ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNOLÓGOS

DESARROLLO DEL SISTEMA WEB DE GESTIÓN DE CAMPEONATOS DE FÚTBOL LIGA LOMA DE PUENGASÍ

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN ANÁLISIS DE SISTEMAS INFORMÁTICOS

EDWIN DAVID CASA VELASCO
edwin.casa01@epn.edu.ec

Directora: ING. IVONNE FERNANDA MALDONADO SOLIZ, MSc.
ivonne.maldonadof@epn.edu.ec

Codirectora: ING. LUZ MARINA VINTIMILLA JARAMILLO, MSc.
marina.vintimilla@epn.edu.ec

Quito, marzo 2020

DECLARACIÓN

Yo, Casa Velasco Edwin David, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 114 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación -COESC-, soy titular de la obra en mención y otorgo una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional. Entregaré toda la información técnica pertinente. En el caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.

CASA VELASCO EDWIN DAVID

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por el señor Casa Velasco Edwin David, con cédula de identidad 1752284933, bajo nuestra supervisión.

Ing. Ivonne Maldonado, MSc.	Ing. Luz Marina Vintimilla, MSc.
DIRECTORA	**COODIRECTORA**

AGRADECIMIENTO

Primero, agradezco a Dios por brindarme sabiduría, fuerza, valor y perseverancia para no darme por vencido y lograr cumplir este escalón importante en mi vida.

A mi madre Mercedes y a mi padre Segundo, los cuales son un pilar fundamental en mi vida, y fueron muy importantes en esta etapa, ya que siempre estuvieron brindándome su apoyo de manera incondicional.

A mis hermanos y hermanas, por estar siempre conmigo apoyándome y sin permitir que decaiga y abandone mis sueños.

A mis amigos, con quienes aprendí, compartí tanto tristezas como alegrías, personas gratas en mi vida, ya que supieron brindarme motivación y apoyo a lo largo de este proceso.

Finalmente, doy gracias a todos los profesores que me supieron brindar su conocimiento, en especial agradezco a mi directora de tesis que fue una gran ayuda para hacer posible este proyecto.

EDWIN DAVID CASA VELASCO

DEDICATORIA

Dedico este trabajo a todas las personas que creyeron en mí, en especial a mi familia, ya que, ellos son mi motivación para seguir adelante y con su apoyo, motivación incondicional me ayudan a ser mejor cada día y a no darme por vencido en cada una de las metas que me propongo.

EDWIN DAVID CASA VELASCO

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

La implementación de un sistema Web en el cual se almacene, controle y gestione la información de los diferentes campeonatos de fútbol realizados, ayudará a la directiva de La liga Barrial Loma de Puengasí a mantener la gran cantidad de datos que se obtiene de cada campeonato de fútbol de forma ordena y sistematizada, proporcionando a los implicados información confiable, de esta manera se evitará el almacenamiento masivo de información en documentos físicos, facilitando la obtención de manera información específica de una manera sencilla y rápida.

El presente trabajo de titulación corresponde al desarrollo de un sistema web de gestión de campeonatos de fútbol para la Liga Deportiva Loma de Puengasí, llamada Sysfut, aplicando la metodología *Extremme Programming* (XP), que, a través de las diferentes vistas, permitirá el almacenamiento de la diferente información acerca de los campeonatos, mediante un Sistema Gestor de Base de Datos (SGBD) PostgreSQL. El procesamiento de la información se lo realizará mediante una APIREST desarrollado con el microframework Flask, el cual se comunicará mediante un Mapeo Relacional de Objetos (ORM) implementado con SQLAlchemy. EL producto final, facilitará la creación de campeonatos de manera rápida, eficiente y segura, mediante la automatización de los diferentes procesos presentes en cada evento deportivo, el cual es organizado anualmente por la mencionada liga deportiva.

Palabras claves: Extreme Programming (XP), Sistema Web, campeonatos de fútbol, PostrgreSQL, APIREST, Flask, ORM, SQLAlchemy.

ABSTRACT

The implementation of a web system in which the information of the different soccer championships will be stored, controlled and managed, will help the management of La Liga Barrial Loma de Puengasí *to maintain the large amount of data obtained from each soccer championship In an orderly and systematized way, providing those involved with reliable information, in this way the massive storage of information in physical documents will be avoided, facilitating the obtaining of specific information in a simple and fast way.*

The present degree work corresponds to the development of a web system for the management of soccer championships for the Loma de Puengasí Sports League, called Sysfut, applying the Extremme Programming (XP) methodology, which, through the different views, will allow storage of the different information about the championships, through a PostgreSQL Database Management System (DBMS). The information will be processed through an APIREST developed with the Flask microframework, which will be communicated through a Relational Object Mapping (ORM) implemented with SQLAlchemy. The final product will facilitate the creation of championships quickly, efficiently, and safely, by automating the different processes present in each sporting event, which is organized annually by the aforementioned sports league.

Keywords: *Extreme Programming (XP), Web System, soccer championships, PostrgreSQL, APIREST, Flask, ORM, SQLAlchemy.*

1. INTRODUCCIÓN

Las Ligas Barriales congregar a diferentes equipos de fútbol a ser partícipes de los eventos deportivos que se celebran anualmente. Cada Liga Barrial organiza y celebra campeonatos con un reglamento aprobado por la filial unión de ligas. La organización de dichos eventos actualmente se los realiza mediante el registro de la información de manera física en actas, documentos; los cuales generan problemas por la falta de acceso y difusión lenta de la información por parte de los representantes de cada equipo y la directiva de la liga deportiva.

1.1 Planteamiento del problema

La Liga Deportiva Loma de Puengasí anualmente realiza eventos deportivos, pueden participar equipos de diferentes localidades, los cuales no posean sanciones o adeuden en las diferentes ligas filiales que existen en la ciudad de Quito. Este evento deportivo es organizado por la directiva de la mencionada liga, que se encarga de coordinar la participación de todos los clubes afiliados y que han sido aceptados por el directorio ampliado de la liga.

En la actualidad la Liga Deportiva Loma de Puengasí no cuenta con un sistema web para la creación de campeonatos de fútbol, cada campeonato desarrollado se lo realiza manualmente, almacenando la información en documentos físicos; cada campeonato se lo inicia con un reglamento de competencia que maneja cada liga filial.

El presidente de la Liga Deportiva Loma de Puengasí supo manifestar que el registro de la información de cada partido se lo puede visualizar, después de la reunión o sesión que mantiene en dicha entidad. Todos los lunes mientras dure el campeonato, por lo cual la información de cada partido no se puede evidenciar instantáneamente cuando finaliza cada partido; además manifestó que las diferentes tablas informativas que gestiona la Liga Deportiva Loma de Puengasí, solamente las pueden visualizar los directivos y presidentes de cada equipo, la información no se hace pública debido a la falta de un medio de comunicación existente entre la entidad y los diferentes participantes.

Se conoció que las sanciones son aplicadas de forma manual a cada jugador, esto se refiere a que por cada partido jugado, se registra toda la información en una hoja de vocalía, la cual debe ser leída con atención para así proceder a registrar y aplicar las sanciones correspondientes, lo cual genera el problema, ya que en ocasiones el encargado de la comisión mencionada no asiste a la sesión semanal, por lo cual no se

pueden registrar las sanciones y en ocasiones la información deja de ser integra y existen confusiones que genera sanciones adicionales a los participantes.

A través de una entrevista con un participante de la Liga Deportiva Loma de Puengasí se pudo constatar que en ocasiones si un representante no asiste a una sesión, este no se puede informar de la fecha y hora en la que se programó su partido, ya que no existe un medio de comunicación externo a la sesión, deben averiguar a los diferentes participantes, para así poder asistir a su encuentro y a su obligación otorgada en cada sesión.

1.2 Objetivo general

Desarrollar un sistema web de gestión de campeonatos de fútbol de la Liga Deportiva Loma de Puengasí.

1.3 Objetivos específicos

- OBJ1: Determinar los requisitos del Sistema Web.
- OBJ2: Diseñar el modelo de base de datos e interfaces de usuario para el Sistema Web.
- OBJ3: Construir el Sistema Web a partir del OBJ1 y OBJ2.
- OBJ4: Probar la funcionalidad y lógica del Sistema Web

1.4 Alcance

El sistema web ayudará a la creación rápida y correcta de las fechas a jugarse en el campeonato, las sanciones de cada jugador, la cantidad de tarjetas amarillas y rojas, ya que esto se tomará en cuenta para generar una alerta, de que a un jugador se le ha aplicado una sanción, también se implementará la lista de jugadores por equipo, los cuales podrán ser visualizados con facilidad por la directiva de la Liga Deportiva Loma de Puengasí, y por los diferentes presidentes de los clubs de fútbol pertenecientes a esta liga. La implementación tiene la finalidad de evitar gastos innecesarios en procesos, los cuales se automatizarán con la implementación del sistema web. Además, se obtendrá la información de manera actualizada e inmediata, cada vez que finalice un encuentro, evitando esperas prolongadas para saber resultados y sanciones aplicadas a los participantes de los encuentros.

Para garantizar la confiabilidad e integridad de la información el sistema web contará con un control de acceso a cada usuario, con un proceso de verificación de la autenticación y la autorización del usuario para acceder a los diferentes recursos e información expuesta por el sistema. Además de llevar un control de cada proceso

realizado por usuario, mediante archivos en texto plano que contendrán información descriptiva de lo que realiza cada uno.

Los diferentes usuarios que se manejarán en el sistema serán:

- El usuario invitado, el cual podrá visualizar información descriptiva de la liga sin necesidad de autentificación.
- El usuario administrador, el cual se encargará de otorgar permisos de acceso, creación de usuarios.
- El usuario presidente, el cual será el encargado de mantener la página con información actualizada; además, podrá modificar, actualizar y visualizar la información del sistema web.
- El usuario delegado será el encargado de registrar el equipo y sus jugadores en la disciplina correspondiente, mismo que podrá visualizar y modificar información del equipo que registro en el sistema.

Ya registrados los equipos, cada usuario participante podrá visualizar el reglamento de competencia del año vigente, solicitando la aceptación de dicho documento al usuario. Una vez completado el registro se deberá esperar la aceptación del registro por parte de la directiva de la liga.

Una vez con el registro de 7 o más equipos registrados el sistema procederá a la creación automática de la programación por fechas, los horarios se los establecerá en las sesiones que se realizarán dependiendo lo establecido en el reglamento.

Finalmente, el sistema permitirá la generación de diferentes reportes en formato Excel y estará disponible en el sistema Web.

1.5 Estructura del documento

Para el desarrollo del sistema web se utilizó la metodología de desarrollo ágil Extreme Programming (XP), misma que proporciona retroalimentación continua entre el cliente y el desarrollador, mediante entregables funcionales del sistema.

El presente documento se encuentra dividido en cuatro secciones que se detallan a continuación:

En la sección I se realiza la introducción, se especifica, el objetivo general, los objetivos específicos, el alcance del proyecto y la justificación del porqué se realiza el desarrollo del sistema web.

En la sección II se detalla el uso de la metodología de desarrollo, se describe el análisis y levantamiento de requerimientos, la arquitectura del sistema, el diseño del modelo la base de datos y las herramientas que se utilizaron para el desarrollo del sistema web.

En la sección III se detalla cómo se llevó a cabo el desarrollo del sistema Web, además de la presentación de los resultados obtenidos de las pruebas realizadas.

Finalmente, en la sección VI se presentan las conclusiones y recomendaciones obtenidas en el proceso de desarrollo del proyecto.

2. METODOLOGÍA

En esta sección se detalla la metodología utilizada como base para el desarrollo del sistema web *Extreme Programming*.

2.1. Metodología de desarrollo Extreme Programming (XP)

El sistema web para la gestión de campeonatos de fútbol de la Liga Deportiva Loma de Puengasí fue desarrollada mediante la metodología ágil de ingeniería de software *Extreme Programming (XP)*, que proporciona retroalimentación continua entre el cliente y el desarrollador, usada para el levantamiento de requisitos y corrección de errores. Cada proceso que se realiza está definido por iteraciones. Cada iteración consiste en cuatro etapas, las cuales son: análisis, desarrollo, diseño y pruebas. Al finalizar las etapas se generarán entregables funcionales, los cuales fueron analizados por el cliente quien proporcionó retroalimentación [1].

XP permitió la comunicación fluida entre los participantes del proyecto (directiva de la Liga Deportiva Loma de Puengasí y desarrollador), adaptabilidad a cambios en requerimientos que surgieron en el proceso de generación de entregables funcionales, simplicidad en las soluciones a los problemas que se planteen, solucionando dudas sin dificultad ya que se trabajó en paralelo con el cliente, calidad del software, debido a que, mediante el desarrollo guiado por pruebas se pudo verificar la funcionalidad antes de ser implementada. Además, la calidad en XP aplica el método *refactoring*[1], el cual facilitó el mantenimiento del código fuente, permitiendo detectar errores en algoritmos en menos tiempo [2].

2.1.1. Planificación (planning game)

A diferencia de la planificación del Sprint en Scrum, la dinámica de planificación de XP llevada a cabo al inicio de la iteración, suele ser de la siguiente manera [3]:

- El cliente define la lista de las funcionalidades deseadas para el sistema, las cuales son escritas con formato de “Historia de Usuario”, en estas se presentó la manera de gestionar la información de cada campeonato de futbol a celebrar.
- Estimar el esfuerzo que demanda desarrollar las historias de usuario, así como el esfuerzo disponible para el desarrollo (las iteraciones en XP

[1] **Refactoring:** proceso de reestructuración del código de computadora existente.

suelen durar 1 a 4 semanas), en este caso fueron estimadas de acuerdo con el nivel de complejidad.

- El cliente decide que Historias de Usuario desarrollar y en qué orden, manejadas por prioridad de negocio. Se seleccionó en qué orden se gestiona un campeonato para conjuntamente con el cliente establecer las prioridades a cada historia de usuario.

2.1.2. Comunicación *(on-site customer)*

En XP se requiere que el cliente esté dispuesto a participar activamente del proyecto, contando con la disponibilidad suficiente y necesaria, para interactuar con el equipo en todas las fases del proyecto [3]. La participación activa de los directivos de la Liga Deportiva Loma de Puengasí permitió que los procesos de levantamiento de requisitos, implementación y pruebas se lleven a cabo de forma efectiva.

La comunicación despejó dudas existentes en cada una de las iteraciones realizadas, logrando solucionar conflictos presentes en tiempo real acerca de la gestión de campeonatos mediante la colaboración del cliente. Las personas participantes en el desarrollo del software fueron usuarios reales (directiva de la Liga Deportiva Loma de Puengasí, presidentes de clubes deportivos, jugadores pertenecientes a clubes) los cuales son conocedores del tema, de la funcionalidad a obtener en el sistema y objetivos a alcanzar con la implementación de este.

2.1.3. Diseño

Esta práctica, deriva de un famoso principio técnico del desarrollo de software: *KISS*[2], que puede ser resumido “hacer lo mínimo indispensable, tan legible como sea posible” [4]. Principio utilizado en el diseño de las interfaces, permitiendo que estas sean amigables con el usuario, facilitando la usabilidad del sistema.

2.1.4. Pruebas

Entre las pruebas que XP propone se tiene [1]:

[2] **KISS2-“Keep it simple, stupid!”:** patrón de diseño de software que menciona que cualquier sistema va a funcionar mejor si se mantiene sencillo que si se vuelve complejo. Es decir que la sencillez tiene que ser una meta en el desarrollo y que la complejidad innecesaria debe ser eliminada.

- **Test Unitarios (TDD):** consisten en testear el código de manera unitaria(individual), mientras se va programando. La aplicación de TDD en el desarrollo proporcionó retroalimentación de los errores al momento del consumo de los diferentes métodos desarrollados.
- **Test de aceptación:** está enfocado a las pruebas de funcionalidad, es decir al comportamiento funcional del código, basada en criterios de aceptación establecida por el cliente para la verificación funcional del sistema, la cual ayudó a verificar errores en generación de procesos como fue el caso de la programación de partidos.

2.1.5. Entregables *(short releases)*

XP es una metodología que realiza entregas en breves lapsos de tiempo, incrementando pequeñas funcionalidades en cada iteración, permitiéndole al cliente tener una mejor experiencia con el software, debido a que lo que deberá probar como nuevo, será poco, y fácilmente asimilable, pudiendo sugerir mejoras de implementación más simples [4].

La presentación del producto en cada iteración ayudó a la mejora continua del software debido a que se hallaron fallos en funcionalidades que fueron solucionadas conforme las pruebas realizadas en cada entrega.

2.2. Análisis y levantamiento de requisitos

Parte fundamental para el desarrollo del software consiste en obtener la información sobre los requerimientos funcionales y requerimientos no funcionales del software.

En esta etapa se realizó reuniones con la directiva de la Liga Deportiva Loma de Puengasí, con el objetivo de recopilar la información necesaria para replicar cada uno de los procesos a realizar para la correcta gestión y administración de los campeonatos a realizar.

En base a la información recopilada se realizó las historias de usuario las cuales ayudaron a cubrir las necesidades planteadas por los usuarios mediante niveles de prioridad. Además, se estableció a partir de la información obtenida la infraestructura a utilizar para el desarrollo del sistema, el diseño de la base de datos, elaboración de prototipos de interfaces del sistema web y selección de las herramientas para el correcto desarrollo y despliegue del software.

2.3. Arquitectura del sistema web

El sistema web está basado en el modelo–vista–controlador (MVC), este modelo de estructura se fundamenta en la separación de cada uno de sus elementos, trabajando así uno indistintamente del otro, pero manteniendo una comunicación fluida entre componentes.

La Fig. 1 muestra el diagrama de la infraestructura del software con el fin de comprender el diseño a utilizar, su comunicación y su implementación.

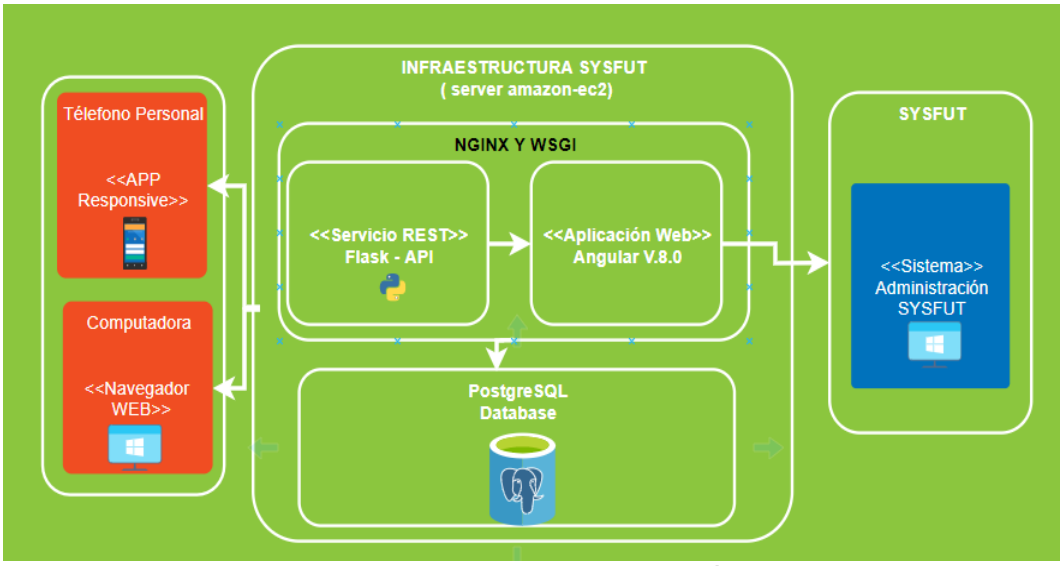

Fig. 1: Diagrama de distribución

La infraestructura implementada consiste en la comunicación de un APIREST entre angular y Flask, utilizando métodos GET y POST, y consumiendo información de la base de datos PostgreSQL (alojada en un servidor amazon-ec2), lo que permite que cada usuario pueda acceder desde un dispositivo móvil o una computadora al sistema web, asegurándole la disponibilidad, seguridad y fluidez de la información.

2.4. Diseño del modelo de la base de datos

El modelo entidad relación para el sistema web consta de 14 entidades, cada una de ellas con sus respectivos atributos. La base de datos se la modelo mediante el ORM[3] de SQLAlchemy, tal como muestra Fig. 2.

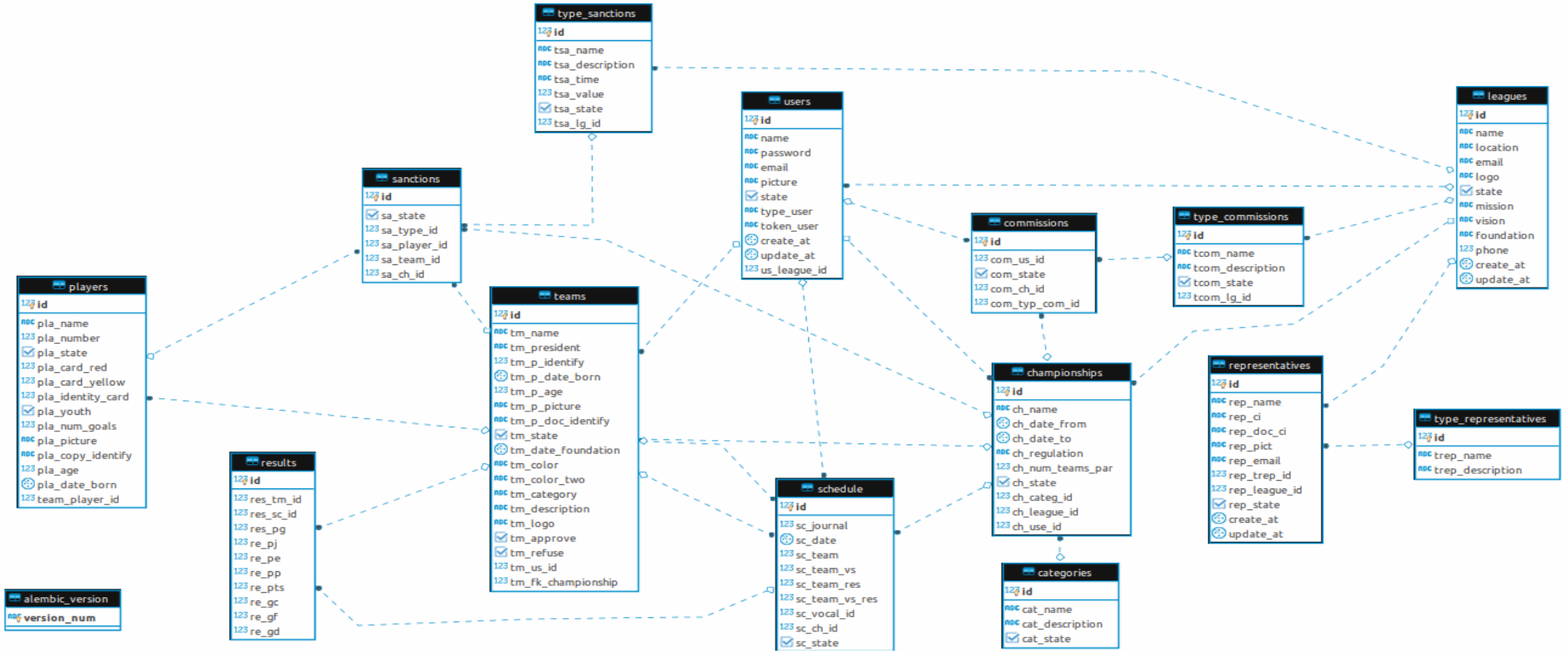

Fig. 2: Base de Datos

[3] **ORM:** Objeto para mapeo relacional.

2.5. Prototipo del sistema web

El prototipo del sistema web (Véase Manual Técnico - 5. Prototipo del Sistema) se elaboró de acuerdo con los requerimientos presentados por los usuarios. Para el modelado se hizo uso de la herramienta de creación de bocetos *NinjaMock,* la cual permitió la creación de los prototipos, facilitando la revisión de los conceptos básicos, necesidades del software en la etapa inicial del proyecto mediante la creación del maquetado del sitio web [5].

2.6. Herramientas para el desarrollo del sistema Web

2.6.1. Python

Es un lenguaje de programación interpretado de tipado dinámico cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma y disponible en varias plataformas, el cual proporciona diferentes recursos, gracias a ello se logró la implementación de las diversas funcionalidades como son: envió de correos electrónicos, tareas programadas, seguridad y generación de reportes [6].

2.6.2. MicroFramework Flask

Flask es un marco pequeño para la mayoría de los estándares, lo suficientemente pequeño como para ser llamado "microframework". El ser pequeño no significa que lo haga menos que otros marcos. Flask fue firmado como un marco extensible desde cero; proporciona un núcleo sólido con los servicios básicos, mientras que las extensiones proporcionan el resto. Posee dependencias principales. El enrutamiento, la depuración y la puerta de enlace del servidor web [7].

Por ello, este microframework ayudó en el retorno de información procesada para la posterior presentación en la plataforma web mediante un RESTAPI con el uso de la dependencia de enrutamiento para la generación de los diferentes tipos de métodos.

2.6.3. SQLAlchemy

SQLAlchemy es una biblioteca de Python creada por Mike Bayer, para proporcionar una interfaz de alto nivel a bases de datos relacionales como: Oracle, DB2, MySQL, PostgreSQL y SQLite. SQLAlchemy intenta ser discreto con su Código de Python, que le permite asignar objetos Python antiguos simples a tablas de bases de datos sin cambiar sustancialmente su código Python existente. SQLAlchemy incluye un

lenguaje de expresión SQL independiente del servidor de bases de datos y un mapeador relacional de objetos (ORM) que le permite usar SQL para persistir sus objetos de aplicación automáticamente [8].

Gracias a SQLAlchemy se facilitó la creación de tablas mediante objetos los cuales son manejados mediante el ORM, brindando seguridad a cada una de las consultas evitando inyecciones SQL mediante el formateo de la consulta, además proporcionó rendimiento y flexibilidad en cada consulta realizada para la abstracción de información necesaria.

2.6.4. Angular

Es un *framework*[4] de desarrollo web que utiliza *TypeScript*[5] y separa el *back-end*[6] del *front-end*[7]*.* Para que funcione de una manera sencilla y óptima utiliza directivas y atributos del vocabulario HTML sosteniendo la semántica, sin necesidad de emplear librerías externas como jQuery o Underscore.js [9].

Mediante *TypeScript* se generó la relación entre el APIREST y las vistas a ser presentadas con la información correspondiente obtenida del *back-end* de la aplicación.

2.6.5. PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacionales de código abierto que comenzó bajo la licencia BSD, pero ahora se llama la Licencia PostgreSQL (TPL). Para todos los efectos, es BSD con licencia [10].

Gracias a PostgreSQL el sistema web logra almacenar la información de manera segura, integra, ofreciendo disponibilidad y seguridad.

2.6.6. NGINX

Es un servidor web, proxy inverso y balanceador de carga *Open Source* que soporta a millones de páginas web en todo el mundo. Funciona de una forma no

[4] ***Frameworks*:** esquema o estructura que se establece y que se aprovecha para desarrollar y organizar un software determinado

[5] ***TypeScript*:** lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft.

[6] **B*ack-end:*** manipulación, procesamiento de datos.

[7] ***Front-end:*** presentación del producto de acuerdo con las especificaciones del cliente.

tradicional en comparación a los servidores web alternativos como Apache. Su arquitectura se enfoca más que nada de forma asincrónica, basándose enteramente en eventos, donde cada petición que viene desde un navegador web se maneja utilizando un solo hilo. Esto hace que use muy pocos recursos, sobre todo en cuanto a uso de memoria, gestándose así una plataforma para servir peticiones web liviana y muy rápida [11].

Este servidor web controla los múltiples procesos de trabajo que demanda el sistema web, gracias a ello al final se logra realizar el despacho de las solicitudes que vienen desde los clientes (navegadores web), es un clásico escenario de modelo cliente servidor.

2.6.7. WSGI

WSGI (Web Server Gateway Interface) es una interfaz de bajo nivel que permite la comunicación entre diferentes *frameworks* y servidores web, haciendo portables las aplicaciones entre ambos componentes [12].

En este caso, es quien permite la comunicación y salida a internet mediante el sistema web, de manera segura, trabajando juntamente con el servidor web NGINX para el procesamiento y retorno de información.

3. RESULTADOS Y DISCUSIONES

En esta sección se detalla los requerimientos funcionales que fueron implementados en el sistema web.

3.1. Requerimientos funcionales

Los requerimientos funcionales que muestra la TABLA I, son un conjunto de entradas, comportamientos y salidas, los cuales describen la interacción entre el sistema y su ambiente, independientemente de su implementación. El ambiente se refiere al usuario y a cualquier otro sistema externo que interactúe con el sistema.

TABLA I: Requerimientos funcionales del sistema web

ID	Requerimiento
RFSW001	Iniciar sesión, cerrar sesión
RFSW002	Página de presentación de la liga deportiva
RFSW003	Crear, visualizar, editar usuarios
RFSW004	Crear, visualizar, editar ligas deportivas
RFSW005	Crear, visualizar, editar equipos
RFSW006	Crear, visualizar, editar jugadores
RFSW007	Crear, visualizar, editar sanciones
RFSW008	Crear, visualizar, editar programación
RFSW009	Crear, visualizar, editar campeonatos
RFSW010	Crear, visualizar, editar resultados
RFSW011	Crear, visualizar, editar comisiones

3.1.1. Requerimientos no funcionales

Los requerimientos no funcionales que muestra la TABLA II, representan características generales y restricciones del sistema web.

TABLA II: Requerimientos no funcionales del sistema web

ID	Requerimiento no funcional
RNFSW001	El sistema debe contar con manuales de usuario estructurados adecuadamente.
RNFSW002	El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.
RNFSW003	La aplicación web debe poseer un diseño “Responsive”
RNFSW004	El sistema debe tener una disponibilidad del 99,99% de las veces en que un usuario intente accederlo.

3.2. Usuarios del sistema web

Existen cuatro tipos de usuarios detallados a continuación:

- **Administrador:** Este usuario posee todos los permisos sobre el sistema, puede eliminar, editar y actualizar toda la información que contiene el sistema web. Usuario que no pertenece a ninguna liga deportiva, es el primer usuario creado por el desarrollador, el cual proporciona las credenciales para la autenticación en el sistema.
- **Presidente:** Este usuario accede al sistema web mediante dos identificadores: correo electrónico y contraseña. Además, puede actualizar información relevante de la Liga Deportiva Loma de Puengasí, así como crear usuarios, crear campeonatos y modificar la información.
- **Delegado:** Este usuario podrá acceder el sistema con sus respectivas credenciales, las cuales son entregadas por el usuario presidente. Además, podrá crear, visualizar y modificar la información que haya registrado.
- **Invitado:** Este usuario no necesita de autenticación en el sistema ya que solamente podrá visualizar la información referente a los campeonatos deportivos sin posibilidad de modificarla.

3.3. Restricciones de desarrollo

- El sistema web está desarrollado en base a los procedimientos que se llevan a cabo en la Liga Deportiva Loma de Puengasí.
- Los usuarios delegados, presidente, y administrador deben autenticarse con sus respectivas credenciales para poder hacer uso de las diferentes funcionalidades del sistema.
- El usuario con perfil administrador será el encargado de administrar el sistema y dar solución a problemas presentes en cuanto a información.
- El usuario presidente será el presidente de la liga barrial el cual gestionará la información de los campeonatos.
- Los usuarios registrados deberán tener acceso al correo electrónico registrado debido a que este le permitirá activar la cuenta en el sistema.

3.4. Consideraciones

- El administrador del sistema web debe contar con previos conocimientos sobre administración de servidores y gestor de base de datos PostgreSQL.
- Para el uso del sistema web se deberá contar con una conexión a internet, ya que el sistema se aloja en un servidor Linux de Amazon y diversas funcionalidades requieren de este servicio.
- Los usuarios delegado y presidente deben tener conocimiento básico acerca del manejo de un computador.

3.5. Requerimientos específicos para el sistema web

En esta sección se detalla las consideraciones y funcionalidades de la línea de negocio desarrolladas.

3.5.1. Administración de usuarios y perfiles.

El usuario administrador asigna el rol de usuario presidente, el cual es el encargado del manejo de los perfiles de accesos y permisos al sistema.

3.5.2. Autenticación de usuarios

Los usuarios pueden iniciar sesión en el sistema con sus credenciales, una vez que hayan activado su cuenta mediante el enlace enviado al correo electrónico registrado. Las credenciales son entregadas por el usuario con perfil presidente:

- **Correo electrónico:** registrado al momento de la creación del usuario
- **Contraseña:** asignada por el usuario con perfil presidente

3.5.3. Restablecer contraseña

Los usuarios pueden restablecer su contraseña mediante un enlace que es enviado al correo electrónico registrado en el sistema.

3.5.4. Administración de campeonatos

El usuario con perfil presidente es el encargado de la creación y modificación de campeonatos. Solo puede existir un campeonato activo.

3.5.5. Administración de sanciones

El administrador asigna un usuario con perfil delegado para la creación y modificación de las sanciones, mismas que deberán estar basadas en el reglamento vigente del campeonato.

3.5.6. Administración de equipos y jugadores

El usuario con perfil delegado es el encargado de crear y, modificar equipos y jugadores (estén activos o inactivos). Cada delegado creará un equipo por campeonato el cual podrá ser editado únicamente por el mismo.

3.5.7. Generación de calendario de juegos

El sistema genera automáticamente el calendario de juegos en base a los requerimientos de la Liga Deportiva Loma de Puengasí.

3.6. Requerimientos de ambiente de desarrollo

Para el desarrollo del sistema web se utilizó el *framework* angular en su versión 8.0 basado en el lenguaje *typescript,* junto con el microframework Flask, basado en Python en su versión 3.6, además de hacer uso de SQLAlchemy como ORM.

3.7. Historias de usuario

Las historias de usuario, en XP, son una técnica para especificar los requisitos del software. Son tarjetas donde el cliente describe brevemente las características que el sistema debe poseer, las cuales son redactadas por el desarrollador y deben ser lo suficientemente comprensibles, y delimitadas para ser implementarlas en el menor tiempo posible [13].

En base a las reuniones realizadas, las necesidades y los procesos a cubrir con el software se generaron 35 historias de usuarios (Véase Manual Técnico – 4. Historias de Usuario) las cuales facilitaron el análisis y levantamiento de requerimientos para el respectivo desarrollo del sistema web. La TABLA III, muestra un ejemplo de las historias de usuarios generadas.

TABLA III: Historia de usuario Nro.15: Crear campeonato

	Historia de Usuario
Identificador: HU15	**Usuario:** presidente

Nombre historia: Crear campeonato.	
Prioridad en Negocio (Alto/Medio/Baja): Alta	**Riesgo en Desarrollo (Alto/Medio/Baja):** Media
Iteración asignada: 6	
Responsable: David Casa	
Descripción: El usuario con rol presidente será el encargado de la creación de los diferentes campeonatos, el cual constará con los campos de nombre, fecha inicio, fecha fin, reglamento, estado, categoría. Además de tener la opción de ingresar las diferentes comisiones.	
Observación: Para el ingreso de las comisiones se debe tener el campeonato creado y con equipos disponibles, ya que las comisiones las integran los equipos.	

3.8. Desarrollo del sistema

En esta sección se detalla la configuración previa para el despliegue del sistema web a un ambiente de desarrollo.

3.8.1. Configuración del ambiente de desarrollo

Como primer paso, se instaló Python en su versión 3.6, angular cli y Flask, cada una de estas herramientas fueron utilizadas en el proceso de desarrollo y pruebas en el sistema web. Además, se instaló el gestor de paquetes de Python, Pip[8] el cual permitió la instalación de paquetes de PyPI[9] necesarios para el desarrollo del proyecto. PostgreSQL como sistema gestor de base de datos, utilizado tanto en la etapa de desarrollo como en el de pruebas.

3.8.2. Creación del sistema web

En primer lugar, se realizó la creación de un nuevo proyecto en angular mediante el uso del comando ng new my-dream-app, mediante angular cli, como muestra la Fig. 3.

[8] **Pip:** sistema de gestión de paquetes.

[9] **PyPI:** repositorio de software para el lenguaje de programación Python.

```
> ng new my-dream-app
```

Fig. 3: Comando para crear un nuevo proyecto en angular

Para levantar el proyecto se ejecutó el comando ng serve en la ubicación del proyecto, lo que iniciará el servidor en la dirección http://localhost:4200 como muestra la Fig. 4.

```
PS C:\Users\Hp\Desktop\TESIS\software\futgol> ng serve
Browserslist: caniuse-lite is outdated. Please run next command `npm update`
 10% building 3/3 modules 0 activei [wds]: Project is running at http://localhost:4200/webpack-dev-server/
i [wds]: webpack output is served from /
i [wds]: 404s will fallback to //index.html

chunk {main} main.js, main.js.map (main) 722 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 251 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.09 kB [entry] [rendered]
chunk {scripts} scripts.js, scripts.js.map (scripts) 106 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 3.25 MB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 15.8 MB [initial] [rendered]
Date: 2020-02-11T07:18:27.964Z - Hash: 27f932f3c00a8f1b0461 - Time: 62655ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
i [wdm]: Compiled successfully.
```

Fig. 4: Comando para ejecutar proyecto angular

3.8.3. Estructura del proyecto Angular

Angular crea una estructura de directorios, organizando la información en base al modelo – vista - controlador (MVC), ordenando así los diferentes componentes a desarrollar, la estructura creada se puede observar en Fig. 5.

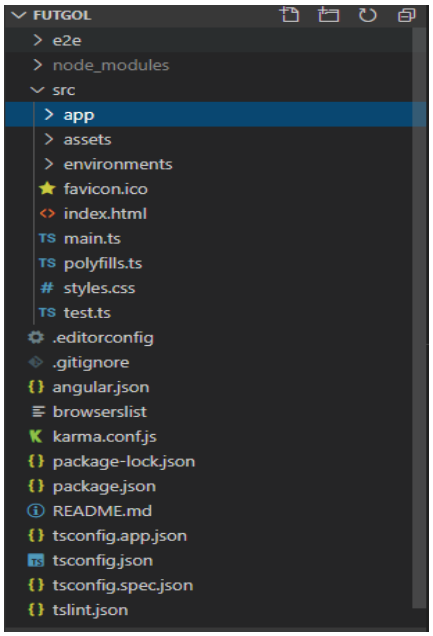

Fig. 5: Estructura de proyecto angular

- **Directorio src:** aloja toda la lógica, vistas y controladores del proyecto.
- **Directorio app:** contiene toda la información en cuanto a los componentes y las vistas.

- **Directorio assets:** aloja el contenido multimedia a usar directamente en la aplicación.
- **Directorio node_modules:** aloja todos los componentes utilizados en el desarrollo del software y los necesarios para ejecutar el proyecto angular.

3.8.4. Creación del RESTful API

Para la creación de un RESTful[10] API mediante Flask se deben crear los directorios *models*, *templates* y los archivos app.py, config.py y manage.py, como se muestra en la Fig. 6.

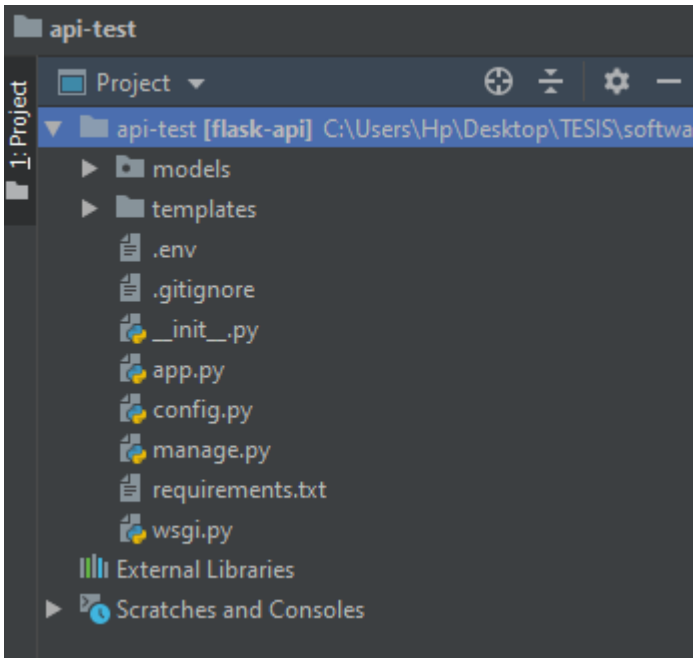

Fig. 6: Estructura proyecto Flask

- **Directorio *models*:** contiene todos los modelos necesarios para la posterior creación de la base de datos mediante el ORM de SQLAlchemy.
- **Directorio *templates*:** contiene los *templates*, paginas a usar para las diferentes redirecciones que realizara el api y además para mantener los formatos de los correos que envía el sistema web.
- **Archivo app.py:** contiene los métodos y lógica del procesamiento de la información.
- **Archivo manage.py:** contiene la configuración de los comandos a usar en la etapa de desarrollo.

[10] **RESTful:** conjunto de restricciones que se utilizarán para crear servicios web.

- **Archivo config.py:** contiene la información de la conexión a la base de datos.

3.8.5. Creación y configuración de la base de datos

La Fig. 7 muestra el ingreso al gestor de base de datos PgAdmin4 mediante la URL http://127.0.0.1:50291/browser/, mientras que la Fig. 8 muestra la creación de la base de datos.

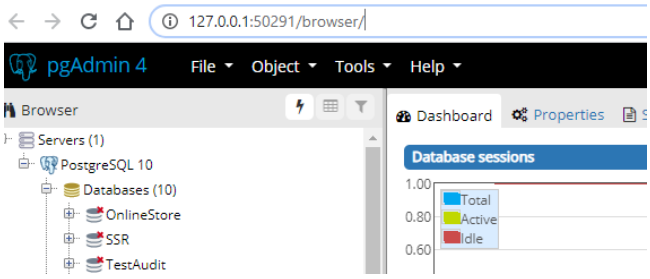

Fig. 7: Gestor de base de datos

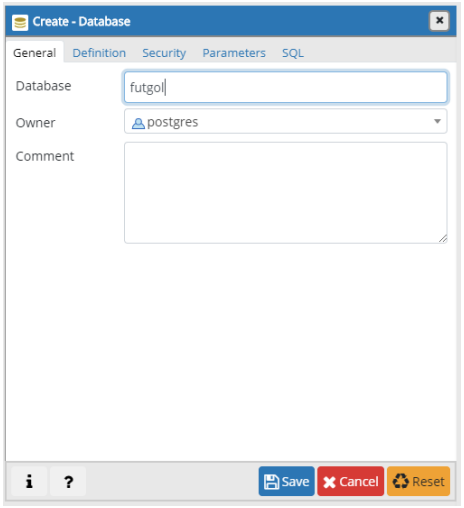

Fig. 8: Creación de base de datos

El modelo de base de datos se creó mediante el ORM de SQLAlchemy, el cual facilita la modificación de la base de datos sin necesidad de ejecución de comandos de lenguaje de definición de datos (DDL) [14], siendo práctico al momento de desarrollar y generar cambios los cuales se manejan mediante archivos de migración que son almacenas en un directorio creado por la herramienta.

3.8.6. Consumo de RESTful API

Un proceso fundamental en el sistema web es la consulta, ingresos y actualización de la información, lo cual se logra mediante los métodos desarrollados, los cuales están alojados en la url:

htttp://localhost:5000/{metodo}

El uso de *JavaScript Object Notation* (JSON)[11] que no es más que una colección de pares de nombre / valor, de formato ligero usado para el intercambio de datos [15], logra la comunica con el sistema web.

Mientras que la Transferencia de Estado representativa (REST), un protocolo de comunicación sin caché, que se comunica mediante HTPP [16], facilita la interacción entre la aplicación y la manipulación de datos.

Estructura del retorno generado por RESTful API

La Fig. 9 ilustra el retorno de la información del RESTful API, con la información referente al método consultado, cada método a consultar proporciona un formato de información diferente en formato JSON.


```
[
  {
    "create_at": "Thu, 23 Jan 2020 07:12:20 GMT",
    "email": "",
    "foundation": "1990-11-11",
    "id": 1,
    "location": "Loma de Puengasi",
    "logo": "https://dl.dropbox.com/s/jd2vkr3edye1mqr/LigaIndependienteLomadePuengasig5017.png?dl=0",
    "mission": "Atraer gran cantidad de equipos",
    "name": "Liga Independiente Loma de Puengasi",
    "phone": 992507969,
    "state": true,
    "update_at": "Thu, 23 Jan 2020 07:12:20 GMT",
    "vision": "Encaminar a la juventud al deporte"
  }
]
```


Fig. 9: Formato de retorno RESTful API

3.8.7. Construcción de módulos del sistema web

El sistema web está basado en los módulos: equipos, campeonatos, sanciones, resultados y programación.

La página principal de cada módulo se la puede visualizar de manera detallada en manual de usuario (Véase Manual Técnico – 12. Manual de Usuario).

Además, el sistema web cuenta con las validaciones necesarias para que la información a ingresar y consultar se mantenga integra y el software funcione acorde a las especificaciones. En el manual de usuario se puede observar los mensajes de

[11] **JSON:** formato de texto ligero para el intercambio de datos.

validaciones que contiene cada formulario del sistema (Véase Manual Técnico – 12. Manual de Usuario).

Creación de la página de inicio

La página de inicio se encuentra configurada en la IP 54.208.143.236, tal como muestra la Fig. 10.

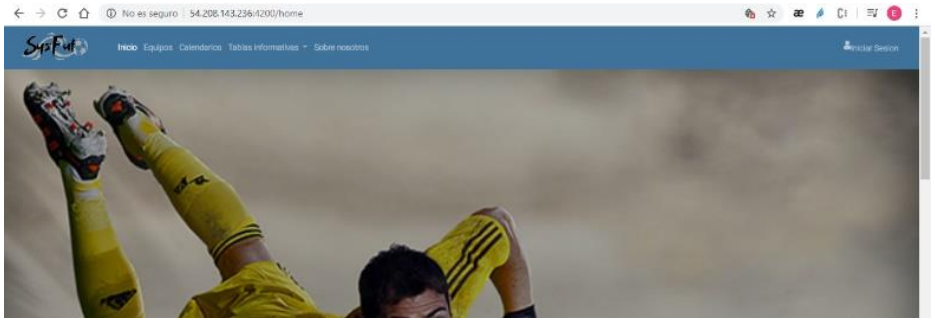

Fig. 10: Página principal SysFut

El sistema web cuenta con diseño *responsive*[12] en cada una de sus páginas, logrando así la adaptación de la página mediante la redimensión de sus elementos automáticamente, para obtener una vista amigable para el usuario como se visualiza en la Fig. 11.

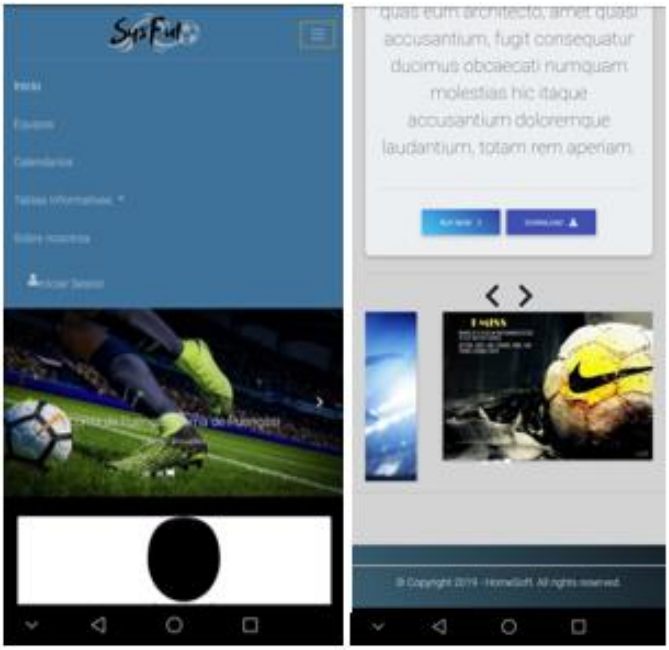

Fig. 11: Diseño responsive SysFut

Creación del módulo de autenticación y registro de usuarios

[12] ***Responsive:*** diseño web adaptativo a distintos dispositivos

Para el desarrollo de este módulo se creó un método RESTful para validar que las credenciales ingresadas sean correctas. Por defecto existirá un usuario administrador creado por el desarrollador el cual deberá autenticarse mediante las credenciales para realizar la gestión de los procesos del software, tal como muestra la Fig. 12.

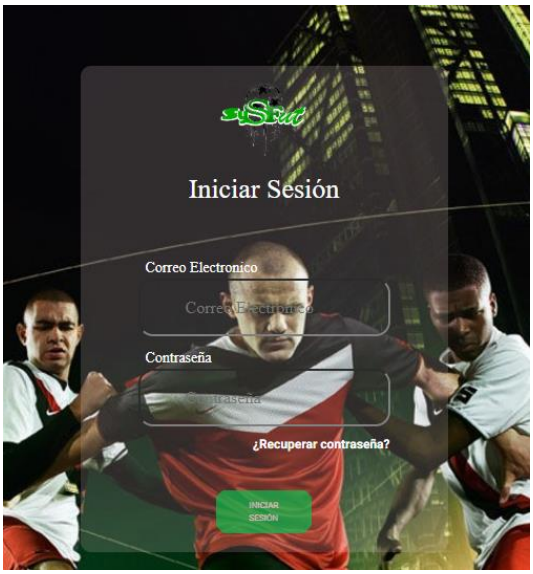

Fig. 12: Módulo de autenticación

Cabe mencionar que los usuarios podrán ver y acceder al contenido que únicamente están autorizados, esto se lo realiza mediante la validación de tipo de perfil que posea el usuario. Además, si el usuario intenta ingresar a una página, ruta que no existe el sistema automáticamente le redirigirá a la página principal del sistema.

La seguridad de las vistas se lo maneja mediante los *guards*[13] que proporciona angular, en cuanto a las paginas inexistentes se lo realiza mediante el *routing*[14] usado para las rutas.

Módulo de usuarios

Este módulo permite al usuario con perfil presidente registrar, modificar y visualizar los usuarios, tipo de representantes, representantes además de poder generar reportes e ingreso de forma masiva de la información en cuanto a usuarios, ver la Fig. 13..

[13] **Guards:** permitir la navegación a una ruta solicitada

[14] **Routing:** facilita la comunicacion entre los diferentes componentes

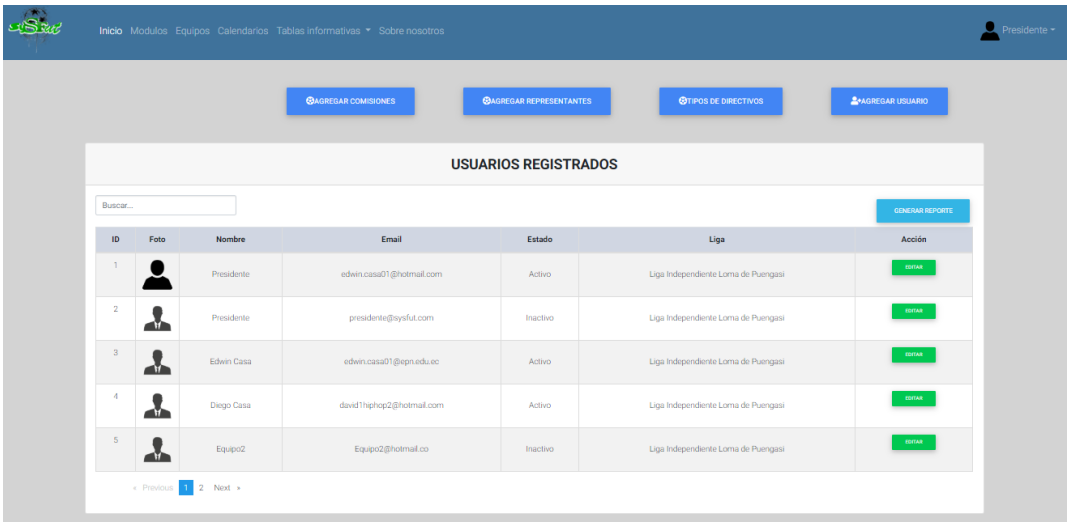

Fig. 13: Módulo usuarios

Módulo de campeonatos

En este módulo el usuario autorizado puede registrar, modificar y visualizar la información completa de cada campeonato registrado, activar el campeonato, descargar el reglamento del campeonato, visualizar la lista de equipos participantes, registro de las diferentes comisiones, ver la Fig. 14.

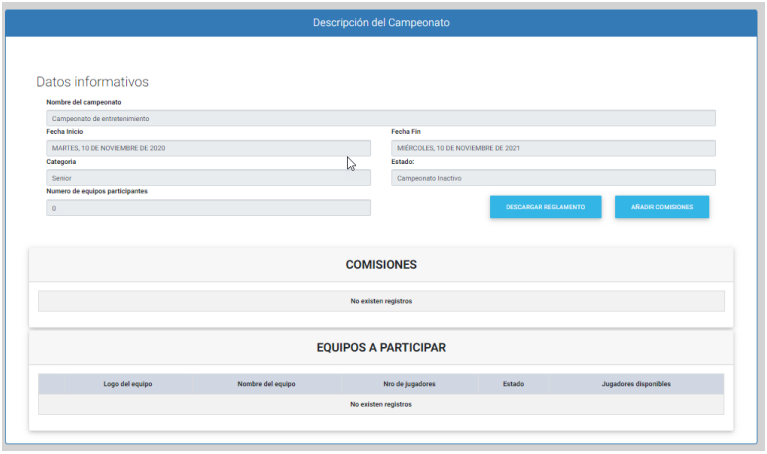

Fig. 14: Módulo de campeonatos

Módulo de equipos

El usuario con perfil presidente puede visualizar todos los equipos inscritos, posibles participantes. El usuario con perfil delegado puede visualizar su equipo, editar y visualizar los demás equipos que se encuentren aprobados en el campeonato seleccionado, además, puede registrar jugadores, visualizar la nómina de jugadores inscritos, exportar e importar información y descargar la lista de carnets de jugadores, ver la Fig. 15.

Al momento de registrar un equipo o un jugador se deben realizar todas las validaciones correctamente, las cuales se describen a detalle en el manual de usuario (Véase Manual Técnico - 12. Manual de Usuario).

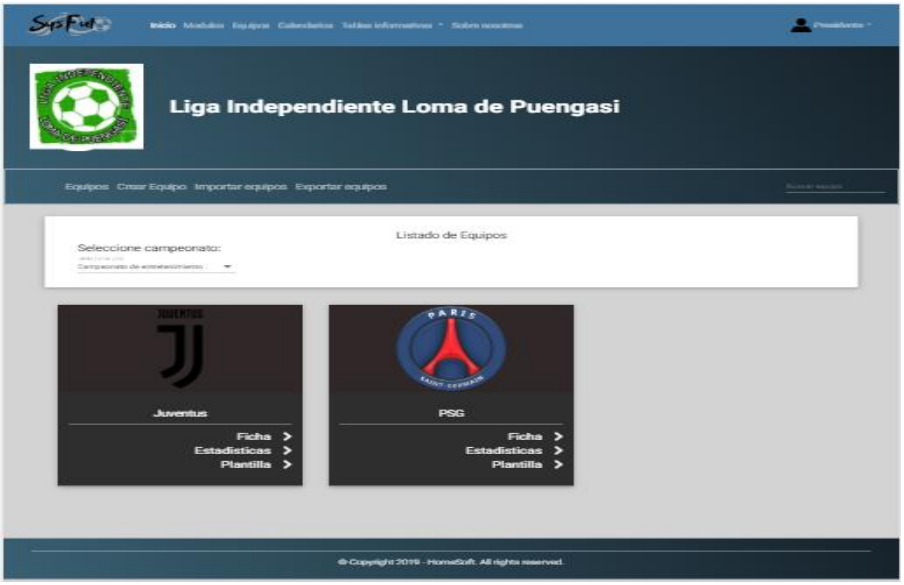

Fig. 15: Módulo equipos

Módulo de sanciones

En este módulo, se puede realizar el registro, modificación, visualización de los tipos de sanciones y sanciones aplicadas tanto a jugadores como a equipos, ver la Fig. 16.

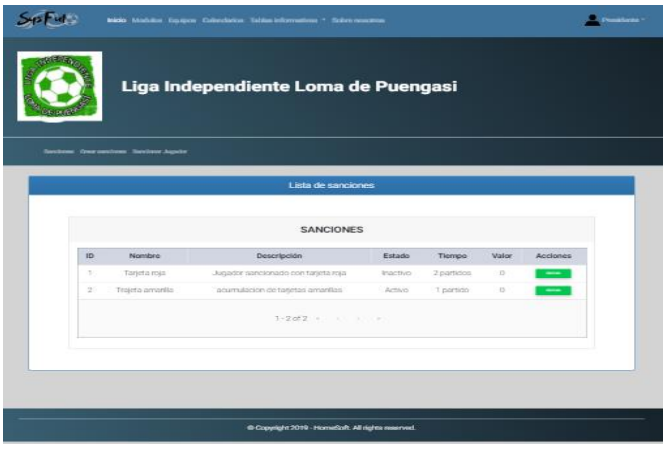

Fig. 16: Módulo de sanciones

Módulo de programación

En este módulo, el usuario con perfil presidente puede generar la programación por campeonato, teniendo en cuenta que el mínimo de equipos para la programación es de 7, ver la Fig. 17.

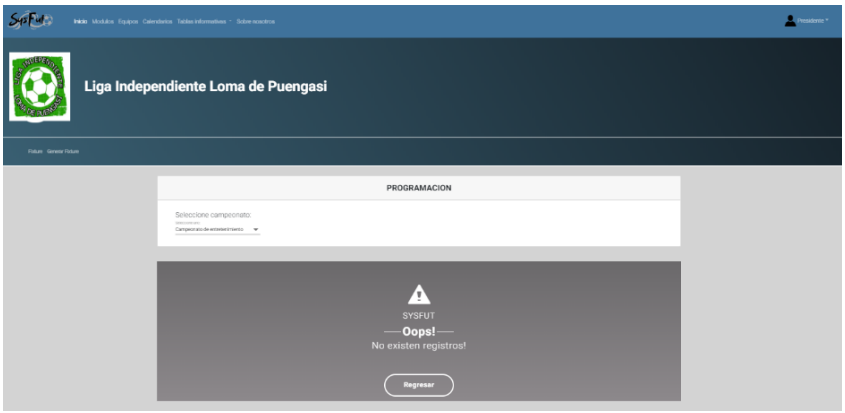

Fig. 17: Módulo de programación

3.9. Pruebas del sistema web

Se realizó tres tipos de pruebas: funcionalidad, aceptación y usabilidad al sistema web, con la finalidad de comprobar que este cumpla con los requerimientos y objetivos planteados por el usuario.

3.9.1. Pruebas de funcionalidad

Se realizaron pruebas de funcionalidad con los navegadores web de escritorio más comunes, bajo la plataforma de Windows con el objetivo de comprobar el correcto desempeño del sistema web en dichos navegadores, observando que el sistema web funciona sin ningún problema en estos navegadores. La TABLA IV muestra los resultados de las pruebas realizadas.

TABLA IV: Pruebas de funcionalidad en navegadores web.

Navegador	**Versión**	**Resultado de funcionalidad**
Google Chrome	79.0.3945.130 (*Build oficial*)	Totalmente funcional
Mozilla Firefox	72.0.2 (64-bit)	Totalmente funcional
Opera	5.803.135.127	Totalmente funcional
Microsoft Edge	44.18362.449.0	Totalmente funcional

3.9.2. Pruebas de aceptación

Las pruebas de aceptación se realizaron con la directiva de la Liga Deportiva Loma de Puengasí, con el objetivo de verificar que las funcionalidades implementadas en el sistema web satisfagan los requerimientos levantados. La TABLA V muestra un ejemplo las pruebas de aceptación enfocadas a la línea de negocio, para observar todas las pruebas realizadas véase Manual Técnico – 6. Pruebas de Aceptación.

TABLA V: Prueba de aceptación Nro.10: Generar programación.

Pruebas de aceptación	
Identificador de prueba (ID): PA10	**Identificador de historia (ID):** HU27
Nombre prueba de aceptación: Generar programación.	
Descripción: Ingreso al sistema web y generar la programación de un campeonato.	
Condiciones de prueba: Tener conexión a Internet. Iniciar sesión con perfil presidente. Tener un campeonato activo.	
Pasos de ejecución: Ingresar al módulo programación. Presionar la opción generar programación. Seleccionar campeonato. Presionar el botón generar programación.	
Resultado deseado: Generación de programación automáticamente.	
Evaluación de la prueba: Resultado satisfactorio, la programación se genera de forma automática y correctamente, además se puede visualizar los resultados.	

3.9.3. Pruebas de usabilidad

Para verificar la usabilidad en el sistema web se realizado las pruebas con 12 usuarios, 11 usuario con perfil delegado y 1 usuario con perfil presidente para que así se pueda cumplir el flujo del proceso de manera completa, cada uno realizo las siguientes tareas:

- Iniciar sesión
- Registrar un campeonato
- Registrar un Equipo
- Registrar jugadores
- Generar programación
- Registrar resultados
- Visualizar los carnets
- Cerrar sesión

El evaluador guio a los usuarios con el fin de verificar el cumplimiento de cada una de las tareas, al final se solicitó que contesten una encuesta (Véase Manual Técnico – 10. Encuesta de Usabilidad de Software) obteniendo una serie de resultados (Véase Manual Técnico – 11. Resultados de la Encuesta de Usabilidad de Software).

A través de los resultados obtenidos de las encuestas aplicadas a los diferentes usuarios participantes de las pruebas, se determinó que el sistema web Sysfut tiene la aceptación por parte del cliente y usuarios. Además, se obtuvo comentarios y recomendaciones los cuales se los puede tener en cuenta para implementaciones en futuras versiones del software.

4. CONCLUSIONES Y RECOMENDACIONES

En base a las pruebas y encuestas realizadas al sistema web se plantea las conclusiones y recomendaciones obtenidas de los comentarios del cliente y los usuarios.

4.1 CONCLUSIONES

El sistema web para la gestión de campeonatos de la Liga Deportiva Loma de Puengasí, satisface las necesidades y cumple con los objetivos planteados, permitiendo así la automatización de los procesos que se llevan a cabo, en cuanto a la gestión de información.

El sistema web facilita a todos los usuarios obtener información actualizada, tanto de campeonatos en curso como finalizado; tales como: programación, resultados, plantilla de jugadores, listado de equipos, tabla de máximos anotadores, tabla de posiciones, de forma inmediata y segura.

En cuanto a la información sobre resultados, estos no podrán ser modificados una vez que el registro haya finalizado, evitando así la manipulación de la información, con el fin de proporcionar información integra y confiable.

Las herramientas de software libre utilizadas para el desarrollo del presente proyecto integrador fueron: Python, Flask, SQLAlchemy, PostgreSQL, Angular, las cuales permitieron cumplir con los requerimientos establecidos por el cliente en iteraciones cortas y con un coste mínimo.

El sistema web fue desarrollado con una arquitectura Modelo – Vista - Controlador (MVC), lo cual permitirá que el software pueda ser escalable y fácil de mantener debido a la separación de capaz existente.

El uso del microframework Flask facilitó el desarrollo de la APIRESTful, debido a que la curva de aprendizaje de dicha herramienta no es elevada. Además, se puede integrar de manera sencilla paquetes de Python lo cual facilita al desarrollo.

Con la automatización de procesos como son: creación de campeonatos, registro de sanciones, registro de equipos, generación de programación, registro de jugadores, se logra eliminar las actas y documentos físicos, ya que el sistema web junto con la API

realizan consultas en tiempo real de la información, proporcionando vistas detalladas de cada proceso.

Para garantizar el correcto funcionamiento del sistema web se realizaron pruebas de usabilidad, funcionalidad y aceptación del sistema. Los resultados de cada una de las pruebas fueron satisfactorios, lo cual indico que el sistema web cumple con los requerimientos establecidos por el cliente.

4.2 RECOMENDACIONES

Para el uso del sistema web el usuario debe disponer de conexión a internet, ya que los archivos e imágenes son cargados a la nube para así mantener la información integra, y evitar costes de implementación adicionales.

El usuario presidente debe tener conocimientos básicos acerca del flujo de procesos que se tiene que llevar a cabo para la creación de un nuevo campeonato, ya que esto permitirá explotar al máximo al sistema web desarrollado.

Para una futura versión se debe añadir el módulo de inscripciones el cual permita generar reportes, visualizar la información en cuanto a valores de dinero que ingresan y salen de los fondos de la Liga Deportiva Loma de Puengasí.

Actualmente el sistema presenta información a través de Internet, aprovechando la difusión de este medio de comunicación en dispositivos móviles y la tendencia a disminuir su costo se recomienda ampliar el acceso del sistema a estos dispositivos y así ofrecer otra alternativa al usuario final de la aplicación.

Con el transcurso del tiempo el volumen de datos manejados por el sistema crecerá por lo que es recomendable medir continuamente el desempeño del servidor Web y de la base de datos para que no se vea afectado al desempeño del sistema.

El uso del manual de usuario ayudará a tener una perspectiva clara acerca de los diferentes flujos del sistema evitando así el uso erróneo del sistema, despejando dudas de los errores presentes en cada formulario, además, se describe a detalle la línea de negocio del sistema.

REFERENCIAS BIBLIOGRÁFICAS

[1] F. J. R. Laínez, Desarrollo de Software ÁGIL: Extreme Programming y Scrum, Madrid: IT Campus Academy, 2015.

[2] Chromatic, Extreme Programming Pocket Guide: Team-Based Software Development, Graveinstein Highway North: O'Reilly Media, Inc., 2003.

[3] J. Joskowicz, Reglas y Prácticas en eXtreme Programming, España, 2008.

[4] S. Ambler, Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process, New York: ilustrada, 2002.

[5] NinjaMock, «NinjaMock,» 27 02 2020. [En línea]. Available: https://ninjamock.com/.

[6] N. N. C. Menezes, Introducción a la programación con Python: Algoritmos y lógica de programación para principiantes, Sao Paulo: Novatec Editora, 2017.

[7] G. Miguel, Flask Web Development, Sebastopol: Meghan Blanchette and Rachel Roumeliotis, 2014.

[8] R. Copeland, Essential SQLAlchemy, Sebastopol: Mary E. Treseler, 2008.

[9] Angular, «Angular,» 27 02 2020. [En línea]. Available: https://angular.io/guide/architecture.

[10] R. Obe y L. Hsu, PostgreSQL: Up and Running, Sebastopol: Meghan Blanchette, 2012.

[11] V. Kholodkov, Nginx Essentials, Birmingham : Packt Publishing Ltd, 2015.

[12] A. F. Montoro, PYTHON 3 al descubierto, Madrid: RC Libros, 2012.

[13] A. A. H. C. Jeffries R., Extreme Programming Installed, Addison-Wesley., 2001.

[14] T. E. Chicano, UF1472 - Lenguajes de definición y modificación de datos SQL, Malaga: Editorial Elearning, 2015.

[15] B. Smith, Beginning JSON Expert's voice in Web development, New York: Apress, 2015.

[16] L. Richardson, M. Amundsen y S. Ruby, RESTful Web APIs: Services for a Changing World, Sebastopol: O'Reilly Media, Inc, 2013.

ANEXOS

MANUAL TÉCNICO

1. Carta de aceptación del proyecto
2. Carta de módulo financiero
3. Diagrama de Base de datos
4. Historias de Usuario
5. Prototipo del Sistema
6. Pruebas de Aceptación
7. Formato Excel importar usuarios
8. Formato Excel importar equipos
9. Formato Excel importar sanciones
10. Encuesta de Usabilidad de Software
11. Resultados de la Encuesta de Usabilidad de Software
12. Manual de Usuario
13. Carta de entrega del proyecto