
This is the **published version** of the bachelor thesis:

Orozco Lorente, Adrià; Talló Sendra, Marc, dir. Aplicación para la gestión de equipos de fútbol. 2022. (958 Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/264118>

under the terms of the  license

Aplicación para la gestión de equipos de fútbol

Adrià Orozco Lorente

Resumen - Este proyecto pretende ser una base de innovación respecto a la gestión de clubes de fútbol base, la cual no ha progresado tecnológicamente y no dispone de la automatización de un amplio abanico de procesos que facilitan la labor de muchos de sus empleados. De esta forma se opta a conseguir una herramienta capaz de automatizar todos los procesos rutinarios en el ámbito de la gestión de un club de forma centralizada, rápida y sencilla para todos los empleados y válida para todos los rangos de edad, añadiendo de igual forma funcionalidades que puedan ser de utilidad para todos los entrenadores, tales como disponer de información estadística de su equipo, o un muro común de entrenamientos para facilitar su labor de desarrollo de los mismos durante la semana.

Palabras Clave— Entrenamiento, convocatoria, asistencia, geolocalización, estadísticas, aplicación, entrenador, coordinador.

Resum - Aquest projecte pretén ser una base d'innovació respecte a la gestió de clubs de futbol base, la qual no ha progressat tecnològicament i no disposa de l'automatització d'un ampli ventall de processos que faciliten la labor de molts dels seus empleats. D'aquesta manera s'opta a aconseguir una eina capaç d'automatitzar tots els processos rutinaris en l'àmbit de la gestió d'un club de forma centralitzada, ràpida i senzilla per a tots els empleats i vàlida per a tots els rangs d'edat, afegint d'igual forma funcionalitats que puguin ser d'utilitat per a tots els entrenadors, com ara disposar d'informació estadística del seu equip, o una mur comú d'entrenaments per a facilitar la seva labor de desenvolupament dels mateixos durant la setmana.

Paraules Clau— Entrenament, convocatòria, assistència, geolocalització, estadístiques, aplicació, entrenador, coordinador.

Abstract—This project aims to be a base for innovation in the management of soccer clubs, which has not progressed technologically and does not have the automation of a wide range of processes that facilitate the work of many of its employees. In this way, the aim is to achieve a tool capable of automating all routine processes in the field of management of a club in a centralized, fast and simple way for all employees and valid for all age ranges, adding features that can be useful for all coaches, such as having statistical information of your team, or a common wall of training to facilitate their work of development of them during the week.

Index Terms—Training, citation, attendance, geolocation, statistics, application, coach, coordinator.

1 INTRODUCCIÓN

Como entrenador de equipos de fútbol y coordinador de una de las categorías de fútbol base del mismo club en el que trabajo, me encuentro en la necesidad de tener una aplicación en la que poder llevar a cabo las tareas de gestión tanto de un equipo en particular, como de un conjunto de equipos de forma centralizada, sin la necesidad de recurrir tanto a métodos tradicionales, como a una amplia variedad de aplicaciones distintas para llevar a cabo diferentes tareas. Teniendo en cuenta los progresos realizados durante el desarrollo y bases establecidas en el informe inicial y en el primer informe de seguimiento, se establece el flujo de este documento de tal manera que muestre el resultado final del proyecto y sus desviaciones

En este instante del proyecto se ha realizado la codificación de la aplicación creando cada funcionalidad según su prioridad establecida en los requisitos además de pasar

ciertos test para asegurar la calidad de las funcionalidades implementadas. Por otra parte se ha realizado la fase de test para evitar errores que imposibiliten el uso correcto de la aplicación. Para ello se asignó Trello como principal herramienta para el registro de tareas y Github para el control de versiones del código a implementar. Todo esto se ha realizado teniendo en cuenta toda la planificación y diseños realizados en las fases anteriores.

Los contenidos desarrollados corresponden a aquellos requisitos que la aplicación debía contener, y se muestran a continuación junto la explicación al problema anterior:

1. **Seguimiento de asistencia de los entrenadores y jugadores**, donde para los primeros la única forma existente actualmente para llevarla a cabo es

mediante un método tradicional. Por otra parte la asistencia de jugadores es controlada de la anterior manera en algunos clubes, mientras que otros optan por la inclusión de nuevas tecnologías como pueden ser ficheros compartidos en la nube mediante documentos de Google Drive¹, por ejemplo o hojas de cálculo como las que ofrece Microsoft Excel², que son enviadas por correo electrónico o impresas i mostradas en el club a final de mes.

2. **Creación de entrenamientos** para sus respectivos equipos, si bien no es una tarea que deba automatizarse, siempre puede ser útil guardar un registro de estos, tanto para poder re-aprovecharlos durante el transcurso de la temporada, para que quede constancia de los aspectos trabajados en un registro global, o poder compartirlos con otros entrenadores y así poder dedicar un tiempo menor a la elaboración de estos.
3. **Lista de actividades en relación a la competición** en la que participan los fines de semana. Entre estas se encuentran:

3.1 Realización de convocatorias (Elemento donde queda constancia de los jugadores que pueden ir al partido del fin de semana),

Para la realización de esta tarea se da el mismo caso que para las asistencias de los jugadores, y se realizan tanto de forma manual, como con documentos compartidos en la nube, incluso de forma verbal sin llevar un control específico de la situación.

3.2 Fichas de partido, las cuales constan en un resumen de lo sucedido en un partido, incluyendo los minutos jugados por cada jugador, la posición ocupada en el campo, goles, y demás información que podría ser de utilidad para el uso estadístico y para tener un registro completo de la temporada.

Es importante remarcar en este último punto, que al menos en Cataluña para otorgar las subvenciones³ a jugadores con dificultades económicas, se exige a los clubes un registro de estos dos aspectos comentados de un equipo aleatorio. Por lo cual resulta imprescindible tener un registro automatizado sobre estos aspectos, que hasta la fecha debe realizar algún encargado de forma manual, y en ocasiones con información de dudosa veracidad, al no tener registros sobre los mismos.

4. **Registro de la información personal de los jugadores**, o sus tutores legales en caso de que sean menores de edad. Para ello al igual que en casos anteriores se pueden utilizar herramientas que se han comentado con anterioridad.
5. **Registro de estadísticas de jugadores**, las cuales son extraídas de las mencionadas fichas de partido y no existe hasta ahora un método automático para

llevarlo a cabo, y debe encargarse una persona en concreto para llevar a cabo esta recopilación de datos cada fin de semana

2 ESTADO DEL ARTE

Teniendo en cuenta las características especificadas en la sección anterior se realiza un estudio de mercado con objetivo de analizar aplicaciones similares y determinar el potencial de la herramienta a desarrollar.

Como principal aplicación móvil, se encuentra 'Football Coach App', la cual permite registrar a los entrenadores la información sobre su equipo para tener un registro de jugadores y estadísticas, si es cierto que la aplicación a desarrollar contará con un sistema similar al que desempeña esta aplicación, el objetivo global de la herramienta a crear es la gestión del día a día del club y no solo la información estadística de los jugadores.

Por otra parte existe 'MatchApp' que es una de las aplicaciones más potentes en el ámbito de fútbol base, pero esta, más bien esta orientada a realizar un seguimiento de los encuentros que tiene un equipo a lo largo de la temporada, incluyendo calendario y resultados.

Derivado al ámbito de la gestión de clubes se encuentra 'FutApp gestión de equipos', la cual es una herramienta con la que se puede tener el control de un club, pero solo permite introducir la información de los equipos y sus próximos entrenamientos.

En otros mercados, se ha encontrado una versión alemana de una aplicación llamada 'B42 Die Fussball Team App' la cual nos ofrece herramientas para la gestión de un equipo en concreto de manera que cabe la posibilidad de establecer los días de entrenamiento del equipo y poner ejercicios que se realizarán.

Habiendo observado las distintas herramientas encontradas en el mercado, se establece un amplio potencial en la herramienta a desarrollar debido a la inclusión de funcionalidades no vistas en el mercado, además de una gestión de clubes centralizada y sencilla la cual no hay ninguna herramienta que lo permita.

3 OBJETIVOS

Teniendo en cuenta todo lo especificado en las secciones anteriores y los documento anteriores se ha planteado como propuesta el desarrollo de una aplicación móvil, en la que se pueden realizar todas las funciones ya comentadas de forma sencilla, intuitiva y en una disposición centralizada para así facilitar las tareas rutinarias de la gestión de ya no solo un equipo, si no de un club modesto de fútbol.

La aplicación debe contar con un **sistema de usuarios** con diferentes roles en base a la posición que ocupen en el club para el que trabajen con tal de así limitar las funcionalidades a las que tendrá acceso así como a la información que podrá recuperar de la plataforma., estableciendo estos como Coordinador y Entrenador, donde un usuario puede ocupar ambos roles.

En esta etapa final del proyecto se han mantenido los objetivos marcados previamente, incluso se han añadido mejoras a los diseños para facilitar la gestión de un club.

Por lo que las funcionalidades incluidas en la aplicación consisten en que cualquier usuario será capaz de marcar su asistencia desde la aplicación mediante un **sistema de geolocalización**, la posibilidad de **marcar la asistencia de los jugadores, actualizarla, realizar convocatorias, y fichas de partido** desde la misma y el **desarrollo de un muro** en la aplicación donde los usuarios puedan subir sus entrenos para compartírselos, aunque se declara como fase secundaria del desarrollo.

Se continúa mencionando las funciones que pueden tener los **roles**, de forma que los **coordinadores** pueden introducir todos los datos relevantes de sus **equipos y jugadores**.

También se propone unificar de igual manera en el mismo aplicativo las funciones respectivas a **convocatoria y fichas de resumen del partido**.

4 METODOLOGIA

Para conseguir los objetivos planteados, se considera vital una planificación detallada con objetivos semanales para el correcto desarrollo del trabajo planteado.

Para ello se utiliza una metodología de desarrollo incremental⁴ a través de iteraciones semanales donde se repite un proceso de trabajo en concreto donde se especifican las tareas a realizar semanalmente teniendo en cuenta el tiempo disponible.

De esta manera se consigue gestionar y sincronizar las tareas del proyecto en base al objetivo final.

Para ello se planteó inicialmente la división del proyecto en fases, tales como:

-Planificación, donde se estudian las fortalezas del proyecto, los riesgos asociados, la planificación temporal y la asignación de recursos a cada etapa del mismo.

-Análisis, donde se especifica lo que debe hacer el software que se implementará, las necesidades reales, y la comprensión adecuada de los requerimientos del sistema.

-Diseño donde se decide la estructura general del mismo y las posibles soluciones de implementación.

-Implementación, donde se produce la codificación del software de manera iterativa con una correcta estructuración del código definida por la fase de diseño.

-Pruebas, donde se busca detectar los errores cometidos en fases anteriores para corregirlos, antes de mostrar el resultado a los usuarios.

-Instalación

En esta etapa del proyecto se han realizado todas las fases, y es momento de instalar la aplicación en un entorno real.

5 HERRAMIENTAS

Para la realización de las tareas especificadas se ha realizado desarrollo de una aplicación móvil realizada en el lenguaje Dart⁵ mediante la SDK de Flutter⁶.

Por otra parte en relación al Back-end se ha hecho uso de **Firestore**⁷ como plataforma para el desarrollo de la aplicación, la cual cuenta con la posibilidad de mantener un sistema personalizado de actualización, así como una base de datos **Nosql**⁸ funcional a tiempo real donde poder almacenar los datos necesarios, así como un sistema de almacenamiento que será útil tanto para posibles plantillas, como para la subida de archivos de cada usuario, como videos, imágenes para los entrenamientos, entre otros.

Por otra parte y referente a la gestión del proyecto cabe destacar el uso de **Trello**¹⁰ para la planificación de los sprints y el detallado de las tareas y **GitHub**¹¹ para la estructura y control de versiones del repositorio donde se realice la codificación.

6 PLANIFICACIÓN

Las tareas del proyecto han sido completadas con éxito y se ha conseguido establecer un flujo correcto de tareas y evitar problemas posteriores, estas tareas han consistido en:

-Captura de requisitos funcionales (funciones del sistema de software que se desarrollará)

-Captura de requisitos no funcionales y restricciones (especificaciones de criterios que pueden usarse para juzgar el sistema en vez de comportamientos específicos, así como limitaciones a establecer)

-Realización de prototipos (diseños iniciales de la aplicación)

-Descripción de casos de uso, incluyendo todas las condiciones necesarias, excepciones y dependencias

-Diseño de la base de datos.

-Montaje de la base de datos e integración con el front-end de la aplicación.

-Documentación asociada a la codificación

-Checklist de test

-Pruebas de funcionalidad (Conjunto de pruebas para verificar el correcto comportamiento de un componente)

-Documentación asociada a cambios

-Scrapping de equipos de cataluña (tarea no registrada en la planificación inicial pero su bajo tiempo de codificación hace que no se produzcan desviaciones)

-Funcionalidades de la aplicación

-Test y correcciones

-Pruebas en entorno real

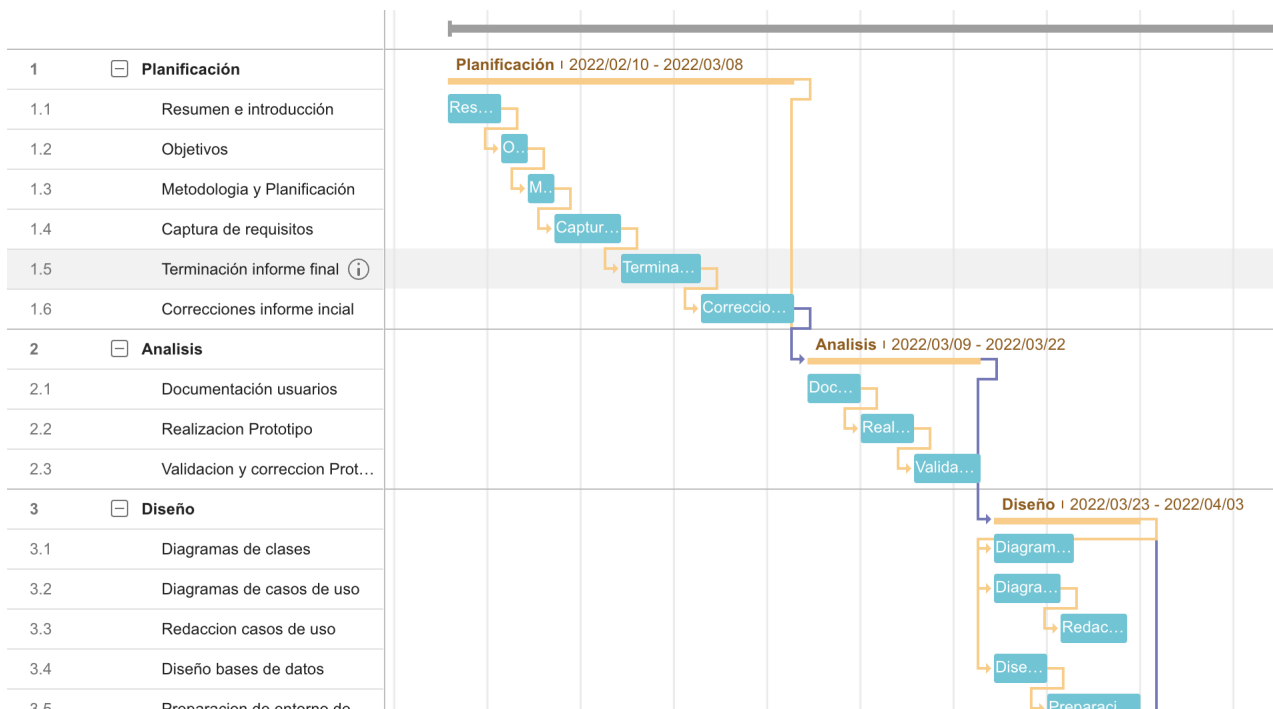
El proyecto ha seguido correctamente la planificación inicial y ha respetado los plazos de cada fase de manera satisfactoria por lo que se han llevado cabo las fases

propuestas además de las actividades marcadas de la siguiente manera:

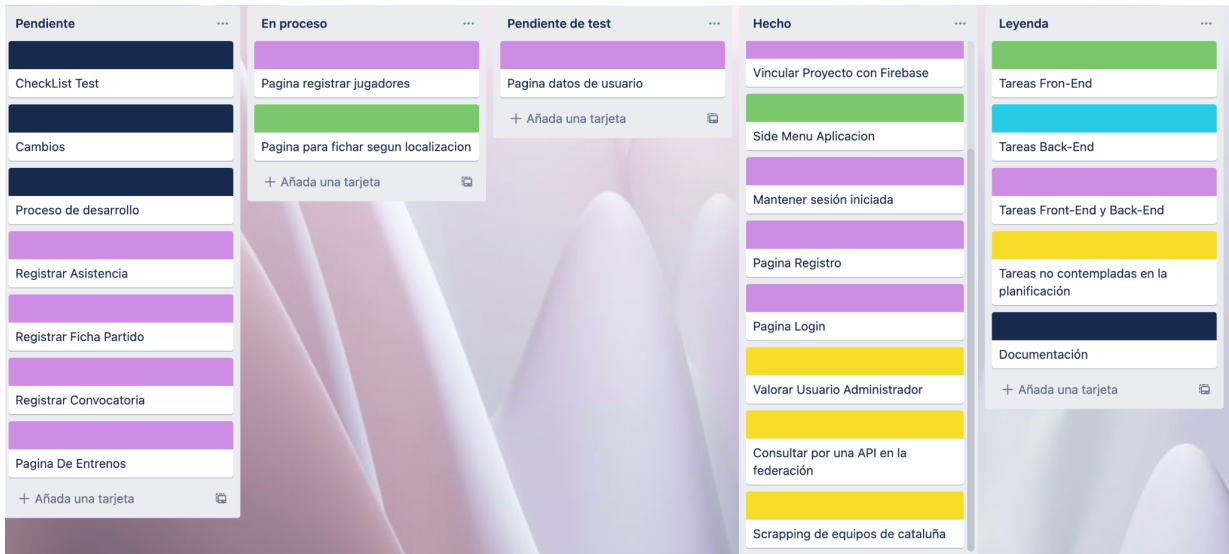
-Software en funcionamiento

Se muestra una parte de la planificación inicial completada a continuación:

1. **08/02/2022 - 06/03/2022 Fase de planificación** ✓
-Establecimiento de bases del proyecto, introducción, objetivos, metodología, planificación inicial.
2. **07/03/2022 - 20/03/2022 Fase de análisis** ✓
-Captura de requisitos funcionales, no funcionales y restricciones.
-Realización, validación y corrección de prototipos .
3. **21/04/2022 - 10/04/2022 Fase de diseño** ✓
-Diagramas UML (casos de uso)
-Documentación de los casos de uso.
-Diseño de la base de datos
4. **11/04/2022 - 12/06/2022 Fase de codificación** ✓
-Fase en la que se desarrollará de manera simultánea la codificación junto a la elaboración de la documentación asociada a esta, incluidas las checklist de los test, primeras pruebas funcionales y la documentación asociada a cambios.
5. **13/06/2022 - 26/06/2022 Fase de pruebas** ✓
-Pruebas finales
-Pruebas de usuarios
-Corrección de errores i documentación final
6. **27/06/2022 Fase de instalación**



Se adjunta el diagrama completo como pdf en la carpeta origen con nombre Diagrama de Gantt.pdf
Además se adjunta una captura de pantalla de un ejemplo de planificación de tareas durante la fase de codificación::



7 REQUISITOS FUNCIONALES

A continuación se especifican los requisitos funcionales que se han implementado en el proyecto junto con sus prioridades, las cuales quedan especificadas como A (Alta), M (Media) y B (Baja).

Id	Descripción	Prioridad
RF-01	La aplicación ha de permitir al usuario registrarse	A
RF-02	La aplicación ha de permitir al usuario iniciar sesión	A
RF-03	La aplicación ha de permitir al usuario recuperar su contraseña	M
RF-04	La aplicación ha de permitir al usuario ver el muro de entrenamientos	M
RF-05	La aplicación ha de permitir al usuario ver los detalles de un entrenamiento	M
RF-06	La aplicación ha de permitir al usuario valorar un entrenamiento	B
RF-07	La aplicación ha de permitir al usuario subir un entrenamiento	A
RF-08	La aplicación ha de permitir al usuario aportar material gráfico a un entrenamiento	B
RF-09	La aplicación ha de permitir al usuario la opción de ver sus entrenamientos	M
RF-10	La aplicación ha de permitir al usuario visualizar un video de entrenamiento	B
RF-11	La aplicación ha de permitir al usuario eliminar un entrenamiento	B
RF-12	La aplicación ha de permitir al usuario rellenar la lista de asistencia de sus jugadores	A
RF-13	La aplicación ha de permitir al usuario editar la lista de asistencia de sus jugadores	A
RF-14	La aplicación debe permitir al usuario marcar su asistencia	A

RF-15	La aplicación ha de permitir al usuario realizar la ficha de un partido	A
RF-16	La aplicación ha de permitir al usuario visualizar la ficha de un partido	A
RF-17	La aplicación ha de permitir al usuario rellenar la convocatoria de un partido	A
RF-18	La aplicación ha de permitir al usuario visualizar la convocatoria de un partido	A
RF-19	La aplicación ha de permitir al usuario introducir la información de sus jugadores	A
RF-20	La aplicación ha de permitir al usuario editar la información de los jugadores	A
RF-21	La aplicación ha de permitir al usuario observar las estadísticas de los jugadores	M
RF-22	La aplicación debe ser capaz de actualizar las estadísticas de los jugadores	A
RF-23	La aplicación ha de permitir al usuario editar su perfil	B
RF-24	La aplicación debe ser capaz de renovar las convocatorias tras semana	B
RF-25	La aplicación debe ser capaz de renovar las fichas de partido tras cada semana	B
RF-26	La aplicación debe ser capaz de renovar las listas de asistencia cada mes	B
RF-27	La aplicación ha de permitir al usuario consultar un registro de convocatorias	M
RF-28	La aplicación ha de permitir al usuario consultar un registro de fichas de partido	M
RF-29	La aplicación ha de permitir al usuario consultar un registro de asistencia de los jugadores	M
RF-30	La aplicación ha de permitir al usuario pausar y reproducir un video de un entrenamiento	B
RF-31	La aplicación ha de permitir ver en vista de calendario todos los entrenos de su equipo	B

7 REQUISITOS NO FUNCIONALES

A continuación se especifican los requisitos no funcionales que se han implementado en el proyecto junto con sus prioridades, las cuales quedan especificadas como A (Alta), M (Media) y B (Baja).

Id	Descripción	Prioridad
RNF-01	El sistema ha de proteger los datos personales de manera cifrada	A
RNF-02	El sistema ha de ocultar la información personal de un usuario a otros usuarios	A
RNF-03	El sistema ha de contar con una interfaz gráfica agradable al usuario y que permita realizar las tareas en los mínimos pasos posibles	A
RNF-04	El sistema se debe adaptar a los sistemas operativos Android y iOS	A
RNF-05	El sistema ha de permitir ver el contenido de forma correcta en cualquier tamaño de pantalla	A

RNF-06	El sistema ha de tener un tiempo de espera inferior a 2 segundos	M
RNF-07	La autenticación del usuario deberá quedar almacenada a parte de la base de datos que alimenta el sistema	A
RNF-08	Los datos almacenados en la base de datos deben tener un identificador único para poder diferenciarlos	A
RNF-09	La aplicación necesitará conexión a internet para funcionar	A
RNF-10	La aplicación deberá contar con permisos de ubicación, y almacenamiento para funcionar	A
RNF-11	La aplicación necesitará un mínimo de 20MB disponibles en el sistema para ser instalada	A
RNF-12	La aplicación debe mostrar mensajes de excepción en cada funcionalidad	B

8 VALORACIÓN DE COSTES Y RIESGOS

Durante el proyecto se han ido ajustando los costes del mismo de tal manera que se han modificado según las necesidades, finalmente el resultado es el siguiente:

Perfil	Precio/hora (€)	Total horas
Analista	30	30
Diseñador	10	50
Front-End Developer	20	140
Back-End Developer	20	20
Tester	27	70

Las horas necesarias se extraen de una aproximación realizada en el diagrama de gantt de la planificación y son adaptadas por la carga necesaria de trabajo realizadas durante esta fase, donde han hecho falta más recursos a dedicar en el front-end de la aplicación, ya que al utilizar un Back-end as a service no hace falta implementar clases ni llamadas api para poder realizar según que funciones.

Coste Total: $(30*30) + (10*50) + (20*140) + (20*20) + (27*70) = 6490 \text{ €}$

Se añade el software de Creative-Tim explicado en la sección 9.3 → 100€

Nuevo Total: 6590€

Con el anterior desglose de precios y teniendo en cuenta tanto la planificación establecida, como las herramientas que se están utilizando para el desarrollo del proyecto, se detectan riesgos, tales como:

1. **Retraso de las tareas.** El hecho de que las tareas no se realicen en el tiempo establecido comporta un aumento de costes en el proyecto, aunque el buen desarrollo del proyecto minimiza este riesgo..
2. **Errores en tareas principales.** Una fase del proyecto ejecutada incorrectamente, puede derivar en que este deba realizar de nuevo ciertas tareas, cosa que comporta además de un aumento en el coste total, el riesgo de que el proyecto no pueda desarrollarse en el tiempo establecido. Por lo que resulta de vital importancia la validación de cada apartado de forma minuciosa antes de pasar al siguiente.
En este instante no se puede descartar todavía este riesgo y es algo que se corrobora durante la codificación, pero de momento podemos concluir que este riesgo se ha minimizado
3. **Imprevistos.** En este apartado se contemplan los riesgos que se pueden derivar de imprevistos no controlados. Las herramientas usadas para el desarrollo del proyecto son de uso gratuito y pertenecientes a proveedores externos. El hecho de que uno de estos servicios sufra un ataque, o tenga alguna caída de servicio, puede afectar no solo a los tiempos de entrega si no que si es algo más grave puede afectar gravemente en los costes, necesitando más horas para migrar el sistema a nuevos entornos y añadiendo horas para realizar estas tareas no previstas. Donde aquí se añade el fallo de alguna de las funcionalidades de la API de google maps añadida, la cual será explicada posteriormente, sin la cual no seríamos capaces de obtener las direcciones reales de los campos de fútbol donde se realizan los entrenamientos.

9 ANÁLISIS DE DISEÑO

En este apartado se pretende realizar un análisis completo de

las características de diseño que debe contener el proyecto antes de ser realizado, por lo que se detallan de forma exhaustiva las funcionalidades, desglose de actividades y casos de uso, teniendo en cuenta análisis previos.

9.1 DESGLOSE DE USUARIOS

La aplicación a desarrollar consta de un sistema de autenticación de usuarios los cuales se dividirán en dos roles diferentes: Coordinador y Entrenador, pudiendo un usuario ocupar los dos roles a la vez.

Se exponen a continuación de manera detallada las acciones que pueda llevar a cabo cada rol de usuario, con ciertas modificaciones llevadas a cabo desde el registro de la aplicación para que las acciones se realicen de manera más óptima con el menor número de pasos posibles.

USUARIO COORDINADOR:

Como usuario coordinador tengo la capacidad de...
-Registrar Club
-Registrar Equipos
-Establecer, modificar y eliminar Horarios
-Establecer, modificar y eliminar Localización de entrenamientos
-Ver asistencia de los equipos en seguimiento
-Ver asistencia de los entrenadores de equipos en seguimiento
-Ver convocatorias de equipos en seguimiento
-Ver Estadísticas de jugadores de equipos en seguimiento
-Ver fichas de partido de equipos en seguimiento
-Registrar, modificar y eliminar Jugadores

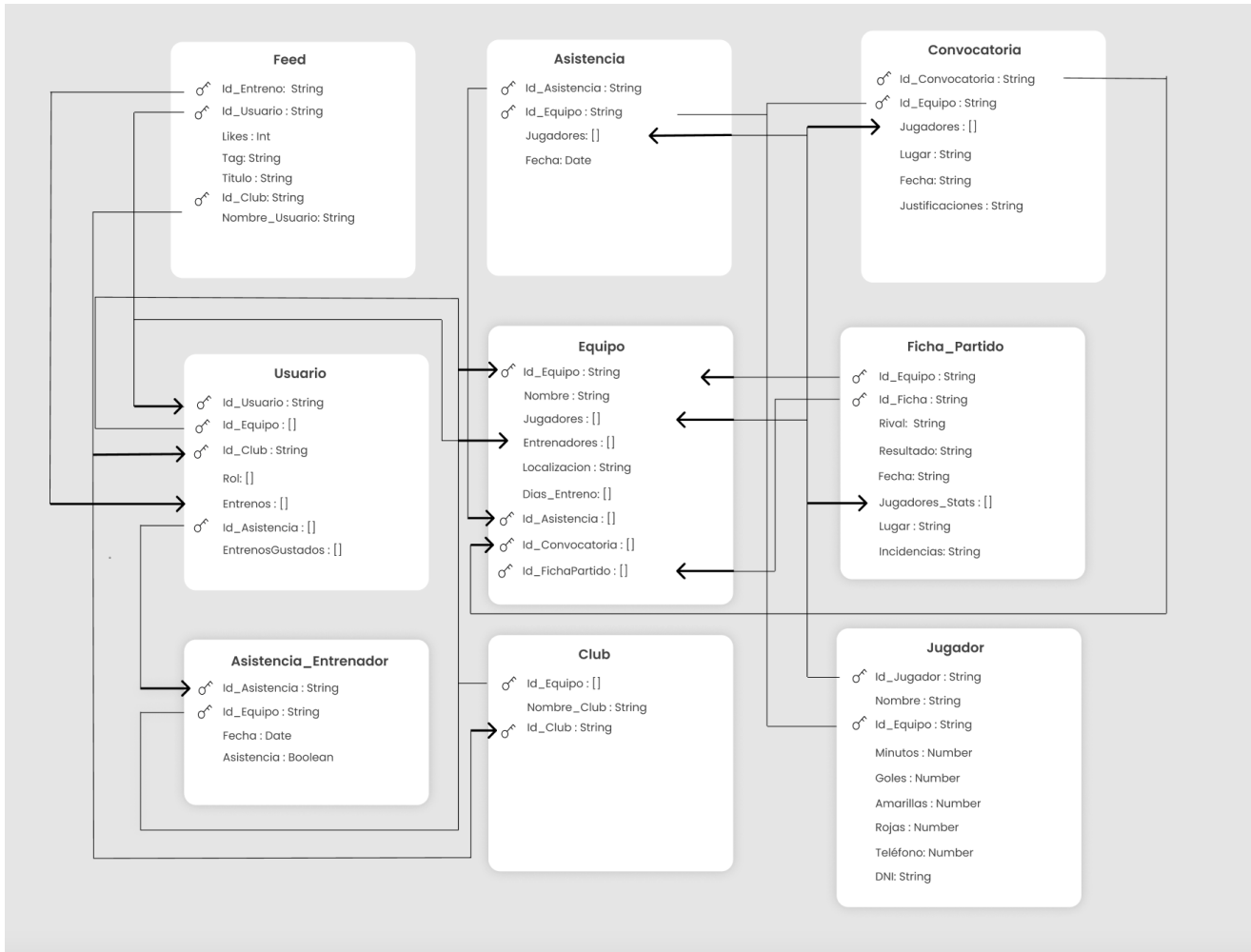
USUARIO ENTRENADOR:

Como usuario entrenador tengo la capacidad de...
-Registrar, modificar y eliminar Jugadores
-Seleccionar club y equipo
-Añadir, modificar y eliminar Jugadores
-Añadir, modificar y eliminar Información personal de los jugadores
-Asignar y eliminar Equipos para seguimiento

-Realizar, modificar y ver asistencias
-Realizar, modificar y ver convocatorias
-Realizar, modificar y ver fichas de partido
-Ver Estadísticas de jugadores de mi equipo
-Ver y modificar mi perfil
-Fichar mi asistencia
-Recibir notificaciones de recordatorio
-Subir , modificar y eliminar Entrenamientos
-Ver otros entrenamientos
-Valorar otros entrenamientos
-Buscar entrenamientos

9.2 MONTAJE BASE DE DATOS

Se adjunta el diseño de la base de datos como imagen en la carpeta origen con nombre BaseDeDatos.png también visible en la página siguiente.



9.3 PROTOTIPOS

Con el objetivo de conseguir un diseño sencillo e intuitivo, con el que además, poder describir correctamente el flujo de actividades, se realizaron varios prototipos en una primera versión de la aplicación, los cuales se han aprovechado para la codificación de la aplicación. Durante la codificación se

han realizado algunas mejoras respectivas a paletas de colores, posicionamiento de ciertos campos de texto, incluso mejoras de diseño respecto a ciertos principios de User-Experience y creación de iconos personalizados que se muestran posteriormente.

Se adjunta todos los diseños en la carpeta origen con nombre Diseños TFF.fig (También accesibles desde <https://www.figma.com/file/YWReN3AbpYU7uOHRmnNXWI/Dise%C3%B1os-TFG?node-id=0%3A1>)

10 CASOS DE USO

Para definir de forma correcta el flujo de actividades se realiza la descripción de los casos de uso para cada una de las funcionalidades que se tendrán en cuenta para la aplicación. A continuación se listan todos los casos de uso que aparecerán en la aplicación. Cabe destacar que los casos de uso siguen siendo los mismos con modificaciones en el caso de uso del registro y Login.

Id	Descripción
CU-01	-Registrarse
CU-02	-Iniciar Sesión
CU-03	-Cerrar Sesión

CU-04	-Ver mi perfil
CU-05	-Visualizar asistencia
CU-06	-Añadir asistencia
CU-07	-Modificar asistencia
CU-08	-Visualizar mis entrenos
CU-09	-Ver detalles de entreno
CU-10	-Editar perfil
CU-11	-Visualizar entrenos
CU-12	-Subir entrenos
CU-13	-Eliminar entreno
CU-14	-Filtrar entreno
CU-15	-Ver jugadores
CU-16	-Añadir jugador

11 IMPLEMENTACIÓN DEL SERVIDOR

El hecho de utilizar Firebase como servidor, facilita mucho esta tarea, ya que este funciona como un Back-End as a service por lo tanto es posible tener acceso a todo tipo de funcionalidades necesarias según el tipo de plan adquirido. La implementación de este servidor es muy sencilla de integrar con flutter debido a que ambos son herramientas de Google, por lo que simplemente con instalar la librería de firebase para flutter y añadir a la línea de configuración de firebase en flutter y el path al proyecto creado en firebase es suficiente. Además para este proyecto con el plan gratuito es suficiente, y automáticamente al crear el proyecto se obtiene un proyecto de google con el que se pueden habilitar ciertas APIs interesantes.

Entre los servicios ofrecidos por Firebase, los que se usan en este proyecto son los siguientes.

-Authentication.

Nos ofrece las llamadas necesarias para el registro, inicio de sesión, cierre de sesión y recuperación de contraseñas, además del almacenamiento cifrado de los usuarios con un único id. Con la librería de Firebase podemos acceder a todas estas funciones mediante la clase Auth declarándose en la parte inicial del código.

-Firestore Database (Data, rules and indexes).

Nos ofrece las llamadas necesarias para almacenar datos en colecciones en formato NoSQL generando documentos con un único id, actualizar, y eliminar estos datos. Con una instancia de firestore desde la librería Firebase es suficiente para poder realizar estas operaciones las cuales tienen filtros por parámetros,

CU-17	-Editar jugador
CU-18	-Eliminar jugador
CU-19	-Fichar asistencia
CU-20	-Crear convocatoria
CU-21	-Editar convocatoria
CU-22	-Ver convocatoria
CU-23	-Valorar entreno
CU-24	-Crear ficha de partido
CU-25	-Editar ficha de partido
CU-26	-Ver fichas de partido

Para ver el contenido concreto de cada caso de uso revisar el archivo Casos De Uso.pdf

documentos y colecciones para su sencillo uso.

Además Firebase nos ofrece el apartado de reglas para determinar quién puede escribir, modificar y borrar datos de las colecciones incluso para usuarios no autenticados (Regla establecida en el registro de la aplicación para crear un usuario con su respectivo club).

Por otra parte, existe la posibilidad de crear índices para realizar ciertas consultas parametrizadas de forma eficiente y en caso de que varios parámetros sean necesarios, como es el caso de buscar equipos de un club que contienen un entrenador en concreto y no disponen de los días de entrenamiento establecidos.

-Storage

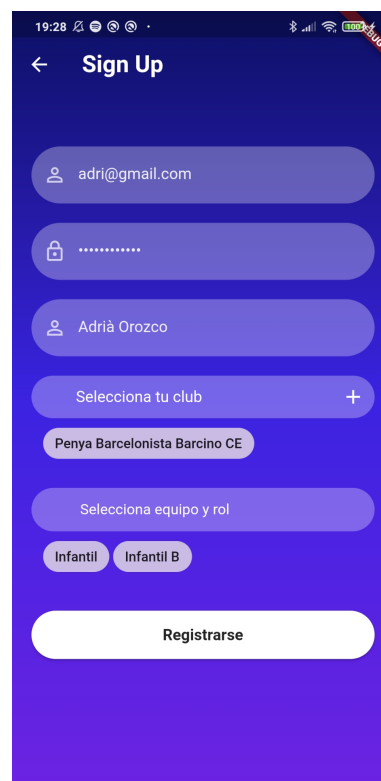
Nos ofrece el servicio necesario para el almacenamiento de archivos, además de las funciones necesarias para la misma subida y descarga de estos además de poder visualizar el contenido de un archivo y usarlo en el código. Con una simple instancia de storage desde la librería firebase podemos acceder a las funciones necesarias para conseguir o subir archivos.

12 RESULTADOS

Teniendo en cuenta el resultado del proyecto se muestran a continuación algunas de las funcionalidades reales de la aplicación, correspondientes a las funciones de asistencia de equipos, registro, selección de equipos y clubes, y registro de información de equipos. Cabe destacar que todas las funciones han sido testadas y diseñadas para cualquier tipo de situación, donde el apartado con más condiciones y funcionalidades ha sido el setUp de

entrenadores y coordinadores, que con objetivo de ahorrar costes de almacenamiento en la base de datos se ha realizado un scraping¹⁵ con un programa Python¹⁶ para obtener los clubes de cataluña los cuales han sido almacenados en un archivo json guardado en el cloud de firebase¹⁷ al cual actualizando sus normas de acceso podemos seleccionar el club navegando por su contenido de manera online para seleccionar club, y que la creación de equipos se realice a la hora del registro siempre que estos contengan información. Así como el set-up de entrenadores y coordinadores que se construye de manera dinámica a la vez que cualquier usuario se registra comprobando los coordinadores asociados a sus clubes y los equipos previamente registrados, previniendo la duplicidad de ciertos datos. Por otra parte destacar también la inclusión de la API de google places¹⁸ con objetivo de establecer direcciones reales y obtener a partir de ellas información de coordenadas, latitud y longitud para previamente realizar los cálculos estimados para el registro de asistencia con geolocalización. Por otra parte se han añadido varias funcionalidades extras para facilitar la experiencia de usuario tales como vista de calendario para los entrenamientos junto a la creación de eventos automáticos, Con esto, se muestran algunos de los resultados conseguidos.

El resto pueden ser accesibles desde el archivo 'Resultados.fig' accesibles desde la carpeta origen y también visibles desde el link: <https://www.figma.com/file/cYmKgLGazDBjAd3GaMoPui/Resultados?node-id=0%3A1>



13 FASES DE TEST

Respecto a este apartado se han realizado ciertos test pertinentes durante el desarrollo para evitar errores tardíos con un coste de solución más elevado posteriormente. Estos test han consistido en la realización de 'Path coverage' de manera que se estima todas las posibilidades de uso sobre una funcionalidad y se controlan todos estos casos. Por ejemplo en la pantalla del registro:

- Dejar campos vacíos
- email incorrecto
- contraseña incorrecta
- no seleccionar club
- no seleccionar rol y equipo
- Creación de club no registrado
- Actualización de club registrado
- Creación de equipo no registrado
- Actualización de equipo registrado
- Comprobación si existe coordinador para esa categoría en ese club
- Añadir equipo a coordinador creado
- Crear coordinador y añadir equipos ya creados

Esto se ha realizado para cada una de las funcionalidades implementadas con sus respectivos condicionales.

En la fase específica de test que se ha llevado a cabo una vez finalizada la codificación, tiene lugar ciertos test relativos al diseño y a los requisitos no funcionales así como aquellos relacionados con los mensajes de error de la aplicación de las estructuras try-catch y aquellos no descubiertos gracias a las pruebas en un entorno real.

Entre estos test encontramos ciertos parámetros que no estaban bien diseñados como mensajes de error los cuales aparecían con una simple frase y se han reemplazado con sus respectivos nuevos iconos así como ciertos márgenes de la aplicación que sobrepasaba los límites con un gran tamaño de datos los cuales también han sido solucionados.

Otra de las opciones importantes a destacar es aquellos mensajes de error producidos por las instancias de Firebase los cuales no se estaban mostrando en la aplicación.

Y por último se han detectado algunos errores ortográficos en ciertos mensajes que también han sido corregidos.

El hecho de haber llevado en paralelo una de las fases de test junto a la codificación ha hecho que esta segunda, tenga un coste de tiempo mayor debido a la complejidad que requiere realizar estos pasos, pero ha ayudado a minimizar el número de errores encontrados a posteriori y así se ha conseguido evitar la corrección de errores de alta prioridad después del desarrollo, cosa que hubiera ocasionado un empleo de tiempo mayor para esta fase y quizás en consecuencia, generar ciertos conflictos con otras funcionalidades

desarrolladas posteriormente.

14 CODIFICACIÓN Y ESTRUCTURA DE FICHEROS

Para la realización de la codificación y la estructuración del código se divide el proyecto principal en varias subcarpetas para el buen manejo de los ficheros y la buena estructuración de los componentes

La estructuración es la siguiente:

```

/lib
  /Models
  /Providers
  /Reusable_widgets
  /screens
  /storage
  /utils
main.dart

```

En la carpeta Models están los archivos relacionados con la transformación de los datos a archivos json para realizar acciones sobre la base de datos Json. Hay un modelo para cada colección de datos de la base de datos.

En la carpeta Providers están los archivos relacionados con las operaciones realizadas sobre la base de datos de manera que almacena las funciones necesarias para cada colección, separadas por archivos por cada una de ellas.

En la carpeta Reusable_widgets se almacenan ciertos componentes que son reutilizables entre otras pantallas de la aplicación así como TextFields con su respectivos validators.

En la carpeta screens se encuentra cada una de las pantallas diseñadas para la aplicación la cual cada una de ellas corresponde a un caso de uso.

En la carpeta storage encontramos los archivos relacionados con las funciones necesarias para la subida, bajada y transformación de archivos a variables globales para el posterior uso dentro de la app. Estos archivos corresponden a aquellos que se almacenan en Storage de Firebase.

En la carpeta utils encontramos ciertas declaraciones de algunos estilos necesarios para la aplicación como temas y colores personalizados (con gradientes o combinaciones).

En el archivo main.dart se encuentra el punto de inicio de la aplicación que según el estado dirige a las rutas indicadas además de inicializar los temas y las variables globales así como la autenticación del usuario.

Respecto a las clases implementadas encontramos una para cada modelo de datos que se encarga de inicializar las variables de un documento sobre una

colección en concreto y proporciona funciones para la transformación a formato json.

Por otra parte existe una clase para cada uno de los casos de uso con su respectivo Widget principal para mostrar en pantalla, sus funciones para carga de datos, y atributos personalizados para el manejo del estado del caso de uso en concreto.

15 CONCLUSIÓN

Este proyecto pretende conseguir ser pionero en el ámbito de la gestión de clubes de fútbol base, obteniendo así como resultado una aplicación capaz de manejar todas las funciones básicas que se deben desempeñar durante el día a día en cualquier club de fútbol, de manera sencilla y centralizada.

Entre los resultados a los que se hace referencia con los objetivos marcados, se encuentra la automatización de las tareas que consigue el ahorro de tiempo para los encargados, los cuales podrán dedicar este tiempo a otras funciones.

Por otra parte se consigue obtener información a

tiempo real de todos los jugadores de un equipo así como sus estadísticas durante la temporada para favorecer un mejor control global; De igual manera que se consigue obtener la información relativa a los entrenadores, de forma que se pueda llevar un control de sus obligaciones.

Con el trabajo realizado durante el proyecto, se concluye de forma satisfactoria de manera que se obtiene una herramienta con un amplio potencial para llevar a cabo todas las funcionalidades necesarias de forma rápida, eficaz y centralizada, que hasta la fecha, no disponen de unos procesos definidos ni centralizados para su realización.

Por último, como posibles ampliaciones para el proyecto se plantea la posibilidad de conectar con coordinadores de otros equipos para poder concretar de forma más sencilla amistosos de pretemporada, y la adición de un usuario directivo que pueda controlar el registro de entrenadores y coordinadores a su club para así evitar registros erróneos y llevar un control completo del club.

BIBLIOGRAFIA

- [1] "Almacenamiento en la nube para casa y el trabajo - Google Drive". Google. <https://www.google.com/intl/es-es/drive/>.
- [2] "Software de hojas de cálculo Microsoft Excel | Microsoft 365". Microsoft – Cloud, Computer, Apps und Gaming. <https://www.microsoft.com/es-es/microsoft-365/excel>.
- [3] "Oficina virtual de tràmits | Ajuntament de Barcelona". Seu electrònica. <https://seuelectronica.ajuntament.barcelona.cat/oficinavirtual/es/tramit/20200001413>.
- [4] "Desarrollo iterativo e incremental". Proyectos Ágiles. <https://proyectosagiles.org/desarrollo-iterativo-incremental/>.
- [5] "Dart programming language". Dart programming language | Dart. <https://dart.dev/>.
- [6] "Flutter - Build apps for any screen". Flutter - Build apps for any screen.
- [14] Download @ Creative Tim. <https://www.creative-tim.com/>.
- [15] "¿Qué es el web scraping?" IONOS Digitalguide.
- [16] "Welcome to Python.org". Python.org. <https://www.python.org/>.
- [17] "Cloud Firestore | Firebase Documentation". Firebase. <https://firebase.google.com/docs/firestore>.
- [18] "Overview | Places API | Google Developers". Google Developers. <https://developers.google.com/maps/documentation/places/web-service/overview>.
- any screen. https://flutter.dev/?gclid=CjwKCAiAx8KQBhAGEiwAD3EiP8s2OrHNRIlf8Imq5-VSKLZpOE4i2Dg5ZCI88HtSsbl8SnBKt60PGEBoCeuOOAvD_BwE&gclidsrc=aw.ds.
- [7] "Firebase". Firebase. <https://firebase.google.com/?hl=es>.
- [8] "Explicación Sobre Las Bases De Datos NoSQL". MongoDB. <https://www.mongodb.com/es/nosql-explained>.
- [9] Figma. <https://www.figma.com/ui-design-tool/>.
- [10] "Sobre nosotros: la historia, los logotipos y los clientes de Trello | Trello". Trello. <https://trello.com/es/about>.
- [11] "Build software better, together". GitHub. <https://github.com/about>.
- [12] "herramienta de código abierto que utiliza descripciones textuales simples para dibujar diagramas UML". PlantUML.com. <https://plantuml.com/es/>.
- [13] "Premium Bootstrap Themes and Templates". Premium Bootstrap Themes and Templates: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-el-web-scraping/>.
- [19] "multi_select_flutter | Flutter Package". Dart packages. https://pub.dev/packages/multi_select_flutter.
- [20] "Material Design". Material Design. <https://material.io/components/time-pickers/flutter>.
- [21] "JSON". JSON. <https://www.json.org/json-es.html>.
- [22] "Firebase Authentication | Firebase Documentation". Firebase. <https://firebase.google.com/docs/auth?hl=es-41>.