



**UNIVERSIDAD SAN FRANCISCO DE QUITO**

**Colegio de Ciencias e Ingeniería**

**Torneos Instantáneos de Fútbol en cualquier parte del mundo:  
Propuesta de implementación de una aplicación web que permita la  
generación y la administración de torneos de todos los equipos  
involucrados que se enfrentan entre sí y elige un ganador a través de la  
acumulación de puntos.**

**Ricardo Herrera Petit**

**Daniel Fellig, M.Sc., Director de Tesis**

Tesis de grado presentada como requisito para la obtención del título de Ingeniero en  
Sistemas

Quito, abril de 2014

**Universidad San Francisco de Quito  
Colegio de Ciencias e Ingeniería**

**HOJA DE APROBACION DE TESIS**

**Torneos Instantáneos de fútbol en cualquier parte del mundo: Propuesta de implementación de una aplicación web que permita la generación y la administración de torneos de todos los equipos involucrados que se enfrentan entre sí y elige un ganador a través de la acumulación de puntos.**

**Ricardo Herrera Petit**

Daniel Fellig, M.Sc.  
Director de Tesis

.....

Fausto Pasmay, M.Sc.  
Miembro del Comité de Tesis

.....

Daniel Fellig, M.Sc.  
Miembro del Comité de Tesis

.....

Mauricio Iturralde, Ph.D.  
Miembro del Comité de Tesis

.....

Ximena Córdova, Ph.D.  
Decana de la Escuela de Ingeniería  
Colegio de Ciencias e Ingeniería

.....

Quito, abril de 2014

Quito, abril de 2014  
© DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído la Política de Propiedad Intelectual de la Universidad San Francisco de Quito y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo de investigación quedan sujetos a lo dispuesto en la Política.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo de investigación en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Firma:

-----

Nombre: Ricardo Herrera Petit

C. I.: 1722839113

Fecha: Quito, abril de 2014

## RESUMEN

El presente proyecto de tesis trata de la creación de una aplicación web en lenguaje de programación Python y la ayuda del Framework Django, para controlar la generación y administración de torneos de fútbol con acceso desde cualquier parte del mundo desde cualquier explorador de internet.

La aplicación provee un súper administrador creado por el sistema y un módulo para registrar usuarios que deseen utilizar la plataforma de forma independiente. Para esta aplicación se implementaron cuatro funcionalidades: registro de usuarios, creación de usuarios y asignación de permisos, creación de torneos y cálculos realizados para definir un ganador del torneo en formato de liga. Los cálculos realizados proporcionan una tabla de posiciones por torneo en la cual se ordenan los equipos por puntos, diferencia de goles y goles a favor respectivamente.

La aplicación permite diferenciar tipos de Usuarios. Existe el usuario digitador que tiene el permiso de actualizar los resultados de los partidos del torneo que fue asignado. Existe el usuario representante de equipo que podrá ver únicamente la tabla de posiciones del equipo al cual fue asignado. Por último, existen torneos públicos los cuales podrán ser vistos por cualquier persona con acceso a internet sin tener que registrarse.

## ABSTRACT

The present thesis project consists in the development of a web application in the Python programming language and Django framework to control the generation and administration of soccer tournaments with access from any Internet browser around the world.

The application provides a super administrator created by the system and a module to register any users using an independent platform. For this application, four functionalities were developed: user registration, user creation with privileges, tournament creations, and computations to determine a champion in tournaments in league format. The computations provide a standings table in which is ordered by points, goal difference and goals in favor respectively.

The application allows to differentiate many types of users. There is the user that inputs data to update the scores of the matches from a tournament it was assigned. There is the user's team representative who can view only the standings of the team it was assigned to. Lastly, there are public tournaments that can be viewed by any type of user without registration.

## Tabla de Contenido

RESUMEN .....	5
ABSTRACT .....	6
Tabla de Contenido .....	7
Tabla de Figuras.....	10
Tabla de Tablas .....	12
Capítulo 1 - Introducción.....	13
Antecedentes.....	13
Justificación del Proyecto .....	16
Objetivo General.....	17
Objetivos Específicos .....	17
Metas .....	23
Actividades .....	24
Capítulo 2 - Marco Teórico.....	25
Tecnología a ser Utilizada .....	25
Lenguajes de programación.....	25
Frameworks para desarrollar aplicaciones web .....	27
Base de Datos .....	28
ORM.....	29
Metodología.....	30
Tecnologías Investigadas .....	30
Diseño.....	31
Desarrollo .....	32
Revisión Literaria .....	34
Capítulo 3 –Tecnologías e Implementación .....	37
Tecnologías Utilizadas Para la Aplicación de Generación de Torneos.....	37
Introducción a MVT y ORM de Django .....	37
Desarrollo de la Aplicación de Generador y Administrador de Torneos.....	39
Diseño Inicial de la Base de Datos .....	39
Diseño final de la Base de Datos .....	42
Arquitectura de la aplicación de Generador y Administrador de Torneos de Fútbol. ....	43
Configuraciones Iniciales .....	44
Controladores de Django .....	45
Módulo de Súper Administrador de Django.....	46
.....	47
Diseño de Django .....	47

Vista de Django (Template).....	48
Diagramas UML.....	51
.....	51
.....	51
Requerimientos de la aplicación.....	52
Casos de Uso .....	52
Capítulo 4 – Análisis y Resultados.....	58
Objetivo 1 .....	59
Objetivo 2.....	60
Objetivo 3.....	60
Objetivo 4.....	60
Objetivo 5.....	61
Objetivo 6.....	61
Objetivo 7.....	62
Objetivo 8.....	62
Objetivo 9.....	63
Objetivo 10.....	63
Objetivo 11.....	64
Objetivo 12.....	65
Objetivo 13.....	66
Capítulo 5.....	67
Conclusiones.....	67
Recomendaciones.....	68
Desarrollo Futuro.....	70
Referencias.....	72
Anexo 1: Manual de Usuario (Administrador de Torneos) para el Administrador y Generador de Torneos de fútbol.....	75
Registrarse y Entrar al Sistema.....	75
Crear un Torneo.....	77
Eliminar un Torneo.....	79
Crear un Equipo.....	80
Eliminar un Equipo.....	82
Generar Partidos (Automáticamente).....	83
Crear Partidos (Manualmente).....	85
Actualizar resultados (Usuario Digitador).....	86
Ver la tabla de posiciones (Usuario Representante de Equipo).....	87
Crear Rol de Usuario Digitador.....	88

Asignar Permisos al Rol de Usuario Digitador .....	88
Crear Rol de Usuario Representante de Equipos .....	89
Asignar Permisos al Rol de Representante de Equipos:.....	90
Anexo 2: Manual del módulo del Súper Administrador y para los módulos de Administrador y Generador de Torneos de fútbol. ....	91
Ingresar al módulo súper administrativo: .....	91
.....	92
¿Cómo cambiar la contraseña? .....	92
.....	94
Crear un usuario .....	94
Editar un usuario.....	95
Crear un Torneo .....	97
Editar un Torneo .....	99
Crear un Equipo.....	100
Editar un Equipo.....	102
Crear una jornada.....	103
Editar una jornada.....	104
Crear un Partido.....	105
Editar un partido .....	107
Anexo 3: Detalle de la aplicación del Generador y Administrador de torneos de fútbol. .	108
Módulo de Registro .....	108
ClaseRegistrarse .....	108
Clase RegistrarDigitador.....	108
Clase RegistrarRepresentante .....	108
Función Asignar Permisos .....	108
Función SuccessPermission.....	109
Clase Usuario.....	109
Decoradores .....	109
Módulo de Torneo.....	109
Función create.....	109
Función tournaments .....	109
Función tournament.....	110
Función deletetournament .....	110
Módulo de Partidos.....	110
Función Filtermatches .....	110
Función createStandings.....	110
Función insertarpartido.....	111

Función editarpartido.....	111
Función success .....	111
Función creatematches .....	111

## Tabla de Figuras

Figura 1.1: Diseño Principal de la Base de Datos .....	39
.....	41
Figura 1.2: Diseño de la base de Datos después de Actualización con Django .....	41
.....	44
Figura 2.1 Estructura de directorio de aplicación del Generador y Administrador de torneos de fútbol.....	44
Figura 2.2 Estructura de directorio de aplicación general .....	46
Figura 2.3 Estructura de directorio de Archivos Estáticos .....	47
Figura 2.4 Estructura de directorio de archivo de aplicación de Torneos .....	49
Figura 2.5 Diagrama UML de clases para la Aplicación de Generación de Torneos .....	51
Figura 2.6 Segundo Diagrama UML de clases para la Aplicación de Generación de Torneos .....	51
Figura 2.7 Tercer Diagrama UML de clases para la Aplicación de Generación de Torneos.....	52
.....	55
Figura 3.1 Caso de Uso de Súper Administrador .....	55
.....	56
Figura 3.2 Caso de Uso de Administrador de Torneos .....	56
.....	57
Figura 3.3 Caso de Uso de Usuario Visita, Representante y Digitador.....	57
Figura 4.1: Pantalla de Inicio del Sistema o Home .....	75
Figura 4.2: Pantalla de Registro .....	75
Figura 4.3: Pantalla de Login .....	76
Figura 4.4: Pantalla de Bienvenida al Sistema .....	77
Figura 4.5: Pantalla de Creación de Torneo.....	78
Figura 4.6: Pantalla con la lista de Torneos.....	80
Figura 4.7: Pantalla para Eliminación de Torneo .....	80
Figura 4.8: Pantalla para la Creación de un Equipo .....	81
Figura 4.9: Pantalla de Creación de Equipo Exitosa .....	82
Figura 4.10: Pantalla de Listado de Equipos .....	82
Figura 4.11: Pantalla de Eliminación de Equipos.....	83
Figura 4.12: Pantalla de Selección de Torneos para Generar Partidos .....	83
Figura 4.13: Pantalla Previa a la Generación de Partidos.....	84
Figura 4.14: Pantalla de Confirmación de Creación de Partidos.....	84
Figura 4.14: Pantalla de Inserción Manual de Partidos.....	85
Figura 4.15: Pantalla de Ingreso de Resultados.....	86
Figura 4.16: Pantalla Para Ver Standings .....	87
Figura 4.17: Pantalla de Ejemplo de la Tabla de Posiciones de un Torneo .....	88
Figura 4.18: Pantalla de Asignar Permisos a Usuario Digitador .....	89
Figura 4.19: Pantalla de Asignación de Equipos a Representante de Equipos.....	90
Figura 5.1: Página Principal de Entrada para el Súper Administrador .....	91
Figura 5.2: Pantalla Principal del Súper Administrador .....	92
Figura 5.3: Pantalla de Listado de Usuarios del Súper Administrador.....	92
Figura 5.4: Pantalla de Listado del Usuario Admin.....	93

Figura 5.5: Pantalla de Edición de Usuario Admin .....	93
Figura 5.6: Pantalla de Cambio de Contraseña para Usuario Admin .....	94
Figura 5.7: Pantalla de Cambio de Contraseña Exitosa.....	94
Figura 5.8: Pantalla de Agregar Usuario .....	95
Figura 5.9: Pantalla Completa de los Datos de edición de Usuario .....	96
Figura 5.10: Pantalla Para Agregar Torneos .....	98
Figura 5.11: Pantalla Para Agregar Torneos con Calendario.....	98
Figura 5.12: Pantalla de Listado de Torneos .....	100
Figura 5.13: Pantalla de Creación de Equipos.....	101
Figura 5.14: Pantalla de Listado de Equipos .....	102
Figura 5.15: Pantalla de Ingreso de Datos a Jornadas .....	103
Figura 5.16: Pantalla de Ingreso de Datos de Partidos .....	105

## Tabla de Tablas

Tabla 1: Tablas de insertadas por el Desarrollador, Django y Django-Guardian .....	42
Tabla 2: Explicación Breve de Directorio General de Django .....	46
Tabla 3: Explicación de Archivos Estáticos .....	47
Tabla 4: Archivos Generales de Módulos .....	49
Tabla 5: Metas Alcanzadas .....	59

## Capítulo 1 - Introducción

### Antecedentes

Actualmente, existen muchos torneos de fútbol que utilizan un software para poder administrarlos. Los torneos mencionados anteriormente son los torneos con un formato de liga. La liga consiste en un conjunto de dos o más equipos que tienen partidos entre sí. Los partidos consisten de dos equipos, un equipo local y un equipo visitante. Generalmente en las ligas, todos los equipos involucrados juegan entre sí dos veces debido a que se intercambian los roles de equipo local y equipo visitante. Esto quiere decir que los partidos consisten de todas las permutaciones posibles que existen para los equipos inscritos en una liga. Cada liga tiene sus propios equipos y todas las ligas son independientes entre sí. Un ganador se define a través de la acumulación de puntos. Un equipo acumula puntos por partido. Si el equipo resulta ser el ganador del partido, acumulará tres puntos, si el equipo resulta empatar con el otro; entonces acumulará un punto. Sin embargo, si el equipo resulta ser perdedor no obtendrá ningún punto. Estos puntos de los partidos se suman durante toda la liga, el campeón es el que tiene más puntos y así se ordenan los equipos en la tabla de posiciones. La tabla de posiciones es un listado ordenado de los equipos en base al puntaje. Es posible que dos equipos tengan la misma cantidad de puntos, en este caso se ordena por diferencia de goles. La diferencia de goles es la resta total entre los goles convertidos por el equipo y los goles recibidos por el equipo. Esta referencia se utiliza como segundo criterio para elegir a un campeón. Es decir, cuando haya equipos con la misma cantidad de puntos, el que tenga la mayor diferencia de goles será el campeón y se utilizará este segundo criterio para el ordenamiento de la tabla de posiciones. Otra posibilidad se define cuando los dos equipos tienen la misma cantidad de puntos y la misma diferencia de goles. En este caso se utiliza un tercer criterio de ordenamiento, los goles a favor. Los goles a favor son la cantidad total de los goles convertidos del equipo en todos los partidos de la liga. Si los equipos tienen

la misma cantidad de puntos y diferencia de goles, el equipo que tenga la mayor cantidad de goles a favor será clasificado como ganador y se utilizará este tercer criterio para el ordenamiento en la tabla de posiciones. En caso de que los equipos tengan la misma cantidad de puntos, diferencia de goles y goles a favor, dependerá del organizador de la liga establecer un desempate y definir un ganador. Existen aplicaciones web que permiten hacer la generación de torneos y la administración de los mismos pero de manera pagada.

Existen otras aplicaciones que permiten hacer la generación de los torneos pero con muy poca flexibilidad, el torneo de eliminación directa. El torneo de eliminación directa es una competencia entre un conjunto de equipos que se emparejan para jugar partidos. En este caso los partidos son únicos y siempre existe un ganador. Luego se vuelven a emparejar los ganadores en otros partidos. Este ciclo se repite hasta que exista un único partido, esto se llama la final del torneo. La final del torneo es un partido decisivo para determinar al campeón del torneo. Generalmente, solo empiezan las siguientes cantidades de equipos 2, 4, 8, 16, ...  $2^n$ . Después de que los equipos están registrados en el torneo, se emparejan equipos aleatoriamente en partidos en el primer nivel del torneo. Un partido consiste de dos equipos de los cuales tiene que existir un ganador. Si durante el partido, los equipos empatan entonces se elige a un ganador a través de penales. Los penales son cinco remates libres al arco. El equipo que consiga anotar más penales que el otro será el ganador. Si después de los cinco remates no existe un ganador, entonces continúa la ejecución de penales hasta determinar uno. Los equipos ganadores de los partidos del primer nivel pasan al siguiente y se enfrentan con otros equipos ganadores por la ruta establecida en el horario. Esto continúa así hasta que existan solamente dos equipos para enfrentarse, la final. En la final el equipo ganador es el equipo campeón del torneo, es decir que los equipos tienen un número máximo de partidos jugados dependiendo de su rendimiento. Si el número de equipos no sigue la secuencia mencionada, entonces los organizadores hacen arreglos al horario que haya otra secuencia

establecida.

Otro tipo de aplicación implementa el torneo con fases de grupos y luego se convierte en torneo de eliminación directa, los torneos mixtos. Las fases de grupos es un conjunto de equipos de formato liga, de los cuales clasifican dos equipos. En esta etapa del torneo sí existen los empates. Cuando estos dos equipos clasifican, entonces proceden a las reglas del torneo de eliminación directa. En los torneos mixtos se registran los equipos, el número de equipos normalmente es una secuencia como la de los torneos de eliminación directa. Existen aplicaciones para crear este tipo de torneos mixtos pero no son gratuitas y se limitan a tener solo dos torneos por usuario.

Se necesita una aplicación que pueda generar este manejo de creación y administración de torneos de forma fácil, rápida y accesible. La aplicación deberá permitir a cualquier persona con acceso a internet crear su torneo. Esa persona podrá crear los equipos participantes en ese torneo. La aplicación generará los partidos de forma aleatoria con una opción de edición por si el usuario está insatisfecho con el orden generado de los partidos. La aplicación se asegurará de que ningún equipo juegue contra sí mismo. La aplicación tendrá dos modos. El primer modo será para crear los torneos que sean solo de ida, es decir que un equipo juegue solamente una vez contra todos los otros equipos. El segundo modo, será para crear los partidos de ida y vuelta, en este caso el equipo jugará contra todos los otros equipos al menos dos veces. El usuario podrá ingresar los datos de los partidos generados y la aplicación generará la tabla de posiciones ordenando por puntos, goles diferencia y goles a favor.

Se propone implementar una aplicación web, centralizada y gratuita. De esta manera, cualquier persona que se registre al sistema podrá generar sus torneos en formato de liga, en cada uno de ellos podrá crear sus equipos, generar los partidos, editar los partidos si está insatisfecho con el orden e ingresar los resultados de los partidos. El sistema responderá

automáticamente proveyendo una tabla de posiciones dinámica que cambiará cuando se editen los resultados de los partidos del sistema mostrando el rendimiento de los equipos hasta la finalización de la liga. Además el usuario podrá cambiar la privacidad del torneo a privado o público. Un torneo privado será un torneo el cual solo el dueño del torneo y las personas que el dueño provea acceso puedan ver la tabla de posiciones. Un torneo público, será un torneo que cualquiera pueda ver la tabla de posiciones del mismo.

### **Justificación del Proyecto**

El fútbol es uno de los deportes más populares del mundo. Existen varios tipos de torneos de que se generan y administran. Los torneos son internacionales, nacionales, regionales, institucionales y virtuales. Existen también torneos de eliminación directa, torneos de formato de liga y torneos mixtos. La propuesta del proyecto es brindar facilidades a los administradores torneos para generar sus partidos y también para poder administrarlos. De esta manera, los administradores podrán tener un mejor control multiplataforma para sus torneos.

Los usuarios beneficiados de esta aplicación no serán solamente los administradores de los torneos sino también otro tipo de usuarios como los usuarios representantes de equipos, usuarios que ingresan los datos y usuarios visita. Los usuarios administradores tendrán el control de la creación de su torneo, la generación de sus partidos, y podrán contemplar el rendimiento de sus equipos para cada uno de los torneos que estos usuarios crearon. Estos usuarios podrán ser organizaciones que necesitan del sistema para administrar sus torneos o personas naturales que se registran al sistema y quieren crear su propio torneo por diversión ya sea real o virtual. Otro beneficio es que los usuarios representantes de sus equipos podrán conocer el rendimiento de su equipo. Los jugadores de cada equipo de instituciones, como son las universidades, podrán seguir el rendimiento de los torneos internos que administran. Un ejemplo de esto sería el campeonato interno de la Universidad

San Francisco de Quito que se realiza todos los semestres. Así, este tipo de usuario podrá saber la posición de su equipo a lo largo del torneo. Los usuarios visita podrán notar el rendimiento de los equipos de todos los torneos que sean públicos. De esta manera podrán ver una tabla dinámica de posiciones y será una forma de patrocinar el sistema.

Justamente, la propuesta es facilitar la administración y generación de torneos de fútbol. De ese modo distintos tipos de usuarios podrán ver el rendimiento de sus equipos, el manejo de torneos simultáneos y lo podrán hacer desde cualquier parte del mundo con tan solo tener acceso a Internet.

### **Objetivo General**

El objetivo final es crear una aplicación web que permita generar y administrar torneos de fútbol públicos y privados, los torneos privados podrán ser vistos por el dueño del torneo y otras personas a las que el dueño permita el acceso. Los torneos públicos podrán ser vistos por cualquier persona. El sistema se implementará mediante tecnología de software libre que ofrece el desarrollo de aplicaciones web.

### **Objetivos Específicos**

- 1 Diseño de la Base de Datos
  - Crear una entidad-relación de la base de datos
  - Diseñar tabla torneos para crear torneos
    - i. Id del Torneo
    - ii. Nombre del Torneo
    - iii. Fecha de inicio del Torneo
    - iv. Fecha Fin del Torneo
    - v. Booleano si es ida y vuelta
    - vi. Booleano público o privado

- Diseñar tabla equipos para crear equipos en torneos
  - i. Id del equipo
  - ii. Nombre del Equipo
  - iii. Descripción del Equipo (Opcional)
  - iv. Torneo al que corresponde
- Las tablas de usuarios y privilegios no se diseñan debido a que la tecnología que se utiliza para desarrollar esta aplicación ya los provee.
- Diseñar la tabla partidos para poder hacer cálculos e implementar con estos datos una tabla de posiciones
  - i. Id del Partido
  - ii. Equipo Local
  - iii. Equipo Visitante
  - iv. Gol de Equipo Local
  - v. Gol de Equipo Visitante
- Diseñar la tabla jornadas para establecer una jornada a cada partido en un torneo
  - i. Id de la Jornada
  - ii. Nombre de la Jornada
  - iii. Fecha de la Jornada
- Verificar si la base de datos está normalizada
- Normalizarla sino lo está
- Verificar de nuevo si la base de datos está normalizada
- Empezar con la implementación de la base de datos

## 2. Implementar la base de datos

- Correr el script para implementar la base de datos

- Generar datos de prueba
  - Crear un plan de restauración por si se dañan los datos
    - i. Hacer un respaldo automático cada día para tener los datos restaurados que serán guardados en el VPS.
  - Cargar datos de prueba para posteriormente hacer las pruebas (los cuales podrán ser eliminados al final del proyecto)
  - Cargar los usuarios principales como el súper administrador y el usuario visita
    - i. Usuario Visita podrá:
      - 1. Hacer login sin registrarse
      - 2. Ver la tabla de posiciones de los torneos públicos
    - ii. Súper Administrador podrá:
      - 1. Eliminar usuarios
      - 2. Eliminar torneos
      - 3. Eliminar equipos
      - 4. Editar registro de partidos
      - 5. Editar torneos
      - 6. Editar usuarios (para ponerles los permisos correspondientes)
3. Preparar el entorno de desarrollo para la aplicación en Linux ( Ubuntu 13.0 )
- Instalar PIP
  - Instalar Django
  - Instalar Django-Registration
  - Instalar Django-Guardian
  - Instalar Virtualenv
  - Instalar librería de MySQL para Python

- Instalar driver de MySQL para Python
- Cambiar configuraciones de Django para conectarse a la base de datos creada en MySQL y sincronizarla con las tablas de Django

#### 4. Implementar el módulo de súper administrador

- El súper administrador tendrá control para la edición y eliminación de torneos públicos, torneos privados, equipos y partidos.
- El súper administrador tendrá el control de la edición y eliminación de usuarios pero no podrá crearlos. (Los usuarios se crean desde el módulo de registro).
- Este módulo tendrá el control de la edición y de la eliminación de todos los usuarios del sistema, sin embargo los usuarios serán registrados por sí mismo en el módulo de registro que se menciona a continuación.

#### 5. Implementar el módulo de Registro

- Este módulo será para que cualquier persona puede registrarse a la base de datos y administrar sus torneos
- Tiene la funcionalidad de cargar los datos de algún usuario que quiso registrarse al sistema para poder crear y administrar su torneo.

#### 6. Implementar el módulo de Login

- Este módulo será implementado para poder entrar al sistema
- Se investigará como se hace un login seguro
- Se implementará el login seguro en base a la investigación
- El login del usuario visita será implementado con un botón debido a que cualquier persona pueda ver los torneos públicos

##### i. Privilegios Relacionados con usuarios

1. El súper administrador tendrá una contraseña específica al inicio del sistema y un usuario específico con el cual podrá entrar y de esta manera hacer todos los cambios administrativos correspondientes.
2. El digitador será creado por el administrador de torneos, y le dará acceso para que únicamente registre los resultados de los partidos para ciertos torneos.
3. El usuario del equipo representativa será creado por el administrador de torneos que le dará acceso para ver el rendimiento de su equipo en los torneos.

#### 7. Implementar módulo de administrador de torneos

- Administrador de torneos puede:
  - i. Crear un torneo
  - ii. Establecer torneo como público o privado
  - iii. Crear Equipos para el torneo
  - iv. Crear usuario representante de equipo
  - v. Crear usuario digitador
  - vi. Ingresar datos a los partidos del torneo
  - vii. Editar torneo
  - viii. Editar Equipos
- Torneos públicos: Son los torneos de los que el usuario visita puede ver.
- Torneo privado: Son torneos que solo los usuarios con el acceso permitido pueden verlos.

#### 8. Implementar el módulo de generación de torneo

- Este módulo será implementado para generar cualquier tipo de torneo con su respectivo nombre del administrador de torneos correspondiente.
- Se generará el torneo y el administrador de torneos podrá elegir si el torneo es público o privado.

#### 9. Implementar el módulo de usuario digitador

- El administrador de torneos será el que pueda crear un usuario digitador para su torneo, el cual digitará los resultados de los partidos.

#### 10. Implementar el módulo de creación de equipos por torneo

- En este módulo se realizará la creación de equipos del torneo ya creado.
- El administrador de torneos creará los equipos si es que no han sido creados
- Si ya han sido creados en otro torneo por otro administrador tendrá la opción de importarlos

#### 11. Implementar el módulo de generación de partidos

- En este módulo se crearán los partidos que la aplicación automáticamente generará.
- El administrador de torneos podrá editar manualmente la generación de partidos si no le gustó el orden.
- El usuario digitador o el administrador de torneos ingresarán los resultados de los partidos.

#### 12. Implementar el módulo de tabla de posiciones

- En este módulo se pretende generar, según el orden de puntos, luego por puntos de diferencia y finalmente por puntos a favor para ordenar los equipos, y visualizar cuál de estos es el campeón y su rendimiento.
- Se seguirá las reglas del formato de liga para ordenar los equipos y seleccionar un campeón.

### 13. Probar la aplicación

- Se hará una prueba con los datos cargados en la base de datos.
- Se hará otra prueba cargando los datos directamente desde el sistema
- Se hará una prueba con cada usuario para establecer si la aplicación funciona como debería

### 14. Subir la aplicación

- Se debe poner la aplicación en línea luego de que esté probada para que la gente desde cualquier parte del mundo la pueda utilizar.
- Probar la aplicación en línea con creados en el momento.

## **Metas**

- 1 Investigar sobre los requerimientos de seguridad para poder implementar un login seguro.

Investigar sobre las tecnologías que se ofrecen para desarrollo web.

1. Identificar la tecnología más adecuada para realizar la aplicación, se debe considerar presupuesto, licencias, equipos para el desarrollo.
2. Implementar la funcionalidad de la aplicación, la generación de torneos con sus diferentes usuarios, ver el rendimiento de los equipos y mostrar el ganador.
3. Implementar la generación de los torneos según los equipos y automáticamente generar los partidos para todos los equipos en la liga. Si la generación de los partidos no es del agrado del usuario, éste puede cambiarlo manualmente.
4. Implementar la funcionalidad de mostrar la tabla de posiciones y mostrarla de manera entendible al usuario.
5. Implementar la funcionalidad de los distintos privilegios para los usuarios como son el súper administrador, el administrador de torneos, el usuario digitador, el usuario

visita y el usuario representante del equipo para que vean únicamente lo que estén autorizados a ver.

6. Integrar todos los módulos para realizar una aplicación completamente funcional para los usuarios.
7. Hacer pruebas con datos ficticios generados con la aplicación en línea para poder asegurar la funcionalidad total de la misma.

### **Actividades**

1. Diseño de la Base de Datos
1. Implementar la base de datos en línea
1. Preparar el entorno de desarrollo para empezar con la aplicación
2. Implementar el módulo de súper administrador
3. Implementar el módulo de Registro
4. Implementar módulo de administrador de torneos
5. Implementar el módulo de Login
6. Implementar el módulo de generación de torneo
7. Implementar el módulo de usuario digitador
8. Implementar el módulo de creación de equipos por torneo
9. Implementar el módulo de generación de partidos
10. Implementar el módulo de tabla de posiciones
11. Probar la aplicación
12. Subir la aplicación

## **Capítulo 2 - Marco Teórico**

### **Tecnología a ser Utilizada**

#### **Lenguajes de programación**

##### **Definición**

Un lenguaje de programación es una serie de instrucciones enviadas al ordenador para poder implementar un programa (Kernighan & Ritchie, 1988). Las instrucciones que son enviadas al ordenador son escritas por una persona que entiende el lenguaje de programación. Esta persona se denomina programador. En el mundo existen varias lenguas para que las personas de diferentes regiones se comuniquen entre sí. Del mismo modo, existen diferentes lenguajes de programación que se utilizan para programas en las computadoras. El campo de la informática es bastante amplio debido a la gran cantidad de lenguajes de programación que existen.

##### **Python**

Python es un lenguaje de programación orientado a objetos (Sanner, n.d.). Esto quiere decir que utiliza objetos y clases para poder crear un código mantenible. Un código mantenible es aquel que es de fácil lectura, donde un cambio es localizado de una forma modular. Python provee estructuras de datos de alto nivel que son diccionarios, listas y otro contenido dinámico que facilita el desarrollo de la aplicación (Sanner, n.d.). De esta manera, el programador puede ser mucho más productivo, concentrándose en el dominio de su aplicación. Python tiene librerías muy útiles para la aplicación que se propone desarrollar; la generación de torneos. Un torneo puede ser de ida y vuelta o solo de ida. Los torneos de ida y vuelta generan partidos de equipos por medio de permutaciones. Una permutación es el acto de reagrupar objetos o valores donde el orden importa. En el caso de los torneos de ida y vuelta se utiliza la permutación de equipos para crear partidos. Los torneos de ida generan partidos de equipos por medio de combinaciones. Una combinación es una

permutación en la que el orden sí importa. En particular, se utiliza la librería de Python llamada `itertools` para poder realizar combinaciones y permutaciones de los partidos de un torneo. Esta librería tiene varios métodos que utiliza para iterar información. Existe un método, `combinations`, para ejecutar combinaciones de objetos y un método, `permutations`, para ejecutar permutaciones de objetos. Las librerías de Python son bloques de código para que facilita la reutilización de funciones comunes.

## **Java**

Java es otro lenguaje de programación orientado a objetos (Lindholm, Yellin, Bracha, & Buckley, 2013). Java fue un programa desarrollado para que sus programas puedan correr en cualquier tipo de plataforma (Lindholm et al., 2013). Antes de Java, los programas eran escritos en un cierto tipo de plataforma y podían correr únicamente en esa plataforma. Si se quería correr el mismo programa en otra plataforma se necesitaba volver a portar el código y eso requería de esfuerzos adicionales. Como se requería de software que se ejecutara a través de redes de computadoras y que su código se ejecutará en cualquier otro computador sin importar la plataforma se inventó Java (Lindholm et al., 2013). Por esta razón, muchas personas prefieren programar en Java. A diferencia de Java, Python tiene una sintaxis más simple. Es decir, se puede escribir las mismas funciones que en Java, con menos palabras. El uso de sangrías en el formato para escribir código en Python permite que este código sea más legible. Las librerías en Python son mucho más completas que en Java. Python fue un lenguaje creado para científicos. Esto quiere decir que tiene más librerías matemáticas, que otros idiomas como por ejemplo las combinaciones y permutaciones que son sumamente necesarias para el desarrollo de una aplicación de generación de torneos.

## **Frameworks para desarrollar aplicaciones web**

### **Antecedentes**

Antes, desarrollar aplicaciones web era bastante tedioso. Los desarrolladores escribían todas y cada una de sus páginas en HTML que provocaba una pérdida de tiempo en el proceso de implementación (Holovaty & Kaplan-Moss, 2009). Debido a esto los desarrolladores inventaron un servidor web que podía hacer que los programas generaran páginas HTML dinámicamente llamado “CGI Common Gateway Interface”. (Holovaty & Kaplan-Moss, 2009). Sin embargo, CGI tenía el problema de que el código era difícil de entender y no se podía reutilizar (Holovaty & Kaplan-Moss, 2009). En programación, la reutilización y claridad de código es muy importante para crear aplicaciones mantenibles sin mayores esfuerzos. Además, CGI utiliza un proceso por cada cliente lo cual es muy costoso en recursos. Ahora, solo se necesita de un único hilo por cliente para responder a los procesos de varios clientes.

Así es como surgió PHP. PHP es un lenguaje de programación que se ejecuta del lado del servidor y luego muestra al cliente una página HTML (Welling & Thomson, 2003). Esto resolvió algunos de los problemas del desarrollo web, pero no todos. Debido a que el código de PHP se puede escribir dentro de la página HTML implica que los programadores no tenían que aprender mucho más que el lenguaje del que ya sabían (Welling & Thomson, 2003). Sin embargo, PHP tenía sus propios problemas con el código porque no respetaba el modelo MVC (Holovaty & Kaplan-Moss, 2009). Otro problema que tenía PHP es que no brindaba protecciones sobre vulnerabilidades de seguridad (Holovaty & Kaplan-Moss, 2009). Vulnerabilidades de seguridad como ataques de Denial of Service (DoS), falsificación de identificación y generación de claves por fuerza bruta. Denial of Service ocurre cuando se envían muchos pedidos al servidor. Si el servidor no puede manejar los pedidos, entonces éste se congela. La falsificación de identificación es hacerse pasar por otro usuario. La

generación de claves por fuerza bruta es un proceso iterativo que consigue descifrar la clave de un usuario. PHP, por ser un lenguaje del lado del servidor, tenía estas y muchas otras vulnerabilidades de seguridad. Cuando los desarrolladores aprendían sobre seguridad era demasiado tarde. Por eso se han desarrollado frameworks de seguridad para crear aplicaciones mantenibles y seguras.

## **Django**

Django es un framework para desarrollar aplicaciones web que utiliza Python como lenguaje de programación (Anisenkov et al., 2006). Existen muchas ventajas al utilizar un framework como Django para generar aplicaciones web en lugar de hacerlo archivo por archivo o con PHP. Una de las ventajas de Django es su propia configuración de archivos para conectarse a la base de datos (Holovaty & Kaplan-Moss, 2009). El archivo de configuración de la base de datos de Django es utilizado por todos los archivos de la aplicación. Si no existiera este archivo se tendría que implementar la configuración de la base de datos en todos y cada uno de los archivos del sistema. Esto sería ineficiente si fuese necesario una migración de la base de datos. En ese caso se tendría que cambiar la configuración en todos los archivos de la aplicación. Aunque se utilicen métodos de copiado y pegado, el desarrollador podría olvidarse de algún archivo, lo que puede provocar un error en el funcionamiento de la aplicación. Además, encontrar un error por la mala configuración podría tardar mucho tiempo, en especial si la aplicación tiene muchos archivos. Muchos de estos y otros problemas son los que Django resuelve y de esa manera se puede desarrollar aplicaciones seguras y mantenibles.

## **Base de Datos**

“Un Sistema de Manejo de Base de Datos (en inglés DBMS Database Management System) es un conjunto de elementos interrelacionados y una serie de programas que permiten a varios usuarios tener acceso a estos archivos ya sea para consultarlos o

actualizarlos” (León, 2008) . Es decir, es un conjunto de datos con cierta estructura. Muchas aplicaciones funcionan guardando datos que serán utilizados después, es por esto que se crearon las bases de datos. Las operaciones permitidas en una base de datos relacional son: la inserción, edición y eliminación de los datos. De esta manera se puede tener un mayor control sobre la información que provee la aplicación.

### **MySQL**

MySQL es la base de datos de código abierto más popular que existe (Widenius & Axmark, 2002). Una ventaja de MySQL es su código abierto que es totalmente gratuito. Otra ventaja es que varios expertos han revisado el código y lo han modificado para tener un rendimiento óptimo. El código abierto permite a otro programador hacer las modificaciones que crea que son necesarias y así puede evolucionar su producto hacia uno mejor. Además, existe bastante documentación para poder generar las sentencias SQL necesarias para la aplicación que se propone desarrollar.

### **ORM**

ORM significa en inglés Object Relationship Mapping (Plehanova, 2009). En otras palabras, quiere decir que a cada tabla de la base de datos las trata como a un objeto. Se tienen muchas ventajas cuando se utiliza un ORM. Por ejemplo, en Django todas las sentencias de insertar, editar, eliminar y el 90% de las sentencias SQL se pueden escribir a través del código, sin tener que utilizar el lenguaje de la base de datos (Plehanova, 2009). Por esta razón se podría utilizar cualquier base de datos si se necesitara cambiar el motor. La desventaja de solo utilizar un ORM provoca una pérdida de utilización de los datos en ciertas relaciones complejas. Sin embargo, Django facilita la flexibilidad de utilizar sentencias SQL crudas para operaciones más complicadas (Plehanova, 2009). Así es como un framework como Django simplifica el desarrollo de clientes de base de datos.