**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR**

**Unidad Académica de Formación Técnica y Tecnológica – PUCE TEC**

# APLICACIÓN MÓVIL PARA LA GESTIÓN Y ORGANIZACIÓN DE LA LIGA BARRIAL EDÉN DEL VALLE

**Proyecto de titulación previo a la obtención del título de: Tecnólogo Superior en Desarrollo de Software**


**Autores:**

**Bryan Marcelo Cabrera Cevallos**

**Cristian Santiago Altamirano Tuquerrez**

**Tutor: José Luis Galarraga Cañizares**


**Quito, Ecuador**

**2026**

## Dedicatoria

A mis padres, por haberme brindado las herramientas necesarias para convertirme en un profesional con oportunidades, por abrirme el camino correcto a través del ejemplo de esfuerzo, perseverancia y dedicación. Sus valores y enseñanzas se reflejan en cada uno de mis logros y en las metas que continúo trazando a lo largo de mi vida académica y profesional.

A mi hijo, mi mayor fuente de inspiración, por ser el impulso que me motiva a alcanzar cada objetivo y enseñarme la maravillosa dualidad de ser padre y profesional al mismo tiempo. Gracias por recordarme cada día que todo es posible cuando se hace con amor y determinación.

A mi esposa, mi compañera de vida y apoyo incondicional, por ser mi refugio en los momentos difíciles, por su paciencia, comprensión y por recordarme siempre que soy capaz. Gracias por caminar a mi lado y compartir conmigo cada etapa de este proceso.

**Atentamente**

Bryan Marcelo Cabrera Cevallos

## Dedicatoria

A mi Familia dedico con mi amor y gratitud este logro a mi madre Marcia Verónica Tuquerrez de la Cruz, por su apoyo incondicional económico junto a su pareja Napo Chamorro Quien ha Sido mi fuerza y guía en cada paso. Gracias Por creer en mi por su esfuerzo incansable y por brindarme no solo ese apoyo moral haciendo posible este sueño que prometí alcanzarlo hace mucho tiempo atrás enseñándome que los sueños se alcanzan con unión y Solidaridad.

A mi hermano David Sebastián Altamirano Tuquerrez por estar siempre Presente por ese respaldo constante y ser ese ejemplo de lucha y perseverancia.

A mi hijo Estiven Santiago Altamirano Burga quien se convirtió en esa inspiración para seguir adelante. Todo mi esfuerzo ha sido pensando en su bienestar y su futuro.

A mi novia Ana Rodríguez, por no dudar de mi creer en mis capacidades por esa motivación a retomar mis estudios y acompañarme con paciencia y amor en cada etapa de este proceso.

Y en especial a la memoria de mi padre, Fidel Virgilio Altamirano Martínez quien partió hace 20 años, aunque no este físicamente conmigo, su recuerdo y sus enseñanzas han guiado cada paso que doy. Hoy cumplo la promesa que es convertirme en un Profesional digno de su Nombre y Orgullo de nuestra Familia.

**Atentamente**

Cristian Santiago Altamirano Tuquerrez

**Tabla de contenidos**

## Lista de tablas

## Lista de Figuras

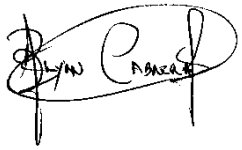
## DECLARACIÓN y AUTORIZACIÓN

Yo, **Bryan Marcelo Cabrera Cevallos** con C.I. 1717472185 autor(a) del trabajo de titulación intitulado: **"Aplicación móvil para la gestión y organización de la liga barrial Edén del Valle"**, previa a la obtención del título de **Tecnólogo Superior en Desarrollo de Software** en la Unidad Académica de Formación Técnica y Tecnológica PUCE TEC:

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE el referido trabajo de titulación, respetando las políticas de propiedad intelectual de Universidad.

Quito, 6 de febrero de 2026



Bryan Marcelo Cabrera Cevallos

C.I. 1717472185

**DECLARACIÓN y AUTORIZACIÓN (Cristian)**

Yo, **Cristian Santiago Altamirano Tuquerrez** con C.I. 172204267-6 autor del trabajo de titulación intitulado: **"Aplicación móvil para la gestión y organización de la liga barrial Edén del Valle"**, previa a la obtención del título de **Tecnólogo Superior en Desarrollo de Software** en la Unidad Académica de Formación Técnica y Tecnológica PUCE TEC:

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE el referido trabajo de titulación, respetando las políticas de propiedad intelectual de Universidad.

Quito, 6 de febrero de 2026

Cristian Santiago Altamirano Tuquerrez

C.I. 172204267-6

## Agradecimientos

A Dios, por darme la fortaleza, la sabiduría y la perseverancia necesarias para culminar este importante proyecto de mi vida.

A mis padres, por su amor incondicional, su apoyo constante y por enseñarme el valor del esfuerzo y la dedicación. Sin ustedes, nada de esto habría sido posible.

A mi esposa, por comprender mis ausencias y alentarme en cada paso de este camino académico.

A mi hijo, por ser ese motor que me impulsa cada día de mi vida y me llena de ganas de ser mejor a nivel personal y profesional.

A mis docentes, por su guía, paciencia y valiosos consejos durante el desarrollo de este trabajo. Su acompañamiento fue fundamental para alcanzar los objetivos propuestos.

A mis compañeros y amigos, por compartir conmigo este proceso, por las horas de estudio, las conversaciones y el ánimo en los momentos más difíciles.

Finalmente, agradezco a todas las personas que, de una u otra forma, contribuyeron al desarrollo y culminación de esta tesis.

**Atentamente**

Bryan Marcelo Cabrera Cevallos

## Agradecimientos

Quiero expresar mi más sincero agradecimiento:

A **Dios**, por estar a mi lado y no dejarme caer, por la salud y la sabiduría necesarias para superar cada obstáculo que se presentó en el camino.

A mis **docentes a Diana Molina** por compartir sus conocimientos, su paciencia y su dedicación los cuales fueron fundamentales para mi formación académica y profesional y esa guía tan personalizada Asia mi persona.

A mi **tutor de tesis**, **José Luis Galarraga Cañizares** por su guía, su orientación constante y sus valiosas recomendaciones que permitieron mejorar y fortalecer este trabajo.

A la Universidad Católica **por brindarme las herramientas y los espacios necesarios para desarrollar mis capacidades y crecer como estudiante y futuro profesional.**

A mis **compañeros de carrera**, por el compañerismo, las experiencias compartidas y el apoyo mutuo en cada etapa del proceso por esos momentos inolvidables y la lucha constante.

Y finalmente, a mi **familia**, que siempre estuvo a mi lado brindándome su amor, apoyo y comprensión incondicional. Sin ustedes, este logro no habría sido posible.

**Atentamente**

## Introducción

La presente investigación se centrará en crear una aplicación móvil para la gestión y organización de la liga barrial “Edén del Valle”, la cual permitirá optimizar y automatizar los procesos operativos y administrativos de la misma. Este proyecto aborda una necesidad especifica que tienen este tipo de organizaciones, ya que al ser una entidad que se autogestiona manualmente con los aportes de los equipos participantes, suelen carecer de los recursos suficientes para invertir en tecnología, lo cual limita la eficiencia, fiabilidad y transparencia en la gestión realizada dentro de la institución.

En la actualidad el uso de las tecnologías de la información y comunicación ha transformado el día a día de cualquier tipo de organización o personas, facilitando mucho el acceso a la información, permitiendo una mayor eficiencia y transparencia en cualquier tipo de proceso por más sencillo que este sea, por ello utilizando este concepto se busca generar las herramientas necesarias que aporten en la gestión de la liga barrial.

El desarrollo de esta aplicación se fundamentará en los principios de usabilidad, accesibilidad y sostenibilidad tecnológica, empleando metodologías agiles que simplifiquen los tiempos de desarrollo y permitan obtener entregables de forma temprana, esta implementación busca agilizar las tareas administrativas y operativas de la liga barrial, así como transparentar el acceso a la información para todos los involucrados.

Finalmente, esta investigación es de carácter social, ya que busca promover la digitalización de procesos en un entorno comunitario, fortaleciendo así el uso de las TICs,

la participación ciudadana y el deporte barrial, con todas las limitaciones que esto representa, de este modo el proyecto responde un compromiso con la innovación, democratización y acceso a la tecnología.

**Antecedentes**

La liga barrial Edén del Valle es una organización construida el 1 de febrero del 2016, esta liga se formó con el propósito de fomentar el deporte entre la comunidad sin distinción de género o de edad, es  por lo que la liga a través de sus diferentes categorías ha buscado formar a diferentes deportistas en varios ámbitos como futbol, básquet y ecua vóley.

Actualmente la liga barrial Edén del Valle cuenta con 2 categorías de futbol, las cuales son masculino – categoría única y femenino – categoría única, adicionalmente en determinadas fechas del año, proceden a realizar torneos cortos en otros deportes como los mencionados anteriormente, sin embargo el futbol representa el mayor ingreso y operación para la liga ya que prácticamente los campeonatos suelen durar de 8 a 10 meses, por lo que el proyecto de titulación se enfocará en cubrir las necesidades operativas y administrativas de este deporte.

Adicionalmente la liga cuenta actualmente un total de 20 equipos entre hombres y mujeres, los cuales cuentan alrededor de 26 jugadores como miembros de cada club, dando un total aproximado de 520 jugadores, los cuales se verán beneficiados por la aplicación, asi mismo existe una cantidad incalculable de hinchas los cuales acuden cada fin de semana al complejo deportivo con la finalidad de alentar a sus equipos y seguirlos lo que los convierte en un actor adicional para nuestra aplicación en vista que su sistema es solo manual.

## Planteamiento del problema

En el contexto de las ligas barriales de Ecuador, organizaciones de gran importancia socio cultural que fomentan la cohesión comunitaria, se observa una brecha tecnológica significativa. La Liga Barrial "Edén del Valle" no es una excepción su gestión operativa depende de procesos manuales tanto canales de comunicación informales como hojas de cálculo, documentos físicos y grupos de WhatsApp.

La liga tiene varios procesos que se ejecutan de forma manual y que buscan automatizarse con esta aplicación, entre los cuales están la gestión de partidos y hojas de vocalías, control de tarjetas amarillas, rojas y goles, gestión de sanciones, organización de partidos y tabla de posiciones, aplicación de bonificaciones, gestión y control financiero, aplicación de multas.

Esta metodología tradicional genera problemas tangibles como la descentralización de la información, alta propensión a errores humanos en el registro de resultados y sanciones, una comunicación fragmentada que dificulta el acceso oportuno a datos cruciales para los equipos y la comunidad.

Este problema técnico se enmarca en un contexto socioeconómico donde las ligas barriales son administradas por voluntarios y operan con presupuestos limitados, esta realidad económica les impide acceder a soluciones de software comerciales, que suelen tener costos elevados no están adaptadas a sus necesidades específicas. Sin embargo, existe una alta penetración de teléfonos inteligentes entre los participantes, lo que

evidencia una oportunidad desaprovechada: la comunidad posee la tecnología de acceso (dispositivos móviles), pero la organización carece de la herramienta digital adecuada para capitalizarla y optimizar su funcionamiento satisfaciendo las necesidades tanto de la liga como los jugadores.

**Justificación**

La implementación de esta herramienta digital es fundamental para centralizar la información, automatizar tareas mejorando significativamente la transparencia y el acceso a los datos para directivos, jugadores y la comunidad en general.

Desde la perspectiva técnica como tecnológica, la relevancia del proyecto radica en la adopción de un enfoque de aplicaciones hibridas. Esta decisión estratégica se justifica al permitir que a partir de una única base de código se generen aplicaciones móviles y web, sin embargo, en este alcance se realizará específicamente una aplicación para Android.

Esto no solo optimiza los recursos al reducir drásticamente el tiempo los costos de desarrollo, sino que también garantiza una experiencia de usuario homogénea de alta calidad facilitando así una mayor adopción y consolidando un ecosistema tecnológico sostenible fácil mantenimiento para la organización.

Para la construcción de la solución, se emplearán metodologías ágiles, concretamente el marco de trabajo Scrum, para una gestión de proyecto adaptativa e iterativa. En el plano tecnológico, el frontend se desarrollará utilizando el framework react native con typescript y otros componentes que nos permitirán aprovechar al máximo las utilidades para crear una APP vistosa, asegurando una interfaz de usuario reactiva y moderna.

Para el backend, se integrará el framework react native, la base de datos PostgreSQL y Spring boot, así mismo se utilizará métodos de autenticación basados en JWT, implementando esta APP dentro de Android estudio.

**Objetivo General**

Desarrollar una aplicación móvil con React Native orientada a la gestión integral de la Liga Barrial *"Edén del Valle"*, exportándola a Android Studio aplicando la metodología ágil Scrum para automatizar los procesos administrativos, centralizar la información mejorando la comunicación entre la organización los equipos y la comunidad.

**Objetivos Específicos**

- Levantar los requerimientos funcionales y no funcionales del sistema mediante entrevistas y reuniones con los directivos de la liga, para establecer el alcance y las prioridades del producto en el marco de un backlog inicial.
- Diseñar la arquitectura del software y la base de datos así como la interfaz de usuario (UI/UX) creando prototipos y flujos de navegación que garanticen una experiencia intuitiva y accesible para los diferentes perfiles de usuario.
- Implementar la aplicación móvil en React Native, desarrollando los componentes visuales, integrando la lógica de negocio y asegurando su compatibilidad mediante la exportación y compilación en Android Studio.
- Aplicar la metodología ágil Scrum durante el desarrollo del sistema, organizando el trabajo en sprints, asignando roles y utilizando tableros de

gestión de tareas para fomentar la colaboración y la entrega incremental del producto.

- Realizar pruebas de calidad (unitarias, de integración y de usabilidad) en cada iteración, con el fin de validar las funcionalidades, corregir errores y garantizar la estabilidad de la aplicación antes de su despliegue final.

# Capítulo I: Levantamiento de Requisitos y Diseño del Sistema

## Análisis de requerimientos

(Juura, 2024) menciona que el análisis de requisitos es el proceso en el cual se descubren y se documentan todos los requerimientos que el cliente desea solicitar de acuerdo con el sistema que se esté desarrollando, el objetivo de este proceso es gestionarlos de forma correcta durante todo el ciclo de vida del sistema y verificar su cumplimiento cuando se realice una retrospectiva del proyecto al cierre de este.

Dentro del desarrollo de software el análisis de requisitos es una parte fundamental, que según (Khalid, Rasheed, & Khaleeq, 2025) comprende de 4 fases: Elicitación, análisis y diseño, especificación y validación, cada una de estas fases se basan en el entendimiento de lo que requiere el usuario para solventar un problema o una necesidad y en proponer a través de diferentes herramientas diversas soluciones que ayuden a mitigar dicha necesidad.

El análisis de requerimiento para este proyecto consistirá en un levantamiento presencial con el cliente, en donde se definirán todas los requerimientos funcionales y no funcionales que debe tener la aplicación, así como también se definirán los límites del proyecto, creando un alcance y especificando que funcionalidades quedarán fuera de este proyecto, finalmente se realizarán los diagramas de caso de uso para que se pueda abordar cada requerimiento desde un diferente punto de vista.

## Alcance del proyecto

Como parte del alcance del proyecto a continuación en la figura 1 se muestran los requerimientos funcionales que van a ser considerados dentro del proyecto.

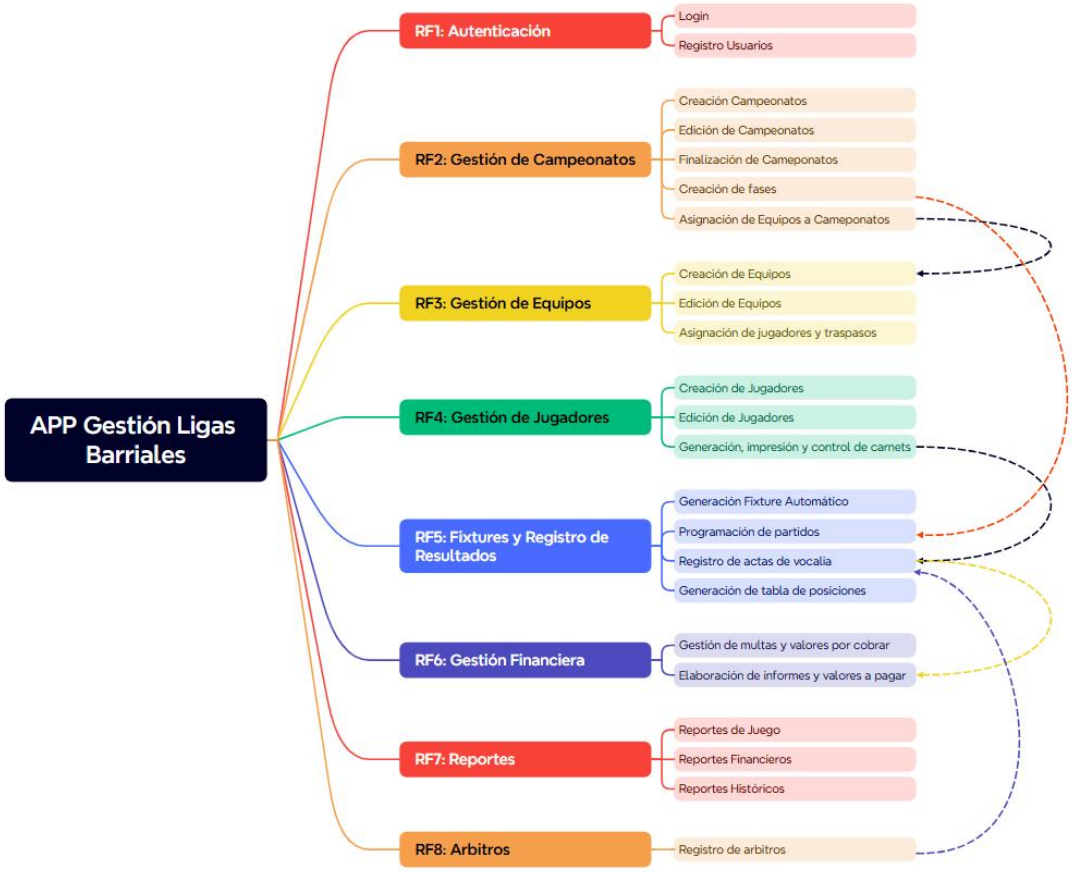



*Figura 1: Diagrama de Requerimientos*

## Requerimientos funcionales

Los requerimientos funcionales constituyen características y funciones del sistema, que permiten a los usuarios completar diferentes tareas para lo cual fue diseñada dicha aplicación (Mazurkiewiz, 2023), por ello dentro del análisis de requerimientos es tan importante identificar cuales son las funcionalidades que serán el core de nuestro aplicación, para que a partir se pueda generar la línea base que nos permita iniciar con el prototipado de la solución, definición de entidades y el posterior desarrollo de la aplicación.

La identificación de estos requerimientos funcionales se los realizará a través de reuniones especificas con el usuario en la cual se definen los comportamientos, funciones y capacidades específicas que el software debe proporcionar para satisfacer las necesidades de los usuarios (Mehta, Mehta, & Bindal, 2022), dentro de este levantamiento se espera definir todos los módulos que requiere tener el sistema así como cada una de las acciones que realizará el usuario en la aplicación.

La aplicación móvil para la gestión de la liga barrial Edén del Valle tendrá los siguientes requerimientos funcionales:

***Módulo de Autenticación***

- Página de registro de incluyendo los roles administrador y directivos
- Página de login
- Página de recuperación de contraseña

***Módulo de Gestión de Campeonatos:***

- Formulario de creación y configuración de campeonatos
- Selección de campeonatos
- Selección de equipos
- Búsqueda de jugadores por equipo
- Edición de campeonatos
- Creación de fases del torneo, sorteo de grupos se realizará de manera Aleatoria, eliminatorias
- Página de asignación de equipos a las fases del torneo

***Módulo de Gestión de Equipos***

- Formulario de creación y edición de equipos

- Asignación de jugadores a equipos
- Gestión de imágenes logos de los equipos
- Página de traspasos de jugadores tendrá el rol coordinado directivo y administrador
- Gestión de imágenes logos de equipos

***Módulo de gestión de jugadores:***

- Formulario de creación y edición de jugadores
- Gestión de imágenes jugadores
- Generación de plantillas para carnets
- Lectura y control de carnets a través de código QR
- Plantilla de impresión de carnets físicos

***Módulo de gestión de calendario de partidos y registro de resultados***

- Generación automática del fixture
- Modificación de partidos y asignación de árbitros y vocales
- Generación de actas de vocalía
- Edición de actas de vocalía, escaneo de jugadores
- Cierre de hoja de vocalía e informes
- Generación y actualización de tabla de posiciones

***Módulo de Reportes***

- Creación e impresión de reportes de partido y actas de vocalía
- Creación e impresión de reportes de equipos y jugadores

**Funcionalidades fuera de alcance**

- Integración con sistemas de cobranzas o pagos
- Soporte a otras ligas o equipos que no formen parte de Liga Barrial Edén del Valle.
- Gestión de Facturación
- Reportes estadísticos avanzados o análisis de datos
- Soporte para múltiples idiomas
- Soporte a otros deportes diferentes al futbol
- Gestión de marketing digital
- Costos de licenciamiento respecto a la publicación de la aplicación en Android y IOS

**Requerimientos no funcionales**

Al contrario de los requisitos funcionales, estos tipos de requisitos definen la calidad del servicio que se va a brindar a través de la plataforma, así como también la interfaz de usuario que tendrá y el tiempo que tomará cada proceso, y finalmente las restricciones sobre cómo deben producirse y entregarse dichos servicios a través del sistema (Bellavista, 2024).

Los requisitos no funcionales definen los aspectos que de cierta forma son intangibles para el usuario pero que determinan la calidad del software, entre estos requisitos están la velocidad de mi aplicación, la seguridad y la integridad de datos (Altexsoft Editorial Team, 2023), todos estos definen compromisos que la aplicación en un determinado número de transacciones adquiere con el usuario.

A continuación, se detallan los requisitos no funcionales que tendrá la aplicación para la gestión de la liga Edén del Valle:

- La aplicación debe cargar la pantalla principal en menos de 3 segundos bajo condiciones de una red estándar.
- Las operaciones de consulta como tabla de posiciones, fixture, etc. deben responder en un lapso de 5 segundos.
- El sistema debe soportar al menos 100 usuarios de manera concurrente, sin perder rendimiento.
- La aplicación debe usar autenticación a través de un JWT, evitando exponer cualquier información sensible a través de las APIs.
- Las contraseñas deben almacenarse de forma segura, siendo estas encriptadas a través de cualquier herramienta o librería.
- Todas las comunicaciones entre el front y back deben ser a través de HTTPS.
- La aplicación debe implementar un control basado en RBA (Role based Access)
- La interfaz debe ser intuitiva y debe cumplir con un diseño responsive para los diferentes tipos de móviles y tablets.
- El sistema debe tener una disponibilidad anual del 99%

**Arquitectura empresarial**

La arquitectura empresarial es un framework que permite relacionar los procesos de negocio con las tecnologías de información, garantizando que exista un coherencia entre todos los niveles y sistemas organizaciones, este tipo de arquitectura provee una estructura que alinea las capacidades del negocio, datos, aplicaciones y tecnología para

alcanzar los objetivos trazados por la compañía y apoyar en la transformación digital de la misma (Sidana, Harminder, & Devi, 2022).

A continuación, se presenta el diagrama de arquitectura empresarial que se utilizará en este proyecto, esta arquitectura se organiza en 4 capas: Arquitectura de negocio, arquitectura de aplicación, arquitectura de datos y arquitectura tecnológica, este diagrama toma como referencia el modelo TOGAF para describir cada uno de los componentes que la conforman.

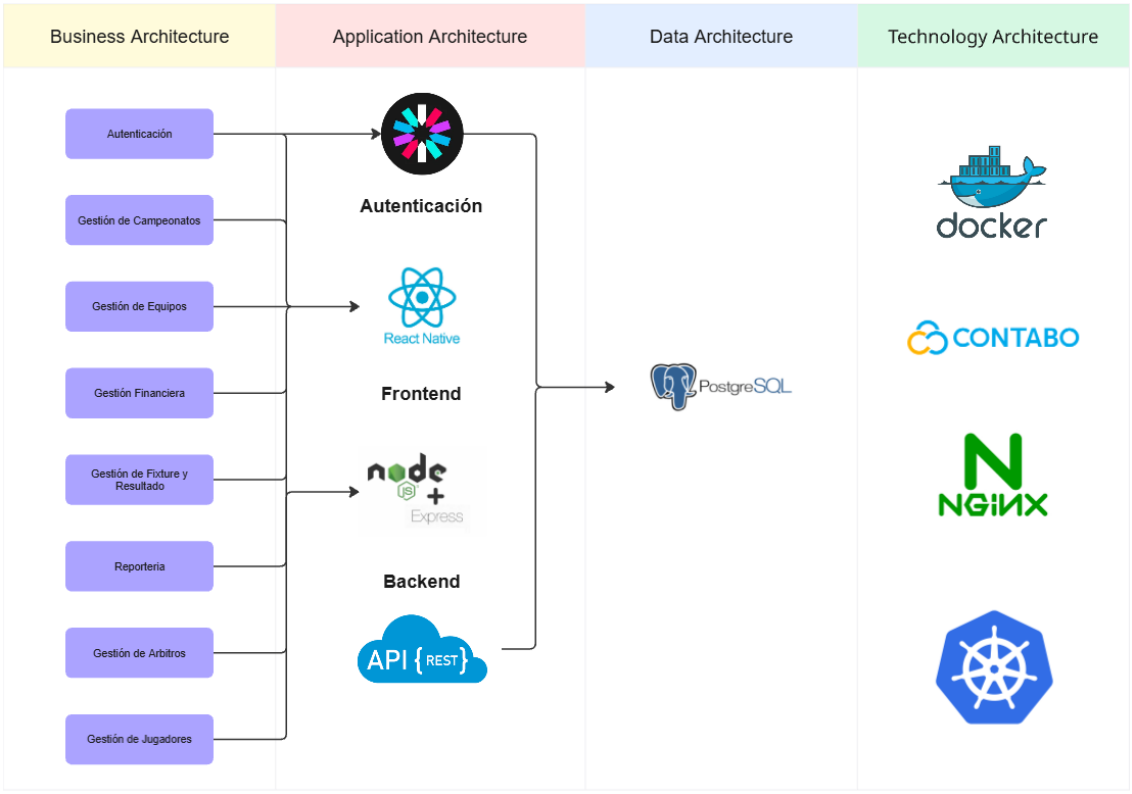



*Figura 2: Diagrama de arquitectura empresarial*

### *Arquitectura de negocio*

En la primera columna del diagrama se representan todos los procesos de negocio que serán plasmados dentro de la aplicación, estos son esenciales para el funcionamiento de la liga, cada una de estas actividades corresponden a las diferentes funciones que

realizan varios actores en la organización dentro de un entorno real, y constituyen los pilares en los cuales se trabajará para crear la solución tecnológica propuesta.

***Arquitectura de aplicación***

La segunda columna muestra como estos procesos de negocio se transformarán en diferentes componentes de software, esto contendrá 3 capas:

- **Autenticación**: Esta se realizará a través de JWT y será el responsable del manejo de sesiones y credenciales.
- **Frontend**: La aplicación móvil estará realizada en React Native, este framework simplifica los tiempos de desarrollo entre los SO de IOs y Android.
- **Backend**: El backend se realizará a través de una arquitectura de microservicios y se realizará en java spring boot, este backend servirá como el intermediario entre el front y la base de datos.

***Arquitectura de datos***

En esta sección se muestra la información relacionada al almacenamiento de datos, para ello se utilizará la aplicación PostgreSQL la cual es una base de datos relacional, esta capa asegurará que la información sea persistente y consistente para los otros componentes.

***Arquitectura tecnológica***

Esta última sección presenta las tecnologías que se utilizarán para desplegar, ejecutar y escalar nuestra aplicación en caso de ser requerido en un futuro, las tecnologías a usar serán docker, contabo, nginx y kubernetes, estas tecnologías también van orientadas hacia nuestra estructura de microservicios.

## Diagrama de Casos de Uso

Los casos de uso son una herramienta fundamental para representar las interacciones entre los usuarios y el sistema, estos describen un conjunto de acciones que producen un resultado valioso para el actor. Atreves de ellos describen los pasos o acciones que producen a la obtención del resultado del valor para el usuario. Según (Pressman, R. S. 2010 p.134) "un caso de uso describe un conjunto de acciones que un sistema realiza y que producen un resultado observable de valor para un actor particular" (p.132) los casos de uso no solo sirven como un medio de comunicación entre los desarrolladores y los interesados del proyecto, si no que contribuyen a documentar los comportamientos del sistema de manera clara y estructurada.

## Casos de Uso Liga Edén del Valle

### *Gestión Ligas Barreales Administrador*

El diagrama de casos de uso mostrado en la figura 3 representa las principales funcionalidades de la aplicación móvil para realizar la gestión de Ligas Barriales, donde el actor principal es el administrador**.**

Este usuario tiene el acceso a distintos procesos del sistema, tales como la gestión de Campeonatos, gestión de equipos, jugadores, árbitros, fixture y resultados, gestión Financiera y reportaría. Cada uno de estos casos de uso permite modelar las distintas interacciones que el administrador realiza con el sistema, también detalla las acciones con resultados esperados.

De acuerdo con (Ph.D, 2010), "un caso de uso describe un conjunto de acciones que un sistema realiza y que producen un resultado observable de valor para un actor en particular"(p.132). por su parte (Sommerville, 2011) sostiene que los casos de uso nos

permiten identificar y especificar los requisitos funcionales del sistema atreves de escenarios que representan las interacciones entre el usuario y la aplicación.

En este sentido el diagrama permite comprender de forma visual como el administrador interactúa con los distintos módulos del sistema.

Por ejemplo: puede Crear, Modificar campeonatos, realizar la aprobación de Equipos, registrar árbitros, generando los reportes financieros y consultar resultados de los partidos. Este tipo de representación facilitara la comunicación entre los desarrolladores y el cliente del proyecto, garantizando que los requerimientos funcionales del sistema se acoplen de manera clara y estructurada.

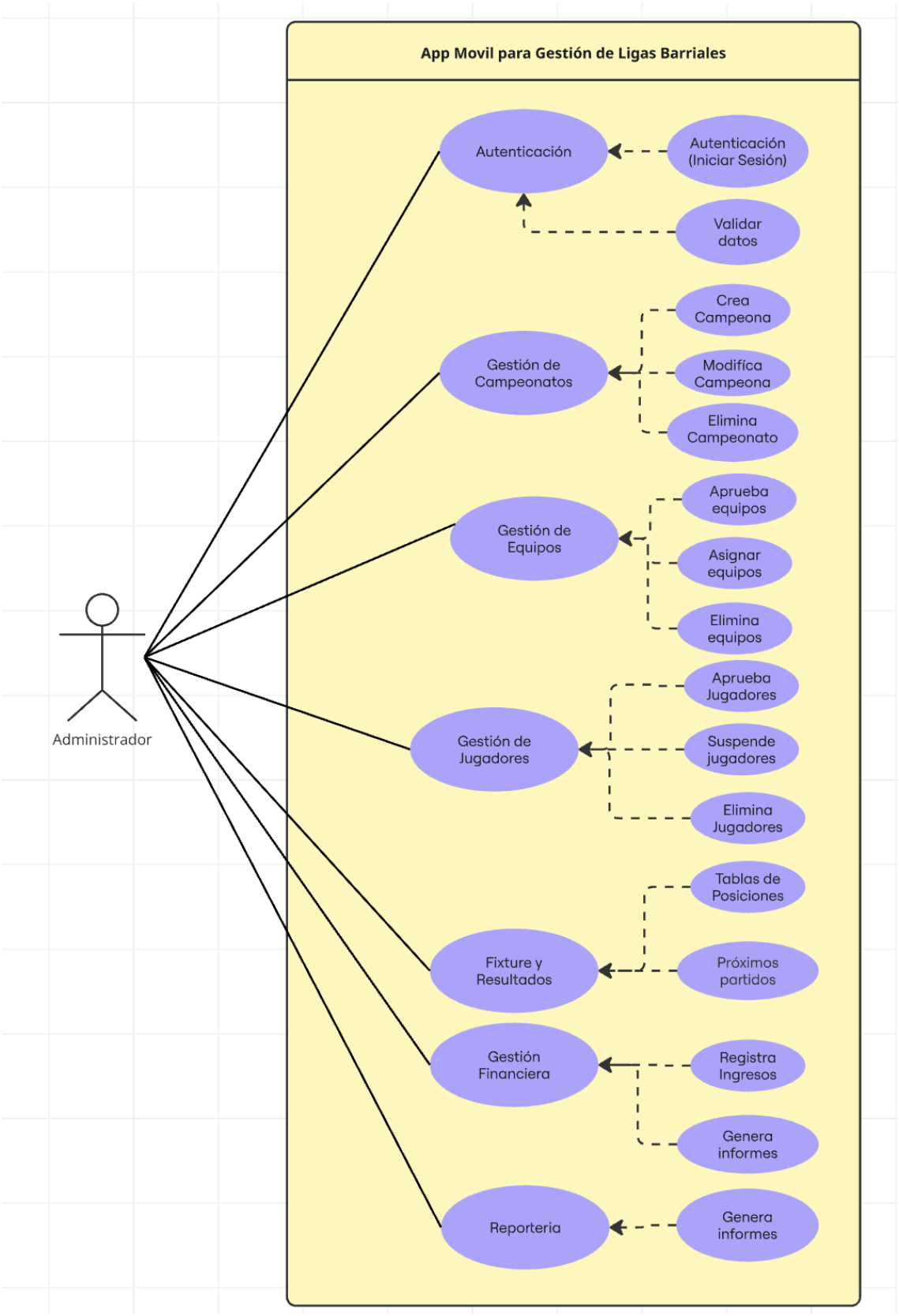



*Figura 3: Diagrama de casos de uso administrador*

***Gestión Ligas Barreales Directivo***

La figura 4 muestra el diagrama de casos de uso correspondiente al actor directivo**,** quien cumple un rol administrativo dentro de la aplicación Móvil para la gestión de ligas barriales. Este usuario tiene el acceso a las diferentes funcionalidades relacionadas con la autentificación, la gestión de jugadores, consulta de resultados y la revisión de reportes.

Entre las principales acciones que este puede realizar se encuentran: registrar y actualizar los datos de los jugadores, también elimina o dar de baja a los mismos consulta fixtures y resultados de los partidos, revisa las tablas de posiciones y goleadores del campeonato y también accede a los reportes financieros y estadísticos del equipo.

De acuerdo con (Sommerville, 2011), los casos de uso permiten describir los requisitos funcionales del sistema a través de escenarios que representan las interacciones entre el usuario y el software. En este Contexto el Actor Directivo interactúa con el sistema para obtener la información y ejecutar tareas que facilitan la administración y el control del equipo. Así este diagrama contribuye a visualizar de forma clara las responsabilidades y límites de acción del directivo dentro de nuestra aplicación móvil.

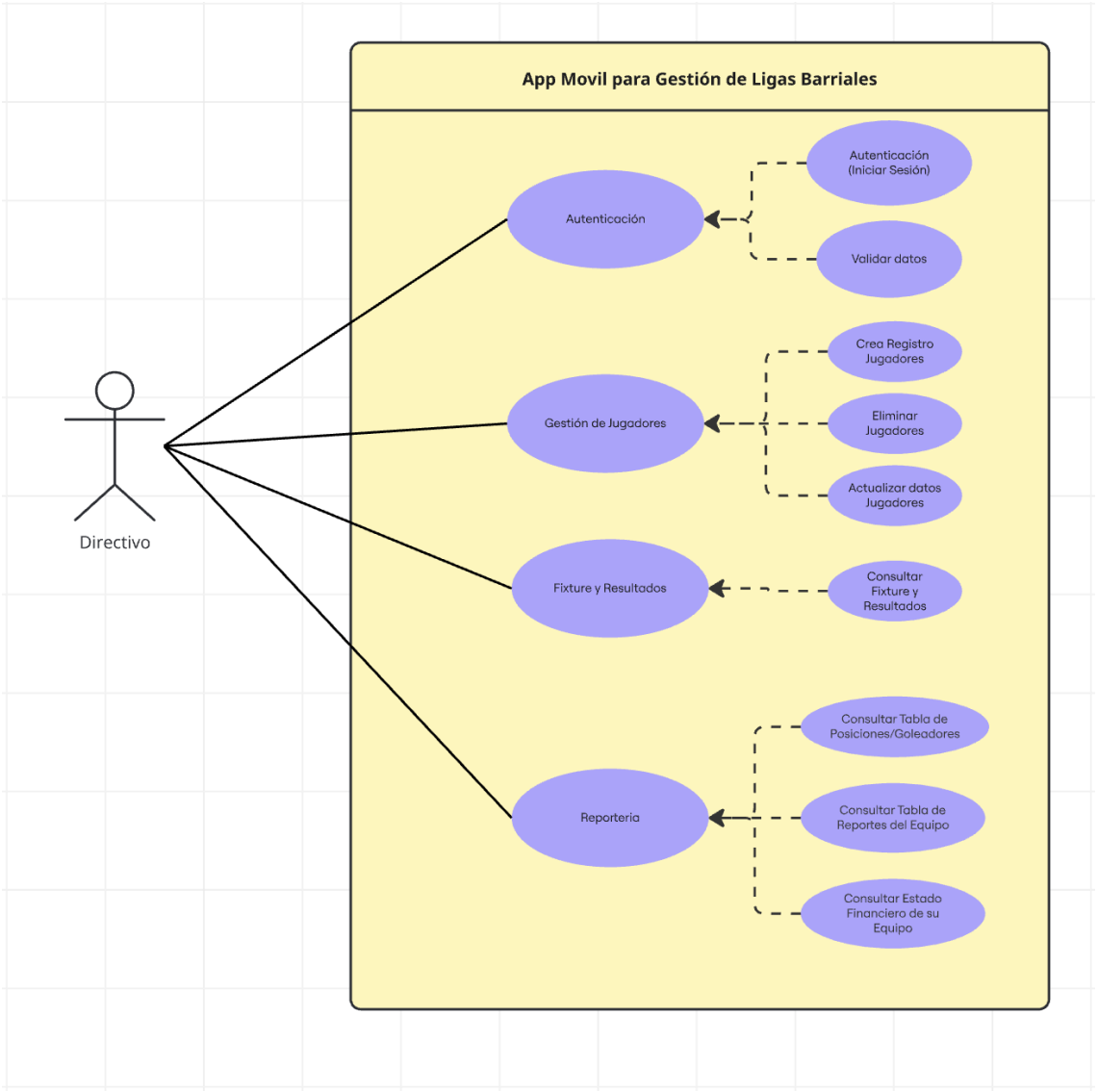



*Figura 4: Diagramas de casos de uso directivo*

***Gestión Ligas Barreales Jugadores Publico General.***

La figura 5 representa el diagrama de casos de uso correspondiente al actor Jugador y Publico General el cual agrupa a los usuarios que utilizan la aplicación móvil para consultar información sobre la liga los resultados y las estadísticas.

Este actor tiene accesos a las funcionalidades como autentificar el registro la consulta de fixtures y resultados la visualización de tablas de posiciones con el acceso a estadísticas individuales y colectivas de los equipos y jugadores adicional puede hacer el uso de consultar las noticias y novedades relacionadas con la liga lo que esto promueve la participación activa y la transparencia de la información deportiva.

Según (Grady Booch, 2000), “Los casos de uso son una técnica que permite capturar y describir el comportamiento del sistema del punto de vista del usuario especificando como el actúa con él para lograr un objetivo particular” (p. 57)

Accediendo a estos datos actualizados sin modificar su estructura principalmente como consumidor de la app móvil y su información, accediendo a los datos del sistema sin modificar su estructura. Esto contribuye a mantener informada a la comunidad deportiva y reforzar la usabilidad desde la perspectiva del usuario final.

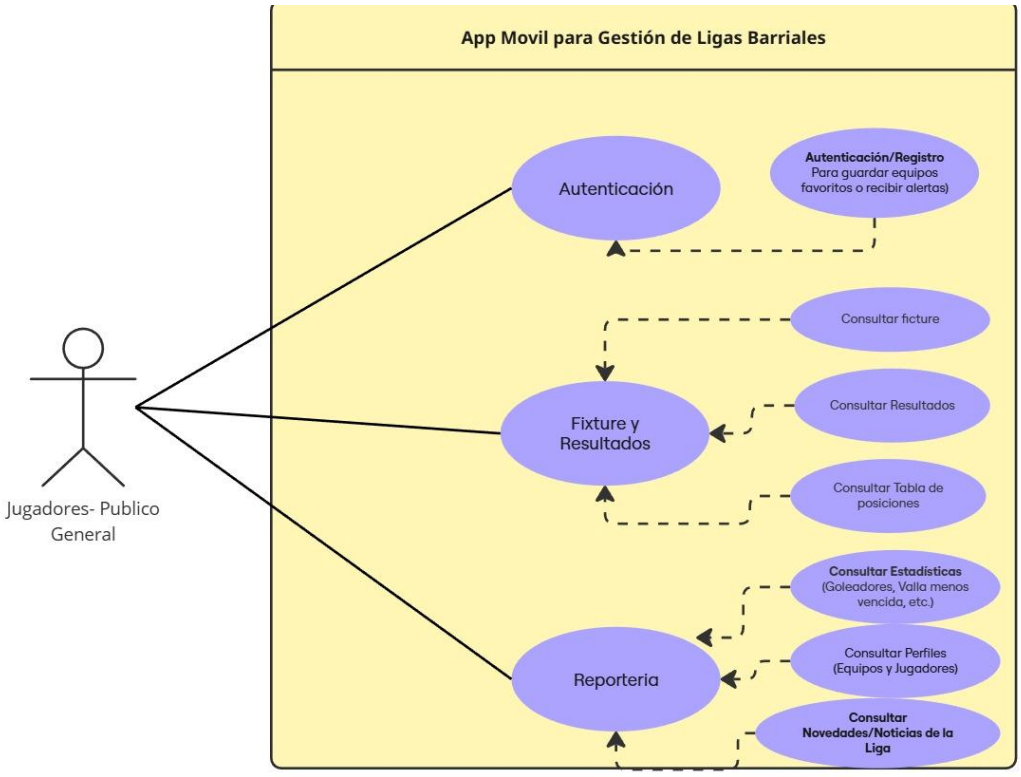



*Figura 5: Diagramas de caso de uso jugadores - publico general*

**Definición de Esquema o Diagrama Entidad Relación**

El diagrama entidad relación es una representación gráfica que permite modelar la estructura lógica de la base de datos. Este tipo de diagrama fue propuesto por (Chen, 1976)y se utiliza ampliamente para identificar los elementos fundamentales de la información que manejará un sistema: las entidades sus atributos y las relaciones entre ellas.

Según (Coronel, 2019), el modelo entidad relación constituye la base del diseño de la base de datos relacionales coma ya que facilita la comprensión del dominio del problema antes de implementar tablas físicas en un sistema de gestor de base de datos (SGBD).

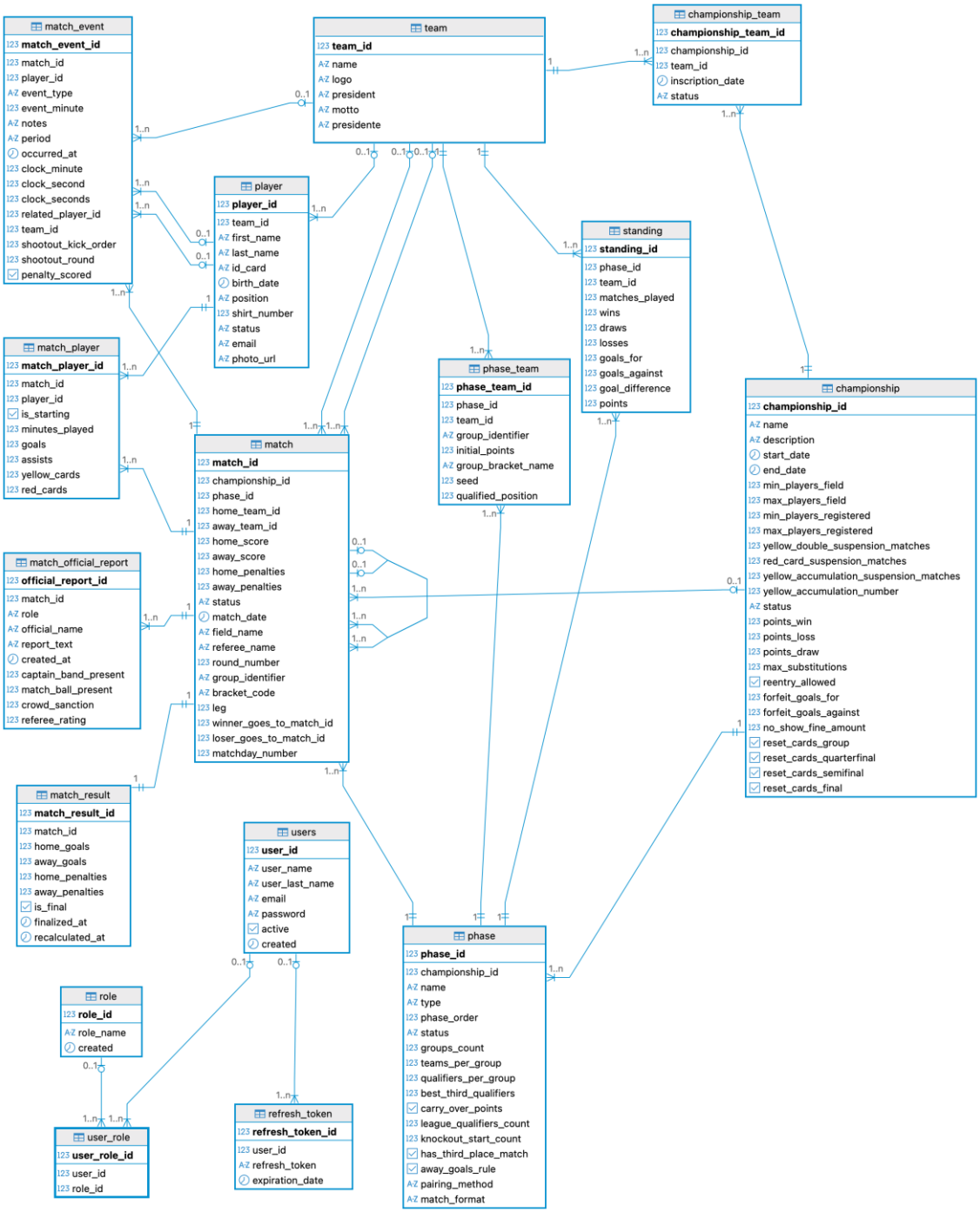



*Figura 6: Diagrama de Entidad Relación*

**Descripción del diagrama entidad relación de la APP "Gestión de Ligas barriales.**

Este diseño representa un modelo de base de datos relacional orientada a la gestión de ligas barriales, este modelo está adaptado para soportar todos los procesos descritos anteriormente como: Registrar campeonatos, equipos, fases, calendarización de partidos, registro de eventos, tablas de posiciones, etc.

Para garantizar la consistencia de seguridad y datos el modelo utiliza claves primarias, foraneas, reestricciones como Check o Unique y diferentes reglas de eliminación que permitan a la base de datos evitar cualquier error humano.

**2.1 Entidades Principales**

El sistema se compone de las siguientes entidades:

1. Championship: Esta tabla representa el campeonato o torneo principal y contiene la información general del campeonato como fechas de inicio y fin, reglas de participación y diferentes parámetros configurables del torneo. Esta entidad es la tabla principal del modelo ya que se relaciona con diferentes tablas del modelo.
2. Team: Esta tabla registra a los equipos que conforman los diferentes campeonatos la relación principal es un equipo puede participar en múltiples campeonatos y puede tener múltiples jugadores asociados.
3. Player: Esta tabla registra a los jugadores que pueden participar en un equipo, dentro de esta tabla se aplicó una restricción UNIQUE en el campo email y en el campo ID Card
4. Championship Team: Esta tabla se utiliza como una tabla intermedia que representa la inscripción de un equipo dentro de un campeonato. Aquí también se

aplicó una restricción para evitar que el mismo equipo se registre 2 veces en un mismo campeonato.

5. Phase: Esta tabla registra todas las fases que pertenecen a un campeonato, incluye diversos atributos que configuran las reglas en la fase del campeonato, cada fase está asociada a un championship id y se han aplicado restricciones de tipo check para asegurar que existan campos obligatorios en cada fase,
6. Phase Team: Esta tabla permite registrar la participación de un equipo dentro de una fase especifica, esta tabla intermedia tiene una relación de muchos a muchos, aquí se aplicó una restricción UNIQUE para evitar que se registren equipos duplicados en una fase.
7. Match: La tabla representa a los partidos de un campeonato, soporta el registro de marcadores en las distintas fases que pueden llegar a existir y permite modelar si un equipo avanza o no de fase.
8. Match Result: La tabla guarda la información acerca de los marcados de cada partido para la configuración de la tabla de posiciones.
9. Match Event: Esta tabla permite configurar los eventos ocurridos durante un partido como goles, tarjetas, sustituciones, penales, etc.
10. Match Player: Esta tabla permite registrar a los jugadores dentro de un partido, con esta tabla podemos obtener las estadísticas por temporada.
11. Standing: Esta tabla permite almacenar la información que corresponde a la tabla de posiciones.
12. Match Official Report: Esta tabla almacena la información de los informes que generan los árbitros, vocales y veedores.

13. Users: Esta tabla permite registrar a los usuarios, la contraseña por cuestiones de seguridad se almacena de forma encriptada.
14. Role: Esta tabla permite definir los roles que existirán en mi APP.
15. User Role: Esta tabla permite registrar a los usuarios de acuerdo a su role.
16. Refresh Token: Esta tabla permite registrar los tokens de refresco JWT

### *2.2 Relaciones del Modelo*

El modelo tiene las siguientes relaciones:

- Championship: Esta tabla se relaciona con:
    - Phase de 1 a muchos
    - Championship team de 1 a muchos
    - Match de 1 a muchos
- Team: Esta tabla tiene relación con:
    - Player de 1 a muchos
    - Championship team de 1 a muchos
    - Phase Team de 1 a muchos
    - Standing de 1 a muchos
    - Match de 1 a muchos
    - Match Event de 1 a Muchos
- Phase: Esta table tiene relación con:
    - Phase Team de 1 a muchos
    - Standing de 1 a muchos
    - Match de 1 a muchos
- Match: Esta tabla tiene relación con:

    - Match Event de 1 a muchos
    - Match Player de 1 a muchos
    - Match Official Report de 1 a Muchos
    - Match Result de 1 a muchos
- Player: Esta tabla tiene relación con:
    - Match Event de 1 a muchos
    - Match Player de 1 a muchos
- Users: Esta tabla tiene relación con:
    - Refresh Token de 1 a muchos
    - User Role de 1 a muchos
    - Role con user role de 1 a muchos

**Metodología de desarrollo Scrum Basado en los casos de uso.**

El desarrollo de futespirito se ejecutó mediante un enfoque ágil Aplicando el marco de trabajo Scrum para organizar el desarrollo de esta app de manera interactiva y también decir que fue incremental.

Scrum definió esas responsabilidades en distintos casos que nos permitieron hacer entregas de ciclos cortos con inspección y adaptación continua.

Este enfoque nos permitió mediante una herramienta priorizar la entrega de software funcional teniendo esa capacidad al cambio durante el proyecto sin ver afectado el valor y calidad de nuestra APP Futespirito.

**Equipo de trabajo y organización de responsabilidades.**

El equipo de desarrollo de esta APP Futespirito está conformada por Bryan Cabrera y Cristian Altamirano dado el tamaño de equipo ambos asumimos

responsabilidades de desarrollo(developers) las cuales se distribuyeron las tareas de Coordinación. Priorizando y revisando de acuerdo a la necesidad de cada Sprint, manteniendo siempre ese objetivo de asegurar incrementos funcionalidades al cierre de cada ciclo.

En términos de especialización técnica se trabajó de forma colaborativa separando actividades por componentes del sistema frontend Backend y base de Datos.


*Figura 7: Product backlog distribuido*

**Planificación por Sprints y entregables**

El trabajo se organizó en sprints priorizando primero funcionalidades habilitadoras estructura base del proyecto, conexión a base de datos luego separado en

gestión y módulos del dominio del futbol ejemplo equipos campeonatos, partidos ,reportes sprint produce un incremento que nos aportó cierto valor a la App.

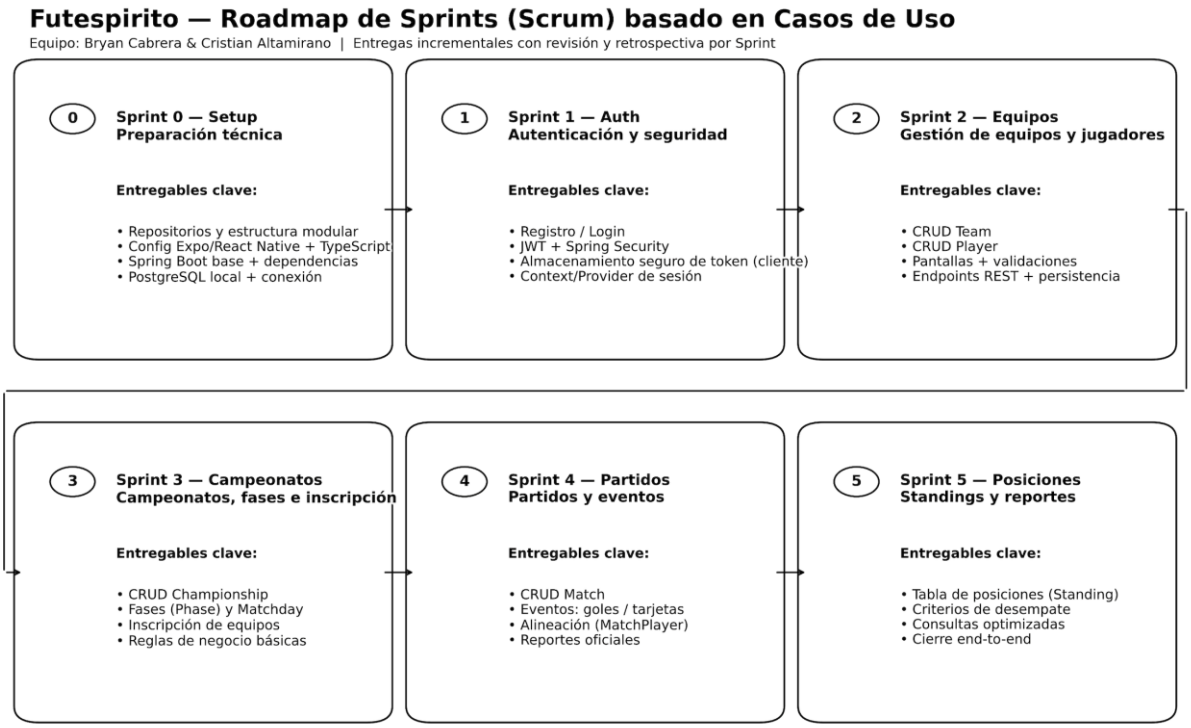



*Figura 8: Sprints futespirito*

# Capítulo II

## Construcción del Sistema

### Desarrollo y Arquitectura en General (Futespirito)

Futespirito se está implementando bajo una arquitectura Cliente-Servidor en donde el frontend se aplica la tecnología de React native incluido Expo ,consumiendo los servicios del Backend  (api rest en SpringBoot) mediante solicitudes HTTP donde se intercambian con nuestros JSON permitiendo una separación clara de responsabilidades el cliente gestiona la interfaz y experiencia del usuario al mismo tiempo el servidor centraliza la lógica del negocio con seguridad y persistencia.

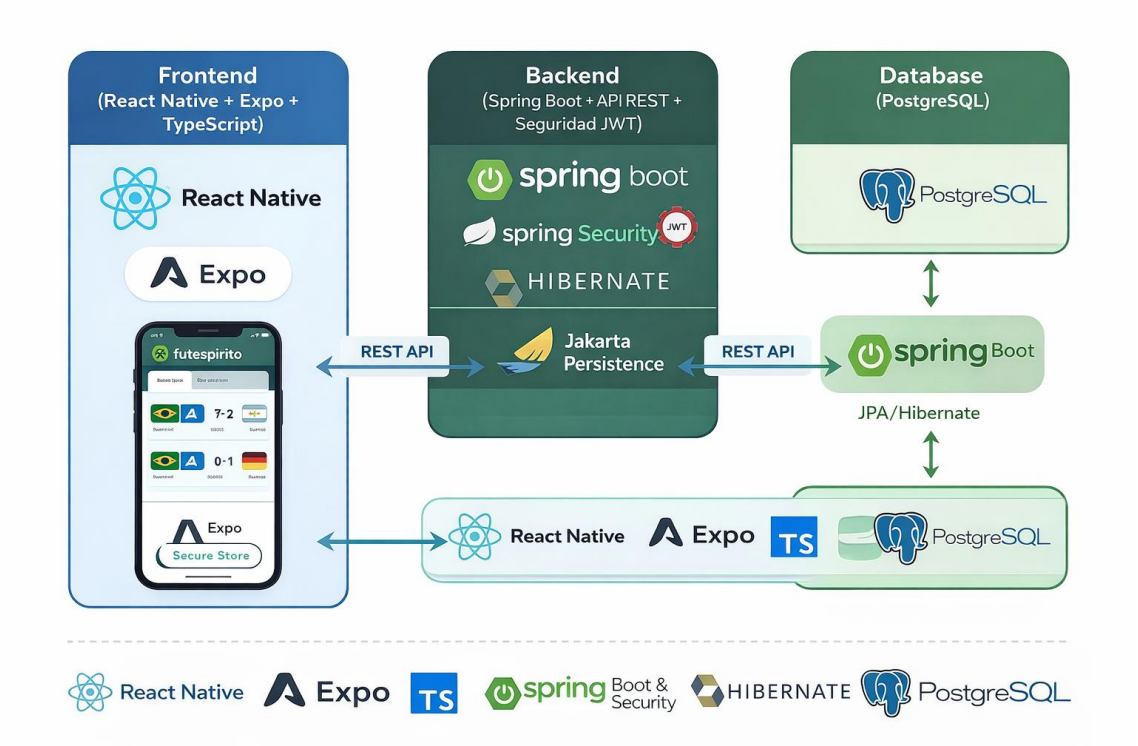



*Figura 9: Implementación de tecnologías*

**Tecnologias utilizadas**

1. **React Native** framework que usamos para construir las interfaces móviles mediante componentes reutilizables con soporte y manejo de los estados hacia nuestros hooks.
2. **Expo** es la plataforma que implementamos para crear las apps para el empaquetado en Android basado en el modelo de negocio.
3. **Type Script** lenguaje que se extiende de Java script tipado estático para mejorar y tener la mantenibilidad y reducción de errores en el desarrollo
4. **Spring Boot** Framework para construir aplicaciones en el backend con java facilitando la configuración y despliegue de los servicios web.
5. **Spring segurity y JWT** la seguridad está basada en la implementación de tokens para la autentificación y la automatización de las seguridades de los usuario y roles.
6. **PostgreSQL** base de datos relacional utilizada para las persistencias de datos conb estructuras basadas en las tablas anteriores mencionadas y SQL.
7. **JPA es un ORM** que mapea las entidades en java donde nos gestiona las tablas relacionales y Gestiona persistencia.

**Metodología de Desarrollo**

***Framework de desarrollo Móvil***

Para el desarrollo de esta aplicación se ha utilizado el framework llamado react native a nivel de frontend, un framework es un marco de trabajo que contiene ciertas

reglas que ya han sido validadas y probadas por otros desarrolladores y permite construir software de manera rápida, ya que la comunidad ha creado muchas librerías que permiten adaptar el desarrollo a lo que se requiera.

Según (Meta, 2024) react native es un framework de desarrollo móvil, el cual permite construir aplicaciones nativas para IOS y para Android, usando javascript y la biblioteca de react adaptada para móviles, este framework permite desarrollar aplicaciones de manera rápida, sin la necesidad de recrear la misma aplicación en los 2 sistemas operativos como los con IOS y Android, adicionalmente integra conceptos existentes en react web como hooks, componentes, react navigator, etc dentro del desarrollo móvil.

***Estructura del Frontend***

El forntend de futespirito está acompañado de expo CLI lo que nos ayuda a optimizar el desarrollo y la depuración hecha en multiplataforma. Además contamos con lenguaje TypeScript que nos ayuda al uso de los componentes, servicios, código y la mantenibilidad de los modelos hacia los componentes.

**Organización por capas y módulos (Futespirito)**

Para una mejor organización del código este se ha estructurado en diferentes capas la cual separa la estructura visual de la lógica, consumo de APIs, navegación, etc, a continuación, se muestran las capas existentes a nivel de frontend.

1. **Capas de Presentacion UI (interfas del usuario)**

Directorio ubicado en src/presentation esta capa contiene toda la información acerca de los componentes visuales como iconos, layouts, componentes y pantallas, en este sentido partimos desde lo más básico como iconos y componentes y terminamos con

la creación de layouts y pantallas, permitiendo que estas puedan ser reutilizadas cuando sea necesario.

La navegación y la creación de providers permite al usuario manejar de manera correcta la lógica de navegación entre las pantallas, así como el uso de modales de carga que permiten continuar y validar errores en el flujo.

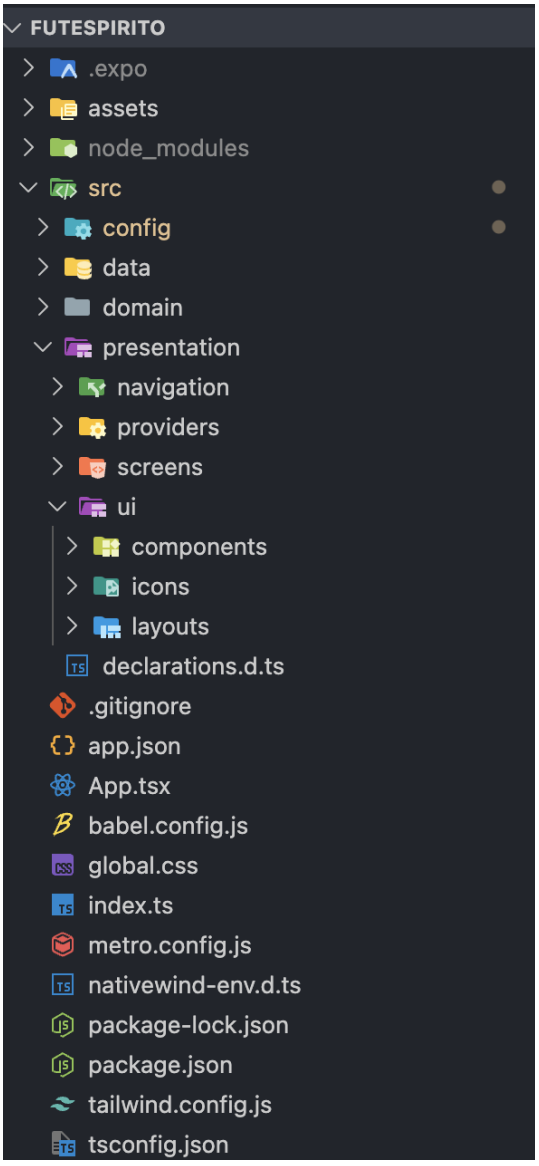



*Figura 10: Vista general del frontend (interfaz de usuario)*

### 2. Capa de Domain

Ubicada en src/domain esta es la capa que organiza la lógica del negocio por módulos como por ejemplo el Auth y el Championship mensionados en las tablas de los módulos anteriormente. Esta capa está acompañada de los Hooks personalizados que encapsulan los flujos del negocio de futespirito como ejemplo esta la autenticación de usuario, aparte de eso incluye el esquema de providers ejemplo el (authContext) que comparte el estado global a nuestra API.

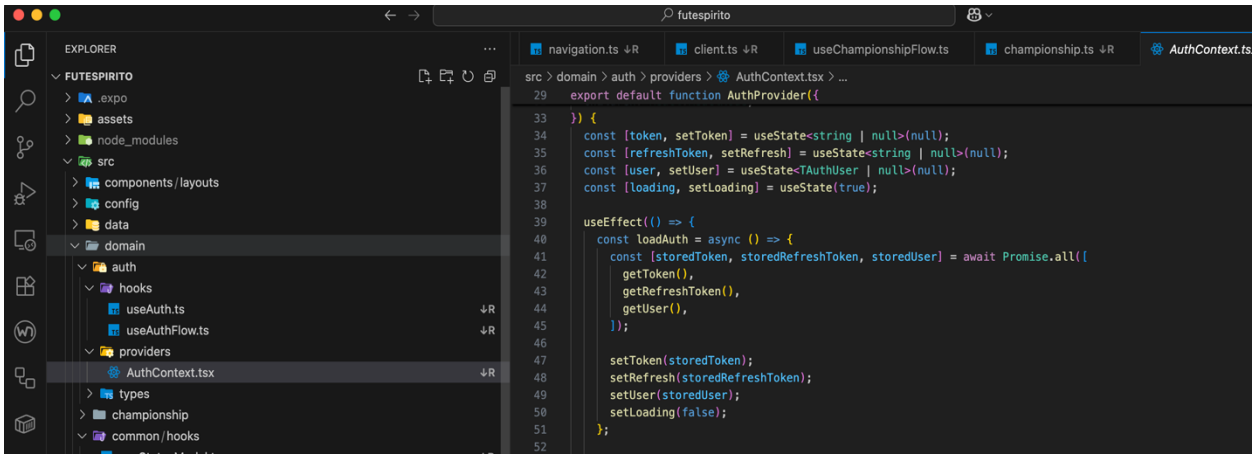



*Figura 11: Capa de Domain*

### 3. Capa de datos

Ubicada en src/data/ esta capa contiene toda la lógica de consumo de servicios rest como por ejemplo los endpoints, client, errores y validaciones de red, adicionalmente tiene toda la estructura relacionada a los llamados de los servicios como los DTO e interfaces requeridas en cada servicio como parte del request y response. Esta capa de datos también contiene el código necesario para la gestión del local storage, el cual es requerido para ejecutar ciertos procesos de futespirito.

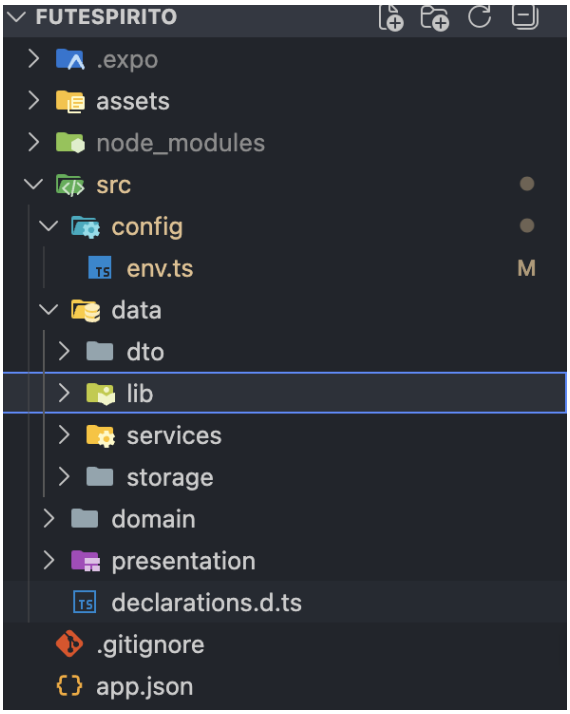



*Figura 12: Capa de Datos*

## 4. Configuracion y entorno

Ubicada en el directorio src/config ejemplo env.ts se configura y centraliza parámetros de URLS y otras constantes que conectan nuestra interfaz gráfica con nuestros servicios web.

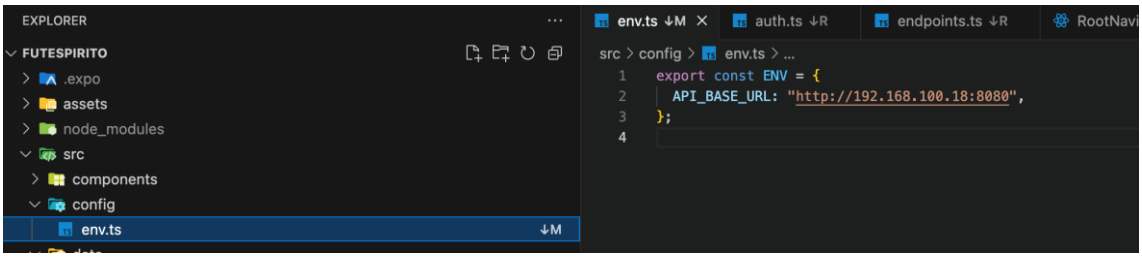



*Figura 13: Configuración del ambiente*

**Manejo del Contexto y Estado**

Para el manejo del contexto y estado en la aplicación se implementó un manejo de estado basado en react hooks y context API, esto con la finalidad de centralizar información compartida entre las diferentes pantallas y componentes, estos hooks que se crearon dentro del proyecto nos permitieron administrar el estado local de cada componente, en especial en casos como la autenticación o manejo de modales.

Entre los principales hooks que utilizamos se encuentran:

- useState: Usado para almacenar datos temporales y establecer banderas de carga o mensajes
- useEffect: Usado para consultar datos de un servicio rest, cargar pantallas o actualizar la información luego de cambiar ciertos valores.
- useMemo y useCallback: Usados para optimizar el renderizado de ciertos componentes.

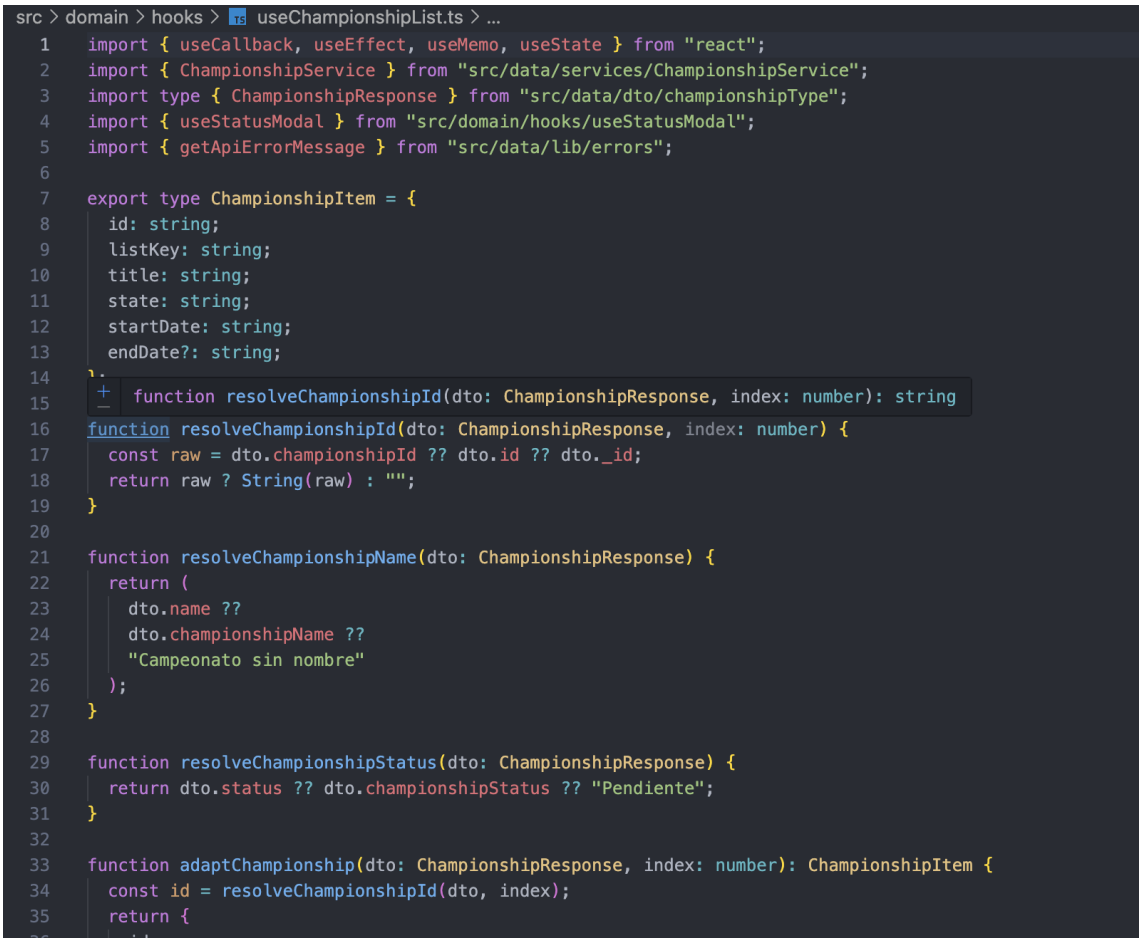



*Figura 14: Uso de Hooks*

Por otro lado, también usamos Context API el cual nos permitió manejar un estado global para aquellos datos que deben ser accesibles desde varias partes de la aplicación como:

- Información del usuario en contexto
- Información del campeonato en contexto
- Token de acceso (Usado para el llamado a otras APIs)

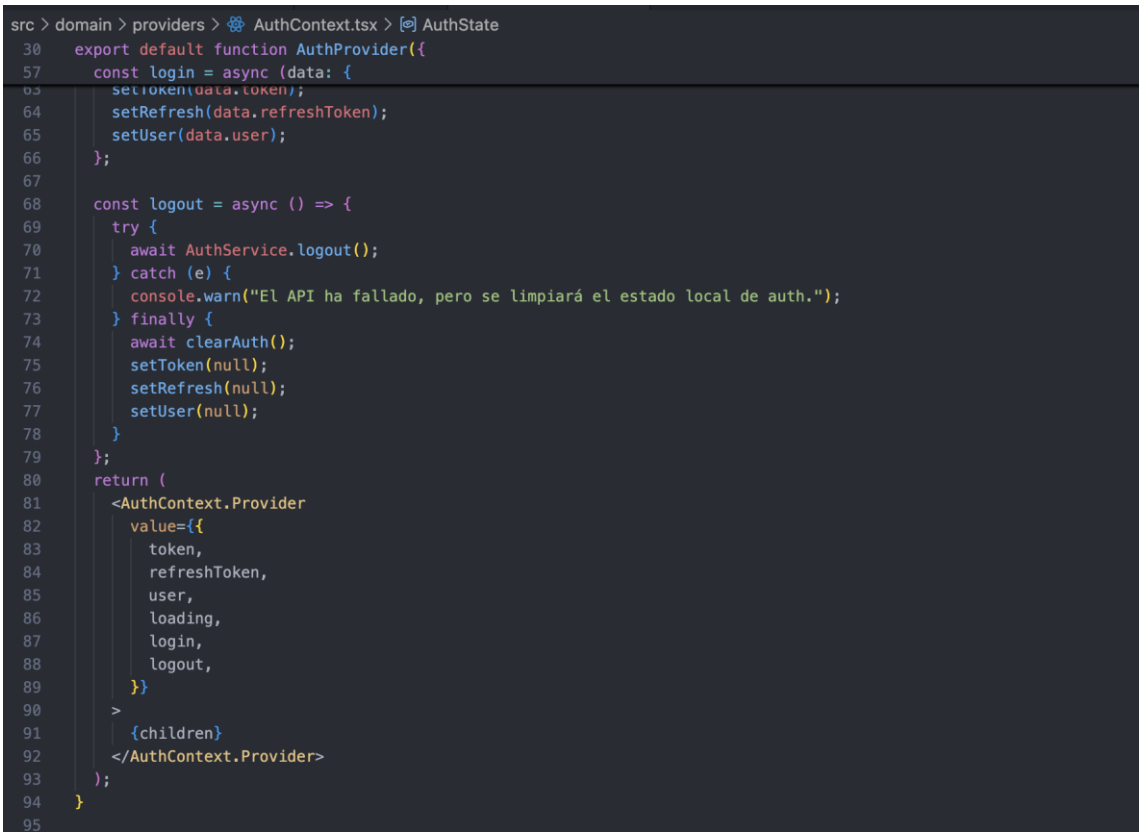



*Figura 15: Uso de Context API*

## Navegación en react native

Para gestionar el flujo de las pantallas dentro de la aplicación de futespirito se utilizó la librería propia de react llamada react navigation, esta es una librería que nos permite implementar una navegación modular y estructurada, ya que nos permite soportar escenarios como navegación por stacks, menús, y navegación basada en la autenticación


Figura 16: Navegación en react native

## Consumo de Servicios Rest

Para el consumo de servicios rest utilizamos la librería de axios, la cual nos permite realizar solicitudes http y https de forma eficiente, a través de esta librería se pudieron realizar las siguientes operaciones:

- Autenticación del usuario
- Obtención de la información de campeonatos, equipos, jugadores y partidos

- Registro de eventos por partido
- Consulta de tabla de posiciones y estadísticas
- Manejo de errores, etc.

src > data > services > ChampionshipService.ts > ...

```ts
import axios from "axios";
import { api } from "src/data/lib/apiClient";
import { ENDPOINTS } from "src/data/lib/endpoints";
import { requestWithTimeout } from "src/data/lib/network";
import { extractMessageFromResponseData } from "src/data/lib/errors";
import type {
  ChampionshipResponse,
  CreateChampionshipRequest,
  CreateChampionshipResponse,
} from "src/data/dto/championshipType";
import type {
  ChampionshipPhaseRequest,
  ChampionshipPhaseResponse,
} from "src/data/dto/championshipPhaseType";

function throwIfErrorPayload(data: any, fallbackMessage: string) {
  const statusCode = (data as any)?.status;
  if (typeof statusCode === "number" && statusCode >= 400) {
    const message = extractMessageFromResponseData(data) || fallbackMessage;
    const error: any = new Error(message);
    error.response = { data };
    throw error;
  }
}

export const ChampionshipService = {
  async create(
    payload: CreateChampionshipRequest
  ): Promise<CreateChampionshipResponse> {
    try {
      const response = await requestWithTimeout(
        api.post<CreateChampionshipResponse>(ENDPOINTS.championship.create, payload)
      );

      const data = response.data;
      throwIfErrorPayload(
        data,
```

*Figura 17: Consumo de servicios rest*

## Framework Backend

Para el backend se utilizó el framework spring boot, este nos ayuda según (VMware, 2026) a crear aplicaciones independientes, listas para producción y basadas

en spring de java, este framework es muy popular debido a su escalabilidad y a su vez a la gran comunidad que tiene, lo cual facilita la creación de aplicaciones o servicios rest.

### *Estructura del Backend*

El backend de futespirito está construido con Spring Boot con el cual exponemos varias APIs para las diferentes funcionalidades del sistema ejemplo: Autenticación, gestión de campeonatos, equipos partidos, calendarios etc. Spring Boot facilita la creación de servicios web y su organización permitiendo estructurar nuestro proyecto de manera clara por capas, paquetes y módulos.

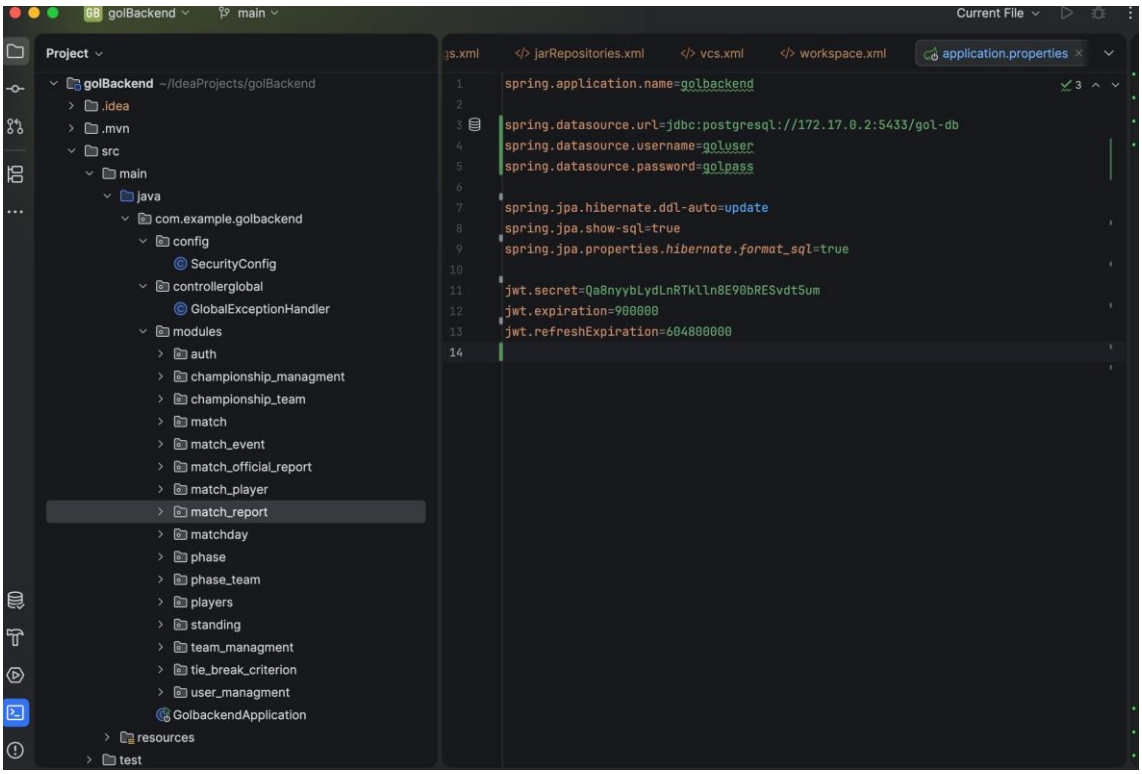



*Figura 18: Estructura del Backend*

El backend se encuentra estructurado de la siguiente manera:

### *Capa Controller*

Este representa el acceso a la aplicación, cada módulo tiene su propio controler y expone los endpoint rest, la principal responsabilidad de controller es recibir solicitudes http (GET, POST, PUT, DELETE) con sus respectivos parámetros o body así lo requiere y retorna respuestas estructuradas.

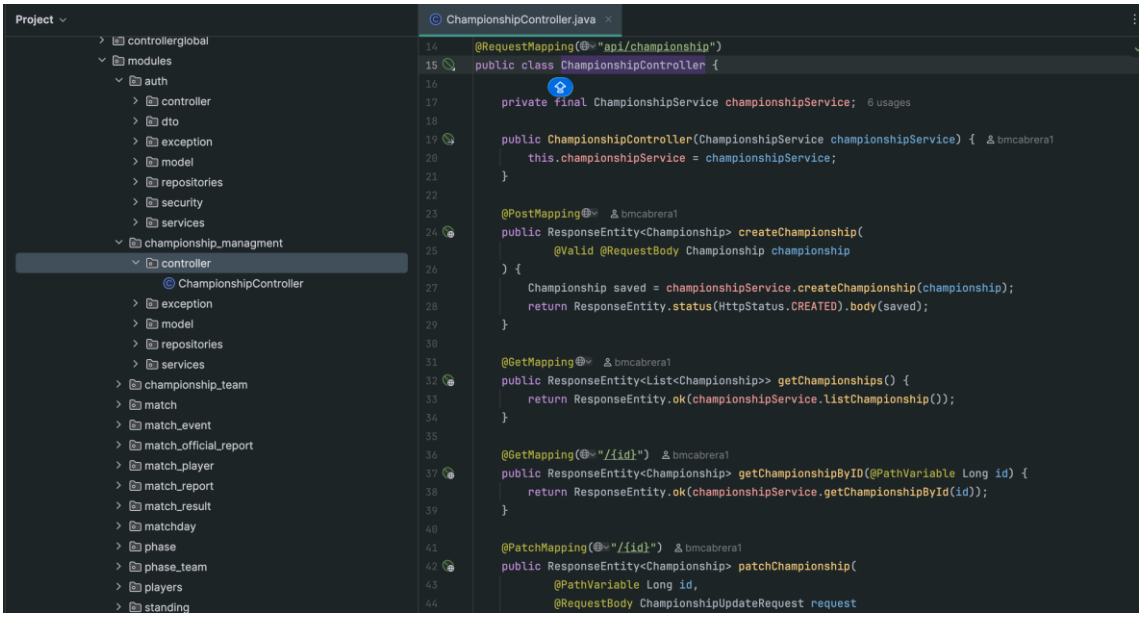



*Figura 19: Capa Controller*

### *Capa Service*

Esta capa contiene la lógica del negocio en el sistema, es la encargada de aplicar todas las reglas y validaciones necesarias, así como también es responsable de implementar las operaciones de guardado, eliminado u operaciones más complejas dentro de mi BDD.

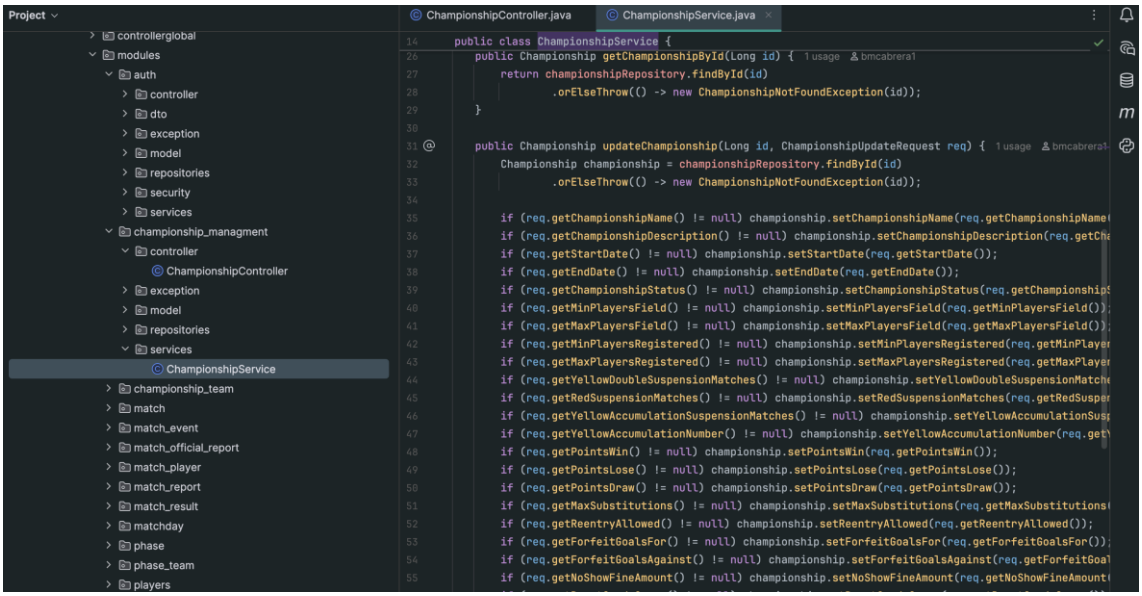



*Figura 20: Capa Service*

### *Capa Repository*

Esta capa se encarga de la persistencia y el acceso a los datos para ello se ha utilizado JPA, lo que permite interactuar con la base de datos, mediante repositorios, esta capa es responsable, de ejecutar consultas hacia la bdd, interactúa con la base de datos eliminando, actualizando o creando registros.

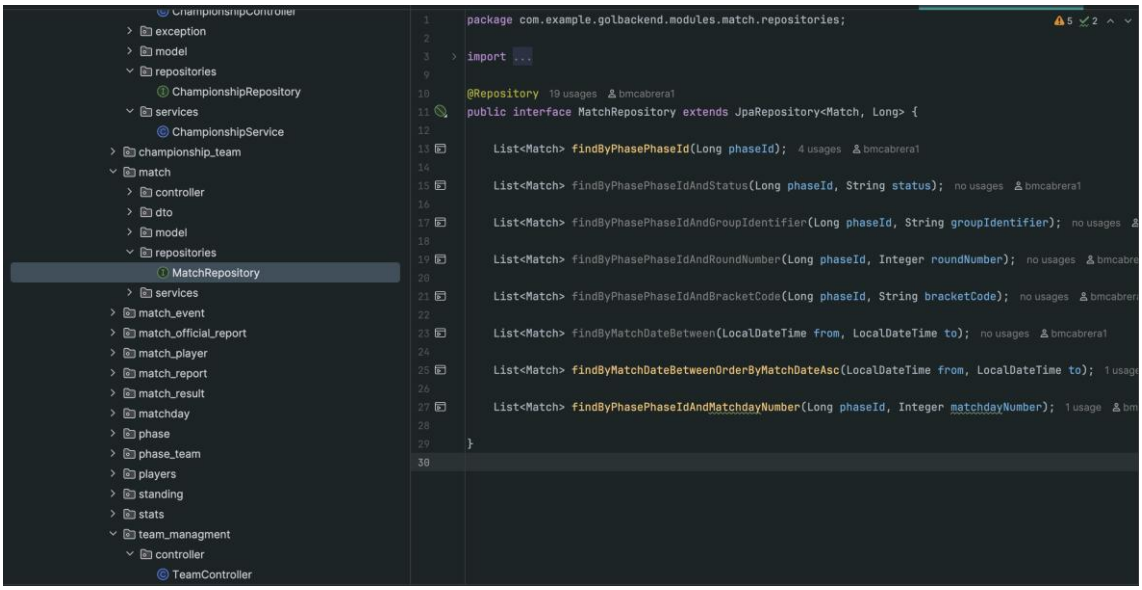



*Figura 21: Capa Repository*

### *Capa DTOs y Model*

La capa model es aquella que define mi entidad como tal y me permite acceder a mis datos usando mi bdd, en muchos de los modelos se utilizando DTOs como una capa adicional para controlar la información que se envía y se recibe a través del APIs proporcionando así una capa de seguridad adicional.

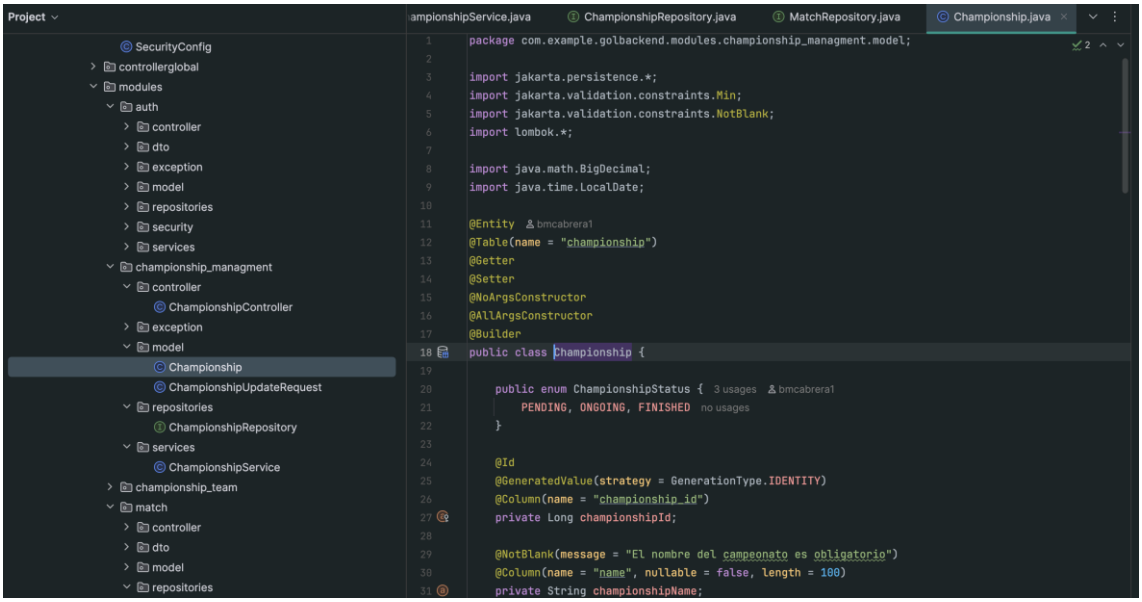



*Figura 22: Capa de entidades y DTOs*

**Organización por módulos funcionales.**

El backend está organizado por fixtures o módulos carpetas o paquetes que separan la lógica por el dominio del negocio en esta estructura vamos a ver módulos como auth, user_management, team_management, Champions_management, Match, Players etc. Estas son nomenclaturas del negocio del futbol.

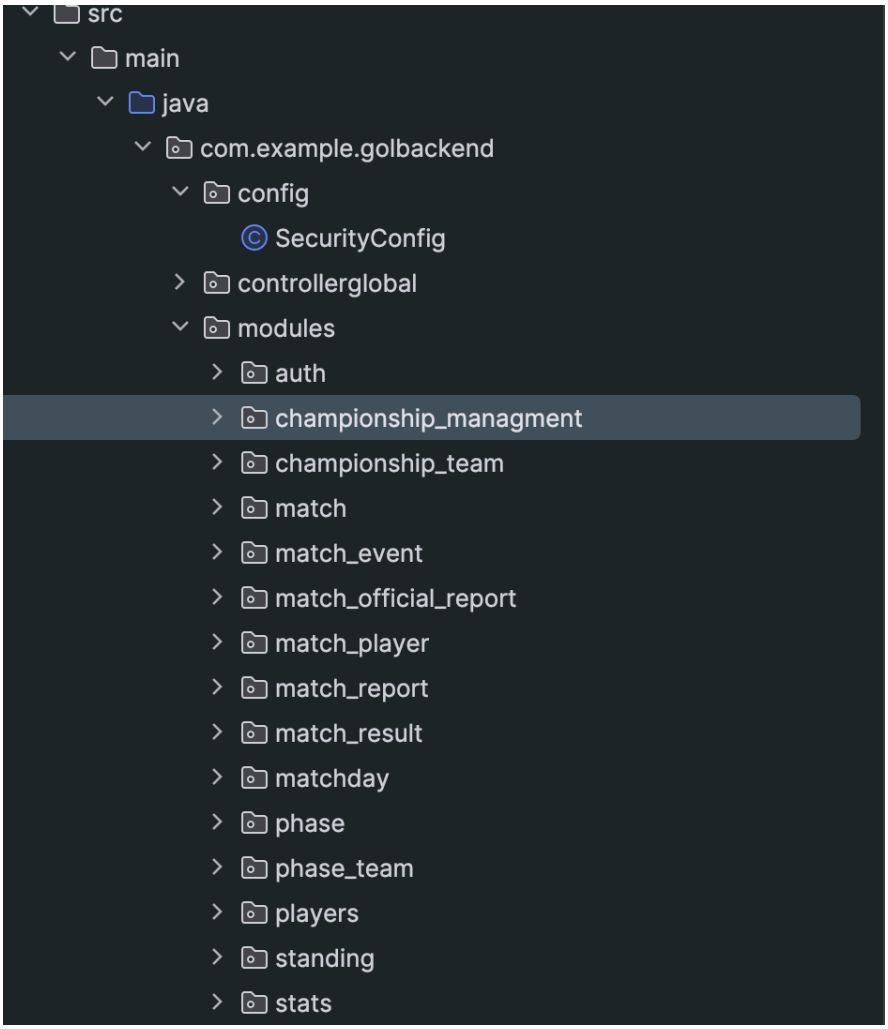



*Figura 23: Organización del backend en módulos*

**Manejo de autenticación del Cliente.**

En la App de Futespirito se gestiona la autenticación de la siguiente manera realizando el almacenamiento mediante tokens, esto permite al usuario mantenerse de manera más segura en los dispositivos móviles.

**Configuración de Seguridad y manejo de errores.**

En cuestión de seguridad en futespirito utilizamos Spring segurity como base de Control de acceso integrando autenticación con JWT (json web tokens) en la aplicación.

El manejo de excepciones **se** realiza con el uso de un GlobalExceptionHandler, el cual da ese enfoque centralizado para transformar errores en respuestas HTTP consistentes.

Para la persistencia el backend utiliza el estándar Jakarta Persistence junto al ORM hibernate para mapear las entidades del dominio a las relaciones de las tablas en PostgreSQL.

**Base de Datos PostgreSQL**

En lo que corresponde a nuestra APP futespirito gestionamos la base de datos Relacional donde la estructura se basa en tablas (create Table) donde estas se relacionan para facilitar la integridad referencial del dominio (usuarios Equipos Campeonatos Partidos Jugadores Posiciones etc.

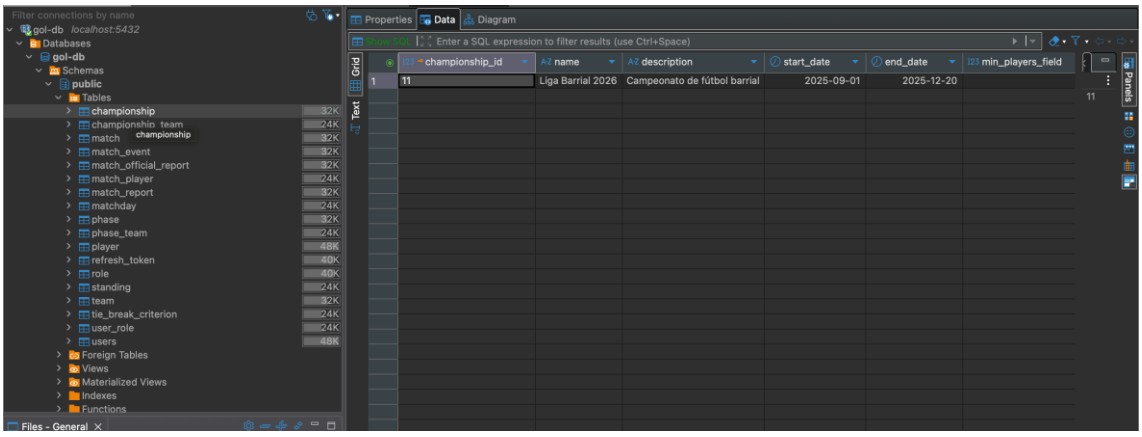



*Figura 24: Esquema de base de datos*

**Herramienta de control de versión**

La herramienta de control de versión utilizada fue Github según (GitHub, 2026) esta es una plataforma que almacenar, escribir y compartir código junto a otros usuario, facilitando el trabajo entre varías personas, dentro de esta herramienta lo que se realizo fue crear 2 repositorios, un repositorio para el backend y otro para el frontend, en cada

uno de los repositorios se crearon diversas ramas las cuales no permitieron organizar de mejor manera el desarrollo de los diferentes features.

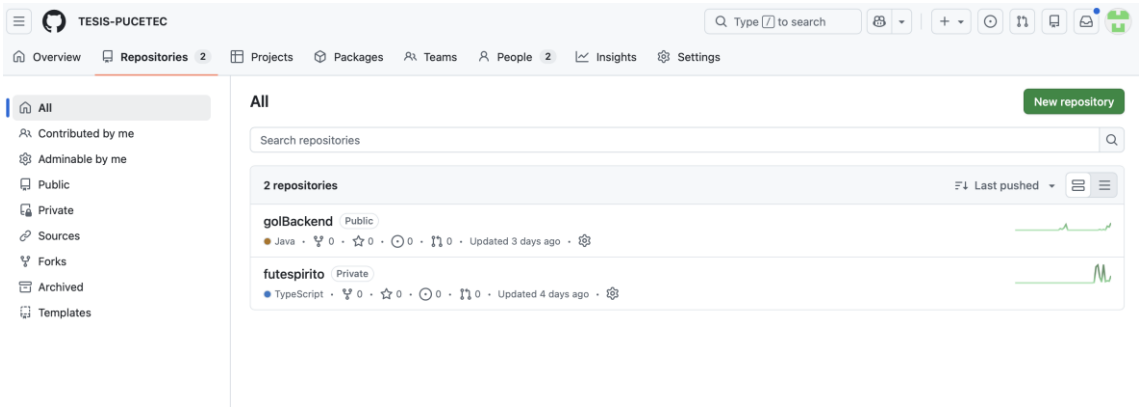



*Figura 25: Repositorios de futespirito*

# Capítulo III

## Pruebas y Estabilización

Para verificar el correcto funcionamiento del App móvil y del backend, se ejecutaron pruebas orientadas a verificar los principales procesos de la misma como:

- Registro y Logín de usuarios
- Registro de campeonatos
- Configuración de fases y equipos
- Configuración de sorteo
- Creación de jugadores y equipos
- Carga de imágenes como Logos y Fotos
- Consulta de partidos, tablas de posiciones etc.

Para las pruebas se utilizaron las siguientes herramientas como postman, expo go y la base de datos postgreSQL para validaciones de persistencia.

### Resultados de las pruebas ejecutadas

Se ejecutaron un total de 56 casos de pruebas como parte de pruebas de la aplicación y los resultados fueron los siguientes:

| Módulo | Total de Casos de Pruebas | Resultados |
|---|---|---|
| Login y Registro | 7 | 7 Aprobados |
| Creación de Campeonatos | 5 | 4 Aprobados<br>1 Issue Reportado |
| Configuración de fases | 4 | 4 Aprobados |

| | | |
|---|---|---|
| Creación de sorteos | 4 | 2 Aprobados<br>2 Issues Reportados |
| Creación y edición de equipos | 6 | 4 Aprobados<br>2 Issues Reportados |
| Creación y edición de jugadores | 6 | 6 Aprobados |
| Registro de eventos y hojas de vocalía | 10 | 7 Aprobados<br>3 Issues Reportados |
| Actualización y Visualización de estadísticas | 3 | 3 Aprobados |
| Actualización y Visualización de partidos | 3 | 2 Aprobados<br>1 Issue reportado |
| Actualización y Visualización de tabla de posiciones | 3 | 3 Aprobados |
| Generación de reportes | 5 | 5 Aprobados |

Tabla 1: Resultados Casos de pruebas

**Casos de prueba ejecutados**

A continuación, se detallan alguno de los casos de pruebas que se ejecutaron:

| **Campo** | **Detalle** |
|---|---|
| Nombre del Caso | Realizar Login en el sistema |
| Objetivo | Se requiere realizar el login dentro del sistema. |

| | |
|---|---|
| Precondiciones | API activa, usuarios registrados |
| Resultado Esperado | Se debe realizar el login esperando que se realicen todas las validaciones a nivel de formulario, como usuario con email valido, password con la longitud correcta, al finalizar el login debe aparecer un mensaje de confirmación indicando que el login fue exitoso |
| Resultado Obtenido | Aprobado |

Tabla 2: Caso de prueba 1

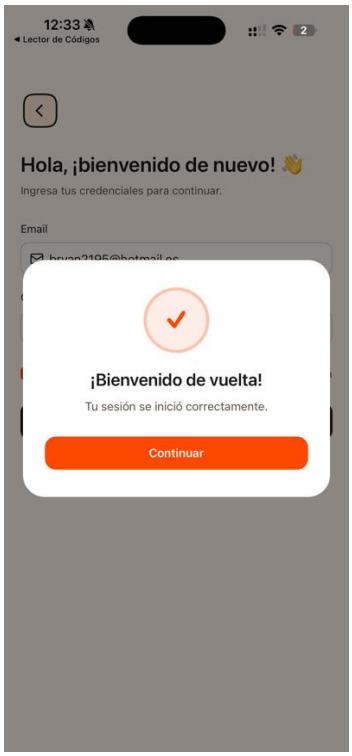



*Figura 26: Login*

| **Campo** | **Detalle** |
|---|---|
| Nombre del Caso | Realizar Registro en el sistema |
| Objetivo | Se requiere realizar el registro dentro del sistema. |
| Precondiciones | API activa, usuarios no registrados |
| Resultado Esperado | Se debe realizar el registro esperando que se realicen todas las validaciones a nivel de formulario, como usuario con email valido, password con la longitud correcta, longitud mínima en nombre, apellido, al finalizar el registro debe aparecer un mensaje de confirmación indicando que el registro fue exitoso |
| Resultado Obtenido | Aprobado |

Tabla 3: Caso de prueba 2

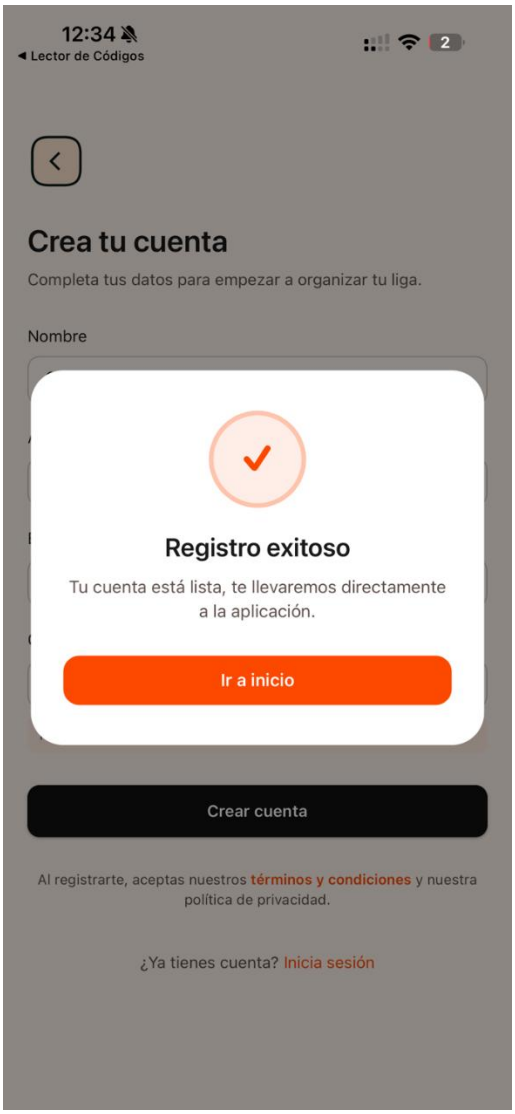



*Figura 27: Registro*

| **Campo** | **Detalle** |
|---|---|
| Nombre del Caso | Crear un campeonato |
| Objetivo | Se requiere realizar la creación del campeonato dentro del sistema. |
| Precondiciones | API activa |

| | |
|---|---|
| Resultado Esperado | Se debe verificar que el formulario valide todas las reglas del negocio y permita la creación del campeonato. |
| Resultado Obtenido | Fallido: Issue reportado en la pantalla, no todos los títulos son iguales a nivel de tipo grafica. |

Tabla 4: Caso de prueba 3

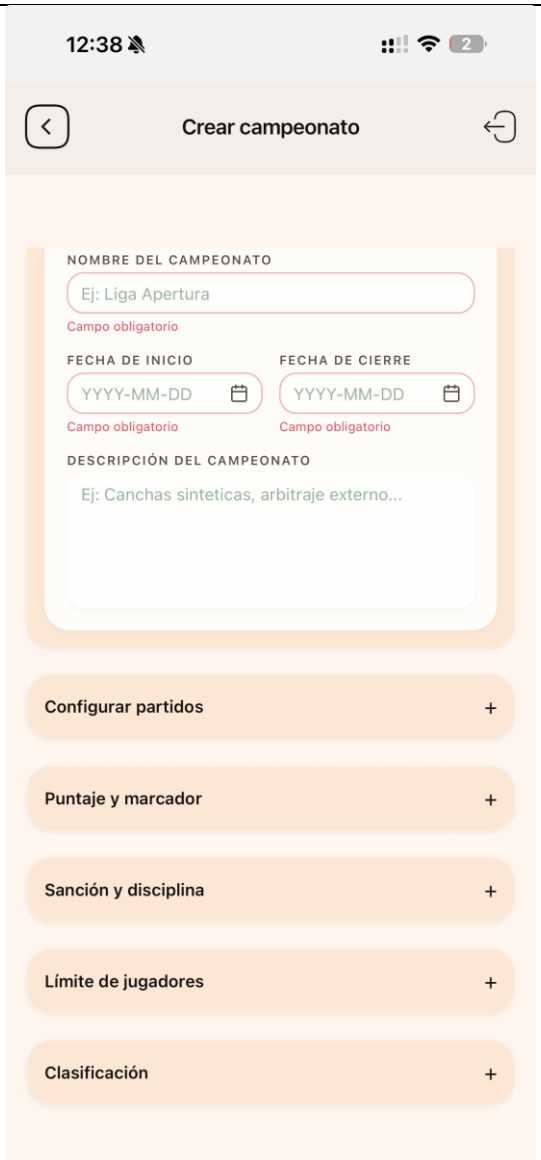



*Figura 28:Creación Campeonatos*

| Campo | Detalle |
| --- | --- |
| Nombre del Caso | Creación de un equipo con logo |
| Objetivo | Se requiere crear un equipo con logo. |
| Precondiciones | API Disponible, Cloudinary service disponible. |
| Resultado Esperado | Se debe poder pasar las validaciones del formulario para crear un equipo, adicionalmente se debe permitir la carga de un logo. |
| Resultado Obtenido | Aprobado |

Tabla 5: Caso de prueba 4


*Figura 29: Creación Equipo*

| **Campo** | **Detalle** |
|---|---|
| Nombre del Caso | Verificar tabla de posiciones |
| Objetivo | Se requiere verificar que la tabla de posiciones se muestre de manera correcta. |

| | |
|---|---|
| Precondiciones | API activa, resultados registrados |
| Resultado Esperado | Se debe acceder a la sección de tabla de posiciones y se debe verificar que los resultados, goles a favor, goles en contra y número de partidos jugadores |
| Resultado Obtenido | Fallido, se reportó un issue ya los goles registrados no se muestran en la tabla de posiciones. |

Tabla 6: Caso de prueba 5

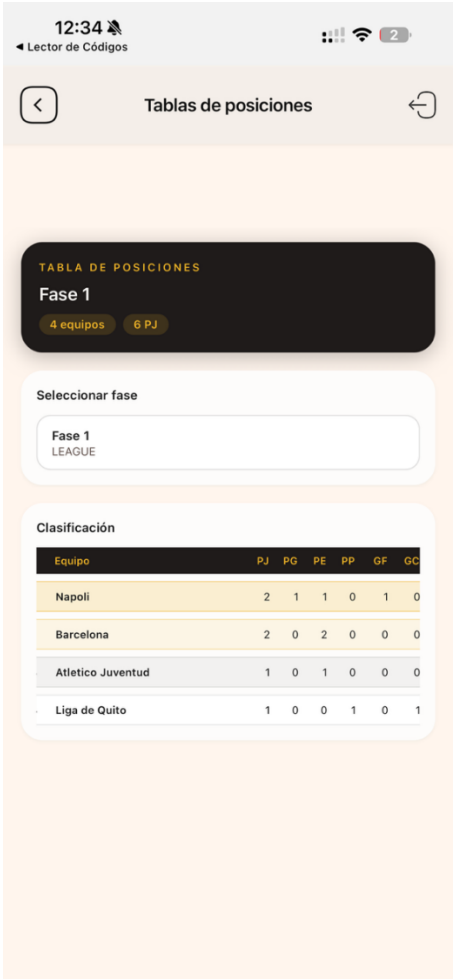



*Figura 30: Tabla de Posiciones*

**Conclusiones**

La implementación del backend bajo un enfoque basado en microservicios permitió estructurar las funcionalidades del sistema de manera modular y ordenada, facilitando la separación de responsabilidades entre componentes, optimizando el mantenimiento del código y permitiendo una integración más eficiente con el frontend móvil. Adicionalmente, este enfoque habilita una escalabilidad futura más controlada, ya que cada servicio puede evolucionar o expandirse sin comprometer el funcionamiento general del sistema.

El uso de contenedores mediante Docker Compose permitió estandarizar el entorno de ejecución del sistema, asegurando consistencia entre los ambientes de desarrollo, pruebas y producción. Esta estrategia redujo significativamente problemas asociados a diferencias de configuración, dependencias o versiones, mejorando la confiabilidad del despliegue y disminuyendo el tiempo requerido para replicar el sistema en distintos entornos.

La correcta definición de variables de entorno, junto con el uso de volúmenes para persistencia de datos, permitió que el backend pueda reiniciarse o actualizarse de manera segura en entornos productivos. Esto garantizó continuidad operativa durante despliegues o reinicios del servidor, evitando pérdida de información y fortaleciendo la estabilidad general del sistema.

El desarrollo del frontend mediante React Native permitió optimizar el tiempo de implementación al construir una aplicación multiplataforma con una sola base de código. Esta característica facilitó el despliegue tanto en iOS como en Android, reduciendo el

esfuerzo de mantenimiento y asegurando una experiencia consistente para el usuario final, independientemente del sistema operativo utilizado.

La implementación de componentes reutilizables dentro del frontend contribuyó significativamente a la escalabilidad y mantenibilidad del proyecto, ya que permitió estandarizar patrones de diseño y comportamiento en la interfaz. Esto redujo la duplicidad de código, mejoró la consistencia visual de la aplicación y facilitó la incorporación de nuevas funcionalidades sin afectar la estabilidad del sistema.

**Recomendaciones**

Se recomienda fortalecer el proceso de despliegue en producción incorporando un flujo de versionamiento y automatización que permita aplicar cambios de manera controlada, reduciendo riesgos operativos y asegurando que cada actualización del sistema mantenga estabilidad y trazabilidad.

Se sugiere implementar mecanismos de monitoreo y análisis de logs tanto en el backend como en la infraestructura del servidor, con el fin de identificar de forma temprana posibles fallos, consumos anómalos de recursos o comportamientos inesperados que puedan afectar la continuidad del servicio.

Es recomendable consolidar una estrategia formal de seguridad, considerando el manejo adecuado de credenciales, el uso de HTTPS, el control de accesos y la exposición mínima de puertos, garantizando que el sistema pueda operar en un entorno productivo con menores vulnerabilidades.

Se aconseja continuar con la optimización del frontend mediante prácticas orientadas a rendimiento y experiencia de usuario, tales como validaciones más robustas, manejo eficiente de listas, reducción de renders innecesarios y pruebas en dispositivos reales, con el objetivo de asegurar un funcionamiento fluido en diferentes condiciones y equipos.

## Referencias bibliográficas


Khalid, S., Rasheed, U., & Khaleeq, U. (2025). Ensuring quality in software requirement engineering process: A comparative study. *ScienceDirect*. doi:https://doi.org/10.1016/j.eij.2025.100754

Juura, T. (2024). Artificial Intelligence in Requirements engineering. *Artificial Intelligence in Requirements engineering*. University of Technology LUT.

Ph.D, R. S. (2010). *INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO.* MÉXICO : María Teresa Zapata Terrazas.

Sommerville, I. (2011). *INGENIERÍA DE SOFTWARE.* Mexíco: Luis M. Cruz Castillo.

Grady Booch, I. J. (2000). *EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE.* Madrid: Universidad de Málaga.

Mazurkiewiz, E. (10 de Mayo de 2023). *Functional vs non-functional requirements*. Obtenido de inwedo: https://inwedo.com/blog/functional-and-non-functional-requirements/?utm_source=chatgpt.com

Mehta, A., Mehta, N., & Bindal, I. (2022). Maximizing integrative learning in software development teams: A systematic review of key drivers and future research agenda. *Journal of Systems and Software*, 4.

Bellavista, P. (2024). A Review of Non-Functional Requirements Analysis Throughout the SDLC. *MDPI Open Access Journal*, 2.

Altexsoft Editorial Team. (29 de Diciembre de 2023). *Nonfunctional Requirements in Software Engineering: Examples, Types, Best Practices*. Obtenido de Altexsoft: https://www.altexsoft.com/blog/non-functional-requirements/

Sidana, N., Harminder, K., & Devi, P. (2022). Schiff Base Modified Paper Test Strips for Naked Eye Detection of Copper Ions in Mixed Aqueous Media. *IEEE Sensors Journal*, 4.

Chen, P. P. (1976). *ACM Transactions on Database Systems.* Estados Unidos: Toward a unified.

Coronel, C. &. (2019). *Database Systems: Design, Implementation, and Management.* Estados Unidos: Cengage.

Meta. (2024). *React Native*. Obtenido de React Native: https://reactnative.dev/

VMware. (2026). *Spring Boot Reference Documentation*. Obtenido de Spring Boot: : https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/

GitHub. (2026). *Acerca de GitHub y Git*. Obtenido de GitHub Docs: https://docs.github.com/es/get-started/start-your-journey/about-github-and-git