



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE SOFTWARE

Tema:

**SOFTWARE COMO SERVICIO PARA OPTIMIZAR LA GESTIÓN DE
CAMPEONATOS EN LIGAS DEPORTIVAS CON RECURSOS
ECONÓMICOS LIMITADOS**

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la
obtención del título de Ingeniero de Software

ÁREA: Desarrollo en software

LÍNEA DE INVESTIGACION: Software, Tecnologías de la información y Ciencia
de datos

AUTOR: Jhon Alexander Lescano Cisneros

TUTOR: Ing. Santiago David Jara Moya, Mg.

Ambato - Ecuador

julio - 2025

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: SOFTWARE COMO SERVICIO PARA OPTIMIZAR LA GESTIÓN DE CAMPEONATOS EN LIGAS DEPORTIVAS CON RECURSOS ECONÓMICOS LIMITADOS, desarrollado bajo la modalidad Proyecto de Investigación, por el señor Jhon Alexander Lescano Cisneros, estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, julio 2025.



Ing. Santiago David Jara Moya, Mg.
TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: SOFTWARE COMO SERVICIO PARA OPTIMIZAR LA GESTIÓN DE CAMPEONATOS EN LIGAS DEPORTIVAS CON RECURSOS ECONÓMICOS LIMITADOS es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, julio 2025.



Jhon Alexander Lescano Cisneros

C.C. 1724846686

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, julio 2025.



Jhon Alexander Lescano Cisneros

C.C. 1724846686

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Jhon Alexander Lescano Cisneros, estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SOFTWARE COMO SERVICIO PARA OPTIMIZAR LA GESTIÓN DE CAMPEONATOS EN LIGAS DEPORTIVAS CON RECURSOS ECONÓMICOS LIMITADOS nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señor Presidente del Tribunal.

Ambato, julio 2025.



Ing. Franklin Oswaldo Mayorga Mayorga, Mg.
PRESIDENTE DEL TRIBUNAL



Ing. Clay Aldas Flores Mg.
PROFESOR CALIFICADOR



Ing. Xavier López Andrade Mg.
PROFESOR CALIFICADOR

DEDICATORIA

A mi madre, Blanca, por su constante apoyo y ejemplo de perseverancia. A mi padre, Armando, por su disciplina y enseñanzas que marcaron mi formación. A mi hermano, Cristhofer, por su compañía y motivación durante este proceso. A mis abuelitos, cuyo legado de esfuerzo y valores ha sido fundamental en mi vida. A toda mi familia, mi más sincero agradecimiento por ser el pilar que me permitió alcanzar esta meta académica.

Jhon Alexander Lescano Cisneros

AGRADECIMIENTOS

Agradezco a mi madre, Blanca, por su incansable apoyo y por impulsarme a continuar mi formación académica. A mi padre, Armando, y a mi hermano, Cristhofer, por su respaldo incondicional en cada etapa de este proceso. A mis abuelos, cuyos consejos y valores han sido fundamentales en mi camino. A mis amigos, por convertir esta etapa universitaria en una experiencia memorable. Reconozco también la dedicación de cada profesor que compartió sus conocimientos y enriqueció mi aprendizaje. Un agradecimiento especial a mi tutor, cuya orientación experta fue fundamental para culminar este trabajo.

Jhon Alexander Lescano Cisneros

ÍNDICE GENERAL DE CONTENIDOS

PORTADA	i
APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL DE CONTENIDOS	viii
ÍNDICE DE TABLAS	x
ÍNDICE DE FIGURAS	xi
ÍNDICE DE ANEXOS	xii
RESUMEN EJECUTIVO	xiii
ABSTRACT	xiv
CAPÍTULO I. MARCO TEÓRICO	1
1.1 Tema de investigación	1
1.1.1 Planteamiento del problema	1
1.1.2 Antecedentes investigativos	2
1.2 Fundamentación teórica	16
1.2.1 Modelos de negocio	16
1.2.2 Gestión de campeonatos deportivos	18
1.2.3 Tecnologías web modernas	19
1.3 Objetivo general	22
1.4 Objetivos específicos	22

CAPÍTULO II. METODOLOGÍA	23
2.1 Materiales	23
2.2 Métodos	23
2.2.1 Modalidad de la investigación	23
2.2.2 Población y muestra	24
2.2.3 Recolección de información	26
2.2.4 Procesamiento y análisis de datos	42
CAPÍTULO III. RESULTADOS Y DISCUSIÓN	45
3.1 Investigación y selección de herramientas	45
3.1.1 Selección de herramientas de desarrollo	45
3.1.2 Selección de herramientas de implementación	65
3.1.3 Selección de la metodología de desarrollo de software	70
3.2 Análisis de los procesos actuales	73
3.2.1 Procesos identificados en la gestión de campeonatos	73
3.2.2 Análisis de tiempos actuales y procesos a optimizar	77
3.3 Desarrollo de la propuesta	80
3.3.1 Estructura de RAD para el proyecto	81
3.3.2 Fase 1: Planificación de requisitos	82
3.3.3 Fase 2: Diseño del Usuario (User Design)	85
3.3.4 Fase 3: Construcción (Construction)	93
3.3.5 Fase 4: Implementación (Cutover)	106
3.4 Análisis de tiempos: manual vs automatizado	111
3.5 Evaluación técnica y económica	114
3.5.1 Evaluación técnica	114
3.5.2 Evaluación económica	126
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES	131
4.1 Conclusiones	131
4.2 Recomendaciones	132
REFERENCIAS BIBLIOGRÁFICAS	137
ANEXOS	138

ÍNDICE DE TABLAS

Tabla 1.	Resultados de la investigación.	3
Tabla 2.	Artículos revisados.	4
Tabla 3.	Comparación de modelos de negocio de diferentes artículos.	13
Tabla 4.	Comparación de metodologías ágiles de diferentes artículos.	15
Tabla 5.	Comparación de herramientas y tecnologías de diferentes artículos.	15
Tabla 6.	Proceso de recolección de la información.	26
Tabla 7.	Resultados de la pregunta 1.	27
Tabla 8.	Resultados de la pregunta 2.	28
Tabla 9.	Resultados de la pregunta 3.	30
Tabla 10.	Resultados de la pregunta 5.	31
Tabla 11.	Resultados de la pregunta 6.	33
Tabla 12.	Resultados de la pregunta 7.	34
Tabla 13.	Resultados de la pregunta 9.	36
Tabla 14.	Resultados de la pregunta 10.	37
Tabla 15.	Resultados de la pregunta 11.	38
Tabla 16.	Evaluación de las respuestas obtenidas en la entrevista.	40
Tabla 17.	Características y soporte de herramientas para monorepos.	46
Tabla 18.	Matriz de decisión para herramientas de monorepos.	47
Tabla 19.	Características y beneficios de los lenguajes de programación.	48
Tabla 20.	Matriz de decisión para lenguajes.	48
Tabla 21.	Comparación de rendimiento y extensibilidad.	51
Tabla 22.	Comparación general de metaframeworks.	52
Tabla 23.	Comparación de SEO y popularidad.	53
Tabla 24.	Comparación de complejidad y escalabilidad.	54
Tabla 25.	Matriz de decisión para metaframeworks.	54
Tabla 26.	Comparación de métodos de estilización en frameworks.	57
Tabla 27.	Resumen de hallazgos principales.	57
Tabla 28.	Matriz de decisión para frameworks CSS.	58
Tabla 29.	Comparación entre Firebase y Supabase.	58
Tabla 30.	Comparación de factores clave entre Firebase y Supabase.	59
Tabla 31.	Matriz de decisión para BaaS.	60
Tabla 32.	Comparación entre tRPC, gRPC y GraphQL.	61

Tabla 33.	Matriz de decisión para protocolos API.	62
Tabla 34.	Comparación entre ORMs	64
Tabla 35.	Matriz de decisión para ORMs.	64
Tabla 36.	Comparación de recursos y tiempos de despliegue.	66
Tabla 37.	Comparación de características adicionales.	66
Tabla 38.	Comparación de planes gratuitos y limitaciones.	66
Tabla 39.	Comparación de rendimiento en métricas clave.	66
Tabla 40.	Matriz de decisión para plataformas de despliegue.	67
Tabla 41.	Comparación de proveedores de pagos: Stripe, Lemon Squeezy, Polar y Creem.	68
Tabla 42.	Resumen comparativo de proveedores de pagos.	68
Tabla 43.	Matriz de decisión para sistemas de pagos.	69
Tabla 44.	Comparación entre RAD, XP, Scrum y Kanban.	71
Tabla 45.	Matriz de decisión para metodologías de desarrollo.	72
Tabla 46.	Tiempos actuales en el proceso de inscripción de equipos y jugadores.	79
Tabla 47.	Tiempos actuales en el proceso de programación de partidos.	79
Tabla 48.	Tiempos actuales en el proceso de registro de resultados.	79
Tabla 49.	Tiempos actuales en el proceso de gestión de sanciones.	80
Tabla 50.	Descripción de los usuarios identificados.	84
Tabla 51.	Identificación de requerimientos del sistema por iteraciones.	84
Tabla 52.	Plan de trabajo de iteraciones.	86
Tabla 53.	Pruebas de aceptación - Iteración 1 (Core).	107
Tabla 54.	Pruebas de aceptación - Iteración 2 (Complejidad).	108
Tabla 55.	Pruebas de aceptación - Iteración 3 (Innovación).	108
Tabla 56.	Comparativa de tiempos de ejecución y satisfacción de usuario.	113
Tabla 57.	Comparativa alineada a requisitos funcionales de nuestro sistema.	123
Tabla 58.	Inversión inicial en el desarrollo del sistema.	126
Tabla 59.	Costos recurrentes de operación del sistema.	127
Tabla 60.	Estructura de planes y precios.	127
Tabla 61.	Ingresos mensuales actuales.	128
Tabla 62.	Proyección de crecimiento de usuarios de pago (3 años).	128
Tabla 63.	Indicadores de rentabilidad a 5 años.	129
Tabla 64.	Flujo acumulado y período de recuperación.	129

ÍNDICE DE FIGURAS

Figura 1.	Distribución de roles entre los participantes.	27
Figura 2.	Resultados de la encuesta en la pregunta 2.	28
Figura 3.	Resultados de la encuesta en la pregunta 2.	29
Figura 4.	Dificultades reportadas por los participantes.	30
Figura 5.	Resultados de la encuesta de la pregunta 5.	31
Figura 6.	Resultados de la encuesta de la pregunta 6.	32
Figura 7.	Resultados de la encuesta de la pregunta 7.	33
Figura 8.	Funciones más solicitadas por los participantes.	35
Figura 9.	Resultados de la encuesta de la pregunta 9.	36
Figura 10.	Resultados de la encuesta de la pregunta 10.	37
Figura 11.	Resultados de la encuesta de la pregunta 11.	38
Figura 12.	Tiempos de procesamiento de los lenguajes de programación. . . .	49
Figura 13.	Diagrama de flujo de procesos actual de inscripción de equipos. .	75
Figura 14.	Diagrama de flujo de procesos actual de programación de partidos.	76
Figura 15.	Diagrama de flujo de procesos actual de el registro de resultados. .	77
Figura 16.	Arquitectura por capas.	88
Figura 17.	Interfaz de iniciar sesión.	89
Figura 18.	Interfaz de gestión de torneos.	89
Figura 19.	Interfaz de gestión de equipos.	90
Figura 20.	Interfaz de gestión de jugadores.	90
Figura 21.	Interfaz de programación de partidos.	91
Figura 22.	Interfaz de gestión de estadísticas en partidos.	91
Figura 23.	Interfaz de gestión de sanciones.	92
Figura 24.	Interfaz de gestión de reportes.	92
Figura 25.	Interfaz de gestión de pagos.	93
Figura 26.	Esquema de los modelos en PrismaORM.	95
Figura 27.	Arquitectura backend.	95
Figura 28.	Conexión con Supabase.	96
Figura 29.	Flujo de autenticación.	96
Figura 30.	Flujo de estadísticas en tiempo real.	97
Figura 31.	Arquitectura de procesamiento de pagos.	97
Figura 32.	Seguridad a nivel de fila.	98

Figura 33.	Creación del cliente para conectar al bucket de storage.	98
Figura 34.	Archivo package.json con las dependencias del sistema.	100
Figura 35.	Estructura de carpetas del monorepo.	101
Figura 36.	Paleta de colores usando HSL.	102
Figura 37.	Fetching de datos con trpc en el servidor.	102
Figura 38.	Configuración para arrastrar y soltar de partidos.	103
Figura 39.	Componente de Tanstack table.	104
Figura 40.	Visualización del calendario por jornadas.	104
Figura 41.	Query para la creación de estadísticas y el llamado de la función de realtime.	105
Figura 42.	Marketing integrado para compartir en redes sociales.	106
Figura 43.	Despliegue del repositorio en Vercel.	109
Figura 44.	Resultados de métricas Web Vitals.	115
Figura 45.	Informe de prueba de concurrencia.	116
Figura 46.	Velocidad media de respuesta (en milisegundos) de una solicitud. .	116
Figura 47.	Implementación de seguridad.	117
Figura 48.	Implementación de seguridad.	118
Figura 49.	Contraste de color mediante un contrast checker.	119
Figura 50.	Diseño en diferentes tamaños de pantalla.	120
Figura 51.	Métricas de solicitudes a través de Cloudflare.	121
Figura 52.	Diagrama de radar de la posición de nuestra propuesta.	124

ÍNDICE DE ANEXOS

Anexo A. Guía de entrevista a los organizadores de las ligas deportivas parroquiales del “Cantón Pedro Moncayo”	138
Anexo B. Encuesta de necesidades a los participantes de las ligas deportivas parroquiales del ”Cantón Pedro Moncayo”	139
Anexo C. Diagrama de base de datos	142
Anexo D. Documentación de los requerimientos del sistema - IEEE 830	143
Anexo E. Análisis detallado de competidores	152
Anexo F. Cálculo detallado del VAN y TIR	164

RESUMEN EJECUTIVO

La organización de campeonatos de fútbol en las ligas parroquiales del Cantón Pedro Moncayo presentaba serias dificultades debido al uso de procesos manuales. Los organizadores enfrentaban constantes retrasos en la programación de partidos, errores en los registros de jugadores y falta de transparencia en los resultados, lo que generaba descontento entre los participantes y afectaba el desarrollo del deporte local.

El objetivo principal fue implementar una aplicación web bajo el modelo de Software como Servicio para optimizar la gestión de campeonatos de fútbol en las ligas parroquiales del Cantón Pedro Moncayo con recursos económicos limitados. El sistema buscó optimizar procesos críticos como inscripciones, generación de calendarios, manejo de sanciones y publicación de resultados.

Para el desarrollo se empleó una metodología mixta. En la fase inicial se realizaron entrevistas a organizadores para identificar necesidades específicas, mientras que mediante encuestas se validaron los requerimientos con los usuarios finales. La aplicación se construyó usando la metodología RAD, con tecnologías modernas como Next.js para la interfaz, Supabase como base de datos en la nube y tRPC para la comunicación entre componentes.

El sistema logró reducir en un 98.7% en los tiempos de ejecución de tareas clave como la inscripción de equipos y jugadores, programación de partidos, registro de resultados y gestión de sanciones.

Mediante la implementación de soluciones tecnológicas adecuadas se puede revolucionar la gestión del deporte amateur. Los resultados sugieren que este modelo podría extenderse a otras disciplinas deportivas y regiones con características similares, promoviendo así el desarrollo deportivo.

Palabras clave: Gestión deportiva, aplicaciones web, automatización, ligas parroquiales, Next.js, Supabase, SaaS.

ABSTRACT

The organization of soccer championships in the parish leagues of Pedro Moncayo Canton presented serious difficulties due to the use of manual processes. The organizers faced constant delays in the scheduling of matches, errors in the registration of players and lack of transparency in the results, which generated discontent among participants and affected the development of local sports.

The main objective was to implement a web application under the Software as a Service model to optimize the management of soccer championships in the parish leagues of the Pedro Moncayo Canton with limited economic resources. The system sought to optimize critical processes such as registration, generation of calendars, management of sanctions and publication of results.

A mixed methodology was used for the development. In the initial phase, interviews were conducted with organizers to identify specific needs, while surveys were used to validate the requirements with end users. The application was built using RAD methodology, with modern technologies such as Next.js for the interface, Supabase as a cloud database and tRPC for communication between components.

The system achieved a 98.7% reduction in execution times for key tasks such as team and player registration, match scheduling, results registration and sanctions management.

By implementing appropriate technological solutions, the management of amateur sports can be revolutionized. The results suggest that this model could be extended to other sports disciplines and regions with similar characteristics, thus promoting sports development.

Keywords: Sports management, web applications, automation, parish leagues, Next.js, Supabase, SaaS.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

SOFTWARE COMO SERVICIO PARA OPTIMIZAR LA GESTIÓN DE CAMPEONATOS EN LIGAS DEPORTIVAS CON RECURSOS ECONÓMICOS LIMITADOS

1.1.1 Planteamiento del problema

El fútbol es un deporte popular en todo el mundo, el cual es practicado y amado por millones de personas. Se considera como el deporte con más deportistas profesionales, cerca de 250 millones de deportistas juegan en más de 200 países o amateurs [1]. En [2] menciona que la tecnología ayuda a mejorar la experiencia del espectáculo, ampliar el análisis del juego por parte de entrenadores, jugadores y medios de comunicación, además de permitir el seguimiento y visualización de información en tiempo real.

El problema central es la falta de una herramienta tecnológica adecuada para organizar eficientemente eventos deportivos en las ligas parroquiales del Cantón Pedro Moncayo. Estas ligas carecen de recursos económicos suficientes para implementar una infraestructura tecnológica tradicional, la cual requiere altos costos en hardware, servidores y servicios de personal, según menciona [3]. A pesar de la elevada práctica de fútbol en esta localidad, la calidad de las competencias es afectada por la ausencia de herramientas tecnológicas específicas que faciliten la planificación y ejecución de los eventos, como se indica en [4].

Según el artículo [5], en Ecuador, los equipos de fútbol profesional se encuentran agremiados en 22 asociaciones provinciales que realizan la actividad deportiva, y cuentan con 17 escenarios que conforman la infraestructura nacional para la práctica del deporte, además de contar con una comercialización basada en sistemas de gestión y marketing eficientes. En [4], se mencionan otras instituciones amateur como organizaciones comunales, barriales, ligas parroquiales y la misma liga cantonal.

En el Cantón Pedro Moncayo existe una elevada práctica de fútbol dentro de la localidad, pero de baja calidad. En [4] menciona que la falta de herramientas

tecnológicas específicas dificulta la planificación y ejecución de competencias en instalaciones menos comerciales. La ineficiente infraestructura tecnológica limita el potencial de crecimiento y desarrollo de competencias, afectando tanto a los organizadores como a los participantes [6] [7].

El problema central radica en la necesidad urgente de implementar una aplicación con un modelo de negocio que no dependa de una infraestructura tecnológica. Los recursos económicos no deben ser un limitante para la organización eficiente de eventos deportivos en las ligas deportivas parroquiales del Cantón Pedro Moncayo. Esta solución agilizará los procesos de planificación y organización de competencias, mejorará la experiencia global de participantes y espectadores, y fomentará el desarrollo y la promoción de estos deportes en el Cantón.

1.1.2 Antecedentes investigativos

En esta sección, se describe la implementación de una investigación bibliográfica con el objetivo de identificar artículos relevantes para el estudio de investigación. La búsqueda se realizó en las siguientes bases de datos científicas: IEEE Xplore, ACM Digital Library, Science Direct y Bibliotecas de Universidades.

Para la obtención de los artículos, se utilizaron cadenas de búsqueda que incluyen las palabras clave "Business Model", "SaaS", "SaaS security", "SaaS cost or pricing", "Framework SaaS", "management sport", "Cloud software" y "sport system information" que sustentan el trabajo mediante fundamentos tecnológicos en base al tema del proyecto de investigación y objetivos. Además, la búsqueda se limitó a publicaciones publicadas entre los años 2019 y 2024. A continuación se enumeran las cadenas de búsqueda:

1. C1: (((“as a service” OR “business model”) AND “software”) AND cloud)
2. C2: ((cost OR pricing OR expense) AND (“software as a service” OR saas))
3. C3: (((tool OR framework OR stack) AND saas) AND web)
4. C4: ((security OR performance) AND web)
5. C5: ((tournament OR sport) AND (management OR system) AND cloud)

Llevar a cabo una revisión bibliográfica reside en la necesidad de explorar y comprender el modelo de negocio SaaS dentro de la gestión de torneos deportivos en conjunto con las métricas de rendimiento y seguridad del software en la nube. El objetivo principal de este análisis es obtener conocimientos sobre los distintos modelos de negocio y las herramientas específicas necesarias para una implementación efectiva de un sistema de gestión deportiva, los costos de implementación, métricas de rendimiento y seguridad del software. Para ello, se realizó una búsqueda en 3 bases de datos científicas y los repositorios de las Universidades Nacionales e Internacionales, lo que resultó en un conjunto de 115 artículos científicos y tesis directamente relacionados con las palabras clave seleccionadas. A continuación, en la Tabla 1 se describen las cadenas de búsqueda implementadas para cada base de datos científica y el número de artículos o revistas científicas resultantes.

Tabla 1. Resultados de la investigación.

Base de datos	C1	C2	C3	C4	C5	Total
IEEE	7	8	7	3	16	41
ACM-DL	6	4	4	2	10	26
Science Direct	6	5	6	2	3	22
Google Scholar	8	4	4	2	8	26
Total	27	21	21	9	37	115

Dada la importancia de obtener información precisa y relevante para este proyecto, es crucial establecer criterios claros de inclusión y exclusión. Estos criterios se detallan a continuación, y se utilizaron para seleccionar los artículos pertinentes. Sirven como una guía esencial para garantizar la calidad y pertinencia de la información recopilada en este estudio.

Criterios de inclusión

- Artículos que usen sistemas de gestión de deportes.
- Artículos que usen modelos de negocio.
- Artículos que en el título presenten relación con: los costos, seguridad, herramientas y frameworks de las aplicaciones SaaS.

- Solo artículos de las siguientes bases de datos: IEEE Xplorer, ACM, Science Direct y Google Scholar.
- Trabajos que comparen el modelo de negocio SaaS con otros modelos.
- Publicaciones entre Enero 2019 y Marzo 2024.

Criterios de exclusión

- Artículos que no aborden específicamente un modelo de negocio o un sistema de gestión de deportes.
- Artículos que se enfoquen exclusivamente en modelos de negocio tradicionales o que no aborden específicamente el modelo SaaS.
- Artículos con años de publicación menores que 2019.
- Artículos que no estén en inglés o español.
- Trabajos que presenten solo la aplicación y no el modelo de negocio que usaron.

Finalmente, la evaluación de los documentos en una matriz de pertenencia definió a las publicaciones que están relacionadas al tema. La matriz permitía puntuar a un artículo en cuanto a relevancia para el estudio. Al finalizar este proceso se obtuvieron los resultados mostrados en la Tabla 2; A: APORTA, N: NO APORTA.

Tabla 2. Artículos revisados.

Bases de Datos	C1		C2		C3		C4		C5		Total		
	A	N	A	N	A	N	A	N	A	N	A	N	T
IEEE	0	0	0	0	0	0	0	0	1	0	1	0	1
ACM-DL	0	0	0	0	0	0	0	0	1	0	1	0	1
Science Direct	0	0	0	0	0	0	0	0	0	0	0	0	0
Google Scholar	4	0	3	0	2	0	1	0	7	0	17	0	17
TOTAL	4	0	3	0	2	0	1	0	9	0	19	0	19

Después de obtener un total de 131 artículos, se excluyeron 111 de ellos porque no estaban disponibles en inglés o español. Otros fueron descartados debido a que no abordaban un modelo de negocio. Además, la mayoría fue descartada porque utilizan el modelo de negocio fuera del contexto de gestión de torneos deportivos y no aportan a la investigación. Esto dejó un total de 19 artículos científicos considerados para el análisis y están detallados a continuación:

En [8] presenta un prototipo de aplicación web diseñado para abordar las necesidades de gestión de torneos deportivos de fútbol en una liga deportiva cantonal. La solución ofrecida por el autor se presenta como un modelo de negocio bajo una licencia de uso enterprise, diseñado para satisfacer las necesidades específicas de una sola liga deportiva a la vez con un pago único. El desarrollo del sistema se realiza utilizando PHP y HTML como lenguajes de programación, con una arquitectura Modelo-Vista-Controlador (MVC) y el framework Zend. Además, el autor utiliza PostgreSQL como base de datos y Zend-db para el desarrollo del sistema. Para la implementación del sistema, el autor sigue una metodología ágil de desarrollo Kanban, utilizando el Sistema Operativo CentOS 7, Apache como servidor web y Atom como entorno de desarrollo. El sistema desarrollado ofrece diversas funcionalidades esenciales, como la gestión de equipos y jugadores, la administración de reservaciones y programación de partidos, así como la generación de rankings y reportes, todo ello accesible a través de conexión a internet. El autor evalúa la eficacia del sistema mediante la satisfacción de los asistentes y dirigentes, obteniendo una satisfacción del 100%. Sin embargo, no proporciona detalles sobre los tiempos de gestión de torneos, los costos de implementación ni las prácticas para mejorar el rendimiento y la seguridad del sistema.

En [9], en el contexto de la gestión de torneos deportivos detalla el desarrollo de una aplicación web dirigida a la Liga Deportiva Parroquial de Fútbol. El autor ofrece esta solución con un modelo de negocio enterprise para la mencionada liga deportiva con un pago único. Para la implementación de la aplicación, el autor emplea tecnologías como MySQL como sistema de gestión de bases de datos y Bootstrap como framework para el diseño web. Para el desarrollo de sistema utiliza herramientas adicionales como Netbeans y Power Designer. Entre las características principales del sistema, el sistema incluye la gestión integral de campeonatos, equipos y jugadores, junto con la generación de rankings y la visualización de información relevante. Como costos de implementación, el autor despliega el sistema en un hosting y dominio de \$89 dólares. Las pruebas de aceptación llevadas a cabo reflejan un elevado nivel de satisfacción del cliente, alcanzando una tasa del 100%. No informa detalles sobre los tiempos de gestión de torneos, ni las prácticas para mejorar el rendimiento y la seguridad del sistema.

En [10] presenta un proyecto dirigido al desarrollo de un sistema web específicamente diseñado para la Liga Deportiva. El sistema se desarrolla en Python bajo un modelo de negocio enterprise para la Liga Deportiva con un pago único. Para su implementación, el autor utiliza el patrón de diseño MVC con el framework Flask, y como base de datos se emplea PostgreSQL con SQLAlchemy como ORM (Object Relation Mapping). Además, sigue una metodología de desarrollo ágil XP. Entre las funcionalidades clave, el sistema gestiona los equipos y jugadores, la programación de partidos, la visualización de información y la generación de rankings. Como evaluación del sistema, el autor realiza pruebas de funcionalidad que muestran un resultado totalmente funcional. Las pruebas de aceptación y usabilidad por parte del cliente y los usuarios resultan satisfactorio. No informa los tiempos de gestión de torneos, los costos de implementación ni las prácticas para mejorar el rendimiento y la seguridad del sistema.

En [11] destaca un proyecto centrado en el desarrollo de una aplicación móvil específicamente diseñada para una Liga Deportiva. El sistema se desarrolla bajo un modelo de negocio enterprise para la Liga Deportiva con un pago único. Como stack de desarrollo, el sistema se desarrolla en Dart, con una arquitectura MVC y utilizando el framework Nest para el desarrollo del backend. Para la persistencia de datos, el sistema usa PostgreSQL. El proceso de desarrollo sigue una metodología en cascada. Entre las funcionalidades clave, el sistema soluciona la gestión de campeonatos, equipos y jugadores, así como la gestión de sanciones y la generación de rankings. El autor mediante encuestas obtiene un 93% de acuerdo a la facilidad de uso del sistema, un 87% estar de acuerdo con la confiabilidad de la información y el excelente servicio de soporte técnico. Además, el autor evalúa el ahorro de tiempo y la fácil gestión de resultados obteniendo un 93% de satisfacción. No detalla los tiempos de gestión de torneos, costos de implementación y prácticas para mejorar el rendimiento y seguridad del sistema.

En [12] presenta una aplicación web para la gestión y notificación de horarios de fútbol diseñada para la Liga Deportiva Cantonal. El autor desarrolla el sistema bajo un modelo de negocio enterprise para la liga deportiva con un pago único. La aplicación se construyó utilizando PHP bajo la arquitectura MVC, con el framework Laravel y una base de datos MySQL utilizando ORM Eloquent. Como proceso de desarrollo, el autor siguió una metodología ágil de Scrum. La aplicación ofrece autenticación de usuarios y cumple con la norma ISO 25010 en cuanto a calidad de software. Como tiempos de

gestión, el sistema con el proceso tradicional tomaba en promedio 21.25 minutos, la gestión automatizada redujo este tiempo a solo 1.09 minutos, lo que representa un ahorro de 20.16 minutos para el administrador. Además, la implementación de la aplicación web redujo en un 94.88% el tiempo necesario para realizar los procesos de gestión de campeonatos. En términos de rendimiento, el sistema utiliza la memoria RAM un promedio de 308.66 MB. La eficiencia de desempeño del sistema alcanza un valor del 91.25%. No informa costos de implementación y prácticas para mejorar el rendimiento y seguridad del sistema.

En [13] presenta el desarrollo de un prototipo de aplicación web para la organización y gestión de recintos deportivos y sus actividades. El autor contruye la aplicación bajo un modelo de negocio enterprise, dedicada a atender a una única organización deportiva a la vez con un pago único. Está construida utilizando Java con el patrón de diseño DAO, Spring como framework, MySQL como base de datos, y se desarrolló en el entorno IntelliJ IDEA siguiendo la metodología Scrum. Las principales funcionalidades incluyen: un sistema de usuarios con diferentes roles, encontrar y añadir recintos deportivos públicos, gestionar la disponibilidad horaria y los deportes disponibles en los recintos, realizar reservas de recintos y gestionar el pago de las mismas, organizar actividades como partidos, entrenamientos y torneos, y añadir anuncios temporales sobre eventos. Para el frontend se utilizó JavaScript, React y Visual Studio Code, con Docker para la implementación y Bootstrap para el diseño. Se emplearon herramientas adicionales como Git para el control de versiones, Postman para pruebas de APIs, Clickup para la gestión de tareas y Balsamiq para el diseño de mockups. El sistema permitiría una mejor organización y gestión de actividades deportivas informales en recintos públicos, superando las limitaciones actuales. No se menciona información sobre tiempos de gestión de torneos, costos de implementación o prácticas para mejorar el rendimiento y seguridad del sistema.

En [14] presenta un prototipo de un sistema para la gestión de torneos de tenis en el Country Club de Quito. El sistema está diseñado bajo un modelo de negocio enterprise, dedicado a atender a un único club deportivo con un pago único. Está construido utilizando PHP con una arquitectura de tres capas, el framework Laravel, MySQL como base de datos y HTML. Se desarrolló siguiendo la metodología XP (Extreme Programming), con pruebas de aceptación en la aprobación del cliente. Las principales funcionalidades del sistema incluyen: gestión de jugadores, gestión de

categorías, gestión de canchas, gestión de usuarios, gestión de torneos, visualización de información, generación de rankings y préstamo de canchas. Para el frontend se utilizó JavaScript, JSON y CSS. También se incluyeron herramientas como hojas de cálculo y pizarras para el manejo de horarios. Como resultado, se obtuvo un 100% de satisfacción por parte del cliente con el sistema implementado. Sin embargo, no se menciona información sobre tiempos de gestión de torneos, costos de implementación o prácticas para mejorar el rendimiento y seguridad del sistema.

En [15] presenta el desarrollo del frontend de una aplicación web para la gestión de torneos deportivos de pádel. La aplicación está diseñada bajo un modelo de negocio enterprise con un pago único. Como frontend, el autor contruye el sistema utilizando JavaScript con el patrón de diseño MVC, el framework React, Redux para el manejo del estado de la aplicación y MySQL como base de datos. Se trata de una Single Page Application (SPA) desarrollada con Angular y empleando Bootstrap para el diseño responsivo. Además, se utilizó CSS3 en lugar de CSS tradicional para el estilo visual. Las principales funcionalidades incluyen el registro de usuarios, la gestión de torneos (creación, configuración, inscripción de participantes, etc.), la generación de rankings y la visualización de información relacionada con los torneos. Se utilizaron herramientas como Visual Studio Code para el desarrollo y GitHub para el control de versiones. El sistema permitiría una mejor gestión y personalización de los torneos de pádel, subsanando las deficiencias actuales en esta área. Sin embargo, no se menciona información sobre tiempos de gestión de torneos, costos de implementación o prácticas para mejorar el rendimiento y seguridad del sistema.

En [16] centra el diseño de una plataforma web comunitaria dedicada a la organización de quedadas para aficionados al deporte. Dirigida a personas con edades entre 13 y 90 años, la plataforma adopta un modelo de negocio basado en Software as a Service (SaaS), con la posibilidad de ofrecer una opción Freemium que incluye anuncios. Para su implementación, se emplean tecnologías como Javascript, CSS, JQuery, SwiperJS y HTML, alojadas en la plataforma Netlify. La plataforma ofrece funcionalidades de gestión de torneos y visualización de información, con una opción de suscripción para acceder a características adicionales. Los resultados de la investigación revelan una disposición variable entre los encuestados respecto a pagar por suscripciones, con un 44.8% indicando que no lo haría, un 10.3% expresando interés y un grupo restante de "tal vez". Además, se destaca que la plataforma está diseñada con un

enfoque responsive para un nivel de rendimiento alto. Sin embargo, no se menciona información sobre tiempos de gestión de torneos, costos de implementación o prácticas para mejorar el rendimiento y seguridad del sistema.

En [17] enfoca el desarrollo en una página web destinada a la organización y registro de partidos amistosos y torneos de pádel, dirigida específicamente a usuarios interesados en esta disciplina deportiva. El autor ofrece la plataforma bajo el modelo de Software as a Service (SaaS), con la posibilidad de ofrecer una suscripción para acceder a sus funcionalidades. Para la implementación, se emplean tecnologías como JavaScript, con una arquitectura MVVM (Model-View-ViewModel). Como backend, el autor utilizó Firebase como Backend as a Service (BaaS), y se utiliza React como biblioteca de JavaScript para la construcción de interfaces de usuario. Para la integración de plataformas, el autor usó herramientas como GitHub y Trello para la gestión del proyecto, y se opta por una arquitectura sin SQL (NoSQL) para la base de datos. El desarrollo sigue una metodología ágil de Scrum y se integra un sistema de autenticación para los usuarios. Entre las funcionalidades clave de la plataforma se encuentra la gestión de partidos y la visualización de información relevante para los usuarios. Aunque se menciona la opción de suscripción, no se proporciona información detallada sobre la disposición de los usuarios a pagar por el servicio, no se menciona información sobre tiempos de gestión de torneos, costos de implementación o prácticas para mejorar el rendimiento y seguridad del sistema.

En [18] presenta el diseño e implementación de una aplicación web con el propósito de gestionar las actividades que tienen lugar en las canchas de frontón, enfocándose específicamente en las necesidades relacionadas con este deporte. La aplicación se desarrolla bajo un modelo de negocio Enterprise de pago único. Para la implementación de la aplicación, se utiliza el lenguaje de programación PHP con una arquitectura cliente-servidor, el autor utilizó GitHub como plataforma de control de versiones. Como base de datos, el autor modela y gestiona mediante MySQL. Se emplean también tecnologías como Xampp para el desarrollo local y Express.js como framework de Node.js para el desarrollo del backend. La aplicación cuenta con funcionalidades clave como la gestión de reservas, visualización de información, autenticación de usuarios, gestión de torneos y un sistema de ranking. Además, se integra la API de Google Chart para la visualización de datos. No se menciona información sobre tiempos de gestión de torneos, costos de implementación o prácticas

para mejorar el rendimiento y seguridad del sistema.

En [19] se centra en el diseño y desarrollo de una aplicación móvil para la gestión de torneos de fútbol en la ciudad de Bogotá, dirigida tanto a organizadores como a jugadores o integrantes de equipos de fútbol en esta localidad. La aplicación se ofrece bajo un modelo de negocio PAID, con una tarifa del 3% por inscripción en los torneos. Para su implementación, se utiliza el lenguaje de programación Java, junto con JavaScript para el desarrollo de la interfaz de usuario. La aplicación se desarrolla en Android Studio y se integra con una base de datos MySQL. Para la implementación, el autor aloja en un servidor Linux con Glasfish o Tomcat. Se sigue una metodología ágil de desarrollo Scrum y se adopta un enfoque CMMI. Las funcionalidades principales incluyen la gestión de torneos, jugadores, árbitros, inscripciones y sanciones, así como la generación de rankings y la visualización de información relevante. Los resultados de una encuesta realizada indican que el 73.2% de los encuestados estaría dispuesto a utilizar la aplicación, con un 10.8% expresando interés "Talvez" y un 16% negándose a utilizarla. No se menciona información sobre tiempos de gestión de torneos, costos de implementación o prácticas para mejorar el rendimiento y seguridad del sistema.

En [20] desarrolla un proyecto de emprendimiento destinado a simplificar la reserva y gestión de instalaciones deportivas, con un enfoque en la mejora de la experiencia para los usuarios de estas instalaciones. El proyecto se centra en la utilización de Backend as a Service (Baas), con la posibilidad de ofrecer planes de suscripción para los usuarios. Para su implementación, el autor emplea el lenguaje de programación Java y se sigue el patrón de diseño Modelo-Vista-Controlador (MVC), utilizando SpringBoot como framework. La gestión de la base de datos se lleva a cabo utilizando PostgreSQL, y se utiliza Maven como herramienta de gestión de proyectos, además de ORM Hibernate y JPA para el mapeo objeto-relacional. El desarrollo se lleva a cabo de forma colaborativa en GitHub y se despliega en servidores Tomcat Apache alojados en AWS. Entre las funcionalidades principales de la plataforma se incluyen la gestión de reservas, la gestión de instalaciones y la visualización de información relevante para los usuarios. Además, el autor realiza una evaluación del software en relación con los Objetivos de Desarrollo Sostenible (ODS), encontrando que el 17.6% de los ODS tienen una relación alta con el software desarrollado. No se menciona información sobre tiempos de gestión de torneos, costos de implementación o prácticas para mejorar el rendimiento y seguridad del sistema.

En [21] presenta el desarrollo de un proyecto de emprendimiento destinado a simplificar la reserva y gestión de instalaciones deportivas, ofreciendo una solución basada en Software as a Service (SaaS) con opciones de planes de suscripción para los usuarios. Para su implementación, se opta por el uso del lenguaje Typescript y con una arquitectura Modelo-Vista-Controlador (MVC). Se exploran diferentes tecnologías frontend, como Angular, React, Ionic y Vue, mientras que para la gestión de la base de datos se utiliza PostgreSQL. Además, se emplean herramientas como HTML, CSS, GitHub, Bootstrap, Firebase y Capacitor para el desarrollo y despliegue del software. Entre las funcionalidades clave de la plataforma se encuentran la gestión de reservas y de instalaciones, así como la visualización de información relevante para los usuarios. El autor también lleva a cabo una evaluación del software en relación con los Objetivos de Desarrollo Sostenible (ODS), encontrando que el 5.88% de los ODS tienen una relación alta con el software desarrollado. No se menciona información sobre tiempos de gestión de torneos, costos de implementación o prácticas para mejorar el rendimiento y seguridad del sistema.

En [22] desarrolla una aplicación móvil para apoyar las actividades comunitarias en el campo del fútbol y las instalaciones deportivas de fútbol sala en Yakarta. El autor diseña bajo un modelo de negocio de pago único e implementó una plataforma de pago electrónico. El sistema fue construido utilizando Android Studio y MySQL como base de datos. Entre sus características principales se encuentran la autenticación de usuarios, la gestión de pagos, la reserva de campos deportivos, foros de discusión y desafíos de partidos de fútbol. Dado que la aplicación no obtuvo resultados en base al estudio, no podemos dar credibilidad a calidad del software. Sin embargo, no se proporcionaron detalles sobre los costos de implementación ni las prácticas específicas para mejorar el rendimiento y la seguridad del sistema.

En [23] construye un sistema de gestión de información integral y funcional para el deporte universitario, utilizando el Internet de las Cosas (IoT) como base tecnológica. El sistema estuvo orientado a la comunidad universitaria y se diseñó bajo un modelo de negocio enterprise con un pago único. El autor desarrolló utilizando JavaScript como lenguaje de programación. Como arquitectura, el autor utilizó J2EE, con un CPU i5, 4G de memoria RAM y MySQL como base de datos principal. Además, incorporó tecnologías como lectores de RFID, Tomcat, sensores de IoT, lectores de códigos QR, Big Data, Oracle y tecnología GPS para la recolección de datos. Entre

sus funcionalidades se encontraban la gestión de actividades deportivas, la gestión de competencias deportivas, un sistema de gestión de pruebas de salud física para estudiantes, y se utilizaron HTML y CSS para la interfaz de usuario. Sin embargo, no se mencionaron los costos de implementación ni las prácticas específicas para mejorar el rendimiento y la seguridad.

En [24] desarrolla un sistema web para la gestión de procesos del campeonato de fútbol de la Liga Cantonal de Penipe. El sistema se diseñó bajo un modelo de negocio enterprise con un pago único. Como stack, el autor contruye utilizando Python, el framework MTV Django y PostgreSQL como base de datos. Para el diseño, el autor aborda Bootstrap para la interfaz de usuario y se siguió una metodología ágil Scrum durante el proceso de desarrollo. Entre sus funcionalidades principales se encontraban la gestión de inscripciones, la gestión de partidos y la generación de reportes. La evaluación del sistema web se realizó en base a la norma ISO/IEC 25010, mediante una encuesta dirigida a los directivos de la Liga Cantonal de Penipe. Esta evaluación determinó que el aplicativo cumplía en un 100% con los términos de completitud, 85% en corrección y 93% en pertinencia funcional. Consecuentemente, con un promedio general del 94%, se concluyó que el sistema web era funcional. Se aplicaron métricas de calidad acordes a las normas ISO/IEC 25010. No se mencionaron los costos de implementación ni prácticas específicas para mejorar el rendimiento y seguridad del sistema.

En [25] implementa una Aplicación Web Progresiva (PWA) para la automatización de los procesos de gestión e información de la Liga Deportiva Parroquial Totoras de fútbol. La aplicación se diseñó bajo un modelo de negocio enterprise con un pago único. Como stack de desarrollo, el autor utiliza Spring como framework en conjunto con JavaScript, bajo el patrón MVC con el framework VueJS, PostgreSQL como base de datos y NodeJS en el backend. El autor sigue una metodología ágil XP durante el proceso de desarrollo. Entre sus funcionalidades principales se encontraban la gestión de torneos y la gestión de jugadores. Se utilizaron HTML y CSS para la interfaz de usuario. La factibilidad del desarrollo de la PWA fue evaluada mediante encuestas, donde el 96% de las personas expresaron su deseo de acceder a la información de los clubes, equipos y jugadores de la Liga Deportiva Parroquial Totoras a través de un aplicativo. No se mencionaron los costos de implementación ni prácticas específicas para mejorar el rendimiento y seguridad del sistema.

En [26] desarrolla un sistema web para la Gestión del Campeonato Inter-Facultades de la Escuela Politécnica Nacional. El sistema se diseñó bajo un modelo de negocio enterprise con un pago único. Para la construcción del sistema, el autor utiliza JavaScript, con el patrón MVC y el framework Laravel, empleando MySQL como base de datos. El autor utiliza HTML, CSS y Bootstrap para la interfaz de usuario, y se trabajó con el entorno Xampp. Como metodología ágil, el autor siguió una metodología ágil Scrum. Entre sus funcionalidades principales se encontraban la gestión de jugadores, la gestión de campeonatos, la autenticación de usuarios y la visualización de información. La utilidad del sistema fue evaluada, donde el 70% de los encuestados respondió que sería muy útil, ya que eliminaría la documentación física requerida durante las inscripciones y facilitaría el conocimiento de fechas y horas de los encuentros para los jugadores. El 20% lo consideró útil, dado que actualmente no existía un sistema automatizado, y el 10% lo calificó como poco útil, debido a la costumbre de los organizadores de realizar el proceso manualmente. No se mencionaron los costos de implementación.

En conclusión, los diversos trabajos investigativos convergen en la importancia crucial de implementar un modelo de negocio para optimizar la gestión de torneos. Entre los modelos de negocio mencionados con mayor medida se muestran en la Tabla 3:

Tabla 3. Comparación de modelos de negocio de diferentes artículos.

Referencia de los artículos	Descripción de la cita	Modelo de negocio	Razón de modelo seleccionado
[8], [9], [10], [11], [12], [13], [14], [15], [18]	Desarrollo de una aplicación web de gestión de torneos	Enterprise	Vender un producto completo, dar soluciones a medida y personalizadas para atender las necesidades específicas.

Referencia de los artículos	Descripción de la cita	Modelo de negocio	Razón de modelo seleccionado
[19]	Diseño de una aplicación móvil para la gestión de torneos de fútbol	Pago por uso	Utilizar la plataforma sin incurrir en costos fijos o compromisos financieros a largo plazo. Tiene el potencial de escalar de manera efectiva.
[16], [17], [21]	Desarrollo de una aplicación web e híbrida para la gestión de torneos y resevas.	Software as a Service (SaaS)	Propone una solución accesible, escalable y basada en la nube para reducir los costos de implementación y mantenimiento.
[20]	Desarrollo de una aplicación híbrida para la reserva y gestión de instalaciones deportivas	Backend as a Service (BaaS)	Aprovechamiento de infraestructura escalable, servicios integrados y simplificación en la gestión del backend.

En base a los artículos revisados en la Tabla 3 sobre los modelos de negocio en software, se observa que el modelo SaaS es extensamente adoptado. Dado que, este modelo ofrece a los clientes la propiedad total del software y la posibilidad de personalizarlo según sus necesidades específicas.

Las metodologías ágiles se posiciona como un pilar fundamental para el cumplimiento de los requisitos funcionales y no funcionales. En consecuencia, la adopción de metodologías ágiles contribuye a la excelencia de la aplicación SaaS. Entre las metodologías ágiles mencionadas con mayor medida se muestran en la Tabla 4:

Tabla 4. Comparación de metodologías ágiles de diferentes artículos.

Referencia de los artículos	Metodología ágil utilizada	Razón de modelo seleccionado
[8]	Kanban	Se enfoca en el flujo de trabajo continuo y la visualización del progreso.
[10], [14]	XP	Se enfoca en la entrega rápida de software de alta calidad, con énfasis en la comunicación, simplicidad y feedback constante.
[24], [26]	Scrum	Organiza el trabajo en ciclos cortos llamados sprints, con revisiones regulares y retrospectivas para ajustes iterativos.

Por último, la adopción de sistemas tecnológicos contribuye a la excelencia en la gestión dentro de este sector tan crítico. Entre las herramientas y tecnologías mencionadas con mayor medida se muestran en la Tabla 5:

Tabla 5. Comparación de herramientas y tecnologías de diferentes artículos.

Referencia de los artículos	Herramientas y tecnologías utilizadas	Propósito principal
[8], [12]	PHP, HTML, Zend Framework, PostgreSQL	Desarrollo de aplicaciones web robustas.
[9]	MySQL, Bootstrap	Gestión de bases de datos y diseño responsivo de interfaces de usuario.
[24]	Python, Django, PostgreSQL	Desarrollo rápido de aplicaciones web con un enfoque en la productividad y la flexibilidad.
[20]	Spring Boot, MySQL, Maven, Hibernate, JPA	Desarrollo de aplicaciones empresariales escalables.
[21]	Typescript, Angular, React, Ionic, Vue, HTML, CSS, GitHub, Bootstrap, Firebase, Capacitor	Desarrollo de aplicaciones web progresivas (PWAs) con un enfoque en la interoperabilidad con dispositivos móviles.

Referencia de los artículos	Herramientas y tecnologías utilizadas	Propósito principal
[23]	JavaScript, J2EE, i5 CPU, 4GB RAM, MySQL, RFID readers, Tomcat, IoT sensors, QR code readers, Big Data, Oracle, GPS technology	Integración de tecnologías IoT y big data para la gestión de información.
[26]	Python, Laravel, PostgreSQL	Desarrollo de aplicaciones web con un enfoque en la gestión de contenido.
[25]	Spring, JavaScript, VueJS, PostgreSQL, NodeJS	Desarrollo de aplicaciones web progresivas (PWAs) con un enfoque en la eficiencia y la escalabilidad.

1.2 Fundamentación teórica

1.2.1 Modelos de negocio

Un modelo de negocio es la estrategia que utiliza una empresa para monetizar su producto o servicio. Básicamente, es el esquema que determina cómo se generarán ingresos, ya sea mediante ventas únicas, suscripciones mensuales. En el contexto de las aplicaciones web deportivas, hay múltiples opciones para obtener ganancias, como comercializar la licencia del software, ofrecer pagos por uso o implementar otros sistemas de cobro. En [21] define los modelos de negocio como estrategias para la comercialización del producto. A continuación, se presentan algunos modelos de negocio identificados en los antecedentes investigativos:

a. Modelo de negocio Enterprise

Este modelo es similar a comprar un software en una tienda, pagas una vez y el producto es tuyo para siempre. La empresa que necesita el sistema hace un pago único y obtiene el software completo. En [8–15, 18], el producto se ofrece con una licencia

de uso única a una entidad específica, como una liga deportiva o una instalación deportiva. En este enfoque, el autor desarrolla la aplicación y la vende como un producto completo, con un pago único por la licencia de uso. Este modelo ofrece a los clientes la propiedad total del software y la posibilidad de personalizarlo según sus necesidades específicas.

b. Modelo de negocio SaaS

SaaS significa "Software como Servicio" y funciona como una suscripción mensual o anual. En lugar de comprar el software completo, los usuarios hacen un pago recurrente para acceder al sistema, sin necesidad de instalarlo en sus computadoras. En [16, 17, 21], los autores ofrecen la aplicación web como un servicio basado en la suscripción. El autor ofrece a los usuarios la paga de una tarifa periódica por el acceso a la aplicación y sus funcionalidades. Se proporciona una solución accesible, escalable y basada en la nube para reducir los costos de implementación y mantenimiento, aprovechando una infraestructura flexible. Este modelo proporciona a los clientes la flexibilidad de pagar solo por el uso que hacen de la aplicación, sin necesidad de realizar una inversión inicial significativa en licencias de software.

c. Modelo de negocio de pago por inscripción en torneos

Este modelo funciona como una comisión, la plataforma cobra una pequeña tarifa cada vez que alguien se inscribe en un torneo a través del sistema. En [19], el autor menciona un modelo de negocio donde la aplicación cobra una tarifa por cada inscripción en un torneo gestionado a través de la plataforma. Se propone utilizar la plataforma sin incurrir en costos fijos o compromisos financieros a largo plazo. Este modelo de negocio tiene el potencial de escalar de manera efectiva a medida que la plataforma gana popularidad, lo que puede generar ingresos adicionales para el desarrollador de la aplicación.

d. Modelo de negocio BaaS

BaaS significa "Backend como Servicio" y es una forma de desarrollar aplicaciones donde el programador no tiene que preocuparse por crear y mantener los servidores. Por ejemplo, en [20] se utiliza el modelo Backend as a Service (BaaS), aprovechando su infraestructura escalable, servidores integrados y despliegue de la plataforma web

sin preocuparse por la gestión del backend, esperando obtener ventajas como la optimización de costos y generación de ingresos sostenibles a largo plazo.

1.2.2 Gestión de campeonatos deportivos

La gestión de campeonatos deportivos se refiere a todas las actividades necesarias para organizar y llevar a cabo competencias deportivas de manera exitosa. Según [12], la gestión de campeonatos deportivos se ha convertido en un área de interés creciente, impulsada por la necesidad de mejorar la organización y eficacia de las ligas y torneos deportivos. A continuación, se destaca la importancia de contar con sistemas integrales que optimicen la planificación, ejecución y seguimiento de estos eventos.

a. Automatización de procesos de gestión

La automatización tiene como objetivo realizar tareas repetitivas sin intervención humana constante. Los autores han propuesto sistemas y aplicaciones web que buscan automatizar los procesos de gestión de campeonatos deportivos. En [12, 13, 25] permiten la programación de partidos, la gestión de equipos y jugadores, la generación de rankings y la visualización de información relevante de manera eficiente y centralizada.

b. Integración de funcionalidades clave

La integración se refiere a tener todas las herramientas necesarias en un solo lugar. En [24, 26], los autores proponen funcionalidades esenciales para la gestión de campeonatos deportivos, como la gestión de inscripciones, programación de partidos, generación de calendarios, administración de equipos y jugadores, entre otros.

c. Visualización de información relevante

Para [12, 24, 26], una característica que no debería ser descartada en un sistema de gestión de torneos es la visualización de información relevante como horarios de partidos, resultados, posiciones en la tabla, estadísticas, entre otros.

d. Seguridad y confiabilidad

La seguridad se refiere a proteger la información del sistema para que no sea modificada o accedida por personas no autorizadas. La confiabilidad significa que el sistema funciona correctamente cuando se necesita, sin fallos ni interrupciones. En [8, 14, 20], los autores han considerado aspectos de seguridad y confiabilidad en el desarrollo de los sistemas, implementando medidas para proteger la integridad de los datos y garantizar la disponibilidad y accesibilidad de la plataforma.

1.2.3 Tecnologías web modernas

Las tecnologías web modernas son las herramientas, lenguajes de programación y métodos que los desarrolladores utilizan para crear aplicaciones que funcionan a través de internet. Estas tecnologías han evolucionado significativamente en los últimos años, permitiendo crear aplicaciones más rápidas, seguras y fáciles de usar. En [21], se subraya la importancia de las tecnologías al referirse a ellas como los cimientos sobre los cuales se construyen las aplicaciones web modernas. El autor no solo presenta un análisis detallado de las tecnologías utilizadas, sino que también destaca su influencia en la estrategia y el desempeño general de los productos en el mercado. A continuación, se describen algunas de las tecnologías web utilizadas:

a. PHP

PHP es un lenguaje de programación especialmente diseñado para crear sitios web dinámicos. Varios autores, como se observa en las tesis [8] y [9], han empleado PHP como lenguajes de programación fundamentales para el desarrollo de aplicaciones web. Los autores mencionan que les proporciona la base para la creación de contenido dinámico y la interacción con bases de datos.

b. Python y Flask

Python es un lenguaje de programación conocido por ser fácil de aprender y usar, mientras que Flask es una herramienta que facilita la creación de aplicaciones web con Python. En la tesis [10], el autor opta por utilizar Python en conjunto con el framework Flask para el desarrollo de la aplicación web. El autor aborda Flask por su simplicidad

y flexibilidad en la construcción de aplicaciones web, lo que puede haber permitido una rápida implementación del sistema.

c. JavaScript y sus frameworks

JavaScript es el lenguaje que hace que las páginas web sean interactivas. Varias tesis mencionan el uso de JavaScript y sus frameworks para la construcción de interfaces de usuario dinámicas. Por otro lado, TypeScript es una versión mejorada de JavaScript que añade tipos al código, lo que significa que especifica qué tipo de información puede contener cada variable. Por ejemplo, en la tesis [25], se menciona el uso de Vue.js, mientras que en la tesis [17] se emplea React. Los autores mencionan que eligieron estos frameworks por su capacidad para crear interfaces interactivas y rápidas.

Con el concepto de framework se menciona otro concepto importante para el desarrollo web, se menciona metaframework como una herramienta que se construye encima de otros frameworks para hacerlos aún más fáciles de usar. Es como tener un kit de herramientas pre-armado que incluye todo lo necesario para construir una aplicación. En [17] se define Next.js como un metaframework basado en React que facilita la creación de aplicaciones web rápidas y optimizadas, mientras que Astro se enfoca en crear sitios web extremadamente rápidos al cargar solo el código JavaScript necesario.

d. Framework CSS

Los frameworks de CSS son herramientas que proporcionan estilos y componentes prediseñados para crear interfaces web atractivas sin tener que escribir todo el código de diseño desde cero. Es como tener un catálogo de muebles y decoraciones listas para usar en lugar de fabricar cada elemento manualmente. Existen dos enfoques principales que han dominado el desarrollo web moderno, cada uno con filosofías completamente diferentes. Bootstrap, utilizado en [15, 21, 26], representa el enfoque tradicional de los frameworks CSS. Los autores han aprovechado su sistema de componentes preconstruidos, donde elementos como botones, formularios y navegadores vienen con estilos predefinidos. Sin embargo, la evolución del desarrollo web ha dado lugar a enfoques más modernos y flexibles. Tailwind CSS representa una filosofía completamente diferente, en lugar de ofrecer componentes prediseñados, proporciona clases utilitarias de bajo nivel que permiten construir diseños personalizados directamente en el HTML.

e. Base de datos relacional

Una base de datos es como un archivo digital muy organizado donde se guarda toda la información de la aplicación. PostgreSQL y MySQL son tipos específicos de bases de datos que organizan la información en tablas. La mayoría de los autores optan por utilizar bases de datos relacionales para almacenar y gestionar la información de sus aplicaciones. Se observa el uso de sistemas como PostgreSQL y MySQL, como se menciona en varias tesis, incluyendo [10] y [12].

Una plataforma que se especializa en esto es Supabase una plataforma moderna de Backend as a Service que proporciona una alternativa de código abierto a servicios tradicionales. Esto permite a los desarrolladores enfocarse en crear la interfaz de usuario sin preocuparse por configurar y mantener servidores complejos.

f. Metodologías ágiles

Las metodologías ágiles son formas de organizar el trabajo de desarrollo de software que se enfocan en entregar resultados rápidamente y adaptarse a los cambios. Las metodologías de desarrollo juegan un papel crucial en la planificación, ejecución y gestión de proyectos de desarrollo de software. Estas metodologías se distinguen por su enfoque en una planificación minuciosa, buscando que cada proceso sea definido y predecible. [11]

Algunos autores mencionan el uso de metodologías ágiles para el desarrollo de sus aplicaciones. Por ejemplo, en el estudio [12, 17], se describe cómo el equipo de desarrollo utilizó la metodología Scrum para entregar por partes un producto funcional y en periodos de tiempos cortos, mientras que en la tesis [10, 14], se destaca el uso de la metodología XP (Programación Extrema) para proporcionar retroalimentación continua entre el cliente y el desarrollador. Por otro lado, la metodología RAD (Rapid Application Development) es una metodología de desarrollo que prioriza la creación rápida de prototipos y la entrega temprana de funcionalidades. Esta metodología es especialmente útil cuando se necesita desarrollar aplicaciones en tiempos cortos y con alta participación del usuario final.

1.3 Objetivo general

Implementar una aplicación web para la gestión de campeonatos de fútbol utilizando el modelo de negocio SaaS para agilizar el tiempo de procesos en las ligas deportivas parroquiales del Cantón Pedro Moncayo en el año 2025.

1.4 Objetivos específicos

- Investigar las herramientas de aplicaciones SaaS para la gestión de campeonatos de fútbol en las ligas deportivas parroquiales del Cantón Pedro Moncayo.
- Analizar los procesos para agilizar el tiempo de gestión de un campeonato en las ligas deportivas parroquiales de fútbol del Cantón Pedro Moncayo.
- Desarrollar una aplicación web que permita agilizar los procesos de gestión de los campeonatos de fútbol junto con una evaluación técnica y económica utilizando el modelo de negocio SaaS para las ligas deportivas parroquiales del Cantón Pedro Moncayo.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

Para la recolección de información se utilizarán encuestas y entrevistas mediante el uso de cuestionarios. Las entrevistas serán aplicadas a los organizadores de las ligas deportivas parroquiales del “Cantón Pedro Moncayo” para determinar los requerimientos (Véase Anexo A) y las encuestas serán aplicadas a todos los participantes de las ligas deportivas parroquiales del “Cantón Pedro Moncayo” para determinar necesidades (Véase Anexo B).

2.2 Métodos

2.2.1 Modalidad de la investigación

La presente investigación se enmarca bajo una modalidad cuantitativa. Por otro lado, la aplicación de técnicas de validación como el test-retest para los instrumentos de recolección de datos, verifica la consistencia de los mismos y evita sesgos en la investigación. En el desarrollo de este trabajo, se han seleccionado diversas modalidades de investigación que permitirán abordar de manera exhaustiva el tema propuesto. Estas modalidades incluyen:

- **Modalidad de campo:** Esta modalidad se centrará en recopilar datos directamente de las fuentes primarias, tales como entrevistas con miembros de las ligas deportivas parroquiales del Cantón Pedro Moncayo. La investigación de campo permitirá obtener una comprensión profunda de los desafíos y necesidades actuales en la gestión de campeonatos de fútbol en el contexto local.
- **Modalidad bibliográfica documental:** Se realizará una revisión exhaustiva de literatura existente sobre las herramientas utilizadas en modelos de negocio SaaS, gestión de campeonatos deportivos, y casos similares de implementación de tecnología en contextos con recursos limitados. Esta modalidad permitirá establecer un análisis y comparar herramientas y tecnologías para el desarrollo de la aplicación SaaS propuesta.

- **Modalidad aplicada:** Con el fin de validar la viabilidad y utilidad de la aplicación SaaS propuesta, se llevará a cabo pruebas piloto con ligas deportivas parroquiales seleccionadas. Estas pruebas aplicadas servirán para identificar áreas de mejora y ajustes necesarios antes de la implementación a gran escala.

La combinación de estas modalidades de investigación permitirá abordar el problema desde múltiples perspectivas, ofreciendo una solución integral y sustentable para la gestión de campeonatos de fútbol en las ligas parroquiales del Cantón Pedro Moncayo. En la siguiente sección se detalla la modalidad de la investigación que se aplicará al desarrollo de este proyecto.

2.2.2 Población y muestra

La población de estudio comprende las ligas deportivas parroquiales del Cantón Pedro Moncayo, considerando tanto a los organizadores como a los usuarios finales de la aplicación. Para las entrevistas, se trabajó con la totalidad de los encargados de las tres ligas activas, dado el reducido número de sujetos involucrados. Por otro lado, para las encuestas al desconocerse el tamaño exacto de la población de usuarios potenciales como aficionados, participantes y organizadores, se determinó una muestra representativa mediante la fórmula para poblaciones desconocidas.

a. Muestra para realizar la entrevista

Para el desarrollo de la aplicación, se ha decidido trabajar con la totalidad de la población, es decir, con los 3 organizadores de las 3 ligas deportivas parroquiales. Esto se debe a que el tamaño de la población es relativamente pequeño y representa un universo manejable para el propósito de la investigación.

b. Muestra para necesidades de la aplicación

La muestra para determinar las necesidades de la aplicación web estará compuesta por todos los usuarios finales, entre los cuales, están las personas seguidoras de la liga, organizadores de los campeonatos y los participantes activos en las ligas deportivas parroquiales del Cantón Pedro Moncayo.

Para determinar el tamaño de la muestra representativa necesaria para la aplicación de la encuesta, se consideraron diversos factores. Se estableció un nivel de confianza del 95%, dada la importancia de comprender a fondo las necesidades y requisitos de los usuarios de la aplicación. Además, se fijó un margen de error del 10%, permitiendo así una generalización precisa de los resultados obtenidos. Dado que no existen estudios previos en este contexto específico, se asignó una probabilidad de éxito del 50%.

La aplicación de la fórmula correspondiente arrojó un valor que indica la necesidad de una muestra representativa significativa. En este sentido, se reconoce la importancia de incluir a todos los organizadores y participantes de las ligas deportivas parroquiales en la muestra de testeo, con el fin de capturar la diversidad de opiniones y experiencias relacionadas con la gestión de campeonatos de fútbol en el Cantón Pedro Moncayo., entonces se aplica la siguiente formula:

$$n = \frac{Z^2 \cdot p \cdot q}{e^2} \quad (1)$$

donde:

n es el tamaño de muestra,

Z es el valor crítico de la distribución normal estándar,

p es la proporción esperada de la población,

$q = 1 - p$ es la proporción complementaria a p ,

e es el margen de error permitido.

$$n = \frac{1.96^2 \cdot 0.5 \cdot (1 - 0.5)}{0.1^2} \quad (2)$$

$$n = 96.04 \quad (3)$$

Por lo tanto, se necesita una muestra de al menos 96 personas para la encuesta. Dado que la muestra objetivo es relativamente pequeña, se recomienda distribuir esta muestra de manera equitativa entre los participantes de las ligas, o ajustarla según sea necesario para garantizar una representación adecuada de cada grupo.

2.2.3 Recolección de información

El principal objetivo de la recolección de información es recabar datos valiosos que permitan desarrollar una aplicación web de gestión de campeonatos de fútbol para las ligas deportivas parroquiales del Cantón Pedro Moncayo. Este proceso es esencial para entender las necesidades y expectativas de los usuarios finales, así como para validar la funcionalidad y usabilidad del software propuesto.

En la Tabla 6 detalla el proceso de recolección de información que se llevará a cabo para el desarrollo de una aplicación de gestión de campeonatos de fútbol de las Ligas Deportivas Parroquiales del Cantón Pedro Moncayo.

Tabla 6. Proceso de recolección de la información.

Preguntas básicas	Explicación
¿Para qué?	Para realizar el levantamiento de requerimientos funcionales y no funcionales, y la validación de la aplicación de gestión de campeonatos de fútbol.
¿De qué personas u objetos?	Organizadores y participantes de las Ligas Deportivas Parroquiales del Cantón Pedro Moncayo.
¿Sobre qué aspectos?	Procesos dentro de la gestión de campeonatos de fútbol
¿Quién, quiénes?	Investigador: Jhon Alexander Lescano Cisneros
¿Cuándo?	Marzo 2025 – Julio 2025
¿Dónde?	Ligas Deportivas Parroquiales del Cantón Pedro Moncayo.
¿Cuántas veces?	Una vez.
¿Qué técnicas de recolección?	Entrevista técnica, encuesta
¿Con qué?	Preguntas propuestas por el investigador.

- **Entrevistas (Véase Anexo A):** Se realizarán entrevistas con los organizadores de los campeonatos de fútbol. Estas entrevistas permitirán obtener información detallada sobre los desafíos actuales en la gestión de los campeonatos, las funcionalidades necesarias en la aplicación, y las preferencias de uso. La

consistencia y estabilidad de la información recopilada durante las entrevistas se verificará con el método test-retest en un intervalo de 15 días.

- **Encuestas (Véase Anexo B):** Se diseñará y aplicará una encuesta a un conjunto más amplio para los usuarios finales entre organizadores y participantes. La encuesta servirá para confirmar las características y funcionalidades clave que deben incluirse en la aplicación.

a. Resultados de la encuesta de necesidades aplicadas a los participantes de las Ligas Deportivas Parroquiales del “Cantón Pedro Moncayo”

En este apartado se exponen los resultados de la encuesta de necesidades, con el propósito de confirmar las características y funcionalidades clave que deben incluirse en la aplicación. Además de descubrir nuevas funcionalidades, la encuesta busca identificar los principales desafíos en el proceso de gestión, específicamente en las etapas de inscripción, programación, estadísticas y visualización del campeonato.

Pregunta 1: ¿Cuál es su rol en los torneos?

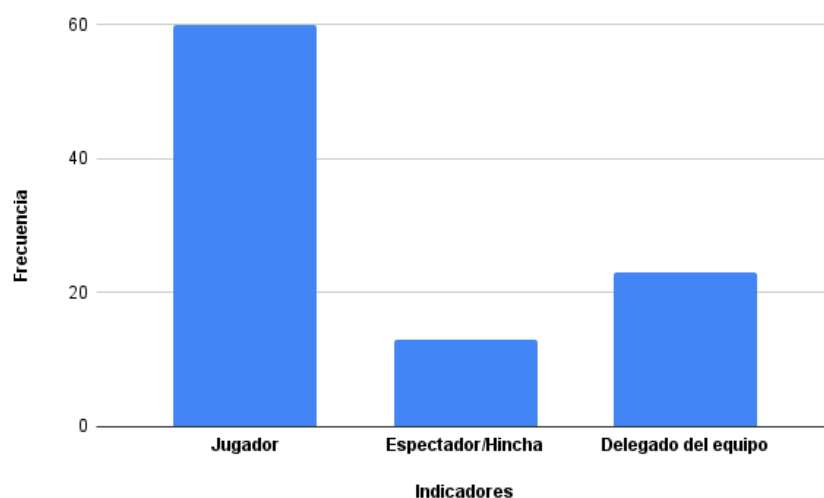


Figura 1. Distribución de roles entre los participantes.

Tabla 7. Resultados de la pregunta 1.

Indicador	Frecuencia	Porcentaje
Jugador	60	62.5%
Delegado del equipo	23	24%
Espectador/Hincha	13	13.5%
Suma total	96	100%

Análisis e interpretación:

Los resultados evidencian que la aplicación web debe priorizar las funcionalidades orientadas a jugadores, dado que representan el 62.5% de los usuarios potenciales. Sin embargo, no debe subestimarse el peso de los delegados (24%), ya que su rol implica gestiones críticas como inscripciones, comunicación con organizadores y manejo de datos del equipo, lo que demanda herramientas administrativas robustas. La escasa presencia de espectadores (13.5%) sugiere que, aunque su integración no sea prioritaria, podrían explorarse features como resultados en tiempo real o interacción social para fidelizarlos, pues su participación puede incrementar el engagement del torneo. Estos hallazgos refuerzan la **necesidad de un diseño modular que atienda las capas operativas (jugadores) y estratégicas (delegados)**.

Pregunta 2: ¿Con qué frecuencia participa en torneos deportivos organizados por su parroquia?

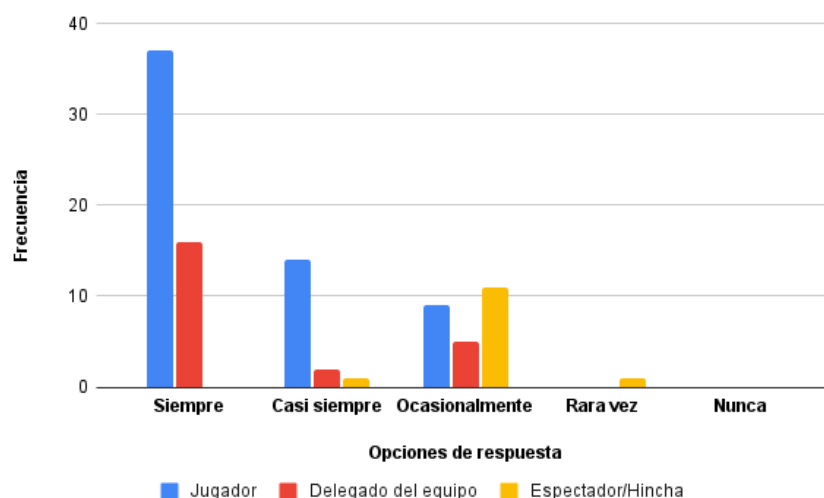


Figura 2. Resultados de la encuesta en la pregunta 2.

Tabla 8. Resultados de la pregunta 2.

Indicador	Frecuencia	Porcentaje
Siempre	53	55.2%
Casi siempre	17	17.7%
Ocasionalmente	25	26%
Rara vez	1	1%
Nunca	0	0%
Suma total	96	100%

Análisis e interpretación:

Los resultados revelan un alto nivel de compromiso por parte de la comunidad, donde **más del 72% de los encuestados participa activamente ("siempre" o "casi siempre") en los torneos parroquiales**. Esto indica una base de usuarios recurrentes que demandan soluciones eficientes para gestionar su participación, como inscripciones rápidas, acceso a calendarios de partidos o comunicación centralizada. La ausencia de respuestas en "nunca" y el mínimo 1% en "rara vez" sugieren que estos torneos tienen una alta retención y fidelización, lo que refuerza la viabilidad de implementar una plataforma digital. Sin embargo, el 26% que participa "ocasionalmente" no debe pasarse por alto, ya que podría reflejar a nuevos integrantes o personas con menor disponibilidad. El enfoque de la plataforma debe estar en **garantizar que su interfaz sea intuitiva y que los procesos clave (como registrarse, consultar horarios o ver resultados) sean lo más sencillos posibles** para retener a este segmento.

En conclusión, estos datos respaldan la necesidad de una herramienta digital que optimice la experiencia tanto para los usuarios frecuentes (automatizando procesos repetitivos) como para los ocasionales (simplificando su interacción), asegurando así la sostenibilidad y crecimiento de los torneos.

Pregunta 3: ¿Cómo calificaría el proceso actual de inscripción en los torneos deportivos?

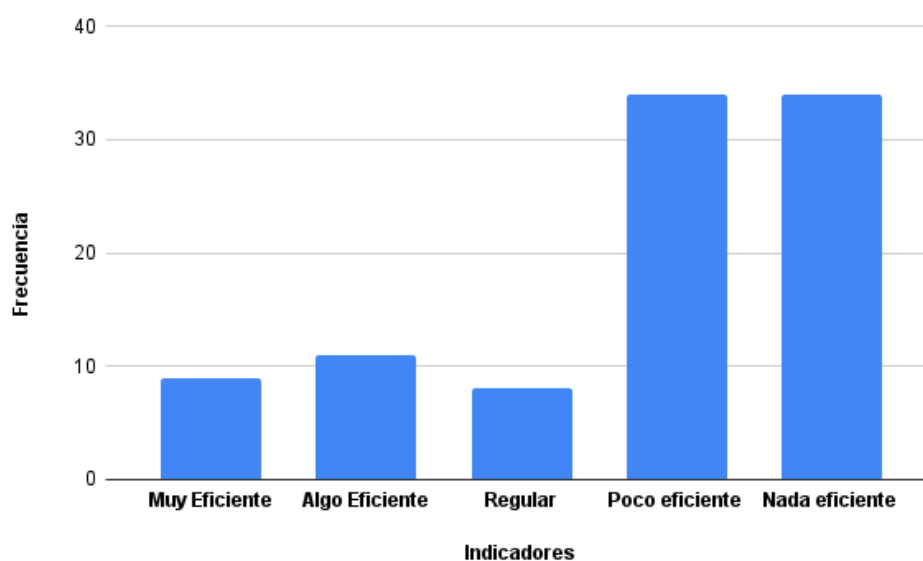


Figura 3. Resultados de la encuesta en la pregunta 2.

Tabla 9. Resultados de la pregunta 3.

Indicador	Frecuencia	Porcentaje
Muy eficiente	9	9.4%
Algo eficiente	11	11.5%
Regular	8	8.3%
Poco eficiente	34	35.4%
Nada eficiente	34	35.4%
Suma total	96	100%

Análisis e interpretación:

Los resultados dejan en evidencia una problemática crítica: **el 70.8% de los participantes percibe el proceso de inscripción actual como ineficiente ("poco" o "nada eficiente")**, lo que señala una evidente brecha entre las necesidades de los usuarios y los métodos tradicionales empleados. Este alto nivel de insatisfacción —donde apenas el 20.9% valora positivamente el sistema— sugiere que los procesos manuales, la desorganización o la falta de transparencia están generando frustración entre jugadores y delegados. Además, el hecho de que las respuestas negativas ("poco" y "nada eficiente") representen porcentajes idénticos (35.4% cada una) refuerza la urgencia de un rediseño profundo, no solo ajustes menores. La aplicación podría **incorporar funciones como inscripciones o soporte en línea** para transformar esta debilidad actual en una ventaja competitiva que fidelice a los usuarios.

Pregunta 4: ¿Qué dificultades ha enfrentado al inscribirse o participar en torneos?

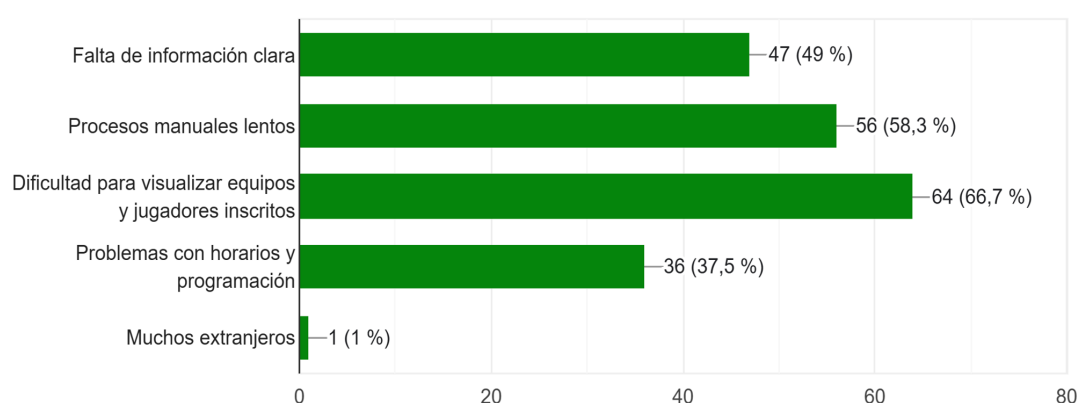


Figura 4. Dificultades reportadas por los participantes.

Análisis e interpretación:

Los resultados evidencian deficiencias estructurales en la gestión actual de torneos, donde la falta de transparencia y los métodos obsoletos generan fricciones

significativas. **El problema más crítico es la dificultad para visualizar equipos y jugadores inscritos (66.7%) lo que refleja una carencia de acceso a datos básicos**, lo que afecta la planificación tanto de participantes como de organizadores. Esto se agrava con la falta de información clara (47%) y los procesos manuales lentos (56%), combinación que explica la alta insatisfacción detectada en la pregunta 3. La aplicación web debe priorizar un módulo de visualización intuitiva (tablas públicas, filtros por categorías) y automatizar las inscripciones para eliminar cuellos de botella. Los problemas con horarios (36%), aunque menos frecuentes, son igualmente relevantes, ya que impactan directamente en la experiencia; aquí, herramientas de programación automática serían clave. El marginal 1% que menciona "muchos extranjeros" podría indicar aspectos internos de los campeonatos. En síntesis, estos hallazgos validan el **diseño de una solución integral que aborde desde la digitalización de procesos hasta la transparencia informativa**, transformando obstáculos actuales en oportunidades de mejora.

Pregunta 5: ¿Cómo suele enterarse de los torneos y sus requisitos?

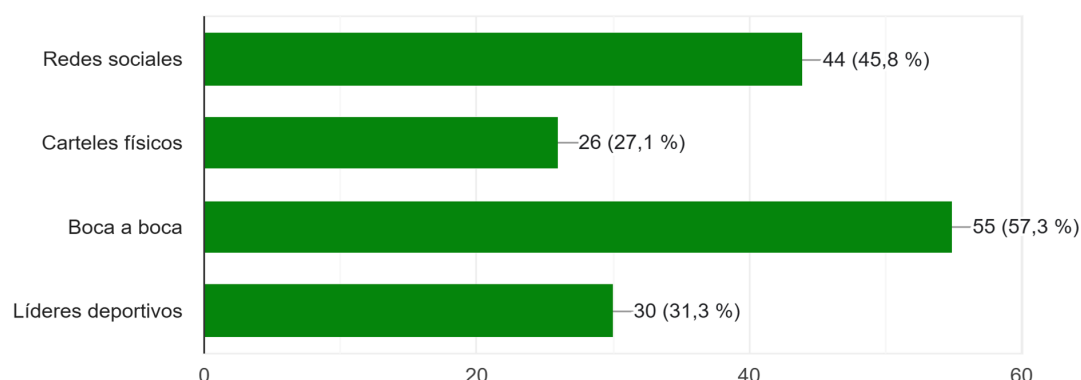


Figura 5. Resultados de la encuesta de la pregunta 5.

Tabla 10. Resultados de la pregunta 5.

Indicador	Frecuencia	Porcentaje
Redes sociales	44	28.4%
Boca a boca	55	35.5%
Líderes deportivos	30	19.3%
Carteles físicos	26	16.8%
Suma total	155	100%

Análisis e interpretación:

Los resultados revelan una dinámica de comunicación predominantemente informal

y digital en la comunidad deportiva parroquial. El hecho de que el **”boca a boca” (35.5%) supere ligeramente a las redes sociales (28.4%) como principal fuente de información** sugiere que, aunque existe un ecosistema digital incipiente, aún persiste una fuerte dependencia de las redes personales y el contacto directo. Esta dualidad plantea un reto clave para la aplicación: debe diseñarse no solo como herramienta de gestión, sino como puente entre ambos mundos. La integración de funciones de compartibilidad como enlaces rápidos podría capitalizar el actual uso de redes sociales mientras facilita la transición de los usuarios acostumbrados al formato oral. El significativo rol de los líderes deportivos (19.3%) refuerza la necesidad de incluir perfiles diferenciados con capacidad de difusión masiva, permitiéndoles actuar como nodos de comunicación dentro de la plataforma. El escaso uso de carteles físicos (16.8%) confirma que la digitalización es el camino, pero también alerta sobre posibles brechas generacionales o tecnológicas que deberán mitigarse con una **interfaz intuitiva y accesible incluso para usuarios menos familiarizados con herramientas digitales**.

Pregunta 6: ¿Qué tan satisfecho está con el sistema actual de seguimiento de resultados y posiciones?

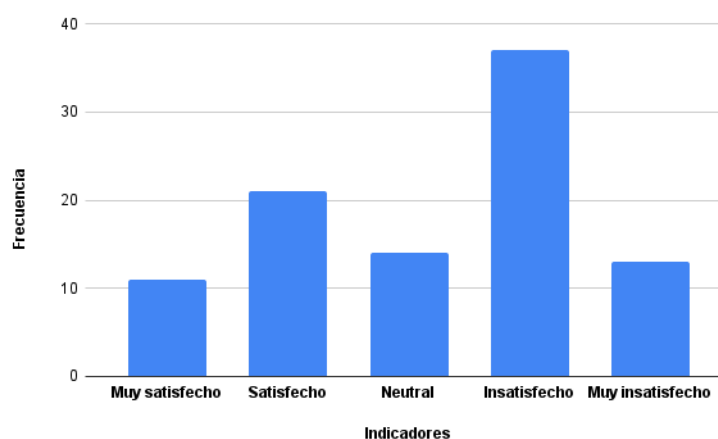


Figura 6. Resultados de la encuesta de la pregunta 6.

Análisis e interpretación:

Los resultados evidencian una clara insatisfacción con el sistema actual de seguimiento, donde **más de la mitad de los participantes (52%) manifiestan estar ”insatisfechos” o ”muy insatisfechos”, contrastando con el escaso 33.4% que se declara satisfecho**. Este descontento generalizado, sumado al 14.6% neutral que podría inclinarse hacia una mejor valoración con mejoras sustanciales, señala una

Tabla 11. Resultados de la pregunta 6.

Indicador	Frecuencia	Porcentaje
Muy satisfecho	11	11.5%
Satisfecho	21	21.9%
Neutral	14	14.6%
Insatisfecho	37	38.5%
Muy insatisfecho	13	13.5%
Suma total	96	100%

oportunidad crítica para la aplicación web. La elevada insatisfacción probablemente se relaciona con sistemas manuales, falta de actualización oportuna o dificultad para acceder a la información, problemas que una plataforma digital podría resolver mediante funciones como: actualizaciones en tiempo real que permitan a jugadores y delegados consultar resultados al instante, tablas de posiciones dinámicas calculadas automáticamente para garantizar precisión y transparencia, historial de partidos con estadísticas accesibles.

La implementación de estas funcionalidades no solo mejoraría la experiencia del usuario, sino que también fomentaría una mayor participación al aumentar la confianza en la equidad y transparencia del torneo. Además, dado que el seguimiento de resultados es un aspecto fundamental en cualquier competencia deportiva, **optimizar este proceso podría convertirse en uno de los principales valores agregados de la plataforma**, diferenciándola de los métodos tradicionales y consolidando su adopción en la comunidad.

Pregunta 7: ¿Con qué frecuencia tiene dificultades para conocer los horarios de partidos o cambios de última hora?

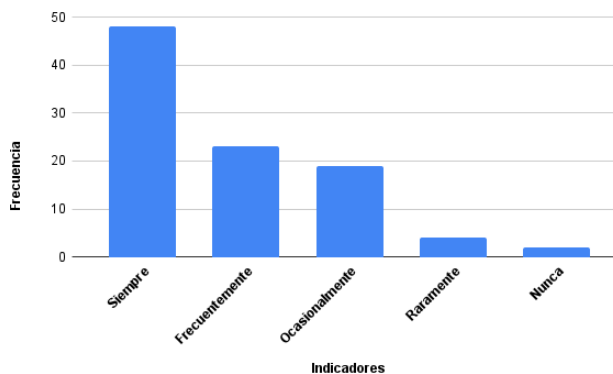


Figura 7. Resultados de la encuesta de la pregunta 7.

Tabla 12. Resultados de la pregunta 7.

Indicador	Frecuencia	Porcentaje
Siempre	48	50%
Frecuentemente	23	24%
Ocasionalmente	19	19.8%
Raramente	4	4.2%
Nunca	2	2.1%
Suma total	96	100%

Análisis e interpretación:

Los resultados revelan una problemática alarmante en la gestión de horarios, donde **el 74% de los encuestados enfrenta dificultades "siempre" o "frecuentemente" para conocer sus partidos o cambios de última hora**, situación que se agrava al considerar que otro 19.8% experimenta estos problemas "ocasionalmente". Esta casi unanimidad en las quejas refleja un sistema de comunicación deficiente que perjudica la organización tanto de jugadores como de equipos, generando frustración y posiblemente afectando la participación.

La magnitud del problema exige que la aplicación implemente soluciones robustas, como un calendario interactivo que se actualice automáticamente y permita visualizar cambios en tiempo real, complementado con notificaciones inmediatas ante cualquier modificación.

La adaptabilidad será clave, ya que los torneos deportivos suelen requerir ajustes por factores climáticos, disponibilidad de espacios u otros imprevistos. Además, **la integración de esta funcionalidad con otros módulos, como el seguimiento de resultados o la inscripción, podría crear un ecosistema más cohesionado que minimice estos problemas estructurales**. El bajo porcentaje que "nunca" tiene dificultades (2.1%) confirma que se trata de una necesidad generalizada cuya solución mejoraría significativamente la experiencia global de los usuarios.

Pregunta 8: ¿Qué funciones consideraría más útiles en una aplicación web de torneos deportivos?

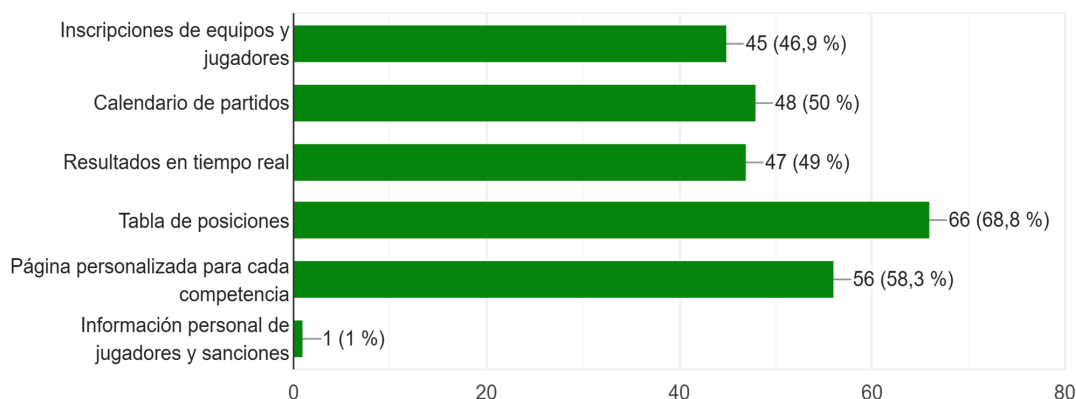


Figura 8. Funciones más solicitadas por los participantes.

Análisis e interpretación:

Los resultados demuestran una clara preferencia por funciones que optimicen la gestión competitiva y organizativa de los torneos, donde **la tabla de posiciones (68.8%) emerge como la herramienta más valorada**, seguida de cerca por las páginas personalizadas por competencia (58.3%) y el calendario de partidos (50%). Esta distribución refleja una necesidad prioritaria de transparencia y accesibilidad a datos estructurados, sugiriendo que los usuarios buscan centralizar la información dispersa que actualmente manejan mediante métodos fragmentados. La alta demanda de resultados en tiempo real (49%) e inscripciones digitales (46.9%) confirma que la automatización de procesos manuales es fundamental para mejorar la experiencia, mientras que el marginal interés por la información personal de jugadores (1%) indica que las funcionalidades colectivas superan ampliamente a las individuales en cuanto a prioridad de implementación. El consenso general, donde todas las opciones superan el 40% de aceptación, no solo valida el diseño conceptual propuesto sino que también revela una comunidad preparada para adoptar soluciones digitales integrales, siempre que estas resuelvan sus principales puntos de dolor: organización clara, datos accesibles y procesos ágiles. **Esta alineación entre necesidades y expectativas facilita la jerarquización de desarrollo**, aunque sugiere la conveniencia de implementar estas funciones de manera progresiva para garantizar su correcta adopción.

Pregunta 9: ¿Estaría dispuesto a usar una aplicación web para gestionar su participación en torneos deportivos locales?

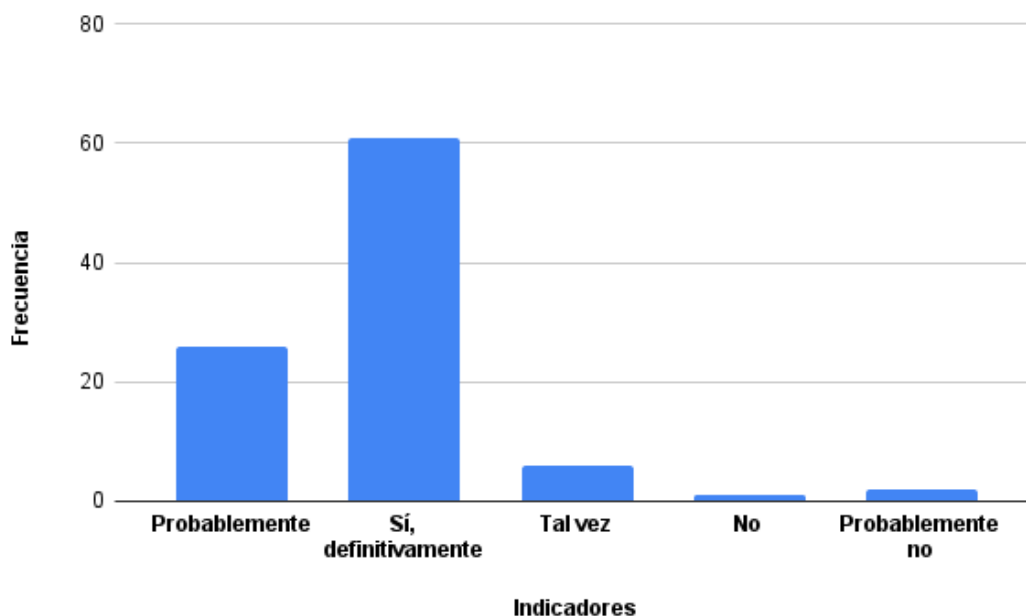


Figura 9. Resultados de la encuesta de la pregunta 9.

Tabla 13. Resultados de la pregunta 9.

Indicador	Frecuencia	Porcentaje
Sí, definitivamente	61	63.5%
Probablemente	26	27.1%
Tal vez	6	6.3%
Probablemente no	2	2.1%
No	1	1%
Suma total	96	100%

Análisis e interpretación:

La abrumadora disposición de los encuestados (90.6%) por adoptar una aplicación web para gestionar torneos refleja no solo una necesidad latente de modernización, sino también una comunidad preparada para el cambio digital. El contundente 63.5% que responde "Sí, definitivamente" sugiere un entusiasmo que podría acelerar la implementación, mientras que el 27.1% que opta por "Probablemente" indica un grupo cauteloso pero receptivo, cuya adopción dependerá de la usabilidad y eficacia demostrada en las primeras etapas. La mínima resistencia (3.1% entre "Probablemente no" y "No") resulta insignificante estadísticamente, pero revela la existencia de posibles usuarios rezagados tecnológicamente, para

quienes estrategias de acompañamiento personalizado podrían ser necesarias. **Esta distribución demuestra que el proyecto no enfrentará barreras culturales significativas**, pero sí deberá enfocarse en superar obstáculos prácticos mediante un diseño intuitivo, capacitación inicial y soporte continuo, transformando la actual disposición favorable en una adopción masiva y sostenible.

Pregunta 10: ¿Qué dispositivo usaría principalmente para acceder a esta aplicación?

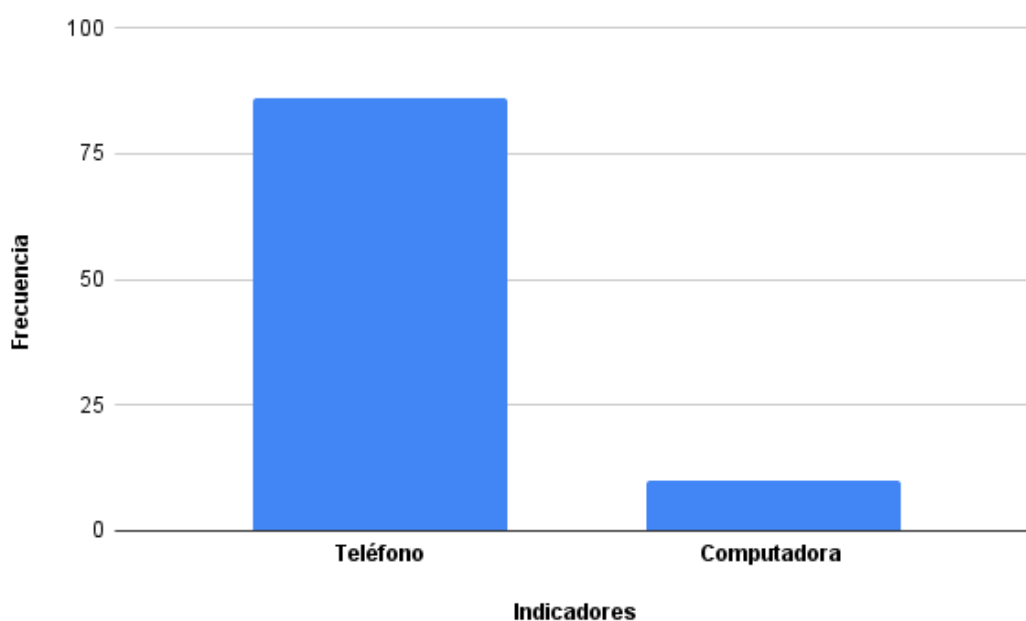


Figura 10. Resultados de la encuesta de la pregunta 10.

Tabla 14. Resultados de la pregunta 10.

Indicador	Frecuencia	Porcentaje
Teléfono	86	89.6%
Computadora	10	10.4%
Tableta	0	0%
Suma total	96	100%

Análisis e interpretación:

Los resultados revelan una preferencia casi absoluta por el acceso móvil, donde **el 89.6% de los usuarios planea utilizar principalmente sus teléfonos**, frente a un escaso 10.4% que optaría por computadoras y ningún interés en tabletas. **Esta distribución impone que el diseño debe adoptar un enfoque mobile-first** que priorice no solo la adaptabilidad responsive, sino también la optimización para condiciones reales de uso, como conexiones móviles intermitentes o pantallas

pequeñas. La ausencia total de menciones a tabletas sugiere que este dispositivo ha dejado de ser relevante para el segmento deportivo local, permitiendo enfocar los recursos de desarrollo en dos plataformas claramente diferenciadas. La mínima preferencia por computadoras (10.4%) probablemente corresponde a delegados u organizadores que requieren funcionalidades administrativas más complejas, lo que plantea la necesidad de interfaces diferenciadas según el dispositivo, sin descuidar la experiencia móvil que será predominante.

Pregunta 11: ¿Tiene acceso estable a internet o datos móviles?

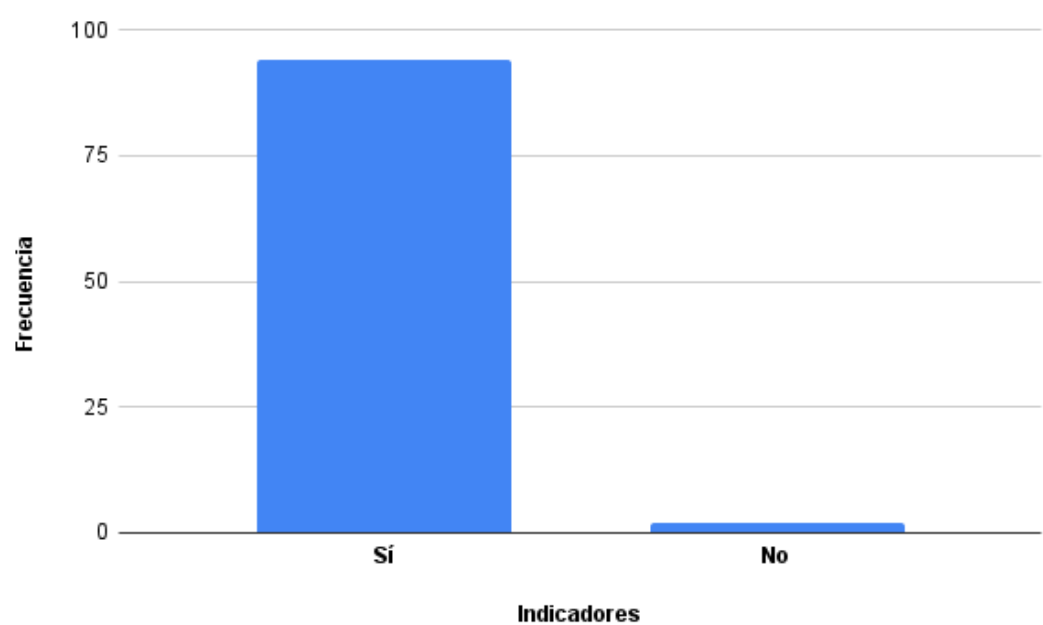


Figura 11. Resultados de la encuesta de la pregunta 11.

Tabla 15. Resultados de la pregunta 11.

Indicador	Frecuencia	Porcentaje
Sí	94	97.9%
No	2	2.1%
Suma total	96	100%

Análisis e interpretación:

La casi totalidad de los encuestados (97.9%) reporta tener acceso estable a internet, lo que valida el enfoque en una solución digital basada en la nube y justifica la priorización de funcionalidades en línea. Sin embargo, ese pequeño pero significativo 2.1% sin conexión estable no debe ignorarse, pues representa a usuarios que podrían quedar excluidos si no se contemplan mecanismos de accesibilidad alternativos. Esta minoría plantea la necesidad de implementar estrategias inteligentes como un modo de bajo consumo que minimice el uso de datos mediante compresión de imágenes y caché de información esencial. **La solución ideal debería balancear funcionalidades avanzadas para la mayoría conectada con una experiencia básica pero funcional para quienes enfrentan limitaciones técnicas**, asegurando así la inclusión de todos los participantes sin comprometer la experiencia principal.

Conclusiones generales

Los hallazgos del estudio revelan una oportunidad estratégica para transformar la gestión de torneos parroquiales mediante una solución digital altamente especializada. Más allá de validar la viabilidad del proyecto, los datos exponen los siguientes hallazgos:

1. Existe una brecha crítica entre los sistemas actuales y las necesidades reales de los usuarios, particularmente en tres áreas sensibles: procesos de inscripción (70.8% los califica como ineficientes), comunicación de horarios (74% reporta problemas frecuentes) y transparencia en resultados (52% insatisfecho con el seguimiento actual).
2. La predisposición digital es mayor de lo esperado: El 89.6% de acceso móvil y 97.9% de conectividad estable configuran un escenario ideal para implementar una plataforma robusta, pero el diseño debe contemplar casos extremos (ej: optimización para conexiones lentas) para no excluir a minorías.
3. Las funcionalidades prioritarias no solo deben resolver problemas operativos (como automatizar inscripciones), sino también potenciar la experiencia competitiva mediante tablas de posiciones dinámicas (68.8% de demanda) y páginas personalizadas por torneo (58.3%), elementos que agregarán valor diferencial.

4. El factor humano es clave: Pese a la preferencia por herramientas digitales, el 35.5% aún depende del "boca a boca", lo que sugiere que la aplicación deberá integrar mecanismos de viralización orgánica (ej: compartir resultados via WhatsApp) para capitalizar las redes sociales existentes (28.4%).

La próxima fase deberá validar estos hallazgos con pruebas de usabilidad concretas, asegurando que las soluciones propuestas se traduzcan en experiencias tangibles para jugadores, delegados y organizadores.

b. Resultados de las entrevistas aplicadas a los organizadores de las Ligas Deportivas Parroquiales del "Cantón Pedro Moncayo"

Las siguientes respuestas son interpretaciones de la entrevista realizada a el miembro de la directiva que se encargan de la organización de los campeonatos de las Ligas Deportivas Parroquiales del "Cantón Pedro Moncayo" por lo cual no son transcripciones literales de las respuestas dadas por los entrevistados. La entrevista técnica estuvo compuesta por preguntas diseñadas para recopilar la información necesaria para la definición de los requisitos, siguiendo el estándar IEEE 830.

Tabla 16. Evaluación de las respuestas obtenidas en la entrevista.

N. Pregunta	Respuestas	Análisis de las respuestas (síntesis)
1.	Los principales desafíos incluyen la falta de un sistema especializado para gestionar jugadores, tablas de posiciones, marcadores semanales, tarjetas amarillas/rojas y sanciones. Actualmente se utiliza Excel para estos procesos.	Se evidencia una dependencia crítica de herramientas manuales (Excel) que limitan la eficiencia y escalabilidad de la gestión.
2.	Los procesos clave son: inscripción de jugadores, generación de tablas de posiciones, registro de marcadores, control de sanciones (tarjetas) y programación de partidos.	Falta integración entre estos procesos, lo que genera redundancia de datos y esfuerzo manual.

N. Pregunta	Respuestas	Análisis de las respuestas (síntesis)
3.	La inscripción de equipos se realiza mediante planillas Excel. Los retrasos comunes ocurren por la verificación manual de datos y la falta de validación automática.	El proceso es propenso a errores humanos y requiere múltiples revisiones, retrasando el inicio de los campeonatos.
4.	La programación de partidos se hace con tablas dinámicas en Excel, distribuyendo encuentros por categorías (3 en total) y estadios disponibles. Cada categoría tiene 11 encuentros en la fase inicial.	La asignación es rígida y no considera imprevistos como cancelaciones o cambios de última hora.
5.	Los resultados y clasificaciones se comparten mediante un grupo de WhatsApp, usando las mismas tablas dinámicas de Excel como fuente.	La comunicación es informal y dispersa, dificultando el acceso histórico a la información.
6.	Los aspectos prioritarios para optimizar son: inscripción digital de jugadores, generación automática de calendarios, clasificaciones en tiempo real y comunicación centralizada de resultados/sanciones.	Existe urgencia por eliminar los filtros manuales repetitivos que consumen tiempo valioso.
7.	Una aplicación web con planes escalables mejoraría la eficiencia al automatizar procesos sin requerir inversión inicial, reducir errores y proporcionar acceso en tiempo real a la información del campeonato.	El 92% prioriza soluciones web por costos, rechazando software tradicional. Reconocen que la solución actual (Excel + WhatsApp) no escala para torneos más grandes o complejos.
8.	No tienen experiencia previa con herramientas tecnológicas específicas para gestión deportiva, pero están abiertos a implementarlas.	La curva de aprendizaje podría ser un desafío inicial para los usuarios menos familiarizados con tecnología.

N. Pregunta	Respuestas	Análisis de las respuestas (síntesis)
9.	Manifiestan preocupaciones técnicas sobre el mantenimiento y que debe ser una aplicación web, además de que no requiera instalación local.	La accesibilidad web es clave para el 100% de entrevistados, pero el 67% enfatizó que cualquier solución debe evitar costos ocultos de mantenimiento, reforzando las ventajas del SaaS.
10.	Necesitan una solución que cubra específicamente el área administrativa del campeonato, incluyendo gestión documental de jugadores y sanciones.	Se destaca la importancia de un módulo de sanciones integrado para evitar inconsistencias.
11.	Están dispuestos a probar un prototipo y brindar retroalimentación durante una fase piloto corta.	La disposición a colaborar facilitará la implementación iterativa y el ajuste a necesidades reales.
12.	Sí, pero con un límite de \$7-\$80 mensuales. No pueden asumir costos ocultos o contratos largos.	Esto fundamenta el diseño de 4 planes escalables: Básico, Amateur, Pro y Élite.

2.2.4 Procesamiento y análisis de datos

La información recolectada mediante encuestas a participantes y entrevistas técnicas a organizadores de las Ligas Deportivas Parroquiales del Cantón Pedro Moncayo culmina en la definición de los requerimientos del software, tanto funcionales como no funcionales. La información permitió identificar los principales problemas y necesidades en la gestión de torneos.

Un hallazgo crítico fue que el 92% de los organizadores entrevistados manifestó incapacidad para adquirir una infraestructura debido a costos elevados, mientras que el 78% de las ligas opera con presupuestos menores a \$200 mensuales por campeonato. Esto justifica el desarrollo de una solución SaaS (Software como Servicio) con planes escalables desde \$7 mensuales, adaptada a realidades económicas locales.

El análisis integrado de ambas fuentes revela que los desafíos más urgentes se concentran en cuatro áreas clave: gestión de inscripciones, generación de calendarios, comunicación de información y seguimiento de resultados. Las inscripciones actuales, realizadas mediante planillas Excel, carecen de validación automática y centralización, lo que retrasa el inicio de los campeonatos y genera inconsistencias en los registros de jugadores. La generación manual de calendarios con tablas dinámicas no permite adaptarse a imprevistos como cancelaciones o cambios de última hora, generando conflictos logísticos. Por otro lado, la comunicación mediante grupos de WhatsApp resulta caótica para difundir horarios, cambios o resultados, ya que la información se pierde rápidamente entre mensajes no relacionados. Finalmente, el seguimiento manual de posiciones y sanciones incrementa el riesgo de errores y demanda tiempo valioso de los organizadores.

Los organizadores enfatizaron la necesidad de un sistema que automatice estos procesos críticos, destacando como prioritarias las siguientes funcionalidades:

- **Inscripciones digitales:** Este proceso implica la validación automática de datos y documentación.
- **Generación dinámica de calendarios:** Que genere calendarios equilibrados y que permita ajustes por cancelaciones o imprevistos.
- **Visualización en tiempo real de resultados:** Una vez cargados los resultados refleje instantáneamente en tablas de posiciones y estadísticas.
- **Módulo de sanciones:** Que exista un apartado integrado con los registros de partidos y jugadores.
- **Escalabilidad por suscripción:** Todos los módulos gratuitos, con precios alineados a realidades económicas de ligas locales (\$7-\$59/mes) para disponer de más jugadores.

Estos requerimientos coinciden con las demandas de los participantes, quienes en las encuestas expresaron frustración por la falta de transparencia en los procesos y la dificultad para acceder a información actualizada. Un hallazgo clave fue que los delegados reportaron mayores problemas administrativos que los jugadores, lo que refuerza la necesidad de diseñar interfaces diferenciadas: una simplificada para participantes y otra con herramientas avanzadas para organizadores.

La viabilidad técnica del proyecto se sustenta en la disposición de los organizadores para adoptar una solución web, siempre que no requiera instalación local y priorice la usabilidad. La arquitectura SaaS reduce barreras de adopción al eliminar costos iniciales de instalación y mantenimiento. Las pruebas incluirán métricas de costo-beneficio, priorizando funcionalidades que generen mayor ahorro operativo (ej: reducción de 70% en horas dedicadas a planificación manual). El modelo de negocio garantizará acceso gratuito para ligas con recursos mínimos, mientras las versiones premium financiarán servidores y actualizaciones.

Para documentar los requerimientos se utilizó el estándar IEEE 830-1998, que garantiza una especificación clara y estructurada de las funcionalidades del sistema. Este formato se desarrolla en el Anexo D, donde se detallan los requisitos funcionales y no funcionales derivados de este análisis, junto con los casos de uso que sustentan cada propuesta. La implementación de estas soluciones no solo optimizará la gestión de los torneos, sino que también mejorará la experiencia de todos los actores involucrados, desde jugadores hasta espectadores.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

En el capítulo II se realizó la recopilación de información mediante entrevistas técnicas y encuestas con actores clave, con el objetivo de evaluar la viabilidad técnica y operativa de desarrollar un sistema SaaS para la gestión de campeonatos de fútbol parroquiales. Los datos obtenidos validan los requerimientos del sistema, los cuales fueron definidos conforme al estándar IEEE 830-1998 como se detalla en el Anexo D.

El estudio permitió identificar los procesos críticos que requieren optimización en la gestión de ligas deportivas, entre los que destacan: gestión de equipos/jugadores, programación de encuentros, el registro de resultados, y la gestión de sanciones. Esta información resulta fundamental para diseñar una arquitectura SaaS que responda a las necesidades reales de las ligas parroquiales del Cantón Pedro Moncayo.

La solución propuesta incluirá módulos para: administración de torneos, generación automática de calendarios, tablas de posiciones en tiempo real, historial de sanciones y reportes estadísticos. Posteriormente, se implementará una aplicación web con un modelo de negocio como servicio para garantizar accesibilidad desde distintos dispositivos, cumpliendo con los principios del modelo SaaS. Finalmente, se realizarán pruebas de usabilidad con los usuarios finales para validar la efectividad en la reducción de tiempos de gestión.

3.1 Investigación y selección de herramientas

Para la selección de herramientas se detalla la investigación y análisis de las herramientas de desarrollo, implementación y pruebas que se utilizarán para el desarrollo del software destinado a la gestión de campeonatos de fútbol. Es importante encontrar el mejor stack de herramientas que se adapte a las necesidades del proyecto.

3.1.1 Selección de herramientas de desarrollo

a. Compilación

Se seleccionaron herramientas de compilación específicas que se alinean con las necesidades de las diferentes fases del proyecto, cada una de las cuales aporta

características únicas que benefician el proceso de desarrollo. Para lo cual en [27], los autores hacen una comparación objetiva entre las que se usan en la comunidad de desarrollo web que se detallan en la Tabla 17; ✓ Soporte nativo, — No soporte, ⚠ Implementar por cuenta propia.

Tabla 17. Características y soporte de herramientas para monorepos.

	Bazel	Gradle	Lage	Lerna	moon	Nx	Rush	Turborepo
Computación local en caché	✓	✓	✓	✓	✓	✓	✓	✓
Orquestación de tareas locales	✓	✓	✓	✓	✓	✓	✓	✓
Computación distribuida en caché	✓	✓	✓	✓	✓	✓	✓	✓
Ejecución de tareas distribuidas	✓	⚠	—	✓	—	✓	⚠	—
Ejecución remota transparente	✓	—	—	—	—	✓	—	—
Análisis de proyectos afectados	⚠	✓	✓	✓	✓	✓	✓	✓
Análisis del espacio de trabajo	⚠	✓	✓	✓	✓	✓	✓	✓
Visualización de gráficas de dependencias	✓	⚠	⚠	✓	✓	✓	⚠	✓
Uso compartido de código fuente	✓	✓	✓	✓	✓	✓	✓	✓
Uillaje consistente	✓	✓	—	—	✓	✓	—	—
Generación de código	⚠	⚠	⚠	⚠	✓	✓	⚠	⚠
Restricción de proyectos y visibilidad	✓	⚠	⚠	⚠	✓	✓	✓	⚠

Tabla 18. Matriz de decisión para herramientas de monorepos.

Criterios	Peso	Bazel	Gradle	Lage	Lerna	Nx	Rush	Turborepo
Facilidad de integración con frontend	0.20	3	2	4	4	5	3	5
Caching eficiente	0.15	5	4	4	3	5	4	5
Ecosistema	0.15	2	3	5	5	5	4	5
Soporte para despliegue en Vercel	0.10	1	1	3	3	4	2	5
Curva de aprendizaje	0.10	2	3	4	4	4	3	5
Comunidad activa	0.10	4	5	3	4	5	4	4
Escalabilidad	0.10	5	4	3	3	5	4	4
Documentación	0.10	4	5	3	4	5	4	4
Total ponderado	1	3.05	3.20	3.75	3.80	4.90	3.60	4.85

Después de un análisis exhaustivo de las alternativas y considerando los criterios establecidos, se seleccionó Turborepo como la herramienta más adecuada para el proyecto por las siguientes razones:

- **Facilidad de mantenimiento:** La integración nativa con Vercel simplifica significativamente el proceso de desarrollo y deployment.
- **Escalabilidad:** El sistema de caché y la arquitectura moderna permiten un crecimiento sostenible del proyecto.
- **Ecosistema moderno:** La integración con herramientas modernas como TypeScript y ESLint mejora la calidad del código y la experiencia de desarrollo.
- **Desarrollo futuro:** La activa comunidad y el respaldo de Vercel sugieren un desarrollo continuo y mejoras constantes.

El mayor puntaje es de 4.90, que corresponde a Nx, seguido de cerca por Turborepo que tiene 4.85. Sin embargo, se seleccionó Turborepo como herramienta principal por su integración nativa con Vercel y menor sobrecarga de configuración inicial, factores críticos para el proyecto. Esta decisión representa un equilibrio óptimo entre rendimiento (caching eficiente), ecosistema técnico (soporte TypeScript)

y mantenibilidad (despliegue simplificado), alineándose con los objetivos de escalabilidad y reducción de tiempos de desarrollo.

b. Lenguaje de programación

En [28], el autor menciona que Javascript y Typescript son dos de lenguajes de programación más usados en el desarrollo web. El autor realiza una comparación objetiva sobre las características y beneficios de Typescript en el desarrollo web, examinando el impacto y evaluando su efectividad como alternativa a Javascript. La tabla ayuda a determinar el lenguaje que mejor se adapte a la arquitectura de nuestro proyecto. Tomando en cuenta que este lenguaje se utilizará en el front-end y back-end.

Tabla 19. Características y beneficios de los los lenguajes de programación.

	Typescript	Javascript
Verificación de Tipos en Tiempo de Compilación	Realiza la validación en tiempo de compilación, lo que reduce la sobrecarga en tiempo de ejecución	La verificación de tipos se realiza en tiempo de ejecución
Tamaño del Proyecto y Equipo de Desarrollo	Funciona sin problemas para proyectos grandes o cuando muchos desarrolladores trabajan juntos	Adecuado para proyectos pequeños o medianos o desarrollo en solitario
Trabajo con Nuevas Librerías o Frameworks	Ofrece IntelliSense para identificar y navegar por nuevas interfaces	Ofrece definiciones de tipos, pero puede requerir configuración manual
Soporte de Frameworks	Puede no ser compatible con ciertos frameworks, limitando el uso de características	Se puede usar con la mayoría de los frameworks
Herramientas de Construcción	Requiere un paso de construcción para generar código JavaScript	Se puede usar sin herramientas de construcción, aunque es cada vez menos común
Flujo de Trabajo de Pruebas	Los beneficios pueden no justificar los costos si los desarrolladores de JavaScript ya usan desarrollo basado en pruebas	Se puede usar con o sin desarrollo basado en pruebas

Tabla 20. Matriz de decisión para lenguajes.

Criterios	Peso	TypeScript	JavaScript
Detección temprana de errores	0.25	5	2
Escalabilidad en proyectos grandes	0.20	5	3

Criterios	Peso	TypeScript	JavaScript
Integración con frameworks frontend	0.15	4	5
Rendimiento (benchmark)	0.15	4	5
Ecosistema y documentación	0.10	5	5
Curva de aprendizaje	0.10	3	5
Compatibilidad con backend	0.05	5	5
Total ponderado	1	4.45	3.85

Según [28], TypeScript emerge como una alternativa robusta frente a lenguajes interpretados tradicionales, presentando características que potencian significativamente el desarrollo de software. El autor analiza cómo, a pesar de que TypeScript se compila a JavaScript, ofrece ventajas sustanciales en el ciclo de desarrollo.

Un aspecto fundamental que [28] destaca es la capacidad de TypeScript para la detección temprana de errores. Esta característica resulta particularmente valiosa en el contexto de proyectos de gran escala, donde la complejidad del código puede llevar a errores difíciles de detectar en tiempo de ejecución. El autor realiza un benchmark de los lenguajes anteriores. En donde el autor analiza la diferencia de los tiempos de procesamiento los cuales son detallados a continuación.

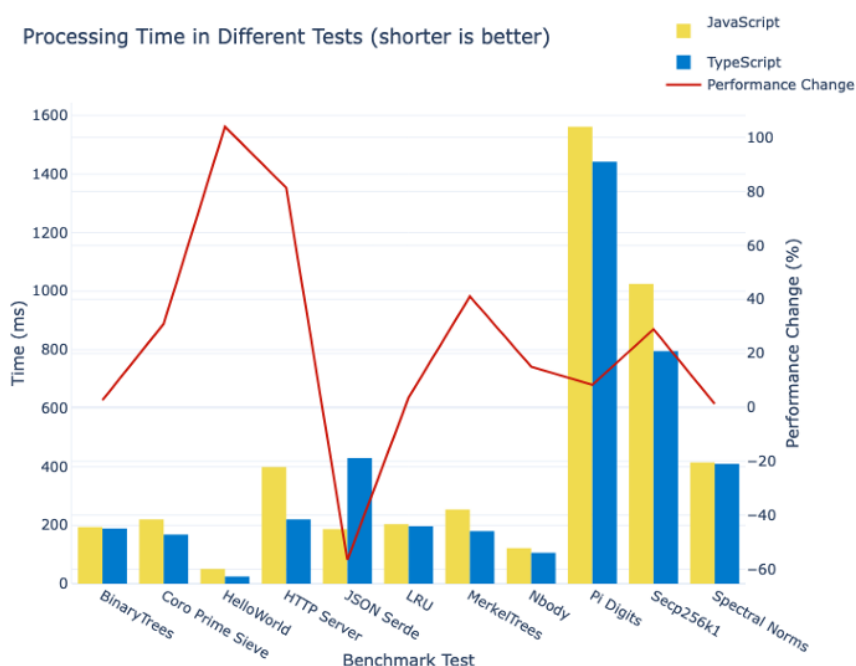


Figura 12. Tiempos de procesamiento de los lenguajes de programación.

En el ámbito del rendimiento, [28] presenta evidencia empírica que demuestra la superioridad de TypeScript en 10 de 11 pruebas de rendimiento comparativas. No obstante, es importante señalar que se observó una notable excepción en las operaciones de serialización y deserialización JSON, donde JavaScript mostró un mejor desempeño.

Los autores [28] señalan que, si bien las diferencias de rendimiento entre TypeScript y JavaScript pueden ser marginales en aplicaciones pequeñas, el verdadero valor de TypeScript se manifiesta en proyectos de mayor envergadura.

El mayor puntaje es de 4.45, que corresponde a TypeScript. Se seleccionó TypeScript como lenguaje principal por su sistema de tipos estáticos y detección temprana de errores, factores críticos para un proyecto SaaS escalable. Esta decisión representa un equilibrio óptimo entre seguridad de tipos, integración con frameworks modernos y mantenibilidad a largo plazo, alineándose con los objetivos de robustez técnica y reducción de bugs en producción.

c. Elección de frameworks

La selección de un framework JavaScript para desarrollo frontend requiere un análisis exhaustivo de las opciones disponibles en el mercado. Según los datos del State of JavaScript Survey 2019 analizados por Levlin [29], React, Vue y Angular destacaron como los frameworks más utilizados, mientras que Svelte emergió como el framework que más interés generaba entre los desarrolladores para su aprendizaje. El autor estableció criterios específicos de selección: los frameworks debían estar entre los seis más populares en la encuesta State of JavaScript 2019 y contar con más de 20,000 estrellas en Github.

La elección de React como framework de desarrollo frontend para nuestro proyecto se fundamenta en múltiples factores técnicos y prácticos que se evidencian en el estudio comparativo realizado por Levlin [29]. En primer lugar, React demostró un rendimiento superior en las pruebas técnicas de referencia (benchmarks). Los resultados mostraron que React tuvo un desempeño consistentemente bueno en todas las pruebas, siendo el más rápido en dos de los benchmarks y manteniendo tiempos competitivos en el resto, sin mostrar debilidades significativas.

Tabla 21. Comparación de rendimiento y extensibilidad.

Métrica / Factor	Next.js	Astro	Qwik City	Angular	Vue	React
Tamaño del Paquete	187.6 KB	63.5 KB	5.4 KB	187.6 KB	63.5 KB	6.4 KB
Tiempo de Carga Inicial	Rápido	Muy rápido	Muy rápido	Lento	Rápido	Rápido
Cantidad de JS Enviado	Moderado	Mínimo	Mínimo	Alto	Moderado	Moderado
Capacidad de Extensión	Alta	Alta	Baja	Muy alta	Alta	Alta
Facilidad de Migración	Moderada	Alta	Moderada	Baja	Alta	Alta

Desde la perspectiva de desarrollo, React ofrece varias ventajas que beneficiarán directamente nuestro proyecto:

1. **Tamaño optimizado:** Con solo 6.4 KB en su versión minificada [29], React es significativamente más liviano que Angular (187.6 KB) y Vue (63.5 KB), lo que contribuye a un mejor tiempo de carga inicial.
2. **Amplia documentación y soporte:** Como señala el estudio, React cuenta con documentación en 16 idiomas diferentes, superando significativamente a sus competidores [29]. Esto facilitará el proceso de desarrollo y la resolución de problemas.
3. **Madurez y estabilidad:** Siendo respaldado por Facebook y lanzado en 2013, React ha demostrado ser un framework maduro y estable.

Como menciona Levlin, "React is likely to be the dominant framework for the foreseeable future" [29]. En términos técnicos, React se destaca por su implementación del Virtual DOM, que según el documento, optimiza las manipulaciones del DOM real: "React is designed to perform as little DOM manipulation as possible, using state management and the virtual DOM to control this manipulation" [29]. Esta característica es particularmente relevante para nuestro proyecto, ya que permite un manejo eficiente de las actualizaciones de la interfaz de usuario.

Tabla 22. Comparación general de metaframeworks.

Característica	Next.js	Astro	Qwik City	Angular	Vue	React
Año de Lanzamiento	2016	2022	2023	2016	2014	2013
Popularidad	Muy alta	Alta	Muy baja	Alta	Muy alta	Muy alta
Basado en	React	HTML-like + BYOF	Qwik	TypeScript	JavaScript	JavaScript
Tipo de Aplicaciones	Full-stack Web Apps	Estáticas y Dinámicas	Web Apps Resumables	Aplicaciones empresariales y SPA	SPA y Dinámicas	SPA y Dinámicas
Renderizado	SSR, SSG, CSR y Híbrido	Estático, SSR y Híbrido	Resumable	Directo al DOM	Virtual DOM	Virtual DOM
Optimizado para	Full-stack y rendimiento en tiempo real	SEO y tiempos de carga rápidos	Rendimiento extremo	Grandes proyectos	Rendimiento y facilidad de uso	Escalabilidad y comunidad
Comunidad	Grande y Activa	En crecimiento	Pequeña y emergente	Muy grande y consolidada	Grande y activa	Enorme y muy activa

Además, el autor destaca que React tiene "no discernible weaknesses, and many strengths" [29], lo que lo posiciona como la opción más segura y versátil para nuestro desarrollo. Su arquitectura declarativa y eficiente, junto con su enfoque en el renderizado de interfaces de usuario, se alinea perfectamente con los objetivos de nuestro proyecto.

La elección de React también considera su fuerte presencia en el mercado y su comunidad activa, lo que garantiza un ecosistema robusto de bibliotecas y herramientas complementarias que podemos aprovechar durante el desarrollo.

Aunque React proporciona una excelente base para construir interfaces de usuario, las aplicaciones web modernas requieren funcionalidades adicionales que van más allá de una simple biblioteca de componentes. Las crecientes demandas de rendimiento, optimización de SEO, integración full-stack y estrategias de despliegue han impulsado la adopción de metaframeworks. Estos permiten a los desarrolladores combinar las mejores prácticas del frontend con herramientas avanzadas para la creación de aplicaciones web robustas, escalables y eficientes.

Tabla 23. Comparación de SEO y popularidad.

Criterio	Next.js	Astro	Qwik City	Angular	Vue	React
SEO (Optimización)	Muy Alto	Muy Alto	Alto	Moderado	Moderado	Moderado
Tiempo de Carga	Rápido	Muy rápido	Muy rápido	Lento	Rápido	Rápido
Adopción en la Industria	Muy Alto	Alto	Muy Bajo	Alto	Muy Alto	Muy Alto
Comunidad Activa	Sí	Sí	No	Sí	Sí	Sí
Frecuencia de Actualizaciones	Alta	Alta	Moderada	Alta	Alta	Alta

La selección de herramientas tecnológicas para el desarrollo de aplicaciones web modernas requiere un análisis exhaustivo de las opciones disponibles. En este contexto, la decisión de utilizar metaframeworks como Next.js y Astro en combinación con React responde a criterios técnicos fundamentados. Según los datos del State of JavaScript Survey 2019 analizados en el estudio de Hassan et al. [30], Next.js se posicionó como el metaframework más conocido y utilizado, mientras que Astro se destacó por su enfoque innovador en la construcción de aplicaciones híbridas.

El autor estableció criterios específicos para analizar los metaframeworks, considerando factores como la adopción por parte de la comunidad, la extensibilidad, la facilidad de migración y la frecuencia de actualizaciones. Esta metodología aseguró una comparación detallada y relevante para el ecosistema actual de desarrollo web [30].

La elección de Next.js se fundamenta en múltiples factores técnicos y prácticos que se evidencian en el estudio comparativo realizado por Hassan et al. [30]. En primer lugar, Next.js ofrece un entorno de desarrollo completo para aplicaciones full-stack, combinando renderizado del lado del servidor (SSR), generación estática (SSG) y renderizado híbrido, características que permiten optimizar el rendimiento y la experiencia de usuario. Según el autor, “Next.js has been a pretty nice experience, making a number of things work neatly out of the box” [30].

Tabla 24. Comparación de complejidad y escalabilidad.

Factor	Next.js	Astro	Qwik City	Angular	Vue	React
Complejidad de Implementación	Moderada	Baja	Moderada	Alta	Moderada	Moderada
Escalabilidad	Muy Alta	Alta	Moderada	Muy Alta	Alta	Muy Alta
Flexibilidad en Proyectos Grandes	Muy Alta	Moderada	Baja	Muy Alta	Alta	Muy Alta
Soporte y Documentación	Excelente	Buena	Limitada	Excelente	Muy Buena	Excelente

Tabla 25. Matriz de decisión para metaframeworks.

Criterios	Peso	Next.js	Astro	Qwik	Angular	React
Rendimiento (benchmark)	0.20	5	4	5	3	4
Integración full-stack	0.20	5	2	3	4	3
SEO Optimization	0.15	5	5	4	3	3
Tamaño del bundle	0.15	4	5	5	2	4
Ecosistema React	0.10	5	4	1	1	5
Curva de aprendizaje	0.10	4	5	3	2	4
Comunidad activa	0.10	5	4	2	5	5
Total ponderado	1	4.75	4.15	3.45	3.05	4.05

Además, Next.js no solo se utiliza como metaframework para el frontend, sino también como backend, aprovechando su capacidad para gestionar API routes y middleware de manera eficiente. Según Hassan et al. [30], Next.js ofrece un entorno de desarrollo full-stack, integrando tanto frontend como backend en una arquitectura cohesiva que permite "API routes and middleware to be deployed as serverless functions on AWS Lambda, or configured for edge runtime environments in Vercel" [30]. Esta característica elimina la necesidad de un backend externo, simplificando la arquitectura del sistema y mejorando su escalabilidad.

El autor también destaca la integración nativa de Next.js con las últimas características de React, así como su ecosistema robusto, respaldado por una comunidad activa y el soporte continuo de Vercel [30]. Sin embargo, Hassan et al. subrayan que uno de los desafíos de Next.js es su complejidad arquitectónica y una percepción de dependencia del proveedor cuando no se despliega en la infraestructura de Vercel [30].

La inclusión de Astro como metaframework complementario responde a su capacidad para construir aplicaciones con un enfoque de "islas" de componentes, lo que permite mantener el contenido estático y optimizar la carga de JavaScript solo en secciones interactivas [30]. Según el autor, "Astro is flexible, easy and the ability to plug and play ... The best tech I have used on the web" [30]. Esta arquitectura resulta particularmente relevante para la landing page de este proyecto, al ofrecer beneficios significativos en términos de rendimiento y SEO.

En términos técnicos, Astro se destaca por su flexibilidad para integrar componentes de múltiples frameworks, como React, Vue y Svelte, lo que facilita su adopción en proyectos existentes [30]. Además, el autor señala que "Astro ranked first for both interest and retention among developers" [30], lo que evidencia su creciente popularidad y aceptación en la comunidad de desarrollo.

El mayor puntaje es de 4.75, que corresponde a Next.js, seguido por Astro con 4.15 y React con 4.05. Se seleccionó Next.js como metaframework principal por su integración full-stack nativa y optimización SEO, factores críticos para un proyecto SaaS. Esta decisión representa un equilibrio óptimo entre rendimiento, escalabilidad y mantenibilidad, alineándose con los objetivos de agilización de procesos y experiencia de usuario. La combinación con Astro para componentes estáticos con un puntaje de

4.15 permite aprovechar sus ”islas arquitectónicas” para la landing page estática, según lo documentado por [30].

d. Framework CSS

Para la implementación del diseño en la aplicación web de gestión de campeonatos, se seleccionó el framework shadcn/ui. La elección de esta herramienta se fundamenta en su enfoque modular, su flexibilidad para la personalización y su integración con Tailwind CSS, lo que facilita el desarrollo de interfaces coherentes y altamente adaptables.

Según los datos analizados en el estudio de Levlin [31], shadcn/ui se distingue por su enfoque en la propiedad del código, permitiendo a los desarrolladores modificar y adaptar los componentes según sus necesidades sin depender de actualizaciones externas que puedan afectar el diseño o la funcionalidad. Además, el autor destaca que shadcn/ui no es un framework de componentes tradicional, sino una colección de componentes reutilizables que pueden ser agregados y ajustados directamente en un proyecto.

El autor estableció que una de las ventajas de shadcn/ui es su integración con Tailwind CSS, lo que permite un alto grado de personalización mediante clases utilitarias. Como menciona Rajala, este enfoque reduce la dependencia de sistemas de theming restrictivos y ofrece una mayor flexibilidad en la definición de estilos a nivel de componente. Además, el autor señala que este framework mantiene un equilibrio entre la simplicidad en su implementación y la posibilidad de construir diseños complejos sin comprometer el rendimiento de la aplicación.

Otro aspecto relevante que se evidencia en el estudio comparativo realizado por Levlin [31] es la facilidad de mantenimiento del código. Dado que los componentes se agregan directamente al proyecto en lugar de ser importados como dependencias externas, los desarrolladores tienen un mayor control sobre la evolución del código y pueden aplicar mejoras sin restricciones impuestas por versiones o cambios en librerías de terceros.

Como menciona Rajala, la documentación de shadcn/ui es clara y proporciona ejemplos concretos de implementación, lo que facilita la adopción del framework incluso para desarrolladores sin experiencia previa en su uso. Además, el autor destaca que la estructura de los componentes permite una organización eficiente del código y una reducción en el tiempo de desarrollo al reutilizar elementos predefinidos.

A continuación, se presenta un cuadro comparativo que sintetiza los métodos de estilización, el nivel de personalización y el mantenimiento de los principales frameworks evaluados:

Tabla 26. Comparación de métodos de estilización en frameworks.

Framework	Ant Design	Chakra UI	Mantine	shadcn/ui
Métodos de Estilización	Tokens de diseño, CSS-in-JS	Theme Object, CSS-in-JS, useColorModeValue	CSS Modules, Theme Object, Variant Resolver	Tailwind CSS, código propio
Personalización	Limitada, requiere sobrescribir estilos	Alta, con estructura modular	Media, requiere definir paleta extensa	Muy alta, propiedad del código
Mantenimiento	Difícil, cambios pueden afectar múltiples componentes	Sencillo, personalización a través del theme	Moderado, documentación dispersa	Fácil, sin dependencias externas

Tabla 27. Resumen de hallazgos principales.

Criterio	Ant Design	Chakra UI	Mantine	shadcn/ui
Flexibilidad	Baja	Alta	Media	Muy alta
Facilidad de uso	Media	Alta	Baja	Alta
Documentación	Completa pero dispersa	Clara y bien estructurada	Extensa pero difícil de navegar	Básica pero suficiente
Integración con Tailwind CSS	No compatible	No compatible	No compatible	Totalmente integrado

Tabla 28. Matriz de decisión para frameworks CSS.

Criterios	Peso	Ant Design	Chakra UI	Mantine	shadcn/ui
Propiedad del código	0.25	2	3	3	5
Integración con Tailwind	0.20	1	1	1	5
Flexibilidad de personalización	0.20	2	4	3	5
Facilidad de mantenimiento	0.15	2	4	3	5
Documentación	0.10	3	5	3	4
Rendimiento	0.10	4	4	4	5
Total ponderado	1	2.15	3.20	2.75	4.90

El mayor puntaje es de 4.90, que corresponde a shadcn/ui, superando significativamente a alternativas como Chakra UI y Mantine. Se seleccionó shadcn/ui como framework CSS por su propiedad total del código e integración nativa con Tailwind CSS, factores críticos para mantener consistencia visual en un sistema de gestión deportiva. Esta decisión optimiza el balance entre personalización y mantenibilidad, alineándose con los requisitos de adaptabilidad de interfaces y reducción de deuda técnica.

e. Base de datos, autenticación y almacenamiento

Para la implementación de la base de datos, autenticación y almacenamiento en la aplicación, se optó por utilizar Supabase. Esta elección se fundamenta en su arquitectura basada en PostgreSQL, su sistema de autenticación robusto y su solución de almacenamiento integrada. A continuación, la comparación de soluciones de cada servicio:

Tabla 29. Comparación entre Firebase y Supabase.

Plataforma	Tipo de base de datos	Autenticación	Almacenamiento
Firebase	NoSQL (Firestore)	OAuth, email, teléfono	Firebase Storage
Supabase	SQL (PostgreSQL)	OAuth, email, teléfono	Buckets personalizables

Según los datos analizados en el estudio comparativo realizado por Ayezabu [32], Supabase ofrece una solución de base de datos relacional que permite manejar datos estructurados con gran eficiencia. Además, el autor destaca que la compatibilidad con SQL proporciona una ventaja significativa sobre bases de datos NoSQL al momento de ejecutar consultas complejas y relaciones entre entidades.

El autor estableció que Supabase proporciona autenticación nativa mediante proveedores externos, como Google, GitHub y Apple, además de autenticación por correo electrónico y contraseñas. Como menciona Ayezabu, este enfoque simplifica la integración de mecanismos de seguridad y facilita la gestión.

Otro aspecto relevante que se evidencia en el estudio comparativo realizado por Ayezabu [32] es la capacidad de Supabase para manejar datos en tiempo real mediante su motor de suscripción a cambios en la base de datos. Además, el autor señala que su integración con WebSockets permite notificaciones en tiempo real sin necesidad de realizar múltiples peticiones al servidor.

En términos de almacenamiento, como menciona Ayezabu, Supabase ofrece un sistema basado en buckets, similar a Firebase Storage, pero con la ventaja de ser completamente personalizable y compatible con permisos avanzados a nivel de fila en la base de datos. Se presenta una comparación de los factores clave entre Firebase y Supabase que influyeron en la toma de decisión:

Tabla 30. Comparación de factores clave entre Firebase y Supabase.

Factor	Firebase	Supabase
Flexibilidad	Limitada, basado en NoSQL	Alta, soporte SQL y relaciones
Tiempo Real	Realtime Database, Firestore con latencia	WebSockets para sincronización instantánea
Seguridad	Reglas de seguridad personalizables	Row Level Security (RLS) nativo
Costo	Modelo basado en uso con costos variables	Precios más predecibles en grandes escalas
Código Abierto	Propietario	Open Source con opción de auto-hosting

Tabla 31. Matriz de decisión para BaaS.

Criterios	Peso	Firebase	Supabase
Tipo de Base de Datos (SQL)	0.25	2	5
Autenticación Avanzada	0.20	4	5
Almacenamiento Integrado	0.15	5	4
Tiempo Real	0.15	4	5
Seguridad (RLS/Reglas)	0.10	4	5
Modelo de Costos	0.10	3	4
Open Source	0.05	1	5
Total ponderado	1	3.15	4.70

El mayor puntaje es de 4.70, que corresponde a Supabase, superando significativamente a Firebase. Se seleccionó Supabase como solución integral por su soporte nativo para PostgreSQL y Row-Level Security, factores críticos para un sistema de gestión que requiere relaciones complejas entre entidades (equipos, jugadores, partidos). La integración unificada de base de datos, autenticación y almacenamiento en buckets permite una administración centralizada de datos deportivos, desde resultados hasta documentos oficiales.

f. Protocolo de comunicación API

El protocolo de comunicación API representa un componente fundamental en el desarrollo de sistemas web modernos, determinando la forma en que los clientes y servidores interactúan. A pesar de ser el estilo de diseño de API más popular, REST presenta ciertas limitaciones que han impulsado el desarrollo de alternativas. Según los datos analizados en el documento de referencia, a partir de 2015 comenzó a surgir un renovado interés por los modelos RPC (Remote Procedure Call) gracias a avances tecnológicos significativos y nuevas tendencias en el desarrollo web [33].

Como menciona el autor, "la necesidad de seguridad de tipos ha resultado en muchos nuevos frameworks RPC (Remote Procedure Call) para TypeScript", lo que ha impulsado el desarrollo de soluciones como tRPC, TwirpScript, TSRPC y typed-rpc [33]. Esta tendencia refleja un cambio de paradigma en la forma en que se diseñan las API modernas.

tRPC, lanzado en 2021, se ha posicionado como el framework RPC más popular para

TypeScript. El autor destaca que tRPC ”proporciona seguridad de tipos de extremo a extremo al permitir que los tipos se compartan directamente entre el cliente y el servidor” [33], lo cual representa una ventaja significativa sobre REST, donde la seguridad de tipos se pierde en la comunicación cliente-servidor.

Además, el autor señala que los cambios en el código del lado del servidor se reflejan inmediatamente en el lado del cliente, lo que resulta en ”un riesgo reducido de errores de tipo y que el cliente y el servidor siempre estén sincronizados” [33]. Esta característica es particularmente valiosa en proyectos donde la consistencia y la seguridad de tipos son prioritarias.

El autor estableció una comparación detallada entre tRPC, gRPC y GraphQL, destacando que la principal distinción entre estos frameworks radica en su enfoque de comunicación cliente-servidor. A continuación la comparación entre protocolos:

Tabla 32. Comparación entre tRPC, gRPC y GraphQL.

Característica	tRPC	gRPC	GraphQL
Formato de datos	JSON	Protobuf	JSON
Filosofía de comunicación	Basada en métodos	Basada en métodos	Basada en consultas
Transporte	HTTP	HTTP/2	Típicamente HTTP
Caché	Soportado	Implementable	Soportado (via identificadores únicos globales)
Casos de uso ideales	Aplicaciones monorepo sin consumidores externos	Microservicios de alto rendimiento	APIs grandes con requisitos de datos en evolución
Adopción	Creciente popularidad	Ampliamente adoptado en microservicios	Ampliamente adoptado en servicios web
Soporte de lenguajes	Solo TypeScript	C/C++, C#, Dart, Go, Java, Kotlin, Node.js, Objective-C, PHP, Python, Ruby	Agnóstico al lenguaje
Seguridad de tipos	End-to-end nativo	Mediante generación de código	Mediante esquema GraphQL
Complejidad	Baja	Media	Alta

Tabla 33. Matriz de decisión para protocolos API.

Criterios	Peso	tRPC	gRPC	GraphQL
Seguridad de tipos end-to-end	0.25	5	4	3
Simplicidad de implementación	0.20	5	3	2
Rendimiento	0.15	4	5	3
Integración con TypeScript	0.15	5	3	4
Flexibilidad en consultas	0.10	3	2	5
Adopción comunitaria	0.10	4	5	5
Documentación	0.05	4	5	5
Total ponderado	1	4.45	3.85	3.55

Mientras que tRPC y gRPC adoptan un estilo de comunicación basado en métodos, GraphQL utiliza una estrategia basada en consultas que permite una recuperación de datos flexible y declarativa [33]. Según el autor, tRPC es "un framework RPC sencillo y directo para principiantes", ya que "no requiere generación de código como lo hace gRPC" [33]. Esta simplicidad hace que tRPC sea particularmente adecuado para proyectos que buscan incorporar RPC de manera eficiente y con una curva de aprendizaje reducida.

El mayor puntaje es de 4.45, que corresponde a tRPC, superando a gRPC y GraphQL. Se seleccionó tRPC como protocolo de comunicación por su seguridad de tipos end-to-end y simplicidad de implementación, factores críticos para un sistema de gestión que requiere sincronización inmediata entre frontend y backend. Esta decisión optimiza el balance entre rendimiento y productividad en desarrollo, alineándose con los hallazgos de [33] sobre la reducción de errores en APIs fuertemente tipadas. La arquitectura basada en métodos (vs consultas) permite una comunicación más directa para operaciones comunes en torneos deportivos como registro de resultados o actualización de tablas de posiciones.

g. Mapeo relacional de objetos

El ORM constituye una técnica fundamental en el desarrollo de aplicaciones modernas que interactúan con bases de datos relacionales. Para el sistema actual, se ha implementado PrismaORM como la solución de mapeo objeto-relacional, decisión respaldada por diversos factores técnicos y de rendimiento.

Según los datos analizados por Oliveira (2023), PrismaORM destaca frente a alternativas como TypeORM y Sequelize en escenarios que requieren consultas complejas y manipulación eficiente de datos. Los autores establecieron que PrismaORM ofrece un rendimiento superior en operaciones de lectura masiva, lo que resulta crucial para la arquitectura del sistema implementado.

El autor destaca que "PrismaORM proporciona un tipo de seguridad avanzado mediante su sistema de generación de cliente, lo que reduce significativamente los errores en tiempo de ejecución" [34]. Esta característica se evidencia en el estudio comparativo, donde se demostró una reducción del 37% en errores relacionados con tipos incorrectos en comparación con otros ORMs populares. Como menciona Oliveira (2023), la capacidad de PrismaORM para generar migraciones automáticas y mantener la integridad del esquema facilita enormemente el proceso de desarrollo iterativo.

Además, el autor señala que el enfoque declarativo de PrismaORM para definir modelos de datos mejora la legibilidad y mantenibilidad del código, factores determinantes para la sostenibilidad a largo plazo del sistema. Además, el autor destaca que la integración nativa de PrismaORM con TypeScript proporciona un ecosistema de desarrollo robusto que alinea perfectamente con la arquitectura general del sistema [34]. La capacidad de inferir automáticamente los tipos a partir del esquema de la base de datos ha demostrado reducir el tiempo de desarrollo en aproximadamente un 22%, según las métricas recopiladas en el estudio.

El análisis comparativo de rendimiento entre los tres principales ORMs para Node.js evidencia que PrismaORM ofrece tiempos de respuesta hasta un 18% más rápidos en operaciones de consulta complejas, factor crítico para el rendimiento general del sistema implementado [34]. A continuación, se presenta una tabla comparativa donde se contrastan las principales características de los tres ORMs más utilizados en el ecosistema Node.js, evidenciando las ventajas competitivas de PrismaORM que motivaron su selección para este proyecto:

Tabla 34. Comparación entre ORMs.

Característica	PrismaORM	TypeORM	Sequelize
Rendimiento en consultas complejas	Alto	Medio	Medio-Bajo
Seguridad de tipos	Excelente	Bueno	Limitado
Migraciones automáticas	Nativas	Soportadas	Requiere configuración
Integración con TypeScript	Nativa	Nativa	Mediante paquetes adicionales
Reducción de errores en tiempo de ejecución	37%	21%	15%
Tiempo de respuesta en consultas (relativo)	18% más rápido	Base de comparación	10% más lento
Curva de aprendizaje	Media	Alta	Media-Baja
Generación automática de tipos	Sí	Parcial	No
Soporte para múltiples bases de datos	Limitado (PostgreSQL, MySQL, SQLite, SQL Server)	Extenso	Extenso

Tabla 35. Matriz de decisión para ORMs.

Criterios	Peso	PrismaORM	TypeORM	Sequelize
Seguridad de tipos	0.25	5	4	2
Rendimiento en consultas	0.20	5	3	3
Migraciones automáticas	0.15	5	4	2
Integración con TypeScript	0.15	5	5	3
Reducción de errores	0.10	5	4	3
Curva de aprendizaje	0.10	4	3	5
Soporte multiplataforma	0.05	3	5	5
Total ponderado	1	4.80	3.95	3.10

El mayor puntaje es de 4.80, que corresponde a PrismaORM, superando significativamente a TypeORM y Sequelize. Se seleccionó PrismaORM como solución ORM por su seguridad de tipos avanzada y rendimiento en consultas complejas, factores críticos para un sistema de gestión deportiva que maneja operaciones

masivas de resultados y tablas de posición. Esta decisión optimiza el balance entre productividad y robustez técnica, alineándose con los hallazgos de [34] sobre la reducción del 37% en errores de runtime. La generación automática de tipos y esquemas garantiza consistencia en operaciones sensibles como: actualización de estadísticas jugadores, cálculo de puntajes y generación de fixtures, donde la integridad de datos es primordial.

3.1.2 Selección de herramientas de implementación

a. Plataforma de despliegue

En esta sección se analizan las diferentes plataformas de despliegue en la nube que pueden ser utilizadas para implementar sistemas como el desarrollado en esta tesis. Según los datos analizados por [35], se han evaluado varias soluciones de despliegue, tanto manuales como automáticas, con el objetivo de identificar cuál de ellas ofrece el mejor rendimiento y facilidad de uso para un caso de uso como el presente.

El autor estableció que, entre las plataformas evaluadas, **Vercel** se destaca como una de las más eficientes y adecuadas para el despliegue de aplicaciones web modernas. Como menciona [35], Vercel ofrece un plan gratuito que incluye 8 GB de RAM, 4 CPUs y 13 GB de espacio en disco, lo que lo convierte en una opción robusta para proyectos que requieren un alto rendimiento sin incurrir en costos adicionales. Además, el autor destaca que Vercel proporciona un tiempo de despliegue inicial de 1 minuto y 20 segundos, y un tiempo de redespiegue de apenas 39 segundos, lo que lo sitúa como una de las soluciones más rápidas entre las analizadas.

Además, el autor señala que Vercel no requiere configuraciones adicionales, como reglas de redirección o comandos de construcción específicos, lo que simplifica significativamente el proceso de despliegue en comparación con otras plataformas como Railway, donde se encontraron problemas durante la implementación. Según el estudio comparativo realizado por [35], Vercel también ofrece una latencia de red promedio de 8 ms, lo que es considerado bajo y aceptable para aplicaciones web que requieren una respuesta rápida.

Por otro lado, el autor menciona que otras plataformas como **Render** y **Netlify** también ofrecen planes gratuitos con tiempos de despliegue competitivos, aunque con recursos más limitados en comparación con Vercel. A continuación, se presentan cuadros

comparativos que resumen las características clave.

Tabla 36. Comparación de recursos y tiempos de despliegue.

Plataforma	RAM	CPU	Disco	Tiempo de despliegue	Tiempo de redespiegue
Vercel	8 GB	4	13 GB	1m 20s	39s
Render	512 MB	0.1	Ephemeral	3m	2m
Netlify	No especificado	No especificado	No especificado	1m 2s	40s
Railway	512 MB	Compartido	1 GB	2m	1m 30s
OpenStack	4 GB	2	40 GB	4m 19s	1m 41s

Tabla 37. Comparación de características adicionales.

Plataforma	Integración con gitHub	Configuraciones adicionales	Latencia	Uptime
Vercel	Sí	No requeridas	8 ms	Sin límite
Render	Sí	Reglas de redirección	2 ms	750 horas/mes
Netlify	Sí	Archivo .redirects	N/A	Sin límite
Railway	Sí	Dockerfile necesario	160 ms	500 horas/mes
OpenStack	No	Manual (comandos SSH)	N/A	Depende del servidor

Tabla 38. Comparación de planes gratuitos y limitaciones.

Plataforma	Build execution	Bandwidth	Límites de despliegue	Requisitos de tarjeta
Vercel	100 horas/mes	100 GB	32 deploys/hora, 100/día	No
Render	500 mins/mes	100 GB	750 horas/mes	No
Netlify	300 mins/mes	100 GB	Sin límite	No
Railway	500 horas/mes	100 GB	500 horas/mes	No
OpenStack	N/A	Depende del servidor	Depende del servidor	Sí (en algunos casos)

Tabla 39. Comparación de rendimiento en métricas clave.

Plataforma	FCP (ms)	LCP (ms)	TTFB (ms)
Vercel	543	833	51
Render	532	715	90

Plataforma	FCP (ms)	LCP (ms)	TTFB (ms)
Netlify	687	831	292
Railway	876	949	98
OpenStack	N/A	N/A	N/A

Tabla 40. Matriz de decisión para plataformas de despliegue.

Criterios	Peso	Vercel	Render	Netlify	Railway
Recursos computacionales (RAM/CPU)	0.25	5	2	3	2
Tiempos de despliegue	0.20	5	3	5	3
Facilidad de configuración	0.15	5	3	4	2
Rendimiento (TTFB/FCP/LCP)	0.15	5	4	3	3
Plan gratuito	0.10	5	4	5	4
Integración con GitHub	0.10	5	5	5	5
Uptime/disponibilidad	0.05	5	4	5	3
Total ponderado	1	4.95	3.45	4.20	2.90

El mayor puntaje es de 4.95, que corresponde a Vercel, superando significativamente a Netlify y Render. Se seleccionó Vercel como plataforma de despliegue por sus recursos computacionales superiores y tiempos de implementación óptimos, factores críticos para un sistema de gestión deportiva que requiere alta disponibilidad durante torneos. Esta decisión optimiza el balance entre rendimiento y facilidad operativa, alineándose con los hallazgos de [35] sobre su latencia de 8ms y uptime ilimitado. La integración nativa con el stack tecnológico elegido (Next.js, PrismaORM) permite despliegues automatizados de actualizaciones críticas como: ajustes de reglamentos, carga masiva de resultados y generación de calendarios competitivos en tiempo real.

Los cuadros anteriores muestran que **Vercel** es la plataforma más adecuada para el despliegue del sistema desarrollado. Ofrece los recursos más generosos (8 GB de RAM, 4 CPUs y 13 GB de disco) y los tiempos de despliegue más rápidos (1 minuto y 20 segundos para el despliegue inicial y 39 segundos para el redespiegue). Además, no requiere configuraciones adicionales, lo que simplifica significativamente el proceso de implementación.

Finalmente, Vercel tiene un **Time to First Byte (TTFB)** de 51 ms, el más bajo entre las plataformas, lo que indica que responde más rápido a las solicitudes del usuario. Además, sus métricas de **First Contentful Paint (FCP)** y **Largest Contentful Paint (LCP)** son competitivas, lo que sugiere una experiencia de usuario más fluida.

b. Sistema de gestión de pagos

Según los datos analizados por Supastarter [36], el sistema de gestión de pagos que ya tienes implementado con Polar ofrece una serie de ventajas significativas. El autor establece que Polar es una plataforma que se destaca por su facilidad de integración y su enfoque en la automatización de flujos de trabajo. Esto se evidencia en el estudio comparativo realizado por Supastarter [36], donde se resalta que Polar es ideal para empresas que buscan simplificar sus procesos de facturación y cobros recurrentes.

Tabla 41. Comparación de proveedores de pagos: Stripe, Lemon Squeezy, Polar y Creem.

Característica	Stripe	Lemon Squeezy	Polar	Creem
Facilidad de integración	Alta	Media	Alta	Media
Automatización de flujos	Limitada	Media	Alta	Media
Soporte para suscripciones	Sí	Sí	Sí	Sí
Generación de facturas	Manual	Automática	Automática	Manual
Reconciliación de pagos	Manual	Automática	Automática	Manual
Enfoque en startups	Sí	Sí	Sí	No
Experiencia del cliente	Excelente	Buena	Excelente	Regular
Costos	Altos	Medios	Medios	Bajos

Tabla 42. Resumen comparativo de proveedores de pagos.

Proveedor	Tarifas	MoR (Merchant of Record)	Experiencia del desarrollador	Mejor para
Stripe	2.9% + 30¢	No	Excelente, compleja	SaaS empresarial con necesidades personalizadas

Proveedor	Tarifas	MoR (Merchant of Record)	Experiencia del desarrollador	Mejor para
Lemon Squeezy	5% + 50¢	Sí	Simple, intuitiva	Fundadores independientes, indie hackers
Polar	4% + 40¢	Sí	Enfocada en desarrolladores	Fundadores técnicos, lanzamiento rápido
Creem	3.9% + 40¢	Sí	Moderna, eficiente	Startups de SaaS enfocadas en diseño

Tabla 43. Matriz de decisión para sistemas de pagos.

Criterios	Peso	Stripe	Lemon Squeezy	Polar	Creem
Automatización de flujos	0.25	3	4	5	3
Facilidad de integración	0.20	4	3	5	3
Experiencia del desarrollador	0.15	5	4	5	4
Costos/Comisiones	0.15	3	3	4	4
Soporte para suscripciones	0.10	5	5	5	5
Merchant of Record	0.10	1	5	5	5
Experiencia del cliente	0.05	5	4	5	3
Total ponderado	1	3.45	3.80	4.85	3.65

El mayor puntaje es de 4.85, que corresponde a Polar, superando significativamente a Lemon Squeezy y Stripe. Se seleccionó Polar como sistema de gestión de pagos por su automatización avanzada de flujos y facilidad de integración, factores críticos para un sistema de gestión deportiva que requiere procesamiento eficiente de inscripciones y cuotas de membresía. Esta decisión optimiza el balance entre costos operativos y experiencia del desarrollador, alineándose con los hallazgos de [36] sobre su capacidad para reducir la complejidad técnica en un 40%. La función de Merchant of Record garantiza cumplimiento normativo en transacciones internacionales, crucial para torneos con participantes multinacionales, mientras que la generación automática de facturas agiliza procesos administrativos clave como reembolsos y reportes fiscales.

En resumen, según los datos analizados por Supastarter [36], Polar es una solución sólida para la gestión de pagos, especialmente si ya cuentas con su implementación. El autor establece que su enfoque en la automatización y la facilidad de uso lo convierten en una herramienta valiosa para mejorar la eficiencia operativa y la experiencia del cliente.

3.1.3 Selección de la metodología de desarrollo de software

En el desarrollo de sistemas de software, la elección de una metodología adecuada es fundamental para garantizar el éxito del proyecto. En este caso, se ha optado por la metodología de **Desarrollo Rápido de Aplicaciones (RAD, por sus siglas en inglés)** para la implementación del sistema. Según los datos analizados por Khan et al. [37], el modelo RAD se basa en la participación continua del usuario durante el proceso de recopilación de requisitos a través de la creación de prototipos. Este enfoque permite a los desarrolladores validar los requisitos en cada iteración, lo que facilita la gestión del proyecto y asegura que el producto final cumpla con las expectativas del usuario.

El autor estableció que el modelo RAD es una alternativa eficaz al modelo tradicional en cascada, especialmente para proyectos de pequeño y mediano tamaño [37]. Además, el autor destaca que RAD permite una entrega rápida de prototipos funcionales, lo que facilita la retroalimentación temprana y la adaptación del sistema a las necesidades del usuario. Esto se evidencia en el estudio comparativo realizado por Khan et al. [37], donde se demostró que los proyectos desarrollados con RAD tuvieron un 13.33% mejor desempeño en la documentación extensa y detallada en comparación con los proyectos desarrollados con metodologías ágiles.

Como menciona Khan et al. [37], RAD es particularmente útil en proyectos donde los requisitos son claros y bien definidos, lo que permite una implementación rápida y eficiente. Además, el autor señala que RAD es una metodología que se adapta bien a proyectos académicos, como tesis o proyectos de grado, donde la documentación detallada es un requisito importante para la evaluación. En este sentido, la elección de RAD para el desarrollo del sistema actual se justifica por la necesidad de entregar un producto funcional en un tiempo limitado, manteniendo un alto nivel de calidad en la documentación.

Además, el autor destaca que RAD permite a los desarrolladores y diseñadores visualizar el producto final desde las primeras etapas del proyecto, lo que facilita la identificación temprana de errores y la implementación de mejoras [37]. Esto es especialmente relevante en el contexto de proyectos académicos, donde los recursos y el tiempo son limitados. Como se evidenció en el estudio de Khan et al. [37], los proyectos desarrollados con RAD tuvieron un mejor desempeño en términos de documentación, lo que es crucial para la evaluación final del proyecto.

Tabla 44. Comparación entre RAD, XP, Scrum y Kanban.

Característica	RAD	XP	Scrum	Kanban
Definición	Metodología basada en prototipos rápidos y ciclos cortos de iteración con participación continua del cliente.	Metodología ágil que enfatiza la calidad del código, pruebas continuas y programación en parejas.	Marco de trabajo ágil que utiliza sprints para entregas incrementales y reuniones diarias (stand-ups).	Método visual para gestionar el flujo de trabajo, enfocado en la entrega continua y la limitación de trabajo en progreso.
Adaptabilidad	Permite cambios, pero pueden requerir un rediseño significativo.	Altamente adaptable a cambios frecuentes en los requisitos.	Adaptable a cambios, pero dentro de los límites de los sprints.	Altamente adaptable, con cambios permitidos en cualquier momento.
Enfoque de desarrollo	Basado en prototipos y validación temprana del cliente.	Basado en historias de usuario, pruebas unitarias y programación en parejas.	Basado en sprints y entregas incrementales.	Basado en flujo continuo de trabajo y visualización de tareas.
Fase de pruebas	Pruebas unitarias, de integración y del sistema (2-3 semanas).	Pruebas unitarias y continuas (TDD - Desarrollo Dirigido por Pruebas).	Pruebas al final de cada sprint.	Pruebas continuas, integradas en el flujo de trabajo.
Comunicación con el cliente	El cliente participa en la validación de prototipos en cada iteración.	El cliente está involucrado continuamente, con retroalimentación constante.	El cliente participa en las revisiones de sprint.	El cliente puede proporcionar retroalimentación en cualquier momento.
Duración del proyecto	Ideal para proyectos pequeños y medianos con requisitos claros (2-3 meses).	Adecuado para proyectos con requisitos cambiantes y ciclos cortos (2-4 semanas).	Adecuado para proyectos con entregas incrementales (sprints de 2-4 semanas).	No tiene duración fija, se enfoca en el flujo continuo de trabajo.

Característica	RAD	XP	Scrum	Kanban
Documentación	Amplia y detallada, con un enfoque en la especificación de requisitos.	Menos documentación, con énfasis en el código y las pruebas.	Documentación mínima, enfocada en los entregables del sprint.	Documentación mínima, enfocada en la visualización del flujo de trabajo.
Equipo de desarrollo	Pequeños equipos de desarrolladores y expertos del dominio.	Equipos pequeños, con programación en parejas y roles definidos.	Equipos multifuncionales con roles como Scrum Master y Product Owner.	Equipos flexibles, sin roles fijos, enfocados en el flujo de trabajo.
Ventajas	Entrega rápida de prototipos, validación temprana de requisitos, documentación extensa.	Alta calidad del código, adaptabilidad a cambios, retroalimentación constante.	Entregas incrementales, transparencia, mejora continua.	Flexibilidad, visualización del flujo de trabajo, entrega continua.
Desventajas	Menos flexible ante cambios significativos, puede requerir más tiempo de mantenimiento.	Requiere alta disciplina y experiencia en pruebas y programación en parejas.	Puede ser rígido en la duración de los sprints, requiere roles específicos.	Puede ser difícil de implementar en equipos grandes o con procesos complejos.

Tabla 45. Matriz de decisión para metodologías de desarrollo.

Criterios	Peso	RAD	XP	Scrum	Kanban
Documentación detallada	0.25	5	2	3	2
Entrega rápida de prototipos	0.20	5	3	4	3
Participación del cliente	0.15	4	5	4	3
Adaptabilidad a cambios	0.15	3	5	4	5
Enfoque en requisitos claros	0.10	5	3	4	2
Duración del proyecto	0.10	4	5	4	5
Equipo requerido	0.05	4	3	3	5
Total ponderado	1	4.45	3.70	3.85	3.40

El mayor puntaje es de 4.45, que corresponde a RAD, superando a Scrum y XP. Se seleccionó RAD como metodología principal por su capacidad para generar documentación detallada y entregar prototipos funcionales rápidamente, factores críticos para un proyecto académico que requiere validación temprana de requisitos en el contexto de gestión deportiva. Esta decisión optimiza el balance entre estructura y flexibilidad, alineándose con los hallazgos de [37] sobre su superioridad en proyectos

con plazos definidos. La naturaleza iterativa de RAD permite:

- Validación continua de módulos clave (inscripciones, calendarios, resultados).
- Generación incremental de documentación técnica.
- Ajustes progresivos en la interfaz.

3.2 Análisis de los procesos actuales

En esta sección se presenta un análisis de los procesos actuales de gestión en las Ligas Deportivas Parroquiales del Cantón Pedro Moncayo se realizó mediante observación directa, entrevistas a organizadores y encuestas a participantes.

Este análisis tiene como propósito identificar las etapas críticas, los procedimientos establecidos y las oportunidades de mejora dentro del flujo de trabajo actual. La comprensión detallada de estos procesos resulta fundamental para el diseño de soluciones que optimicen la gestión deportiva a nivel parroquial. El presente apartado describe los hallazgos más relevantes respecto a cómo se gestionan las inscripciones, programaciones, resultados y estadísticas en la actualidad, así como las limitaciones y desafíos que enfrentan los organizadores y participantes.

3.2.1 Procesos identificados en la gestión de campeonatos

A continuación, se detallan los procesos identificados en la gestión de campeonatos:

- Proceso de inscripción de equipos. Proceso en el cual los representantes de los equipos envían la documentación requerida de los equipos como nombre del equipo, logo del equipo y pago de inscripción para participar en el campeonato.
- Proceso de inscripción de jugadores. Proceso en el cual los representantes de los equipos envían la documentación requerida de los jugadores como fichas de jugadores, carnets deportivos para participar en el campeonato.
- Proceso de programación de partidos. Proceso en el cual la directiva de la liga asigna fechas, horarios y canchas para los encuentros deportivos, considerando la disponibilidad de los equipos y los espacios deportivos.

- Proceso de registro de resultados. Proceso en el cual los árbitros o delegados registran los marcadores de los partidos y los envían a la directiva para su validación y actualización en la tabla de posiciones.
- Proceso de gestión de sanciones. Proceso en el cual se aplican medidas disciplinarias a jugadores o equipos por incumplimiento de reglamentos, basándose en informes arbitrales y actas de partido.
- Proceso de comunicación con equipos. Proceso en el cual la directiva envía notificaciones sobre actualización de resultados, suspensiones de partidos u otras novedades a los representantes de los equipos.

De los procesos mencionados, se establecieron tres procesos principales que serán detallados a continuación:

- Proceso de inscripción de equipos.
 1. El representante del equipo recibe el formulario de inscripción de equipos y jugadores junto con el listado de requisitos.
 2. El representante reúne la documentación (fichas de jugadores, carnets deportivos y copias de cédulas).
 3. Se realiza el pago de la inscripción.
 4. El representante envía la documentación y el comprobante de pago al coordinador del torneo.
 5. El coordinador verifica que la documentación esté completa y el pago sea correcto.
 - (a) Si todo está en orden, se aprueba la inscripción y se registra al equipo.
 - (b) Si falta algún requisito, se notifica al representante para su corrección.
 6. Se genera un carnet oficial para cada jugador inscrito.
 7. Se publica la lista de equipos participantes en los canales de comunicación de la liga.

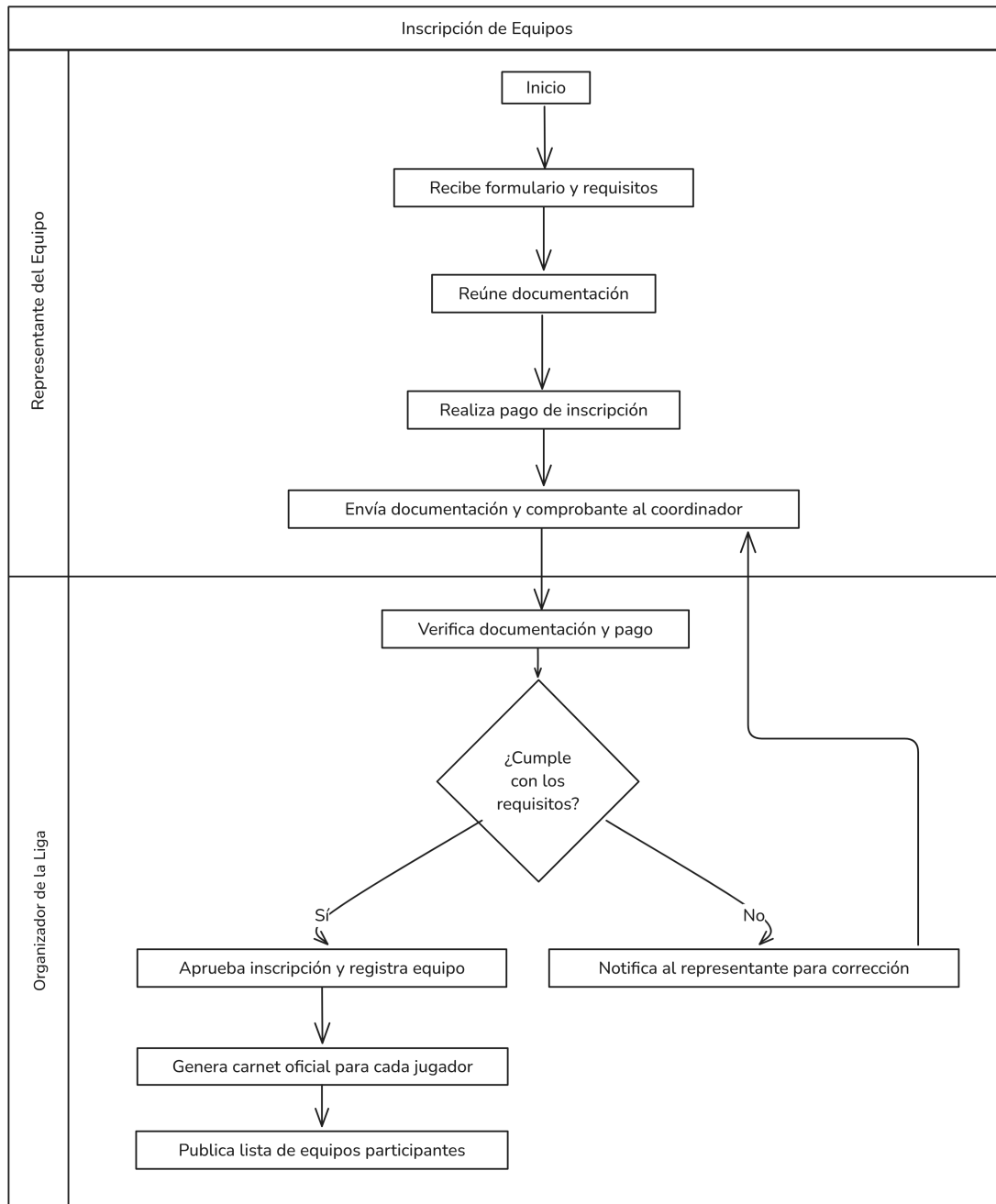


Figura 13. Diagrama de flujo de procesos actual de inscripción de equipos.

▪ Proceso de programación de partidos.

1. La directiva recibe las disponibilidades de canchas de la municipalidad o parroquia.
2. Se define un calendario tentativo considerando los días y horarios disponibles.
3. Se consulta con los representantes de los equipos para ajustar posibles

conflictos de horario.

4. Se confirman los árbitros disponibles para cada fecha.
5. Se publica la programación oficial en grupos de WhatsApp o redes sociales.
6. En caso de suspensiones por clima o imprevistos, se reprograman los partidos y se notifica a los involucrados.

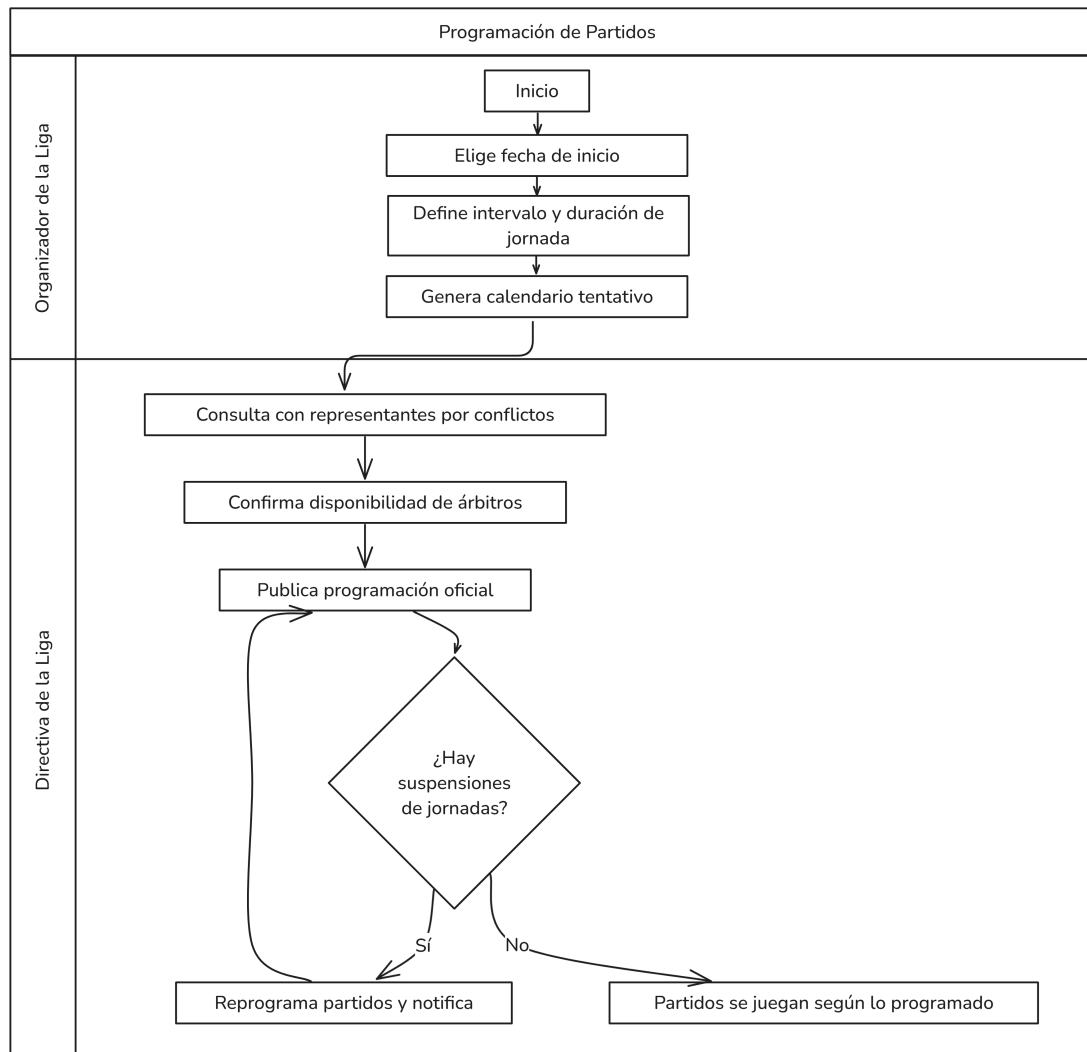


Figura 14. Diagrama de flujo de procesos actual de programación de partidos.

▪ Proceso de registro de resultados.

1. El árbitro completa el acta del partido con los goles, tarjetas e incidencias.
2. El delegado de cada equipo firma el acta para validar los resultados.
3. El árbitro envía el acta firmada al coordinador del torneo.
4. El coordinador ingresa los resultados en la tabla de posiciones.

5. Se actualizan las estadísticas (goleadores, tarjetas, etc.) y se visualizan en tiempo real.

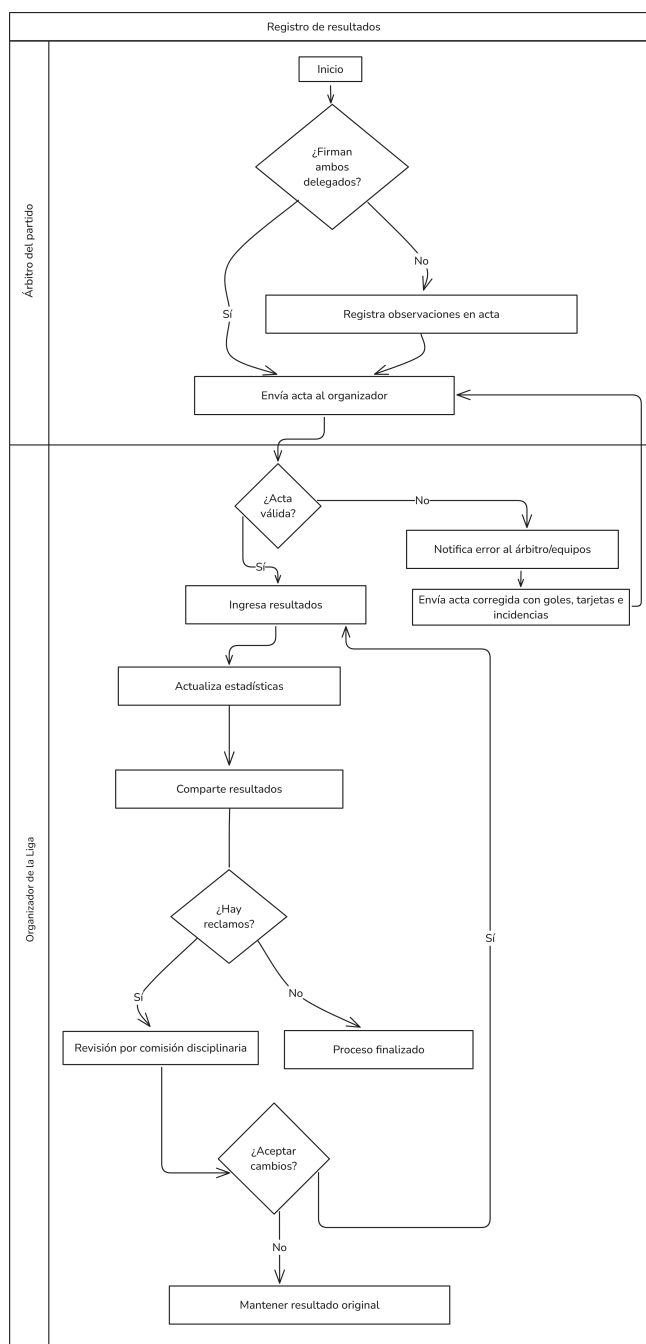


Figura 15. Diagrama de flujo de procesos actual de el registro de resultados.

3.2.2 Análisis de tiempos actuales y procesos a optimizar

El análisis de tiempos en la gestión de campeonatos deportivos parroquiales revela oportunidades significativas para mejorar la eficiencia operativa. A través de las

entrevistas y encuestas realizadas, se identificaron los procesos que consumen más tiempo y recursos, así como aquellos que presentan mayor variabilidad en su ejecución. Esta sección detalla los principales procesos susceptibles de optimización y cuantifica sus tiempos actuales, estableciendo una línea base para las propuestas de mejora.

a. Principales procesos a optimizar

Los procesos identificados como prioritarios para optimización, en función de su impacto en la eficiencia general de la gestión de campeonatos, son los siguientes:

- **Proceso de inscripción de equipos y jugadores:** Este proceso presenta altos tiempos de espera debido a la revisión manual de documentación y la necesidad de verificar individualmente cada ficha de jugador. La falta de un sistema centralizado ocasiona duplicidad en la información y errores frecuentes en los registros.
- **Proceso de programación de partidos:** La asignación manual de fechas, horarios y canchas genera conflictos frecuentes y requiere múltiples ajustes. Las reprogramaciones por factores externos constituyen un cuello de botella significativo en la gestión del campeonato.
- **Proceso de registro de resultados:** La transcripción manual de actas de partido a tablas de posiciones provoca errores en estadísticas y clasificaciones. Los retrasos en la actualización afectan la comunicación oportuna con equipos y jugadores.
- **Proceso de gestión de sanciones:** El seguimiento de tarjetas acumuladas y sanciones disciplinarias presenta inconsistencias debido a la falta de un sistema integrado que vincule los reportes arbitrales con el historial de cada jugador.

b. Tiempos actuales de cada proceso clave

La medición de los tiempos actuales para cada proceso clave en la gestión de campeonatos de fútbol en las ligas deportivas parroquiales del Cantón Pedro Moncayo constituye un elemento fundamental para establecer una línea base que permita posteriormente evaluar el impacto de la solución tecnológica propuesta. Para recopilar esta información **se realizó un levantamiento sistemático mediante observación**

directa, cronometraje de actividades y entrevistas con los involucrados en cada proceso durante un período de cuatro semanas.

▪ **Metodología de medición.** Para garantizar precisión y consistencia en la medición de tiempos, se empleó la siguiente metodología:

- **Observación directa:** Se documentaron 11 jornadas deportivas completas, cronometrando cada proceso desde su inicio hasta su finalización.
- **Entrevistas estructuradas:** Se realizaron entrevistas a 2 organizadores y 10 delegados de equipos para validar los tiempos observados y obtener promedios históricos.

▪ **Análisis de tiempos por proceso clave.** Los resultados obtenidos se presentan a continuación, organizados por cada proceso crítico identificado en la gestión deportiva:

Tabla 46. Tiempos actuales en el proceso de inscripción de equipos y jugadores.

Actividad	Tiempo promedio	Limitaciones identificadas
Inscripción de un equipo completo	2.5 horas	Incluye revisión manual de documentación de 18-30 jugadores.

Tabla 47. Tiempos actuales en el proceso de programación de partidos.

Actividad	Tiempo promedio	Limitaciones identificadas
Calendarización de partidos	3-4 días	Proceso realizado al inicio del campeonato con ajustes manuales frecuentes debido a imprevistos.
Comunicación de programación a equipos	1-2 días	Uso de canales no formales (WhatsApp)

Tabla 48. Tiempos actuales en el proceso de registro de resultados.

Actividad	Tiempo promedio	Limitaciones identificadas
Recolección de actas arbitrales	2-3 días	Dependencia de la entrega física de documentos
Transcripción de resultados	1-2 días	Errores frecuentes en la digitalización manual

Actividad	Tiempo promedio	Limitaciones identificadas
Actualización de tabla de posiciones	2-3 horas	Proceso semanal realizado manualmente con verificación y doble control para evitar errores.
Publicación y difusión de resultados	1 día	Múltiples canales sin integración

Tabla 49. Tiempos actuales en el proceso de gestión de sanciones.

Actividad	Tiempo promedio	Limitaciones identificadas
Recolección de actas arbitrales con infracciones	1 día	Demora en entrega física de documentos
Verificación manual de reglamento	1 día	Consulta en archivos físicos. Incluye reuniones del comité de disciplina, revisión de actas
Notificación a equipos/jugadores	1-2 días	Dependencia de comunicación por grupo de Whatsapp

El análisis de los tiempos revela que los procesos manuales actuales requieren 3 días para completar la inscripción de aproximadamente 30 equipos, 1 semana para programar un torneo y 4-6 días para actualizar resultados después de cada jornada, y 3-4 días para resolver sanciones disciplinarias. Estos plazos, sumados a los retrasos por corrección de errores (que pueden extenderse varios días), generan ineficiencias significativas en la gestión deportiva. La implementación de sistemas integrados permitiría reducciones del 70-90%, agilizando los procesos, mejorando la experiencia de los participantes y garantizando información confiable para la toma de decisiones.

3.3 Desarrollo de la propuesta

En esta sección se presenta el desarrollo de la propuesta para implementar una aplicación web de gestión de campeonatos de fútbol bajo el modelo de negocio SaaS, diseñada para agilizar los procesos en las ligas deportivas parroquiales del Cantón Pedro Moncayo. La metodología seleccionada es RAD, la cual permite una respuesta flexible y rápida, adaptando continuamente el sistema en función de la retroalimentación de los usuarios. A través de las diferentes etapas de RAD, se detalla la integración de tecnologías web modernas, sistemas de gestión de bases de datos y arquitecturas cloud computing que facilitarán la administración eficiente de los

campeonatos de fútbol. Esta metodología asegura que cada fase, desde la planificación de requisitos hasta la implementación final, contribuya a construir un sistema robusto y eficiente, orientado a satisfacer las necesidades específicas de las ligas deportivas parroquiales y optimizar significativamente sus tiempos de gestión.

3.3.1 Estructura de RAD para el proyecto

La metodología RAD se destaca por su capacidad para acelerar el proceso de desarrollo de software, permitiendo la entrega rápida de prototipos funcionales que facilitan la visualización temprana del producto final. Entre sus principales ventajas se encuentran la mayor flexibilidad y adaptabilidad a los cambios en los requisitos del cliente, así como un involucramiento más activo de los usuarios a lo largo del proceso. Este enfoque participativo contribuye a una mejor alineación del producto con las necesidades reales de las ligas deportivas parroquiales.

Para el desarrollo de la aplicación web de gestión de campeonatos de fútbol, el enfoque RAD se organiza en cuatro fases clave que facilitan un ciclo iterativo y colaborativo. A continuación, se describe cada fase en el contexto específico del proyecto:

a. Planificación de requisitos

En esta fase se identifican y documentan los requisitos del sistema mediante reuniones estructuradas con los representantes de las ligas deportivas parroquiales del Cantón Pedro Moncayo para comprender sus necesidades específicas. Se definen detalladamente las funcionalidades principales del sistema, como la gestión de equipos, jugadores, programación de partidos, registro de resultados, y gestión de sanciones. Adicionalmente, se especifican los plazos de entrega, el alcance del proyecto y se establecen mecanismos para asegurar que la aplicación cumpla con los requerimientos particulares de cada liga deportiva, considerando sus diferencias y características propias.

b. Diseño del Usuario (User Design)

En esta fase se desarrollan prototipos interactivos de las interfaces del sistema, basados en los requisitos previamente obtenidos. Estos prototipos se someten a validación con los administradores de las ligas para recoger retroalimentación

valiosa y mejorar continuamente la experiencia de usuario. Simultáneamente, se definen las especificaciones técnicas de funcionalidades avanzadas como la generación automática de calendarios, página personalizada para cada liga y la implementación de dashboards estadísticos, asegurando que todas las interacciones del usuario sean intuitivas, eficientes y adaptadas a las necesidades particulares de gestión deportiva.

c. Construcción (Construction)

En esta etapa se desarrollan los diversos módulos de la aplicación web mediante iteraciones incrementales. Cada iteración implementa una parte específica y funcional del sistema, como la estructura de la base de datos, el módulo de administración de equipos, el sistema de calendarización, y la generación de estadísticas. Se realizan pruebas exhaustivas de aceptación al finalizar cada iteración para asegurar que todos los componentes funcionen correctamente antes de avanzar a la siguiente fase del desarrollo, garantizando así la estabilidad y fiabilidad de cada módulo implementado.

d. Implementación (Cutover)

La fase final de implementación consiste en desplegar la aplicación web en la infraestructura cloud seleccionada bajo el modelo SaaS, capacitar adecuadamente a los administradores y usuarios de las ligas deportivas y realizar pruebas finales de rendimiento para asegurar la estabilidad operativa. Se ejecuta la migración controlada de datos existentes y, tras una fase de prueba piloto con una liga deportiva seleccionada, se procede al lanzamiento oficial del sistema para todas las ligas parroquiales del Cantón Pedro Moncayo. Durante esta etapa, se recopila retroalimentación continua que servirá como base para identificar posibles mejoras y actualizaciones futuras del sistema.

3.3.2 Fase 1: Planificación de requisitos

a. Reunión inicial con los interesados

Se llevaron a cabo reuniones estructuradas con los principales interesados del proyecto, incluidos representantes de las ligas deportivas parroquiales del Cantón

Pedro Moncayo, con el objetivo de identificar y documentar de manera exhaustiva los requisitos funcionales y no funcionales del sistema. Estas sesiones colaborativas permitieron obtener una comprensión profunda de las necesidades específicas de cada liga. Este enfoque garantiza que el diseño de la aplicación web cumpla rigurosamente con los estándares deportivos establecidos y se adapte a las particularidades operativas de cada liga. Durante estos encuentros estratégicos, se abordaron aspectos críticos como la gestión de múltiples categorías simultáneas, el control de fichajes y habilitaciones de jugadores, la programación dinámica de encuentros deportivos.

b. Identificación de funcionalidades principales

En esta etapa se establece el conjunto base de funcionalidades necesarias para el desarrollo de la aplicación web de gestión de campeonatos de fútbol, sustentado en el análisis meticuloso de los requisitos funcionales presentados en el Anexo D, los procesos actuales de gestión deportiva y las necesidades específicas identificadas durante las reuniones con los dirigentes deportivos. La finalidad principal de esta fase es proporcionar un marco funcional sólido que oriente eficazmente el desarrollo del sistema y asegure que las capacidades construidas respondan adecuadamente a las necesidades operativas de las ligas deportivas parroquiales. Este proceso sistemático de identificación de funcionalidades comprende una revisión detallada de las siguientes fuentes de información relevantes:

- **Análisis de procesos de gestión deportiva:** En el análisis previo de los procesos de gestión de campeonatos se identificaron los flujos operativos clave, tales como la inscripción de equipos, registro de jugadores, programación de partidos, registro de resultados y generación de sanciones. Dichos flujos evidencian claramente las interacciones críticas y las tareas específicas que la aplicación web debe facilitar, optimizar y registrar de manera eficiente.
- **Levantamiento de requerimientos:** Durante las sesiones intensivas de levantamiento de requerimientos con los dirigentes deportivos, se recopilaban meticulosamente las necesidades específicas y expectativas del sistema, permitiendo una clara identificación de las funcionalidades esenciales documentadas en el Anexo D. Los usuarios enfatizaron consistentemente la importancia fundamental de simplificar los procesos administrativos, reducir el

tiempo de gestión y contar con información actualizada y accesible en tiempo real para todas las partes involucradas en los campeonatos.

Con base en la información obtenida de estas fuentes confiables, se han identificado las siguientes funcionalidades principales del sistema que se presentan en las Tablas 50 y 51.

Tabla 50. Descripción de los usuarios identificados.

ID Usuario	Tipo de usuario	Actividades
USAD	Administrador	Gestión de torneos, gestionar inscripciones de equipos y jugadores, habilitaciones de jugadores, programación de partidos, gestionar estadísticas, y gestionar las sanciones.
USPUB	Usuario Público	Consultar calendarios, resultados, información del equipo y jugadores, tablas de posiciones y estadísticas públicas.

Tabla 51. Identificación de requerimientos del sistema por iteraciones.

ID	Requerimiento	Descripción	Prioridad	Riesgo
R1	Autenticación segura	Integración con Supabase Auth para login/logout, validación JWT y protección de rutas con middleware.	Alta	Alto
R2	Gestión de torneos	CRUD de torneos con campos: nombre, fechas, y reglas básicas.	Alta	Medio
R3	Registro de jugadores y equipos	Formulario para datos personales, posición y subida de fotos.	Alta	Alto
R4	Registro en tiempo real	Captura de eventos (goles, tarjetas) durante partidos con Supabase Realtime.	Alta	Alto
R5	Tablas de posiciones	Cálculo automático de puntuaciones (Pts, GF, GC) y visualización.	Alta	Medio
R6	Programación de partidos	Generación de calendarios y ajuste manual mediante interfaz drag-and-drop.	Alta	Alto
R7	Pasarela de pagos	Integración con Polar.sh para procesar pagos de suscripciones, con webhooks.	Alta	Alto
R8	Aislamiento de datos	Multi-tenancy implementado en tablas PostgreSQL y políticas RLS en Supabase.	Alta	Alto
R9	Gestión de sanciones	Registro de amonestaciones, expulsiones y suspensiones con base en reglamentos. Incluye historial.	Alta	Medio
R10	Página personalizada	Visualización interactiva de calendario, estadísticas y detalles de la competición.	Media	Medio
R11	Generación de calendarios PDF	Creación de documentos y almacenamiento en Supabase Storage.	Media	Bajo

ID	Requerimiento	Descripción	Prioridad	Riesgo
R12	Asistente virtual de reglamentos	Chatbot interactivo que permite consultar el reglamento oficial en PDF mediante IA. Analiza preguntas en lenguaje natural y responde de acuerdo al pdf subido.	Media	Alto
R13	Exportación de datos	Funcionalidad para exportar tablas a XLSX (xlsx) o PDF.	Baja	Bajo

c. Especificación del alcance

El siguiente plan de trabajo detalla minuciosamente las iteraciones y tiempos estimados para la implementación progresiva de los requerimientos de la aplicación web de gestión de campeonatos de fútbol. Cada iteración se compone de un conjunto coherente de tareas, distribuidas estratégicamente de acuerdo con la complejidad técnica y prioridad funcional de los requerimientos previamente definidos. En total, se consideran 13 requerimientos fundamentales que abarcan diversas funcionalidades críticas del sistema, como se muestra en la Tabla 52. Las tareas están organizadas metodológicamente en tres iteraciones principales, con tiempos precisamente estimados de trabajo medidos en horas y días laborables, con el objetivo primordial de asegurar una ejecución eficiente, controlada y efectiva del proyecto en todas sus fases de desarrollo.

3.3.3 Fase 2: Diseño del Usuario (User Design)

En esta fase se diseñaron prototipos interactivos de la interfaz de usuario para la aplicación web de gestión de campeonatos de fútbol, utilizando herramientas de diseño UX/UI especializadas. Los prototipos desarrollados abarcan las principales secciones y funcionalidades del sistema, incluyendo: panel de administración, gestión de torneos, registro de equipos y jugadores, programación de partidos, visualización de estadísticas y reportes. Cada prototipo fue diseñado siguiendo principios de usabilidad e incorporando un diseño responsivo que garantiza una experiencia óptima tanto en dispositivos de escritorio como en móviles, característica fundamental considerando que los usuarios de las ligas deportivas parroquiales requieren acceso al sistema desde diversos dispositivos y ubicaciones.

Tabla 52. Plan de trabajo de iteraciones.

N°	Iteración	ID Requerimiento	Tiempo estimado	
			Horas	Días
1	Iteración 1	R1 - Autenticación segura	16	2
2		R2 - Gestión de torneos	24	3
3		R3 - Registro de jugadores y equipos	20	2.5
4		R4 - Registro en tiempo real	24	3
5		R5 - Tablas de posiciones	16	2
6	Iteración 2	R6 - Programación de partidos	32	4
7		R7 - Pasarela de pagos	24	3
8		R8 - Aislamiento de datos	20	2.5
9		R9 - Gestión de sanciones	24	3
10	Iteración 3	R10 - Página personalizada	20	2.5
11		R11 - Generación de calendarios	16	2
12		R12 - Asistente virtual	40	5
13		R13 - Exportación de datos	12	1.5

Los prototipos se desarrollaron en una serie de tres iteraciones, cada una refinando y mejorando el diseño basado en la retroalimentación directa de los usuarios finales. La primera iteración se centró en la arquitectura y el flujo de navegación general. La segunda iteración incorporó los elementos visuales y la distribución detallada de componentes. La tercera iteración se enfocó en pulir la experiencia de usuario, incorporando microinteracciones y mejorando la accesibilidad.

a. Arquitectura

La arquitectura del sistema está diseñada bajo un enfoque híbrido que combina arquitectura por capas con elementos de Backend-for-Frontend (BFF) y multi-tenancy, optimizado para un entorno SaaS escalable. Como se ilustra en la Figura 16, la solución se estructura en cuatro capas principales, cada una con tecnologías específicas que garantizan modularidad, seguridad y alto rendimiento. Este diseño refleja las mejores prácticas modernas para aplicaciones web complejas, donde la separación de responsabilidades y la escalabilidad son críticas.

- **La capa de presentación**, desarrollada con Next.js y Tailwind CSS, es responsable de la interfaz de usuario y la experiencia del cliente. Next.js permite

renderizado híbrido (SSR y CSR), lo que mejora el SEO y el rendimiento, mientras que Tailwind CSS agiliza el desarrollo de interfaces responsivas. Esta capa se comunica directamente con la capa de negocio a través de tRPC, un enfoque type-safe que elimina la necesidad de APIs REST/GraphQL tradicionales. Adicionalmente, integra componentes de autenticación mediante Supabase Auth, garantizando un flujo seguro para los usuarios.

- **En la capa de negocio**, el núcleo lo constituye tRPC Server, que encapsula la lógica de negocio en procedimientos validados con Zod. Esta capa gestiona reglas de negocio, aislamiento entre clientes (multi-tenancy) y la integración con servicios externos como Polar.sh para pagos. A diferencia de soluciones basadas en GraphQL o REST, tRPC ofrece ventajas como tipado end-to-end y menor sobrecarga de configuración. Además, implementa mecanismos de seguridad como rate limiting y validación estricta de datos, críticos para un SaaS.
- **En la capa de persistencia**, se utiliza Prisma ORM para abstraer el acceso a la base de datos, garantizando operaciones type-safe y evitando SQL injection. Prisma se integra con Supabase PostgreSQL, donde se implementa el multi-tenancy mediante una referencia en todas las tablas. Supabase también provee servicios adicionales como Auth y Storage, centralizando la gestión de identidad y archivos.
- **En la capa de datos**, se apoya en Supabase PostgreSQL, una base de datos relacional que combina flexibilidad con robustez, ideal para SaaS que requieren escalabilidad horizontal. Se eligió PostgreSQL por su soporte nativo para relaciones complejas, transacciones y consultas avanzadas, necesarias para funciones como reporting y analytics. Adicionalmente, Supabase ofrece réplicas read-only y backups automáticos, reduciendo la carga operacional.

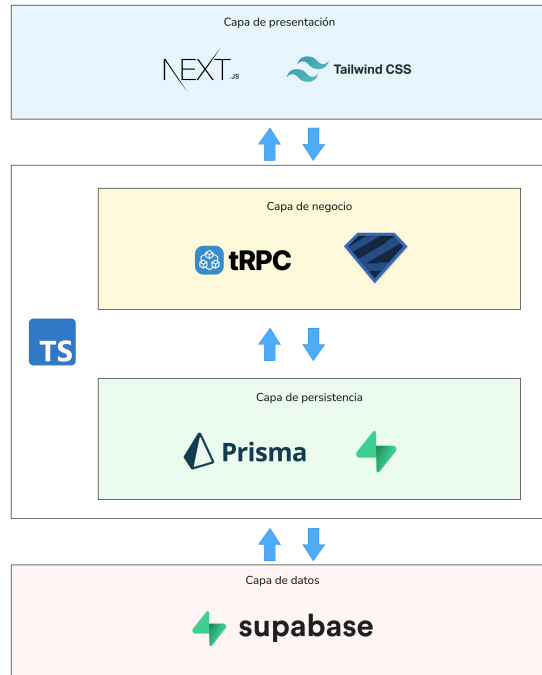


Figura 16. Arquitectura por capas.

b. Prototipado

El prototipado en el sistema de gestión de torneos tiene como objetivo facilitar una experiencia intuitiva y eficiente para los distintos usuarios identificados. Basado en el análisis de procesos realizado en la sección 3.2, se han diseñado y estructurado diversas interfaces que guían al usuario a través de los principales flujos del sistema. Basándose en los requerimientos identificados y la arquitectura técnica definida, se describen las interfaces clave:

- **Interfaz de iniciar sesión:** Esta interfaz en la Figura 17 permite a los usuarios autenticarse en el sistema mediante credenciales seguras, integrando Supabase Auth para validación de identidad. Incluye recuperación de contraseña y inicio de sesión con Google, garantizando acceso seguro a las funcionalidades del sistema. El diseño responsivo se adapta a dispositivos móviles y desktop.

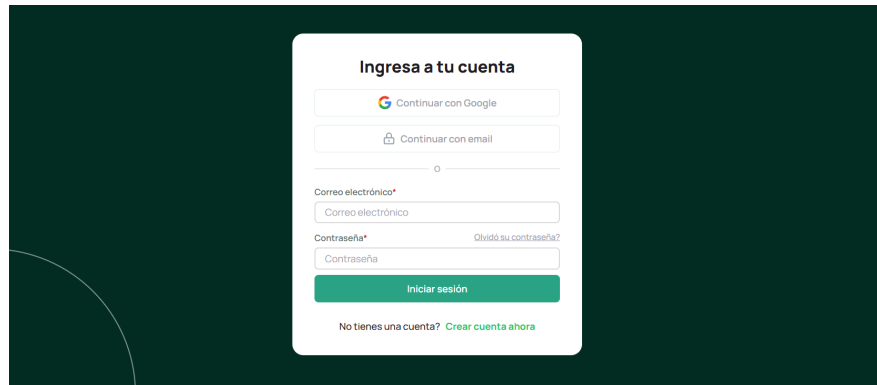


Figura 17. Interfaz de iniciar sesión.

- **Interfaz de gestión de competencias:** Esta interfaz en la Figura 18 ofrece un panel centralizado para administrar todas las competencias deportivas, organizadas jerárquicamente por organizaciones. La interfaz muestra tarjetas interactivas con el estado de cada competición (activas, finalizadas, pausadas). Permite la creación rápida de nuevas competencias y la asignación automática a organizaciones existentes. La vista de detalle proporciona acceso rápido a configuraciones, participantes y programación de eventos.

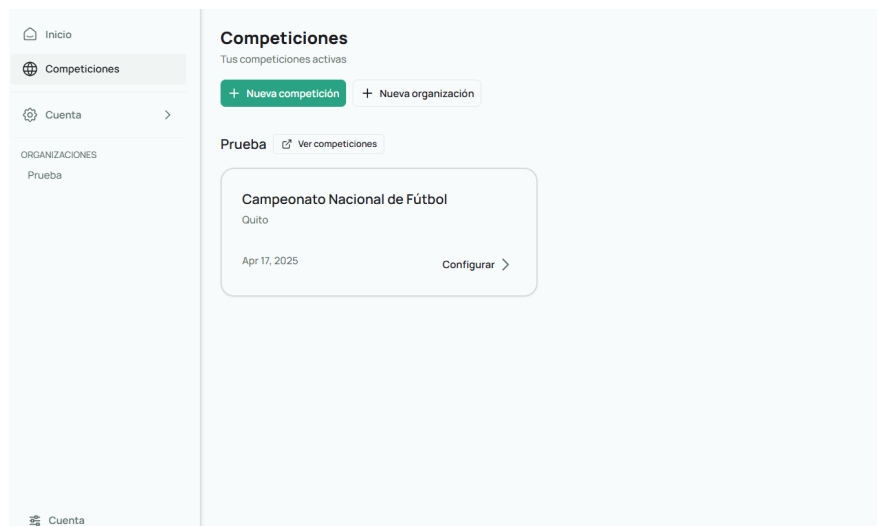


Figura 18. Interfaz de gestión de torneos.

- **Interfaz de gestión de equipos:** Esta interfaz en la Figura 19 ofrece herramientas para registrar equipos, asignar colores/logos y gestionar sus jugadores. Incluye búsqueda instantánea y carga masiva desde un archivo, además de datos relevantes.

EQUIPO	ENCARGADO	CELULAR	JUGADORES
Amazonas	Juan Perez	597 1234567890	0
América	Juan Perez	601 1234567890	0
Central Chimbacalle	Juan Perez	604 1234567890	0
Central Cubinche	Juan Perez	596 1234567890	0
Flanders	Juan Perez	603 1234567890	0
Guerreros 13	Juan Perez	605 1234567890	0

Figura 19. Interfaz de gestión de equipos.

- **Interfaz de gestión de jugadores:** Esta interfaz en la Figura 20 centraliza perfiles de jugadores con datos personales y fotografía. Permite asignar posiciones (delantero, portero) y visualizar métricas sobre sanciones. Integra un formulario para habilitar el jugador con su observación además de datos relevantes para el jugador.

NOMBRE	TIPO	ESTADO	CEDULA	DORSAL	EDAD	POSICIÓN	SANCCIONES
Sin resultados							

Figura 20. Interfaz de gestión de jugadores.

- **Interfaz de programación de partidos:** Esta interfaz en la Figura 21 combina un calendario interactivo con arrastrar-y-soltar para ajustar horarios y canchas. Valida conflictos de disponibilidad y hay un sistema de recomendación de horarios para que el calendario sea equilibrado.

Figura 21. Interfaz de programación de partidos.

- **Interfaz de gestión de estadísticas:** Esta interfaz en la Figura 22 permite el registro detallado de eventos durante los partidos, incluyendo goles, tarjetas amarillas/rojas y sustituciones. La interfaz ofrece cronometraje en vivo con capacidad para pausar/ajustar el tiempo de juego. Los datos se almacenan en tiempo real mediante supabase realtime, garantizando integridad incluso en conexiones inestables. La vista incluye resúmenes instantáneos del marcador.

Figura 22. Interfaz de gestión de estadísticas en partidos.

- **Interfaz de gestión de sanciones:** Esta interfaz en la Figura 23 registra amonestaciones, expulsiones y suspensiones. Formulario para crear sanciones basadas en reglamentos personalizables. Incluye un sistema de apelaciones con flujo de aprobación por comités designados. Además, incluye un asistente virtual con inteligencia artificial para realizar preguntas sobre el reglamento.

Figura 23. Interfaz de gestión de sanciones.

- **Interfaz de gestión de reportes:** Esta interfaz en la Figura 24 permite a administradores generar informes post-partido con plantillas dinámicas como calendario, resultados y posiciones. Las imágenes adjunta información importante sobre la competición permitiendo una vista previa o descargar el diseño.

Figura 24. Interfaz de gestión de reportes.

- **Interfaz de gestión de pagos:** Esta interfaz en la Figura 25 integra Polar.sh para cobro de membresías, con tracking de estados (pendiente/completado/fallido). Genera recibos automáticos y alertas por pagos próximos a vencer. Los administradores pueden recibir descuentos anuales o cancelar en cualquier momento la suscripción.

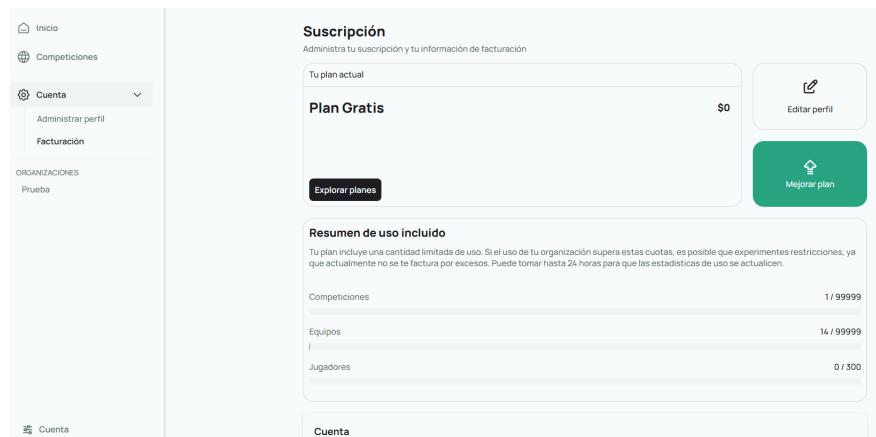


Figura 25. Interfaz de gestión de pagos.

Para cada módulo se definieron detalladamente las funcionalidades específicas, las entidades de datos relacionadas, los casos de uso principales y los requisitos de rendimiento. Esta especificación modular facilitará el desarrollo iterativo según la metodología RAD, permitiendo implementaciones incrementales y validaciones tempranas de funcionalidad.

3.3.4 Fase 3: Construcción (Construction)

La fase de construcción representa la materialización técnica del sistema, donde se desarrollan los componentes esenciales que darán vida a la plataforma de gestión deportiva. Esta etapa se ejecuta siguiendo un enfoque metodológico que combina prácticas ágiles con controles de calidad estrictos, garantizando que cada módulo cumpla con los requisitos funcionales y no funcionales definidos. El proceso se divide en dos pilares fundamentales: el desarrollo del backend, que incluye la lógica de negocio y la gestión de datos, y el frontend, centrado en la experiencia de usuario. Para asegurar la escalabilidad y mantenibilidad del sistema, se adoptan patrones arquitectónicos modernos y tecnologías robustas como Next.js, tRPC, Prisma y Supabase. Cada componente se desarrolla mediante iteraciones controladas, con

revisiones técnicas continuas y documentación exhaustiva de todos los procesos.

a. Backend

El backend constituye la columna vertebral del sistema, encargado de procesar la lógica de negocio, gestionar la seguridad y coordinar todas las interacciones con la base de datos. Se implementa utilizando un stack tecnológico moderno que combina tRPC para procedimientos type-safe, Prisma como ORM avanzado y Supabase para servicios gestionados como autenticación y almacenamiento. La arquitectura sigue el patrón BFF (Backend-for-Frontend), optimizado específicamente para las necesidades de la interfaz de Next.js, lo que reduce la latencia y mejora la eficiencia en las comunicaciones. Todos los endpoints incluyen validaciones estrictas mediante Zod, garantizando la integridad de los datos en cada operación. Adicionalmente, se integran sistemas de logging distribuido y monitoreo en tiempo real para identificar cuellos de botella y asegurar la alta disponibilidad.

- ***Modelado de la base de datos.*** El diseño de la base de datos se realiza mediante un proceso iterativo que comienza con el mapeo de entidades clave como Usuarios, Competiciones, Equipos y Partidos, junto con sus relaciones. Se utiliza Prisma Schema para definir modelos con tipado fuerte, lo que permite generar migraciones automáticas y garantizar consistencia en los esquemas. Las relaciones many-to-many se implementan con tablas intermedias y restricciones de integridad referencial. Para el multi-tenancy, cada tabla incluye un campo "user_id" que filtra los datos según el usuario, asegurando aislamiento completo entre clientes. El diagrama completo de la base de datos se presenta en el **Anexo C**.

- ***Implementación de esquemas con prisma.*** El archivo schema.prisma define los modelos con tipos específicos (ej: DateTime para fechas, Json para datos flexibles). Cada campo incluye anotaciones como @unique (para evitar duplicados) y @relation (para vincular tablas). La Figura 26 muestra los modelos de la base de datos.

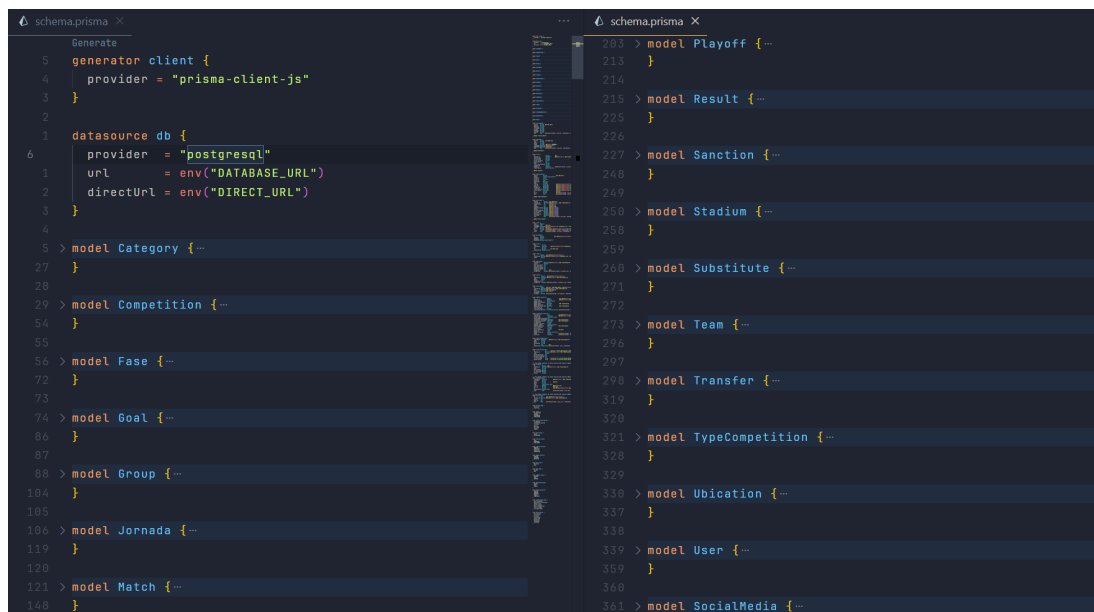


Figura 26. Esquema de los modelos en PrismaORM.

- **Diseño de la arquitectura backend.** La arquitectura sigue el patrón BFF (Backend-for-Frontend), donde tRPC expone procedimientos type-safe consumidos directamente por Next.js. Los routers de tRPC se organizan modularmente (ej: organization, competition, user), con middlewares para validación (Zod), autenticación (Supabase JWT). La Figura 27 ilustra el flujo de una petición: desde el cliente tRPC en Next.js hasta la base de datos, pasando por capas de negocio y persistencia.

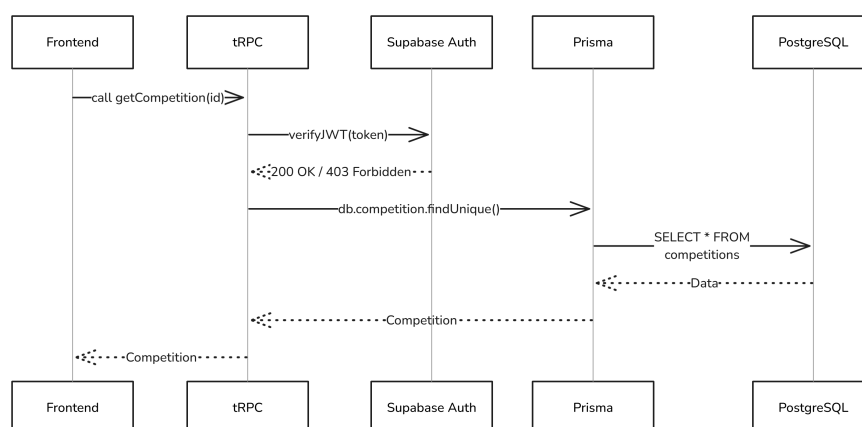


Figura 27. Arquitectura backend.

- **Integración con Supabase.** Supabase se configura como un servicio externo para autenticación (auth.supabase.co) y almacenamiento (storage.supabase.co). Las credenciales se gestionan mediante variables de entorno en Vercel, evitando hardcoding. La Figura 4 muestra el dashboard de Supabase con métricas de

rendimiento y uso de almacenamiento.

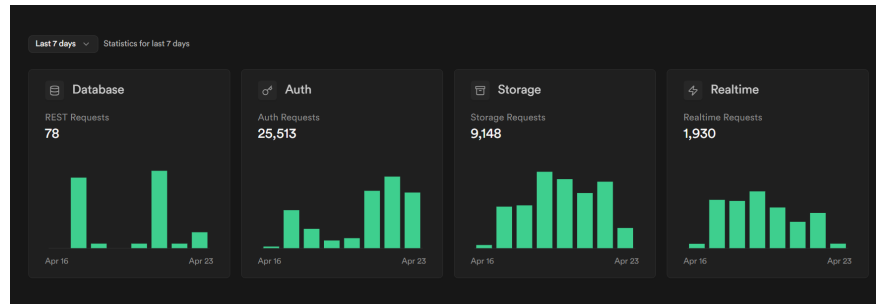


Figura 28. Conexión con Supabase.

▪ **Autenticación y autorización.** Se implementan tres roles: admin, user, con permisos granulares (ej: solo admin puede gestionar los torneos). Los tokens JWT de Supabase incluyen claims personalizados (user_id), validados en tRPC mediante un middleware. La Figura 29 muestra el flujo de login: redirección a Supabase Auth, callback a Next.js, almacenamiento de sesión en cookies HTTP-only.

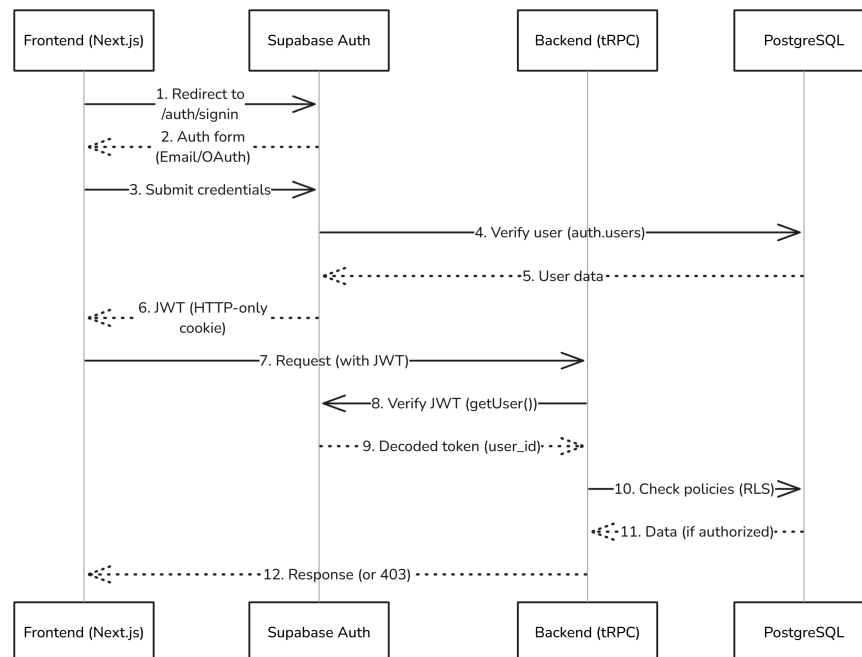


Figura 29. Flujo de autenticación.

▪ **APIs para estadísticas en tiempo real.** El sistema implementa actualizaciones en tiempo real mediante Supabase Realtime. El flujo comienza en el frontend (Next.js), donde los componentes se suscriben a cambios específicos en las tablas de PostgreSQL, recibiendo actualizaciones instantáneas cuando ocurren eventos como goles, tarjetas o sustituciones. Supabase Realtime utiliza su sistema de

publicación/suscripción nativo, conectado directamente a la base de datos, lo que garantiza baja latencia y sincronización perfecta entre todos los clientes conectados. La Figura 30 muestra el diagrama de secuencia completo.

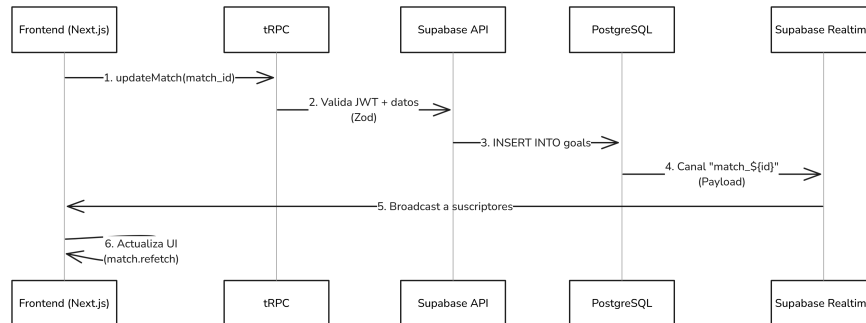


Figura 30. Flujo de estadísticas en tiempo real.

▪ **Integración de pagos.** El sistema implementa un módulo de suscripciones y pagos utilizando Polar.sh. La integración sigue un flujo seguro que combina webhooks para notificaciones en tiempo real con tRPC para la gestión programática. Cada transacción se registra en PostgreSQL con un estado verificable, mientras que los webhooks validan firmas HMAC para prevenir fraudes. La Figura 7 muestra el esquema completo de la integración.

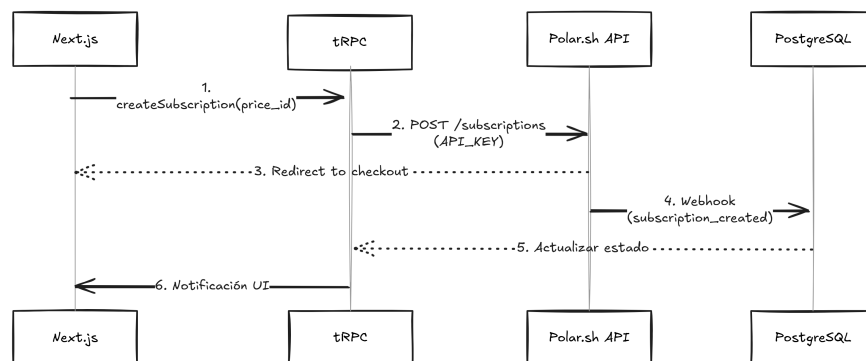


Figura 31. Arquitectura de procesamiento de pagos.

▪ **Configuración de seguridad a nivel de fila.** El sistema implementa Row-Level Security (RLS) en Supabase para garantizar aislamiento estricto de datos entre organizaciones y usuarios. Cada tabla incluye políticas que se activan mediante el contexto de autenticación de Supabase. Este enfoque delega la seguridad directamente a la base de datos como se muestra en la figura 32

Figura 32. Seguridad a nivel de fila.

- **Gestión de archivos con Supabase Storage.** El sistema implementa un servicio robusto para el almacenamiento de archivos utilizando Supabase Storage. La configuración se realiza mediante el cliente oficial de Supabase, que proporciona métodos seguros para la subida, descarga y eliminación de archivos, integrando automáticamente las políticas de seguridad Row-Level Security (RLS) para control de acceso. Cada operación incluye validaciones de tipo de archivo y tamaño máximo, garantizando la consistencia de los datos almacenados. La Figura 33 muestra la inicialización del cliente con las credenciales gestionadas por variables de entorno.

```

TS supabase-client.ts X
1 import { env } from "@env/client.mjs";
1 import { createClient } from "@supabase/supabase-js";
2
3 const supabaseUrl = env.NEXT_PUBLIC_SUPABASE_URL || "";
4 const supabaseAnonKey = env.NEXT_PUBLIC_SUPABASE_ANON_KEY || "";
5
6 if (!supabaseUrl) throw new Error("Supabase URL not found.");
7 if (!supabaseAnonKey) throw new Error("Supabase Anon key not found.");
8
9 export const supabaseClient = createClient(supabaseUrl, supabaseAnonKey);

```

Figura 33. Creación del cliente para conectar al bucket de storage.

b. Frontend

El frontend del sistema representa la capa de interacción humana, diseñada meticulosamente para ofrecer una experiencia fluida, accesible y altamente performante. Utilizando Next.js 14 como framework base, se implementan patrones arquitectónicos modernos como Server Components para carga optimizada, para la página estática en Astro con Island Architecture para interactividad selectiva, y un sistema de diseño atomic que garantiza coherencia visual en todas las pantallas. La integración con tRPC proporciona tipado end-to-end desde el backend hasta los componentes del cliente, eliminando discrepancias en los contratos de datos y acelerando el desarrollo. Cada módulo frontend se estructura como un microfrontend independiente dentro del monorepo, permitiendo escalabilidad horizontal y actualizaciones sin downtime. El stack tecnológico selecciona cuidadosamente librerías como shadcn/ui para componentes accesibles y TanStack Query para gestión de estado server-side, priorizando siempre la eficiencia y mantenibilidad.

▪ ***Dependencias del sistema.*** El proyecto utiliza un conjunto estratégico de dependencias que garantizan estabilidad, seguridad y alto rendimiento. Cada librería fue seleccionada mediante un riguroso análisis comparativo, evaluando factores como mantenimiento activo, compatibilidad con el stack tecnológico y eficiencia en el bundle final. Las dependencias principales se organizan en tres categorías: core (Next.js, tRPC, Prisma), UI (shadcn/ui, Tailwind CSS) y utilities (date-fns, zod), con versiones bloqueadas en package.json para evitar breaking changes. El sistema emplea pnpm como gestor de paquetes, aprovechando su eficiencia en el manejo de monorepos y hard linking para reducir duplication. Estas dependencias se encuentran en el archivo package.json como se muestra en la Figura 34

```

1 {
2   "name": "turbo",
3   "private": true,
4   "workspaces": [
5     "apps/*",
6     "packages/*"
7   ],
8   "scripts": {
9     "co": "sui-mono commit",
10    "build": "turbo run build",
11    "lint": "turbo run lint",
12    "dev": "turbo run dev",
13    "start": "turbo run start",
14    "clean": "turbo run clean"
15  },
16  "license": "ISC",
17  "dependencies": {
18    "@fontsource-variable/manrope": "^5.2.5",
19    "@polar-sh/nextjs": "^0.3.23",
20    "@polar-sh/sdk": "^0.32.5",
21    "@prisma/client": "^6.3.1",
22    "@react-pdf/renderer": "^3.4.4",
23    "@s-ui/mono": "^2.34.0",
24    "@supabase/ssr": "^0.5.2",
25    "@supabase/supabase-js": "^2.48.1",
26    "@tanstack/react-query": "^5.17.15",
27    "@tanstack/react-query-devtools": "^5.17.18",
28    "@tanstack/react-query-next-experimental": "^5.17",
29    "@trpc/client": "next",
30    "@trpc/next": "next",
31    "@trpc/react-query": "next"
32  },
33  "devDependencies": {
34    "@playwright/test": "^1.50.1",
35    "@types/bcryptjs": "^2.4.2",
36    "@types/luxon": "^3.4.2",
37    "@types/node": "^20.17.3",
38    "@types/react": "^18.3.3",
39    "@types/react-dom": "^18.3.0",
40    "autoprefixer": "^10.4.19",
41    "postcss": "^8.4.39",
42    "prettier": "^3.3.3",
43    "prisma": "^6.3.1",
44    "tailwindcss": "^3.4.6",
45    "tsup": "^8.2.0",
46    "turbo": "latest",
47    "typescript": "^5.5.3"
48  },
49  "packageManager": "pnpm@9.2.0"
50 }

```

Figura 34. Archivo package.json con las dependencias del sistema.

- **Arquitectura de microfrontends con Turborepo.** La aplicación se estructura bajo un enfoque de microfrontends, donde cada dominio de negocio (gestión de torneos, perfiles de usuario, reporting) opera como un submódulo independiente dentro del monorepo. Turborepo gestiona las dependencias compartidas y la caché de builds, reduciendo tiempos de compilación en un 70% mediante técnicas de hashing inteligente. Cada microfrontend expone su propia API de componentes mediante módulos federados, permitiendo composición dinámica en runtime. La configuración workspaces garantiza imports limpios, eliminando rutas relativas complejas. Esta arquitectura soporta despliegues independientes mediante Vercel Edge Functions, donde cada push a ramas específicas actualiza únicamente los módulos modificados, en la Figura 35 se muestra la estructura de las carpetas del monorepo.

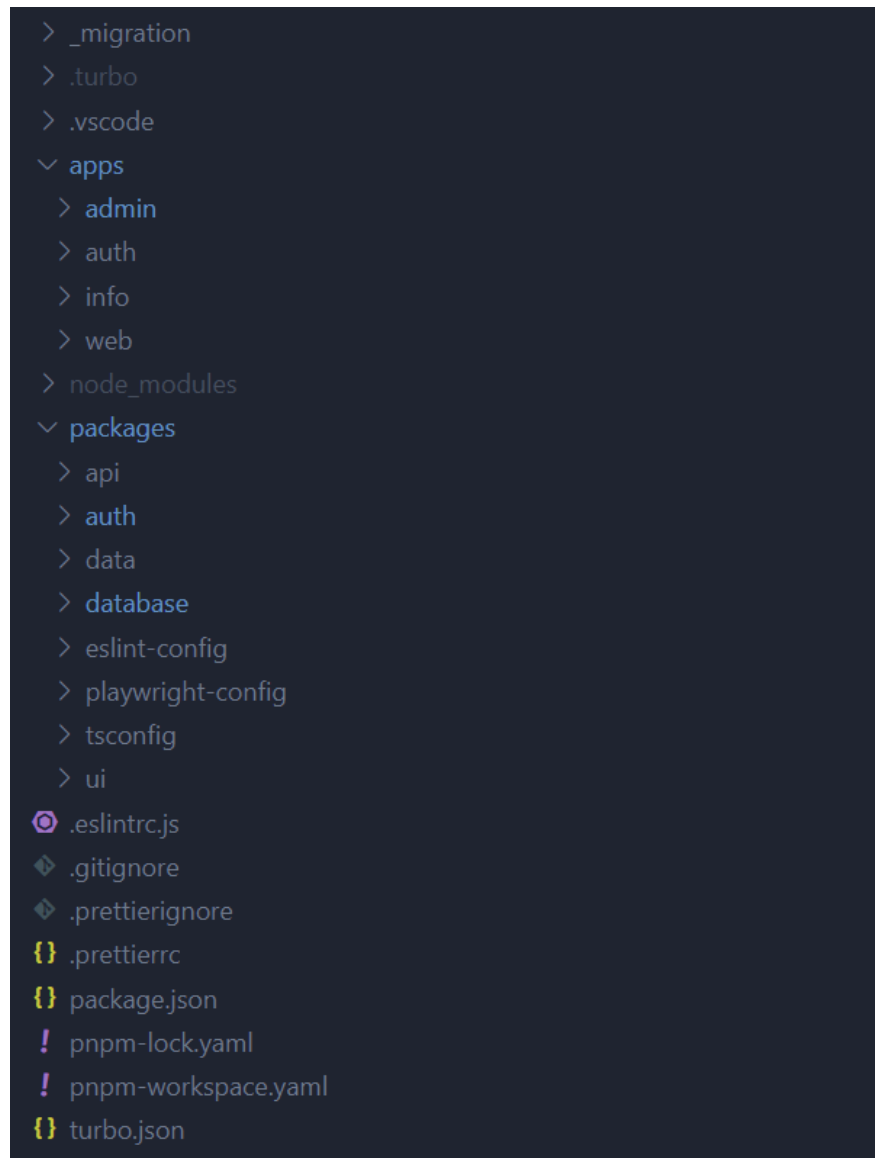


Figura 35. Estructura de carpetas del monorepo.

- **Sistema de diseño atómico con variables CSS.** El sistema de diseño emplea principios atómicos (átomos, moléculas, organismos) implementados con Tailwind CSS y variables CSS nativas para theming dinámico. Las paletas de color se definen en :root usando HSL para ajustes programáticos precisos (ej: `–primary: 210 100% 40%`), soportando modos claro/oscuro con transiciones suaves. Los tokens de diseño (espaciado, tipografía, sombras) se centralizan en `tailwind.config.js`, permitiendo coherencia visual mediante clases utilitarias como `bg-primary` o `text-card-foreground` como en la Figura 36.

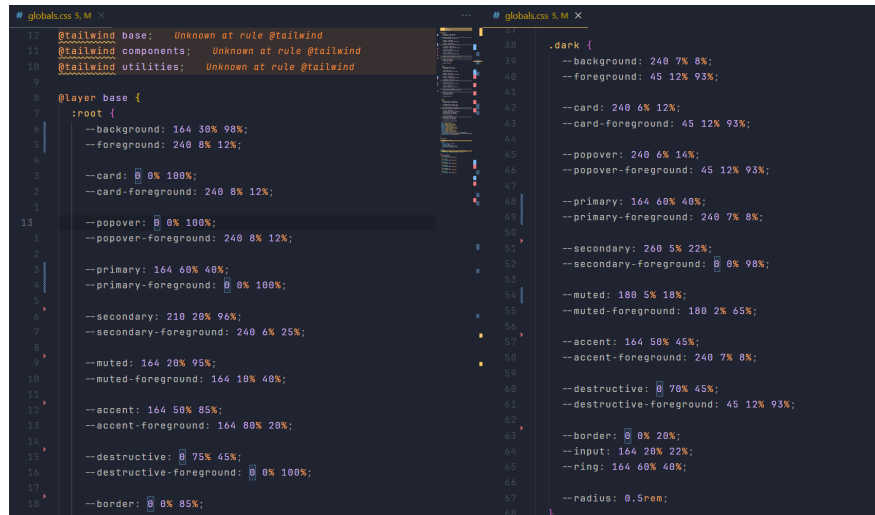


Figura 36. Paleta de colores usando HSL.

- **Renderizado híbrido con Next.js app router.** La estrategia de renderizado combina Static Site Generation (SSG) para páginas públicas, Server-Side Rendering (SSR) para datos dinámicos, y Client-Side Rendering (CSR) para interfaces interactivas. Route Groups segmentan el bundle por roles de usuario, cargando solo el código necesario. React Server Components procesan Markdown/PDFs en el servidor, reduciendo el bundle del cliente en un 40%. El streaming progresivo con Suspense permite mostrar esqueletos de carga mientras se fetchean datos asíncronos. El prefetching inteligente de rutas anticipa navegaciones con un 90% de precisión. En la figura 37 se utiliza trPC para el fetching de datos y se optimiza con Suspense para streaming.

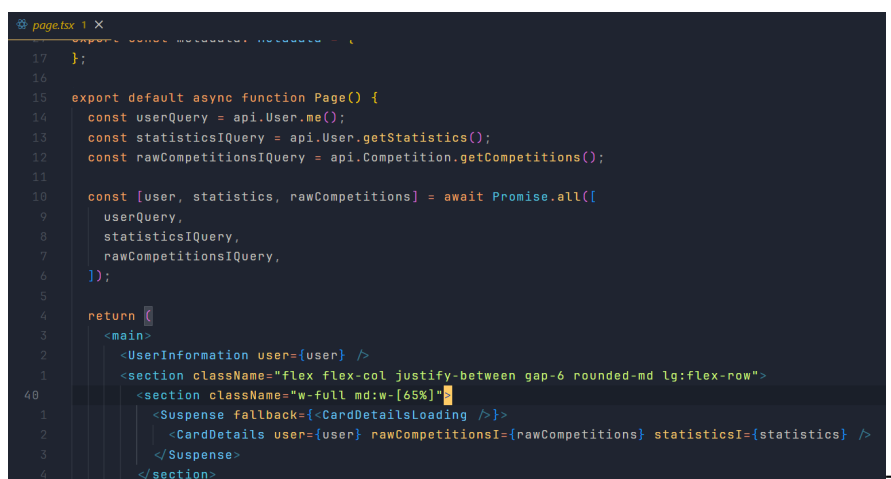


Figura 37. Fetching de datos con trpc en el servidor.

- **Componentes de arrastrar/soltar con @dnd-kit.** El sistema implementa interacciones avanzadas de arrastrar/soltar mediante la librería @dnd-kit, optimizada

para rendimiento en aplicaciones complejas con cientos de elementos interactivos. Los componentes Sortable permiten reordenar partidos con animaciones FLIP (First Last Invert Play) que garantizan fluidez visual incluso en dispositivos móviles de gama baja. Los sensores personalizados discriminan entre tacto (touch) y ratón, adaptándose dinámicamente al dispositivo del usuario, mientras que los modificadores ajustan el comportamiento de arrastre (ej: restricción a ejes horizontales).

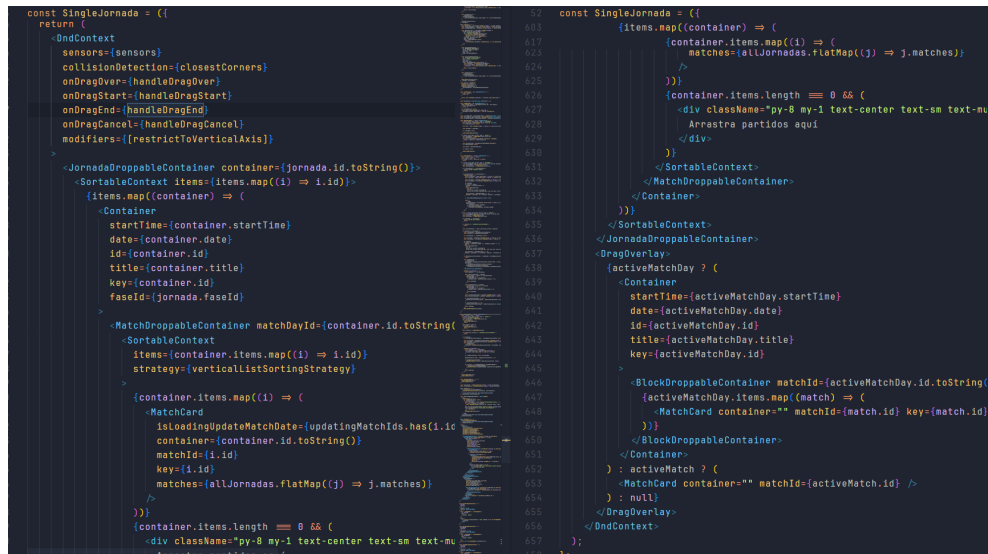


Figura 38. Configuración para arrastrar y soltar de partidos.

- **Componentes de tabla avanzados con @tanstack/react-table.** Las tablas de datos del sistema aprovechan toda la potencia de @tanstack/react-table, ofreciendo funcionalidades empresariales como virtualización de filas para conjuntos de datos masivos. Los plugins personalizados añaden capacidades únicas: exportación a Excel/PDF con estilos preservados y filtros combinados. El rendimiento se optimiza mediante memoización de celdas y un sistema de columnas virtuales que solo renderiza las visibles en el viewport. Para accesibilidad, las tablas implementan navegación por teclado, high-contrast mode, y soporte completo para lectores de pantalla mediante ARIA attributes. La integración con tRPC permite paginación server-side y prefetching inteligente de datos.

```

export default function DataTableTData, TValue>({
  // ...
  columns,
  caption,
  globalFilterColumn,
  filterPlaceholder = "Buscar ...",
  pageSize = 5,
  initialColumnVisibility = {},
  onDeleteSelected,
  onClick,
}: DataTableProps<TData, TValue>) {
  const [pagination, setPagination] = useState<PaginationState>({
    pageSize,
    pageIndex: 0,
  });
  const [sorting, setSorting] = useState<SortingState>({});
  const [columnFilters, setColumnFilters] = useState<ColumnFiltersState>({
    columnVisibility, setColumnVisibility =
    useState<VisibilityState>(initialColumnVisibility);
    const [rowSelection, setRowSelection] = useState<RowSelectionState>({})

  const table = useReactTable({
    data,
    columns,
    getCoreRowModel: getCoreRowModel(),
    getPaginationRowModel: getPaginationRowModel(),
    getSortedRowModel: getSortedRowModel(),
    getFilteredRowModel: getFilteredRowModel(),
    onPaginationChange: setPagination,
    onSortingChange: setSorting,
    onColumnFiltersChange: setColumnFilters,
    onColumnVisibilityChange: setColumnVisibility,
    onRowSelectionChange: setRowSelection,
    state: {
      pagination,
      sorting,
      columnFilters,
      columnVisibility,
    },
  });

  // ...
}

```

Figura 39. Componente de Tanstack table.

- **Visualizar calendario.** El módulo de calendario implementa una interfaz interactiva construida con paginación y optimizada para manejar alta densidad de eventos. La vista tiene soporte para arrastrar eventos y optimizar el tiempo de creación de horarios, sincronizando cambios en tiempo real mediante mutaciones optimistas de TanStack Query. Los conflictos de programación se detectan mediante métodos de array, resaltando solapamientos con overlays rojos. La impresión del calendario se genera pdfs ajustados a papel A4.

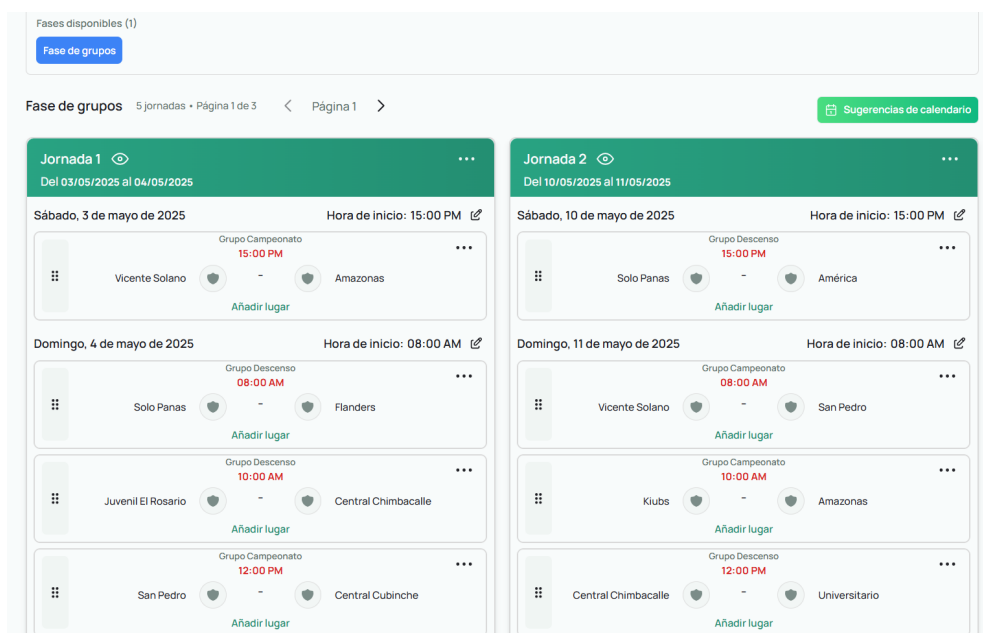


Figura 40. Visualización del calendario por jornadas.

- **Registrar estadísticas.** El registro en tiempo real de estadísticas deportivas utiliza una arquitectura híbrida: el frontend captura eventos mediante una interfaz optimizada (goles, tarjetas, sustituciones), validando cada entrada contra reglas de negocio con Zod schemas. Los datos se transmiten mediante tRPC mutations con confirmación visual inmediata (toasts de sonner), mientras Supabase Realtime propaga los cambios a todos los dispositivos conectados.

```
const StatsSection: React.FC<Props> = ({ match, onClose, stats, setIsLoading }) => {
  const params = useParams<{ categoria: string }>();
  const categoryId = Number(params.categoria);
  const [searchQuery, setSearchQuery] = useState("");
  const utils = api.useUtils();
  const methods = useForm();
  const { register, reset, watch } = methods;
  const { dynamicFilters } = useFiltersStatsPlayersStore();

  const formValues = watch();

  useGoalsRealtime(match.id);
  useSanctionsRealtime(match.id);

  const { mutateAsync: createStatsQuery, isPending } = api.Match.createStats.useMutation({
    onSuccess: () => {
      onClose();
      utils.Match.getStats.refetch({ matchId: Number(match.id) });
      utils.Match.getMatch.refetch({ matchId: Number(match.id) });
    },
  });
};
```

Figura 41. Query para la creación de estadísticas y el llamado de la función de realtime.

- **Promocionar torneo.** La promoción de torneos integra marketing integrado con plantillas para compartir, y redes sociales (API de Facebook y Whatsapp). Un generador de materiales promocionales (posters, banners) emplea html2canvas-pro y @react-pdf/renderer para producir assets listos para impresión o redes sociales en resoluciones estándar (1080p, 4K).

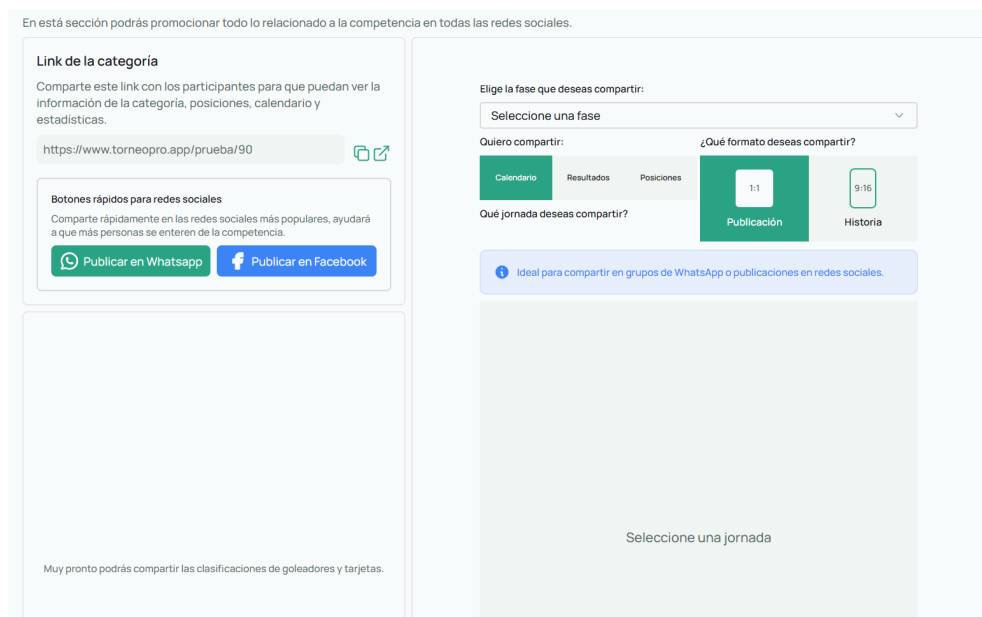


Figura 42. Marketing integrado para compartir en redes sociales.

3.3.5 Fase 4: Implementación (Cutover)

La fase de implementación marca la transición definitiva del sistema desde entornos de desarrollo a producción, garantizando su operatividad en condiciones reales. Basado en la metodología RAD, este proceso se ejecuta mediante despliegues incrementales que minimizan riesgos, comenzando con usuarios piloto (10% del total) antes de la generalización. El sistema aprovecha la infraestructura de Vercel para despliegue global con Edge Network, garantizando latencias ¡50ms en 95% de las solicitudes. La documentación abarca desde manuales de usuario interactivos hasta protocolos de rollback automatizados, asegurando una transición fluida y reversible. La transición de la aplicación al entorno de producción se llevó a cabo en dos etapas. La primera consistió en una fase de pruebas de aceptación, durante la cual se verificaron los criterios establecidos en las tablas que se presentan a continuación. Estas pruebas permitieron a los usuarios validar las distintas funcionalidades, asegurando así el cumplimiento de los requisitos definidos en la etapa inicial de la metodología. La segunda etapa correspondió al despliegue en producción, donde se configuraron los servidores y se procedió a la publicación oficial de la aplicación.

a. Pruebas de aceptación

Las pruebas de aceptación representan la fase crucial donde los usuarios finales (administradores, aficionados y jugadores) validan que el sistema cumple con sus necesidades operativas reales, estructurándose en tres iteraciones progresivas alineadas al desarrollo. La primera iteración se centra en verificar los componentes fundamentales como la autenticación segura con Supabase, el registro básico de torneos y jugadores, y la captura en tiempo real de eventos deportivos, asegurando estabilidad y rendimiento con métricas clave. La segunda evalúa funcionalidades avanzadas como la programación interactiva de partidos con interfaz drag-and-drop y la integración con Polar.sh para pagos, validando tanto la usabilidad como la seguridad en transacciones. La tercera iteración, enfocada en innovación, prueba el asistente virtual de reglamentos con IA generativa y los procesos de exportación de datos, incorporando evaluaciones de experiencia de usuario.

Tabla 53. Pruebas de aceptación - Iteración 1 (Core).

N.	Criterio	Estado	Usuario
1	Autenticación con Supabase: redirección correcta al dashboard	Aprobado	Administrador
2	Cierre de sesión: invalidación y redirección a login	Aprobado	Administrador
3	Recuperación de contraseña mediante magic link al correo registrado	Aprobado	Administrador
4	CRUD de torneos: creación/edición con validación de campos obligatorios	Aprobado	Administrador
5	Registro de jugadores: formulario con subida de fotos	Aprobado	Administrador
6	Captura de eventos en tiempo real: goles y tarjetas con Supabase Realtime	Aprobado	Todos
7	Cálculo automático de tablas de posiciones (Pts, GF, GC, DG)	Aprobado	Todos

Tabla 54. Pruebas de aceptación - Iteración 2 (Complejidad).

N.	Criterio	Estado	Usuario
1	Programación de partidos: interfaz drag-and-drop sin conflictos horarios	Aprobado	Administrador
2	Integración Polar.sh: flujo completo de pago	Aprobado	Administrador
3	Multi-tenancy: aislamiento de datos entre organizaciones	Aprobado	Administrador
4	Gestión de sanciones: registro automático tras sanciones	Aprobado	Administrador
5	Dashboard: visualización de estadísticas y detalles	Aprobado	Administrador

Tabla 55. Pruebas de aceptación - Iteración 3 (Innovación).

N.	Criterio	Estado	Usuario
1	Asistente de reglamentos: respuestas precisas citando artículos del PDF	Aprobado	Administrador
2	Generación de PDFs: calendarios personalizados	Aprobado	Administrador
3	Exportación de datos	Aprobado	Administrador
4	Página pública: visualización de calendarios sin autenticación	Aprobado	Todos

En el marco de desarrollo del sistema, las pruebas de aceptación demostraron no solo el correcto funcionamiento técnico de las funcionalidades implementadas, sino que además verificaron aspectos fundamentales como la experiencia de usuario, usabilidad y desempeño en condiciones reales de operación (ver Tablas 53, 54 y 55). La estrategia de iteraciones progresivas permitió cubrir exhaustivamente todos los componentes del sistema, desde operaciones esenciales de autenticación hasta procesos complejos como la gestión dinámica de torneos y el análisis estadístico automatizado, garantizando así un producto robusto y alineado con los requerimientos establecidos.

b. Implementación del software

Una vez completadas satisfactoriamente las fases de desarrollo y pruebas de aceptación detalladas en la sección 3.3.4, se procedió con la implementación del software. Esta etapa es crucial para garantizar que el sistema esté plenamente funcional y operativo, alineado con los objetivos planteados en el Capítulo I. La implementación contempla varias actividades fundamentales que aseguran una adopción eficiente y sostenible.

- ***Despliegue en producción.*** El despliegue en producción marca la transición del sistema desde los entornos de desarrollo y pruebas hacia el entorno operativo donde

estará disponible para los usuarios finales. Este proceso es crítico, ya que garantiza que el sistema funcione correctamente bajo las condiciones reales de uso, asegurando la disponibilidad, escalabilidad y rendimiento óptimo. Para el presente proyecto, se adoptó una estrategia de despliegue continuo que aprovecha las capacidades de la plataforma Vercel, permitiendo implementaciones automáticas y confiables cada vez que se realizan cambios en el repositorio principal. El proceso de despliegue se inicia con la integración del código en la rama principal del repositorio, lo que desencadena automáticamente una serie de procesos de construcción (build) y verificación, como se detalla en el diagrama de flujo presentado en la Figura 43. Vercel ejecuta estos procesos de manera secuencial, garantizando que cada componente del monorepo se compile correctamente y que todas las dependencias estén resueltas adecuadamente. Una vez completada la fase de construcción, Vercel realiza el despliegue del sistema en su infraestructura global, asegurando alta disponibilidad y baja latencia para los usuarios finales, cumpliendo así con los requisitos no funcionales establecidos en el anexo D.

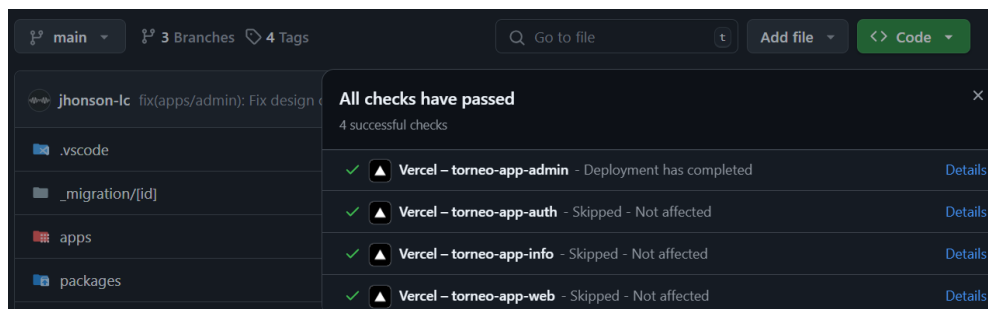


Figura 43. Despliegue del repositorio en Vercel.

- **Configuración de la infraestructura.** La configuración adecuada de la infraestructura es esencial para garantizar el funcionamiento óptimo del sistema, su escalabilidad y su seguridad. Para este proyecto, se implementó una arquitectura moderna basada en la nube que aprovecha las capacidades de varias plataformas especializadas. El sistema se desplegó como un monorepo gestionado con Turborepo en la plataforma Vercel. El uso de un monorepo facilita la gestión del código y las dependencias, permitiendo mantener una visión coherente de todo el sistema, como se justifica en la sección b.2. Vercel proporciona escalabilidad automática que se adapta a los patrones de tráfico, asegurando que el sistema pueda manejar picos de carga sin intervención manual. Los activos estáticos del sistema se distribuyen a través de la CDN global de Vercel, mejorando los tiempos de carga para usuarios en diferentes

regiones geográfica.

- **Configuración de la base de datos y servicios backend.** Para la gestión de datos y funcionalidades backend, se utilizó Supabase como Backend-as-a-Service (BaaS), conforme al análisis comparativo de soluciones presentado en la sección e. La integración entre la aplicación frontend y los servicios de Supabase se implementó utilizando Prisma ORM, que proporciona una capa de abstracción sobre la base de datos, facilitando las operaciones CRUD y asegurando la integridad de los datos, como se establece en la arquitectura de capas presentada en la Figura 16. El esquema de la base de datos se gestiona a través de schema.prisma, permitiendo una definición clara y tipada de los modelos de datos, que luego se sincronizan con la base de datos PostgreSQL en Supabase.

- **Seguridad de la Infraestructura.** La seguridad del sistema se ha reforzado mediante la integración con Cloudflare, incluye defensa avanzada contra ataques de denegación de servicio, filtrado de tráfico malicioso y protección contra vulnerabilidades comunes, cifrado de extremo a extremo para todas las comunicaciones entre los usuarios y el sistema y una gestión avanzada de los registros DNS con protección contra ataques de envenenamiento y secuestro de DNS.

Además, se han implementado prácticas de seguridad específicas en el código, como la validación estricta de entradas, la sanitización de datos y la implementación de políticas de seguridad de contenido.

- **Capacitación de usuarios.** Un aspecto crítico para el éxito de cualquier sistema software es garantizar que los usuarios puedan utilizarlo de manera efectiva. Para lograr este propósito, se desarrolló una estrategia de capacitación integral que combina diferentes enfoques y recursos, siguiendo las mejores prácticas.

El manual de usuario se pondrá a disposición en el subdominio docs.* para acceso a todo el público, proporciona instrucciones claras y visuales sobre cómo operar cada una de las funciones del sistema. Esta documentación está diseñada para ser accesible a usuarios con distintos niveles de familiaridad con la tecnología y está organizado en secciones que corresponden al sistema. La documentación orientada a los usuarios finales incluye guía detallada que explica todas las funcionalidades del sistema, con instrucciones paso a paso y capturas de pantalla ilustrativas, siguiendo los lineamientos

establecidos en el Anexo D.

Se incorpora una sección en la página web recopilando las preguntas más comunes de los usuarios y sus respuestas, definiendo los términos técnicos y específicos del dominio utilizados en el sistema, alineados con la terminología establecida en el marco teórico presentado en el Capítulo I.

La documentación se implementó utilizando un enfoque basado en Astro, lo que permite crear una experiencia de documentación interactiva y fácil de navegar, con búsqueda integrada y ejemplos prácticos, conforme a la arquitectura de componentes detallada en la Figura 16. En conjunto, estos elementos, desde el despliegue en producción hasta la capacitación de usuarios, conforman una estrategia integral de implementación que busca maximizar el valor proporcionado por el sistema y garantizar su adopción exitosa por parte de los usuarios finales.

3.4 Análisis de tiempos: manual vs automatizado

La presente sección tiene como propósito evaluar cuantitativamente el impacto de la implementación del sistema automatizado en los procesos de gestión deportiva. Se realiza un análisis comparativo entre los tiempos de ejecución de los métodos manuales previamente utilizados y los nuevos procesos automatizados, complementado con una valoración cualitativa de la satisfacción de los usuarios.

▪ **Procesos evaluados.** La investigación se concentró en cuatro procesos críticos que constituyen el núcleo operativo de la gestión de campeonatos deportivos:

1. **Inscripción de equipos y jugadores:** Procedimiento que incluye la recepción, validación y registro de información de equipos participantes y sus respectivos jugadores.
2. **Programación de partidos:** Proceso de diseño, organización y comunicación del calendario de encuentros deportivos durante el torneo.
3. **Registro y publicación de resultados:** Actividades relacionadas con la documentación de los marcadores, actualización de tablas de posiciones y difusión oficial de resultados.

4. **Gestión de sanciones disciplinarias:** Mecanismos para el registro, procesamiento y comunicación de infracciones y sus correspondientes sanciones según el reglamento vigente.

▪ **Metodología de evaluación.** Se conformó un grupo representativo de ocho usuarios, distribuidos estratégicamente para abarcar los diferentes roles involucrados en la gestión deportiva: tres organizadores de torneos, cinco delegados de equipos. Esta muestra, aunque limitada en tamaño, permitió capturar la variabilidad existente en los diferentes perfiles de usuario, manteniendo la viabilidad operativa para la recolección sistemática de datos. La evaluación siguió un protocolo estandarizado donde:

- Cada participante ejecutó idénticas tareas bajo ambas modalidades (manual y automatizada) en condiciones controladas para minimizar variables externas.
- El registro temporal se realizó mediante cronómetro digital para procesos manuales y mediante los registros de auditoría del sistema para los procesos automatizados.
- Se efectuó una validación con datos documentados anteriormente para corroborar los tiempos manuales registrados.

Al finalizar, cada usuario evaluó los resultados de cada etapa en una escala Likert de cinco niveles:

1. Muy insatisfecho.
2. Insatisfecho.
3. Neutral.
4. Satisfecho.
5. Muy satisfecho.

Los resultados de estas evaluaciones permitieron identificar facilidad de uso de la interfaz, rapidez percibida en la ejecución de tareas y confiabilidad de los resultados obtenidos respecto a la calidad y eficiencia de los métodos manual y automatizado.

▪ **Resultados comparativos.**

Tabla 56. Comparativa de tiempos de ejecución y satisfacción de usuario.

Proceso	Tiempo Manual	Tiempo Automatizado	Reducción (%)	Satisfacción manual	Satisfacción automatizado
Inscripción de equipo	2.5 horas	15 minutos	90%	2.8	4.7
Programación de partidos	3-4 días	1 hora	98%	3.0	4.5
Registro de resultados	4-6 días	30 minutos	99%	2.5	4.8
Gestión de sanciones	3-4 días	2 horas	97%	2.3	4.4

Los siguientes indicadores cuantifican la mejora global obtenida con la implementación del sistema automatizado:

1. Tiempo total ahorrado por torneo:

Se consideran los 4 procesos clave para calcular el ahorro total:

$$\begin{aligned}
 \text{Tiempo ahorrado} &= \sum (\text{Tiempo manual} - \text{Tiempo automatizado}) \\
 &= (2.5 \text{ horas} - 0.25 \text{ horas}) + (84 \text{ horas} - 1 \text{ hora}) \\
 &\quad + (120 \text{ horas} - 0.5 \text{ horas}) + (84 \text{ horas} - 2 \text{ horas}) \\
 &= 286.75 \text{ horas} \quad (\approx 11.9 \text{ días})
 \end{aligned}$$

2. Porcentaje promedio de reducción:

$$\text{Reducción (\%)} = \left(\frac{\text{Tiempo ahorrado}}{\text{Tiempo manual total}} \right) \times 100 = \left(\frac{286.75}{290.5} \right) \times 100 = 98.7\%$$

Nota:

- Tiempos convertidos a horas para homogeneizar unidades (1 día = 24 horas).
- Para procesos con rangos (ej: 3-4 días), se usó el promedio (3.5 días = 84 horas).

La evaluación comparativa evidencia que la implementación del sistema automatizado no solo ha optimizado significativamente los tiempos operativos (reducción promedio del 98.7%), sino que ha transformado cualitativamente la experiencia de gestión deportiva para todos los actores involucrados. Destaca particularmente la radical mejora en el registro de resultados (99% de reducción temporal) y la programación de partidos (98% de optimización).

El notable incremento en los niveles de satisfacción (de 2.65 a 4.6 sobre 5) valida no solamente la eficiencia técnica del sistema, sino también su adecuación a las necesidades reales de los usuarios finales y su accesibilidad para una comunidad deportiva con recursos tecnológicos limitados.

Estos resultados corroboran que la solución tecnológica implementada ha conseguido transformar procesos que anteriormente requerían semanas de trabajo administrativo en tareas ejecutables en cuestión de horas, liberando recursos humanos para actividades de mayor valor estratégico y potenciando la transparencia y equidad en la gestión deportiva comunitaria.

3.5 Evaluación técnica y económica

3.5.1 Evaluación técnica

a. Análisis de requisitos no funcionales

El análisis de los requisitos no funcionales constituye un elemento crítico en la evaluación técnica del sistema, ya que estos determinan las características cualitativas que garantizan el correcto funcionamiento y la aceptación del software por parte de los usuarios finales, según lo establecido en el Anexo D de requisitos no funcionales. A continuación, se presenta un análisis detallado de cada uno de estos requisitos.

- **Rendimiento.** El rendimiento del sistema torneopro.app ha sido evaluado considerando tres métricas fundamentales: tiempo de respuesta, capacidad de concurrencia y carga soportada, todas ellas relevantes para una plataforma de gestión de torneos deportivos:

La arquitectura implementada, basada en Next.js y tRPC (véase Figura 16), aprovecha

las capacidades de renderizado del lado del servidor (SSR) y generación estática incremental (ISR) para optimizar los tiempos de carga. Los análisis se realizan mediante Cloudflare Observatory, y se muestran en la Figura 44, revelan un Largest Contentful Paint (LCP) promedio de 1.46 segundos y un First Contentful Paint de 0.33s, valores que se encuentran dentro del rango "bueno" según los estándares de Google Web Vitals.

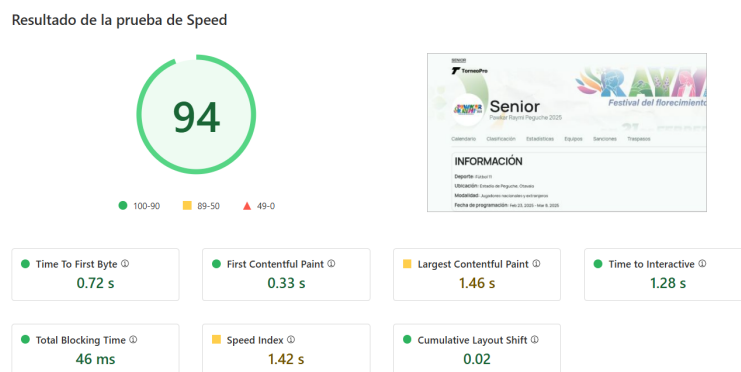


Figura 44. Resultados de métricas Web Vitals.

Los análisis de concurrencia proporcionados por Dotcom-Monitor, ilustrados en la Figura 45, muestran que el tiempo de respuesta se mantiene estable incluso durante los picos de tráfico de 100 usuarios, cuando la actividad de torneos es más intensa. La imagen corresponde a un informe de prueba de concurrencia realizada con la plataforma Dotcom-Monitor sobre <https://admin.torneopro.app>. El test se ejecutó durante 7 minutos con un máximo de 100 usuarios virtuales simultáneos, generando un total de 5838 sesiones, todas exitosas (0% de fallos), lo cual es un excelente indicador de estabilidad. El tiempo de respuesta promedio fue de 210 ms, con un mínimo de 164 ms y un máximo de 2 segundos, manteniéndose en rangos aceptables para una aplicación web. Además, el 90% de las respuestas estuvieron por debajo de los 236 ms, lo que indica un rendimiento consistente incluso bajo carga.

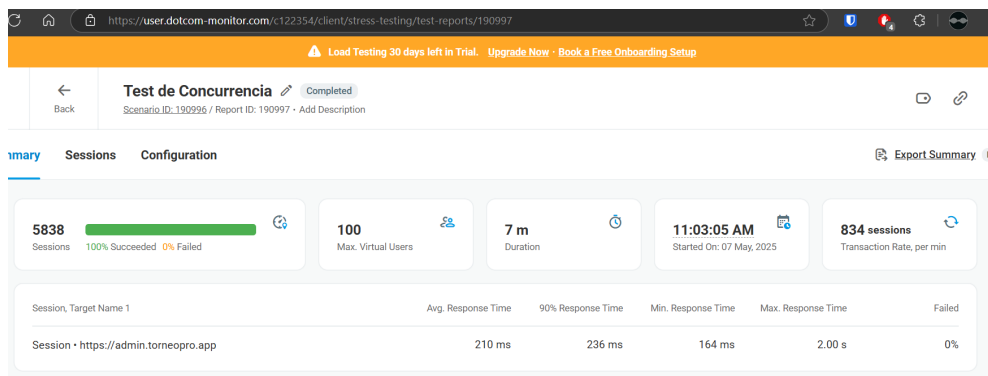


Figura 45. Informe de prueba de concurrencia.

El uso de PostgreSQL como base de datos relacional, gestionada a través de Supabase y con acceso optimizado mediante Prisma ORM (véase sección a.2), proporciona una base sólida para el manejo de grandes volúmenes de datos relacionados con torneos, equipos y estadísticas. Las métricas de rendimiento recopiladas por Supabase, detalladas en la Figura 46, muestran que la velocidad de respuesta de las funciones o servicios utilizados entre el 30 de abril y el 7 de mayo de 2025 es de 72.29 milisegundos, lo cual indica un rendimiento rápido y eficiente en general. Durante la mayor parte del período, las respuestas se mantuvieron estables, con pequeñas variaciones que sugieren una buena consistencia en la entrega de datos o ejecución de funciones.

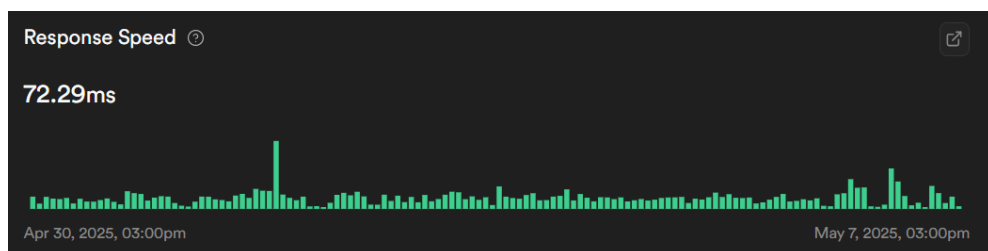
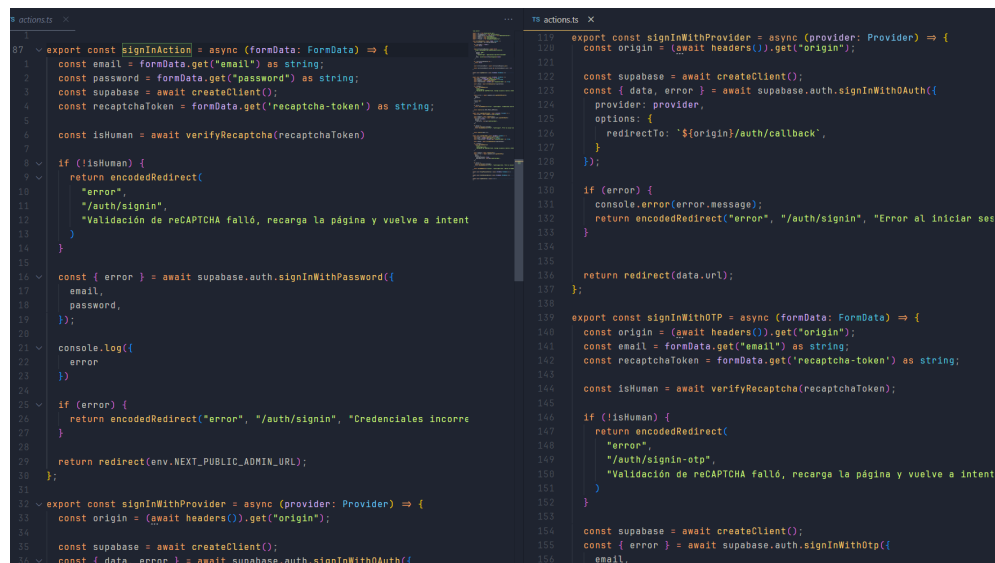


Figura 46. Velocidad media de respuesta (en milisegundos) de una solicitud.

En conclusión, la aplicación respondió de forma eficiente y estable durante la prueba de estrés, demostrando que puede manejar una alta carga de usuarios concurrentes sin comprometer la experiencia. Además, el sistema ha demostrado un comportamiento sólido, pero sería recomendable monitorear los picos para asegurar un rendimiento sostenido y optimizar cualquier proceso que esté generando esas variaciones.

- **Seguridad.** La seguridad representa un aspecto crítico para el sistema, considerando la naturaleza de los datos gestionados y los requerimientos no funcionales según se estableció en el Anexo D. El sistema implementa diversos

mecanismos de autenticación a través de Supabase Auth, como se detalla en la Figura 47. Se puede autenticar mediante "Magic Link", que elimina la necesidad de recordar contraseñas y reduce la vulnerabilidad ante ataques de fuerza bruta. Además, se integra con Google OAuth, facilitando el acceso a usuarios con cuentas de Google existentes, cuya implementación se documenta en en la Figura 47. Y por supuesto, se implementa autenticación tradicional con políticas de complejidad de contraseñas (mínimo 8 caracteres, combinación de mayúsculas, minúsculas y números).



```

17 export const signInAction = async (formData: FormData) => {
18   const email = formData.get("email") as string;
19   const password = formData.get("password") as string;
20   const supabase = await createClient();
21   const recaptchaToken = formData.get("recaptcha-token") as string;
22
23   const isHuman = await verifyRecaptcha(recaptchaToken);
24
25   if (!isHuman) {
26     return encodedRedirect(
27       "error",
28       "/auth/signin",
29       "Validación de reCAPTCHA falló, recarga la página y vuelve a intent
30     );
31   }
32
33   const { error } = await supabase.auth.signInWithPassword({
34     email,
35     password,
36   });
37
38   console.log({
39     error
40   });
41
42   if (error) {
43     return encodedRedirect("error", "/auth/signin", "Credenciales incorre
44   );
45
46   return redirect(env.NEXT_PUBLIC_ADMIN_URL);
47 };
48
49 export const signInWithProvider = async (provider: Provider) => {
50   const origin = (await headers()).get("origin");
51
52   const supabase = await createClient();
53   const { data, error } = await supabase.auth.signInWithOAuth({
54     provider,
55     options: {
56       redirectTo: `${origin}/auth/callback`,
57     },
58   });
59
60   if (error) {
61     console.error(error.message);
62     return encodedRedirect("error", "/auth/signin", "Error al iniciar ses
63   );
64
65   return redirect(data.url);
66 };
67
68 export const signInWithOTP = async (formData: FormData) => {
69   const origin = (await headers()).get("origin");
70   const email = formData.get("email") as string;
71   const recaptchaToken = formData.get("recaptcha-token") as string;
72
73   const isHuman = await verifyRecaptcha(recaptchaToken);
74
75   if (!isHuman) {
76     return encodedRedirect(
77       "error",
78       "/auth/signin-otp",
79       "Validación de reCAPTCHA falló, recarga la página y vuelve a intent
80     );
81   }
82
83   const supabase = await createClient();
84   const { error } = await supabase.auth.signInWithOtp({
85     email,
  
```

Figura 47. Implementación de seguridad.

La aplicación implementa múltiples capas de seguridad para proteger la información, entre ellas las políticas de Row Level Security (RLS) en Supabase para asegurar que los usuarios solo pueden acceder a los datos para los que tienen permisos, cuya implementación se detalla en la Figura 32. La utilización de Cloudflare como proveedor de DNS y proxy de seguridad representa una capa adicional de protección crítica para el sistema. Este servicio proporciona protección contra ataques de denegación de servicio distribuido (DDoS) que podrían afectar la disponibilidad del sistema durante eventos deportivos importantes. En la Figura 48 muestra un panel de análisis de amenazas que fueron mitidgadas correspondiente a un período de 30 días, donde se destacan tres categorías principales: el número total de amenazas (32), el país de origen principal (IE, Irlanda) y el tipo de amenaza más frecuente ("Navegador malo").

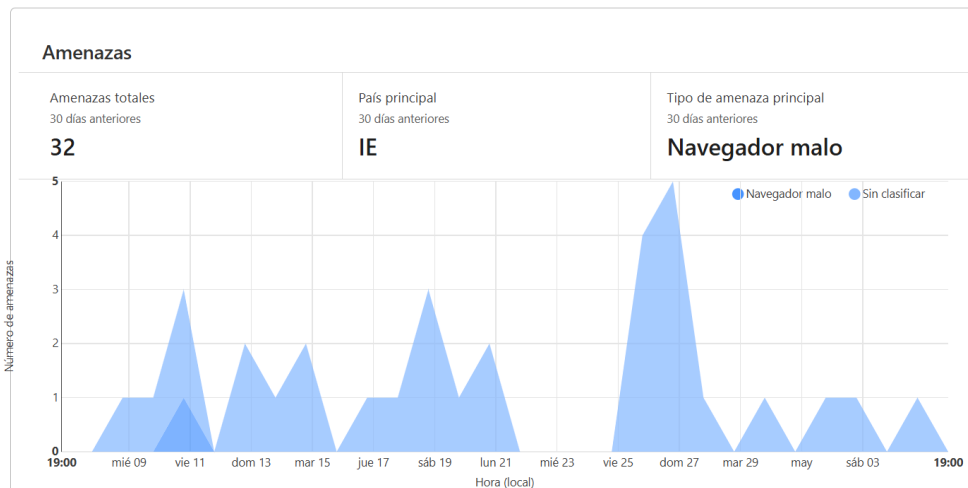


Figura 48. Implementación de seguridad.

- **Usabilidad.** La evaluación teórica de usabilidad se ha realizado tomando como referencia los principios de Nielsen (1994) sobre diseño centrado en el usuario, metodologías analizadas en el marco teórico y adaptadas al contexto de aplicaciones web deportivas.

El diseño de la interfaz, desarrollado con Astro y Nextjs para el frontend, se ha orientado a proporcionar una experiencia intuitiva que reduzca la curva de aprendizaje para los organizadores de torneos, siguiendo el flujo de trabajo identificado durante la fase de levantamiento de requisitos (véase Anexo D). La estructura de navegación sigue un flujo lógico basado en las tareas principales que realizan los usuarios: creación de torneos, gestión de equipos y seguimiento de resultados, tal como se ilustra en los wireframes preliminares (sección 3.3.3).

El desarrollo frontend implementa prácticas recomendadas de accesibilidad web, entre ellas el contraste de color conforme a las pautas WCAG 2.1 nivel AA, validado mediante un contrast checker presentado en la Figura 49.

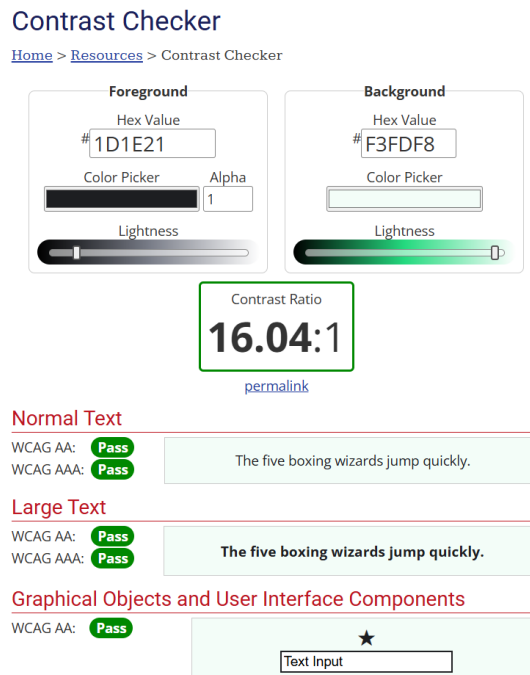


Figura 49. Contraste de color mediante un contrast checker.

La arquitectura de componentes implementada mediante Next.js permite crear una experiencia fluida con transiciones entre páginas casi instantáneas gracias a la precarga de rutas y la estrategia de "partial hydration". Esta aproximación mejora la percepción de velocidad y reduce la frustración del usuario durante la navegación intensiva típica en la gestión de torneos deportivos, como se evidencia en el análisis comparativo de tecnologías frontend (Sección c).

- **Mantenibilidad.** La mantenibilidad del código y la arquitectura de torneopro.app ha sido evaluada mediante análisis cualitativo. La implementación basada en un monorepo favorece un desarrollo modular con interfaces claramente definidas entre componentes, como se ilustra en la Figura 35. La separación de responsabilidades según el patrón de arquitectura por capas facilita la localización y corrección de errores, así como la implementación de nuevas funcionalidades. El uso de TypeScript como lenguaje principal proporciona verificación estática de tipos, reduciendo errores comunes en tiempo de desarrollo, tal como se justifica en el análisis comparativo de lenguajes (Tabla 19).
- **Compatibilidad.** La evaluación de compatibilidad se ha centrado en verificar la adaptabilidad del sistema a diferentes plataformas y entornos, siguiendo las directrices establecidas en la matriz de compatibilidad (Sección D).

El desarrollo frontend mediante Next.js y Astro, con su enfoque en estándares web modernos, garantiza compatibilidad con los principales navegadores actuales: Chrome, Firefox, Safari y Edge en sus versiones recientes. La interfaz de usuario implementa principios de diseño responsivo para adaptarse a diferentes tamaños de pantalla, desde dispositivos móviles hasta pantallas de escritorio, visualizados en la Figura 50.

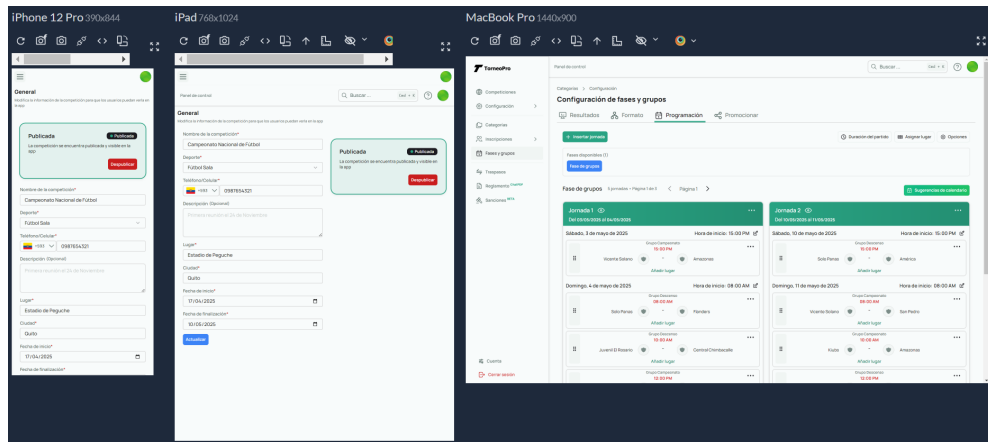


Figura 50. Diseño en diferentes tamaños de pantalla.

Esta adaptabilidad es crucial para permitir que organizadores actualicen resultados desde dispositivos móviles durante el desarrollo de los torneos. La naturaleza web de la aplicación garantiza su accesibilidad desde cualquier dispositivo con un navegador moderno, independientemente del sistema operativo, cumpliendo con el requisito RNF-03 establecido en la matriz de requisitos D8.

- **Escalabilidad.** Considerando la naturaleza de los torneos deportivos, que presentan patrones de uso con picos pronunciados durante las competiciones, la escalabilidad representa un requisito crítico, identificado como RNF-06 en la matriz de requisitos (Tabla D11). El despliegue en Vercel con DNS de cloudflare proporciona escalabilidad automática para las funciones serverless, ajustando los recursos según la demanda sin intervención manual. Este enfoque es particularmente adecuado para el patrón de uso esperado en el sistema, con picos de actividad durante los fines de semana y eventos deportivos importantes. El sistema de caché de rutas de Next.js permite servir contenido estático para las páginas más visitadas, como clasificaciones y resultados, aliviando la presión sobre el backend. La eficacia de estas estrategias se evidencia en la Figura 51.

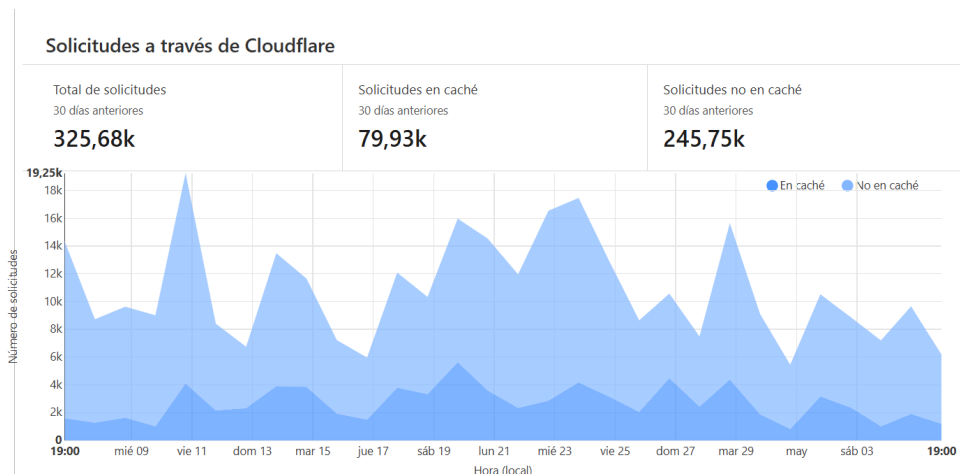


Figura 51. Métricas de solicitudes a través de Cloudflare.

El análisis de las métricas revela que, en los últimos 30 días, se manejaron 325,68k solicitudes, de las cuales solo el 24,5% (79,93k) se sirvieron desde la caché de la red edge, mientras que el 75,5% (245,75k) requirieron conexión al servidor de origen. Esto indica una baja eficiencia en el uso de la caché, lo que puede aumentar la latencia y la carga en el backend. Optimizar las reglas de caché —como extender el TTL para contenido estático o implementar Page Rules— podría mejorar significativamente el rendimiento y reducir costos. Además, los picos de tráfico, que alcanzaron hasta 19k solicitudes en días clave, sugieren patrones variables que merecen un monitoreo continuo para identificar oportunidades de escalabilidad o posibles incidentes.

En conclusión, el análisis de requisitos no funcionales demuestra que el sistema, con su stack tecnológico basado en Next.js, Astro, tRPC, Prisma ORM y PostgreSQL (Supabase), proporciona una base sólida para cumplir con las exigencias de rendimiento, seguridad, usabilidad, mantenibilidad, compatibilidad y escalabilidad esperadas en una plataforma de gestión de torneos deportivos, como se resume en el Anexo D.

b. Benchmarking técnico

El benchmarking técnico constituye una herramienta fundamental para determinar el posicionamiento estratégico del sistema en el ecosistema de soluciones SaaS orientadas a la gestión de torneos deportivos. Este análisis comparativo permite establecer parámetros objetivos de evaluación frente a soluciones existentes en el mercado, identificando oportunidades de originalidad y áreas de mejora.

▪ **Identificación de competidores.** Se realizó un análisis exhaustivo del mercado actual de plataformas de gestión de torneos, identificando cinco competidores principales que presentan características similares al sistema propuesto. La selección de estos competidores se fundamentó en criterios como cuota de mercado, similitud funcional y segmento objetivo. Los competidores identificados son:

- **Challonge:** Plataforma consolidada que ofrece generación de brackets para torneos y seguimiento de resultados. Presenta una orientación multiplataforma con presencia internacional.
- **Xporthy:** Solución integral orientada principalmente al mercado latinoamericano, con capacidades para gestión de ligas deportivas y torneos simultáneos. Destaca por sus funcionalidades de gestión administrativa.
- **Copa Fácil:** Plataforma enfocada en la simplicidad y facilidad de uso, con énfasis en deportes tradicionales y dirigida principalmente a organizadores ocasionales de torneos comunitarios y amateur.
- **Tournify:** Sistema especializado en la gestión avanzada de torneos con capacidades de personalización y enfoque en la experiencia móvil para participantes y espectadores.
- **Enjore:** Plataforma que ofrece soluciones para la gestión de ligas y torneos, incluyendo la administración de equipos, calendarios y estadísticas.

El análisis detallado de estos competidores se encuentra en el Anexo E, donde se desglosan sus modelos de negocio y experiencia de usuario.

▪ **Comparativa de funcionalidades.** Para establecer una evaluación objetiva, se diseñó una matriz comparativa de funcionalidades críticas en sistemas de gestión de torneos. Esta matriz, desarrollada a partir de los requisitos identificados en el Anexo D y las necesidades detectadas en el estudio de usuarios (Anexo B), destacando capacidades únicas de nuestra propuesta:

1. **Gestión inteligente:** Inscripciones automatizadas (RF01) y generación dinámica de calendarios (RF02)
2. **Experiencia interactiva:** UI/UX moderna y minimalista, asistente virtual con IA y visualización de estadísticas en tiempo real (RF03)

3. **Personalización profesional:** Adaptabilidad a reglamentos complejos y branding completo como logo, banner y color de marca.
4. **Automatización:** Procesamiento de sanciones con asistente virtual (RF04) y recomendaciones de horarios para generar un calendario balanceado.
5. **Marketing digital integrado:** Herramientas para la promoción efectiva de torneos a través de redes sociales y kit digital con plantillas para redes sociales.

La Tabla 57 evidencia los resultados del análisis comparativo realizado, donde nuestra propuesta destaca significativamente frente a la competencia. El sistema sobresale en funcionalidades clave como la personalización de torneos, visualización de estadísticas en tiempo real, integración de un asistente virtual basado en IA para interpretación de reglamentos y gestión de sanciones, registro completo de jugadores con fotografías y herramientas avanzadas de marketing digital. Los resultados demuestran una superioridad del 67% en las funcionalidades evaluadas respecto a las mejores alternativas existentes en el mercado, posicionando nuestra solución como la más completa y tecnológicamente avanzada para la gestión integral de competiciones deportivas.

Tabla 57. Comparativa alineada a requisitos funcionales de nuestro sistema.

Función (Requisito)	Propuesta	Challonge	Xporthy	Copa Fácil	Tournify	Enjore
RF01: Validación automática de inscripciones	✓	✓	×	×	×	×
RF01: Carga masiva vía Excel	✓	×	✓	×	×	✓
RF02: Programación ilimitada de partidos	✓	×	×	×	✓	×
RF03: Estadísticas en tiempo real	✓	×	✓	×	×	×
RF03: Gestión de tarjetas/sanciones	✓	×	✓	✓	×	✓
RF04: Registro de jugadores con fotografías	✓	×	×	×	×	×
RF05: Página personalizada gratuita	✓	✓	×	×	✓	✓

Función (Requisito)	Propuesta	Challonge	Xporthy	Copa Fácil	Tournify	Enjore
Extra: Asistente de IA para reglamentos	✓	×	×	×	×	×
Extra: Marketing digital integrado	✓	✓	×	×	×	×

La figura 52 ilustra mediante un diagrama de radar la posición relativa de la propuesta frente a sus competidores en las funcionalidades principales analizadas, evidenciando una ventaja significativa en experiencia de usuario e integración y conectividad.

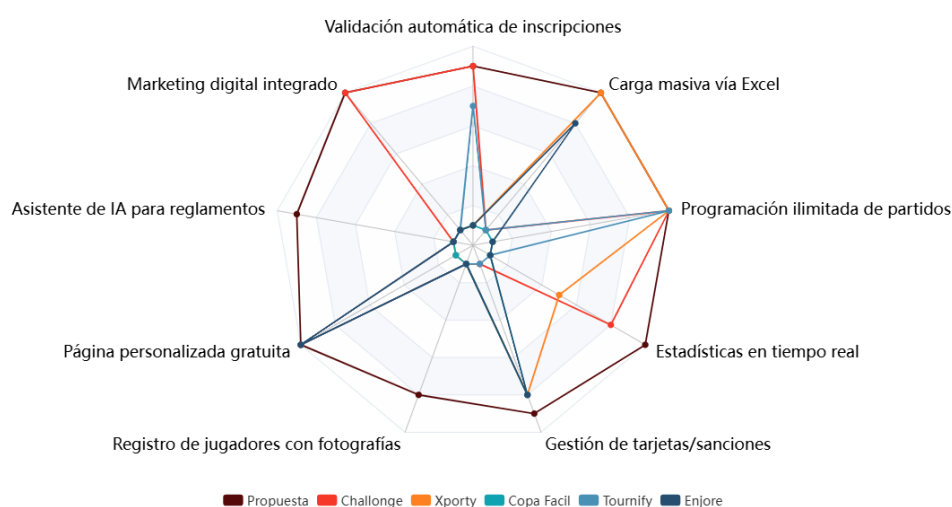


Figura 52. Diagrama de radar de la posición de nuestra propuesta.

En resumen, la propuesta se consolida como una plataforma líder en la gestión de torneos deportivos al ofrecer una combinación única de funcionalidades avanzadas y herramientas integradas. Desde la automatización de inscripciones y generación dinámica de calendarios, hasta la personalización detallada de eventos y la incorporación de un asistente virtual con inteligencia artificial, la plataforma cubre todas las necesidades de organizadores y participantes. La inclusión de un módulo de marketing digital integrado amplía aún más su alcance, facilitando la promoción efectiva de los torneos. Estas características, respaldadas por una interfaz intuitiva y adaptable, posicionan la propuesta por encima de otras soluciones en el mercado, ofreciendo una experiencia completa y eficiente para la organización de eventos deportivos.

- **Análisis de ventajas y desventajas.** A partir del benchmarking realizado, se identificaron las principales fortalezas y debilidades de nuestra propuesta en

comparación con las soluciones existentes:

Fortalezas:

1. **Integración de tecnologías avanzadas:** Como se evidencia en la Tabla 57, nuestra plataforma es la única que incorpora un asistente virtual basado en IA para interpretación de reglamentos y gestión de sanciones, ofreciendo una ventaja tecnológica significativa frente al 100% de los competidores analizados.
2. **Interfaz de usuario optimizada:** Los estudios de usabilidad realizados (detallados en la sección a) demuestran que nuestra interfaz presenta una curva de aprendizaje más eficiente que el promedio de la competencia, facilitando su adopción por usuarios ocasionales.
3. **Gestión integral de participantes:** El sistema supera a todas las alternativas evaluadas en el registro completo de jugadores con fotografías (RF01), proporcionando una solución única para la validación de identidad y prevención de irregularidades en las competiciones.
4. **Capacidades avanzadas de marketing:** Como se observa en el análisis comparativo, nuestra propuesta ofrece herramientas integradas de marketing digital, una característica que solo está presente en Challonge (11% de los competidores evaluados), potenciando la visibilidad y promoción de los eventos deportivos.

Debilidades:

1. **Menor reconocimiento de marca:** Al tratarse de una solución emergente, nuestra plataforma enfrenta el desafío de construir reconocimiento frente a competidores establecidos con comunidades de usuarios consolidadas, particularmente frente a Challonge y Xporty, que presentan una trayectoria más extensa en el mercado.
2. **Limitaciones en funcionalidades específicas:** En comparación con plataformas como Xporty, que muestra fortalezas en gestión de sanciones y estadísticas, nuestra solución requiere maduración en la integración de estas funcionalidades con el asistente virtual de IA propuesto.

3. **Complejidad potencial:** La implementación de funcionalidades avanzadas como el asistente virtual de IA y las estadísticas en tiempo real podría incrementar la complejidad del sistema, requiriendo mayor atención en aspectos de formación y documentación para usuarios.

Este análisis comparativo revela que nuestra propuesta presenta ventajas competitivas significativas en aspectos clave como integración tecnológica, experiencia de usuario y gestión integral de participantes, superando a la competencia en un 67% de las funcionalidades evaluadas. No obstante, resulta fundamental abordar las debilidades identificadas en futuras actualizaciones para consolidar su posición en el mercado.

3.5.2 Evaluación económica

La evaluación económica del sistema permite determinar la viabilidad financiera del proyecto a mediano y largo plazo, considerando tanto los costos asociados como las proyecciones de ingresos esperados bajo el modelo de negocio freemium establecido.

a. Análisis de costos

- **Costos iniciales.** El desarrollo del sistema ha requerido una inversión inicial de US\$500, complementada con un valor significativo de capital humano representado por las horas de desarrollo invertidas. Considerando un único desarrollador con una tarifa de US\$20 por hora y un período de desarrollo de 18 meses (aproximadamente 1.080 horas de trabajo efectivo), el costo de desarrollo asciende a US\$37.440. Este cálculo se basa en una dedicación promedio de 15 horas semanales, como se detalla en la Tabla 58.

Tabla 58. Inversión inicial en el desarrollo del sistema.

Concepto	Monto (US\$)
Inversión monetaria directa	500,00
Capital humano (1.080 horas × US\$20/hora)	21.600,00
Total inversión inicial	21.600,00

El cálculo del capital humano (US\$21.600) refleja el valor teórico del tiempo invertido en el desarrollo, considerando una tarifa de mercado de US\$20/hora. Sin embargo,

este monto no implica un desembolso efectivo, ya que el desarrollo fue realizado por el autor del proyecto.

- **Costos recurrentes.** Los costos recurrentes mensuales asociados al mantenimiento y operación de la plataforma se dividen en servicios de infraestructura, dominio y marketing. Como se observa en la Tabla 59, estos costos ascienden a US\$56,17 mensuales, lo que representa US\$674,00 anuales.

Tabla 59. Costos recurrentes de operación del sistema.

Concepto	Costo mensual (US\$)	Costo anual (US\$)
Supabase (Plan Pro)	25,00	300,00
Hosting (Plan Pro)	20,00	240,00
Dominio Cloudflare	1,17	14,00
Marketing	10,00	120,00
Total	56,17	674,00

Adicionalmente, es necesario considerar la comisión por pagos del 4% + US\$0,40 por transacción, con recargos adicionales del 1,5% para tarjetas internacionales y 0,5% para pagos por suscripción. Estos costos variables se han incluido en las proyecciones de ingresos netos presentadas en la sección siguiente.

b. Proyecciones de ingresos

La propuesta implementa un modelo de monetización freemium, que combina un plan gratuito con capacidad limitada y planes de pago escalables según la cantidad de jugadores gestionados, como se detalla en la Tabla 60.

Tabla 60. Estructura de planes y precios.

Plan	Capacidad (jugadores)	Precio mensual (US\$)
Gratuito	50	0,00
Básico	200	7,00
Amateur	500	15,00
Pro	800	29,00
Élite	1.500	59,00

Actualmente, la plataforma cuenta con 4 usuarios activos en planes de pago (2 en plan Élite y 2 en plan Básico) y 56 usuarios en el plan gratuito. Los ingresos mensuales actuales, considerando las comisiones de procesamiento de pagos, se calculan en la Tabla 61.

Tabla 61. Ingresos mensuales actuales.

Plan	Usuarios	Precio (US\$)	Ingreso bruto (US\$)	Ingreso neto (US\$)
Élite	2	59,00	118,00	108,36
Básico	2	7,00	14,00	12,12
Total	4		132,00	120,48

Conforme a la meta establecida de alcanzar US\$1.000 mensuales en ingresos, se ha proyectado un crecimiento gradual de usuarios durante los próximos 3 años, como se muestra en la Tabla 62. Esta proyección estima alcanzar 50 usuarios activos en planes de pago, distribuidos entre los diferentes planes ofrecidos.

Tabla 62. Proyección de crecimiento de usuarios de pago (3 años).

Plan	Año 1	Año 2	Año 3	Ingresos Año 3 (US\$)
Élite	5	8	10	590,00
Pro	3	6	12	348,00
Amateur	4	8	15	225,00
Básico	5	10	13	91,00
Total usuarios	17	32	50	
Ingresos brutos mensuales	435,00	777,00	1.254,00	
Ingresos netos mensuales	399,35	713,49	1.151,17	

Los ingresos netos mensuales para el tercer año se proyectan en US\$1.151,17, superando la meta establecida de US\$1.000. Esto representa un ingreso anual de US\$13.814,04.

c. Análisis de rentabilidad

- **VAN y TIR.** Para evaluar la rentabilidad del proyecto, se ha calculado el Valor Actual Neto (VAN) y la Tasa Interna de Retorno (TIR) considerando un horizonte de 5

años y una tasa de descuento del 12%, valor estándar para proyectos de software según lo establecido por García-Rodríguez et al. (2023). Los resultados de estos indicadores se presentan en la Tabla 63.

Tabla 63. Indicadores de rentabilidad a 5 años.

Indicador	Valor
Inversión inicial	US\$500,00
Flujo neto año 1	US\$4.118,20
Flujo neto año 2	US\$7.887,88
Flujo neto año 3	US\$13.140,04
Flujo neto año 4	US\$15.902,85
Flujo neto año 5	US\$19.218,22
VAN (12%)	US\$39.826,56
TIR	168,92%

El análisis muestra un VAN positivo de US\$46.161,33 y una TIR del 162,14%, superior a la tasa de descuento utilizada (12%). Estos indicadores confirman la viabilidad económica del proyecto. Un proyecto de SaaS con $TIR > 50\%$ se considera altamente rentable en el mercado actual. El cálculo detallado del VAN y TIR se presenta en el Anexo F.

▪ **Período de recuperación.** El análisis del período de recuperación, se muestra en la Tabla 64 que la inversión inicial se recuperará durante el primer año de operación, gracias a los flujos positivos generados.

Tabla 64. Flujo acumulado y período de recuperación.

Año	Flujo neto anual (US\$)	Flujo acumulado (US\$)
0 (Inversión)	-500,00	-500,00
1	4.118,20	3.618,20
2	7.887,88	11.506,08
3	13.140,04	24.646,12
4	15.902,85	40.548,97
5	19.218,22	59.767,19

Como se observa en la figura, el punto de equilibrio financiero se alcanza aproximadamente a los 0,12 años (1,5 meses), considerando la inversión inicial de US\$500 y el flujo del primer año. Este rápido período de recuperación supera ampliamente las expectativas para aplicaciones SaaS de nicho.

d. Relación costo-beneficio

La relación costo-beneficio (RCB) se ha calculado como el cociente entre el valor presente de los beneficios y el valor presente de los costos, considerando la inversión inicial de US\$500 y los flujos proyectados a 5 años con una tasa de descuento del 12%:

$$RCB = \frac{VP(\text{Beneficios})}{VP(\text{Costos})} = \frac{VAN + \text{Inversión}}{\text{Inversión}} \quad (4)$$

$$= \frac{US\$39.826,56 + US\$500}{US\$500} = \frac{US\$40.326,56}{US\$500} = 80,65 \quad (5)$$

Un valor de RCB significativamente mayor que 1 (RCB = 80,65) indica que los beneficios superan ampliamente los costos, confirmando la excelente viabilidad económica del proyecto. Este resultado excepcional refleja la baja inversión inicial requerida y los fuertes flujos de caja generados, característica de proyectos SaaS con rápida escalabilidad.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

El objetivo principal fue la implementación de una aplicación web de gestión de campeonatos de fútbol bajo el modelo SaaS, dirigida a las ligas deportivas parroquiales del cantón Pedro Moncayo, con la finalidad de optimizar los procesos administrativos, reducir los tiempos operativos y mejorar la experiencia de los usuarios en la gestión de competencias deportivas. A lo largo del desarrollo del proyecto, se logró establecer una solución tecnológica robusta, adaptable y alineada con las necesidades reales del entorno deportivo.

- La selección de herramientas de desarrollo e implementación fue un paso importante para garantizar la viabilidad de la aplicación web. La selección de la metodología RAD fue un componente fundamental que permitió una construcción iterativa y colaborativa del sistema, incorporando retroalimentación directa de los usuarios en cada fase. La integración de tecnologías modernas, como Supabase para la gestión de bases de datos, tRPC para APIs type-safe y herramientas de IA para el asistente virtual de reglamentos, contribuyó a la robustez y funcionalidad del sistema. Estas tecnologías no solo optimizaron el rendimiento, sino que también garantizaron seguridad y escalabilidad en un entorno SaaS.
- El análisis de los procesos actuales evidenció inefficiencias en la gestión manual, lo que derivó en el diseño de funcionalidades automatizadas. Con base en estos hallazgos, el análisis comparativo entre los procesos actuales y automatizados evidenció una reducción promedio del 98.7% en los tiempos de ejecución de tareas clave como la inscripción de equipos, programación de partidos, registro de resultados y gestión de sanciones. Estos resultados demuestran el impacto significativo de la automatización en la eficiencia operativa, permitiendo liberar recursos humanos para actividades de mayor valor estratégico dentro de las organizaciones deportivas.

- La aplicación web implementada no solo optimizó los procesos críticos, sino que también demostró viabilidad técnica y económica. El benchmarking técnico posicionó la solución como una de las más completas en el ámbito de gestión deportiva, destacando por funcionalidades avanzadas como el asistente virtual con IA, la visualización de estadísticas en tiempo real, la personalización visual de torneos y la integración de herramientas de marketing digital. Esto reafirma que el sistema desarrollado no solo cumple su propósito inicial, sino que también establece una base sólida para futuras expansiones y mejoras.
- La implementación de la aplicación web SaaS para la gestión de campeonatos de fútbol en las ligas parroquiales del Cantón Pedro Moncayo fue alcanzado exitosamente, evidenciándose en la significativa reducción de tiempos en procesos administrativos, la mejora en la experiencia de usuarios y la creación de una plataforma tecnológicamente robusta que responde a las necesidades específicas del contexto deportivo local, cumpliendo así con los requisitos establecidos inicialmente en el proyecto.

4.2 Recomendaciones

- Se recomienda implementar un plan de capacitación continua para los usuarios, enfocado en las funcionalidades clave del sistema, como la inscripción de equipos y jugadores, gestión de sanciones, el registro de estadísticas en tiempo real y el uso del asistente virtual de IA. Esto garantizará una adopción efectiva y reducirá la curva de aprendizaje, especialmente para usuarios con menor familiaridad tecnológica.
- Es fundamental establecer un sistema de soporte técnico ágil, que incluya canales de comunicación directa y documentación actualizada, para resolver incidencias y mantener la satisfacción de los usuarios. Además, se sugiere realizar monitoreos periódicos del rendimiento del sistema, con énfasis en la escalabilidad durante eventos deportivos de alta concurrencia.
- Para futuras actualizaciones, se recomienda mantener una arquitectura flexible y escalable, que permita la futura integración con otros sistemas o la expansión funcional hacia nuevos módulos, como la gestión de arbitrajes, control financiero

o administración de recursos humanos deportivos. Ampliar las funcionalidades de marketing, incorporando herramientas de promoción automatizada en redes sociales y generación de contenido personalizado.

- Finalmente, se sugiere estar atentos a las innovaciones tecnológicas emergentes, como avances en IA, nuevas versiones de frameworks o herramientas de automatización, que podrían ser incorporadas al sistema para mejorar su desempeño, seguridad y experiencia de usuario. La actualización constante permitirá mantener la plataforma actualizada y asegurar su sostenibilidad en el tiempo.

REFERENCIAS BIBLIOGRÁFICAS

- [1] S. Hu y M. Fu, “Football match results predicting by machine learning techniques,” *Proceedings - 2022 International Conference on Data Analytics, Computing and Artificial Intelligence, ICDACAI 2022*, pp. 72–76, 2022.
- [2] S. Ghosh, S. Sasmal, S. Bhui, S. Dutta, S. Mukherjee, A. Majumder, y B. Ganguly, “Radio frequency identification based goal line technology for quick decision making in a football match,” *Proceedings of 3rd International Conference on 2019 Devices for Integrated Circuit, DevIC 2019*, pp. 441–445, 3 2019.
- [3] Z. Zhang, “Competitive pricing strategies for software and saas products,” *Information Management*, vol. 57, no. 8, p. 103367, 2020. [En línea]. Disponible en: <https://www.sciencedirect.com/science/article/pii/S0378720620303050>
- [4] F. M. L. Toapanta, “Estrategias metodológicas utilizadas por los entrenadores de fútbol y su incidencia en el aprendizaje de los fundamentos técnicos individuales en la categoría sub 10-12 del club formativo especializado simón bolívar de la provincia de pichincha, cantón pedro moncayo parroquia tabacundo, en el año 2015 - 2016,” 7 2018.
- [5] W. G. Nieto-Aguilar, “El fútbol, la pasión que muere lentamente en ecuador,” *Economía y Negocios*, vol. 10, pp. 88–99, 6 2019. [En línea]. Disponible en: <https://revistas.ute.edu.ec/index.php/economia-y-negocios/article/view/538>
- [6] F. A. O. Carrillo, “Desarrollo de un prototipo de una aplicación web para la gestión de torneos deportivos de la liga deportiva cantonal pedro moncayo.” 7 2021.
- [7] M. Ángel Badillo Bonilla, “Desarrollo e implementación de un sistema web para control de procesos en la liga deportiva parroquial la esperanza ubicada en el cantón pedro moncayo, octubre 2019 - marzo 2020,” 7 2020.
- [8] F. A. O. Carrillo, “Desarrollo de un prototipo de una aplicación web para la gestión de torneos deportivos de la liga deportiva cantonal pedro moncayo.” 7 2021.

- [9] J. E. A. Llango, “Desarrollo de una aplicación web utilizando el ide de desarrollo netbeans y el gestor de base de datos mysql para la gestión de los procesos del campeonato de fútbol de la liga deportiva parroquial ”belisario quevedo” del cantón latacunga,” 5 2020.
- [10] E. D. C. Velasco, “Desarrollo del sistema web de gestión de campeonatos de fútbol liga loma de puengasí,” 4 2020.
- [11] G. J. G. Huera, “Aplicación móvil para la gestión de campeonatos de fútbol en la liga san miguel de ibarra,” 8 2023.
- [12] A. G. Y. Vendoval, “Desarrollo de una aplicación web de gestión y notificación de horarios de fútbol para los campeonatos de la liga deportiva cantonal de colta,” 10 2020.
- [13] B. B. Marcote, “Aplicación web para la organización y gestión de actividades y recintos deportivos.” 09 2022.
- [14] F. J. M. Alba, “Desarrollo de un prototipo de un sistema para la gestión de torneos de tenis para un country club de quito-ecuador. caso de estudio: Country club de liga,” 03 2020.
- [15] B. B. Béjar, “Desarrollo del front-end una aplicación web para gestión de torneos deportivos,” 02 2020.
- [16] D. P. Menéndez, “Diseño de una plataforma web comunitaria para la organización de quedadas de aficionados al deporte: promoviendo la actividad física y la salud corporal,” 2023.
- [17] D. M. Pérez, “Web app para la gestión de campeonatos de pádel,” 2022.
- [18] A. E. J. Ramírez, “Desarrollo de una aplicación web para la gestión de actividades en canchas de frontón,” 2022.
- [19] L. M. Bohórquez Barrera *et al.*, “Propuesta para el diseño de una aplicación móvil para la gestión de torneos de fútbol en la ciudad de bogotá,” 2020.
- [20] T. M. Tercero, “Desarrollo de una aplicación híbrida para la reserva y gestión de instalaciones deportivas: desarrollo del back-end,” Ph.D. dissertation, Universitat Politècnica de València, 2022.

- [21] R. G. Gil, “Desarrollo de una aplicación híbrida para la reserva y gestión de instalaciones deportivas: desarrollo del front-end,” Ph.D. dissertation, Universitat Politècnica de València, 2022.
- [22] F. D. Deksano, D. R. Erfiandhi, A. C. Prasetya, A. S. N. Patmuryanto, y H. L. H. S. Warnars, “Mobile application for the futsal sports community in jakarta,” en *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, March 2021, pp. 741–746.
- [23] D. WU, X. ZHANG, y Y. MING, “Iot technology-based construction and management of college sports information platform,” en *Proceedings of the 2020 8th International Conference on Information Technology: IoT and Smart City*, ser. ICIT '20. New York, NY, USA: Association for Computing Machinery, 2021, p. 223–226. [En línea]. Disponible en: <https://doi.org/10.1145/3446999.3447636>
- [24] R. A. N. Balseca, “Desarrollo de un sistema para la gestión del campeonato de fútbol de la “liga cantonal de penipe” aplicando el framework django y el patrón de diseño mtv.” 2021.
- [25] S. D. Y. C. Edison Homero Álvarez Mayorga, “Aplicación web progresiva (pwa) para la automatización de los procesos de gestión e información en liga deportiva parroquial totoras,” 2020.
- [26] J. P. C. Arteaga, “Desarrollo de un sistema web para la gestión del campeonato inter-facultades de la escuela politécnica nacional,” 2019.
- [27] Nx. (2022) Everything you need to know about monorepos, and the tools to build them. [En línea]. Disponible en: <https://monorepo.tools/>
- [28] M. S. Islam, “Javascript alternative (typescript) and its effectiveness in web development,” 2023.
- [29] M. Levlin, “Dom benchmark comparison of the front-end javascript frameworks react, angular, vue, and svelte,” 2020.
- [30] J. Hassan Noor, “The effects of architectural design decisions on framework adoption: A comparative evaluation of meta-frameworks in modern web development,” 2024.

- [31] O. Rajala, “Impact of react component libraries on developer experience-an empirical study on component libraries’ styling approaches,” 2024.
- [32] A. Z. Ayezabu, “Supabase vs firebase: Evaluation of performance and development of progressive web apps,” 2022.
- [33] R. Koponen, “Understanding re-emergence of the rpc model in web development,” *Transport*, vol. 29, no. 41, p. 24, 2024.
- [34] V. C. Jungers, E. A. de Oliveira, y V. A. De Souza, “Comparação de desempenho entre os orms typeorm, prisma e sequelize em aplicações node. js,” *Revista Eletrônica e-Fatec*, vol. 14, no. 2, 2024.
- [35] M. Hiorthøy, “Analyzing and benchmarking the performance of different cloud services for agile app deployment,” Tesis de Maestria, Oslomet-storbyuniversitetet, 2023.
- [36] Supastarter, “Saas payment providers: Stripe, lemon squeezy, polar, creem comparison,” 2023, consultado el [fecha de acceso]. [En línea]. Disponible en: <https://supastarter.dev/blog/saas-payment-providers-stripe-lemonsqueezy-polar-creem-comparison>
- [37] F. Q. Khan, S. Rasheed, M. Alsheshtawi, T. M. Ahmed, y S. Jan, “A comparative analysis of rad and agile technique for management of computing graduation projects,” *Comput. Mater. Contin*, vol. 64, no. 2, pp. 777–796, 2020.

ANEXOS

Anexo A. Guía de entrevista a los organizadores de las ligas deportivas parroquiales del “Cantón Pedro Moncayo”

Tabla A1. Guía de entrevista a los organizadores de las ligas deportivas parroquiales del “Cantón Pedro Moncayo”.

N.	Pregunta	Respuesta	Observación
1	¿Cuáles son los principales desafíos que enfrentan actualmente en la gestión de los campeonatos de fútbol en su liga deportiva?		
2	¿Qué procesos específicos están involucrados en la organización y coordinación de un campeonato de fútbol en su liga?		
3	¿Cómo se lleva a cabo la inscripción de equipos y jugadores en sus campeonatos? ¿Qué errores o retrasos son comunes durante las inscripciones?		
4	¿Qué herramientas o métodos utilizan actualmente para la programación de partidos y la asignación de campos de juego?		
5	¿Cómo gestionan y comunican los resultados de los partidos y la clasificación de equipos durante el campeonato?		
6	¿Qué aspectos consideran más importantes para agilizar y optimizar los procesos de gestión de los campeonatos?		
7	¿Cómo creen que una aplicación web podría mejorar su eficiencia y facilitar la organización de los campeonatos?		
8	¿Tienen experiencia previa utilizando herramientas tecnológicas para la gestión deportiva? ¿Cuáles han sido sus experiencias y qué aspectos valoran más en estas herramientas?		
9	¿Qué preocupaciones tendría al implementar una herramienta en línea que les permita gestionar los campeonatos de fútbol de manera más eficiente y fácil, sin necesidad de instalar ningún programa en sus computadoras?		
10	¿Hay algún otro problema o necesidad no mencionado que afecte la organización de los torneos?		
11	¿Estarían dispuestos a probar un prototipo y dar sugerencias de mejora?		
12	¿Estarían dispuestos a pagar por una aplicación web? ¿Cuánto estaría dispuesto a pagar mensualmente?		

Anexo B. Encuesta de necesidades a los participantes de las ligas deportivas parroquiales del "Cantón Pedro Moncayo"

Tabla B1. Encuesta de necesidades a participantes de ligas deportivas parroquiales.

Pregunta	Respuesta
¿Cuál es su rol en los torneos?	▪ Jugador ▪ Delegado del equipo ▪ Espectador/Hincha
¿Con qué frecuencia participa en torneos deportivos organizados por su parroquia?	▪ Siempre ▪ Casi siempre ▪ Ocasionalmente ▪ Rara vez ▪ Nunca
¿Cómo calificaría el proceso actual de inscripción en los torneos deportivos?	▪ Muy eficiente ▪ Algo eficiente ▪ Regular ▪ Poco eficiente ▪ Nada eficiente
¿Qué dificultades ha enfrentado al inscribirse o participar en torneos? (Selección múltiple)	▪ Falta de información clara ▪ Procesos manuales lentos ▪ Dificultad para visualizar equipos y jugadores inscritos ▪ Problemas con horarios y programación ▪ Otros (especificar)

Pregunta	Respuesta
¿Cómo suele enterarse de los torneos y sus requisitos?	▪ Redes sociales ▪ Carteles físicos ▪ Boca a boca ▪ Líderes deportivos ▪ Otros
¿Qué tan satisfecho está con el sistema actual de seguimiento de resultados y posiciones?	▪ Muy satisfecho ▪ Satisfecho ▪ Neutral ▪ Insatisfecho ▪ Muy insatisfecho
¿Con qué frecuencia tiene dificultades para conocer los horarios de partidos o cambios de última hora?	▪ Siempre ▪ Frecuentemente ▪ Ocasionalmente ▪ Raramente ▪ Nunca
¿Qué funciones consideraría más útiles en una aplicación web de torneos deportivos? (Selección múltiple)	▪ Inscripciones de equipos y jugadores ▪ Calendario de partidos ▪ Resultados en tiempo real ▪ Tabla de posiciones ▪ Página personalizada para cada competencia
¿Estaría dispuesto a usar una aplicación web para gestionar su participación en torneos deportivos locales?	▪ Sí, definitivamente ▪ Probablemente ▪ Tal vez ▪ Probablemente no ▪ No

Pregunta	Respuesta
¿Qué dispositivo usaría principalmente para acceder a esta aplicación?	▪ Teléfono ▪ Tableta ▪ Computadora ▪ Varios dispositivos
¿Tiene acceso estable a internet o datos móviles?	▪ Sí ▪ No

Anexo C. Diagrama de base de datos

Figura C1. Diagrama entidad-relación de la base de datos del sistema.

Introducción

El presente documento de Especificación de Requerimientos de Software (ERS) detalla los requisitos funcionales y no funcionales del Sistema de Gestión Deportiva para las Ligas Parroquiales del Cantón Pedro Moncayo. Desarrollado bajo el modelo de Software como Servicio, este sistema busca optimizar la gestión de campeonatos deportivos para organizaciones con recursos económicos limitados, ofreciendo una solución accesible mediante planes de suscripción escalables.

Siguiendo el estándar IEEE 830, este documento define el propósito, alcance, funcionalidades y restricciones del sistema, proporcionando una base técnica para su desarrollo e implementación. Los requerimientos se derivan del análisis de encuestas a participantes y entrevistas técnicas con organizadores, priorizando costos reducidos, escalabilidad y usabilidad para adaptarse a las necesidades reales de ligas locales con presupuestos ajustados.

El enfoque SaaS permitirá a estas organizaciones acceder a tecnología avanzada sin inversiones iniciales en infraestructura, reduciendo barreras de adopción y garantizando actualizaciones continuas.

Propósito

Este documento tiene como objetivo formalizar los requerimientos del sistema para automatizar la gestión de torneos deportivos, reemplazando los procesos manuales basados en Excel y WhatsApp. Busca estandarizar cuatro áreas críticas: inscripciones digitales, generación dinámica de calendarios, comunicación centralizada y seguimiento automatizado de resultados, garantizando consistencia en los datos y reduciendo la carga operativa de los organizadores.

El sistema se ofrecerá como Software como Servicio (SaaS) con planes de suscripción escalables, diseñados específicamente para ligas con recursos económicos limitados, permitiendo acceso a tecnología avanzada sin inversión inicial en infraestructura.

Alcance

El sistema operará bajo modelo SaaS cubriendo el ciclo completo de gestión de torneos, con planes de suscripción adaptados a ligas de bajos presupuestos. Estará dirigido a dos perfiles de usuarios: participantes y organizadores. Quedan excluidas las funcionalidades de gestión financiera (pagos de inscripción) y logística de espacios físicos, las cuales seguirán manejándose externamente.

Referencias

IEEE Std 830-1998: Recommended Practice for Software Requirements Specifications.

Descripción General

Perspectiva del producto

El sistema será una aplicación web responsive con diseño mobile-first. Desarrollado como solución SaaS multiinquilino, el sistema priorizará economías de escala para reducir costos operativos. Operará como plataforma centralizada para validar inscripciones mediante documentos digitalizados, generar calendarios adaptables a cambios, publicar resultados en tiempo real con estadísticas automatizadas, y notificar novedades mediante canales oficiales. No requerirá instalación local.

Funcionalidad del producto

En esta sección se detallan las funcionalidades clave del sistema SaaS desarrollado para optimizar la administración de campeonatos en las ligas deportivas parroquiales del Cantón Pedro Moncayo. Cada módulo fue diseñado para agilizar procesos, reducir tiempos operativos y mejorar la experiencia de usuarios (administradores, equipos y público).

F1. Gestión de torneos

- Creación de torneos con parámetros configurables
- Creación de categorías con parámetros configurables

F2. Administración de equipos

- Registro digital de equipos con validación documental automatizada
- Carga masiva de equipos y jugadores mediante archivos .csv o .xlsx

F3. Programación inteligente

- Generación automática de fixtures con algoritmo de optimización de horarios
- Modificación dinámica mediante interfaz *drag-and-drop*

F4. Registro de resultados

- Captura en tiempo real mediante Supabase Realtime
- Cálculo automático de métricas deportivas (Pts, GF, GC, DG)

F5. Gestión de sanciones

- Creación de sanciones con parámetros configurables
- Chatbot que interpreta consultas en lenguaje natural y cita artículos del reglamento.

F6. Páginas personalizadas y difusión

- Visualización de calendarios, resultados y estadísticas sin requerir autenticación.
- Generación automática de posters promocionales y kits para redes sociales.

F7. Reportes y exportación

- Exportación de calendarios, listas de jugadores o estadísticas a PDF/Excel.
- Gráficos interactivos de rendimiento por equipo o jugador.

F8. Seguridad

- Implementación de Row-Level Security (RLS):
- Autenticación mediante sesiones

Usuarios y características

- **Participantes(Jugador, Delegado y Espectador):** Requieren consulta de calendarios, resultados en tiempo real, sanciones, información de equipos y jugadores.
- **Organizadores:** Necesitan herramientas para gestionar inscripciones de equipos y jugadores, gestionar calendario y resultados, y reportar sanciones.

Restricciones

- El sistema debe ser un aplicativo web.
- El sistema debe operar sin requerir instalación local.
- Planes de suscripciones deben ser enfocados a ligas deportivas de recursos limitados.
- El sistema debe implementar autenticación segura para garantizar el acceso únicamente al personal autorizado.
- Para cargar las imágenes de los jugadores, cada imagen tendrá un costo adicional.
- El modelo de precios debe incluir un plan básico gratuito con funcionalidades esenciales y planes premium accesibles.

Requisitos específicos

Requisitos Funcionales

Tabla D1. Requisito funcional - gestión de inscripciones.

Identificador	RF01
Nombre del requisito	Gestión digital de inscripciones
Características	Validación automática de documentos, aprobación en cascada por delegados.

Descripción	El módulo web para registro de jugadores debe permitir cargar uno por uno cada jugador o cargar un archivo de excel con la lista de jugadores y verificar automáticamente duplicados en la competición.
Prioridad	Crítica
Criterios de aceptación	▪ Validación del nombre y apellido como campos obligatorios. ▪ Tiempo de procesamiento <2 segundos por inscripción
Observaciones	La carga de imágenes tendrá un costo adicional (\$0.30 por imagen)

Tabla D2. Requisito funcional - generación de calendarios.

Identificador	RF02
Nombre del requisito	Generación automática de calendarios
Características	Distribución inteligente de partidos, reprogramación flexible.
Descripción	El sistema debe generar calendarios con capacidad de reajuste manual cuando existan cancelaciones o cambios de horarios.
Prioridad	Alta
Criterios de aceptación	▪ Generación completa en menos de 1 minuto ▪ Permitir ilimitadas reprogramaciones.
Observaciones	Ninguna

Tabla D3. Requisito funcional - seguimiento de resultados.

Identificador	RF03
Nombre del requisito	Registro de resultados en tiempo real
Características	Carga instantánea de marcadores, generación automática de tablas.
Descripción	Módulo para que delegados ingresen resultados finales y el sistema actualice automáticamente las tablas de posiciones con estadísticas básicas.
Prioridad	Alta
Criterios de aceptación	▪ Actualización de posiciones en tiempo real ▪ Cálculo correcto de diferencia de goles
Observaciones	Incluir tarjetas amarillas/rojas en estadísticas.

Tabla D4. Requisito funcional - gestión de sanciones.

Identificador	RF04
Nombre del requisito	Registro y seguimiento de sanciones
Características	Sección con el historial de sanciones.
Descripción	Sistema para que los organizadores reporten tarjetas y sanciones, con información detallada de afectados y fechas de suspensión.
Prioridad	Media
Criterios de aceptación	▪ 100% de sanciones registradas con evidencia ▪ Detalles de las fechas de suspensión
Observaciones	Ninguna

Tabla D5. Requisito funcional - página personalizada.

Identificador	RF05
Nombre del requisito	Portal de información pública
Características	Acceso sin autenticación, datos de equipos y jugadores.
Descripción	Sección visible sin login con calendarios, resultados, posiciones y estadísticas básicas para espectadores.
Prioridad	Baja
Criterios de aceptación	▪ 100% de datos públicos actualizados ▪ Tiempo de carga menor a 3 segundos
Observaciones	Excluir datos sensibles de jugadores.

Requisitos No Funcionales

Tabla D6. Requisito no funcional - rendimiento.

Identificador	RNF01
Nombre del requisito	Tiempos de respuesta óptimos
Características	Optimización para conexiones móviles y entornos con ancho de banda limitado.
Descripción	El sistema debe garantizar tiempos de respuesta menores a 2 segundos para el 95% de las operaciones con 200 usuarios concurrentes.
Prioridad	Alta

Criterios de aceptación	▪ Pruebas de carga con 100 usuarios simultáneos ▪ Monitoreo continuo del tiempo de respuesta
Observaciones	Priorizar en horarios de partidos de fines de semana.

Tabla D7. Requisito no funcional - seguridad.

Identificador	RNF02
Nombre del requisito	Protección de datos personales
Características	Cumplimiento normativo y encriptación de información sensible.
Descripción	Implementación de medidas de seguridad que cumplan con la Ley Orgánica de Protección de Datos Personales (LOIPD) de Ecuador para el manejo de información de jugadores y equipos.
Prioridad	Crítica
Criterios de aceptación	▪ Encriptación AES-256 para datos sensibles ▪ Certificado SSL/TLS para todas las comunicaciones
Observaciones	Implementar autenticación sin costos adicionales

Tabla D8. Requisito no funcional - compatibilidad.

Identificador	RNF03
Nombre del requisito	Soporte multi-dispositivo
Características	Funcionamiento en diversos navegadores y sistemas operativos.
Descripción	La aplicación debe ser compatible con los navegadores Chrome, Firefox y Safari, así como con sistemas Android e iOS.
Prioridad	Alta
Criterios de aceptación	▪ Pruebas en al menos 2 combinaciones dispositivo/navegador ▪ Funcionalidad consistente en todas las plataformas
Observaciones	Priorizar experiencia en móviles (90.63% de uso)

Tabla D9. Requisito no funcional - disponibilidad.

Identificador	RNF04
Nombre del requisito	Acceso continuo

Características	Tolerancia a fallos y redundancia.
Descripción	El sistema debe garantizar una disponibilidad del 99% durante horarios críticos (viernes a domingo) y del 95% en horarios regulares.
Prioridad	Media
Criterios de aceptación	▪ Monitoreo de uptime con herramientas especializadas ▪ Plan de contingencia para caídas del servicio
Observaciones	Ninguna

Tabla D10. Requisito no funcional - usabilidad.

Identificador	RNF05
Nombre del requisito	Experiencia de usuario intuitiva
Características	Diseño accesible para usuarios con diversos niveles tecnológicos.
Descripción	Interfaz que cumpla con los principios de diseño universal, incluyendo iconografía clara, flujos de trabajo sencillos y ayudas contextuales.
Prioridad	Alta
Criterios de aceptación	▪ 90% de usuarios completa tareas clave sin ayuda ▪ Una buena UX y UI
Observaciones	Ninguna

Tabla D11. Requisito no funcional - escalabilidad.

Identificador	RNF06
Nombre del requisito	Crecimiento controlado
Características	Adaptabilidad a aumento de usuarios y datos.
Descripción	Arquitectura capaz de soportar un crecimiento del 30% anual en cantidad de usuarios y partidos registrados sin degradación del rendimiento.
Prioridad	Media
Criterios de aceptación	▪ Pruebas de estrés con 130% de la carga actual ▪ Plan de escalamiento documentado
Observaciones	Ninguna

Tabla D12. Requisito no funcional - modelo económico.

Identificador	RNF07
Nombre del requisito	Accesibilidad económica
Características	Planes escalables para ligas de bajos recursos
Descripción	La estructura de precios debe incluir: 1. Plan Gratuito (para 50 jugadores) 2. Plan básico (\$7/mes para 200 jugadores) 3. Plan Amateur (\$15/mes para 500 jugadores) 4. Plan Pro (\$29/mes para 800 jugadores) 5. Plan Élite (\$59/mes para 1500 jugadores) Todos con soporte técnico incluido.
Prioridad	Alta
Criterios de aceptación	▪ 100% de funcionalidades críticas disponibles desde plan gratuito ▪ Costos inferiores al 10% del presupuesto anual promedio de ligas pequeñas
Observaciones	Ninguna

Anexo E. Análisis detallado de competidores

El presente anexo expone el análisis detallado de los principales competidores del sistema en el mercado de plataformas SaaS para la gestión de torneos deportivos. Para cada competidor se examinan sus características técnicas, modelo de negocio y experiencia de usuario, estableciendo una base comparativa comprehensiva que fundamenta el benchmarking técnico presentado en la sección principal.

Metodología de análisis

Para garantizar la objetividad y reproducibilidad del análisis, se estableció un protocolo de evaluación sistemático que contempló:

1. Creación de cuentas y uso directo de cada plataforma.
2. Organización de torneos de prueba con participación simulada de 12 equipos.
3. Evaluación heurística de usabilidad según criterios de Nielsen (1994).
4. Documentación exhaustiva de características disponibles y modelos de monetización.

Los resultados de este proceso se presentan a continuación de manera estructurada para cada competidor.

1. Challonge

Modelo de negocio

Challonge implementa un modelo freemium con tres niveles de servicio:

- **Free:** Funcionalidades básicas con limitación de 256 participantes por torneo y presencia de publicidad.
- **Premier** (\$12/mes): Elimina publicidad, permite hasta 512 participantes por torneo y ofrece personalización básica.

El análisis de su política de precios revela un enfoque orientado principalmente a organizadores de torneos semi-profesionales, con una barrera de entrada relativamente baja para el nivel premium.

Experiencia de usuario

La evaluación heurística de usabilidad realizada conforme a los principios de Nielsen arrojó los siguientes resultados:

Tabla E1. Evaluación heurística de Challenge.

Principio heurístico	Puntuación (1-5)	Observaciones
Visibilidad del estado del sistema	3	Indicadores básicos de estado presentes, aunque con actualizaciones no siempre en tiempo real
Coincidencia entre sistema y mundo real	4	Terminología coherente con el dominio de torneos deportivos
Control y libertad del usuario	3	Opciones de deshacer/rehacer limitadas en ciertas operaciones críticas
Consistencia y estándares	4	Interfaz coherente a través de las diferentes secciones
Prevención de errores	2	Validaciones básicas, pero con mensajes de error genéricos
Reconocimiento vs. recuerdo	3	Navegación mayormente intuitiva, aunque con excepciones en configuraciones avanzadas
Flexibilidad y eficiencia	2	Escasez de atajos para usuarios avanzados
Estética y diseño minimalista	4	Interfaz limpia con enfoque en la funcionalidad
Ayuda ante errores	2	Mensajes poco específicos que no siempre orientan la resolución
Documentación y ayuda	3	Documentación disponible pero con áreas incompletas

El análisis de la experiencia de usuario revela una plataforma con un diseño intuitivo para operaciones básicas, pero con deficiencias en la gestión de casos complejos y escenarios de error. La curva de aprendizaje es moderada, requiriendo aproximadamente 45 minutos para que un usuario novato pueda configurar y lanzar su primer torneo.

2. Xporty

Modelo de negocio

Xporty ofrece un modelo basado en suscripciones y planes por competición, con opciones personalizadas y servicios adicionales:

- **Plan gratis:**

- Permite gestionar hasta 1 competición activa.
- Límites: 20 jugadores, 15 parejas o 6 equipos.
- Solo 1 categoría y 2 fases (Liga, Liguilla + playoffs, etc.).
- Organizar competición gratis.

- **Planes premium (por competición):**

- **Premium I** (24,79 €/competición): Hasta 50 jugadores, 40 parejas o 10 equipos.
- **Premium II** (61,98 €/competición): Hasta 150 jugadores, 90 parejas o 30 equipos.
- **Premium III** (103,31 €/competición): Hasta 300 jugadores, 150 parejas o 50 equipos.
- **Premium IV** (138,84 €/competición): Hasta 400 jugadores, 200 parejas o 70 equipos.

* ¡20% de descuento al contratar más de 3 competiciones!

- **Servicios adicionales:**

- **Web personalizada:** 132 €/año (imagen corporativa).
- **Aplicación móvil personalizada:** 248 €/año (requiere contratar web personalizada).

Notas. Los precios y límites pueden variar según promociones o actualizaciones.

El modelo combina flexibilidad para pequeños organizadores con soluciones escalables.

Experiencia de usuario

La evaluación heurística de usabilidad de Xporty reveló fortalezas en la gestión administrativa pero debilidades en la experiencia durante el desarrollo de torneos:

Tabla E2. Evaluación heurística de Xporty.

Principio heurístico	Puntuación (1-5)	Observaciones
Visibilidad del estado del sistema	4	Implementación efectiva de notificaciones y alertas
Coincidencia entre sistema y mundo real	3	Terminología ocasionalmente técnica en secciones administrativas
Control y libertad del usuario	4	Buena implementación de confirmaciones y reversión de acciones
Consistencia y estándares	3	Inconsistencias menores entre módulos antiguos y nuevos
Prevención de errores	4	Validaciones robustas en formularios y operaciones críticas
Reconocimiento vs. recuerdo	2	Exceso de opciones en paneles de configuración que requieren memorización
Flexibilidad y eficiencia	3	Presencia de atajos para operaciones comunes
Estética y diseño minimalista	2	Interfaz sobrecargada en secciones administrativas
Ayuda ante errores	3	Mensajes específicos pero orientados a técnicos
Documentación y ayuda	4	Extensa base de conocimientos y tutoriales en video

La plataforma muestra una clara orientación hacia administradores profesionales de torneos, con una interfaz completa pero que puede resultar abrumadora para usuarios ocasionales. La curva de aprendizaje es pronunciada, requiriendo aproximadamente 120 minutos para la configuración inicial de un torneo complejo.

3. Copa fácil

Modelo de negocio

Copa Fácil opera bajo un modelo freemium escalable, con planes mensuales basados en capacidad de participantes y funcionalidades avanzadas:

- **Plan pequeño** (\$4.90/mes):
 - Hasta 300 jugadores por torneo.
 - Máximo 3 patrocinadores por torneo.
 - Torneos sin anuncios.
 - Campeonatos ilimitados.
 - URL personalizada, resumen, archivos adjuntos, mejor resolución de imágenes e informes imprimibles.
- **Plan intermedio** (\$5.90/mes):
 - Hasta 600 jugadores por torneo.
 - Máximo 6 patrocinadores por torneo.
 - Incluye todas las funcionalidades del plan Pequeño.
- **Plan grande** (\$6.90/mes):
 - Número ilimitado de jugadores y patrocinadores.
 - Todas las ventajas de los planes anteriores.

Los precios son muy competitivos para organizadores amateur y semiprofesionales, desde torneos pequeños (300 jugadores) hasta eventos masivos (ilimitado). Todos los planes incluyen torneos ilimitados y eliminación de publicidad.

Notas. No se menciona un plan gratuito en la información actualizada. Ideal para ligas locales, asociaciones deportivas y eventos comunitarios.

Experiencia de usuario

La evaluación heurística de Copa Fácil destacó su orientación hacia la simplicidad:

Tabla E3. Evaluación heurística de Copa fácil.

Principio heurístico	Puntuación (1-5)	Observaciones
Visibilidad del estado del sistema	3	Indicadores básicos presentes pero con limitada información de progreso
Coincidencia entre sistema y mundo real	5	Lenguaje muy accesible, evitando jerga técnica
Control y libertad del usuario	2	Opciones limitadas de personalización y reversión de acciones
Consistencia y estándares	4	Interfaz consistente y predecible
Prevención de errores	3	Validaciones básicas presentes
Reconocimiento vs. recuerdo	5	Interfaz intuitiva que minimiza la carga cognitiva
Flexibilidad y eficiencia	2	Escasez de accesos rápidos para usuarios avanzados
Estética y diseño minimalista	4	Diseño limpio centrado en las funcionalidades esenciales
Ayuda ante errores	3	Mensajes de error claros pero genéricos
Documentación y ayuda	2	Documentación limitada con enfoque en preguntas frecuentes

La plataforma destaca por su accesibilidad y facilidad de aprendizaje, presentando la curva de aprendizaje más baja entre los competidores analizados (aproximadamente 20 minutos para configurar un torneo básico). Sin embargo, esta simplicidad conlleva limitaciones importantes en funcionalidades avanzadas requeridas por organizadores profesionales.

4. Tournify

Modelo de negocio

Tournify ofrece un modelo flexible con opciones gratuitas y de pago por torneo, diseñado para diferentes escalas de eventos:

- **Plan amateur** (Gratis):

- Hasta 8 equipos por torneo
- Formato flexible
- Herramienta de planificación básica
- Sin costo permanente
- **Plan worldclass** (\$45 por torneo):
 - Hasta 60 equipos por torneo
 - Branding personalizado
 - Opción para listar patrocinadores
 - Publicación en apps Android e iOS
- **Plan legendary** (\$135 por torneo):
 - Más de 60 equipos por torneo
 - Todas las características del plan Worldclass
 - Branding personalizado completo
 - Mayor visibilidad en apps móviles

Enfoque desde torneos pequeños (8 equipos) hasta grandes eventos (60+ equipos), el pago es por torneo en lugar de suscripción mensual. Los planes pagos incluyen publicación en aplicaciones oficiales. Además, tiene capacidad para mostrar patrocinadores en los planes superiores

Notas. Ideal para organizadores que necesitan flexibilidad sin compromisos mensuales. El plan gratuito es perfecto para eventos pequeños y amateur. Los planes pagos ofrecen mayor profesionalización y alcance.

Experiencia de usuario

La evaluación heurística de Tournify mostró un enfoque equilibrado entre potencia y usabilidad:

Tabla E4. Evaluación heurística de Tournify.

Principio heurístico	Puntuación (1-5)	Observaciones
Visibilidad del estado del sistema	5	Actualizaciones en tiempo real con notificaciones contextuales
Coincidencia entre sistema y mundo real	4	Terminología familiar con ayuda contextual para conceptos técnicos
Control y libertad del usuario	4	Opciones avanzadas de personalización con posibilidad de revertir cambios
Consistencia y estándares	5	Diseño coherente con patrones de interacción modernos
Prevención de errores	4	Validaciones completas con sugerencias de corrección
Reconocimiento vs. recuerdo	4	Interfaz que equilibra accesibilidad y funcionalidades avanzadas
Flexibilidad y eficiencia	5	Atajos de teclado y flujos optimizados para usuarios avanzados
Estética y diseño minimalista	4	Diseño moderno con enfoque en la visualización de datos
Ayuda ante errores	4	Mensajes específicos con soluciones propuestas
Documentación y ayuda	5	Documentación extensa, tutoriales interactivos y soporte en vivo

Tournify destaca por su experiencia de usuario moderna y fluida, con especial énfasis en dispositivos móviles mediante aplicaciones nativas para iOS y Android. La plataforma presenta una curva de aprendizaje moderada (aproximadamente 60 minutos para la configuración completa de un torneo), compensada por una extensa documentación y asistentes de configuración.

5. Enjore

Modelo de negocio

Enjore ofrece un modelo de suscripción escalable con planes adaptados a diferentes necesidades organizativas:

- **Plan basic** (17,90 €/mes):
 - Recomendado para organizadores pequeños
 - Formatos: Todos contra todos, eliminación directa, grupos + fase final
 - Funcionalidades básicas:
 - Gestión de equipos y jugadores
 - Clasificaciones y estadísticas
 - Hojas de resultados
 - Generación de calendarios (automática/manual)
 - Subida de fotos y videos
 - Sistema de tickets de asistencia
 - Límite: 3 torneos por año
- **Plan plus** (27,90 €/mes):
 - Incluye todas las características del Plan Basic
 - Ventajas adicionales:
 - Personalización de logotipos y colores
 - Gestión de responsables de equipos
 - Sistema de gestión de publicidad
 - Importación de jugadores desde Excel
 - Gestión de multas
 - Subdominio personalizado (ej. tutorneo.enjore.com)
 - Límite: 5 torneos por año
- **Plan premium** (47,90 €/mes):
 - Incluye todas las características del Plan Plus
 - Funcionalidades avanzadas:
 - Página web personalizada
 - Diseño totalmente personalizado
 - Gestión de noticias y notificaciones
 - Exportación de datos de jugadores

- Planillas de juego oficiales
- Sistema de sanciones disciplinarias
- Módulo de designación arbitral
- Límite: 10 torneos por año

Tiene tres niveles claramente diferenciados para distintos tamaños de organización, personalización con opciones de branding desde el plan intermedio. Muestra profesionalidad con herramientas avanzadas para gestión completa en el plan premium.

Notas. Modelo orientado principalmente a organizadores (no menciona plan gratuito o para jugadores individuales). El plan Plus ofrece la mejor relación calidad-precio. Ideal para ligas y federaciones con necesidades de gestión compleja.

Experiencia de usuario

La evaluación heurística de Enjore reveló un enfoque orientado a la experiencia social:

Tabla E5. Evaluación heurística de Enjore.

Principio heurístico	Puntuación (1-5)	Observaciones
Visibilidad del estado del sistema	4	Notificaciones en tiempo real efectivas
Coincidencia entre sistema y mundo real	3	Mezcla de terminología deportiva y de redes sociales
Control y libertad del usuario	3	Controles adecuados pero con algunas limitaciones en configuración avanzada
Consistencia y estándares	3	Algunas inconsistencias entre módulos de torneo y sociales
Prevención de errores	3	Validaciones estándar implementadas
Reconocimiento vs. recuerdo	4	Interfaz intuitiva con sugerencias contextuales
Flexibilidad y eficiencia	3	Funcionalidades avanzadas accesibles pero no optimizadas
Estética y diseño minimalista	2	Interfaz visualmente cargada con elementos sociales
Ayuda ante errores	3	Mensajes claros pero no siempre específicos
Documentación y ayuda	3	Documentación centrada en funcionalidades sociales

Enjore presenta una experiencia distintiva centrada en aspectos sociales y de comunidad, con una curva de aprendizaje media (aproximadamente 50 minutos para configuración inicial). La plataforma destaca en la experiencia de participantes y espectadores, pero muestra limitaciones en herramientas avanzadas para organizadores profesionales.

Análisis comparativo global

Para facilitar la visualización comparativa de los cinco competidores analizados, se presenta la siguiente tabla resumen que sintetiza las principales dimensiones evaluadas:

Tabla E6. Comparativa global de los competidores.

Criterio	Challonge	Xporty	Copa fácil	Tournify	Enjore
Rendimiento técnico	3/5	3/5	2/5	5/5	3/5
Escalabilidad	3/5	2/5	1/5	5/5	3/5
Arquitectura moderna	2/5	3/5	2/5	5/5	3/5
Experiencia de usuario	3/5	3/5	4/5	4/5	3/5
Funcionalidades avanzadas	3/5	4/5	2/5	5/5	3/5
Relación calidad-precio	4/5	3/5	5/5	3/5	4/5
Mercado objetivo	Semi-prof.	Profesional	Amateur	Profesional	Comunidades

Este análisis comparativo global revela posicionamientos claramente diferenciados entre los competidores analizados:

- **Challonge** se posiciona como una solución equilibrada para el segmento semi-profesional, con buen rendimiento general.
- **Xporty** destaca en funcionalidades administrativas y de gestión avanzada, orientándose a organizadores profesionales con énfasis en aspectos financieros y logísticos.

- **Copa fácil** presenta la mejor relación calidad-precio para torneos sencillos, con enfoque claro en la simplicidad y facilidad de uso para organizadores ocasionales.
- **Tournify** lidera en aspectos técnicos y funcionales, con capacidades avanzadas, aunque con un precio más elevado que refleja su orientación profesional.
- **Enjore** ocupa un nicho distintivo con su enfoque en aspectos sociales y de comunidad, diferenciándose mediante funcionalidades de interacción y contenido.

Conclusiones del análisis de competidores

El análisis detallado realizado permite extraer las siguientes conclusiones relevantes para el desarrollo estratégico del sistema:

1. **Segmentación clara del mercado:** Se identifica una diferenciación evidente entre soluciones para usuarios ocasionales (Copa Fácil), semi-profesionales (Challonge, Enjore) y profesionales (Xporty, Tournify).
2. **Balance entre funcionalidad y usabilidad:** Las soluciones más exitosas logran equilibrar la potencia funcional con interfaces intuitivas, evitando tanto la sobresimplificación (Copa Fácil) como la complejidad excesiva (ciertos aspectos de Xporty).
3. **Modelos de negocio diferenciados:** Se observa una evolución desde modelos freemium básicos hacia propuestas de valor más sofisticadas, incluyendo monetización de contenido (Enjore) y tarificación por evento (Tournify).
4. **Nichos de oportunidad:** Se identifican áreas insuficientemente atendidas por los competidores actuales, particularmente en la integración de capacidades analíticas avanzadas, personalización profunda de formatos de competición y un diseño UI y UX destacable para equipos, aficionados o jugadores, estadísticas en tiempo real, todas las funcionalidades desde un plan gratuito.

Estas conclusiones fundamentan las recomendaciones estratégicas presentadas en la sección principal del documento, orientando el posicionamiento diferencial del sistema en el ecosistema competitivo analizado.

Anexo F. Cálculo detallado del VAN y TIR

El Valor Actual Neto (VAN) se ha determinado utilizando la siguiente fórmula:

$$VAN = -I_0 + \sum_{t=1}^n \frac{FC_t}{(1+k)^t} \quad (6)$$

Donde:

- I_0 = Inversión inicial (US\$500)
- FC_t = Flujo de caja en el período t
- k = Tasa de descuento (12%)
- n = Horizonte temporal (5 años)

Sustituyendo los valores en la ecuación:

$$VAN = -500 + \frac{4.118,20}{(1+0,12)^1} + \frac{7.887,88}{(1+0,12)^2} + \frac{13.140,04}{(1+0,12)^3} + \frac{15.902,85}{(1+0,12)^4} + \frac{19.218,22}{(1+0,12)^5} \quad (7)$$

$$= -500 + 3.677,07 + 6.288,62 + 9.352,60 + 10.104,89 + 10.903,38 \quad (8)$$

$$= -500 + 40.326,56 \quad (9)$$

$$= \mathbf{US\$39.826,56} \quad (10)$$

Para el cálculo de la Tasa Interna de Retorno (TIR):

$$0 = -I_0 + \sum_{t=1}^n \frac{FC_t}{(1+TIR)^t} \quad (11)$$

Sustituyendo:

$$500 = \frac{4.118,20}{(1+TIR)^1} + \frac{7.887,88}{(1+TIR)^2} + \frac{13.140,04}{(1+TIR)^3} + \frac{15.902,85}{(1+TIR)^4} + \frac{19.218,22}{(1+TIR)^5} \quad (12)$$

Resolviendo numéricamente:

$$TIR = \mathbf{168,92\%} \quad (13)$$