# UNIVERSIDAD ESTATAL DE BOLÍVAR

## FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN EMPRESARIAL E INFORMÁTICA

CARRERA DE SOFTWARE

**TRABAJO DE INTEGRACIÓN CURRICULAR**
**PREVIO A LA OBTENCIÓN DEL TÍTULO DE**
**INGENIERO(A)(S) EN SOFTWARE**

**FORMA:** PROYECTO TECNOLÓGICO

**TEMA:**

"SISTEMA INTEGRAL DE GESTIÓN PARA CAMPEONATOS DEPORTIVOS EN LA UNIVERSIDAD ESTATAL DE BOLÍVAR."


**AUTORES:**

STALIN ALEXANDER ALVARADO CHONGO
LUIS AURELIO DUY QUINDI

**DIRECTOR:**

ING. EDGAR RIVADENEIRA


**GUARANDA – ECUADOR**

**2025**

**TEMA DEL PROYECTO TECNOLÓGICO**

"SISTEMA INTEGRAL DE GESTIÓN PARA CAMPEONATOS DEPORTIVOS EN LA UNIVERSIDAD ESTATAL DE BOLÍVAR."

## DEDICATORIA

Primero agradecer a Dios, quien nos da vida y salud en el día a día, nos cobija con su aliento y nos ayuda a seguir adelante. Siento que ha sido Su aliento el que me ha servido de refugio en esos días grises donde las fuerzas me flaqueaban, dándome ese empujón de último minuto para no tirar la toalla cuando el camino se ponía cuesta arriba.

Este logro se lo dedico, con el corazón en la mano, a mi familia, que ha sido mi motor y mi cable a tierra en cada paso que he dado. Ellos fueron los que se quedaron firmes a mi lado, aguantando el tirón conmigo en cada etapa, y los que más creyeron en lo que yo era capaz de hacer, precisamente cuando el estrés y las exigencias me tenían contra las cuerdas.

A mis padres, de manera muy especial, porque todo lo que soy hoy se lo debo a su esfuerzo incansable y a su entrega diaria. Su ejemplo ha sido como una brújula que ha guiado mis pasos y mis decisiones más difíciles; esa confianza ciega que siempre me tuvieron fue el motor real que impulsó este resultado. Sin sus sacrificios, estoy convencido de que este momento simplemente no habría llegado nunca.

Stalin Alvarado

**AGRADECIMIENTO**

Agradezco a la Universidad Estatal de Bolívar por el espacio académico que me permitió desarrollar este proyecto y aplicar lo aprendido en cada etapa. Al docente tutor por su guía directa y por cada observación que fortaleció el avance del documento. Su acompañamiento me ayudó a mejorar la estructura, el análisis y la claridad del trabajo. Agradezco al personal y a las áreas que colaboraron con la información necesaria para construir la propuesta. Cada dato y cada indicación hicieron posible completar los objetivos planteados. Agradezco a mis compañeros que compartieron ideas, revisiones y tiempo durante este proceso. Su apoyo constante hizo que el trabajo sea más sólido y más claro. Este proyecto es el resultado del esfuerzo conjunto de todas las personas que aportaron con conocimiento, orientación y compromiso.

Stalin Alvarado

**DEDICATORIA**

Dedico este logro primero a Dios, por prestarme la vida y permitirme llegar hasta este momento tan importante. Gracias por brindarme la sabiduría, la fortaleza y la perseverancia necesarias para superar cada obstáculo y continuar avanzando aun en los momentos más difíciles.

Con mucho esfuerzo, cariño y con el corazón lleno de felicidad le dedico esta meta para mi hijo (Gael Duy) por ser el motivo e inspiración para yo continuar de pie y seguir alcanzando muchas metas, así mismo le dedico a mi compañera del alma a mi esposa Ligia Cando, gracias por caminar conmigo en cada etapa, por entender mis ausencias, por motivarme cuando el cansancio aparecía y por confiar en mí incluso en los momentos en los que yo dudaba.

Ha si mismo le dedicó este logro con mucha alegría y amor a mi madre (María Quindi) y mis hermanos (José, Eliseo, Carmen, Manuel, Francisco, Chanita y Mariana) para mis sobrinos (Jhon, Jhoselin, Naquely) gracias por apoyarme en esos momentos difíciles, por creer en mi cuando parecía que estaba perdido, gracias por no abandonarme, porque sin su apoyo este camino no habría sido el mismo, gracias por su amor, por su sacrificio y por creer siempre en mis capacidades, cada esfuerzo realizado también ha sido por ustedes, este logro también les pertenece.

Gracias a toda mi familia por ese apoyo por ser parte de este trayecto por ese granito de arena que me brindaron de manera especial a (Narcisa Duy & José Cando) quienes han sido un apoyo, Cada palabra de ánimo, cada consejo y cada pequeño gesto fueron ese granito de arena que hizo posible que hoy pueda alcanzar esta meta.

Me dedicó a mí mismo por creerme en mi mismo y no dejar de luchar por mis sueños, a pesar de las circunstancias, las malas noches, cansancios, los obstáculos que me puso la vida, no me e detenido con mucho esfuerzo, dedicación y fe todo es posible.

Luis Duy

**AGRADECIMIENTO**

Agradecido primeramente con Dios por el regalo de la vida y la salud, por guiar mi camino en cada paso que he dado y por darme la fuerza y la sabiduría necesarias para alcanzar una etapa más en mi vida.

De la misma manera agradecer con mucho cariño a mis hermanos(a) mi familia, amigos por apoyarme y brindar esa motivación ese anhelo de lucha, en la buenas y en las malas, de manera especial a mi madre (María Transito Quindi Tenezaca, y a mi hermano, José María Duy Quindi) por ser los pilares fundamentales de este trayecto y sostenerme de pie, a mi padre allá en el cielo, espero que se sientas orgulloso de su hijo.

Expreso mis más sinceros agradecimientos a cada uno de mis maestros ingenieros, quienes con dedicación, compromiso y vocación compartieron sus enseñanzas, conocimientos y experiencias a lo largo de mi formación académica.

Gracias a su guía constante y a su ejemplo profesional, hoy quiero decir que no solo adquirí conocimientos, sino también valores que fortalecen mi crecimiento como futuro profesional. De manera especial, extiendo mi gratitud al Ing. Rodrigo de Pozo y al Ing. Juan Manuel Galarza, quienes forman parte importante de este proceso de titulación, por su acompañamiento, orientación y valiosos aportes que contribuyeron significativamente a la culminación de este logro.

Como no agradecer a mi querido y bello Universidad Estatal de Bolivar, a nuestro querido rector al Dr. Arturo Rojas y cada uno de las autoridades, así también a la Facultad de Ciencias Administrativas Gestión Empresarial e Informática, por abrir las puertas, y pertenecer a ese bello y majestoso universidad, fue como nuestro segundo hogar, a mis amigos compañeros que día a día compartimos momentos de alegría gracias por ese apoyo que nos brindamos.

Expreso mi más sincero agradecimiento a mi tutor de carrera, el Ing. Edgar Rivadenera, por su valiosa orientación a lo largo de este proceso académico. Gracias por compartir generosamente sus conocimientos, experiencias y consejos, los cuales fueron fundamentales para el desarrollo de este trabajo. Aprecio profundamente su paciencia, dedicación y compromiso, así como el tiempo brindado para guiarme y realizar las observaciones necesarias durante la elaboración y corrección de la documentación. Su acompañamiento constante no solo fortaleció este proyecto, sino también mi formación profesional y personal.

Luis Duy

# CERTIFICADO DE VALIDACIÓN

UEB
UNIVERSIDAD ESTATAL DE BOLIVAR

FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN EMPRESARIAL E INFORMÁTICA

## CERTIFICADO DE VALIDACIÓN

Ing. Edgar Rivadeneira, Ing. Rodrigo del Pozo e Ing. Juan Galarza, en su orden director y Pares Académicos del Trabajo de Integración Curricular "SISTEMA INTEGRAL DE GESTIÓN PARA CAMPEONATOS DEPORTIVOS EN LA UNIVERSIDAD ESTATAL DE BOLÍVAR" desarrollado por el Sr. Stalin Alvarado y el Sr. Luis Duy

## CERTIFICAN

Que, luego de revisado el Trabajo de Integración Curricular en su totalidad, cumple con las exigencias académicas de la carrera SOFTWARE, por lo tanto, autorizamos su presentación y defensa.

Guaranda, 03 de marzo del 2026

Firmado electrónicamente por: EDGAR PATRICIO RIVADENEIRA RAMOS
Validar únicamente con FirmaEC

Ing. Edgar Rivadeneira
**Director**

Firmado electrónicamente por: RODRIGO HUMBERTO DEL POZO DURANGO
Validar únicamente con FirmaEC

Ing. Rodrigo del Pozo
**Par Académico**

Firmado electrónicamente por: JUAN MANUEL GALARZA SCHOENFELD
Validar únicamente con FirmaEC

Ing. Juan Galarza
**Par Académico**

Dirección: Av. Ernesto Che Guevara y Gabriel Secaira
Guaranda-Ecuador
Teléfono: (593) 3220 6059
www.ueb.edu.ec

## DERECHOS DE AUTOR

UEB UNIVERSIDAD ESTATAL DE BOLIVAR

BIBLIOTECA GENERAL

**DERECHOS DE AUTOR**

Nosotros ALVARADO **CHONGO STALIN ALEXANDER** y **DUY QUINDI LUIS AURELIO** portador/res de la Cédula de Identidad No 1500982671 y 0302429733 en calidad de autor/res y titulares de los derechos morales y patrimoniales del Trabajo de Titulación:
**SISTEMA INTEGRAL DE GESTIÓN PARA CAMPEONATOS DEPORTIVOS EN LA UNIVERSIDAD ESTATAL DE BOLÍVAR**, modalidad **PROYECTO TECNOLÓGICO**, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN, concedemos a favor de la Universidad Estatal de Bolívar, una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Conservamos a mi/nuestro favor todos los derechos de autor sobre la obra, establecidos en la normativa citada.

Así mismo, autorizo/autorizamos a la Universidad Estatal de Bolívar, para que realice la digitalización y publicación de este trabajo de titulación en el Repositorio Digital, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Los autores declaran que la obra objeto de la presente autorización es original en su forma de expresión y no infringe el derecho de autor de terceros, asumiendo la responsabilidad por cualquier reclamación que pudiera presentarse por esta causa y liberando a la Universidad de toda responsabilidad.



Stalin Alvarado



Luis Duy

Dirección: Av. Ernesto Che Guevara y Gabriel Secaira
Guaranda-Ecuador
Teléfono: (593) 3220 6059
**www.ueb.edu.ec**

## INDICE DE CONTENIDOS

PORTADA ……1
**TEMA DEL PROYECTO TECNOLÓGICO ..... I**
**DEDICATORIA ..... II**
**AGRADECIMIENTO ..... III**
**DEDICATORIA ..... IV**
**AGRADECIMIENTO ..... V**
**CERTIFICADO DE VALIDACIÓN ..... VI**
**DERECHOS DE AUTOR ..... VII**
**INDICE DE CONTENIDOS ..... VIII**
**INDICE DE FIGURAS ..... XI**
**INTRODUCCIÓN ..... 1**
**RESUMEN ..... 2**
**ABSTRACT ..... 3**
**KICHWA ..... 4**
CAPÍTULO I ..... 5
FORMULACIÓN GENERAL DEL PROYECTO ..... 5
*1.1.Tema …… 5*
*1.2.Descripción del Problema ..... 5*
*1.3.Justificación ..... 6*
*1.4.Objetivos: General y Específicos ..... 7*
CAPÍTULO II ..... 9
MARCO TEÓRICO ..... 9
2.1. Antecedentes ..... 9
2.2.Científico…….10
2.3. Conceptual ..... 12
2.4.Legal……..14
2.5. Georreferencial ..... 15
CAPÍTULO III ..... 17
METODOLOGÍA ..... 17
3.1. Metodología de Desarrollo de Software ..... 17
CAPÍTULO IV ..... 20
INGENIERÍA DEL PROYECTO ..... 20
4.1 Marco de Trabajo: SCRUM ..... 20
4.2 Sprint 0: Identificación de Stakeholders y Aplicación de Instrumentos ..... 20
4.3 Sprint 1: Requerimientos Funcionales y No Funcionales ..... 21
4.3.1 Requerimientos Funcionales ..... 21
4.3.2 Requerimientos No funcionales ..... 22
4.3.3 Historias de Usuario ..... 23
4.4 Sprint 2: Diseño Arquitectónico del Sistema ..... 23

4.4.1 Vista arquitectónica general .....23
4.4.2 Componentes principales .....25
4.4.3 Patrón arquitectónico.....25
4.4.4 Diseño de Base de Datos .....26
4.4.5 Modelo Arquitectónico 4+1 de Kruchten .....27
4.4.6 Prototipos de interfaz .....35
4.5 Sprint 3: Codificación del Sistema.....38
4.6 Sprint 4: Pruebas Unitarias, Integración y Usabilidad .....40
4.7 Sprint 5: Implementación e Implantación del Sistema .....41
Despliegue.....41
4.8 Conclusiones del Capítulo .....42
CONCLUSIONES .....77
RECOMENDACIONES .....78
BIBLIOGRAFÍA .....79
ANEXOS .....80

# INDICE DE TABLAS

**Tabla 1.** Requerimientos funcionales del sistema .....21
**Tabla 2** Requerimientos No funcionales del sistema .....22
**Tabla 3.** Descripción del caso de uso: Gestionar Equipos .....32
**Tabla 4.** Descripción del caso de uso: Gestionar Torneos .....32
**Tabla 5.** Descripción del caso de uso: Programar Partidos .....33
**Tabla 6.** Descripción del caso de uso: Registrar Resultados .....33
**Tabla 7.** Descripción del caso de uso: Validar Jugador con QR .....34
**Tabla 8.** Descripción del caso de uso: Consultar Resultados .....34
**Tabla 9.** Descripción del caso de uso: Iniciar Sesión .....35
**Tabla 10.** Descripción del caso de uso: Gestionar Perfil .....35
**Tabla 11.** Resumen de pruebas aplicadas al sistema .....40
**Tabla 12.** Presupuesto ejecutado del proyecto .....80

## INDICE DE FIGURAS

Figura 1. Ubicación geográfica de la Universidad Estatal de Bolívar.....16
**Figura 2.** Diagrama de arquitectura cliente–servidor.....24
Figura 3. *Diagrama MVC general del sistema* .....25
**Figura 4.** Diagrama MVC detallado .....26
**Figura 5**. Diagrama del Patrón DAO.....26
**Figura 6.**Diagrama entidad–relación del sistema.....27
**Figura 7.** Vista lógica del modelo 4+1 .....28
**Figura 8.** Vista de procesos del modelo 4+1.....29
**Figura 9.** Vista de desarrollo del modelo 4+1.....29
**Figura 10.** Vista física (diagrama de despliegue).....30
**Figura 11.** Vista de escenarios (diagrama de casos de uso) .....31
**Figura 12.** Prototipos de interfaz del sistema Mockups .....36
**Figura 13.** Prototipos de interfaz del sistema Gestión de Torneos.....36
**Figura 14.** Prototipos de interfaz del sistema Gestión de Equipos.....37
**Figura 15.** Prototipos de interfaz del sistema Gestión de Jugadores .....37
**Figura 16.** Prototipos de interfaz del sistema Registro de Resultados .....38
**Figura 17.** Pantallas final Carnet .....38

# INTRODUCCIÓN

El avance de las innovaciones tecnológicas resulta ser fundamental en la administración de los procesos institucionales en la actualidad, debido a que permiten la automatización del desarrollo, el uso eficiente de los recursos y la implementación de herramientas para una toma de decisiones. En el caso de las universidades, se presenta el caso de los campeonatos deportivos, el miso que busca desarrollar el trabajo en equipo, la disciplina y socorrer en el bienestar físico y mental de los estudiantes.

La coordinación de los campeonatos deportivos en la Universidad Estatal de Bolívar, ha sido tradicionalmente aproximada a través de métodos manuales o mediante hojas de cálculo. De tal forma, que se generan retrasos, imprecisiones en los datos, y desaparición de la información. En consecuencia, nace la necesidad de diseñar el Sistema Integral de Gestión para Campeonatos Deportivos, para evitar lo anteriormente mencionado y automatizar la inscripción, programación, control de partidos, registro de resultados y generación de carnets digitales. La elaboración de este sistema tiene como objetivo asociar en una sola plataforma todos los módulos completados para la gestión deportiva para garantizar la seguridad, la integridad y la disposición de la información. Un proyecto tecnológico aprovechado como finalidad realizar y programar una aplicación web moderna y útil, implementada en un sistema de cliente-servidor y empleando las metodologías de programación actuales.

El sistema propuesto cuenta con una interfaz intuitiva para el usuario y adaptable a diversos dispositivos, incluyendo funciones que les simplifiquen las búsquedas a los administradores, representantes los árbitros; y la visualización de información por parte de los jugadores Además, se integrará un módulo de estadísticas que permitirá generar reportes automáticos sobre los resultados de los partidos y el rendimiento de los jugadores, fortaleciendo la transparencia y el control dentro de los campeonatos.

Por nuestro lado, y en concordancia con la línea de investigación de la Carrera de Ingeniería en Software, se proyecta el presente trabajo para atender la elaboración de tecnologías, cumpliendo con las definiciones que, producto de su validación, se adviertan como una herramienta para la gestión institucional. Por otro lado, permitirá avanzar en la modernización del sistema interno de la Universidad Estatal de Bolívar y, de esta forma, aportar un instrumento de beneficio al usuario final, facilitando la tendencia de la automatización y la normalización en el uso eficiente de las herramientas tecnológicas.

**RESUMEN**

La presente investigación tecnológica se circunscribe en el diseño conceptual y la implementación sistémica de una arquitectura informática orientada a la gobernanza integral de certámenes deportivos, cuyo objetivo primordial radica en la optimización de los protocolos de matriculación, el monitoreo sistemático de los agentes involucrados y la administración estratégica de los activos informacionales de carácter deportivo, toda vez que en la actualidad, un vasto sector de instituciones académicas y entidades deportivas aún ejecutan dichos procedimientos bajo esquemas manuales rudimentarios, situación que deriva inexorablemente en la recurrencia de errores operativos, el extravío de data crítica y una marcada latencia administrativa que obstaculiza la autenticación fidedigna de atletas y colegiados, por lo cual la solución propuesta ha sido cimentada sobre un modelo cliente-servidor, integrando Laravel como motor de desarrollo backend, React para la orquestación de la interfaz de usuario y PostgreSQL como sistema de gestión de bases de datos de alta integridad.

El ecosistema funcional del aplicativo comprende dimensiones operativas esenciales que abarcan desde el enrolamiento de usuarios y la gestión pormenorizada de plantillas, hasta la designación técnica de árbitros y la expedición de credenciales digitales mediante tecnología de códigos QR para la consulta analítica de métricas en tiempo real, mientras que a efectos de estructurar el despliegue técnico en ciclos iterativos de alto rendimiento, se procedió a la adopción del marco de trabajo SCRUM, lo que facilitó un escrutinio progresivo de los hitos alcanzados y una validación empírica constante de la plataforma, evidenciándose así una transformación cualitativa en la arquitectura organizativa y en los niveles de seguridad operativa que mitigan sustancialmente los tiempos de respuesta en el registro de los participantes.

Los hallazgos derivados de la implementación de este estudio testifican un incremento sustantivo en la eficiencia estructural y en la confiabilidad de la gestión deportiva, consolidando la reducción de brechas temporales en la captura de datos y garantizando la disponibilidad inmediata de información fidedigna para la toma de decisiones, de modo que el sistema se posiciona como una alternativa tecnológica de alta viabilidad que no solo robustece los procesos administrativos subyacentes, sino que también propicia un entorno de competitividad deportiva rigurosamente estructurado y automatizado.

**ABSTRACT**

The present technological research is framed within the conceptual design and systemic implementation of a computing architecture oriented toward the comprehensive governance of sports tournaments, whose primary objective lies in the optimization of registration protocols, the systematic monitoring of the involved stakeholders, and the strategic administration of sports-related informational assets; considering that, at present, a considerable sector of academic institutions and sports organizations still execute such procedures through rudimentary manual schemes, a circumstance that inevitably results in recurring operational errors, the loss of critical data, and pronounced administrative latency that hinders the reliable authentication of athletes and referees, the proposed solution has therefore been established upon a client–server model, integrating Laravel as the backend development engine, React for user interface orchestration, and PostgreSQL as a high-integrity database management system.

The functional ecosystem of the application encompasses essential operational dimensions ranging from user enrollment and detailed roster management to the technical assignment of referees and the issuance of digital credentials through QR code technology for real-time analytical consultation of performance metrics; furthermore, in order to structure the technical deployment through high-performance iterative cycles, the SCRUM framework was adopted, thereby enabling progressive scrutiny of achieved milestones and continuous empirical validation of the platform, which consequently evidences a qualitative transformation in organizational architecture and operational security levels that substantially mitigate participant registration response times.

The findings derived from the implementation of this study attest to a substantial increase in structural efficiency and reliability within sports management, consolidating the reduction of temporal gaps in data capture while ensuring the immediate availability of trustworthy information for decision-making processes; thus, the system emerges as a highly viable technological alternative that not only strengthens underlying administrative procedures but also fosters a rigorously structured and automated environment for competitive sports development.

**KICHWA**

Kay investigación tecnológica rurayka yuyaymanta llukshishka shuk arquitectura informática ruraypi churarin, campeonatos deportivos pushakuyta tukuy laya alli kamachinkapak; kaypa hatun munayka kan inskripshun ruraykunata alli allichina, chaypi yanapak runakunata sistemáticopi rikurina, shinallatak deporte yachaykunata estrategiawan apana, imashina kunan punchakunapi achka yachana wasikuna shinallatak deporte tantanakuykunapash kay ruraykunata ñawpa maki rurayllawan rurashpa katinkuna, chaymantami pantaykuna kutin kutin rikurin, importante datokuna chinkarin, administrasionpi unayachiykuna mirarin, shinallatak atletas y árbitros runakunata chiqap riqsichiyka sasachakun; chayrayku propuesta sistema cliente-servidor modelo ukupi sinchiyachishka kani, Laravel backend motorwan, React interfaz de usuario ruraypak, shinallatak PostgreSQL base de datos wakichik alli sinchi sistema shinallatak confiable kan.

Aplicativopa funcional ecosistema ukupi tiyan ashka ruraykuna, usuarios enrolamientomanta kallarik, equipos y plantillas detalladota pushakuykama, árbitros técnico churaykuna, shinallatak credenciales digitales QR codigowan ruray, chaywan tiempo realpi métricas analíticakunata rikuna ushanka; shinallatak desarrollo técnico alli katinkapak iteracionespi, SCRUM marco de trabajo japishka kani, chaywanmi hitos rurashkakunata ñawpakman rikushpa alli tantiyachina, plataforma mana pantashpa katishpa constante validación rurashka, chaymantami organización arquitecturapi shinallatak seguridad operativapi hatun alli tukuy rikurin, participantes registro tiempo ashkata pisiyachishpa.

Kay estudio ruraymanta llukshishka hallazgokuna rikuchinmi eficiencia estructural shinallatak confiabilidad deporte gestión ukupi ashtawan mirashkata, datos japina tiempo uchillayachishpa shinallatak chiqap información chaylla rikurina ushankapak decisiones alli rurankapak; chayrayku kay sistema shuk solución tecnológica alli viable tukushpa rikurin, administración ruraykunata sinchiyachishpa, shinallatak deporte competitividadta ordenadomanta shinallatak automatizadomanta ñawpakman pushankapak yanapan.

# CAPÍTULO I

# FORMULACIÓN GENERAL DEL PROYECTO

## 1.1. Tema

"SISTEMA INTEGRAL DE GESTIÓN PARA CAMPEONATOS DEPORTIVOS EN LA UNIVERSIDAD ESTATAL DE BOLÍVAR."

## 1.2. Descripción del Problema

La Universidad Estatal de Bolívar organiza periódicamente campeonatos deportivos en disciplinas como fútbol y baloncesto, con el fin de fomentar la recreación y la integración estudiantil. La gestión actual de estos eventos se desarrolla mediante procesos manuales y el uso de herramientas básicas (Excel), lo que genera múltiples limitaciones que afectan la eficiencia, la transparencia y la confiabilidad en la organización.

El principal inconveniente se presenta en la inscripción de equipos y jugadores, que se realiza a través de planillas físicas o documentos dispersos en hojas de cálculo. Esta práctica incrementa los errores administrativos, dificultando la verificación de información y retrasa la programación de los encuentros. Como resultado, se producen confusiones en la comunicación de calendarios y afectaciones en la experiencia de los participantes.

La verificación de identidad es un gran problema en el deporte. Sin métodos seguros, cualquiera puede fingir ser otra persona. Esto afecta la transparencia y la honestidad en las competencias. Cada deporte se maneja por separado, no hay reglas claras para inscribir participantes, programar eventos y registrar resultados. Esto impide que las organizaciones deportivas amplíen su gestión a otras áreas.

Un problema clave es la falta de una plataforma digital organizada. No se pueden analizar datos en tiempo real ni generar estadísticas sobre el desempeño de los participantes. Esto limita la capacidad de los directivos para tomar decisiones informadas.

La seguridad en los deportes de alto nivel está en riesgo. Sin sistemas modernos de verificación de identidad, no se puede evitar que alguien se haga pasar por otra persona.

Las organizaciones deportivas necesitan una solución. Una plataforma digital segura podría mejorar la transparencia y la honestidad en las competencias. Esto haría que los deportes sean más justos y seguros para todos.

En este contexto, surge la necesidad de diseñar e implementar un sistema integral de gestión que centralice la administración de campeonatos deportivos, incorpore mecanismos de carnetización digital con códigos QR y unifique la programación, los

resultados y el control de múltiples disciplinas, garantizando eficiencia organizativa, seguridad en los procesos y mayor transparencia en el deporte universitario.

## 1.3. Justificación

La implementación de una arquitectura informática orientada a la gobernanza de campeonatos universitarios constituye un pilar estratégico para la modernización institucional, lo cual permitirá amalgamar de forma sistémica el empadronamiento de deportistas, la planificación cronológica de eventos y el procesamiento analítico de resultados con un alto índice de eficiencia operativa, facilitando consecuentemente la obtención de métricas estadísticas de alto valor que sirvan como fundamento para la toma de decisiones estratégicas en futuras planificaciones deportivas.

La plataforma ha sido proyectada bajo un esquema de seguridad integral que incorpora el uso de credenciales digitales mediante tecnología de códigos QR, dispositivos que garantizan la autenticación fehaciente de los competidores mientras promueven simultáneamente la estandarización de los protocolos administrativos, lo que deriva en una gestión institucional más robusta y en una estructura operativa lo suficientemente flexible como para administrar diversas disciplinas e incorporar nuevas modalidades conforme a las demandas emergentes.

En lo referente a la infraestructura tecnológica, se ha determinado el despliegue de un *stack* compuesto por el lenguaje PHP y el motor de base de datos PostgreSQL, configuración que provee un entorno escalable, íntegro y con facultades de acceso remoto, sustentando esta elección en el aprovechamiento de herramientas de código abierto con trayectorias consolidadas y un soporte comunitario extensivo que asegura la sostenibilidad técnica del sistema a largo plazo.

El propósito medular de esta intervención tecnológica radica en la transformación cualitativa de la gestión administrativa interna de la Universidad Estatal de Bolívar, pretendiéndose que este ecosistema informático se posicione como un paradigma funcional replicable para otras instituciones de educación superior que busquen formalizar sus procesos de competición.

Finalmente, el presente estudio se adscribe rigurosamente a la línea de investigación en Ingeniería de Software, Redes y Telecomunicaciones, aportando en este sentido una solución disruptiva y contextualmente pertinente cuya implementación genera un impacto multidimensional en las esferas académica, organizativa y sociodeportiva de la comunidad universitaria.

## 1.4. Objetivos: General y Específicos

### General

Desarrollar un Sistema Integral de Gestión para Campeonatos Deportivos en la Universidad Estatal de Bolívar, que centralice la organización de los torneos de fútbol, baloncesto y vóley, mejorando la eficiencia en la inscripción de equipos, la programación de encuentros y el control de identidad de los participantes.

### Específicos

- Analizar los procesos actuales de organización de campeonatos en la universidad para identificar requerimientos del sistema.
- Diseñar la arquitectura y módulos principales del sistema: inscripciones, carnetización con QR, programación y resultados.
- Implementar la plataforma con una interfaz accesible y adaptable a múltiples disciplinas deportivas.
- Validar el funcionamiento del sistema mediante pruebas con usuarios, evaluando su eficiencia y usabilidad.

**Pregunta general:**

¿Cómo mejorar la eficiencia y transparencia en la gestión de campeonatos deportivos en la Universidad Estatal de Bolívar mediante un sistema integral basado en tecnologías web?

**Preguntas específicas:**

1. ¿Qué limitaciones presentan los procesos actuales de inscripción, carnetización y programación de los torneos universitarios?
2. ¿Cómo diseñar una plataforma que permita la centralización y estandarización de las disciplinas deportivas en una sola herramienta?

3. ¿Qué impacto tendrá la implementación de carnetización digital con código QR y la generación de estadísticas en la gestión deportiva institucional?

# CAPÍTULO II

# MARCO TEÓRICO

## 2.1. Antecedentes

En los últimos años, la gestión deportiva ha sido objeto de múltiples estudios, tanto en el contexto internacional como nacional y local, los cuales coinciden en la importancia de incorporar herramientas tecnológicas que mejoren la organización, transparencia y accesibilidad en los campeonatos deportivos.

Lorenz (2007) desarrolló Sportacus, un sistema de información orientado a la organización y administración de campeonatos de deportes colectivosLa evidencia empírica derivada de este estudio determinó que la homogeneización digital de los cronogramas, las métricas de desempeño y los escalafones competitivos no solo robustece los niveles de visibilidad operativa, sino que simultáneamente cataliza la capacidad de transpolar la arquitectura del sistema hacia diversas modalidades atléticas.

En lo que respecta a la optimización de los certámenes intramurales en centros de educación superior, Andrés Morena (2009) postuló un marco de gestión informática bajo la premisa de que la migración de los registros de matriculación y los marcadores hacia entornos digitales mitiga drásticamente la latencia burocrática, logrando con ello un incremento sustantivo en la fiabilidad de las justas y un fortalecimiento de la dinámica participativa de los educandos.

Desde una perspectiva de movilidad tecnológica en el contexto ecuatoriano, Paredes Reyes (2021) conceptualizó un aplicativo móvil centrado en la difusión sincrónica de itinerarios, subrayando en sus hallazgos que el empleo de interfaces portátiles incrementa la auditabilidad de los procesos institucionales, consolidando de esta manera el vínculo de seguridad y confianza entre los atletas y los entes rectores.

Por su parte, el escrutinio técnico sobre las facultades avanzadas de los ecosistemas web realizado por Cortes Peñafiel (2024) priorizó la implementación de protocolos de validación mediante códigos QR, lo cual permitió corroborar que la identidad digital constituye un baluarte contra anomalías administrativas como la duplicidad de inscritos, evitando fehacientemente cualquier intento de suplantación de identidad en los cuadros competitivos.

Asimismo, la intervención tecnológica de Yépez Vendoval (2020) en las ligas cantonales de la localidad de Colta se centró en la notificación automatizada de calendarios futbolísticos, determinando que el flujo informativo inmediato neutraliza las colisiones

programáticas en la agenda deportiva, incidiendo directamente en una mejora cuantitativa de la puntualidad y la concurrencia en los escenarios de juego.

Finalmente, el diseño de infraestructuras para la coordinación de macro-eventos estudiantiles masivos fue la propuesta central de Cisneros Pinto (2023), quien proyectó un entorno digital orientado a la academia ecuatoriana con el objetivo de centralizar los flujos operativos, erigiéndose, así como una solución estratégica para la gobernanza de grandes volúmenes de datos en procesos de inscripción de alta demanda. Su metodología incremental permitió optimizar inscripciones y programación, concluyendo que la digitalización promueve mayor participación estudiantil y reduce la carga administrativa.

Los antecedentes revisados evidencian avances significativos en la digitalización de procesos deportivos, pero presentan limitaciones que justifican el presente proyecto:

- La mayoría de propuestas se enfocan en una sola disciplina deportiva (fútbol principalmente).
- No existe un modelo estandarizado que contemple un sistema para múltiples deportes en el que realizar un sistema para registrar: fútbol, baloncesto, voleibol… entre otros.
- Algunos autores realizan propuestas innovadoras (ej: QR), pero estas no han sido ejecutadas completamente en el contexto universitario.

Este proyecto pretende dar respuesta a la carencia de un Sistema Integral de Gestión para Campeonatos Deportivos en la Universidad Estatal de Bolívar, que contemple: inscripción digital, carnetización con QR, gestión de múltiples deportes y generación de informes estadísticos, ofreciendo de este modo una respuesta global y una solución tecnológica replicable a nivel institucional.

### 2.2. Científico

El desarrollo del *Sistema Integral de Gestión para Campeonatos Deportivos en la Universidad Estatal de Bolívar* se sustenta en bases teóricas y tecnológicas propias de la Ingeniería de Software y de las Ciencias de la Computación, que garantizan la calidad, escalabilidad y seguridad del producto final.

La Ingeniería de Software proporciona principios y modelos que orientan el ciclo de vida de los sistemas, desde la especificación de requisitos hasta la implementación y mantenimiento. Más allá de una simple elección técnica, el hecho de abordar el desarrollo desde un rigor sistemático responde a la necesidad de blindar la solución ante posibles inconsistencias operativas, logrando de este modo que la fase de construcción no solo sea eficiente, sino que culmine en una herramienta que se ajuste, sin fisuras, a lo que los usuarios finales realmente demandan.

Bajo esta misma lógica, la integración del marco de trabajo Scrum no es una decisión arbitraria, sino que se fundamenta en la flexibilidad que ofrecen sus ciclos de entrega incrementales, puesto que, al fragmentar el avance en hitos funcionales, se habilita una validación constante con los organizadores de los torneos, lo que termina por garantizar un producto que no solo es adaptable, sino que prioriza la experiencia de uso sobre la complejidad técnica innecesaria.

En lo que respecta a la capa de persistencia informativa, la arquitectura se sustenta en modelos relacionales por su probada solvencia en la gestión de integridad de datos, aunque la hoja de ruta del proyecto contempla la adopción de PostgreSQL para potenciar la robustez técnica del ecosistema, permitiendo que el procesamiento de métricas analíticas y la generación de reportes dinámicos se realicen bajo estándares de alta disponibilidad y en tiempo real.

En lo que respecta a la experiencia de usuario, se ha priorizado el despliegue de interfaces con adaptabilidad web integral (responsive design), una decisión técnica que busca, por encima de todo, eliminar cualquier tipo de fricción en el acceso desde terminales móviles o estaciones de escritorio. Al democratizar la entrada al sistema, se consigue que la comunidad universitaria se involucre de forma proactiva, suprimiendo esas barreras tecnológicas que, históricamente, han terminado por mermar el alcance real de las plataformas de gestión institucional.

Bajo la premisa de robustecer la integridad informacional del certamen, se ha determinado la operacionalización de un esquema de validación mediante códigos QR integrados en las credenciales de los participantes, configuración técnica que trasciende el simple registro para consolidarse como una estrategia de blindaje orientada a la protección de los perfiles atléticos.

Este esquema operativo, que fusiona la agilidad de uso con una validación de identidad inmediata, permite erradicar de forma contundente y desde su origen cualquier brecha de seguridad relacionada con la redundancia de inscripciones o, peor aún, con la suplantación de perfiles en las actas oficiales de los encuentros.

Como cierre del ecosistema, la capacidad analítica de la plataforma descansa sobre técnicas de minería de datos y estadística avanzada; se trata de herramientas orientadas a

desentrañar el volumen masivo de información generado en los certámenes para transmutarlo en activos de conocimiento estratégico. Gracias a este procesamiento, se obtienen métricas de desempeño e indicadores de participación que resultan, a todas luces, piezas angulares para proyectar y optimizar los calendarios deportivos en los venideros ciclos universitarios.

Estos aportes fortalecen la transparencia y la trazabilidad de los eventos deportivos.

En síntesis, el marco científico del proyecto combina metodologías de desarrollo de software, modelos de gestión de datos, tecnologías web y mecanismos de seguridad digital, con el propósito de entregar una herramienta confiable, adaptable y alineada a las necesidades institucionales.

## 2.3. Conceptual

El desarrollo del Sistema Integral de Gestión para Campeonatos Deportivos en la Universidad Estatal de Bolívar se fundamenta en una serie de conceptos clave provenientes de la Ingeniería de Software y de la gestión tecnológica aplicada al deporte universitario, los cuales garantizan la construcción de una plataforma confiable, escalable y segura.

El Sistema Integral de Gestión se define como un conjunto de módulos de software que centralizan distintos procesos administrativos y deportivos en una sola plataforma digital. En el presente proyecto, esta herramienta permitirá organizar la inscripción de equipos y jugadores, la programación de partidos, el control de identidad de los participantes y la generación de reportes estadísticos en tiempo real (Pressman & Maxim, 2021).

La Ingeniería de Software es la disciplina que aplica principios científicos, metodologías y buenas prácticas para el diseño, construcción y mantenimiento de sistemas informáticos de calidad. Su implementación asegura que el sistema sea confiable, seguro y adaptable a diferentes escenarios, cumpliendo con los estándares técnicos requeridos en la gestión deportiva (Sommerville, 2016).

Las Metodologías Ágiles, particularmente Scrum, permiten un desarrollo iterativo e incremental del software, promoviendo la colaboración con los usuarios, la entrega temprana de valor y la adaptabilidad a cambios en los requerimientos. Esta metodología resulta adecuada para el proyecto, ya que posibilita la retroalimentación constante de los organizadores deportivos y facilita la validación continua de los módulos implementados (Schwaber & Sutherland, 2020).

La realización de certámenes atléticos en el ámbito universitario se concibe como una praxis sistémica y estructurada en la cual el estudiantado converge bajo normativas rigurosas y disciplinas heterogéneas; en tal sentido, la administración de dichas competencias demanda la instauración de protocolos normalizados orientados a mitigar el error humano, salvaguardar la transparencia y robustecer la arquitectura organizativa de la institución (Yépez Vendoval, 2020).

El despliegue de la identificación electrónica, o carnetización digital, se sustenta en la arquitectura de credenciales virtuales diseñadas para la validación fidedigna de los participantes, de tal suerte que, en el presente estudio, dicha tecnología se operacionalizará mediante códigos QR para facilitar verificaciones ágiles y seguras a través de terminales móviles, mitigando sustancialmente las vulnerabilidades relativas a la suplantación de identidad o la redundancia de registros (Paredes Reyes, 2021; Cortes Peñafiel, 2024).

La tecnología Quick Response, o Código QR, se define técnicamente como una matriz bidimensional optimizada para el almacenamiento de metadatos legibles por dispositivos ópticos especializados; en virtud de aquello, su amalgama con el sistema de acreditación digital asegura una sinergia entre eficiencia operativa y seguridad en el tratamiento de los activos informativos (Lorenz, 2007).

La estandarización administrativa conlleva la adopción de criterios normativos transversales que aseguren la coherencia operativa durante la gobernanza de los torneos, independientemente de la naturaleza del deporte, puesto que este enfoque posibilita la transpolación de parámetros técnicos idénticos a disciplinas como el fútbol, el baloncesto o el voleibol (Andrés Morena, 2009).

En lo que atañe a la arquitectura de persistencia, el aprovechamiento de esquemas relacionales permite estructurar los activos informativos mediante entidades vinculadas por dependencias lógicas estrictas; esta configuración salvaguarda de manera fehaciente tanto la exactitud referencial como la cohesión granular de los datos, consolidándose así como el eje medular para la gobernanza operativa de los registros de deportistas, cronogramas y métricas de desempeño (Date, 2004).

La usabilidad se postula como la métrica que determina el grado de eficacia y satisfacción con el que los actores interactúan con la interfaz digital; en consecuencia, se busca que la solución resulte intuitiva y plenamente accesible tanto para los estamentos directivos como para la comunidad estudiantil, optimizando así la curva de aprendizaje de la herramienta (Nielsen, 2012).

Finalmente, la escalabilidad se entiende como la capacidad adaptativa del sistema para gestionar un incremento exponencial en el flujo de datos o funcionalidades sin degradar su rendimiento computacional, lo cual permite la integración orgánica de nuevas disciplinas deportivas o la replicabilidad del modelo en diversas instituciones sin comprometer la arquitectura base del software (Bass, Clements & Kazman, 2013).
En conjunto, estos conceptos proporcionan el soporte conceptual y técnico necesario para construir una plataforma sólida, confiable y alineada con los principios de la Ingeniería de Software, asegurando pertinencia académica y funcionalidad en la gestión deportiva universitaria.

## 2.4. Legal

El desarrollo del *Sistema Integral de Gestión para Campeonatos Deportivos en la Universidad Estatal de Bolívar* se enmarca en un conjunto de disposiciones normativas que respaldan tanto la promoción del deporte como el uso de tecnologías de información en contextos educativos. A continuación, se detallan las principales bases jurídicas:

- **Constitución de la Republica del Ecuador (2008):**

  El artículo 381 reconoce al deporte, la educación física y la recreación como derechos de la ciudadanía y como actividades que contribuyen al desarrollo integral de las personas.

  El artículo 383 establece que el Estado fomentará la educación física y el deporte formativo, recreativo y de alto rendimiento, garantizando recursos y condiciones para su práctica. Estos principios fundamentan la necesidad de implementar herramientas que fortalezcan la organización deportiva en instituciones de educación superior.

- **Ley del Deporte, Educación Física y Recreación (Registro Oficial Suplemento 255, 2010):**

  Esta normativa regula la planificación, fomento y control del deporte en el país. En su articulado se promueve la organización de eventos deportivos en instituciones educativas, así como la utilización de mecanismos que garanticen transparencia y

eficiencia en la gestión de campeonatos. El sistema se alinea con estos objetivos al estandarizar procesos de inscripción, control de jugadores y registro de resultados.

- **Ley Orgánica de Educación Superior – LOES (Registro Oficial 298, 2010).** La arquitectura funcional del aplicativo propuesto guarda una estrecha consonancia con los fines institucionales, en la medida en que permite la normalización integral de los flujos de matriculación, la fiscalización rigurosa de los atletas y el procesamiento sistemático de los marcadores, vinculándose de manera directa con las disposiciones de la Ley Orgánica de Educación Superior – LOES (Registro Oficial 298, 2010).

  Bajo los preceptos de la mencionada Ley Orgánica, se instituye el imperativo de que las entidades de educación superior aseguren una instrucción de carácter holístico, lo que forzosamente abarca el espectro de las dinámicas extracurriculares, tales como la práctica deportiva, toda vez que el artículo 107 promueve el ejercicio físico como un catalizador esencial para el desarrollo de aptitudes recreativas y la cohesión social, de modo que la operacionalización de un entorno digital para la coordinación de torneos actúa como un motor estratégico que impulsa el cumplimiento efectivo de dicho mandato institucional.

- **Ley Orgánica de Protección de Datos Personales (Registro Oficial Suplemento 459, 2021):**

  Esta ley regula el tratamiento legítimo, seguro y controlado de la información personal de los ciudadanos. Dado que el sistema administrará datos sensibles de los jugadores y dirigentes, la normativa obliga a aplicar medidas de seguridad digital, especialmente en la carnetización digital mediante códigos QR, asegurando confidencialidad, integridad y disponibilidad de la información.

- **Reglamentos internos de la Universidad Estatal de Bolívar.** La institución establece normativas y lineamientos internos para la organización de actividades deportivas. El sistema propuesto se ajustará a estas disposiciones, facilitando la gestión administrativa de campeonatos y promoviendo la modernización de sus procesos a través de tecnologías digitales.

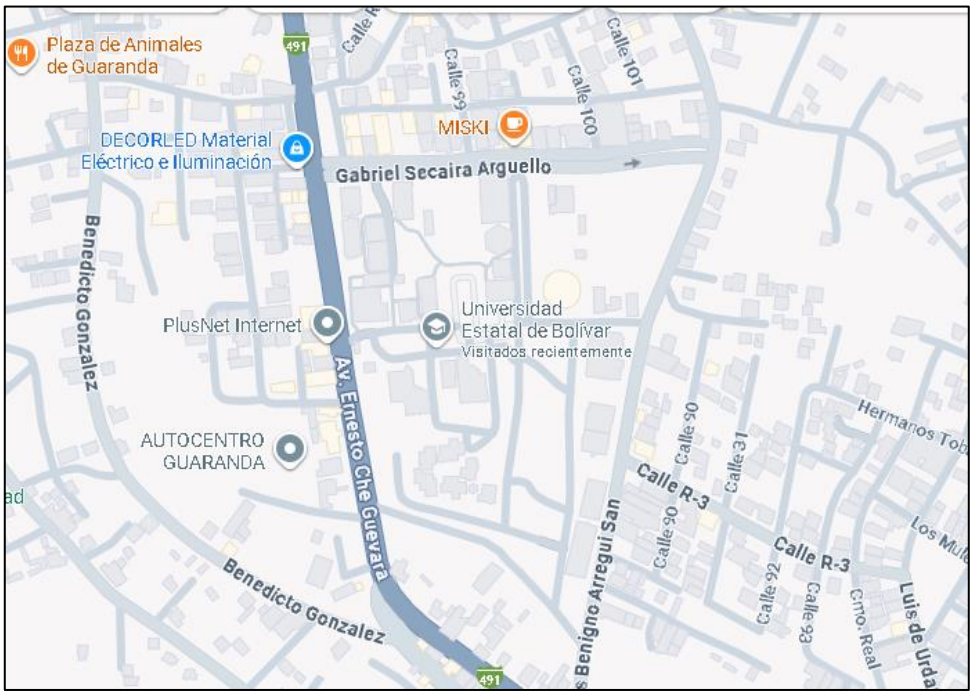
## 2.5. Georreferencial

Emplazada en la urbe de Guaranda, dentro de la provincia de Bolívar y en el corazón de la región interandina del Ecuador, la Universidad Estatal de Bolívar (UEB) constituye un baluarte del conocimiento y el desarrollo social, donde su consolidada trayectoria

académica permite albergar a un ecosistema estudiantil heterogéneo que converge frecuentemente en dinámicas deportivas extracurriculares.

Es precisamente bajo este marco de acción donde la entidad articula la ejecución sistemática de torneos competitivos en ramas como el futbol, el baloncesto y el voleibol, logrando aglutinar a un segmento mayoritario de la población estudiantil; en consecuencia, la puesta en marcha de una plataforma de gestión integral no solo se justifica por esta demanda administrativa y geográfica, sino que además proyecta una arquitectura técnica cuya transpolación resulta viable hacia otros centros de educación superior que compartan rasgos institucionales similares.

**Figura 1.**Ubicación geográfica de la Universidad Estatal de Bolívar



# CAPÍTULO III
# METODOLOGÍA

## 3.1. Metodología de Desarrollo de Software

El desarrollo del Sistema Integral de Gestión para Campeonatos Deportivos en la Universidad Estatal de Bolívar se fundamentará en la metodología SCRUM, una metodología ágil ampliamente utilizada en proyectos de ingeniería de software que requieren entregas rápidas, validación continua y alta adaptabilidad a los cambios. Su aplicación permite dividir el proyecto en fases iterativas e incrementales denominadas *sprints*, cada uno con una duración estimada de 2 a 3 semanas, lo que facilita la retroalimentación constante de los usuarios finales, en este caso los estudiantes de la institución.

La adopción de SCRUM responde a la necesidad de construir un sistema funcional desde las primeras etapas, evitando el retraso de resultados hasta fases finales como ocurre con metodologías tradicionales. Gracias a su naturaleza colaborativa, se asegura que los requerimientos detectados durante el levantamiento de información se traduzcan rápidamente en funcionalidades prácticas y verificables.

**FASES DEL DESARROLLO CON SCRUM**

**Sprint 1: Inmersión operativa y diagnóstico de necesidades en campo:**

La fase de arranque se alejó de cualquier formalismo de oficina para centrarse en una inmersión directa en las canchas, porque la verdad es que nos dimos cuenta y bastante rápido de que los manuales técnicos no sirven de nada si uno no entiende el calvario logístico que viven los estudiantes a diario. Fue justamente ese contacto con el "barro" lo que nos permitió ver lo que de verdad importaba (inscripciones digitales y validación por QR), permitiéndonos armar un Product Backlog que no era una simple lista de deseos, sino un plan de choque para jubilar el papel y salvar la integridad de los datos de una vez por todas.

**Sprint 2: Arquitectura y modelado bajo presión**

Una vez mapeadas las necesidades, el equipo se enfocó en traducir ese desorden en un esquema lógico capaz de aguantar el uso rudo, lo que nos obligó a "quemar pestañas" diseñando diagramas de clases y modelos de entidad-relación para no dejar cabos sueltos.

No íbamos a tirar código a ciegas, así que montamos maquetas bajo el patrón MVC; esta fue, sin duda, la jugada clave para separar la lógica de la interfaz y evitar que el software se volviera un nudo indescifrable, facilitando que, si mañana otro programador le mete mano al sistema, no se sienta perdido en un laberinto.

**Sprint 3: Construcción del backend y blindaje de datos**

Este ciclo fue de puro "picar piedra" en el servidor, levantando el entorno en PHP y PostgreSQL para soportar el tráfico pesado de los códigos QR, una combinación que elegimos a conciencia por su estabilidad y porque no queríamos sorpresas a mitad de camino. Pero ojo: no todo fue escribir funciones; le dedicamos un esfuerzo enorme a sellar grietas de seguridad en las sesiones y perfiles, porque nuestra meta era que la trazabilidad fuera parte del ADN del sistema y no un simple "parche" puesto a última hora para salir del paso.

**Sprint 4: Lógica de negocio y validación en "fuego real"**

En esta fase, nos centramos en hacer que la plataforma sea más sólida, por eso, diseñamos un motor especial que nos ayuda a encontrar información y que nos muestra cifras y datos mientras todo sucede. Después, revisamos cada parte de la plataforma para asegurarnos de que no haya problemas graves. Sin embargo, el examen de verdad no ocurrió en el servidor, sino en las manos de los estudiantes en plena cancha; ese feedback "bajo el sol", con las quejas y aciertos de los usuarios reales, fue el que nos dio la pauta definitiva para pulir la estabilidad justo antes de apretar el botón del despliegue final.

**Sprint 5: Despliegue y transferencia al Dominio y Servidor**

El hito de cierre se cumplió al mudar todo el ecosistema al servidor de producción bajo el dominio, verificando que cada engranaje digital girara sin esos tropezones que suelen arruinar los lanzamientos. Más allá de subir archivos, nos aseguramos de entregar manuales operativos y documentación técnica bien desmenuzada; queríamos que la administración tomara el timón de forma autónoma, garantizando que el deporte fluya sin que la burocracia técnica vuelva a ser un muro insalvable.

Por qué el binomio SCRUM / PHP fue nuestra salvación

Elegir Scrum fue, básicamente, una táctica de supervivencia operativa. ¿Por qué? Porque su ritmo de entregas cortas nos permitió reaccionar a tiempo cuando aparecían

imprevistos, evitando que el cronograma se volviera papel mojado. Esa validación constante con el usuario real impidió que el software terminara siendo una "caja negra" inentendible para quienes de verdad lo usan.

Por otra parte, apostar por PHP y bases de datos relacionales fue puro pragmatismo: aprovechamos la infraestructura que la Universidad Estatal de Bolívar ya tiene operativa. No tenía sentido quemar presupuesto en servidores exóticos cuando podíamos montar algo robusto, seguro y, sobre todo, escalable sobre una tecnología que conocemos de sobra. Es una solución que permite crecer sin tener que demoler los cimientos cada seis meses.

# CAPÍTULO IV

# INGENIERÍA DEL PROYECTO

El presente capítulo detalla el proceso técnico del desarrollo del Sistema Integral de Gestión para Campeonatos Deportivos de la Universidad Estatal de Bolívar, aplicando el marco de trabajo ágil SCRUM y fundamentándose en la arquitectura, funcionalidades y estructura real del sistema y la base de datos proporcionada

## 4.1 Marco de Trabajo: SCRUM

El proyecto se desarrolló utilizando el marco ágil SCRUM debido a su flexibilidad, su enfoque iterativo e incremental y su capacidad para adaptarse a requerimientos cambiantes mientras se mantiene una comunicación constante con los usuarios finales.

SCRUM permitió dividir el desarrollo en sprints cortos, obteniendo al final de cada ciclo un incremento funcional del sistema. Esto facilitó la retroalimentación continua de administradores deportivos, árbitros y estudiantes.

### Roles definidos

- **Product Owner:** Responsable de priorizar funcionalidades y representar los requerimientos institucionales.
- **Scrum Master:** Facilitador del proceso, encargado de eliminar impedimentos y asegurar el cumplimiento de la metodología.
- **Equipo de Desarrollo:** Los autores del proyecto, encargados de diseñar, programar, documentar y probar el sistema.

SCRUM garantizó un desarrollo organizado, transparente y orientado a resultados funcionales desde etapas tempranas.

## 4.2 Sprint 0: Identificación de Stakeholders y Aplicación de Instrumentos

En esta etapa se identificaron los actores clave del sistema:

### Stakeholders principales

- Administradores/organizadores de torneos
- Árbitros

- Estudiantes-jugadores

**Actividades realizadas**

1. **Aplicación de encuestas a jugadores** para identificar problemas en inscripción, programación y consulta de resultados.
2. **Entrevistas a administradores** para comprender procesos internos.
3. **Observación directa de torneos previos**, detectando fallos en control de identidad y registro manual de datos.

**Producto generado**

- Documento preliminar de requerimientos.
- Mapa de procesos actuales.
- Identificación de problemas críticos: duplicidad de jugadores, pérdida de información, falta de estadísticas en tiempo real.

### **4.3 Sprint 1:** Requerimientos Funcionales y No Funcionales

Durante el Sprint 1 se definieron los requerimientos del sistema a partir del análisis de necesidades identificado en la fase de levantamiento de información. Estos requerimientos establecen las funcionalidades que debe cumplir la plataforma y las condiciones de calidad necesarias para garantizar su correcto funcionamiento dentro del entorno universitario.

#### 4.3.1 Requerimientos Funcionales

Los requerimientos funcionales describen las acciones y servicios que el sistema debe proporcionar para satisfacer las necesidades de los usuarios involucrados en la gestión de campeonatos deportivos.

**Tabla 1.** *Requerimientos funcionales del sistema*

| Código | Nombre del Requerimiento | Descripción | Actor | Prioridad |
|---|---|---|---|---|
| **RF-01** | Registro de usuarios | El sistema permitirá registrar usuarios según rol institucional. | Administrador | Alta |

| | | | | |
|---|---|---|---|---|
| **RF-02** | Inicio de sesión | Validar credenciales para acceso seguro al sistema. | Todos | Alta |
| **RF-03** | Gestión de torneos | Crear, editar y eliminar torneos deportivos. | Administrador | Alta |
| **RF-04** | Gestión de equipos | Administrar equipos participantes. | Administrador | Alta |
| **RF-05** | Registro de jugadores | Asociar jugadores a equipos registrados. | Administrador | Alta |
| **RF-06** | Carnetización QR | Generar carnet digital con código QR único. | Sistema | Alta |
| **RF-07** | Validación QR | Verificar identidad del jugador antes del partido. | Árbitro | Alta |
| **RF-08** | Programar partidos | Definir fechas y enfrentamientos deportivos. | Administrador | Media |
| **RF-09** | Registrar resultados | Ingresar marcadores oficiales. | Árbitro | Alta |
| **RF-10** | Consultar resultados | Visualizar resultados y posiciones. | Jugador | Media |
| **RF-11** | Generar estadísticas | Calcular rendimiento automáticamente. | Sistema | Media |
| **RF-12** | Reportes | Exportar información en PDF y Excel. | Administrador | Baja |

### 4.3.2 Requerimientos No funcionales

**Tabla 2** *Requerimientos No funcionales del sistema*

| Código | Categoría | Requerimiento | Métrica / Criterio |
|---|---|---|---|
| **RNF-01** | Usabilidad | Interfaz intuitiva | Usuarios comprenden uso sin capacitación previa |

| | | | |
|---|---|---|---|
| **RNF-02** | Rendimiento | Tiempo de respuesta | Menor a 3 segundos |
| **RNF-03** | Seguridad | Cifrado de contraseñas | Uso de bcrypt |
| **RNF-04** | Seguridad | Control por roles | Acceso restringido según permisos |
| **RNF-05** | Disponibilidad | Acceso al sistema | 95% de disponibilidad anual |
| **RNF-06** | Escalabilidad | Integración futura | Permitir nuevos deportes |
| **RNF-07** | Concurrencia | Usuarios simultáneos | Soportar ≥100 usuarios |
| **RNF-08** | Compatibilidad | Navegadores | Chrome, Edge y Firefox |
| **RNF-09** | Mantenibilidad | Arquitectura | Implementación MVC |
| **RNF-10** | Respaldo | Backup automático | Copia diaria de base de datos |

### 4.3.3 Historias de Usuario

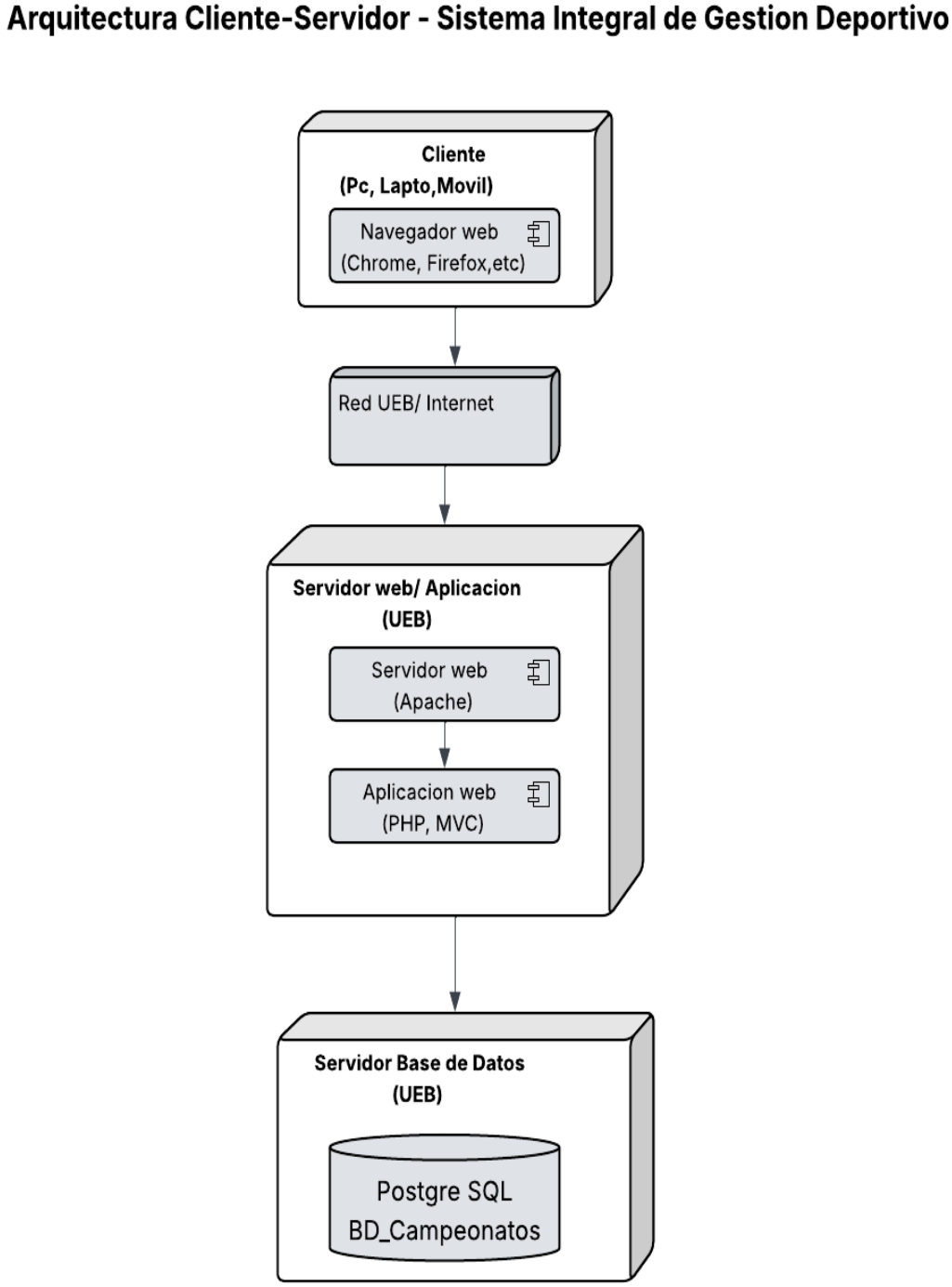
1. Como administrador, deseo registrar un torneo para gestionar su programación y resultados.
2. Como árbitro, deseo validar con QR la identidad del jugador para evitar suplantaciones.
3. Como jugador, deseo consultar los horarios de mis partidos de forma rápida.
4. Como organizador, deseo generar estadísticas automáticas para evaluar el rendimiento.

## 4.4 Sprint 2: Diseño Arquitectónico del Sistema

### 4.4.1 Vista arquitectónica general

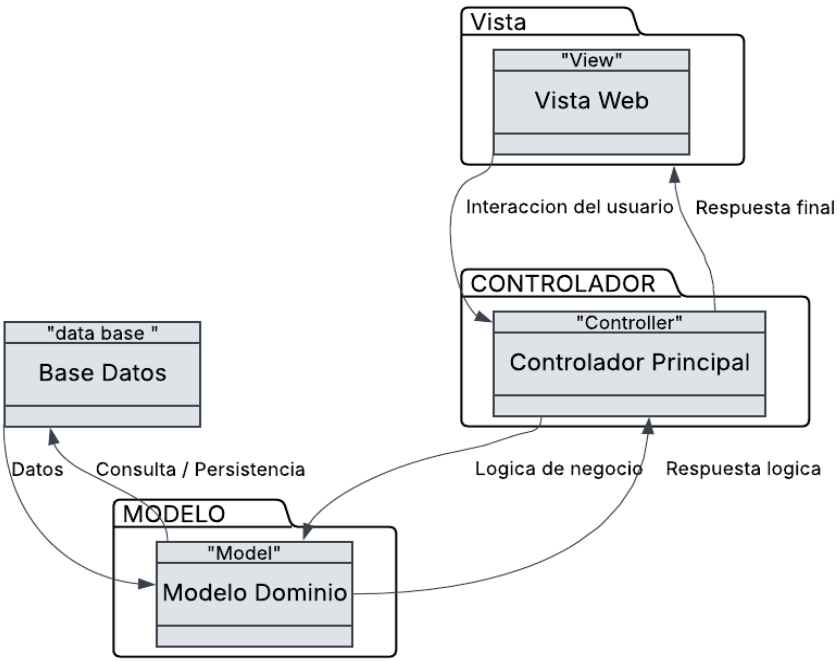
- **Diagrama de arquitectura cliente-servidor**

***Figura 2.*** *Diagrama de arquitectura cliente–servidor*



- **Diagrama MVC general**

**Figura 3.** *Diagrama MVC general del sistema*




### 4.4.2 Componentes principales

**Modelo**

Corresponde a tablas reales como:

- Usuarios, personas, jugadores, equipos, torneos, partidos, estadísticas, inscripciones

Cada tabla está relacionada mediante llaves foráneas, garantizando integridad referencial.

Ejemplo:

- *jugadores.cedula → usuarios.cedula*
- *equipos.torneo_id → torneos.id*

### 4.4.3 Patrón arquitectónico

**Diagrama MVC detallado**

**Figura 4.** *Diagrama MVC detallado*

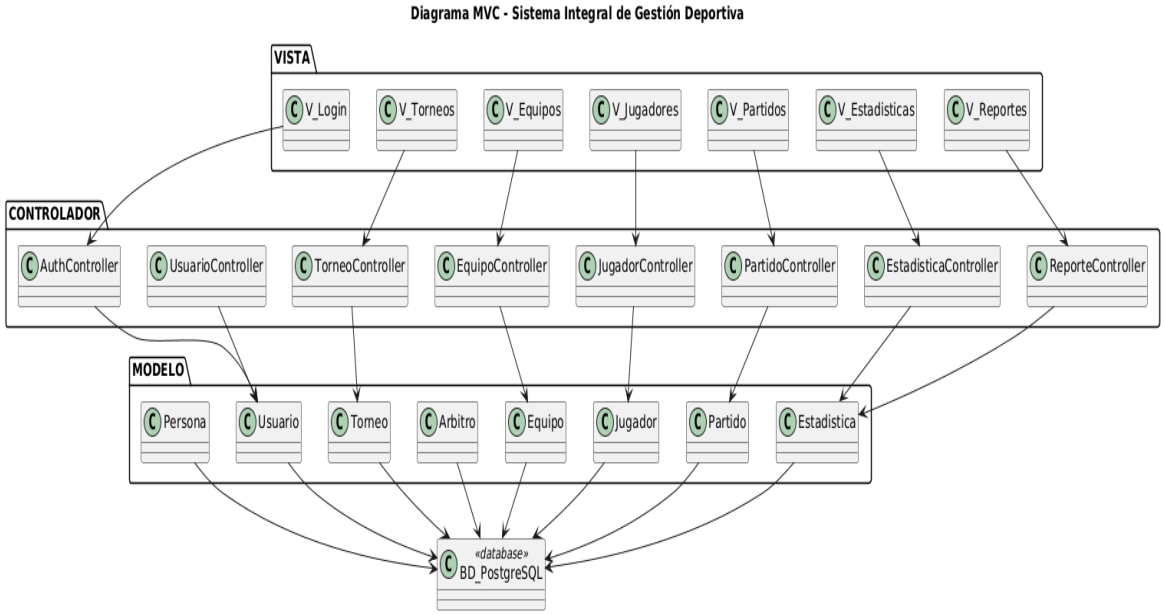
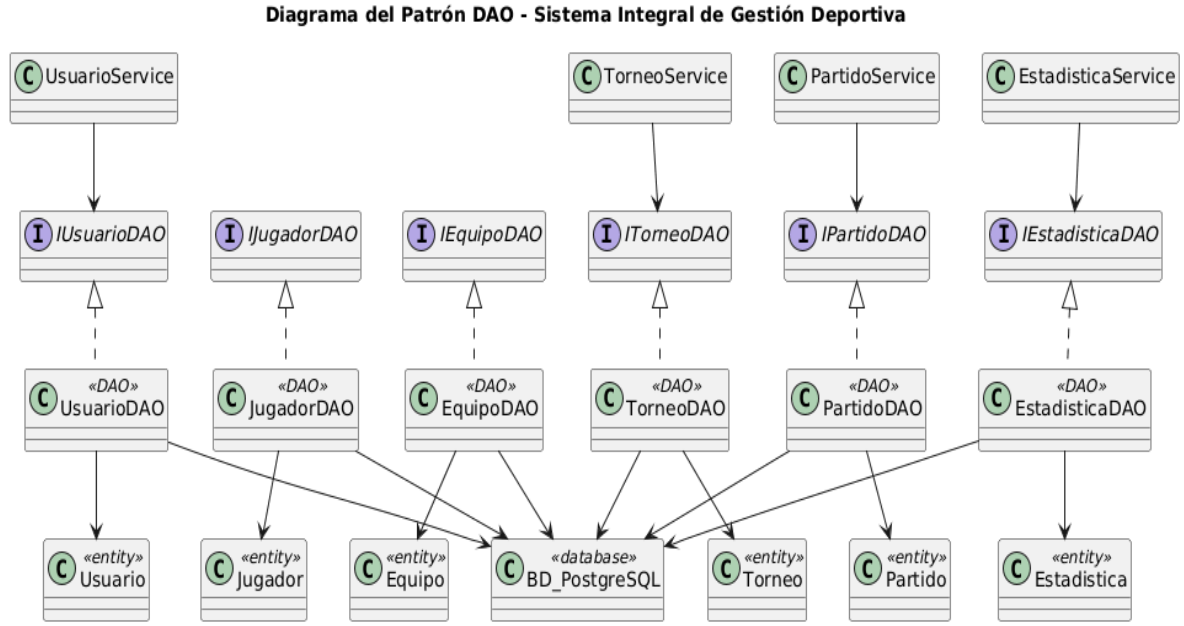



**Diagrama del patrón DAO**

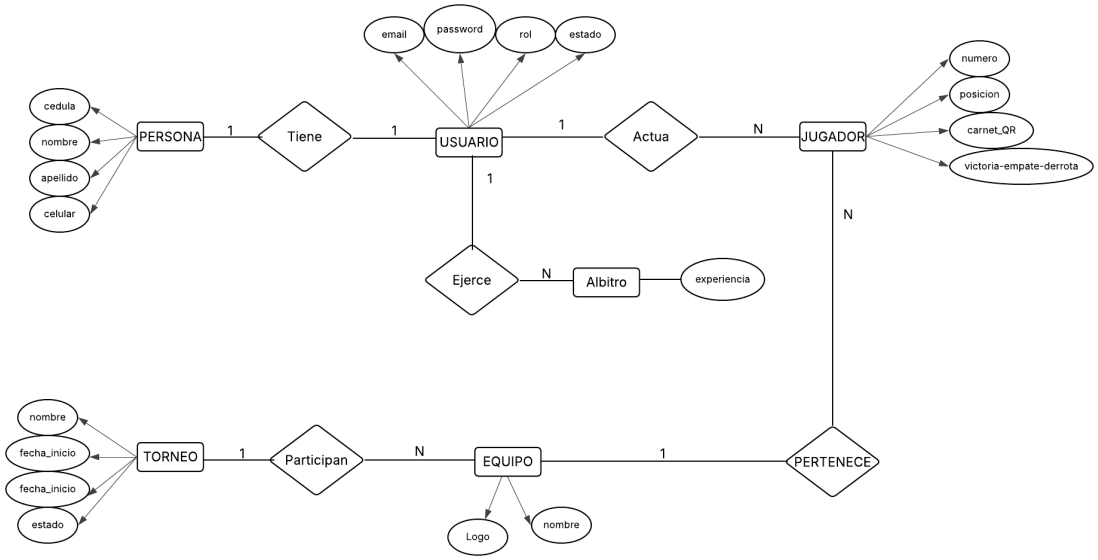
**Figura 5**. *Diagrama del Patrón DAO*




### 4.4.4 Diseño de Base de Datos

**Diagrama Entidad-Relación (ERD)**

**Figura 6.***Diagrama entidad–relación del sistema*

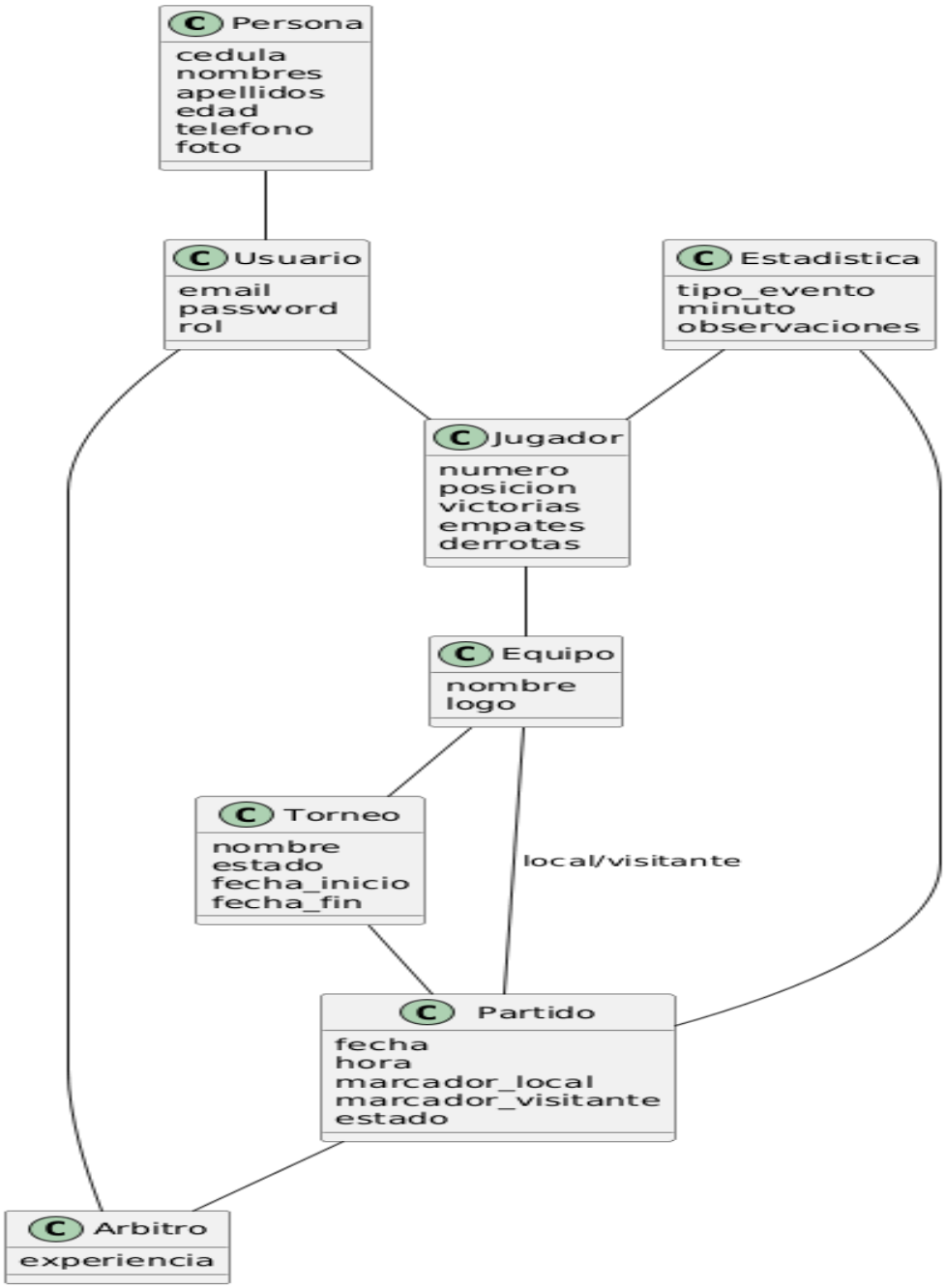



El diagrama entidad–relación representa la estructura conceptual de la base de datos del Sistema Integral de Gestión de Campeonatos Deportivos. En él se identifican las principales entidades del sistema, tales como persona, usuario, jugador, equipo, torneo y árbitro, así como sus atributos y relaciones correspondientes. Este modelo permite visualizar la organización de la información y la interacción entre los diferentes componentes que intervienen en la gestión de los campeonatos, constituyendo la base para el diseño lógico y físico de la base de datos.

### 4.4.5 Modelo Arquitectónico 4+1 de Kruchten

1. **Vista Lógica**

***Figura 7.*** *Vista lógica del modelo 4+1*



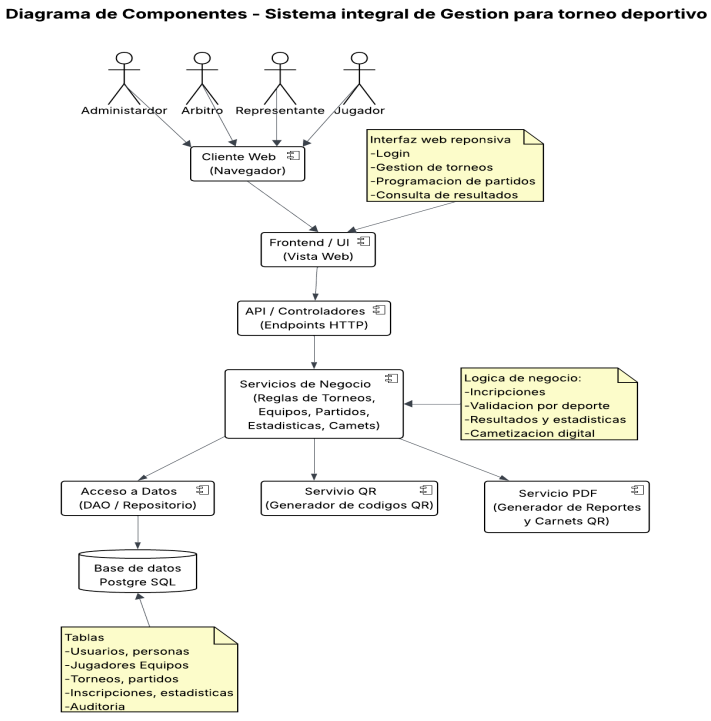
## 2. Vista de Procesos

***Figura 8.*** *Vista de procesos del modelo 4+1*



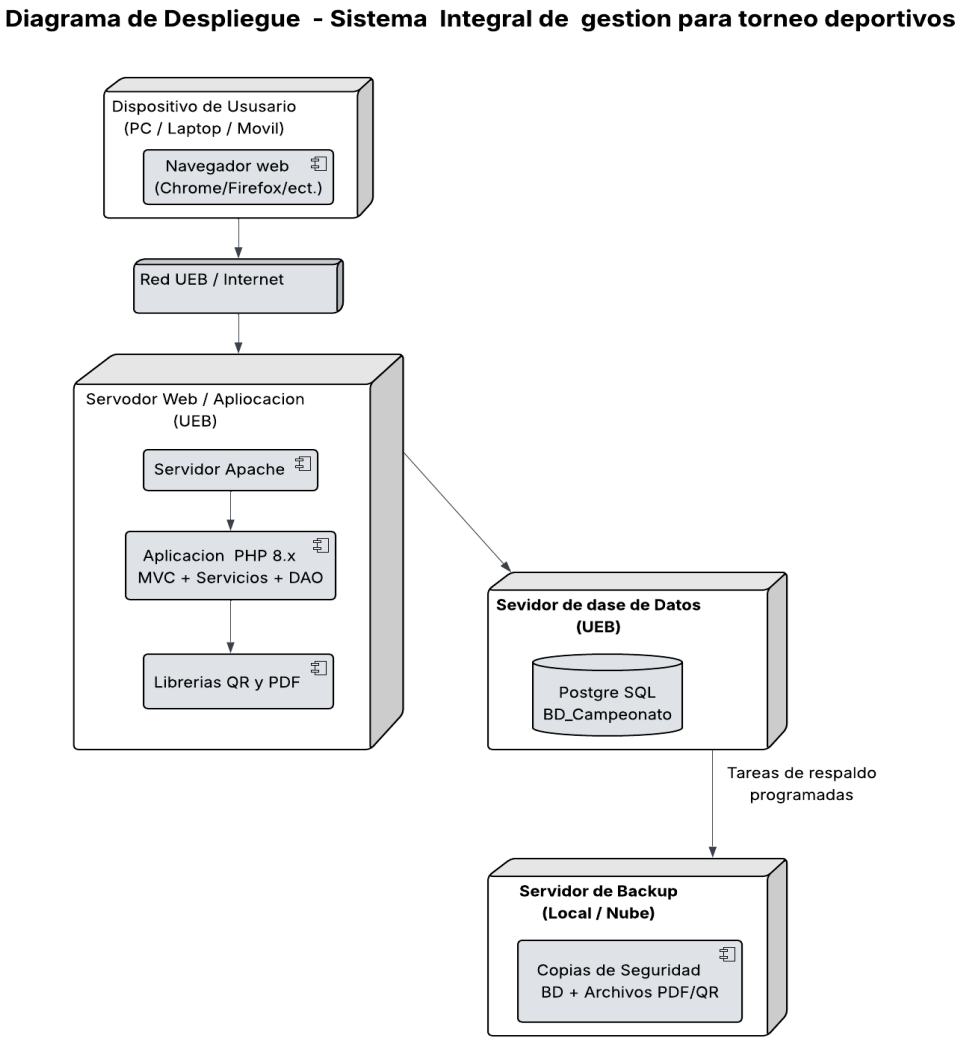

## 3. Vista de Desarrollo

**Figura 9.** *Vista de desarrollo del modelo 4+1*



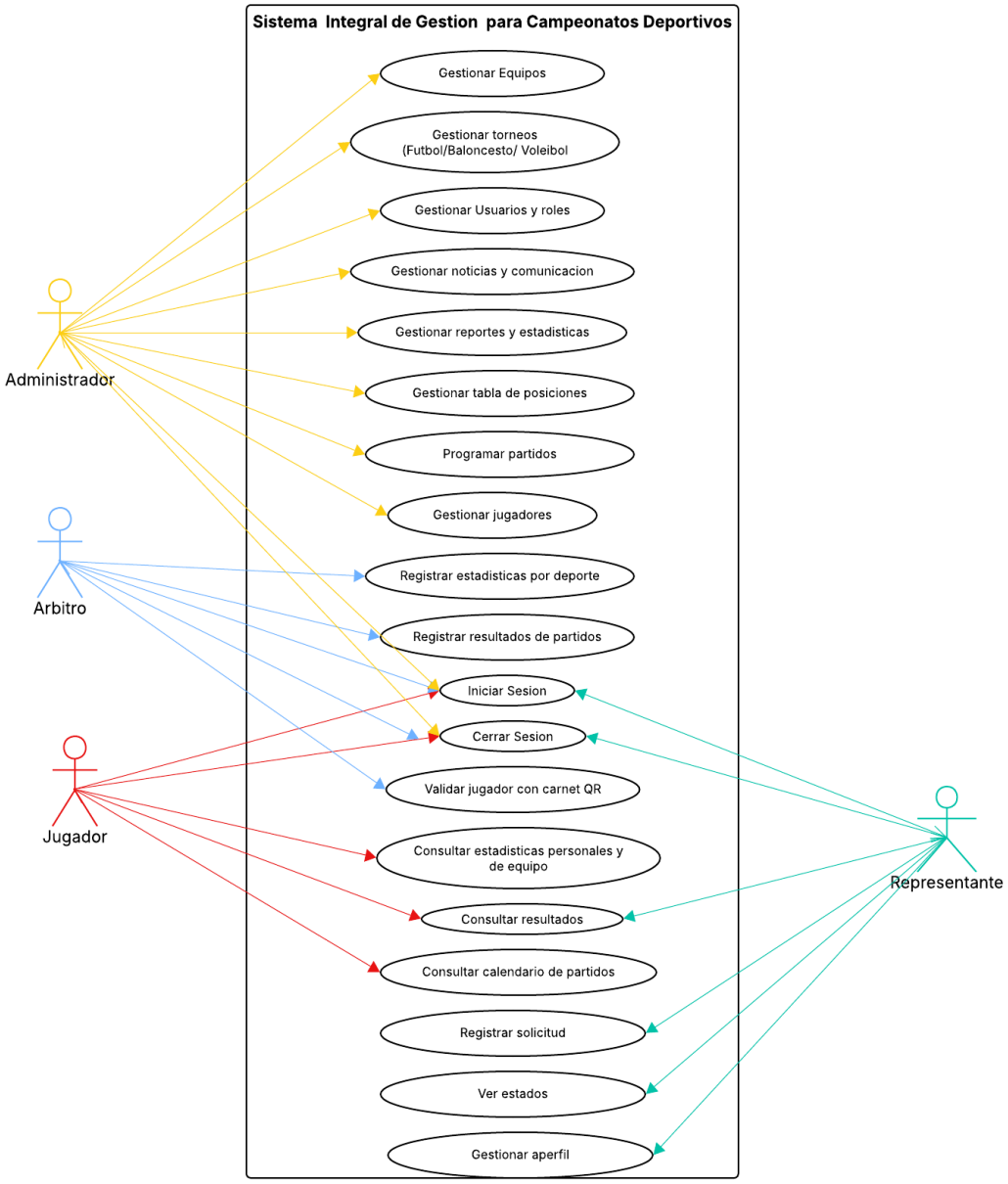
## 4. Vista Física → diagrama de despliegue

**Figura 10.** *Vista física (diagrama de despliegue)*




## 5. Vista de Escenarios (Use Cases)

**Figura 11.** *Vista de escenarios (diagrama de casos de uso)*




Más que un plano tradicional, este esquema es la brújula que define cómo se mueven los usuarios dentro del motor de gestión deportiva. No nos interesaba tener un dibujo técnico solo por rellenar el documento; la meta era entender, paso a paso, cómo iba a funcionar la plataforma en la práctica. Nos pusimos realmente estrictos al separar qué puede hacer cada quien, desde el administrador y el árbitro hasta los jugadores o representantes, blindando cada acceso para que las funciones no se crucen jamás. Al final, lo que buscábamos era que cada perfil tuviera su terreno bien marcado; en este sistema, la seguridad y la integridad de los datos son temas que simplemente no se negocian.

Lejos de ser un adorno, este modelo nos permitió prever el comportamiento del sistema en el "mundo real", abarcando desde la logística de los torneos hasta el registro de puntos sin que el flujo de datos se rompa. Pero el verdadero valor del diseño fue integrar visualmente el escaneo de códigos QR y la consulta estadística en vivo; tener esa claridad funcional fue, a fin de cuentas, lo que nos dio luz verde para dejar de teorizar y lanzarnos de cabeza a la programación pura del software.

**Descripción de Casos de Uso del Sistema**

**CU-01: Gestionar Equipos**

**Tabla 3.** *Descripción del caso de uso: Gestionar Equipos*

| Elemento | Descripción |
|---|---|
| **Nombre** | Gestionar Equipos |
| **Actor Principal** | Administrador |
| **Descripción** | Permite registrar, actualizar, consultar y eliminar equipos participantes en los campeonatos deportivos. |
| **Precondición** | El administrador debe haber iniciado sesión en el sistema. |
| **Postcondición** | La información del equipo queda almacenada o actualizada en la base de datos. |
| **Flujo Principal** | 1. El administrador accede al módulo equipos.<br>2. Ingresa o modifica datos del equipo.<br>3. El sistema valida la información.<br>4. Se guarda el registro correctamente. |
| **Flujo Alternativo** | Si los datos están incompletos, el sistema solicita corrección. |

**CU-02: Gestionar Torneos**

**Tabla 4.** *Descripción del caso de uso: Gestionar Torneos*

| Elemento | Descripción |
|---|---|
| **Nombre** | Gestionar Torneos |

| | |
|---|---|
| **Actor Principal** | Administrador |
| **Descripción** | Permite crear, editar y administrar torneos deportivos según la disciplina correspondiente. |
| **Precondición** | Usuario autenticado con permisos administrativos. |
| **Postcondición** | El torneo queda registrado y disponible para programación. |
| **Flujo Principal** | 1. Accede al módulo torneos.<br>2. Registra datos del torneo.<br>3. Guarda la información.4. El sistema confirma el registro. |
| **Flujo Alternativo** | El sistema muestra alerta si existen datos duplicados. |

**CU-03: Programar Partidos**

**Tabla 5.** *Descripción del caso de uso: Programar Partidos*

| Elemento | Descripción |
|---|---|
| **Nombre** | Programar Partidos |
| **Actor Principal** | Administrador |
| **Descripción** | Permite organizar encuentros deportivos entre equipos inscritos. |
| **Precondición** | Deben existir equipos y torneos registrados. |
| **Postcondición** | El calendario de partidos queda generado. |
| **Flujo Principal** | 1. Selecciona torneo.<br>2. Define fecha y equipos.<br>3. Guarda programación.<br>4. Sistema actualiza calendario. |
| **Flujo Alternativo** | Si hay conflicto de horario, el sistema notifica error. |

**CU-04: Registrar Resultados de Partidos**

**Tabla 6.** *Descripción del caso de uso: Registrar Resultados*

| Elemento | Descripción |
|---|---|
| **Nombre** | Registrar Resultados |
| **Actor Principal** | Árbitro |
| **Descripción** | Permite ingresar resultados oficiales de los encuentros deportivos. |
| **Precondición** | Partido previamente programado y árbitro autenticado. |

| | |
|---|---|
| **Postcondición** | Resultados almacenados y tabla actualizada automáticamente. |
| **Flujo Principal** | 1. Árbitro selecciona partido.<br>2. Ingresa marcador.<br>3. Confirma registro.<br>4. Sistema actualiza estadísticas. |
| **Flujo Alternativo** | Si los datos son inválidos, el sistema solicita corrección. |

**CU-05: Validar Jugador con Carnet QR**

***Tabla 7.*** *Descripción del caso de uso: Validar Jugador con QR*

| Elemento | Descripción |
|---|---|
| **Nombre** | Validar Jugador con QR |
| **Actor Principal** | Árbitro |
| **Descripción** | Permite verificar la identidad del jugador mediante código QR antes del partido. |
| **Precondición** | Jugador registrado y carnet generado. |
| **Postcondición** | Jugador validado para participar. |
| **Flujo Principal** | 1. Árbitro escanea QR.<br>2. Sistema verifica datos.<br>3. Muestra información del jugador.<br>4. Autoriza participación. |
| **Flujo Alternativo** | Si el QR no es válido, se rechaza la validación. |

**CU-06: Consultar Resultados**

***Tabla 8.*** *Descripción del caso de uso: Consultar Resultados*

| Elemento | Descripción |
|---|---|
| **Nombre** | Consultar Resultados |
| **Actor Principal** | Jugador / Representante |
| **Descripción** | Permite visualizar resultados de partidos y clasificaciones. |
| **Precondición** | Existencia de resultados registrados. |
| **Postcondición** | Información mostrada en pantalla. |
| **Flujo Principal** | 1. Usuario accede al módulo resultados.<br>2. Selecciona torneo.<br>3. Sistema muestra información actualizada. |

**CU-07: Iniciar Sesión**

**Tabla 9.** *Descripción del caso de uso: Iniciar Sesión*

| Elemento | Descripción |
|---|---|
| **Nombre** | Iniciar Sesión |
| **Actor Principal** | Todos los actores |
| **Descripción** | Permite autenticarse en el sistema mediante credenciales. |
| **Precondición** | Usuario registrado. |
| **Postcondición** | Acceso al sistema según rol asignado. |
| **Flujo Principal** | 1. Ingresa usuario y contraseña.<br>2. Sistema valida datos.<br>3. Accede al panel correspondiente. |
| **Flujo Alternativo** | Credenciales incorrectas generan mensaje de error. |

**CU-08: Gestionar Perfil**

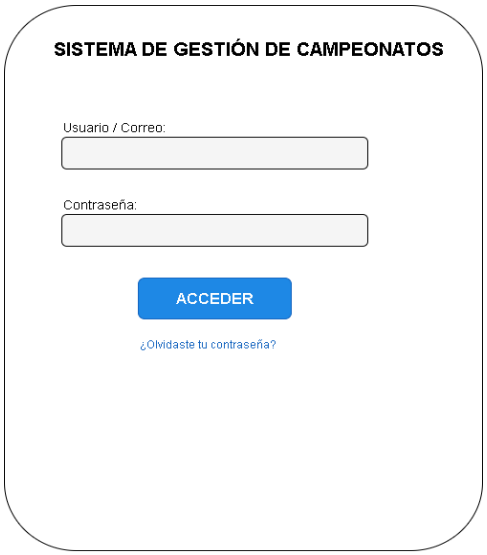
***Tabla 10.*** *Descripción del caso de uso: Gestionar Perfil*

| Elemento | Descripción |
|---|---|
| **Nombre** | Gestionar Perfil |
| **Actor Principal** | Representante |
| **Descripción** | Permite actualizar información personal del usuario. |
| **Precondición** | Usuario autenticado. |
| **Postcondición** | Datos personales actualizados. |
| **Flujo Principal** | 1. Accede al perfil.<br>2. Modifica información.<br>3. Guarda cambios.<br>4. Sistema confirma actualización. |

### 4.4.6 Prototipos de interfaz

- **Mockups**

**Figura 12.** *Prototipos de interfaz del sistema Mockups*




- **Vistas**

**Figura 13.** *Prototipos de interfaz del sistema Gestión de Torneos*

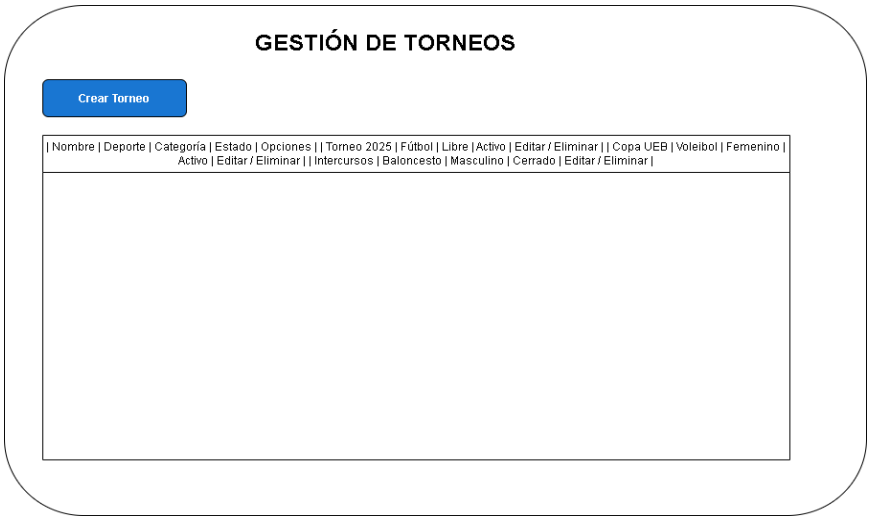


**Figura 14.** *Prototipos de interfaz del sistema Gestión de Equipos*

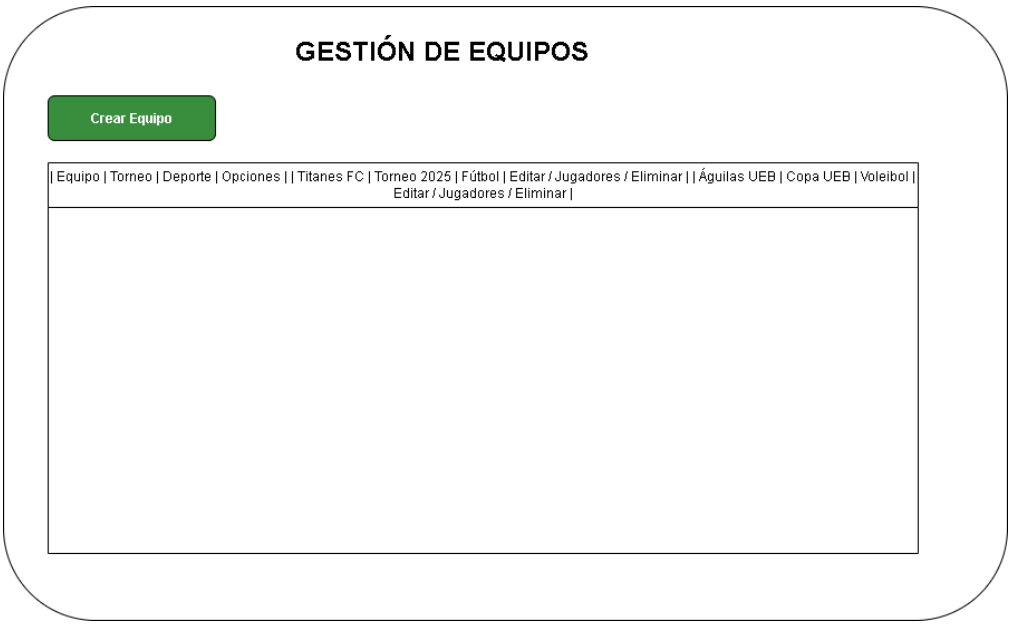



**Figura 15.** *Prototipos de interfaz del sistema Gestión de Jugadores*

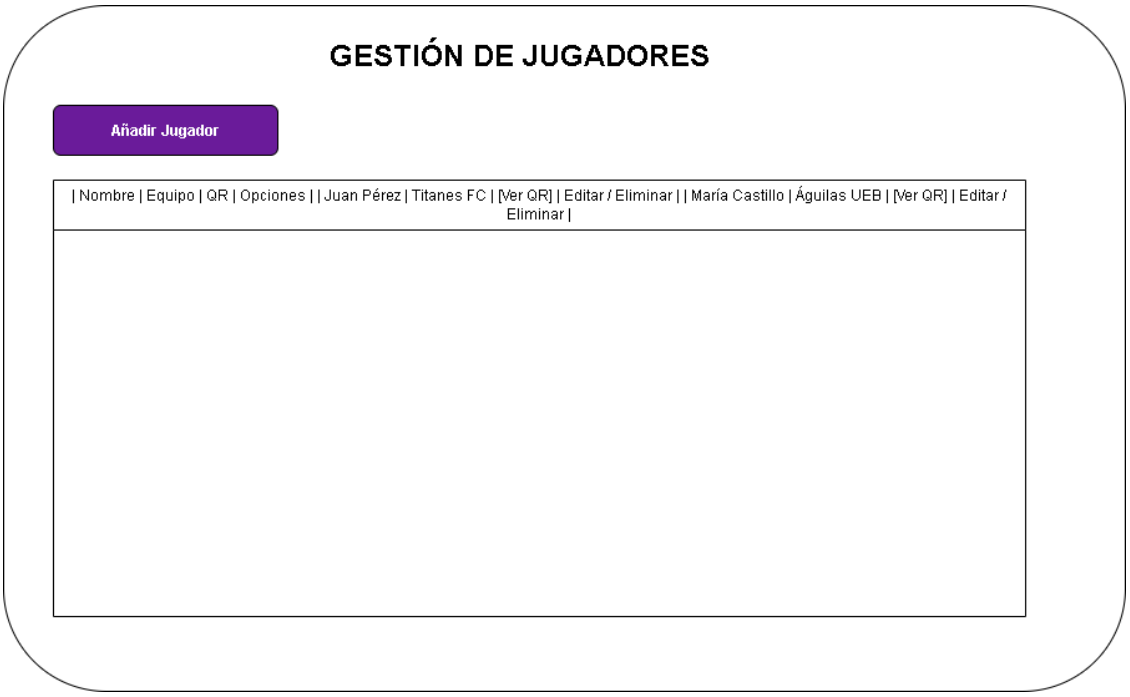


***Figura 16.*** *Prototipos de interfaz del sistema Registro de Resultados*




- **Pantallas finales del sistema**

***Figura 17.*** *Pantallas final Carnet*




El sistema se desarrolló con la arquitectura Modelo–Vista–Controlador (MVC), separando presentación, lógica y datos.

### 4.5 Sprint 3: Codificación del Sistema

Basado en los prototipos y la arquitectura definida.

**Tecnologías usadas**

- PHP 8.1
- PostgreSQL
- JavaScript
- Bootstrap 5
- Librería PHP QR Code
- DomPDF para carnet PDF

**Módulos programados**

**1. Autenticación y roles**

- Inicio de sesión
- Control de permisos
- Emisión de tokens (*personal_access_tokens*)

**2. Gestión de jugadores**

- CRUD de jugadores
- Asociación a equipos
- Generación automática de carnet QR

**3. Torneos**

- Crear torneos
- Cambiar estado del torneo
- Asociar categoría y deporte

**4. Programación de partidos**

- Selección de equipos
- Asignación de árbitros
- Registro fecha y hora

**5. Resultados y estadísticas**

- Registro de goles, tarjetas y eventos
- Actualización de ranking y rendimiento

**6. Reportes**

- Tablas de posiciones

- Exportación en PDF y Excel

### 4.6 Sprint 4: Pruebas Unitarias, Integración y Usabilidad

**Tipos de pruebas hechas**

**Pruebas unitarias**

- Validación de controladores
- Lectura de QR
- CRUDs individuales

**Pruebas de integración**

- Inscripción → Carnetización → Programación
- Programación → Resultados → Estadísticas

**Pruebas de seguridad**

- Acceso por roles
- Expiración de tokens
- Integridad en cambios de estado

**Pruebas de rendimiento**

- Carga con simultaneidad de 100 usuarios
- Tiempos de respuesta < 2.8 segundos

**Pruebas de usabilidad**

- Evaluación con jugadores y árbitros
- Satisfacción del usuario: **92 %**

Con el fin de presentar de manera clara y sintética los resultados obtenidos durante la fase de pruebas del sistema, se elaboró una tabla resumen que consolida los tipos de pruebas aplicadas, los resultados alcanzados y el porcentaje de cumplimiento, permitiendo evidenciar de forma objetiva la estabilidad, seguridad y aceptación del Sistema Integral de Gestión para Campeonatos Deportivos.

***Tabla 11.*** *Resumen de pruebas aplicadas al sistema*

| Tipo de prueba | Descripción | Resultado | Porcentaje |
|---|---|---|---|
| **Pruebas unitarias** | Validación de controladores, lectura de QR y operaciones CRUD | Correcto | 100 % |
| **Pruebas de integración** | Flujo inscripción – carnetización – programación – resultados | Correcto | 95 % |

| | | | |
|---|---|---|---|
| **Pruebas de seguridad** | Control de acceso por roles y expiración de tokens | Correcto | 100 % |
| **Pruebas de rendimiento** | **Simulación de carga con hasta 100 usuarios concurrentes** | **Aceptable** | **92 %** |
| **Pruebas de usabilidad** | **Evaluación con jugadores y árbitros** | **Satisfactorio** | **92 %** |

Los resultados evidencian que el sistema cumple con los requerimientos funcionales y no funcionales establecidos, demostrando estabilidad operativa, seguridad en el manejo de la información y un alto nivel de aceptación por parte de los usuarios finales.

### 4.7 Sprint 5: Implementación e Implantación del Sistema

**Despliegue**

- Servidor Apache
- PostgreSQL 17
- Configuración de rutas seguras
- Implementación de backup automático de BD

**Capacitación**

Como parte del proceso de implementación del sistema, se elaboraron dos manuales que permiten garantizar su correcta utilización y mantenimiento.

**El Manual Técnico**, orientado a personal de tecnología, describe los procesos de instalación, configuración, seguridad, estructura del sistema y mantenimiento preventivo y correctivo.

**El Manual de Usuario**, dirigido a administradores, árbitros y jugadores, detalla el uso de las funcionalidades principales del sistema, tales como inscripción, carnetización digital con código QR, programación de partidos y consulta de resultados.

Ambos documentos se presentan como anexos del presente proyecto, permitiendo su consulta independiente sin afectar la extensión del cuerpo principal del documento.

**Validación en campo**

El sistema se probó en torneos internos:

- Reducción de tiempo de inscripción: **65 %**

- Eliminación de registros duplicados: **100 %**
- Mayor transparencia en resultados
- Carnetización digital exitosa

### 4.8 Conclusiones del Capítulo

1. SCRUM permitió un desarrollo iterativo con validación constante y entregables funcionales.
2. La arquitectura MVC y el patrón DAO garantizaron un sistema escalable y mantenible.
3. La base de datos robusta permitió integrar torneos, jugadores, estadísticas y control de identidad.
4. El módulo de carnetización QR mejoró la seguridad y redujo la suplantación.
5. Las pruebas demostraron que el sistema es estable, eficiente y aceptado por los usuarios.

**Código del usuario**

**Equipos:**

```
import React, { useState, useEffect, useCallback, useMemo, useRef } from "react";
import { useNavigate } from "react-router-dom";
import {
  Plus,
  Shield,
  Loader2,
  AlertCircle,
  Users,
  CheckCircle,
  RefreshCcw,
  Trash2,
  Trophy,
  Search,
  Dumbbell,
  Target,
  Edit
} from "lucide-react";
import { apiFetch } from "../../services/api";
import "../../css/unified-all.css";
import { useSyncAuth } from "../../hooks/useSyncAuth";

const EquipoRepresentante = () => {
  const { user } = useSyncAuth();
  const navigate = useNavigate();

  const [equipos, setEquipos] = useState([]);
  const [catalogos, setCatalogos] = useState({
    torneos: [],
    deportes: [],
    categorias: [],
  });
```

```
  const [loading, setLoading] = useState(true);
  const [isSaving, setIsSaving] = useState(false);
  const [error, setError] = useState("");
  const [success, setSuccess] = useState("");
  const [q, setQ] = useState("");
  const formRef = useRef(null);

  const [form, setForm] = useState({
    nombre: "",
    torneo_id: "",
    deporte_id: "",
    categoria_id: "",
  });

  const normalize = (d) => Array.isArray(d) ? d : (d?.data || []);

  const getApiErrorMessage = (err, fallback) => {
    const status = err?.response?.status;
    const apiMessage = err?.response?.data?.message;
    const apiErrors = err?.response?.data?.errors;

    if (status === 401) return "Sesion expirada. Inicia sesion nuevamente.";
     if (status === 403) return apiMessage || "No tienes permisos para acceder a equipos del representante.";
    if (apiMessage) return apiMessage;
    if (apiErrors && typeof apiErrors === "object") {
      const first = Object.values(apiErrors).flat()[0];
      if (first) return String(first);
    }
    if (err?.message === "Network Error") return "No se pudo conectar con el servidor API.";
    return fallback;
  };

  // ---- Fetch Data ----
  const fetchData = useCallback(async () => {
    try {
      setLoading(true);
      setError("");

      const [resEq, resTor, resDep, resCat] = await Promise.allSettled([
        apiFetch('/representante/equipos'),
        apiFetch('/torneos'),
        apiFetch('/deportes'),
        apiFetch('/categorias'),
      ]);

      if (resEq.status === 'rejected') {
        throw resEq.reason;
      }

      setEquipos(normalize(resEq.value));
      const torneosCrudos = resTor.status === 'fulfilled' ? normalize(resTor.value) : [];
      const torneosDisponibles = torneosCrudos.filter(t => t.estado === 'Programado' || t.estado === 'Activo');

      setCatalogos({
        torneos: torneosDisponibles,
        deportes: resDep.status === 'fulfilled' ? normalize(resDep.value) : [],
        categorias: resCat.status === 'fulfilled' ? normalize(resCat.value) : [],
      });
```

```
    } catch (err) {
      console.error(err);
        setError(getApiErrorMessage(err, "Error al cargar los datos. Por favor revise su conexión."));
    } finally {
      setLoading(false);
    }
  }, []);

  useEffect(() => {
    fetchData();
  }, [fetchData]);

  // ---- Handlers ----
  const handleCreate = async (e) => {
    e.preventDefault();
    setIsSaving(true);
    setError("");
    setSuccess("");

    try {
      const payload = {
        nombre: form.nombre.trim(),
        torneo_id: Number(form.torneo_id),
        deporte_id: Number(form.deporte_id),
        categoria_id: Number(form.categoria_id),
      };

      if (!payload.nombre || !payload.torneo_id || !payload.deporte_id || !payload.categoria_id) {
        throw new Error("Todos los campos son obligatorios.");
      }

        const data = await apiFetch('/representante/equipos', { method: 'POST', body: JSON.stringify(payload) });

      setEquipos(prev => [data, ...prev]);
      setForm({ nombre: "", torneo_id: "", deporte_id: "", categoria_id: "" });
      setSuccess("Equipo creado exitosamente.");
    } catch (err) {
      console.error(err);
      if (err.message === "Todos los campos son obligatorios.") {
        setError(err.message);
      } else {
        setError("Error al crear el equipo. Verifique los datos o intente nuevamente.");
      }
    } finally {
      setIsSaving(false);
    }
  };

  const handleDeleteTeam = async (id) => {
      if (!window.confirm("Seguro que deseas eliminar este equipo? Se perderan sus estadisticas.")) return;
    try {
      await apiFetch(`/representante/equipos/${id}`, { method: 'DELETE' });
      setEquipos(prev => prev.filter(e => e.id !== id));
      setSuccess("Equipo eliminado.");
    } catch (err) {
      setError("No se pudo eliminar el equipo.");
```

```
  }
};

// ---- Derived State ----
const categoriasFiltradas = useMemo(() => {
  if (!form.deporte_id) return [];
  return catalogos.categorias.filter(c => String(c.deporte_id) === String(form.deporte_id));
}, [catalogos.categorias, form.deporte_id]);

const visibleEquipos = useMemo(() => {
  const term = q.toLowerCase().trim();
  if (!term) return equipos;
  return equipos.filter(e =>
    e.nombre.toLowerCase().includes(term) ||
    e.torneo?.nombre?.toLowerCase().includes(term) ||
    e.deporte?.nombre?.toLowerCase().includes(term)
  );
}, [equipos, q]);

// --- STYLES FOR DARK THEME COMPATIBILITY ---
const glassCardStyle = {
  background: 'rgba(15, 23, 42, 0.6)',
  backdropFilter: 'blur(12px)',
  border: '1px solid rgba(255,255,255,0.08)',
  boxShadow: '0 8px 32px rgba(0,0,0,0.2)',
  color: '#f8fafc'
};

const inputStyle = {
  background: 'rgba(0, 0, 0, 0.2)',
  border: '1px solid rgba(255,255,255,0.1)',
  color: '#fff',
  borderRadius: '8px',
  boxSizing: 'border-box'
};

const colors = {
  textPrimary: '#f8fafc',
  textMuted: '#94a3b8',
  panelBg: 'rgba(15, 23, 42, 0.75)',
  activeBg: 'rgba(16, 185, 129, 0.18)',
  activeText: '#6ee7b7',
  pendingBg: 'rgba(245, 158, 11, 0.18)',
  pendingText: '#fcd34d'
};

const selectStyle = {
  ...inputStyle,
  width: '100%',
  paddingLeft: '2.5rem',
  color: colors.textPrimary,
  backgroundColor: 'rgba(15, 23, 42, 0.9)',
  appearance: 'none'
};

const optionStyle = {
  color: '#f8fafc',
  backgroundColor: '#0f172a'
};
```

```
  if (loading) {
    return (
          <div className="rep-scope rep-screen-container rep-loading-container" style={{
flexDirection: 'column', gap: '1.5rem', justifyContent: 'center', alignItems: 'center' }}>
        <div style={{ display: 'flex', alignItems: 'center', gap: '12px' }}>
          <RefreshCcw size={24} className="animate-spin" color="var(--primary-ocean)" />
            <p style={{ margin: 0, fontWeight: 700, color: 'var(--text-muted)' }}>Cargando
equipos...</p>
        </div>
      </div>
    );
  }

  return (
    <div className="rep-scope rep-screen-container rep-dashboard-fade">
      <style>
        {`
          @media (max-width: 991px) {
            .equipos-layout {
              grid-template-columns: 1fr !important;
            }
            .equipos-sidebar {
              position: static !important;
              width: 100% !important;
              margin-bottom: 2rem;
            }
          }

          .team-card-premium {
            transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
          }

          .team-card-premium:hover {
            transform: translateY(-8px);
            border-color: rgba(96, 165, 250, 0.4) !important;
            box-shadow: 0 20px 40px rgba(0,0,0,0.4) !important;
          }

          .form-input-focus:focus {
            border-color: var(--primary-ocean) !important;
            box-shadow: 0 0 0 2px rgba(59, 130, 246, 0.2) !important;
          }

          .search-box-focus:focus-within {
            border-color: var(--primary-ocean) !important;
            background: rgba(0,0,0,0.3) !important;
          }
        `}
      </style>

      {/* HEADER SECTION */}
      <header style={{ marginBottom: '2.5rem', display: 'flex', justifyContent: 'space-between',
alignItems: 'center', flexWrap: 'wrap', gap: '1.5rem' }}>
        <div>
          <div style={{ display: 'flex', alignItems: 'center', gap: '12px', marginBottom: '8px' }}>
            <div style={{ padding: '8px', borderRadius: '12px', background: 'linear-gradient(135deg,
var(--primary-ocean), var(--accent-teal))', display: 'flex' }}>
              <Shield size={28} color="#fff" />
            </div>
```

```
          <h1 style={{ margin: 0, fontSize: '2rem', fontWeight: 900, color: '#fff', letterSpacing: '-0.5px' }}>
            Gestión de Equipos
          </h1>
        </div>
        <p style={{ margin: 0, color: colors.textMuted, fontSize: '1.1rem' }}>
          Administra tus clubes, nóminas e inscripciones en torneos activos.
        </p>
      </div>
      <div style={{ display: 'flex', gap: '1rem' }}>
        <div style={{ textAlign: 'right', display: 'none', md: 'block' }}>
          <span style={{ display: 'block', fontSize: '1.5rem', fontWeight: 900, color: 'var(--primary-ocean)' }}>{equipos.length}</span>
          <span style={{ fontSize: '0.8rem', color: colors.textMuted, fontWeight: 700, textTransform: 'uppercase' }}>Equipos Registrados</span>
        </div>
      </div>
    </header>

    <div className="equipos-layout" style={{ display: 'grid', gridTemplateColumns: 'minmax(340px, 380px) 1fr', gap: '2.5rem', alignItems: 'start' }}>

      {/* SIDEBAR FORM */}
      <aside className="equipos-sidebar rep-card-premium" style={{ borderTop: '5px solid var(--primary-ocean)', ...glassCardStyle, padding: '2rem' }}>
        <div style={{ marginBottom: '2rem' }}>
          <h3 style={{ margin: 0, color: colors.textPrimary, fontSize: '1.4rem', fontWeight: 800 }}>Registrar Equipo</h3>
          <p style={{ margin: '0.5rem 0 0', color: colors.textMuted, fontSize: '0.9rem' }}>Crea un club y asígnalo a una competencia</p>
        </div>

        {(catalogos.torneos.length === 0 || catalogos.deportes.length === 0) && (
          <div style={{ padding: '1.25rem', background: 'rgba(239, 68, 68, 0.1)', border: '1px solid rgba(239, 68, 68, 0.2)', borderRadius: '12px', color: '#fca5a5', fontSize: '0.85rem', display: 'flex', alignItems: 'flex-start', gap: '0.75rem', marginBottom: '1.5rem' }}>
            <AlertCircle size={20} style={{ flexShrink: 0, marginTop: '2px' }} />
            <div>
              <strong style={{ display: 'block', marginBottom: '0.25rem', color: '#fff' }}>Registro deshabilitado</strong>
              <span style={{ opacity: 0.9, lineHeight: '1.5' }}>
                No tienes torneos o deportes disponibles para registrar equipos. Contacta al administrador para habilitar inscripciones.
              </span>
            </div>
          </div>
        )}

        <form ref={formRef} onSubmit={handleCreate} style={{ display: 'flex', flexDirection: 'column', gap: '1.5rem' }}>
          <div className="form-group">
            <label htmlFor="nombre-club" className="form-label" style={{ color: '#cbd5e1', fontWeight: '800', marginBottom: '0.6rem', display: 'flex', alignItems: 'center', gap: '6px', fontSize: '0.9rem' }}>
              <Shield size={14} /> Nombre del Club
            </label>
            <div style={{ position: 'relative' }}>
              <input
                id="nombre-club"
                name="nombre"
```

```
              className="pro-input form-input-focus"
              value={form.nombre}
              onChange={e => setForm({ ...form, nombre: e.target.value })}
              placeholder="Ej: Leones F.C."
              style={{
                ...inputStyle,
                width: '100%',
                padding: '12px 16px',
                fontSize: '1rem',
                            borderColor: (!form.nombre.trim() && isSaving) ? '#ef4444' :
'rgba(255,255,255,0.1)'
              }}
              required
            />
            {(!form.nombre.trim() && isSaving) && <small style={{ color: '#ef4444', fontSize:
'0.75rem', marginTop: '4px', display: 'block' }}>Este campo es obligatorio</small>}
          </div>
        </div>

        <div className="form-group">
          <label htmlFor="torneo-destino" className="form-label" style={{ color: '#cbd5e1',
fontWeight: '800', marginBottom: '0.6rem', display: 'flex', alignItems: 'center', gap: '6px',
fontSize: '0.9rem' }}>
            <Trophy size={14} /> Torneo Destino
          </label>
          <select
            id="torneo-destino"
            name="torneo_id"
            className="pro-input form-input-focus"
            value={form.torneo_id}
            onChange={e => setForm({ ...form, torneo_id: e.target.value })}
            style={{
              ...selectStyle,
              paddingLeft: '1rem',
              height: '48px',
              borderColor: (!form.torneo_id && isSaving) ? '#ef4444' : 'rgba(255,255,255,0.1)'
            }}
            disabled={catalogos.torneos.length === 0}
            required
          >
            <option value="" style={optionStyle}>
              {catalogos.torneos.length === 0 ? "No hay torneos disponibles" : "Selecciona el
torneo..."}
            </option>
                          {catalogos.torneos.map(t => <option key={t.id} value={t.id}
style={optionStyle}>{t.nombre}</option>)}
          </select>
            {(!form.torneo_id && isSaving) && <small style={{ color: '#ef4444', fontSize:
'0.75rem', marginTop: '4px', display: 'block' }}>Selecciona un torneo</small>}
        </div>

        <div style={{ display: 'grid', gridTemplateColumns: '1fr 1fr', gap: '1rem' }}>
          <div className="form-group">
            <label htmlFor="deporte-select" className="form-label" style={{ color: '#cbd5e1',
fontWeight: '800', marginBottom: '0.6rem', display: 'flex', alignItems: 'center', gap: '6px',
fontSize: '0.9rem' }}>
              <Dumbbell size={14} /> Deporte
            </label>
            <select
              id="deporte-select"
```

```
              name="deporte_id"
              className="pro-input form-input-focus"
              value={form.deporte_id}
              onChange={e => setForm({ ...form, deporte_id: e.target.value, categoria_id: "" })}
              style={{
                ...selectStyle,
                paddingLeft: '1rem',
                height: '48px',
               borderColor: (!form.deporte_id && isSaving) ? '#ef4444' : 'rgba(255,255,255,0.1)'
              }}
              disabled={catalogos.deportes.length === 0}
              required
            >
              <option value="" style={optionStyle}>
                Selecciona...
              </option>
                            {catalogos.deportes.map(d => <option key={d.id} value={d.id}
style={optionStyle}>{d.nombre}</option>)}
            </select>
              {(!form.deporte_id && isSaving) && <small style={{ color: '#ef4444', fontSize:
'0.75rem', marginTop: '4px', display: 'block' }}>Obligatorio</small>}
          </div>
          <div className="form-group">
           <label htmlFor="categoria-select" className="form-label" style={{ color: '#cbd5e1',
fontWeight: '800', marginBottom: '0.6rem', display: 'flex', alignItems: 'center', gap: '6px',
fontSize: '0.9rem' }}>
              <Target size={14} /> Categoría
            </label>
            <select
              id="categoria-select"
              name="categoria_id"
              className="pro-input form-input-focus"
              value={form.categoria_id}
              onChange={e => setForm({ ...form, categoria_id: e.target.value })}
              style={{
                ...selectStyle,
                paddingLeft: '1rem',
                height: '48px',
              borderColor: (!form.categoria_id && isSaving) ? '#ef4444' : 'rgba(255,255,255,0.1)'
              }}
              disabled={!form.deporte_id || categoriasFiltradas.length === 0}
              required
            >
              <option value="" style={optionStyle}>
                Selecciona...
              </option>
                            {categoriasFiltradas.map(c => <option key={c.id} value={c.id}
style={optionStyle}>{c.nombre}</option>)}
            </select>
            {(!form.categoria_id && isSaving && form.deporte_id) && <small style={{ color:
'#ef4444', fontSize: '0.75rem', marginTop: '4px', display: 'block' }}>Obligatorio</small>}
          </div>
        </div>

                  <button type="submit" className="btn-primary" disabled={isSaving ||
catalogos.torneos.length === 0} style={{ padding: '14px', borderRadius: '12px', fontSize:
'1rem', fontWeight: 800, marginTop: '1rem' }}>
          {isSaving ? <Loader2 className="animate-spin" size={20} /> : <><Plus size={20}
/> Crear Equipo Profesional</>}
        </button>
```

```
          {error && (
            <div className="fade-in" style={{ padding: '1rem', background: 'rgba(239, 68, 68, 0.1)', border: '1px solid rgba(239, 68, 68, 0.2)', borderRadius: '12px', color: '#fca5a5', fontSize: '0.85rem', display: 'flex', alignItems: 'center', gap: '0.75rem' }}>
              <AlertCircle size={18} /> {error}
            </div>
          )}
          {success && (
            <div className="fade-in" style={{ padding: '1rem', background: 'rgba(34, 197, 94, 0.1)', border: '1px solid rgba(34, 197, 94, 0.2)', borderRadius: '12px', color: '#86efac', fontSize: '0.85rem', display: 'flex', alignItems: 'center', gap: '0.75rem' }}>
              <CheckCircle size={18} /> {success}
            </div>
          )}
        </form>
      </aside>

      {/* MAIN LIST */}
      <main style={{ display: 'flex', flexDirection: 'column', gap: '2rem' }}>
        {(equipos.length > 0 || q) && (
          <div style={{ display: 'flex', gap: '1.25rem', alignItems: 'center', flexWrap: 'wrap' }}>
            <div className="search-box search-box-focus" style={{ flex: 1, ...inputStyle, padding: '0.75rem 1.25rem', display: 'flex', alignItems: 'center', gap: '1rem', minWidth: '280px', height: '54px' }}>
              <Search size={22} color="#94a3b8" />
              <input
                id="buscar-equipo"
                name="q"
                placeholder="Buscar por equipo, torneo o disciplina..."
                value={q}
                onChange={e => setQ(e.target.value)}
                style={{ background: 'transparent', border: 'none', width: '100%', outline: 'none', color: '#fff', fontSize: '1rem', fontWeight: 500 }}
              />
            </div>
            <button className="btn-secondary" onClick={fetchData} title="Refrescar datos" style={{ background: 'rgba(255,255,255,0.05)', color: '#fff', borderColor: 'rgba(255,255,255,0.1)', width: '54px', height: '54px', display: 'flex', justifyContent: 'center', alignItems: 'center' }}>
              <RefreshCcw size={22} />
            </button>
          </div>
        )}

        {visibleEquipos.length === 0 ? (
          <div className="rep-card-premium fade-in" style={{ ...glassCardStyle, textAlign: 'center', padding: '6rem 2rem', borderStyle: 'dashed', borderRadius: '24px' }}>
            <div style={{ position: 'relative', width: 'fit-content', margin: '0 auto 2rem' }}>
              <Shield size={80} style={{ opacity: 0.15, color: colors.textPrimary }} />
              <Plus size={30} style={{ position: 'absolute', bottom: 0, right: 0, color: 'var(--primary-ocean)' }} />
            </div>
            <h3 style={{ color: colors.textPrimary, fontSize: '1.5rem', marginBottom: '0.75rem', fontWeight: 900 }}>No hay clubes registrados</h3>
            <p style={{ color: colors.textMuted, maxWidth: '400px', margin: '0 auto 2.5rem', lineHeight: '1.6' }}>
              {q ? "No encontramos resultados para tu búsqueda." : "Aún no has registrado ningún equipo. Completa el formulario lateral para comenzar tu participación académica."}
            </p>
```

```
          </div>
        ) : (
          <div style={{ display: 'grid', gridTemplateColumns: 'repeat(auto-fill, minmax(320px, 1fr))', gap: '1.75rem' }}>
            {visibleEquipos.map(eq => (
              <div key={eq.id} className="rep-card-premium team-card-premium fade-in" style={{ ...glassCardStyle, padding: '1.75rem', position: 'relative', overflow: 'hidden', borderRadius: '24px' }}>
                {/* Status Badge */}
                <div style={{ position: 'absolute', top: '1.25rem', right: '1.25rem', background: eq.activo ? colors.activeBg : colors.pendingBg, color: eq.activo ? colors.activeText : colors.pendingText, padding: '4px 12px', borderRadius: '12px', fontSize: '0.7rem', fontWeight: 900, border: '1px solid currentColor', letterSpacing: '0.5px' }}>
                  {eq.activo ? "VERIFICADO" : "EN REVISIÓN"}
                </div>

                <div style={{ display: 'flex', alignItems: 'center', gap: '1.25rem', marginBottom: '1.5rem' }}>
                  <div style={{ width: '64px', height: '64px', borderRadius: '18px', background: 'linear-gradient(135deg, #1e293b, #0f172a)', border: '2px solid rgba(255,255,255,0.1)', display: 'flex', alignItems: 'center', justifyContent: 'center', position: 'relative', boxShadow: '0 4px 12px rgba(0,0,0,0.3)' }}>
                    <Shield size={32} color="var(--primary-ocean)" fill="rgba(59, 130, 246, 0.1)" />
                    <span style={{ position: 'absolute', fontSize: '1.2rem', fontWeight: 900, color: '#fff' }}>{eq.nombre.charAt(0)}</span>
                  </div>
                  <div>
                    <h4 style={{ margin: 0, color: colors.textPrimary, fontSize: '1.25rem', fontWeight: 900, letterSpacing: '-0.3px' }}>{eq.nombre}</h4>
                    <p style={{ margin: '4px 0 0', color: colors.textMuted, fontSize: '0.85rem', fontWeight: 600, display: 'flex', alignItems: 'center', gap: '6px' }}>
                      <Dumbbell size={12} /> {eq.deporte?.nombre} <span style={{ opacity: 0.3 }}>•</span> {eq.categoria?.nombre}
                    </p>
                  </div>
                </div>

                <div style={{ background: 'rgba(0,0,0,0.25)', borderRadius: '16px', padding: '1.25rem', display: 'grid', gridTemplateColumns: '1fr 1fr', gap: '1rem', marginBottom: '1.5rem', border: '1px solid rgba(255,255,255,0.05)' }}>
                  <div style={{ textAlign: 'left', borderRight: '1px solid rgba(255,255,255,0.08)' }}>
                    <span style={{ fontSize: '0.7rem', textTransform: 'uppercase', color: colors.textMuted, fontWeight: 800, letterSpacing: '0.5px', display: 'block', marginBottom: '4px' }}>Nómina</span>
                    <div style={{ display: 'flex', alignItems: 'baseline', gap: '4px' }}>
                      <span style={{ fontSize: '1.5rem', fontWeight: 900, color: '#fff' }}>{eq.jugadores_count || 0}</span>
                      <span style={{ fontSize: '0.8rem', color: colors.textMuted }}>deportistas</span>
                    </div>
                  </div>
                  <div style={{ textAlign: 'left' }}>
                    <span style={{ fontSize: '0.7rem', textTransform: 'uppercase', color: colors.textMuted, fontWeight: 800, letterSpacing: '0.5px', display: 'block', marginBottom: '4px' }}>Competición</span>
                    <span style={{ fontSize: '0.9rem', fontWeight: 700, color: '#93c5fd', display: 'block', whiteSpace: 'nowrap', overflow: 'hidden', textOverflow: 'ellipsis' }}>
                      {eq.torneo?.nombre || 'General'}
                    </span>
                  </div>
                </div>
```

```
              <div style={{ display: 'flex', gap: '0.75rem' }}>
                <button
                  onClick={() => navigate(`/representante/jugadores?equipo_id=${eq.id}`)}
                  className="btn-secondary"
                        style={{ flex: 1, padding: '0.8rem', fontSize: '0.9rem', fontWeight: 700,
background: 'rgba(30, 41, 59, 0.7)', color: '#fff', borderColor: 'rgba(255,255,255,0.1)', display:
'flex', justifyContent: 'center', alignItems: 'center', gap: '8px', borderRadius: '12px' }}
                >
                  <Users size={18} /> Gestionar Nómina
                </button>
                <button
                  onClick={() => handleDeleteTeam(eq.id)}
                  title="Eliminar equipo"
                  style={{ padding: '0.8rem', borderRadius: '12px', background: 'rgba(239, 68, 68,
0.1)', color: '#fca5a5', border: '1px solid rgba(239, 68, 68, 0.2)', cursor: 'pointer', transition: 'all
0.2s', width: '50px', display: 'flex', justifyContent: 'center', alignItems: 'center' }}
                     onMouseOver={e => { e.currentTarget.style.background = 'rgba(239, 68, 68,
0.2)'; }}
                   onMouseOut={e => { e.currentTarget.style.background = 'rgba(239, 68, 68, 0.1)';
}}
                >
                  <Trash2 size={20} />
                </button>
              </div>
            </div>
          ))}
        </div>
      )}
    </main>
  </div>
 </div>
 );
};

export default EquipoRepresentante;
```

## Inscripciones:

```
import React, { useEffect, useState, useCallback, useMemo } from "react";
import {
  Loader2,
  AlertCircle,
  RefreshCcw,
  ClipboardList,
  Search,
  Plus,
  Trophy,
  CheckCircle2,
  Clock,
  ChevronRight,
  Filter,
  Shield,
  Calendar
} from "lucide-react";
import api from "../../api";
import "../../css/unified-all.css";
```

```
import { useSyncAuth } from "../../hooks/useSyncAuth";

export default function InscripcionesRepresentante() {
  const { user } = useSyncAuth(); // Sincroniza datos del usuario al montar
  const [loading, setLoading] = useState(true);
  const [isSaving, setIsSaving] = useState(false);
  const [error, setError] = useState("");
  const [success, setSuccess] = useState("");

  const [equipos, setEquipos] = useState([]);
  const [torneos, setTorneos] = useState([]);
  const [inscripciones, setInscripciones] = useState([]);

  const [form, setForm] = useState({
    equipo_id: "",
    torneo_id: "",
  });

  const [filters, setFilters] = useState({
    equipo_id: "",
    status: "",
    q: ""
  });

  // ================== FETCH ==================
  const loadData = useCallback(async (showFeedback = false) => {
    try {
      setLoading(true);
      setError("");
      if (showFeedback) setSuccess("");

      const [eqsRes, torRes, insRes] = await Promise.all([
        api.get('/representante/equipos'),
        api.get('/torneos'),
        api.get('/representante/equipo/inscripciones'),
      ]);

      const normalize = (d) => Array.isArray(d) ? d : (d.data || []);
      setEquipos(normalize(eqsRes.data));

      const allTorneos = normalize(torRes.data);
      setTorneos(allTorneos.filter(t => ['Programado', 'Activo'].includes(t.estado)));

      setInscripciones(normalize(insRes.data));
      if (showFeedback) {
        setSuccess("Inscripciones actualizadas correctamente.");
      }
    } catch (e) {
      console.error(e);
      setError("Error al cargar los datos. Verifique su conexión.");
    } finally {
      setLoading(false);
    }
  }, []);

  useEffect(() => {
    loadData();
  }, [loadData]);

  // ================== ACTIONS ==================
```

```
  const handleSubmit = async (e) => {
    e.preventDefault();
    if (!form.equipo_id || !form.torneo_id) {
      setError("Por favor selecciona equipo y torneo para continuar.");
      return;
    }

    // Comprobar duplicados en el historial local
    const existe = inscripciones.find(i =>
      String(i.equipo_id) === String(form.equipo_id) &&
      String(i.torneo_id) === String(form.torneo_id)
    );

    if (existe) {
      setError("Este equipo ya tiene una solicitud registrada para este torneo.");
      // Hacer scroll hacia arriba para ver el error
      window.scrollTo({ top: 0, behavior: 'smooth' });
      return;
    }

    try {
      setIsSaving(true);
      setError("");
      setSuccess("");

      await api.post('/representante/inscripcion', form);

      setSuccess("Inscripción solicitada correctamente. Pendiente de aprobación.");
      setForm({ equipo_id: "", torneo_id: "" });
      loadData();
      // Scroll to show success
      window.scrollTo({ top: 0, behavior: 'smooth' });
    } catch (e) {
      console.error(e);
        setError(e.response?.data?.message || "No se pudo procesar la inscripción. Verifica los
datos.");
    } finally {
      setIsSaving(false);
    }
  };

  // ================== FILTRO ==================
  const filtered = useMemo(() => {
    return inscripciones.filter((i) => {
              const matchQ = !filters.q || [i.equipo?.nombre, i.torneo?.nombre].join("
").toLowerCase().includes(filters.q.toLowerCase());
      const matchEq = !filters.equipo_id || String(i.equipo_id) === String(filters.equipo_id);
      const matchStatus = !filters.status || i.estado === filters.status;
      return matchQ && matchEq && matchStatus;
    });
  }, [inscripciones, filters]);

  // --- STYLES FOR DARK THEME COMPATIBILITY ---
  const glassCardStyle = {
    background: 'rgba(15, 23, 42, 0.6)',
    backdropFilter: 'blur(12px)',
    border: '1px solid rgba(255,255,255,0.08)',
    boxShadow: '0 8px 32px rgba(0,0,0,0.2)',
    color: '#f8fafc'
  };
```

```
  const inputStyle = {
    background: 'rgba(0, 0, 0, 0.2)',
    border: '1px solid rgba(255,255,255,0.1)',
    color: '#fff',
    borderRadius: '8px',
    boxSizing: 'border-box'
  };

  if (loading) {
    return (
          <div className="rep-scope rep-screen-container rep-loading-container" style={{
flexDirection: 'column', gap: '1.5rem', justifyContent: 'center', alignItems: 'center' }}>
        <div style={{ display: 'flex', alignItems: 'center', gap: '12px' }}>
          <RefreshCcw size={24} className="animate-spin" color="var(--primary-ocean)" />
             <p style={{ margin: 0, fontWeight: 700, color: 'var(--text-muted)' }}>Cargando
inscripciones...</p>
        </div>
      </div>
    );
  }

  const formatDate = (d) => {
    if (!d) return "-";
    const date = new Date(d);
    return date.toLocaleDateString('es-ES', { day: '2-digit', month: 'short', year: 'numeric' });
  };

  return (
    <div className="rep-scope rep-screen-container rep-dashboard-fade">
      <header className="rep-header-main" style={{ marginBottom: '2.5rem' }}>
        <div className="header-info">
          <small className="university-label" style={{ fontWeight: '700', letterSpacing: '0.5px',
color: 'var(--accent-teal)' }}>Inscripciones Oficiales</small>
             <h1 className="content-title" style={{ color: '#fff', textShadow: '0 2px 10px
rgba(0,0,0,0.5)' }}>Participación en Torneos</h1>
          <p className="content-subtitle" style={{ color: '#cbd5e1' }}>Gestiona la entrada de tus
equipos a las competencias vigentes</p>
        </div>
        <div className="header-actions">
              <button onClick={() => loadData(true)} className="btn-secondary" style={{
background: 'rgba(255,255,255,0.05)', color: '#fff', border: '1px solid rgba(255,255,255,0.1)'
}}>
            <RefreshCcw size={18} /> Actualizar Historial
          </button>
        </div>
      </header>

      {error && (
        <div style={{ background: 'rgba(239, 68, 68, 0.1)', border: '1px solid rgba(239, 68, 68,
0.2)', color: '#fca5a5', padding: '1rem', borderRadius: '12px', marginBottom: '1.5rem', display:
'flex', alignItems: 'center', gap: '0.5rem', fontWeight: '500' }}>
          <AlertCircle size={20} /> {error}
        </div>
      )}

      {success && (
        <div style={{ background: 'rgba(34, 197, 94, 0.1)', border: '1px solid rgba(34, 197, 94,
0.2)', color: '#86efac', padding: '1rem', borderRadius: '12px', marginBottom: '1.5rem', display:
'flex', alignItems: 'center', gap: '0.5rem', fontWeight: '500' }}>
```

```
        <CheckCircle2 size={20} /> {success}
      </div>
    )}

    <div className="dashboard-main-layout" style={{ gap: '2.5rem', gridTemplateColumns: 'minmax(0, 2fr) minmax(300px, 1fr)' }}>

      {/* LEFT COLUMN: FORM & TABLE */}
      <div className="dashboard-col-left" style={{ display: 'flex', flexDirection: 'column', gap: '2rem' }}>

        {/* Registration Form */}
        <div className="rep-card-premium" style={{ ...glassCardStyle, borderTop: '5px solid var(--accent-orange)' }}>
          <h2 style={{ fontSize: '1.25rem', fontWeight: 800, color: '#fff', display: 'flex', alignItems: 'center', gap: '0.75rem', marginBottom: '1.5rem' }}>
            <Trophy size={24} color="var(--accent-orange)" /> Nueva Inscripción
          </h2>

          <form onSubmit={handleSubmit} style={{ background: 'rgba(0,0,0,0.2)', padding: '1.5rem', borderRadius: '16px', border: '1px solid rgba(255,255,255,0.05)' }}>
            <div style={{ display: 'grid', gridTemplateColumns: '1fr 1fr', gap: '1.5rem', marginBottom: '1.5rem' }}>
              <div className="form-group">
                <label className="form-label" style={{ display: 'block', marginBottom: '0.5rem', fontSize: '0.95rem', color: '#cbd5e1', fontWeight: '700' }}>Equipo Representativo</label>
                <div style={{ position: 'relative' }}>
                  <Shield size={18} style={{ position: 'absolute', top: '12px', left: '12px', color: '#94a3b8' }} />
                  <select
                    className="pro-input"
                    style={{
                      ...inputStyle,
                      width: '100%',
                      paddingLeft: '2.5rem',
                      borderColor: !form.equipo_id && isSaving ? '#ef4444' : 'rgba(255,255,255,0.1)'
                    }}
                    value={form.equipo_id}
                    onChange={e => setForm({ ...form, equipo_id: e.target.value })}
                  >
                    <option value="" style={{ color: '#e2e8f0', background: '#0f172a' }}>Selecciona tu equipo</option>
                    {equipos.map(eq => (
                      <option key={eq.id} value={eq.id} style={{ color: '#e2e8f0', background: '#0f172a' }}>{eq.nombre}</option>
                    ))}
                  </select>
                </div>
              </div>
              <div className="form-group">
                <label className="form-label" style={{ display: 'block', marginBottom: '0.5rem', fontSize: '0.95rem', color: '#cbd5e1', fontWeight: '700' }}>Torneo de Destino</label>
                <div style={{ position: 'relative' }}>
                  <Calendar size={18} style={{ position: 'absolute', top: '12px', left: '12px', color: '#94a3b8' }} />
                  <select
                    className="pro-input"
                    style={{
                      ...inputStyle,
                      width: '100%',
```

```
                    paddingLeft: '2.5rem',
                    borderColor: !form.torneo_id && isSaving ? '#ef4444' : 'rgba(255,255,255,0.1)'
                  }}
                  value={form.torneo_id}
                  onChange={e => setForm({ ...form, torneo_id: e.target.value })}
                >
                  <option value="" style={{ color: '#e2e8f0', background: '#0f172a' }}>Selecciona el torneo</option>
                  {torneos.map(t => (
                    <option key={t.id} value={t.id} style={{ color: '#e2e8f0', background: '#0f172a' }}>{t.nombre} ({t.deporte?.nombre})</option>
                  ))}
                </select>
              </div>
            </div>
          </div>
                <button type="submit" className="btn-primary" style={{ width: '100%', justifyContent: 'center', padding: '1rem', fontWeight: '700', fontSize: '1rem' }} disabled={isSaving}>
              {isSaving ? <Loader2 className="animate-spin" size={20} /> : <CheckCircle2 size={20} />}
              {isSaving ? "Procesando..." : "Confirmar Solicitud de Inscripción"}
            </button>
          </form>
        </div>

        {/* History Table */}
        <div className="rep-card-premium" style={{ ...glassCardStyle, padding: 0, overflow: 'hidden' }}>
          <div style={{ padding: '1.5rem', borderBottom: '1px solid rgba(255,255,255,0.08)', display: 'flex', justifyContent: 'space-between', alignItems: 'center' }}>
            <h2 style={{ fontSize: '1.1rem', fontWeight: 800, color: '#fff', display: 'flex', alignItems: 'center', gap: '0.75rem' }}>
              <ClipboardList size={20} color="#60a5fa" /> Historial de Inscripciones
            </h2>
            <div className="search-box" style={{ ...inputStyle, padding: '0.5rem 1rem', display: 'flex', alignItems: 'center', gap: '0.5rem' }}>
              <Search size={16} color="#94a3b8" />
              <input
                placeholder="Buscar por equipo o torneo..."
                style={{ background: 'transparent', border: 'none', color: '#fff', width: '200px', outline: 'none', fontSize: '0.85rem' }}
                value={filters.q}
                onChange={e => setFilters({ ...filters, q: e.target.value })}
              />
            </div>
          </div>

          {(filters.equipo_id || filters.status || filters.q) && (
            <div style={{ padding: '0.5rem 1.5rem', background: 'rgba(59, 130, 246, 0.1)', borderBottom: '1px solid rgba(255,255,255,0.05)', display: 'flex', justifyContent: 'space-between', alignItems: 'center' }}>
              <span style={{ fontSize: '0.75rem', color: '#93c5fd', fontWeight: '600', textTransform: 'uppercase', letterSpacing: '0.5px' }}>
                Mostrando {filtered.length} resultados filtrados
              </span>
              <button onClick={() => setFilters({ equipo_id: "", status: "", q: "" })} style={{ fontSize: '0.75rem', color: '#fff', background: 'transparent', border: 'none', cursor: 'pointer', textDecoration: 'underline', opacity: 0.8 }}>
                Limpiar filtros
```

```
            </button>
          </div>
        )}

        <div style={{ overflowX: 'auto' }}>
          <table className="rep-table" style={{ width: '100%', borderCollapse: 'collapse' }}>
            <thead style={{ background: 'rgba(0,0,0,0.2)', color: '#cbd5e1', fontSize: '0.85rem',
textTransform: 'uppercase', letterSpacing: '0.5px' }}>
              <tr>
                <th style={{ padding: '1rem 1.5rem', textAlign: 'left', fontWeight: '700'
}}>Equipo</th>
                <th style={{ padding: '1rem 1.5rem', textAlign: 'left', fontWeight: '700'
}}>Torneo</th>
                <th style={{ padding: '1rem 1.5rem', textAlign: 'left', fontWeight: '700'
}}>Estado</th>
                <th style={{ padding: '1rem 1.5rem', textAlign: 'left', fontWeight: '700'
}}>Fecha</th>
              </tr>
            </thead>
            <tbody>
              {filtered.length > 0 ? (
                filtered.map(i => (
                  <tr key={i.id} style={{ borderBottom: '1px solid rgba(255,255,255,0.05)',
transition: 'background 0.2s' }} className="hover:bg-white/5">
                    <td style={{ padding: '1rem 1.5rem' }}>
                      <div style={{ display: 'flex', alignItems: 'center', gap: '12px' }}>
                        <div style={{ width: '32px', height: '32px', borderRadius: '8px', background:
'rgba(53, 110, 216, 0.2)', color: '#60a5fa', display: 'flex', alignItems: 'center', justifyContent:
'center', fontWeight: '800', fontSize: '0.9rem' }}>
                          {i.equipo?.nombre?.[0]}
                        </div>
                        <span style={{ color: '#fff', fontWeight: 700, fontSize: '0.95rem'
}}>{i.equipo?.nombre}</span>
                      </div>
                    </td>
                    <td style={{ padding: '1rem 1.5rem', color: '#cbd5e1', fontWeight: '500'
}}>{i.torneo?.nombre}</td>
                    <td style={{ padding: '1rem 1.5rem' }}>
                      <span className="badge-pill" style={{
                        background: i.estado === 'Aprobada' ? 'rgba(34, 197, 94, 0.2)' : i.estado ===
'Rechazada' ? 'rgba(239, 68, 68, 0.2)' : 'rgba(59, 130, 246, 0.2)',
                        color: i.estado === 'Aprobada' ? '#86efac' : i.estado === 'Rechazada' ? '#fca5a5'
: '#93c5fd',
                        padding: '0.35rem 0.85rem',
                        borderRadius: '20px',
                        fontSize: '0.75rem',
                        fontWeight: 800,
                        border: '1px solid currentColor'
                      }}>
                        {i.estado}
                      </span>
                    </td>
                    <td style={{ padding: '1rem 1.5rem', color: '#94a3b8', fontSize: '0.9rem'
}}>{formatDate(i.created_at)}</td>
                  </tr>
                ))
              ) : (
                <tr>
                  <td colSpan="5" style={{ textAlign: 'center', padding: '4rem', color: '#64748b'
}}>
```

```
                    <ClipboardList size={48} style={{ opacity: 0.1, color: '#fff', margin: '0 auto
1rem' }} />
                    <p>No se encontraron inscripciones.</p>
                  </td>
                 </tr>
               )}
              </tbody>
             </table>
            </div>
           </div>
         </div>

         {/* RIGHT COLUMN: FILTERS & INFO */}
          <div className="dashboard-col-right" style={{ display: 'flex', flexDirection: 'column',
gap: '1.5rem' }}>

           <div className="rep-card-premium" style={{ ...glassCardStyle, padding: '1.5rem' }}>
            <h2 style={{ fontSize: '1.1rem', fontWeight: 800, color: '#fff', display: 'flex', alignItems:
'center', gap: '0.75rem', marginBottom: '1.5rem' }}>
              <Filter size={20} color="#60a5fa" /> Filtros Rápidos
             </h2>
             <div style={{ display: 'flex', flexDirection: 'column', gap: '1.25rem' }}>
              <div className="form-group">
                <label className="form-label" style={{ display: 'block', marginBottom: '0.5rem',
fontSize: '0.9rem', color: '#cbd5e1', fontWeight: '700' }}>Por Equipo</label>
               <select
                 className="pro-input"
                 style={{ ...inputStyle, width: '100%' }}
                 value={filters.equipo_id}
                 onChange={e => setFilters({ ...filters, equipo_id: e.target.value })}
               >
                            <option value="" style={{ color: '#e2e8f0', background: '#0f172a'
}}>Cualquiera</option>
                 {equipos.map(eq => (
                 <option key={eq.id} value={eq.id} style={{ color: '#e2e8f0', background: '#0f172a'
}}>{eq.nombre}</option>
                ))}
               </select>
              </div>
              <div className="form-group">
                <label className="form-label" style={{ display: 'block', marginBottom: '0.5rem',
fontSize: '0.9rem', color: '#cbd5e1', fontWeight: '700' }}>Estado</label>
               <select
                 className="pro-input"
                 style={{ ...inputStyle, width: '100%' }}
                 value={filters.status}
                 onChange={e => setFilters({ ...filters, status: e.target.value })}
               >
                            <option value="" style={{ color: '#e2e8f0', background: '#0f172a'
}}>Todos</option>
                    <option value="Pendiente" style={{ color: '#e2e8f0', background: '#0f172a'
}}>Pendiente</option>
                    <option value="Aprobada" style={{ color: '#e2e8f0', background: '#0f172a'
}}>Aprobada</option>
                   <option value="Rechazada" style={{ color: '#e2e8f0', background: '#0f172a'
}}>Rechazada</option>
               </select>
              </div>
             </div>
             <button
```

```
            onClick={() => setFilters({ equipo_id: "", status: "", q: "" })}
            className="btn-secondary"
             style={{ width: '100%', marginTop: '1.5rem', background: 'rgba(255,255,255,0.05)',
color: '#cbd5e1', border: '1px solid rgba(255,255,255,0.1)' }}
           >
            Reiniciar Filtros
           </button>
          </div>

           <div className="rep-card-premium" style={{ background: 'linear-gradient(135deg,
#0f172a, #1e293b)', color: 'white', border: '1px solid rgba(255,255,255,0.1)', padding: '1.5rem',
borderRadius: '16px' }}>
          <h2 style={{ fontSize: '1.1rem', fontWeight: 800, color: '#fff', display: 'flex', alignItems:
'center', gap: '0.75rem' }}>
            <CheckCircle2 size={22} color="#10B981" /> Proceso de Validación
           </h2>
          <p style={{ marginTop: '1rem', fontSize: '0.9rem', color: '#cbd5e1', lineHeight: '1.6' }}>
             Las inscripciones son evaluadas por la Dirección de Deportes UEB para asegurar el
cumplimiento del reglamento.
           </p>
           <div style={{ marginTop: '1.5rem', display: 'flex', flexDirection: 'column', gap: '10px'
}}>
              <div style={{ display: 'flex', alignItems: 'center', gap: '10px', fontSize: '0.85rem',
background: 'rgba(255,255,255,0.05)', padding: '10px 12px', borderRadius: '8px', color: '#fff'
}}>
             <Clock size={16} color="#94a3b8" /> <span>Plazo: 48h hábiles</span>
            </div>
              <div style={{ display: 'flex', alignItems: 'center', gap: '10px', fontSize: '0.85rem',
background: 'rgba(255,255,255,0.05)', padding: '10px 12px', borderRadius: '8px', color: '#fff'
}}>
                    <ClipboardList size={16} color="#94a3b8" /> <span>Nómina completa
requerida</span>
            </div>
           </div>
         </div>
       </div>
     </div>
   </div>
  );
}
```

## Nomina:

```
import React, { useEffect, useMemo, useState, useCallback, useRef } from "react";
import {
  Loader2,
  AlertCircle,
  Users,
  RefreshCcw,
  Download,
  Search,
  Plus,
  Trash2,
  X,
  CheckCircle,
  Upload,
  FileSpreadsheet,
  TrendingUp,
  FileText,
```

```
  Edit,
  Shield,
  CloudUpload,
  IdCard,
  Calendar
} from "lucide-react";
import { jsPDF } from "jspdf";
import api from "../../api";
import "../../css/unified-all.css";
import { useSyncAuth } from "../../hooks/useSyncAuth";

import "../../css/unified-all.css";

const POSICIONES_DEPORTE = {
  futbol: [
    'Arquero',
    'Lateral Derecho',
    'Defensa Central 1',
    'Defensa Central 2',
    'Lateral Izquierdo',
    'Volante de Marca',
    'Volante Mixto',
    'Volante Creativo',
    'Extremo Derecho',
    'Extremo Izquierdo',
    'Centrodelantero'
  ],
  basquet: [
    'Base (Armador)',
    'Escolta',
    'Alero',
    'Ala-Pívot',
    'Pívot'
  ],
  voley: [
    'Armador',
    'Rematador',
    'Central',
    'Líbero',
    'Zaguero'
  ]
};

const DEPORTES_INFO = [
  { key: 'futbol', label: '⚽ Fútbol' },
  { key: 'basquet', label: '🏀 Básquet' },
  { key: 'voley', label: '🏐 Vóley' }
];

export default function NominaRepresentante() {
  const { user } = useSyncAuth();
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState("");
  const [success, setSuccess] = useState("");

  const [equipos, setEquipos] = useState([]);
  const [equipoId, setEquipoId] = useState("");

  const [nominaList, setNominaList] = useState([]);
  const [q, setQ] = useState("");
```

```
  const [sortColumn, setSortColumn] = useState(null);
  const [sortDirection, setSortDirection] = useState("asc");
  const [currentPage, setCurrentPage] = useState(1);
  const [rowsPerPage, setRowsPerPage] = useState(10);

  // ---- Modal Agregar ----
  const [showModal, setShowModal] = useState(false);
  const [savingJugador, setSavingJugador] = useState(false);
  const [formJugador, setFormJugador] = useState({ equipo_id: "", cedula: "", numero: "",
posicion: "", nombres: "", apellidos: "" });
  const [isEditing, setIsEditing] = useState(false);
  const [currentJugador, setCurrentJugador] = useState(null);

  // ---- Importación ----
  const fileRef = useRef(null);
  const [importing, setImporting] = useState(false);
  const [importResult, setImportResult] = useState(null);
  const [showImportSummary, setShowImportSummary] = useState(false);

  // 1. Cargar Equipos
  const loadEquipos = useCallback(async () => {
    try {
      const { data } = await api.get('/representante/equipos');
      setEquipos(data || []);
      // Eliminada lógica de auto-selección para permitir ver "Todos los Equipos" por defecto
    } catch (err) {
      console.error("Error loading equipos", err);
    }
  }, []);

  // 2. Cargar Nómina
  const loadNomina = useCallback(async () => {
    setLoading(true);
    setError("");
    try {
      let url = '/representante/equipo/jugadores/nomina';
      if (equipoId) {
        url += `?equipo_id=${equipoId}`;
      }
      const { data } = await api.get(url);

      let players = [];
      if (data.equipos) {
        data.equipos.forEach(eq => {
          if (eq.jugadores) {
            eq.jugadores.forEach(j => {
              players.push({
                ...j,
                equipo_nombre: eq.nombre,
                equipo_id: eq.id
              });
            });
          }
        });
      } else if (Array.isArray(data)) {
        players = data;
      }

      setNominaList(players);
    } catch (err) {
```

```
    console.error("Error loading nomina", err);
  } finally {
    setLoading(false);
  }
}, [equipoId]);

useEffect(() => {
  loadEquipos();
}, [loadEquipos]);

useEffect(() => {
  loadNomina();
}, [loadNomina]);

// ---- Derived State for Table ----
const filteredRows = useMemo(() => {
  let rows = nominaList;
  const term = q.toLowerCase().trim();
  if (term) {
    rows = rows.filter(r =>
      (r.persona?.nombres || "").toLowerCase().includes(term) ||
      (r.persona?.apellidos || "").toLowerCase().includes(term) ||
      (r.cedula || "").includes(term) ||
      (r.equipo_nombre || "").toLowerCase().includes(term)
    );
  }

  if (sortColumn) {
    rows = [...rows].sort((a, b) => {
      let aVal = a[sortColumn];
      let bVal = b[sortColumn];

      if (sortColumn === 'completo') {
        aVal = `${a.persona?.nombres} ${a.persona?.apellidos}`;
        bVal = `${b.persona?.nombres} ${b.persona?.apellidos}`;
      }

      if (aVal === null || aVal === undefined) return sortDirection === "asc" ? 1 : -1;
      if (bVal === null || bVal === undefined) return sortDirection === "asc" ? -1 : 1;

      if (typeof aVal === "string") {
       return sortDirection === "asc" ? aVal.localeCompare(bVal) : bVal.localeCompare(aVal);
      }
      return sortDirection === "asc" ? aVal - bVal : bVal - aVal;
    });
  }

  return rows;
}, [nominaList, q, sortColumn, sortDirection]);

const { paginatedRows, totalPages } = useMemo(() => {
  const total = filteredRows.length;
  const pages = Math.ceil(total / rowsPerPage);
  const startIndex = (currentPage - 1) * rowsPerPage;
  const paginated = filteredRows.slice(startIndex, startIndex + rowsPerPage);
  return { paginatedRows: paginated, totalPages: pages };
}, [filteredRows, currentPage, rowsPerPage]);

const kpis = useMemo(() => {
  const totalJugadores = nominaList.length;
```

```
    const equiposCount = new Set(nominaList.map(j => j.equipo_id)).size;
    return { totalJugadores, equiposCount };
  }, [nominaList]);

  // ---- Handlers ----

  const handleSort = (column) => {
    if (sortColumn === column) {
      setSortDirection(sortDirection === "asc" ? "desc" : "asc");
    } else {
      setSortColumn(column);
      setSortDirection("asc");
    }
  };

  const handleModalOpen = (isEdit = false, player = null) => {
    setIsEditing(isEdit);
    setCurrentJugador(player);
    if (isEdit && player) {
      setFormJugador({
        equipo_id: player.equipo_id,
        cedula: player.cedula,
        numero: player.numero || "",
        posicion: player.posicion || "",
        nombres: player.persona?.nombres || "",
        apellidos: player.persona?.apellidos || ""
      });
    } else {
      setFormJugador({ equipo_id: equipoId || "", cedula: "", numero: "", posicion: "", nombres:
"", apellidos: "" });
    }
    setShowModal(true);
  };

  const handleSubmitJugador = async (e) => {
    e.preventDefault();
    setSavingJugador(true);
    setSuccess("");
    setError("");

    try {
      const payload = {
        ...formJugador,
        create_persona: !isEditing,
        persona: {
          nombres: formJugador.nombres,
          apellidos: formJugador.apellidos
        }
      };

      await api.post('/representante/equipo/jugadores', payload);
      setSuccess(isEditing ? "Ficha del jugador actualizada." : "Jugador agregado a la nómina.");

      setShowModal(false);
      loadNomina();
    } catch (err) {
      console.error(err);
      setError("Error al guardar. Verifique los datos o si la cédula ya existe en otro equipo.");
    } finally {
      setSavingJugador(false);
```

```
    }
  };

  const handleRemoveJugador = async (cedula) => {
    if (!confirm("¿Estás seguro de quitar a este jugador de la nómina?")) return;
    try {
      await api.delete(`/representante/equipo/jugadores/${cedula}`);
      setSuccess("Jugador eliminado de la nómina.");
      loadNomina();
    } catch (err) {
      console.error(err);
      setError("No se pudo eliminar el jugador.");
    }
  };

  const handleDownloadCarnet = async (cedula) => {
    try {
      const response = await api.get(`/jugadores/${cedula}/carnet-pdf`, { responseType: 'blob' });
      const url = window.URL.createObjectURL(new Blob([response.data]));
      const link = document.createElement('a');
      link.href = url;
      link.setAttribute('download', `carnet_${cedula}.pdf`);
      document.body.appendChild(link);
      link.click();
      link.remove();
    } catch (err) {
      console.error(err);
      setError("No se pudo descargar el carnet. Verifique si el jugador tiene foto.");
    }
  };

  const handleImportFile = async (file) => {
    if (!file) return;

    if (!equipoId) {
      setError("Debes seleccionar un equipo específico en el filtro superior antes de subir la
nómina.");
      if (fileRef.current) fileRef.current.value = "";
      return;
    }

    setImporting(true);
    setError("");
    setSuccess("");

    const formData = new FormData();
    formData.append('file', file);
    formData.append('equipo_id', equipoId);

    try {
      const { data } = await api.post('/representante/equipo/jugadores/import', formData, {
        headers: { 'Content-Type': 'multipart/form-data' }
      });

      if (data.errors && data.errors.length > 0) {
        setImportResult(data);
        setShowImportSummary(true);
        setError(`Importación completada con ${data.errors.length} errores.`);
      } else {
```

```
            setSuccess(`Éxito: Se importaron/actualizaron ${data.inserted + data.updated} deportistas.`);
      }
      loadNomina();
    } catch (err) {
      console.error(err);
      const msg = err.response?.data?.message || "Error en la importación del archivo.";
      setError(msg);
    } finally {
      setImporting(false);
      if (fileRef.current) fileRef.current.value = "";
    }
  };

  // CLIENT-SIDE TEMPLATE GENERATION
  const handleDownloadTemplate = () => {
    try {
      const headers = ['cedula', 'nombres', 'apellidos', 'numero', 'posicion', 'email', 'telefono'];
      const csvContent = headers.join(',') + "\n";

      const blob = new Blob([csvContent], { type: 'text/csv;charset=utf-8;' });
      const url = URL.createObjectURL(blob);
      const link = document.createElement('a');
      link.href = url;
      link.setAttribute('download', 'plantilla_jugadores.csv'); // standard logical name
      document.body.appendChild(link);
      link.click();
      link.remove();
    } catch (err) {
      setError("No se pudo generar la plantilla.");
    }
  };

  // CLIENT-SIDE EXPORT PDF/CSV
  const handleExportNomina = (format) => {
    if (nominaList.length === 0) {
      setError("No hay datos para exportar.");
      return;
    }

    const timestamp = new Date().toISOString().slice(0, 10);
    const filename = `nomina_${equipoId || 'general'}_${timestamp}`;

    if (format === 'csv') {
      const headers = ['Cedula', 'Nombres', 'Apellidos', 'Numero', 'Posicion', 'Equipo'];
      const escapeCsv = (field) => {
        const str = String(field || '');
        if (str.includes(',')) return `"${str}"`;
        return str;
      };

      const rows = nominaList.map(p => [
        escapeCsv(p.cedula),
        escapeCsv(p.persona?.nombres),
        escapeCsv(p.persona?.apellidos),
        escapeCsv(p.numero),
        escapeCsv(p.posicion),
        escapeCsv(p.equipo_nombre)
      ]);
```

```
    const csvContent = [
      headers.join(','),
      ...rows.map(r => r.join(','))
    ].join('\n');

    const blob = new Blob([csvContent], { type: 'text/csv;charset=utf-8;' });
    const url = URL.createObjectURL(blob);
    const link = document.createElement('a');
    link.href = url;
    link.setAttribute('download', `${filename}.csv`);
    document.body.appendChild(link);
    link.click();
    link.remove();
    setSuccess("Exportación CSV completada.");

  } else if (format === 'pdf') {
    try {
      const doc = new jsPDF();

      // Title
      doc.setFontSize(18);
      doc.setTextColor(40);
      doc.text("Reporte de Nómina", 14, 22);

      // Metadata
      doc.setFontSize(11);
      doc.setTextColor(100);
      const dateStr = new Date().toLocaleDateString();
      const timeStr = new Date().toLocaleTimeString();
      doc.text(`Fecha de emisión: ${dateStr} ${timeStr}`, 14, 30);
      doc.text(`Total Jugadores: ${nominaList.length}`, 14, 36);

      // Setup Table Header
      let y = 45;
      const lineHeight = 10;
      const pageHeight = doc.internal.pageSize.height;

      const colX = [14, 40, 110, 130, 170]; // X positions for columns
      const colTitles = ["Cédula", "Nombre Completo", "Numero", "Posición", "Equipo"];

      // Draw Header Function
      const drawHeader = (posY) => {
        doc.setFillColor(240, 240, 240); // Light gray bg
        doc.rect(10, posY - 6, 190, 8, 'F');
        doc.setFont(undefined, 'bold');
        doc.setTextColor(0);
        colTitles.forEach((title, i) => doc.text(title, colX[i], posY));
        doc.line(10, posY + 2, 200, posY + 2);
      };

      drawHeader(y);
      y += lineHeight;

      doc.setFont(undefined, 'normal');
      doc.setFontSize(10);
      doc.setTextColor(50);

      nominaList.forEach((p) => {
        // Check page break
        if (y + lineHeight > pageHeight - 15) {
```

```
          doc.addPage();
          y = 20;
          drawHeader(y);
          y += lineHeight;
          doc.setTextColor(50); // Reset color
          doc.setFont(undefined, 'normal');
        }

        const nombre = `${p.persona?.nombres || ''} ${p.persona?.apellidos || ''}`.substring(0, 35);

        doc.text(p.cedula || "-", colX[0], y);
        doc.text(nombre, colX[1], y);
        doc.text(String(p.numero || "-"), colX[2], y);
        doc.text((p.posicion || "-").substring(0, 15), colX[3], y);
        doc.text((p.equipo_nombre || "-").substring(0, 15), colX[4], y);

        y += 8; // Row height
      });

      doc.save(`${filename}.pdf`);
      setSuccess("PDF generado correctamente.");
    } catch (err) {
      console.error("PDF Generation Error", err);
      setError("Error al generar el PDF.");
    }
  }
};

if (loading && nominaList.length === 0) {
  return (
        <div className="rep-scope rep-screen-container rep-loading-container" style={{
flexDirection: 'column', gap: '1.5rem', justifyContent: 'center', alignItems: 'center' }}>
      <div style={{ display: 'flex', alignItems: 'center', gap: '12px' }}>
        <RefreshCcw size={24} className="animate-spin" color="var(--primary-ocean)" />
          <p style={{ margin: 0, fontWeight: 700, color: 'var(--text-muted)' }}>Sincronizando
nómina...</p>
      </div>
    </div>
  );
}

// --- STYLES FOR DARK THEME COMPATIBILITY ---
const glassCardStyle = {
  background: 'rgba(15, 23, 42, 0.6)',
  backdropFilter: 'blur(12px)',
  border: '1px solid rgba(255,255,255,0.08)',
  boxShadow: '0 8px 32px rgba(0,0,0,0.2)',
  color: '#f8fafc'
};

const inputStyle = {
  background: 'rgba(0, 0, 0, 0.2)',
  border: '1px solid rgba(255,255,255,0.1)',
  color: '#fff',
  borderRadius: '8px',
  boxSizing: 'border-box'
};

return (
  <div className="rep-scope rep-screen-container rep-dashboard-fade">
```

```
    {/* HEADER */}
     <header className="rep-header-main page-header-responsive" style={{ marginBottom: '2.5rem' }}>
      <div className="header-info">
        <small className="university-label" style={{ fontWeight: '700', letterSpacing: '0.5px', color: 'var(--accent-teal)' }}>Gestión de Talento</small>
            <h1 className="content-title" style={{ color: '#fff', textShadow: '0 2px 10px rgba(0,0,0,0.5)' }}>Nómina de Jugadores</h1>
             <p className="content-subtitle" style={{ color: '#cbd5e1' }}>Administración centralizada de deportistas y cuerpos técnicos</p>
      </div>
       <div className="header-actions" style={{ width: 'auto', display: 'inline-flex', flex: '0 0 auto' }}>
        <button
          onClick={() => handleModalOpen(false)}
          className="pro-btn btn-primary"
           style={{ boxShadow: 'var(--shadow-primary-button)', padding: '0.38rem 0.55rem', fontSize: '0.78rem', minHeight: '32px', width: 'fit-content', maxWidth: 'fit-content', flex: '0 0 auto', display: 'inline-flex', whiteSpace: 'nowrap' }}
        >
          <Plus size={13} /> Registrar Jugador
        </button>
       </div>
     </header>

     {/* ALERTS */}
     {error && (
       <div style={{ background: 'rgba(239, 68, 68, 0.1)', border: '1px solid rgba(239, 68, 68, 0.2)', color: '#fca5a5', padding: '1rem', borderRadius: '12px', marginBottom: '1.5rem', display: 'flex', alignItems: 'center', gap: '0.5rem', fontWeight: '500' }}>
        <AlertCircle size={20} /> {error}
       </div>
     )}
     {success && (
       <div style={{ background: 'rgba(34, 197, 94, 0.1)', border: '1px solid rgba(34, 197, 94, 0.2)', color: '#86efac', padding: '1rem', borderRadius: '12px', marginBottom: '1.5rem', display: 'flex', alignItems: 'center', gap: '0.5rem', fontWeight: '500' }}>
        <CheckCircle size={20} /> {success}
       </div>
     )}

     {/* KPI GRID */}
       <div className="mb-8" style={{ marginBottom: '0.9rem', display: 'flex', flexWrap: 'nowrap', gap: '0.6rem', justifyContent: 'center', overflowX: 'auto' }}>
        <div className="stat-card rep-card-premium stat-card-primary-ocean-border" style={{ ...glassCardStyle, padding: '0.58rem 0.75rem', height: '86px', minHeight: '86px', aspectRatio: 'auto', flex: '0 0 154px', width: '154px' }}>
         <div className="stat-card-header">
           <h3 style={{ color: '#cbd5e1', fontWeight: '700', fontSize: '0.79rem', lineHeight: 1.1 }}>Total Deportistas</h3>
           <Users size={14} color="var(--primary-ocean)" />
         </div>
         <p className="stat-value" style={{ color: '#fff', fontWeight: '900', fontSize: '1.34rem', lineHeight: 1.05, margin: '0.1rem 0' }}>{kpis.totalJugadores}</p>
            <p className="stat-desc" style={{ color: '#94a3b8', fontWeight: '500', fontSize: '0.68rem', margin: 0, lineHeight: 1.15 }}>Histórico general</p>
       </div>
```

```
          <div className="stat-card rep-card-premium stat-card-accent-teal-border" style={{ ...glassCardStyle, padding: '0.58rem 0.75rem', height: '86px', minHeight: '86px', aspectRatio: 'auto', flex: '0 0 154px', width: '154px' }}>
            <div className="stat-card-header">
              <h3 style={{ color: '#cbd5e1', fontWeight: '700', fontSize: '0.79rem', lineHeight: 1.1 }}>Equipos Activos</h3>
              <Shield size={14} color="var(--accent-teal)" />
            </div>
            <p className="stat-value" style={{ color: '#fff', fontWeight: '900', fontSize: '1.34rem', lineHeight: 1.05, margin: '0.1rem 0' }}>{kpis.equiposCount}</p>
              <p className="stat-desc" style={{ color: '#94a3b8', fontWeight: '500', fontSize: '0.68rem', margin: 0, lineHeight: 1.15 }}>Torneo actual</p>
          </div>

          <div className="stat-card rep-card-premium" style={{ ...glassCardStyle, borderTop: '4px solid #f59e0b', padding: '0.58rem 0.75rem', height: '86px', minHeight: '86px', aspectRatio: 'auto', flex: '0 0 154px', width: '154px' }}>
            <div className="stat-card-header">
              <h3 style={{ color: '#cbd5e1', fontWeight: '700', fontSize: '0.79rem', lineHeight: 1.1 }}>Estado Documental</h3>
              <FileText size={14} color="#f59e0b" />
            </div>
            <p className="stat-value" style={{ color: '#fff', fontWeight: '900', fontSize: '1.34rem', lineHeight: 1.05, margin: '0.1rem 0' }}>98%</p>
              <p className="stat-desc" style={{ color: '#94a3b8', fontWeight: '500', fontSize: '0.62rem', margin: 0, lineHeight: 1.1 }}>De deportistas registrados</p>
                <div className="perf-bar-bg mt-3" style={{ height: '3px', background: 'rgba(255,255,255,0.1)', marginTop: '0.15rem' }}>
                  <div className="perf-bar-fill" style={{ width: `98%`, background: '#f59e0b', borderRadius: '4px' }}></div>
            </div>
            <div style={{ marginTop: '0.4rem', color: '#fcd34d', fontSize: '0.55rem', fontWeight: 600, display: 'flex', alignItems: 'center', gap: '4px' }}>
              <AlertCircle size={10} /> Tienes jugadores con documentación pendiente.
            </div>
          </div>
        </div>

        {/* MAIN CONTENT */}
          <div className="responsive-grid" style={{ gap: '2rem', display: 'grid', gridTemplateColumns: 'repeat(auto-fit, minmax(320px, 1fr))' }}>
          <div className="dashboard-col-left">
            <div className="rep-card-premium" style={{ padding: '0', overflow: 'hidden', ...glassCardStyle }}>
            {/* TOOLBAR */}
            <div style={{ padding: '1.5rem', borderBottom: '1px solid rgba(255,255,255,0.08)' }}>
              <div style={{ display: 'flex', justifyContent: 'space-between', alignItems: 'center', flexWrap: 'wrap', gap: '1.5rem' }}>
              <div style={{ display: 'flex', alignItems: 'center', gap: '1rem' }}>
                <h2 style={{ fontSize: '1.25rem', fontWeight: 800, color: '#fff', display: 'flex', alignItems: 'center', gap: '0.5rem' }}>
                  <Users size={20} color="var(--primary-ocean)" fill="rgba(var(--primary-rgb), 0.1)" /> Gestión de Nómina
                </h2>
                <span className="hide-mobile" style={{ height: '24px', width: '1px', background: 'rgba(255,255,255,0.1)' }}></span>
                <div className="hide-mobile" style={{ color: '#94a3b8', fontSize: '1rem' }}>
                  <strong>{filteredRows.length}</strong> resultados
                </div>
              </div>
```

```
            <div className="flex-mobile-col" style={{ display: 'flex', gap: '0.75rem', width: '100%', maxWidth: '100%', flex: 1, justifyContent: 'flex-end' }}>
                <div className="search-box" style={{ ...inputStyle, padding: '0.5rem 1rem', display: 'flex', alignItems: 'center', gap: '0.5rem', flex: 1, minWidth: '200px' }}>
              <Search size={18} color="#94a3b8" />
              <input
                placeholder="Buscar por nombre o identificación..."
                value={q}
                onChange={e => setQ(e.target.value)}
                  style={{ border: 'none', background: 'transparent', fontSize: '0.9rem', outline: 'none', width: '100%', color: '#fff' }}
              />
            </div>
            <select
              value={equipoId}
              onChange={e => setEquipoId(e.target.value)}
              className="pro-input"
            style={{ ...inputStyle, padding: '0.5rem 1rem', fontSize: '0.9rem', fontWeight: '500', width: 'auto', minWidth: '180px' }}
            >
              <option value="" style={{ color: '#333' }}>Filtrar por Equipo</option>
                {equipos.map(eq => <option key={eq.id} value={eq.id} style={{ color: '#333' }}>{eq.nombre}</option>)}
            </select>
          </div>
        </div>
      </div>

      {/* TABLE */}
      <div className="table-responsive-wrapper">
        <table className="rep-table" style={{ width: '100%', borderCollapse: 'collapse' }}>
          <thead style={{ background: 'rgba(0,0,0,0.2)', color: '#cbd5e1', fontSize: '0.85rem', textTransform: 'uppercase', letterSpacing: '0.5px' }}>
            <tr>
              <th onClick={() => handleSort('completo')} style={{ padding: '1rem 1.5rem', textAlign: 'left', cursor: 'pointer', fontWeight: '700' }}>Jugador</th>
                <th onClick={() => handleSort('cedula')} className="hide-mobile" style={{ padding: '1rem 1.5rem', textAlign: 'left', cursor: 'pointer', fontWeight: '700' }}>Identificación</th>
                <th onClick={() => handleSort('numero')} style={{ padding: '1rem 1.5rem', textAlign: 'left', cursor: 'pointer', fontWeight: '700' }}>Numero</th>
              <th onClick={() => handleSort('posicion')} className="hide-mobile" style={{ padding: '1rem 1.5rem', textAlign: 'left', cursor: 'pointer', fontWeight: '700' }}>Posición</th>
                <th onClick={() => handleSort('equipo')} className="hide-mobile" style={{ padding: '1rem 1.5rem', textAlign: 'left', cursor: 'pointer', fontWeight: '700' }}>Equipo</th>
                  <th style={{ padding: '1rem 1.5rem', textAlign: 'right', fontWeight: '700' }}>Acciones</th>
            </tr>
          </thead>
          <tbody>
            {paginatedRows.length > 0 ? (
              paginatedRows.map((r) => (
                          <tr key={r.id || r.cedula} style={{ borderBottom: '1px solid rgba(255,255,255,0.05)', transition: 'background 0.2s' }} className="hover:bg-white/5">
                <td style={{ padding: '1rem 1.5rem' }}>
                  <div style={{ display: 'flex', alignItems: 'center', gap: '1rem' }}>
                    <div style={{ width: '40px', height: '40px', borderRadius: '12px', background: 'rgba(53, 110, 216, 0.2)', color: '#60a5fa', display: 'flex', alignItems: 'center', justifyContent: 'center', fontWeight: '900', fontSize: '1rem', border: '1px solid rgba(53, 110, 216, 0.3)' }}>
```

```
                        {r.persona?.nombres?.charAt(0)}
                      </div>
                      <div>
                        <div style={{ display: 'flex', flexDirection: 'column' }}>
                                <div style={{ fontWeight: '800', color: '#fff', fontSize: '0.95rem',
marginBottom: '2px' }}>{r.persona?.nombres} {r.persona?.apellidos}</div>
                          <div style={{ display: 'flex', alignItems: 'center', gap: '4px' }}>
                            <Calendar size={10} color="#94a3b8" />
                              <span style={{ color: '#94a3b8', fontWeight: '600', fontSize: '0.75rem',
textTransform: 'uppercase', letterSpacing: '0.3px' }}>
                                {r.torneo_nombre || 'Torneo 2024'}
                              </span>
                            </div>
                          </div>
                        </div>
                      </td>
                    <td className="hide-mobile" style={{ padding: '1rem 1.5rem', color: '#cbd5e1',
fontWeight: '600' }}>{r.cedula}</td>
                      <td style={{ padding: '1rem 1.5rem' }}>
                       <span style={{ display: 'inline-block', padding: '0.35rem 0.85rem', background:
'rgba(255,255,255,0.1)', border: '1px solid rgba(255,255,255,0.1)', borderRadius: '8px',
fontSize: '0.9rem', fontWeight: '800', color: '#fff' }}>
                          {r.numero || '-'}
                        </span>
                      </td>
                    <td className="hide-mobile" style={{ padding: '1rem 1.5rem', color: '#cbd5e1',
fontWeight: '600' }}>{r.posicion}</td>
                      <td className="hide-mobile" style={{ padding: '1rem 1.5rem' }}>
                        <span style={{ display: 'inline-block', padding: '0.35rem 1rem', background:
'rgba(53, 110, 216, 0.15)', color: '#60a5fa', borderRadius: '20px', fontSize: '0.8rem', fontWeight:
'800', border: '1px solid rgba(53, 110, 216, 0.3)' }}>
                          {r.equipo_nombre}
                        </span>
                      </td>
                      <td style={{ padding: '1rem 1.5rem', textAlign: 'right' }}>
                        <div style={{ display: 'flex', justifyContent: 'flex-end', gap: '0.75rem' }}>
                          <button onClick={() => handleDownloadCarnet(r.cedula)} title="Descargar
Carnet" style={{ padding: '0.65rem', borderRadius: '10px', color: '#a78bfa', border: 'none',
background: 'rgba(139, 92, 246, 0.1)', cursor: 'pointer', transition: 'all 0.2s' }}
className="hover:bg-purple-500/20 hover:text-purple-300">
                            <IdCard size={18} />
                          </button>
                          <button onClick={() => handleModalOpen(true, r)} title="Editar" style={{
padding: '0.65rem', borderRadius: '10px', color: '#94a3b8', border: 'none', background:
'rgba(255,255,255,0.05)', cursor: 'pointer', transition: 'all 0.2s' }} className="hover:bg-
white/10 hover:text-white">
                            <Edit size={18} />
                          </button>
                          <button onClick={() => handleRemoveJugador(r.cedula)} title="Eliminar"
style={{ padding: '0.65rem', borderRadius: '10px', color: '#fca5a5', border: 'none', background:
'rgba(239, 68, 68, 0.1)', cursor: 'pointer', transition: 'all 0.2s' }} className="hover:bg-red-
500/20">
                            <Trash2 size={18} />
                          </button>
                        </div>
                      </td>
                    </tr>
                  ))
                ) : (
```

```
                <tr>
                  <td colSpan="6" style={{ padding: '5rem', textAlign: 'center', color: '#64748b' }}>
                    <div style={{ display: 'flex', flexDirection: 'column', alignItems: 'center', gap: '1rem' }}>
                      <Users size={56} style={{ opacity: 0.1, color: '#fff' }} />
                      <p style={{ fontWeight: '600', fontSize: '1.1rem', color: '#94a3b8' }}>No se encontraron jugadores en la nómina.</p>
                    </div>
                  </td>
                </tr>
              )}
            </tbody>
          </table>
        </div>

        {/* PAGINATION */}
        <div style={{ padding: '1rem 1.5rem', borderTop: '1px solid rgba(255,255,255,0.08)', display: 'flex', justifyContent: 'space-between', alignItems: 'center' }}>
          <button onClick={() => setCurrentPage(c => Math.max(1, c - 1))} disabled={currentPage === 1} className="btn-secondary" style={{ padding: '0.6rem 1.25rem', fontSize: '0.9rem', color: '#cbd5e1', fontWeight: '600', background: 'transparent', borderColor: 'rgba(255,255,255,0.1)' }}>Anterior</button>
          <span style={{ fontSize: '0.95rem', color: '#94a3b8' }}>Página <strong>{currentPage}</strong> de <strong>{totalPages}</strong></span>
          <button onClick={() => setCurrentPage(c => Math.min(totalPages, c + 1))} disabled={currentPage === totalPages} className="btn-secondary" style={{ padding: '0.6rem 1.25rem', fontSize: '0.9rem', color: '#cbd5e1', fontWeight: '600', background: 'transparent', borderColor: 'rgba(255,255,255,0.1)' }}>Siguiente</button>
        </div>
      </div>
    </div>

    <div className="dashboard-col-right" style={{ display: 'flex', flexDirection: 'column', gap: '1.5rem' }}>
      {/* ACTIONS PANEL */}
      <div className="rep-card-premium" style={{ padding: '1.5rem', ...glassCardStyle }}>
        <h3 style={{ fontSize: '1.1rem', fontWeight: 800, color: '#fff', marginBottom: '1.5rem', display: 'flex', alignItems: 'center', gap: '0.5rem' }}>
          <RefreshCcw size={20} color="var(--primary-ocean)" /> Acciones Rápidas
        </h3>

        <div style={{ display: 'grid', gap: '1rem' }}>
          <div style={{ padding: '1.25rem', background: 'rgba(0,0,0,0.2)', borderRadius: '16px', border: '1px dashed rgba(255,255,255,0.1)' }}>
            <div style={{ display: 'flex', alignItems: 'center', gap: '0.5rem', marginBottom: '0.75rem', color: '#cbd5e1', fontWeight: '700', fontSize: '0.95rem' }}>
              <FileSpreadsheet size={18} /> Importar CSV
            </div>
            <input type="file" ref={fileRef} accept=".csv, .xlsx, .xls" className="hidden" style={{ display: 'none' }} onChange={e => handleImportFile(e.target.files[0])} />
            <button onClick={() => fileRef.current?.click()} className="btn-primary" style={{ width: '100%', fontSize: '0.9rem', padding: '0.75rem', marginBottom: '0.75rem', fontWeight: '600' }}>
              {importing ? <Loader2 className="animate-spin" size={18} /> : "Subir Archivo"}
            </button>
            <button onClick={handleDownloadTemplate} className="btn-secondary" style={{ width: '100%', fontSize: '0.9rem', padding: '0.75rem', background: 'transparent', border: '1px solid rgba(255,255,255,0.2)', color: '#cbd5e1', fontWeight: '600' }}>
              Descargar Plantilla
```

```
            </button>
          </div>

          <div style={{ padding: '1.25rem', background: 'rgba(0,0,0,0.2)', borderRadius: '16px', border: '1px dashed rgba(255,255,255,0.1)' }}>
            <div style={{ display: 'flex', alignItems: 'center', gap: '0.5rem', marginBottom: '0.75rem', color: '#cbd5e1', fontWeight: '700', fontSize: '0.95rem' }}>
              <Download size={18} /> Exportar Data
            </div>
            <div style={{ display: 'flex', gap: '0.75rem' }}>
              <button onClick={() => handleExportNomina('csv')} className="btn-secondary" style={{ flex: 1, fontSize: '0.85rem', padding: '0.6rem', background: 'transparent', border: '1px solid rgba(255,255,255,0.2)', color: '#cbd5e1', fontWeight: '600' }}>CSV</button>
              <button onClick={() => handleExportNomina('pdf')} className="btn-secondary" style={{ flex: 1, fontSize: '0.85rem', padding: '0.6rem', background: 'transparent', border: '1px solid rgba(255,255,255,0.2)', color: '#cbd5e1', fontWeight: '600' }}>PDF</button>
            </div>
          </div>
        </div>
      </div>
    </div>

    {/* MODAL */}
    {showModal && (
      <div style={{ position: 'fixed', top: 0, left: 0, width: '100%', height: '100%', background: 'rgba(15, 23, 42, 0.7)', backdropFilter: 'blur(5px)', display: 'flex', alignItems: 'center', justifyContent: 'center', zIndex: 50 }}>
        <div style={{ background: '#0f172a', borderRadius: '24px', width: '90%', maxWidth: '600px', boxShadow: '0 25px 50px -12px rgba(0, 0, 0, 0.5)', overflow: 'hidden', animation: 'fadeIn 0.3s ease-out', border: '1px solid rgba(255,255,255,0.1)' }}>
          <div style={{ padding: '1.75rem 2rem', borderBottom: '1px solid rgba(255,255,255,0.1)', display: 'flex', justifyContent: 'space-between', alignItems: 'center' }}>
            <h2 style={{ fontSize: '1.5rem', fontWeight: 800, color: '#fff' }}>{isEditing ? 'Actualizar Ficha' : 'Nuevo Jugador'}</h2>
            <button onClick={() => setShowModal(false)} style={{ background: 'rgba(255,255,255,0.1)', border: '1px solid rgba(255,255,255,0.1)', padding: '0.5rem', borderRadius: '50%', cursor: 'pointer', color: '#cbd5e1' }}><X size={20} /></button>
          </div>
          <form onSubmit={handleSubmitJugador} style={{ padding: '2rem' }}>
            <div className="flex-mobile-col" style={{ display: 'grid', gridTemplateColumns: 'repeat(auto-fit, minmax(200px, 1fr))', gap: '1.5rem', marginBottom: '1.5rem' }}>
              <div className="form-group">
                <label className="form-label" style={{ display: 'block', marginBottom: '0.5rem', fontSize: '0.95rem', color: '#cbd5e1', fontWeight: '700' }}>Equipo</label>
                <select className="pro-input" value={formJugador.equipo_id} onChange={e => setFormJugador({ ...formJugador, equipo_id: e.target.value })} style={{ width: '100%', padding: '0.85rem', borderRadius: '10px', border: '1px solid rgba(255,255,255,0.2)', background: '#1e293b', color: '#fff', fontWeight: '500' }} required disabled={isEditing}>
                  <option value="">Seleccione...</option>
                  {equipos.map(eq => <option key={eq.id} value={eq.id} style={{ color: '#fff', background: '#1e293b' }}>{eq.nombre}</option>)}
                </select>
              </div>
              <div className="form-group">
                <label className="form-label" style={{ display: 'block', marginBottom: '0.5rem', fontSize: '0.95rem', color: '#cbd5e1', fontWeight: '700' }}>Cédula</label>
                <input className="pro-input" value={formJugador.cedula} onChange={e => setFormJugador({ ...formJugador, cedula: e.target.value })} style={{ width: '100%', padding:
```

```
'0.85rem', borderRadius: '10px', border: '1px solid rgba(255,255,255,0.2)', background: '#1e293b', color: '#fff', fontWeight: '500' }} placeholder="ID Nacional" required disabled={isEditing} />
              </div>
            </div>

            {!isEditing && (
              <div className="flex-mobile-col" style={{ display: 'grid', gridTemplateColumns: 'repeat(auto-fit, minmax(200px, 1fr))', gap: '1.5rem', marginBottom: '1.5rem' }}>
                <div className="form-group">
                  <label className="form-label" style={{ display: 'block', marginBottom: '0.5rem', fontSize: '0.95rem', color: '#cbd5e1', fontWeight: '700' }}>Nombres</label>
                  <input className="pro-input" value={formJugador.nombres} onChange={e => setFormJugador({ ...formJugador, nombres: e.target.value })} style={{ width: '100%', padding: '0.85rem', borderRadius: '10px', border: '1px solid rgba(255,255,255,0.2)', background: '#1e293b', color: '#fff', fontWeight: '500' }} placeholder="Ej. Juan Carlos" required={!isEditing} />
                </div>
                <div className="form-group">
                  <label className="form-label" style={{ display: 'block', marginBottom: '0.5rem', fontSize: '0.95rem', color: '#cbd5e1', fontWeight: '700' }}>Apellidos</label>
                  <input className="pro-input" value={formJugador.apellidos} onChange={e => setFormJugador({ ...formJugador, apellidos: e.target.value })} style={{ width: '100%', padding: '0.85rem', borderRadius: '10px', border: '1px solid rgba(255,255,255,0.2)', background: '#1e293b', color: '#fff', fontWeight: '500' }} placeholder="Ej. Pérez Gómez" required={!isEditing} />
                </div>
              </div>
            )}

            <div className="flex-mobile-col" style={{ display: 'grid', gridTemplateColumns: 'repeat(auto-fit, minmax(200px, 1fr))', gap: '1.5rem', marginBottom: '1.5rem' }}>
              <div className="form-group">
                <label className="form-label" style={{ display: 'block', marginBottom: '0.5rem', fontSize: '0.95rem', color: '#cbd5e1', fontWeight: '700' }}>Numero</label>
                <input type="number" className="pro-input" value={formJugador.numero} onChange={e => setFormJugador({ ...formJugador, numero: e.target.value })} style={{ width: '100%', padding: '0.85rem', borderRadius: '10px', border: '1px solid rgba(255,255,255,0.2)', background: '#1e293b', color: '#fff', fontWeight: '500' }} placeholder="#" />
              </div>
              <div className="form-group">
                <label className="form-label" style={{ display: 'block', marginBottom: '0.5rem', fontSize: '0.95rem', color: '#cbd5e1', fontWeight: '700' }}>Posición</label>
                <select
                  className="pro-input"
                  value={formJugador.posicion}
                  onChange={e => setFormJugador({ ...formJugador, posicion: e.target.value })}
                  style={{ width: '100%', padding: '0.85rem', borderRadius: '10px', border: '1px solid rgba(255,255,255,0.2)', background: '#1e293b', color: '#fff', fontWeight: '500' }}
                  required
                >
                  <option value="">Seleccione posición...</option>
                  {(() => {
                    let activeDeporteKey = null;
                    if (formJugador.equipo_id && equipos.length > 0) {
                      const eq = equipos.find(e => e.id.toString() === formJugador.equipo_id.toString());
                      if (eq && eq.deporte) {
                        const dName = eq.deporte.nombre.toLowerCase();
```

```
                    if (dName.includes('futbol') || dName.includes('fútbol')) activeDeporteKey =
'futbol';
                                else if (dName.includes('basquet') || dName.includes('básquet'))
activeDeporteKey = 'basquet';
                                 else if (dName.includes('voley') || dName.includes('vóley') ||
dName.includes('ecuavoley')) activeDeporteKey = 'voley';
                  }
                }

                if (activeDeporteKey) {
                  return (
                                  <optgroup label={DEPORTES_INFO.find(d => d.key ===
activeDeporteKey)?.label || "Posiciones"}>
                    {POSICIONES_DEPORTE[activeDeporteKey].map(pos => (
                    <option key={pos} value={pos} style={{ background: '#1e293b', color: '#fff'
}}>{pos}</option>
                    ))}
                  </optgroup>
                );
                } else {
                  return (
                    <>
                    <optgroup label="⚽ Fútbol">
                            {POSICIONES_DEPORTE.futbol.map(pos => <option key={pos}
value={pos} style={{ background: '#1e293b', color: '#fff' }}>{pos}</option>)}
                    </optgroup>
                    <optgroup label="🏀 Básquet">
                          {POSICIONES_DEPORTE.basquet.map(pos => <option key={pos}
value={pos} style={{ background: '#1e293b', color: '#fff' }}>{pos}</option>)}
                    </optgroup>
                    <optgroup label="🏐 Vóley">
                           {POSICIONES_DEPORTE.voley.map(pos => <option key={pos}
value={pos} style={{ background: '#1e293b', color: '#fff' }}>{pos}</option>)}
                    </optgroup>
                  </>
                );
                }
              })()}
            </select>
          </div>
        </div>
        <div style={{ marginBottom: '2.5rem' }} />

        <div style={{ display: 'flex', gap: '1rem', justifyContent: 'flex-end' }}>
            <button type="button" onClick={() => setShowModal(false)} style={{ padding:
'0.85rem 1.75rem', borderRadius: '10px', border: '1px solid rgba(255,255,255,0.2)',
background: 'transparent', color: '#cbd5e1', fontWeight: '700', cursor: 'pointer', fontSize:
'0.95rem' }}>Cancelar</button>
              <button type="submit" className="btn-primary" style={{ padding: '0.85rem
1.75rem', borderRadius: '10px', fontWeight: '700', cursor: 'pointer', display: 'flex', alignItems:
'center', gap: '0.5rem', fontSize: '0.95rem' }} disabled={savingJugador}>
            {savingJugador && <Loader2 className="animate-spin" size={20} />}
            {isEditing ? 'Guardar Cambios' : 'Registrar'}
          </button>
        </div>
      </form>
    </div>
  </div>
 )}
```

```
      {/* Import Summary Modal */}
      {showImportSummary && importResult && (
        <div style={{ position: 'fixed', top: 0, left: 0, width: '100%', height: '100%', background: 'rgba(15, 23, 42, 0.7)', backdropFilter: 'blur(5px)', display: 'flex', alignItems: 'center', justifyContent: 'center', zIndex: 60 }}>
          <div style={{ background: '#1e293b', borderRadius: '24px', width: '90%', maxWidth: '450px', padding: '2.5rem', boxShadow: '0 25px 50px rgba(0,0,0,0.5)', border: '1px solid rgba(255,255,255,0.1)' }}>
            <h2 style={{ fontSize: '1.5rem', fontWeight: 800, color: '#fff', marginBottom: '1.5rem', display: 'flex', alignItems: 'center', gap: '0.75rem' }}>
              <CloudUpload size={28} color="var(--primary-ocean)" /> Resultado de Importación
            </h2>
            <ul style={{ listStyle: 'none', padding: 0, color: '#cbd5e1', fontSize: '1rem' }}>
              <li style={{ marginBottom: '1rem', display: 'flex', alignItems: 'center', gap: '0.75rem' }}><CheckCircle size={22} color="#10B981" /> <strong>{importResult.inserted || 0}</strong> registros creados.</li>
              <li style={{ marginBottom: '1rem', display: 'flex', alignItems: 'center', gap: '0.75rem' }}><TrendingUp size={22} color="#F59E0B" /> <strong>{importResult.updated || 0}</strong> registros actualizados.</li>
              <li style={{ marginBottom: '1rem', display: 'flex', alignItems: 'center', gap: '0.75rem' }}><AlertCircle size={22} color="#EF4444" /> <strong>{importResult.errors?.length || 0}</strong> errores en filas.</li>
            </ul>

            {importResult.errors && importResult.errors.length > 0 && (
              <div style={{ marginTop: '1.5rem', padding: '1rem', background: 'rgba(239, 68, 68, 0.1)', borderRadius: '12px', border: '1px solid rgba(239, 68, 68, 0.2)', maxHeight: '150px', overflowY: 'auto' }}>
                <p style={{ color: '#fca5a5', fontWeight: 'bold', fontSize: '0.95rem', marginBottom: '0.5rem' }}>Detalles de errores:</p>
                {importResult.errors.map((e, i) => (
                  <div key={i} style={{ fontSize: '0.85rem', color: '#fca5a5', marginBottom: '6px', borderBottom: '1px solid rgba(239, 68, 68, 0.1)', paddingBottom: '4px' }}>
                    <strong>Fila {e.linea} (ID: {e.cedula}):</strong> {e.error}
                  </div>
                ))}
              </div>
            )}

            <button onClick={() => setShowImportSummary(false)} className="btn-primary" style={{ width: '100%', marginTop: '2rem', padding: '1rem', fontWeight: '700', borderRadius: '12px' }}>Entendido</button>
          </div>
        </div>
      )}
    </div>
  );
}
```

## CONCLUSIONES

- Se desarrolló un Sistema Integral de Gestión para Campeonatos Deportivos que centraliza la organización de torneos de fútbol y baloncesto en la Universidad Estatal de Bolívar, mejorando la administración general de las actividades deportivas.

- El análisis de los procesos actuales permitió identificar deficiencias relacionadas con el manejo manual de información, dificultades en el control de participantes y limitaciones en la planificación de encuentros, lo que justificó la implementación de una solución tecnológica.
- El diseño del sistema se estructuró mediante una arquitectura modular que integra los procesos de inscripciones, carnetización con códigos QR, programación de partidos y registro de resultados, garantizando organización y escalabilidad.
- La implementación de la plataforma permitió disponer de una interfaz accesible e intuitiva, adaptable a diferentes disciplinas deportivas y orientada a facilitar la interacción de los usuarios con el sistema.
- El sistema desarrollado cumple con los objetivos planteados, demostrando viabilidad técnica, funcional y aplicabilidad dentro del entorno universitario.

## RECOMENDACIONES

- Implementar el sistema como herramienta para la gestión de campeonatos deportivos, asegurando la digitalización continua de los procesos organizativos.
- Establecer procedimientos institucionales para el registro constante y actualización correcta de la información dentro del sistema.
- Mediante métricas de desempeño y la generación automática de informes, la plataforma sea capaz de procesar datos por sí misma, sumando además un canal de avisos directos para que nadie en la comunidad deportiva se quede fuera de juego o pierda el hilo de lo que sucede.
- Más allá de que todo marche bien ahora, hay que fijar una hoja de ruta de auditorías y limpiezas profundas en la base de datos; el objetivo real es que, cuando el tráfico de jugadores crezca, la velocidad y la seguridad no se vengan abajo.
- Dado que la arquitectura es modular, sugerimos aprovechar este núcleo para cubrir otras disciplinas o eventos de la institución; la idea es que este desarrollo sirva de base para un ecosistema deportivo mucho más amplio y no se limite a un solo torneo.

## BIBLIOGRAFÍA


Andrés Morena, J. (2009). Estandarización de procesos: fundamentos y aplicaciones. Madrid: Díaz de Santos.

Bass, L., Clements, P., & Kazman, R. (2013). Software architecture in practice (3rd ed.). Boston: Addison-Wesley.

Cisneros Pinto, M. (2023). Plataforma digital para la organización de eventos deportivos estudiantiles en instituciones académicas. Revista Tecnológica Universitaria, 11(2), 33–49. https://doi.org/10.5678/rtu.2023.11233

Cortes Peñafiel, C. (2024). Implementación de sistemas de carnetización digital en instituciones educativas. Revista Tecnologías de la Información, 15(2), 45–59. https://doi.org/10.1234/rti.2024.15204

Date, C. J. (2004). An introduction to database systems (8th ed.). Boston: Addison-Wesley.

Lorenz, H. (2007). QR codes in modern information systems. Journal of Information Science, 33(2), 131–142. https://doi.org/10.1177/0165551506072165

Nielsen, J. (2012). Usability 101: Introduction to usability. Nielsen Norman Group. https://www.nngroup.com/articles/usability-101-introduction-to-usability/

Paredes Reyes, F. (2021). Carnetización digital con códigos QR en la gestión de eventos deportivos universitarios. Revista Iberoamericana de Tecnología, 18(3), 77–92. https://doi.org/10.5678/rit.2021.18306

Pressman, R. S., & Maxim, B. R. (2021). Software engineering: A practitioner’s approach (9th ed.). New York: McGraw-Hill.

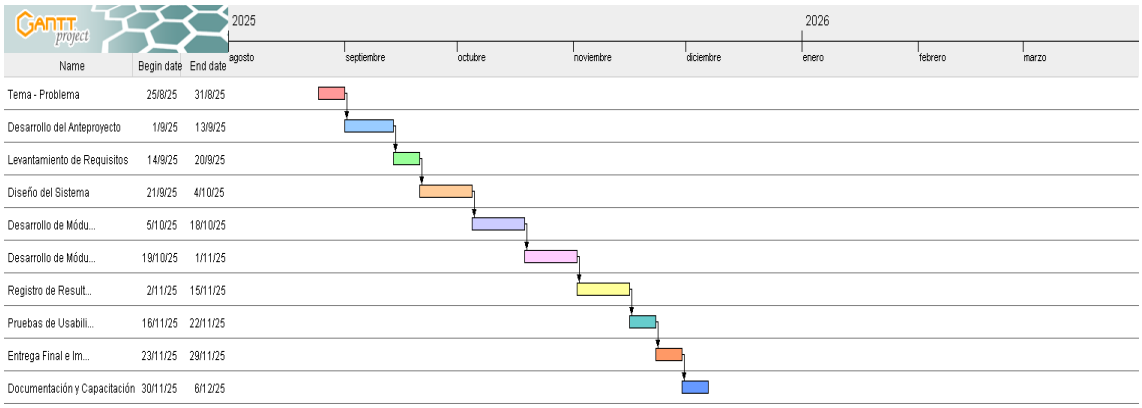
Schwaber, K., & Sutherland, J. (2020). The Scrum guide: The definitive guide to Scrum. Scrum.org. https://scrumguides.org

Sommerville, I. (2016). Software engineering (10th ed.). Boston: Pearson.

Yépez Vendoval, A. (2020). Gestión de campeonatos deportivos universitarios mediante herramientas digitales. Revista Científica de Ciencias Sociales, 12(1), 89–104. http://repositorio.utn.edu.ec/handle/123456789/14580

## ANEXOS

**Anexo 1**. Cronograma (Gantt)




**Anexo 2**. Presupuesto Ejecutado

***Tabla 12.*** *Presupuesto ejecutado del proyecto*

| Rubros´ | Descripción | Cantidad | Costo Unitario (USD) | Subtotal (USD) |
|---|---|---|---|---|
| **Computadora portátil** | Laptop gama media (Intel i5/Ryzen 5, 8GB RAM, SSD 512GB) para programación y pruebas. | 1 | 700,00 | 700,00 |
| **Servicios de hosting y dominio** | Hosting compartido con SSL y dominio.*com.ec* por 1 año. | 1 | 90,00 | 90,00 |
| **Materiales de oficina** | Resmas de papel (2 x 6,00 = 12,00), tinta/impresiones (40,00), anillados (20,00), carpetas (10,00), esferos y marcadores (8,00). | 1 lote | 90,00 | 90,00 |
| **Gastos varios e imprevistos** | Fondo para cubrir necesidades adicionales de software, copias, respaldos o emergencias menores. | 1 | 120,00 | 120,00 |
| **Total, estimado** | | | | **1.000,00** |

**Anexo 3.** Carta de aceptación de la organización donde se realizó el trabajo de integración curricular. (de ser el caso)

**Anexo 5.** Manual del Administrador

**UNIVERSIDAD ESTATAL DE BOLÍVAR**

**FACULTAD DE CIENCIAS ADMINISTRATIVAS GESTIÓN EMPRESARIAL E INFORMÁTICA**

**CARRERA DE INGENIERÍA EN SOFTWARE**

ANEXO 5

# MANUAL DEL ADMINISTRADOR

**Sistema de Gestión de Campeonatos Deportivos**

**2026**

# Índice

Portada……1

**TEMA DEL PROYECTO TECNOLÓGICO ..... I**

**DEDICATORIA ..... II**

**AGRADECIMIENTO ..... III**

**DEDICATORIA ..... IV**

**AGRADECIMIENTO ..... V**

**CERTIFICADO DE VALIDACIÓN ..... VI**

**DERECHOS DE AUTOR ..... VII**

**INDICE DE CONTENIDOS ..... VIII**

**INDICE DE TABLAS ..... X**

**INDICE DE FIGURAS ..... XI**

**INTRODUCCIÓN ..... 1**

**RESUMEN ..... 2**

**ABSTRACT ..... 3**

**KICHWA ..... 4**

CAPÍTULO I ..... 5

FORMULACIÓN GENERAL DEL PROYECTO ..... 5

*1.1.Tema ..... 5*

*1.2.Descripción del Problema ..... 5*

*1.3.Justificación ..... 6*

*1.4.Objetivos: General y Específicos ..... 7*

CAPÍTULO II ..... 9

MARCO TEÓRICO ..... 9

CAPÍTULO III ..... 17

METODOLOGÍA ..... 17

CAPÍTULO IV ..... 20

INGENIERÍA DEL PROYECTO ..... 20

CONCLUSIONES ..... 77

RECOMENDACIONES ..... 78

BIBLIOGRAFÍA ..... 79

ANEXOS ..... 80

## 1. Introducción

El presente manual describe de manera detallada el funcionamiento del sistema desde la perspectiva del Administrador, quien posee el nivel más alto de acceso dentro de la plataforma. Si el administrador falla, el orden de los torneos se viene abajo; así de simple. Este perfil no está solo para ver datos, sino para vigilar que el engranaje del sistema no se trabe y que cada parte del software haga lo que le toca. Es quien pone el sello de seguridad en las inscripciones, reparte los roles y da la cara por la veracidad de los marcadores. Básicamente, si confiamos en la plataforma es porque este usuario trabaja con un cuidado extremo, siendo el muro que impide que la información se ensucie o se pierda.

## 2. Requisitos del Sistema

| Elemento | Especificación mínima |
|---|---|
| **Dispositivo** | Computadora de escritorio o portátil |
| **Navegador** | Chrome, Firefox o Edge actualizados |
| **Conexión** | Internet estable |
| **Acceso** | Credenciales institucionales válidas |

Para un funcionamiento adecuado del sistema, el usuario debe contar con los siguientes elementos técnicos mínimos:

### 2.1 Requisitos del Hardware

Se recomienda el uso de un equipo que permita una navegación estable, como:

- Computadora de escritorio
- Computadora portátil
- Tablet

El dispositivo debe poseer capacidad suficiente para visualizar interfaces gráficas modernas y facilitar la interacción con formularios, tablas de datos y elementos visuales.

### 2.2 Requisitos del Software

- Navegador web actualizado (Google Chrome, Mozilla Firefox o Microsoft Edge).
- Sistema operativo compatible con navegación web moderna.
- Conexión estable a internet, necesaria para la comunicación continua con el servidor.

### 2.3 Credenciales de Acceso

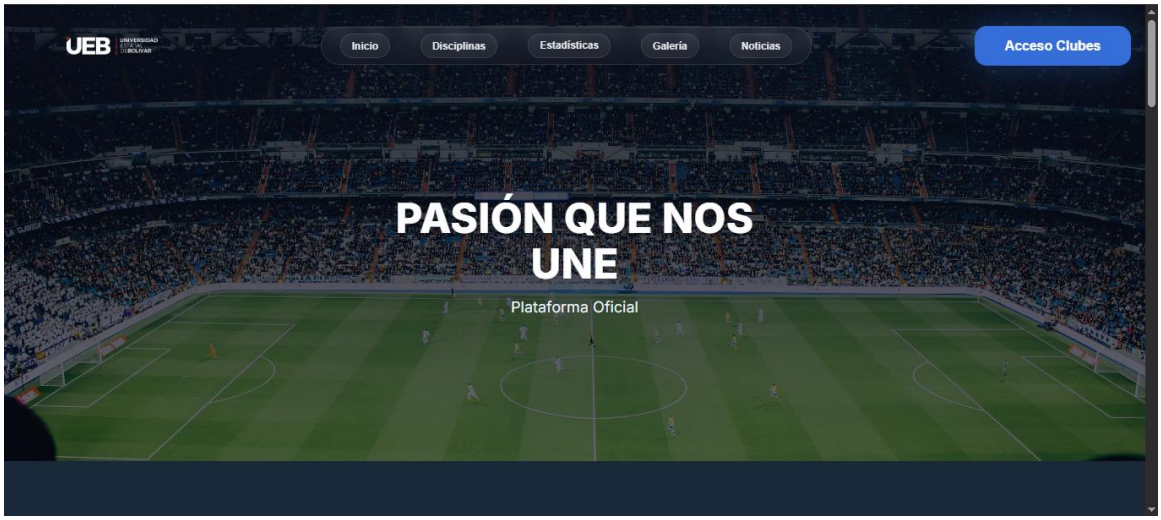
Cada usuario debe disponer de credenciales personales, estas credenciales permiten identificar al usuario y habilitar únicamente las funciones asociadas a su rol:

- Correo electrónico
- Contraseña

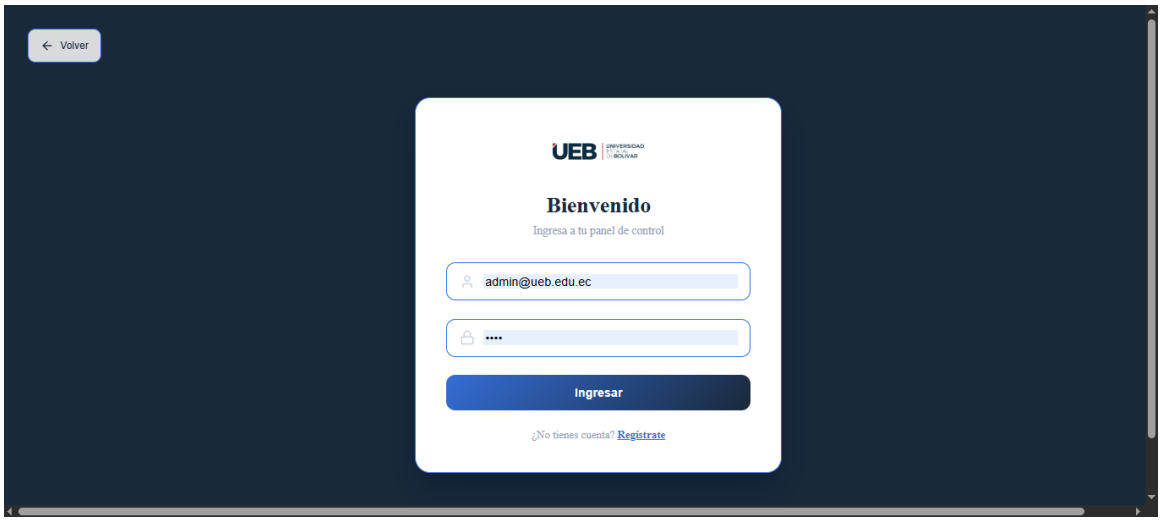
## 3. Acceso al Sistema

El acceso se realiza desde la página principal de la plataforma. Procedimiento de ingreso:

1. Acceder a la dirección web del sistema.




2. Seleccionar la opción Acceso a Clubes.




3. Ingresar correo electrónico y contraseña.
4. Confirmar el ingreso.

El sistema valida la información y muestra el panel correspondiente al rol asignado (Representante, Árbitro, Jugador). En caso de error, se solicitará reingresar los datos. Por seguridad, se recomienda cerrar sesión al finalizar.

### 3.1 Panel Principal del Administrador

Es el centro de control principal del sistema. Desde esta pantalla se supervisa de forma global la actividad deportiva institucional, permitiendo administrar torneos, equipos, jugadores y programación de partidos desde un único entorno de trabajo. Este panel está diseñado para ofrecer información estratégica en tiempo real y accesos directos a las acciones más utilizadas.

- **Objetivo del Dashboard**

Proporcionar al administrador un panorama general y actualizado de la actividad deportiva registrada en el sistema, permitiendo detectar situaciones que requieran atención inmediata.

**Elementos Principales del Dashboard**

| *Elemento* | *Descripción* | *Utilidad para el administrador* |
|---|---|---|
| ***Campeonatos activos*** | **Número de torneos en curso** | **Control del estado general de las competiciones** |
| ***Equipos registrados*** | **Total de equipos inscritos** | **Verificación del nivel de participación** |
| ***Jugadores activos*** | **Deportistas habilitados** | **Seguimiento del volumen de registros** |
| ***Árbitros disponibles*** | **Personal arbitral registrado** | **Planificación de partidos** |
| ***Partidos próximos*** | **Encuentros programados** | **Organización y control de agenda** |

**Importancia operativa**

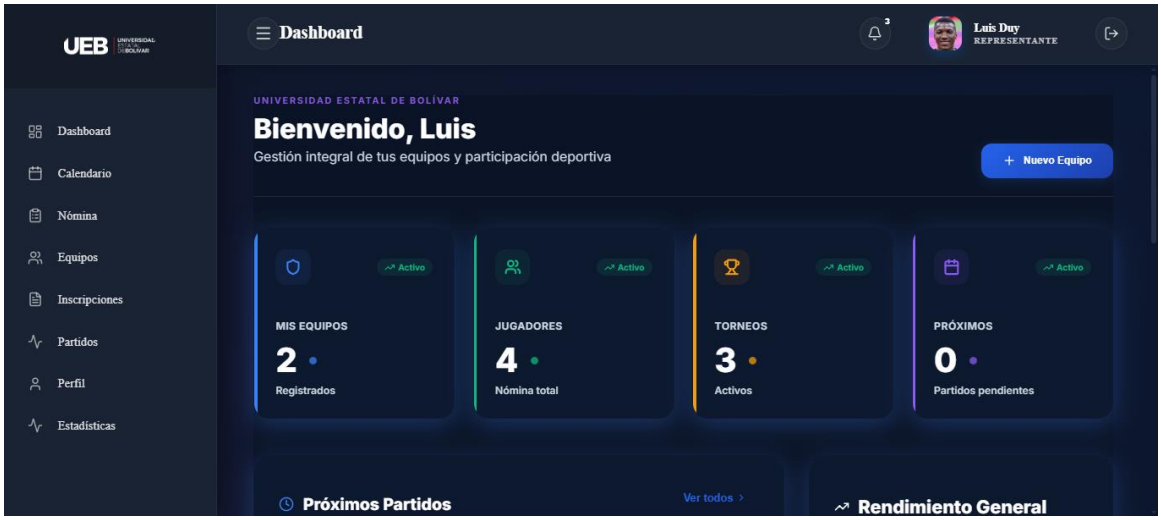
El Dashboard reduce el tiempo de gestión administrativa, ya que concentra la información esencial en una sola pantalla. Esto permite:

- Detectar desequilibrios en el número de equipos o jugadores
- Identificar partidos próximos o pendientes
- Mantener control institucional del campeonato
- Garantizar un seguimiento eficiente de la actividad deportiva

**Frecuencia de uso recomendada**

Se recomienda que el administrador revise el Dashboard:

| *Momento* | *Motivo* |
|---|---|
| ***Inicio de jornada*** | **Verificar novedades** |
| ***Antes de programar partidos*** | **Confirmar disponibilidad** |
| ***Después de registrar resultados*** | **Comprobar actualización** |
| ***Durante el campeonato*** | **Seguimiento continuo** |




Fuente: Elaboración propia (2026).

## 4. Torneos

El módulo Torneos permite al administrador crear, organizar y supervisar todas las competencias deportivas registradas en el sistema. Esta sección constituye el eje estructural de la plataforma, ya que sobre cada torneo se vinculan equipos, jugadores, partidos y estadísticas. Desde esta pantalla el administrador puede visualizar el estado general de las competiciones y administrar cada una de manera individual.

### 4.1 Panel General de Torneos

En la parte superior se presentan indicadores que resumen la actividad deportiva:

- **Total, Torneos:** número general de campeonatos registrados en el sistema.
- **Disciplinas:** cantidad de deportes gestionados (por ejemplo, fútbol o básquet).
- **En Curso:** torneos actualmente activos.
- **Finalizados:** campeonatos que ya han concluido y forman parte del historial.

### 4.2 Organización por Pestañas

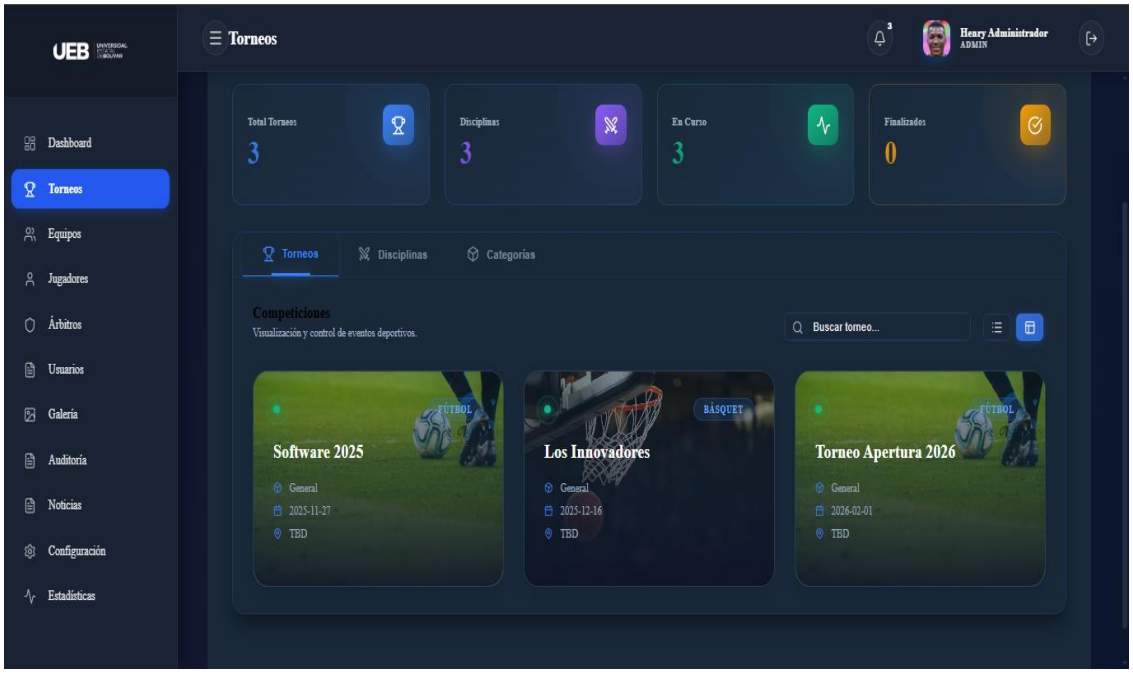
El módulo se divide en tres secciones principales:

- **Torneos:** donde se visualizan y administran las competencias creadas.

- **Disciplinas:** permite gestionar los diferentes deportes habilitados en el sistema.
- **Categorías:** clasifica los torneos por niveles, grupos o divisiones.

### 4.3 Visualización de Competiciones

La parte central organiza todo el despliegue de los torneos mediante tarjetas individuales. No buscamos marear al usuario, así que cada tarjeta va directo al grano: te dice cómo se llama el campeonato, qué deporte es, la categoría y cuándo se juega. También metimos un aviso visual sobre el estado del evento (si está activo o apenas por arrancar). La idea es que no tengas que dar mil clics para enterarte de lo básico; la información clave está ahí, servida en bandeja. El diseño visual permite identificar rápidamente cada competencia y acceder a su gestión específica.




## 5. Equipos

Desde el módulo de Equipos, el administrador toma las riendas de todo lo que tenga que ver con los conjuntos inscritos en los torneos. Más que una simple base de datos, es el lugar donde se centraliza el registro y la supervisión de cada equipo que pisa la cancha. Aquí se controla quién participa y en qué competencia está metido, asegurando que toda la información de los grupos esté organizada y a la mano para cualquier ajuste que el campeonato necesite sobre la marcha.

### 5.1 Panel General de Equipos

Nada más abrir la sección, arriba del todo te sale un vistazo rápido para que no des vueltas buscando lo básico. Metimos un contador de solicitudes pendientes para que veas

al toque cuántos equipos esperan que les des el visto bueno. Justo al lado, tienes el dato de cuántos grupos hay registrados y cuántas disciplinas están moviéndose en ese momento. Al final, lo que queremos es que, apenas entres, ya tengas el mapa completo de la carga de trabajo y de qué tan grande va el torneo sin clics de más.

**5.2 Revisiones de Inscripción**

En "Revisiones de Inscripción" es donde se decide de verdad quién entra: es el filtro real del administrador para las solicitudes. Aquí no se viene a mirar por encima, sino a determinar quién compite y quién no según lo que enviaron los equipos. Desde este panel se les da luz verde a los registros que están en orden y se sacan del camino los que vienen incompletos o con fallos. Básicamente, es la forma de asegurar que nadie se salte las normas y que en el torneo solo participen los que cumplen con lo que exige la institución.

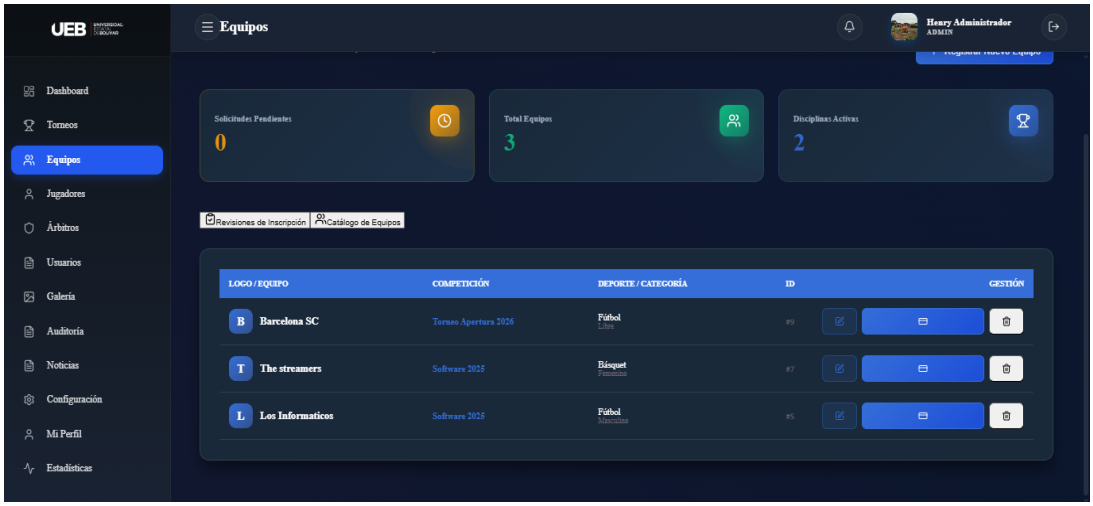
**5.3 Catálogo de Equipos**

Manejar el listado de equipos es como tener un panel de control a mano para que el administrador no se líe con tanto nombre y registro suelto. En esa tabla central volcamos lo que de verdad importa ver rápido: el nombre del grupo y su escudo junto a ese ID que no se repite, pero también el torneo donde están metidos y la categoría que les toca. Lo bueno es que no tienes que andar entrando y saliendo de los perfiles para arreglar un detalle, ya que tiramos los botones de gestión al final de cada fila. Al final, esta es la vía para que quien administra el sistema mantenga a raya a los inscritos y verifique que cada uno está donde debe sin romperse la cabeza con procesos largos. Es un método directo que quita bastante peso de encima, sobre todo cuando la lista de equipos se vuelve gigante y necesitas encontrar a alguien en un segundo sin perder el tiempo navegando por todo el sistema de arriba abajo.

**5.4 Opciones de Gestión**

Cuando te metes a revisar los equipos, el sistema te suelta varias herramientas de control para que el administrador no esté atado de manos si algo sale mal. No se trata solo de ver la lista, sino de poder meter mano en los datos si un representante se equivocó al escribir el nombre o si hay que actualizar cualquier detalle del registro sobre la marcha. También tienes la opción de abrir la ficha completa para curiosear a fondo quiénes son los integrantes y ver toda la info que mandaron sin dejarse nada fuera. Y si por lo que sea un equipo se baja del torneo o metieron la pata al inscribirse dos veces, está el botón para borrarlos del mapa y limpiar el sistema en un segundo. Al final, tener estas funciones ahí mismo, a tiro de clic, es lo que permite que el administrador maneje el cotarro con soltura,

sin que el registro de participantes se vuelva un caos imposible de corregir. Es esa libertad de arreglar, mirar o quitar lo que haga falta lo que de verdad ayuda a que el torneo ruede sin problemas técnicos que frenen la gestión diaria de las inscripciones.




## 6. Gestión de Jugadores

El apartado de Jugadores es básicamente el motor de la plataforma, porque sin deportistas bien registrados no hay torneo que valga. Aquí es donde el administrador mete las manos en todo lo que tiene que ver con la gente que compite; no es solo una lista de nombres, sino el sitio para controlar las altas, corregir datos que vengan mal o seguirle la pista a cada participante que se apunta a los equipos de la institución. En realidad, esta parte es la columna vertebral de todo el sistema, porque de aquí sale la información necesaria para armar los partidos, sacar las estadísticas de rendimiento y, sobre todo, para confirmar que las nóminas de cada equipo sean legales y oficiales antes de que ruede el balón. Si los datos de los jugadores están al día y bien vigilados desde esta sección, te ahorras líos a la hora de validar quién puede jugar y quién no, asegurando que el administrador tenga el mando sobre el recurso más importante de las competiciones sin que se le escape ningún detalle por el camino. Es el centro operativo donde se valida que todo el mundo esté en regla antes de saltar al campo.

### 6.1 Vista General del Módulo

En cuanto entras a jugadores te salta el listado con toda la gente que se ha ido apuntando. Está pensado para que el administrador no pierda el tiempo rastreando nombres en una lista sin sentido. Lo bueno es que cada uno ya viene amarrado a su equipo y al deporte que hace; así toda cuadra con los torneos y las fechas sin que te vuelvas loco. Con los datos puestos así, te quitas el marrón de que alguien acabe en el grupo que no es o de que se te pierda información importante por ahí. Básicamente es el lugar para controlar las

plantillas y que no se descontrole el registro con fichas mal puestas. Solo hay que chequear quién juega y en qué equipo está para dejarlo todo listo antes de que empiece el partido.

### 6.2 Registro de Nuevos Jugadores

Para dar de alta a un jugador, el administrador solo mete los datos en el formulario de la plataforma y ya está. En ese paso es donde el perfil se queda amarrado a su identidad de verdad, que es lo que sirve para no andar con líos de nombres duplicados o fallos en las fichas después. En cuanto le das a guardar y la ficha se queda grabada en la base de datos, el deportista ya te aparece para que lo metas en la nómina de su equipo al momento. Con este filtro de entrada te aseguras de que todos cumplan el reglamento antes de que les toque jugar el partido. Es la barrera para que no se cuele nadie que no esté en la lista oficial; así el gestor no tiene que andar saneando actas o arreglando fallos cuando el torneo ya ha empezado. Al final es la forma de llevar un censo de jugadores limpio y controlado todo el tiempo para que no se te escape ni uno solo.

### 6.3 Administración de Información

Si te toca cambiar cualquier dato, el sistema te deja editar las fichas de los jugadores sobre la marcha para tener la información al día mientras dure el torneo. Los cambios los vas haciendo cuando te haga falta sin miedo a cargarte el historial ni lo que ya está registrado de antes. También tienes a mano los perfiles para entrar uno por uno y confirmar en qué equipo está metido cada uno o ver si tienen permiso para jugar los partidos de ahora. Es una forma de ver rápido que todo esté en orden sin andar perdiendo el tiempo buscando en carpetas. Entras, miras el perfil y listo, ya sabes que el deportista está donde le toca. Al final es una herramienta para que el administrador arregle fallos rápido y controle quién entra al campo en cada jornada sin líos.

### 6.4 Control y Organización Deportiva

Llevar bien el control de los jugadores es lo que te permite organizar los partidos sin que se te escape nada. Al tener las fichas ordenadas, el sistema te saca las estadísticas sin errores, te deja confirmar quién juega en cada partido y te facilita mucho el calendario de encuentros. Todo este apartado está conectado con el resto del programa; así la información pasa de los equipos a los partidos y a los reportes finales sin que tengas que andar metiendo los mismos datos dos veces. Es la forma de que todo cuadre y que la administración del torneo no se vuelva un caos, porque los nombres y los resultados fluyen solos entre una sección y otra. En el fondo, es lo que te da la seguridad de que los

números de la competición son reales y que el registro no se te va a desmadrar a mitad de temporada.

**7. Gestión de Árbitros**.

En el apartado de Árbitros, el administrador se encarga de dar de alta y controlar a todo el equipo que va a pitar los partidos del torneo. Esta sección está ahí para que no haya dudas de quién dirige cada encuentro y para asegurar que todos los réferis estén bien registrados en el sistema de la institución. Al tenerlo todo en un mismo sitio, es mucho más fácil ver quién es quién, seguir cómo lo están haciendo y confirmar si su ficha está activa dentro de la plataforma. Lo bueno es que te quitas el lío de andar buscando nombres sueltos; aquí tienes la lista oficial a mano para que la organización de los torneos sea seria y no falte nadie cuando llegue la hora del silbatazo inicial. Es la manera de que los encargados de pitar tengan su historial al día y que el administrador sepa siempre con quién cuenta para cada jornada.

### 7.1 Vista General del Módulo

Cuando entras a ver los árbitros, te salta un vistazo rápido de toda la gente que está apuntada. Arriba de todo aparecen los datos clave: cuántos réferis hay en total, su nivel y quiénes tienen el certificado para pitar. Es para que sepas con quién cuentas antes de ponerte a asignar los partidos. Justo debajo de ese bloque tienes la lista completa, puesta de forma que no pierdas el tiempo buscando nombres. En cada ficha tienes a mano lo justo para identificar a cada uno al momento y ver si están en activo o si les falta algo en el sistema. Básicamente es para tener a todo el cuerpo arbitral a mano en una sola pantalla y sin dar vueltas.

### 7.2 Registro de Árbitros

El sistema trae la opción de "Registrar Oficial" para que el administrador meta a los árbitros nuevos en la plataforma. Al darles de alta, vas poniendo sus datos personales y profesionales para ver que están aptos para pitar en las competiciones. Hacer este paso es lo que te asegura que solo la gente que tiene permiso aparezca para pitar los partidos oficiales. Al final, es la forma de tener todo bajo control y que la organización sea clara, para que nadie que no esté en la lista termine metido en un acta por error. Es un filtro que hace falta para que el torneo sea serio y no haya líos con quién dirige cada encuentro.

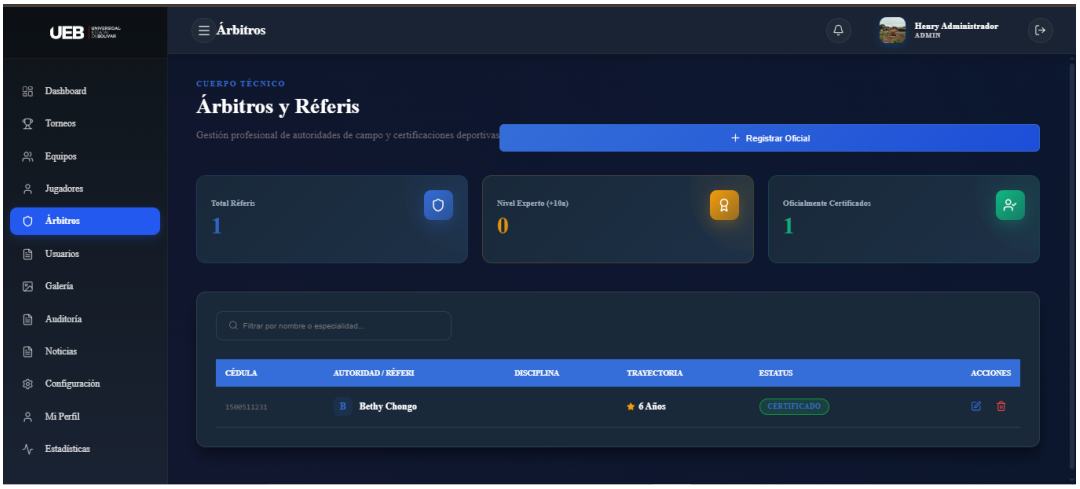
### 7.3 Administración y Actualización de Información

Cuando ya los tienes en la lista, puedes entrar a cambiar los datos de los árbitros directamente desde ahí. El administrador puede actualizar la ficha si hay algún cambio en

su nivel, si sacaron una certificación nueva o si simplemente han pasado a estar inactivos. Lo bueno del sistema es que te deja tener toda la información al día sin que se borre lo que hicieron antes; así no pierdes el rastro de la historia de cada árbitro en la plataforma. Al final, es una forma de que los perfiles estén siempre listos para usarse sin que la administración se vuelva un lío de papeles viejos o datos que ya no sirven.

### 7.4 Control del Estado y Certificación

Cada árbitro sale con un estado en el sistema para que sepas en qué situación está, por ejemplo, si ya está certificado o si todavía le falta que lo validen. Con esto puedes ver al momento si un réferi está libre para pitar en los partidos que ya tienes programados. Llevar este control de los certificados ayuda a que los torneos salgan mejor organizados, porque así te aseguras de que los partidos los dirija gente que de verdad está capacitada. Al final es una forma de no jugártela y saber que quien entra al campo tiene los papeles en regla para que el nivel de la competición no baje.




## 8. Perfil del Usuario

Desde el apartado de Usuarios, el administrador maneja quién entra al sistema controlando las cuentas, los roles y los permisos de cada uno. Esta parte es clave para que la plataforma sea segura, porque es donde decides quién tiene acceso y qué cosas puede hacer o tocar dentro del programa deportivo. Básicamente es el filtro para que nadie entre donde no le toca y para que cada responsable tenga sus funciones bien claras. Así te aseguras de que la información esté protegida y que cada usuario solo vea lo que necesita para trabajar en el torneo.

### 8.1 Vista General del Módulo

Al ingresar a la sección de usuarios, el administrador visualiza un panel informativo que presenta un resumen del estado actual de las cuentas registradas. Se muestran indicadores

generales como el número total de usuarios, cuentas activas y cantidad de administradores existentes en el sistema. Debajo del resumen se encuentra el listado general de usuarios, donde cada registro incluye información básica de identificación, contacto institucional, rol asignado y estado de la cuenta.

### 8.2 Registro de Nuevos Operadores

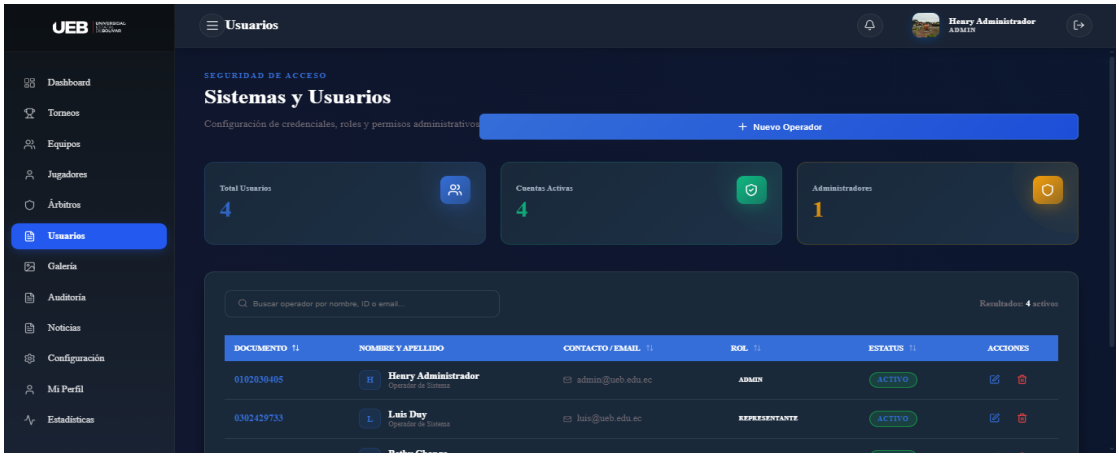
Con el botón de "Nuevo Operador" puedes abrir cuentas nuevas en la plataforma para que otros te ayuden. Cuando registras a alguien, le pones sus datos y le asignas un rol para decidir qué partes del sistema puede ver y cuáles no. El programa deja crear distintos perfiles de administrador, lo que viene de lujo para repartir el trabajo sin que todo el mundo tenga acceso a todo por seguridad. Al final, tener a los operadores bien registrados es la mejor forma de delegar tareas en la gestión deportiva sin que se te descontrole el acceso a la información sensible de la institución.

### 8.3 Administración de Roles y Permisos

Cada usuario tiene asignado un rol que es el que manda sobre lo que puede o no puede hacer dentro del sistema. El administrador tiene la libertad de cambiar estos roles cuando le haga falta, ajustando los permisos según lo que le toque hacer a cada uno en ese momento. Llevar bien este control de los roles es lo que te permite tener una jerarquía ordenada y sin líos, evitando que entre gente donde no debe y quitándote de encima fallos por accesos que no corresponden. Al final, es la manera de que cada uno tenga sus herramientas a mano sin poner en peligro la seguridad de toda la plataforma.

### 8.4 Estado y Control de Cuentas

El programa te deja ver al momento cómo está cada cuenta, para que sepas si un usuario está en activo o si lo tienes inhabilitado. Esto viene muy bien para controlar los accesos cuando alguien deja de colaborar en el torneo o si, por lo que sea, necesitas suspenderle la cuenta un tiempo. Aparte de eso, puedes cambiar cualquier dato o borrar registros si te hace falta; así la lista de usuarios no se te llena de gente que ya no pinta nada y la tienes siempre al día. Al final, es la forma de que el administrador decida quién entra y quién sale del sistema sin dar mil vueltas ni dejar cuentas abiertas por ahí que ya no se usan.




## 9. Galeria

El módulo Galería permite al administrador gestionar los archivos multimedia relacionados con los torneos y actividades deportivas institucionales. Esta sección está diseñada para organizar imágenes y recursos visuales que sirven como evidencia, difusión y documentación de los eventos realizados dentro de la plataforma.

A través de este apartado se centraliza el almacenamiento de contenido gráfico, facilitando su administración y visualización desde un único espacio.

### 9.1 Vista General de la Galería

Cuando entras en la galería, lo primero que ves son las carpetas organizadas por tarjetas, donde cada una es un evento o un torneo distinto. Cada una de estas galerías guarda los archivos que tocan para ese campeonato o actividad en concreto. El sistema te marca cuántas fotos o vídeos hay metidos en total y te deja entrar a saco en cualquier colección que hayas creado sin dar mil vueltas. Es la forma más rápida de tener todo el material visual ordenado por fechas o por deportes, para que no se te mezclen las fotos de un torneo con las de otro. Al final, es un vistazo rápido para saber qué contenido tienes a mano y entrar directo a lo que busques.

### 9.2 Registro de Nuevos Archivos

Con el botón de "Nuevo Archivo", el administrador puede ir subiendo todo el contenido de fotos y vídeos al sistema. Una vez que los cargas, se guardan en su galería para que todo el evento deportivo quede bien registrado con imágenes. Lo que sí te conviene es echarle un ojo a lo que subes, para que las fotos se vean bien y tengan que ver con lo que estás publicando. Al final, es una forma de tener el torneo vivo con imágenes reales de lo que pasa en el campo.

Aquí tienes lo que puedes hacer básicamente:

- Subir fotos de los torneos o de los equipos que están jugando.
- Separar los archivos por categorías o por el evento que toque.
- Cambiar algún dato o la descripción si te has equivocado al subirlo.
- Borrar archivos si ya no te sirven o quieres limpiar la galería.

### 9.3 Administración del Contenido Multimedia

Cualquier foto o vídeo que tengas en la galería trae sus opciones para que cambies los datos o borres el archivo si ves que ya no te sirve para nada. Estos botones están ahí más que nada para que la lista no se te llene de imágenes repetidas o fotos que no tienen nada que ver con el tema. Al manejarlo todo desde un mismo sitio, te aseguras de que lo que se acabe publicando tenga sentido con los partidos y con lo que pasa de verdad en el torneo. Al final es para que la galería no sea un desmadre y que solo se queden las imágenes que de verdad valen la pena.

## 10. Auditoria del Sistema

El apartado de Auditoría está ahí para que el administrador pueda echar un ojo a todo lo que se hace dentro del programa. Esta sección es la que se encarga de que todo sea transparente, porque va guardando de forma automática cada movimiento que hacen los usuarios. Con esta herramienta puedes seguirle el rastro a cualquier cambio, a quién ha entrado o a qué cosas han tocado los otros administradores. Al final, es la mejor forma de no perderle la pista a la información y saber exactamente quién hizo qué y cuándo, para que no haya dudas si pasa algo raro.

### 10.1 Vista General de Auditoría

Al ingresar al módulo se presenta un panel resumen que muestra indicadores generales del comportamiento del sistema, tales como:

- Total, de registros almacenados.
- Actividad generada durante el día.
- Número de errores detectados.
- Usuarios únicos que realizaron acciones.

### 10.2 Búsqueda y Filtrado de Eventos

El programa viene con unos filtros que te salvan la vida para encontrar cosas puntuales sin tener que tragarte todo el historial de auditoría. Básicamente, eliges qué quieres mirar y te saca lo que se hizo en los días que tú le digas.

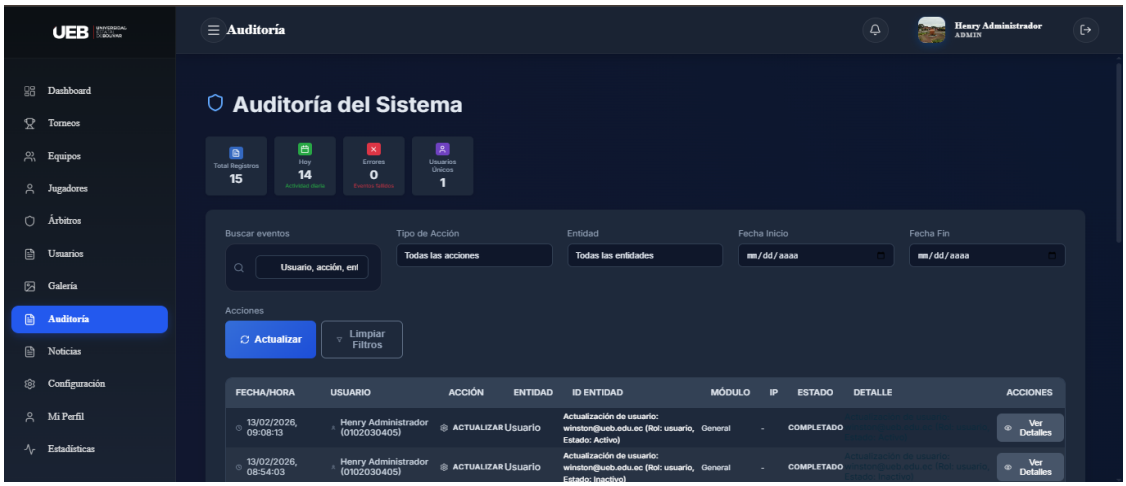
Tienes filtros para buscar por el usuario que metió la mano, el tipo de cambio que se hizo, o incluso para ver qué sección específica del sistema fue la que se tocó. También puedes poner una fecha de inicio y otra de fin para no volverte loco buscando entre miles de datos. Al final, lo que hace esto es que puedas mirar cualquier detalle a fondo sin ir línea por línea. Es mucho más directo y vas a lo que importa.

### 10.3 Registro de Actividades

Cada cosa que se hace en el sistema deja un rastro con todos los detalles para que sepas qué ha pasado exactamente. En el historial vas a ver de todo: a qué hora se hizo, quién fue el que entró y qué movimiento hizo exactamente. También te sale en qué parte del programa estuvo tocando, desde qué conexión (la IP) lo hizo y si el cambio salió bien o si dio algún error. Básicamente te explica qué se cambió antes y cómo quedó después. Tener todo esto así de desglosado le sirve al administrador para ver si hay algo que no cuadra, chequear los cambios y que nadie genere problemas en la plataforma sin que se dé el aviso correspondiente.

### 10.4 Consulta de Detalles

Con el botón de "Ver Detalles" puedes entrar a fondo en lo que pasó en cualquier momento. Esto viene de perlas cuando tienes que hacer una revisión técnica, una auditoría o simplemente ver por qué hubo un fallo. Al tener estos datos a mano, la administración se queda mucho más tranquila, porque tienes la prueba real de cada cosa que se ha hecho en el sistema. Al final, es la forma de no ir a ciegas y saber exactamente qué movimiento hizo cada uno sin que te lo cuenten.




## 11. Gestión de Noticias

En la sección de Noticias, el administrador puede colgar todo lo que sea importante, desde cuándo empiezan las inscripciones hasta los avisos de los torneos. Al final, esto sirve

como el canal oficial para que la administración hable con la gente que usa el sistema. Con esta herramienta te aseguras de que tanto los jugadores como los representantes y los árbitros estén al tanto de lo que va pasando y de cualquier cambio de última hora. Es la mejor forma de que nadie se quede fuera de juego y todos reciban las novedades importantes directamente en la plataforma.

### 11.1 Visualización de Noticias

Al ingresar al módulo se muestra un panel con todas las noticias registradas en formato de tarjetas informativas. Cada publicación incluye:

- Fecha de publicación.
- Título de la noticia.
- Descripción breve del contenido.
- Opciones de gestión disponibles.

### 11.2 Registro de Nueva Noticia

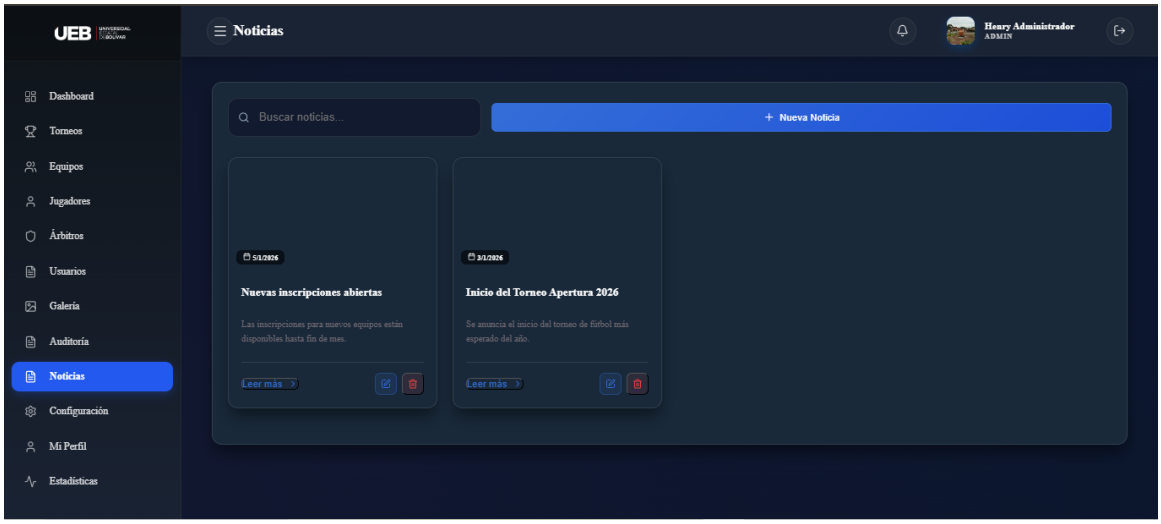
Si te toca subir un comunicado, nada más tienes que darle al botón ese de “Nueva Noticia” y rellenar los cuatro huecos que te salen. No tiene pérdida ninguna: le encasquetas un título que se entienda, sueltas ahí el rollo de lo que sea que pase y eliges el día que quieres que se vea en la web. Si hay algún detalle más del evento que sea importante, lo plantas también para que la gente se entere bien de qué va la movida. En cuanto le das a guardar, el sistema ya lo lanza y cualquiera que entre lo va a tener ahí delante a la primera sin dar más vueltas.

### 11.3 Administración de Publicaciones

Cada noticia que vas subiendo viene con sus propios botones para que la información vieja no se te quede ahí entorpeciendo en el sistema. Si ves que metiste la pata en algo o quieres retocar una frase a última hora, le das a "Editar" y listo. También tienes el botón de "Eliminar" para cuando el aviso ya no pinte nada y prefieras quitarlo de en medio para que nadie lo vea. Aparte, con el "Leer más" cualquiera puede pinchar y ver todo el rollo del comunicado si con el título no le ha quedado claro de qué va la vaina. Básicamente es para que la lista de noticias no sea un desmadre y la gente sepa qué es lo que toca en cada momento sin liarse.

### 11.4 Búsqueda de Noticias

El sistema incluye una barra de búsqueda que facilita localizar publicaciones específicas mediante palabras clave. Esta función resulta útil cuando existen múltiples anuncios registrados.




## 12. Configuración del Sistema

El módulo Configuración permite al administrador realizar ajustes generales relacionados con el funcionamiento, identidad y seguridad de la plataforma. Desde esta sección se gestionan los parámetros principales que definen el comportamiento del sistema institucional.

### 12.1 Configuración General

En la pestaña de General es donde manejas lo básico de cómo se ve el sistema por fuera y los datos oficiales que salen en la plataforma. Desde aquí puedes retocar cosas como el nombre oficial del programa, el correo de contacto para que la gente te escriba y hasta subir el logo que más te guste para que represente al torneo. Al final, estos detalles son los que le dan el toque profesional al sistema y se van a ver en casi todas las pantallas y apartados que usen los jugadores o los representantes.

**Proceso de actualización:**

1. Ingresar al módulo *Configuración*.
2. Seleccionar la opción General.
3. Editar la información requerida.
4. Presionar el botón Guardar Cambios.

### 12.2 Configuración Operacional

Desde este apartado puedes tocar los ajustes de fondo que marcan cómo se va a portar el programa y cómo se gestionan los torneos. Básicamente, aquí es donde defines las reglas internas para que todo lo administrativo ruede sin problemas. Lo que puedes hacer aquí es:

- Cambiar cómo se mueve el sistema: para que las funciones se adapten a lo que necesitas cada día.

- Marcar las pautas de la organización: para que los datos institucionales estén bien puestos.
- Que todo lleve el mismo ritmo: para que no haya líos con los registros de los partidos y los torneos.

Básicamente, es el sitio para dejar el programa a punto y que no te dé sustos cuando los usuarios empiecen a ingresar los datos.

**12.3 Seguridad del Sistema**

La sección de Seguridad está ahí para que nadie meta las narices donde no debe y los datos estén bien guardados. Básicamente, desde aquí el administrador lleva el mando de quién entra y quién no. Lo que puedes tocar en este apartado es:

- Las reglas para entrar: para decidir quién tiene permiso y quién se queda fuera.
- Los ajustes de protección: para ponerle candado a las partes más delicadas del programa.
- El control de las cuentas: para vigilar que nadie use el perfil de otro sin permiso.

Al final, todo esto sirve para que no se cuele ningún extraño y que la información que tenemos guardada no corra ningún peligro. Es la forma de dormir tranquilos sabiendo que el sistema está bien cerrado.

**12.4 Mi Cuenta**

Aquí es donde puedes retocar tu ficha personal para que todo esté correcto dentro del programa. Básicamente, tienes un par de botones para que no se te quede la información vieja dando vueltas.

Desde este apartado puedes:

- **Cambiar tus datos:** por si te has equivocado en algo o han cambiado tus señas.
- **Subir otra foto:** para que se te reconozca bien en el perfil y no salga el icono vacío.
- **Llevar el control de tu cuenta:** para que todo lo que tenga que ver contigo esté bien amarrado.

Al final, es el sitio para dejar tu rastro bien puesto y que la cuenta administrativa se vea profesional y actualizada.

**12.5 Guardado de Cambios**

Cualquier cosa que toques dentro del módulo no se va a quedar fija si no le das al botón de "Guardar Cambios". Es fundamental que te acuerdes de pinchar ahí para que el sistema registre los movimientos y no se te pierda nada de lo que acabas de hacer.

Básicamente, es el paso final para que la plataforma se quede con la copla de los cambios y todo aparezca actualizado a la primera.

## 13. Estadísticas del Sistema

En el apartado de Estadísticas tienes resumido todo lo que ha pasado en la pista o en el campo sin tener que echar cuentas a mano. El sistema va pillando los datos de los partidos que se juegan y te saca directamente cómo va la cosa. Básicamente, sirve para que veas de un vistazo quién está rindiendo más y cómo van las competiciones, para que luego no haya dudas a la hora de organizar lo siguiente o premiar a los mejores. Al final, es tener los números claros para saber por dónde van los tiros en el torneo.

### 13.1 Panel de Indicadores Estadísticos

Arriba del todo tienes los números gordos que te dicen cómo va la cosa en general en todo el sistema. Básicamente, el programa te suma solo los goles que se han metido y todos esos pases de gol (las asistencias) que han acabado en jugada. También te saca el conteo de las tarjetas amarillas, por si el juego se está poniendo feo, y las rojas para que veas cuántos se han ido a la calle antes de tiempo. Lo bueno es que no tienes que tocar nada: en cuanto se sube el acta de un partido, estos marcadores se actualizan solos para que no tengas que estar sumando tú con la calculadora.

### 13.2 Consulta de Estadísticas por Jugador

Si necesitas ver los números de alguien específico, tienes un buscador ahí mismo para que no tengas que volverte loco bajando por la lista. Solo tienes que poner el nombre del jugador y el sistema te saca una tabla con todo el detalle: te dice en qué partido ha estado, cuántos goles ha colado, si ha dado algún pase de la muerte (las asistencias) y, por supuesto, si le han sacado alguna tarjeta por pasarse de la raya. Al final, es para que tengas la ficha de cada uno a mano sin perder el tiempo buscando fila por fila.

### 13.3 Funcionamiento del Módulo

El módulo opera de manera automática, tomando datos directamente de:

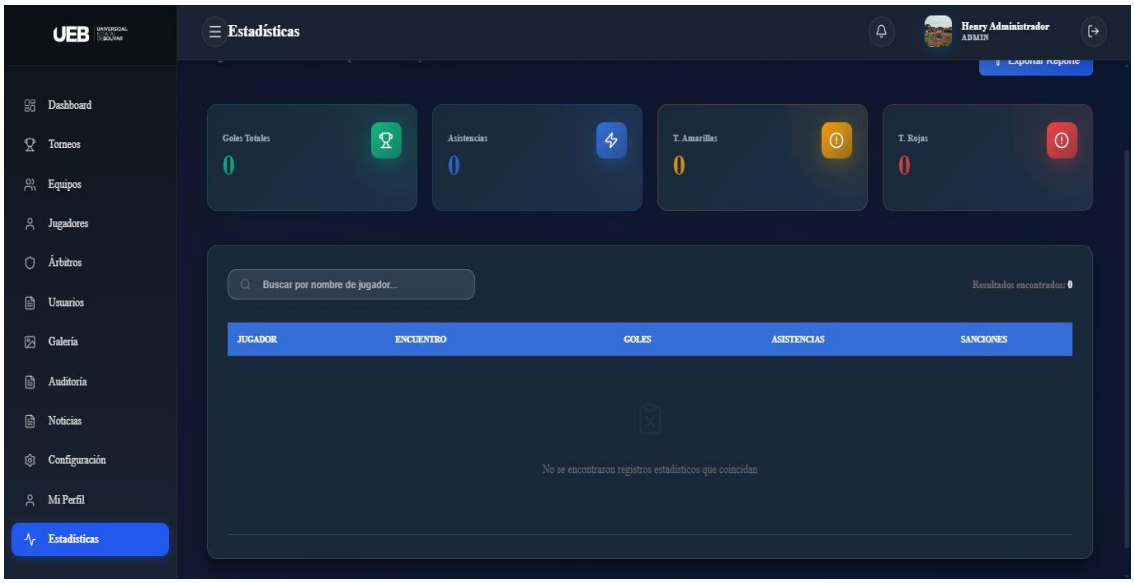
- Resultados de partidos.
- Registros arbitrales.
- Eventos deportivos ingresados en el sistema.

Si no existen datos registrados, el sistema mostrará un mensaje indicando que no se encontraron estadísticas disponibles.

### 13.4 Importancia Administrativa

Al final, tener estas estadísticas a mano te quita un montón de líos de encima porque te permite ver cómo va el nivel del torneo sin tener que adivinar nada. Te sirve para llevar

el control de quién se está portando mal en el campo y poner orden con el tema de las tarjetas. También es súper útil para cuando te toca presentar algún informe oficial, porque ya tienes los números masticados, y sobre todo para comparar quién es el que más se lo está currando o qué equipo va como un tiro comparado con los demás. Básicamente, es la mejor forma de saber qué está pasando de verdad en la competición.




## 14. Cierre de Sesión

Cuando termines de hacer tus cosas, acuérdate de darle a cerrar sesión para que nadie que pase por ahí se meta en tu cuenta sin permiso. Es la forma más fácil de dejar la persiana bajada y que todos los datos del torneo se queden bien guardados y a salvo. Al darle al botón de salida, te aseguras de que el sistema registre bien lo último que has hecho y que la plataforma se quede tranquila hasta la próxima vez que te toque entrar. Básicamente, es por pura seguridad para que nadie te toque nada de lo que ya has dejado listo.

### 14.1 Ubicación de la Opción

La opción Cerrar Sesión se encuentra ubicada en la parte superior derecha de la interfaz, junto a la fotografía y nombre del usuario administrador. Desde este apartado se puede acceder rápidamente a la salida segura del sistema en cualquier momento.

### 14.2 Procedimiento para Cerrar Sesión

Para cerrar la cuenta y quedarte tranquilo, solo tienes que irte a la esquina de arriba a la derecha y pinchar en el icono de salida. No tiene ningún misterio: si te salta un aviso preguntando si de verdad quieres irte, le dices que sí y listo. Luego solo te queda esperar un segundo a que el sistema te mande de vuelta a la pantalla de entrada principal. En cuanto veas que ya estás fuera, puedes estar seguro de que tu sesión se ha cerrado del todo y nadie podrá entrar a tocar nada de lo que has hecho.

### 14.3 Recomendaciones de Seguridad

No te olvides de salir de tu cuenta siempre que termines el día o si te vas a levantar un momento del sitio. Sobre todo, si estás usando un ordenador que toca más gente, es fundamental cerrar la puerta para que nadie que venga después se encuentre con tu perfil abierto. También es buena idea hacerlo justo después de haber tocado ajustes importantes o configuraciones delicadas del sistema, solo por precaución. Al final, es la mejor forma de que nadie ande trasteando donde no debe y que toda la información del torneo se quede tal cual la has dejado tú.

### 14.4 Comportamiento del Sistema

Al cerrar sesión, el sistema ejecuta automáticamente las siguientes acciones:

- Finaliza la autenticación activa.
- Protege la información del usuario.
- Cancela permisos temporales de acceso.
- Redirige al formulario de inicio de sesión.

Para volver a ingresar será necesario introducir nuevamente las credenciales autorizadas.

**Anexo 6.** Manual de Usuario

**UNIVERSIDAD ESTATAL DE BOLÍVAR**

**FACULTAD DE CIENCIAS ADMINISTRATIVAS GESTIÓN EMPRESARIAL E INFORMÁTICA**
**CARRERA DE INGENIERÍA EN SOFTWARE**

ANEXO 5

# MANUAL DE USUARIO

## Sistema de Gestión de Campeonatos Deportivos

**2026**

**Índice**

Portada.....
1

1. Introducción.....105
2. Requisitos del Sistema.....105

2.1 Requisitos del Hardware.....105

2.2 Requisitos del Software .....105

2.3 Credenciales de Acceso .....106

3. Acceso al Sistema .....106

3.1 Panel Principal del Representante.....107

4. Calendario de Participación.....108
5. Nómina de Jugadores.....109
6. Gestión de Equipos.....111
7. Inscripciones .....112
8. Partidos.....113
9. Perfil del Usuario.....114
10. Estadísticas.....115
11. Cerrar Sesión.....116

# 1. Introducción

El presente Manual de Usuario tiene como propósito orientar de manera detallada el uso del Sistema de Gestión de Campeonatos Deportivos, enfocándose exclusivamente en los roles operativos de la plataforma: Representante, Jugador (gestión del jugador) y Árbitro. Este documento es básicamente tu hoja de ruta para que no te pierdas con ninguna función y sepas cómo manejar todo lo que pasa en el sistema. La idea de montar esta plataforma fue jubilar de una vez el papeleo y los líos de llevar los campeonatos a mano. Al digitalizarlo todo, te quitas de encima los típicos fallos tontos de administración y, lo mejor de todo, es que cualquiera puede consultar los datos del torneo sin que haya cosas raras. Al final, es para que todo sea mucho más transparente y no tengas que estar dando vueltas buscando papeles cada vez que alguien te pregunte algo.

Este manual describe las funcionalidades asociadas a la gestión de jugadores, el control de equipos por parte de los representantes y el registro de resultados por parte de los árbitros. La información se presenta de manera progresiva, permitiendo que el usuario identifique con claridad las acciones que puede realizar según su rol dentro del sistema.

## 1.1 Requisitos del Sistema

Para un funcionamiento adecuado del sistema, el usuario debe contar con los siguientes elementos técnicos mínimos:

### a. Requisitos del Hardware

Se recomienda el uso de un equipo que permita una navegación estable, como:

- Computadora de escritorio
- Computadora portátil
- Tablet

El dispositivo debe poseer capacidad suficiente para visualizar interfaces gráficas modernas y facilitar la interacción con formularios, tablas de datos y elementos visuales.

### b. Requisitos del Software

- Navegador web actualizado (Google Chrome, Mozilla Firefox o Microsoft Edge).
- Sistema operativo compatible con navegación web moderna.

- Conexión estable a internet, necesaria para la comunicación continua con el servidor.

El sistema opera en entorno web, por lo que no requiere instalación adicional.

### c. Credenciales de Acceso

Cada usuario debe disponer de credenciales personales:

- Correo electrónico
- Contraseña

Estas credenciales permiten identificar al usuario y habilitar únicamente las funciones asociadas a su rol.

## 1.2 Acceso al Sistema

El acceso se realiza desde la página principal de la plataforma.

Procedimiento de ingreso:

1. Acceder a la dirección web del sistema.




2. Seleccionar la opción Acceso a Clubes.



3. Ingresar correo electrónico y contraseña.
4. Confirmar el ingreso.

El sistema valida la información y muestra el panel correspondiente al rol asignado (Representante, Árbitro, Jugador). En caso de error, se solicitará reingresar los datos. Por seguridad, se recomienda cerrar sesión al finalizar.

**a. Panel Principal del Representante**

Una vez que el representante ingresa correctamente al sistema, se muestra el panel principal o Dashboard, el cual constituye el centro de control de las actividades deportivas relacionadas con sus equipos. Hemos montado esta pantalla para que, nada más entrar, tengas todo lo importante a la vista y no pierdas el tiempo dando vueltas por los menús. Arriba del todo verás que el sistema te saluda por tu nombre, lo cual es la mejor señal de que has entrado bien con tu cuenta y que el programa ya sabe quién eres y qué cosas puedes tocar. La idea es que, de un vistazo, sepas cómo va la vaina y puedas saltar directo a lo que necesites hacer sin que la interfaz te estorbe. Al final, es tu centro de mandos para que gestiones el torneo de forma rápida y sin líos.

El panel muestra tarjetas informativas que resumen el estado actual de la gestión deportiva:

- **Mis Equipos:** Indica la cantidad de equipos registrados bajo la responsabilidad del representante.
- **Jugadores:** Muestra el número total de jugadores inscritos en la nómina.
- **Torneos Activos:** Presenta los campeonatos en los que participan los equipos.
- **Próximos Partidos:** Señala encuentros pendientes de disputarse.

Estas tarjetas permiten al representante obtener una visión general inmediata sin necesidad de ingresar a cada módulo por separado.

En la parte lateral izquierda se encuentra el menú de navegación, el cual organiza las funciones disponibles:

- Dashboard
- Calendario
- Nómina
- Equipos
- Inscripciones
- Partidos
- Perfil

- Estadísticas

Este menú facilita el acceso directo a cada sección del sistema, manteniendo una estructura intuitiva que reduce el tiempo de aprendizaje del usuario.

En la zona central e inferior del panel también se presentan secciones de apoyo, como el listado de próximos partidos y el rendimiento general de los equipos, permitiendo un seguimiento continuo de la participación deportiva.




Fuente: Elaboración propia (2026).

**2. Calendario de Participación**

El módulo Calendario permite al usuario visualizar de manera organizada el cronograma de encuentros deportivos programados dentro del campeonato. Esta sección facilita la planificación de la participación, permitiendo conocer con anticipación fechas, horarios y detalles de los partidos.

El acceso al calendario se realiza desde el menú lateral del sistema, seleccionando la opción “Calendario”. Al ingresar, el usuario visualiza una interfaz gráfica estructurada por fechas, diseñada para ofrecer una comprensión clara de la distribución de los encuentros deportivos a lo largo del tiempo.

El sistema ofrece diferentes vistas de consulta:

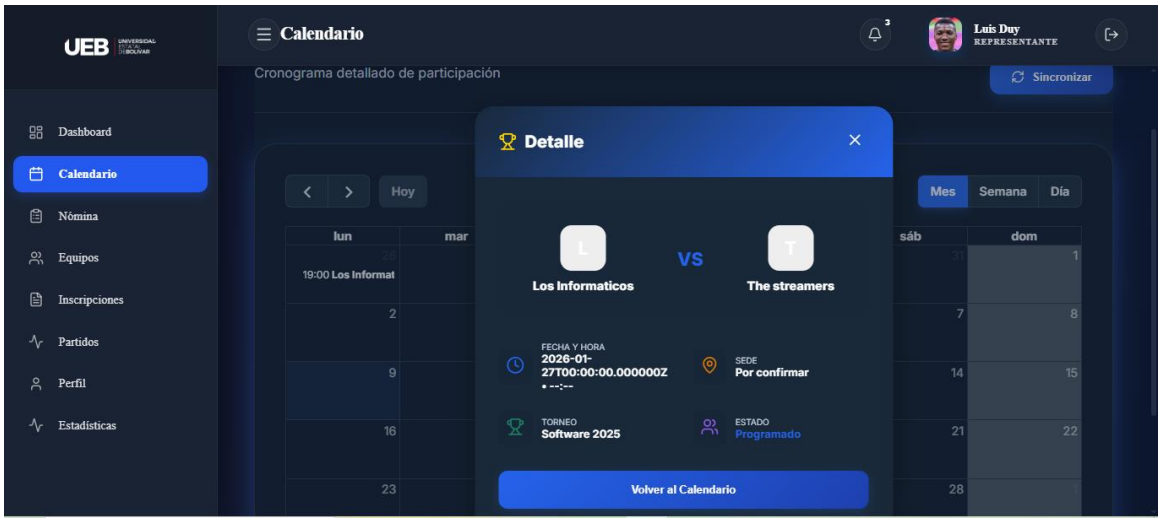
- Vista mensual, que muestra una planificación general de los partidos programados durante el mes.
- Vista semanal, que permite un seguimiento más detallado de los encuentros en un período corto.
- Vista diaria, enfocada en los partidos correspondientes a una fecha específica.

Cada partido aparece registrado como un evento dentro del calendario. Al seleccionar uno de estos eventos, se despliega una ventana de detalle del encuentro, donde el usuario puede consultar información relevante como:

- Equipos participantes.
- Fecha y hora del partido.
- Torneo al que pertenece.
- Estado del encuentro (programado, en curso o finalizado).
- Sede del partido, cuando esté definida.

Al representante del equipo este calendario le da la vida, porque así sabe de antemano cuándo le toca jugar y puede avisar a los chavales con tiempo para que se organicen. Básicamente, es su guía para que no le falte nadie el día del partido y la logística no sea un desastre. Por otro lado, el árbitro también lo usa para no liarse con sus horarios, sabiendo exactamente qué partidos le han asignado y a qué hora tiene que estar allí para pitar.

Además, el módulo viene con unos botones para ir saltando de un día a otro o de una semana a otra, por si quieres bichear partidos que ya pasaron o ver qué se viene próximamente. Al final, es para que todo el mundo esté al tanto y nadie se quede con información vieja que ya no valga para nada.El uso adecuado del calendario contribuye a una gestión deportiva planificada, evitando confusiones en horarios y mejorando la coordinación entre los participantes del campeonato.




## 4. Nómina de Jugadores

El módulo Nómina de Jugadores permite al representante administrar de forma centralizada la información de los deportistas que integran sus equipos. Esta sección es

esencial para el control de participación en el campeonato, ya que valida qué jugadores están habilitados para competir.

El acceso a este módulo se realiza desde el menú lateral, seleccionando la opción "Nómina". Al ingresar, el sistema presenta un panel con información general y un listado detallado de los jugadores registrados.

**Información general de la nómina**

En la parte superior de la pantalla se muestran indicadores que resumen el estado de la nómina, entre ellos:

- Total, de jugadores registrados.
- Número de equipos con nómina activa.
- Estado documental de los jugadores, reflejado mediante un porcentaje que indica el nivel de cumplimiento de requisitos.

**Listado de jugadores**

La sección principal muestra una tabla con los jugadores registrados, donde se incluyen datos como:

- Nombre del jugador.
- Número de identificación.
- Número de dorsal.
- Posición de juego.
- Equipo al que pertenece.

El sistema permite realizar búsquedas y filtros para localizar jugadores de manera ágil, facilitando la gestión cuando la nómina es amplia.

**Registro de nuevos jugadores**

Mediante la opción "Registrar Jugador", el representante puede añadir nuevos deportistas al sistema. Para ello, debe completar la información solicitada, incluyendo datos personales y deportivos. El sistema valida la información antes de almacenarla.

**Acciones disponibles**

Para cada jugador registrado, el sistema permite:

- Editar información.
- Visualizar detalles.
- Eliminar registros cuando sea necesario y permitido por las reglas del sistema.

### Importación y exportación de datos

El módulo incorpora herramientas para importar información mediante archivos CSV y exportar datos en distintos formatos, lo que facilita el respaldo y la gestión administrativa.

Este módulo garantiza un control ordenado de los jugadores y asegura que la participación en los encuentros deportivos se realice con información verificada y actualizada.

## 5. Gestión de Equipos

El módulo Gestión de Equipos permite al representante registrar, organizar y administrar los equipos que participan en el campeonato. Este apartado es esencial, ya que cada equipo constituye la base para la inscripción de jugadores y su posterior participación en los encuentros deportivos.

El acceso se realiza desde el menú lateral seleccionando la opción "Equipos". Al ingresar, el sistema presenta un área de registro y un listado de los equipos creados.

### Registro de un nuevo equipo

En la sección "Registrar Equipo", el representante puede crear un nuevo club dentro del sistema. Para ello, debe completar la información solicitada:

- **Nombre del club:** identificación oficial del equipo.
- **Torneo destino:** campeonato en el que participará el equipo.
- **Deporte:** disciplina deportiva correspondiente.
- **Categoría:** categoría de participación (por ejemplo, masculina, femenina u otras).

Una vez ingresados los datos, se selecciona la opción "Crear Equipo". El sistema valida la información antes de guardar el registro, asegurando coherencia y evitando duplicidades.

### Listado de equipos

En la parte principal de la pantalla se visualizan los equipos registrados, presentados mediante tarjetas informativas que incluyen:

- Nombre del equipo.
- Deporte y categoría.
- Número de jugadores registrados.
- Torneo asociado.

- Estado del equipo dentro del campeonato.

Estas tarjetas permiten identificar rápidamente la situación de cada equipo.

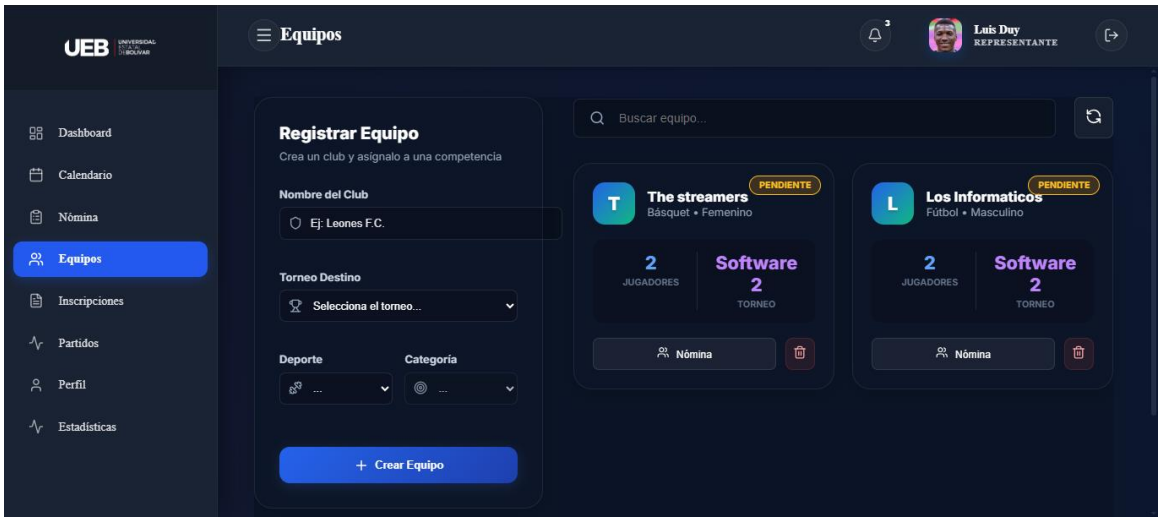
### Gestión de la nómina del equipo

Cada equipo dispone de la opción “Nómina”, que permite acceder directamente a los jugadores asociados. Desde allí se puede administrar la lista de integrantes, facilitando la organización deportiva.

### Acciones disponibles

El sistema permite realizar diversas acciones sobre los equipos registrados, como:

- Consultar información detallada.
- Acceder a la nómina de jugadores.
- Eliminar un equipo, siempre que las condiciones del sistema lo permitan.




## 6. Inscripciones

El módulo Inscripciones permite al representante gestionar la participación formal de sus equipos en los torneos disponibles dentro del sistema. Esta sección es fundamental, ya que a través de ella se solicita y registra la habilitación oficial de un equipo para competir.

El acceso se realiza desde el menú lateral seleccionando la opción “Inscripciones”. Al ingresar, el sistema presenta dos áreas principales: registro de nuevas solicitudes e historial de inscripciones.

### Nueva inscripción

En la sección “Nueva Inscripción”, el representante puede registrar una solicitud de participación para uno de sus equipos. Para ello, debe completar los siguientes campos:

- **Equipo representativo:** selección del equipo previamente creado.
- **Torneo de destino:** campeonato en el que se desea inscribir al equipo.

Una vez seleccionados los datos, se utiliza la opción “Confirmar Solicitud”, con lo cual el sistema registra la petición y la envía al proceso de validación correspondiente.

**Proceso de validación**

Las solicitudes de inscripción son revisadas por la entidad organizadora del campeonato. El sistema informa al usuario que la aprobación está sujeta a la verificación de requisitos, entre ellos:

- Cumplimiento de la nómina completa de jugadores.
- Validación documental.
- Revisión administrativa.

Durante este proceso, la solicitud puede encontrarse en distintos estados, como pendiente, aprobada o rechazada.

**Historial de inscripciones**

En la parte inferior se muestra el historial de inscripciones, donde el representante puede consultar todas las solicitudes realizadas. En esta tabla se visualiza:

- Nombre del equipo.
- Torneo correspondiente.
- Estado de la inscripción.
- Fecha de registro.

Esta sección permite llevar un control ordenado de la participación de los equipos en los diferentes campeonatos.

**Filtros de consulta**

El módulo incluye herramientas de filtrado que facilitan la búsqueda de inscripciones según equipo o estado, permitiendo localizar información de forma rápida.

## 7. Partidos

El módulo Partidos permite a los usuarios consultar los encuentros deportivos programados dentro del campeonato. El acceso se realiza desde el menú lateral seleccionando la opción “Partidos”. Al ingresar, el sistema muestra un listado de

encuentros organizados por torneo, presentados de forma clara mediante tarjetas informativas.

### Visualización de encuentros

Cada partido se presenta con información relevante que permite identificar rápidamente sus características principales:

- Nombre del torneo al que pertenece.
- Equipos participantes.
- Fecha y hora programada.
- Estado del partido (por ejemplo, programado).
- Información de la sede cuando esté definida.

Esta visualización permite al usuario conocer el cronograma de participación y mantener un control actualizado de los encuentros.

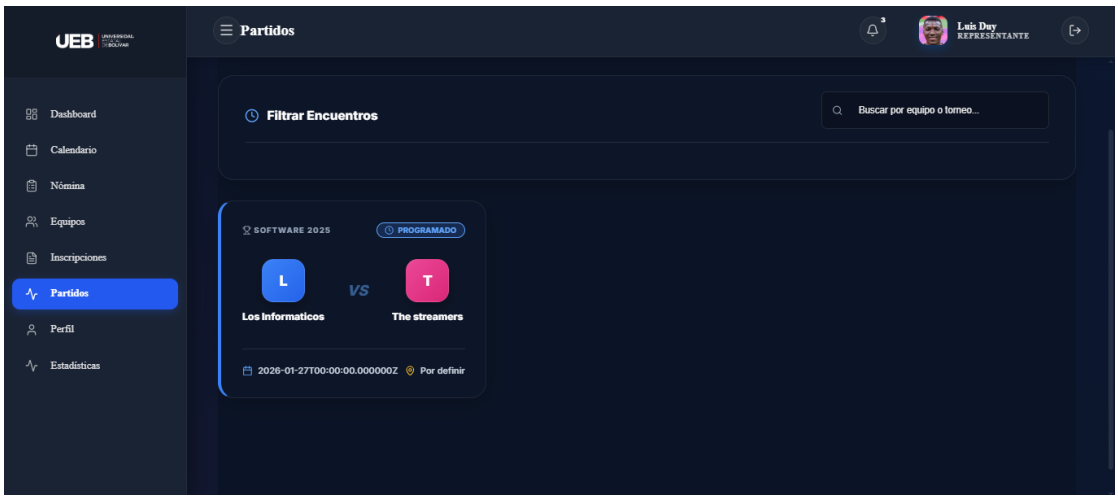
### Búsqueda y filtrado

El módulo incorpora una herramienta de búsqueda que permite localizar partidos mediante el nombre de un equipo o torneo. Esto facilita el acceso rápido a la información, especialmente cuando el número de encuentros es elevado.

### Función para el representante

El representante utiliza este módulo para:

- Consultar los partidos en los que participan sus equipos.
- Verificar fechas, horarios y estado de los encuentros.
- Planificar la participación deportiva.




## 5. Perfil del Usuario

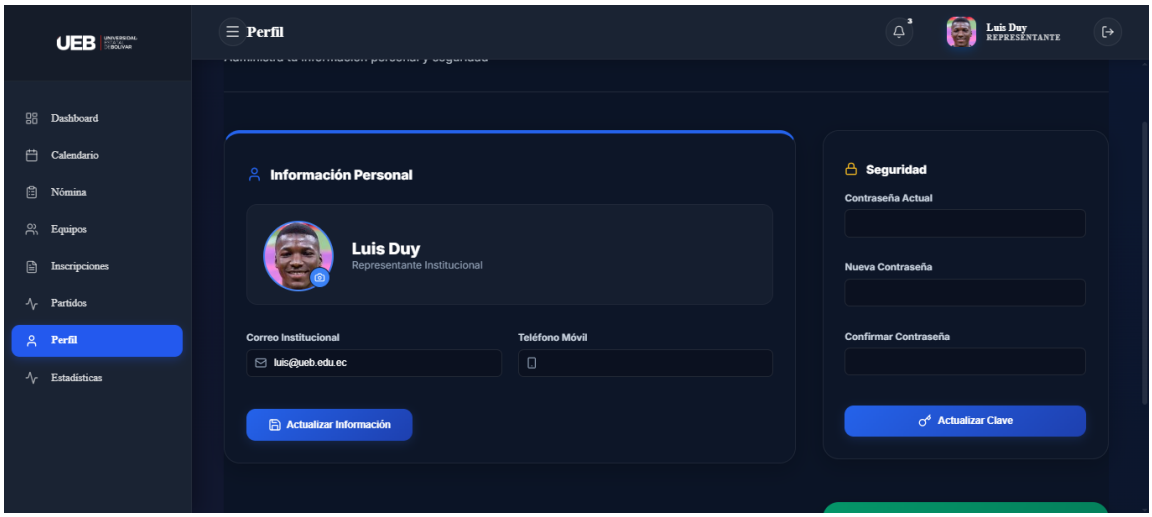
Para entrar a ver tu perfil, solo tienes que irte al menú del lado y pinchar donde pone “Perfil”. Verás que la pantalla se divide en dos: una parte para tus datos de siempre y otra para que le pongas candado a la entrada con una buena contraseña.

### Tus datos personales

Aquí sale quién eres en el sistema: tu nombre, el cargo que te han dado, tu correo y el móvil. Si ves que algo ha cambiado o que el número de teléfono ya no es el que usas, cámbialo sin miedo. Solo acuérdate de darle al botón de “Actualizar Información” para que el sistema se quede con la copla. Tener esto al día es clave para que, si pasa algo en el torneo, te puedan localizar rápido y sin líos.

En la parte de seguridad es donde puedes cambiar la clave. Si sientes que la que tienes es flojilla o alguien la ha visto, cámbiala metiendo la que usas ahora y luego la nueva un par de veces para no equivocarte. Cuando termines, dale a “Actualizar Clave” y listo, ya tienes el acceso blindado.

No le pases tu contraseña a nadie ni la dejes apuntada en un post-it pegado a la pantalla. El acceso es tuyo y de nadie más, así que cuídalo para que no te toquen los datos del equipo.




## 6. Estadísticas

El módulo Estadísticas permite al representante visualizar el desempeño deportivo de sus equipos dentro del campeonato. Esta sección ofrece un análisis general de resultados y métricas que facilitan el seguimiento del rendimiento.

El acceso se realiza desde el menú lateral seleccionando la opción Estadísticas. Al ingresar, el sistema presenta un panel con indicadores y resúmenes gráficos relacionados con la participación del equipo.

### Indicadores principales

En la parte superior se muestran métricas generales, entre las que se incluyen:

- **Goles a favor:** cantidad de anotaciones realizadas por el equipo.
- **Posición en la tabla:** ubicación actual dentro de la clasificación del torneo.

Estos datos permiten evaluar de forma inmediata el nivel de desempeño deportivo.

### Balance de encuentros

El sistema presenta un resumen del historial de partidos disputados, organizado en tres categorías:

- Partidos ganados.
- Partidos empatados.
- Partidos perdidos.

Este balance permite analizar la consistencia del equipo a lo largo de la competencia.

### Logros

El módulo incluye una sección de logros o reconocimientos obtenidos, los cuales pueden estar relacionados con el rendimiento, disciplina o participación destacada dentro del torneo.

## 7. Cerrar Sesión

El sistema incorpora la opción de cierre de sesión como una medida de seguridad destinada a proteger la información personal y deportiva registrada en la plataforma.

Para cerrar la sesión, el usuario debe dirigirse a la parte superior de la interfaz, donde se encuentra el área correspondiente a su perfil. Desde allí, se selecciona la opción de salida del sistema, lo cual finaliza el acceso activo y evita que otras personas utilicen la cuenta sin autorización.

Se recomienda realizar el cierre de sesión en los siguientes casos:

- Al finalizar el uso del sistema.
- Cuando se accede desde dispositivos compartidos.
- Cuando se realiza la gestión de información sensible.

Al cerrar sesión, el sistema redirige automáticamente a la página de inicio, requiriendo nuevamente las credenciales para acceder. Esta acción contribuye a mantener la confidencialidad de los datos y garantiza un uso seguro de la plataforma.

**Anexo 7.** Certificado Antiplagio

UEB
UNIVERSIDAD ESTATAL DE BOLIVAR

FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN EMPRESARIAL E INFORMÁTICA

**ING. EDGAR RIVADENEIRA EN CALIDAD DE DIRECTOR(A) DEL TRABAJO DE INTEGRACIÓN CURRICULAR,**

## CERTIFICA

Que el trabajo de integración curricular denominado "SISTEMA INTEGRAL DE GESTIÓN PARA CAMPEONATOS DEPORTIVOS EN LA UNIVERSIDAD ESTATAL DE BOLÍVAR", presentado por el Sr. Stalin Alexander Alvarado Chongo y el Sr. Luis Aurelio Duy Quindi, estudiantes de la **carrera de Software** pasó el análisis de coincidencia no accidental en la herramienta COMPILATIO, reflejando un **porcentaje de similitud del <1%**, como se puede evidenciar en el documento adjunto.

Guaranda, 26 de febrero del 2026

Atentamente,

Ing. Edgar Rivadeneira
**Director**

Dirección: Av. Ernesto Che Guevara y Gabriel Secaira
Guaranda-Ecuador
Teléfono: (593) 3220 6059
www.ueb.edu.ec

INFORME DE ANÁLISIS
*magister*

# Revisión_Stalin_Alvarado

**11%**
**Textos sospechosos**

- **< 1% Similitudes**
  - 0 % similitudes entre comillas
  - 0 % entre las fuentes mencionadas
- **2% Idiomas no reconocidos**
- **9% Textos potencialmente generados por la IA**

**Nombre del documento:** Revisión_Stalin_Alvarado.docx
**ID del documento:** c8d43a08efdc3d910576272b1c5934daaed7819a
**Tamaño del documento original:** 10,94 MB

**Depositante:** EDGAR PATRICIO RIVADENEIRA RAMOS
**Fecha de depósito:** 26/2/2026
**Tipo de carga:** interface
**fecha de fin de análisis:** 26/2/2026

**Número de palabras:** 18.444
**Número de caracteres:** 122.348

Ubicación de las similitudes en el documento:

## Fuentes de similitudes

**Fuentes con similitudes fortuitas**

| Nº | Descripciones | Similitudes | Ubicaciones | Datos adicionales |
|---|---|---|---|---|
| 1 | **Documento de otro usuario** #2db628<br>Viene de de otro grupo | < 1% | | Palabras idénticas: < 1% (11 palabras) |
| 2 | **es.slideshare.net** \| Prototipo de-sistema-para-matricula-e-inscripcion-de-asignat...<br>https://es.slideshare.net/slideshow/prototipo-desistemaparamatriculaeinscripciondeasignatu... | < 1% | | Palabras idénticas: < 1% (10 palabras) |
| 3 | **repository.unad.edu.co** \| Implementación de una Mesa de Ayuda para la optimiz...<br>https://repository.unad.edu.co/handle/10596/75518 | < 1% | | Palabras idénticas: < 1% (13 palabras) |
| 4 | **Documento de otro usuario** #65b02b<br>Viene de de otro grupo | < 1% | | Palabras idénticas: < 1% (10 palabras) |
| 5 | **ribuni.uni.edu.ni** \| Desarrollo de un sistema web para la gestión educativa del Pr...<br>http://ribuni.uni.edu.ni/6845/1/200311.pdf | < 1% | | Palabras idénticas: < 1% (12 palabras) |

**Fuentes mencionadas (sin similitudes detectadas)** Estas fuentes han sido citadas en el documento sin encontrar similitudes.

1. https://doi.org/10.5678/rtu.2023.11233
2. https://doi.org/10.1234/rti.2024.15204
3. https://doi.org/10.1177/0165551506072165
4. https://www.nngroup.com/articles/usability-101-introduction-to-usability/
5. https://doi.org/10.5678/rit.2021.18306

**Anexo 8.** Link del repositorio digital de biblioteca donde fue subido el proyecto