

MISTER APP

Autor: Javier Moral Garrido
Tutor: Mario Marugán Cancio

Ciclo formativo de desarrollo de
aplicaciones multiplataforma

Curso 2019-2020

IFP Madrid

Mayo, 2020

Resumen

Hoy en día el análisis de datos es clave, y en el mundo del fútbol no iba a ser menos, se guardan todos los datos de las grandes competiciones profesionales y algunas semi profesionales, pero ¿y el resto? El fútbol profesional es solo una pequeña parte de todo lo que genera este deporte, hay una gran infinidad de equipos y divisiones no profesionales o de aficionados, y de todos estos equipos no se guarda ningún tipo de estadísticas más que las victorias y los goles, y creo que cualquier equipo que quiere competir se debe conocer mejor y saber con más detalles sus puntos débiles y fuertes. Por eso hay muchos entrenadores que apuntan las estadísticas, a papel, de cada partido.

Por esto último, este trabajo va a ser una aplicación que permita coger las estadísticas de cada partido de tu equipo desde el teléfono móvil del entrenador. La aplicación contará con un apartado de gestión del equipo donde se podrá añadir, eliminar y modificar los jugadores del equipo, también habrá un registro donde se guarden todos los partidos disputados y se pueda acceder a cada partido para verlo con más detalle. El usuario tendrá una vista sencilla a la hora de anotar una estadística, puede ser una estadística de tu equipo, o del equipo contrario. El usuario tendrá que elegir el tiempo del partido, y los datos de cada usuario serán exclusivos del mismo.

Al realizar una aplicación para teléfonos móviles el trabajo se realizará desde Android Studio, siendo exclusiva de Android, el lenguaje usado será Java, y la base de datos a usar será Firebase, una base de datos en la nube que nos permite realizar consultas potentes y en tiempo real.

Índice general

Índice de Figuras

<u>1. Introducción</u>	1
<u>1.1 Introducción al tema</u>	1
<u>1.2 Estructura del documento</u>	2
<u>2. Justificación del proyecto</u>	3
<u>2.1 Elección del proyecto</u>	3
<u>2.2 Objetivo principal</u>	3
<u>2.3 Objetivos parciales</u>	3
<u>2.4 Medios utilizados</u>	3
<u>2.4.1 Hardware</u>	3
<u>2.4.2 Software</u>	3
<u>3. Estado del arte</u>	5
<u>3.1 Evolución en el desarrollo de aplicaciones móviles</u>	5
<u>3.1.1 Android Studio</u>	6
<u>3.1.2 Java</u>	7
<u>3.1.3 Flutter</u>	7
<u>3.2 Modelo-Vista-Controlador (MVC)</u>	8
<u>3.3 Bases de datos</u>	9
<u>3.3.1 Bases de datos Relacionales</u>	9
<u>3.3.1.1 SQLite</u>	9
<u>3.3.1.2 Realm</u>	10
<u>3.3.2 Bases de datos no SQL</u>	10
<u>3.3.2.1 Firebase</u>	10
<u>3.4 Aplicaciones similares</u>	11
<u>3.4.1 Mis Estadísticas de Fútbol</u>	11
<u>4. Metodología de trabajo</u>	12
<u>4.1 Metodología ágil SCRUM</u>	12
<u>4.2 Evolución por fases</u>	13
<u>4.2.1 Fase I</u>	13
<u>4.2.1 Fase II</u>	13
<u>4.2.1 Fase III</u>	13
<u>4.2.1 Fase IV</u>	13
<u>4.2.1 Fase V</u>	14
<u>4.3 Entorno tecnológico</u>	14
<u>4.3.1 Herramientas para la gestión de proyectos</u>	14
<u>4.3.1.1 Github</u>	14

4.3.2 Herramientas para el modelado del software y la elaboración de la documentación	14
4.3.2.1 Draw.io.	14
4.3.3 Herramientas y tecnologías para el desarrollo del proyecto	14
4.3.3.1 Android Studio	14
4.3.3.1 Java	14
4.3.3.1 Firebase.	15
5. Resultados	16
5.1 Iteración 0	16
5.1.1 Plan de proyecto	16
5.1.1.1 Ambitos del sistema	16
5.1.1.2 Vision general.	17
5.1.1.3 Perspectiva del producto.	17
5.1.1.4 Restricciones	17
5.1.2 Viabilidad	18
5.1.2.1 Viabilidad tecnica.	18
5.1.2.1 Viabilidad legal.	18
5.2 Iteración 1	19
5.2.1 Administrar equipo	19
5.3 Iteración 2	19
5.4 Iteración 3	20
5.5 Iteración 4	21
5.6 Iteración 5	21
5.7 Iteración 6	21
6. Conclusiones y propuestas	22
6.1 Conclusiones	22
6.2 Propuestas futuras	22
Bibliografía	23

Índice de figuras

3.1 : Ciclo de vida de una actividad	6
3.2 : Funcionamiento MVC	8

1. Introducción

1.1 Introducción al tema

Se ha elegido realizar esta aplicación como proyecto porque solo se lleva un registro de partidos en las ligas profesionales y algunas semi profesionales. Dichas ligas es solo una pequeña parte de lo que hay en España por lo menos, y todos esos equipos que están en divisiones más bajas no pueden ver sus propias estadísticas más detalladas, y si quisiera llevarlas tendría que ser a mano, como sucede en mi equipo, por ejemplo, y digitalizarlo lo haría más cómodo, es uno de los motivos que han dado a pie a decantarme por realizar esta aplicación. A demás, al llevar el registro en el teléfono móvil, es más sencillo acceder a los datos y se puede acceder en cualquier momento a diferencia de un registro guardado a papel. El objetivo de esta aplicación es poder guardar las principales estadísticas de los jugadores y de los partidos en tiempo real, y también poder acceder a estas estadísticas cuando el usuario lo desee, con el fin de que los entrenadores que no tienen acceso a este tipo de estadísticas lo tengan. Y de esta manera poder ayudarles a llevar una mejor gestión de su equipo, un ejemplo sería que en el último partido se hubieran registrado una gran cantidad de tiros, pero solo unos pocos se dirigieran a puerta. Sabiendo esto, el entrenador puede preparar ejercicios de disparos a portería de cara al próximo partido, y así intentar ser más efectivos.

Una característica que destaca en esta aplicación es la sencillez cuando el partido está en marcha. El usuario no puede perder mucho tiempo apuntando un dato, por eso se intenta hacer de una forma fácil y sencilla, pudiendo, simplemente, apuntar un dato a un jugador y a un partido con 3 toques a la pantalla, dependiendo de si se anota una estadística a favor, donde es necesario guardar quien la ha llevado a cabo, o si es una estadística en contra que solo se apunta el dato.

La aplicación también cuenta con la posibilidad de ver gráficos estadísticos y tablas de su equipo en diferentes momentos de la temporada, los últimos cinco partidos, si encaja o marca más goles jugando de local o de visitante. De esta manera el usuario ve cómo se desarrolla y evoluciona su equipo y se pueden ver datos interesantes como quien es el jugador que más faltas recibe de tu equipo, o al que más tarjetas le sacan, los máximos goleadores y asistentes de tu equipo, entre otras estadísticas. Puede ser una aplicación que ayude a cualquier equipo, porque al tener acceso a las estadísticas del equipo se puede ver que está pasando, el estado del equipo, si algo no funciona en los últimos partidos, si el equipo necesita mejorar una faceta específica del juego, o cosas como estas que se pueden mejorar en dinámica de equipo, y que pueden ayudar a la preparación de próximos enfrentamientos.

Hay que declarar que los usuarios también podrán gestionar la creación, modificación y eliminación de los jugadores de su equipo, las estadísticas de cada jugador no podrán ser modificadas por el usuario, la única manera de eliminar alguna estadística de un jugador sería eliminando un partido en el que haya jugado, eliminándolo se eliminarán las estadísticas correspondientes a ese partido. Los partidos no pueden ser modificados dado que la aplicación está hecha para coger las estadísticas en tiempo real, pero si pueden ser eliminados posteriormente si así lo desea el usuario.

Esta aplicación no podrá ser usada sin haber iniciado sesión previamente, el usuario puede modificar el nombre del club y los minutos de sus partidos. También podrá eliminar el

club y volver a crear uno iniciando sesión de nuevo, una cuenta solo va a estar enlazada con un equipo.

Sabiendo el tema de la realización del proyecto, tenía muy claro el entorno y el lenguaje a usar, decidiendo así Android Studio como entorno, dado que la aplicación sería desarrollada en el sistema operativo de Android que es el más usado en todo el mundo actualmente, y Android Studio es el entorno con más soporte en el desarrollo de aplicaciones Android, es también el recomendado por Google y fue creado principalmente para desarrollar aplicaciones móviles en Android. El lenguaje será Java, uno de los más populares a nivel mundial y para el desarrollo en aplicaciones móviles igual. La metodología aplicada sería una metodología ágil, para tener una mayor adaptación a diferentes circunstancias que pudieran ocurrir, y ser más flexibles respecto a esto.

1.2 Estructura del documento

En este apartado se mostrarán de forma resumida lo que se encontrará en cada punto de los apartados del documento.

1. **Introducción:** En este primer apartado se dará una idea del tema del que se trata en este TFG, y del entorno utilizado para su realización.
2. **Justificación del proyecto:** En la siguiente sección se darán a conocer los motivos por los cuales se ha decidido hacer este proyecto, además de los objetivos a los que se pretende llegar con esta aplicación.
3. **Estado del arte:** En este apartado se encontrarán términos teóricos sobre el desarrollo de la aplicación.
4. **Métodos de trabajo:** A continuación, se mostrará la metodología de trabajo utilizada, y las fases que se han realizado para completar este proyecto.
5. **Resultados:** En esta sección se mostrarán los resultados de tras usar la metodología indicada.
6. **Conclusiones:** En este apartado se mostrarán las conclusiones tras la realización de los objetivos y se propondrán mejoras que se pretenden implementar más adelante.
7. **Bibliografía:** Por último, se hará referencia a los libros o artículos citados en el proyecto, y la bibliografía consultada.

2. Justificación del proyecto

2.1 Elección del proyecto

Uno de los principales motivos para la elección de este proyecto ha sido que juego en un equipo de fútbol aficionado, y nuestro entrenador recoge a papel las estadísticas de cada partido, el desarrollar esta aplicación puede ser una forma más cómoda y sencilla de anotar estadísticas a demás recogería más datos, y más específicos, como es el caso de estadísticas individuales para cada jugador. Otra motivación ha sido el poder ser capaz de desarrollar una aplicación móvil por mí mismo, siempre quise hacer una por mi cuenta, tenía muchas ganas de hacerlo.

2.2 Objetivo principal

El objetivo principal de este proyecto es el desarrollo de una aplicación que permita a los entrenadores de fútbol semi profesionales y aficionados tener guardadas las estadísticas de su equipo y jugadores. Y de esta manera poder ayudar a la planificación de su equipo.

A parte del objetivo principal hay otros parciales que se mostrarán a continuación.

2.3. Objetivos parciales

Los objetivos parciales que se han desarrollado en la realización del proyecto han sido:

1. Seguir aprendiendo y desarrollando conceptos sobre el lenguaje Java y el entorno Android Studio.
2. Aprendizaje de la base de datos Cloud Firestore de Firebase y del control de versiones Git.
3. Realizar el diseño de una Base de Datos para el almacenamiento de los datos en la aplicación.
4. Desarrollo de una aplicación con las tecnologías mencionadas.

2.4 Medios utilizados

2.4.1 Hardware

- Ordenador portátil HP elitebook 8440. 4 GB de ram 225 GB de SSD, y procesador intel i5 CPU.

2.4.2 Software

- Android Studio v3.5.1 como entorno de desarrollo utilizado.
- Java v8.
- Cloud Firestore de Firebase como base de datos en la nube.

- Control de versiones Git, ayudado de GitHub como medio de almacenamiento del proyecto, con actualizaciones conforme se iba avanzando en el proyecto.

3. Estado del arte

En el presente punto se expondrán el estado actual de las tecnologías usadas en este proyecto. Primeramente, se explicará la evolución en el desarrollo de aplicaciones móviles y las diferentes opciones que había para la realización del proyecto. Seguidamente se explicará el patrón Modelo-Vista-Controlador que ha sido el utilizado en la realización del presente proyecto. Y finalmente se comparará con aplicaciones con una función similar a la de la presente aplicación, la toma de estadísticas, en directo, de un partido de fútbol.

3.1 Evolución en el desarrollo de aplicaciones móviles

El desarrollo en aplicaciones para dispositivos móviles se ha visto disparada en esta última década, gracias, en parte, a la gran aceptación que han tenido este tipo de dispositivos en nuestra vida diaria, llegando al punto de ser imprescindibles para algunas personas en determinadas circunstancias.

En la actualidad existen aplicaciones, para dispositivos móviles, de cualquier cosa y para cualquier utilidad, aunque no siempre se adaptan y abarcan todas las necesidades de todos los usuarios, y es por esto, que es fácil encontrar aplicaciones con una misma función general, pero que pueden ser muy diferentes en cuanto a funcionalidad y a la manera de llevar a cabo dicha función. Por esto la gran versatilidad a la hora de crear diferentes aplicaciones que se adapten a las necesidades de cualquiera es un gran rasgo que tienen las aplicaciones móviles. Este tipo de aplicaciones sirve para ayudar a las personas a realizar la función que deseen de la manera más fácil posible, y estas aplicaciones se usan tanto, y tienen tanto éxito, en parte, porque están al alcance de nuestra mano en prácticamente cualquier situación.

A demás debido a la gran adaptación y aceptación de estas aplicaciones nativas, las empresas están empezando a crear aplicaciones híbridas, que son adaptaciones de aplicaciones web para móviles.

Las aplicaciones para Android hoy en día se desarrollan en torno a las actividades, estas son la toma de contacto entre usuario y la aplicación. Consiste en una pantalla que es la que verá el usuario, y el desarrollador es el encargado de crear la interfaz de dicha pantalla, añadiendo diferentes componentes que tiene el desarrollador para crear la interfaz deseada. A continuación, en la figura [3.1](#) se muestra el ciclo de vida de una actividad, esto son las diferentes fases por las que puede pasar la actividad de una aplicación durante el tiempo en el que usuario este usándola:

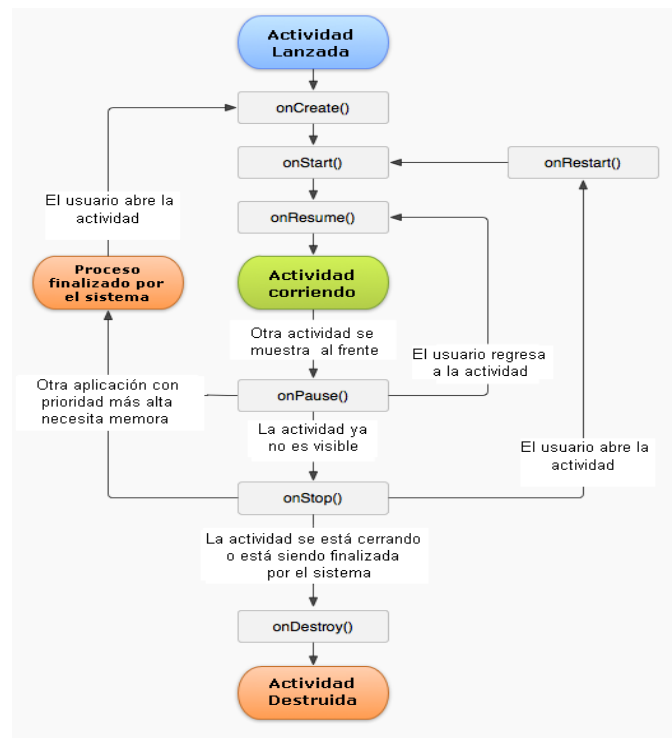


Figura 3.1: Ciclo de vida de una actividad. Imagen extraída de:
<https://ismaelfcf.wordpress.com/2013/07/01/actividades-y-su-ciclo-de-vida/>

Como se ha comentado antes, normalmente al desarrollar en aplicaciones móviles para Android se va a usar el patrón Modelo-Vista-Controlador, donde a través de la vista el usuario será capaz de interactuar con la aplicación y de mandar eventos al controlador, donde se encuentra la parte lógica del programa pudiendo comunicarse al modelo que es capaz de cargar datos de la base de datos.

A continuación, se presentarán las principales herramientas de las que disponemos hoy en día para poder llevar a cabo el desarrollo de aplicaciones móviles para Android, en los que destacan AndroidStudio, Flutter,

3.1.1 Android Studio

Este entorno de desarrollo para crear aplicaciones nativas para dispositivos móviles Android, también es el recomendado por Google. Tras el lanzamiento del primer mercado de aplicaciones en Android, en 2008, conocido como Android Market (renombrado a Play Store en 2012), por parte de Google, se empezaron a desarrollar algunas aplicaciones para dispositivos Android, pero por aquel entonces la utilidad y las capacidades que tenían los móviles no era comparable a lo que se vive hoy en día y la variedad de aplicaciones no era muy grande. Tras el paso de unos pocos años, la tecnología en dispositivos móviles se incremento exponencialmente y cada vez había mas demanda para crear aplicaciones móviles, entonces se usaba Eclipse como IDE predefinido para desarrollar este tipo de aplicaciones que, aunque no estaba enfocado en este tipo de aplicaciones, se podían desarrollar. Entonces en 2014 se desarrolló un IDE que estaría pensado para el desarrollo de estas aplicaciones móviles, y fue Android Studio, y así Eclipse fue reemplazado como IDE oficial para este desarrollo.

Este entorno nos proporciona una ejecución en tiempo real de la aplicación gracias a que se puede ejecutar en nuestro dispositivo Android o también podemos crear emuladores con diferentes tamaños de pantalla y diferentes versiones de Android, para así poder ver como se ejecutaría la aplicación en otro tipo de dispositivos.

También es capaz de ejecutar las compilaciones de forma muy rápida debido a que no es necesario reiniciar la aplicación cada vez que ejecutamos o depuramos. El principal problema de este IDE es que consume una gran cantidad de recursos si lo comparamos con lo que consumirían otros IDEs. A continuación, se muestran algunas de sus características principales:

- Integra la herramienta Grandle, la cual automatiza y gestiona nuestros proyectos.
- Tiene la capacidad de incluir el control de versiones con Git, Mercurial o Github accediendo a sus repositorios.
- Incorpora un sistema de alertas por errores sintácticos en tiempo real, o de estabilidad de la aplicación antes de ejecutarla.
- Proporciona la capacidad de mostrar una vista previa en el diseño de las aplicaciones.
- Integración con la nube mediante Cloud Plataform, en el cual se encontrarán diferentes herramientas para desarrollar las aplicaciones.

3.1.2 Java

Como lenguaje de programación para programar en Android Studio se encuentra Java, es un lenguaje multiplataforma y uno de los lenguajes oficiales para programar en este entorno. Cuenta con un gran soporte dado que es uno de los lenguajes más usados en todo el mundo. Sus principales características son:

- Código robusto, quiere decir que gracias a que es un lenguaje orientado a objetos, es complicado que se corrompa la memoria del dispositivo ni datos de otras aplicaciones o del sistema operativo.
- Gestión eficiente y automática de la liberación de memoria.
- Al ser orientado a objetos, permite crear componentes de una forma más sencilla para ser reutilizados.
- Son aplicaciones seguras, dado que no se accede a zonas delicadas de memoria o de sistema.

3.1.3 Flutter

Flutter es un SDK para el desarrollo de aplicaciones híbridas, que han sido comentadas anteriormente, hoy en día todavía está creciendo y no tiene mucho soporte, pero según los expertos, será el sustituto de Android Studio. Algunas de sus características son:

- La simplicidad que tiene y las facilidades que aporta, está pensado para que programar sea mas sencillo y más agradable para el programador, con implementaciones de desarrollo ágil para ver cambios de forma instantánea. Además aporta facilidades como implementar componentes en las aplicaciones.
- Rendimiento nativo, aporta widgets que hacen más ameno el uso de la aplicación, gracias a que está lleno de animaciones para las aplicaciones.
- Alto rendimiento y velocidad en las aplicaciones en las que se usa.

3.2 Modelo-Vista-Controlador (MVC)

En este apartado se explicará cómo funciona el flujo de ejecución usando este patrón MVC. En la imagen [3.2](#) se puede observar de forma gráfica su funcionamiento. En primer lugar se explicará en que consisten las diferentes partes del patrón y su función:

- **El modelo:** Tiene acceso a la base de datos y se encarga de manejar los datos de la aplicación y de relacionarlos con la base de datos. También es el encargado de realizar consultas hacia la base de datos.
- **La vista:** Es el responsable de mostrar los datos al usuario, de tal manera que pueda interactuar con la aplicación. La vista es capaz de mostrar los datos de la aplicación, pero no los puede manejar.
- **El controlador:** Recibe los eventos del usuario y actúa en función a lo que le pide el usuario a través de la vista. El controlador tiene todo el código funcional y enlaza la vista con el modelo.

Y a continuación se muestra como funciona el flujo de ejecución:

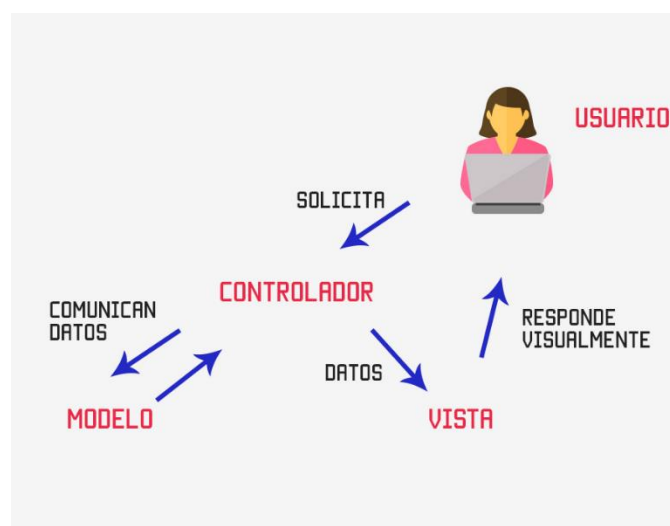


Figura 3.2: Funcionamiento MVC. Imagen extraída de: <https://medium.com/academia-hack/frameworks-mvc-y-ruby-on-rails-1afc3c501c49>

Como se ha podido observar en la imagen:

- El usuario tiene acceso a la vista y al realizar cualquier interacción con ella, se activa el controlador.
- Dicho controlador recoge el evento que realiza el usuario y lo gestiona.
- El controlador puede acceder a otras vistas y mostrarlas al usuario, y también puede acceder a los modelos.
- De esta manera, el modelo le proporcionará al controlador los datos solicitados, y tras obtener los datos que le proporciona el modelo puede mostrarlos en la vista y seleccionar en que vista quiere mostrarlos.

3.3 Bases de datos

En este apartado se verán algunas bases de datos con las que poder contar a la hora de desarrollar una aplicación para dispositivos móviles, en primer lugar, veremos las bases de datos relacionales y seguido las no relacionales.

3.3.1 Bases de datos Relacionales

Las bases de datos relacionales son las que más se usan a día de hoy, se basan en organizar la base de datos mediante tablas que guardan están relacionadas mediante el tipo de datos que contienen. Las más usadas para aplicaciones Android nativas son SQLite y realm.

3.3.1.1 SQLite

Es la base de datos más usada en las aplicaciones Android. Es un gestor de base de datos embebido, lo que quiere decir, que no está implementada fuera del programa, si no que está integrada en él gracias a un conjunto de librerías. Destaca por su simplicidad y rapidez a la hora del desarrollo, algunas características de SQLite son:

- Es gratuito y de código abierto.
- No hace falta el soporte de un servidor, gracias a las librerías comentadas anteriormente, ni necesita configuración.
- Software libre, lo que quiere decir que es de código abierto y los desarrolladores pueden acceder a su dominio.
- Permite almacenar datos de forma persistente, esto quiere decir que los datos se mantendrán en la aplicación en caso de que se apague el dispositivo, por ejemplo.
- Solo carga los datos que necesite en cada momento, lo que se reduce el consumo de memoria. De esta manera no lee toda la base de datos de la aplicación de golpe.

3.3.1.2 Realm

Esta base de datos destaca por su rapidez y está enfocada a grandes bases de datos y a trabajar con grandes cantidades de datos, tanto para dispositivos Android como para dispositivos iOS. Algunas de sus características son:

- Trabaja como ORM, lo que quiere decir que no es necesario usar lenguaje SQL para acceder a la base de datos, lo hace todo mediante objetos.
- Mayor rapidez que SQLite a la hora de realizar consultas a la base de datos.

3.3.2 Bases de datos no SQL

Este tipo de base de datos no siguen el modelo relacional, ni una estructura fija como tablas, tampoco usan el lenguaje SQL como principal a la hora de realizar consultas, aunque puede soportar dicho lenguaje. Las bases de datos no SQL emplean una arquitectura distribuida, esto quiere decir que los datos en la base de datos pueden estar de forma redundante.

Estas bases de datos se crearon principalmente para trabajar con una gran cantidad de datos, dado que con grandes tamaños de bases de datos y mucha cantidad de información las bases de datos relacionales pueden ver su rendimiento afectado. Además, tienen un escalado horizontal, que puede mejorar el rendimiento con la inclusión de más nodos (servidores). La base de datos que veremos de este tipo para el desarrollo en dispositivos móviles será Firebase.

3.3.2.1 Firebase

Firebase es una base de datos en tiempo real donde los datos se almacenarán en formato JSON, Firebase proporciona medidas analíticas de las aplicaciones, también, crear reglas en la base de datos, donde se podrá elegir quien puede, o no, acceder a determinados datos. Cuenta con un sistema de autenticación, que es útil para crear usuarios en la base de datos.

Destacan las siguientes características:

- Admite estructuras de datos flexibles y jerárquicas
- Almacena datos en colecciones, y dentro de ellas en documentos. Dentro de los documentos puede haber más colecciones, sub-colecciones, y los datos entre documentos no tienen por que tener los mismos valores.
- Consultas potentes y específicas, se puede acceder a los datos buscando por colecciones o por determinados documentos, también se puede filtrar por los campos de los documentos.
- Los datos se indexan de forma predeterminada, esto quiere decir que, al realizar consultas, su rendimiento será proporcional a los resultados y no a los datos de la base de datos.
- Tiene asistencia sin conexión, lo que quiere decir que los datos se guardan en caché y cuando se vuelva a tener conexión se subirán los cambios a la base de datos.

3.4 Aplicaciones similares

A continuación, se comentarán algunas aplicaciones similares a este TFG que consiste en la gestión de estadísticas de un equipo de fútbol, hay muchas aplicaciones que aunque tengan un apartado para esto, no se centren especialmente, y se centren más en otros rasgos como la gestión de eventos y el contacto con los jugadores mediante la aplicación o la creación de ejercicios... Solo se mostrarán aquellas cuyo propósito principal sea la gestión de estadísticas.

3.4.1 Mis Estadísticas de Fútbol

Es una aplicación que sirve para la gestión de estadísticas de un equipo de fútbol en tiempo real. Algunas características son:

- Da la opción a crear nuestro equipo junto a nuestros jugadores.
- Permite ver diferentes estadísticas y promedios por jugador en tu equipo.
- Gestión de los jugadores de tu equipo
- Puedes ver un registro de los partidos jugados.
- Tiene filtros para mejorar la interacción con el usuario.

Al haber sido probada, la aplicación, la aplicación desarrollada en el presente TFG podría mejorar la toma de alguna estadística, también que sean más específicas diciendo el jugador, mejora visual a la hora de mostrar las estadísticas o cosas como que el entrenador pueda elegir la configuración para sus partidos.

4. Metodología de trabajo

En este apartado se verán los métodos de trabajo que han sido aplicados en el desarrollo del proyecto, en este caso la metodología ágil, SCRUM. A demás se mostrarán las fases que tendrá este proyecto.

4.1 Metodología ágil SCRUM

No se puede hablar de la utilización de SCRUM como tal, porque esta metodología está pensada para ser usada en grupos de trabajo, y en este caso, dado que la realización del proyecto ha sido llevada a cabo de forma individual, no los hay. No obstante, se intentará adaptar dicha metodología al trabajo individual sin perder la filosofía y la esencia de SCRUM, que es la capacidad de adaptarse a los requisitos del proyecto según se requiera y se vaya avanzando en la realización de este. A demás, esta metodología está basada en sprints, y es uno de sus puntos claves, que pueden ser adaptados fácilmente al trabajo individual.

Como se ha mencionado anteriormente, SCRUM se basa en ir adaptándose a las diversas circunstancias que puedan aparecer en el proyecto mientras este se va desarrollando. Esto quiere decir que no se sigue ningún plan previo de desarrollo, favoreciendo así su adaptabilidad. SCRUM cuenta con tres documentos o componentes propios:

- **Product Backlog:** Aquí se muestran los requisitos del proyecto a realizar, es decir, que funcionalidades debe tener, nuevas características a incorporar, cambios en algunas funcionalidades que ya estaban, corrección de errores o algunos cambios en el proyecto. Estos requisitos no son fijos y se van modificando y actualizando conforme se van haciendo iteraciones en el proyecto. Como se observa, nada es fijo y es en parte gracias a esto que se tiene una gran capacidad de adaptación y flexibilidad en el proyecto. Destacar también, que cada funcionalidad o característica que hay, está priorizada y posee un tiempo de estimación.
- **Product Sprint:** La nueva funcionalidad elegida en el Backlog se distribuye mediante tareas, estas tareas en su conjunto construyen un incremento que será el objetivo al que se pretende llegar tras el sprint, este objetivo es la funcionalidad que se eligió en el Backlog. El tamaño de cada tarea debe oscilar entre 2 y 16 horas, si pasa de ese rango, se debería dividir en dos tareas distintas, el estado de cada tarea será visible para todos los miembros del grupo.
- **Incremento:** Es la funcionalidad desarrollada que se eligió en el Backlog y que se desarrolló en el Sprint.

Algunas características y funcionalidades que proporciona la metodología SCRUM serían:

- **Temprana detección de errores:** Al ser una metodología mediante sprints, el proyecto se va actualizando poco a poco y cada vez que hay una iteración se revisa dicha iteración, de esta manera es más fácil detectar errores y minimiza los riesgos en el futuro del desarrollo.

- **Continua evolución:** Como se ha mencionado en el punto anterior el proyecto se va incrementando poco a poco y de manera continua, lo que le permite estar en una continua adaptación y evolución.
- **Organización:** Los equipos en sus correspondientes sprints, pueden tomar las decisiones que consideren oportunas para conseguir un mejor resultado. Dentro de cada equipo no hay roles y cada uno aporta sus conocimientos y capacidades de forma abierta con el fin de entregar una iteración de alta calidad.

4.2 Evolución por fases

En esta sección se va a mostrar y a explicar cada una de las fases que se han llevado a cabo durante la realización de la aplicación propuesta, así como su finalidad y el resultado que se ha obtenido.

4.2.1 Fase I

En esta primera fase se asegura la viabilidad del proyecto a realizar y el alcance de este.

4.2.2 Fase II

En la siguiente fase se ha trabajado con la realización de diferentes diagramas UML, en este caso han sido el diagrama E/R para tener las ideas más claras en cuanto a la modulación de la base de datos y a las relaciones entre sus tablas, y el diagrama de Clases

4.2.3 Fase III

Esta fase se ha dedicado al aprendizaje de la base de datos Cloud Firestore, de Firebase, dado que ha sido la primera vez que el alumno ha utilizado esta base de datos, se ha decidido dedicar 1 o 2 semanas a ver tutoriales de su utilización y a leer la documentación oficial.

4.2.4 Fase IV

En esta fase se ven las diferentes iteraciones de las que consta la presente aplicación, al haber usado la metodología SCRUM, cualquier iteración podría ser susceptible a modificaciones o actualizaciones. A continuación, se muestran las diferentes iteraciones planeadas:

- **Iteración 0:** Realización del ante proyecto y plan de proyecto.
- **Iteración 1:** Modulación de Usuarios.
- **Iteración 2:** Modulación de Jugadores.
- **Iteración 3:** Poder iniciar partidos y atribuirles estadísticas a los jugadores y a los partidos.

- **Iteración 4:** Administrar partidos.
- **Iteración 5:** Mostrar estadísticas en gráficos.
- **Iteración 6:** Enviar Correo al desarrollador.

4.2.5 Fase V

Es la última fase, donde se redacta la documentación del proyecto y se cierra este TFG

4.3 Entorno tecnológico

En esta sección se mostrarán las herramientas usadas para el desarrollo del presente TFG.

4.3.1 Herramientas para la gestión de proyectos

4.3.1.1 Github

GitHub es un gestor de proyectos en la nube que permite el control de versiones gracias a Git. Permite crear repositorios de forma pública o privada, esta última es de pago, y también permite añadir colaboradores al repositorio, y de esta manera, trabajar de manera conjunta. Al mantener el repositorio público, cualquier persona puede descargar el código pero no modificar el código fuente.

4.3.2 Herramientas para el modelado del software y la elaboración de la documentación

4.3.2.1 Draw.io

Draw.io es una herramienta de creación de diagramas online, y utilizado en la presente aplicación para realizar los diagramas UML. Draw.io proporciona una gran cantidad de variantes, y gracias a ello se pueden crear una gran cantidad de diferentes tipos de diagramas.

4.3.3 Herramientas y tecnologías para el desarrollo del proyecto

4.3.3.1 Android Studio

Como entorno de desarrollo se decidió utilizar Android Studio, explicado en la sección [3.1.1](#)

4.3.3.2 Java

Como lenguaje de programación se decidió utilizar Java, explicado en la sección [3.1.2](#)

4.3.3.3 Firebase

Como base de datos se decidió utilizar Firebase, explicado en la sección [3.3.2.1](#)

5. Resultados

En este apartado se redactarán las iteraciones explicadas en el punto anterior, y que han sido realizadas para llevar a cabo el presente proyecto.

5.1 Iteración 0

En esta primera iteración se ve si la idea de la aplicación puede tener utilidad y ser viable. Tras poner las ideas sobre la mesa el alumno decide intentar llevar a cabo la presente aplicación comenzando con el anteproyecto, y para ello sería necesario pensar en que tecnologías usar, cual puede ser mejor y más rentable para la aplicación. Tras la realización y posterior confirmación del anteproyecto, el alumno decide empezar con la realización de este proyecto.

5.1.1 Plan de proyecto

5.1.1.1 Ámbitos del sistema

La aplicación para desarrollar realizará las siguientes tareas:

- La principal tarea que debe llevar a cabo la aplicación es guardar estadísticas individuales de cada jugador en cada partido y del partido en cuestión.
- El entrenador tiene asociada una cuenta con Google y cada entrenador tiene asociado un equipo.
- El entrenador tiene total control sobre la gestión de sus jugadores, siempre y cuando no sean las estadísticas, que se recogen en tiempo real en cada partido.
- Tras tener jugadores creados el entrenador puede comenzar partidos donde debe elegir si su equipo juega de local o visitante e indicar el nombre del equipo rival. También tiene que completar la convocatoria del partido.
- Una vez en el partido, el entrenador deberá comenzar el partido para poder guardar estadísticas, tras comenzar, el entrenador puede seleccionar si quiere guardar una estadística a favor o en contra, en caso de ser a favor, debe indicar el jugador que realiza la acción. El entrenador tiene que tener acceso a realizar cambios durante el partido.
- Al finalizar el partido el entrenador tiene la opción de guardar observaciones sobre el partido.
- El entrenador tiene acceso a un buscador para buscar un partido en concreto. Y acceso a diferentes gráficos estadísticos.

- El entrenador puede ver las estadísticas generales de un jugador y también las de el partido que seleccione, mostrándose ahí las estadísticas de los jugadores que jugaron ese partido
- El entrenador tiene acceso a enviar un mensaje, mediante el correo electrónico, al desarrollador en caso de tener alguna duda o sugerencia en cuestión.
- El entrenador puede modificar el nombre del equipo y los minutos de juego.
- El entrenador tiene la opción de cerrar sesión.
- El sistema tiene como objetivos:
- Ayudar a los entrenadores a obtener las estadísticas que quieran en todo momento.
- Ayudar a los entrenadores durante el partido, automatizando algunas tareas.
- Gestionar los jugadores, pudiendo crearlos, eliminarlos y modificarlos.
- Gestionar los partidos, pudiendo crearlos, eliminarlos y buscarlos.

5.1.1.2 Visión general

La presente aplicación proporciona a los entrenadores poder tener las estadísticas de sus jugadores y de sus partidos a mano en cualquier momento, pudiendo así llevar a cabo una mejor planificación de su equipo.

5.1.1.3 Perspectiva del producto

Para el uso de la aplicación solo será necesario tener una cuenta de Google, que si se ha podido descargar la aplicación, este requisito ya estaría cumplido. Por lo demás, simplemente la interacción de un usuario y el sistema.

Como la aplicación solo está pensada para entrenadores la vista del usuario siempre será la misma:

- Interface de entrenador/a.

5.1.1.4 Restricciones

La aplicación debe cumplir con la ley de protección de datos, es decir, nunca se podrán revelar los datos de los entrenadores.

Para el desarrollo de la aplicación se va a usar, como lenguaje de programación, Java en el entorno de desarrollo Android Studio. La base va a ser Cloud Firestore de Firebase.

Se deberá obtener los datos de acceso de cada entrenador, en este caso la cuenta de Google, para poder lograr un sistema seguro. A demás se facilitará el uso de la aplicación con

vistas sencillas e intuitivas, para que el manejo de esta se simplifique y sea más cómodo para los usuarios.

5.1.2 Viabilidad

En este apartado se tratará la cuestión de si el proyecto es viable o no y con la intención de ponerlo en marcha, en caso de que sea exitoso.

5.1.2.1 Viabilidad técnica

Este análisis se centra en la viabilidad del proyecto, a nivel de software, y a nivel de conocimientos.

En cuanto a las tecnologías que pueden ser aplicadas para la realización de este proyecto se encuentran:

- Ordenador portatil HP elitebook 8440
- Android Studio
- Git
- Java
- Libre office
- Google Chrome

Como se puede observar, todo es open source, y el ordenador era propiedad del alumno en cuestión así que no supone ningún gasto. Por lo tanto, no hay ningún inconveniente a nivel técnico que impida al alumno la correcta realización de este proyecto.

Y en cuanto a los conocimientos del alumno, el alumno ha sido capaz de adquirir conocimientos por su propia cuenta como así se muestra en la fase II. Por lo que en este caso tampoco hay ningún inconveniente en cuanto a conocimientos técnicos.

Dado que los dos puntos han sido positivos el alumno está capacitado para la correcta realización del presente proyecto.

5.1.2.2 Viabilidad legal

Este análisis asegura que no se incumple ninguna norma o ley establecida. Dado que se quiere garantizar lo establecido en el proyecto, como se ha comentado en las restricciones se llevará a cabo la *Ley Orgánica 15/1999 de Protección de Datos de Carácter Personal*.

Puesto que los datos que se guardan sobre los jugadores los guarda el entrenador, no hay ninguna fiabilidad de que esos datos sean reales o no, por lo que no haría falta aplicar las normativas respecto a la recogida de datos a un mayor de edad, visto esto, se cumplen todos los requisitos del artículo 5 de la LOPD, y no se aplicará en ningún momento el derecho (ARCO).

5.2 Iteración I

En cuanto a esta iteración se desarrolla el módulo de Usuario. Este es el único tipo de usuario que permite la aplicación y está pensado para que sea el entrenador de un equipo de fútbol.

Lo primero que se le pedirá al usuario es iniciar sesión con Google, si no selecciona ninguna cuenta podrá navegar por algunas partes de la aplicación, pero no tendrá funcionalidad ninguna.

Tras iniciar sesión se mostrará la pantalla de inicio, que sirve para crear partidos, pero no dejará crearlos si no hay jugadores.

El usuario podrá cerrar sesión y volver a iniciarla cuando desee.

5.2.1 Administrar equipo

Crear equipo

La primera acción que se le pedirá al usuario después de la de iniciar sesión, es la de configurar los datos del equipo, siempre y cuando el usuario no tienda un club creado ya, donde se le pedirá al usuario establecer un nombre del club y el tiempo de cada parte de los partidos.

Modificar equipo

Una vez creado el equipo, se pueden modificar los datos establecidos, es decir, el nombre del club y los minutos de cada parte.

5.3 Iteración II

En esta iteración se modelarán a los jugadores, para poder usar la aplicación es necesario tener jugadores en el club y además el usuario los podrá administrar como desee.

El usuario deberá ir a la pestaña jugadores, y desde hay tendrá varias opciones.

Crear jugadores

Sin jugadores en el club no se podrá empezar un partido, así que en este apartado el usuario va a poder crear jugadores introduciendo su nombre, la posición que ocupa en el campo, el número de su camiseta y la edad.

El número de la camiseta no es un campo requerido, y no dejará crear a dos jugadores con el mismo nombre en un mismo equipo, igual pasa con el número de la camiseta.

Una vez creado un jugador correctamente, se mostrará en una lista donde se encuentran todos los jugadores ordenados por posición.

Modificar jugadores

Al mantener pulsado un jugador, este jugador puede ser modificado y se cargará una nueva vista con sus datos actuales.

Eliminar jugadores

La otra opción al mantener pulsado un jugador es la de eliminarlo, al hacer esto se eliminará dicho jugador.

Consultar información

Al seleccionar un jugador, se mostrarán los datos generales de este.

5.4 Iteración III

Cuando el usuario tenga más de un jugador y escriba el nombre del equipo rival y si su equipo juega como local o visitante, el usuario podrá crear un partido con 11 jugadores titulares y un máximo de 22 jugadores sumando los titulares y los suplentes.

Tras completar todo correctamente, el usuario puede comenzar el partido, y el tiempo empezará a correr, cuando el contador llegue a cero aparecerá otro que cuenta el tiempo añadido, y aparecerá un botón para el final del primer tiempo, desde ese momento el usuario será capaz de añadir estadísticas al partido.

En la pantalla hay un botón para salir del partido, pero si se pulsa y se confirma, no se guardarán los datos de ese partido.

Al pulsar el botón de estadística a favor, se mostrará una lista con todas las opciones de estadísticas que tiene la aplicación, al seleccionar una, se mostrará una lista con los jugadores titulares y al seleccionar uno, se le habrá sumado esa estadística.

Al pulsar el botón de estadística en contra, solo se mostrará una lista con las opciones que hay para las estadísticas de los rivales, al pulsar en una, esta se actualizará.

El usuario puede cambiar a la ventana “estadísticas” donde se mostrarán las estadísticas anotadas durante el partido.

Al pulsar el botón de final del primer tiempo, pedirá una segunda confirmación por si acaso ha sido un error, si se confirma, aparecerá un botón para empezar la segunda parte y el tiempo se reiniciará. Durante el descanso no se puede hacer ninguna acción.

Al iniciar la segunda parte, la funcionalidad será igual que en la primera, y aparecerá un botón de finalizar partido que tendrá la misma función que el de finalizar primera parte.

Al pulsar el botón finalizar y confirmar, aparecerá un “box” para escribir observaciones del partido y un botón para enviar y confirmar el partido.

Durante el partido, el entrenador podrá seleccionar el icono de cambio, donde primero deberá elegir el jugador que será sustituido, y luego el que le sustituirá.

5.5 Iteracion IV

El usuario puede ser capaz de administrar sus partidos en la pestaña “registros”, puede buscar por nombre un partido específico. Y filtrar por algunas características del partido.

Al pulsar un partido, se mostrarán los jugadores de este y sus estadísticas, así como las estadísticas del partido y las observaciones.

Eliminar partidos

Al mantener pulsado en un partido, saldrá una opción para eliminar ese partido, se eliminarán también los datos de los jugadores que jugaron ese partido.

5.6 Iteracion V

En la pestaña “info” al usuario se le mostrarán unos cuantos gráficos a cerca de los partidos disputados, y de los disputados de local y de visitante.

Podrá observar también un balance sobre sus últimos cinco partidos, e información adicional sobre su equipo.

5.7 Iteracion VI

En esta ultima iteración el usuario va a ser capaz de enviar un correo al desarrollador, por si tiene alguna duda personal, o alguna sugerencia sobre la aplicación.

Simplemente deberá de escribir un asunto y un mensaje desde la aplicación, y al pulsar el botón de enviar, aparecerá un box de aplicaciones donde estará disponible la opción de Gmail, al pulsar en ella el mensaje se copiará al correo y solo tendrá que enviar dicho correo.

6. Conclusiones y propuestas

En este punto se muestran las conclusiones de la realización del presente proyecto y se expondrán el cumplimiento o no de los objetivos marcados en el punto 2 del trabajo. También el alumno propondrá propuestas a futuro en el desarrollo de la aplicación.

6.1 Conclusiones

En este apartado se darán a conocer si se han conseguido los objetivos parciales del proyecto, y se dará una justificación de por qué se han podido, o no, superar.

1. El objetivo 1. Ha sido completado, dado que el proyecto ha sido realizado utilizando ambas herramientas, y se ha fomentado el aprendizaje de estas.
2. El objetivo 2. Ha sido completado, dado que el alumno centró unos días en el aprendizaje, de este nuevo servicio para él, de base de datos y su posterior implementación en el proyecto ayudó a mejorar los conocimientos sobre Firebase y en cuanto a Git, el alumno ha creado un repositorio con su proyecto y lo ha ido actualizando en función de las nuevas implementaciones. Así que si se puede decir que este objetivo está cumplido.
3. El objetivo 3. Completado con éxito gracias a los diagramas realizados y entregados junto al proyecto.
4. El objetivo 4. Se ve cumplido al finalizar este TFG, viendo que se han usado las tecnologías mencionadas.

Tras verse cumplidos todos los objetivos secundarios, el objetivo principal que es desarrollar una aplicación que ayude a los entrenadores de fútbol a tener unas estadísticas más detalladas de su equipo.

6.2 Propuestas futuras

Dado que la intención del alumno es seguir desarrollando la aplicación y en un futuro lanzarla a la Play Store, se tiene intención de añadir los siguientes puntos:

- Correcta implementación en tablets
- Gráficos más detallados y de diferentes momentos de la temporada
- Sistema para comparar jugadores
- Automatización de la toma de estadísticas, por ejemplo, que, al anotar un tiro a puerta, se anote un tiro también-
- Límite de cambios y que el entrenador pueda elegir cuantos

Bibliografía

Firestore : <https://firebase.google.com/docs?hl=es-419>
Firestore : https://www.youtube.com/channel/UC_Fh8kvtkVPkeihBs42jGcA
Git : <https://gist.github.com/dasdo/9ff71c5c0efa037441b6>
Git : <https://www.hostinger.es/tutoriales/comandos-de-git>
Android : <https://developer.android.com/docs>
SCRUM : <https://clouding.io/blog/scrum-en-la-metodologia-agil-te-explicamos-que-es-y-como-funciona/>
SCRUM : [https://en.wikipedia.org/wiki/Scrum_\(software_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development))
Realm: <https://blog.desdelinux.net/realm-base-de-datos-nativa-para-dispositivos-moviles/>
SQLite : <https://openwebinars.net/blog/sqlite-para-android-la-herramienta-definitiva/>
SQLite : <https://sg.com.mx/revista/17/sqlite-la-base-datos-embebida>
Flutter: https://flutter.dev/?gclid=EAIaIQobChMIq6DI2Jq06QIV1JTVCh0dAQSOEA_AYA_SAAEgLSEPD_BwE&gclid=aw.ds