

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

**DESARROLLO DE UN PROTOTIPO DE UNA APLICACIÓN WEB
PARA LA GESTION DE TORNEOS DEPORTIVOS DE LA LIGA
DEPORTIVA CANTONAL PEDRO MONCAYO**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA Y REDES DE INFORMACIÓN**

FABIÁN ANDRÉS OLMEDO CARRILLO

DIRECTOR: M.Sc. XAVIER ALEXANDER CALDERÓN HINOJOSA

Quito, julio 2021

AVAL

Certifico que el presente trabajo fue desarrollado por Fabián Andrés Olmedo Carrillo, bajo mi supervisión.

M.Sc XAVIER ALEXANDER CALDERÓN HINOJOSA
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Fabián Andrés Olmedo Carrillo, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

FABIÁN ANDRÉS OLMEDO CARRILLO

DEDICATORIA

Dedico este trabajo de Titulación con todo mi amor y cariño a la estrella más reluciente del firmamento, a ese ángel primoroso que cada día de mi vida me acompaña y nunca me deja solo. Mi siempre amada y recordada Mamita Pía.

AGRADECIMIENTO

A Dios y mi querida Mama Nati, por brindarme la salud, guiarme por el buen camino y permitirme cumplir una meta más.

Al mejor papá del mundo mundial, ya que sin él nada hubiese podido ser, gracias por ser quién eres y brindarme todo el amor, el cariño, tus enseñanzas y nunca faltarme con nada.

A mi hermana Milagritos que, a pesar de su carácter difícil, siempre está conmigo en los momentos que la necesito, y a sus modos de consentirme.

A mi abuelito Tito quien me enseñó a ser un hombre de bien y siempre saber llevar en alto la lealtad y la honradez.

A mi mamá Margarita por todas sus malas noches y lágrimas para ver que su hijo cumpla sus metas y no permitir que en el aprendizaje de la vida me desvíe del sendero correcto.

A mi director, M.Sc_Xavier Calderón por darme la confianza en este proyecto de titulación, espero no haberle defraudado.

A mis queridos Milhouses: Juan, Negro, Omy, Ale, Pacu con quienes he compartido una amistad tan fraterna y sincera que podría confundirse con hermandad.

A mis Pécoras del Mal: Charly, Cris, Jenny, Suka, Pancho, con quienes forme otra familia más en mi vida politécnica, compartiendo experiencias y locuras que, si contáramos, las personas no nos creerían.

A mis niños redes: Ricky y Black, Negro querido mi eterno agradecimiento por aguantarme y enseñarme a programar y por los acolites en las materias.

A mis secretarias Verito y Alexita que aguantaron mis mandatos e impertinencias, pero siempre me recibieron con cariño y con una sonrisa.

A ti Vale, por enseñarme muchas cosas el tiempo que compartimos, también por ayudarme a pasar las materias más complicadas a tu puro estilo.

A mis amigos PICHONES, mi otra familia gracias por compartir tantas experiencias.

A mi querida gran jefa Eve, por ser esa maravillosa persona, acolite y berraca que siempre ha estado pendiente de este humilde personaje desde mi arribo a SIDEVOX.

A la empresa SIDEVOX quien me ha dado la oportunidad de crecer como profesional y como persona.

A mis niños padawans: Jimmy, Luis, Cris y Fanny por ser excelentes personas y brindarme su cariño y apoyo.

Estas últimas líneas las he reservado para una persona tan especial, mi amiga mi confidente, la que me conoces más que yo mismo, la que siempre me apoyado, ayudado, regañado y corregido, con quien hemos estado en las buenas y en las malas, a quien a pesar de haberle sacado de quicios nunca me dejo a mi suerte. Gracias muchas gracias por ser quién eres te quiero mucho Eri.

Fabo

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	VI
RESUMEN	VIII
ABSTRACT	IX
1. INTRODUCCIÓN.....	1
1.1 OBJETIVOS	2
1.2 ALCANCE	2
1.3 MARCO TEÓRICO.....	4
2. METODOLOGÍA.....	12
2.1. DEFINICIÓN DE REQUERIMIENTOS.....	12
2.1.1. SITUACIÓN ACTUAL DE LA GESTIÓN DE TORNEOS ORGANIZADOS POR LA LIGA CANTONAL PEDRO MONCAYO.....	12
2.1.2. ENTREVISTAS	13
2.1.3. ANÁLISIS DE REQUERIMIENTOS	17
2.1.3.1. Requerimientos Funcionales	18
2.1.3.2. Requerimientos No Funcionales.....	18
2.2. DISEÑO	19
2.2.1. TABLERO DE ACTIVIDADES KANBAN.....	19
2.2.2. DISEÑO DE LA CAPA DE DATOS.....	20
2.2.2.1. Diagrama Entidad-Relación.....	20
2.2.3. DISEÑO DE LA CAPA DE LÓGICA DE NEGOCIOS	23
2.2.3.1. Diagramas de Casos de Uso.....	23
2.2.3.2. Diagramas de Clases	27
2.2.4. DISEÑO DE LA CAPA DE PRESENTACIÓN	30
2.3. IMPLEMENTACIÓN.....	35
2.3.1. ACTUALIZACIÓN DEL TABLERO KANBAN	35
2.3.2. INSTALACIÓN DE LAS HERRAMIENTAS NECESARIAS.....	36

2.3.2.1.	Instalación del Sistema Operativo CentOS 7.....	36
2.3.2.2.	Instalación del servicio web Apache	41
2.3.2.3.	Instalación de PostgreSQL 9.6	41
2.3.2.4.	Instalación de Zend Framework.....	42
2.3.2.5.	Instalación de Atom	43
2.3.3.	IMPLEMENTACIÓN DE LA CAPA DE DATOS	44
2.3.4.	IMPLEMENTACIÓN DE LA CAPA LOGICA DE NEGOCIO	45
2.3.4.1.	Codificación de Clases	45
2.3.4.2.	Codificación del Servicio Web	49
2.3.5.	IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN.....	56
3.	RESULTADOS Y DISCUSIÓN	65
3.1	ACTUALIZACIÓN DEL TABLERO KANBAN.....	65
3.2	PRUEBAS DE FUNCIONAMIENTO.....	65
3.2.1	INICIAR SESIÓN EN LA APLICACIÓN WEB	66
3.2.2	GESTIÓN DE UN TORNEO.....	67
3.2.3	GESTIÓN DE UN EVENTO	71
3.2.4	GESTIÓN DE JUGADORES.....	74
3.2.5	GESTIÓN DE UN USUARIO.....	77
3.2.6	PRUEBAS DE FUNCIONAMIENTO DEL MÓDULO AGENDA	80
3.2.7	PRUEBAS DE FUNCIONAMIENTO DEL MÓDULO ESTADÍSTICAS	82
3.2.8	PRUEBAS DE FUNCIONAMIENTO DEL MÓDULO REPORTES	85
3.2.9	GESTIÓN DE UNA RESERVACIÓN.....	87
3.3	PRUEBAS DE VALIDACIÓN.....	90
3.4	TABLERO KANBAN FINAL.....	94
4.	CONCLUSIONES Y RECOMENDACIONES.....	96
4.1.	CONCLUSIONES.....	96
4.2.	RECOMENDACIONES	97
5.	REFERENCIAS BIBLIOGRÁFICAS	98
	ANEXOS	100

RESUMEN

Este trabajo de Titulación propone el desarrollo de un prototipo de aplicación web para la gestión de torneos de la Liga Deportiva Cantonal Pedro Moncayo. El prototipo está conformado por una base de datos y un servicio web.

El primer capítulo presenta los objetivos, el alcance que tendrá el trabajo de Titulación; además se describen de manera general fundamentos teóricos para el desarrollo del prototipo. Se describe la metodología ágil Kanban usada en el desarrollo de este prototipo.

El segundo capítulo se enfoca en detallar la metodología empleada para el desarrollo de este prototipo. Se presentan dos etapas la de Diseño e Implementación.

El tercer capítulo se detallan las pruebas de funcionamiento realizadas al prototipo para verificar su correcto funcionamiento.

El cuarto capítulo se presentan las conclusiones y recomendaciones, realizadas al término de este proyecto.

PALABRAS CLAVE: aplicación web, Zend Framework, Kanban, PostgreSQL, PHP, Apache

ABSTRACT

This degree work proposes the development of a web application prototype for the management of tournaments of the Pedro Moncayo Cantonal Sports League. The prototype is made up of a database and a web service.

The first chapter presents the objectives, the scope that the degree work will have; In addition, the theoretical foundations for the development of the prototype are described in a general way. The agile Kanban methodology used in the development of this prototype is described.

The second chapter focuses on detailing the methodology used for the development of this prototype. Two stages are presented: Design and Implementation.

The third chapter details the functional tests carried out on the prototype to verify its correct operation.

The fourth chapter presents the conclusions and recommendations made at the end of this project.

KEYWORDS: web application, Zend Framework, Kanban, PostgreSQL, PHP, Apache

1. INTRODUCCIÓN

Las aplicaciones web en la actualidad tienen una gran acogida por clientes y han ganado su popularidad gracias a su facilidad de acceso a la información y la practicidad de los navegadores web, ya que estos están disponibles en varios de los equipos que los usuarios manejan en su diario vivir: laptops, celulares, tablets entre otros; a eso se le suma que las aplicaciones web son fáciles de mantener, actualizar y no necesitan ser distribuidas e instaladas en los dispositivos de los potenciales usuarios.

El deporte atrae un sin número de fanáticos en todas las disciplinas; es por esto, que se necesita del apoyo de la tecnología para gestionar los torneos, competencias y brindar al usuario final información de calidad acorde a los gustos de cada uno de ellos.

En la web se encuentra varios softwares que pueden facilitar dicha gestión, pero se tiene una limitante económica para la personalización y no cumplen con requisitos específicos de pequeñas ligas o federaciones.

En el presente estudio técnico se ha desarrollado un prototipo de aplicación web para la gestión de torneos deportivos de la Liga Deportiva Cantonal Pedro Moncayo.

Esta aplicación es desarrollada para facilitar y automatizar la gestión de torneos, proveer información actualizada a los fanáticos del deporte cantonal, así como generar una agenda personal en la que pueden incluir los eventos deportivos de su agrado, estadísticas y notificaciones de los eventos agendados.

En este estudio técnico se detallan conceptos que facilitan el entendimiento del desarrollo de una aplicación web, así como las herramientas usadas para el desarrollo y la implementación de la aplicación usando una metodología ágil llamada Kanban.

Esta aplicación web posee una interfaz gráfica con varios módulos, los cuales se detallan más adelante, estos módulos son los necesarios para que la aplicación cumpla con los requerimientos del cliente y de los usuarios finales.

En capítulos posteriores encontraremos resultados del desarrollo del prototipo y para finalizar las conclusiones y las recomendaciones obtenidas en este estudio técnico.

1.1 OBJETIVOS

El objetivo general de este Proyecto Técnico es el de “Desarrollar un prototipo de una aplicación web para la gestión de torneos deportivos de la Liga Deportiva Cantonal Pedro Moncayo”.

Los objetivos específicos del Proyecto Técnico son:

- Describir los fundamentos teóricos para el desarrollo del prototipo de aplicación web.
- Diseñar los componentes que forman parte del prototipo de aplicación web: base de datos y aplicación web.
- Implementar los componentes del prototipo de aplicación web para la gestión de eventos y la administración de la liga.
- Analizar los resultados de las pruebas realizadas sobre el prototipo.

1.2 ALCANCE

El Trabajo de Titulación propone el desarrollo de un prototipo de aplicación web que permitirá la gestión de torneos deportivos de la Liga Deportiva Cantonal Pedro Moncayo, el cual estará formado por una base de datos y una aplicación web.

La Figura 1.1 muestra un esquema general del prototipo, donde se visualiza la propuesta de aplicación web para la gestión de torneos deportivos.



Figura 1.1 Esquema del prototipo de aplicación web para la gestión de torneos deportivos.

A continuación, se describen las herramientas que se utilizarán en la implementación del módulo web:

- Se utilizará un servidor web Apache.
- Se implementará una base de datos utilizando PostgreSQL.
- Se desarrollará una aplicación web utilizando el *framework* Zend.
- Se desarrollarán las interfaces gráficas para el usuario con la herramienta correspondiente.

Se tendrán tres tipos de usuarios:

- **Administrador:** quien se encargará de la creación de torneos, definición de la agenda del torneo mediante la aplicación web, así como la asignación de los escenarios correspondientes a cada encuentro, de crear, editar o eliminar a todos los usuarios sean estos deportistas, fanáticos o dirigentes deportivos.
- **Asistentes** (usuarios comunes): quienes podrán ver información de eventos, seleccionar los que sean de su interés, crear una agenda personalizada, ver reportaría y estadísticas del torneo, recibir notificaciones de próximos eventos y reservar cualquier escenario para un evento particular.

Los Asistentes al evento o fanáticos aficionados que ingresen por primera vez al sistema prototipo deberán registrarse para crear una cuenta.

- **Dirigentes deportivos:** quienes representan a cada una de las distintas delegaciones participantes tendrán la funcionalidad del asistente, con la diferencia que ellos podrán inscribir a sus deportistas representantes de cada disciplina.

El registro de los dirigentes será realizado exclusivamente por el administrador del sistema.

El sistema prototipo contará con al menos los siguientes módulos (Requisitos mínimos):

- **Administración de Usuario**

Este módulo contará con la capacidad de crear, leer, editar y eliminar los usuarios del sistema.

- **Administración de Disciplinas**

Este módulo contará con la capacidad de crear, leer, editar y eliminar las disciplinas que formarán parte de los torneos

- **Inscripción de deportistas.**

Este módulo contará con la capacidad de crear, leer, editar y eliminar las deportistas que formarán parte de los torneos y de cada una de las delegaciones., En este módulo se mostrará al dirigente deportivo y al administrador, además se podrá hacer una carga masiva mediante un archivo con extensión .csv.

- **Agenda**

En este módulo se mostrarán los eventos deportivos disponibles de acuerdo con la fecha actual, donde el usuario podrá seleccionar y ver la información de dicho evento como es la fecha y hora, el lugar a desarrollarse y los participantes. El usuario final podrá crear su agenda personalizada para recibir en el correo electrónico notificaciones del evento. al correo electrónico.

- **Administración de Torneos**

Este módulo contará con la capacidad de crear, leer, editar y eliminar los torneos del sistema.

- **Estadísticas**

Se mostrará un cuadro con los resultados de las competiciones por cada disciplina además de las posiciones en el medallero de cada torneo.

- **Reportería**

En este módulo se mostrarán reportes asociados a cada disciplina, delegación o deportista.

- **Reserva de escenarios.**

Este módulo contará con la capacidad de crear, leer, editar y eliminar las reservaciones de los escenarios deportivos para eventos particulares. El usuario final recibirá la notificación de aprobación o negación de la reserva del establecimiento deportivo solicitado.

1.3 MARCO TEÓRICO

A continuación, se detallarán conceptos y herramientas tecnológicas que ayudan a la implementación del prototipo de aplicación web. También se detallará la metodología ágil Kanban que permite la organización y planificación para el desarrollo del prototipo.

1.3.1 SERVIDOR WEB

Un Servidor web es una computadora, es decir, hardware y software ubicada en algún lugar del planeta, que se encarga de almacenar y entregar contenido a los clientes a través de

un sitio web. El contenido entregado por el servidor puede ser variado como, por ejemplo: imágenes, texto, video, datos de aplicación entre otros. Cuando los clientes realizan una petición al sitio web lo hacen a través del protocolo *HyperText Transfer Protocol* (HTTP), el cual permite la transferencia de información por medio del internet [1].

HyperText Transfer Protocol (HTTP) es un sencillo protocolo cliente-servidor, que permite intercambiar información entre los clientes web y el servidor HTTP. Esta soportado sobre los servicios de conexión TCP/IP, es decir, que el servidor escucha un puerto TCP (por defecto el 80), a la espera de solicitudes de conexión por parte de los clientes web.

HTTP se basa en operaciones sencillas de solicitud-respuesta. Por ejemplo, un cliente web realiza una petición desde su navegador al servidor web, al realizar esta petición el cliente establece una conexión con el servidor web y envía un mensaje con los datos la solicitud. El servidor responde a esta petición con un mensaje similar el cual contiene el estado de la solicitud y su posible resultado. Toda operación puede adjuntar un recurso u objeto sobre el que actúan. [2]

Los recursos u objetos que actúan como entrada o salida de un comando HTTP están clasificados por su descripción *Multipurpose Internet Mail Extensions* (MIME¹). De este modo permite al protocolo intercambiar cualquier tipo de datos. [2]

Características principales del protocolo HTTP:

- La comunicación entre cliente y servidor se realiza con caracteres de 8 bits, por lo que se puede transmitir cualquier tipo de datos o archivos.
- Existen varios verbos que un cliente puede usar para comunicarse con el servidor. La Tabla 1.1. muestra algunos de los verbos más usados.

Tabla 1.1. Verbos HTTP

Verbo	Descripción
GET	Solicita la representación de un recurso en específico.
HEAD	Realiza la misma acción que el GET, pero sin el cuerpo del mensaje.
POST	Se usa para enviar un recurso al servidor.

¹ MIME: Especificaciones para el intercambio de información a través del Internet y que es transparente al usuario final

PUT	Se envía recurso al servidor y también actualiza un recurso existente.
DELETE	Elimina un recurso existente.

- Cada operación HTTP conlleva una conexión al servidor la cual es liberada al término de la misma. Actualmente se pueden realizar una conexión que se mantiene activa durante un periodo de tiempo y permite realizar varias transacciones secuenciales. Este mecanismo se lo conoce como *Keep Alive*. [2]
- Cada petición realizada al servidor es tratada de forma independiente del resto, es decir que las peticiones no se afectan o no influye en nada las transacciones anteriores.

1.3.2 BASE DE DATOS

Una base de datos es básicamente es un sistema computarizado de almacenamiento de datos interrelacionados entre si correspondientes a un mismo contexto, ordenados de manera sistemática sin redundancias perjudiciales, para que posterior el usuario pueda acceder, eliminar, editar, agregar o actualizar dichos datos. Se podría comparar a una base de datos con un armario electrónico o un contenedor, donde se archivan registros o información que el usuario pueda utilizar. Las bases de datos son independientes a las aplicaciones que requieran usarlas [3].

1.3.3 APLICACIONES WEB

Una aplicación web es un software que se encuentra instalado en un servidor web, codificado en un lenguaje soportado por los navegadores web, lo que permite a los clientes acceder al contenido mediante dichos navegadores, haciendo uso del internet o intranet; en consecuencia, las aplicaciones web no tienen la necesidad de ser instaladas en los equipos de los clientes para su uso.

Las aplicaciones web se distinguen tres niveles: el nivel superior que interacciona con el usuario por medio de un navegador web, que es el encargado de la interpretación del código, el nivel intermedio es el que procesa los datos y el nivel inferior la base de datos [4].

Un navegador web es un aplicativo que permite al cliente acceder o interactuar con información de páginas o sitios web. El navegador tiene la capacidad de interpretar el

código que se encuentra en el servidor que por lo general suele ser escrito en HTML² y JavaScript³ y mostrar al usuario como una interfaz gráfica.

Las aplicaciones web en la actualidad han aumentado su popularidad debido a la facilidad de mantener y actualizar dichas aplicaciones ya que basta con actualizar en el servidor y así evitar la instalación a millones de clientes. Además, las aplicaciones web son independientes a los sistemas operativos.

1.3.4 POSTGRESQL

PostgreSQL es un potente sistema de base de datos relacional de objetos desarrollado por la Universidad de California, es de código abierto, utiliza y amplía el lenguaje SQL combinando con muchas características que almacenan y escala la carga de datos más complicadas. [5]

PostgreSQL garantiza su estabilidad debido a que utiliza multiprocesos en vez de multihilos, así si, un proceso falla este no afectará al resto y el sistema continuará funcionando. [6]

PostgreSQL debido a su arquitectura probada, confiabilidad, integridad de datos además de su alto rendimiento se ha ganado una consolidada reputación entre los desarrolladores y empresas. PostgreSQL se ejecuta en los principales sistemas operativos y cumple con ACID⁴ desde 2001. [7]

PostgreSQL además de ser gratuito y de código abierto tiene muchas características que ayudan a crear aplicaciones, a proteger la integridad de los datos y crear entornos tolerantes a fallas. También, administra los datos sin importar el tamaño del conjunto de datos. [7]

1.3.5 SERVICIO WEB APACHE

Un servicio web es un conjunto de protocolos y estándares que permite la comunicación entre dos sistemas mediante el uso de la red o Internet, es decir un servicio web permite intercambiar datos entre aplicaciones que manejan lenguajes distintos. Un servicio web

² HTML (*HyperText Markup Languaje*): es un lenguaje de marcado que se utiliza para el desarrollo de páginas web.

³ JavaScript: Es un lenguaje de programación interpretado orientado a objetos que permite implementar funciones complejas en páginas web.

⁴ ACID: (*Atomicity, Consistency, Isolation, Durability*):son las características de los parámetros que permiten clasificar las transacciones de la gestión de bases de datos.

puede ser implementado en diferentes sistemas operativos o arquitecturas [8]. Los servicios web tienen como protocolo básico XML, que se usa como formato de los mensajes de datos y como base de los protocolos SOAP, WSDL y UDDI [9].

XML (eXtensible Markup Language): es el lenguaje de etiquetas o marcas que se utiliza para transportar, organizar y almacenar datos y documentos. [10]

SOAP (Simple Object Access protocol): es un protocolo de mensajería basado en XML, que permite comunicar programas que corren en distintos sistemas operativos, es decir que nos indica como codificar los mensajes para que estos puedan circular entre las aplicaciones. [11]

WSDL (Web Services Description Language): define un mecanismo estándar para describir y localizar un servicio web, WSDL usa un vocabulario XML y de esta forma permite un entendimiento entre distintas tecnologías y arquitecturas. [8]

UDDI (Universal Description, Discovery and Integration): este protocolo proporciona un mecanismo estándar para registrar y localizar los servicios web. Los directorios UDDI se comportan como una guía telefónica de servicios web. [12]

Apache es un servidor HTTP de código abierto seguro, eficiente y extensible que proporciona servicios HTTP en sincronía con los estándares actuales. Apache está orientado a sistemas operativos modernos como UNIX y Windows.

Apache carece de una interfaz gráfica para su configuración, aunque es altamente configurable. Admite la negociación de contenidos y bases de datos de autenticación. Soporta lenguajes como PHP⁵ y ASP⁶ con los módulos adecuados.

Apache se encarga de garantizar la comunicación entre el cliente y el servidor. Cuando un cliente por medio del navegador web solicita la información al servidor Apache, este responde con la información solicitada, mediante el protocolo HTTP.

1.3.6 ZEND FRAMEWORK

Zend Framework es una colección de paquetes PHP, que se pueden utilizar para el desarrollo de aplicaciones y servicios web. Proporciona un código orientado a objetos

⁵ PHP: es un lenguaje de programación de uso general que se adapta especialmente al desarrollo web.

⁶ ASP: es un entorno para aplicaciones web desarrollado y comercializado por Microsoft, permite el desarrollo de aplicaciones web dinámicas.

usando un amplio espectro de características del lenguaje. Zend Framework usa Composer como administrador de dependencia de paquetes, es decir, es aquel que almacena e instala las librerías necesarias para el desarrollo web; PHPUnit⁷ para probar todos los paquetes; y Travis CI⁸ como servicio de Integración Continua. Zend Framework también sigue los estándares PHP-FIG, que estandarizan los conceptos de programación en PHP e incluye una implementación de PSR-7 para interfaces de mensajes HTTP [13]. La Tabla 1.2 resume las interfaces de mensajes HTTP

Tabla 1.2. Interfaces de mensajes HTTP

Interfaces de mensajes	Descripción
interface MessageInterface	Esta interfaz define los métodos comunes a cada uno de los mensajes HTTP.
interface RequestInterface	Representación de una solicitud del lado del cliente.
interface ServerRequestInterface	Representación de una solicitud HTTP entrante del lado del servidor.
interface ResponseInterface	Representación de una respuesta saliente del lado del servidor.
interface StreamInterface	Esta interfaz proporciona un contenedor para las operaciones más comunes, incluida la serialización de todo el flujo en una cadena.
interface UriInterface	Esta interfaz está destinada a representar URI de acuerdo con RFC 3986 y proporcionar métodos para las operaciones más comunes
interface UploadedFileInterface	Objeto de valor que representa un archivo cargado a través de una solicitud HTTP

Zend *Framework* nos permite desarrollar las aplicaciones web mucho más rápido, debido a que posee varios componentes que se pueden utilizar como código base para la aplicación. Zend *Framework* ayuda a tener sitios web seguros con una estructura consistente y fácil de mantener, gracias al patrón modelo-vista-controlador que separa la lógica del negocio y la capa de presentación. [14]

Zend *Framework* provee componentes como filtros y validadores para los datos ingresados en los formularios, escapar la salida HTML, algoritmos criptográficos, detectores de humanos y falsificación de petición en sitios cruzados. [15]

⁷ PHPUnit: Se utiliza para pruebas de programadores PHP.

⁸ Travis CI: Se utiliza para pruebas de código.

Zend *Framework* usa el concepto de módulos que permite reutilizar los modelos, vistas y controladores, haciendo a un sitio web escalable. También permite usar las bases de datos de manera orientada a objetos en lugar de utilizar directamente consulta SQL. [15]

1.3.7 METODOLOGÍA ÁGIL KANBAN

La palabra Kanban viene del japonés que significa tarjeta con signos. La metodología ágil Kanban está basada en una idea simple “Trabajo en progreso”, que se enfoca en realizar tareas pendientes y los principios más importantes, pueden ser divididos en cuatro principios básicos y seis prácticas [16].

Principios Básicos:

- Empezar con lo que se hace ahora: Kanban puede ser aplicado a cualquier flujo real de trabajo para identificar los problemas ya que no es necesario realizar cambios drásticos. [16]
- Comprometerse a buscar e implementar cambios incrementales y evolutivos: Los cambios que deben implementarse son pequeños incrementales y continuos, no se realizan cambios grandes debido a la resistencia y la incertidumbre [16].
- Respetar los procesos, las responsabilidades y los cargos actuales: Kanban no prohíbe el cambio, pero tampoco lo prescribe [16].
- Animar el liderazgo en todos los niveles [16].

Prácticas de Kanban:

- Visualizar el flujo de trabajo: en todo proyecto se necesita conocer las actividades a realizarse desde el inicio hasta la entrega del proyecto, así se puede mejorar y hacer los ajustes a cada una de las actividades. Kanban propone tener un tablero que al menos conste de tres columnas, que serían: La tarea por empezar, en progreso y finalizada, así se podrá detectar los cuellos de botella y seguir el progreso del proyecto [16]. En la Figura 1.2 se muestra un ejemplo de un tablero Kanban



Figura 1.2 Ejemplo de Tablero Kanban

- Eliminar las interrupciones: En esta práctica se busca poner límites al trabajo en progreso ya que si no hay límites no existe Kanban. Se debe implementar el sistema de arrastre a una parte del flujo de trabajo o a su totalidad, así se podrá identificar áreas problemáticas en el flujo del trabajo.
- Gestionar el flujo: el gestionar el flujo permite alcanzar el objetivo de tener un trabajo rápido e ininterrumpido y así evitar retrasos.
- Hacer las políticas explícitas: No se puede mejorar lo que no se entiende, por este motivo el proceso debe ser explícito, publico y bien definido.
- Circuitos de retroalimentación: Se debe tener reuniones con frecuencia, para realizar revisiones y entregas de los avances del proyecto. Se recomienda reuniones de 10 a 15 minutos.
- Mejorar colaborando: Los grupos de trabajo pueden tener ideas distintas, pero la colaboración ayuda a tener un mejor éxito en la ejecución del proyecto.

Por lo expuesto anteriormente la metodología ágil Kanban se encuentra actualmente en una de las más usadas, ya que su aplicación puede ser a distintos tipos de proyectos. Las empresas grandes usan tableros digitales, esto les permite distribuir a todo el equipo de trabajo sin la necesidad de reunirlos.

2. METODOLOGÍA

En este capítulo, se desarrollan tres temas principales que son: Definición de Requerimientos, Diseño e Implementación del prototipo de aplicación web para la gestión de torneos de la Liga Deportiva Cantonal Pedro Moncayo.

En la Definición de Requerimientos, primero se establece la situación actual de la organización y ejecución de los torneos organizados por la Liga Cantonal Pedro Moncayo. Luego, en base a las encuestas realizadas a los miembros del Directorio de la Liga Cantonal y su análisis correspondiente, se definen los Requerimientos Funcionales y No Funcionales con los que cumplirá el prototipo.

En el Diseño de la aplicación, Primero se detalla el Tablero Kanban que permite tener una organización adecuada para el desarrollo del prototipo. Después el diseño se divide en 3 Capas: Datos, Lógica de Negocio y Presentación, que forman parte de los componentes del prototipo.

En cuanto a la Implementación, está basado en la codificación del diseño de los componentes que forman parte del prototipo.

2.1. DEFINICIÓN DE REQUERIMIENTOS

En este apartado, se define la situación actual de la gestión de los torneos organizados por la Liga Deportiva Cantonal Pedro Moncayo. Luego, se detallan los requerimientos funcionales y no funcionales en base a las entrevistas realizadas y procesadas a los miembros del directorio de la Liga Cantonal.

2.1.1. SITUACIÓN ACTUAL DE LA GESTIÓN DE TORNEOS ORGANIZADOS POR LA LIGA CANTONAL PEDRO MONCAYO

La Liga Deportiva Cantonal Pedro Moncayo se constituye en la máxima entidad del deporte del cantón. Su actividad está sujeta a las disposiciones contenidas en la Ley de Educación Física y Recreación, es una entidad de Derecho Privado, autónoma y con personería jurídica, su organización es dependiente de Concentración Deportiva de Pichincha. Tiene como propósito esencial la planificación, fomento, organización, ejecución, coordinación y control de los deportes como medio formativo, el desarrollo integral de las personas en todas las edades para el cumplimiento eficaz de sus obligaciones como miembros de la sociedad. [17]

En la actualidad, La Liga Deportiva Cantonal Pedro Moncayo presenta un problema en la gestión de sus torneos al realizarlos de manera manual, ya que, al reunir a una gran cantidad de atletas, entrenadores, dirigentes deportivos, jueces y escenarios deportivos,

se necesita de un gran despliegue en la organización, ejecución, recopilación de la información requerida de cada uno de los participantes y la difusión de la información de los torneos, así como también, se obtiene una gran cantidad de documentación recibida en hojas, libros de actas, entre otras y que son almacenadas y organizadas en carpetas en un archivero de metal, por ejemplo: listado de jugadores, actas de juego, calendarios, y resultados.

Con el fin de mejorar la gestión de torneos, se plantea el diseño e implementación de un prototipo de aplicación web que permita a la Liga Deportiva Cantonal Pedro Moncayo desempeñar su gestión de una mejor manera. La Liga Cantonal se beneficiará a través del desarrollo de este trabajo, ya que, el software propuesto automatizará procesos que se realizan de forma manual; previniendo la pérdida de documentos y ahorrando espacio físico. Por otra parte, brindará eficiencia en la notificación de los eventos deportivos a través de correo electrónico.

2.1.2. ENTREVISTAS

Se aplicó una entrevista dirigida a tres miembros del directorio de la Liga Deportiva Cantonal Pedro Moncayo, quienes son los encargados de la organización de los torneos. La entrevista formulada consta de nueve preguntas y el modelo se presenta en el ANEXO A.

La primera pregunta permite conocer si en la organización de torneos ha existido pérdida de documentación relevante; En la Figura 2.1 se presentan los resultados obtenidos, en los cuales el 100% de los entrevistados considera que si ha existido pérdida de documentación.



Figura 2.1 Resultados de la pregunta 1

La segunda pregunta busca conocer si se realizan tareas manuales en la organización de torneos. En la Figura 2.2, se muestran los resultados donde el 100% de los encuestados afirma que se realizan tareas manuales.



Figura 2.2 Resultados de la pregunta 2

La tercera pregunta busca conocer si la Liga Deportiva Cantonal cuenta con un registro de los deportistas. Los resultados obtenidos se presentan en la Figura 2.3, donde el 100% de los encuestados indica que no se tiene un registro de los deportistas.



Figura 2.3 Resultados de la pregunta 3

La cuarta pregunta y sus resultados se muestran en la Figura 2.4, donde se busca conocer si la Liga Cantonal tiene un registro de sus establecimientos deportivos a lo cual el 100% de los entrevistados indicó que no.



Figura 2.4 Resultados de la pregunta 4

La quinta pregunta busca conocer si es necesario el desarrollo de la aplicación web, a la cual el 100% de los encuestado manifestó estar de acuerdo. Los resultados se muestran en la Figura 2.5

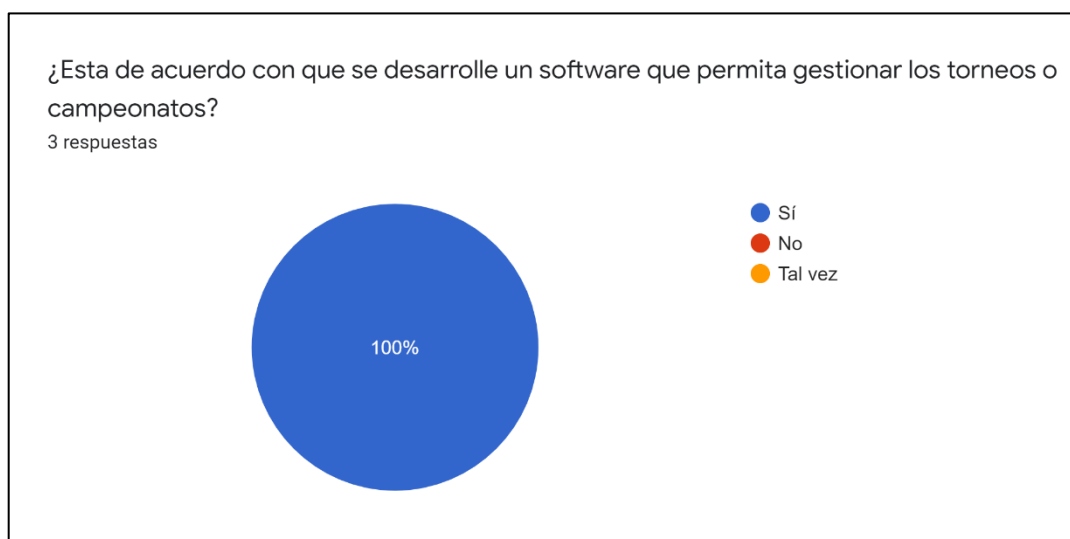


Figura 2.5 Resultados de la pregunta 5

La sexta pregunta busca conocer el modo de difusión de los eventos deportivos organizados por la Liga Deportiva Cantonal. Los resultados obtenidos se presentan en la Figura 2.6, donde el 66,7% de los encuestados manifestó que por redes sociales y el otro 33,3% mediante perifoneo.



Figura 2.6 Resultados de la pregunta 6

La séptima pregunta busca conocer si la aplicación será utilizada para la gestión de los torneos, a lo cual el 100% de los entrevistados manifestaron que si la utilizarían. Los resultados obtenidos se muestran en la Figura 2.7.

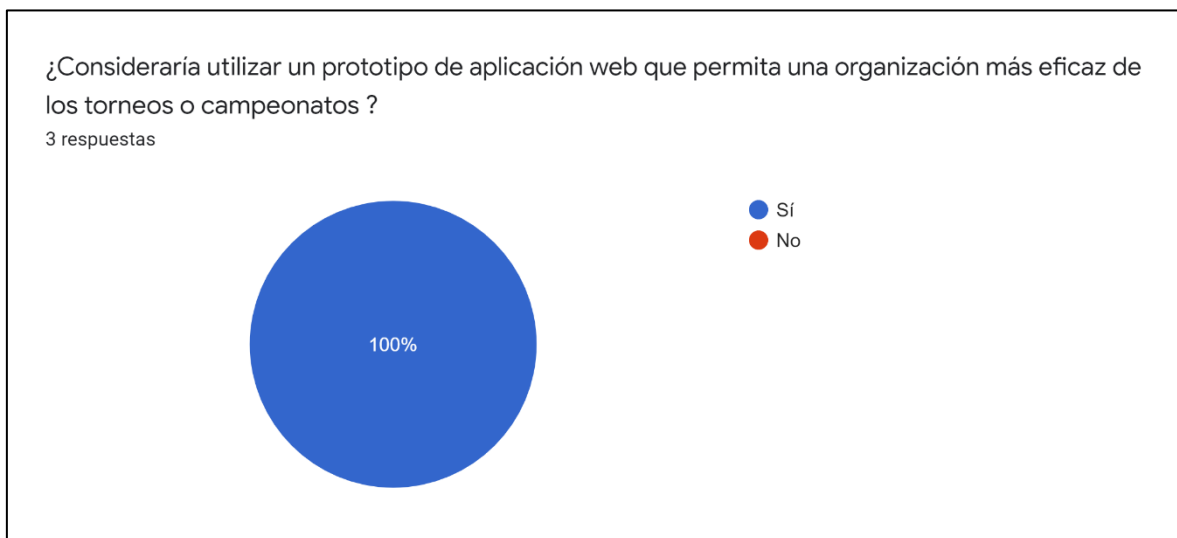


Figura 2.7 Resultados de la pregunta 7

La octava pregunta tiene como objetivo establecer el tiempo adecuado para el envío de notificaciones de los eventos deportivos. La Figura 2.8 presenta los resultados obtenidos, se observa que para los entrevistados los tiempos mostrados en las opciones son adecuados para este tipo de eventos.

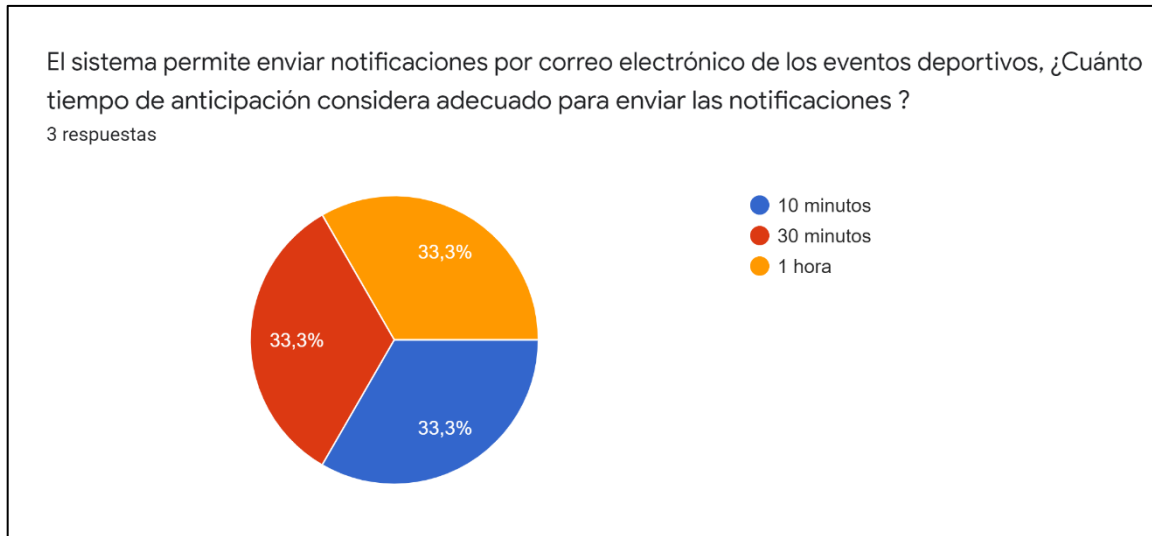


Figura 2.8 Resultados de la pregunta 8

La novena pregunta busca establecer si el inicio de sesión para la manipulación de la aplicación es importante, a lo que el 100% de entrevistados manifestó que si es importante. Los resultados obtenidos se muestran en la Figura 2.9.



Figura 2.9 Resultados de la pregunta 9

2.1.3. ANÁLISIS DE REQUERIMIENTOS

Con la información obtenida en las entrevistas a los miembros del directorio de la Liga Cantonal Pedro Moncayo y el levantamiento de información durante el desarrollo del Plan de este Trabajo de Titulación, se establecen los requerimientos para la aplicación web, los cuales son descritos a continuación.

2.1.3.1. Requerimientos Funcionales

- Se tendrán tres tipos de usuarios: administrador, dirigente, asistente (usuario común)
- El administrador se encargará de la creación, modificación y eliminación de torneos, eventos, usuarios, escenarios deportivos y estadísticas.
- El administrador reestablecerá la contraseña de los dirigentes.
- El administrador será el encargado de entregar las credenciales de acceso a los dirigentes.
- El dirigente podrá ver y crear los deportistas de sus delegaciones.
- El dirigente podrá ver las estadísticas, eventos y reportes
- El dirigente podrá gestionar una agenda y gestionar reservaciones.
- Los asistentes o usuarios comunes deberán registrarse en el sistema y luego iniciar sesión para acceder a las funcionalidades.
- Los asistentes podrán ver los eventos disponibles y agregar los de su interés a una agenda personalizada, recibir notificaciones de los eventos agregados a la agenda.
- Los asistentes podrán ver el módulo de estadísticas donde se muestran medalleros y tablas de posiciones.

2.1.3.2. Requerimientos No Funcionales

- Se usa un modelo cliente servidor para el prototipo de aplicación web.
- Los datos se almacenarán en una base de datos PostgreSQL.
- La aplicación web se desarrollará con Zend Framework.
- Las contraseñas serán cifradas con md5 antes de ser almacenadas en la base de datos.

2.2. DISEÑO

En esta sección, se detalla el diseño del prototipo de aplicación web para la gestión de torneo que realiza la Liga Deportiva Cantonal Pedro Moncayo. Primero se aplica la metodología Kanban, para lo cual se define el tablero de actividades que permite el desarrollo de este trabajo.

En la Capa de Datos se realiza un diagrama relacional siguiendo las buenas prácticas definidas en la documentación del lenguaje SQL. Mientras que, la Capa de Lógica de Negocio incluye el diagrama de casos de uso y diagrama de clases. Finalmente, en la Capa de Presentación se muestran los bosquejos de la aplicación.

2.2.1. TABLERO DE ACTIVIDADES KANBAN

Trello es una solución fácil, gratuita, flexible y visual para gestionar proyectos y organizarlo todo [18]. Esta aplicación web está orientada para administración de proyectos basada en metodología Kanban y se encuentra disponible para sistemas operativos IOS, Android y Windows.

Trello permite distribuir pendientes o tareas de un proyecto en base a un tablero principal, en el cual se generan diversas listas con títulos para cada etapa del proyecto. En cada lista se pueden agregar varias tarjetas que contienen las actividades a realizar, dichas tarjetas pueden ser editadas y ordenadas ya que usa la modalidad *drag & drop* (arrastrar y soltar).

Haciendo uso de la aplicación Trello, se realiza el Tablero de actividades Kanban para el desarrollo del prototipo de aplicación web. En la Figura 2.10 se observa el tablero Kanban el cual presenta las siguientes listas: “Lista de Tareas” que se refiere al conjunto de actividades a realizar de todo el proyecto, “En Proceso” como su nombre lo dice son las actividades que están en proceso de desarrollo, de acuerdo con el avance del proyecto. Mientras que, la lista “Hecho” es el conjunto de actividades completadas. Además, cada una de las tareas tiene un color específico que hace referencia a la fase del proyecto, los colores son: “Amarillo” que corresponde a la Fase Teórica, “Verde” las tareas correspondientes a la Fase de Diseño, “Azul” son las actividades de la Fase de Implementación, por último, “Rojo” que son las actividades de la Fase de Pruebas.

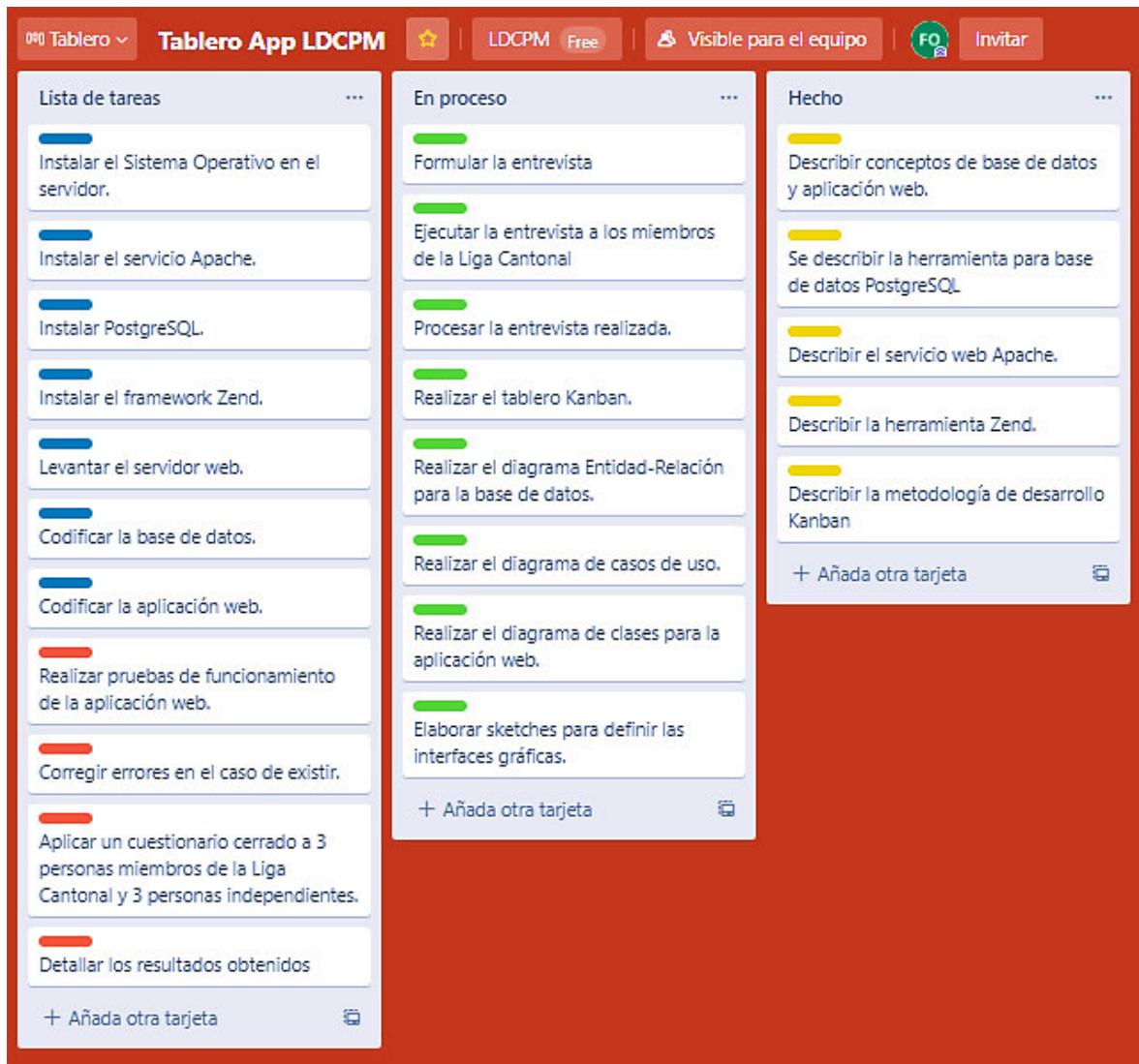


Figura 2.10. Tablero de Actividades Kanban.

2.2.2. DISEÑO DE LA CAPA DE DATOS

En esta sección, al usar PostgreSQL, se realiza el diseño del diagrama entidad relación de la base de datos que se va a utilizar teniendo en cuenta los requerimientos obtenidos previamente.

2.2.2.1. Diagrama Entidad-Relación

Un diagrama entidad-relación, es un tipo de diagrama que ilustra cómo las "entidades", tales como: personas, objetos o conceptos, se relacionan entre sí dentro de un sistema. Los diagramas ER se usan para diseñar o depurar bases de datos relacionales. [19]

En la Figura 2.2 se aprecia el diagrama entidad relación para la base de datos SQL, que permitirá el almacenamiento de los datos para el prototipo de aplicación web. Para evitar el conflicto con caracteres especiales, en su mayoría tanto las entidades como sus atributos

se encuentra escritos en inglés. Este diagrama fue desarrollado con el software Valentina Studio el cual permite realizar diagramas para PostgreSQL, MySQL en otras. Valentina Studio [20] es un gestor de base de datos SQL, permite editar o diagramar esquemas, descubre la base de datos de tipo SQL y la muestra como diagrama visual, también tiene edición SQL, así como reportes y formularios que hacen de este software robusto y muy utilizado por desarrolladores.

En este diagrama se considera que la Liga Cantonal necesita saber a qué disciplina y delegación pertenece un atleta o entrenador, para tener registrado en su base de datos, también necesita saber los jueces que serán parte del torneo, así como los escenarios deportivos a usarse para el desarrollo de las competencias, adicionalmente se necesita saber los eventos deportivos que va a disputar cada una de las delegaciones con sus correspondientes disciplinas registradas en un torneo. Una vez iniciado el torneo se requiere saber la posición en la que se encuentra cada una de las delegaciones de acuerdo con las disciplinas participantes, las estadísticas y reportes que se vayan generando en cada uno de los eventos.

Al termino de cada torneo se requiere conocer el medallero general, así como medalleros individuales de cada una de las disciplinas registradas.

Por lo tanto, en la Figura 2.11 se muestra como la entidad “*Events*” se relaciona con las entidades “*Tournament*”, “*Referee*”, “*Escenarios*” y “*Disciplinas*”. A su vez, tenemos entidades “*Events*” específicas para cada disciplina tales como: “*Events_futbol*”, “*Events_basket*”, “*Events_Chess*” entre otras, que fueron obtenidas de los requerimientos, las cuales se relacionan con las entidades “*Positions*”, “*Medalleros*”, “*Events*”. La entidad “Reserva Escenario” permitirá guardar los registros de las reservaciones a eventos privados realizados por usuarios naturales.

La entidad “*Sys_Users*”, permitirá el acceso al sistema después de que se haya registrado, esta entidad se relaciona con “*Admin*” y “*Admin_Profile*” la cual asocia el perfil de cada usuario y su nivel de permiso para acceder a las funcionalidades del sistema. La información de un atleta es almacenada en la entidad “*Deportistas*” y para registrar la disciplina a la cual pertenece se utilizará la entidad “*Dis_tournament_canton*”. Por último, tenemos la entidad “*Agenda*” la cual guarda los eventos que sean de interés para el usuario.

En la Tabla 2.1 se explica como ejemplo los atributos que forman parte de la entidad “*Sys_Users*”, las entidades restantes se encuentran en el **ANEXO B**.

Tabla 2.1. Entidad “*Sys_Users*”

Entidad “ <i>Sys_Users</i> ”		
Atributo	Tipo de Dato	Descripción
id	<i>Serial</i>	Identificador único de un usuario.
_canton	<i>Int</i>	<i>Foreign key</i> , Identificador único de un cantón.
Username	<i>Text</i>	Nombre de un usuario para <i>login</i> .
Firstname	<i>Text</i>	Nombres del Usuario.
Lastname	<i>Text</i>	Apellidos del usuario.
<i>Profile</i>	<i>Text</i>	Perfil del usuario, para indicar si tiene el rol de Administrador, Dirigente o Usuario natural.
Email	<i>Text</i>	Correo Electrónico de un usuario.
<i>Password</i>	<i>Text</i>	Contraseña cifrada del usuario.
<i>Status</i>	<i>Boolean</i>	Indicador para cuando el usuario ingrese por primera vez.
<i>Created</i>	<i>Timestamp</i>	Indicador de la fecha de creación del usuario.

2.2.3. DISEÑO DE LA CAPA DE LÓGICA DE NEGOCIOS

En esta capa, se diseña toda la lógica necesaria a la capa de presentación, para que el usuario a través de una interfaz gráfica pueda acceder a las funcionalidades que presenta la aplicación web. Para esto se utilizará los diagramas de Casos de Uso y el diagrama de clases.

2.2.3.1. Diagramas de Casos de Uso

En el diagrama de casos de uso se representan las funcionalidades del sistema desde la perspectiva del usuario (actor en UML⁹). Este actor no es necesariamente un humano

⁹ UML: lenguaje de modelado de sistemas de software.

también puede ser un rol del sistema externo que accede a otro sistema. De este modo, el diagrama de casos de uso muestra la relación entre un actor y sus requisitos o expectativas del sistema, sin representar las acciones que tienen lugar o ponerlas en un orden lógico. [21]

En el presente prototipo se definen tres tipos de actores: Administrador, Asistentes y Dirigentes.

En la Figura 2.12 se muestra los tres actores principales y su interacción con el prototipo.

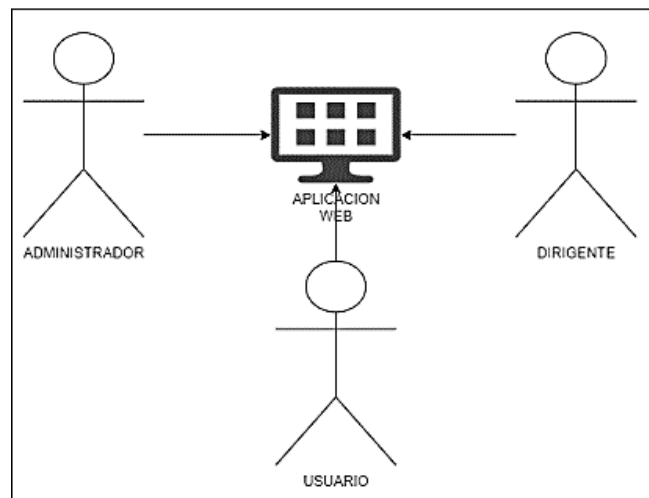


Figura 2.12 Actores que interactúan con el sistema

A continuación, se detalla el rol de cada uno:

- **Administrador:** este actor hace referencia a un Usuario al que se le ha otorgado el rol de Administrador. Un Administrador es el encargado de la creación de torneos, definición de la agenda del torneo mediante la aplicación web, así como la asignación de los escenarios correspondientes a cada encuentro, de crear, editar o eliminar a todos los usuarios sean estos deportistas, fanáticos o dirigentes deportivos, también puede gestionar las estadísticas de cada torneo y las reservaciones de los escenarios deportivos. En la Figura 2.13 se muestra el caso de uso para el administrador.

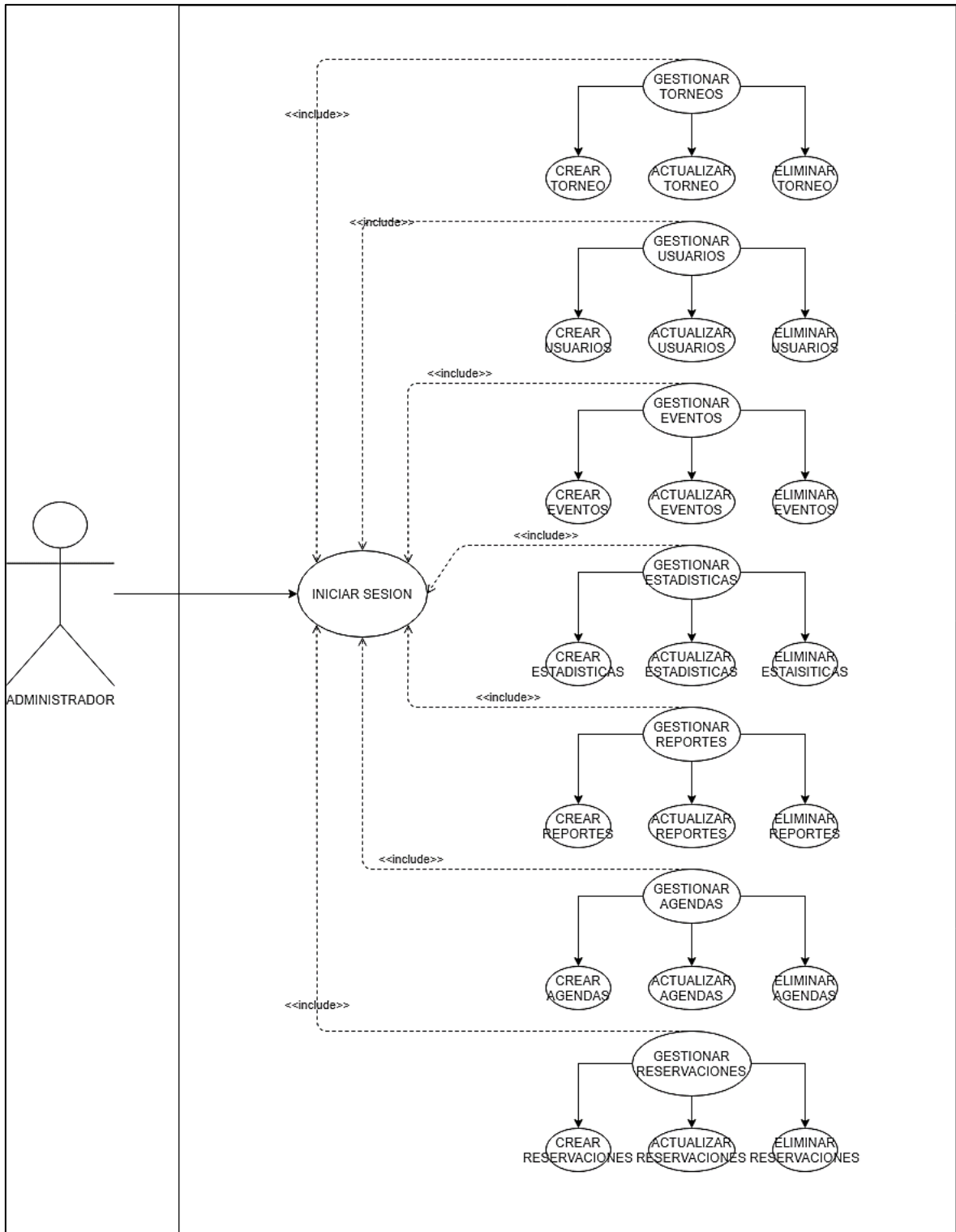


Figura 2.13 Diagrama de caso de uso Administrador.

- **Asistentes (usuarios comunes):** este actor hace referencia a cualquier usuario que desee asistir a los eventos deportivos y realizar reservaciones de los escenarios para eventos particulares. Los asistentes podrán ver información de eventos, seleccionar los que sean de su interés, crear una agenda personalizada,

ver reportaría y estadísticas del torneo, recibir notificaciones de próximos eventos y reservar cualquier escenario para un evento particular. Los Asistentes al evento o aficionados que ingresen por primera vez al sistema prototipo deberán registrarse para crear una cuenta. En la Figura 2.14 se muestra el caso de uso para los asistentes o usuarios comunes.

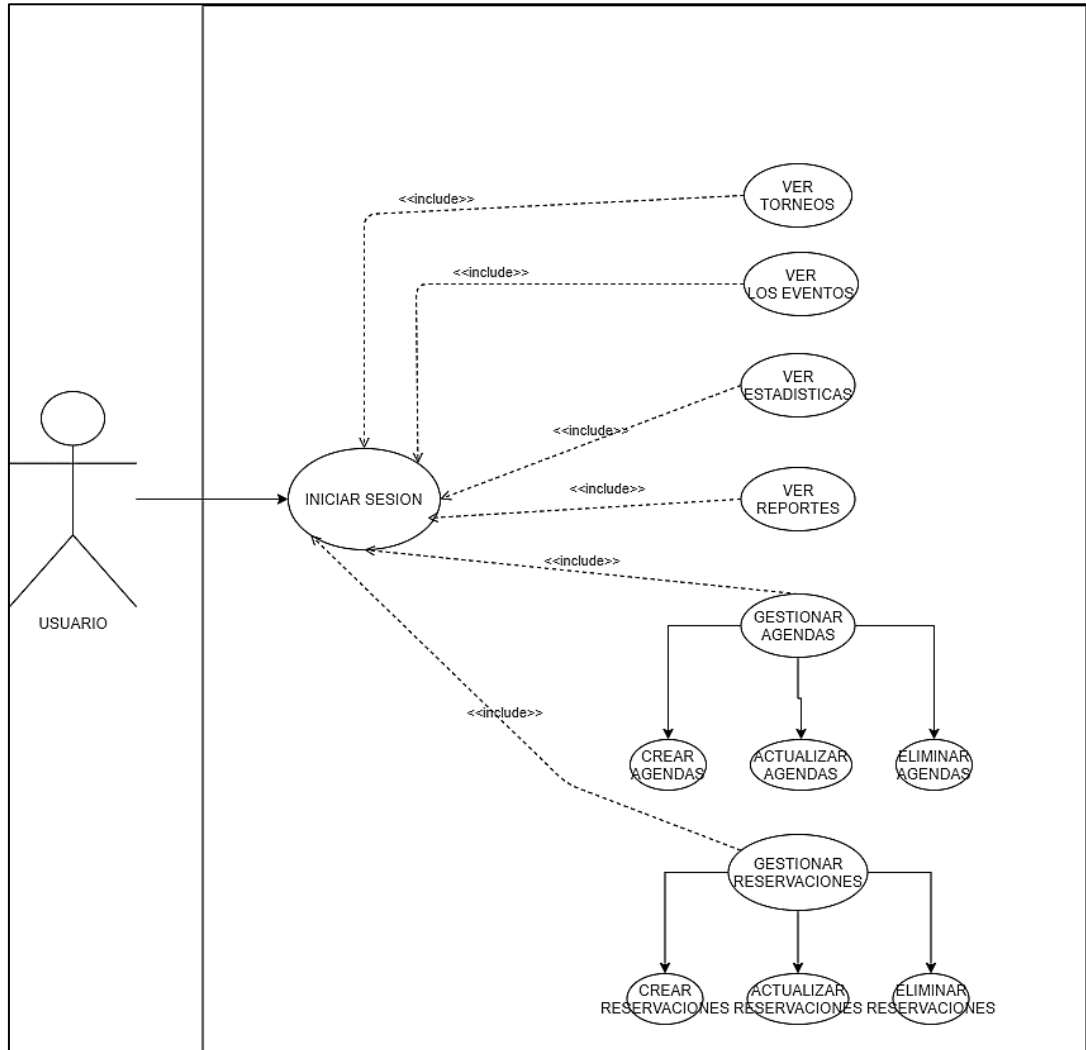


Figura 2.14 Diagrama caso de uso Usuario

- **Dirigentes:** este actor hace referencia a un Usuario al que se le ha otorgado el rol de Dirigente. Un Dirigente representa a cada una de las distintas delegaciones participantes, tendrá la funcionalidad del asistente, con la diferencia que ellos podrán inscribir a sus deportistas representantes de cada disciplina. El registro de los dirigentes será realizado exclusivamente por el administrador del sistema. En la Figura 2.15 se muestra el caso de uso del actor Dirigente.

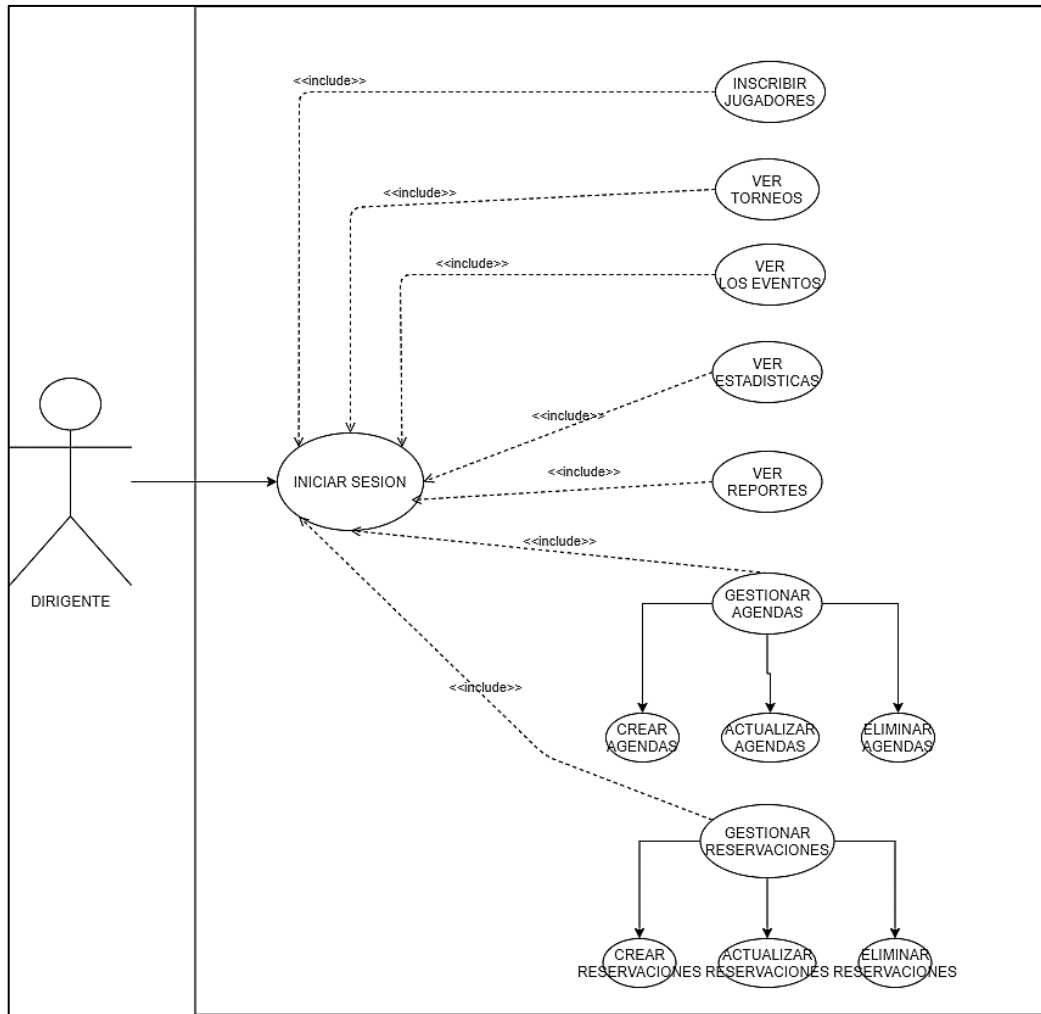


Figura 2.15 Diagrama caso de uso Dirigente.

2.2.3.2. Diagramas de Clases

El diagrama de clases forma parte de los diagramas de estructura UML, en la cual permite representar de mejor manera los elementos y componentes de un sistema. El diagrama de clases es orientado al modelo de programación orientado a objetos, en donde se definen las clases, sus atributos, funciones y relaciones que posteriormente se utilizarán en la fase de implementación del sistema [22].

En la representación de una clase se tiene un rectángulo que se divide en tres zonas utilizando líneas horizontales. La primera zona se utiliza para el nombre de la clase, la zona siguiente se escriben los atributos de la clase, uno por línea, y en la última zona se incluyen los métodos o funciones que la clase ofrece.

En la Figura 2.16 se muestra el diagrama de clases de la aplicación web para la gestión de torneos de la Liga Cantonal Pedro Moncayo. Para realizar este diagrama se utilizó el software Draw.io [23].

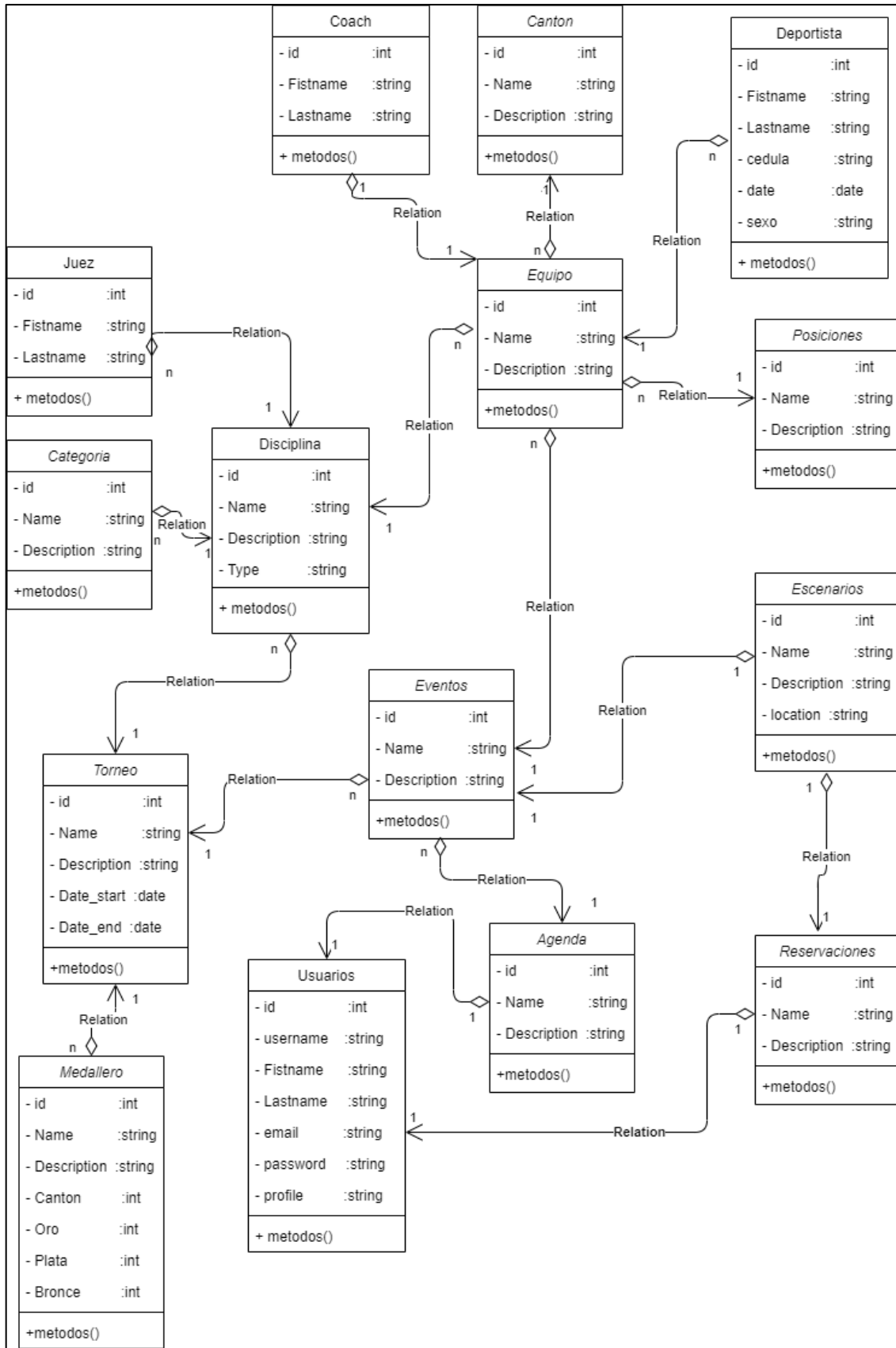


Figura 2.16 Diagrama de Clases de la Aplicación web.

Es importante mencionar que por falta de espacio en el Diagrama de Clases no se han colocado detalladamente los métodos, estos métodos han sido incorporados en base a la documentación de “Zend framework” y que permiten realizar acciones CRUD (*Create, Read, Update, Delete*).

En el diagrama, la clase “Disciplina” y las clases “Juez”, “Equipo” y “Categoría” tienen una multiplicidad de “uno” a “n” tal que una disciplina puede tener varias categorías, equipos y jueces. la clase “Equipo” tiene una multiplicidad de “uno” a “n” con la clase “Deportistas” ya que un equipo puede tener varios deportistas participantes. Además, la clase “Equipo” tiene una multiplicidad de “uno” a “uno” con la clase “Coach” de modo que cada equipo tiene un solo *coach*. La clase “Cantón” y la clase “Equipo” tienen una multiplicidad de “uno” a “n” por lo que un cantón puede tener varios equipos, al igual la clase “Posiciones” tiene una multiplicidad de “uno” a “n” con la clase “Equipo” ya que una tabla de posiciones puede tener varios equipos.

Por otro lado, la clase “Usuarios” tiene una multiplicidad de “uno” a “uno” con las clases “Agenda” y “Reservaciones” puesto que un usuario podrá gestionar su agenda personal y las reservaciones de los escenarios deportivos para eventos privados. La clase “Agenda” posee una multiplicidad de “uno” a “n” con la clase “Evento” ya que un usuario puede agregar a su agenda varios eventos que sean de su interés.

Conjuntamente, se tiene la clase “Eventos” que tiene una multiplicidad de “uno” a “n” con las clases “Equipo”, debido a que un evento puede tener varios equipos, también tiene una multiplicidad con la clase “Escenario” de “uno” a “uno” porque cada evento se desarrolla en un escenario. La clase “Reservaciones” muestra una multiplicidad de “uno” a “uno” con la clase “Escenario” indicando que por cada reservación puede asociarse solo un escenario. Por último, se tiene la clase “Torneo” la cual tiene una multiplicidad de “uno” a “n” con las clases “Medallero”, “Evento”, “Disciplina” esto denota que un torneo puede abarcar varios eventos con varias disciplinas y con su medallero correspondiente.

Como ejemplo, en la Tabla 2.2 se indica el detalle de la clase “Torneo”. El detalle de las clases restantes se encuentra en el **ANEXO C**.

Tabla 2.2. Detalle de la clase “Torneo”

Torneo		
Atributo	Tipo de dato	Descripción
id	int	Identificador único del torneo
Name	string	Nombre del torneo

Description	String	Información breve del torneo
date_start	date	Fecha de inicio del torneo
date_end	date	Fecha de fin del torneo

2.2.4. DISEÑO DE LA CAPA DE PRESENTACIÓN

Esta capa, también conocida como la interfaz gráfica es la encargada de la interacción con el usuario final, para lo cual debe ser amigable. Esta capa presenta el sistema, entrega y captura información al usuario. El diseño de los bosquejos (*sketch*) para esta capa se los realiza con una herramienta en línea con el nombre de Ninjamock [24]. Es importante mencionar que la aplicación está distribuida por módulos para cubrir los requerimientos de la Liga Cantonal, por lo que las interfaces diseñadas son las principales de cada módulo.

A continuación, se presenta los *sketch* principales realizados para los módulos detallados anteriormente.

En la Figura 2.17 se muestra el diseño para el inicio de sesión, el cual es necesario para utilizar los módulos del sistema. El Usuario ingresa un nombre de usuario y contraseña para el inicio de sesión. Si el Usuario es nuevo, debe registrarse primero para poder realizar el proceso, en caso de ser un usuario con el perfil de Dirigente debe ser previamente creado por el Administrador.

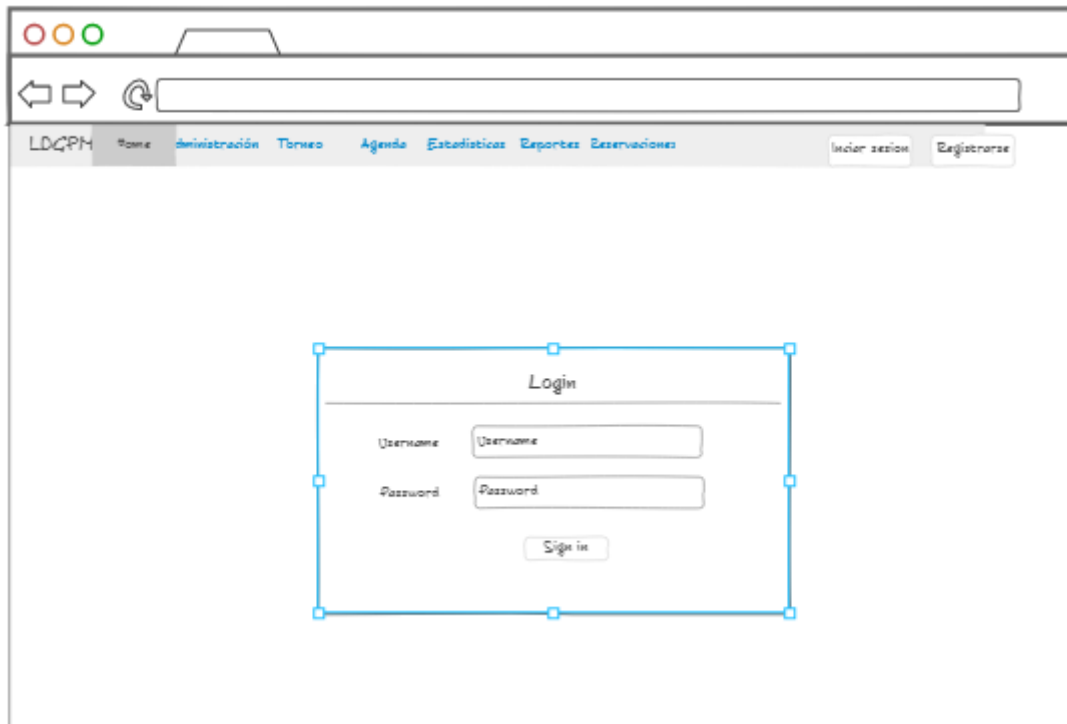


Figura 2.17 Interfaz para el inicio de sesión

La Figura 2.18 presenta el sketch para el módulo Torneo, es el encargado de la gestión de los torneos y los eventos deportivos. Esta interfaz presenta una barra de navegación con todos los módulos, en la opción de Torneo se despliega una lista la cual permite gestionar torneos o eventos. En la interfaz Torneos tenemos una tabla en la que se detallan todos los torneos creados, además tiene un botón que permite la creación de nuevos torneos.

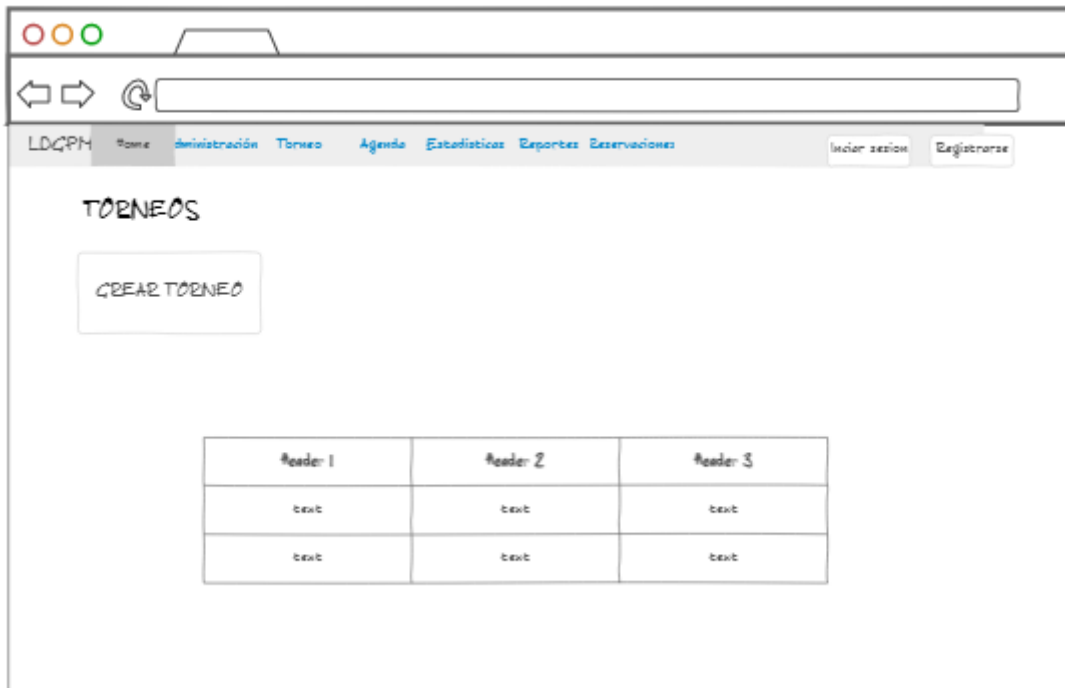


Figura 2.18 Interfaz para el módulo Torneo.

En la Figura 2.19 se observa el sketch diseñado para la interfaz principal del módulo agenda. En este módulo observamos la lista de eventos de interés que el usuario ha seleccionado, el usuario puede eliminar de la lista los eventos que ya no sean de su interés.

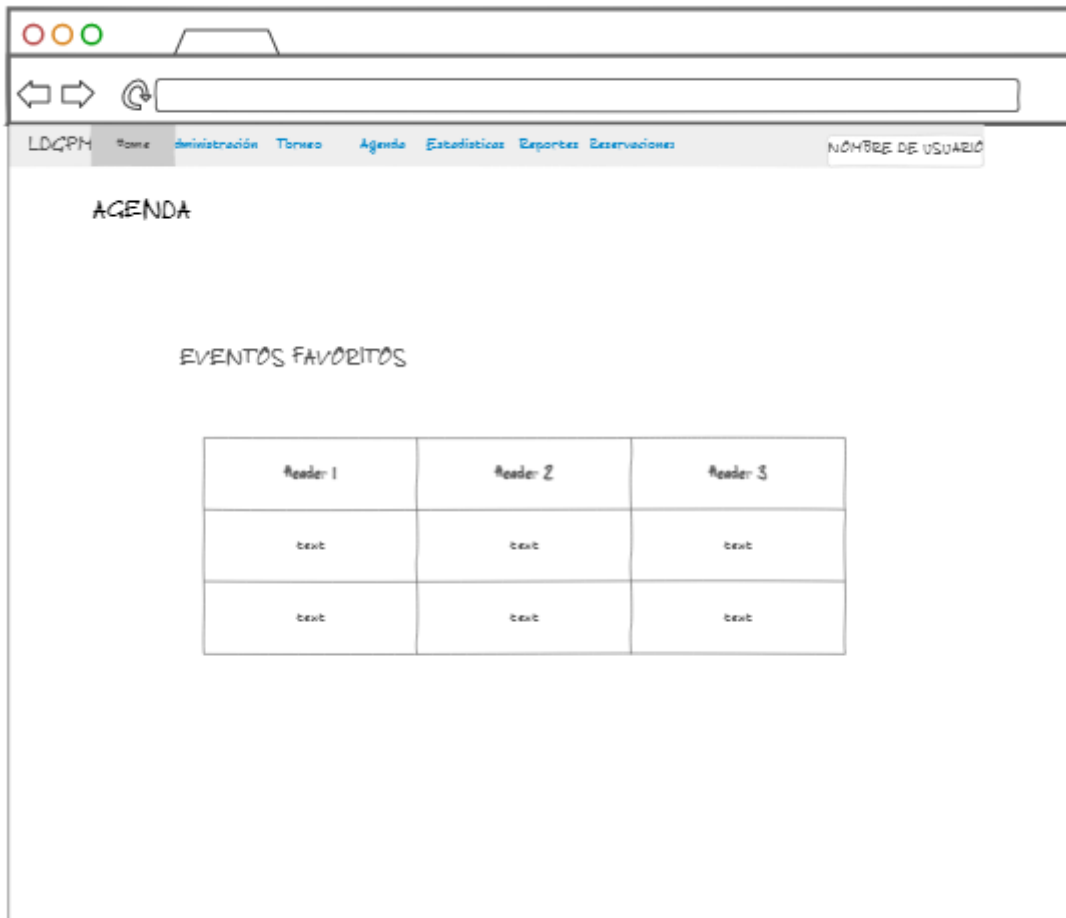


Figura 2.19 Interfaz módulo Agenda

La Figura 2.20 presenta la interfaz del módulo Estadísticas, en el cual el usuario puede acceder a los medalleros y tablas de posiciones de cada uno de los torneos. El sketch cuenta con la barra de navegación de todos los módulos al igual que con los sketches anteriores. En la página principal se encuentra un listado de acceso a los medalleros y a las tablas de posiciones.



Figura 2.20 Interfaz del módulo Estadísticas

La Figura 2.21 presenta el bosquejo de la interfaz del módulo de Reportes, como se observa, este sketch mantiene la base de los anteriores y muestra una lista de los reportes disponibles, los cuales realizan una redirección para su visualización.

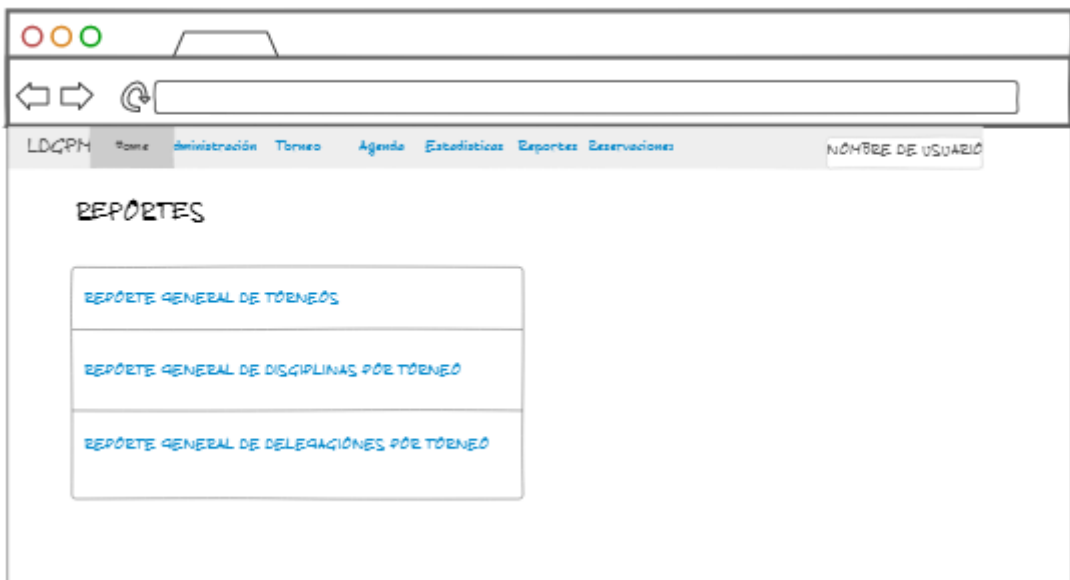


Figura 2.21 Interfaz del módulo Reportes

El bosquejo de la interfaz del módulo Reservaciones se muestra en la Figura 2.22. Este módulo al igual que los anteriores mantiene su base. Permite al Usuario ver el historial de

reservaciones que posee, y gestionar una nueva reserva por medio del botón crear reserva.

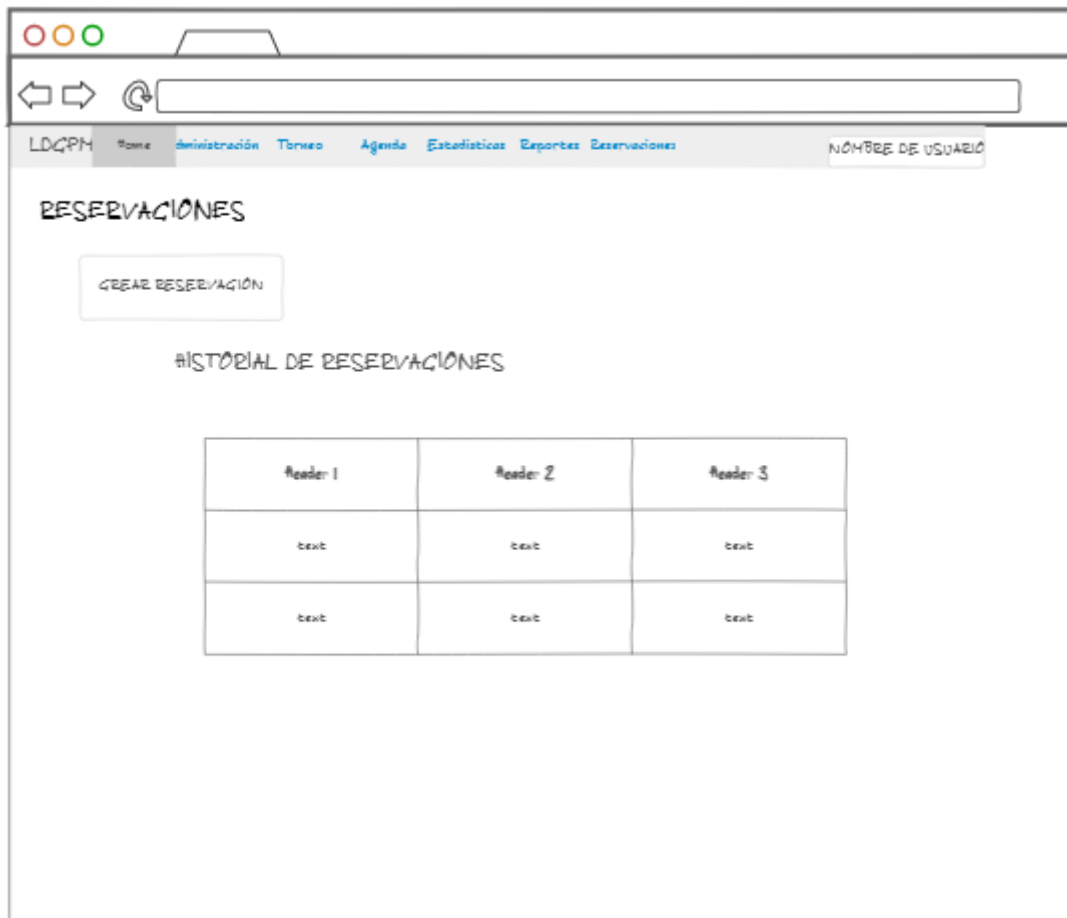


Figura 2.22 Interfaz para el módulo Reservaciones.

En la Figura 2.23 se presenta el bosquejo de la interfaz para el módulo de Administración. Ese módulo estará disponible solamente para el usuario Administrador. Para acceder a este módulo, en la barra de navegación se encuentra una lista desplegable con las opciones para administrar que son: Usuarios, Disciplinas, Cantones, Categorías. El sketch mantiene la base de los anteriores y en cada opción cuenta con una tabla donde está el detalle de cada uno de los usuarios, disciplinas, cantones y categorías registradas. También, tiene un botón en el cual puedes crear un nuevo ítem de las opciones antes mencionadas.



Figura 2.23 Interfaz para el módulo de Administración.

2.3. IMPLEMENTACIÓN

En esta fase del proyecto, se actualiza el Tablero Kanban con las tareas a cumplirse durante esta etapa. Se prepara el ambiente de desarrollo necesario, es decir, instalando las herramientas que se ajusten a la necesidad del proyecto. En el proceso de codificación se inicia con la Capa de Datos, seguido de la Capa de Negocio y finalmente la Capa Presentación, correspondiente al diseño realizado.

2.3.1. ACTUALIZACIÓN DEL TABLERO KANBAN

En la Figura 2.24 se muestra el avance del proyecto ya con las tareas actualizadas en el tablero. Al término de las tareas correspondientes a la etapa de Diseño, estas son trasladadas a la columna de "Hecho". Mientras que, en la columna "En proceso" es actualizada con las tareas de la fase de implementación. Finalmente, la columna "Lista de Tareas" muestra aquellas tareas que serán desarrolladas en la fase de pruebas.

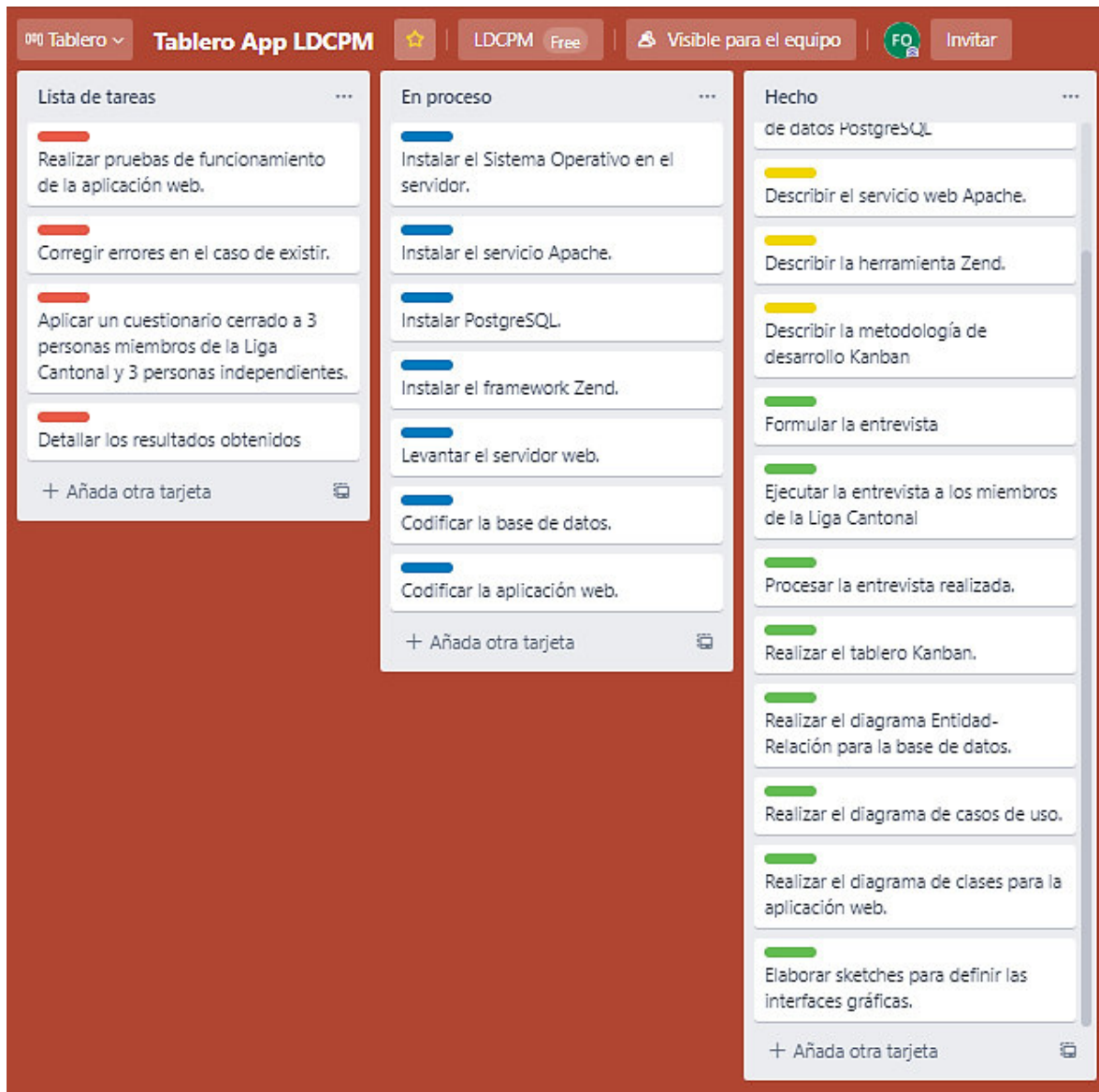


Figura 2.24 Tablero Kanban de la Fase de Implementación

2.3.2. INSTALACIÓN DE LAS HERRAMIENTAS NECESARIAS

En este apartado, se describe la instalación de las herramientas necesarias para el desarrollo del prototipo de aplicación web. Además de la instalación de los complementos necesarios del servidor para el correcto funcionamiento del proyecto.

2.3.2.1. Instalación del Sistema Operativo CentOS 7

Para preparar el servidor que alojará el prototipo de aplicación web, se instala el sistema operativo CentOS 7, el cual carece de interfaz gráfica para optimizar los recursos del servidor.

En la Figura 2.25 se muestra la pantalla inicial del proceso de instalación. Para esta instalación se preparó un “Kickstart”¹⁰ con las configuraciones iniciales y así lograr que el proceso de instalación sea más rápido.

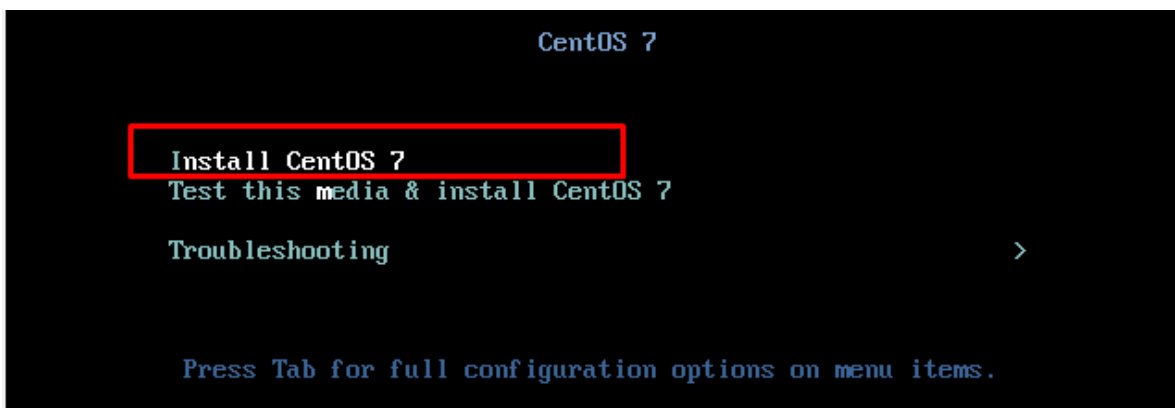


Figura 2.25 Pantalla inicial de instalación

Para poder ejecutar el *Kickstart*, se debe acceder a la configuración presionando la tecla TAB. En la Figura 2.26 se indica como debe estar la configuración para que el Kickstart se ejecute. Una vez realizada esta configuración se procede a presionar la combinación de teclas CTRL+ x para iniciar el proceso de instalación del sistema operativo.



Figura 2.26 Configuración para el Kickstart

El paso siguiente es la creación de las particiones del disco. En la Figura 2.27 muestra la opción que se debe escoger para el destino de la instalación la cual nos direcciona a la ventana donde se selecciona el disco para la instalación y la opción para configurar las particiones, la cual se muestra en le Figura 2.28. Una vez seleccionado lo antes mencionado se presiona en el botón “Done” como indica la Figura 2.29.

¹⁰ Kickstart: sirve para hacer una instalación desatendida y automática del sistema operativo CentOS

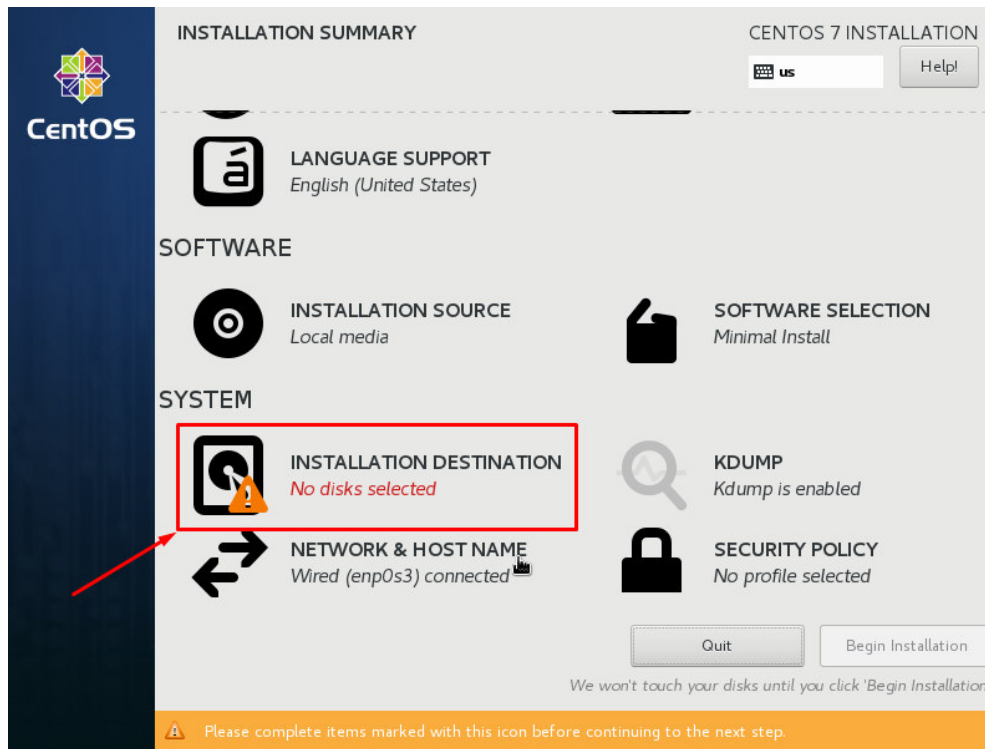


Figura 2.17 Disco destino de instalación

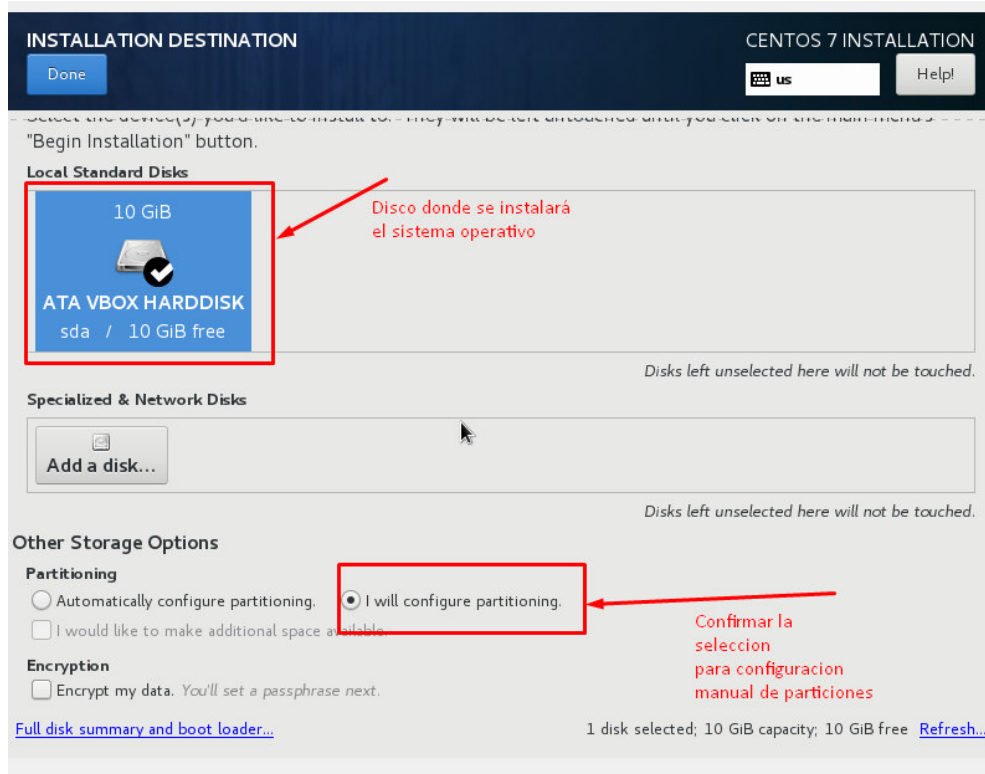


Figura 2.28 Disco y modo de particionamiento.



Figura 2.29 Botón Done.

Continuando con el proceso, en la Figura 2.30 muestra la opción que se debe escoger para realizar una partición automática del disco. Sin embargo, es necesario eliminar la partición “/home” y expandir la partición “/root” como se muestra en la Figura 2.31. Una vez terminada esta configuración, se presiona el botón Done.

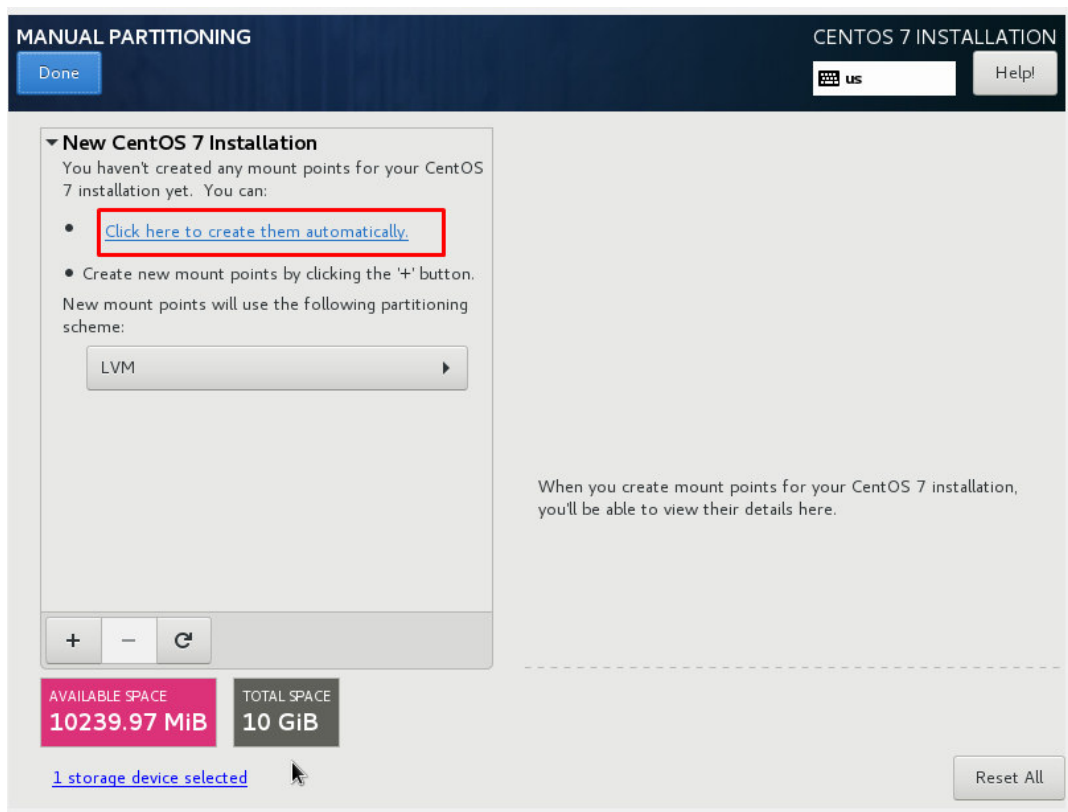


Figura 2.30 Partición automática.

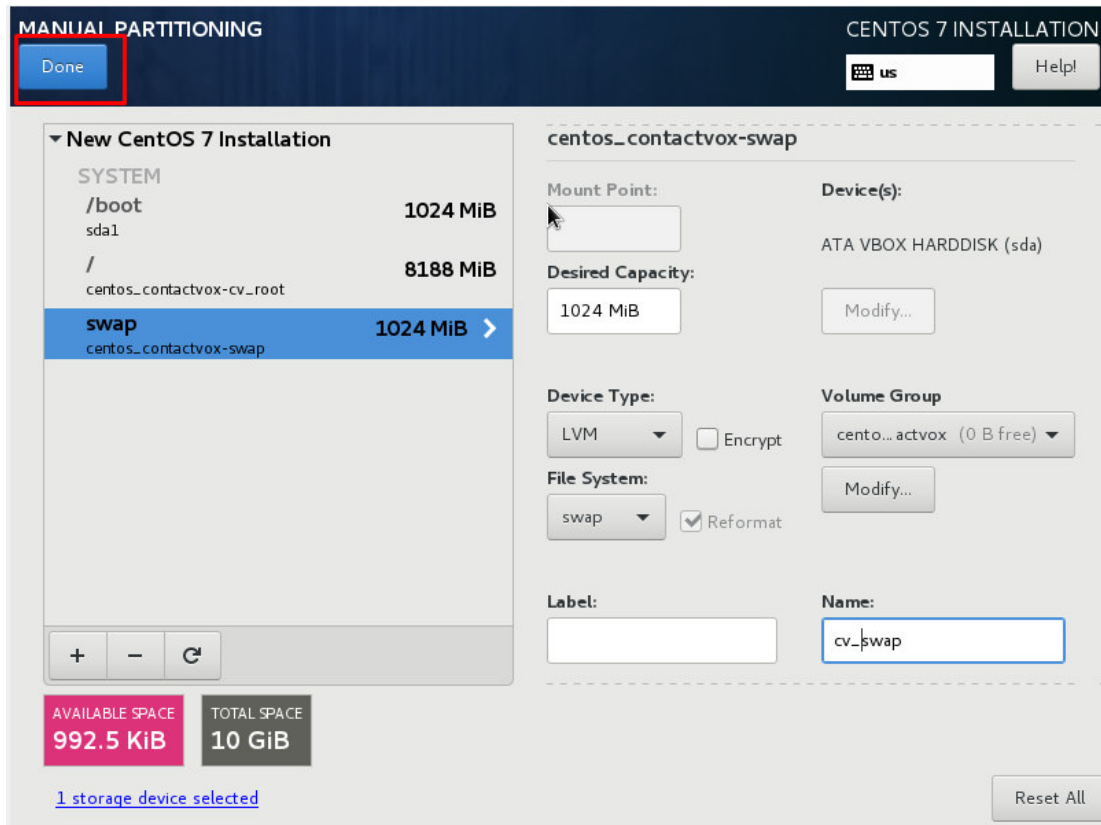


Figura 2.31 Partición final.

Finalmente, presionar “iniciar instalación” como se muestra en la Figura 2.32 y esperar mientras se cumple el proceso.

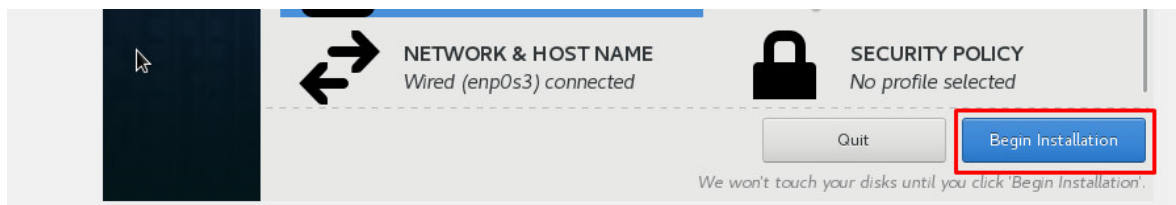


Figura 2.32 Iniciar Instalación.

Terminada la instalación del sistema operativo, el servidor se reinicia de manera automática. Después del reinicio, se muestra un terminal con el *login* correspondiente para ingresar al servidor, como se indica en la Figura 2.33.

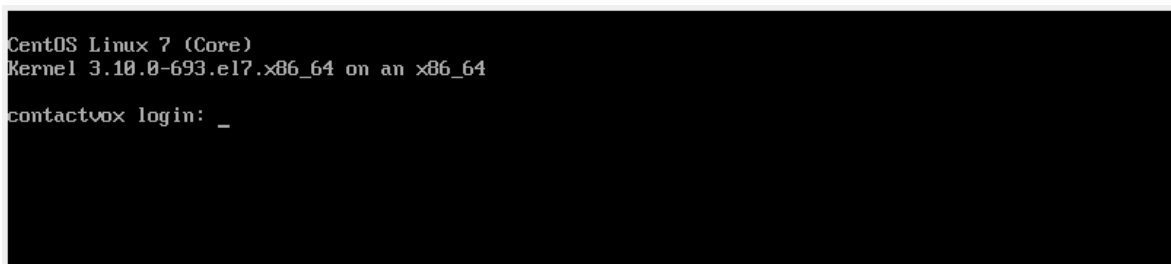


Figura 2.33 Terminal del servidor CentOS.

2.3.2.2. Instalación del servicio web Apache

Para instalar el servicio web, se debe abrir un terminal y ejecutar el comando indicado en el Código 2.1. Para instalar paquetes o servicio en el sistema operativo CentOS se usa el comando “yum -y install”.

```
[root@contactvox ~]# yum -y install httpd
```

Código 2.1 Comando para instalar Apache.

2.3.2.3. Instalación de PostgreSQL 9.6

En el Código 2.2 se indica el comando utilizado para la instalación de PostgreSQL-9.6 y todos sus componentes, estos componentes permiten la integración con varios sistemas y servicios.

```
# yum -y install postgresql96-server postgresql96-plperl postgresql96-contrib postgresql96-odbc postgresql96-devel
```

Código 2.2 Comando para instalar PostgreSQL-9.6

Una vez instalado todos los paquetes se procede a inicializar la base de datos ejecutando el comando mostrado en el Código 2.3.

```
# /usr/pgsql-9.6/bin/postgresql96-setup initdb
```

Código 2.3 Comando para inicializar PostgreSQL

En la Figura 2.34 se muestra la configuración de la autenticación para el ingreso a la base de datos, puede ser “trust” o “md5”. El archivo para modificar es “pg_hba.conf”.

```
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all trust
# IPv4 local connections:
host all all 127.0.0.1/32 trust
host all all 192.168.0.0/16 trust
```

Figura 2.34 Configuración de autenticación de PostgreSQL.

Continuando con la configuración, se debe editar el archivo “visudo”. El comando utilizado para editar este archivo se muestra en el Código 2.4. En el archivo visudo se debe agregar la línea que se indica en la Figura 2.35, la cual permite el ingreso a la base de datos con el usuario “postgres”.

```
# visudo
```

Código 2.4 Comando para acceder al archivo visudo.

```
###  
## Allow root to run any commands anywhere  
root    ALL=(ALL)        ALL  
postgres    ALL=(ALL)        NOPASSWD:ALL  
## Allows members of the 'sys' group to run networking, software,  
## service management apps and more.  
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES,  
  
## Allows people in group wheel to run all commands  
%wheel  ALL=(ALL)        ALL
```

Figura 2.35 Configuración del archivo visudo.

Para finalizar la instalación y configuración de PostgreSQL, copiar los archivos “odbc” al directorio lib64 usando el comando indicado en el Código 2.5 e iniciar el servicio con el comando mostrado en el Código 2.6.

```
# cp /usr/pgsql-9.6/lib/psqlodbc* /usr/lib64/
```

Código 2.5 Comando para copiar los archivos odbc.

```
[root@contactvox ~]# service postgresql-9.6 start  
Redirecting to /bin/systemctl start postgresql-9.6.service
```

Código 2.6 Comando para iniciar el servicio de PostgreSQL.

2.3.2.4. Instalación de Zend Framework

El proceso de instalación de “Zend *Framework*” comienza con la implementación de paquetes específicos requeridos por este *framework*. Primero, se instala todos los paquetes PHP indicados en la documentación de Zend. Los comandos usados para esta instalación se indica en el Código 2.7.

```
[root@contactvox ~]# yum install -y php php-pdo php-mbstring php-pgsql php-xml php-common php-opcache  
php-gd php-curl php-intl php-git php-cli
```

Código 2.7 Comandos para instalar paquetes PHP.

Otro de los complementos necesarios para este *framework* es el “composer¹¹”. En el Código 2.8 se muestran los comandos para la instalación.

```
1 curl -sS https://getcomposer.org/installer | php
2 mv composer.phar /usr/bin/composer
3 chmod +x /usr/bin/composer
```

Código 2.8 Comandos de instalación del composer.

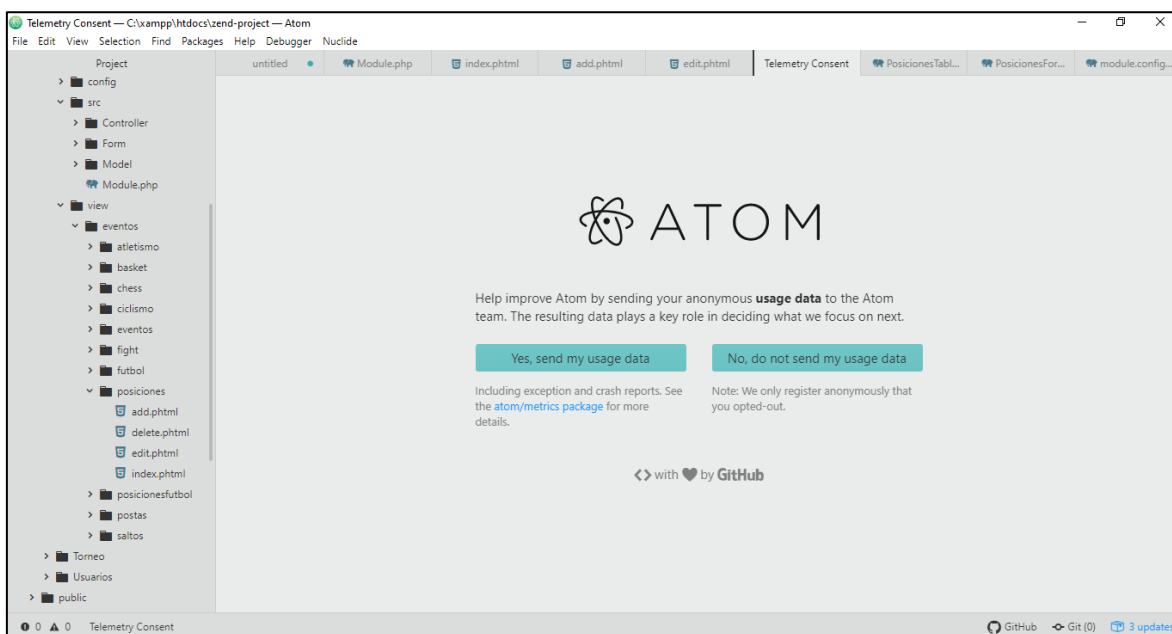
Para culminar con la instalación de Zend Framework, se debe crear el proyecto con los comandos que se indican en el Código 2.9.

```
1 cd /var/www/html
2 composer create-project zendframework/skeleton-application zend-project
3 cd zend-project/public
```

Código 2.9 Comandos para crear el proyecto.

2.3.2.5. Instalación de Atom

Atom es un editor de código fuente de código abierto y está disponible para plataformas Windows, Linux, MacOS. Se debe descargar el instalador desde la página oficial de Atom [25] de acuerdo con el sistema operativo que se requiere. Se ejecuta el instalador para iniciar el asistente de instalación. Terminada la instalación se muestra la página de bienvenida como indica la Figura 2.36.



¹¹ Composer: Es un sistema de gestión de paquetes para programar en PHP.

Figura 2.36 Página de Bienvenida de Atom.

2.3.3. IMPLEMENTACIÓN DE LA CAPA DE DATOS

En esta sección, se presenta un ejemplo de la codificación de la Capa de Datos basado en la etapa de Diseño. Durante la codificación de las capas de Datos, Lógica de Negocio y Presentación, en su gran mayoría se utiliza el idioma inglés, tratando de cumplir con las buenas prácticas de programación y así evitar los caracteres especiales como vocales acentuadas y la ñ que pueden provocar que, al momento de almacenar los datos, estos sufran alteraciones o a su vez en el momento de ejecución existan fallos.

En el código 2.10 se muestra la creación de la base de datos llamada `ldcpm_db` donde se almacena toda la información del sistema. En la siguiente línea, se establece la conexión a la base de datos a usarse.

```
postgres=# create database ldcpm_db;
CREATE DATABASE
postgres=# \c ldcpm_db
You are now connected to database "ldcpm_db" as user "postgres".
ldcpm_db=#
```

Código 2.10 Creación de la base de datos `ldcpm_db`.

El Código 2.11 muestra la creación del esquema *private* y establece la contraseña para el usuario `postgres`.

```
1 DROP SCHEMA IF EXISTS private CASCADE;
2 CREATE SCHEMA private;
3 ALTER ROLE postgres WITH PASSWORD 'P@r@n0ld';
```

Código 2.11 Creación del esquema *private* y contraseña del usuario `postgres`.

El Código 2.12 corresponde a la creación de la tabla "Deportistas". En la línea 1 elimina la tabla en caso de que ya exista. En la línea 2, se establece el nombre de la tabla y el número de columnas que tendrá de acuerdo con la cantidad de atributos especificados entre la línea 3 y 12. En la línea 3 se define la columna "id" como llave primaria y es del tipo serial y no puede ser nula, al igual que el resto de los campos ya que poseen el tipo NOT NULL.

```

1 DROP TABLE IF EXISTS private.deportistas CASCADE;
2 CREATE TABLE private.deportistas (
3   "id" serial PRIMARY KEY,
4   "_canton" INT NOT NULL REFERENCES private.canton ON DELETE RESTRICT,
5   "firstname" TEXT NOT NULL,
6   "lastname" TEXT Not NULL,
7   "date" DATE NOT NULL,
8   "cedula" TEXT NOT NULL,
9   "sexo" private.disc_sex Not NULL,
10  "_distc" INT NOT NULL REFERENCES private.disc_tournament_canton ON DELETE RESTRICT,
11  "status" BOOLEAN NOT NULL DEFAULT FALSE, /*Para cuando se habilitan*/
12  "created" TIMESTAMP NOT NULL DEFAULT now()
13 );

```

Código 2.12 Creación de la tabla Deportistas.

El código restante se encuentra en el **ANEXO D**.

2.3.4. IMPLEMENTACIÓN DE LA CAPA LOGICA DE NEGOCIO

En este apartado, se presenta la implementación de las clases y el servicio web.

Zend *framework* permite crear aplicaciones impulsadas por una base de datos utilizando el paradigma Modelo-Vista-Controlador. Zend utiliza un sistema de módulos para organizar el código específico de la aplicación dentro de cada módulo. [13]

En la Figura 2.37 se muestra el módulo Torneo donde se despliegan tres directorios que se detallan a continuación:

- *Config*: Este directorio contiene la configuración de las rutas para acceder a los recursos del servicio web.
- *Src*: Este directorio contiene otros directorios con los modelos y controladores que permitirán ejecutar las acciones y la interacción con la base de datos.
- *View*: Este directorio contiene los archivos para la interfaz gráfica.

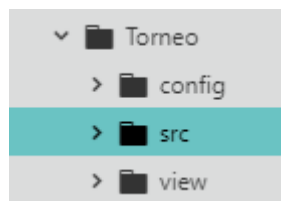


Figura 2.37 Módulo Torneo.

2.3.4.1. Codificación de Clases

Como se mencionó anteriormente Zend Framework organiza el código por directorios, en el directorio “src” se encuentra otro directorio llamado “Model” el cual tiene archivos con la

clase y los métodos de interacción con la base de datos. Estos archivos se pueden observar en la Figura 2.38

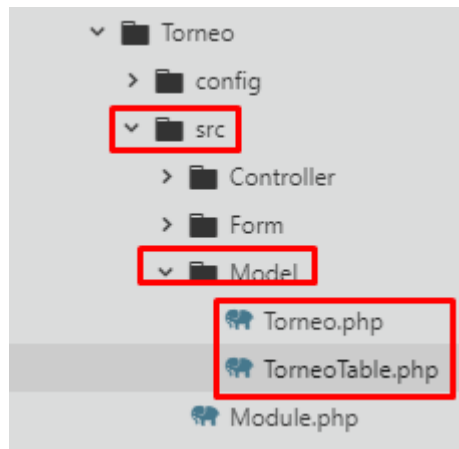


Figura 2.38 Directorio Model

El archivo Torneo.php es una clase del tipo PHP. En el código 2.13, la línea 14 define el nombre de la clase que implementa una interfaz que permite validar los datos recibidos. Entre la línea 16 y 20 se definen los atributos que tiene esta clase. En la línea 24, se crea el método `exchangeArray()` que es el encargado de copiar los datos de la matriz proporcionada a las propiedades de nuestra clase.

```
14 class Torneo implements InputFilterAwareInterface
15 {
16     public $id;
17     public $name;
18     public $description;
19     public $date_start;
20     public $date_end;
21
22     private $inputFilter;
23
24     public function exchangeArray(array $data)
25     {
26         $this->id = isset($data['id']) ? $data['id'] : null;
27         $this->name = isset($data['name']) ? $data['name'] : null;
28         $this->description = isset($data['description']) ? $data['description'] : null;
29         $this->date_start = isset($data['date_start']) ? $data['date_start'] : null;
30         $this->date_end = isset($data['date_end']) ? $data['date_end'] : null;
31     }
```

Código 2.13 Clase Torneo.

En el Código 2.14 se muestra un método que permite la validación de los datos recibidos.


```

40 public function getInputFilter()
41 {
42     if ($this->inputFilter) {
43         return $this->inputFilter;
44     }
45
46     $inputFilter = new InputFilter();
47
48     $inputFilter->add([
49         'name' => 'id',
50         'required' => true,
51         'filters' => [
52             ['name' => ToInt::class],
53         ],
54     ]);
55
56     $inputFilter->add([
57         'name' => 'name',
58         'required' => true,
59         'filters' => [
60             ['name' => StripTags::class],
61             ['name' => StringTrim::class],
62         ],
63         'validators' => [
64             [
65                 'name' => StringLength::class,
66                 'options' => [
67                     'encoding' => 'UTF-8',
68                     'min' => 1,
69                     'max' => 100,
70                 ],
71             ],
72         ],
73     ]);

```

Código 2.14 Método para validar datos.

El archivo “TorneoTable.php” se encuentran los métodos estándar que se encargan de gestionar el acceso a las tablas, usando la librería Zend-db. Los métodos principales son: fetchAll(), save(), update(), get(), delete().

El Código 2.15 presenta los métodos fetchAll() el cual realiza un consulta general a la base de datos y el método getTorneo() realiza una consulta específica en base al id a la base de datos y retorna el valor requerido.

```

public function fetchAll()
{
    return $this->tableGateway->select();
}

public function getTorneo($id)
{
    $id = (int) $id;
    $rowset = $this->tableGateway->select(['id' => $id]);
    $row = $rowset->current();
    if (!$row) {
        throw new RuntimeException(sprintf(
            'Could not find row with identifier %d',
            $id
        ));
    }

    return $row;
}

```

Código 2.15 Métodos fetchAll() y getTorneo()

Los métodos saveTorneo() y deleteTorneo(), son mostrados en el código 2.16. El primer método recibe los datos del torneo, valida si el id recibido es diferente de cero e inserta en la base de datos, mientras que, el segundo método recibe el id del torneo y elimina el registro de la base de datos.

```

36 public function saveTorneo(Torneo $torneo)
37 {
38
39     $data = [
40         // 'id' => $torneo->id,
41         'name' => $torneo->name,
42         'description' => $torneo->description,
43         'date_start' => $torneo->date_start,
44         'date_end' => $torneo->date_end,
45     ];
46
47     $sid = (int) $torneo->id;
48
49
50     if ($sid === 0) {
51         $this->tableGateway->insert($data);
52         return;
53     }
54
55     try {
56         $this->getTorneo($sid);
57     } catch (RuntimeException $e) {
58         throw new RuntimeException(sprintf(
59             'Cannot update torneo with identifier %d; does not exist',
60             $sid
61         ));
62     }
63
64     $this->tableGateway->update($data, ['id' => $sid]);
65 }
66
67 public function deleteTorneo($id)
68 {
69     $this->tableGateway->delete(['id' => (int) $id]);
70 }
71 }

```

Código 2.16 Métodos saveTorneo() y deleteTorneo()

Cabe mencionar que, a modo de ejemplo se detalla el módulo Torneo, los módulos restantes se encuentran en el **ANEXO E**.

2.3.4.2. Codificación del Servicio Web

Para codificar el servicio web, primero levantaremos el servidor Apache. Con el comando mostrado en el Código 2.17 accederemos al archivo de configuración de Apache. En la Figura 2.39 se indica la configuración que debe tener el archivo antes mencionado, se puede observar que el DocumentRoot y el Directory deben tener la misma ruta a donde apunta.

```
# vim /etc/httpd/conf/httpd.conf
```

Código 2.17 Comando para acceder al archivo de configuración de Apache

```
#
DocumentRoot "/var/www/html/zend-project/public"
#
# Relax access to content within /var/www.
#
# Further relax access to the default document root:
<Directory "/var/www/html/zend-project/public">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
#   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.4/mod/core.html#options
# for more information.
#
Options Indexes FollowSymLinks

#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   Options FileInfo AuthConfig Limit
#
AllowOverride All

#
# Controls who can get stuff from this server.
#
Require all granted
</Directory>
```

Figura 2.39 Archivo de configuración de Apache.

Finalmente, se inicia el servicio de Apache con el comando mostrado en el Código 2.18.

```
# service httpd start
```

Código 2.18 Comando para iniciar Apache.

Continuando con la codificación de la capa Lógica de Negocio, se debe configurar la conexión a la base de datos. Zend Framework tiene dos archivos de configuración donde se coloca esta conexión. En el Código 2.19 se indica la conexión a la base de datos contenida en el archivo "global.php". En el Código 2.20 se especifican las credenciales de acceso a la base de datos que se colocan en el archivo "local.php".

```

16 // ...
17 'db' => array (
18     'driver' => 'Pdo',
19     'dsn' => 'pgsql:dbname=ldcpm_prueba; host=192.168.100.250; port=5432',
20     'driver_options' => array (
21     ),
22 ),

```

Código 2.19 Conexión a la base de datos

```

14 return [
15     'db' => array(
16         'username' => 'postgres',
17         'password' => 'P@r@n01d',
18     ),
19 ];

```

Código 2.20 Credenciales de acceso a la base de datos

A continuación, se presenta el controlador del módulo de Torneo en el cual se especifican métodos asociados a las acciones que el usuario puede realizar en la aplicación web. El Código 2.21 presenta los métodos `indexAction()`, este método permite almacenar en una variable los registros obtenidos de la base de datos. Y el método `addAction()`, que obtiene los datos ingresados por el usuario y los guardar en la base de datos como un nuevo registro. En la línea 35 se valida que sea una petición del tipo `post`.

```

21     public function indexAction()
22     {
23         return new ViewModel([
24             'torneos' => $this->table->fetchAll(),
25         ]);
26     }
27
28     public function addAction()
29     {
30         $form = new \Torneo\Form\TorneoForm();
31         $form->get('submit')->setValue('Add');
32
33         $request = $this->getRequest();
34
35         if (!$request->isPost()) {
36             return ['form' => $form];
37         }
38
39         $torneo = new \Torneo\Model\Torneo();
40         $form->setInputFilter($torneo->getInputFilter());
41         $form->setData($request->getPost());
42
43         if (!$form->isValid()) { //\Zend\Debug\Debug::dump($form->getData());
44             return ['form' => $form];
45         }
46         $torneo->exchangeArray($form->getData());
47         $this->table->saveTorneo($torneo);
48         return $this->redirect()->toRoute('torneo');
49     }

```

Código 2.21 Métodos indexAccion() y addAction().

El código 2.22 presenta el método editAccion() que es el encargado de actualizar los registros específicos de acuerdo con el id. Entre la línea 53 a la 61 se recupera el id del registro, con este id se consulta en la base de datos y se obtiene el registro. Entre la línea 63 y 68 muestran los valores obtenidos en el formulario para su correspondiente edición o actualización, en las líneas restantes se realiza el proceso del guardado del registro actualizado.

```

51 public function editAction()
52 {
53     $id = (int) $this->params()->fromRoute('id', 0);
54     if (0 === $id) {
55         return $this->redirect()->toRoute('torneo', ['action' => 'add']);
56     }
57     try {
58         $torneo = $this->table->getTorneo($id);
59     } catch (\Exception $e) {
60         return $this->redirect()->toRoute('torneo', ['action' => 'index']);
61     }
62
63     $form = new \Torneo\Form\TorneoForm();
64     $form->bind($torneo);
65     $form->get('submit')->setAttribute('value', 'Edit');
66
67     $request = $this->getRequest();
68     $viewData = ['id' => $id, 'form' => $form];
69
70     if (!$request->isPost()) {
71         return $viewData;
72     }
73     $form->setInputFilter($torneo->getInputFilter());
74     $form->setData($request->getPost());
75
76     if (!$form->isValid()) {
77         return $viewData;
78     }
79
80     $this->table->saveTorneo($torneo);
81
82     return $this->redirect()->toRoute('torneo');
83 }

```

Código 2.22 Método *editAction()*.

Finalmente, se define el método `deleteAccion()`, que permite la eliminación de cualquier registro. En el Código 2.23 entre la línea 87 y 90 se recupera el id del registro que se desea eliminar. En la línea 94 se pregunta al usuario si desea eliminar el registro, si el resultado es si, se procede a eliminar de la base de datos con el método `delete()` detallado en la línea 98. Una vez eliminado el registro, el método redirecciona el usuario a la interfaz principal del módulo Torneo.

```

85     public function deleteAction()
86     {
87         $id = (int) $this->params()->fromRoute('id', 0);
88         if (!$id) {
89             return $this->redirect()->toRoute('torneo');
90         }
91
92         $request = $this->getRequest();
93         if ($request->isPost()) {
94             $del = $request->getPost('del', 'No');
95
96             if ($del == 'Yes') {
97                 $id = (int) $request->getPost('id');
98                 $this->table->deleteTorneo($id);
99             }
100
101             // Redirect to List of albums
102             return $this->redirect()->toRoute('torneo');
103         }
104
105         return [
106             'id' => $id,
107             'torneo' => $this->table->getTorneo($id),
108         ];
109     }
110 }

```

Código 2.23 Método deleteAccion()

Cabe recalcar que los métodos antes detallados se replican en todos los módulos de la aplicación web y se encuentran en el **ANEXO E**.

Al concluir la definición de las acciones que realiza el controlador, se debe configurar los servicios tanto para el controlador como para la base de datos. En el Código 2.24 se detalla el método getServiceConfig(), que permite enlazar el modelo con la base de datos y definir la tabla a la que se accederá a realizar el CRUD, es decir, configura el servicio de acceso a la base de datos. El Código 2.25 muestra el método getControllerConfig() en donde se define el controlador para el módulo Torneo.


```

17 public function getServiceConfig()
18 {
19     return [
20         'factories' => [
21             Model\TorneoTable::class => function($container) {
22                 $tableGateway = $container->get(Model\TorneoTableGateway::class);
23                 return new Model\TorneoTable($tableGateway);
24             },
25             Model\TorneoTableGateway::class => function ($container) {
26                 $dbAdapter = $container->get(AdapterInterface::class);
27                 $resultSetPrototype = new ResultSet();
28                 $resultSetPrototype->setArrayObjectPrototype(new Model\Torneo());
29                 return new TableGateway('tournament', $dbAdapter, null, $resultSetPrototype);
30             },
31         ],
32     ];
33 }

```

Código 2.24 Método getServiceConfig()

```

34 public function getControllerConfig()
35 {
36     return [
37         'factories' => [
38             Controller\TorneoController::class => function($container) {
39                 return new Controller\TorneoController(
40                     $container->get(Model\TorneoTable::class)
41                 );
42             },
43         ],
44     ];
45 }

```

Código 2.25 Método getControllerConfig()

En el directorio “config” del módulo Torneo se encuentra el archivo module.config.php en el que se definen las rutas que tendrá el módulo. En el código 2.26 se muestra el nombre de la ruta, es “torneo” y es de tipo “segmento”. La ruta del segmento nos permite especificar marcadores de posición en el patrón de URL, que se asignarán a los parámetros con nombre en la ruta coincidente. En este caso, la ruta /torneo[/:action[/:id]] coincidirá con cualquier URL que comience con /torneo. El siguiente segmento será un nombre de acción opcional y, finalmente, el siguiente segmento se asignará a una identificación opcional. Los corchetes indican que un segmento es opcional. La sección de restricciones nos permite asegurarnos de que los caracteres dentro de un segmento sean los esperados, por lo que tenemos acciones limitadas para comenzar con una letra y luego los caracteres posteriores solo son alfanuméricos, guiones bajos o guiones. También limitamos la identificación a dígitos.

```

8 return [
9
10     'router' => [
11         'routes' => [
12             'torneo' => [
13                 'type' => Segment::class,
14                 'options' => [
15                     'route' => '/torneo[:action][:id]]',
16                     'constraints' => [
17                         'action' => '[a-zA-Z][a-zA-Z0-9_-]*',
18                         'id' => '[0-9]+',
19                     ],
20                 ],
21             ],
22         ],
23     ],
24 ];

```

Código 2.26 Configuración de rutas del módulo Torneo.

2.3.5. IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN

Zend Framework para la Capa Presentación tiene la librería Zend-form que está diseñada como un puente entre sus modelos y la capa de vista. Comprende una capa delgada de objetos que representan elementos de un formulario, así como métodos para vincular datos desde y hacia el mismo [13]. Además, para las vistas Zend utiliza archivos phtml¹², estilos Bootstrap¹³, los cuales combina lenguajes CSS y JavaScript para dar interactividad a las páginas.

Al iniciar la aplicación web se muestra una página PHTML, ésta representa el *layout* (plantilla) que se utiliza durante toda la aplicación. En el Código 2.27 se muestra la cabecera de este archivo donde se importan las hojas de estilo, se definen el título de la pestaña del navegador y otros datos que sirven de información para la página web. En el Código 2.28 se indica el cuerpo de la página, que consta de la barra de navegación con todas las opciones disponibles para el usuario.

¹² Phtml: archivos que permite tener lenguaje PHP y HTML en el mismo.

¹³ Bootstrap: Framework usado para dar estilo e interactividad a páginas web.

```

3 <html lang="en">
4 <head>
5 <meta charset="utf-8">
6 <?=$this->headTitle('LOCPM')->setSeparator(' - ')->setAutoEscape(false) ?>
7
8 <?=$this->headMeta()
9 >->appendName('viewport', 'width=device-width, initial-scale=1.0')
10 >->appendHttpEquiv('X-UA-Compatible', 'IE=edge')
11 ?>
12
13 <!-- Le styles -->
14 <?=$this->headLink(['rel' => 'shortcut icon', 'type' => 'image/vnd.microsoft.icon', 'href' => $this->basePath() . '/img/favicon.ico'])
15 >->prependStylesheet($this->basePath('css/style.css'))
16 >->prependStylesheet($this->basePath('css/bootstrap.min.css'))
17 ?>
18
19 <!-- Scripts -->
20 <?=$this->inlineScript()
21 >->prependFile($this->basePath('js/bootstrap.min.js'))
22 >->prependFile($this->basePath('js/jquery-3.3.1.min.js'))
23 ?>
24 <script type="text/javascript">
25 >$(document).ready(function () {
26 >    $('<span class="dropdown-toggle">').dropdown();
27 >});
28 </script>
29 </head>

```

Código 2.27 Cabecera del archivo layout.phtml

```

30 <body>
31 <nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark" role="navigation">
32 <div class="container" style="max-width: 11470px">
33 ><div class="navbar-header" style="margin-left: 45px">=
45 <div class="collapse navbar-collapse" id="navbarSupportedContent">
46 <ul class="navbar-nav mr-auto">
47 <li class="nav-item active">
48 <a class="nav-link" href="<?=$this->url('home') ?>">Home <span class="sr-only">(current)</span></a>
49 </li>
50 <li class="nav-item dropdown">
51 <a class="nav-link dropdown-toggle active" href="#" id="navbarDropdown" role="button" data-toggle="dropdown" aria-
52 >haspopup="true" aria-expanded="false" >Administración </a>
53 <div class="dropdown-menu" aria-labelledby="navbarDropdown">=
54 </li>
55 <li class="nav-item dropdown">
56 <a class="nav-link dropdown-toggle active" href="#" id="navbarDropdown" role="button" data-toggle="dropdown" aria-
57 >haspopup="true" aria-expanded="false" >Torneos </a>
58 <div class="dropdown-menu" aria-labelledby="navbarDropdown">=
59 </li>
60 <li class="nav-item">
61 <a class="nav-link" href="<?=$this->url('home') ?>">Agenda <span class="sr-only">(current)</span></a>
62 </li>
63 <li class="nav-item">
64 <a class="nav-link" href="<?=$this->url('home') ?>">Estadísticas <span class="sr-only">(current)</span></a>
65 </li>
66 <li class="nav-item">
67 <a class="nav-link" href="<?=$this->url('home') ?>">Reportes <span class="sr-only">(current)</span></a>
68 </li>
69 <li class="nav-item active">
70 <a class="nav-link" href="<?=$this->url('reserva') ?>">Reservaciones <span class="sr-only">(current)</span></a>
71 </li>
72 </ul>
73 </div>=
74 </div>
75 </nav>

```

Código 2.28 Cuerpo del archivo layout.phtml

En la Figura 2.40 se muestra la página principal que se presenta al usuario al abrir la aplicación web, el contenido de la página se encuentra codificado en el archivo `index.phtml` el cual se observa en el Código 2.29. Entre la línea 1 a la 6 se define el título de la página y la imagen, contenidas en un recuadro con color de fondo. El resto de las líneas comprenden la codificación de los recuadros informativos de Misión, Visión e Historia.



Figura 2.40 Página principal de la aplicación web

```

1 <div class="jumbotron" >
2   <h1 style="text-align: center">Bienvenido al sitio Web de la <span class="zf-blue">Liga Cantonal Pedro Moncayo</span></h1>
3
4   <h2 style="text-align: center"> </span></h2>
5
6 </div>
7
8 <div class="row">
9
10  <div class="col-md-4">
11    <div class="card">
12      <div class="card-header">
13        <h2 class="card-title">Misión</h2>
14      </div>
15      <div class="card-body">
16        <p>
17          Tiene como propósito esencial la planificación, fomento
18          organización, ejecución, coordinación y control de los
19          deportes como medio formativo, el desarrollo integral de
20          las personas en todas las edades para el cumplimiento
21          eficaz de sus obligaciones como miembros de la sociedad.
22        </p>
23      </div>
24    </div>
25  </div>
26 </div>
27
28 <div class="col-md-4">
29  <div class="card">
30    <div class="card-header">
31      <h2 class="card-title">Visión</h2>
32    </div>
33    <div class="card-body">
34      <p>
35        Ser semillero de grandes deportistas integros que
36        sirvan a la sociedad y al País. Alcanzar la maxima
37        distincion deportiva y sobre salir entre todos los
38        cantones del territorio Nacional.
39      </p>

```

Código 2.29 Código fuente del archivo index.phtml

A continuación, se muestra los componentes necesarios para el módulo Torneo. En la Figura 2.41 se presenta la vista principal de este módulo, en la que se detallan los Torneos que se encuentra creados en la aplicación. En el Código 2.30 se indica el código fuente de esta vista. En la línea dos se crea una variable con el título de la vista, la cual se usa también para nombrar la pestaña del navegador. En la línea 9 se hace una referencia en el texto, el cual redirecciona a la página donde está el formulario de creación de los torneos. En las líneas 12 a la 19 se crea la tabla con sus cabeceras donde alojan los registros creados. Además, en la última columna se añaden las opciones de edición y borrado que al igual que la creación nos direccionan a las paginas correspondientes a cada una de estas acciones. Entre la línea 20 y 26 se realiza el poblado de la tabla con los registros obtenidos de la base de datos.

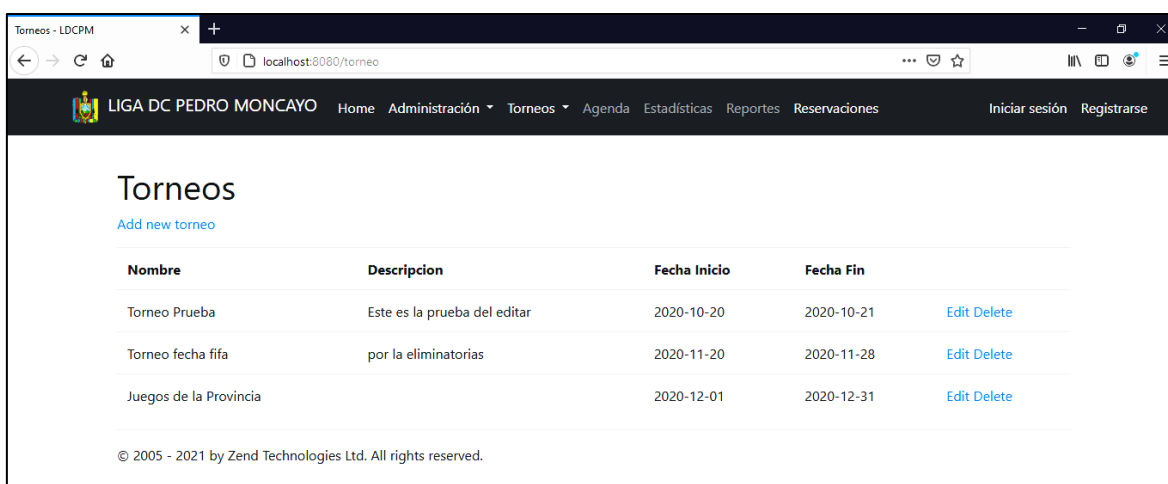


Figura 2.41 Vista principal del módulo Torneo.

```

1 <?php
2
3 $title = 'Torneos';
4 $this->headTitle($title);
5 ?>
6 <h1><?= $this->escapeHtml($title) ?></h1>
7 <p>
8
9     <a href="<?= $this->url('torneo', ['action' => 'add']) ?>">Add new torneo</a>
10 </p>
11
12 <table class="table">
13 <tr>
14     <th>Nombre</th>
15     <th>Descripcion</th>
16     <th>Fecha Inicio</th>
17     <th>Fecha Fin</th>
18     <th>&nbsp;</th>
19 </tr>
20 <?php foreach ($torneos as $torneo) : ?>
21     <tr>
22         <td><?= $this->escapeHtml($torneo->name) ?></td>
23         <td><?= $this->escapeHtml($torneo->description) ?></td>
24         <td><?= $this->escapeHtml($torneo->date_start) ?></td>
25         <td><?= $this->escapeHtml($torneo->date_end) ?></td>
26         <td>
27             <a href="<?= $this->url('torneo', ['action' => 'edit', 'id' => $torneo->id]) ?>">Edit</a>
28             <a href="<?= $this->url('torneo', ['action' => 'delete', 'id' => $torneo->id]) ?>">Delete</a>
29         </td>
30     </tr>
31 <?php endforeach; ?>
32 </table>
33

```

Código 2.30 Código fuente de la vista principal del módulo Torneo

En la Figura 2.42 se presenta la vista que tiene el formulario de creación de torneos, para desarrollar este formulario, se usa Zend-form cuyo código fuente de los elementos utilizado se indica en el Código 2.31. En la línea 12 se define el método que se usa para la recepción de los datos, en este caso es el método post. Entre la línea 15 y 18 se define el id, este no se muestra en el formulario, pero se enviará al momento de realizar la petición. Entre la 19 y 31 se definen dos cajas de texto para ingresar el nombre del torneo y su descripción. En las líneas 35 a la 43 definimos calendarios para seleccionar las fechas de inicio y finalización del torneo y, por último, está el botón que desencadena la acción de crear.

En el Código 2.32 se presenta el código fuente de la vista para la acción de crear un nuevo torneo. El archivo tiene como nombre add.phtml. En la línea 7 se ingresa el nombre de la vista. De la línea 9 a la 19 se asignan los elementos del formulario para proceder a dibujarlos. En la línea 20 se prepara el URL. En la línea 21 se prepara el formulario. De la línea 25 a la 45 se distribuye los elementos en el formulario. En la línea 46 se obtiene el id para enviar en la petición y por último se obtiene el valor del botón que será usado por el controlador para ejecutar la acción requerida.

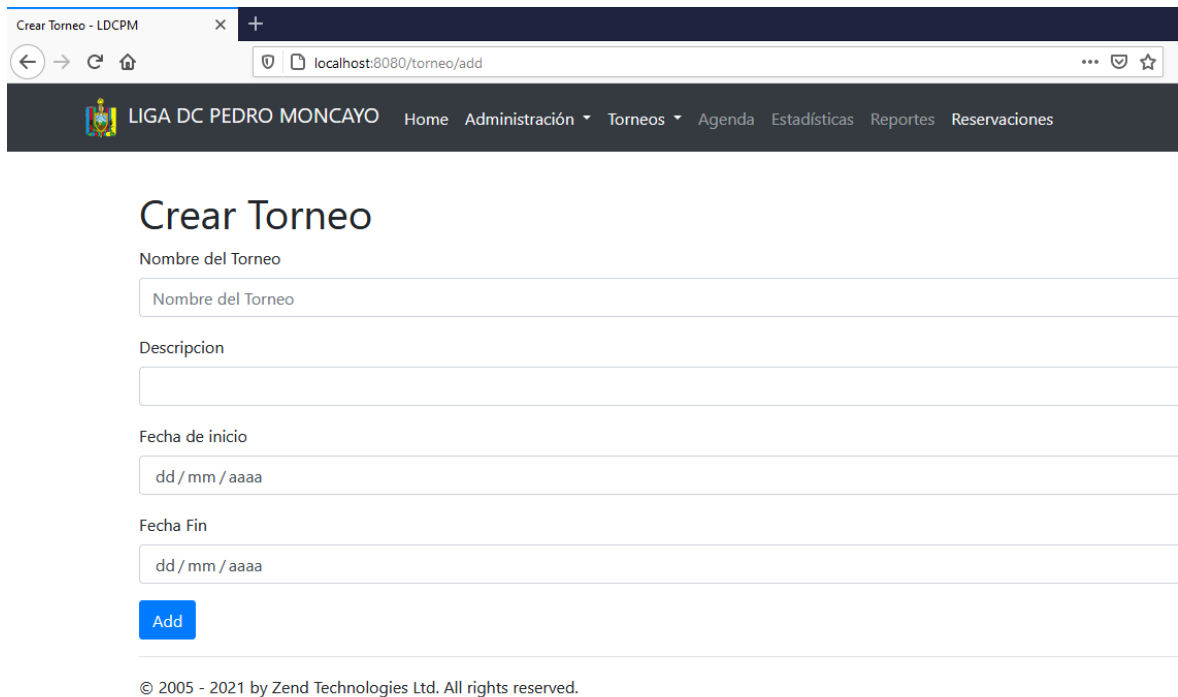


Figura 2.42 Vista para la creación de torneos

```

11     parent::__construct('TorneoForm');
12     $this->setAttributes(array(
13         'method' => 'post'
14     ));
15     $this->add([
16         'name' => 'id',
17         'type' => 'hidden',
18     ]);
19     $this->add([
20         'name' => 'name',
21         'type' => 'text',
22         'options' => [
23             'label' => 'Nombre del Torneo',
24         ],
25     ]);
26     $this->add([
27         'name' => 'description',
28         'type' => 'text',
29         'options' => [
30             'label' => 'Descripcion',
31         ],
32     ]);
33     $this->add([
34         'name' => 'date_start',
35         'type' => Element\Date::class,
36         'options' => [
37             'label' => 'Fecha de inicio',
38         ],
39     ]);$this->add([
40         'name' => 'date_end',
41         'type' => Element\Date::class,
42         'options' => [
43             'label' => 'Fecha Fin',
44         ],
45     ]);
46     $this->add([
47         'name' => 'submit',
48         'type' => 'submit',
49         'attributes' => [
50             'value' => 'Go',
51             'id' => 'submitbutton',

```

Código 2.31 Código de los elementos del formulario.


```

7 <h1><?= $this->escapeHtml($title) ?></h1>
8 <?php
9 $ntorneo = $form->get('name');
10 $ntorneo->setAttribute('class', 'form-control');
11 $ntorneo->setAttribute('placeholder', 'Nombre del Torneo');
12 $dtorneo = $form->get('description');
13 $dtorneo->setAttribute('class', 'form-control');
14 $fitorneo = $form->get('date_start');
15 $fitorneo->setAttribute('class', 'form-control');
16 $fftorneo = $form->get('date_end');
17 $fftorneo->setAttribute('class', 'form-control');
18 $submit = $form->get('submit');
19 $submit->setAttribute('class', 'btn btn-primary');
20 $form->setAttribute('action', $this->url('torneo', ['action' => 'add']));
21 $form->prepare();
22 echo $this->form()->openTag($form);
23 ?>
24 <?php ?>
25 <div class="form-group">
26     <?= $this->formLabel($ntorneo) ?>
27     <?= $this->formElement($ntorneo) ?>
28     <?= $this->formElementErrors()->render($ntorneo, ['class' => 'help-block']) ?>
29 </div>
30 <div class="form-group">
31     <?= $this->formLabel($dtorneo) ?>
32     <?= $this->formElement($dtorneo) ?>
33     <?= $this->formElementErrors()->render($dtorneo, ['class' => 'help-block']) ?>
34 </div>
35 <div class="form-group">
36     <?= $this->formLabel($fitorneo) ?>
37     <?= $this->formElement($fitorneo) ?>
38     <?= $this->formElementErrors()->render($fitorneo, ['class' => 'help-block']) ?>
39 </div>
40 <div class="form-group">
41     <?= $this->formLabel($fftorneo) ?>
42     <?= $this->formElement($fftorneo) ?>
43     <?= $this->formElementErrors()->render($fftorneo, ['class' => 'help-block']) ?>
44 </div>
45 <?php
46 echo $this->formHidden($form->get('id'));
47 echo $this->formSubmit($form->get('submit'));

```

Código 2.32 Código fuente de la vista para crear torneos.

Para la edición de torneos se utiliza los mismos componentes del formulario y el código fuente de la vista es similar al de la creación de torneos, por lo tanto, en el Código 2.33 se muestra el cambio respecto a la vista para crear torneos.

```

$form->setAttribute('action', $this->url('torneo', ['action' => 'edit', 'id' => $id]));
$form->prepare();

```

Código 2.33 Diferencia con la vista de crear torneos.

Por último, tenemos la vista para eliminar los registros, el código fuente se indica en el Código 2.34. En la línea 3 se define la URL con los parámetros necesarios para que el

controlador realice la acción correspondiente. Entre la línea 10 y 12 se muestra un mensaje de confirmación para la eliminación del registro. En la Figura 2.43 se indica la vista para la eliminación de los torneos.

```
1 <?php
2 $title = 'Eliminar Torneo';
3 $url = $this->url('torneo', ['action' => 'delete', 'id' => $id]);
4
5 $this->headTitle($title);
6 ?>
7 <h1><?= $this->escapeHtml($title) ?></h1>
8
9 <p>
10     Esta seguro de Eliminar Torneo
11     "<?= $this->escapeHtml($torneo->name) ?>" by
12     "<?= $this->escapeHtml($torneo->description) ?>"?
13 </p>
14
15 <form action="<?= $url ?>" method="post">
16 <div class="form-group">
17     <input type="hidden" name="id" value="<?= (int) $torneo->id ?>" />
18     <input type="submit" class="btn btn-danger" name="del" value="Yes" />
19     <input type="submit" class="btn btn-success" name="del" value="No" />
20 </div>
21 </form>
```

Código 2.34. Código fuente de la vista para eliminar torneos.

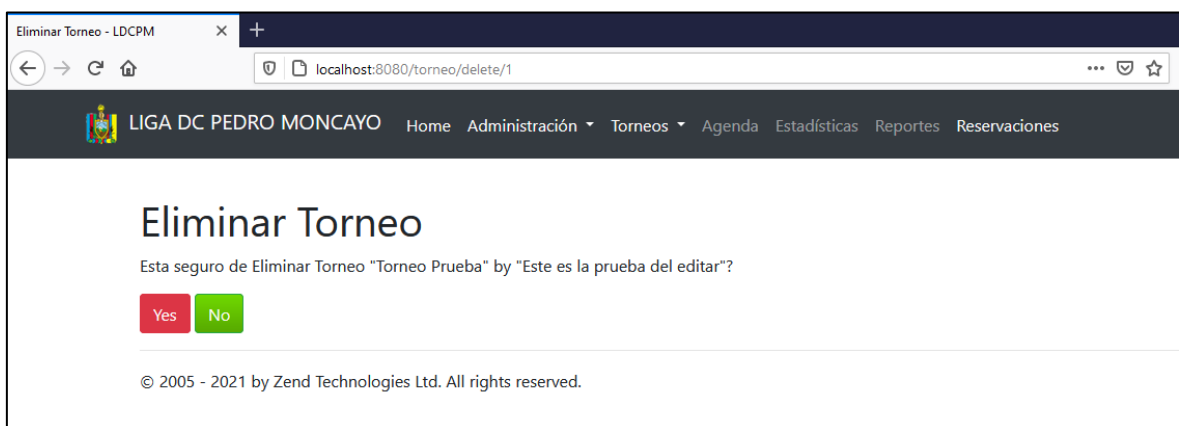


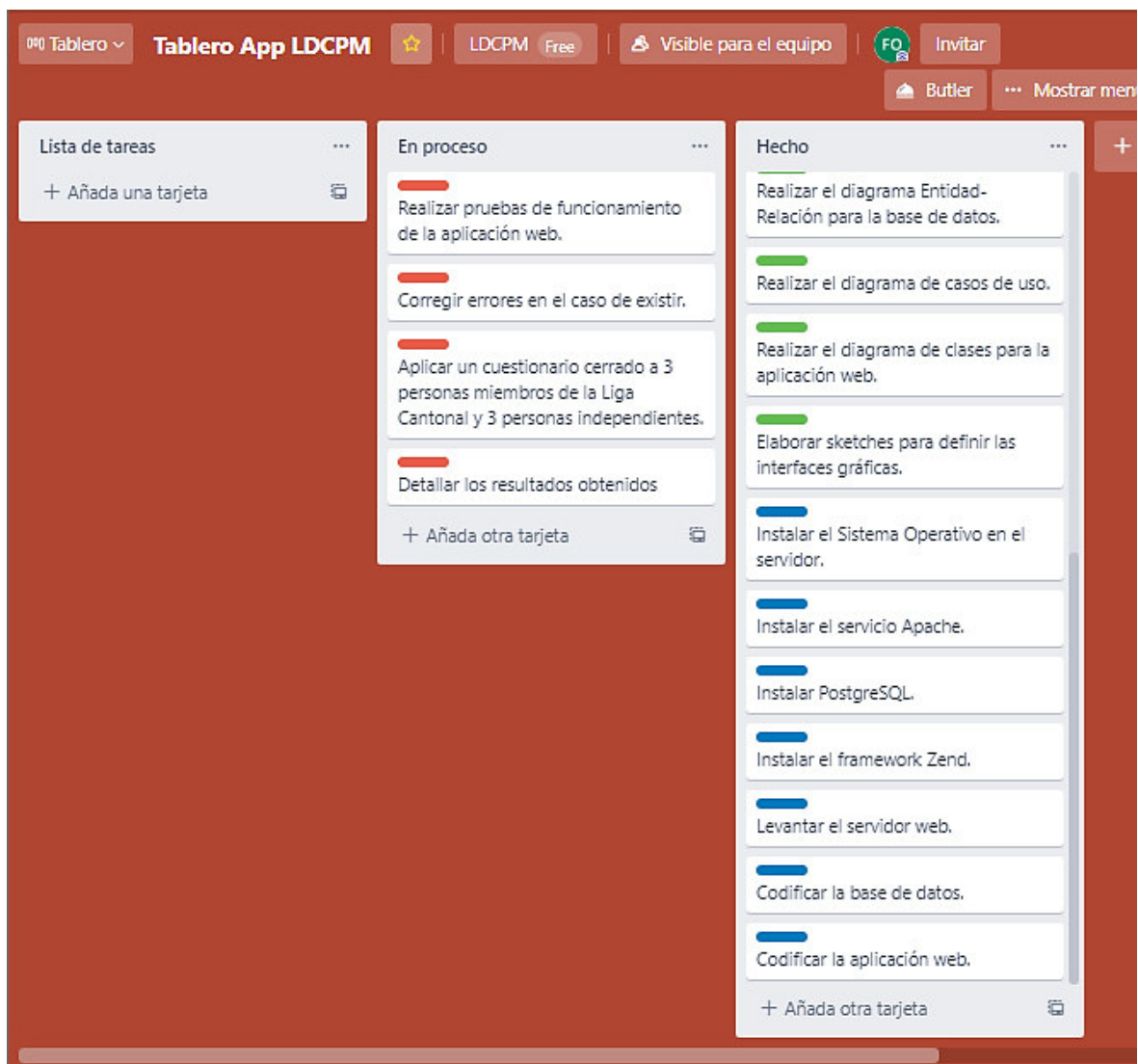
Figura 2.43. Vista para eliminar torneos.

3. RESULTADOS Y DISCUSIÓN

En esta sección se presentan los resultados obtenidos en las pruebas realizadas al prototipo de aplicación web, con las cuales se valida el cumplimiento de los requerimientos.

3.1 ACTUALIZACIÓN DEL TABLERO KANBAN

En la Figura 3.1 se muestra el tablero actualizado con las tareas correspondientes a la fase de pruebas en la columna “En proceso” y las tareas de implementación en la columna “Hecho”



3.2 PRUEBAS DE FUNCIONAMIENTO

En este apartado se realiza las pruebas en conjunto de todos los componentes de la aplicación web, para validar que funcionen de manera correcta con el servidor y la base de datos.

3.2.1 INICIAR SESIÓN EN LA APLICACIÓN WEB

En la Figura 3.2 se presenta la interfaz gráfica para el inicio de sesión en la aplicación web, en esta prueba se valida que el usuario pueda iniciar sesión con el nombre de usuario y contraseña.

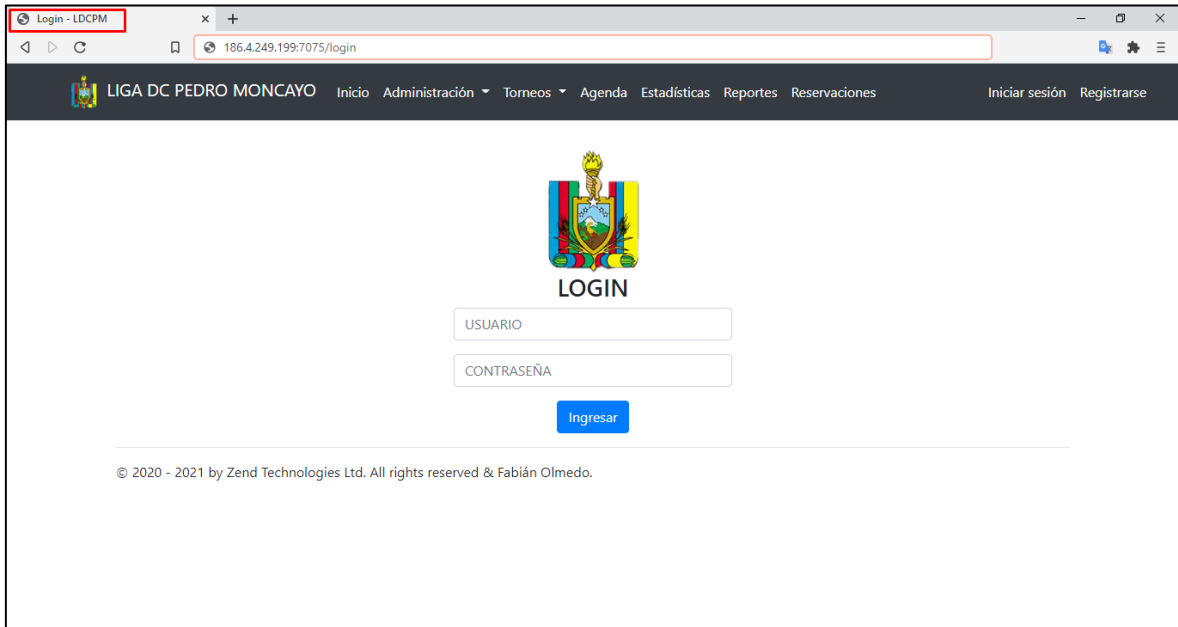


Figura 3.2 Ingreso a la aplicación web

En la Figura 3.3 se indica un mensaje cuando al iniciar sesión se realiza la validación y cualquiera de los datos son erróneos.

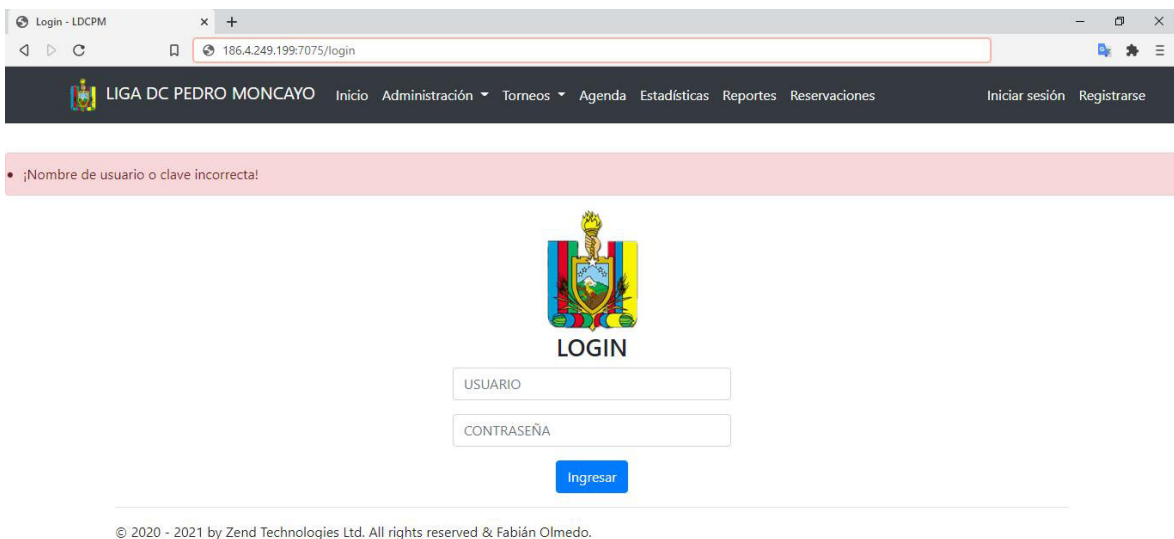


Figura 3.3 Ingreso fallido

Si las credenciales son correctas la aplicación redirige a la pantalla principal como se muestra en la Figura 3.4. En esta pantalla en la parte superior derecha se muestra el nombre de usuario de la persona que ingreso al sistema.



Figura 3.4 Página principal después del inicio de sesión.

3.2.2 GESTIÓN DE UN TORNEO

En este apartado, se presentan las pruebas que permiten validar el funcionamiento de la gestión de un torneo; que comprende: la visualización de todos los torneos almacenados en el sistema; ingreso, actualización y eliminación de los eventos.

La página principal de este módulo se indica en la Figura 3.5 y se visualiza la información de los torneos creados anteriormente. Para agregar un nuevo torneo la aplicación web presenta un formulario exhibido en la Figura 3.6. Los campos obligatorios no pueden ser vacíos, en este caso, mostrará un aviso indicando que no puede estar vacío el campo.

Torneos

[Añadir Torneo](#)

Nombre	Descripción	Fecha de Inicio	Fecha de Finalización	
Torneo fecha fifa	por la eliminatorias	2020-11-20	2020-11-28	Editar Borrar
Juegos de la Provincia		2020-12-01	2020-12-31	Editar Borrar
Torneo Prueba	Este es la prueba del editar	2020-10-20	2020-10-21	Editar Borrar
Torneo Validacion	torneo pruebas	2020-12-28	2021-01-31	Editar Borrar
Torneo Carnaval		2021-03-01	2021-03-31	Editar Borrar

© 2020 - 2021 by Zend Technologies Ltd. All rights reserved & Fabián Olmedo.

Figura 3.5 Página principal del módulo Torneo.

Crear Torneo

Nombre del Torneo

- Value is required and can't be empty

Descripcion

Fecha de inicio

- Value is required and can't be empty

Fecha Fin

- Value is required and can't be empty

[Add](#)

Figura 3.6 Aviso de validación de campos vacíos.

Una vez ingresados los datos correctamente, el torneo se guarda y redirecciona a la página principal de los torneos donde se muestra ya el último registro creado, como se indica en las Figuras 3.7 y 3.8.

Crear Torneo

Nombre del Torneo

- Value is required and can't be empty

Descripcion

Fecha de inicio

- Value is required and can't be empty

Fecha Fin

- Value is required and can't be empty

[Add](#)

Figura 3.7 Formulario de creación completo

Torneos

[Añadir torneo](#)

Nombre	Descripcion	Fecha Inicio	Fecha Fin	
Torneo Prueba	Este es la prueba del editar	2020-10-20	2020-10-21	Edit Delete
Torneo fecha fifa	por la eliminatorias	2020-11-20	2020-11-28	Edit Delete
Juegos de la Provincia		2020-12-01	2020-12-31	Edit Delete
Torneo Validacion	torneo para pruebas de funcionamiento	2020-12-28	2021-01-31	Edit Delete

© 2005 - 2021 by Zend Technologies Ltd. All rights reserved.

Figura 3.8 Lista actualizada de los torneos.

La información del torneo se almacena en la base de datos en la tabla Tournament, en la Figura 3.9 se indica el registro de la base de datos.

id	name	description	date_start	date_end
5	Torneo Validacion	torneo para pruebas de funcionamiento	2020-12-28	2021-01-31

Figura 3.9 Registro del torneo almacenado en la base de datos.

A continuación, se realiza la prueba de la edición de los datos del torneo. En la Figura 3.10 se muestra el formulario que permite la actualización de los campos. Si los campos ingresados son correctos finalizada la edición, redirige a la página principal del torneo.



Editar Torneo

Nombre del Torneo
Torneo Validacion

Descripcion
torneo validado

Fecha de inicio
28/12/2020

Fecha Fin
31/01/2021

Edit

Figura 3.10 Formulario de edición de torneos.

En la Figura 3.11 se muestra el registro actualizado en la base de datos.

```

cucpml prueba=# select * from private.tournament order by id desc,
id | name | description | date_start | date_end
-----+-----+-----+-----+-----
5 | Torneo Validacion | torneo validado | 2020-12-28 | 2021-01-31

```

Figura 3.11 Registro Actualizado de la Base de Datos.

Por último, para proceder a borrar cualquiera de los torneos mostrados en el listado; la aplicación web pide una confirmación de que realmente desea borrar el torneo. En la Figura 3.12 se muestra el mensaje de confirmación.



Figura 3.12 Mensaje de Confirmación

3.2.3 GESTIÓN DE UN EVENTO

En este apartado se realizan las pruebas de funcionamiento al módulo de eventos. Al igual que el módulo anterior la gestión comprende de la visualización, ingreso, actualización y eliminación de los eventos.

La página principal de este módulo presenta el listado de todos los eventos y a partir de ahí se redirecciona a las diferentes acciones que se quiera realizar. En la Figura 3.13 se presenta esta página.

Nombre	Descripción	Fecha	
Futbol Sub 15 PM vs Cayambe		2020-01-01 03:00:00	Edit Delete
Futbol Sub 14 Pedro Moncayo vs Cayambe		2020-12-02 09:00:00	Edit Delete
Futbol Sub 14 PQ vs PVM		2020-12-02 11:00:00	Edit Delete
Futbol Sub 14 Los Bancos vs Mejia		2020-12-03 09:00:00	Edit Delete
Futbol Sub 14 PM vs Rumiñahui		2020-12-03 11:00:00	Edit Delete

Figura 3.13 Página principal el módulo Eventos

En la Figura 3.14 se presenta el formulario encargado de la recolección de datos ingresados por el usuario. Al igual que el otro módulo este valida que no existan campos vacíos que sean obligatorios de ingresar.

Crear Evento

Nombre del Evento

Descripción

Fecha del Evento

1 de enero de 2021, 00:00

Seleccione el Torneo

Seleccione la Disciplina

Seleccione el Escenario

Seleccione el Juez

Add

Figura 3.14 Formulario para crear eventos

La Figura 3.15 muestra el registro ingresado en la base de datos, el cual se almacena en la tabla Events .

```

c0cpm_prueba=# select * from private.events order by id desc;
 id |      name      | description |      date      | _torneo | _escenarios | _disciplina | _referee |
-----+-----+-----+-----+-----+-----+-----+-----+
  7 | Evento Validado |            | 2021-01-01 00:00:00 | 5 | 1 | 3 | 1 | 20
21-01-11 02:04:35.133909

```

Figura 3.15 Registro almacenado en la base de datos

La siguiente acción que se realiza la prueba de funcionamiento es la de editar un evento. El formulario asignado para la edición del evento se presenta en la Figura 3.16, cabe

mencionar que al igual que el formulario de creación, también se valida que los campos obligatorios no queden vacíos.

Editar Evento

Nombre del Evento
Evento Validado

Descripción

Fecha del Evento
1 de enero de 2021, 00:00

Seleccione el Torneo
Torneo fecha fifa

Seleccione la Disciplina
Futbol peso gallo

Seleccione el Escenario
Estadio LDCPM

Seleccione el Juez
Gustavo Velasquez

Editar

Figura 3.16 Formulario para editar eventos.

En la Figura 3.17 se observa que el registro de la base de datos fue actualizado.

id	created	name	description	date	_torneo	_escenarios	_disciplina	_referee
7	21-01-11 02:04:35.133909	Evento Validado Editado	ya se edito	2021-01-01 00:00:00	5	1	3	1 20

Figura 3.17 Registro actualizado del evento.

Para eliminar un evento es necesario pulsar en la opción correspondiente, a continuación, se presente un mensaje solicitando la confirmación para eliminar el evento seleccionado; en caso de ser afirmativa la confirmación, el evento se elimina de la base de datos caso contrario se mantiene. En la Figura 3.18 se muestra el mensaje de confirmación.

Eliminar Evento

Esta seguro de Eliminar el Evento "Evento Validado Editado"?

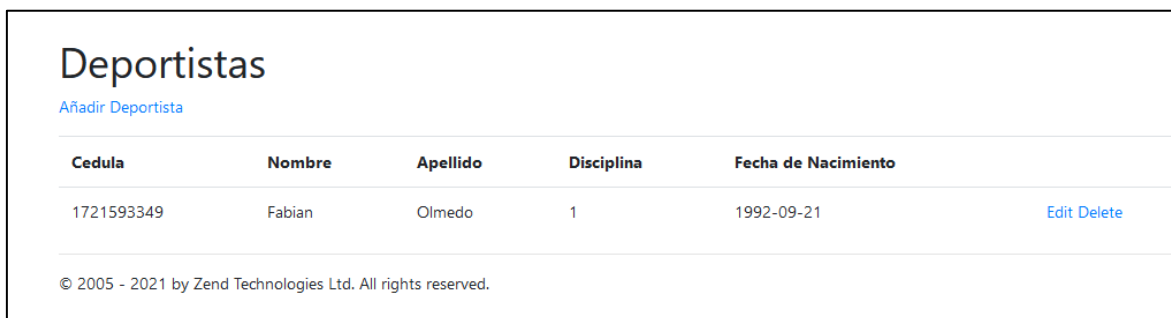
Yes No

Figura 3.18 Mensaje de confirmación para eliminar eventos.

3.2.4 GESTIÓN DE JUGADORES

Para acceder a la gestión de los jugadores es necesario ir al módulo de torneos donde tendremos la opción para visualizar, crear, actualizar y eliminar a los deportistas.

En la página principal de la gestión de los Deportistas, la aplicación web presenta el listado de todos los deportistas cargados en el sistema, además de las acciones que permiten cumplir con los requerimientos, esta página principal se muestra en la Figura 3.19.



Cedula	Nombre	Apellido	Disciplina	Fecha de Nacimiento	
1721593349	Fabian	Olmedo	1	1992-09-21	Edit Delete

© 2005 - 2021 by Zend Technologies Ltd. All rights reserved.

Figura 3.19 Página principal para la gestión de deportistas.

Para añadir a un nuevo jugador se debe escoger la opción correspondiente a la acción de crear un deportista. En la Figura 3.20 se presenta el formulario para la creación de un nuevo deportista. Cabe mencionar que al igual que los módulos anteriores el formulario valida que los campos obligatorios se encuentren llenos.

En la Figura 3.21 se muestra el registro en la base de datos, el cual es almacenado en la tabla Deportistas.

CREAR Deportista

Seleccione el Cantón

Pedro Moncayo

Nombre

Juan Andres

Apellido

Carrillo

Fecha de Nacimiento

05 / 02 / 2013

Cedula

1721546693

Sexo

Masculino

Seleccione la Disciplina

Futbol PM

Add

Figura 3.20 Formulario para crear deportista.

id	_canton	firstname	lastname	date	cedula	sexo	_distc	status	created
4	1	Juan Andres	Carrillo	2013-02-05	1721546693	Masculino	1	f	2021-01-11 04:25:27.686816

Figura 3.21 Registro nuevo almacenado en la base de datos

Para modificar la información de un jugador seleccionado, esta se carga en el formulario, se puede realizar los cambios que el usuario considere necesarios, como se indica en la Figura 3.22 en donde se añadió el segundo apellido del deportista y una vez actualizado, redirige a la página principal de la gestión de disciplinas.

Editar Deportista

Seleccione el Cantón

Pedro Moncayo

Nombre

Juan Andres

Apellido

Carrillo Narvaes

Fecha de Nacimiento

05 / 02 / 2013

Cedula

1721546693

Sexo

Masculino

Seleccione la Disciplina

Futbol PM

[Edit](#)

Figura 3.22 Formulario para editar deportista.

Los cambios realizados en el deportista se ven reflejados en la base de datos como se muestra en la Figura 3.23

id	_canton	firstname	lastname	date	cedula	sexo	_distc	status	created
4	1	Juan Andres	Carrillo Narvaes	2013-02-05	1721546693	Masculino	1	f	2021-01-11 04:25:27.686816

Figura 3.23 Registro actualizado del deportista.

Si se requiere eliminar un deportista, en la página principal de la gestión se debe seleccionar la opción correspondiente, realizada esta acción la aplicación muestra un mensaje de confirmación para eliminar el registro como se indica en la Figura 3.24. En caso de ser afirmativa la respuesta, el registro es eliminado completamente de la base de datos.



Figura 3.24 Mensaje de confirmación para eliminar Deportista

3.2.5 GESTIÓN DE UN USUARIO

La aplicación web permite al usuario administrador gestionar los usuarios. En este apartado se realizan las pruebas correspondientes a la gestión de un usuario.

Para ingresar al módulo gestión de Usuarios, en la barra de navegación se debe seleccionar Administración y seguidamente Usuarios. Esta acción redirige a la vista principal del módulo como indica la Figura 3.25, en la cual se muestra un listado de todos los usuarios creados en el sistema, además de todas las acciones concernientes a la gestión.

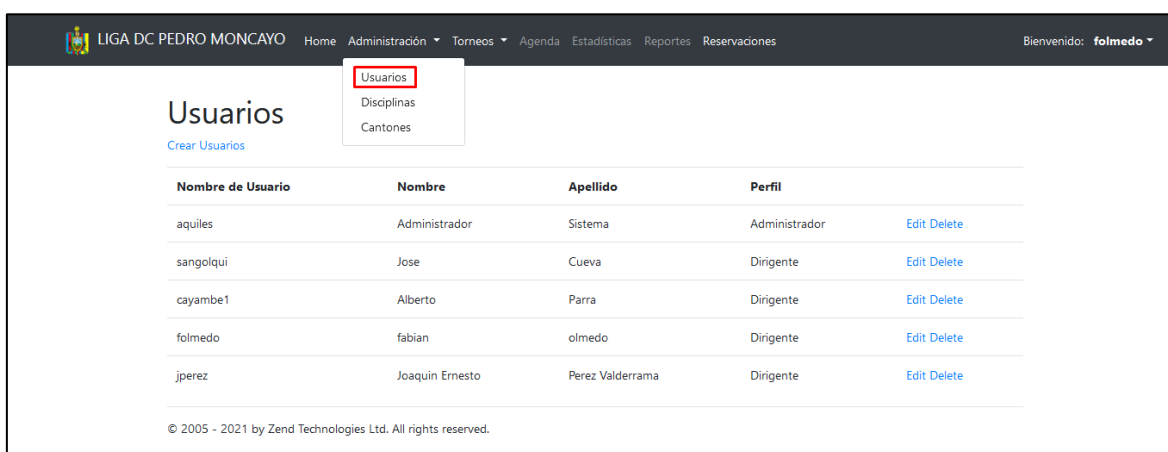
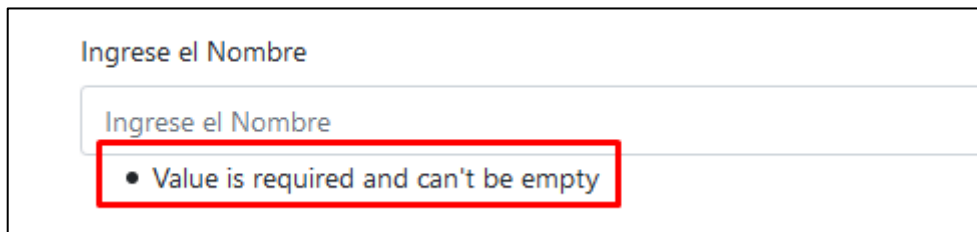


Figura 3.25 Vista principal del módulo Usuarios

A continuación, se presentan las pruebas de registro y actualización y eliminación de un usuario.

Para crear un usuario la aplicación web presenta un formulario. Si los campos requeridos se dejan vacíos, el sistema presenta un aviso de que el campo debe ser ingresado como

se indica en la Figura 3.26. En la Figura 3.27 se presenta un ejemplo del ingreso de un nuevo usuario.



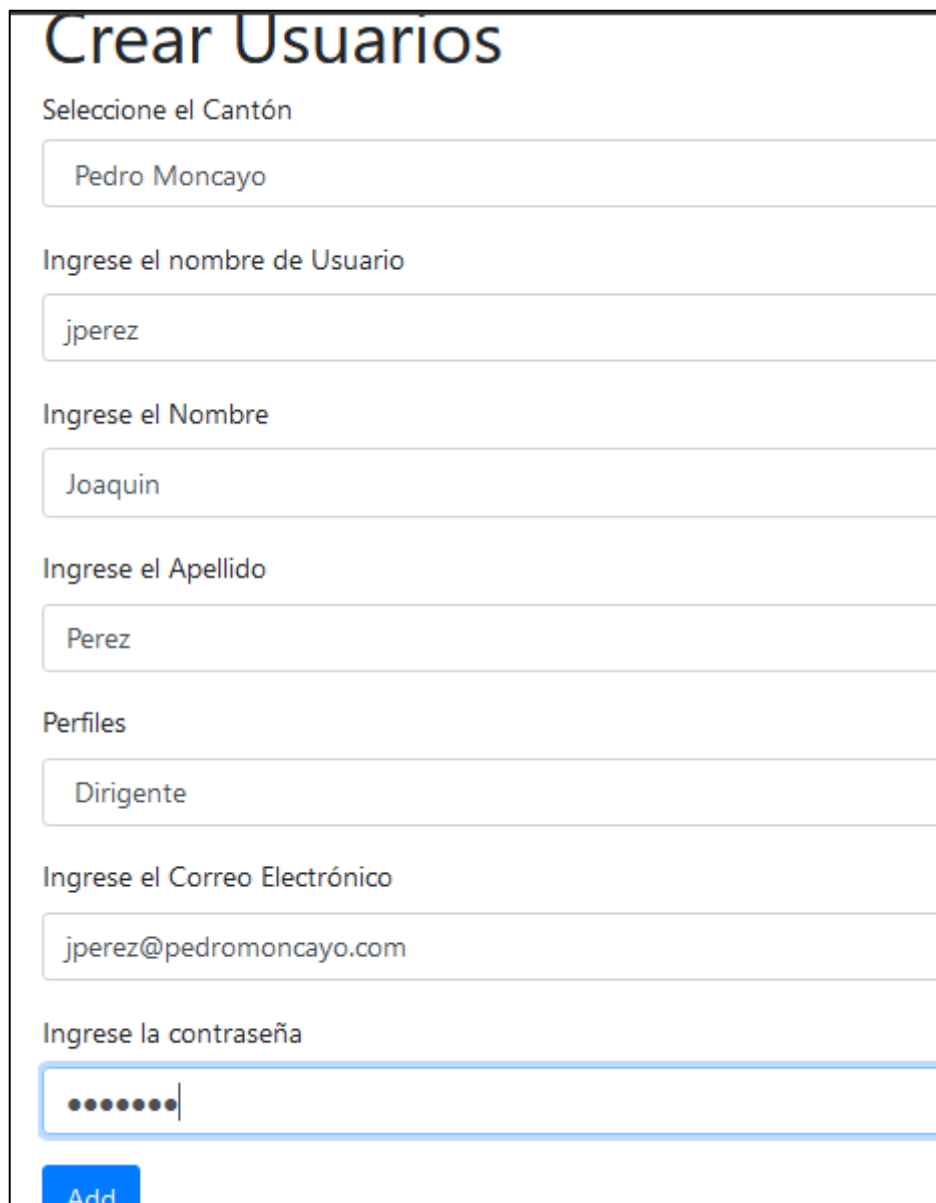
Ingrese el Nombre

Ingrese el Nombre

- Value is required and can't be empty

Detailed description: This image shows a close-up of a web form. At the top, the text 'Ingrese el Nombre' is displayed. Below it is a text input field containing the same text 'Ingrese el Nombre'. A red rectangular box highlights an error message below the input field: a bullet point followed by the text 'Value is required and can't be empty'.

Figura 3.26 Aviso que el campo no puede ser vacío.



Crear Usuarios

Seleccione el Cantón

Pedro Moncayo

Ingrese el nombre de Usuario

jperez

Ingrese el Nombre

Joaquin

Ingrese el Apellido

Perez

Perfiles

Dirigente

Ingrese el Correo Electrónico

jperez@pedromoncayo.com

Ingrese la contraseña

.....

Add

Detailed description: This image is a screenshot of a web form titled 'Crear Usuarios'. The form contains several input fields: a dropdown menu for 'Seleccione el Cantón' with 'Pedro Moncayo' selected; a text field for 'Ingrese el nombre de Usuario' with 'jperez'; a text field for 'Ingrese el Nombre' with 'Joaquin'; a text field for 'Ingrese el Apellido' with 'Perez'; a dropdown menu for 'Perfiles' with 'Dirigente' selected; a text field for 'Ingrese el Correo Electrónico' with 'jperez@pedromoncayo.com'; and a password field for 'Ingrese la contraseña' with seven dots. At the bottom left, there is a blue button labeled 'Add'.

Figura 3.27 Formulario para crear nuevo usuario

La información ingresada, es almacenada en la base de datos en la tabla sys_users como se presenta en la Figura 3.28.

id	_canton	username	firstname	lastname	profile	email	password
6	1	jperez	Joaquin	Perez	Dirigente	jperez@pedromoncayo.com	96ac9a11d94d8f982ba476aa4b5ef503

Figura 3.28 Registro nuevo almacenado en la base de datos.

La aplicación permite actualizar la información de un usuario. Para acceder a esta funcionalidad el usuario debe escoger la opción correspondiente a la edición de un usuario en la vista principal del módulo. La información del usuario seleccionado se carga en el formulario como se muestra en la Figura 3.29 y el administrador puede modificar los campos que considere necesarios; en esta prueba se agregó el segundo nombre y apellido. Si la información es almacenada de manera exitosa la aplicación redirige a la vista principal del módulo.

Editar Usuario

Ingrese el nombre de Usuario

Ingrese el Nombre

Ingrese el Apellido

Seleccione el Cantón

Perfiles

Ingrese el Correo Electrónico

Ingrese la contraseña

Figura 3.29 Formulario para editar usuario

En la Figura 3.30 se muestra el registro actualizado en la base de datos.

id	canton	username	firstname	lastname	profile	email	
6	1	jperez	Joaquin Ernesto	Perez Valderrama	Dirigente	jperez@pedromoncayo.com	96ac9a11d94d8f982ba476aa4b5ef503

Figura 3.30 Registro actualizado en la base de datos

Si el administrador desea eliminar un usuario se debe seleccionar la opción en la vista principal del módulo, hecho esto la aplicación muestra un mensaje de confirmación para eliminar el registro, si la respuesta es afirmativa el usuario se elimina de la aplicación caso contrario se mantiene. La Figura 3.31 muestra este mensaje.

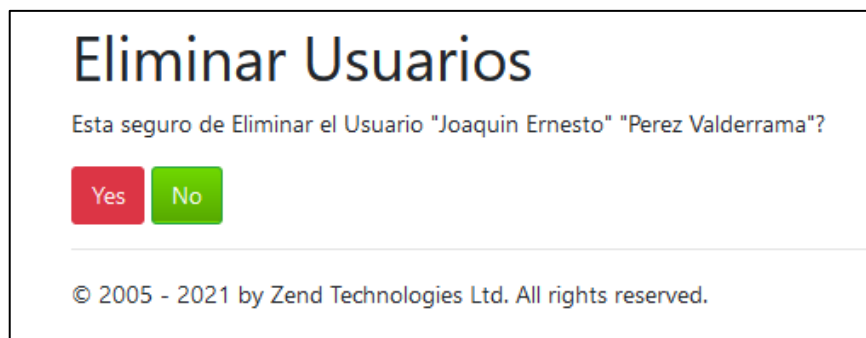


Figura 3.31 Mensaje para eliminar usuario.

3.2.6 PRUEBAS DE FUNCIONAMIENTO DEL MÓDULO AGENDA

La aplicación web permite a los usuarios tener una agenda con los eventos deportivos de interés. A continuación, se presentan las pruebas de visualización, agregación y eliminación de los eventos de la agenda.

La Figura 3.32 presenta el listado de eventos asociados a la agenda del usuario, se accede a esta vista escogiendo en la barra de navegación la opción Agenda.

Mi Agenda

[Añadir Evento](#)

Nombre	Descripcion	Fecha	
Futbol Sub 15 PM vs Cayambe		2020-01-01 03:00:00	Delete
Futbol Sub 14 Pedro Moncayo vs Cayambe		2020-12-02 09:00:00	Delete
Futbol Sub 14 PQ vs PVM		2020-12-02 11:00:00	Delete
Futbol Sub 14 Los Bancos vs Mejia		2020-12-03 09:00:00	Delete
Futbol Sub 14 PM vs Rumiñahui		2020-12-03 11:00:00	Delete
Evento Validado Editado	ya se edito	2021-01-01 00:00:00	Delete

© 2005 - 2021 by Zend Technologies Ltd. All rights reserved.

Figura 3.32 Vista principal del módulo agenda.

Si el usuario desea agregar nuevos eventos deportivos a su agenda, en la vista principal debe seleccionar añadir eventos, la cual redirige a otra vista donde se visualiza los eventos disponibles; al lado derecho de cada evento esta una opción para añadir como se muestra en la Figura 3.33.

Añadir Eventos a Mi Agenda

Nombre	Descripcion	Fecha	
Futbol Sub 10 PM vs Cayambe		2020-01-01 03:00:00	Añadir
Futbol Sub 14 Pedro Moncayo vs Cayambe		2020-12-02 09:00:00	Añadir

Figura 3.33 Formulario para añadir eventos a la agenda.

En la Figura 3.34 se muestran los registros ingresados en la base de datos al añadir nuevos eventos a la agenda. La tabla en la que se almacena los datos es agenda .

id	name	_users	_event
1	mi Agenda	1	1
2	mi Agenda	1	3

Figura 3.34 Registros ingresados en la tabla agenda

Finalmente, si el usuario desea eliminar un evento de la agenda que ya no sea de su interés, en la página principal del módulo, se debe seleccionar la opción correspondiente a eliminar. Al realizar esta acción la aplicación muestra un mensaje de confirmación de la

eliminación del evento, si la respuesta es positiva el evento es retirado de la agenda. En la Figura 3.35 se muestra el mensaje de confirmación.

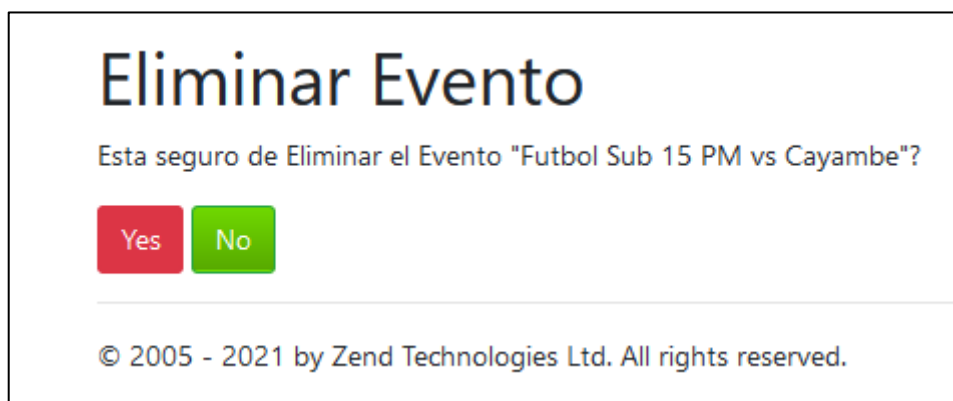


Figura 3.35 Mensaje de confirmación para eliminar un evento de la agenda.

3.2.7 PRUEBAS DE FUNCIONAMIENTO DEL MÓDULO ESTADÍSTICAS

La aplicación web en el módulo estadísticas presenta los medalleros y las tablas de posiciones. Las pruebas consisten en visualizar estos medalleros y tablas de posiciones.

Al ingresar al módulo de Estadísticas, la aplicación direcciona a la vista principal en la cual se muestra tres ítems como indica la Figura 3.36.

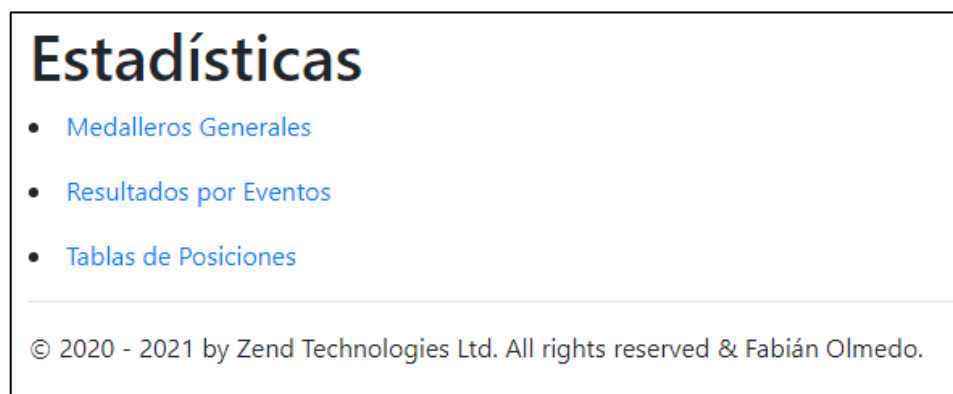


Figura 3.36 Vista principal del módulo Estadísticas.

Al seleccionar la opción de Medalleros generales la aplicación muestra un listado de todos los medalleros disponibles de acuerdo con cada torneo como se indica en la Figura 3.37.

Medalleros Generales			
Nombre	Descripcion	Fecha	
MG Torneo Fecha Fifa		2020-12-01	Ver
MG Juegos de la Provincia		2020-12-15	Ver

Figura 3.37 Listado de Medalleros.

Para la prueba de funcionamiento se ha escogido uno de los medalleros. En la Figura 3.38 se muestra el medallero general de un torneo, donde se visualiza el cantón, las medallas de oro, plata y bronce, además, del total de medallas obtenidas.

MG Torneo Fecha Fifa				
Canton	Oro	Plata	Bronce	Total
Pedro Moncayo	0	1	1	2
Cayambe	0	1	1	2
Rumiñahui	0	1	1	2
Mejía	0	1	1	2
Puerto Quito	0	1	1	2
Pedro Vicente Maldonado	0	1	1	2
Los Bancos	0	1	1	2

Figura 3.38 Ejemplo de un medallero general

Si el usuario escoge la opción de resultados por evento, la aplicación le muestra un listado de todas las disciplinas disponibles específico considerando siempre la relación directa con el torneo como se indica en la Figura 3.39.

Listado de Disciplinas
<ul style="list-style-type: none"> • Resultados Eventos Fútbol • Resultados Eventos Basket • Resultados Eventos Ajedrez • Resultados Eventos Combate • Resultados Eventos Atletismo • Resultados Eventos Ciclismo
© 2020 - 2021 by Zend Technologies Ltd. All rights reserved & Fabián Olmedo.

Figura 3.39 Listado por disciplina.

Para la prueba hemos tomado la disciplina futbol. En la Figura 3.40 se presenta listado con la información correspondiente a cada evento.

Torneo	Evento	Cantón Local	Local	Cantón Visitante	Visitante	
Juegos de la Provincia	Futbol Sub 14 Pedro Moncayo vs Cayambe	Pedro Moncayo	Futbol PM	Cayambe	Futbol C	Ver
Juegos de la Provincia	Futbol Sub 14 PQ vs PVM	Puerto Quito	Futbol PQ	Pedro Vicente Maldonado	Futbol PVM	Ver
Juegos de la Provincia	Futbol Sub 14 Los Bancos vs Mejía	Los Bancos	Futbol B	Mejía	Futbol M	Ver
Juegos de la Provincia	Futbol Sub 10 PM vs Rumiñahui	Pedro Moncayo	Futbol PM	Rumiñahui	Futbol R	Ver
Torneo Carnaval	Futbol PM vs Futbol Cayambe	Pedro Moncayo	Futbol PM	Cayambe	Futbol Cayambe	Ver

© 2020 - 2021 by Zend Technologies Ltd. All rights reserved & Fabián Olmedo.

Figura 3.40 Resultados de eventos por disciplina.

Finalmente, si el usuario desea ver las tablas de posiciones, la aplicación web le muestra un listado de las tablas de posición asociadas a una disciplina y a un torneo específico como se indica en la Figura 3.41

Nombre	Disciplina	Torneo	
Tabla de posiciones Sub 15	Futbol	Torneo Juegos de la Provincia	Ver

© 2005 - 2021 by Zend Technologies Ltd. All rights reserved.

Figura 3.41 Listado de Tablas de posiciones.

En la Figura 3.42 se presenta la tabla de posiciones para la disciplina de futbol la cual muestra los cantones y todas las estadísticas.

Tabla de Posiciones Sub 15								
Canton	PJ	PG	PE	PP	GA	GC	GD	Pts
Pedro Moncayo	5	4	1	0	20	5	15	13
Rumiñahui	5	4	1	0	10	5	5	13
Pedro Vicente Maldonado	5	2	1	2	10	10	0	7
Cayambe	5	1	1	3	8	12	-4	4

© 2005 - 2021 by Zend Technologies Ltd. All rights reserved.

Figura 3.42 Ejemplo de una tabla de posiciones

3.2.8 PRUEBAS DE FUNCIONAMIENTO DEL MÓDULO REPORTES

La aplicación web en el módulo reportes presenta tres reportes generales. Las pruebas consisten en visualizar estos reportes.

Al ingresar al módulo de Reportes, la aplicación direcciona a la vista principal en la cual se muestra tres ítems como indica la Figura 3.43.



Figura 3.43 Listado de Reportes

Al ingresar al Reporte General de Torneos, se debe seleccionar el torneo y la aplicación web mostrará datos generales como el total de disciplinas participantes, el número total de eventos realizados y el Total de medallas entregadas como se indica en la Figura 3.44.

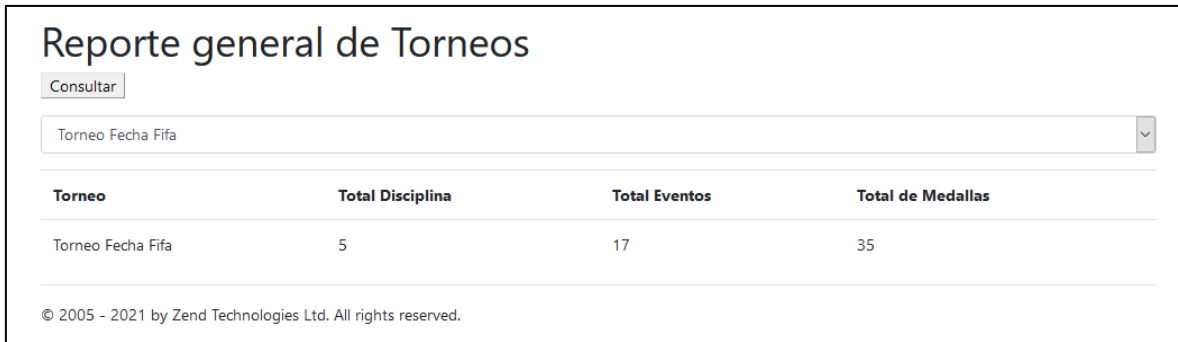


Figura 3.44 Reporte general de torneos.

En la Figura 3.45 se muestra el Reporte general de disciplinas en torneo, este reporte indica datos generales del torneo en base a la disciplina seleccionada. Para obtener este reporte se debe especificar el Torneo y la Disciplina.

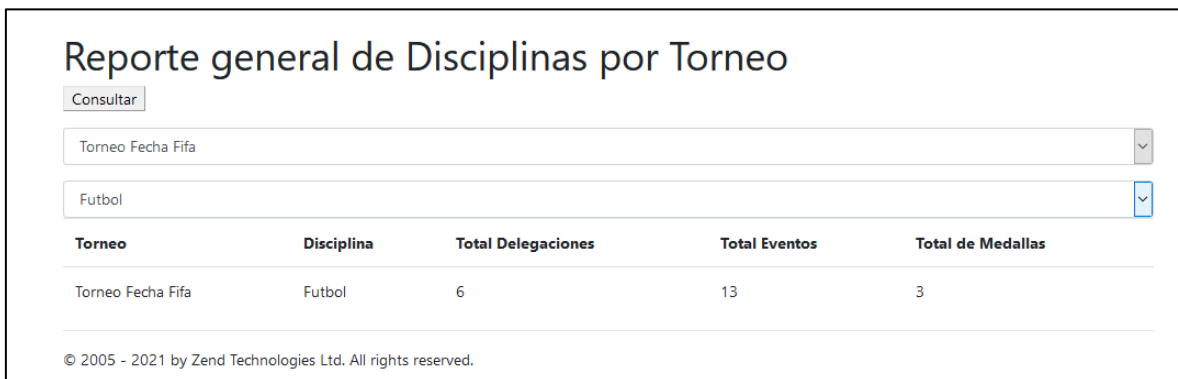


Figura 3.45 Reporte general de disciplinas por torneo.

Por último, en la Figura 3.46 se muestra el Reporte general de delegaciones por torneo, al igual que los reportes anteriores, se muestra valores generales como el número de disciplinas que presento cada delegación, el número de eventos en los que participó y el total de medallas obtenidas. Para consultar este reporte se debe especificar el torneo y la delegación.

Reporte General de Delegaciones por Torneo

Consultar

Torneo Fecha Fifa

Pedro Moncayo

Torneo	Canton	Total Disciplinas	Total Eventos	Total de Medallas
Torneo Fecha Fifa	Pedro Moncayo	6	37	5

© 2005 - 2021 by Zend Technologies Ltd. All rights reserved.

Figura 3.46 Reporte general de delegaciones por torneo.

3.2.9 GESTIÓN DE UNA RESERVACIÓN

La aplicación web permite a los usuarios realizar reservaciones de los escenarios deportivos para eventos privados. En esta sección se realizan las pruebas de funcionamiento de la gestión de una reservación.

A continuación, se presentan las pruebas de visualización, creación, actualización y eliminación de una reservación.

Para acceder al módulo de reservaciones el usuario debe escoger en la barra de navegación la opción reservaciones y la aplicación le redirecciona a la vista principal de este módulo. En la Figura 3.47 se muestra un listado de reservaciones y las acciones correspondientes a una gestión.

RESERVACIONES

[Añadir Reservación](#)

Nombre	Descripción	Fecha	Escenario	Reservado	
Reservacion FO	Reservacion para entrenar	2020-12-04 16:00:00	1	1	Edit Delete

© 2005 - 2021 by Zend Technologies Ltd. All rights reserved.

Figura 3.47 Vista principal del módulo reservaciones.

Para crear una reservación el sistema al igual que módulos anteriores, presenta un formulario como se muestra en la Figura 3.48. Este formulario tiene validadores en los campos que son obligatorios para que no se dejen vacíos. Con los campos llenos de manera correcta se crea la nueva reservación y la aplicación direcciona a la vista principal del módulo.

En la Figura 3.49 se muestra el registro en la base de datos, el cual se almacena en la tabla reserva_escenarios.

Crear Reservación

Nombre de la Reservación

Descripción

Fecha de la Reservación

1 de enero de 2021, 00:00

Tiempo de Reservación (horas)

Seleccione el Escenario

Seleccione el Usuario

Habilitar Reservación

Add

Figura 3.48 Formulario para crear reservaciones.

id	name	description	date	time	_escenario	_user	state
5	REUNION COVID19		2021-01-01 00:00:00	0	1	1	t

Figura 3.49 Registro almacenado en la tabla reserva_escenarios

La aplicación permite actualizar la información de una reservación. La información de la reservación seleccionada se carga en el formulario como se indica en la figura 3.50 y el usuario puede modificar los campos necesarios, en esta prueba se agregó una descripción

a la reservación. Si la información actualizada es exitosa el sistema redirecciona a la vista principal del módulo.

Editar Reservación

Nombre de la Reservación

Descripción

Fecha de la Reservación

1 de enero de 2021, 00:00

Tiempo de Reservación (horas)

Seleccione el Escenario

Seleccione el Usuario

Habilitar Reservación

Figura 3.50 Formulario para editar una reservación.

Como la actualización del registro de una reservación fue exitoso, en la figura 3.51 se verifica en la base de datos el registro actualizado.

id	name	description	date	time	_escenario	_user	state
5	REUNION COVID19	RESERVACIO EDITADA	2021-01-01 00:00:00	2	1	1	t

Figura 3.51 Registro actualizado de una reservación en la base de datos

Finalmente, si el usuario desea eliminar una reservación la aplicación web muestra un mensaje de confirmación como indica la Figura 3.52. Si la respuesta es afirmativa borra completamente el registro de la reservación caso contrario lo mantiene.

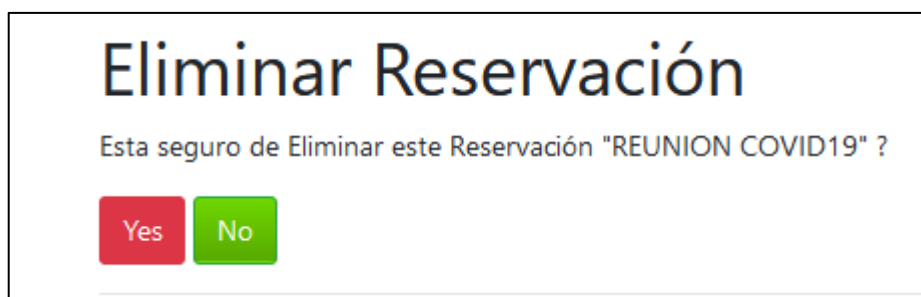


Figura 3.52 Mensaje de confirmación para eliminar una reservación.

3.3 PRUEBAS DE VALIDACIÓN

Las pruebas de validación del prototipo de aplicación web se realizaron mediante cuestionarios de satisfacción cerrados a diferentes usuarios. Los cuestionarios fueron aplicados de la siguiente manera: 3 personas pertenecientes a la Liga Cantonal, que usaron la aplicación con el tipo de usuario Administrador y Dirigente y a 3 personas que utilizaron como Asistente.

Los modelos de cuestionarios se encuentran en el ANEXO F.

En la Tabla 3.1 se presentan los resultados del cuestionario de satisfacción a usuario de tipo Administrador.

Tabla 3.1. Resultados del cuestionario de satisfacción a usuario Administrador

Nº	Pregunta	Respuesta	
		SI	NO
1	¿Pudo ingresar a la aplicación web con el nombre de usuario aquiles y la contraseña paradise?	100%	0%
2	La aplicación web, ¿le permitió agregar un torneo?	100%	0%
3	La aplicación web, ¿le permitió agregar un evento?	100%	0%
4	La aplicación web, ¿le permitió agregar una reservación?	100%	0%
5	La aplicación web, ¿le permitió agregar un usuario nuevo?	100%	0%
6	La aplicación web, ¿le permitió modificar la información de un torneo?	100%	0
7	La aplicación web, ¿le permitió modificar la información de un evento?	100%	0

8	La aplicación web, ¿le permitió modificar la información de un usuario?	100%	0%
9	La aplicación web, ¿le permitió modificar la información de una reservación?	100%	0%
10	La aplicación web, ¿le permitió eliminar un usuario?	100%	0%
11	La aplicación web, ¿le permitió eliminar una reservación?	100 %	0%

La pregunta uno, tenía como objetivo comprobar el control de acceso a la aplicación, arrojando como resultado que el total de los entrevistados pudo ingresar con las credenciales entregadas.

La pregunta dos validó, que la información de un torneo nuevo se almacenó de manera correcta.

La pregunta tres comprobó, que la información de un evento nuevo se almacenó de manera correcta asociando a un torneo, por lo que el total de los entrevistado pudo hacerlo.

La pregunta cuatro validó, que la información de una reservación nueva se almacenó de manera correcta.

La pregunta cinco busco probar, que la información de un usuario nuevo se almacenó de manera correcta ya que el 100% de encuestados lo realizó con éxito.

La pregunta seis, se comprobó, que la funcionalidad de la aplicación para ejecutar cambios en la información es correcta.

La pregunta siete, validó que se puede actualizar la información de los eventos ya que todos los entrevistado lograron hacerlo.

La pregunta ocho, se comprobó que la información de los usuarios se puede actualizar si así lo requiere el administrador, ya que el 100% de los encuestados logró realizar esta tarea.

La pregunta nueve, muestra que el 100% de los encuestados pueden realizar esta acción lo que garantiza que el sistema funciona de la manera correcta.

La pregunta diez indica que todos los encuestados lograron eliminar usuario, esto prueba que si el administrador desea puede eliminar los usuarios, de acuerdo a las necesidades de la Liga Cantonal.

La pregunta once indica que se pudo eliminar sin ningún inconveniente las reservaciones así se garantiza que se elimine cuando ya no se desee la reservación.

La Tabla 3.2 presenta los resultados del cuestionario de satisfacción que se aplicó a las personas con el rol de Dirigente.

Tabla 3.2. Resultados del cuestionario de satisfacción a usuario Dirigente

Nº	Pregunta	Respuesta	
		SI	NO
1	¿Pudo ingresar a la aplicación web con el nombre de usuario folmedo y la contraseña paradise?	100%	0%
2	La aplicación web, ¿le permitió ver los torneos?	100%	0%
3	La aplicación web, ¿le permitió ver los eventos?	100%	0%
4	La aplicación web, ¿le permitió agregar los deportistas?	100%	0%
5	La aplicación web, ¿le permitió agregar eventos a su agenda?	100%	0%
6	La aplicación web, ¿le permitió eliminar los eventos de su agenda?	100%	0
7	La aplicación web, ¿le envió la notificación de un evento 30 minutos antes de iniciarse?	100 %	0
8	La aplicación web, ¿le permitió ver los reportes?	100%	0
9	La aplicación web, ¿le permitió ver las estadísticas?	100%	0%
10	La aplicación web, ¿le permitió crear de una reservación?	100%	0%
11	La aplicación web, ¿le permitió modificar la información de una reservación?	100%	0%
12	La aplicación web, ¿le permitió eliminar una reservación?	100 %	0%

La pregunta uno, tenía como objetivo comprobar el control de acceso a la aplicación, arrojando como resultado que el total de los entrevistados pudo ingresar con las credenciales entregadas.

La pregunta dos busca validar que el usuario pueda visualizar los torneos, los resultados muestran que el total de los encuestados pudo realizar esta tarea.

La pregunta tres valida que el usuario pueda leer los eventos en el sistema, los resultados muestran que el 100% de los entrevistados pudo ver los eventos.

La pregunta cuatro valida si el dirigente puede crear los deportistas de su delegación, los resultados indica que la totalidad de los encuestado logró realizar esta tarea.

Los resultados de la pregunta cinco muestran que todos los encuestados pudieron agregar eventos a su agenda.

Los resultados mostrados en la pregunta seis indican que los usuarios en su totalidad pudieron eliminar los eventos que yo no son de su interés.

Los resultados de la pregunta siete garantiza que los usuarios recibieron la notificación del evento próximo.

La pregunta ocho valida si los usuarios pueden ver los reportes, los resultados obtenidos indican que el 100% de los encuestados pudo acceder a los reportes.

La pregunta nueve comprueba que los usuarios puedan acceder y ver los medalleros y tablas de posiciones de los torneos ya que el 100% de los encuestados logró realizar esta tarea.

La pregunta diez arroja como resultado que el 100% de los encuestados creo una reservación nueva.

La pregunta once garantiza que un usuario puede modificar la información de una reservación propia ya que todos los encuestados lograron cumplir con esta tarea.

La pregunta doce valida que el usuario pueda eliminar una reservación propia ya que todos los encuestados lo hicieron.

La Tabla 3.3 presenta los resultados del cuestionario de satisfacción que se aplicó a las personas con el rol de Asistente.

Tabla 3.3. Resultados del cuestionario de satisfacción a usuario Asistente

Nº	Pregunta	Respuesta	
		SI	NO
1	¿Pudo ingresar a la aplicación web con el nombre de usuario pperlaza y la contraseña paradise?	100%	0%
2	La aplicación web, ¿le permitió ver los torneos?	100%	0%
3	La aplicación web, ¿le permitió ver los eventos?	100%	0%

4	La aplicación web, ¿le permitió agregar eventos a su agenda?	100%	0%
5	La aplicación web, ¿le permitió eliminar los eventos de su agenda?	100%	0
6	La aplicación web, ¿le envió la notificación de un evento 30 minutos antes de iniciarse?	100 %	0
7	La aplicación web, ¿le permitió ver los reportes?	100%	0
8	La aplicación web, ¿le permitió ver las estadísticas?	100%	0%
9	La aplicación web, ¿le permitió crear de una reservación?	100%	0%
10	La aplicación web, ¿le permitió modificar la información de una reservación?	100%	0%
11	La aplicación web, ¿le permitió eliminar una reservación?	100 %	0%

La encuesta de satisfacción del Asistente es igual a la del Dirigente con excepción de la creación de deportistas y los resultados arrojados en todas las preguntas son iguales ya que el 100% de los encuestados logró realizar cada una de las tareas indicadas en el cuestionario.

3.4 TABLERO KANBAN FINAL

Finalizadas todas las etapas de Diseño, Implementación y Pruebas de este Proyecto Técnico se presenta el Tablero Kanban Final en la Figura 3.

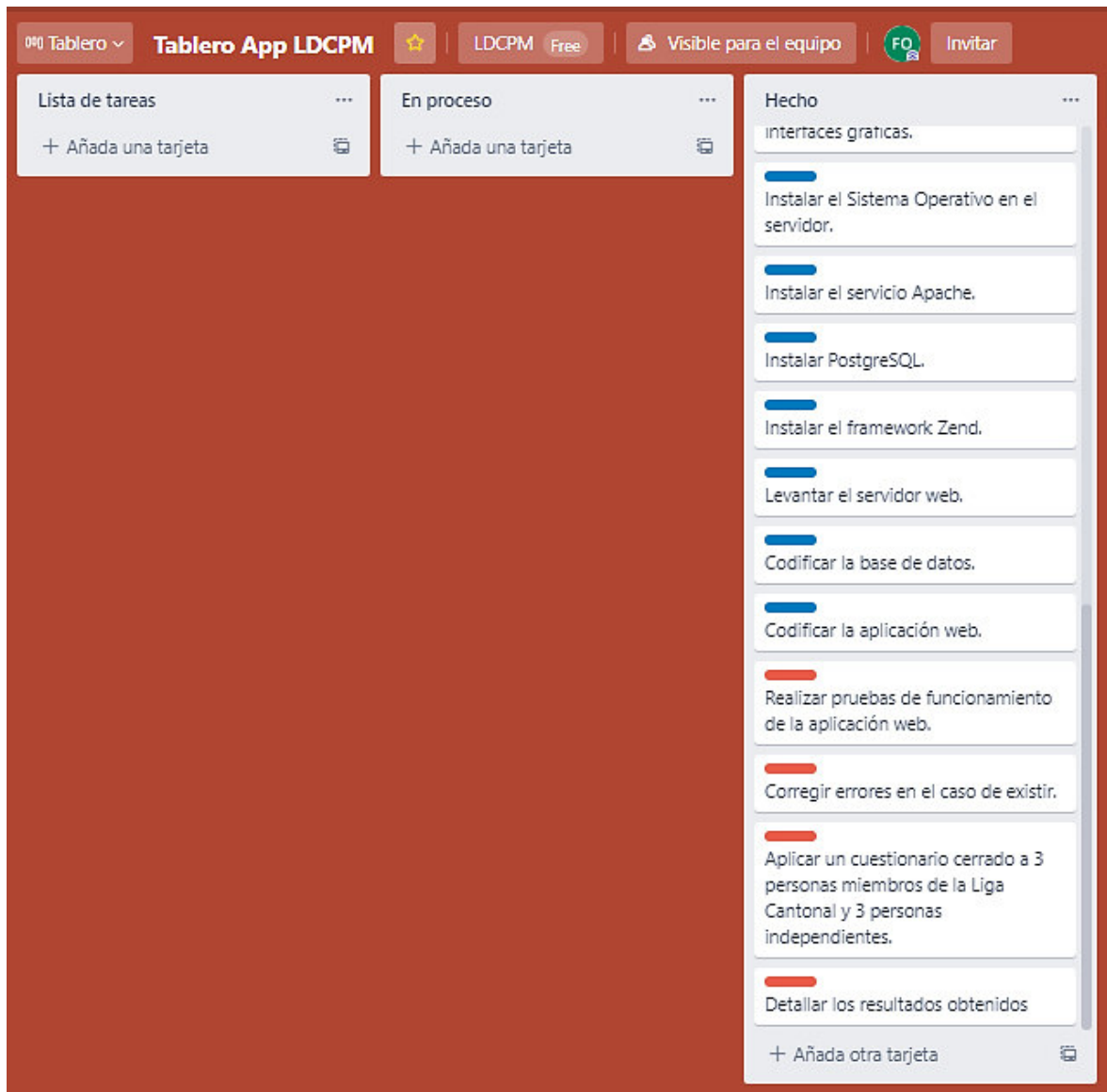


Figura 3. Tablero Kanban Final.

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- Se desarrolló un prototipo de aplicación web para la gestión de torneos de la Liga Deportiva Cantonal Pedro Moncayo. Esta aplicación permite una mejora considerable en la forma actual de gestionar los torneos, solucionando de esta manera los problemas de pérdida de documentación, el traslado a las instalaciones para recibir información, así como el acceso fácil a los registros de la gestión de los torneos y las reservaciones.
- Se analizaron lenguajes y herramientas de programación que se utilizaron para el desarrollo de este prototipo. Se utilizó PHP y HTML como lenguajes de programación para la aplicación web, lo cuales son importantes y muy usados en el desarrollo de dichas aplicaciones.
- Se establecieron los requerimientos funcionales y no funcionales mediante entrevistas a miembros del directorio de la Liga Deportiva Cantonal Pedro Moncayo y también en base a mi conocimiento como miembro activo de la Liga, en la cual se determinó las falencias que se tenía al gestionar un evento de manera manual.
- Se implementó la aplicación web con la ayuda de Zend Framework el cual es un *framework* robusto que ofrece escalabilidad, ya que posee filtros para seguridad, una gran variedad de componentes y organiza la programación por módulos, lo que facilita el desarrollo de cualquier aplicación. Además, que Zend es uno de los *frameworks* más usados en el mundo.
- Se utilizó el paradigma de programación modelo, vista, controlador para el desarrollo del prototipo de aplicación web, lo que facilita realizar mejoras o escalar la aplicación de una manera más ordenada y específica.
- Se utilizó la metodología ágil Kanban para organizar las tareas de cada una de las etapas del desarrollo del prototipo. Esta metodología ayudó a gestionar las tareas gracias a su facilidad visual.
- Se implementó una base de datos relacional en PostgreSQL definiendo relaciones basadas en las consultas a realizar, mismas que se facilitaron gracias a la librería Zend-db.
- Las pruebas de funcionamiento de la aplicación se realizaron en dos de los navegadores más usados que son Google Chrome y Mozilla Firefox, en donde

corren sin ningún inconveniente. Además, que dichas pruebas se realizaron en ambiente de pruebas de manera controlada y en producción con el servidor publicado a la internet.

- Los resultados obtenidos en la fase de pruebas permitieron verificar que el funcionamiento del prototipo sea el adecuado y así cumplir con los Requerimientos Funcionales y no Funcionales.

4.2. RECOMENDACIONES

- Se recomienda implementar una aplicación móvil para sistemas operativos iOS y Android para expandir el alcance del proyecto y transformar la aplicación web en un sistema distribuido.
- Se recomienda como futuras actualizaciones implementar un rol específico para los vocales asignados a cada uno de los eventos deportivos. Actualmente no se tiene dicho rol ya que se ha considerado que este cargo lo van a asumir miembros certificados de la Liga Cantonal. También como actualización se puede implementar la geolocalización para que los asistentes o fanáticos interesados puedan tener la ubicación exacta en donde se van a desarrollar los eventos de su preferencia. Otra actualización muy útil que se puede implementar es el módulo de los encuentros en vivo en donde se pueda presenciar el minuto a minuto de cada evento deportivo.
- En vista de que el servidor fue publicado en la internet, se recomienda implementar seguridad a nivel de servidor como un firewall para evitar ataques de cualquier tipo.
- Se recomienda usar para cualquier tipo de proyecto la metodología Kanban ya que tiene el control de las tareas de manera ordenada que logra culminar el proyecto, es sumamente visual y didáctica.
- Se recomienda realizar el correcto dimensionamiento del servidor para que la aplicación pueda correr sin ningún tipo de inconvenientes tomando en cuenta la escalabilidad y la demanda de usuarios.
- Acorde a la realidad en la que vivimos se recomienda tener las aplicaciones y servicios en la nube ya que estas garantizan una alta disponibilidad, tomando en cuenta el giro del negocio.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] J. J. Gutiérrez Cañizares, UF1271 - Instalación y configuración del software de servidor web, Elearning, S.L., 2015.
- [2] D. Álvarez Suárez, «e-Reading,» [En línea]. Available: <http://bibing.us.es/proyectos/abreproy/10967/fichero/Memoria+por+capitulos%252F5-Descripcion+del+protocolo+HTTP.pdf>. [Último acceso: 9 Septiembre 2020].
- [3] C. Date, Introducción a los sistemas de bases de datos, Pearson Educación, 2001.
- [4] S. Luján Mora, Programación en Internet: Cliente web, San Vicente: Club Universitario, 2001.
- [5] P. Mariuxi Zea Ordóñez, J. Molina Ríos y F. F. Redrován Castillo, ADMINISTRACIÓN DE BASES DE DATOS CON POSTGRESQL, 3Ciencias, 2017.
- [6] M. P. Zea Ordóñez, J. R. Molina Ríos y F. F. Redrován Castillo, ADMINISTRACIÓN DE BASES DE DATOS CON POSTGRESQL, Alicante: Area de Desarrollo e Invovación, S.L, 2017.
- [7] T. P. G. D. Group, «PostgreSQL,» [En línea]. Available: <https://www.postgresql.org/>. [Último acceso: 27 Enero 2020].
- [8] J. M. Vara Mesa, M. López y J. Verde Marin, Desarrollo web en entorno servidor, Madrid: RA-MA S.A, 2014.
- [9] A. Ramos Martín y M. J. Ramos Martín, Aplicaciones Web, Madrid: Paraninfo SA., 2014.
- [10] J. J. Tortajada Cordero, Cómo construir documentos XML: DTD y Esquema XML, 2014.
- [11] I. Lee, Selected Readings on Information Technology and Business Systems Management, IGI Global, 2008.
- [12] J. Tyler y D. Chappell, Java Web Services, United States of America: O'Reilly Media, Inc., 2002.
- [13] A. Contributors, «Zend Framework,» [En línea]. Available: <https://framework.zend.com/>. [Último acceso: 27 Enero 2020].
- [14] A. Omelak, Zend Framework 3. Developer's Guide, CreateSpace Independent Publishing Platform, 2018.
- [15] O. Krivtsov, «Usinf Zend Framework 3,» 5 Octubre 2018. [En línea]. Available: <https://olegkrivtsov.github.io/using-zend-framework-3-book/html/>. [Último acceso: 9 Septiembre 2020].
- [16] D. karaivanov, C. Stalev y I. Bisser, «Kanbanize,» 2020. [En línea]. Available: <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban>. [Último acceso: 1 Septiembre 2020].

- [17] D. d. L. C. P. Moncayo, «Presupuesto de Inversion Deportiva,» Tabacundo, 2009.
- [18] Atlassian, «Trello,» Atlassian, [En línea]. Available: <https://trello.com/>. [Último acceso: 10 Octubre 2020].
- [19] L. S. Inc, «Lucidchart,» Lucid Software Inc, [En línea]. Available: https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion#discovery__top. [Último acceso: 15 Octubre 2020].
- [20] P. Software, «Valentina,» Paradigma Software, [En línea]. Available: <https://www.valentina-db.com/>. [Último acceso: 09 Septiembre 2020].
- [21] I. I. E. S.L.U, «Ionos,» 1&1 IONOS España S.L.U, 2020. [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagrama-de-casos-de-uso/>. [Último acceso: 1 Diciembre 2020].
- [22] M. P. o. G. Framework, «DiagramasUML,» Magazine Pro on Genesis Framework , [En línea]. Available: <https://diagramasuml.com/diagrama-de-clases/>. [Último acceso: 12 Diciembre 2020].
- [23] G. Alder, «Draw.io,» Septiembre 2016. [En línea]. Available: <https://app.diagrams.net/>. [Último acceso: 15 Octubre 2020].
- [24] NinjaMock, «NinjaMock,» 2016. [En línea]. Available: <https://ninjamock.com/>. [Último acceso: 18 Octubre 2020].
- [25] GitHUB, «Atom,» GitHub Inc, 26 Febrero 2017. [En línea]. Available: <https://atom.io/>.
- [26] L. Carvajal, Metodología de la Investigación Científica. Curso general y aplicado, 28 ed., Santiago de Cali: U.S.C., 2006, p. 139.
- [27] T. A. S. Foundation, «The Apache Software Fundation,» [En línea]. Available: <https://www.apache.org/>. [Último acceso: 25 Enero 2020].

ANEXOS

ANEXO A. Preguntas de las entrevistas para la toma de requerimientos.

ANEXO B. Tablas detalladas de la base de datos y Diagrama Entidad Relación.

ANEXO C. Clases detalladas del diagrama de clases.

ANEXO D. Código de la base de datos.

ANEXO E. Código de la aplicación web.

ANEXO F. Cuestionarios de satisfacción de la aplicación web.

Nota: Todos los anexos se encuentran adjuntos en el CD.

ORDEN DE EMPASTADO