

Capítulo 4 – Análisis y Resultados

Al terminar la aplicación desarrollada con Django se han cumplido los objetivos planteados al principio de la propuesta. Los objetivos fueron planteados para cumplir con la funcionalidad de una aplicación que genere y administre torneos de fútbol. Los objetivos que se cumplieron durante la implementación de este proyecto fueron:

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

Meta	Cumplido
Investigar sobre los requerimientos de seguridad para poder implementar un login seguro.	Sí
Investigar sobre las tecnologías que se ofrecen para el desarrollo web.	Sí
Identificar la tecnología más adecuada para realizar la aplicación, se debe considerar presupuesto, licencias, equipos para el desarrollo.	Sí
Implementar la funcionalidad de la aplicación, la generación de torneos con sus diferentes usuarios, ver el rendimiento de los equipos y mostrar el ganador.	Sí
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.	Sí
Implementar la funcionalidad de mostrar la tabla de posiciones y mostrarla de manera entendible al usuario	Sí
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 esté autorizados a ver.	Sí
Integrar todos los módulos para realizar una aplicación completamente funcional para los usuarios.	Sí
Hacer pruebas con datos ficticios generados con la aplicación en línea para poder asegurar la funcionalidad de la misma.	Sí

Tabla 5: Metas Alcanzadas

Objetivo 1

El objetivo 1 se cumple ya que se hizo el diseño de la base de datos. En el diseño se pueden ver todas las entidades que van a ser usadas por la aplicación. Estas entidades son torneos, equipos, partidos y jornadas. Cada una de estas es utilizada por la aplicación. Además, se extiende el diseño de la base de datos debido a que Django agrega sus propias

tablas para los usuarios y sus respectivos permisos. Así se prepara el ambiente de desarrollo de la aplicación para poder alcanzar los siguientes objetivos.

Objetivo 2

El objetivo dos se cumple debido a que los datos ya están cargados en línea.

Objetivo 3

Debido a que se utiliza Django, este objetivo se cumple automáticamente. Django provee un módulo de súper administrador. Este módulo de Django permite la inserción, edición y borrado de los usuarios. También permite el cambio de los privilegios asignados a cada usuario. Al instalar Django se puede crear un súper administrador con su propia clave. Django otorga todos los privilegios al súper administrador sin tener que dárselos explícitamente.

Para poder darle permisos a de crear otras tablas de la base de datos resultó ser bastante sencillo. Lo único que se necesitaba hacer era registrar la tabla con el súper administrador. Django provee un archivo en el cual esto se puede hacer de manera fácil. De esta manera, las tablas creadas en la base de datos pueden ser insertadas, actualizadas y eliminadas sin ningún problema. La tecnología ayudó bastante con el módulo del súper administrador.

Objetivo 4

El objetivo 4 se alcanzó debido a una tecnología que se provee en Django. Existe un módulo que se llama Django-Registration. Este módulo permite hacer formularios para el registro de usuarios. En los formularios de registro el usuario ingresa los campos de usuario y contraseña. Tiene un campo de verificación de contraseña que el sistema ya se encarga de verificar si la contraseña fue correcta. También, codifica la clave para que no se guarde en la base de datos. Esto es una medida de seguridad para que si alguna persona entra a la base de datos no pueda descifrar la clave ya que está encriptada. Este es un módulo seguro de registro

debido a que se utilizan en muchas aplicaciones web y es un módulo hecho por los desarrolladores de Django.

Lo que se modificó en la lógica a este módulo fueron los permisos. El administrador de torneos debe tener todos los permisos para sus torneos, partidos y equipos. Así que al momento que una persona se registra como administrador de torneos obtiene todos estos permisos. Sin embargo, solo tiene permisos para los equipos que el usuario cree. Si otro usuario crea otro torneo con otros equipos entonces el primer usuario no podrá ver que dichos torneos.

Objetivo 5

Este fue el objetivo más difícil de completar. Este objetivo requería de varios otros para que esté completo en su totalidad. El administrador de torneos debía crear un torneo, crear equipos para ese torneo, generar los partidos y ver la tabla de posiciones. La segunda actividad que debía tener el administrador de torneos era poder crear usuarios digitadores, usuarios representantes de equipos y usuarios visitas. Al final, los usuarios visitas se crearon como usuarios anónimos. Estos usuarios anónimos son los usuarios que no necesitaban de hacer login. En el sistema se presenta un link de torneos públicos. Este link es el único acceso que tienen los usuarios anónimos.

Objetivo 6

Debido a que Django ya ofrece un modelo de registro, también ofrece el modelo de login. Django provee un formulario para el usuario en el cual verifica si el usuario y la contraseña son correctos. Si no lo son entonces el usuario no entra al sistema. Si el usuario si es correcto entonces entra a la página principal. Sin embargo, solo puede acceder a los módulos del sistema que el usuario tenga permiso y eso depende de qué tipo de usuario es.

Objetivo 7

El módulo de generación de torneo fue un objetivo bastante sencillo de cumplir. Debido a que Django tiene un ORM no fue necesario de mucha programación para realizar las inserciones en las bases de datos. Los formularios proveídos por Django modelan la clase Torneo y despliega toda la información necesaria que el usuario debe llenar. Al hacer clic en crear, se crea el torneo. Es decir, se inserta en la base de datos sin ningún problema.

Además aquí se le agrega el permiso al usuario a este torneo. Esto se utilizando la tecnología Django-Guardian que fue instalada para alcanzar este objetivo. Django-Guardian permite dar permisos según la instancia de un objeto. Entonces cuando se crea un torneo, solo el usuario que creó el torneo tiene permisos para verlo.

Objetivo 8

El objetivo 8 es parte del objetivo 5. Para completar este objetivo se utilizaron dos módulos de Django. El primer módulo fue el de Django-Registration. Este módulo es el que permite hacer el registro al sistema. Sin embargo, aquí es donde al usuario digitador se le debía otorgar el permiso para el cierto torneo. Lo que se hizo primero fue generar un usuario con un tipo de permiso. Este permiso era “usuario_digitador”. Esto se utiliza después para desplegar todos los usuarios digitadores que existen. El administrador de torneos es el encargado de asignarle los permisos al usuario digitador. Los usuarios digitadores aparecen en pantalla. El administrador de torneos puede asignar uno de sus torneos a alguno de los usuarios digitadores. Este usuario digitador podrá únicamente actualizar los resultados de los partidos del torneo al cual fue asignado. No tendrá opción crear torneos, equipos o editarlos en el sistema. EL objetivo fue alcanzado y probado de manera eficaz. De la misma manera se hizo con el usuario representante de equipos, solo que este usuario solo puede ver la tabla de posiciones no tiene permiso de editarla.

Objetivo 9

Este objetivo fue bastante fácil de alcanzar. Se hace de manera muy similar al módulo de creación de torneos. Lo único distinto es que se debió seleccionar un torneo al cuál agregar los equipos. Se utilizó el ORM de Django, los formularios de Django que modela la tabla para que el usuario ingrese los datos.

Objetivo 10

Este objetivo fue el más largo y más difícil de desarrollar. Esto es debido a que la mayor parte de la funcionalidad se aplica aquí. La generación de partidos se hizo automática y manual. La generación automática consiste en lo siguiente: si un torneo es de sólo ida se hacen combinaciones para generar los partidos. Es decir el equipo A jugará contra el equipo B una sola vez. Sin embargo, si un torneo es de ida y vuelta se hacen permutaciones para generar los partidos. Es decir, el equipo A jugará contra el equipo B y después jugará el equipo B contra el equipo A en dos partidos distintos. Se logró implementar esto a través de las librerías creadas por Python que ya tienen implementadas las funciones para hacer combinaciones y permutaciones. Luego de generar los partidos se los presenta al usuario. Debido a que los partidos consisten de dos equipos el sistema verifica el número de equipos que existe en el torneo. Si no existe ningún equipo en el torneo, el sistema le informa al usuario que debe registrar equipos en su torneo para poder generar partidos. Si el torneo consta de un solo equipo el sistema le advierte al usuario de que por lo menos agregue otro equipo en el torneo. A la final, el sistema cuenta con al menos dos equipos por torneo para poder generarlos.

La segunda forma de generar los partidos es de forma manual. Un usuario administrador de torneos puede insertar partidos. Al insertar los partidos el sistema verifica que no inserte partidos donde un equipo juegue contra sí mismo. El usuario tiene la responsabilidad de insertar los partidos que guste y que desee. Después de que los partidos

sean generados manual o automáticamente el usuario administrador de torneos o el usuario digitador también puede actualizar los resultados. Una vez actualizado los resultados el partido desaparece de la lista de partidos y no se puede volver a actualizar. Este borrado no afecta a la tabla de posiciones.

Objetivo 11

Este objetivo se realiza después de cumplir el objetivo anterior. Para crear una tabla de posiciones se necesita la generación de partidos. La tabla de posiciones consiste en: partidos jugados (PJ), partidos ganados (PG) , partidos empatados (PE) , partidos perdidos (PP), goles a favor (GF), goles en contra (GC), goles diferencia (GD) y puntos obtenidos (Pts). El sistema hace una búsqueda en la base de datos de los partidos. Los partidos que han sido marcados como jugados son los únicos que se toman en cuenta. Al crear partidos, se crea automáticamente como partidos no jugados. Al actualizar el resultado de los partidos se cambia el estado a partidos jugados.

Debido a que la base de datos tiene una tabla partidos con equipo local y equipo visitante, fue muy fácil filtrar los datos. Se escribió un algoritmo para poder generar la tabla de posiciones. Para los partidos jugados simplemente se cuenta cuantos partidos jugados existen en el torneo. Para el resto de puntos se hace un proceso ligeramente distinto. Primero se filtran los equipos locales. De los equipos locales se filtran los goles que tuvieron como locales. Si el gol de casa es mayor al gol de visitante entonces se le otorga al equipo local un partido ganado. Si el gol de casa es menor al gol de visitante se le otorga un partido perdido. Si los goles son iguales entonces se le otorga un partido empatado al equipo local. Del mismo modo se les otorgan los partidos a equipos visitantes.

Para los goles existe una lógica ligeramente diferente. Se filtran los equipos locales en la base de datos. De esos equipos locales se suma los goles que se hicieron localmente. El mismo equipo local pudo ser visitante. Después de filtrar como visitante se suma los goles

que convirtieron como visitantes los equipos. Una vez que se tiene los goles convertidos como locales y visitantes se los suma. De esta forma es como se obtiene los goles a favor de cada uno de los equipos. Los goles en contra se generan de la misma manera, pero en vez de sumar los goles convertidos se suman los goles recibidos.

La generación de goles diferencia y puntos fue la parte más fácil de la implementación. Los goles diferencia es simplemente la resta entre goles a favor y goles en contra. La fórmula para calcular los puntos es la siguiente: $\text{partidos ganados} \times 3 + \text{partidos empatados}$. Es decir se obtiene tres puntos por partido ganado y un punto por partido empatado. No se obtiene puntos por partidos perdidos.

Se utilizó una lista de diccionarios de Python para generar la tabla de posiciones. Esta lista de diccionarios ordena la tabla de posiciones de los partidos por puntos, goles diferencia y luego goles a favor. Esto se utiliza para hacer desempates si los equipos empataran por puntos. Si empatan por puntos, gana el equipo de mayor gol diferencia. Si están empatados por puntos y por gol diferencia gana el que el equipo que convirtió más goles.

Objetivo 12

Este objetivo no es difícil pero es el más necesario. Después de desarrollar una aplicación es necesario probarla. Se hizo pruebas de casos comunes. Los casos comunes son el número de equipos que existen en las ligas. Normalmente existen de cuatro, doce, y veinte equipos. Se utilizó torneos reales para probar la aplicación. Las pruebas fueron un éxito total ya que la funcionalidad del sistema alcanzó los objetivos planteados. Se resolvió los casos de empate como debían ser. El único problema fue al generar permutaciones de veinte partidos el sistema tardó unos treinta segundos en generarlos. Sin embargo, la generación de la tabla de posiciones no demora nada.

Objetivo 13

En este se compró un VPS (Virtual Private Server) que es una máquina virtual de Linux. Se puede tener acceso a través de la IP. El costo del VPS es de \$5 mensuales.

Debido a que es muy poco común que la gente se acuerde de un número para entrar a una aplicación se compró un dominio. El dominio tiene un costo de \$10 anuales. El dominio que se compró fue tourngen.com.

Capítulo 5

Conclusiones

- Todos los objetivos planteados en la propuesta para la generación de una aplicación web que pueda generar y administrar partidos de fútbol fueron cumplidos.
- Todos los usuarios del mundo con acceso a internet cuentan con una interfaz para generar y administrar sus torneos de fútbol. Así pueden crear torneos nacionales, internacionales, institucionales, regionales y virtuales.
- La interfaz cuenta con un módulo para la creación de distintos usuarios que tienen diferentes privilegios. Tiene un súper administrador quien tiene el control de todas las interfaces y todos los privilegios. Un administrador de torneos que se puede registrar solo y puede crear sus propios torneos públicos o privados. Un usuario digitador que es creado por el administrador de torneos y tiene permiso únicamente de cambiar los resultados de algún torneo que se le haya asignado. Un usuario representante de equipos que es creado por el administrador de torneos y tiene permiso solamente de visualizar la tabla de posiciones de un equipo que se le haya asignado. Un usuario visita o usuario anónimo que puede ver la tabla de posiciones de todos los torneos públicos.
- La generación de partidos de esta aplicación puede ser automática. Es decir, que los partidos que se generen para los torneos de un formato de liga son creados por permutaciones o por combinaciones dependiendo del tipo de partido.
- La generación de los partidos puede ser manual. En otras palabras, el administrador de torneos puede ingresar un nuevo partido a un torneo. De esta manera puede agregar más partidos de los que se crean automáticamente y se toman en cuenta para crear la tabla de posiciones.

- El diseño de la funcionalidad de la aplicación presentado por los casos de uso fue muy útil para su implementación. Sin los casos de uso, el diseño de la funcionalidad del sistema y alcanzar los objetivos no hubiese sido posible.

Recomendaciones

- Antes de desarrollar cualquier aplicación, es necesario tener bien definidos los requerimientos de la misma. En este caso los requerimientos fueron obtenidos por la propuesta. Definiendo detalladamente cada uno de los procesos de la propuesta y alcanzando todo los objetivos fue como se pudo realizar la tesis. Es sumamente necesario definir los requerimientos para establecer el alcance de un proyecto con el cliente. Para proyectos futuros con clientes se debe hacer un análisis detallado del proyecto y firmar un contrato para establecer únicamente funcionalidades acordadas. En caso de que se requiera una nueva funcionalidad, que no esté entre las acordadas, se deberá hacer un nuevo contrato cambiando el alcance, el costo del proyecto y agregando la nueva funcionalidad.
8. Es recomendable tener un amplio conocimiento sobre las tecnologías que se utilizan para poder desarrollar una aplicación. En esta aplicación se utilizó Python como lenguaje de programación ya que se utilizó el framework de Django. Se hizo un estudio con varios tutoriales sobre Django hasta conocerlo profundamente. Habían muchos módulos, como el de registro de usuarios, que se podían implementar de manera fácil y sencilla con Django pero había que conocer el framework. Al tener un amplio conocimiento sobre las tecnologías, la implementación de la aplicación es más factible.
- La base de datos no fue un problema en el momento del desarrollo porque se utilizó el ORM de Django. Sin embargo, se debe tener varias consideraciones con la base de datos. Debido a que es una aplicación web, es una aplicación que necesita

mantenimiento y además debe ser escalable. Se debe saber qué hacer si la base de datos crece exponencialmente en el futuro. ¿Qué hacer con los torneos? Se recomienda establecer un límite de torneos que se permitan guardar. También, cuando se elimine un torneo exportarlo a un archivo CSV para quede el reporte histórico. Establecer un espacio de memoria para estos archivos históricos. Tener un plan de contingencia por si se acaba el espacio en el VPS. Se sugiere hacer un sistema de tolerancia ante fallos para que la aplicación no se deteriore en un futuro. Aún con todas las prevenciones posibles, puede resultar que exista un fallo. Se debe estar preparado para cualquier tipo de fallo y poder solucionarlo lo más rápido posible. La aplicación web tendrá que tener mantenimiento a lo largo de su existencia.

- Es de suma importancia tener una tolerancia de fallos sobre todo si la base de datos se perjudica. Uno de los eventos catastróficos que se debe considerar es la eliminación completa de la base de datos. Se debe definir un espacio de memoria para recuperar la información. Se debe definir si se debe guardar la información en un lugar aislado a la aplicación para mayor seguridad de los datos. También lo que se debe establecer es la frecuencia a la que se deben guardar los datos. Cualquier cosa puede suceder en el caso de la informática y se debe estar preparado para todo. Aunque prepararse completamente no es posible porque siempre existe el factor sorpresa, se debe mantener siempre un plan de contingencia para los mayores desastres.
- La eficiencia es una parte fundamental de las aplicaciones web. Al usuario no le gusta esperar. La generación de partidos cuando son 32 equipos, toma treinta segundos y debería solucionarse. Una solución es hacer templates con tipos de torneos comunes predeterminados. De esta manera, el sistema solo actualizará el nombre de los

equipos y no tendrá que desperdiciar tiempo en crear permutaciones o combinaciones.

Desarrollo Futuro

- Esta aplicación tiene un gran potencial futuro por las siguientes razones: se puede extender a crear cualquier tipo de torneo, se podría obtener datos de torneos reales creando tablas de posiciones en tiempo real y se podría incluir una tabla de goleadores por torneo.
 - Por el momento solo utiliza torneos que son de formato de liga. En el futuro se podría realizar que esta aplicación haga cualquier tipo de torneo. En los mundiales hay una fase de grupos y de esa fase de grupos se seleccionan equipos clasificados para competir en un campeonato de eliminación directa. Al finalizar el campeonato de eliminación directa se obtiene el ganador del mundial.
- 2 Se podría implementar para que un usuario pueda crear un torneo de eliminación directa. Se debería crear una forma para que el usuario pueda elegir qué tipo de torneo crear.
- Hay muchos torneos los cuales los datos se presentan en internet en tiempo real. A la gente le gusta tener toda la información en una solo lugar. Es por eso que se podría obtener datos de torneos reales que están siendo actualizados minuto a minuto. Luego de obtener esos datos se podría crear automáticamente el torneo en el sistema. Y así, el sistema podría generar la tabla de posiciones. Los datos se podrían obtener con web services. Y a estos torneos se los podría asignar como públicos para que cualquier persona pueda visualizar la tabla de posiciones de su equipo favorito.
 - En los torneos es muy importante el rendimiento de los jugadores. Por eso es que se debería implementar una tabla de jugadores. En el fútbol, los jugadores hacen goles, obtienen tarjetas amarillas y tarjetas rojas. En un torneo es de suma importancia saber

qué jugador hizo más goles o qué jugador tiene una tarjeta amarilla o roja. Debido a que si tiene dos tarjetas amarillas en un partido , no puede jugar el siguiente. Así, en el sistema se podría controlar qué jugadores son los que pueden jugar el siguiente partido y además tener una estadística de quien fue el mejor jugador en el torneo. Es decir, quien convirtió más goles.