Es el componente más básico de la Web. Define el significado y la estructura del contenido web. Además de HTML, generalmente se utilizan otras tecnologías para describir la apariencia/presentación de una página web (CSS) o la funcionalidad/comportamiento (JavaScript) (Corporation's, Mozilla, s.f.).

5.7.4. *CSS*

Es un lenguaje de programación que permite crear y realizar cambios en un documento HTML de forma rápida y sencilla. En otras palabras, se trata de un lenguaje para el manejo del diseño, cambios, mejoras, presentación y personalización de páginas web (next_u, s.f.).

5.7.5. *Javascript*

JavaScript es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas. Como lenguaje de scripting del lado del servidor, se trata de una de las principales tecnologías de la World Wide Web (Amazon, s.f.).

5.7.6. *PHP*

Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML (The PHP Group, s.f.).

5.7.7. *Laravel*

Laravel es un framework de aplicaciones web con una sintaxis expresiva y elegante. Un framework web proporciona una estructura y un punto de partida para crear tu aplicación. Laravel se esfuerza por brindar una increíble experiencia de desarrollador al tiempo que proporciona características poderosas como inyección de dependencia exhaustiva, una capa de abstracción

de base de datos expresiva, colas y trabajos programados, pruebas unitarias y de integración, y más. (laravel.com, 2011)

5.8. Metodología en Cascada

Es el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior (Pressman, 1995).

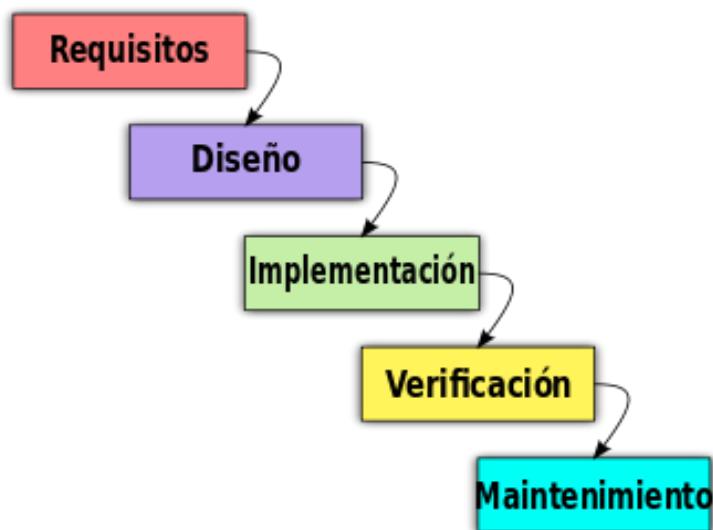
De acuerdo al investigador Winston Royce que propuso dicho modelo, los beneficios de esta metodología surgen cuando no existen fechas inmediatas de implementación por lo cual se tiene un tiempo apropiado para desarrollar cada fase. Es importante mencionar que para que este modelo tenga un menor índice de riesgo, es necesario que los requerimientos sean muy claros y se hayan establecido de manera oficial en la primera parte del proyecto.

VI. ANÁLISIS Y PRESENTACIÓN DE LOS RESULTADOS

6.1. Diseño metodológico

Para el desarrollo de este sistema, se utilizará la metodología de desarrollo de software en cascada por su simplicidad, la cual sugiere un enfoque sistemático en donde cada fase se inicia apenas termine su fase anterior constando de 5 fases que se muestran en el siguiente gráfico.

Figura 7: Gráfico de las fases de la Metodología de desarrollo Cascada



6.1.1. Requisitos

En esta fase se levantarán los requisitos del sistema, es decir, se definirán las características y funcionalidades que debe tener el software. Esto permitirá tener una visión clara de las expectativas y necesidades para la realización del proyecto. Con el cumplimiento y finalización de esta etapa, se pretende satisfacer el objetivo específico 1: Definir las necesidades y funcionalidades esenciales del sistema web de gestión de torneos de ajedrez, en colaboración con la Academia “Estrellas del Ajedrez”.

6.1.1.1. Especificación de requisitos.

Para poder llevar a cabo esta fase se pretende realizar entrevistas con el Experto Nacional y candidato a Maestro FIDE Steven Sánchez, esto con el fin de recopilar una serie de requerimientos del proyecto, de esta manera también se lograr entender cuáles son los procesos que debe cumplir el sistema.

6.1.2. Diseño

El modelo de diseño seleccionado es el Patrón por Capas. Este patrón organiza los roles y procesos del sistema en distintas capas, permitiendo una separación clara de responsabilidades. Cada capa se encarga de un aspecto específico del sistema, lo que facilita la coordinación y el encapsulamiento durante el desarrollo del proyecto. Esta estructura proporciona una abstracción que simplifica la gestión y satisfacción de las solicitudes del sistema.

Figura 8: Arquitectura patrón en capas



Con la finalización de esta etapa, se cumple el objetivo específico 2: Diseñar la arquitectura del sistema considerando los módulos de gestión, torneos, tablas y resultados para el funcionamiento eficiente del mismo.

6.1.3. Implementación

En esta fase, se iniciará el desarrollo del código fuente del sistema. Se implementarán prototipos y pruebas para identificar y corregir errores a medida que surjan.

Se utilizarán las siguientes herramientas:

6.1.3.1. Base de datos

MySQL se empleará como motor de base de datos relacional debido a su rendimiento, seguridad y amplia adopción. Para la gestión y administración, se usará PhpMyAdmin, que facilita la creación de tablas, inserción de datos, ejecución de consultas y administración de permisos a través de una interfaz gráfica intuitiva.

6.1.3.2. Desarrollo del código

El Backend se desarrollará utilizando un stack tecnológico moderno y robusto siendo estas Laravel 12, PHP 8.2+, javascript. En el Front-End, se utilizarán Blade, Bootstrap y Tailwind CSS para crear interfaces de usuario interactivas, editadas en Visual Studio Code.

6.1.3.3. Personalización del modelo

Se adaptará el sistema a los requisitos específicos de la Academia “Estrellas del Ajedrez”, ajustando los parámetros del software a sus necesidades particulares.

6.1.3.4. Entregables

Se proporcionará una base de datos para almacenar información sobre jugadores, partidas y torneos, así como una aplicación web que incluirá varias pantallas funcionales:

- Pantalla de inicio

- Pantalla de recuperación de contraseña
- Pantalla principal
- Pantalla de ingreso de partidas
- Pantalla de lectura de partidas
- Pantalla de administración de torneos
- Pantalla de ingreso de jugadores
- Pantalla de búsqueda de partidas
- Pantalla de resultados generales
- Pantalla de ingresos de resultados
- Pantalla de ingreso de usuarios
- Pantalla de ingreso de academias
- Pantalla de análisis de partidas
- Pantalla de administración de usuarios

Con la finalización de esta etapa, se cumple el objetivo específico 3: Implementar el sistema web utilizando tecnologías HTML, CSS, JS, MySQL, PHP, entre otros, para su desarrollo según los requerimientos establecidos.

6.1.4. Verificación

Se llevará a cabo una fase de pruebas exhaustivas para asegurar que el sistema funcione de acuerdo con los requisitos establecidos. Los usuarios finales participarán en estas pruebas para evaluar el desempeño del sistema en condiciones reales. Cualquier error identificado durante esta fase será corregido para garantizar que el sistema funcione de manera óptima.

Con la finalización de esta etapa, se cumple el objetivo específico 4: Realizar pruebas exhaustivas del sistema web para garantizar la funcionalidad, usabilidad y seguridad de la plataforma.

6.1.5. *Mantenimiento*

En esta etapa, se llevará a cabo el mantenimiento continuo del sistema para abordar cualquier problema o fallo identificado por el cliente. Se documentarán los problemas encontrados para su posterior corrección y actualización del sistema, garantizando que se mantenga alineado con las expectativas y necesidades del usuario.

Con el avance en esta fase, se asegura el cumplimiento del objetivo específico 5: Realizar un seguimiento continuo con el objetivo de identificar las áreas de mejora del sistema.

6.2. Fase de requisitos

La Fase de requisitos establece las bases para el diseño y desarrollo del Sistema Web de gestión de torneos de ajedrez para la Academia Estrellas del Ajedrez, resaltando las necesidades de los usuarios y partes interesadas.

6.2.1. Requerimientos funcionales

Tabla 1

Requerimiento funcional No. 1

Identificador: RF01	Nombre: Gestión de usuarios	
Tipo: Necesario	Crítico: Si	Prioridad: Alta
Entrada: <ul style="list-style-type: none">- Correo electrónico del usuario- Contraseña- Rol asignado- Estado de actividad del usuario		Salida: <ul style="list-style-type: none">- Ventana pop-up con mensaje de estado: completado/error- Actualización de la tabla de usuarios en caso de éxito
Descripción <ul style="list-style-type: none">- El sistema debe permitir crear, modificar y eliminar cuentas de usuario.		
Postcondición <ul style="list-style-type: none">- Mostrar una ventana pop-up que notifique el estado de la acción realizada (completado/no completado), así como una breve descripción del mismo.- Automáticamente, se actualizará la tabla correspondiente a la lista de usuarios.		
Gestión de errores y excepciones <ul style="list-style-type: none">- Si el formulario es enviado con un campo incompleto, el sistema mostrará un mensaje de error debajo del campo correspondiente.- Si trata de ingresar un registro que ya existe, el sistema mostrará un mensaje de error en una ventana pop-up.- Si desea actualizar la contraseña de un usuario registrado, debe ingresar la nueva contraseña y confirmar en un segundo campo de texto.		
Criterios de aceptación <ul style="list-style-type: none">- El correo debe cumplir con un formato válido (ejemplo: <u>correo@dominio.com</u>).- El correo a registrar/actualizar no debe coincidir con otro ya existente en el sistema.		

Nota: La tabla describe el proceso de registro, actualización y eliminación de usuarios, así como el manejo de errores correspondiente.

Tabla 2*Requerimiento funcional No. 2*

Identificador: RF02	Nombre: Inicio de sesión	
Tipo: Necesario	Crítico: Si	Prioridad: Alta
Entrada: <ul style="list-style-type: none"> - Correo electrónico del usuario - Contraseña 		Salida: <ul style="list-style-type: none"> - Redirección al home del sistema en caso de éxito - Mensaje de error en caso de contraseña o correo incorrectos
Descripción <ul style="list-style-type: none"> - El sistema debe permitir el inicio de sesión por medio de los usuarios creados. Asimismo, debe facilitar un apartado de recuperación de contraseña. 		
Postcondición <ul style="list-style-type: none"> - El usuario es redirigido al home del sistema en donde el contenido dependerá del rol que tenga asignado. - En el caso de la recuperación de contraseña, el usuario recibirá un token al correo electrónico asignado a su cuenta, que podrá utilizar para acceder al sistema. 		
Gestión de errores y excepciones <ul style="list-style-type: none"> - Si el formulario es enviado con un campo incompleto, el sistema mostrará un mensaje de error debajo del campo correspondiente. - Si trata la contraseña o el correo electrónico es incorrecto, el sistema mostrará un mensaje de error en una ventana pop-up. - Si desea recuperar la contraseña, el usuario recibirá un token a su correo electrónico; sin embargo, si hace eso durante 5 veces seguidas en un lapso de tiempo menor a 5 minutos, el sistema no le permitirá solicitar más tokens hasta luego de 15 minutos. 		
Criterios de aceptación <ul style="list-style-type: none"> - El usuario debe ingresar un correo y contraseña que hayan sido registrados previamente en el sistema. 		

Nota: La tabla describe el proceso de inicio de sesión y recuperación de contraseña, así como el manejo de errores correspondiente.

Tabla 3*Requerimiento funcional No. 3*

Identificador: RF03	Nombre: Gestión de roles y permisos	
Tipo: Necesario	Crítico: Si	Prioridad: Alta
Entrada: <ul style="list-style-type: none"> - Rol a realizar asignación/ remoción - Permiso a ser asignado/retirado 		Salida: <ul style="list-style-type: none"> - Ventana pop-up con mensaje de estado: completado/error - Actualización de la tabla de usuarios en caso de éxito
Descripción <ul style="list-style-type: none"> - El sistema debe permitir asignar y eliminar permisos a los roles registrados. 		
Postcondición <ul style="list-style-type: none"> - Mostrar una ventana pop-up que notifique el estado de la acción realizada (completado/no completado), así como una breve descripción del mismo. 		
Gestión de errores y excepciones <ul style="list-style-type: none"> - Si existe un inconveniente al momento de asignar un permiso o hay un problema al momento de realizar la acción, el sistema mostrará una ventana pop-up con un mensaje de error y una breve descripción del mismo. 		
Criterios de aceptación <ul style="list-style-type: none"> - Seleccionar un permiso que no haya sido asignado aún al rol con el que se está interactuando. 		

Nota: La tabla describe el proceso de asignación y remoción de permisos a los roles registrados; asimismo, describe el manejo de errores correspondiente.

Tabla 4*Requerimiento funcional No. 4*

Identificador: RF04	Nombre: Gestión de colaboradores de las academias	
Tipo: Necesario	Crítico: Si	Prioridad: Alta
Entrada: <ul style="list-style-type: none"> - Datos personales del colaborador - Número de teléfono - Correo (para el inicio de sesión en el sistema) - Academia de procedencia - Estado 		Salida: <ul style="list-style-type: none"> - Ventana pop-up con mensaje de estado: completado/error - Actualización de la tabla de colaboradores en caso de éxito
Descripción <ul style="list-style-type: none"> - El sistema debe permitir crear, modificar y eliminar los registros de los colaboradores de las academias que pueden acceder y gestionar información relacionada con sus estudiantes al sistema. 		
Postcondición <ul style="list-style-type: none"> - Mostrar una ventana pop-up que notifique el estado de la acción realizada (completado/no completado), así como una breve descripción del mismo. - Automáticamente, se actualizará la tabla correspondiente a la lista de colaboradores. 		
Gestión de errores y excepciones <ul style="list-style-type: none"> - Si el formulario es enviado con un campo incompleto, el sistema mostrará un mensaje de error debajo del campo correspondiente. - Si se trata de ingresar/actualizar un registro que ya existe o hay un problema al momento de realizar la acción, el sistema mostrará una ventana pop-up con un mensaje de error y una breve descripción del mismo. 		
Criterios de aceptación <ul style="list-style-type: none"> - El No. de identificación del colaborador a ingresar/modificar debe ser diferente a la de los colaboradores registrados. 		

Nota: La tabla describe el proceso de registro, actualización y eliminación de los colaboradores, así como el manejo de errores correspondiente.

Tabla 5*Requerimiento funcional No. 5*

Identificador: RF05	Nombre: Gestión de estudiantes de las academias	
Tipo: Necesario	Crítico: Si	Prioridad: Alta
Entrada: <ul style="list-style-type: none"> - Datos personales del colaborador - Número de teléfono - Correo (para el inicio de sesión en el sistema) - Academia de procedencia - Club - Fecha de inscripción - Estado 		Salida: <ul style="list-style-type: none"> - Ventana pop-up con mensaje de estado: completado/error - Actualización de la tabla de estudiantes en caso de éxito
Descripción <ul style="list-style-type: none"> - El sistema debe permitir crear, modificar y eliminar a los estudiantes de las academias para que puedan acceder a la información relacionada con los torneos en los que participan o han participado. 		
Postcondición <ul style="list-style-type: none"> - Mostrar una ventana pop-up que notifique el estado de la acción realizada (completado/no completado), así como una breve descripción del mismo. - Automáticamente, se actualizará la tabla correspondiente a la lista de estudiantes. 		
Gestión de errores y excepciones <ul style="list-style-type: none"> - Si el formulario es enviado con un campo incompleto, el sistema mostrará un mensaje de error debajo del campo correspondiente. - Si se trata de ingresar/modificar un registro que ya existe o hay un problema al momento de realizar la acción, el sistema mostrará una ventana pop-up con un mensaje de error y una breve descripción del mismo. 		
Criterios de aceptación <ul style="list-style-type: none"> - El no. de identificación del estudiante a ingresar/modificar, debe ser diferente al de los estudiantes registrados. 		

Nota: La tabla describe el proceso de registro, actualización y eliminación de los estudiantes, así como el manejo de errores correspondiente.

Tabla 6*Requerimiento funcional No. 6*

Identificador: RF06	Nombre: Gestión de academias	
Tipo: Necesario	Crítico: Si	Prioridad: Alta
Entrada: <ul style="list-style-type: none"> - Nombre de la academia - Correo de la academia - No. de identificación de su representante - Dirección - Ciudad de procedencia - Estado 		Salida: <ul style="list-style-type: none"> - Ventana pop-up con mensaje de estado: completado/error - Actualización de la tabla de academias en caso de éxito - Actualización de los menús de opciones para la selección de academias en los formularios que lo requieren
Descripción <ul style="list-style-type: none"> - El sistema debe permitir crear, modificar y eliminar a las academias para que puedan ser utilizadas al momento de inscribir a un estudiante o colaborador. 		
Postcondición <ul style="list-style-type: none"> - Mostrar una ventana pop-up que notifique el estado de la acción realizada (completado/no completado), así como una breve descripción del mismo. - Automáticamente, se actualizará la tabla correspondiente a la lista de academias. 		
Gestión de errores y excepciones <ul style="list-style-type: none"> - Si el formulario es enviado con un campo incompleto, el sistema mostrará un mensaje de error debajo del campo correspondiente. - Si se trata de ingresar/modificar un registro que ya existe o hay un problema al momento de realizar la acción, el sistema mostrará una ventana pop-up con un mensaje de error y una breve descripción del mismo. 		
Criterios de aceptación <ul style="list-style-type: none"> - El nombre de la academia a ingresar/modificar, debe ser diferente a la de las academias registradas. 		

Nota: La tabla describe el proceso de registro, actualización y eliminación de las academias, así como el manejo de errores correspondiente.

Tabla 7*Requerimiento funcional No. 7*

Identificador: RF07	Nombre: Gestión de torneos	
Tipo: Necesario	Crítico: Si	Prioridad: Alta
Entrada: <ul style="list-style-type: none"> - Nombre del torneo - Fecha, hora y lugar en donde se llevará a cabo el torneo - No. de rondas que tendrá - La federación (si lo tuviera) que lo organiza - El no. de identificación de los árbitros que tendrá - El no. de identificación del director - El no. de identificación del organizador - Estado 		Salida: <ul style="list-style-type: none"> - Ventana pop-up con mensaje de estado: completado/error - Actualización de la tabla de torneos en caso de éxito - Actualización de los menús de opciones para la selección de federaciones en los formularios que lo requieren
Descripción <ul style="list-style-type: none"> - El sistema debe permitir crear, modificar y eliminar torneos, considerando detalles como los controles de tiempo y las categorías del torneo. 		
Postcondición <ul style="list-style-type: none"> - Mostrar una ventana pop-up que notifique el estado de la acción realizada (completado/no completado), así como una breve descripción del mismo. - Automáticamente, se actualizará la tabla correspondiente a la lista de torneos. 		
Gestión de errores y excepciones <ul style="list-style-type: none"> - Si el formulario es enviado con un campo incompleto, el sistema mostrará un mensaje de error debajo del campo correspondiente. - Si se trata de ingresar/modificar un registro que ya existe o hay un problema al momento de realizar la acción, el sistema mostrará una ventana pop-up con un mensaje de error y una breve descripción del mismo. 		

Criterios de aceptación

- El nombre y la fecha del torneo a ingresar/modificar, debe ser diferente a la de los torneos ya registrados.

Nota: La tabla describe el proceso de registro, actualización y eliminación de los torneos, así como el manejo de errores correspondiente.

Tabla 8*Requerimiento funcional No. 8*

Identificador: RF8	Nombre: Gestión de inscripciones en los torneos	
Tipo: Necesario	Crítico: Si	Prioridad: Alta
Entrada: <ul style="list-style-type: none"> - Nombre del primer jugador - Nombre del segundo jugador (jugador rival) 		Salida: <ul style="list-style-type: none"> - Ventana pop-up con mensaje de estado: completado/error - Actualización de la tabla de inscripciones en caso de éxito - Actualización de los menús de opciones para la selección de participantes inscritos en un torneo específico, en los formularios que lo requieren
Descripción <ul style="list-style-type: none"> - El sistema debe permitir la inscripción y eliminación de participantes en los torneos. 		
Postcondición <ul style="list-style-type: none"> - Mostrar una ventana pop-up que notifique el estado de la acción realizada (completado/no completado), así como una breve descripción del mismo. - Automáticamente, se actualizará la tabla correspondiente a la lista de inscripciones. 		
Gestión de errores y excepciones <ul style="list-style-type: none"> - Si el formulario es enviado con un campo incompleto, el sistema mostrará un mensaje de error debajo del campo correspondiente. - Si se trata de ingresar un registro que ya existe o hay un error al momento de realizar la acción, el sistema mostrará una ventana pop-up con un mensaje de error y una breve descripción del mismo. 		
Criterios de aceptación <ul style="list-style-type: none"> - Seleccionar los participantes a enfrentarse en el formulario correspondiente. 		

Nota: La tabla describe el proceso de registro y eliminación de las inscripciones, así como el manejo de errores correspondiente.

Tabla 9*Requerimiento funcional No. 9*

Identificador: RF9	Nombre: Gestión de partidas en los torneos	
Tipo: Necesario	Crítico: Si	Prioridad: Alta
Entrada: <i>Las partidas son generadas directamente por el sistema con los emparejamientos, por lo que no se requiere de una entrada de datos manual</i>		Salida: <ul style="list-style-type: none"> - Ventana pop-up con mensaje de estado: completado/error - Actualización de la tabla de partidas en caso de éxito
Descripción <ul style="list-style-type: none"> - El sistema debe permitir el registro, actualización y eliminación de partidas en los torneos. Postcondición <ul style="list-style-type: none"> - Mostrar una ventana pop-up que notifique el estado de la acción realizada (completado/no completado), así como una breve descripción del mismo. - Automáticamente, se actualizará la tabla correspondiente a la lista de partidas. 		
Gestión de errores y excepciones <ul style="list-style-type: none"> - Si el formulario es enviado con un campo incompleto, el sistema mostrará un mensaje de error debajo del campo correspondiente. - Si se trata de ingresar/actualizar un registro que ya existe o hay un error al momento de realizar la acción, el sistema mostrará una ventana pop-up con un mensaje de error y una breve descripción del mismo. 		
Criterios de aceptación <ul style="list-style-type: none"> - La partida debe estar formada por un estudiante y torneo activos, y un estudiante solo puede estar inscrito en una partida por ronda. 		

Nota: La tabla describe el proceso de registro, actualización y eliminación de las partidas, así como el manejo de errores correspondiente.

Tabla 10*Requerimiento funcional No. 10*

Identificador: RF10	Nombre: Sistema de emparejamiento automático	
Tipo: Necesario	Crítico: Si	Prioridad: Alta
Entrada: <ul style="list-style-type: none"> - Torneo a emparejar 		Salida: <ul style="list-style-type: none"> - Detalles de la partida a disputarse
Descripción <ul style="list-style-type: none"> - El sistema debe permitir el emparejamiento automático de sus partidas, basados en los puntajes ELO, historial y experiencia de los jugadores registrados en el sistema. 		
Postcondición <ul style="list-style-type: none"> - Mostrar una ventana pop-up que notifique el estado de la acción realizada (completado/no completado), así como una breve descripción del mismo. - Automáticamente, se actualizará la tabla correspondiente a la lista de partidas. 		
Gestión de errores y excepciones <ul style="list-style-type: none"> - Si el torneo seleccionado no tiene participantes inscritos, el sistema mostrará una ventana pop-up con un mensaje pop-up y una breve descripción del mismo. 		
Criterios de aceptación <ul style="list-style-type: none"> - El torneo debe contar con participantes registrados para emparejar. 		

Nota: La tabla describe cómo el sistema permite realizar el emparejamiento automático de las partidas.

Tabla 11*Requerimiento funcional No. 11*

Identificador: RF11	Nombre: Gráfica de torneos y resultados	
Tipo: Necesario	Crítico: Si	Prioridad: Alta
Entrada:		Salida: <ul style="list-style-type: none">▪ Gráfica con las estadísticas personales o generales según sea el rol.
Descripción <ul style="list-style-type: none">▪ El sistema debe mostrar en el home, un gráfico con el estadístico personal de los resultados en las partidas y/o de los torneos en general, según sea el rol.		
Postcondición		
Gestión de errores y excepciones		
Criterios de aceptación		

Nota: La tabla describe cómo el sistema mostrará un gráfico con la información de las estadísticas personales y/o generales en el desempeño y desarrollo de los torneos.

Tabla 12*Requerimiento funcional No. 12*

Identificador: RF12	Nombre: Auditoría/Control de cambios	
Tipo: Deseable	Crítico: No	Prioridad: Media
Entrada:		Salida: <ul style="list-style-type: none">- Actualización periódica de la tabla correspondiente a la auditoría/control de cambios
Descripción <ul style="list-style-type: none">- El sistema debe tener un registro en donde se reflejen las interacciones CRUD entre el usuario y el sistema.		
Postcondición <ul style="list-style-type: none">- Actualización periódica de la tabla correspondiente a la auditoría/control de cambios, cada que se complete con éxito una función CRUD.		
Gestión de errores y excepciones		
Criterios de aceptación		

Nota: Esta tabla explica el proceso del historial de control de cambios en los registros del sistema web.

6.2.2. Requerimientos no funcionales

- **Usabilidad:** el sistema debe contar con una interfaz intuitiva y fácil de usar que permita a los usuarios gestionar torneos y participantes sin dificultad.
- **Escalabilidad:** el sistema debe ser escalable, por lo que debe permitir agregar más academias, federaciones, torneos y usuarios sin que su rendimiento disminuya.
- **Accesibilidad:** el sistema debe ser accesible desde diferentes dispositivos sin importar la resolución de su pantalla; asimismo, debe ser compatible con los navegadores más comunes.

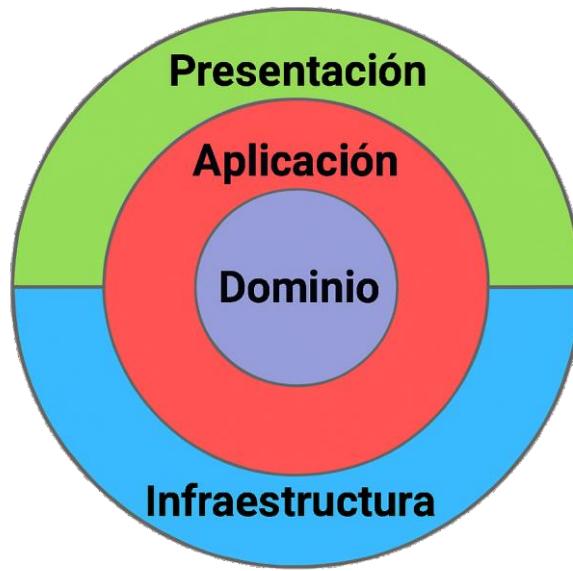
6.3. Fase de diseño

En esta fase, los requisitos se convierten en un plan técnico detallado que permitirá guiar la construcción del Sistema Web de gestión de torneos de ajedrez de la Academia Estrellas del Ajedrez. En esta etapa, se define la arquitectura del software, el diseño de la base de datos, las interfaces de usuario asegurando que el sistema cumpla con los objetivos funcionales y no funcionales.

6.3.1. Arquitectura del sistema – *Clean arquitecture*

Clean Architecture se fundamenta en un enfoque en capas que promueve una separación clara de responsabilidades. Este enfoque, permite la escalabilidad, el mantenimiento y la evolución del sistema, permitiendo que cada componente sea independiente y fácil de modificar sin afectar a otras partes del sistema. Esta arquitectura incluye las siguientes capas:

Figura 9: Arquitectura general del sistema



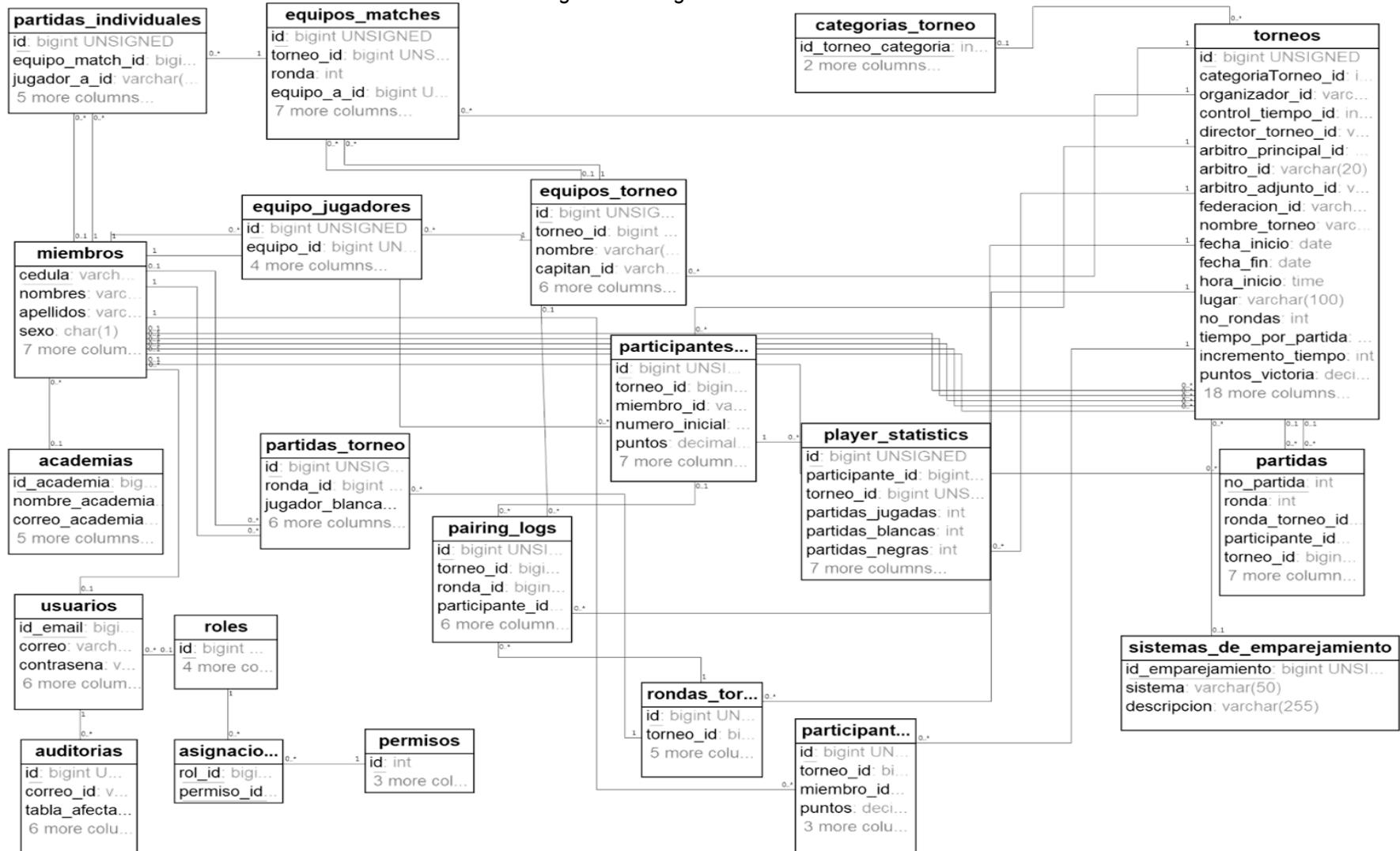
Nota: La figura muestra la arquitectura limpia del software, organizada en capas concéntricas que separan la lógica de negocio, los casos de uso, la interfaz de usuario y los servicios externos.

6.3.1.1. Capa de dominio (Domain)

La capa de dominio representa el núcleo del sistema; aquí, se definen las entidades y las reglas de negocio que determinan su comportamiento. En esta capa se encuentran las clases fundamentales que representan los conceptos clave del sistema, como los torneos, jugadores, partidas, entre otros, que comprenden el contexto del sistema de gestión de torneos de ajedrez.

- **Entidades:** Se incluyen clases como Torneos, Miembros, Partidas, que agrupan tanto los atributos (por ejemplo, Identificación, Nombre, Fecha, Resultados) como los métodos que implementan la lógica de negocio asociada.
- **Servicios de dominio:** Esta capa también puede contener servicios que gestionan procesos complejos que involucran varias entidades, como la programación de partidas o la validación de resultados.

Figura 10: Diagrama de clases del sistema



Nota: Esta ilustración muestra el diagrama de clases UML del Sistema Web de gestión de torneos de ajedrez para la academia

Estrellas del Ajedrez, detallando sus entidades

6.3.1.1.1. Reglas del negocio del Sistema de gestión de torneos para la Academia Estrellas del ajedrez

1. Gestión de Usuarios y Roles:
 - a. **Registro de Usuarios:** Los usuarios se registran en el sistema asignándoles un rol, el cual determina los permisos y las funciones que pueden acceder dentro de la plataforma.
 - b. **Asignación de Roles:** Cada usuario puede estar asociado a un único rol, el cual se define a través de la entidad *roles* y establece sus capacidades y el alcance de sus acciones dentro del sistema.
 - c. **Gestión de permisos:** Los roles pueden estar vinculados a uno o más permisos, los cuales se gestionan mediante la entidad *asignaciones_permisos*, y definen las acciones que un usuario está autorizado a realizar en el sistema.
 - d. **Recuperación de contraseña y gestión de contraseñas:** En caso de olvidar su contraseña, el usuario podrá restablecerla desde la página de inicio de sesión mediante un formulario que solicitará su correo electrónico asociado. El sistema enviará un token al correo, el cual podrá usarse para acceder y restablecer la contraseña. El sistema no devolverá la contraseña original y se asegurará de encriptarla en la base de datos.
2. Gestión de permisos:
 - a. **Gestión de permisos asignados:** Los permisos pueden ser asignados o eliminados de los roles correspondientes según sea necesario; todo esto será gestionado a través de la entidad *asignaciones_permisos*.
3. Gestión de estudiantes y colaboradores:
 - a. **Registro de estudiantes y colaboradores:** Para registrar tanto a estudiantes como a colaboradores de la academia o academias externas con las que se realizan competencias, se les pedirá su número de identificación y datos personales, los cuales serán almacenados en la entidad *Miembros*.

b. **Asignación de usuarios:** A los colaboradores y estudiantes se les asignará un usuario para que puedan gestionar a los estudiantes de sus academias, su desempeño, puntajes ELO y torneos (en el caso de los colaboradores), o para hacer un seguimiento de su rendimiento personal (en el caso de los estudiantes).

4. Gestión de academias:

a. **Registro de academias:** Las academias que participan habitualmente en los torneos podrán ser registradas proporcionando su nombre, la identificación del representante de la academia (quien debe estar previamente registrado como colaborador), la ciudad de origen, entre otros detalles. Este proceso será gestionado a través de la entidad *Academias*.

5. Gestión de federaciones:

a. **Registro de federaciones:** Las federaciones se registran junto con el país al que están asociadas y se utilizan tanto en los torneos como parte de la información de los colaboradores o estudiantes registrados. Las federaciones se almacenan en la entidad *Federaciones*.

6. Gestión de inscripciones:

a. **Inscripciones a torneos:** Las inscripciones de los estudiantes en los torneos se procesarán solicitando el número de identificación del estudiante y el torneo en el que participará (ambos previamente registrados). Tanto el estudiante como el torneo deben estar activos, y las inscripciones se registrarán en la entidad *Participantes*.

7. Gestión de torneos:

a. **Creación de torneos:** Para la creación de un torneo, se requieren datos generales como el nombre, hora, fecha y lugar en el que se llevará a cabo. A continuación, se solicitan detalles adicionales como la categoría del torneo, formato, sistema de emparejamiento, entre otros. Finalmente, se identifican los actores involucrados en la organización del

torneo (director, organizador y árbitros) y se establece un estado para determinar si el torneo ha finalizado. Toda esta información se reflejará en la entidad *Torneos*.

Es importante señalar que este proceso se refiere exclusivamente a los torneos realizados en modalidad presencial.

8. Gestión de partidas y emparejamiento automático

- a. **Registro de partidas:** Las partidas de ajedrez en un torneo involucran a dos jugadores compitiendo entre sí, pero la recopilación de datos abarca varios factores clave. Además de los resultados de la partida, se deben registrar detalles como el color asignado a cada jugador, el número de mesa donde se lleva a cabo la partida, el tiempo empleado durante el juego y el sistema de desempate utilizado en caso de empate o abandono por parte de alguno de los jugadores, entre otros aspectos relevantes.
- b. **Sistema de emparejamiento:** El sistema de emparejamiento varía según el tipo de formato de torneo utilizado:
 - **Por eliminación:** Los jugadores se emparejan directamente en rondas eliminatorias. En la primera ronda, se crean los emparejamientos de forma aleatoria o según el ranking previo de los jugadores. Los ganadores de cada partida avanzan a la siguiente ronda, mientras que los perdedores quedan eliminados. En la siguiente ronda, los jugadores ganadores se emparejan entre sí, y así sucesivamente hasta que se determina un solo campeón.
 - **Round Robin:** Los emparejamientos se realizan de forma predeterminada, con cada jugador enfrentándose una vez a cada otro jugador. A menudo, el emparejamiento se organiza en un calendario preestablecido que asegura que no haya repeticiones de partidas.
 - **Sistema suizo:** Los jugadores se emparejan en cada ronda según su puntaje acumulado. En la primera ronda, los jugadores se emparejan de forma aleatoria o con

base en su ranking inicial. A partir de la segunda ronda, los emparejamientos se hacen entre jugadores con puntajes similares, buscando que aquellos con más victorias se enfrenten entre sí, mientras que los jugadores con menos victorias se emparejan entre ellos. Si hay un número impar de jugadores, puede haber un "empate técnico", en el que un jugador se enfrenta a un "jugador fantasma" y obtiene una victoria automática.

Cabe destacar que, en la academia, la mayoría de los torneos se realizan con el sistema suizo, debido a la naturaleza equilibrada de los emparejamientos que este sistema ofrece, permitiendo que los jugadores compitan de manera justa durante todo el evento.

9. Validaciones de datos:

- a. **Validaciones de entrada:** Se implementan validaciones para garantizar que todos los campos requeridos sean completados de manera correcta al registrar o actualizar la información de los torneos, participantes o la referente a los mismos.
- b. **Control de estado activo:** Se gestionan adecuadamente los estados *Activo* de los torneos, participantes y otras entidades, asegurando que solo se procesen los elementos relevantes dentro del sistema de gestión de torneos.
- c. **Importación de datos:** Se importan los datos mediante un archivo .csv que debe seguir un formato predefinido. Una vez completada la importación, el sistema generará un mensaje que muestra un resumen de los registros procesados, indicando el estado de la importación, como la cantidad de registros importados, no importados, incompletos, entre otros, para proporcionar una visión clara del proceso.
- d. **Exportación de datos:** Se exportan los datos en formato .csv, permitiendo que los usuarios obtengan la información relevante de torneos, participantes, puntajes ELO, y otras entidades en un formato estructurado y fácilmente accesible para su análisis o almacenamiento externo.

6.3.1.2. Capa de aplicación (Application)

La Capa de Aplicación actúa como un intermediario entre la Capa de Dominio y las demás capas del sistema, siendo la encargada de coordinar las acciones que los usuarios pueden realizar y gestiona la lógica necesaria para cumplir con estos requerimientos dentro del contexto del sistema de gestión de torneos de ajedrez.

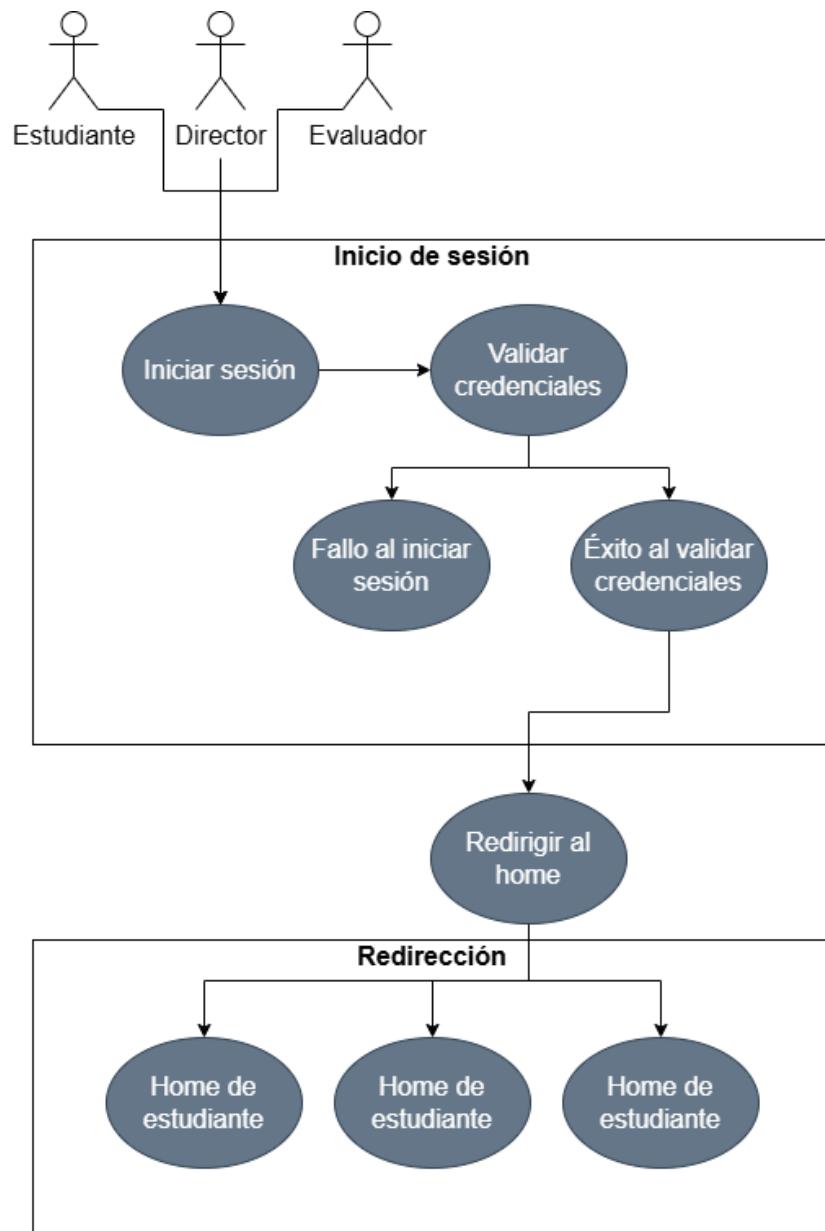
Responsabilidades de la Capa de Aplicación

- **Casos de Uso:** Esta capa define los principales casos de uso del sistema, tales como Registrar Torneo, Inscribir Participante, Asignar Puntaje ELO, entre otros. Estos casos de uso describen cómo los usuarios interactúan con el sistema, detallando los actores involucrados, las acciones realizadas y los resultados esperados en cada proceso.
- **Servicios de Aplicación:** La Capa de Aplicación implementa la lógica que coordina la ejecución de los casos de uso, invocando las entidades del dominio y los repositorios necesarios para completar las tareas solicitadas. Los servicios de aplicación son responsables de coordinar las distintas operaciones, asegurando que se apliquen correctamente las reglas de negocio establecidas, como la validación de la inscripción en torneos o la actualización de puntajes.
- **Interacción con Repositorios:** La Capa de Aplicación interactúa con las interfaces de repositorios para acceder a los datos requeridos sin conocer su implementación interna. Esto separa la lógica de negocio del acceso a los datos, facilitando el mantenimiento del sistema y las pruebas, y permitiendo que las operaciones se lleven a cabo de manera eficiente y modular.

Diagramas de Casos de Uso

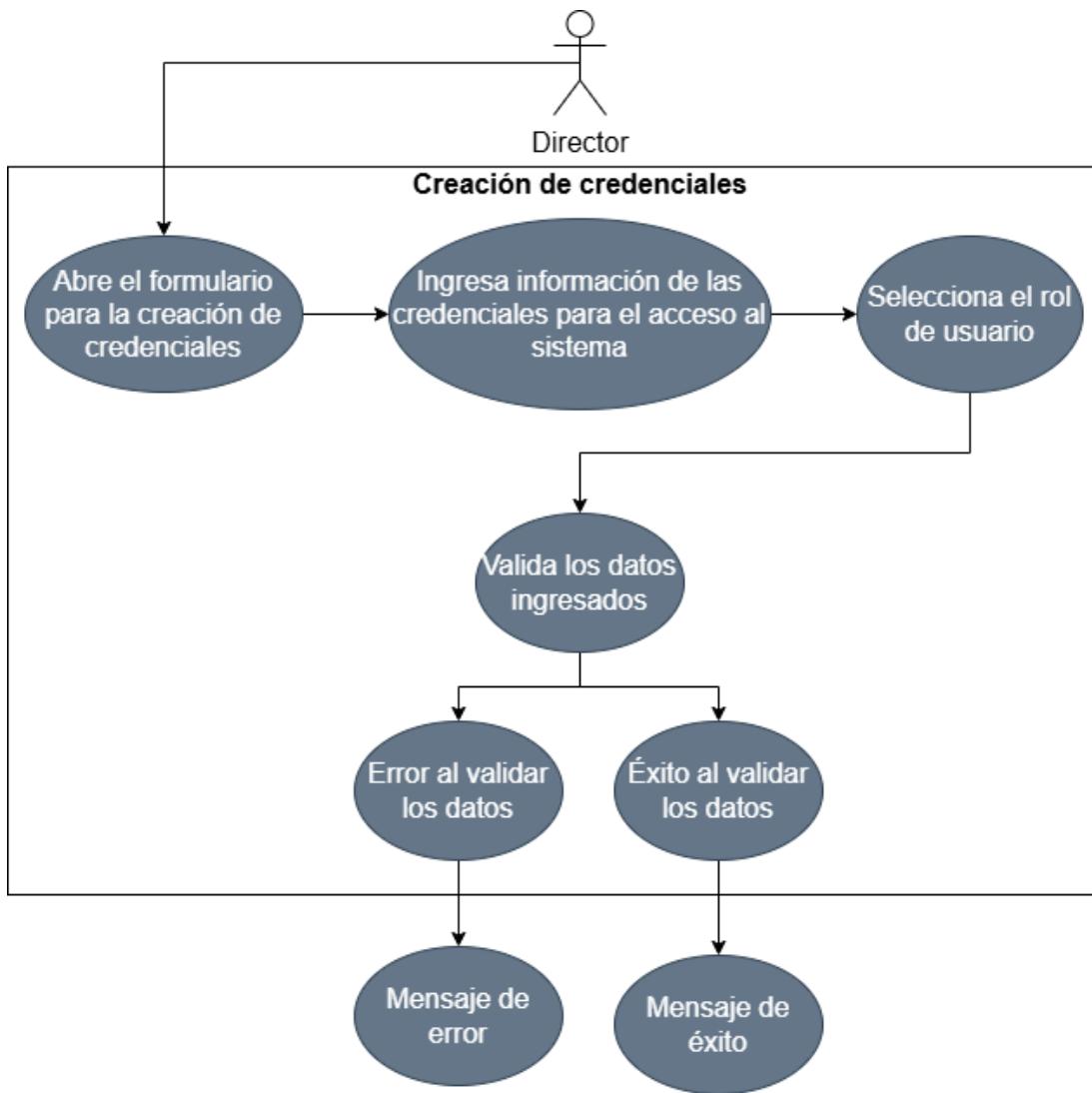
Para ilustrar cómo los usuarios interactúan con el sistema, se incluyen los siguientes diagramas de casos de uso. Estos diagramas muestran las interacciones entre los actores y el sistema, proporcionando una representación visual clara de las funcionalidades disponibles, los flujos de trabajo y los pasos que se deben seguir para completar cada acción dentro del sistema de gestión de torneos.

Figura 11: Diagrama de caso de uso No. 1



Nota: La ilustración muestra el Caso de Uso correspondiente a la autenticación de usuario.

Figura 12: Diagrama de caso de uso No. 2



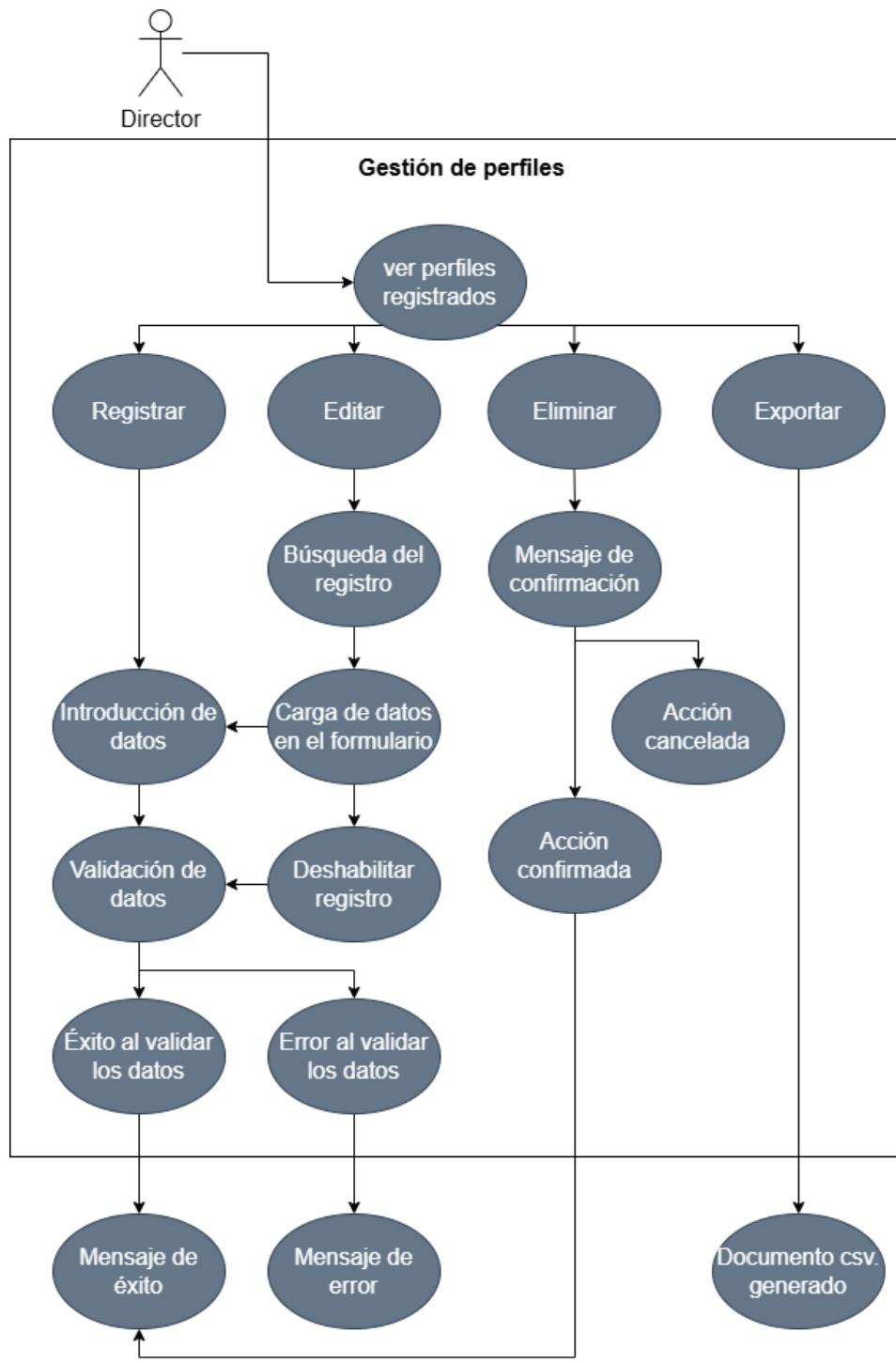
Nota: La ilustración muestra el Caso de Uso correspondiente a la creación de las credenciales de los usuarios.

Es importante destacar que este proceso está restringido exclusivamente a los administradores del sistema, quienes tienen la autorización para generar y gestionar las credenciales de acceso.

Adicionalmente, el rol asignado a cada usuario durante este proceso determina de manera fundamental las funciones y permisos dentro de la plataforma. La academia define tres tipos de roles principales:

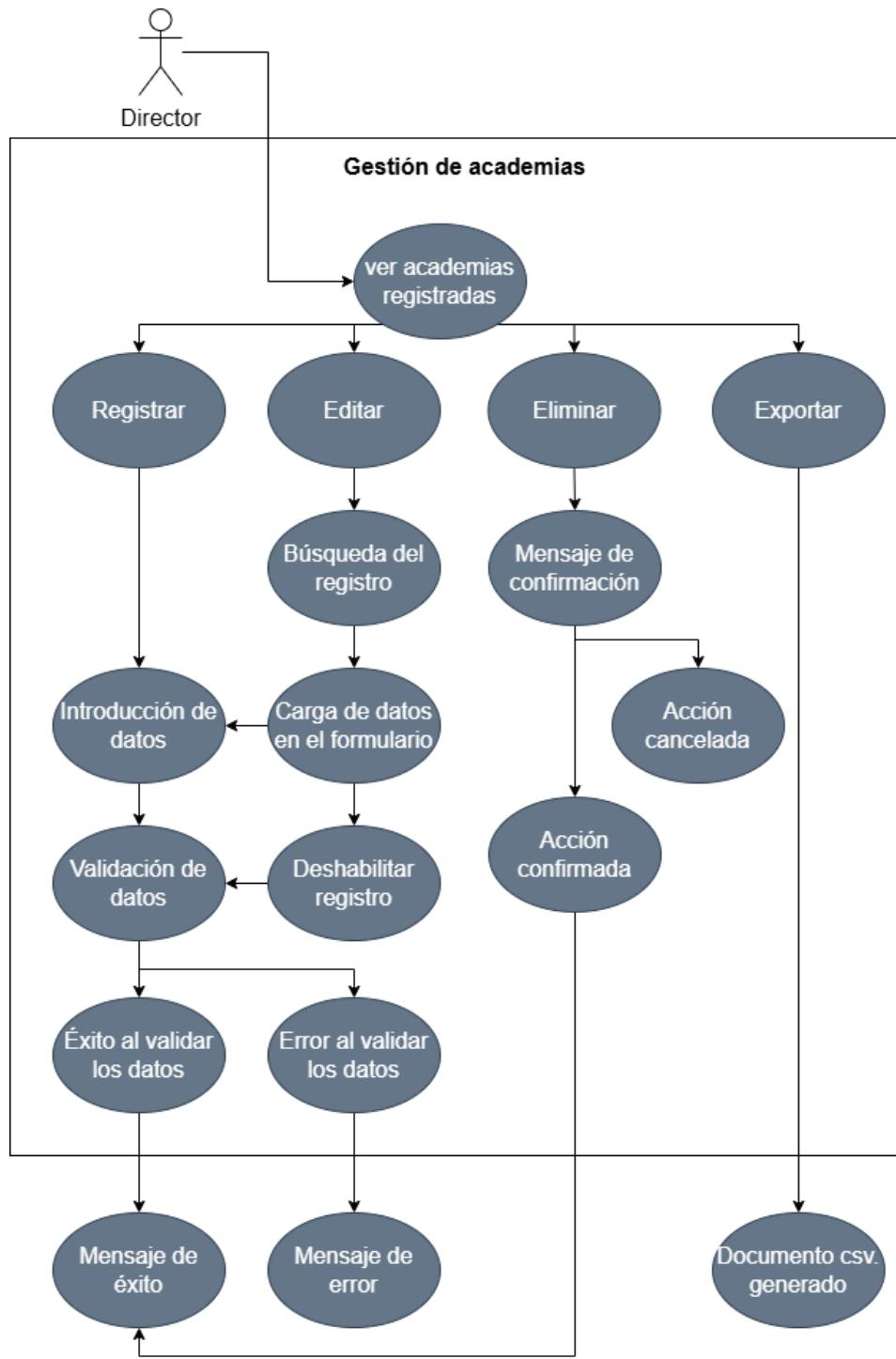
- **Evaluador:** Acceso a gestión de estudiantes, gestión de emparejamientos y puntajes. En este contexto, el rol de evaluador está definido para los instructores de la academia.
- **Director:** Permisos administrativos extendidos, reportes de resultados, historial de cambios en el sistema y gestión global de la academia Estrellas del Ajedrez. El rol de evaluador es el equivalente a un rol Administrador.
- **Estudiante:** Acceso a su perfil personal y consulta de los torneos en los que están inscritos, así como de los resultados y emparejamientos de esos torneos. Este rol carece de funciones de gestión en el sistema; su único propósito es brindar los detalles y estadísticas personales en el desarrollo de los torneos de ajedrez.

Figura 13: Diagrama de caso de uso No. 3



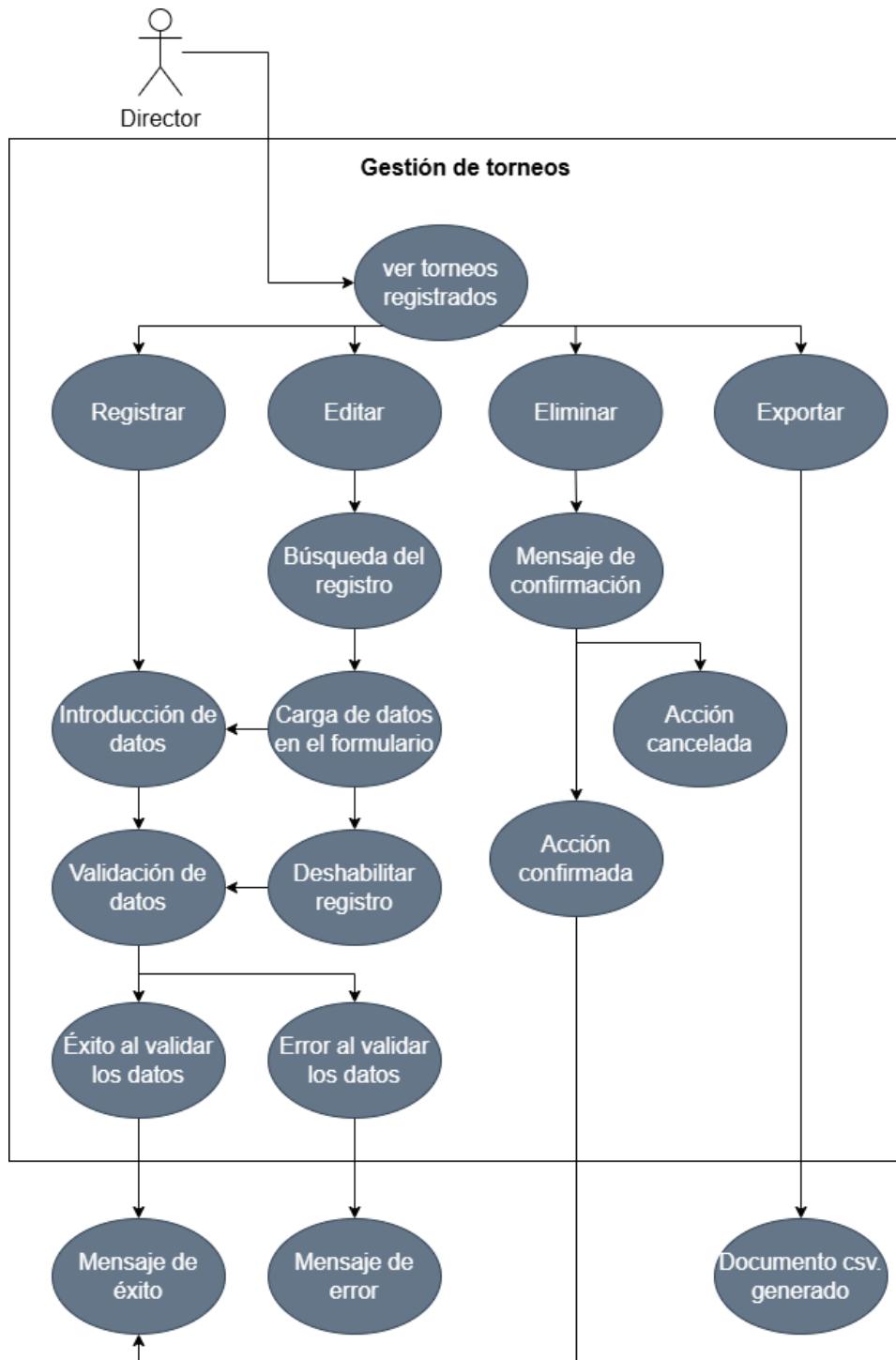
Nota: La ilustración muestra el Caso de Uso correspondiente al módulo de gestión de perfiles de usuario.

Figura 14: Diagrama de caso de uso No. 4



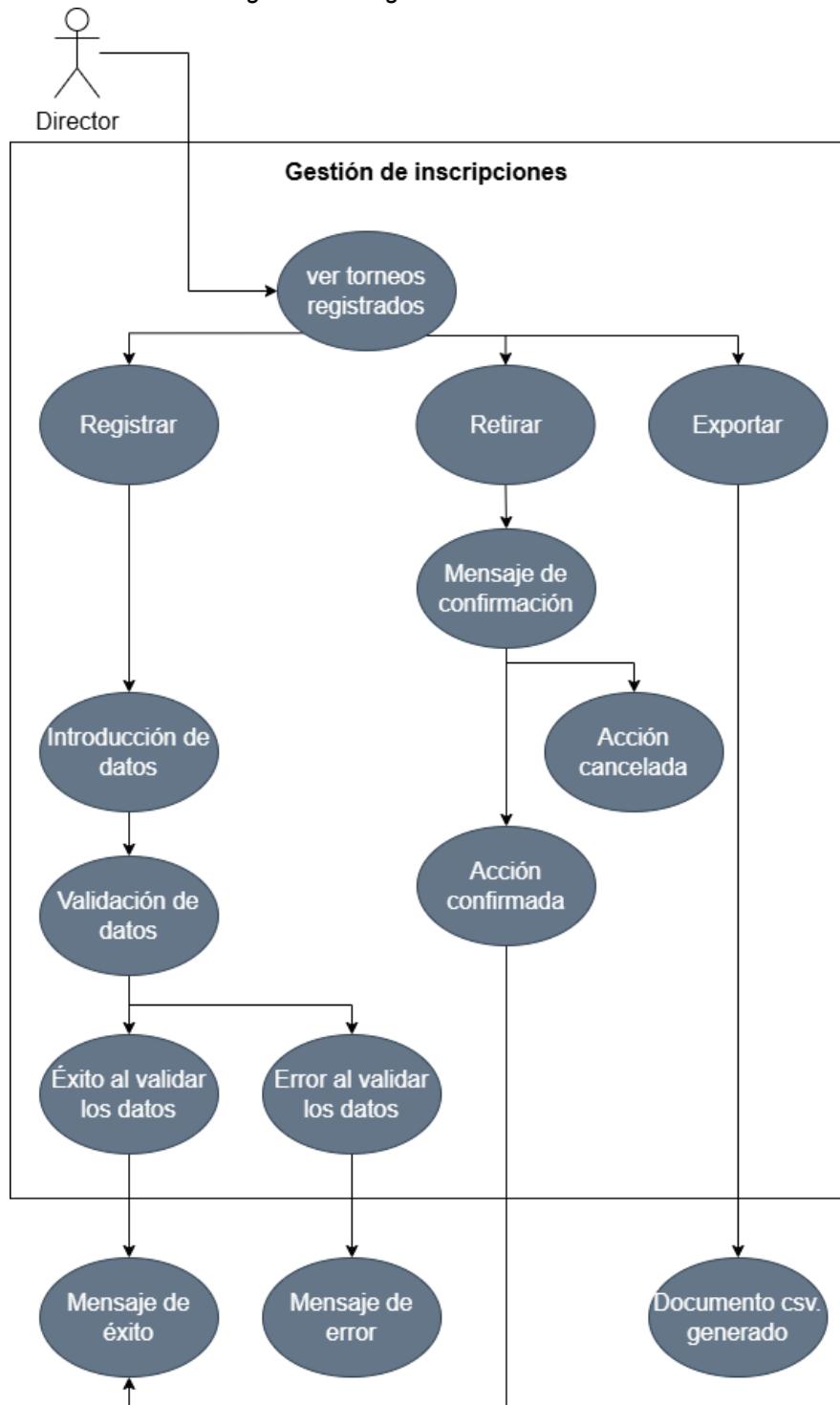
Nota: La ilustración muestra el Caso de Uso correspondiente al módulo de gestión de academias

Figura 15: Diagrama de caso de uso No. 5



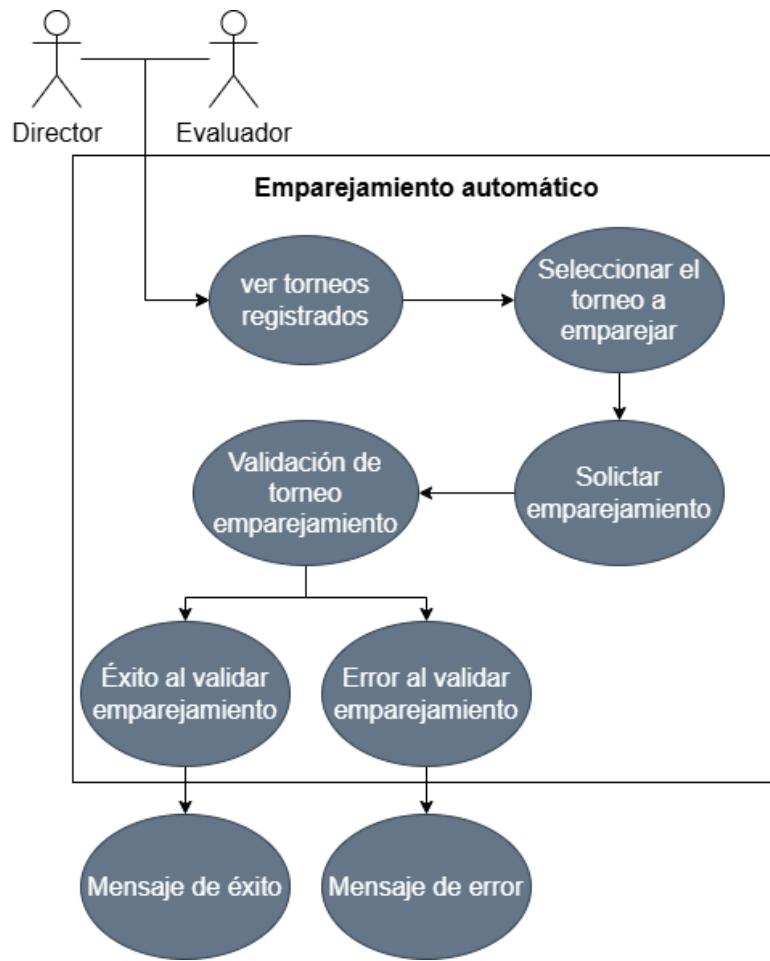
Nota: La ilustración muestra el Caso de Uso correspondiente al módulo de gestión de torneos.

Figura 16: Diagrama de caso de uso No. 6



Nota: La ilustración muestra el Caso de Uso correspondiente al módulo de gestión de inscripciones.

Figura 17: Diagrama de caso de uso No. 7



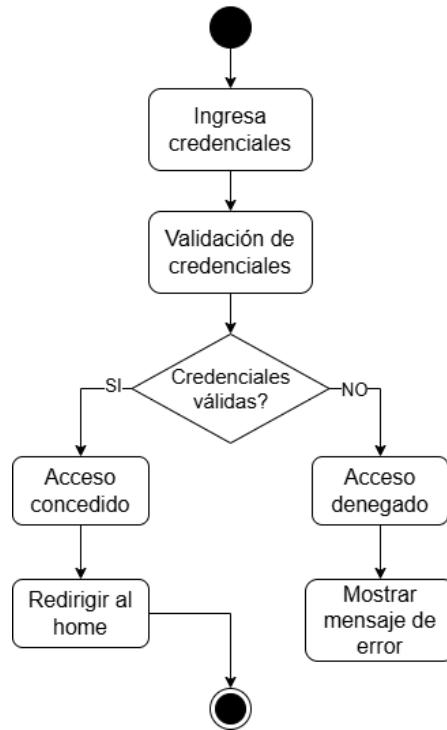
Nota: La ilustración muestra el Caso de Uso correspondiente al proceso de emparejamiento automático.

Diagrama de Actividades

Los diagramas de actividades describen los flujos de trabajo del sistema, representando de manera visual las secuencias de acciones y decisiones tanto de los usuarios como del sistema.

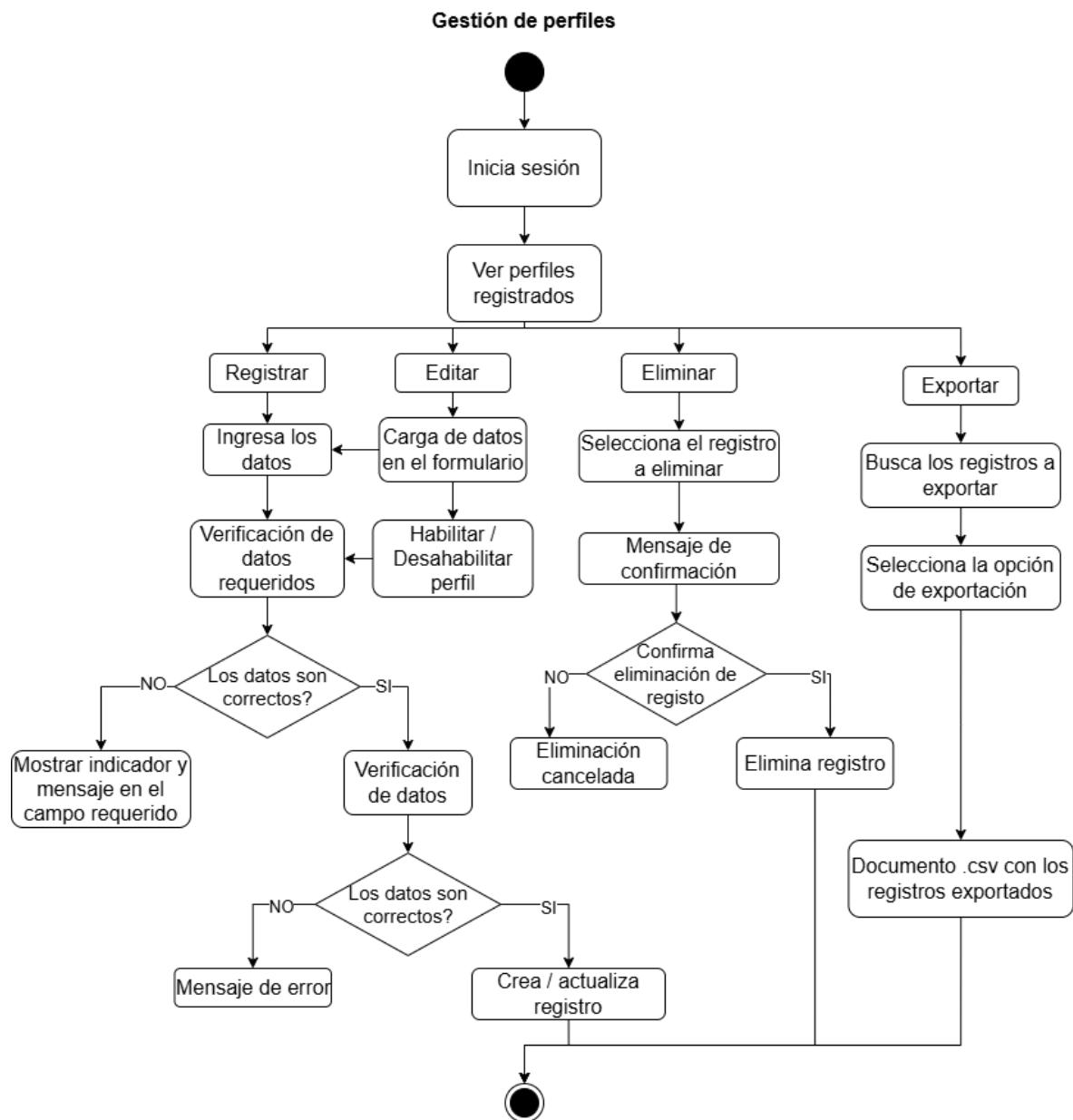
Figura 18: Diagrama de actividades No. 1

Inicio de sesión



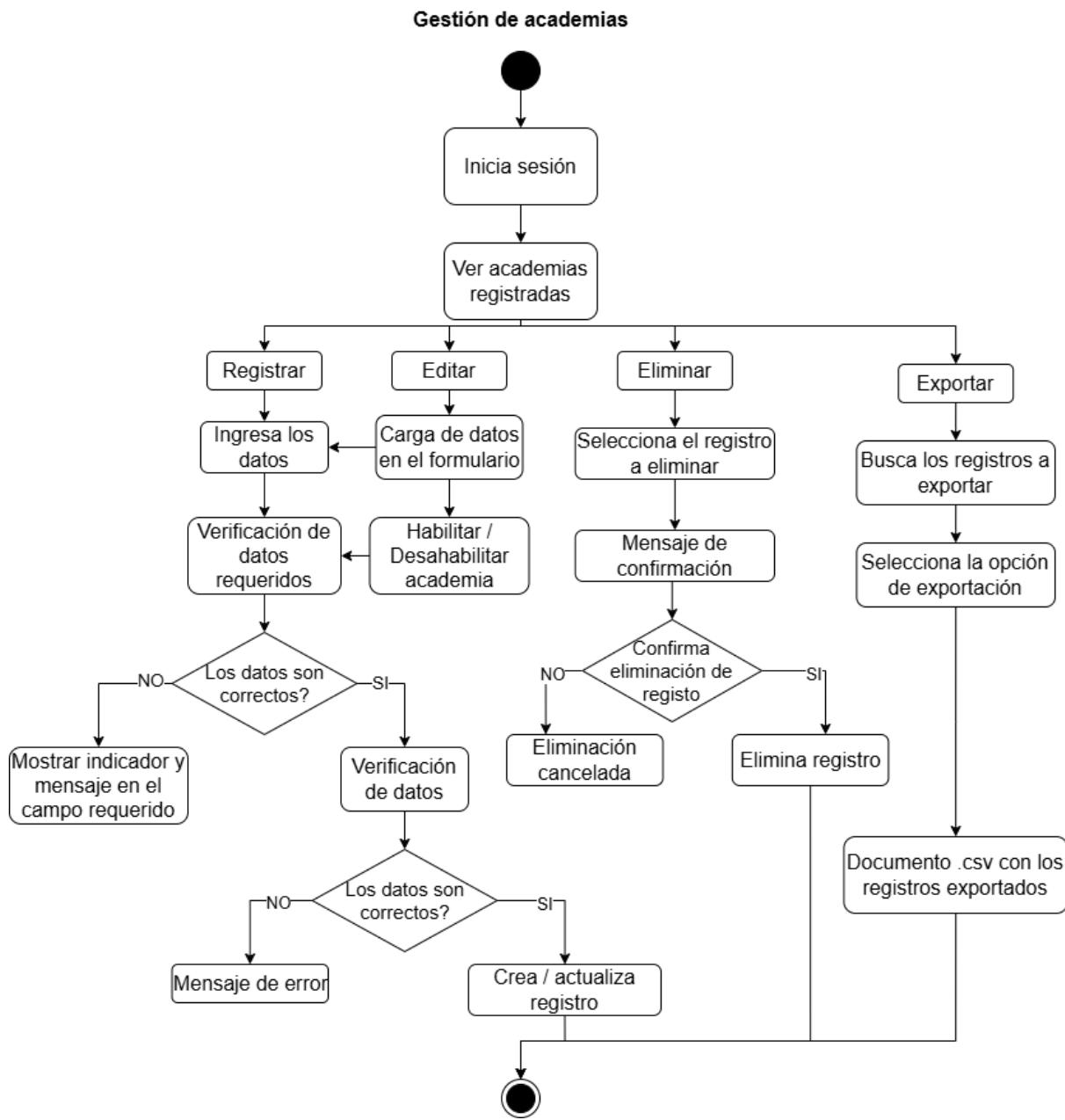
Nota: La ilustración muestra el Diagrama de Actividades correspondiente al proceso de autenticación.

Figura 19: Diagrama de actividades No. 2



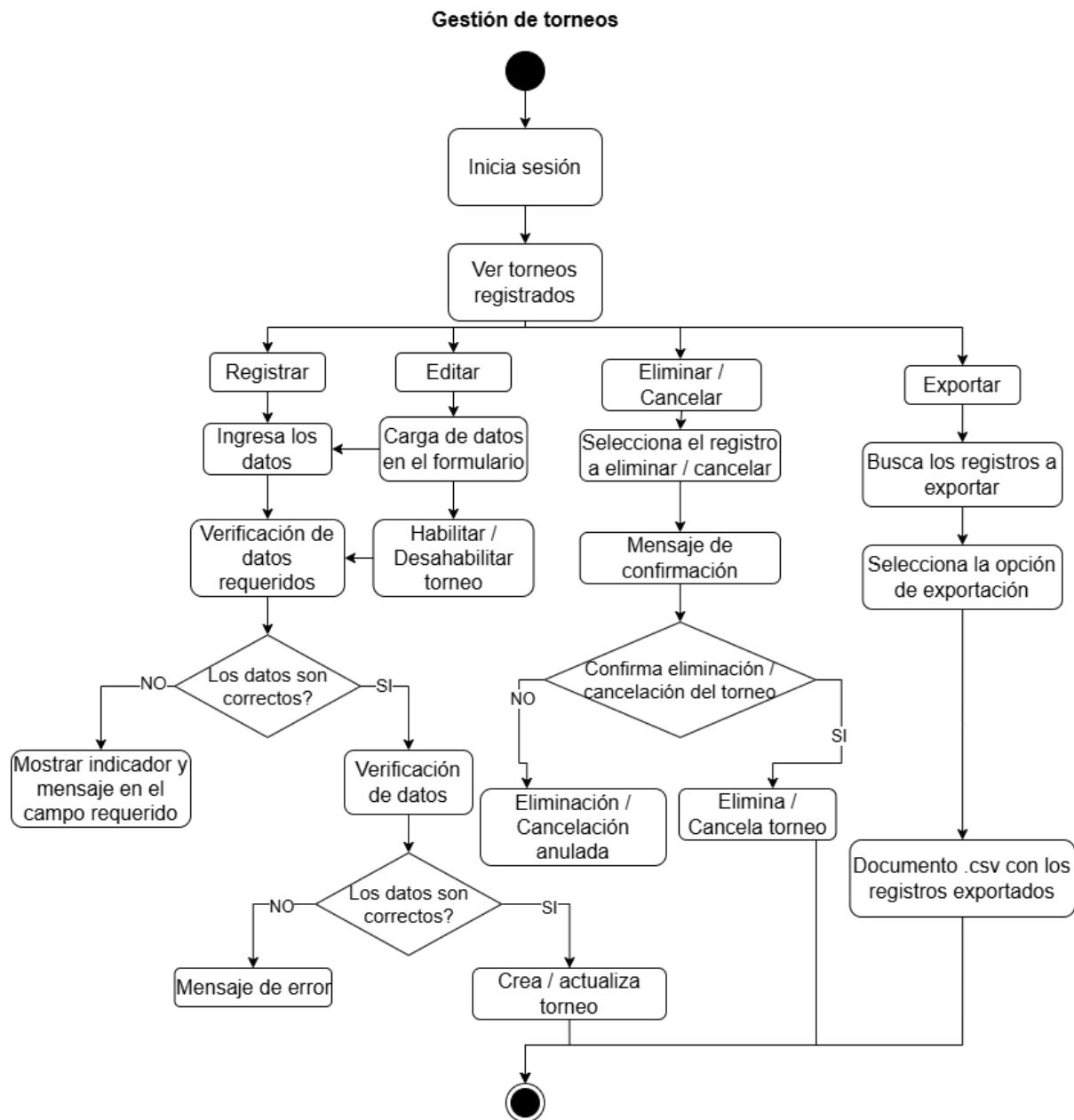
Nota: La ilustración muestra el Diagrama de Actividades correspondiente al módulo de gestión de perfiles.

Figura 20: Diagrama de actividades No. 3



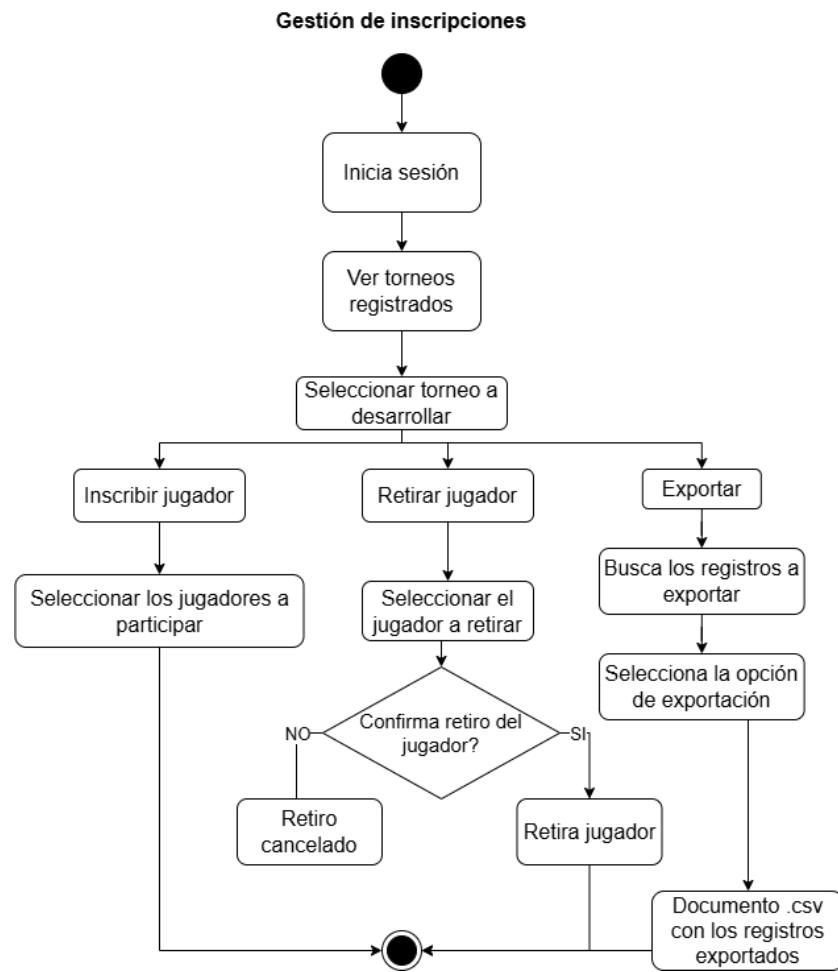
Nota: La ilustración muestra el Diagrama de Actividades correspondiente al módulo de gestión de academias.

Figura 21: Diagrama de actividades No. 4



Nota: La ilustración muestra el Diagrama de Actividades correspondiente al módulo de gestión de torneos.

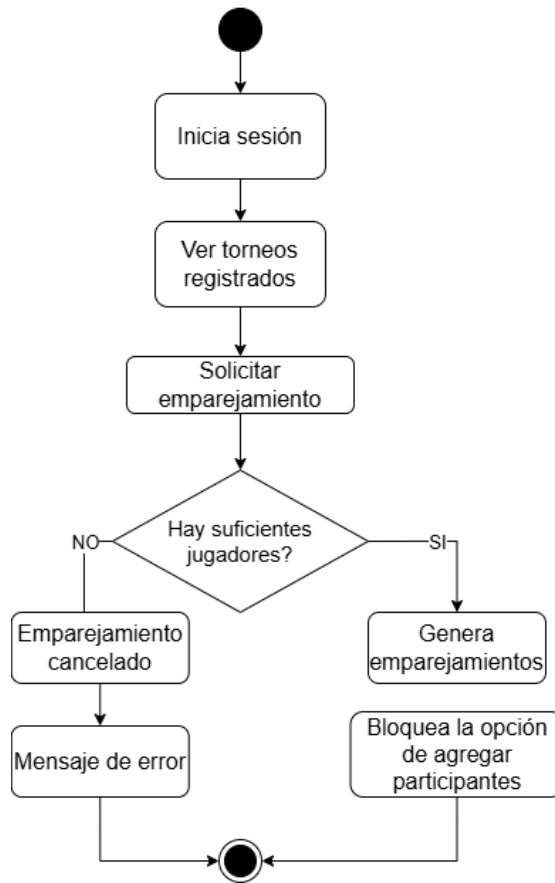
Figura 22: Diagrama de actividades No. 5



Nota: La ilustración muestra el Diagrama de Actividades correspondiente al módulo de gestión de inscripciones.

Figura 23: Diagrama de actividades No. 6

Emparejamiento automático

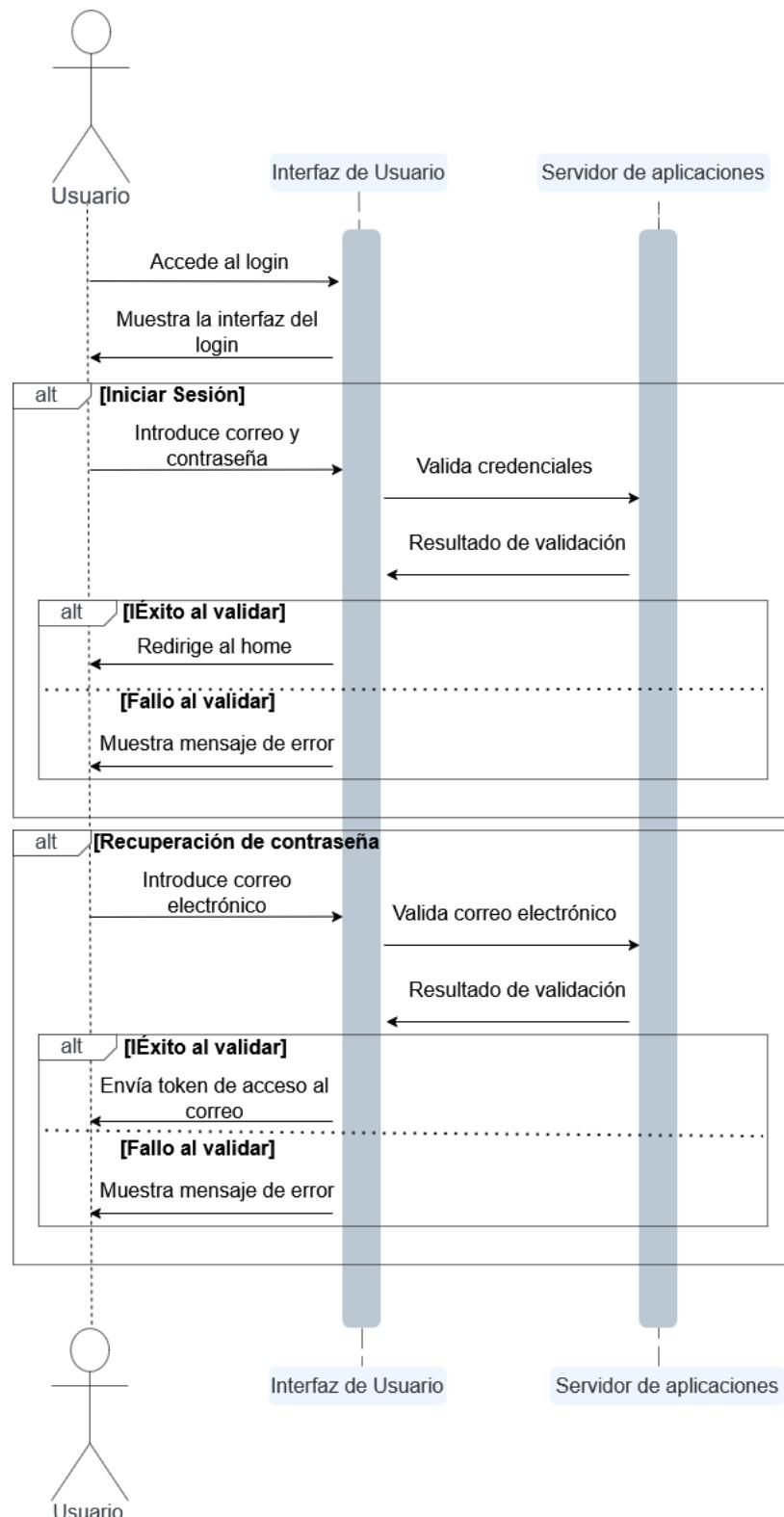


Nota: La ilustración muestra el Diagrama de Actividades correspondiente al emparejamiento automático de partidas.

Diagrama de Secuencia

Los Diagramas de Secuencia ilustran cómo interactúan los usuarios con el sistema a través de sus diferentes capas, detallando el flujo y la secuencia de las llamadas y respuestas. Su propósito es mostrar de manera clara cómo se lleva a cabo la lógica de negocio y los procesos internos del sistema, permitiendo comprender mejor la dinámica de las interacciones y el comportamiento del sistema a lo largo de un escenario específico.

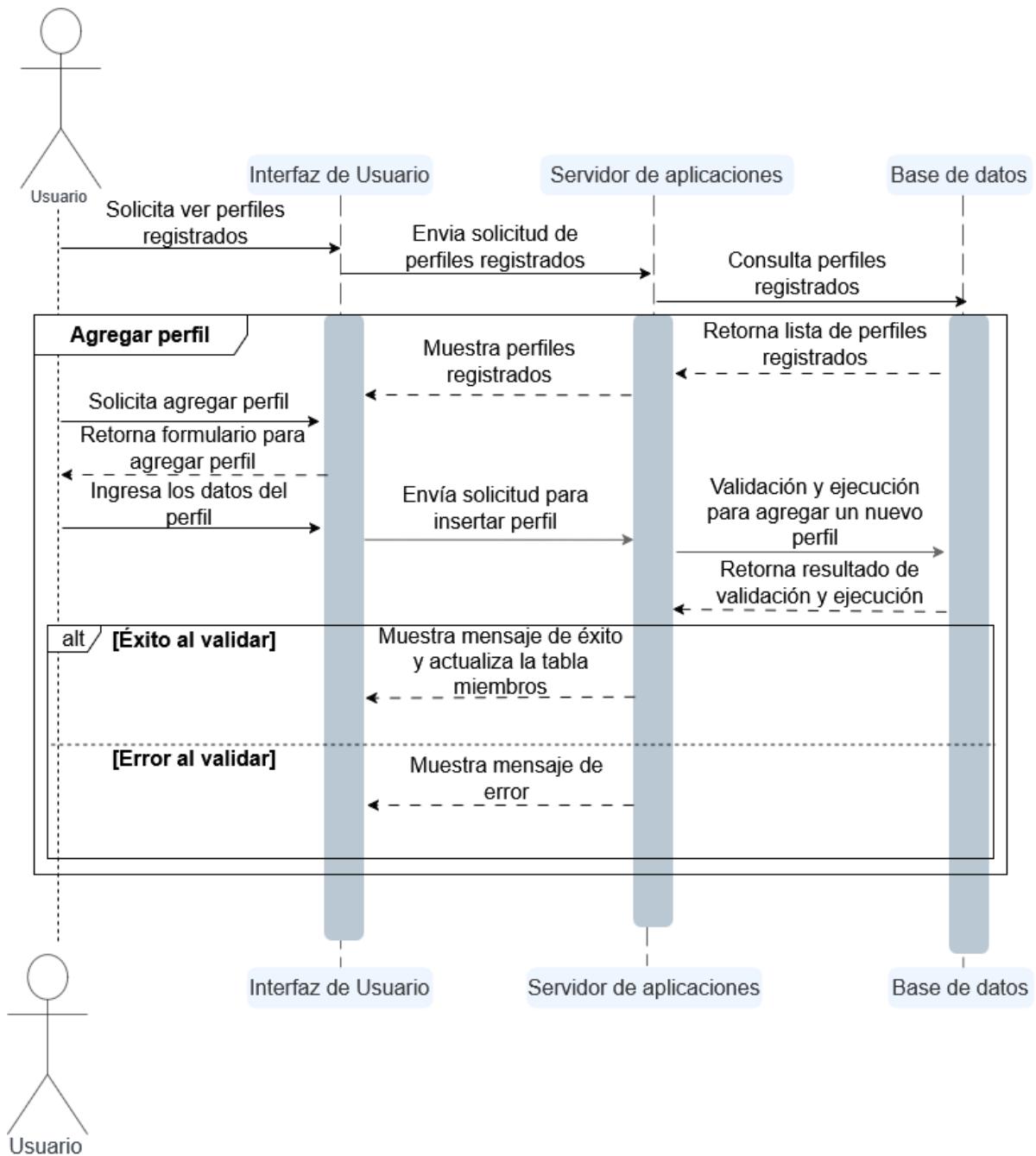
Figura 24: Diagrama de secuencia No. 1



Nota: La ilustración muestra el Diagrama de Secuencia correspondiente al inicio de sesión

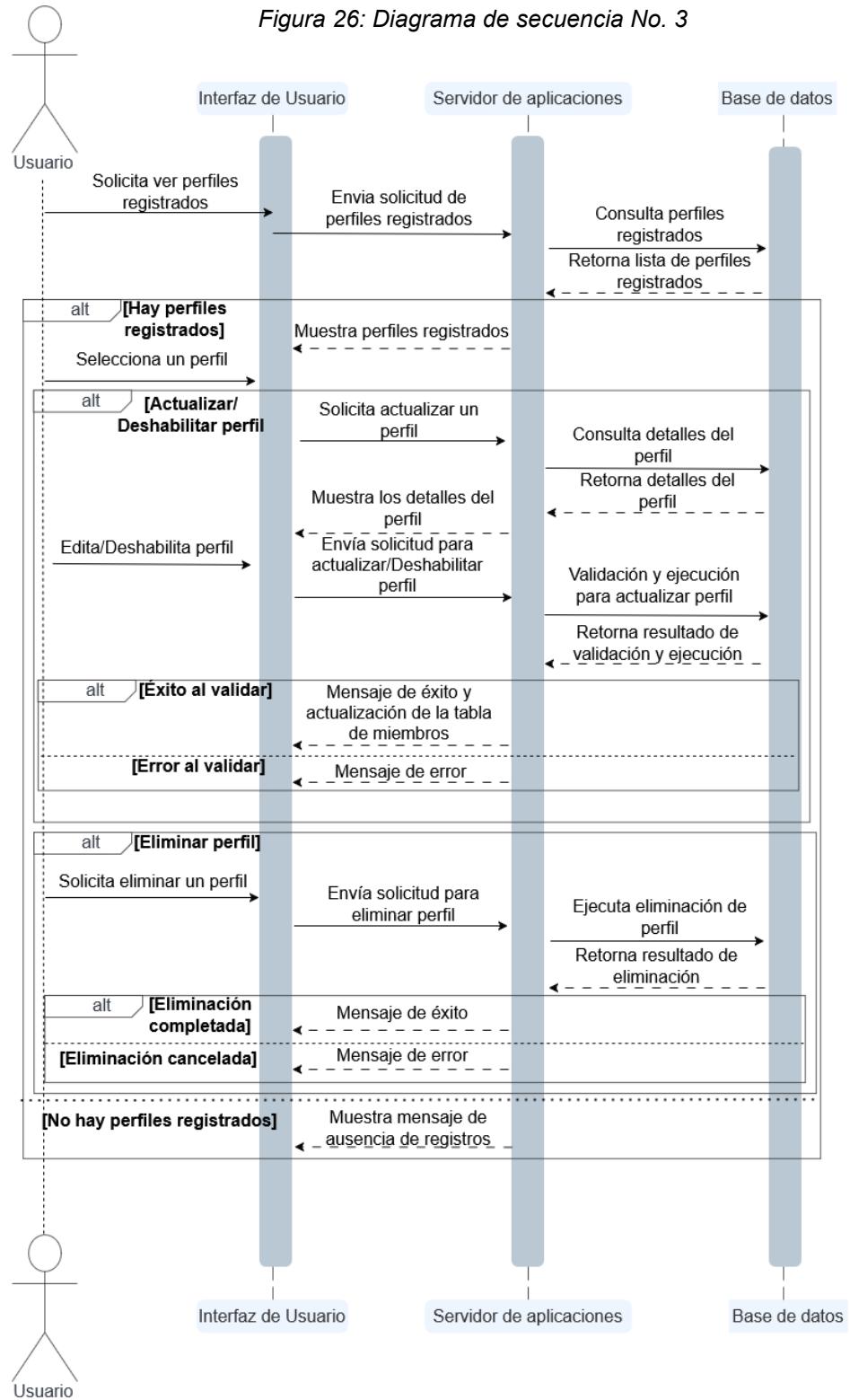
y recuperación de contraseña.

Figura 25: Diagrama de secuencia No. 2



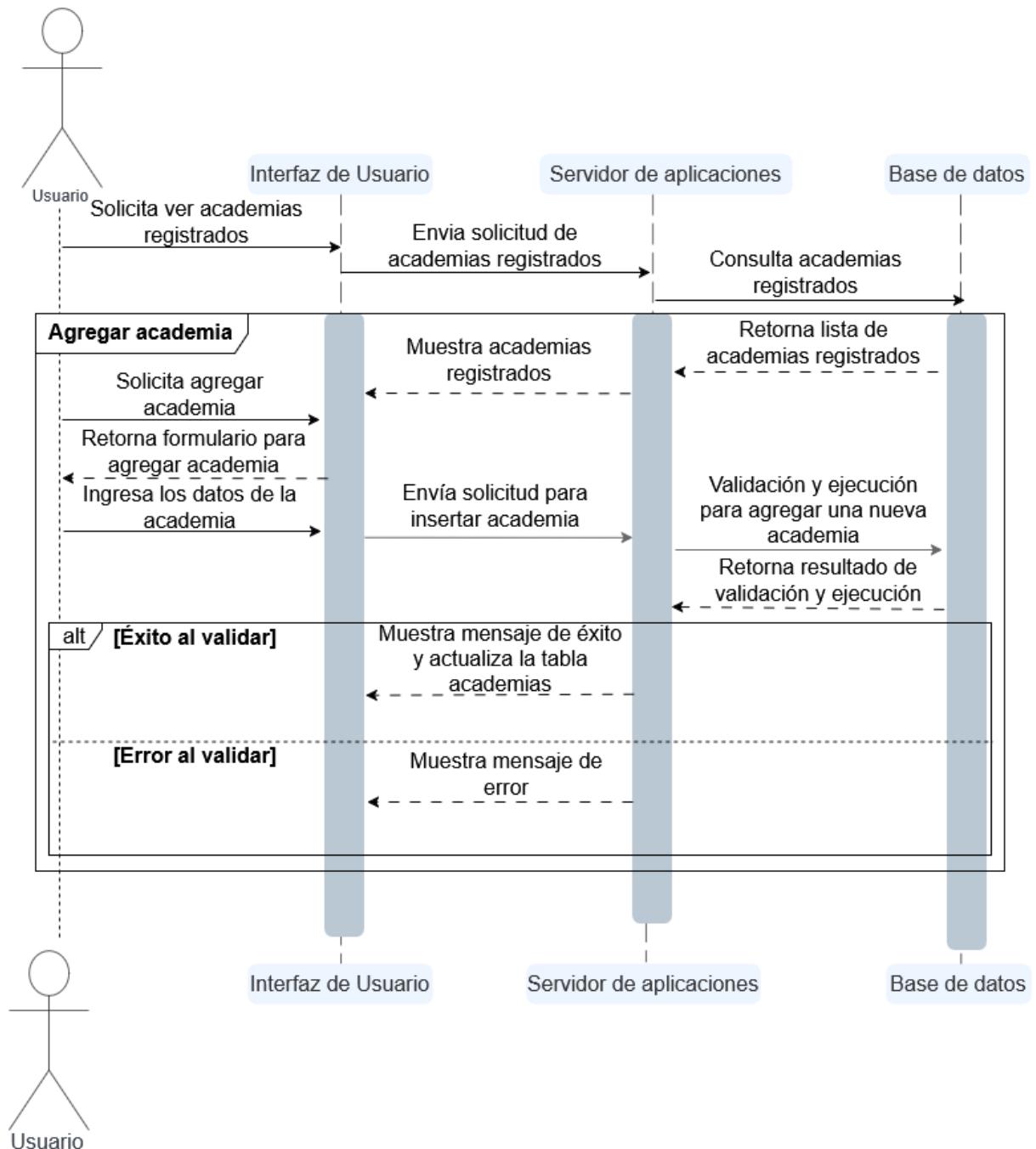
Nota: La ilustración muestra el Diagrama de Secuencia correspondiente al proceso para registrar perfiles de usuario

Figura 26: Diagrama de secuencia No. 3



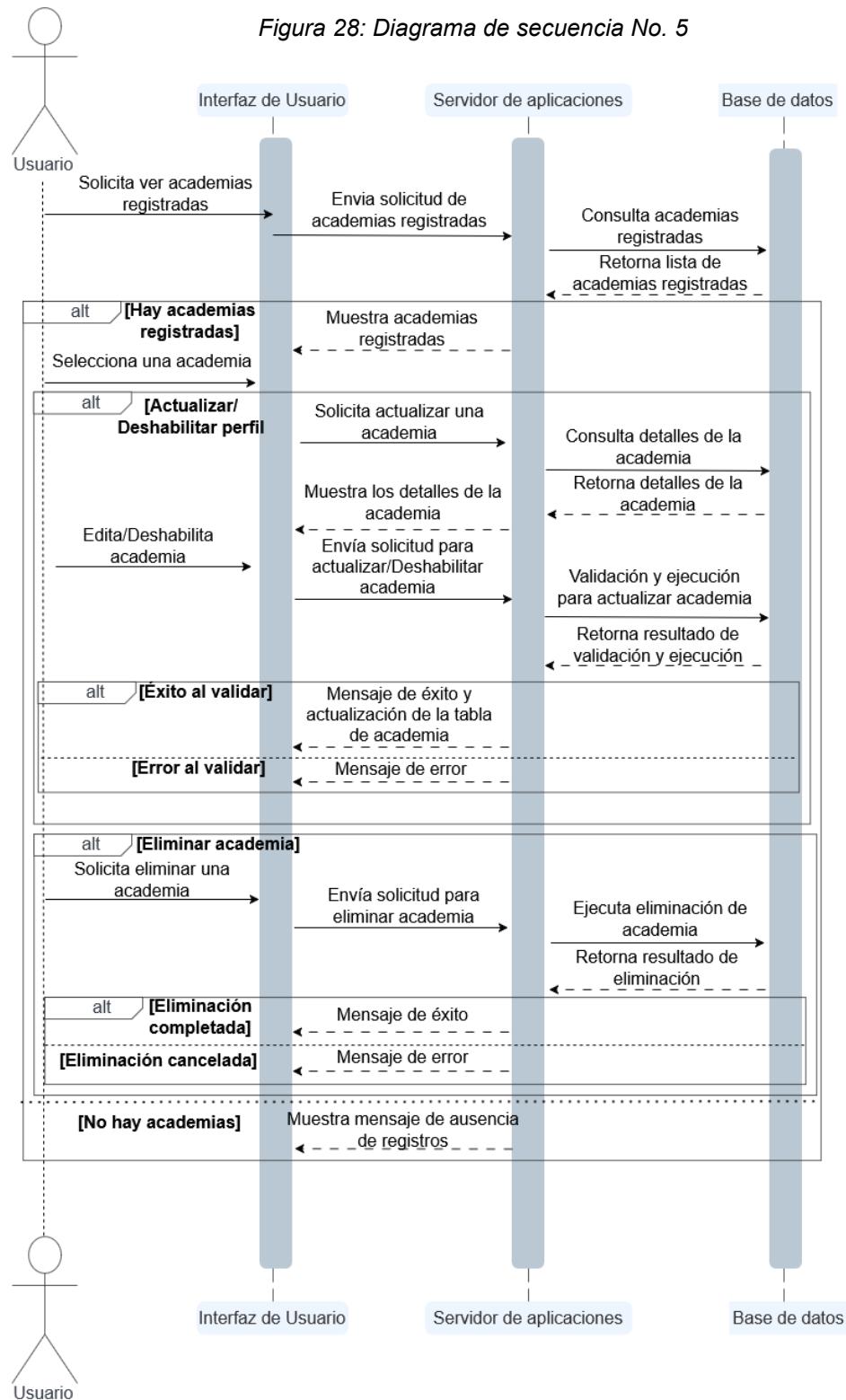
Nota: La ilustración muestra el Diagrama de Secuencia correspondiente al proceso de modificación y eliminación de perfiles de usuarios registrados.

Figura 27: Diagrama de secuencia No. 4



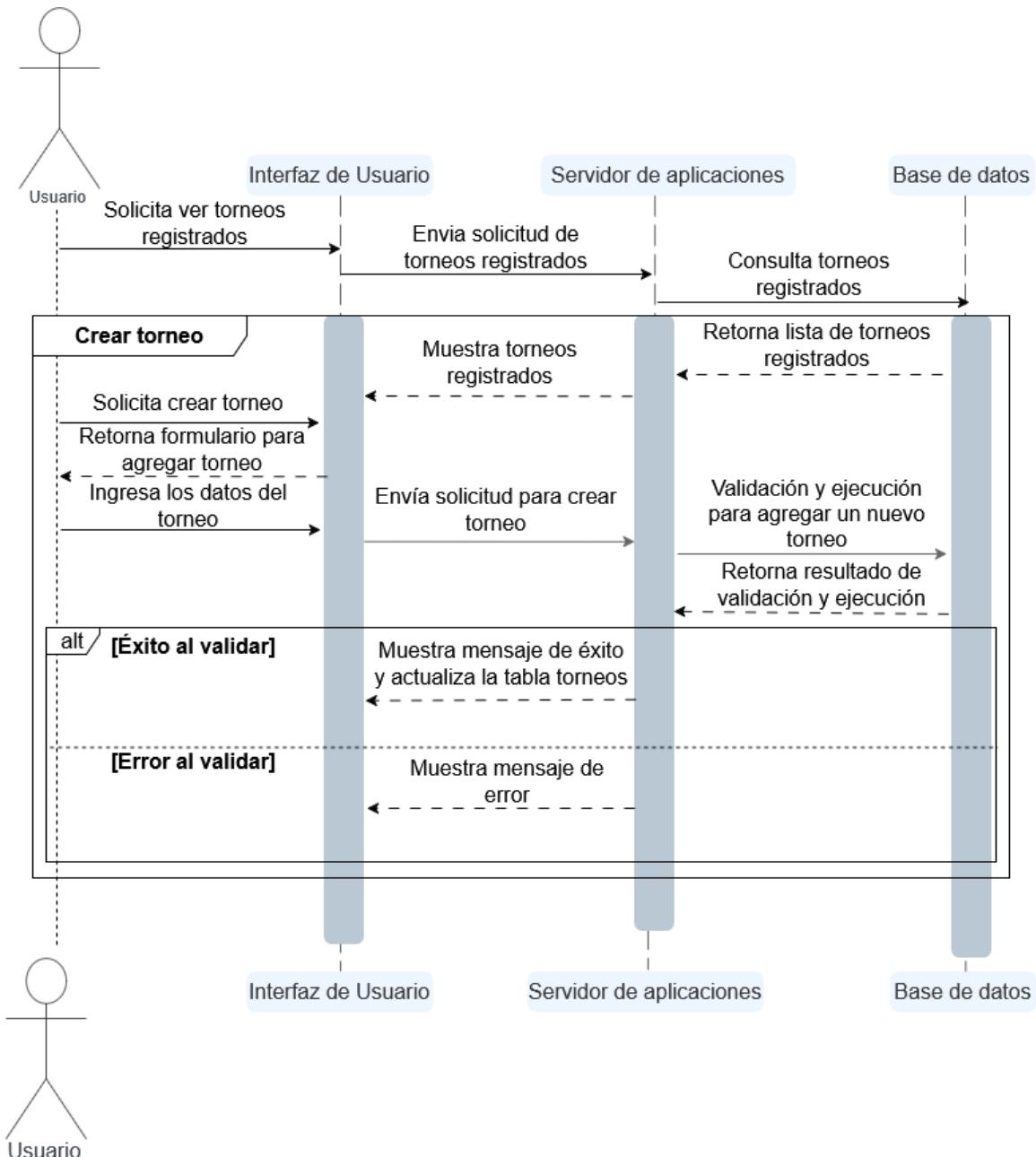
Nota: La ilustración muestra el Diagrama de Secuencia correspondiente al proceso para registrar una academia.

Figura 28: Diagrama de secuencia No. 5



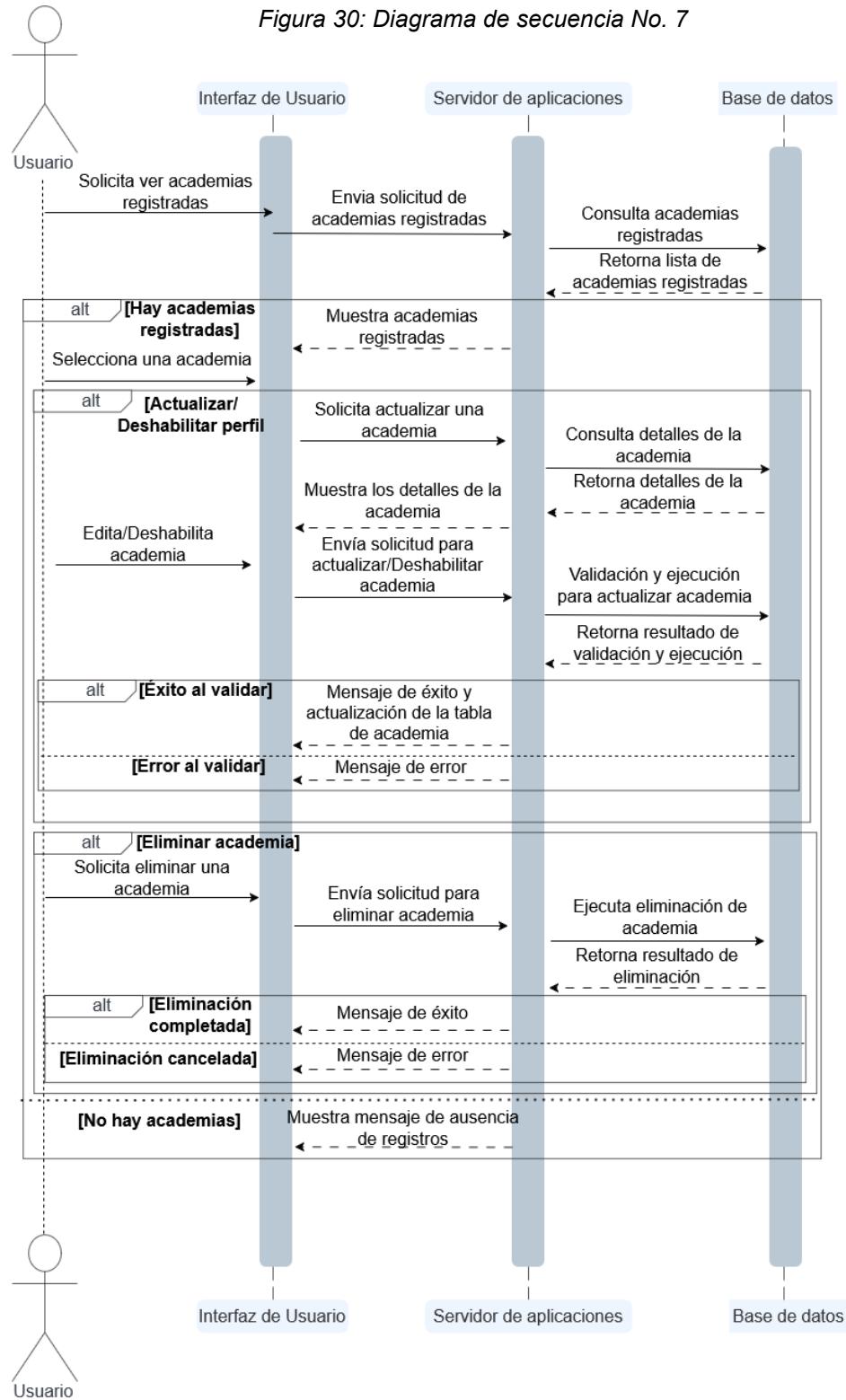
Nota: La ilustración muestra el Diagrama de Secuencia correspondiente al proceso para modificar y eliminar academias registradas.

Figura 29: Diagrama de secuencia No. 6



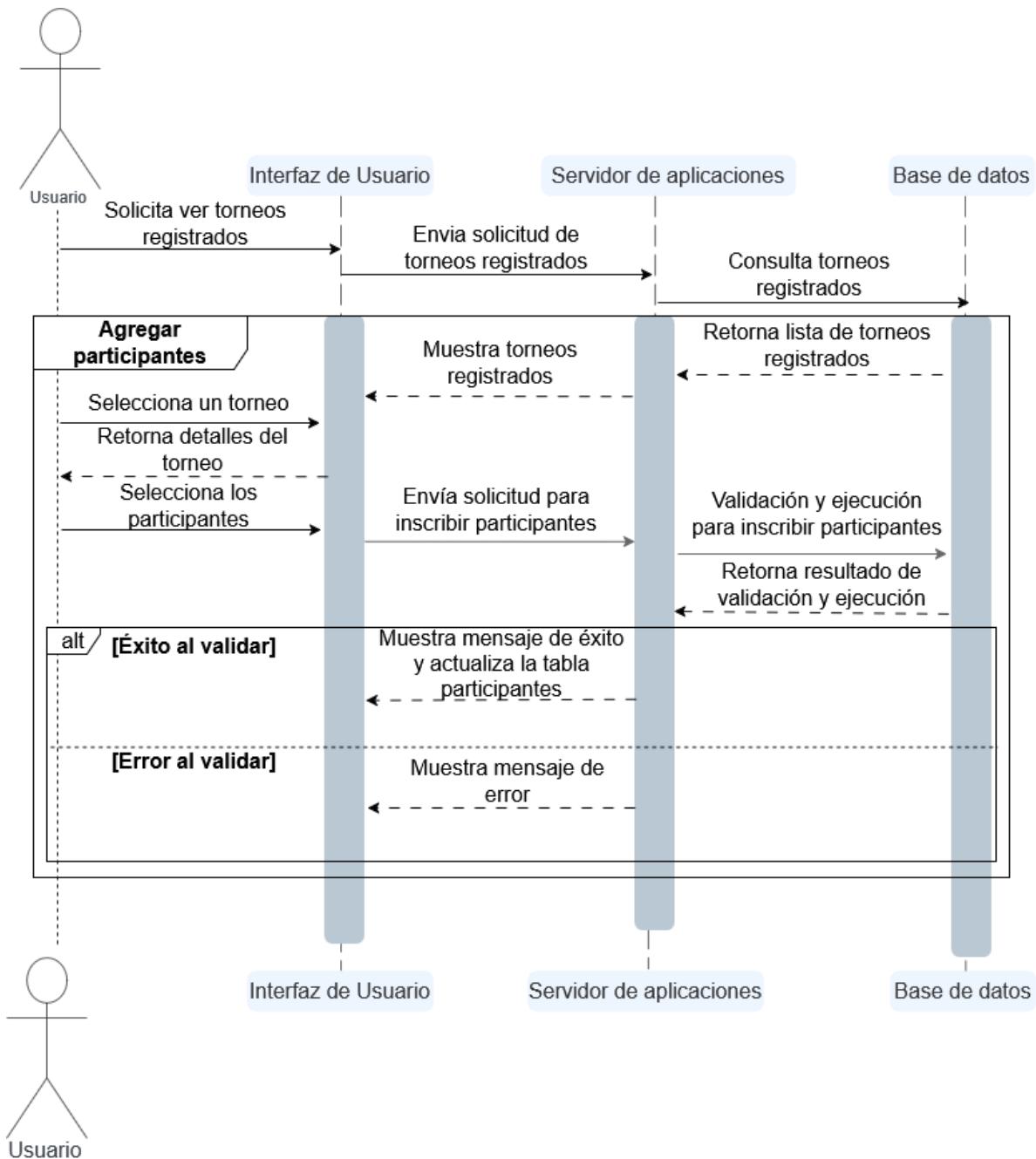
Nota: La ilustración muestra el Diagrama de Secuencia correspondiente al proceso para registrar torneos.

Figura 30: Diagrama de secuencia No. 7



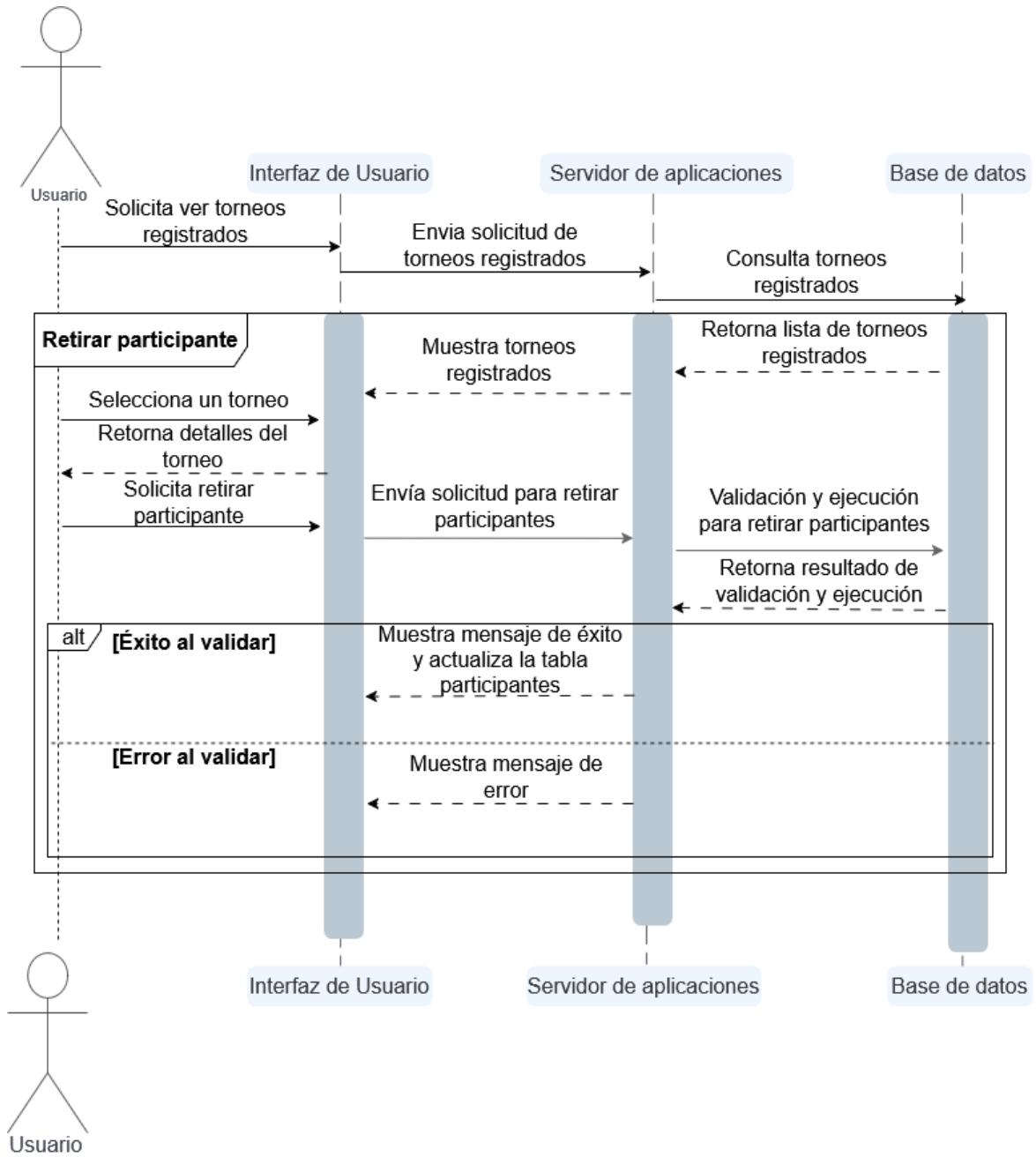
Nota: La ilustración muestra el Diagrama de Secuencia correspondiente al proceso para modificar y eliminar torneos registrados.

Figura 31: Diagrama de secuencia No. 8



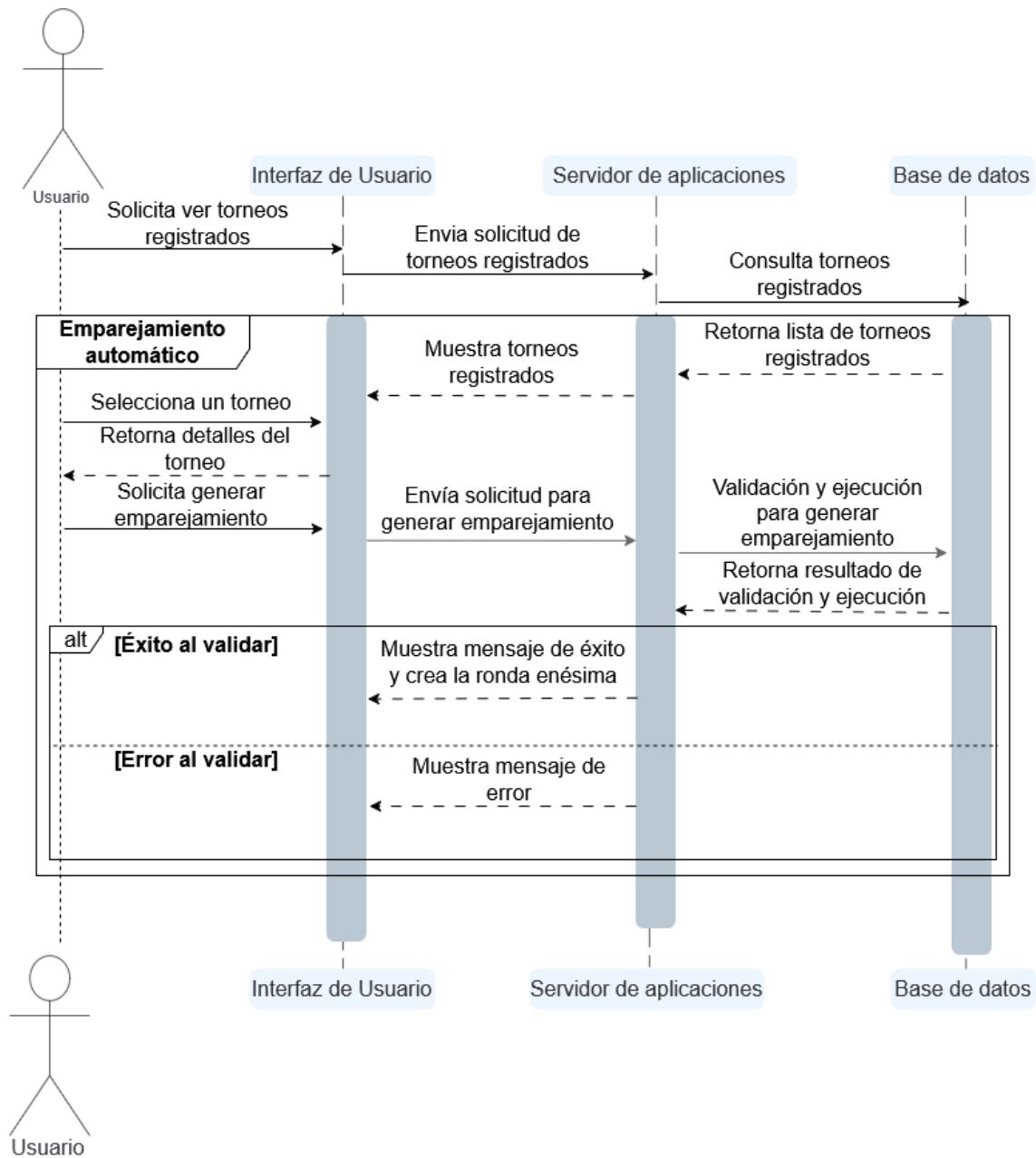
Nota: La ilustración muestra el Diagrama de Secuencia correspondiente al proceso para registrar inscripciones.

Figura 32: Diagrama de secuencia No. 9



Nota: La ilustración muestra el Diagrama de Secuencia correspondiente al proceso para eliminar inscripciones registradas.

Figura 33: Diagrama de secuencia No. 10



Nota: La ilustración muestra el Diagrama de Secuencia correspondiente al proceso para el emparejamiento automático de rondas.

6.3.1.3. Capa de infraestructura (Infrastructure)

La Capa de Infraestructura juega un rol fundamental en la arquitectura del sistema web de gestión de torneos de ajedrez, ya que se encarga de la interacción con los recursos externos necesarios, tales como bases de datos y almacenamiento de información. Su función principal es facilitar el acceso a estos recursos de manera organizada, eficiente y segura, permitiendo que las demás capas del sistema operen de manera fluida y sin contratiempos.

Responsabilidades de la Capa de Infraestructura

La Capa de Infraestructura desempeña diversas funciones entre las que se incluyen:

- **Gestión de Datos:** Esta capa es responsable de implementar los repositorios necesarios para manejar las operaciones de acceso y gestión de datos relacionados con los torneos, jugadores, resultados y otros elementos clave del sistema. Esto abarca las operaciones de creación, lectura, actualización y eliminación (CRUD) de la información almacenada.
- **Integración con Servicios Externos:** La infraestructura facilita la comunicación con servicios externos, como APIs de terceros y cualquier otra herramienta necesaria para el funcionamiento del sistema web de gestión de torneos.
- **Configuración y Gestión de Recursos:** Esta capa se encarga de la configuración y administración de los recursos técnicos esenciales, tales como la conexión a bases de datos y otros recursos de infraestructura que aseguran la estabilidad y eficiencia del sistema durante la organización y ejecución de los torneos.

Repositorios

Los repositorios en la Capa de Infraestructura son implementaciones específicas de las interfaces definidas en la Capa de Dominio. Estos repositorios se encargan de:

- **Acceso a Bases de Datos:** Proporcionan métodos para acceder y manipular los datos en la base de datos, utilizando tecnologías como Eloquent y Query Builder, con la base de

datos MySQL. Estos repositorios permiten realizar operaciones CRUD de manera eficiente, optimizando el acceso a la información.

- **Implementación de Patrones de Diseño:** Se pueden aplicar patrones de diseño, como el patrón Unit of Work y el patrón Repository, para facilitar la gestión de transacciones y la abstracción del acceso a los datos. Además, la interacción con el frontend se gestiona mediante Ajax, lo que permite cargar dinámicamente los datos en tablas utilizando DataTables, sin necesidad de recargar la página.

Comunicación con la Capa de Aplicación

La Capa de Infraestructura se comunica con la Capa de Aplicación a través de interfaces, lo que ofrece varias ventajas clave para el funcionamiento del sistema de gestión de torneos de ajedrez. Este enfoque facilita tanto el desacoplamiento como la flexibilidad en el diseño de la aplicación.

- **Desacoplamiento:** Al utilizar interfaces entre las capas, conseguimos un desacoplamiento entre la lógica de negocio, que se encuentra en la Capa de Aplicación, y la lógica de acceso a datos en la Capa de Infraestructura. Esto no solo mejora la organización y mantenimiento del sistema, sino que también facilita la realización de pruebas unitarias y de integración.
- **Flexibilidad:** La estructura basada en interfaces permite cambiar la implementación de los repositorios de forma sencilla sin tener que modificar las capas superiores del sistema. En el contexto de un sistema como el de gestión de torneos de ajedrez, esto es crucial ya que nos brinda la flexibilidad para adaptarnos rápidamente a nuevas tecnologías o cambios en los requisitos del sistema.

Diagrama de Infraestructura

A continuación, se presentarán los diagramas clave que muestran la arquitectura de la Capa de Infraestructura del sistema de gestión de torneos de ajedrez. Estos diagramas incluyen

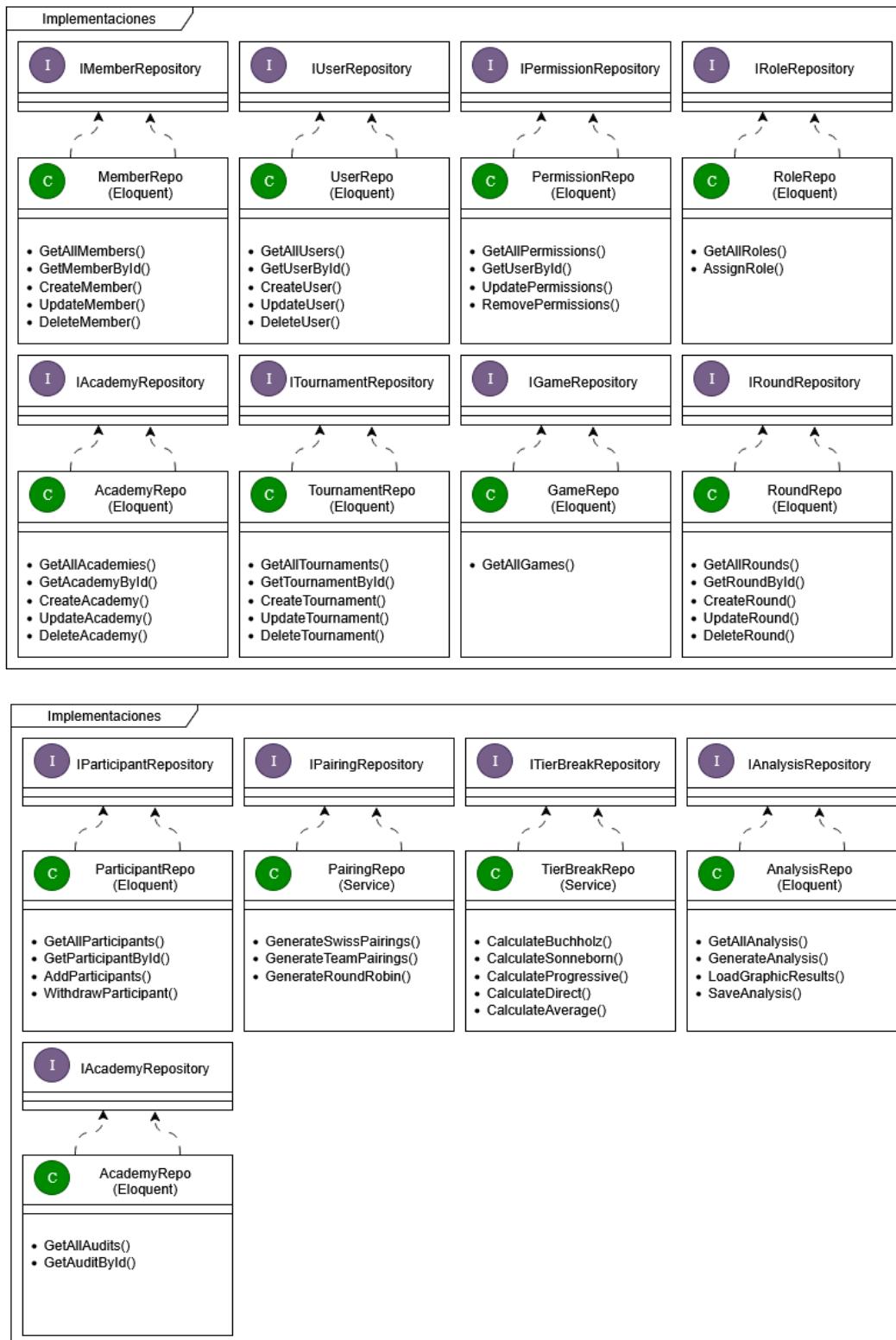
el Diagrama de Arquitectura, las Interfaces de Repositorios, las Implementaciones de Repositorios y el Diagrama de Base de Datos. Por medio de estos diagramas, se proporcionará una visión más clara de cómo está estructurado el sistema y cómo interactúan los diferentes componentes entre sí. Esto facilita la comprensión respecto a cómo fluye la información a través del sistema.

Figura 34: Diagrama de Interfaces de Repositorio

Interfaces de repositorio			
I IMemberRepository	I IUserRepository	I IPermissionRepository	I IRoleRepository
<ul style="list-style-type: none"> • GetAllMembers() • GetMemberById() • CreateMember() • UpdateMember() • DeleteMember() 	<ul style="list-style-type: none"> • GetAllUsers() • GetUserById() • CreateUser() • UpdateUser() • DeleteUser() 	<ul style="list-style-type: none"> • GetAllPermissions() • GetUserById() • UpdatePermissions() • RemovePermissions() 	<ul style="list-style-type: none"> • GetAllRoles() • AssignRole()
I IAcademyRepository	I ITournamentRepository	I IGameRepository	I IRoundRepository
<ul style="list-style-type: none"> • GetAllAcademies() • GetAcademyById() • CreateAcademy() • UpdateAcademy() • DeleteAcademy() 	<ul style="list-style-type: none"> • GetAllTournaments() • GetTournamentById() • CreateTournament() • UpdateTournament() • DeleteTournament() 	<ul style="list-style-type: none"> • GetAllGames() 	<ul style="list-style-type: none"> • GetAllRounds() • GetRoundById() • CreateRound() • UpdateRound() • DeleteRound()
I IParticipantRepository	I IPairingRepository	I ITierBreakRepository	I IAnalysisRepository
<ul style="list-style-type: none"> • GetAllParticipants() • GetParticipantById() • AddParticipants() • WithdrawParticipant() 	<ul style="list-style-type: none"> • GenerateSwissPairings() • GenerateTeamPairings() • GenerateRoundRobin() 	<ul style="list-style-type: none"> • CalculateBuchholz() • CalculateSonneborn() • CalculateProgressive() • CalculateDirect() • CalculateAverage() 	<ul style="list-style-type: none"> • GetAllAnalysis() • GenerateAnalysis() • LoadGraphicResults() • SaveAnalysis()
I IAuditRepository			
<ul style="list-style-type: none"> • GetAllAudits() • GetAuditById() 			

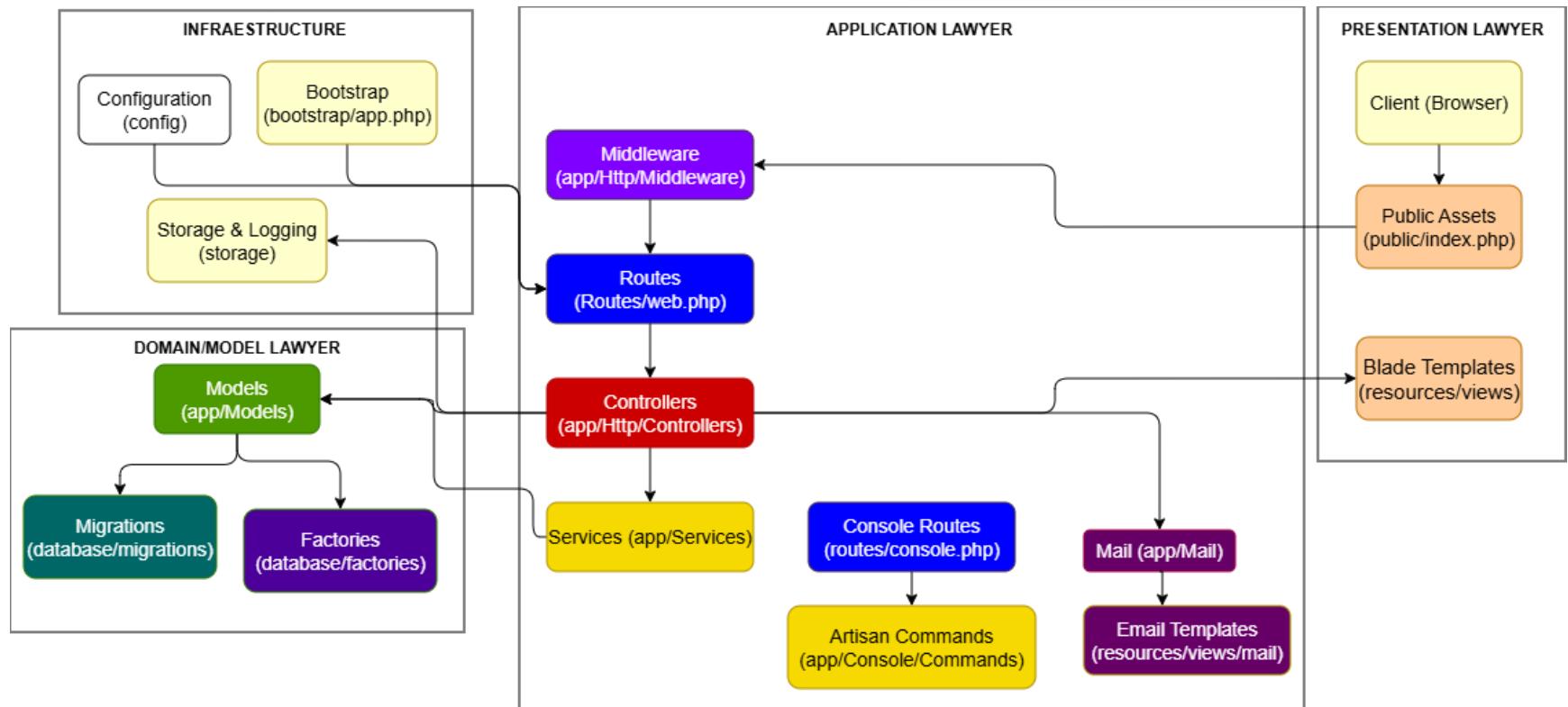
Nota: La ilustración muestra un conjunto de interfaces de repositorios para gestionar entidades como Tournament, Member, Academy, entre otros. Estas interfaces encapsulan las operaciones de acceso a datos para mantener un diseño limpio y desacoplado.

Figura 35: Diagrama de Implementaciones de Repositorio



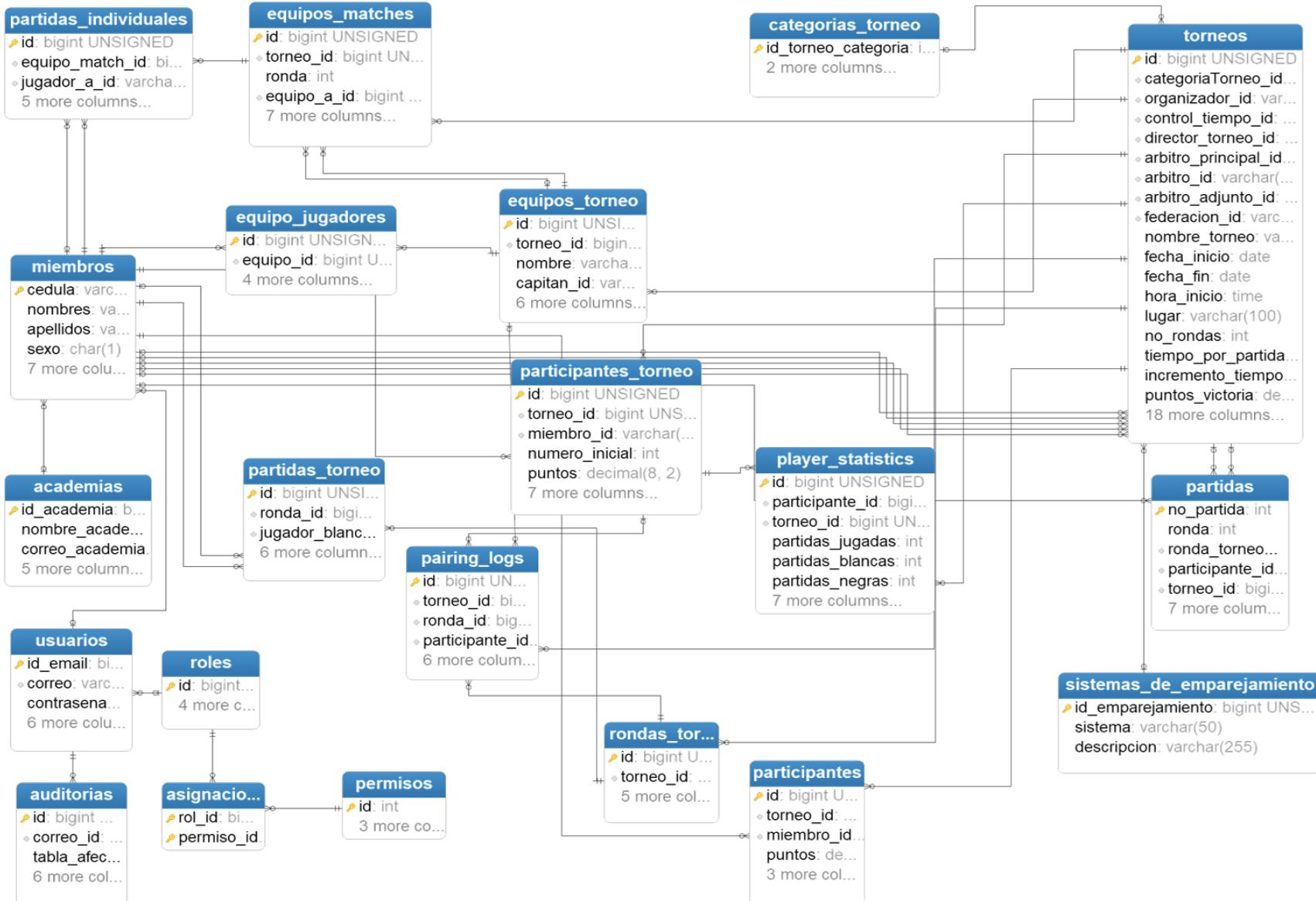
Nota: La ilustración muestra las implementaciones de los repositorios.

Figura 36: Diagrama de Arquitectura de Software



Nota: La ilustración muestra una arquitectura basada en principios de separación de responsabilidades, distribuyendo el sistema en capas bien definidas: Infrastructure, Application Layer, Domain Model Layer y Presentation Layer. Esta organización promueve un bajo acoplamiento, alta cohesión y facilita la mantenibilidad, escalabilidad y prueba del software

Figura 37: Diagrama de Base de Datos



Nota. La ilustración muestra el esquema de base de datos para el Sistema Web de gestión de torneos de ajedrez.

6.3.1.4. Capa de presentación (Presentation)

La Capa de Presentación es la parte del sistema que permite a los usuarios interactuar directamente con la aplicación. En el caso del sistema de gestión de torneos de ajedrez, esta capa puede implementarse como una interfaz gráfica para el usuario o a través de una API que proporciona servicios a otras aplicaciones. Su función principal es garantizar que la comunicación entre el usuario y la lógica del negocio sea fluida y accesible, asegurando que las funcionalidades del sistema estén siempre al alcance de los usuarios de manera fácil e intuitiva.

La Capa de Presentación asume varias responsabilidades claves:

- **Interfaz de Usuario:** Se encarga de ofrecer una interfaz gráfica amigable e intuitiva que permita a los usuarios interactuar con el sistema de manera sencilla. En el caso de un frontend, esta interfaz debe facilitar tareas como la creación y gestión de torneos, la asignación de jugadores, y la visualización de resultados. Para la API, proporciona endpoints RESTful que permiten a otras aplicaciones acceder a los recursos del sistema.
- **Validación de Entradas:** Antes de enviar cualquier dato a la Capa de Aplicación, esta capa valida la información introducida por los usuarios, asegurándose de que sea correcta y esté en el formato adecuado.
- **Presentación de Resultados:** Esta capa también es responsable de mostrar los resultados de las acciones realizadas por el usuario, como la confirmación de la creación de un torneo o la actualización del estado de un jugador en un torneo. Esta retroalimentación es crucial para una experiencia de usuario fluida y satisfactoria.
- **Gestión de Navegación:** La Capa de Presentación gestiona la navegación entre las diferentes vistas y componentes de la aplicación, asegurando que los usuarios puedan desplazarse entre las secciones del sistema sin dificultades.

Figura 38: Wireframe de visualización de registros

Titulo 1	Titulo 2	Titulo 3	Actions
Registro 1	Registro 1	Registro 1	Editar Eliminar
Registro 2	Registro 2	Registro 2	Editar Eliminar
Registro 3	Registro 3	Registro 3	Editar Eliminar
Registro 4	Registro 4	Registro 4	Editar Eliminar
Registro 5	Registro 5	Registro 5	Editar Eliminar
Registro 6	Registro 6	Registro 6	Editar Eliminar
Registro 7	Registro 7	Registro 7	Editar Eliminar
Registro 8	Registro 8	Registro 8	Editar Eliminar
Registro 9	Registro 9	Registro 9	Editar Eliminar
Registro 10	Registro 10	Registro 10	Editar Eliminar
Registro 11	Registro 11	Registro 11	Editar Eliminar
Registro 12	Registro 12	Registro 12	Editar Eliminar
Registro 13	Registro 13	Registro 13	Editar Eliminar
Registro 14	Registro 14	Registro 14	Editar Eliminar
Registro 15	Registro 15	Registro 15	Editar Eliminar
Registro 16	Registro 16	Registro 16	Editar Eliminar

Nota: La ilustración muestra el wireframe de una interfaz con una barra de navegación superior (logo, botones, navbar) y una sección principal con una tabla de visualización de registros, acompañada con opciones para agregar, realizar búsquedas, editar y eliminar, en la parte superior de la tabla.

Interacción con la Capa de Aplicación

La Capa de Presentación se encarga de interactuar directamente con el usuario, permitiéndole realizar acciones dentro del sistema de gestión de torneos. Esta capa se comunica con la Capa de Aplicación para ejecutar las funcionalidades del sistema.

Esta comunicación ocurre principalmente de dos formas:

- **Envío de solicitudes:** Cuando un usuario realiza una acción, como registrar un nuevo torneo o consultar una lista de participantes, la interfaz envía esa información a la capa de aplicación, que se encarga de procesarla adecuadamente.
- **Recepción de respuestas:** Una vez que la solicitud ha sido atendida, la capa de aplicación responde con los datos necesarios (por ejemplo, la confirmación del registro o una lista de resultados), los cuales son devueltos a la capa de presentación para ser mostrados de forma clara y comprensible al usuario.

6.4. Fase de desarrollo

La fase de desarrollo es una de las etapas más cruciales dentro del modelo en cascada, ya que permitió transformar los requerimientos previamente analizados en un sistema funcional. Para ello, se aplicaron metodologías modernas y herramientas que garantizarán la calidad, escalabilidad y mantenibilidad de la aplicación.

6.4.1. Preparación del entorno de desarrollo

Se llevó a cabo la configuración del entorno tecnológico necesario para la implementación:

- Framework: Laravel 12.0 como base para el desarrollo del sistema.
- Lenguaje: PHP 8.2+, por su amplia compatibilidad y soporte en aplicaciones web modernas.
- Base de datos: MySQL, debido a su rendimiento, seguridad y escalabilidad.

Asimismo, se definió una arquitectura MVC (Modelo-Vista-Controlador) que permitió separar las responsabilidades de la lógica de negocio, la gestión de datos y la presentación, asegurando un desarrollo modular y organizado.

6.4.2. Implementación de los Módulos

La implementación del sistema se abordó mediante una arquitectura modular, permitiendo así una mayor escalabilidad, mantenibilidad y facilidad de pruebas. Cada módulo fue desarrollado de forma independiente, lo que facilitó su documentación, validación y posterior integración al sistema general.

A continuación, se detallan los módulos más relevantes implementados en el sistema, especificando sus funcionalidades principales, interfaces y consideraciones técnicas.

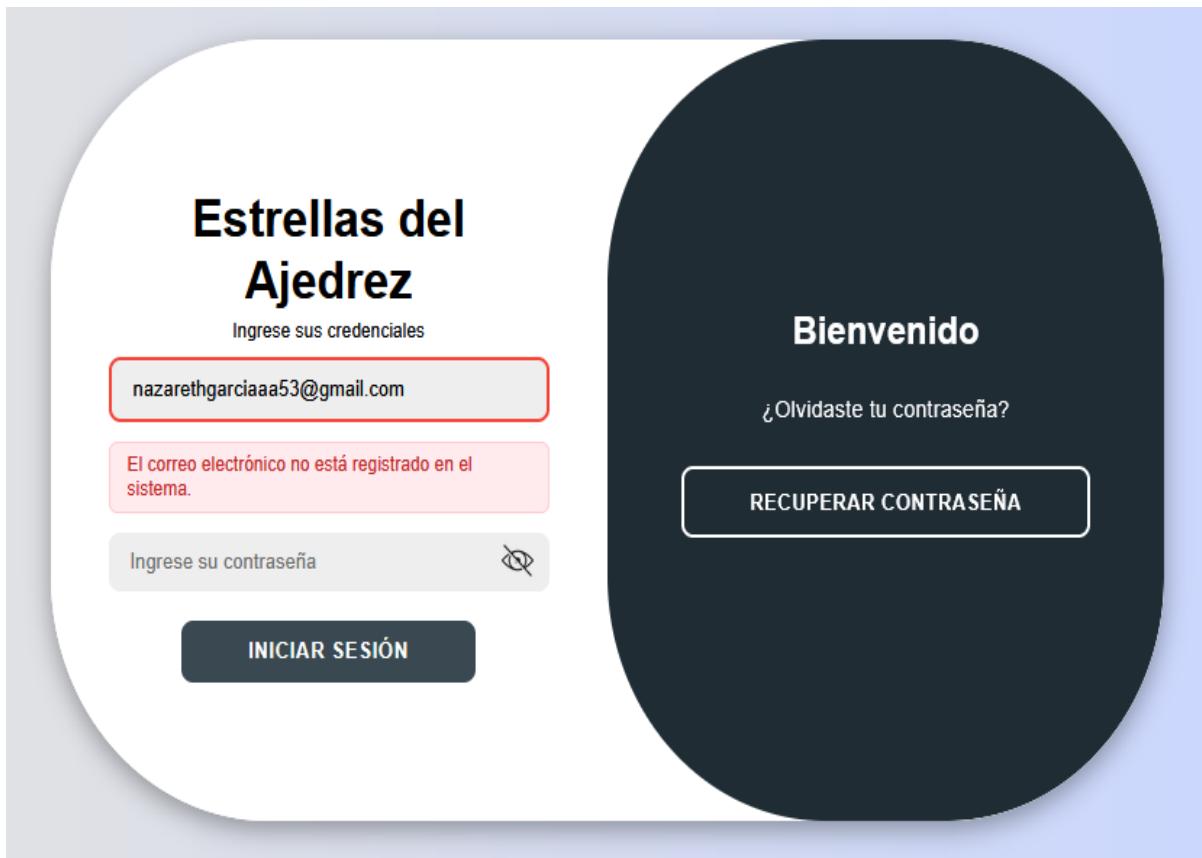
Módulo de autenticación

El módulo de autenticación tiene como propósito garantizar el acceso seguro al sistema, permitiendo únicamente a usuarios registrados iniciar sesión con sus credenciales. El proceso se desarrolla mediante un formulario de login sencillo y funcional, donde el usuario debe ingresar su correo electrónico y contraseña.

El sistema valida las credenciales verificando la existencia del usuario, su estado activo y la correcta combinación de correo y contraseña mediante hash seguro. Una vez autenticado, el usuario es redirigido automáticamente al dashboard principal.

En caso de errores, como credenciales incorrectas, campos vacíos o usuarios inactivos, el sistema despliega mensajes de error específicos, mejorando la experiencia del usuario y facilitando la corrección de los datos ingresados.

Figura 39: Interfaz de inicio de sesión con credenciales incorrectas



Nota: La ilustración muestra la interfaz de inicio de sesión, compuesta por un formulario de inicio de sesión ubicado en el centro de la pantalla. Tras intentar iniciar sesión con un correo electrónico no registrado, el sistema despliega un mensaje de error en la parte inferior del input correspondiente, indicando que el correo proporcionado no está registrado en el sistema.

Además, se incorporó una funcionalidad de recuperación de contraseña, la cual permite a los usuarios solicitar un restablecimiento mediante su correo electrónico, recibiendo instrucciones para generar una nueva contraseña de forma segura.

Figura 40: Interfaz de recuperación de contraseña



Nota: La ilustración muestra la interfaz de recuperación de contraseña posterior al envío de la solicitud de recuperación.

Figura 41: Correo para el restablecimiento de la contraseña



¡Hola!

Has solicitado restablecer tu contraseña en Estrellas del Ajedrez. Para continuar con el proceso, haz clic en el siguiente botón:

Restablecer Contraseña

⚠ Aviso de Seguridad:

Este enlace expirará en 1 hora por motivos de seguridad. Si no solicitaste restablecer tu contraseña, puedes ignorar este correo.

Si el botón no funciona, puedes copiar y pegar el siguiente enlace en tu navegador:

<https://estrellasajedrez.com/password/reset/W3HVBTxxlb6p0rJjAcGGFxxf8zzkxwU3uuiUSNxUbSmrWPR74mwDancIW3cy>

...

Nota: La ilustración muestra el correo recibido para restablecer la contraseña.

La autenticación se implementó utilizando sesiones web seguras con regeneración de tokens, almacenadas en base de datos para mayor control y persistencia. El sistema incluye protección contra ataques comunes como CSRF mediante tokens de verificación, y previene ataques de fijación de sesión mediante la regeneración automática de identificadores de sesión, reforzando la seguridad general del sistema.

Gestión de usuarios

Este módulo permite administrar a los usuarios registrados en el sistema, ofreciendo funcionalidades para crear, editar, eliminar y listar usuarios. Su diseño se basó en una interfaz dinámica que facilita la interacción del administrador con los datos de los usuarios.

Al acceder al módulo, se presenta una tabla con los usuarios existentes, la cual permite realizar búsquedas, ordenamientos y acciones directas (como editar o eliminar).

Figura 42: Interfaz del módulo de gestión de usuarios

CORREO	ROL	ESTADO	ACCIONES
engellargaespaldavargas@gmail.com	Administrador	Activo	
estudiante@chessistem.com	Estudiante	Activo	
evaluador@chessistem.com	Evaluador	Activo	
gestor@chessistem.com	Gestor	Inactivo	

Nota: La ilustración muestra la interfaz del módulo de gestión de usuarios.

Para la creación de nuevos usuarios, se despliega un formulario modal intuitivo donde se ingresan los datos esenciales: correo electrónico, contraseña (con generación automática si no se especifica), rol asignado y estado de activación. El sistema incluye validaciones robustas que garantizan la integridad de los datos y previenen duplicaciones de correos electrónicos.

El módulo implementa un sistema de permisos granular basado en roles (Administrador, Evaluador, Estudiante, Gestor), donde cada rol tiene permisos específicos para diferentes acciones del sistema. Los administradores pueden gestionar la asignación de permisos mediante

una interfaz visual que permite asignar o revocar permisos específicos a cada rol, manteniendo un registro detallado de auditoría de todas las modificaciones.

Figura 43: Formulario del módulo de gestión de usuarios

Nuevo usuario

Correo electrónico *

Rol *

Seleccione un rol

Contraseña *

Confirmar contraseña *

Estado Activo

Crear usuario

Cancelar

Nota: La ilustración muestra el formulario para la gestión de usuarios.

Figura 44: Formulario del módulo de gestión de permisos

Gestionar permisos

Rol: Administrador

academias

Seleccionar todos

Crear una nueva academia

Editar academia

Eliminar academia

Ver listado de academias

Ver detalles de academia

Cancelar Guardar Cambios

Nota: La ilustración muestra el formulario para la gestión de permisos en donde el usuario puede controlar los permisos que tendrá cada rol por cada módulo existente en el sistema.

Internamente, todas las operaciones se comunican con la API del sistema a través de peticiones HTTP seguras con validación de permisos en cada endpoint, garantizando que solo usuarios autorizados puedan realizar modificaciones. El sistema incluye logging completo de acciones para auditoría y seguimiento de cambios, reforzando la seguridad y trazabilidad de la gestión de usuarios en el sistema de torneos de ajedrez.

Módulo de gestión de torneos

El módulo de gestión de torneos es el núcleo del sistema, proporcionando una plataforma integral para la organización y administración completa de competencias de ajedrez. Al acceder al módulo, se presenta una tabla interactiva con todos los torneos registrados, permitiendo

búsquedas avanzadas, filtros por estado (Activo, Finalizado, Cancelado) y acciones directas como editar, duplicar o gestionar participantes.

Figura 45: Interfaz del módulo de gestión de torneos

The screenshot shows a tournament management interface. At the top, there is a header with the title 'Torneos' and a blue button '+ Nuevo Torneo'. Below the header are two buttons: 'Buscar' (with a magnifying glass icon) and 'Exportar' (with a download icon). The main content area displays a tournament card for 'Copa Nacional de Ajedrez - Jul 2025', which started on July 29, 2025, at 09:00 AM. The card is marked as 'Finalizado' (Completed). The card is divided into two sections: 'Detalles Generales' (General Details) and 'Organización' (Organization). The 'Detalles Generales' section contains the following data:

Categoría:	Sin categoría
Lugar:	Medellín, Colombia
Total de rondas:	5
Participantes inscritos:	10
Rondas disputadas:	5
Rondas a disputar:	0

The 'Organización' section contains the following data:

Organizador:	Nazareth García
Director:	Nazareth García
Árbitro:	Luis García
Árbitro Principal:	Luis García
Árbitro Adjunto:	Luis García
Federación:	Sin federación
Formato:	Sin formato
Sistema:	Sin sistema

At the bottom of the card are three buttons: 'Ver Detalles' (View Details), 'Editar' (Edit), and 'Eliminar' (Delete).

Nota: La ilustración muestra la interfaz del módulo de gestión de torneos.

Figura 46: Formulario para la gestión de torneos

💡 Información del Torneo

Nombre del Torneo *

Fecha de Inicio *

🕒 Hora de Inicio *

📍 Lugar *

⚖️ Equipo Organizador

Organizador *

Director del Torneo *

Árbitro *

Árbitro Adjunto *

Árbitro Principal *

📋 Detalles Técnicos

⌚ Número de Rondas * (Mínimo 3, máximo 9)

🏆 Categoría *

🕒 Sistema de Emparejamiento *

⌚ Control de Tiempo *

👤 ¿Torneo por equipos?

🏅 Criterios de Desempate ⓘ

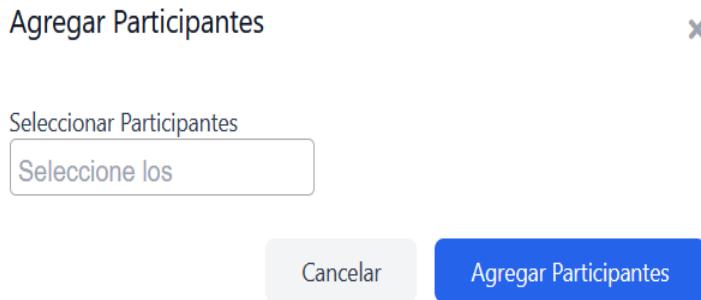
Buchholz

Sonneborn-Berger

Progresivo

Nota: La ilustración muestra el formulario para la gestión de torneos.

Figura 47: Interfaz del módulo para la gestión de participantes en un torneo



Nota: La ilustración muestra el módulo para la gestión de participantes en un torneo.

Módulo de gestión de partidas

El módulo de gestión de partidas permite el registro y seguimiento completo de todas las partidas de ajedrez dentro de los torneos. Incluye la creación automática de partidas según el sistema de emparejamiento seleccionado (Suizo, Round Robin, Eliminación Directa), registro de resultados en tiempo real, y actualización automática de puntos y posiciones de los participantes.

El sistema facilita la administración de rondas con asignación de mesas, control de colores (blancas/negras), registro de tiempos de juego y manejo de abandonos. Incluye funcionalidades avanzadas como generación automática de emparejamientos para Round Robin, cálculo de coeficientes de desempate (Buchholz, Sonneborn-Berger) y actualización en tiempo real de las tablas de posiciones cuando se registran resultados de partidas.

Figura 48: Interfaz del módulo para la gestión de partidas

The screenshot shows a web-based chess tournament management system. At the top, there are filters: 'Fecha: 15/09/2025', 'Lugar: UNI', 'Tipo: Individual', and 'Estado: Activo'. On the right, a link '← Volver al torneo' (Back to tournament) is visible. Below these, a blue button 'Rd.1' indicates the current round. The main area is titled 'Partidas de la Ronda 1' (Matches of Round 1). It displays a table of results for three games:

Tablero	Jugador A	Elo	Pts.	Color	Resultado	Color	Jugador B	Elo	Pts.
1	Carlos González Pérez	-	0.0	<input type="checkbox"/>	1-0/0-1	<input checked="" type="checkbox"/>	Ana Hernández García	-	0.0
2	Eiffel Sevilla Castillo	-	0.0	<input type="checkbox"/>	1-0/0-1	<input checked="" type="checkbox"/>	Juan Sánchez Rodríguez	-	0.0
3	Roberto Martínez	-	0.0	<input type="checkbox"/>	1-0/0-1	<input checked="" type="checkbox"/>	Sofía Torres López	-	0.0

At the bottom right of the table area is a blue button 'Guardar Todos los Resultados' (Save all results).

Nota: La ilustración muestra el módulo para la gestión de partidas y resultados.

Módulo de gestión de miembros

El módulo de gestión de miembros tiene como propósito centralizar y administrar toda la información de los jugadores de ajedrez registrados en el sistema. Este módulo permite mantener un registro completo de cada miembro, incluyendo sus datos personales, información de contacto, afiliación académica y datos deportivos relevantes.

El proceso de gestión se desarrolla mediante una interfaz intuitiva que presenta una tabla completa con todos los miembros registrados, la cual permite realizar búsquedas avanzadas, filtros por diversos criterios y acciones directas como crear, editar, eliminar y ver detalles de cada miembro. Para la creación de nuevos miembros, se despliega un formulario estructurado que captura información esencial como cédula de identidad, nombres completos, fecha de nacimiento, género, información de contacto, afiliación a academia y nivel de ELO cuando aplique.

Figura 49: Interfaz del módulo de gestión de miembros

The screenshot shows a web-based application for managing members. At the top, there is a header with the title "Miembros" and a blue button "Nuevo Miembro" with a plus sign. Below the header are two buttons: "Buscar" (Search) with a magnifying glass icon and "Exportar" (Export) with a download icon. The main content area displays a member profile for "Ana María López". The profile includes the member's name, a small profile picture, their ID number "0987654321", their club "Estrellas del Ajedrez", and a green status indicator "Activo" (Active). The profile is divided into two sections: "Datos personales" (Personal data) and "Detalles secundarios" (Secondary details). The "Datos personales" section contains the following information: **Sexo:** Masculino, **Fecha de nacimiento:** 03 de agosto del 2002, and **Teléfono:** +505 1234 5679. The "Detalles secundarios" section contains the following information: **Fecha de inscripción:** 14 de septiembre del 2025, **ELO:** -, **Correo de acceso:** -, **Rol:** -, **Torneos jugados:** 4, and **Torneos activos a jugar:** 1. At the bottom of the profile section, there are three buttons: "Ver detalles" (View details) with a magnifying glass icon, "Editar" (Edit) with a pencil icon, and "Eliminar" (Delete) with a trash can icon.

Nota: La ilustración muestra la interfaz del módulo de gestión de miembros.

Figura 50: Formulario del módulo de gestión de miembros

Nuevo Miembro

Datos Personales

Cédula *

Nombres * **Apellidos ***

Sexo *

Fecha de nacimiento *
 dd / mm / aaaa 

Teléfono *

Estado  **Activo**

Datos Académicos

Academia * **Fecha de inscripción ***
 28 / 09 / 2025 

ELO
 Ej: 1500

Correo del sistema
 Sin correo asignado

Nota: La ilustración muestra el formulario para la gestión de miembros.

La gestión de miembros se integra estrechamente con otros módulos del sistema, permitiendo la vinculación con usuarios del sistema, academias, ciudades y torneos, facilitando

así la administración integral de la información de los jugadores de ajedrez y su participación en las actividades del sistema.

Módulo de academias

Este módulo se encarga de la administración de las instituciones de ajedrez afiliadas a la academia, definiendo la estructura organizacional que vincula a los miembros con sus respectivas entidades. Su interfaz, diseñada para ser intuitiva y funcional, permite registrar y gestionar academias, capturando información institucional completa, incluyendo datos de contacto, representante y ubicación geográfica.

Además, incorpora herramientas de búsqueda y filtrado que facilitan la localización de academias según criterios como ubicación, estado operativo o nombre. También se incluyen controles que permiten activar o desactivar academias en función de su situación administrativa actual.

Este módulo está estrechamente integrado con el módulo de gestión de miembros, permitiendo establecer relaciones jerárquicas entre jugadores y las instituciones a las que pertenecen. Esta integración mejora la organización interna del sistema y facilita la administración de los usuarios en función de su afiliación institucional.

Figura 51: Interfaz del módulo de gestión de academias

The screenshot shows a web-based application for managing chess academies. At the top, there is a header with the title 'Academias' and a blue button with a '+' icon labeled 'Nueva Academia' (New Academy). Below the header are two buttons: 'Buscar' (Search) with a magnifying glass icon and 'Exportar' (Export) with a download icon. The main content area features a dark header bar with the text 'Academia de Ajedrez Bluefields' and a green 'Activo' (Active) status indicator. The interface is divided into two main sections: 'Contacto' (Contact) on the left and 'Detalles adicionales' (Additional details) on the right. The 'Contacto' section contains the following information: **Correo:** academia@bluefieldsajedrez.com, **Teléfono:** 2572-7070, **Representante:** Linda Campbell, **Dirección:** Barrio Central, Bluefields, and **Ciudad:** Bluefields, Costa Caribe Sur (Nicaragua). The 'Detalles adicionales' section shows two statistics: **Participantes registrados:** 0 and **Torneos participados:** 0. At the bottom right of the main content area are three buttons: 'Ver Detalles' (View Details) with a magnifying glass icon, 'Editar' (Edit) with a pencil icon, and 'Eliminar' (Delete) with a trash can icon.

Nota: La ilustración muestra la interfaz del módulo de gestión de academias.

Figura 52: Formulario del módulo de gestión de academias

Nueva Academia

Generalidades

Nombre

Representante

Ciudad

Dirección

Estado Activo

Contacto

Correo

Teléfono

✓ Guardar Academia

Nota: La ilustración muestra el formulario para la gestión de academias.

Emparejamientos

Esta función genera automáticamente los enfrentamientos entre jugadores durante los torneos. Soporta distintos sistemas de emparejamiento, como Suizo, Round Robin y Eliminación Directa.

Tiene en cuenta factores como puntuación, alternancia de colores y evita repeticiones. En torneos por equipos, crea partidas individuales por tablero y asigna colores de forma equilibrada.

Figura 53: Interfaz del módulo de participantes con la función de Generar Emparejamiento

Participantes (8 registrados)
 Agregar Participantes
 Generar Emparejamiento

Buscar
 Exportar

No.	Nombre	Elo	FED	1.Rd	2.Rd	3.Rd	4.Rd	5.Rd	Pts.	Des 1	Des 2	Des 3	Acciones
	Ana María López	-	NCA	0.0	0.00	0.00	0.00	0.00		Retirar			
	Carlos González Pérez	-	NCA	0.0	0.00	0.00	0.00	0.00		Retirar			
	Carlos Magno García	-	NCA	0.0	0.00	0.00	0.00	0.00		Retirar			
	Eiffel Sevilla Castillo	-	NCA	0.0	0.00	0.00	0.00	0.00		Retirar			
	Ian Eiffel Castillo	-	NCA	0.0	0.00	0.00	0.00	0.00		Retirar			
	José Martínez Sánchez	-	NCA	0.0	0.00	0.00	0.00	0.00		Retirar			
	Roberto Martínez	-	NCA	0.0	0.00	0.00	0.00	0.00		Retirar			
	Sofia Rodríguez	-	NCA	0.0	0.00	0.00	0.00	0.00		Retirar			

Nota: La ilustración muestra un módulo con el listado de participantes en un torneo con la función de *Generar Emparejamiento* habilitada. Esta función estará disponible siempre y cuando el torneo esté activo y no se haya realizado ya un empateamiento.

Auditorías y registro de actividad

Este módulo se encarga de registrar de forma automática todas las acciones realizadas dentro del sistema, permitiendo mantener un historial detallado de operaciones relevantes. Cada registro incluye información clave como el usuario responsable, la acción ejecutada, la entidad afectada, la fecha, hora y dirección IP.

La interfaz permite consultar, filtrar y exportar los registros, facilitando el seguimiento de actividades y el control administrativo. Este componente garantiza la trazabilidad de los cambios

en el sistema, siendo fundamental para auditorías internas, externas y el cumplimiento de políticas de seguridad.

Figura 54: Interfaz del módulo de auditorías

The screenshot shows the 'Auditoría del Sistema' (System Audit) interface. At the top, there are 'Buscar' (Search) and 'Exportar' (Export) buttons. Below that, a header bar displays the date and time as 'Edición - 27 de septiembre del 2025, 11:41 PM' and the session information as 'Equipo: 127.0.0.1 | Usuario: nazarethgarcia53@gmail.com | Módulo afectado: Miembros'. The main content area is divided into two sections: 'Valor previo' (Previous Value) and 'Valor posterior' (Post Value). Both sections display a JSON object representing a member's profile. The 'Valor previo' object is as follows:

```
{  
  "cedula": "0987654321",  
  "nombres": "Ana María",  
  "apellidos": "López",  
  "sexo": "M",  
  "fecha_nacimiento": "03-08-2002",  
  "fecha_inscripcion": "14-09-2025",  
  "estado": "Activo",  
  "academia": "Estrellas del Ajedrez",  
  "elo": "",  
  "correo": "",  
  "telefono": "+505 1234 5678"  
}
```

The 'Valor posterior' object is as follows:

```
{  
  "cedula": "0987654321",  
  "nombres": "Ana María",  
  "apellidos": "López",  
  "sexo": "M",  
  "fecha_nacimiento": "03-08-2002",  
  "fecha_inscripcion": "14-09-2025",  
  "estado": "Activo",  
  "academia": "Estrellas del Ajedrez",  
  "elo": "",  
  "correo": "",  
  "telefono": "+505 1234 5679"  
}
```

Nota: La ilustración muestra el módulo de auditorías correspondientes al registro de actividad

Exportación de registros

El sistema implementa la exportación de datos mediante dos modalidades principales. Desde el backend se generan archivos CSV completos, con validación de permisos y registro automático en el sistema de auditorías; estos archivos incluyen un timestamp y utilizan codificación UTF-8 para asegurar compatibilidad. Por otro lado, la exportación desde el frontend permite descargar únicamente los datos filtrados y visibles en la interfaz, excluyendo columnas no relevantes y generando archivos con formato compatible con Excel. Ambas modalidades

garantizan control de acceso, trazabilidad y respetan los filtros aplicados, combinando eficiencia y exhaustividad según las necesidades del usuario.

Dashboard principal

El dashboard principal del sistema de gestión de torneos de ajedrez actúa como la página de inicio, ofreciendo un acceso centralizado a todas las funcionalidades disponibles, adaptándose dinámicamente según los permisos del usuario autenticado. Incluye secciones clave como el perfil personal, gestión de usuarios y academias, administración de torneos, y auditorías, mostrando solo las opciones correspondientes a los privilegios asignados.

Además, cuenta con un panel de estadísticas y gráficos que presenta el desempeño individual y datos administrativos del sistema, junto con un módulo para el análisis de partidas recientes, facilitando la revisión y creación de nuevos análisis. La interfaz utiliza un sistema de permisos dinámico que ajusta la visibilidad de elementos según el rol del usuario, y está diseñada con un enfoque responsive que garantiza una experiencia óptima en dispositivos móviles y de escritorio. Visualmente, el dashboard utiliza iconografía específica, colores consistentes y efectos de interacción para mejorar la usabilidad.

Figura 55: Dashboard principal del sistema

Dashboard

Bienvenido al sistema de gestión de torneos de ajedrez

Mi Perfil
Edita tu perfil

Usuarios
Gestión de usuarios del sistema

Academias
Ver y editar academias

Torneos
Administración de torneos

Acciones Rápidas

[Nuevo Torneo](#)
 [Gestionar Academias](#)
 [Gestionar Miembros](#)
 [Auditoría](#)

Desempeño

Historial de partidas

Periodo: septiembre de 2024 - septiembre de 2025

Mes	Victorias	Derrotas	Empates	Torneos Inscritos	Partidas Jugadas
Sep	5	0	0	0	2
Oct	15	2	8	25	5
Nov	17	5	3	26	4
Dic	10	4	7	27	4
Ene	11	6	5	25	4
Feb	8	7	5	21	4
Mar	7	8	11	24	5
Abt	2	9	1	13	1
May	3	7	1	12	1
Jun	0	0	0	2	0
Jul	7	4	1	17	4
Ago	4	1	0	2	0
Sep	0	0	0	0	0

Estadísticas de Torneos

Estadísticas Completas de Torneos

Periodo: septiembre de 2024 - septiembre de 2025

Mes	Total Torneos	Torneos Pendientes	Torneos Completados	Total Participantes	Torneos en Curso
Sep	1,5	0	0	25	0
Oct	4,5	0	0	65	0
Nov	3,5	0	0	45	0
Dic	5,5	0	0	70	0
Ene	5,5	0	0	65	0
Feb	3,5	0	0	55	0
Mar	5,5	0	0	70	0
Abt	2,5	0	0	55	0
May	5,5	0	0	70	0
Jun	5,5	0	0	70	0
Jul	3,5	0	0	55	0
Ago	1,5	0	0	25	0
Sep	3,5	0	0	55	0

Ánalysis de Partidas

[Nuevo Análisis](#)
 [Ver más análisis](#)

Fecha	Blancas	Negras	Errores	Brillantes	Blunders	Evaluación	Acciones
2025-09-14	Nazareth	Miguel	B:10 N:15	B:4 N:4	B:2 N:1	Partida equilibrada con oportunidades para ambos b...	Ver
2025-09-14	Nazareth	Carmen	B:3 N:12	B:3 N:2	B:8 N:0	Partida equilibrada con oportunidades para ambos b...	Ver

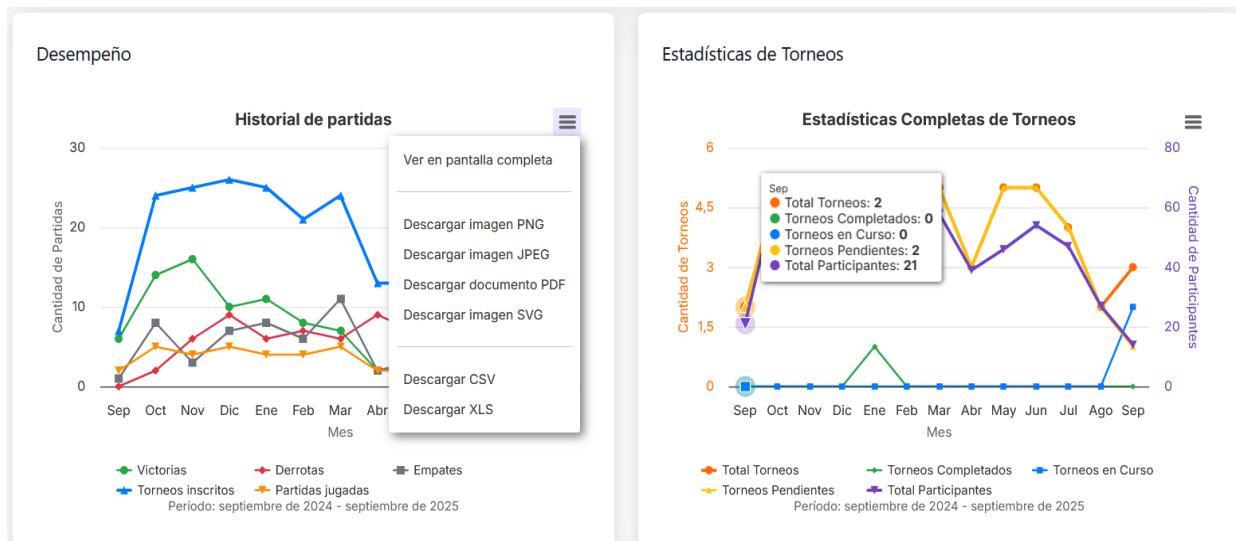
Nota: La ilustración muestra el dashboard del sistema desde una cuenta con rol de administrador.

98

Estadísticas de desempeño y actividad competitiva

El sistema incorpora visualizaciones interactivas que permiten analizar tanto el rendimiento individual del usuario como la actividad general de los torneos gestionados. El gráfico de historial de partidas muestra la evolución personal a lo largo del tiempo mediante métricas mensuales de victorias, empates y derrotas, facilitando la identificación de tendencias y patrones de desempeño. Por su parte, el gráfico de estadísticas globales presenta datos consolidados sobre la actividad competitiva del sistema, incluyendo torneos organizados, número de participantes y partidas disputadas. Estas herramientas permiten a los usuarios y administradores tomar decisiones informadas, evaluar el crecimiento del entorno competitivo y planificar futuras acciones estratégicas.

Figura 56: Gráficos de desempeño y actividad competitiva



Nota: La ilustración los gráficos de desempeño y actividad competitiva, con algunas de las funciones que pueden realizarse con los gráficos.

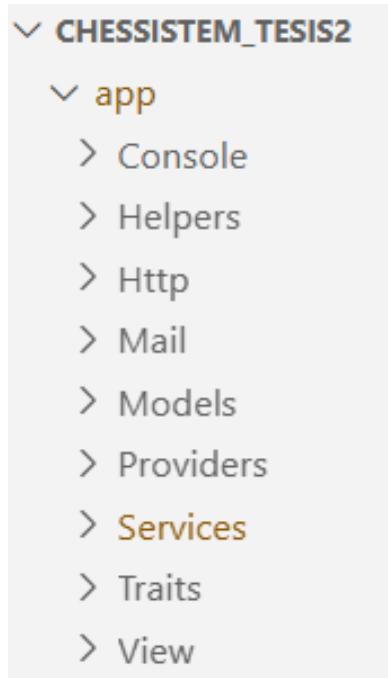
6.4.3. Arquitectura del Sistema

La solución adoptó un patrón MVC (Model-View-Controller), complementado con una arquitectura en capas bien definida, que incluye los siguientes componentes:

Patrón MVC Principal

- Modelos (Models): Modelos Eloquent que representan las entidades del dominio del ajedrez, incluyendo torneos, miembros, partidas, academias, federaciones, usuarios, roles y permisos.
- Vistas (Views): Vistas Blade organizadas en módulos funcionales, con componentes reutilizables y diseños responsivos (layouts).
- Controladores (Controllers): Controladores especializados que gestionan la lógica de presentación y coordinan la interacción entre modelos y vistas.

Figura 57: Patrón MVC del sistema



Nota: La figura muestra la estructura del patrón MVC implementada en el sistema, donde los Controladores manejan la lógica de presentación, los Modelos representan las entidades de

negocio, y los Servicios encapsulan la lógica de negocio compleja, complementados por Middleware, Traits y Helpers para funcionalidades transversales.

Servicios de Negocio

- Servicios de Emparejamiento: Implementan algoritmos de emparejamiento suizo para torneos individuales y por equipos.
- Servicios de Simulación: Permiten validar emparejamientos antes de su aplicación en el sistema.
- Servicios de Permisos: Centralizan la lógica de verificación de permisos y control de acceso.
- Servicios de Auditoría: Gestionan el registro de eventos y acciones dentro del sistema para fines de trazabilidad.
- Servicios de Comunicación: Manejan el envío de correos electrónicos automatizados o manuales.

Figura 58: Servicios principales del negocio/

```
class SwissPairingService
{
    use PairingLogs, PairingValidations;

    private Torneo $torneo;
    private Collection $participantes;
    private Collection $partidasAnteriores;
    private RondaTorneo $ronda;

    public function __construct(Torneo $torneo)
    {
        $this->torneo = $torneo;
        $this->participantes = $torneo->participantes()
            ->with('miembro')
            ->where('activo', true)
            ->get();
        $this->partidasAnteriores = collect();

        foreach ($torneo->rondas as $ronda) {
            $this->partidasAnteriores = $this->partidasAnteriores->merge($ronda->partidas);
        }
    }
}
```

Nota: La figura muestra un fragmento del SwissPairingService, servicio que implementa el algoritmo de emparejamiento suizo para torneos individuales, incluyendo validación de oponentes y asignación de colores balanceados.

Middleware para Seguridad y Control

- Autenticación: Verificación de usuarios autenticados en el sistema.
- Autorización: Control de acceso basado en roles y permisos definidos.
- Seguridad: Protección contra ataques CSRF y configuración segura de proxies.
- Validación: Middleware encargado de la validación de datos y del control de sesiones activas.

Figura 59: Middleware implementado en el sistema

```
class CheckRole
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @param string|int ...$roles
     * @return mixed
     */
    public function handle(Request $request, Closure $next, ...$roles)
    {
        if (!$request->user()) {
            return redirect('/login');
        }

        $userRoleId = $request->user()->rol_id;

        foreach ($roles as $role) {
            if ($userRoleId == $role) {
                return $next($request);
            }
        }

        return redirect('/home')->with('error', 'No tienes permiso para acceder a esta sección');
    }
}
```

Nota: La figura muestra un fragmento del CheckRole, middleware que controla el acceso basado en roles y permisos, verificando la autenticación del usuario y validando sus permisos antes de permitir el acceso a rutas protegidas.

Traits y Helpers para reutilización

- Traits: Implementan funcionalidades específicas reutilizables, como validaciones de emparejamientos o sistemas de registro (logging).
- Helpers: Proporcionan funciones auxiliares para la verificación de permisos y la ejecución de operaciones comunes.

Figura 60: Helpers implementados

```
trait PairingValidations
{
    protected function validarByeRepetido(ParticipanteTorneo $participante): bool
    {
        $byesAnteriores = $this->torneo->partidas()
            ->where(function($query) use ($participante) {
                $query->where('jugador_blanca_id', $participante->miembro_id)
                    ->whereNull('jugador_negra_id');
            })
            ->count();

        return $byesAnteriores < $this->torneo->max_byes_por_jugador ?? 1;
    }

    protected function validarColoresConsecutivos(ParticipanteTorneo $participante, string $color): bool
    {
        $ultimasPartidas = $this->torneo->partidas()
            ->where(function($query) use ($participante) {
                $query->where('jugador_blanca_id', $participante->miembro_id)
                    ->orWhere('jugador_negra_id', $participante->miembro_id);
            })
            ->orderByDesc('ronda')
            ->take(2)
            ->get();

        if ($ultimasPartidas->count() < 2) {
            return true;
        }

        $coloresConsecutivos = $ultimasPartidas->map(function($partida) use ($participante) {
            return $partida->jugador_blanca_id == $participante->miembro_id ? 'B' : 'N';
        });

        return !($coloresConsecutivos->every(fn($c) => $c == $color));
    }
}
```

Nota: La figura muestra un fragmento del `PairingValidations`, trait que implementa funcionalidades específicas reutilizables para validaciones de emparejamientos, incluyendo verificación de byes repetidos, colores consecutivos y diferencias de puntos entre jugadores.

Esta arquitectura garantiza un desarrollo escalable, seguro y con bajo acoplamiento, lo que facilita el mantenimiento y la evolución del sistema. La separación clara de responsabilidades permite que cada capa se enfoque en su función específica, mientras que los servicios encapsulan la lógica de negocio compleja inherente al dominio del ajedrez.

6.4.4. Optimizaciones y Rendimiento

Con el objetivo de garantizar un sistema eficiente, seguro y confiable, se implementaron diversas medidas orientadas a la optimización del rendimiento y a la mejora de la arquitectura general.

6.4.4.1. Optimización de Base de Datos y Consultas

Se aplicaron varias estrategias para optimizar el acceso a los datos y mejorar la eficiencia de las operaciones en la base de datos:

- **Uso de Eloquent ORM con consultas optimizadas y relaciones bien definidas:**
 - Implementación de relaciones Eloquent estructuradas que facilitan el acceso eficiente a datos relacionados.
 - Definición de scopes personalizados para cargar múltiples relaciones de manera eficiente.
 - Uso de métodos de consulta que limitan los campos seleccionados cuando no es necesario recuperar todos los datos.
- **Normalización de la base de datos para reducir redundancias y mejorar la consistencia:**
 - Diseño de una estructura de base de datos normalizada, con tablas especializadas para representar entidades del dominio.
 - Implementación de claves foráneas y restricciones de integridad referencial para asegurar la coherencia de los datos.
 - Uso de índices adecuados para optimizar el rendimiento de las consultas más frecuentes.
- **Empleo de Eager Loading para prevenir el problema de N+1 queries y optimizar los tiempos de respuesta:**
 - Aplicación sistemática de carga anticipada de relaciones en los controladores.

- Uso de *scopes* especializados para cargar múltiples relaciones de forma conjunta y eficiente.
- Implementación de Eager Loading en consultas complejas que involucran múltiples entidades relacionadas.

6.4.4.2. Optimización de Código y Arquitectura

Se adoptaron prácticas arquitectónicas y técnicas de programación para optimizar la ejecución del código y facilitar el mantenimiento del sistema:

- **Implementación de sistemas de caché para operaciones frecuentes:**
 - Configuración de múltiples *drivers* de caché adaptados a los distintos entornos de ejecución.
 - Empleo de sistemas de caché persistente para almacenar datos frecuentemente accedidos.
 - Configuración avanzada de caché para operaciones críticas que requieren alto rendimiento.
- **Control de dependencias mediante *Service Providers* y *Middleware*, favoreciendo la mantenibilidad:**
 - Implementación de *Service Providers* para la gestión centralizada y desacoplada de servicios.
 - Uso de inyección de dependencias para minimizar el acoplamiento entre componentes y facilitar las pruebas unitarias.
 - Desarrollo de *Middleware* optimizados para autenticación, autorización y validación, con integración de sistemas de *logging*.

6.4.4.3. Medidas de Seguridad Implementadas

Con el fin de garantizar la integridad, confidencialidad y disponibilidad de la información, se implementaron múltiples mecanismos de seguridad en distintos niveles del sistema.

1. Protección CSRF:

- Activación del middleware `VerifyCsrfToken` de forma global en el grupo de middleware `web`.
- Configuración del manejo de tokens CSRF en archivos `.htaccess` para permitir su uso en peticiones AJAX.
- Inclusión automática de tokens CSRF en todos los formularios HTML y validación obligatoria en solicitudes de tipo `POST`, `PUT` y `DELETE`.
- Configuración de cookies seguras con los atributos `httpOnly` y `SameSite` para prevenir ataques de secuestro de sesión (*session hijacking*).

2. Validación de Entrada:

- Uso de clases especializadas `FormRequest` para una validación robusta de los datos ingresados por los usuarios.
- Definición de reglas de validación específicas por campo, con mensajes de error personalizados y adaptados al contexto.
- Validación adicional en controladores mediante el método `validate()` para operaciones sensibles.
- Verificación de existencia de registros relacionados a través de reglas como `exists` en formularios vinculados a modelos.

3. Control de Acceso:

- Aplicación global del middleware de autenticación `auth` para proteger rutas sensibles.

- Desarrollo de un middleware personalizado `CheckRole` para implementar un control de acceso granular basado en roles de usuario.
- Implementación de verificación de permisos en controladores, con redirección automática para usuarios no autorizados.
- Uso de *helpers* específicos para comprobar permisos de acceso a módulos y funcionalidades concretas del sistema.

4. Protección contra XSS (*Cross-Site Scripting*):

- Escape automático de datos en las vistas Blade mediante la sintaxis `{}{ }{}`, que previene la ejecución de scripts maliciosos.
- Sanitización de entradas de usuario y componentes Blade para evitar la inyección de código JavaScript.
- Configuración de cookies con atributos de seguridad que impiden su acceso desde scripts del lado del cliente.

5. Auditoría y Registro de Actividades (*Logs*):

- Implementación de un sistema centralizado de auditoría a través del `AuditService`, encargado de registrar actividades críticas del sistema.
- Generación de logs estructurados para eventos de autenticación, autorización y operaciones sensibles.
- Registro detallado de cambios en modelos, incluyendo trazabilidad de valores antes y después de cada operación.
- Uso de modelos especializados para la auditoría, que capturan información contextual como la sesión activa y la dirección IP del usuario.
- Registro de decisiones de emparejamiento y operaciones relacionadas con torneos, garantizando una trazabilidad completa de eventos clave.

6.4.4.4. Protección contra Inyección SQL

Aunque no se implementaron filtros manuales, el sistema está protegido automáticamente gracias al uso de *prepared statements* de Eloquent ORM, lo que mitiga riesgos de inyección SQL.

Figura 61: Protección Automática contra Inyección SQL

```
public function participantesPorTorneo(Torneo $torneo): JsonResponse
{
    $participantes = $torneo->participantes()
        ->with('miembro')
        ->orderBy('posicion')
        ->orderBy('puntos', 'desc')
        ->get();

    return ParticipanteResource::collection($participantes)->response();
}
```

Nota: La ilustración muestra un ejemplo de cómo el método *participantesPorTorneo()* utiliza Eloquent ORM para realizar consultas seguras mediante relaciones de modelos (*\$torneo->participantes()*) y métodos de ordenamiento (*->orderBy()*).

Laravel convierte automáticamente todas las operaciones de Eloquent ORM en prepared statements internamente, garantizando que los datos del usuario nunca se mezclen directamente con las consultas SQL, proporcionando protección robusta contra inyección SQL sin necesidad de implementación manual adicional.

6.4.5. Integración y control de versiones

En esta fase se implementaron mecanismos de control de versiones y estrategias de integración que permitieron mantener la trazabilidad del código fuente, asegurar la organización de los cambios y preparar el sistema para un eventual despliegue en entornos de producción.

6.4.5.1. Control de Versiones

- **Repositorio Git/GitHub:** Se utilizó GitHub como repositorio centralizado para almacenar y gestionar el código fuente.
- **Commits semánticos:** Los mensajes de commits se realizaron siguiendo buenas prácticas, de forma descriptiva y significativa para reflejar los cambios efectuados.
- **Branching strategy:** Se trabajó en ramas separadas (feature branches) para facilitar el desarrollo modular y evitar conflictos en la rama principal.
- **Tags para versiones:** Se aplicaron etiquetas a versiones estables del sistema, lo que permite identificar puntos de control en el ciclo de desarrollo.
- **Historial de cambios:** El repositorio conserva el registro completo de las modificaciones realizadas, garantizando la trazabilidad.

6.4.5.2. Preparación para Despliegue

Aunque el sistema no fue desplegado en producción, se dejó preparada la estructura necesaria para su futura implementación:

Configuración de Entornos

- Se configuraron variables de entorno para diferenciar los ambientes de desarrollo y producción.
- Se incluyó un archivo `.env.example` como referencia para la configuración inicial.
- Se documentó la configuración de la base de datos tanto para desarrollo como para producción.

Scripts de Migración

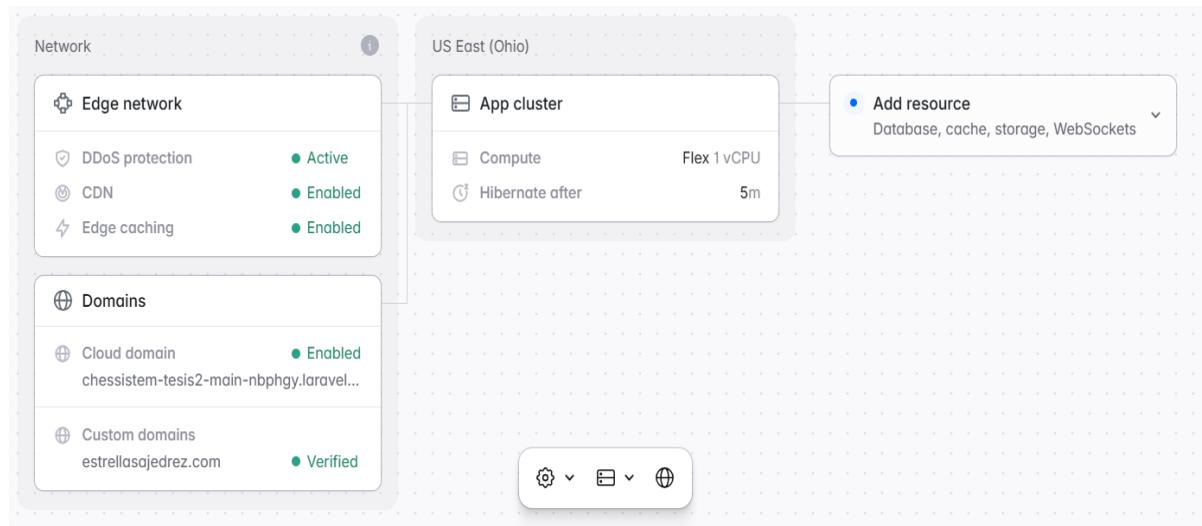
- Se desarrollaron migraciones automatizadas de la base de datos.
- Se implementaron seeders con datos de prueba para validar el correcto funcionamiento.

- Se configuraron comandos Artisan para la inicialización del sistema.

Estructura de Despliegue

- La aplicación fue preparada para su ejecución en servidores web (Apache/Nginx).
- Se documentaron los requerimientos de sistema necesarios para el despliegue.
- Se estableció la estructura base que permitiría una futura publicación del sistema en un servidor.

Figura 62: Environment del sistema desplegado desde Laravel cloud.



Nota: la Ilustración muestra la verificación del dominio y la infraestructura del sistema correctamente desplegado en Laravel Cloud.

Figura 63: Variables de entorno personalizadas.

Custom environment variables

Add your custom environment variables in the format `KEY=value`, each on a new line.

 You are overwriting 5 injected variables.

```
1 APP_KEY=base64:wyc2nBX418yzhZMu6fmgsfkYCgAnxekI3bCCy/Ch15o=
2 APP_NAME=Chessistem
3 APP_ENV=production
4 APP_DEBUG=false
5 APP_URL=https://estrellasajedrez.com
6 MAIL_MAILER=resend
7 RESEND_KEY=re_a6UWrwdm_6hNGWhotDSFLdHDFVsa5QNPe
8 MAIL_FROM_ADDRESS=no-reply@estrellasajedrez.com
9 MAIL_FROM_NAME="Estrellas del Ajedrez"
10 DB_URL=mysql://root:jtyEQcLKRQQwJaBWRRQufbpPJAzhLtqu@switchyard.proxy.rlwy.net:43401/railw
11 DB_CONNECTION=mysql
12 DB_HOST=switchyard.proxy.rlwy.net
13 DB_PORT=43401
```

Nota: la Ilustración muestra las variables de entorno personalizadas configuradas en Laravel Cloud.

6.5. Fase de Pruebas y Validación

Una vez concluido el desarrollo, se realizó la fase de pruebas con el objetivo de garantizar que el sistema cumpliera con los requerimientos establecidos y ofreciera un funcionamiento confiable. Para ello, se aplicaron **pruebas de caja negra**, orientadas a validar la funcionalidad desde la perspectiva del usuario final.

Figura 64: Base de datos con las tablas utilizadas para las pruebas

<input type="checkbox"/> academias	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	1	InnoDB	utf8mb4_unicode_ci	32.0 KB	
<input type="checkbox"/> activity_logs	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	
<input type="checkbox"/> analisis_partidas	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	9	InnoDB	utf8mb4_unicode_ci	64.0 KB	
<input type="checkbox"/> asignaciones_permisos	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	55	InnoDB	utf8mb4_unicode_ci	32.0 KB	
<input type="checkbox"/> auditorias	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	3	InnoDB	utf8mb4_unicode_ci	32.0 KB	
<input type="checkbox"/> cache	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	
<input type="checkbox"/> cache_locks	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	
<input type="checkbox"/> categorias_torneo	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	6	InnoDB	utf8mb4_unicode_ci	16.0 KB	
<input type="checkbox"/> ciudades	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	17	InnoDB	utf8mb4_unicode_ci	32.0 KB	
<input type="checkbox"/> controles_tiempo	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	5	InnoDB	utf8mb4_unicode_ci	16.0 KB	
<input type="checkbox"/> control_tiempo_torneos	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	
<input type="checkbox"/> departamentos	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	17	InnoDB	utf8mb4_unicode_ci	32.0 KB	
<input type="checkbox"/> elo_categorias	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	1	InnoDB	utf8mb4_unicode_ci	16.0 KB	
<input type="checkbox"/> equipos_matches	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	0	InnoDB	utf8mb4_unicode_ci	64.0 KB	
<input type="checkbox"/> equipos_torneo	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	
<input type="checkbox"/> equipo_jugadores	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	0	InnoDB	utf8mb4_unicode_ci	48.0 KB	
<input type="checkbox"/> failed_jobs	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	
<input type="checkbox"/> federaciones	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	1	InnoDB	utf8mb4_unicode_ci	32.0 KB	
<input type="checkbox"/> fides	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	16	InnoDB	utf8mb4_unicode_ci	48.0 KB	
<input type="checkbox"/> jobs	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	
<input type="checkbox"/> job_batches	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	
<input type="checkbox"/> miembros	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	29	InnoDB	utf8mb4_unicode_ci	48.0 KB	
<input type="checkbox"/> migrations	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	44	InnoDB	utf8mb4_unicode_ci	16.0 KB	
<input type="checkbox"/> pairing_logs	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	0	InnoDB	utf8mb4_unicode_ci	80.0 KB	
<input type="checkbox"/> paises	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	2	InnoDB	utf8mb4_unicode_ci	16.0 KB	
<input type="checkbox"/> participantes	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	4	InnoDB	utf8mb4_unicode_ci	48.0 KB	
<input type="checkbox"/> participantes_torneo	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	2	InnoDB	utf8mb4_unicode_ci	48.0 KB	
<input type="checkbox"/> partidas	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	6	InnoDB	utf8mb4_unicode_ci	80.0 KB	
<input type="checkbox"/> partidas_individuales	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	0	InnoDB	utf8mb4_unicode_ci	64.0 KB	
<input type="checkbox"/> partidas_torneo	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	0	InnoDB	utf8mb4_unicode_ci	64.0 KB	
<input type="checkbox"/> password_resets	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	
<input type="checkbox"/> permisos	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	39	InnoDB	utf8mb4_unicode_ci	16.0 KB	
<input type="checkbox"/> player_statistics	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	0	InnoDB	utf8mb4_unicode_ci	48.0 KB	
<input type="checkbox"/> puntajes_el0	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	16	InnoDB	utf8mb4_unicode_ci	32.0 KB	
<input type="checkbox"/> roles	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	4	InnoDB	utf8mb4_unicode_ci	16.0 KB	
<input type="checkbox"/> rondas_torneo	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	
<input type="checkbox"/> sessions	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	4	InnoDB	utf8mb4_unicode_ci	48.0 KB	
<input type="checkbox"/> Consola											

Nota: la ilustración muestra el gestor de base de datos con todas las tablas creadas.

6.5.1. Objetivos de la Fase de Pruebas

- Verificar el cumplimiento de los **requisitos funcionales y no funcionales**.
- Identificar posibles defectos en la interacción con los usuarios.
- Establecer métricas de calidad que permitan evaluar el sistema antes de su despliegue.

6.5.2. Estrategia de Pruebas

La estrategia definida incluyó:

- **Metodología:** pruebas de caja negra, enfocadas en entradas, procesos y salidas.
- **Tipos de pruebas implementadas:**
 - Pruebas funcionales.
 - Pruebas de rendimiento.
 - Pruebas de usabilidad.
- **Entorno de pruebas:** base de datos de ensayo, entornos controlados y casos de usuario simulados.

6.5.3. Casos de Prueba Implementados

Se elaboraron casos de prueba representativos de las funcionalidades críticas:

- **Pruebas de autenticación:** validación de inicio de sesión, registro y recuperación de contraseña. Estas funcionalidades fueron ilustradas previamente en la **Figura 39 (interfaz de inicio de sesión con credenciales incorrectas)**, la **Figura 40 (Interfaz de recuperación de contraseña)** y la **Figura 41 (Correo para el restablecimiento de la contraseña)**, donde se evidencian los diferentes escenarios considerados durante el proceso de validación.

- **Pruebas de torneos:** creación, edición, cancelación, eliminación, participantes y actualización de clasificaciones.

Figura 65: Tablas con torneos creados

The screenshot shows a user interface for tournament management. At the top, there is a header with the text "Torneos" and a button labeled "+ Nuevo Torneo". Below the header, a green notification bar displays the message "Torneo creado exitosamente." (Tournament created successfully). Underneath this, there are two buttons: "Buscar" (Search) and "Exportar" (Export). The main content area features a tournament card for "Torneo de fiestas patrias", which was created on "8 de October de 2025, 09:00 AM". The card includes a status indicator "Activo" (Active) and two sections: "Detalles Generales" (General Details) and "Organización" (Organization). The "Detalles Generales" section lists: Categoría: Sub-2000; Lugar: Auditorio 3; Total de rondas: 9; Participantes inscritos: 0; Rondas disputadas: 0; and Rondas a disputar: 9. The "Organización" section lists: Organizador: María Rodríguez; Director: Rosa Sánchez; Árbitro: José Martínez; Árbitro Principal: Luis García; Árbitro Adjunto: Rosa Sánchez; Federación: Sin federación; Formato: Clásico; and Sistema: Sistema Suizo. At the bottom of the card are four buttons: "Ver Detalles" (View Details), "Editar" (Edit), "Cancelar" (Cancel), and "Eliminar" (Delete).

Nota: la ilustración muestra la tabla con torneos creados de forma exitosa.

Figura 66: Formulario de edición de torneo

Información del Torneo

Nombre del Torneo *	Fecha de Inicio *
Torneo de fiestas patrias	08 / 10 / 2025
Hora de Inicio *	Lugar *
09:00 a.m.	Auditorio 3

Equipo Organizador

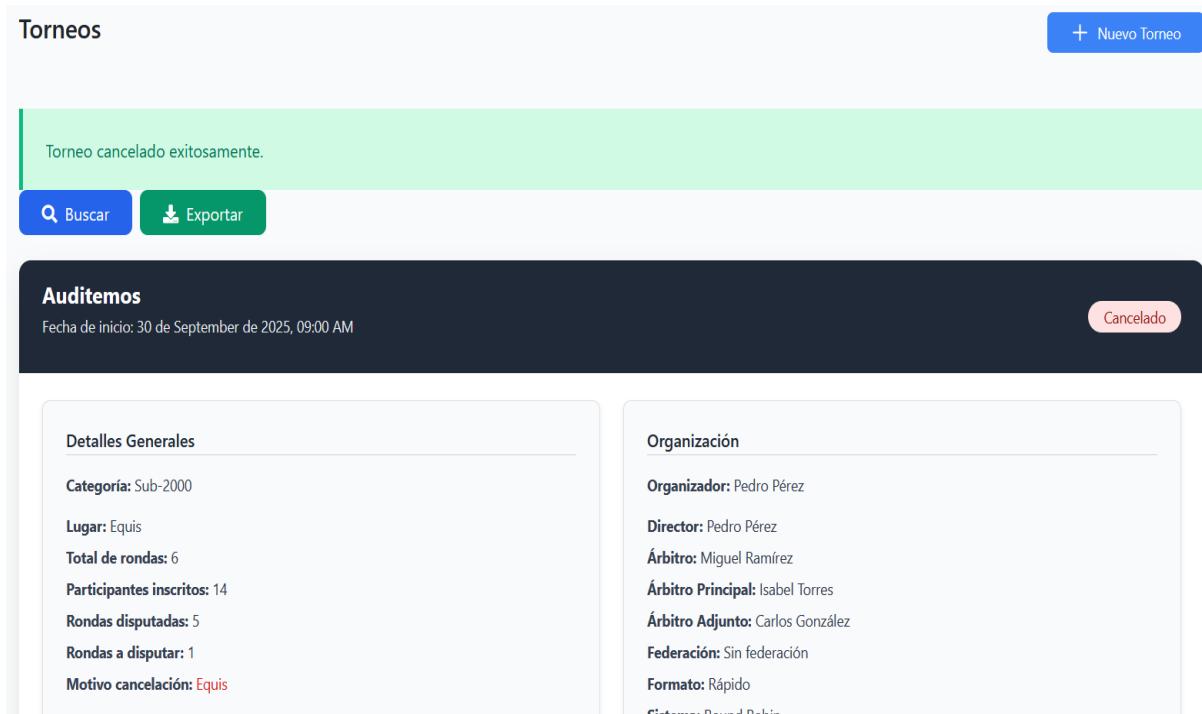
Organizador *	Director del Torneo *
María Rodríguez	Rosa Sánchez
Árbitro *	
José Martínez	
Árbitro Adjunto *	
Rosa Sánchez	
Árbitro Principal *	
Luis García	

Detalles Técnicos

Número de Rondas* (Mínimo 3, máximo 9)	Categoría*
9	Sub-2000
Sistema de Emparejamiento*	Control de Tiempo*
Sistema Suizo	Clásico (90+30)
¿Torneo por equipos?	
Individual	

Nota: la ilustración muestra el formulario de edición de un torneo previamente creado.

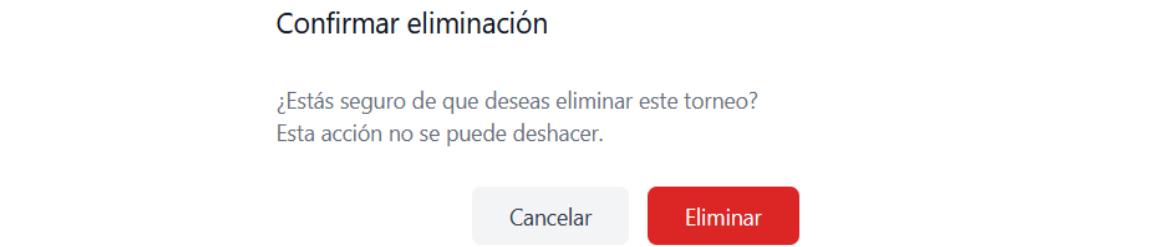
Figura 67: Torneo cancelado



The screenshot shows a software interface for managing tournaments. At the top, a green banner displays the message "Torneo cancelado exitosamente." (Tournament canceled successfully). Below this, there are two buttons: "Buscar" (Search) and "Exportar" (Export). In the top right corner, there is a blue button with a plus sign and the text "Nuevo Torneo" (New Tournament). The main content area is titled "Auditemos" and shows the following information: "Fecha de inicio: 30 de September de 2025, 09:00 AM" and a "Cancelado" (Cancelled) status indicator. The interface is divided into two columns: "Detalles Generales" (General Details) and "Organización" (Organization). The "Detalles Generales" column lists: Categoría: Sub-2000; Lugar: Equis; Total de rondas: 6; Participantes inscritos: 14; Rondas disputadas: 5; Rondas a disputar: 1; and Motivo cancelación: Equis. The "Organización" column lists: Organizador: Pedro Pérez; Director: Pedro Pérez; Árbitro: Miguel Ramírez; Árbitro Principal: Isabel Torres; Árbitro Adjunto: Carlos González; Federación: Sin federación; and Formato: Rápido.

Nota: la ilustración muestra el mensaje de éxito al momento de cancelar un torneo.

Figura 68: Modal de confirmación para eliminación de torneo



The screenshot shows a confirmation modal titled "Confirmar eliminación" (Confirm deletion). The text inside the modal reads: "¿Estás seguro de que deseas eliminar este torneo? Esta acción no se puede deshacer." (Are you sure you want to delete this tournament? This action cannot be undone.). At the bottom of the modal are two buttons: "Cancelar" (Cancel) and a larger red button labeled "Eliminar" (Delete).

Nota: la ilustración muestra un modal de confirmación al momento de eliminar un torneo.

Figura 69: Modal de incorporación de participantes



Nota: La ilustración representa el modal de selección de participantes, diseñado para facilitar la incorporación de usuarios al torneo mediante la elección directa desde la lista de miembros registrados.

Figura 70: Clasificación de participantes

Nombre	Elo	FED	1.Rd	2.Rd	3.Rd	4.Rd	5.Rd
Laisha Nazaret Acevedo García	1400	NCA	b *	0.0	0.00	0.00	0.00
Jose Onell Guerrero Salazar	1885	NCA	w *	0.0	0.00	0.00	0.00
Jonathan Muñoz	1750	NCA	b *	0.0	0.00	0.00	0.00
Patricia Alvarez Gutierrez	1849	NCA	w *	0.0	0.00	0.00	0.00

Nota: La ilustración presenta una tabla de clasificación de participantes en un torneo.

- **Pruebas de partidas:** resultados de cada ronda, emparejamientos según el tipo de torneo y el análisis de partidas.

Figura 71: Resultados obtenidos en un torneo

Nombre	Elo	FED	1.Rd	2.Rd	3.Rd	4.Rd	5.Rd	6.Rd
Laisha Nazaret Acevedo García	1400	NCA	b *	0.0	0.00	0.00	0.00	Retirar
Jose Onell Guerrero Salazar	1885	NCA	w *	0.0	0.00	0.00	0.00	Retirar
Jonathan Muñoz	1750	NCA	b *	0.0	0.00	0.00	0.00	Retirar
Patricia Alvarez Gutierrez	1849	NCA	w *	0.0	0.00	0.00	0.00	Retirar

Nota: La ilustración muestra una tabla general con la información de los resultados y puntos obtenidos en un torneo.

Figura 72: Emparejamiento de una ronda

Partidas de la Ronda 1

Tablero	Jugador A	Elo	Pts.	Color	Resultado	Color	Jugador B	Elo	Pts.
1	Anderson Steven Garcia Montoya	1793	0.0	■	1-0	■	Carlos Cano	1910	0.0
2	CM Ian Eiffel Castillo Sevilla	2095	0.0	■	1-0	■	Cristhoffer Israel Pavon Ruiz	1730	0.0
3	Felix Alberto Espinoza Guerrero	1908	0.0	■	1-0	■	Cristiana Yorlene Martinez Orozco	1755	0.0
4	Halinda Mariel Real Miranda	1479	0.0	■	1-0	■	Heidi Paola Bolaños Garcia	1595	0.0
5	Josue Miguel Rugama	1598	0.0	■	1-0	■	Jainner Sanchez	1755	0.0
6	Manuel Flores	1954	0.0	■	1-0	■	Laisha Nazaret Acevedo Garcia	1400	0.0
7	MF Maximiliano Rocha	2158	0.0	■	1-0	■	Michael Stefan Berrios Cuadra	1566	0.0
8	Heriberto Ortiz	1801	0.0	■	1-0	■	Jimmy Alberto Perez Salazar	1952	0.0
9	Patricia Alvarez Gutierrez	1849	0.0	■	1-0	■	Omar Emilio Galeano Cabrales	1907	0.0
10	Jose Onell Guerrero Salazar	1885	0.0	■	1-0	■	Jonathan Muñoz	1750	0.0

Guardar Todos los Resultados

Nota: La ilustración muestra un emparejamiento de una ronda en específico de un torneo.

Figura 73: Interfaz de análisis de una partida

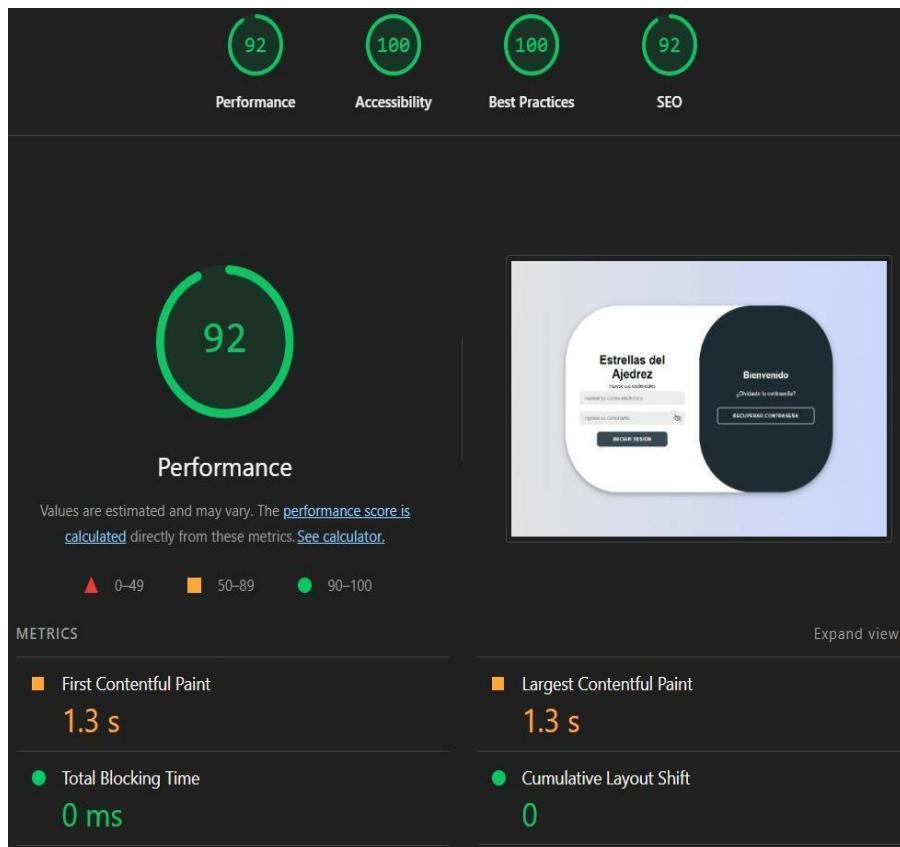
Análisis de la Partida

Jugadores		Errores y Aciertos	
Blancas:	Desconocido		
Negras:	Desconocido		
Fecha:	2025-08-13		
Evaluación General		Blancas	
Partida muy larga y compleja con múltiples fases. Excelente lucha táctica que terminó en tablas por repetición.		3	3
		2	1
		1	1

Nota: La ilustración muestra la interfaz de un análisis terminado de una partida.

- **Pruebas de rendimiento:** respuesta del sistema ante múltiples usuarios concurrentes.

Figura 74: Resultado de rendimiento



Nota: La ilustración muestra el resultado de las pruebas de rendimiento.

Figura 75: Pruebas de rendimiento terminal.

```
storage\logs\laravel.log:230146:[2025-10-12 22:51:02] local.INFO: Jugadores: 9
storage\logs\laravel.log:230147:[2025-10-12 22:51:02] local.INFO: Tiempo: 0.1183 segundos
> storage\logs\laravel.log:230254:[2025-10-12 22:55:57] local.INFO: ===== TIEMPO DE EMPAREJAMIENTO =====
storage\logs\laravel.log:230255:[2025-10-12 22:55:57] local.INFO: Jugadores: 9
storage\logs\laravel.log:230256:[2025-10-12 22:55:57] local.INFO: Tiempo: 0.0315 segundos
> storage\logs\laravel.log:231979:[2025-10-12 23:08:47] local.INFO: ===== TIEMPO DE EMPAREJAMIENTO =====
storage\logs\laravel.log:231980:[2025-10-12 23:08:47] local.INFO: Jugadores: 50
storage\logs\laravel.log:231981:[2025-10-12 23:08:47] local.INFO: Tiempo: 0.1049 segundos
> storage\logs\laravel.log:232871:[2025-10-12 23:12:09] local.INFO: ===== TIEMPO DE EMPAREJAMIENTO =====
storage\logs\laravel.log:232872:[2025-10-12 23:12:09] local.INFO: Jugadores: 100
storage\logs\laravel.log:232873:[2025-10-12 23:12:09] local.INFO: Tiempo: 0.2233 segundos
```

Nota: la ilustración muestra los resultados obtenidos en los Logs de las pruebas de rendimiento de forma local.

Figura 76: Pruebas de rendimiento en el sistema desplegado.

46	Jugador91 Test100	-	0.0	<input type="checkbox"/>	1-0	<input checked="" type="checkbox"/>	Jugador92 Test100	-	0.0
47	Jugador94 Test100	-	0.0	<input type="checkbox"/>	1-0	<input checked="" type="checkbox"/>	Jugador93 Test100	-	0.0
48	Jugador95 Test100	-	0.0	<input type="checkbox"/>	1-0	<input checked="" type="checkbox"/>	Jugador96 Test100	-	0.0
49	Jugador97 Test100	-	0.0	<input type="checkbox"/>	1-0	<input checked="" type="checkbox"/>	Jugador98 Test100	-	0.0
50	Jugador100 Test100	-	0.0	<input type="checkbox"/>	1-0	<input checked="" type="checkbox"/>	Jugador99 Test100	-	0.0

Nota: la ilustración muestra las evidencias de las pruebas de rendimiento con un total de 100 registros en el sistema una vez desplegado a producción.

6.5.4. Resultados de las Pruebas

- Cobertura superior al **85% de los requerimientos funcionales**.
- Tiempos de respuesta dentro de los márgenes establecidos (<2 segundos por consulta).

Figura 77: Tiempo de respuesta por consulta

	fa-brands-400.woff2	200	font	all.min.css
	favicon.ico	200	x-icon	Other
55 requests	11.0 kB transferred	9.6 MB resources	Finish: 1.55 s	DOMContentLoaded: 1.52 s Load: 1.53 s
Console AI assistance What's new Issues Autofill				

Nota: la ilustración muestra el tiempo exacto de respuesta por consulta del sistema.

- Registro de defectos menores que fueron corregidos antes del despliegue.
- **Pruebas de inyección SQL usando entradas maliciosas:** envío datos especialmente diseñados a una aplicación para detectar y explotar vulnerabilidades en la construcción de consultas SQL con el fin de acceder, modificar o eliminar información.

Figura 78: Cadenas maliciosas utilizadas para el test de inyección SQL

```
$payloads = [
    "user001",           // normal
    "abc'xyz",           // prueba de comilla
    "' OR 1=1 --",      // intento inyección OR
    "' AND 1=2 --",     // intento inyección AND
    "'; SELECT * FROM participantes_torneo --", // intento UNION
    "'; DROP TABLE participantes_torneo --",    // destructivo, no se ejecutará por seguridad
    "' OR SLEEP(5) --"   // intento de retraso
];
```

Nota: La ilustración muestra las cadenas utilizadas para las pruebas de inyección SQL.

Figura 79: Fragmento de código para las pruebas de inyección SQL

```
foreach ($payloads as $p) {
    echo "\nPayload: \"$p\"\n";

    // Consulta concatenada (simula punto vulnerable)
    $sql = "SELECT id, torneo_id, miembro_id FROM participantes_torneo WHERE miembro_id = '$p';

    try {
        $resultado = $pdo->query($sql);
        $rows = $resultado->fetchAll(PDO::FETCH_ASSOC);
        echo "Registros devueltos: " . count($rows) . "\n";

        if (count($rows) > 0) {
            foreach ($rows as $r) {
                echo " - ID: {$r['id']} | Torneo: {$r['torneo_id']} | Miembro: {$r['miembro_id']}\n";
            }
            echo "Estado: Consulta ejecutada con retorno.\n";
        } else {
            echo "No se devolvieron registros.\n";
            echo "Estado: Consulta ejecutada sin retorno.\n";
        }
    } catch (Exception $e) {
        echo "Error SQL: " . $e->getMessage() . "\n";
        echo "Estado: Protección activa o sintaxis inválida.\n";
    }
}
```

Nota: La ilustración muestra un fragmento del código que implementa pruebas de inyección SQL.

Figura 80: Resultados de pruebas de inyección SQL.

```
ario\Documents\GitHub\Chessistem_tesis2
php inject.php
== INICIO PRUEBAS AUTOMÁTICAS DE INYECCIÓN ==

Payload: "user001"
Registros devueltos: 0
No se devolvieron registros.
Estado: Consulta ejecutada sin retorno.

Payload: "abc'xyz"
Error SQL: SQLSTATE[42000]: Syntax error or access violation: 1064 You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'xyz'' at line 1
Estado: Protección activa o sintaxis inválida.

Payload: "' OR 1=1 --"
Error SQL: SQLSTATE[42000]: Syntax error or access violation: 1064 You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '' at line 1
Estado: Protección activa o sintaxis inválida.

Payload: "' AND 1=2 --"
Error SQL: SQLSTATE[42000]: Syntax error or access violation: 1064 You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '' at line 1
Estado: Protección activa o sintaxis inválida.

Payload: ""; SELECT * FROM participantes_torneo --
Registros devueltos: 0
No se devolvieron registros.
Estado: Consulta ejecutada sin retorno.

Payload: ""; DROP TABLE participantes_torneo --
Registros devueltos: 0
No se devolvieron registros.
```

Nota: La Ilustración muestra los resultados obtenidos de las primeras pruebas realizadas de inyección SQL realizadas en el sistema una vez desplegado.

6.5.5. Validación de Requisitos

Se verificó la conformidad con:

- **Requisitos funcionales:** todos los módulos respondieron de acuerdo con lo esperado.
- **Requisitos no funcionales:** el sistema cumplió con estándares de rendimiento, usabilidad y seguridad.

6.5.6. Certificación de Calidad

El sistema fue evaluado y certificado como **apto para su implementación**, cumpliendo los criterios de calidad definidos al inicio del proyecto.

VII. CONCLUSIÓN

El desarrollo del **Sistema Web de Gestión de Torneos de Ajedrez para la Academia “Estrellas del Ajedrez”** permitió dar respuesta a la necesidad de modernizar los procesos de inscripción, organización y seguimiento de competencias, reemplazando la gestión manual e ineficiente realizada anteriormente en hojas de cálculo.

El sistema alcanzó los objetivos planteados a través de las siguientes acciones: se integró la administración de jugadores, academias y torneos en una sola plataforma mediante el diseño de una base de datos relacional en **MySQL**; se automatizaron los emparejamientos bajo diferentes modalidades (Suizo, Round-Robin y Eliminación Directa) mediante algoritmos programados en **PHP y Laravel**; se incorporaron sistemas de desempate definidos en las reglas del ajedrez competitivo; y se desarrolló un módulo de análisis de partidas en modo simulado utilizando archivos **PGN**, lo cual constituye un aporte didáctico para el aprendizaje y evaluación de los ajedrecistas.

Las pruebas de validación confirmaron que el sistema cumple con los requisitos funcionales y no funcionales establecidos, garantizando seguridad, usabilidad, rendimiento y escalabilidad. Además, la incorporación de medidas de seguridad, control de versiones y estrategias de despliegue respalda su confiabilidad y mantenibilidad.

VIII. RECOMENDACIONES

- **Integración de Puntajes ELO:** Incorporar el cálculo y actualización de puntajes ELO dentro de la dinámica de la plataforma, lo cual permitiría enriquecer la gestión de torneos y el seguimiento del rendimiento de los jugadores.
- **Módulo de Partidas en Línea y Entrenamiento:** Incluir la posibilidad de jugar partidas en línea, así como enfrentamientos con motores o bots, con el fin de apoyar el entrenamiento de los ajedrecistas.
- **Mejora del Módulo de Análisis:** Ampliar las capacidades del análisis de partidas, pasando del simulador actual a un motor más avanzado que permita evaluar jugadas con mayor precisión.
- **Adquisición de Dominio Propio:** Registrar y configurar un dominio web oficial para la academia, lo que brindaría mayor profesionalismo, accesibilidad y confianza en la plataforma.

IX. BIBLIOGRAFÍA

- 123ajedrez.com.. (s.f.). Notación ajedrecística. <https://www.123ajedrez.com/misclanea/notacion-ajedrez.html>
- Amazon. (s.f.). ¿Qué es JavaScript? <https://aws.amazon.com/es/what-is/javascript/>
- Arsys. (s.f.). phpMyAdmin: qué es y para qué sirve. <https://www.arsys.es/blog/phpmyadmin-que-es-y-para-que-sirve>
- Colonia Machado, B. A. (2009). Análisis, diseño e implementación de un sistema de administración de torneos del juego de ajedrez [Tesis de licenciatura, Pontificia Universidad Católica del Perú]. <https://doi.org/10.500.12404/1329>
- Corporation's, Mozilla. (s.f.). HTML: Lenguaje de marcado de hipertexto. MDN Web Docs. <https://developer.mozilla.org/es/docs/Web/HTML>
- Daniel Muñoz, F. I. (s.f.). Sistemas de desempate en ajedrez. The Zugzwang Blog. <https://thezugzwangblog.com/reglas-de-desempate-en-ajedrez/>
- Ecured. (2018, febrero 14). Sistemas de eliminación en las competencias deportivas. https://www.ecured.cu/index.php?title=Sistemas_de_eliminaci%C3%B3n_en_las_competencias_deportivas&action=history
- Freelancer.com.. (2024). Freelancer. <https://www.freelancer.com/>
- Granados, M. E. (2020, agosto 14). Sistema de gestión de torneos deportivos [Tesis de grado, Universidad Nacional de Ingeniería]. <https://ribuni.uni.edu.ni/3775/>
- Grupohost. (s.f.). Certificados SSL con validación de dominio. <https://www.host.com.ni/certificados-ssl/ssl-validacion-dominio>
- Herzog, H. (2002, agosto 8). Chess-Results homepage. <https://chess-results.com/>
- Herzog, H. (2014, junio 14). Swiss-Manager manual, 2015. <https://swiss-manager.at/>

Herrera, F. J. (2018, julio 21). ¿Qué es el ELO? Chess.com..

<https://www.chess.com/es/blog/stewiiegriffin/que-es-el-elo>

Hostinger. (2024). Planes de hosting web Nicaragua. <https://www.hostinger.com/>

INATEC. (2024). Panorama del sector tecnológico y empleabilidad TIC en Nicaragua [Entrevista]. Managua, Nicaragua.

Laravel.com.. (2011). Installation. <https://laravel.com/docs/12.x>

Microsoft. (2022). Visual Studio Community. <https://visualstudio.microsoft.com/es/vs/community/>

Neothek.com.. (s.f.). Certificados SSL. <https://www.neothek.com/es-ni/certificados-ssl/>

Next_U. (s.f.). ¿Qué es CSS? <https://www.nextu.com/blog/que-es-css-rc22/>

Nicaragua. (2011). Reglamento de la Ley N.º 729, Ley de Firma Electrónica.

Nicaragua. (2012). Ley de Protección de Datos Personales.

Pressman, R. S. (1995). Ingeniería del software: Un enfoque práctico (pp. 26–30). McGraw Hill.

Service, A. W. (s.f.). ¿Qué es Python? Amazon Web Services. <https://aws.amazon.com/es/what-is/python/>

The PHP Group. (s.f.). Introducción a PHP. <https://www.php.net/manual/es/intro-whatis.php>

Ubalgo, J. (2018, octubre 11). Introducción al ajedrez. Medium.

<https://medium.com/ajedrez/introducci%C3%B3n-ca79a5ee20f>

Workana. (2024). Tarifas promedio de programadores freelance en Centroamérica.

<https://www.workana.com/>

Anónimo. (2018, junio 20). Sistemas de eliminación directa. nanopdf.com..

https://nanopdf.com/download/sistemas-de-eliminacion-1-eliminacion-directa-la_pdf#

X. ANEXOS

10.1. Diccionario de datos

Figura 81: Diccionario de datos del sistema 1

chessystem2

academias

Columna	Tipo	Nulo	Predeterminado	Comentarios
id_academia (Primaria)	bigint	No		
nombre_academia	varchar(100)	No		
correo_academia	varchar(100)	Sí	NULL	
telefono_academia	varchar(20)	Sí	NULL	
representante_academia	varchar(255)	Sí	NULL	
direccion_academia	varchar(255)	Sí	NULL	
ciudad_id	bigint	Sí	NULL	
estado_academia	tinyint(1)	No	1	

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_academia	0	A	No	
academias_ciudad_id_foreign	BTREE	No	No	ciudad_id	0	A	Sí	

activity_logs

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	bigint	No		
created_at	timestamp	Sí	NULL	
updated_at	timestamp	Sí	NULL	

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	

analisis_partidas

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	bigint	No		
partida_id	int	Sí	NULL	
movimientos	longtext	No		
jugador_blanca_id	varchar(20)	Sí	NULL	
jugador_negra_id	varchar(20)	Sí	NULL	
evaluacion_general	text	No		
errores_blanca	int	No		
errores_negra	int	No		
brillantes_blanca	int	No		
brillantes_negra	int	No		
blunders_blanca	int	No		
blunders_negra	int	No		
created_at	timestamp	Sí	NULL	
updated_at	timestamp	Sí	NULL	

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	9	A	No	
analisis_partidas_partida_id_foreign	BTREE	No	No	partida_id	3	A	Sí	
analisis_partidas_jugador_blanca_id_foreign	BTREE	No	No	jugador_blanca_id	5	A	Sí	

Figura 82: Diccionario de datos del sistema 2

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
analisis_partidas_jugador_negras_id_foreign	BTREE	No	No	jugador_negras_id	5	A	Si	

asignaciones_permisos

Columna	Tipo	Nulo	Predeterminado	Comentarios
rol_id (Primaria)	bigint	No		
permiso_id (Primaria)	int	No		

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	rol_id	4	A	No	
asignaciones_permisos_permiso_id_foreign	BTREE	No	No	permiso_id	51	A	No	
				permiso_id	35	A	No	

auditorias

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	bigint	No		
correo_id	varchar(40)	No		
tabla_afectada	varchar(30)	No		
accion	varchar(20)	No		
valor_previo	text	Sí	NULL	
valor_posterior	text	Sí	NULL	
fecha	date	No		
hora	time	No		
equipo	varchar(50)	No		

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	id	2	A	No	
auditorias_correo_id_foreign	BTREE	No	No	correo_id	1	A	No	

cache

Columna	Tipo	Nulo	Predeterminado	Comentarios
key (Primaria)	varchar(255)	No		
value	mediumtext	No		
expiration	int	No		

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	key	0	A	No	

cache_locks

Columna	Tipo	Nulo	Predeterminado	Comentarios
key (Primaria)	varchar(255)	No		
owner	varchar(255)	No		
expiration	int	No		

índices

Figura 83: Diccionario de datos del sistema 3

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	key	0	A	No	

categorias_torneo

Columna	Tipo	Nulo	Predeterminado	Comentarios
id_torneo_categoria (Primaria)	int	No		
categoria_torneo	varchar(50)	No		
descrip_categoria_torneo	varchar(255)	Si	NULL	

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_torneo_categoria	6	A	No	

ciudades

Columna	Tipo	Nulo	Predeterminado	Comentarios
id_ciudad (Primaria)	bigint	No		
depto_id	bigint	No		
nombre_ciudad	varchar(255)	No		

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_ciudad	17	A	No	
ciudades_depto_id_foreign	BTREE	No	No	depto_id	17	A	No	

controles_tiempo

Columna	Tipo	Nulo	Predeterminado	Comentarios
id_control_tiempo (Primaria)	int	No		
formato	varchar(12)	No		
control_tiempo	varchar(15)	No		
descrip_control_tiempo	varchar(100)	Si	NULL	

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_control_tiempo	5	A	No	

control_tiempo_torneos

Columna	Tipo	Nulo	Predeterminado	Comentarios
control_tiempo_id (Primaria)	int	No		
categorias_torneo_id (Primaria)	int	No		

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	control_tiempo_id	0	A	No	
				categorias_torneo_id	0	A	No	
control_tiempo_torneos_categorias_torneo_id_foreign	BTREE	No	No	categorias_torneo_id	0	A	No	

departamentos

Figura 84: Diccionario de datos del sistema 4

Columna	Tipo	Nulo	Predeterminado	Comentarios				
id_depto (Primaria)	bigint	No						
pais_id	bigint	No						
nombre_depto	varchar(255)	No						

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_depto	17	A	No	
departamentos_pais_id_foreign	BTREE	No	No	pais_id	1	A	No	

elo_catgorias

Columna	Tipo	Nulo	Predeterminado	Comentarios				
no_elo (Primaria)	bigint	No						
categoria_elo	varchar(10)	Sí	NULL					
descripcion	varchar(255)	Sí	NULL					
elo_catgorias_estado	tinyint(1)	No	1					

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	no_elo	0	A	No	

equipos_matches

Columna	Tipo	Nulo	Predeterminado	Comentarios				
id (Primaria)	bigint	No						
torneo_id	bigint	No						
ronda	int	No						
equipo_a_id	bigint	No						
equipo_b_id	bigint	Sí	NULL					
puntos_equipo_a	double	Sí	NULL					
puntos_equipo_b	double	Sí	NULL					
resultado_match	tinyint	Sí	NULL					
mesa	int	Sí	NULL					
created_at	timestamp	Sí	NULL					
updated_at	timestamp	Sí	NULL					

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	
equipos_matches_torneo_id_foreign	BTREE	No	No	torneo_id	0	A	No	
equipos_matches_equipo_a_id_foreign	BTREE	No	No	equipo_a_id	0	A	No	
equipos_matches_equipo_b_id_foreign	BTREE	No	No	equipo_b_id	0	A	Sí	

equipos_torneo

Columna	Tipo	Nulo	Predeterminado	Comentarios				
id (Primaria)	bigint	No						
torneo_id	bigint	No						
nombre	varchar(255)	No						
capitan_id	varchar(20)	Sí	NULL					
elo_medio	double	Sí	NULL					
federacion	varchar(10)	Sí	NULL					
logo	varchar(255)	Sí	NULL					

Figura 85: Diccionario de datos del sistema 5

Columna	Tipo	Nulo	Predeterminado	Comentarios			
notas	text	Sí	NULL				
created_at	timestamp	Sí	NULL				
updated_at	timestamp	Sí	NULL				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	
equipos_torneo_torneo_id_foreign	BTREE	No	No	torneo_id	0	A	No	

equipo_jugadores

Columna	Tipo	Nulo	Predeterminado	Comentarios			
id (Primaria)	bigint	No					
equipo_id	bigint	No					
miembro_id	varchar(20)	No					
tablero	int	Sí	NULL				
created_at	timestamp	Sí	NULL				
updated_at	timestamp	Sí	NULL				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	
equipo_jugadores_equipo_id_foreign	BTREE	No	No	equipo_id	0	A	No	
equipo_jugadores_miembro_id_foreign	BTREE	No	No	miembro_id	0	A	No	

failed_jobs

Columna	Tipo	Nulo	Predeterminado	Comentarios			
id (Primaria)	bigint	No					
uuid	varchar(255)	No					
connection	text	No					
queue	text	No					
payload	longtext	No					
exception	longtext	No					
failed_at	timestamp	No	CURRENT_TIMESTAMP				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	
failed_jobs_uuid_unique	BTREE	Sí	No	uuid	0	A	No	

federaciones

Columna	Tipo	Nulo	Predeterminado	Comentarios			
acronimo (Primaria)	varchar(255)	No					
nombre_federacion	varchar(255)	No					
pais_id	bigint	No					
federacion_estado	tinyint(1)	No	1				

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	acronimo	0	A	No	

Figura 86: Diccionario de datos del sistema 6

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
federaciones_pais_id_foreign	BTREE	No	No	pais_id	0	A	No	

fides

Columna	Tipo	Nulo	Predeterminado	Comentarios
fide_id (<i>Primaria</i>)	varchar(255)	No		
cedula_ajedrecista_id	varchar(255)	No		
fed_id	varchar(255)	No		
título	varchar(255)	Si	NULL	
fide_estado	tinyint(1)	No	1	

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	fide_id	16	A	No	
fides_cedula_ajedrecista_id_foreign	BTREE	No	No	cedula_ajedrecista_id	16	A	No	

jobs

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	bigint	No		
queue	varchar(255)	No		
payload	longtext	No		
attempts	tinyint	No		
reserved_at	int	Si	NULL	
available_at	int	No		
created_at	int	No		

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	id	0	A	No	
jobs_queue_index	BTREE	No	No	queue	0	A	No	

job_batches

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	varchar(255)	No		
name	varchar(255)	No		
total_jobs	int	No		
pending_jobs	int	No		
failed_jobs	int	No		
failed_job_ids	longtext	No		
options	mediumtext	Si	NULL	
cancelled_at	int	Si	NULL	
created_at	int	No		
finished_at	int	Si	NULL	

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	id	0	A	No	

miembros

Figura 87: Diccionario de datos del sistema 7

Columna	Tipo	Nulo	Predeterminado	Comentarios
cedula (Primaria)	varchar(20)	No		
nombrres	varchar(50)	Sí	NULL	
apellidos	varchar(50)	Sí	NULL	
sexo	char(1)	Sí	NULL	
fecha_nacimiento	date	Sí	NULL	
fecha_inscripcion	date	Sí	NULL	
estado_miembro	tinyint(1)	Sí	NULL	
elo	int	Sí	NULL	
correo_sistema_id	varchar(40)	Sí	NULL	
academia_id	bigint	Sí	NULL	

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	cedula	29	A	No	
miembros_correo_sistema_id_foreign	BTREE	No	No	correo_sistema_id	8	A	Sí	
miembros_academia_id_foreign	BTREE	No	No	academia_id	2	A	Sí	

migrations

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	int	No		
migration	varchar(255)	No		
batch	int	No		

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	42	A	No	

pairing_logs

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	bigint	No		
torneo_id	bigint	No		
ronda_id	bigint	No		
participante_id	bigint	Sí	NULL	
equipo_id	bigint	Sí	NULL	
tipo_decision	varchar(255)	No		
motivo	varchar(255)	No		
detalles	json	Sí	NULL	
created_at	timestamp	Sí	NULL	
updated_at	timestamp	Sí	NULL	

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	
pairing_logs_torneo_id_foreign	BTREE	No	No	torneo_id	0	A	No	
pairing_logs_ronda_id_foreign	BTREE	No	No	ronda_id	0	A	No	
pairing_logs_participante_id_foreign	BTREE	No	No	participante_id	0	A	Sí	
pairing_logs_equipo_id_foreign	BTREE	No	No	equipo_id	0	A	Sí	

paises

Figura 88: Diccionario de datos del sistema 8

Columna	Tipo	Nulo	Predeterminado	Comentarios
id_pais (Primaria)	bigint	No		
nombre_pais	varchar(255)	No		

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_pais	2	A	No	

participantes

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	bigint	No		
torneo_id	bigint	No		
miembro_id	varchar(255)	No		
puntos	decimal(8,2)	No	0.00	
posición	int	Sí	NULL	
created_at	timestamp	Sí	NULL	
updated_at	timestamp	Sí	NULL	

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	4	A	No	
participantes_torneo_id_foreign	BTREE	No	No	torneo_id	1	A	No	

participantes_torneo

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	bigint	No		
torneo_id	bigint	No		
miembro_id	varchar(20)	No		
numero_inicial	int	Sí	NULL	
puntos	decimal(8,2)	No	0.00	
posición	int	Sí	NULL	
buchholz	double	No	0	
sonneborn_berger	double	No	0	
progresivo	double	No	0	
activo	tinyint(1)	No	1	
created_at	timestamp	Sí	NULL	
updated_at	timestamp	Sí	NULL	

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	2	A	No	
participantes_torneo_torneo_id_miembro_id_unique	BTREE	Sí	No	torneo_id	1	A	No	
participantes_torneo_miembro_id_foreign	BTREE	No	No	miembro_id	2	A	No	

partidas

Columna	Tipo	Nulo	Predeterminado	Comentarios
no_partida (Primaria)	int	No		
ronda	int	Sí	NULL	
ronda_torneo_id	bigint	Sí	NULL	

Figura 89: Diccionario de datos del sistema 9

Columna	Tipo	Nulo	Predeterminado	Comentarios
participante_id	varchar(20)	Sí	NULL	
torneo_id	bigint	Sí	NULL	
mesa	int	Sí	NULL	
color	tinyint(1)	Sí	NULL	
tiempo	time	Sí	NULL	
desempate_utilizado_id	int	Sí	NULL	
estado_abandono	tinyint(1)	Sí	NULL	
resultado	double	Sí	NULL	
movimientos	longtext	Sí	NULL	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	no_partida	6	A	No	
partidas_participante_id_foreign	BTREE	No	No	participante_id	3	A	Sí	
partidas_torneo_id_foreign	BTREE	No	No	torneo_id	1	A	Sí	
partidas_desempate_utilizado_id_foreign	BTREE	No	No	desempate_utilizado_id	1	A	Sí	
partidas_ronda_torneo_id_foreign	BTREE	No	No	ronda_torneo_id	1	A	Sí	

partidas_individuales

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	bigint	No		
equipo_match_id	bigint	No		
jugador_a_id	varchar(20)	No		
jugador_b_id	varchar(20)	Sí	NULL	
tablero	int	No		
resultado	double	Sí	NULL	
created_at	timestamp	Sí	NULL	
updated_at	timestamp	Sí	NULL	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	
partidas_individuales_equipo_match_id_foreign	BTREE	No	No	equipo_match_id	0	A	No	
partidas_individuales_jugador_a_id_foreign	BTREE	No	No	jugador_a_id	0	A	No	
partidas_individuales_jugador_b_id_foreign	BTREE	No	No	jugador_b_id	0	A	Sí	

partidas_torneo

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	bigint	No		
ronda_id	bigint	No		
jugador_blanclas_id	varchar(20)	No		
jugador_negras_id	varchar(20)	Sí	NULL	
resultado	float	Sí	NULL	
mesa	int	Sí	NULL	
notas	text	Sí	NULL	
created_at	timestamp	Sí	NULL	
updated_at	timestamp	Sí	NULL	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	

Figura 90: Diccionario de datos del sistema 10

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
partidas_torneo_ronda_id_foreign	BTREE	No	No	ronda_id	0	A	No	
partidas_torneo_jugador_blanclas_id_foreign	BTREE	No	No	jugador_blanclas_id	0	A	No	
partidas_torneo_jugador_negras_id_foreign	BTREE	No	No	jugador_negras_id	0	A	Sí	

password_resets

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	bigint	No		
correo	varchar(255)	No		
token	varchar(255)	No		
created_at	timestamp	Si	NULL	
expires_at	timestamp	Si	NULL	

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	id	0	A	No	

permisos

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	int	No		
permiso	varchar(30)	No		
descripcion	varchar(100)	No		
grupo	varchar(15)	No		

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	id	36	A	No	

player_statistics

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	bigint	No		
participante_id	bigint	No		
torneo_id	bigint	No		
partidas_jugadas	int	No	0	
partidas_blanclas	int	No	0	
partidas_negras	int	No	0	
byes_recibidos	int	No	0	
flotamientos	int	No	0	
porcentaje_blanclas	double	No	0	
porcentaje_negras	double	No	0	
porcentaje_emparejamientos_repetidos	double	No	0	
created_at	timestamp	Sí	NULL	
updated_at	timestamp	Sí	NULL	

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Si	No	id	0	A	No	
player_statistics_participante_id_torneo_id_unique	BTREE	Si	No	participante_id	0	A	No	
player_statistics_torneo_id_foreign	BTREE	No	No	torneo_id	0	A	No	

Figura 91: Diccionario de datos del sistema 11

puntajes_elos

Columna	Tipo	Nulo	Predeterminado	Comentarios
fide_id_miembro (Primaria)	varchar(255)	No		
no_categoria_elos (Primaria)	bigint	No		
elo	int	No		

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	fide_id_miembro	16	A	No	
				no_categoria_elos	16	A	No	
puntajes_elos_no_categoria_elos_foreign	BTREE	No	No	no_categoria_elos	1	A	No	

roles

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	bigint	No		
nombre	varchar(50)	No		
descripcion	text	Sí	NULL	
created_at	timestamp	Sí	NULL	
updated_at	timestamp	Sí	NULL	

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	4	A	No	

rondas_torneo

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	bigint	No		
torneo_id	bigint	No		
numero_ronda	int	No		
fecha_hora	datetime	No		
completada	tinyint(1)	No	0	
created_at	timestamp	Sí	NULL	
updated_at	timestamp	Sí	NULL	

índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	
torneo_id	BTREE	Sí	No	torneo_id	0	A	No	
numero_ronda	BTREE	Sí	No	numero_ronda	0	A	No	

sessions

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	varchar(255)	No		
user_id	bigint	Sí	NULL	
ip_address	varchar(45)	Sí	NULL	
user_agent	text	Sí	NULL	
payload	longtext	No		
last_activity	int	No		

Figura 92: Diccionario de datos del sistema 12

puntajes_elos

Columna	Tipo	Nulo	Predeterminado	Comentarios
fide_id_miembro (Primaria)	varchar(255)	No		
no_categoria_elos (Primaria)	bigint	No		
elo	int	No		

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	fide_id_miembro	16	A	No	
				no_categoria_elos	16	A	No	
puntajes_elos_no_categoria_elos_foreign	BTREE	No	No	no_categoria_elos	1	A	No	

roles

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	bigint	No		
nombre	varchar(50)	No		
descripcion	text	Sí	NULL	
created_at	timestamp	Sí	NULL	
updated_at	timestamp	Sí	NULL	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	4	A	No	

rondas_torneo

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	bigint	No		
torneo_id	bigint	No		
numero_ronda	int	No		
fecha_hora	datetime	No		
completada	tinyint(1)	No	0	
created_at	timestamp	Sí	NULL	
updated_at	timestamp	Sí	NULL	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	
rondas_torneo_torneo_id_numero_ronda_unique	BTREE	Sí	No	torneo_id	0	A	No	
				numero_ronda	0	A	No	

sessions

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (Primaria)	varchar(255)	No		
user_id	bigint	Sí	NULL	
ip_address	varchar(45)	Sí	NULL	
user_agent	text	Sí	NULL	
payload	longtext	No		
last_activity	int	No		

Figura 93: Diccionario de datos del sistema 13

Columna	Tipo	Nulo	Predeterminado	Comentarios
estado_torneo	tinyint(1)	Sí	1	
torneo_cancelado	tinyint(1)	No	0	
motivo_cancelacion	varchar(255)	Sí	NULL	
sistema_emparejamiento_id	bigint	Sí	NULL	
usr_buchholz	tinyint(1)	No	0	
usr_sonneborn_berger	tinyint(1)	No	0	
usr_desempate_progresivo	tinyint(1)	No	0	
numero_minimo_participantes	int	No	4	
permitir_bye	tinyint(1)	No	1	
max_byes_por_jugador	int	No	1	
diferencia_maxima_puntos	int	No	2	
alternar_colores	tinyint(1)	No	1	
evitar_emparejamientos_repetidos	tinyint(1)	No	1	
maximo_emparejamientos_repetidos	int	No	1	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	
torneos_categoria_torneo_id_foreign	BTREE	No	No	categoriaTorneo_id	0	A	Sí	
torneos_control_tiempo_id_foreign	BTREE	No	No	control_tiempo_id	0	A	Sí	
torneos_organizador_id_foreign	BTREE	No	No	organizador_id	0	A	Sí	
torneos_director_torneo_id_foreign	BTREE	No	No	director_torneo_id	0	A	Sí	
torneos_arbitro_principal_id_foreign	BTREE	No	No	arbitro_principal_id	0	A	Sí	
torneos_arbitro_id_foreign	BTREE	No	No	arbitro_id	0	A	Sí	
torneos_arbitroAdjunto_id_foreign	BTREE	No	No	arbitroAdjunto_id	0	A	Sí	
torneos_federacion_id_foreign	BTREE	No	No	federacion_id	0	A	Sí	
torneos_sistema_emparejamiento_id_foreign	BTREE	No	No	sistema_emparejamiento_id	0	A	Sí	

usuarios

Columna	Tipo	Nulo	Predeterminado	Comentarios
id_email (Primaria)	bigint	No		
correo	varchar(40)	No		
contrasena	varchar(80)	No		
rol_id	bigint	Sí	NULL	
usuario_estado	tinyint(1)	No	1	
intentos_updatePass	int	Sí	NULL	
remember_token	varchar(100)	Sí	NULL	
created_at	timestamp	Sí	NULL	
updated_at	timestamp	Sí	NULL	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id_email	11	A	No	
usuarios_correo_unique	BTREE	Sí	No	correo	11	A	No	
usuarios_rol_id_foreign	BTREE	No	No	rol_id	4	A	Sí	

10.2. Matriz de pruebas de caja negra

Figura 94: Matriz de pruebas de caja negra implementadas 1



ESTRELLAS DEL AJEDREZ

+505 8286-1396
Ciudad Sandino Oro verde, del restaurante TipTop 1 Cuadra al norte 7 Cuadras al Este 75 varas al Norte.

Septiembre 16, 2025

La presente matriz de pruebas corresponde a una serie de casos diseñados bajo la técnica de pruebas de caja negra, con el objetivo de verificar el comportamiento funcional del sistema sin necesidad de conocer su estructura interna o el código fuente. Este enfoque se centra en evaluar si el sistema responde correctamente a una variedad de entradas, produciendo las salidas esperadas de acuerdo con los requisitos funcionales establecidos.

Las pruebas se han estructurado en función de módulos específicos del sistema, considerando entradas válidas, inválidas y condiciones de error.

Asimismo, se incorpora una columna de Estado, en la cual se registra el resultado de cada prueba de la siguiente manera:

- Un símbolo ✓ indica que el caso de prueba fue exitoso, es decir, que el sistema se comportó conforme a lo esperado.
- Un símbolo X señala que la prueba falló, reflejando un comportamiento incorrecto o no previsto del sistema.

Cabe destacar que todas las pruebas fueron ejecutadas por el instructor de la academia, **Ian Eiffel Castillo Sevilla**, quien, como uno de los usuarios finales del sistema, aportó una perspectiva práctica y realista al proceso de validación. Su participación permitió identificar oportunidades de mejora en la experiencia de uso y en la robustez funcional del sistema.

Módulo	ID Prueba	Descripción	Entrada	Resultados esperados	Estado	Observaciones
Autenticación	AUTH-001	Login con credenciales válidas	Credenciales de usuario previamente registrado en el sistema.	Redirección al dashboard del sistema según el rol del usuario.	✓	
Autenticación	AUTH-002	Login con credenciales inválidas	Credenciales de usuario no registrado previamente en la base de datos, credenciales registradas, pero con error en el correo y/o contraseña.	Mensaje de error indicando el origen del problema.	✗	

Figura 95: Matriz de pruebas de caja negra implementadas 2



Módulo	ID Prueba	Descripción	Entrada	Resultados esperados	Estado	Observaciones
Torneos	TORN-001	Crear un nuevo torneo	Datos de torneo completos y válidos.	Mensaje de éxito posterior a la creación del torneo y actualización automática de la tabla de torneos.	✓	
Torneos	TORN-002	Crear un torneo sin datos obligatorios y con una fecha que ya ha transcurrido	Datos incompletos válidos y/o fecha que ya ha transcurrido.	Mensajes de campos requeridos en los campos vacíos y bloqueo de interacciones con el torneo debido a que la fecha ya ha transcurrido.	✓	<ul style="list-style-type: none"> • No permite crear el torneo • Torneo con fecha transcurrida no se puede interactuar
Torneos	TORN-003	Inscribir participantes	Selección de participantes a competir en el torneo.	Recarga del módulo, listado de los participantes registrados y habilitación del botón para generar emparejamientos	✓	
Torneos	TORN-004	Generar emparejamiento	Generar emparejamiento posterior a la selección de participantes.	Emparejamientos generados y habilitación de rondas creadas por el sistema.	✓	<ul style="list-style-type: none"> • Se realizó el emparejamiento con sistema suizo y Round R.
Torneos	TORN-005	Registrar resultados de partida	Resultados obtenidos en los enfrentamientos por ronda.	Actualización de resultados en el módulo de partidas y bloqueo de edición de resultados.	✓	
Torneos	TORN-006	Exportación de registros de torneos	Exportar los registros de torneos.	Descarga de documento .csv con los torneos registrados en el sistema.	✓	

Figura 96: Matriz de pruebas de caja negra implementadas 3



ESTRELLAS DEL AJEDREZ

+505 8286-1396
Ciudad Sandino Oro verde, del restaurante TipTop 1 Cuadra al norte 7 Cuadras al Este 75 varas al Norte.

Módulo	ID Prueba	Descripción	Entrada	Resultados esperados	Estado	Observaciones
Torneos	TORN-007	Editar torneo registrado	Datos actualizados del torneo completos y válidos.	Mensaje de éxito posterior a la edición del torneo y actualización automática de la tabla de torneos.	✓	
Torneos	TORN-008	Eliminar torneo registrado	Selección del torneo a eliminar.	Mensaje de confirmación y eliminación posterior a aceptar eliminación del torneo.	✓	
Torneos	TORN-009	Cancelar torneo registrado	Selección del torneo a cancelar.	Mensaje de confirmación y solicitud del motivo de la cancelación.	✓	
Miembros	MEM-001	Crear un nuevo miembro	Datos completos y válidos correspondientes al nuevo miembro.	Mensaje de éxito y actualización automática de la tabla de miembros posterior al envío del formulario.	✓	
Miembros	MEM-002	Crear un nuevo miembro sin datos obligatorios	Datos incompletos válidos.	Mensajes de campos requeridos en los campos vacíos.	✓	
Miembros	MEM-003	Editar miembros existentes	Datos actualizados del miembro completos y válidos.	Mensaje de éxito posterior a la actualización del registro y actualización automática de la tabla miembros.	✓	
Miembros	MEM-004	Eliminar miembro registrado	Selección del torneo a eliminar.	Mensaje de confirmación y eliminación posterior a aceptar eliminación del miembro.	✓	

Figura 97: Matriz de pruebas de caja negra implementadas 4



ESTRELLAS DEL AJEDREZ

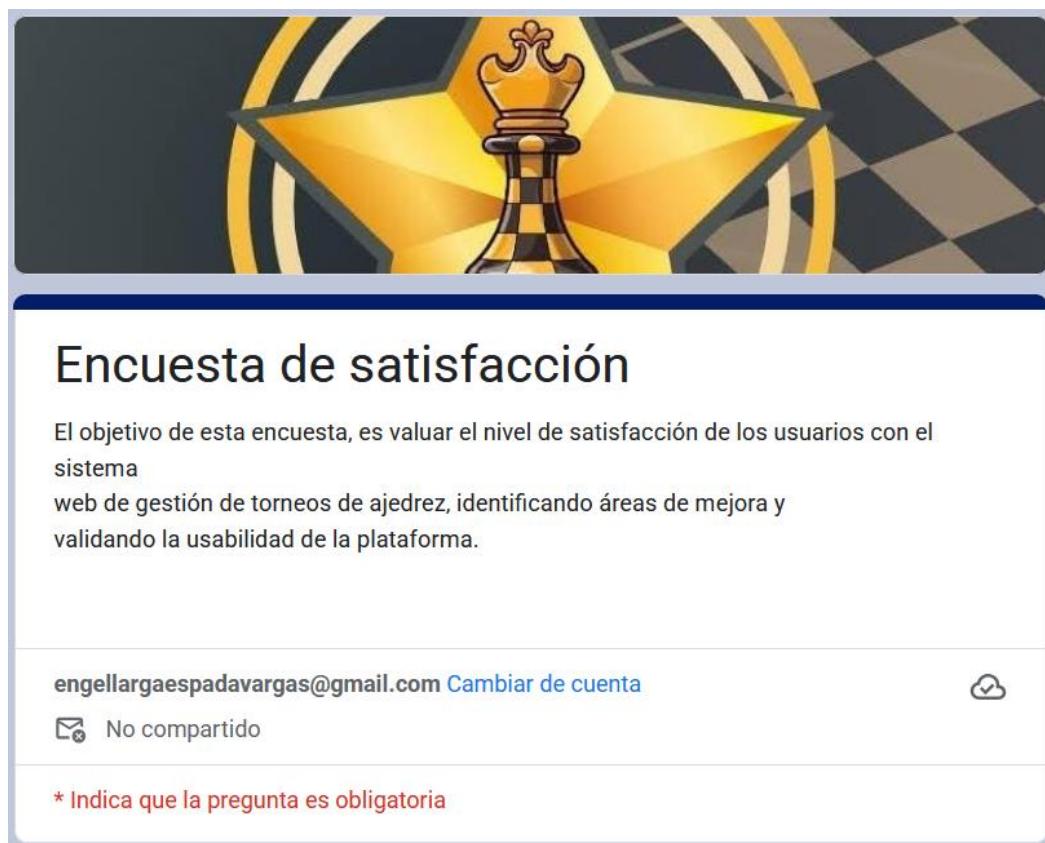
+505 8286-1396
Ciudad Sandino Oro verde, del restaurante TipTop 1 Cuadra al norte 7 Cuadras al Este 75 varas al Norte.

Módulo	ID Prueba	Descripción	Entrada	Resultados esperados	Estado	Observaciones
Miembros	MEM-005	Exportación de registros de miembros	Exportar los registros de miembros.	Descarga de documento .csv con los miembros registrados en el sistema.	✓	
Permisos	PERM-001	Asignar permisos a rol	Selección del rol y grupo a asignar permiso.	Actualización automática de la tabla de permisos.	✓	
Permisos	PERM-002	Remover permisos a rol	Selección del rol y grupo a remover permiso.	Actualización automática de la tabla de permisos.	✓	
Permisos	PERM-003	Verificación permisos	Inicio de sesión con un usuario cuyo rol haya sido modificado en sus permisos.	Visualización de nuevas funciones y/o ocultamiento de funciones posterior a la modificación de su rol.	✓	
Auditorías	AUD-001	Ver registros y exportación de registros de auditorías	Exportar registros de cambios en los distintos módulos del sistema.	Descarga de documento .csv con los cambios registrados en el sistema.	✓	
Estadísticas	EST-001	Ver estadísticas generales	Visualización e interacción con el gráfico de estadísticas generales.	Gráfico dinámico con diversas funciones para visualizar y exportar su información.	✓	
Estadísticas	EST-002	Ver estadísticas personales	Visualización e interacción con el gráfico de estadísticas personales.	Gráfico dinámico con diversas funciones para visualizar y exportar su información.	✓	

Tan Esmeralda Castillo Sevilla
Instructor

10.3. Resultados de encuesta de satisfacción de usuario

Figura 98: Encuesta de satisfacción de usuario



The screenshot shows a survey titled 'Encuesta de satisfacción' (Survey of satisfaction). The objective is to evaluate user satisfaction with the system, identify improvement areas, and validate the platform's usability. The survey is shared with engellargaespaldavargas@gmail.com and is not shared. A note indicates that an asterisk (*) marks required questions.

Encuesta de satisfacción

El objetivo de esta encuesta, es valuar el nivel de satisfacción de los usuarios con el sistema web de gestión de torneos de ajedrez, identificando áreas de mejora y validando la usabilidad de la plataforma.

engellargaespaldavargas@gmail.com Cambiar de cuenta

✉ No compartido

* Indica que la pregunta es obligatoria

Figura 99: Resultados de encuesta de satisfacción 1

¿Qué tan satisfecho se encuentra con el sistema? *(Califique del 1 al 5, donde 1 = Muy Insatisfecho y 5 = Muy Satisfecho)*

22 respuestas

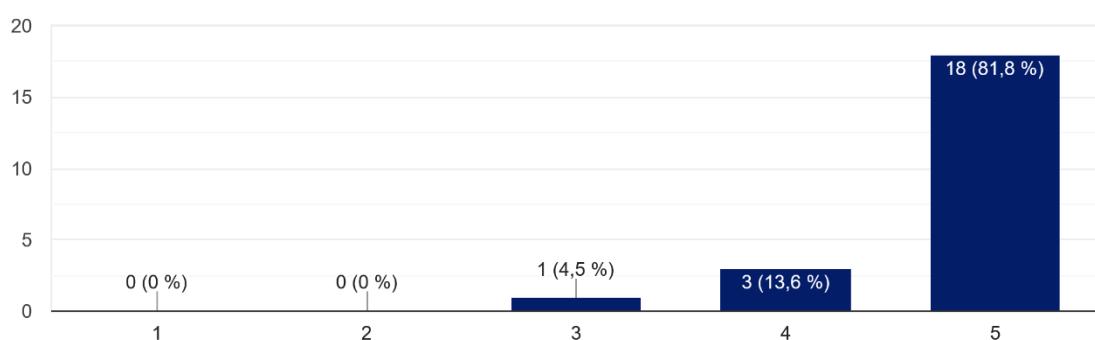


Figura 100: Resultados de encuesta de satisfacción 2

¿Qué tan fácil le resulta navegar por el sistema?

22 respuestas

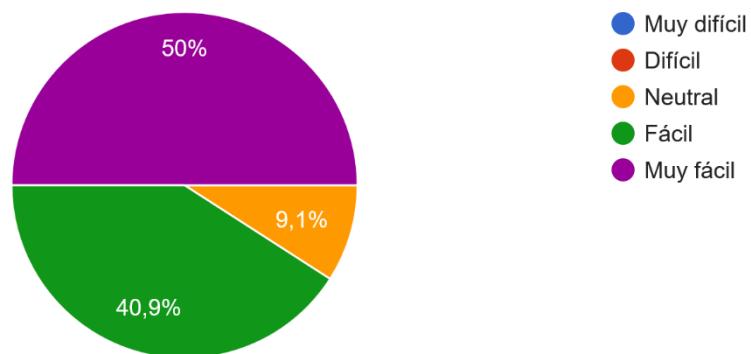


Figura 101: Resultados de encuesta de satisfacción 3

¿Qué tan intuitiva le parece la interfaz del sistema?

22 respuestas

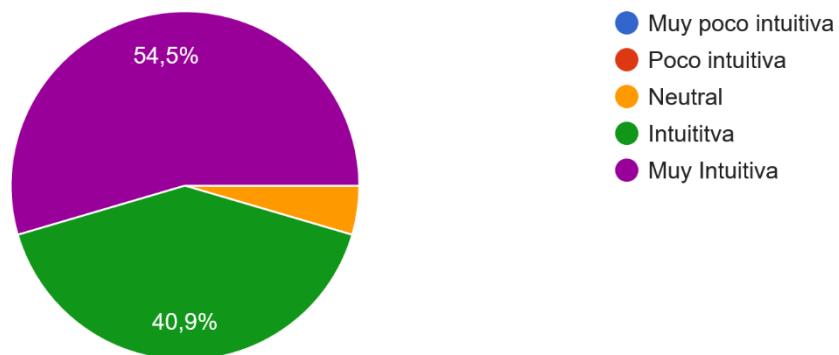


Figura 102: Resultados de encuesta de satisfacción 4

¿Con qué frecuencia necesitó ayuda técnica o soporte?

22 respuestas

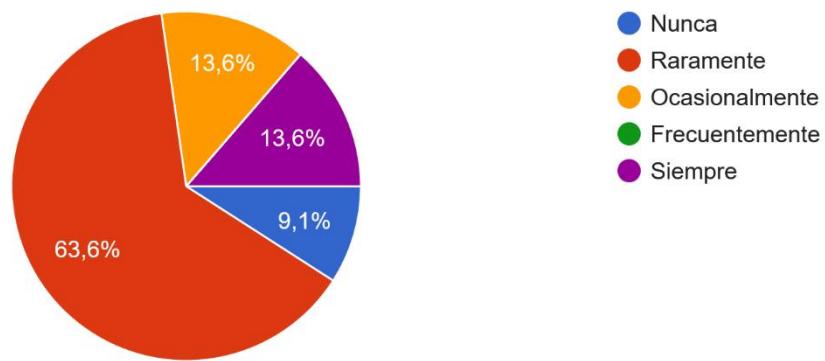


Figura 103: Resultados de encuesta de satisfacción 5

Gestión de Miembros

22 respuestas

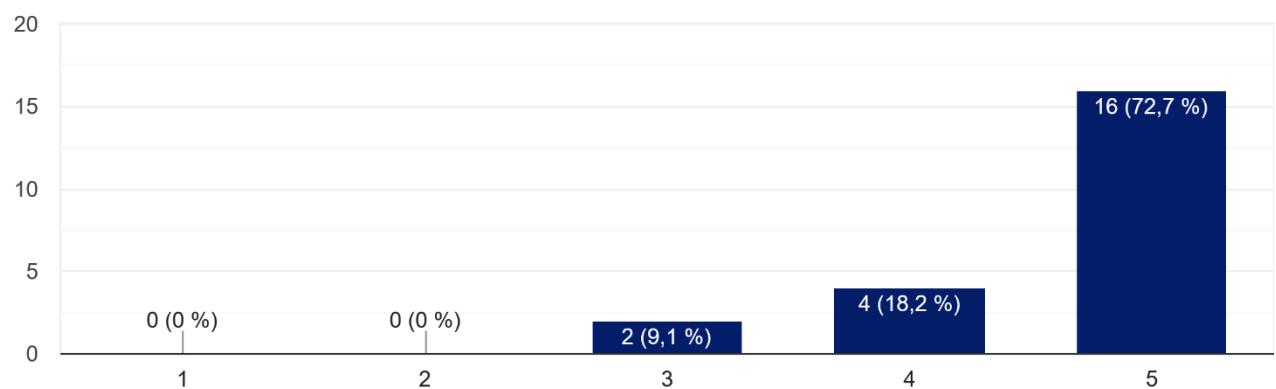


Figura 104: Resultados de encuesta de satisfacción 6

Gestión de Academias

22 respuestas

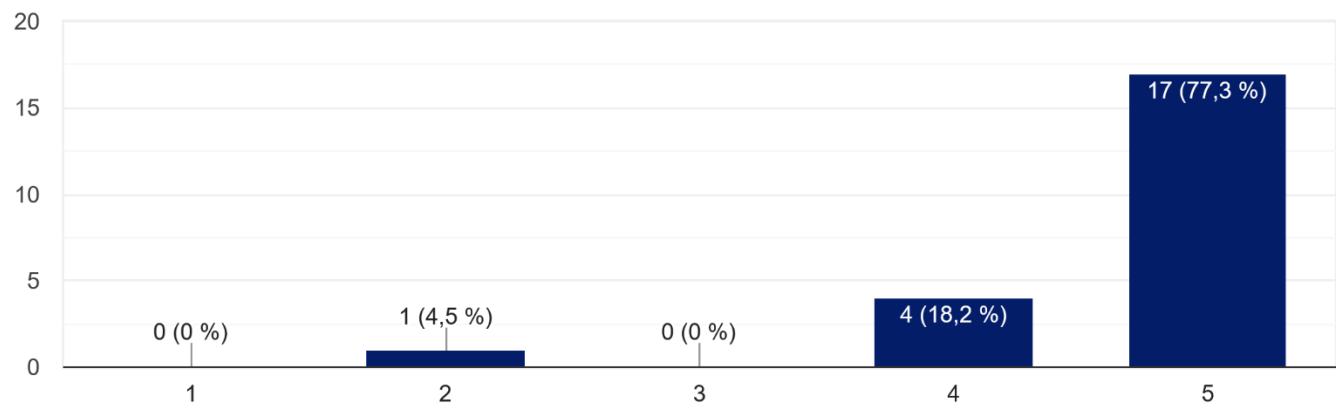


Figura 105: Resultados de encuesta de satisfacción 7

Gestión de Usuarios y Permisos

22 respuestas

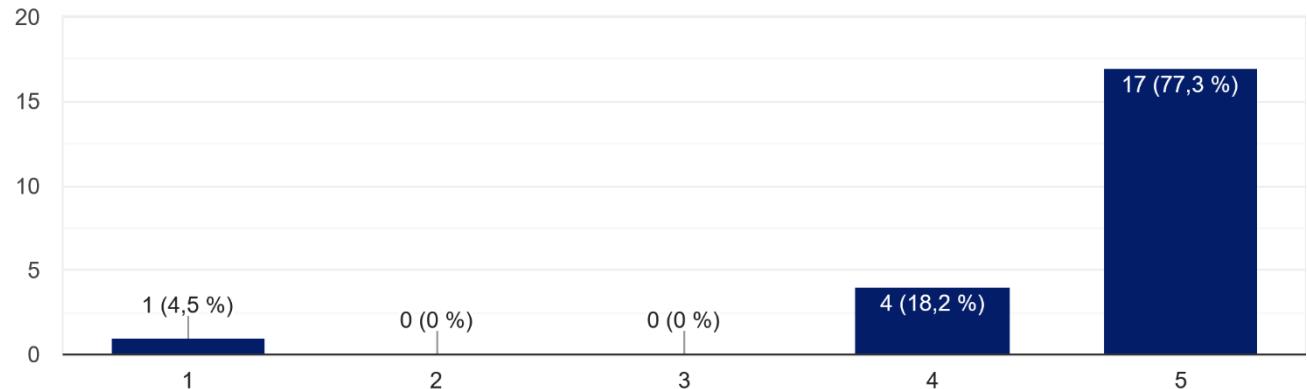


Figura 106: Resultados de encuesta de satisfacción 8

Sistema de Auditoría

22 respuestas

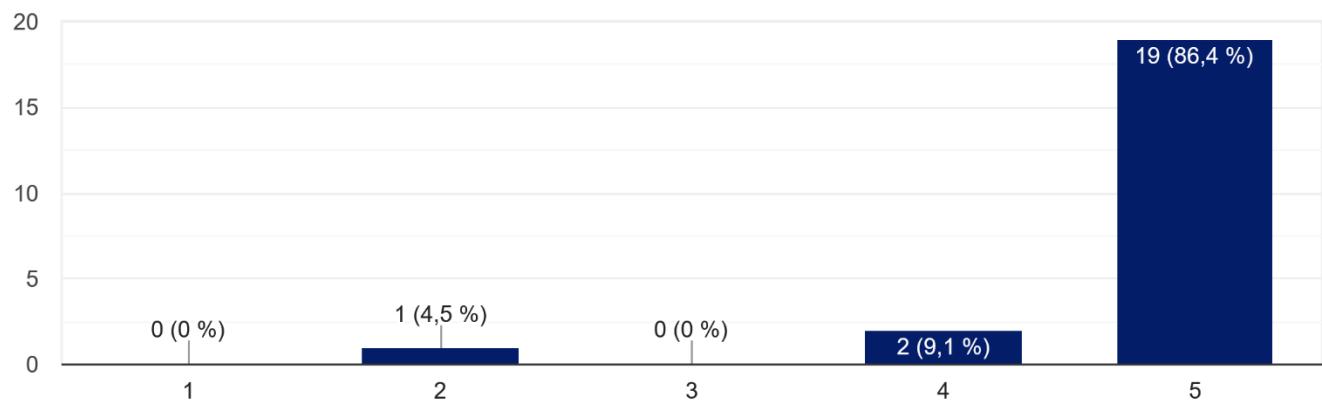


Figura 107: Resultados de encuesta de satisfacción 9

Análisis de Partidas

22 respuestas

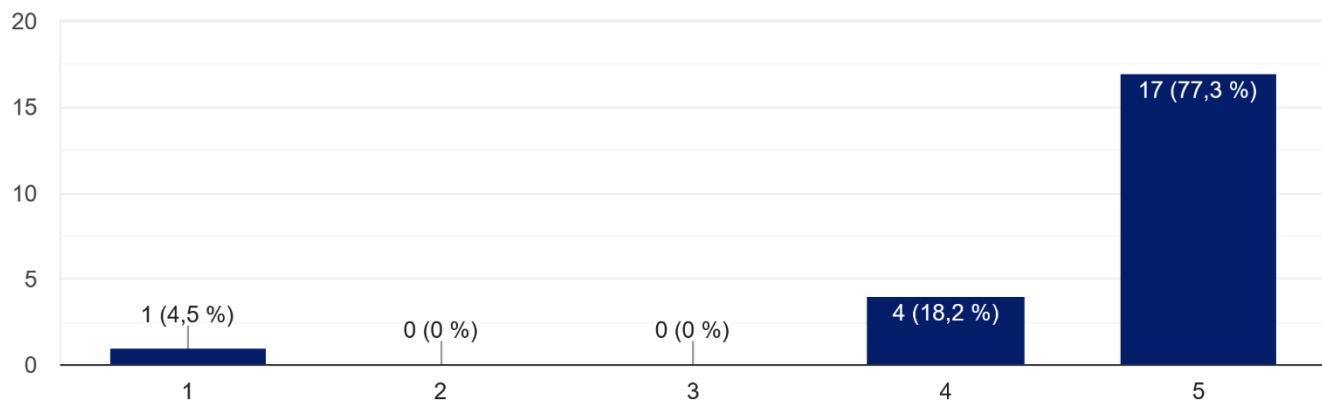


Figura 108: Resultados de encuesta de satisfacción 10

Reportes y Estadísticas

22 respuestas

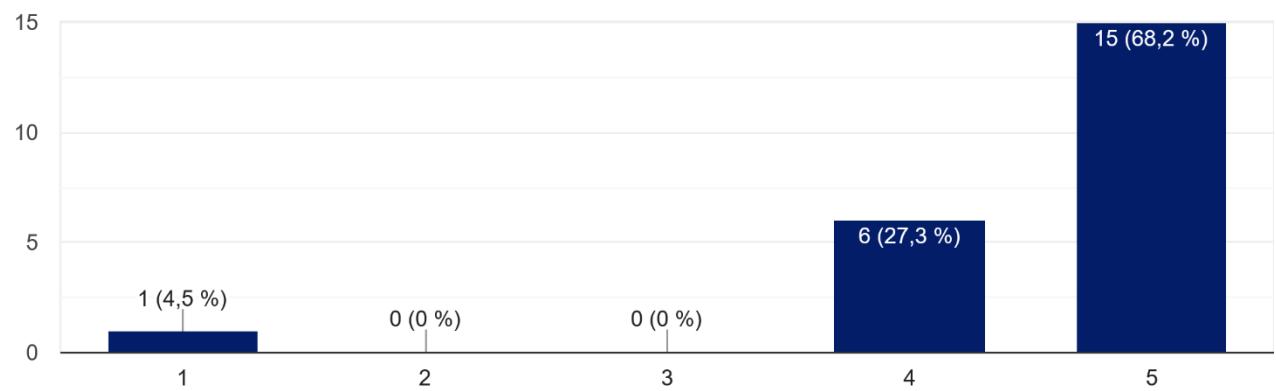


Figura 109: Resultados de encuesta de satisfacción 11

¿Qué tan rápido le parece el sistema?

22 respuestas

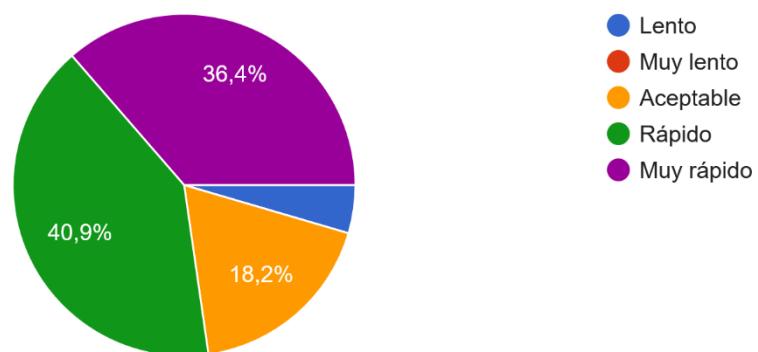


Figura 110: Resultados de encuesta de satisfacción 12

¿Con qué frecuencia experimenta errores o fallos en el sistema?
22 respuestas

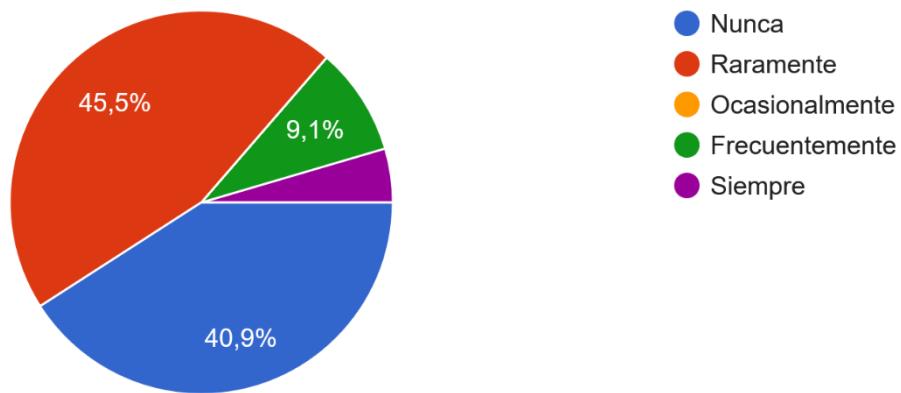


Figura 111: Resultados de encuesta de satisfacción 13

¿Qué tan confiable le parece el sistema?
22 respuestas

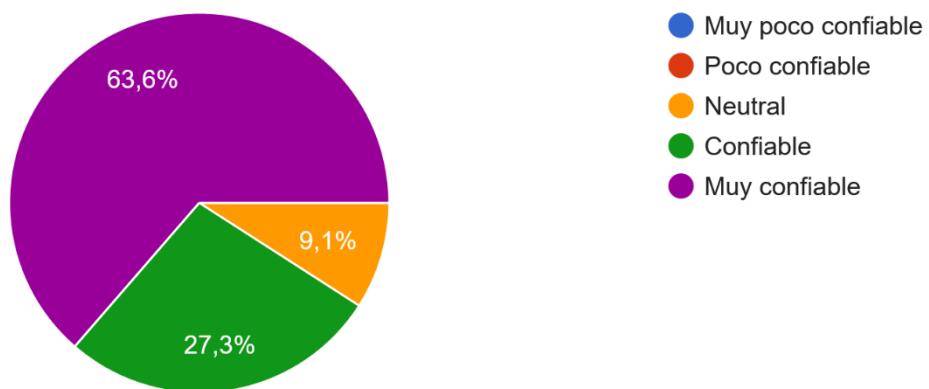


Figura 112: Resultados de encuesta de satisfacción 14

¿Tiene alguna sugerencia específica para mejorar el sistema?

12 respuestas

No

Ninguna, más que excelente

Agregar modo oscuro

Ninguna, todo muy correcto

Agregar modo oscuro

Ninguna

El uso de ciertos íconos pudieran reemplazarse por otros más agradables a la vista. Operativamente cumple con sus funciones.

Deberían de agregar un modo de aspecto oscuro.

Todo excelente, me gusta mucho, una gran ayuda el sistema

10.4. Estudio de Factibilidad

10.4.1. *Factibilidad Técnica*

Análisis de Requisitos Técnicos

El sistema propuesto se desarrolló utilizando tecnologías de código abierto y ampliamente soportadas, como Laravel 12, PHP 8.2+, MySQL, Bootstrap, Tailwind CSS y JavaScript, lo que asegura compatibilidad, escalabilidad y facilidad de mantenimiento.

Para la infraestructura, se propone el uso de servidores en la nube con especificaciones mínimas de 16GB de RAM, 4 vCPUs y 100GB SSD, lo cual garantiza un rendimiento estable. Estos recursos pueden adquirirse mediante servicios de proveedores como AWS, Azure o Digital Ocean, con planes que se adaptan al crecimiento futuro del sistema.

Se concluye que el sistema es técnicamente viable, ya que no requiere hardware especializado ni licencias costosas, aprovechando tecnologías accesibles al entorno académico en Nicaragua.

10.4.2. *Factibilidad Económica*

Costos Estimados

El análisis económico contempla costos asociados al desarrollo inicial, mantenimiento anual, capacitación de usuarios y contingencias.

Tabla 13*Servicios de Mantenimiento o Infraestructura*

Concepto	Descripción	Rango de Costo Estimado (USD)	Rango de Costo Estimado (C\$)	Observaciones
Hosting o Servidor Web	Renta anual de hosting en nube con 4 vCPU, 8–16 GB RAM y 100 GB SSD.	\$300 – \$500 / año	C\$11,100 – C\$18,500 / año	Basado en proveedores como Hostinger, DigitalOcean y Contabo.
Dominio y SSL	Registro de dominio .com/.org y certificado de seguridad SSL.	\$25 – \$40 / año	C\$925 – C\$1,480 / año	Basado en precios de proveedores nicaragüenses como Neothek, Host.com.ni y Neolo. Certificados SSL desde \$29 USD/año.
Mantenimiento técnico	Soporte, copias de seguridad y monitoreo del sistema.	\$400 – \$600 / año	C\$14,800 – C\$22,200 / año	Puede realizarse de forma externa o interna.
Imprevistos o actualizaciones	Cambios no planificados, mejoras de seguridad o diseño.	\$100 – \$200 / año	C\$3,700 – C\$7,400 / año	Monto preventivo.
Total, anual estimado: USD ≈ 1,200 / C\$N ≈ 44,400				

Nota: Estimaciones basadas en precios de hosting y servicios técnicos en Nicaragua publicados por Claro Empresas (2024), Contabo (2024), tarifas de mantenimiento locales según Freelancer.com (2024), (Neothek.com, s.f.), (Grupohost, s.f.)

Tabla 14*Costos de Desarrollo (único)*

Concepto	Descripción	Rango de Costo Estimado (USD)	Rango de Costo Estimado (C\$)	Observaciones
Desarrollo Backend	Programación en Laravel y MySQL.	\$350 – \$500	C\$12,950 – C\$18,500	Proyecto mediano (40–60 horas de trabajo).
Desarrollo Frontend	Interfaz en Bootstrap/Tailwind y JavaScript.	\$300 – \$400	C\$11,100 – C\$14,800	Incluye maquetado adaptable.
Pruebas y Documentación	QA, pruebas funcionales y manual técnico.	\$150 – \$250	C\$5,550 – C\$9,250	Basado en prácticas de QA estándar.
Implementación y capacitación	Puesta en marcha y formación de usuarios.	\$100 – \$200	C\$3,700 – C\$7,400	Taller de uso inicial.
Total estimado (único)		≈ \$1,000 – \$1,200	≈ C\$37,000 – C\$44,400	Pago único por entrega del sistema.

Nota: Costos calculados con base en tarifas promedio cobradas por desarrolladores freelance en Nicaragua (USD \$8–\$12 por hora), según Tarifas promedio de programadores freelance en Centroamérica (Workana, 2024), (Freelancer.com, 2024) y entrevistas locales realizadas por el INATEC (Panorama del sector tecnológico y empleabilidad TIC en Nicaragua. Managua, Nicaragua., 2024)

Tabla 15**Costos Totales**

Categoría	Costo Estimado (USD)	Costo Estimado (C\$)
Mantenimiento e infraestructura (anual)	\$1,200 / año	C\$44,400 / año
Desarrollo inicial (único)	\$1,100 (promedio)	C\$40,700 (promedio)
Total, primer año	≈ \$2,300	≈ C\$85,100

Nota: Elaboración propia con base en estimaciones de mercado nicaragüense 2024–2025, considerando cotizaciones de hosting, tarifas freelance y reportes de costos tecnológicos locales (Freelancer.com, 2024); (Panorama del sector tecnológico y empleabilidad TIC en Nicaragua. Managua, Nicaragua., 2024); (Hostinger, 2024).

En conclusión, los costos estimados resultan accesibles considerando la magnitud del sistema y su proyección a futuro.

Análisis Costos-Beneficios

Beneficios

- Reducción de tiempos en los procesos de inscripción, generación de emparejamientos y registro de resultados.
- Eliminación de errores manuales en el cálculo de desempates y puntuaciones.
- Mayor transparencia y confiabilidad en la gestión de torneos, fortaleciendo la reputación institucional.
- Ahorro de recursos administrativos y materiales, al reducir el uso de papel y tareas repetitivas.
- Contribución a la formación de los jugadores, mediante el análisis automatizado de partidas y estadísticas de desempeño.

Costos de Inversión

- Inversión inicial única (desarrollo del sistema):

USD \approx 1,100 / C\$ \approx 40,700.

(Incluye backend, frontend, pruebas, documentación y capacitación inicial).

- Mantenimiento anual e infraestructura:

USD \approx 1,200 / C\$ \approx 44,400.

(Cubre hosting, dominio, soporte técnico, copias de seguridad e imprevistos).

- Total, primer año (desarrollo + mantenimiento):

USD \approx 2,300 / C\$ \approx 85,100.

Relación Costo–Beneficio

La inversión resulta altamente viable para el contexto nicaragüense, ya que los costos de desarrollo son moderados y se realizan por entrega de proyecto, práctica común en el sector tecnológico local (Workana, 2024), (Panorama del sector tecnológico y empleabilidad TIC en Nicaragua. Managua, Nicaragua., 2024).

El sistema genera un retorno intangible significativo al optimizar la gestión de torneos, reducir los errores humanos y mejorar la experiencia tanto de jugadores como de administradores. Además, su mantenimiento accesible garantiza la sostenibilidad del proyecto a largo plazo sin requerir infraestructura física adicional.

En conclusión, la relación costo-beneficio demuestra que la inversión está plenamente justificada por el impacto positivo en la eficiencia, la calidad de gestión y el desarrollo competitivo de los jugadores.

10.4.3. *Factibilidad Operativa*

Adaptación a las Necesidades del Usuario

El sistema cuenta con una interfaz amigable y accesible desde dispositivos móviles y de escritorio, lo cual facilita la adopción por parte de los usuarios.

La capacitación prevista asegura que los administradores y jugadores se adapten rápidamente al sistema. Además, la academia ya posee experiencia en procesos digitales, lo que minimiza la resistencia al cambio.

La factibilidad operativa es alta, garantizando que el sistema será utilizado de manera efectiva en el entorno real.

Evaluación de Riesgos

- **Fallas técnicas:** Riesgo de caída del servidor → mitigado con respaldos automáticos.
- **Seguridad de la información:** Riesgo de accesos no autorizados → mitigado con cifrado de contraseñas y control de roles.
- **Capacitación insuficiente:** Riesgo de baja adopción → mitigado con talleres y manual de usuario.
- **Limitaciones financieras:** Riesgo de no cubrir mantenimiento → mitigado con planes de financiamiento escalable

Los riesgos identificados son moderados y cuentan con medidas de mitigación adecuadas.

10.4.4. *Estudio de Factibilidad Legal*

Cumplimiento de Normativas, Regulaciones y Propiedad Intelectual

El sistema hace uso de herramientas de software libre, lo cual es legal en Nicaragua siempre que se respeten las licencias (ejemplo: MIT para Laravel, GPL para MySQL).

Validez Jurídica de Operaciones Electrónicas

La Ley No. 729, Ley de Firma Electrónica, otorga respaldo a la validez de registros digitales, lo cual es aplicable a los procesos de inscripción y resultados dentro del sistema. (Nicaragua, 2011)

Contratos y Licencias

El sistema debe cumplir con la Ley No. 787, Ley de Protección de Datos Personales, garantizando la seguridad de la información de los jugadores. Asimismo, la academia deberá establecer políticas de uso, privacidad y respaldo de datos para asegurar el cumplimiento legal. (Nicaragua, 2012)

Área de Conocimiento de Tecnología de la
Información y Comunicación

“Sistema web de gestión de torneos de ajedrez para la Academia “Estrellas del Ajedrez”

Trabajo Monográfico para optar al título de
Ingeniero en Computación

Elaborado por:

Br. Engel Antonio
Largaespada Vargas
Carnet: 2019-0328U

Br. Laisha Nazaret
Acevedo García
Carnet: 2019-1161U

Tutor:

MSc. Luis Eduardo
Chávez Mairena