Pontificia Universidad Católica del Ecuador

Facultad De Ingeniería

Escuela de Sistemas

TEMA:

DISEÑO Y DESARROLLO DE UN PROTOTIPO FUNCIONAL DE UN SISTEMA WEB PARA LA GESTION DE LIGAS BARRIALES

AUTOR:

MATHEW DAVID VARGAS CASTILLO

TRABAJO PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS Y COMPUTACIÓN

QUITO, 2023

DEDICATORIA

Este trabajo va dedicado a mis padres y hermana, quienes siempre han estado a mi lado apoyándome y guiándome en cada paso que he dado en mi vida académica y personal. Su amor incondicional, dedicación y sacrificios han sido la fuerza impulsora detrás de mis logros.

AGRADECIMIENTO

Quiero agradecer principalmente a mis padres, ya que siempre han sido una gran inspiración para mí.

También me gustaría agradecer a mis amigos que siempre están ahí para mí y me apoyan en cada proyecto y aventura en la que participo.

RESUMEN

El presente trabajo se enfoca en las necesidades que tienen las ligas barriales actuales y cómo pueden ser satisfechas mediante el desarrollo de sistemas web. Se identifica una necesidad no cubierta que existe en la gran mayoría de ligas barriales de Quito y del país, lo que lleva al diseño y creación de un sistema web para la administración de las diferentes actividades que se realizan dentro de las ligas barriales. Posteriormente, con la ayuda de la investigación bibliográfica pertinente, se desarrolló un sistema web utilizando las herramientas React y Firebase junto con la metodología XP para el desarrollo. Una vez finalizada la aplicación se concluye que el proceso de desarrollo del sistema requiere de varias etapas, incluida la recopilación de información para realizar un trabajo específico y acorde a los requerimientos del usuario final. Al finalizar el trabajo, se obtuvo como resultado un prototipo funcional de un sistema web que se ajusta con los requerimientos y necesidades de las ligas barriales. Este prototipo puede ser utilizado como base para futuros desarrollos y adaptaciones según las particularidades de cada liga barrial.

ÍNDICE

Tabla de contenido

ÍNDICE DE FIGURAS, GRÁFICOS Y TABLAS IX

INDICE DE ILUSTRACIONES IX

ÍNDICE DE TABLAS X

CAPÍTULO I: INTRODUCCIÓN 1

1. MARCO DE REFERENCIA 1

1.1 JUSTIFICACIÓN 1

1.2 Planteamiento del problema 1

1.3 Objetivo General 2

1.3.1 Objetivos Específicos 2

1.4 Antecedentes 3

1.5 Alcance 4

CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA 5

2. Marco Teórico 5

2.1 Generalidades Marco Espacial 5

2.2 Desarrollo web 5

2.2.1 Desarrollo Frontend 6

2.2.2 Desarrollo Backend 6

2.2.3 Framekork 7

2.2.4 React 8
2.2.5 Firebase 8
2.3 Metodologías de desarrollo 10
2.3.1 Metodología XP 10
2.3.2 Ventajas de XP 10
2.4 Lenguaje Unificado de Modelado 11
2.5 Casos de uso 12
2.6 Gestión deportiva 12
2.7 Ligas Barriales 12
2.8 Diseño y desarrollo de prototipos 13
CAPÍTULO III: METODOLOGÍA 14
3. Metodología de desarrollo del plan de tesis 14
3.1 Investigación Cualitativa 14
3.2 Investigación Aplicativa 15
3.3 Metodología de desarrollo de software 16
3.3.1 Modelo que aplica 16
3.3.1.1 Metodología ágil 16
3.3.1.2 Extreme Programming: 16
CAPÍTULO IV: DESARROLLO DE LA INVESTIGACIÓN 18
4. Desarrollo 18
4.1 Selección Front-end 18

4.1.1. React 18
4.1.2. Angular 18
4.1.3 Comparativa React-Angular 18
4.2. Fase 1: Exploración 19
4.2.1 Alcance y Módulos del proyecto 19
4.2.2 Requerimientos Funcionales 19
4.2.3 Requerimientos No Funcionales 20
4.2.4 Riesgos Identificados 20
4.2.5 Procesos de Gestión de Riesgos 20
4.2.6 Herramientas a utilizar 21
4.3 Fase de Planificación 21
4.3.1 Consolidación de Requerimientos 22
4.3.2 Plan de entregas 22
4.3.3 Fase de Iteraciones 23
4.3.4 Primera Iteración 23
4.3.4.1 Requerimientos Primera Iteracion 23
4.3.4.2 Tareas primera iteración 24
4.3.5 Segunda Iteración 24
4.3.5.1 Requerimientos Segunda Iteración 24
4.3.5.2 Tareas Segunda Iteración 25
4.3.6 Tercera Iteración 25

4.3.6.1 Requerimientos tercera iteración 25
4.3.6.2 Tareas Tercera Iteración 25
4.4 Fase de Diseño 26
4.4.1 Arquitectura 3 capas 26
4.4.2 Casos de Uso 26
4.4.3 Diseño de Interfaces 29
4.5 Fase de codificación 33
4.5.1 Estándares de Codificación 33
4.5.2 Código 34
4.6 Pruebas 34
4.6.1 Pruebas de aceptación 35
4.5.1 Pruebas de caja blanca 37
CONCLUSIONES Y RECOMENDACIONES 41
Conclusiones 41
Recomendaciones 42
BIBLIOGRFÍA 42
Bibliografía 44
GLOSARIO DE TÉRMINOS 46
ANEXOS 48
Anexo A: Pantalla Inicio de sesión 48
Anexo B: Pantalla principal 48

Anexo C: Categorías de la liga.....49

Anexo D: Formulario de contacto.....50

Anexo E: Pantalla CRUD jornadas.....50

Anexo F: Pantalla CRUD equipos.....51

Anexo G: Pantalla Tabla de posiciones.....51

ÍNDICE DE FIGURAS, GRÁFICOS Y TABLAS

INDICE DE ILUSTRACIONES

Ilustración 1 – Caso de Uso Actividad de Usuario 27

Ilustración 2 – Caso de uso Actividad administradores.....28

Ilustración 3 – Diagrama de actividad Usuario.....29

Ilustración 4 – Diseño Usuarios 30

Ilustración 5 – Diseño Equipos categoría.....30

Ilustración 6 – Tabla de posiciones.....31

Ilustración 7 – Jornadas 32

Ilustración 9 – CRUD Administrador 32

ÍNDICE DE TABLAS

Tabla 1 18

Tabla 2 21

Tabla 3 35

Tabla 4 36

Tabla 5 36

Tabla 6 37

Tabla 7 38

Tabla 8 38

Tabla 9 39

Tabla 10 39

CAPÍTULO I: INTRODUCCIÓN

1. MARCO DE REFERENCIA

1.1 JUSTIFICACIÓN

Se plantea desarrollar un sistema web cuyo principal enfoque es la gestión de información de las ligas barriales dentro del Distrito Metropolitano de Quito. Este desarrollo se vuelve importante debido a que muchas de estas ligas carecen de sistemas de información para la visualización de datos que son generados en cada una de las fechas que se juegan dentro de cada campeonato que posee la liga.

De ahí viene la necesidad de desarrollar un sistema y que de esta manera se pueda contar con una herramienta que ayude a gestionar las actividades de las ligas, de manera que vuelva mucho más sencillo que los dirigentes de la liga compartan información de cada club con cada uno de los jugadores pertenecientes a la liga.

El sistema brindará los datos y estadísticas que necesita conocer cada club, y de esta forma, la información será accesible para todos.

1.2 Planteamiento del problema

En la actualidad, el fútbol se encuentra entre los deportes más populares y practicados en Ecuador. Entre una de sus subdivisiones más importantes encontramos a las ligas barriales, donde juegan una gran cantidad de personas para hacer deporte, competir o simplemente divertirse. En el país, se ha observado un aumento en la cantidad de ligas barriales, así como en la cantidad de equipos y jugadores que participan en cada una de ellas en los últimos años.

De acuerdo con el Distrito Metropolitano de Quito, actualmente existen 308 libas barriales y parroquiales que deben gestionar las actividades que llevan a cabo y los resultados obtenidos

por cada equipo dentro de las diferentes competiciones que se disputan al año. Muchas de estas cuentan con sistemas de información que les ayudan a gestionar estos aspectos, pero muchas otras ni siquiera llevan un registro de los jugadores o los partidos que se jugaron, por lo que surge la necesidad de automatizar estos procesos para que se puedan llevar de manera fácil y sencilla.

Muchas veces los jugadores de estas ligas pueden sentirse inconformes o tener problemas para ver los resultados de sus equipos, así como una tabla de posiciones o un calendario de partidos futuros. La mayoría de estas ligas gestionan esto a través de grupos de WhatsApp, donde los directivos son los únicos que reciben la información para comunicarse con el resto del equipo.

Con los previamente dicho, el objetivo de este desarrollo es resolver el problema mencionado anteriormente, ya que para la mayoría de los jugadores resulta difícil encontrar donde ver los datos de su equipo, los puntos que han sumado y su posición en la tabla del campeonato, así como ver las próximas jornadas que se disputarán. Por lo tanto, la aplicación que se está desarrollando tiene como objetivo satisfacer las necesidades de los jugadores y directivos de las ligas barriales.

1.3 Objetivo General

Analizar, diseñar y crear un sistema web para administrar los diversos procedimientos, actividades y campeonatos que se llevan a cabo en las ligas barriales.

1.3.1 Objetivos Específicos

- Determinar cuáles son los principales procesos de gestión y administración que llevan a cabo las ligas barriales para de esta manera dimensionar el alcance de la aplicación.

- Controlar y regular el proceso de inscripción y registro de jugadores dentro de las ligas barriales.
- Evaluar el funcionamiento de la aplicación mediante pruebas de funcionalidad y usabilidad.
- Diseñar interfaces amigables y fáciles de usar para el usuario final.

1.4 Antecedentes

En la actualidad varias de las ligas barriales se han encontrado con varios inconvenientes para gestionar sus actividades como la planificación de eventos, administración de recursos y comunicación entre sus miembros.

Algunos estudios abordan la importancia de que exista una eficiente gestión, como dice (Fernández, RRHDigital, 2019) “La organización dentro de las empresas es fundamental para conseguir alcanzar nuevas metas y objetivos. Contar con herramientas de gestión adecuadas puede ser determinante para que nuestra estrategia sea un éxito o un fracaso”, por lo que contar con herramientas de apoyo para dicha gestión se vuelve fundamental dentro de cualquier liga barrial.

Como respuesta a esta necesidad se han desarrollado varios sistemas de apoyo a la gestión de ligas barriales, el problema se encuentra en que muchos de estos suelen ser caros, poco accesibles y no se adaptan a necesidades específicas de la organización. Por ende, es necesario desarrollar un sistema que se acople a las necesidades de las ligas barriales para que de esta forma puedan mejorar su planificación y gestión de actividades.

Con dicho contexto, surgió la idea de desarrollar un sistema web para las ligas barriales que ayude a estas instituciones a administrar y planificar sus proyectos de forma más eficiente, centrándose en diseñar un sistema de gestión accesible que se acople a los requerimientos

específicos de cada liga y ayude a mejorar la eficiencia en la planificación y gestión de actividades.

1.5 Alcance

A fin de simplificar y agilizar los procesos para registro de jugadores y visualización de resultados de los campeonatos de las ligas barriales, se ha desarrollado un sistema web utilizando el framework de React, que proporciona las herramientas necesarias para el proyecto. La aplicación podrá ser utilizada tanto por los directivos de las ligas barriales como por los jugadores los equipos que forman parte de la liga.

El diseño del sistema web se ha concebido para ofrecer interfaces amigables, sencillas de entender y fáciles de usar. Al ejecutar el sistema, encontraremos los principales módulos desarrollados, que son el de inicio de sesión y registro, recopilación de la información de cada jugador que se inscriba en el equipo. También se incluye un calendario de partidos, donde se pueden visualizar las fechas y lugares de cada partido por equipo, así como los resultados de cada encuentro.

La aplicación cuenta con una tabla de posiciones general, donde se pueden visualizar los puntos y la posición de cada equipo.

CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA

2. Marco Teórico

2.1 Generalidades Marco Espacial

El ámbito de referencia del proyecto es Quito, ya que es uno de los cantones con mayor número de ligas barriales del país, contando con 308 ligas deportivas, donde se desarrollan los principales actores y procesos a los que buscamos darle solución con la creación del sistema web.

El análisis básico de los requerimientos se realizará de forma local, tomando como base y ejemplo las actividades que se realizan dentro de las Ligas Barriales de Cotocollao y San Carlos, para identificar los fundamentos más importantes y necesarios para gestionar la liga dentro del sistema. Se realizó un análisis de las principales actividades que realizan las ligas para proporcionar información a los clubes y jugadores que pertenecen a la liga.

Con este sistema web, se busca principalmente solucionar los problemas de acceso a la información de cada liga, ya que muchas veces no todos los involucrados tienen acceso a datos de resultados y estadísticas de cada una de las fechas que se juega. El sistema se enfocará en los jugadores, ya que son el grupo mayoritario dentro de las ligas barriales, y el acceso a dicha información se convierte en un proceso crucial para ellos.

2.2 Desarrollo web

Cuando hablamos de desarrollo web hablamos de un “diseño, construcción y creación de un sitio web, donde los aspectos más importantes comprenden el diseño, programación y gestión de base de datos dentro del sitio web creado” (OpenClassrooms, 2017). Es un trabajo que se lleva a cabo en segundo plano y se encarga de garantizar que un sitio web tenga una apariencia

profesional, funcionalidad rápida y un rendimiento excelente para brindar la mejor experiencia de usuario posible.

Involucra una variedad de tareas, como el diseño y la creación de páginas web, gestión de bases de datos e implementación de medidas de seguridad. El objetivo del desarrollo web es crear sitios web y aplicaciones web que sean funcionales, eficientes, atractivos visualmente y que brinden una experiencia de usuario satisfactoria.

2.2.1 Desarrollo Frontend

Frontend se refiere a “la sección de un sitio web que los usuarios utilizan de manera directa para interactuar, lo que también se conoce como la interfaz de usuario. Su principal función es permitir que los usuarios accedan y se comuniquen con el sitio web de manera efectiva y eficiente” (Souza, 2020), por lo que es necesario que cumpla con los estándares de usabilidad y diseño para garantizar una buena experiencia de usuario.

Un desarrollo frontend bien diseñado y desarrollado puede mejorar significativamente la experiencia del usuario, lo que a su vez puede llevar a una mayor satisfacción y fidelidad del cliente. También aporta un diseño atractivo y funcional que es crucial para el éxito de cualquier aplicación o sitio web, ya que afecta directamente la percepción del usuario y su experiencia de uso.

2.2.2 Desarrollo Backend

Hace referencia a “una capa de entrada o código de programa al que los usuarios no tienen acceso. Es la parte del software que no entra en contacto directo con los usuarios” (Souza, 2020). Se trata de desarrollar la lógica y funcionalidad interna de un sistema web o aplicación que no es directamente visible para los usuarios finales. Esta parte del sistema se encuentra en el servidor

y se encarga de procesar y almacenar datos, gestionar la seguridad, la autenticación y otras características que aseguran el correcto funcionamiento del software.

Por lo general se utiliza para la gestión y organización de datos, y suele estar formado por uno o varios lenguajes de programación que se utilizan para guiar el desarrollo de estructuras web.

2.2.3 Framekork

Un framework se define como “entorno de trabajo predispuesto, que posee ciertas herramientas y características útiles que agilizan el desarrollo de un proyecto de programación” (Cristancho, 2022). En esencia, el framework simplifica el trabajo del programador al proporcionar una estructura y herramientas predefinidas para trabajar. Al usar una base de datos existente, es posible reducir el tiempo requerido durante el proceso de desarrollo.

Utilizar un framework permite que el programador se centre en la lógica y las funciones del software en desarrollo, sin tener que preocuparse por detalles técnicos como la gestión de la base de datos, la seguridad, la autenticación y la estructura del proyecto. Dado que el framework ofrece soluciones predeterminadas para estos problemas, el programador puede ahorrar tiempo y esfuerzo al evitar tareas repetitivas y aburridas.

Como resultado, el uso de un framework puede aumentar la productividad y eficiencia del proceso de desarrollo al proporcionar una base sólida para construir sobre ella y acelerar la creación de nuevas funcionalidades y características.

2.2.4 React

Según (Deyimar, 2023) "ReactJS es una biblioteca de JavaScript popular para el desarrollo de aplicaciones web y móviles. Creada por Facebook, React incluye una serie de componentes de código reutilizables en JavaScript que se utilizan para construir la interfaz de usuario (UI)". Hoy en día, React se encuentra entre los frameworks más populares y respaldados por los desarrolladores, lo que implica que existe una abundante cantidad de documentación accesible para facilitar el desarrollo del proyecto.

React es una biblioteca de JavaScript que se especializa en la construcción de interfaces de usuario que sean escalables y eficientes a través del uso de componentes reutilizables. Los componentes de React se actualizan de forma automática cuando el estado de la aplicación cambia, lo que permite una programación más eficiente y una mayor velocidad de desarrollo. Debido a su eficacia y facilidad de uso, así como a su activa comunidad de apoyo, hoy en día, React se ha posicionado como una de las herramientas más demandadas para el desarrollo de software.

2.2.5 Firebase

Según (Seidor nts, 2022) Firebase es “una herramienta ágil y sencilla que nos facilita la creación de aplicaciones móviles o web agilizando los tiempos de desarrollo, pero sin perder de vista la calidad necesaria”. Debido a que React es uno de los marcos más populares y apoyados por los desarrolladores en la actualidad, hay una gran cantidad de documentación que ayudará con el progreso del proyecto.

Firebase ofrece varias ventajas, algunas de las cuales son:

- Fácil integración: Firebase es fácil de integrar con otras herramientas y plataformas de desarrollo, lo que lo hace ideal para proyectos de desarrollo de aplicaciones.
- Hosting web: Firebase ofrece alojamiento web gratuito y confiable para aplicaciones web, lo que permite a los desarrolladores implementar y lanzar aplicaciones rápidamente.
- Base de datos en tiempo real: Firebase proporciona una base de datos en la nube que opera en tiempo real, lo que permite a los desarrolladores crear aplicaciones escalables y dinámicas capaces de manejar grandes volúmenes de datos en tiempo real.
- Autenticación y seguridad: Firebase ofrece herramientas de autenticación y seguridad para proteger las aplicaciones y los datos de los usuarios. Esto abarca la posibilidad de autenticar a los usuarios a través de proveedores externos como Google, Facebook y Twitter, así como la opción de implementar autenticación personalizada y control de acceso.
- Funciones en la nube: Firebase proporciona funciones en la nube que permiten a los desarrolladores ejecutar código en la nube en respuesta a eventos del usuario o del sistema, lo que brinda a los desarrolladores la capacidad de crear aplicaciones que pueden crecer de manera efectiva y adaptarse a las necesidades específicas de cada proyecto.
- Comunidad activa: Firebase tiene una comunidad activa de desarrolladores que trabajan juntos para crear bibliotecas y herramientas de código abierto, lo que anima a los desarrolladores a colaborar y compartir información. Esto mejora la colaboración y el aprendizaje durante el desarrollo de proyectos con Firebase.

2.3 Metodologías de desarrollo

Las metodologías de desarrollo de software son técnicas y herramientas que se utilizan para el desarrollo, implementación y mantenimiento de software. En otras palabras, aplicaciones que mejoran la interacción de los usuarios con un dispositivo informático. (Solera, 2022)

Los métodos de desarrollo pueden ser ágiles o tradicionales, según su enfoque y proceso de trabajo. Los métodos tradicionales como Cascada se centran en la planificación detallada antes de la implementación, mientras que los métodos ágiles como Scrum se centran en la iteración rápida y la colaboración continua con las partes interesadas.

Algunos métodos de desarrollo comunes son Scrum, Kanban, Lean, Waterfall, Spiral, XP (Extreme Programming) y RUP (Rational Unified Process). Cada método tiene sus ventajas y desventajas, y es importante elegir la metodología adecuada para cada proyecto y equipo de desarrollo.

2.3.1 Metodología XP

XP es una metodología de desarrollo ágil que se enfoca en la calidad del software, la entrega temprana y frecuente de resultados y la mejora continua del proceso de desarrollo. Los proyectos de desarrollo web que requieren software funcional producido de manera rápida y de alta calidad requieren esta metodología. XP también fomenta la colaboración y la comunicación en el equipo mediante el uso de prácticas como la programación en pareja y el diseño de juegos para aumentar la productividad.

2.3.2 Ventajas de XP

XP se centra en la entrega rápida y temprana de un producto de software. Algunas de sus ventajas son:

- Flexibilidad: XP permite que los requisitos y el alcance del proyecto cambien a medida que avanza, lo que posibilita que el equipo de desarrollo se ajuste a las modificaciones en el mercado y a las necesidades cambiantes del cliente.
- Entrega temprana y regular: XP se enfoca en la entrega temprana y frecuente de software en funcionamiento, lo que permite que el cliente vea y use el producto temprano y brinde comentarios valiosos.
- Calidad: XP enfatiza la calidad desde el inicio del proyecto, utilizando pruebas e integración continuas para garantizar un producto final de calidad.
- Comunicación: XP promueve una comunicación regular y abierta entre el equipo de desarrollo y el cliente, lo cual asegura una comprensión clara de los requisitos y el avance del proyecto por parte de todos los involucrados.
- Mejora continua: Mediante la retroalimentación y la revisión de prácticas, XP promueve la mejora continua en el proceso de desarrollo. Además, busca aumentar la productividad y la calidad del trabajo en equipo mediante la implementación de prácticas como el diseño de juegos y la programación en pareja.

2.4 Lenguaje Unificado de Modelado

El Lenguaje de Modelado Unificado (UML) se define como “un estándar que permite la representación gráfica de elementos, acciones y estados en un sistema, este lenguaje puede utilizarse como modelo para un proyecto, con el objetivo de lograr una organización estructurada de la información” (Ionos, 2018), por lo que el uso de UML facilita a los desarrolladores la presentación de una descripción del sistema que resulte clara y comprensible para aquellos que no tienen experiencia en el campo. "

2.5 Casos de uso

Estos diagramas son “un tipo de diagrama de comportamiento UML que se utiliza comúnmente en el análisis de sistemas. Este tipo de diagrama permite representar gráficamente los distintos roles que existen en un sistema y cómo estos roles interactúan con el mismo” (Creately, 2022).

En un diagrama de casos de uso, los actores se representan como figuras ubicadas fuera del sistema, mientras que los casos de uso se muestran como elipses que se encuentran dentro del sistema. Las flechas se utilizan para ilustrar las interacciones entre los actores y los casos de uso.

El propósito principal del diagrama de casos de uso es brindar una visión general del sistema desde la perspectiva del usuario, mostrando las funcionalidades principales que el sistema ofrece y cómo interactúa con los actores externos, lo que permite identificar y definir los requisitos del sistema de manera más precisa y eficiente. En la fase de análisis y diseño de sistemas de software, el diagrama de casos de uso se utiliza comúnmente para comprender los requisitos del usuario y las funcionalidades necesarias.

2.6 Gestión deportiva

Según (Rodríguez, 2014) la gestión deportiva “está relacionada con las actividades que realiza un administrador, es decir, acciones encaminadas a planear, organizar, dirigir, ejecutar y controlar todo lo relacionado con el deporte”, por lo que se trata de un conjunto de actividades y procesos administrativos necesarios para organizar, coordinar y controlar las diferentes disciplinas deportivas y sus competiciones.

La gestión deportiva abarca la planificación, ejecución y evaluación de actividades y eventos deportivos, así como la gestión de recursos humanos, financieros y materiales. También implica

tomar decisiones estratégicas para asegurar el éxito de las competiciones y la satisfacción de los deportistas y el público en general.

2.7 Ligas Barriales

Son organizaciones deportivas de carácter comunitario que se encargan de promover la práctica del deporte en los barrios y comunidades, a través de la organización de competiciones y eventos deportivos. Estas ligas tienen como objetivo fomentar la participación ciudadana, la integración social y el desarrollo físico y mental de los participantes.

2.8 Diseño y desarrollo de prototipos

Es un proceso iterativo que permite crear un modelo inicial de un producto o sistema, cuya finalidad es evaluar su viabilidad y funcionalidad. Este proceso implica la definición de requerimientos, la elaboración de un diseño preliminar, la implementación y evaluación del prototipo.

CAPÍTULO III: METODOLOGÍA

3. Metodología de desarrollo del plan de tesis

3.1 Investigación Cualitativa

Este estudio se llevará a cabo utilizando una metodología de investigación aplicada, ya que tiene como objetivo poner en práctica los conocimientos adquiridos para resolver un problema específico en la sociedad ecuatoriana.

Se ha elegido la metodología cualitativa, ya que, al trabajar con una población relacionada con las ligas barriales, se espera obtener información sobre las experiencias y percepciones de los participantes asociados a la investigación.

Este tipo de investigación se centra en examinar, describir e interpretar fenómenos sociales complejos mediante la recopilación y análisis de datos no numéricos. Se busca explorar y comprender en profundidad estos fenómenos a través de métodos cualitativos y técnicas como entrevistas, observación participante y análisis de contenido. Por lo que ayuda a analizar los datos recolectados utilizando métodos de análisis temático, análisis narrativo, análisis de discurso y otros enfoques de análisis cualitativo para identificar patrones y temas emergentes.

La metodología cualitativa seleccionada para realizar la investigación es la metodología tradicional, también conocida como investigación de acción participativa, que consiste en colaborar con la comunidad para resolver un problema de interés común. En este caso, se trata de mejorar y facilitar las principales actividades y tareas necesarias para el funcionamiento de las ligas barriales.

De esta manera, se obtendrá información relevante y una mejor comprensión sobre el grupo objetivo de la investigación.

3.2 Investigación Aplicativa

“La investigación aplicada tiene la meta de generar conocimiento con una aplicación al momento y en medio plazo” (Lozada, 2014). Este tipo de investigación se centra en la generación de soluciones prácticas y efectivas para problemas específicos y en la mejora de la vida de las personas y las comunidades.

Su objetivo es obtener un enfoque metodológico que busca aplicar los hallazgos y conocimientos adquiridos en la investigación científica para resolver problemas prácticos en el mundo real.

Al utilizar métodos y procesos de ingeniería se vuelve fundamental utilizar metodologías desarrollo, principalmente porque estas ayudan a guiar el desarrollo y asegurara la calidad del producto final en cuanto a aspectos de seguridad, funcionalidad y consistencia. Actualmente cuando se busca desarrollar un proyecto de software la decisión se encuentra entre escoger una metodología ágil o tradicional, para este proyecto utilizaremos una metodología ágil.

Al finalizar con la investigación aplicada se cuenta con un prototipo o herramienta práctica y útil para la gestión de ligas barriales, lo que puede tener un impacto positivo en la organización y administración de actividades deportivas y sociales en las comunidades. Esto requeriría la aplicación de los principios y metodologías de la investigación científica con el fin de diseñar, desarrollar y probar un sistema web que permita una gestión eficiente y efectiva de las ligas barriales.

3.3 Metodología de desarrollo de software

3.3.1 Modelo que aplica

3.3.1.1 Metodología ágil

En los últimos años, este tipo de metodología ha ganado popularidad porque permite adaptar la forma de trabajo y desarrollo a las condiciones del proyecto, donde su principal objetivo es aumentar la productividad gestionando los proyectos de manera flexible y eficaz. Entre las principales ventajas encontramos:

- Mejora la calidad del producto.
- La participación del cliente en el proceso de desarrollo aumenta la satisfacción del cliente.
- Favorece la mejora en la organización del trabajo en equipo.
- Mayor control de tiempo y costos.

3.3.1.2 Extreme Programming:

XP fomenta la colaboración entre el equipo de desarrollo y el cliente, priorizando la comunicación verbal sobre la escrita. Se enfoca en diseñar y programar las partes del sistema de manera incremental, sin preocuparse demasiado por el futuro, lo que divide el trabajo en iteraciones. Estas iteraciones siguen cinco pasos repetitivos: exploración, planificación, diseño, codificación y prueba, hasta que el proyecto esté completo.

- Exploración: Consiste en realizar una evaluación inicial del proyecto para establecer los requisitos fundamentales del sistema.
- Planificación: Se planifican las tareas que se realizarán en cada iteración. Se establecen los objetivos y se define el alcance del trabajo a realizar.

- Codificación: En esta fase, se escribe el código del sistema y se realizan pruebas unitarias. Los desarrolladores trabajan en parejas, haciendo uso de la programación en pareja, y el código se integra regularmente en el repositorio de código.
- Pruebas: Durante la fase de pruebas en XP, se llevan a cabo pruebas de aceptación para evaluar si el sistema cumple con los requisitos establecidos previamente y se detectan los errores que puedan existir.

El uso de XP incentiva el trabajo colaborativo y promueve la entrega en ciclos cortos. Además, se adapta rápidamente a cualquier cambio que se necesite realizar dentro del sistema, así como automatiza el proceso de pruebas y entrega del producto de software.

CAPÍTULO IV: DESARROLLO DE LA INVESTIGACIÓN

4. Desarrollo

4.1 Selección Front-end

4.1.1. React

React es una biblioteca de JavaScript que permite la creación de componentes interactivos para las interfaces de usuario. Cuando se actualizan los datos en el modelo de objetos del documento (DOM), estos componentes pueden ser diseñados, encapsulados y combinados para crear interfaces complejas que se renderizan de manera eficiente. React es una biblioteca JavaScript fácil de aprender e implementar, con una comunidad activa que apoya el desarrollo y una gran cantidad de documentación para orientarse.

4.1.2. Angular

Angular es un framework que permite la creación de interfaces para aplicaciones web. Este marco de desarrollo, basado en componentes, permite la creación de aplicaciones web escalables. Angular usa un modelo de objeto de documento real capaz de detectar cambios en lugar de uno virtual. Además, Angular es un entorno de desarrollo completo que incluye todas las bibliotecas necesarias al instalar el marco, lo que significa que no necesita importar bibliotecas adicionales. Debido a su enfoque integral, aprender Angular puede ser más difícil porque necesita comprender todos los aspectos del marco para utilizarlo de manera efectiva.

4.1.3 Comparativa React-Angular

Tabla 1

Factores por considerar al elegir un framework de front-end

Criterio	React	Angular
Fácil de aprender	x	
Documentación	x	x
Comunidad	x	x
Experiencia con el framework	x	
Desarrollo de páginas y sistemas web	x	x
Suma/Resultados	5	3

Nota: React posee una mayor cantidad de criterios que Angular

4.2. Fase 1: Exploración

4.2.1 Alcance y Módulos del proyecto

- El prototipo estará compuesto por los módulos de registro de equipos, resultados de partidos, programación de próximos encuentros y registro de jugadores
- El prototipo no incluirá funcionalidades avanzadas como la integración de pagos electrónicos, creación de perfiles de usuario personalizados o visualización de estadísticas avanzadas.

4.2.2 Requerimientos Funcionales

- Registro de usuarios: El sistema debe ofrecer la posibilidad de registro a los usuarios, permitiéndoles crear una cuenta única para acceder al sistema.
- Gestión de ligas: El sistema debe permitir a los administradores de liga crear, modificar y eliminar equipos y jugadores de la liga.

- Calendario de partidos: el sistema debe proporcionar un calendario de partidos para cada liga, donde los usuarios puedan ver la información de los próximos partidos, junto con los resultados de los partidos previos.
- Registro de resultados: El sistema debe permitir a los administradores de la liga registrar los resultados de los partidos y realizar actualizaciones en las tablas de clasificación correspondientes a la liga.
- Control de acceso y permisos: El sistema debe permitir a los administradores de la liga controlar el acceso y los permisos de los usuarios, por ejemplo, restringir el acceso a ciertas funciones o datos según el rol o el nivel de autorización del usuario.
- Diseño responsive: El sistema debe ser diseñado de manera que sea compatible con diversos dispositivos y tamaños de pantalla, incluyendo computadoras de escritorio, tabletas y teléfonos móviles.

4.2.3 Requerimientos No Funcionales

- Usabilidad: el sistema debe ser fácil de entender y usar para todos los usuarios.
- Fiabilidad: el sistema debe estar disponible en cuando se lo requiera.
- Adaptabilidad: el sistema debe ser capaz de adaptarse a diferentes entornos de uso y usuarios, lo que permite la personalización y configuración de ciertos aspectos.

4.2.4 Riesgos Identificados

- Cambios en los requerimientos durante el proceso de desarrollo.
- Problemas con la integración de tecnologías y herramientas.

4.2.5 Procesos de Gestión de Riesgos

- Establecimiento de mecanismos de retroalimentación para minimizar cambios en los requisitos del proyecto.

- Selección cuidadosa de tecnologías y herramientas, y pruebas regulares de su integración.
- Análisis y recopilación correcta de requerimientos.

4.2.6 Herramientas a utilizar

- Lenguaje de programación: JavaScript.
- Framework de desarrollo web: React JS
- Base de datos: Firebase
- Herramienta de control de versiones: Git.

4.3 Fase de Planificación

Tabla 2

Roles de Usuario

Rol	Tipo	Características
Administrador	Todos los privilegios	Puede visualizar información. Permitir llevar a cabo operaciones CRUD en todas las tablas de la base de datos. Puede modificar las imágenes del repositorio.
Estándar	Sin privilegios	Únicamente puede visualizar datos e información

Nota: En la tabla se visualiza los roles que tendrán los usuarios dentro del sistema

4.3.1 Consolidación de Requerimientos

A continuación, se presentará las historias de usuario obtenidas para realizar la codificación de del sistema web:

- RF01: El sistema debe permitir a los usuarios registrarse y crear una cuenta de usuario única para acceder al sistema.
- RF02: Permitir a los administradores de liga crear, modificar y eliminar equipos de la liga.
- RF03: Permitir a los administradores de liga crear, modificar y eliminar jugadores de los equipos de la liga.
- RF04: Proporcionar un calendario de partidos para cada categoría, donde los usuarios puedan ver la información de los próximos partidos y los resultados de los partidos previos.
- RF05: El sistema debe permitir a los administradores de liga registrar los resultados de los partidos.
- RF06: Mostrar tabla de posiciones de por cada una de las categorías de la liga.
- RF07: Restringir el acceso a ciertas funciones o datos según el rol o el nivel de autorización del usuario.
- RF08: Diseño responsive que funcione en ambiente web y dispositivos móviles.

4.3.2 Plan de entregas

Tabla 3

Plan de entregas

Número de iteración	Prioridad	Duración	Número de historia

Primera Iteración	Alta	3 semanas	Del 1 al 4
Segunda Iteración	Media	1 semana	5-6
Tercera Iteración	Baja	1 semana	7-8

Nota: Tabla de tiempo para la entrega del sistema web.

4.3.3 Fase de Iteraciones

Como se ve en la tabla 2 se dividió en 3 iteraciones el desarrollo de aplicación. Cada iteración se ha estimado en términos de semanas y se encuentran ordenadas por prioridad en base a a las funcionalidades que son más importantes para el funcionamiento de la aplicación. Para esta parte la primera iteración se realizarán las 3 primeras historias que son las que más relevancia tienen dentro del sistema.

4.3.4 Primera Iteración

4.3.4.1 Requerimientos Primera Iteración

- RF01: el sistema debe permitir a los usuarios registrarse y crear una cuenta de usuario única para acceder al sistema.
- RF02: permitir a los administradores de liga crear, modificar y eliminar equipos de la liga.
- RF03: permitir a los administradores de liga crear, modificar y eliminar jugadores de los equipos de la liga.
- RF04: proporcionar un calendario de partidos para cada categoría, donde los usuarios puedan ver la información de los próximos partidos y los resultados de los partidos previos.

4.3.4.2 Tareas primera iteración

Al iniciar con el desarrollo de la aplicación, se debe empezar creando el proyecto en React y crear un repositorio en Git para poder llevar un control de las versiones, de manera que se pueda empezar con la codificación de los requerimientos recopilados anteriormente.

Después de haber creado el proyecto, se deben implementar los componentes necesarios junto con las librerías requeridas para la codificación del proyecto. La primera parte de la aplicación está compuesta por las interfaces de visualización de información de cada una de las categorías por las que está compuesta la liga. Estos componentes se encontrarán junto con las rutas para poder navegar por la aplicación.

Es fundamental tener en cuenta que la función principal del sistema es mostrar los resultados y las posiciones de cada categoría. Como resultado, se establecerá una estructura para definir las relaciones entre los componentes y las funciones particulares que cada uno debe desempeñar.

Aparte de las vistas de visualización, se debe crear un login para que los usuarios con acceso de administrador puedan realizar acciones CRUD dentro del sistema. Por ello, se creará un repositorio y tablas dentro de Firebase para poder alojar la información de los jugadores y equipos. Para esto, debemos definir los roles que pueden tener los usuarios al acceder a la aplicación.

4.3.5 Segunda Iteración

4.3.5.1 Requerimientos Segunda Iteración

- RF05: el sistema debe permitir a los administradores de liga registrar los resultados de los partidos.
- RF06: Mostrar tabla de posiciones de por cada una de las categorías de la liga.

4.3.5.2 Tareas Segunda Iteración

Para cumplir con la funcionalidad de los requerimientos establecidos, se crean los componentes donde los administradores podrán registrar los resultados de los partidos una vez culminados. Cada componente incluirá una sección con operaciones CRUD (Crear, Leer, Actualizar y Eliminar) para cada jornada disputada durante la semana, la cual será de uso exclusivo para los administradores. Siendo para los usuarios estándar únicamente a la opción de visualizar los resultados de los partidos.

Mismamente se desarrollarán varias vistas para mostrar las tablas de posiciones de cada categoría existente en la liga, las cuales podrán ser modificadas por los administradores con opciones de agregar equipos y editar los resultados de los partidos en función de cómo se desarrollen los partidos disputados.

Esto se logra mediante el uso de protocolos de seguridad y autenticación para garantizar que solo los usuarios autorizados puedan acceder a las funciones administrativas. Esto se logra mediante el uso de niveles de acceso y autenticación, además del uso de contraseñas seguras.

4.3.6 Tercera Iteración

4.3.6.1 Requerimientos tercera iteración

- RF07: Restringir el acceso a ciertas funciones o datos según el rol o el nivel de autorización del usuario.
- RF08: Diseño responsive que funcione en ambiente web y dispositivos móviles.

4.3.6.2 Tareas Tercera Iteración

Para la última iteración como se mencionó anteriormente se realizará el control de acceso al sistema mediante un Inicio de sesión, el cual restringirá el acceso a ciertas funcionalidades del sistema de forma que se diferencie claramente los usuarios regulares que cuentan con acceso

únicamente de visualización de información y los usuarios administradores que cuentan con más funcionalidades, principalmente de editar, añadir y eliminar información.

4.4 Fase de Diseño

4.4.1 Arquitectura 3 capas

La arquitectura del sistema web adopta un patrón de tres capas que incluye una capa de presentación, una capa de aplicación y una capa de datos.

En la capa de presentación, se encuentran los componentes que permiten la interacción del usuario con el sistema, mostrando y modificando los datos. Su función principal es presentar información y facilitar la comunicación con el usuario.

La capa de aplicación es responsable de procesar la información proporcionada por la capa de presentación y llevar a cabo las operaciones necesarias para cumplir con los requisitos del sistema. Utilizando componentes de React JS, interactúa con la API de Firebase para realizar operaciones CRUD en las tablas de la base de datos del sistema.

La capa de datos se encarga de almacenar y traer los datos hacia el sistema. Se utiliza Firebase, una plataforma en la nube que proporciona bases de datos en tiempo real, autenticación de usuarios, almacenamiento de archivos y otros servicios que se pueden usar para crear aplicaciones web y móviles. Firebase se puede integrar fácilmente con React JS, lo que permite un desarrollo de sistemas rápido y eficiente.

4.4.2 Casos de Uso

En la ilustración 1, se observa una representación macro del funcionamiento del sistema desde la perspectiva del usuario. Dentro del sistema el usuario puede acceder al sistema donde se encontrarán las diferentes categorías que tiene la liga, donde en cada categoría se puede observar la información relacionada a los resultados de los partidos, las jornadas disputadas y por disputarse, junto con la tabla de posiciones de cada una de las categorías de la liga.

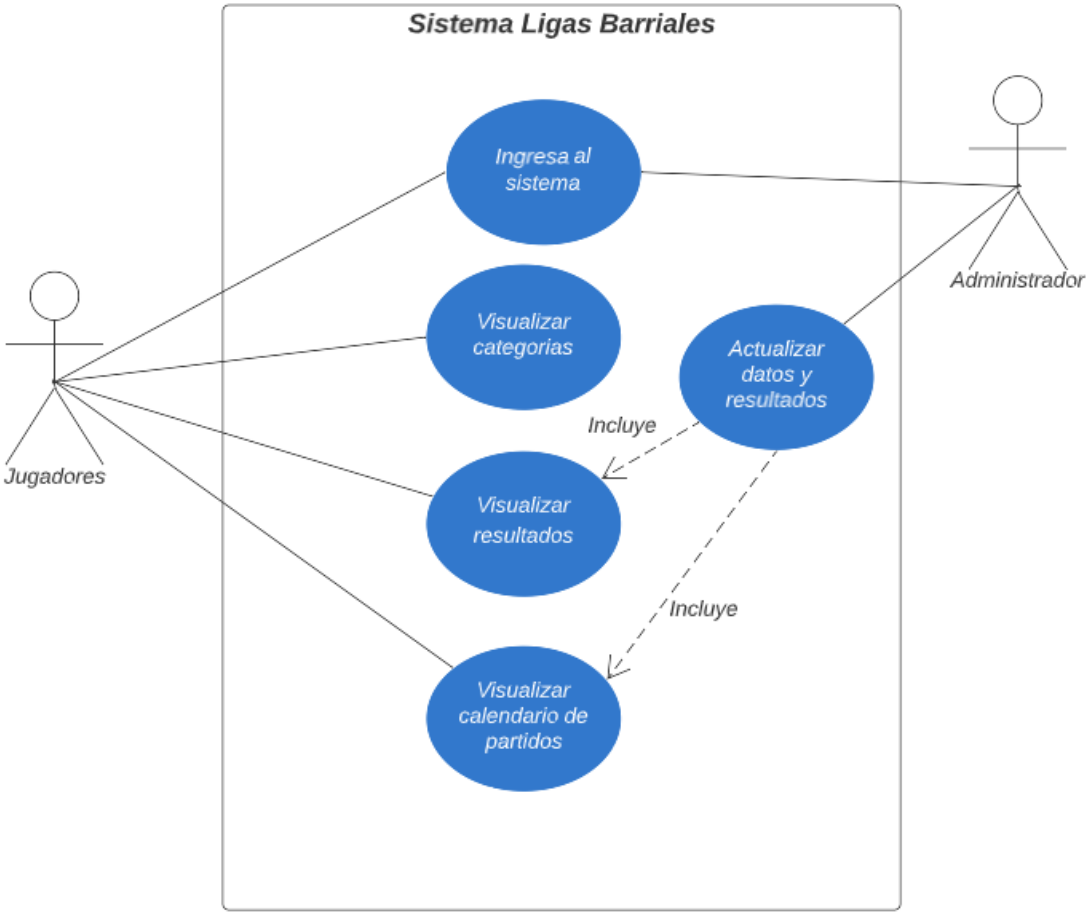


Ilustración 1 – Caso de Uso Actividad de Usuario

En la ilustración 2, se muestra el uso del sistema desde la perspectiva de los administradores o encargados de la liga, quienes tienen acceso a funcionalidades adicionales en comparación con los usuarios estándar. Entre estas funcionalidades se destaca la capacidad de gestionar la información que se muestra en las pantallas de visualización. Los administradores podrán visualizar y gestionar la información de cada categoría, administrar la información de cada uno de los equipos, gestionar información de los jugadores, así mismo como administrarlas jornadas y tabla de posiciones de cada una de las categorías.

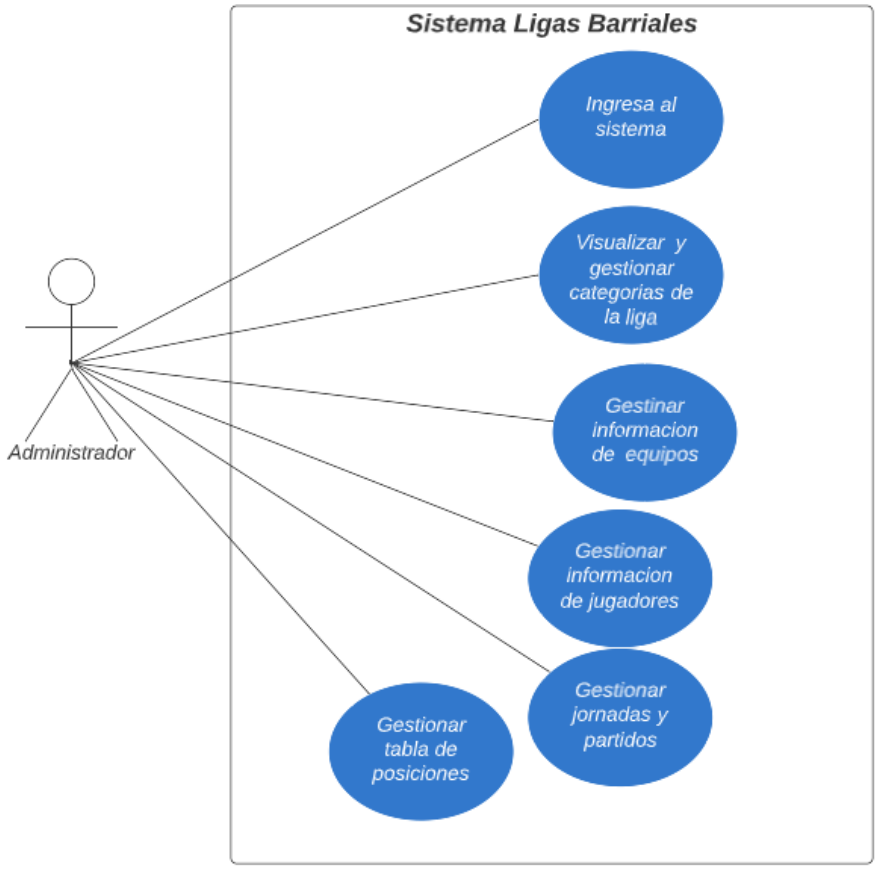


Ilustración 2 – Caso de uso Actividad administradores

En el diagrama de actividades de la ilustración 3 podemos ver el funcionamiento del sistema desde la perspectiva del usuario, donde el usuario puede acceder al foro y escoger las categorías que existen en la liga. Dentro de cada categoría se encuentra los equipos, jornadas, resultados y tabla de posiciones de cada categoría.

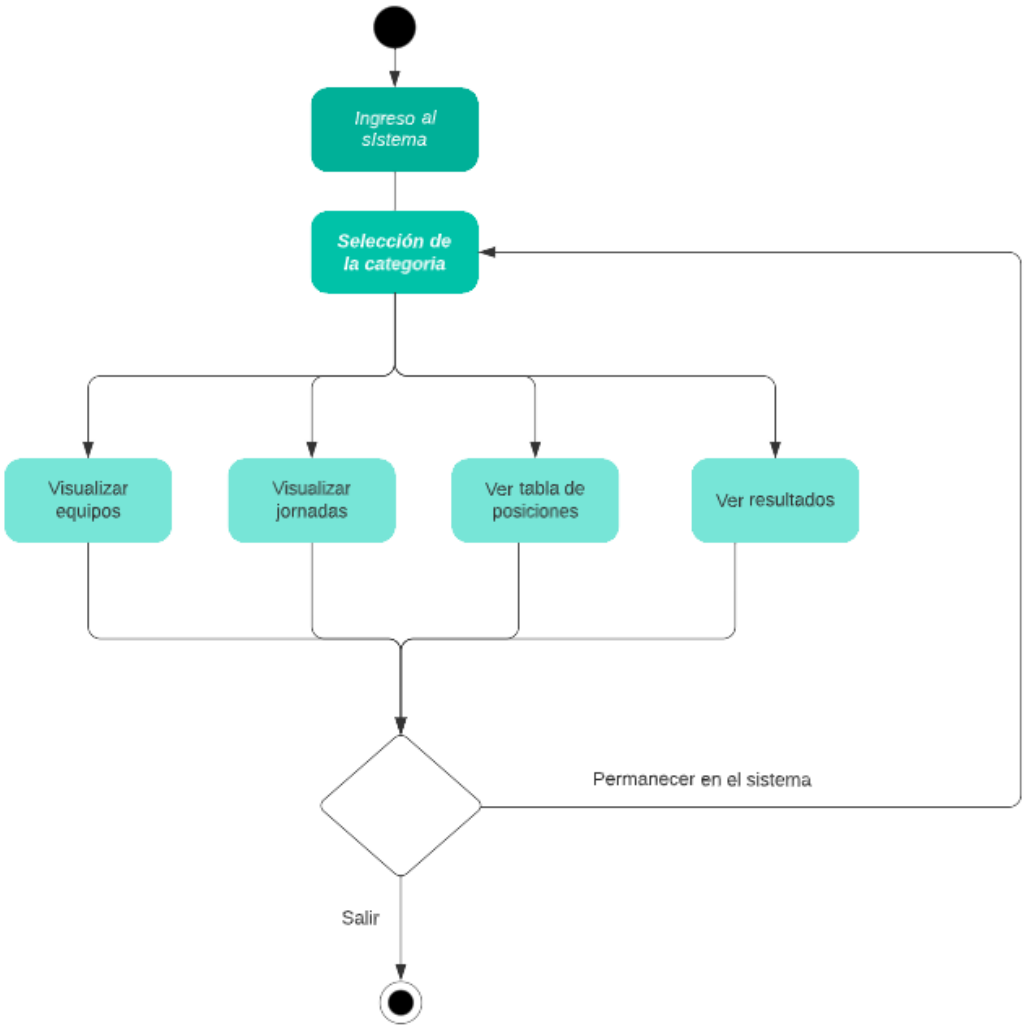


Ilustración 3 – Diagrama de actividad Usuario

4.4.3 Diseño de Interfaces

El diseño parte con 3 apartados que conforman las categorías de la liga barrial, mismos que son botones para acceder a la información que se encuentra dentro de cada categoría.

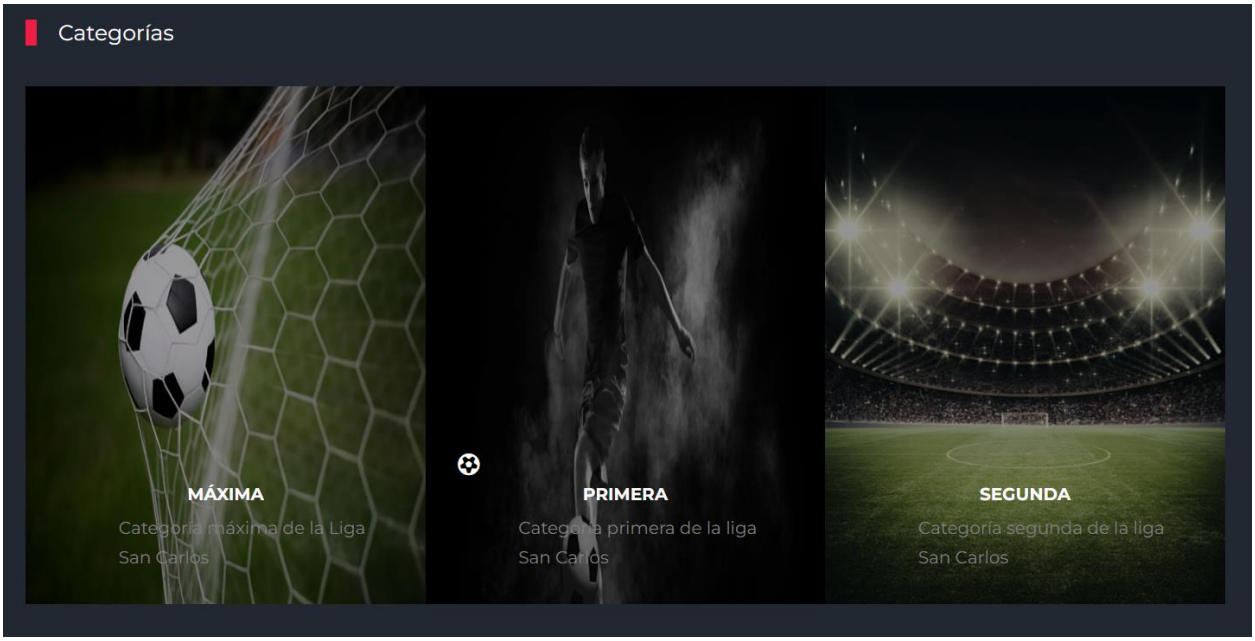


Ilustración 4 – Diseño Usuarios

Dentro de cada categoría tenemos un apartado principal donde se muestran los equipos que se encuentran registrados en la categoría con los campos de nombre, categoría, fecha de fundación y abreviatura, en la parte superior tenemos un menú para movernos por los diferentes apartados que hay en cada categoría.

Ilustración 5 – Diseño Equipos categoría

El diseño para la tabla de posiciones de cada categoría consta de una tabla en la que se puede visualizar cada uno de los equipos junto con la categoría y cuenta con los campos de nombre del equipo, escudo, partidos jugados, partidos perdidos, partidos ganados, partidos empatados, goles a favor, goles en contra, puntos y gol diferencia, donde el criterio de ordenamiento de la tabla son los puntos de clasificación de cada equipo.

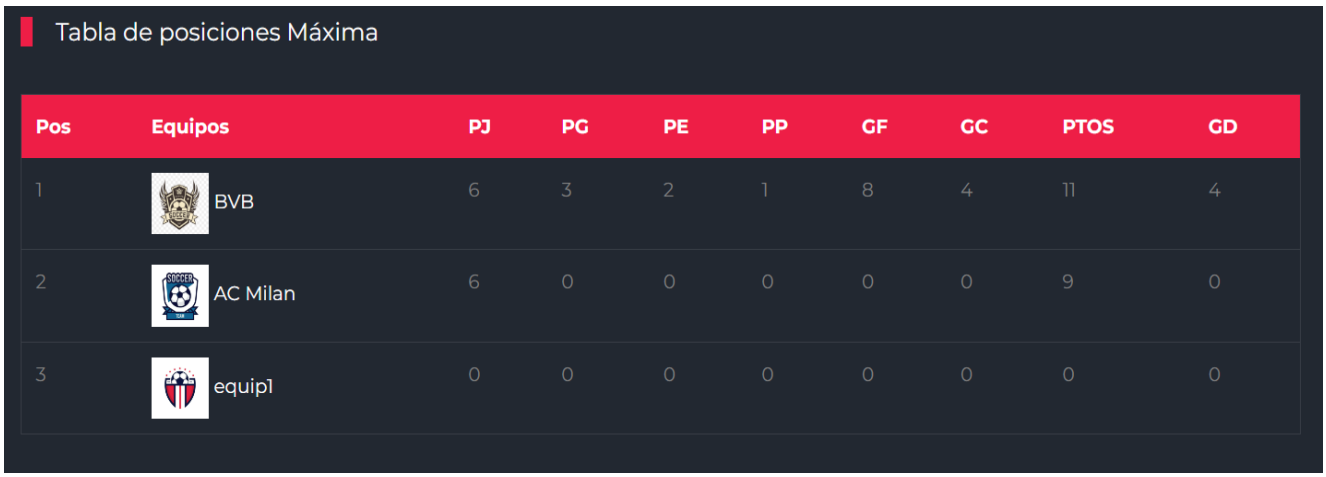


Ilustración 6 – Tabla de posiciones

Para el diseño de las jornadas por disputar se cuenta con un recuadro en el que se puede visualizar la información del partido a jugarse, los campos que se muestran son el escudo de los equipos junto con su nombre, el lugar o cancha en el que se jugará, la fecha del partido, la hora y el número de jornada correspondiente al partido.

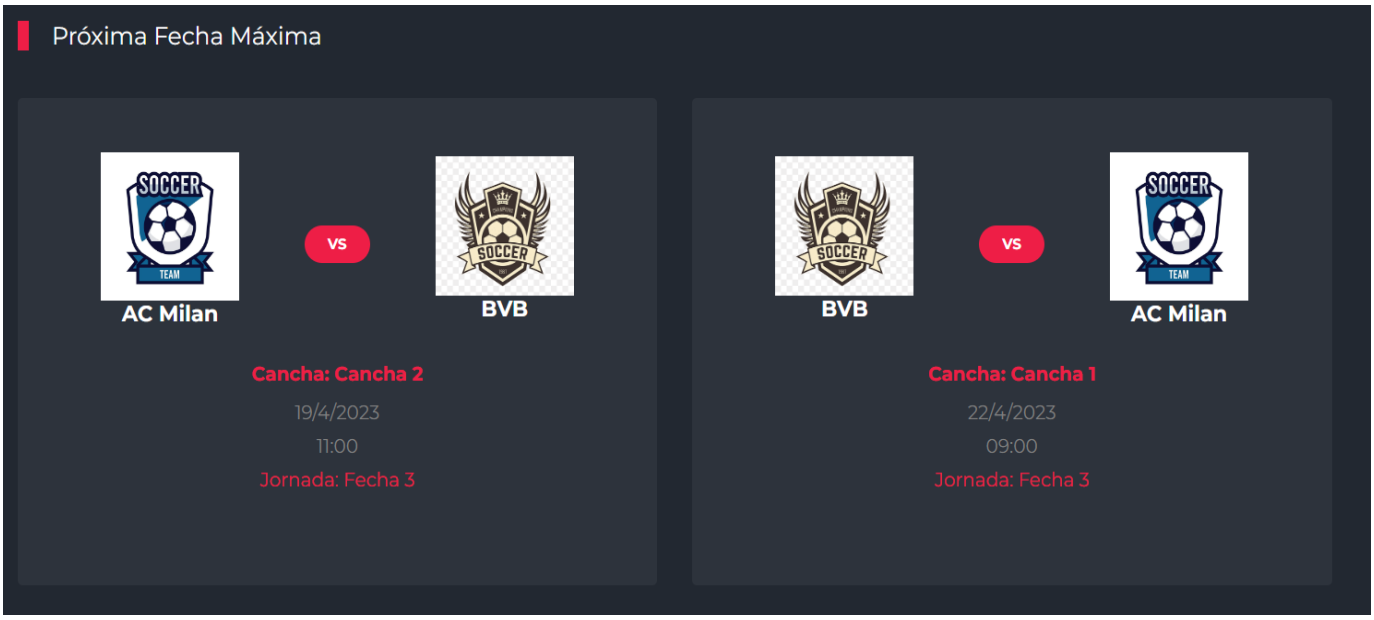


Ilustración 7 – Jornadas

En la vista de administrador contamos con tablas en las que se muestra la misma información que en la vista de usuario, pero en este caso se encuentra en una tabla, la principal diferencia es que en estos apartados contamos con acciones en el cual podemos crear, editar o eliminar registros de las tablas de la base de datos.

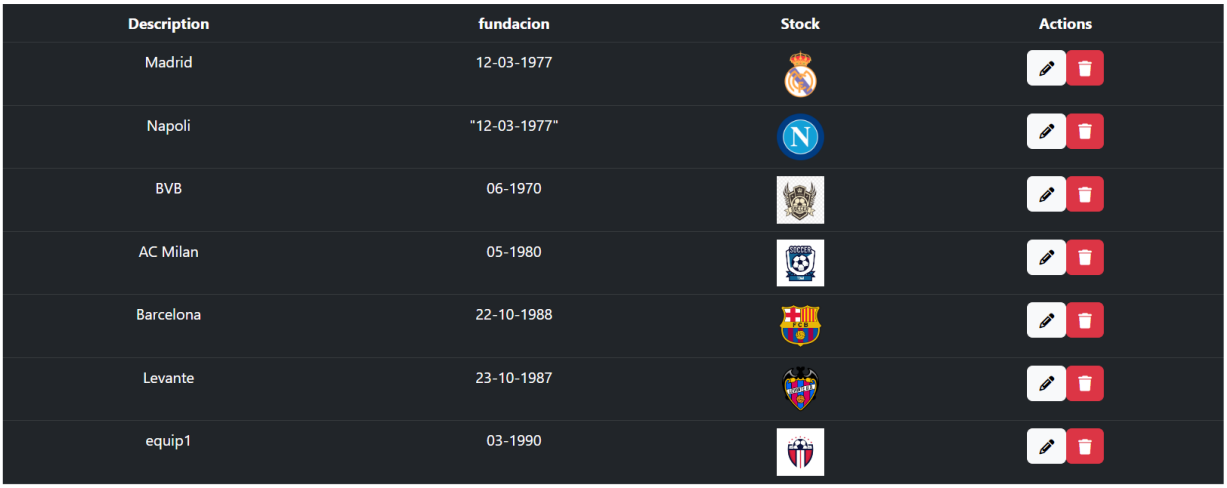


Ilustración 9 – CRUD Administrador

4.5 Fase de codificación

Para llevar a cabo la codificación del sistema, comenzamos creando el proyecto utilizando React JS y descargando las librerías necesarias para el desarrollo, en este caso Firebase para el almacenamiento de datos y react-router-dom para la gestión de las rutas de las diferentes vistas que tendrá el sistema. En este caso, las principales vistas definidas son:

- Usuario:
- Vista Principal
- Categoría Máxima
- Categoría Primera
- Categoría Segunda
- Tabla de Posiciones
- Jornadas
- Equipos

Administrador

- Vista Principal
- CRUD Equipos
- CRUD Jugadores
- Categorías
- Tabla de posiciones
- CRUD Jornadas por categoría
- Resultados

4.5.1 Estándares de Codificación

- Identificadores: El identificador de los capítulos se incrementa automáticamente.
- Variables: Las variables se denominan según su valor.

- Funciones: Las funciones describen su operación o propósito.
- Registro: La captura de datos se realiza utilizando el método correspondiente.

4.5.2 Código

En esta sección se desarrollan los componentes de React JS que establecerán la conexión entre las vistas y la base de datos, permitiendo la visualización y gestión de la información almacenada en cada una de las tablas de la base de datos.

Vista: Contiene el diseño y funciones que son usadas para cada vista.

Hooks: Se encargan de la conexión con Firebase y contienen los siguientes métodos:

• useAuth: maneja la autenticación de usuarios.

• useFirestore: interactúa con la base de datos.

• useStorage: maneja el almacenamiento de archivos.

• Create: permite agregar nuevos datos a Firebase.

• Update: permite actualizar la información que se encuentra en las tablas.

• Get: permite obtener o leer los datos de las tablas.

• Delete: permite eliminar registros de las tablas.

4.6 Pruebas

4.6.1 Pruebas de aceptación

Este tipo de pruebas se realizan para verificar si el sistema cumple con los requisitos que se establecieron previamente, de esta forma determinando si el sistema es apto para ser utilizado en un entorno real.

Tabla 3

Caso de prueba RF01

Pruebas de aceptación	
Código: RF01_P01	Requerimiento: R01
Nombre: Registro de usuario	
Descripción: El sistema debe tener un formulario donde puedan registrarse en la aplicación	
Condiciones: N/A	
Pasos: 1. Dar clic en el apartado de registro al que se ingresa desde el sistema de administrador 2. Completar la información requerida 3. Hacer clic en el botón de registro.	
Resultados previstos: El usuario se registrará exitosamente y será redirigido a la pantalla principal del sistema de administradores.	
Evaluación: Prueba completada satisfactoriamente	

Tabla 4

Caso RF01_P02

Pruebas de aceptación	
Código: RF01_P02	Requerimiento: R01
Nombre: Ingreso de usuario	
Descripción: Llenar los datos del formulario de inicio de sesión para poder ingresar al sistema de administradores	
Condiciones: Previamente haber creado una cuenta	
Pasos: 1. Llenar los datos solicitados 3. Hacer clic en el botón de inicio de sesión.	
Resultados previstos: El usuario administrador logrará iniciar sesión satisfactoriamente y será redirigido a la pantalla principal del sistema.	
Evaluación: Prueba completada satisfactoriamente	

Tabla 5

Caso de prueba RF05_P06

Pruebas de aceptación	
Código: RF05_P06	Requerimiento: RF05
Nombre: Ver jornadas y resultados	
Descripción: Verificar el funcionamiento de las funciones CRUD de la tabla jornadas.	
Condiciones: Previamente haber creado una cuenta	
Pasos: 1. Ingresar al aparatado de jornadas	

2. Escoger una categoría. 3. Verificar que las jornadas se muestren (read) 4. Verificar que los resultados se muestren
Resultados esperados: El usuario administrador logra realizar las operaciones visualizar las jornadas y resultados de los partidos.
Evaluación: Prueba completada satisfactoriamente

Tabla 6

Caso de prueba RF06_P07

Prueba de aceptación	
Código: RF06_P07	Requerimiento: RF06
Nombre: Ver tabla de posiciones	
Descripción: Visualizar la tabla de posiciones de cada una de las categorías de la liga	
Condiciones: N/A	
Pasos: 1. Ingresar al sistema 2. Escoger una categoría. 3. Verificar que se visualice la tabla de posiciones de cada categoria	
Resultados esperados: El usuario pudo visualizar la tabla de posiciones.	
Evaluación de la prueba: Prueba completada satisfactoriamente	

4.5.1 Pruebas de caja blanca

Este tipo de pruebas buscan examinar y evaluar la estructura del código del programa, se basan en el conocimiento detallado de la lógica interna y el diseño del sistema

Tabla 7

Caso de prueba RF02_P03

Pruebas de caja blanca		
Código: RF02_P03	Requerimiento: R02	
Nombre: CRUD equipos de la liga		
Descripción: Verificar el funcionamiento de las funciones CRUD de la tabla equipos		
Condiciones: Previamente haber creado una cuenta		
Entradas	Resultado esperado	Resultado
Mostrar equipos de la liga	Los datos se ven reflejados en la pantalla	OK
Crear equipos	Los equipos se crean correctamente	OK
Editar equipos	Los equipos se editan correctamente	OK
Eliminar equipos	Los equipos se eliminan correctamente	OK

Tabla 8

Caso de prueba RF03_P04

Pruebas de caja blanca		
Código: RF03_P04	Requerimiento: RF03	
Nombre: CRUD JUGADORES de la liga		
Descripción: Verificar el funcionamiento de las funciones CRUD de la tabla jugadores		
Condiciones: Previamente haber creado una cuenta		
Entradas	Resultado esperado	Resultado
Mostrar jugadores de la liga	Los datos se ven reflejados en la pantalla	OK
Crear jugadores	Los registros se crean correctamente	OK
Editar jugadores	Los registros se editan correctamente	OK

Eliminar jugadores	Los registros se eliminan correctamente	OK

Tabla 9

Caso de prueba RF04_P05

Pruebas de caja blanca		
Código: RF04_P05	Requerimiento: RF04	
Nombre: CRUD jornadas de la liga		
Descripción: Verificar el funcionamiento de las funciones CRUD de la tabla jornadas		
Condiciones: Previamente haber creado una cuenta		
Entradas	Resultado esperado	Resultado
Mostrar jornadas de la liga	Los datos se ven reflejados en la pantalla	OK
Crear jornadas	Los registros se crean correctamente	OK
Editar jornadas	Los registros se editan correctamente	OK
Eliminar jornadas	Los registros se eliminan correctamente	OK

Tabla 10

Caso de prueba RF07_P08

Pruebas de caja blanca		
Código: RF07_P08	Requerimiento: RF07	
Nombre: Restringir acceso a usuarios no autorizados		
Descripción: Solo los usuarios administradores podrán acceder a los CRUD del sistema		
Condiciones: Previamente haber creado una cuenta		
Entradas	Resultado esperado	Resultado

Iniciar sesión utilizando una cuenta de administrador.	Se ingresa al sistema de gestión.	OK
No se tiene cuenta de administrador	El usuario no puede acceder al sistema de gestión	OK

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- La creación de un sistema web para gestionar los procesos, operaciones y campeonatos de las ligas barriales es fundamental para mejorar la organización y eficiencia de los campeonatos. A través del sistema en línea, es posible automatizar tareas, facilitar el registro y seguimiento de equipos y jugadores, y agilizar la comunicación entre diferentes operadores.
- La identificación de los principales procesos de gestión realizados por las ligas barriales es esencial para dimensionar el alcance de la aplicación a desarrollar. Al comprender a fondo estos procesos, se pueden determinar de manera más precisa las necesidades y requerimientos del sistema, garantizando así una solución efectiva y adaptada a las características y particularidades de las ligas barriales
- Al finalizar el prototipo podemos concluir que el sistema web para controlar y gestionar el proceso de inscripción y registro de jugadores en las ligas barriales brinda varios beneficios. Al automatizar este proceso, se agiliza y simplifica la gestión de la liga, permitiendo un manejo más eficiente de la información y así mismo reduciendo posibles errores.
- La fase de pruebas es una parte fundamental del sistema, ya que evalúa el prototipo mediante pruebas de caja blanca y de aceptación para garantizar la calidad y eficiencia del sistema. Estas pruebas ayudan a identificar posibles errores, fallos y puntos de mejora en el sistema, tanto en aspectos funcionales como en la experiencia del usuario.
- Tras el diseño del sistema podemos concluir que diseñar interfaces amigables, interactivas y fáciles de usar es un objetivo fundamental en el desarrollo del sistema, ya que tener

interfaces bien diseñadas mejora la experiencia del usuario y contribuye a la eficiencia y productividad al interactuar con el sistema.

Recomendaciones

Para obtener mejores resultados al levantar los requerimientos, es necesario realizar un análisis exhaustivo de los procesos de gestión de las ligas barriales, ya que es fundamental comprender los procesos y actividades de gestión que se llevan a cabo en ellas. De esta forma, se tendrá una visión más clara de lo que se espera del sistema y se podrá adaptarlo a sus necesidades reales.

• Además de ser necesarios y fundamentales para el desarrollo, los frameworks facilitan el trabajo de cualquier desarrollador, ya que contienen componentes ya establecidos y listos para ser utilizados en la aplicación cuando sea necesario. Las plataformas de desarrollo también ofrecen una versión gratuita que permite a los desarrolladores independientes desarrollar sus aplicaciones sin tener que pagar por el hosting, la base de datos u otros requisitos. Se recomienda el uso de React junto con Firebase, ya que React está basado en JavaScript y es fácil de aprender, mientras que Firebase incluye todo lo necesario para el backend de la aplicación.

• Es recomendable implementar y expandir el sistema web para cubrir todas las funcionalidades y procesos de gestión de las ligas barriales, aprovechando los beneficios de automatización, agilidad y reducción de errores. Se deben realizar pruebas exhaustivas, recopilar retroalimentación de los usuarios finales y realizar los ajustes necesarios para garantizar una solución adaptada y altamente satisfactoria.

• Para lograr diseñar interfaces amigables y fáciles de usar, es crucial crear un diseño visualmente atractivo y coherente, asegurándose de que los elementos estén organizados de forma lógica y que el lenguaje utilizado sea claro y comprensible.

Bibliografía

Burns, E. (10 de 09 de 2021). *ComputerWeekly.es*. Obtenido de ComputerWeekly.es: https://www.computerweekly.com/es/definicion/Aprendizaje-profundo-deep-learning#:~:text=El%20aprendizaje%20profundo%20(Deep%20Learning,obtienen%20ciertos%20tipos%20de%20conocimiento.

Creately. (29 de Septiembre de 2022). *creately*. Obtenido de https://creately.com/blog/es/diagramas/tutorial-diagrama-caso-de-uso/

Cristancho, F. (08 de 02 de 2022). *Talently*. Obtenido de Talently: https://talently.tech/blog/que-es-un-framework-en-programacion/

DataScientist. (19 de 04 de 2022). *DataScientist*. Obtenido de DataScientist: https://datascientest.com/es/deep-learning-definicion

Deyimar. (10 de 01 de 2023). *Hostinger*. Obtenido de https://www.hostinger.com.ar/tutoriales/que-es-react#:~:text=ReactJS%20es%20una%20de%20las,usuario%20(UI)%20llamados%20componentes.

Fernández, D. (11 de 06 de 2019). *RRHDigital*. Obtenido de RRHDigital: https://www.rrhhdigital.com/secciones/liderazgo/137086/Claves-para-conseguir-una-buena-gestion-de-empresa-y-de-plantilla#:~:text=Una%20buena%20t%C3%A9cnica%20de%20gesti%C3%B3n,mejorar%20nuestros%20productos%20o%20servicios.

Fernández, D. (11 de 06 de 2019). *RRHDIgital*. Obtenido de RRHDIgital: https://datascientest.com/es/deep-learning-definicion

Ionos. (26 de Octubre de 2018). *Ionos*. Obtenido de https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/uml-lenguaje-unificado-de-modelado-orientado-a-objetos/

Lozada, J. (2014). *Dialnet*. Obtenido de Dialnet: https://dialnet.unirioja.es/servlet/articulo?codigo=6163749

OpenClassrooms. (11 de 09 de 2017). *OpenClassrooms*. Obtenido de OpenClassrooms: https://blog.openclassrooms.com/es/2017/09/11/que-es-el-desarrollo-web/

Rodríguez, A. (24 de Febrero de 2014). *Utel blog*. Obtenido de https://utel.mx/blog/cursos-y-diplomados/que-es-la-gestion-deportiva/#:~:text=La%20gesti%C3%B3n%20deportiva%20est%C3%A1%20relacionada,lo%20relacionado%20con%20el%20deporte.

Seidor nts. (31 de Mayo de 2022). *Desarrollo Mobile*. Obtenido de https://www.nts-solutions.com/blog/firebase-que-es.html

Solera, S. (27 de abril de 2022). *occam*. Obtenido de https://www.occamagenciadigital.com/blog/las-mejores-metodologias-para-un-correcto-desarrollo-de-software

Souza, I. d. (02 de 03 de 2020). *Rockcontent*. Obtenido de Rockcontent: https://rockcontent.com/es/blog/front-end-y-back-end/

Xataka. (28 de 10 de 2016). *Xataka*. Obtenido de Xataka: https://www.xataka.com/robotica-e-ia/deep-learning-que-es-y-por-que-va-a-ser-una-tecnologia-clave-en-el-futuro-de-la-inteligencia-artificial

GLOSARIO DE TÉRMINOS

A

Administrador: Tipo de usuario con privilegios adicionales que puede de realizar tareas de gestión, como agregar equipos, programar partidos y generar informes.

B

Base de datos: Componente del sistema en el que se almacena la información relevante del sistema web.

C

Calendario: Un componente del sistema que muestra los eventos y partidos programados de la liga para facilitar la visualización.

Componente: Dentro de React un componente es la independización de las partes de una interfaz que permite reutilizarlas en diferentes pantallas.

E

Equipo: Un grupo de jugadores que compiten juntos en una liga barrial.

G

Gestión de ligas barriales: Proceso de administrar y organizar las actividades relacionadas con las ligas barriales.

I

Inscripción de equipos: El proceso de registro de los equipos en la liga barrial, incluyendo información sobre los jugadores que los conforman.

Interfaz: Parte del sistema donde el usuario puede interactuar con la aplicación.

P

Programación de partidos: Proceso en el que se establece las fechas, horarios y lugares en los que se llevarán a cabo los partidos entre los equipos de la liga.

Prototipo: Versión inicial o modelo preliminar del sistema web que se desarrollará, generalmente utilizado para probar y validar conceptos antes de la implementación final.

S

Sistema web: Una aplicación basada en la web que permite a los usuarios acceder y gestionar información a través de un navegador de internet.

T

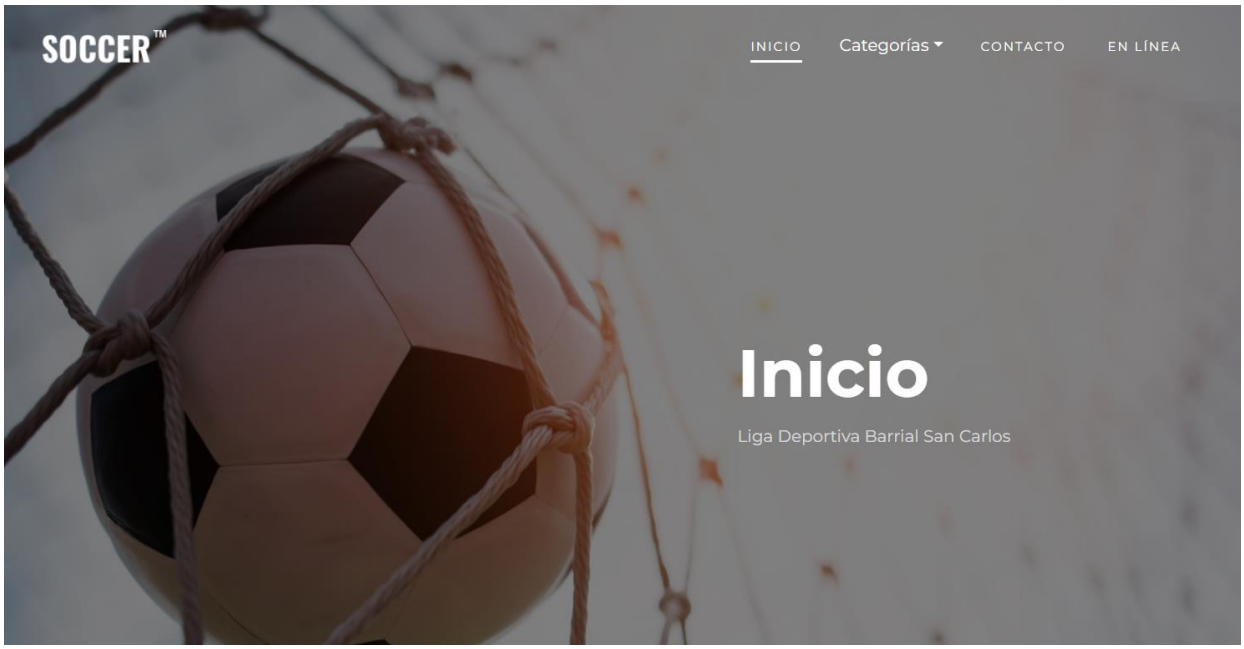
Tabla de posiciones: Tabla en la que se muestra la clasificación de los equipos dependiendo de su rendimiento en los partidos, esta tabla se ordena en base al número de puntos obtenidos.

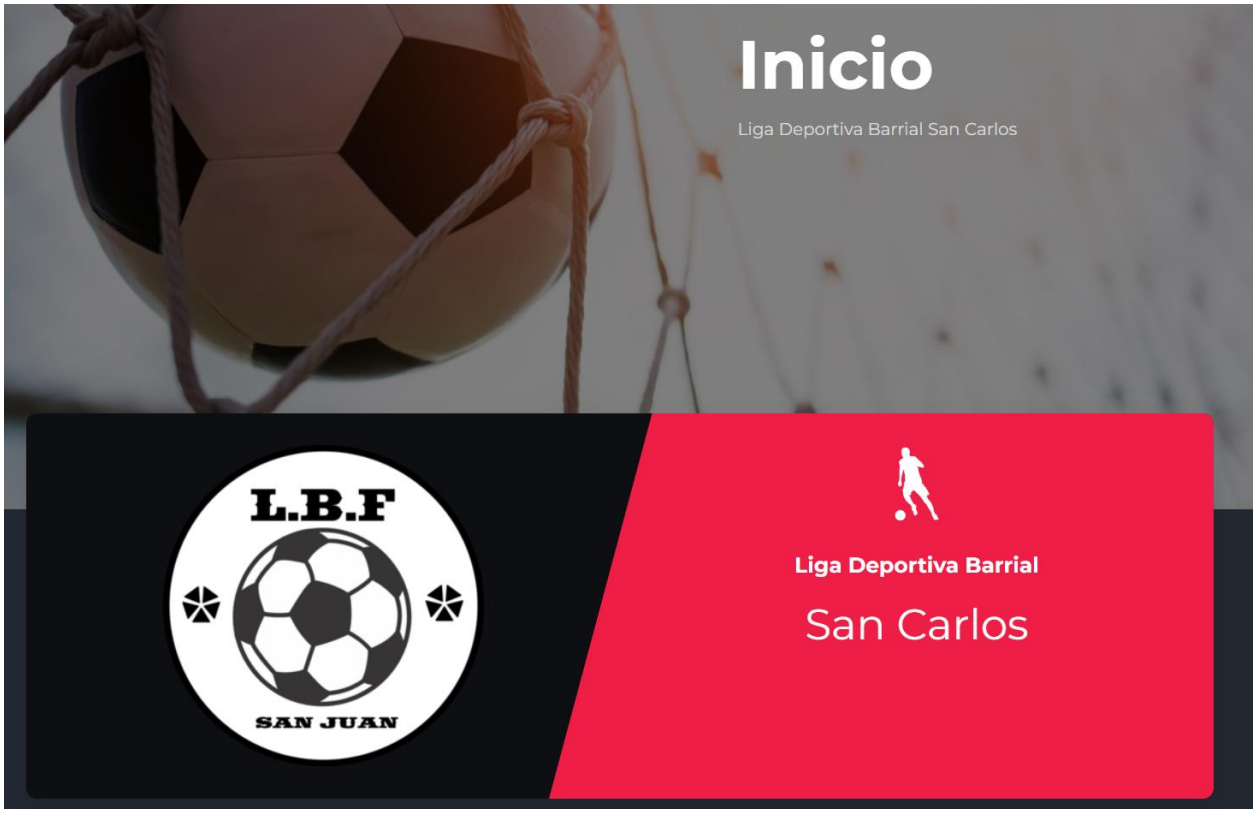
U

Usuario: persona que utiliza e interactúa con el sistema web.

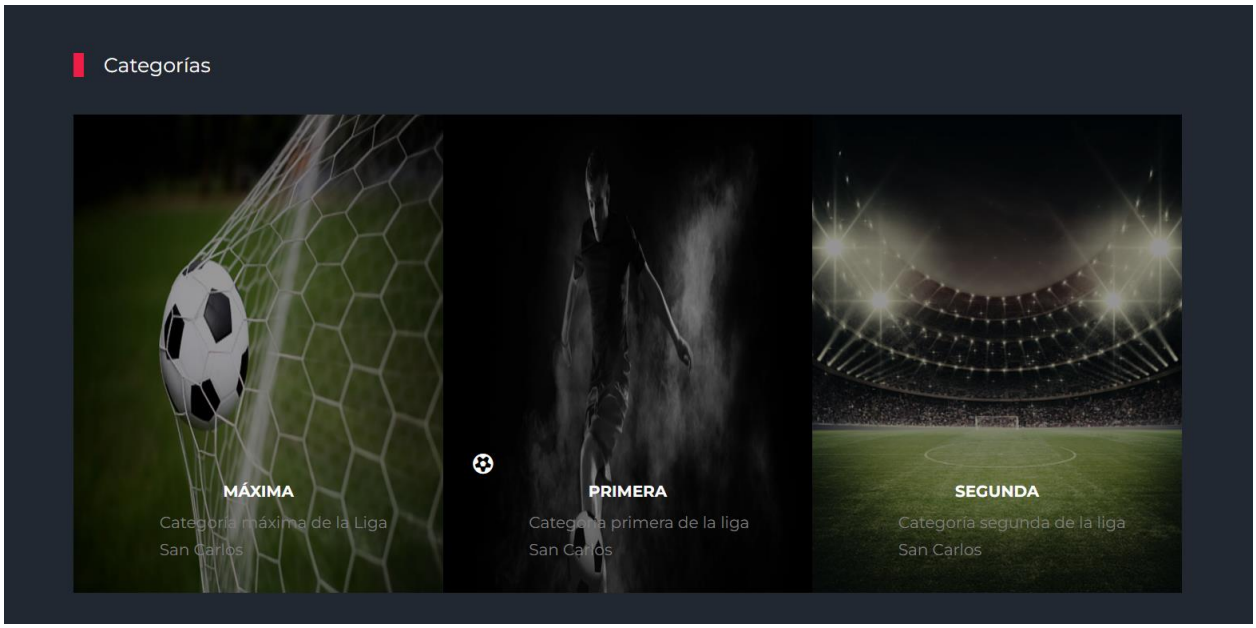
Anexo A: Pantalla Inicio de sesión

Anexo B: Pantalla principal

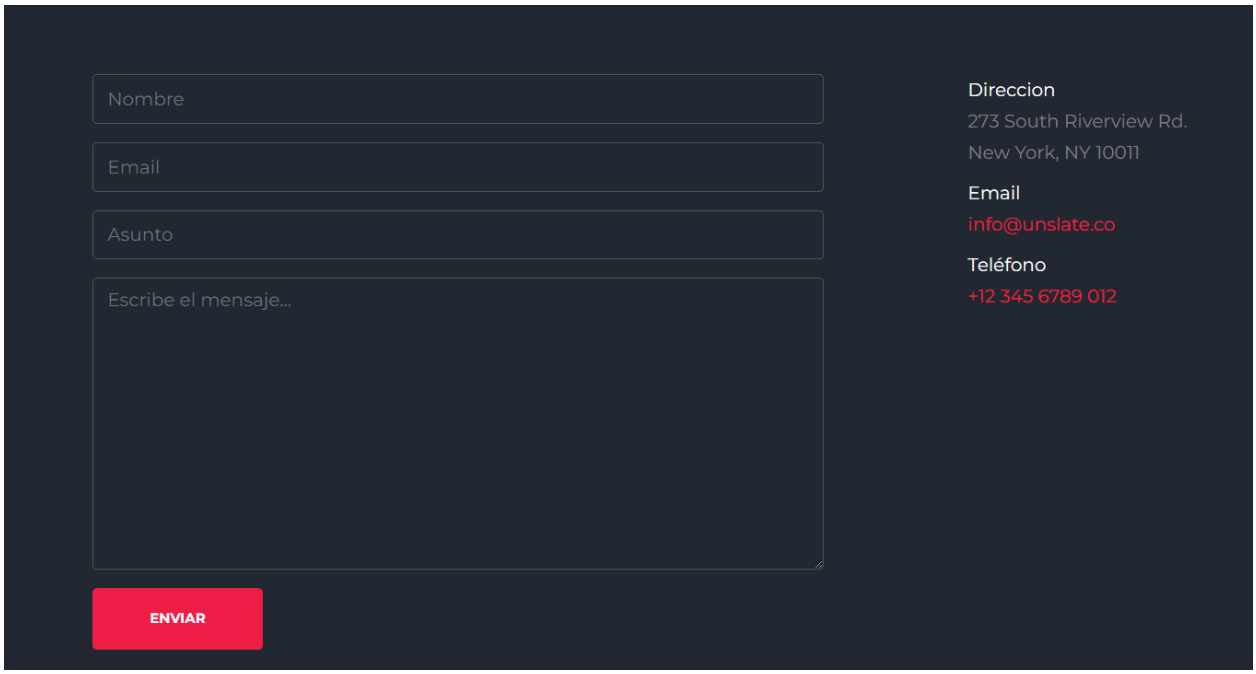




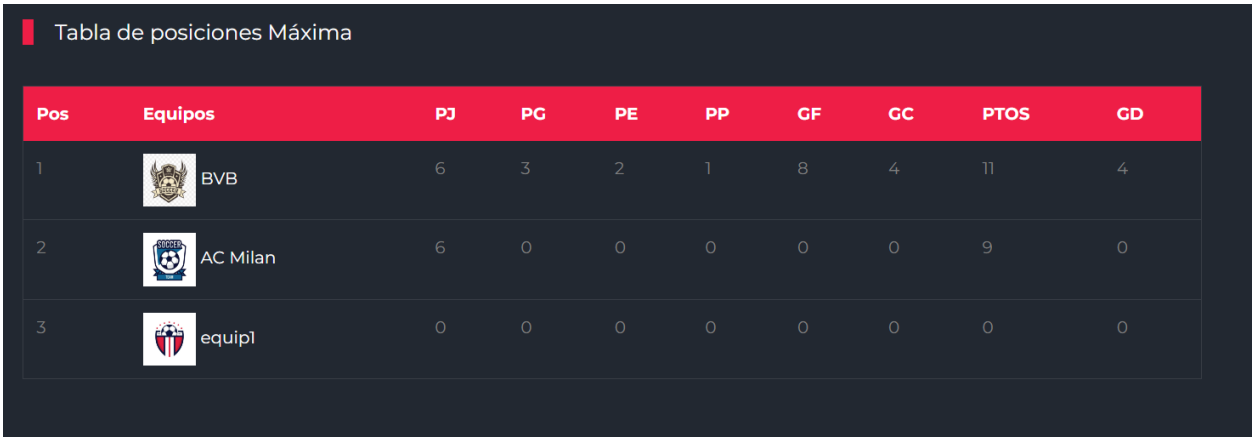
Anexo C: Categorías de la liga



Anexo D: Formulario de contacto



Anexo E: Pantalla CRUD jornadas

Anexo F: Pantalla CRUD equipos

Anexo G: Pantalla Tabla de posiciones