



UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍA INFORMÁTICA DE SEGOVIA

Grado en Ingeniería Informática de Servicios y Aplicaciones

App4CyclingLovers: Aplicación Android para los amantes del ciclismo.

Autor: Rodrigo Álvaro Rodríguez

Tutor: José Vicente Álvarez Bravo

AGRADECIMIENTOS

Agradecer en primer lugar a mi familia que siempre me ha apoyado y se ha interesado por mis avances en el proyecto, pero sin meterme ningún tipo de presión extra para finalizarlo.

Gracias a cada uno de los profesores de la titulación de Ingeniería Informática que han contribuido en inculcarme grandes conocimientos y en prepararme para la realización de este proyecto.

Gracias en especial a José Vicente Álvarez Bravo por aceptar ser el tutor de mi proyecto, y por estar a mi disposición siempre que le he necesitado proporcionándome la ayuda e indicaciones adecuadas.

Gracias a mis amigos (y compañeros) Víctor Herrero y Krystian Drag, que se ofrecieron a colaborar durante el desarrollo del proyecto proporcionándome apoyo y consejos para alcanzar unos mejores resultados.

Y por último gracias a mis amigos que me han acompañado toda la vida y a todo el mundo en general que se ha preocupado por mí a lo largo de este tiempo, ya que han significado un gran impulso para lograr mis objetivos.

RESUMEN

El propósito principal de este Trabajo de Fin De Grado consiste en el desarrollo de una aplicación móvil destinada a plataformas Android que permita a los usuarios conocer las últimas novedades producidas en el mundo del ciclismo, en cualquier momento y en cualquier lugar.

La idea principal es realizar una herramienta sencilla, accesible y fácil de usar, para que cualquier usuario, independientemente de la edad y de sus conocimientos tecnológicos, sea capaz de consultar la información que desee en pocos pasos y sin ningún tipo de dificultad.

En definitiva, se pretende aglutinar en una misma herramienta la información más relevante relativa al ciclismo de primer nivel, como son carreras, equipos y corredores. Los usuarios podrán acceder a dicho contenido de manera concurrente, en lugar de tener que acudir a diversas fuentes de información disponibles en el mercado.

PALABRAS CLAVE

Android, aplicación móvil, ciclismo, carreras, resultados, Cloud Firestore, Scraping Web.

ABSTRACT

This Bachelor's Final Project aims to develop an Android's application which allows users to know the latest news and results in the world of cycling at any time and in any place.

The main idea is to create a simple, accessible and easy-to-use tool which any user, regardless of age and technological knowledge, is able to consult any information in just a few steps without any difficulty.

Ultimately, this project aims to provide in a single app the most relevant information in top-level cycling, regarding races, teams and cyclists. Users will be able to access this content concurrently, instead of obtaining the information from different sources available.

KEY WORDS

Android, mobile application, cycling, races, results, Cloud Firestore, Web Scraping.

Índice General

Índice de Tablas	7
Índice de figuras	9
1. INTRODUCCIÓN	11
1.1 Motivación	11
1.2 Objetivos	11
1.3 Alcance	12
1.4 Organización del documento	13
2. ESTADO DEL ARTE	14
2.1 Mis Marcadores	15
2.2 Cyclingoo	15
2.3 Comparativa de propuestas de mercado	16
3. PLANIFICACIÓN DEL PROYECTO	16
3.1 Metodología de trabajo	16
3.2 Fases de trabajo y planificación temporal	17
3.3 Costes del proyecto	19
3.3.1 Costes de personal	19
3.3.2 Costes Hardware	20
3.3.3 Costes software	20
3.3.4 Costes Añadidos	20
3.4 Estimaciones Iniciales	20
3.4.1 Estimación por puntos de función (Método Albrecht)	21
3.4.2 Estimación por COCOMO	24
3.5 Presupuestos	27
3.5.1 Costes hardware y software	27
3.5.2 Costes Recursos humanos	28
3.5.3 Costes añadidos	29
3.5.4 Presupuestos finales	30
4. ANÁLISIS DEL SISTEMA	32
4.1 Actores del sistema	32
4.2 Requisitos de Usuario	33
4.2.1 Modelo de Casos de Uso	34
4.2.2 Especificación de casos de uso	36
4.3 Reglas de Negocio	54
4.4 Requisitos Funcionales	55
4.5 Requisitos No Funcionales	59

4.5.1 Requisitos de Interfaz Externa-----	59
4.5.2 Atributos de Calidad -----	59
4.6 Requisitos de Información -----	60
5.DISEÑO -----	61
5.1 Arquitectura -----	61
5.1.1 Arquitectura Lógica -----	61
5.1.2 Arquitectura física-----	62
5.2 Modelos de Diseño -----	63
5.2.1 Diagrama de Clases -----	63
5.2.2 Diagramas de Secuencia -----	64
5.2.3 Modelo Lógico de datos -----	65
5.2.4 Diccionario de datos -----	68
5.3 Diseño de Interfaces-----	72
5.3.1 Interfaces -----	72
6. IMPLEMENTACIÓN-----	94
6.1 Requerimientos Hardware y Software -----	94
6.1.1 Software -----	94
6.1.2 Hardware -----	94
6.2 Herramientas empleadas-----	94
6.2.1 Herramientas para el desarrollo móvil. -----	94
6.2.2 Herramientas de Soporte-----	95
6.3 Tecnologías empleadas -----	95
6.4 Consideraciones de Implementación-----	96
6.5 Servidores de bases de datos utilizados -----	97
6.6 Técnicas de Obtención de Datos -----	104
6.7 Organización Interna del Proyecto-----	113
7.PRUEBAS -----	114
7.1 Pruebas de Caja Blanca -----	114
7.2 Pruebas de Caja Negra -----	114
8.MANUAL DE INSTALACIÓN -----	119
8.1 Instalación a través de la Tienda Oficial -----	119
8.2 Instalación a través del Archivo de Ejecución -----	120
9.CONCLUSIONES Y POSIBLES MEJORAS-----	122
9.1 Conclusiones-----	122
9.2 Posibles mejoras-----	123
10. REFERENCIAS -----	124

10.1 Bibliografía -----	124
10.2 Webgrafía-----	125
10.3 Contenidos Audiovisuales -----	126

Índice de Tablas

Tabla 1 Comparativa Herramientas del mercado	16
Tabla 2 Distribución Etapas Planificación.....	19
Tabla 3 Costes Hardware del proyecto	20
Tabla 4 Costes Software del proyecto.....	20
Tabla 5 Costes añadidos al proyecto.....	20
Tabla 6 Entradas de usuario	21
Tabla 7 Salidas externas	21
Tabla 8 Consultas del usuario.....	22
Tabla 9 Ficheros Lógicos Internos	22
Tabla 10 Ficheros lógicos externos	22
Tabla 11 Equivalencia de los Puntos de Función Sin Ajustar	23
Tabla 12 Factores de Complejidad y su valor	23
Tabla 13 Factores Conductores del Coste del proyecto.....	26
Tabla 14 Costes Hardware Totales Estimación Albrecht.....	27
Tabla 15 Costes Hardware Totales Planificación Inicial	28
Tabla 16 Costes Software Totales	28
Tabla 17 Costes Totales RRHH Estimación Albrecht	29
Tabla 18 Costes Totales RRHH Planificación Inicial	29
Tabla 19 Costes Añadidos Estimación Albrecht	29
Tabla 20 Costes Totales añadidos Planificación Inicial.....	29
Tabla 21 Comparativa Costes Totales Proyecto entre Estimación y Planificación.....	30
Tabla 22 Costes Hardware Finales	31
Tabla 23 Costes Software Finales.....	31
Tabla 24 Costes Recursos Humanos Finales.....	31
Tabla 25 Costes Añadidos Finales	31
Tabla 26 Caso de uso "Acceder a la pantalla principal de la aplicación"	36
Tabla 27 Caso de uso "Visualizar Lista Carreras"	37
Tabla 28 Caso de uso " Visualizar datos Carrera"	37
Tabla 29 Caso de uso "Visualizar resultados Corredores y/o Equipos"	38
Tabla 30 Caso de uso "Consultar etapas"	38
Tabla 31 Caso de uso "Seleccionar Etapa"	39
Tabla 32 Caso de uso "Visualizar resultados etapa"	39
Tabla 33 Caso de uso "Consultar si la etapa está en curso".....	40
Tabla 34 Caso de uso "Consultar Clasificación Mundial Corredores"	41
Tabla 35 Caso de uso "Visualizar Lista Equipos"	41
Tabla 36 Caso de uso "Seleccionar Equipo y ver sus datos"	42
Tabla 37 Caso de uso "Consultar Corredores de equipo"	42
Tabla 38 Caso de uso "Seleccionar Equipo favorito"	43
Tabla 39 Caso de uso "Seleccionar Corredor y ver sus datos"	44
Tabla 40 Caso de uso "Acceder sitio web corredor"	44
Tabla 41 Caso de uso "Seleccionar Corredor favorito"	45
Tabla 42 Caso de uso "Consultar favoritos actuales"	45
Tabla 43 Caso de uso "Acceder datos de elementos favoritos"	46
Tabla 44 Caso de uso "Filtrar Corredor por nombre"	47
Tabla 45 Caso de uso "Seleccionar Corredor filtrado y ver sus datos"	47
Tabla 46 Caso de uso "Consultar sitios web de noticias disponibles"	48

Tabla 47 Caso de uso "Filtrar Carrera por nombre"	49
Tabla 48 Caso de uso "Seleccionar Carrera y ver sus datos"	49
Tabla 49 Caso de uso "Consultar medios para compartir App"	50
Tabla 50 Caso de uso "Enviar comentarios"	51
Tabla 51 Caso de uso "Solicitar email de respuesta"	51
Tabla 52 Caso de uso "Acceder sección Favoritos"	52
Tabla 53 Caso de uso "Guardar favorito"	52
Tabla 54 Caso de uso "Seleccionar web y leer noticias"	53
Tabla 55 Caso de uso "Compartir enlace descarga App"	53
Tabla 56 Requisitos Funcionales asociados a los casos de uso	59
Tabla 57 Colecciones de Datos.....	68
Tabla 58 Diccionario Datos-Coleccion "Carreras"	69
Tabla 59 Diccionario Datos- Colección "Equipos"	69
Tabla 60 Diccionario- Datos Colección "Corredores"	70
Tabla 61 Diccionario Datos- Colección "Resultados de Carreras"	71
Tabla 62 Colección "Clasificación Mundial de Corredores"	71
Tabla 63 Interfaz "Pantalla Inicial"	73
Tabla 64 Interfaz "Pantalla Principal"	75
Tabla 65 Interfaz "Detalles Carrera"	77
Tabla 66 Interfaz "Detalles Etapa"	78
Tabla 67 Interfaz "Sección Clasificación"	79
Tabla 68 Interfaz "Sección Favoritos-Acceso Pantalla Principal"	80
Tabla 69 Interfaz "Sección Favoritos-Acceso Seleccion Item favorito"	82
Tabla 70 Interfaz "Menú Lateral"	83
Tabla 71 Interfaz "Sección Corredores"	84
Tabla 72 Interfaz "Sección Carreras"	85
Tabla 73 Interfaz "Sección Noticias"	86
Tabla 74 Interfaz "Pantalla Novedades"	87
Tabla 75 Interfaz "Sección Contacto"	89
Tabla 76 Interfaz "Sección Créditos"	90
Tabla 77 Interfaz "Sección Compartir App"	91
Tabla 78 Interfaz "Detalles Equipo"	92
Tabla 79 Interfaz "Detalles Corredor"	93
Tabla 80 Prueba de Caja Negra: "Seleccionar fecha Calendario"	115
Tabla 81 Prueba de Caja Negra:"Buscador de Corredores"	115
Tabla 82 Prueba de Caja Negra: "Buscador de Carreras"	115
Tabla 83 Prueba de Caja Negra: "Envío de comentarios"	116
Tabla 84 Prueba de Caja Negra: "Acceso a web de noticias"	116
Tabla 85 Prueba de Caja Negra: "Compartir Aplicación"	117
Tabla 86 Prueba de Caja Negra:"Almacenar elemento favorito"	117
Tabla 87 Prueba de Caja Negra:"Consultar listado Equipos"	118
Tabla 88 Prueba de Caja Negra:"Acceder web del corredor"	118
Tabla 89 Prueba de Caja Negra:"Acceso a Detalles Carrera"	118

Índice de figuras

Ilustración 1 Árbol de Características.....	13
Ilustración 2 Equivalencia líneas de código-punto de función.....	25
Ilustración 3 Ponderación Factores Conductores del Coste genéricos	25
Ilustración 4 Modos de desarrollo del Modelo COCOMO INTERMEDIO.....	26
Ilustración 5 Diagrama de Casos de uso 1.....	35
Ilustración 6 Diagrama de Casos de uso 2.....	35
Ilustración 7 Arquitectura lógica de la aplicación	62
Ilustración 8 Arquitectura física de la aplicación	63
Ilustración 9 Diagrama de Clases	64
Ilustración 10 Diagrama de Secuencia "Visualizar Listado de Carreras en una fecha".....	64
Ilustración 11 Diagrama de Secuencia "Consultar resultados de los corredores en una carrera" ..	65
Ilustración 12 Diagrama de Secuencia "Enviar comentario al administrador del sistema".....	65
Ilustración 13 Estructura Base de datos "Cloud Firestore"	66
Ilustración 14 Estructura Documentos en base de datos	66
Ilustración 15 Estructura sub-colecciones en base de datos	67
Ilustración 16 Equivalencia modelo datos Cloud Firestore- modelo de datos programación.....	67
Ilustración 17 Proyectos Cuenta Firebase.....	98
Ilustración 18 Creación Nuevo Protecto Firebase 1.....	98
Ilustración 19 Creación Nuevo Proyecto Firebase 2	98
Ilustración 20 Servicios disponibles Firebase	99
Ilustración 21 Integración Cloud Firestore en la aplicación (1).....	99
Ilustración 22 Integración Cloud Firestore en la aplicación (2).....	100
Ilustración 23 Integración Cloud Firestore en la aplicación (3).....	100
Ilustración 24 Integración Cloud Firestore en la aplicación (4.1).....	101
Ilustración 25 Integración Cloud Firestore en la aplicación (4.2).....	101
Ilustración 26 Procedimiento Extracción Datos de " Equipos inscritos en la temporada"	105
Ilustración 27 Lista de Equipos en la Temporada 2021.....	105
Ilustración 28 Procedimiento Extracción Datos de " Corredores pertenecientes a un Equipo"....	106
Ilustración 29 Procedimiento Modificación Campos Extraídos (1)	106
Ilustración 30 Procedimiento Modificación Campos Extraídos (2)	106
Ilustración 31 Ejemplo Solicitud API "Sportsradar" (1)	107
Ilustración 32 Ejemplo Solicitud API "Sportsradar" (2)	108
Ilustración 33 Integración API "Sporstradar" con entorno "Pycharm".....	108
Ilustración 34 Procedimiento Extracción Datos de " Calendario de Carreras de la Temporada" ..	109
Ilustración 35 Procedimiento Extracción Datos de "Ranking Mundial de Corredores"	109
Ilustración 36 Procedimiento Extracción Datos de "Resultados de una carrera" (1)	110
Ilustración 37 Procedimiento Extracción Datos de "Resultados de una carrera" (3)	110
Ilustración 38 Procedimiento Extracción Datos de "Resultados de una carrera" (2)	110
Ilustración 39 Integración Cloud Firestore con entorno Pycharm (1).....	111
Ilustración 40 Inserción en Cloud Firestore de "Corredores pertenecientes a un Equipo"	112
Ilustración 41 Inserción en Cloud Firestore de " Ranking Mundial de Corredores"	112
Ilustración 42 Instalación App desde Tienda Oficial (2)	119
Ilustración 43 Instalación App desde Tienda Oficial (1)	119
Ilustración 44 Instalación App desde Tienda Oficial (3)	120
Ilustración 45 Instalación App desde Tienda Oficial (5)	120
Ilustración 46 Instalación App desde Tienda Oficial (4).....	120

Ilustración 47 Instalación App a través de archivo ejecución (1).....	121
Ilustración 48 Instalación App a través de archivo ejecución (3).....	121
Ilustración 49 Instalación App a través de archivo ejecución (2).....	121

1. INTRODUCCIÓN

1.1 Motivación

El ciclismo es un deporte con mucha tradición e historia (principalmente en el Continente Europeo). Siempre ha sido considerado como uno de los deportes que comportan un mayor sufrimiento y dedicación, y por lo tanto de mayor gloria para el deportista en el momento de alcanzar éxitos importantes.

En este sentido se desarrollan múltiples competiciones ciclistas en todas partes del mundo, ya que el ciclismo con el paso de los años ha ido adquiriendo un carácter más global, por lo que cada vez hay un mayor seguimiento e interés por parte de los aficionados y amantes de este deporte en numerosos países.

Debido a lo mencionado anteriormente surge una mayor necesidad por parte de los aficionados de poder conocer y acceder a la información y a los resultados de las competiciones ciclistas sobre las que tengan inquietudes e intereses.

Por todo ello se propone en este proyecto la creación de una aplicación móvil que englobe las competiciones ciclistas de mayor prestigio que se disputan a lo largo del año, junto a información complementaria respecto a equipos y corredores que participan en las mismas.

Además, resulta imprescindible puntualizar que en estos momentos no se encuentra en el mercado apenas productos que aglutinen esas características, por lo tanto, resulta una gran oportunidad de negocio.

Otro factor para tener en consideración es el auge de las redes sociales en los últimos años, ya que permiten a cualquier usuario que permanezca conectado con el “mundo del ciclismo” estar al día de las competiciones que se disputan, por lo que, si contaran con una aplicación complementaria para disponer de mayor conocimiento de éstas, la recibirían con los brazos abiertos.

1.2 Objetivos

El objetivo principal de este proyecto es el desarrollo de una aplicación para dispositivos móviles de Android que permita al usuario estar al día de las últimas novedades que surjan en el ámbito del ciclismo mundial.

Una vez enunciado el objetivo principal que debe satisfacer la aplicación, también se pueden enumerar un conjunto de objetivos derivados que ayudan a dotar de una mayor virtud a la aplicación. Dichos objetivos son:

- Centralizar la información relativa al ámbito de ciclismo por categorías (carreras, corredores...).
- Agilizar el acceso a las noticias de ciclismo a los usuarios de la aplicación.
- Fomentar el interés por la práctica del ciclismo en los usuarios de la aplicación.
- Aumentar el uso de la bicicleta en la vida diaria de los usuarios para reducir la contaminación provocada por el vehículo privado.
- Sea fácilmente usable para cualquier persona.

Para la consecución de estos objetivos, la aplicación contará con un trabajo complementario para la obtención y gestión de los datos que garanticen la veracidad y completitud de estos antes de proporcionárselos a los usuarios finales.

1.3 Alcance

El proyecto va destinado a aquellos usuarios que se consideran amantes del ciclismo y desean estar permanentemente conectados con las novedades que se produzcan en este ámbito, aunque no necesariamente excluye a los usuarios que se sienten atraídos e interesados por este deporte, pero viven el ciclismo con una menor intensidad.

Por razones obvias, está pensado para los usuarios que dispongan de un dispositivo electrónico con sistema operativo “Android” y que cuenten con una conexión a Internet estable para poder consultar la aplicación y acceder a todas sus funcionalidades en cualquier momento que lo deseen.

En un primer momento la aplicación estará dotada y cargada con datos relativos a las competiciones ciclistas, equipos y corredores pertenecientes a la Categoría “World Tour” (Nivel de 1ª División) de las temporadas 2020 y 2021, pero en un futuro se podrían ir incorporando nuevas temporadas ciclistas a medida que se vayan desarrollando.

En este sentido, el sistema va a contar con las siguientes funcionalidades principales:

- **Gestión de Carreras:** incluye las características y funcionalidades correspondientes a la búsqueda y visualización de las carreras ciclistas que se disputan, junto con el acceso a la información y resultados de estas. Permite buscar las carreras por su nombre, y por el mes y año en que se disputan.
- **Gestión de Corredores:** incluye las características y funcionalidades correspondientes a la búsqueda de los diferentes Corredores que forman parte de algún equipo durante las temporadas disputadas. Además, también ofrece la posibilidad de acceder a la información presente en su perfil, y escoger uno de ellos como favorito.
- **Gestión de Equipos:** incluye las características y funcionalidades correspondientes a la visualización de los Equipos inscritos en cada temporada ciclista, junto con el acceso a la información y corredores asociados a cada uno de ellos. Además, ofrece la posibilidad de seleccionar un determinado equipo como favorito.
- **Portal de Noticias:** incluye las características y funcionalidades correspondientes al acceso a un determinado sitio web de noticias para conocer las últimas novedades del mundo del ciclismo.
- **Gestión de Comunicaciones:** incluye las características y funcionalidades correspondientes al envío de opiniones, dudas, sugerencias por parte del usuario hacia el encargado de dar soporte al sistema. Incluye la posibilidad de solicitar una respuesta individualizada.
- **Gestión de Favoritos:** incluye las características y funcionalidades correspondientes al acceso directo a la información del corredor y/o equipo escogido como favorito en las respectivas secciones por parte del usuario.

- **Gestión Ranking:** incluye las características y funcionalidades correspondientes a la visualización de la Clasificación Mundial Individual de Corredores (Ranking UCI) en el panorama actual.
- **Compartir Aplicación:** incluye las características y funcionalidades correspondientes al envío del enlace de descarga de la aplicación desde la tienda oficial de “Android” a través de numerosas vías (por ejemplo, a través de Whatsapp o Gmail). En esta primera versión inicial el sistema no se subirá a la tienda de aplicaciones de Android a la espera de la revisión y aprobación final del proyecto.

A continuación, se presenta el árbol de características de la aplicación (ilustración 1):

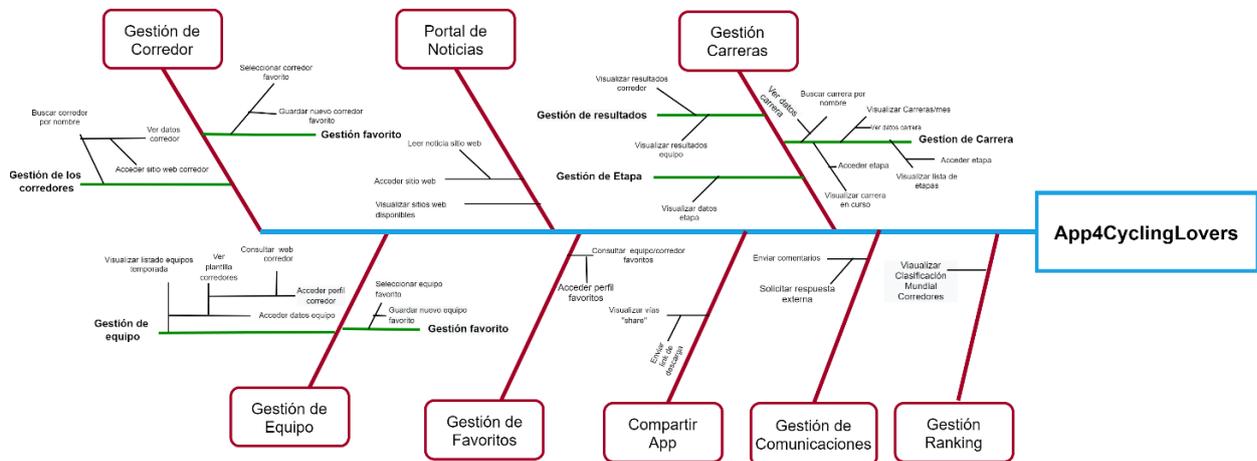


Ilustración 1 Árbol de Características

1.4 Organización del documento

En este apartado se describe de forma detallada como se ha estructurado este documento. La memoria consta de 10 secciones u apartados principales, y en cada uno de ellos se presentará su contenido de una manera bien diferenciada:

1. **Introducción:** Sección donde se describen los conceptos generales del proyecto, como los objetivos que se persiguen, el alcance, la motivación y las características principales del sistema.
2. **Estado del Arte:** Sección donde se reflejan las posibles herramientas similares a la nuestra que existen en el mercado, para analizar las virtudes y defectos en contraposición con la herramienta desarrollada.
3. **Planificación del Proyecto:** Sección donde se refleja la metodología usada en el desarrollo, junto con la planificación temporal seguida y los requerimientos presupuestarios para el desarrollo de la herramienta.

4. **Análisis del sistema:** Sección dónde se refleja los requisitos demandados por el usuario relativos a la funcionalidad de la herramienta que debe garantizar, junto con los actores que intervienen en la misma.
5. **Diseño de Sistema:** Sección donde se refleja la arquitectura física y lógica imprescindible para llevar a cabo la herramienta, junto con la experiencia del usuario.
6. **Implementación:** Sección donde se reflejan las cuestiones relativas al desarrollo de la herramienta, por ejemplo, las tecnologías utilizadas. Además, se explican todas las técnicas utilizadas para la obtención del conjunto de datos cargados en la aplicación.
7. **Pruebas:** Sección donde se reflejan las diferentes actividades realizadas para poner a prueba el correcto funcionamiento de la herramienta.
8. **Manuales:** Sección donde se reflejan los pasos para una correcta instalación de la aplicación en el dispositivo del usuario.
9. **Conclusiones:** Sección dónde se refleja la opinión general relativa al desarrollo final de la herramienta, y los posibles aspectos a mejorar en el futuro.
10. **Bibliografía:** Sección dónde se reflejan las referencias de documentación externas utilizadas durante la realización del proyecto.

2. ESTADO DEL ARTE

Para poder comenzar el proyecto planteado y alcanzar los objetivos propuestos de manera satisfactoria lo primero que debemos hacer es analizar las herramientas de características similares presentes ya en el mercado actual. En nuestro caso, se observa una escasez de herramientas capaces de satisfacer todas las necesidades actuales de los aficionados al ciclismo por lo que se decidió acometer este proyecto.

En el mercado existen numerosas aplicaciones para dispositivos móviles que proporcionen información y resultados al instante de otros deportes que gozan de gran popularidad entre los usuarios de todo el mundo.

Un gran ejemplo se encuentra en la aplicación conocida como “**MisMarcadores**”, por lo que dicha aplicación ha sido tomada como ejemplo en el desarrollo de nuestra herramienta, ya que el objetivo que se pretende lograr es similar, pero en este caso en lo relativo al mundo del ciclismo.

En el ámbito del ciclismo solo hemos localizado un sitio web especializado: “proccyclingstats.com” que dispone de información muy completa y actualizada de todas las carreras que se disputan a lo largo del planeta. Sin embargo, dicha página carece de aplicación disponible en dispositivos móviles.

En este sentido, se encontró una aplicación para dispositivos móviles conocida como “**Cyclingoo**”. A pesar de ser una herramienta para dispositivos móviles a considerar no englobaba alguna de las cuestiones que buscamos facilitar con nuestra herramienta.

Por lo tanto, las herramientas que servirán como referencia antes del desarrollo de nuestro sistema “App4CyclingLovers” son principalmente dos: “MisMarcadores” y “Cyclingoo”.

2.1 Mis Marcadores

Mis Marcadores es una aplicación de resultados para una gran diversidad de deportes (fútbol, tenis, hockey) que te permitirá conocer al instante los cambios que puedan surgir en cada uno de ellos gracias a su servicio de notificaciones.

Además, cuenta con información muy completa y actualizada a la que se puede acceder de manera fácil y rápida en sus diferentes secciones.

La gran ventaja es que dispone de una interfaz muy intuitiva, en la que se despliega un menú lateral dónde debes escoger el deporte a visualizar, y a continuación se presentan las competiciones correspondientes a dicho deporte que se disputan en la fecha seleccionada en un calendario presente en la parte superior de la pantalla. Para acceder a la información más detallada simplemente se deberá pulsar en cualquiera de los elementos mostrados.

Las funcionalidades y características principales son las siguientes:

- Acceso rápido a resultados deportivos.
- Gran disponibilidad de deportes, como, por ejemplo: fútbol, baloncesto, tenis, balonmano, golf, etc.
- Actualizaciones en tiempo real de las situaciones que se producen en cada uno de los eventos.
- Acceso directo a partidos, equipos y competiciones preferidas.
- Servicio de notificaciones y alertas de las competiciones seleccionadas.
- Clasificaciones y detalles de los partidos.
- Comentarios de texto de los partidos en vivo.
- Datos históricos previos de competiciones y equipos.

2.2 Cyclingoo

Cyclingoo es una aplicación de noticias y resultados ciclistas en la que encontrarás las carreras más importantes de la temporada de ciclismo focalizada en las 3 grandes vueltas de la temporada 2020 (Grand Tours: Tour, Giro y Vuelta) además de todos los datos de las carreras de las temporadas anteriores (2016, 2017, 2018 y 2019).

Las funcionalidades y características principales son las siguientes:

- Detallados perfiles de equipos y ciclistas
- Completa información de carreras y etapas
- Noticias sobre ciclismo
- Notificaciones

2.3 Comparativa de propuestas de mercado

En este apartado vamos a reflejar cuales de las funcionalidades presentes en las herramientas enumeradas se encuentran incorporadas actualmente en nuestro sistema, para analizar sus virtudes y defectos (Tabla 1).

	Mis Marcadores	Cyclingoo	App4CyclingLovers
Resultados Deportivos	✓	✓	✓
Cobertura varios deportes	✓	X	X
Eventos al instante	✓	✓	X
Equipos y Corredor favorito	✓	X	✓
Notificaciones	✓	✓	X
Clasificaciones	✓	✓	✓
Archivo Histórico	✓	✓	✓
Perfil Corredores y Ciclistas	X	✓	✓
Noticias	X	✓	✓
Envío Comentarios	✓	X	✓
Versión en inglés	✓	X	✓

Tabla 1 Comparativa Herramientas del mercado

3. PLANIFICACIÓN DEL PROYECTO

En esta sección se van a describir la metodología de trabajo empleada para el desarrollo del proyecto, la organización y la distribución de las fases de trabajo, junto con los costes y estimaciones iniciales del presupuesto requerido para la finalización de este.

3.1 Metodología de trabajo

Para este proyecto se ha decidido utilizar el modelo de “Desarrollo Iterativo Incremental”, el cual consiste en el desarrollo del sistema a través de “mini proyectos” cortos, conocidos como iteraciones. La idea se basa en ir incorporando las características y funcionalidades a la herramienta poco a poco, es decir la herramienta resulta mejorada al final de cada iteración.

El resultado de cada una de las iteraciones es un sistema que puede ser probado, integrado y ejecutado de manera satisfactoria, y cada una de ellas consta de sus propias actividades de:

- Análisis de requisitos
- Diseño
- Implementación
- Pruebas

Además, este modelo de desarrollo nos va a servir para que los requisitos y funcionalidades individuales que teníamos en mente para nuestra herramienta vayan integrándose y adecuándose de una manera más refinada al sistema ya diseñado en cada momento.

Por los motivos descritos considero que este modelo de desarrollo resulta el más adecuado para nuestro proyecto, ya que dispone de una serie de beneficios que nos ayudarán durante el desarrollo de este:

- Mitigación lo más pronto posible de riesgos altos en una posible inestabilidad del sistema.
- Progreso visible en las primeras etapas.
- Temprana retroalimentación, compromiso de los usuarios y adaptación que nos lleva a mejorar el sistema.
- Gestión de la complejidad más sencilla.
- Conocimiento adquirido aprovechable iteración tras iteración.

Sin embargo, este modelo requiere de una metodología de trabajo un poco rígida durante el desarrollo de todo el proyecto, ya que para finalizar cada una de las iteraciones es necesario haber completado todas las fases para garantizar el correcto funcionamiento del sistema antes de añadir nuevas funcionalidades.

En el caso de nuestro proyecto se han considerado las siguientes iteraciones para lograr la incorporación total de las funcionalidades y el mantenimiento satisfactorio del sistema.

- Iteración inicial para el análisis exhaustivo de requisitos.
- iteración para el desarrollo de la interfaz de usuario principal.
- Iteración para la conexión de la aplicación con la base de datos.
- Iteración para la consulta de información relativa al ranking mundial.
- Iteración para la consulta de información relativa a los equipos y corredores.
- Iteración para el sistema de comunicación del usuario con el soporte del sistema.
- Iteración para la consulta de noticias al acceder al sitio web.
- Iteración para el sistema de gestión de las carreras y sus resultados.
- Iteración para el sistema de gestión de corredores y equipos favoritos.
- Iteración para revisión final y documentación.

Por último, resulta importante puntualizar que coincidiendo con las diferentes iteraciones también se ha realizado el trabajo complementario de obtención y carga de los datos correspondientes en la base de datos, para que se pudiera acceder a través de la aplicación.

3.2 Fases de trabajo y planificación temporal

Para la planificación inicial de este proyecto resulta necesario puntualizar diversas cuestiones previas que pueden influir en el desarrollo y por lo tanto afectar a la fecha de finalización de este.

- La comunicación de inicio del proyecto se ubica en torno al mes de marzo de 2020, aunque no se acomete realmente el comienzo de este hasta más adelante.

- La planificación real del proyecto comienza en torno al día 25 de septiembre de 2020, ya que debido a circunstancias personales no se decidió acometer una planificación exhaustiva con anterioridad.
- Los días (salvo alguna excepción) estipulados para trabajar son de lunes a viernes, durante las franjas horarias de por la mañana y por la tarde, sin ningún tipo de limitación.
- La media de horas diarias trabajadas se fijó en 4 horas.
- Los días pertenecientes a periodos vacacionales de carácter nacional, autonómico, y provincial se han eliminado de la planificación, ya que presumiblemente no se van a emplear en el desarrollo del sistema, salvo ciertas excepciones. En nuestro caso se descartarían una cifra cercana a 15 días aproximadamente.
- La falta de experiencia en la utilización de las tecnologías empleadas en el desarrollo del sistema requiere de un trabajo de aprendizaje y familiarización con los componentes bastante exhaustivo, por lo tanto, se ralentizarán ciertos plazos de desarrollo.
- En las primeras fases y actividades del proyecto se empleará más tiempo para completar las iteraciones, ya que el producto partirá de cero y resultará más sencillo integrar los diversos componentes y funcionalidades nuevas a medida que avance el desarrollo del proyecto. Aunque en las fases finales se requerirá un importante trabajo en la realización de pruebas de funcionamiento, ya que la complejidad del sistema será cada vez mayor.

Por lo tanto, en nuestro caso plantearemos el día 10 de abril de 2021 como fecha de finalización estimada así que nuestro proyecto habría tenido una duración global de 198 días aproximadamente, aunque la cifra de días efectivos se reduce a 135 (tabla 2).

A continuación, procedo a reflejar la distribución temporal (siempre de una manera aproximada) de cada una de las etapas seguidas en el desarrollo del proyecto (tabla 2):

ID	Descripción	Fecha Inicio	Fecha Fin	Días empleados
Etapas 1	Análisis de requisitos.	25/09/2020	30/10/2020	26
Etapas 2	Interfaz de usuario principal.	2/11/2020	20/11/2020	15
Etapas 3	Integración con la base de datos.	23/11/2020	30/11/2020	6
Etapas 4	Sección Ranking Mundial.	1/12/20	15/12/2020	11
Etapas 5	Sección Equipos y Corredores.	16/12/2020	22/12/2020	5

Etapa 6	Comunicación usuario con el soporte.	28/12/2020	04/01/2021	4
Etapa 7	Sección de Noticias.	07/01/2021	15/01/2021	7
Etapa 8	Gestión de Carreras y Resultados.	18/01/2021	10/02/2020	18
Etapa 9	Gestión de Equipo y Corredor Favorito.	11/02/2020	23/02/2020	9
Etapa 10	Revisión final.	24/02/2020	10/04/2021	34
Cantidad total de días empleados:				135

Tabla 2 Distribución Etapas Planificación

Por último, cabe destacar que las fases anteriores pueden sufrir alteraciones y alargarse o acortarse respectivamente en función de los conflictos que pudieran surgir durante su desarrollo y el tiempo requerido para poder solventarlos.

3.3 Costes del proyecto

3.3.1 Costes de personal

Estos costes hacen referencia a los recursos humanos empleados durante todo el proceso de desarrollo de la herramienta. Para nuestro proyecto sólo haría falta referirnos al autor de éste, ya que es quien va a hacerse cargo de las distintas fases del proyecto, aunque en cada una de ellas actuará con el rol correspondiente. Los roles para desempeñar son los siguientes:

- Jefe de Proyecto: se encarga de gestionar el equipo de trabajo y revisión y elaboración de la documentación
- Analista de requisitos.
- Diseñador de interfaces de usuario.
- Desarrollador de componentes y su implantación en el sistema.
- Ejecutor de pruebas del funcionamiento del sistema.

Los costes reales de cada uno de estos roles aparecen detallados en el apartado **“3.5.2 Costes Recursos Humanos”** en función del tiempo empleado en cada uno de ellos y del salario base medio definido en la actualidad.

3.3.2 Costes Hardware

Estos costes hacen referencia a los diferentes soportes físicos empleados en la realización del proyecto. Para nuestro proyecto los costes asociados se corresponden con (tabla 3):

Componente	Precio (€)	Vida útil media (años)
Ordenador Portátil 1	700	7
Ordenador Portátil 2	550	7
Dispositivo Móvil 1	490	3
Ratón Inalámbrico	10	4

Tabla 3 Costes Hardware del proyecto

3.3.3 Costes software

Estos costes hacen referencia a los programas y herramientas utilizadas en el desarrollo del proyecto. En nuestro caso no han supuesto ningún valor, ya que disponíamos de algunos de los programas utilizados y el resto son de licencia gratuita, los cuales son (tabla 4):

Componente	Precio (€)
Windows 10 Home	Gratuito
Word (Office 365)	Gratuito
Pycharm Community Edition	Gratuito
Android Studio	Gratuito
Google Drive	Gratuito
Github	Gratuito
Firebase	Gratuito
SporstRadar Api	Prueba Gratuita

Tabla 4 Costes Software del proyecto

3.3.4 Costes Añadidos

Estos costes incrementan el coste total de desarrollo del proyecto, pero no se pueden posicionar en ninguna categoría (tabla 5).

Descripción	Coste/mes (€)
Conexión a Internet (Wifi) + Tarifa soporte móvil	35
Gasto eléctrico	15 aproximadamente

Tabla 5 Costes añadidos al proyecto

3.4 Estimaciones Iniciales

En esta sección se va a detallar la estimación inicial del proyecto en cuanto a sus costes y su duración. Existen diferentes métodos de estimación empleados en el desarrollo software, por lo que en nuestro proyecto nos vamos a decantar por el método de Albrecht con puntos de función y el método conocido como COCOMO.

El objetivo que se persigue es observar cuál de los dos métodos obtiene unos resultados más precisos y así poder compararlo con la planificación inicial realizada para obtener una visión más exhaustiva y precisa con relación a los costes temporales y presupuestarios finales del proyecto.

3.4.1 Estimación por puntos de función (Método Albrecht)

Este método se basa en una métrica que cuantifica la funcionalidad que se debe entregar al usuario para construir la aplicación.

Los parámetros que sirven para evaluar dicha funcionalidad son:

- **Número de entradas:** datos que aporta el usuario al sistema (nombre de ficheros, menú de selección). Siempre actualizan algún fichero interno.
- **Número de salidas:** datos que el sistema aporta al usuario (informes, mensajes)
- **Número de ficheros lógicos internos:** ficheros o bases de datos internos al sistema (es decir sólo los utiliza el sistema, ficheros maestros).
- **Número de ficheros externos:** ficheros o bases de datos externos a los que accede el sistema con el único objetivo de obtener información (es decir son mantenidos por otras aplicaciones).
- **Número de consultas externas:** entradas que requieren de una respuesta por parte del sistema. No modifica los datos del sistema.

Por lo tanto, se necesita detallar los diferentes elementos organizados en sus respectivas categorías (tablas 6, 7, 8, 9 y 10) presentes en nuestro sistema (junto con la complejidad de cada uno de ellos).

Entradas de usuario	
Descripción	Complejidad
Parámetros Búsqueda de Carrera (nombre)	Media
Parámetros Búsqueda de Corredor (nombre)	Media
Parámetros Selección mes/año Carreras	Baja
Mensaje de comentarios al administrador	Baja
Dirección de email para respuesta al usuario	Baja
Selección Equipo favorito	Baja
Selección Corredor Favorito	Baja
Selección método Compartir App	Baja

Tabla 6 Entradas de usuario

Salidas externas	
Descripción	Complejidad
Mensaje del usuario al administrador	Baja
Mensaje generado con enlace descarga app	Baja
Mensajes al usuario	Baja

Tabla 7 Salidas externas

Consultas externas	
Descripción	Complejidad
Consultar listado carreras mes/año	Baja
Consultar detalles carrera	Baja
Consultar resultados corredores	Media
Consultar equipos carrera	Media
Consultar listado etapas de Carrera	Media
Consultar Carrera en curso	Baja
Consultar detalles etapa	Baja
Consultar resultados etapa	Media
Consultar equipos etapa	Media
Consultar Clasificación Mundial UCI	Media
Consultar favoritos	Baja
Consultar listado equipos-temporada	Baja
Consultar detalles equipo	Baja
Consultar plantilla de corredores-equipo	Media
Consultar detalles corredores	Baja
Consultar sitio web del corredor	Baja
Consultar corredores por filtro (nombre)	Media
Consultar sitio web noticias	Baja
Consultar carreras por filtro (nombre)	Media
Consultar vías para compartir app	Baja
Consultar información créditos app	Baja

Tabla 8 Consultas del usuario

Ficheros lógicos internos	
Descripción	Complejidad
Preferencias de corredor/equipo favorito	Baja

Tabla 9 Ficheros Lógicos Internos

Ficheros lógicos externos	
Descripción	Complejidad
Cloud Firestore	Alta

Tabla 10 Ficheros lógicos externos

A continuación, debemos calcular los puntos de función sin ajustar (PFNA) en base a (tabla 11):

Dominio	Complejidad	Peso x Complejidad	Nº total de funciones	Total
Entradas	Baja	x3	6	18
	Media	x4	2	8
	Alta	x6	0	0
Salidas	Baja	x4	3	12
	Media	x5	0	0
	Alta	x7	0	0
Consultas	Baja	x3	12	36
	Media	x4	9	36

	Alta	x6	0	0
Ficheros lógicos internos	Baja	x7	1	7
	Media	x10	0	0
	Alta	x15	0	0
Ficheros lógicos externos	Baja	x5	0	0
	Media	x7	0	0
	Alta	x10	1	10
Puntos de Función Sin Ajustar (PFNA)				127

Tabla 11 Equivalencia de los Puntos de Función Sin Ajustar

Una vez calculados los puntos de función sin ajustar, tenemos que asignar el grado de complejidad a cada uno de los factores de complejidad que afectan a la funcionalidad del sistema (tabla 12). Cada uno de esos factores se evalúan del 0 al 5 en función de su incidencia:

- 0: sin influencia
- 1: incidental
- 2: moderado
- 3: Medio
- 4: Significativo
- 5: esencial

Factor	Valor
1. Respaldo y recuperación	1
2. Comunicaciones de datos	4
3. Funciones distribuidas	0
4. Desempeño Crítico	3
5. Entorno operativo existente	3
6. Entrada de datos en línea	1
7. Actualización archivos maestros	2
8. Complejo de valores de dominio de información	3
9. Complejidad interna	2
10. Código reutilizable	2
11. Conversión/instalación en el diseño	1
12. Instalaciones múltiples	1
13. Aplicación diseñada para cambio	5
14. Velocidad en las transacciones	4
TOTAL	32

Tabla 12 Factores de Complejidad y su valor

El factor de ajuste (FA) se obtiene a partir de la suma de los factores de complejidad anteriormente descritos gracias a la siguiente ecuación:

$$FA = (0.01 \times FC) + 0.65 = (0.01 \times 32) + 0.65 = 0.97$$

Para obtener **los puntos de función ajustados** bastaría con aplicar a los puntos de función no ajustados el factor de ajuste obtenido:

$$PF = PFNA \times FA = 127 * 0.97 = 123.19$$

Por último, debemos calcular la duración estimada del proyecto en base a los resultados obtenidos. La equivalencia en tiempo basándose en valores medios de la industria informática respecto a los puntos de función es: 1 mes (21 días laborables) es igual a 13 puntos de función.

Por lo tanto, la duración del proyecto estimada es igual a: **123.19/13 = 9.47 meses**

3.4.2 Estimación por COCOMO

Este modelo paramétrico de estimación consiste en obtener el tiempo y el esfuerzo empleados en el desarrollo del sistema a partir de las líneas de código obtenidas en la estimación por puntos de función.

Existen diferentes tipologías de este modelo, cada una de las cuales se aplicará en función de las características propias de cada proyecto.

- **COCOMO básico:** para estimaciones iniciales moderadamente precisas, al comienzo del proyecto cuando no se dispone de detalles. Calcula el esfuerzo del desarrollo del software en función del tamaño del programa (LDC). Se aplican sólo valores nominales.
- **COCOMO intermedio:** cuando se dispone de una especificación de requisitos más completa. Calcula el esfuerzo del desarrollo del programa en función del tamaño de este y de un conjunto de factores conductores del costo. Se calcula a partir del valor nominal aplicando a éste la influencia de 15 factores de coste.
- **COCOMO avanzado:** cuando se dispone de una especificación de requisitos completa. Utilizando como base el intermedio, lleva a cabo una evolución del impacto de los factores de costo a lo largo de las distintas fases del proyecto.

Por lo tanto, en base a los modelos definidos y teniendo en cuenta las características principales de nuestro proyecto, consideramos la opción más adecuada el modelo de COCOMO intermedio.

Una vez escogido el modelo de estimación, es necesario establecer las líneas de código requeridas para realizar un punto de función. Para ello empleamos una tabla de equivalencias estándar en función del lenguaje de programación utilizado (ilustración 2).

LOC por punto de función			
Lenguaje	LOC/FP	Lenguaje	LOC/FP
Ensamblador	320	Basic ANSI/Quick/Turbo	64
Macroensamblador	213	Java	53
C	150	Visual C++	34
Fortran	106	Foxpro 2,5	34
Cobol	106	Visual Basic	32
Pascal	91	Delphi	29
Cobol ANSI 85	91	C++	29
Basic	91	Visual Cobol	20
RPG	80	Clipper	19
PL/I	80	Power Builder	16
Ada	71	Hoja de Calculo	6

Ilustración 2 Equivalencia líneas de código-punto de función

En nuestro caso nos quedamos con la equivalencia del lenguaje JAVA, ya que es la base del lenguaje Android. Por lo tanto, las líneas de código de nuestro proyecto son:

$$LDC = PFA \times (\text{Líneas}/PF) = 123.19 \times 53 = 6529.07 \text{ líneas de código}$$

A continuación, como hemos elegido el modelo intermedio de COCOMO, necesitamos establecer la influencia en nuestro proyecto de los factores conductores del coste. Para ello vamos a utilizar como base una tabla con las correcciones del coste estandarizadas para este modelo de estimación (ilustración 3).

Factores conductores del coste (FCC):

Factores Conductores del Coste		Valor de los factores					
		Muy bajo	Bajo	Medio	Alto	Muy alto	Extra
Software	Fiabilidad del software requerido	0,75	0,88	1,00	1,15	1,4	
	Tamaño de la base de datos		0,94	1,00	1,08	1,16	
	Complejidad del software	0,70	0,85	1,00	1,15	1,30	1,65
Hardware	Restricciones de rendimiento en tiempo de ejecución			1,00	1,11	1,30	1,66
	Restricciones de memoria			1,00	1,06	1,21	1,56
	Volatilidad del entorno de la máquina virtual		0,87	1,00	1,15	1,30	
	Tiempo de respuesta requerido		0,87	1,00	1,07	1,15	
Personal	Capacidad de los analistas	1,46	1,19	1,00	0,86	0,71	
	Experiencia con el tipo de aplicación	1,29	1,13	1,00	0,91	0,82	
	Experiencia con el hardware	1,21	1,10	1,00	0,90		
	Experiencia con el lenguaje de programación	1,14	1,07	1,00	0,95		
	Capacidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Proyecto	Técnicas modernas de programación	1,24	1,10	1,00	0,91	0,82	
	Utilización de herramientas software	1,24	1,10	1,00	0,91	0,83	
	Restricciones en la planificación temporal del desarrollo	1,23	1,08	1,00	1,04	1,10	

Ilustración 3 Ponderación Factores Conductores del Coste genéricos

El factor aplicado en nuestro proyecto para cada uno de ellos es (tabla 13):

Fiabilidad del software	Alto	1.15
Tamaño de la base de datos	Medio	1
Complejidad del software	Medio	1
Restricciones en tiempo de ejecución	Medio	1
Restricciones de memoria	Medio	1
Volatilidad máquina virtual	Bajo	0.87
Tiempo de respuesta	Alto	1.08
Capacidad analistas	Medio	1
Experiencia con tipo aplicación	Bajo	1.13
Experiencia con el hardware	Medio	1
Experiencia con lenguaje programación	Medio	1
Capacidad programadores	Medio	1
Técnicas modernas programación	Medio	1
Utilización herramientas software	Muy Alto	0.83
Restricciones en la planificación temporal	Bajo	1.08
Resultado (FCC)		1.0945

Tabla 13 Factores Conductores del Coste del proyecto

En este punto observando el tamaño del proyecto (6529.07 LDC<50KLDC) y que nos encontramos en un entorno estable de desarrollo y con pocas presiones respecto al tiempo de entrega, escogemos el modo de desarrollo Orgánico para calcular el esfuerzo y el tiempo estimado para nuestro proyecto (ilustración 4).

Esfuerzo y tiempo del modelo COCOMO intermedio:

Modo de desarrollo	Persona-mes*	Tiempo desarrollo*
Orgánico ¹	$E = 3,2 \text{ KLDC}^{1,05}$	$TD=2,5 \text{ PM}^{0,38}$
Semiacoplado ²	$E = 3,0 \text{ KLDC}^{1,12}$	$TD=2,5 \text{ PM}^{0,35}$
Empotrado ³	$E = 2,8 \text{ KLDC}^{1,2}$	$TD=2,5 \text{ PM}^{0,32}$

Ilustración 4 Modos de desarrollo del Modelo COCOMO INTERMEDIO

- **Esfuerzo nominal** = $3.2 \text{ KLDC}^{1.05} = 3.2 \times 6.52907^{1.05} = 22.94$ personas – mes
- **Esfuerzo** = Esfuerzo nominal x FCC = $22.94 \times 1.0945 = 25.1078$ personas -mes
- **Tiempo desarrollo** = $2.5 \times \text{PM}^{0.38} = 2.5 \times 25.1078^{0.38} = 8.5088$ meses
- **Nº medio de personas** = $25.1078/8.5088 = 2.9508$ personas
- **Conclusión:** Se necesitan casi 3 personas a tiempo completo para llevar a cabo el desarrollo del proyecto en ese plazo. Pero como el proyecto será desarrollado por una sola persona la duración real asciende a 25,11 meses.

3.5 Presupuestos

En este apartado se detallarán los costes totales del proyecto en función de la estimación obtenida por los puntos de función (Método Albrecht), ya que en la comparativa con la estimación obtenida por el método COCOMO ofrece unos resultados más precisos y ajustados en cuanto a la duración final del proyecto (en este caso 9 meses y medio) de acuerdo con lo establecido en la “Planificación Inicial”.

Por otro lado, se calcularán también dichos costes según la planificación inicial realizada (198 días que equivalen a 6.6 meses aproximadamente), y en caso de no respetar los plazos establecidos se detallará el incremento/decremento de coste en el presupuesto.

Para ello se aglutinarán los costes relativos a los componentes hardware y software, junto con los costes asociados a los recursos humanos y a los costes añadidos por otras cuestiones.

3.5.1 Costes hardware y software

Para evaluar dichos costes se ha tenido en cuenta el porcentaje de uso de los componentes para el desarrollo del proyecto en relación con la vida útil media de cada uno de ellos, y así obtener el coste real asociado al proyecto.

Tabla 14 Costes Hardware Totales Estimación Albrecht

Componente Hardware	Precio (€)	Vida útil media (años)	Porcentaje de uso (%)	Coste real (€)
Ordenador Portátil 1	700	7	11.30%	79.1
Ordenador Portátil 2	550	7	11.30%	62.15
Dispositivo Móvil 1	490	3	26.39%	129.311
Ratón Inalámbrico	10	4	19.79%	1.98
TOTAL (€)				272.54

Tabla 15 Costes Hardware Totales Planificación Inicial

Componente Hardware	Precio (€)	Vida útil media	Porcentaje de uso	Coste real (€)
Ordenador Portátil 1	700	7	7.86%	55.02
Ordenador Portátil 2	550	7	7.86%	43.23
Dispositivo Móvil 1	490	3	18.33%	89.82
Ratón Inalámbrico	10	4	13.75%	1.38
TOTAL (€)				189.45

En el caso de los costes software consideramos que son nulos en ambos presupuestos ya que todos los componentes se obtuvieron de manera gratuita.

Tabla 16 Costes Software Totales

Componente software	Precio
Windows 10 Home	Gratuito
Word (Office 365)	Gratuito
Pycharm Community Edition	Gratuito
Android Studio	Gratuito
Google Drive	Gratuito
Github	Gratuito
Firebase	Gratuito
SporstRadar Api	Prueba Gratuita
TOTAL (€)	0

3.5.2 Costes Recursos humanos

En primer lugar, teniendo en cuenta que el proyecto se desarrolla por una sola persona que desempeña diferentes roles (ya especificados en el apartado de “Costes de Personal”) vamos a agrupar dichos roles en dos categorías más concretas:

- **Analista:** se encarga de realizar las tareas de gestión del equipo de trabajo, análisis de requisitos, diseño de interfaces y elaboración de la documentación.
- **Programador:** se encarga de realizar las tareas de implementación y pruebas del sistema.

Una vez definidos los roles, es necesario establecer cuanto tiempo se desempeñará cada uno de ellos durante el proyecto, ya que el coste que lleva asociado puede variar de manera notable. En este caso vamos a suponer que el rol de analista se desempeñará un 35% del tiempo, mientras que el 65% del tiempo restante se encargará de la labor del programador (siempre de una manera subjetiva y aproximada). Como mediante la estimación por puntos de función no sabemos exactamente las horas diarias que trabaja cada persona fijamos la media de 4 horas que hemos establecido en la planificación inicial.

En este punto, se definen los salarios base medios que pueden llegar a percibir un analista y un programador en cualquier empresa (según el salario base promedio consultado en el portal de empleo “indeed.com”) (tablas 17 y 18):

Tabla 17 Costes Totales RRHH Estimación Albrecht

Rol	Salario/mes (€)	Salario real (4h) (€)	Duración (mes)	Porcentaje Trabajo (%)	Coste total (€)
Analista-funcional	2250	1125	9.5	35%	3740.63
Programador	1967	983.5	9.5	65%	6073.11
TOTAL (€)					9813.74

Tabla 18 Costes Totales RRHH Planificación Inicial

Rol	Salario/mes (€)	Salario real (4h) (€)	Duración (mes)	Porcentaje Trabajo (%)	Coste total (€)
Analista-funcional	2250	1125	6.6	35%	2598.75
Programador	1967	983.5	6.6	65%	4219.22
TOTAL (€)					6817.97

3.5.3 Costes añadidos

Tabla 19 Costes Añadidos Estimación Albrecht

Descripción	Coste/mes (€)	Duración proyecto(mes)	Coste real (€)
Conexión a Internet (Wifi) + Tarifa soporte móvil	35	9.5	332.5
Gasto eléctrico	15 aproximadamente	9.5	142.5
TOTAL (€)			475

Tabla 20 Costes Totales añadidos Planificación Inicial

Descripción	Coste/mes (€)	Duración proyecto(mes)	Coste real (€)
Conexión a Internet (Wifi) + Tarifa soporte móvil	35	6.6	231
Gasto eléctrico	15 aproximadamente	6.6	99
TOTAL (€)			330€

A continuación, vamos a detallar como quedaría el presupuesto con los costes totales teniendo en consideración ambas duraciones de proyecto (tabla 21).

Tabla 21 Comparativa Costes Totales Proyecto entre Estimación y Planificación

Coste total del proyecto			
Modelo	Duración(mes)	Costes (€)	Coste Final (€)
Planificación Inicial	6.6 meses	Personal:6817.97 Hardware: 189.45 Software: 0 Añadidos: 330	7337.42
Estimación método Albrecht	9.5 meses	Personal: 9813.74 Hardware: 272.54 Software: 0 Añadidos: 475	10561.28

Por último, faltaría por especificar el presupuesto final en función de la posible variación de la fecha de entrega con respecto a la establecida en la planificación inicial.

- Añadir los costes generados por retraso en la fecha de entrega.
- Descontar los costes presupuestados por adelanto en la fecha de entrega.
- Mantener los costes por el cumplimiento de la fecha de entrega.

3.5.4 Presupuestos finales

En este punto se van a calcular los costes finales del proyecto en relación con la fecha de entrega de este. Como podemos observar la fecha real de finalización del proyecto se ha alargado hasta el 27 de abril de 2021, es decir 17 días más de lo previsto en la planificación inicial, por lo que se ha generado un incremento sobre los costes presupuestados.

El motivo principal de este retraso obedece a la falta de experiencia en la realización de proyectos de semejantes características, por lo que se cometieron errores en la planificación a la hora de establecer la cantidad de días necesarios para completar los distintos procesos (incluida la elaboración de la documentación). A esto habría que sumar los diferentes imprevistos surgidos en el desarrollo de determinadas funcionalidades, los cuales provocaron un incremento en las tareas de implementación y pruebas para solventarlos.

A continuación, se detallan los costes totales de cada una de las categorías diferenciadas (hardware, recursos humanos, software y costes añadidos) adaptados a la fecha real de finalización del proyecto (véase tablas 22, 23, 24 y 25).

Componente Hardware	Precio (€)	Vida útil media (años)	Porcentaje de uso (%)	Coste real (€)
Ordenador Portátil 1	700	7	8.54%	59.78
Ordenador Portátil 2	550	7	8.54%	46.97
Dispositivo Móvil 1	490	3	19.92%	97.61
Ratón Inalámbrico	10	4	14.94%	1.49
TOTAL (€)				205.85

Tabla 22 Costes Hardware Finales

Componente software	Precio
Windows 10 Home	Gratuito
Word (Office 365)	Gratuito
Pycharm Community Edition	Gratuito
Android Studio	Gratuito
Google Drive	Gratuito
Github	Gratuito
Firebase	Gratuito
SporstRadar Api	Prueba Gratuita
TOTAL (€)	0

Tabla 23 Costes Software Finales

Rol	Salario/mes (€)	Salario real (4h) (€)	Duración (mes)	Porcentaje Trabajo (%)	Coste total (€)
Analista-funcional	2250	1125	7.17	35%	2823.19
Programador	1967	983.5	7.17	65%	4583.60
TOTAL (€)					7406.79

Tabla 24 Costes Recursos Humanos Finales

Descripción	Coste/mes (€)	Duración proyecto(mes)	Coste real (€)
Conexión a Internet (Wifi) + Tarifa soporte móvil	35	7.17	250.95
Gasto eléctrico	15 aproximadamente	7.17	107.55
TOTAL (€)			358.5

Tabla 25 Costes Añadidos Finales

Por lo tanto, la suma total del presupuesto final ascendería a **7971.14 €**, que comparándolo con el presupuesto obtenido en la planificación inicial (**7337.42 €**) supone un incremento de **633.72 € (8.69% más)**. Dicho incremento se considera asumible en relación con los recursos de los que dispone el proyecto, por lo que no se producen daños significativos en la estructura del proyecto.

4. ANÁLISIS DEL SISTEMA

En este apartado se van a describir los distintos aspectos que se han considerado previamente a la realización del diseño e implementación del sistema. Esos aspectos van desde los diferentes agentes participantes en el sistema, hasta los requisitos de todo tipo que debe satisfacer el sistema, acompañados de las especificaciones y diagramas pertinentes para su representación.

4.1 Actores del sistema

Los actores representan a cualquier agente interno o externo que participan en el sistema. En nuestro caso hemos localizado los siguientes:

- **Usuario:** usuario que accede a la aplicación para disfrutar de todas sus características sin necesidad de identificarse en ningún momento. Representa el actor principal del sistema.
- **Cloud Firestore:** servicio en la nube donde se ubica toda la información mostrada al usuario en la aplicación. Se comunica con la aplicación a través de las peticiones de datos realizadas por el usuario. (Actor Externo)
- **Portal Web Noticias:** Portal web de noticias que suministra al usuario las noticias de la página web seleccionada. Se considera un actor externo, ya que simplemente la aplicación carga la página web de noticias sin necesidad de almacenar ningún dato.
- **Mecanismos para Compartir:** herramientas utilizadas para compartir con otros usuarios el enlace de descarga de la aplicación a través de la tienda oficial de Android (PlayStore), como por ejemplo “WhatsApp” o “Gmail”. (Actor Externo)
- **Navegador Internet:** herramienta externa que se comunica con el sistema para desplegar el sitio web del corredor accedido por el usuario, en el caso de que así lo solicite.
- **Plataforma Correo (JavaMail API):** herramienta interna al sistema que proporciona a los usuarios la posibilidad de realizar envíos de mensajes al buzón de correo del administrador del sistema.

4.2 Requisitos de Usuario

Los requisitos de usuario representan las diferentes características que el usuario pretende realizar con el sistema, y por lo tanto este deberá satisfacerlas. En nuestro proyecto van a estar organizados en función del actor que va a ser capaz de desempeñarlos para así facilitar su comprensión. En este sentido, el actor que va a realizar gran parte de las funcionalidades de nuestro sistema es el “**usuario**”, ya que el resto de los actores interactúan con el sistema de manera externa.

Dichos requisitos son los siguientes:

- **RU-01:** El usuario puede acceder a la pantalla principal de la aplicación.
- **RU-02:** El usuario puede visualizar la lista de Carreras que se disputan en un determinado mes y año del Calendario.
- **RU-03:** El usuario puede seleccionar y visualizar los datos de una Carrera.
- **RU-04:** El usuario puede visualizar los resultados de los corredores y equipos de una carrera.
- **RU-05:** El usuario puede consultar si una carrera está formada por diversas etapas.
- **RU-06:** El usuario puede seleccionar una determinada etapa y visualizar sus datos.
- **RU-07:** El usuario puede visualizar los resultados y equipos inscritos en una etapa.
- **RU-08:** El usuario puede consultar (en el caso de que se trate de una carrera por etapas) si una etapa se encuentra disputándose en la fecha actual.
- **RU-09:** El usuario puede consultar la clasificación mundial de los corredores en el momento actual.
- **RU-10:** El usuario puede seleccionar y visualizar la lista de equipos que forman parte de una temporada.
- **RU-11:** El usuario puede seleccionar un determinado equipo y visualizar sus datos.
- **RU-12:** El usuario puede consultar la plantilla de corredores que forman parte de un determinado equipo.
- **RU-13:** El usuario puede seleccionar y escoger un determinado equipo para almacenarlo como favorito.
- **RU-14:** El usuario puede seleccionar el nombre de un corredor y visualizar sus datos.
- **RU-15:** El usuario puede solicitar acceder al sitio web del corredor seleccionado.
- **RU-16:** El usuario puede seleccionar y escoger un determinado corredor para almacenarlo como favorito.
- **RU-17:** El usuario puede consultar el equipo y el corredor almacenado como favorito antes de actualizar los cambios en los favoritos.
- **RU-18:** El usuario puede seleccionar y acceder a los datos de los elementos guardados como favorito.
- **RU-19:** El usuario puede buscar cualquier corredor filtrando por su nombre.
- **RU-20:** El usuario puede seleccionar cualquier corredor filtrado y acceder a sus datos.
- **RU-21:** El usuario puede consultar los sitios web de noticias disponibles.
- **RU-22:** El usuario puede buscar cualquier carrera celebrada en las temporadas 2020 o 2021 filtrando por su nombre.

- **RU-23:** El usuario puede seleccionar cualquier carrera filtrada y acceder a sus datos.
- **RU-24:** El usuario puede consultar las vías para poder compartir el enlace de descarga de la aplicación.
- **RU-25:** El usuario puede enviar opiniones, dudas, o sugerencias al respecto de la aplicación al encargado de esta.
- **RU-26:** El usuario puede solicitar una respuesta de manera externa a la aplicación a través de correo electrónico.
- **RU-27:** El usuario puede acceder a la sección de “Mis Favoritos” desde el icono “Favoritos” situado en la barra de navegación de la parte inferior.
- **RU-28:** El usuario puede guardar el corredor o el equipo seleccionado como favorito en las preferencias de la aplicación.
- **RU-29:** El usuario puede acceder a cualquiera de los sitios web de noticias disponibles y leer las últimas novedades.
- **RU-30:** El usuario puede compartir el enlace de descarga de la aplicación a través de la aplicación que seleccione.

4.2.1 Modelo de Casos de Uso

En este apartado se van a incluir los diferentes diagramas de casos de uso que representan las acciones que serán capaces de realizar los diferentes tipos de actores, es decir se ilustran los requisitos de usuario anteriormente mencionados y la manera en la que interactúan los actores con los mismos. En este tipo de diagrama intervienen conceptos como:

- **Actor:** entidad externa al sistema que se modela y puede interactuar con él.
- **Requisito de usuario:** servicio que se desea que haga el sistema.

Además, las relaciones entre casos de uso y actores pueden ser las siguientes:

- Un actor se comunica con uno o varios requisitos
- Un requisito se comunica con uno o varios requisitos.
- Un requisito extiende otro requisito
- Un requisito incluye otro requisito

Por lo tanto, los diagramas relativos a nuestro proyecto quedarían de la siguiente manera (ilustraciones 5 y 6):

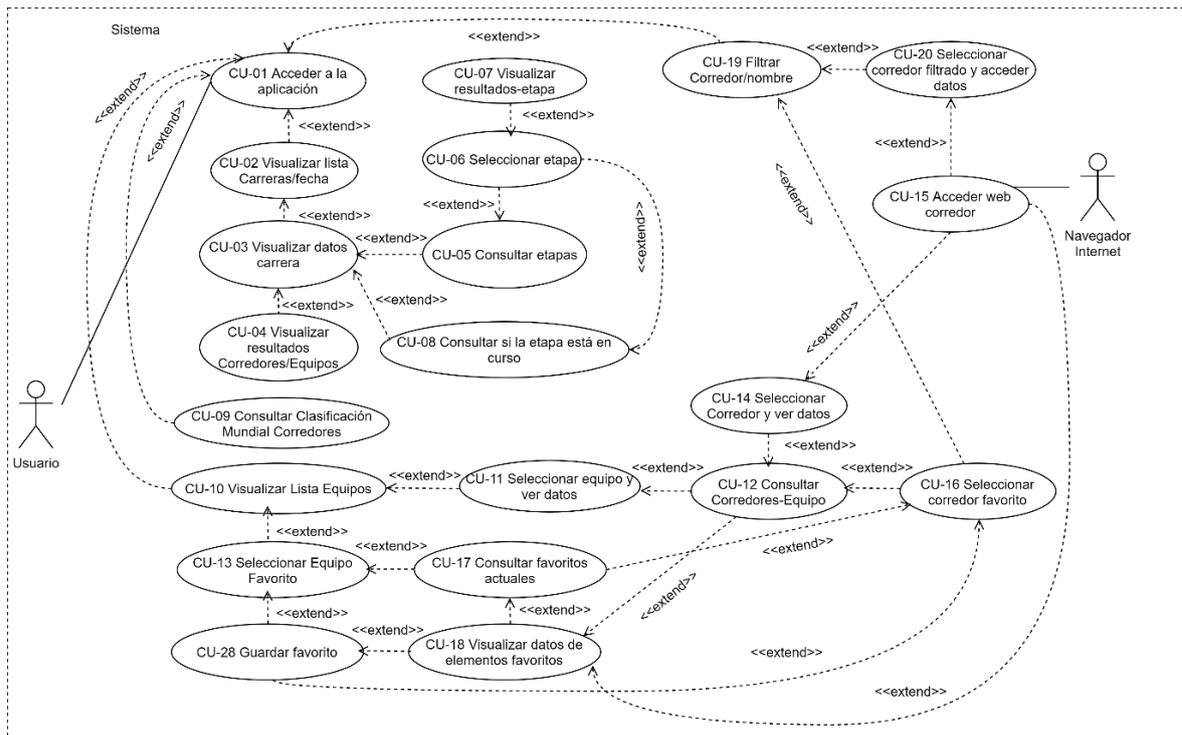


Ilustración 5 Diagrama de Casos de uso 1

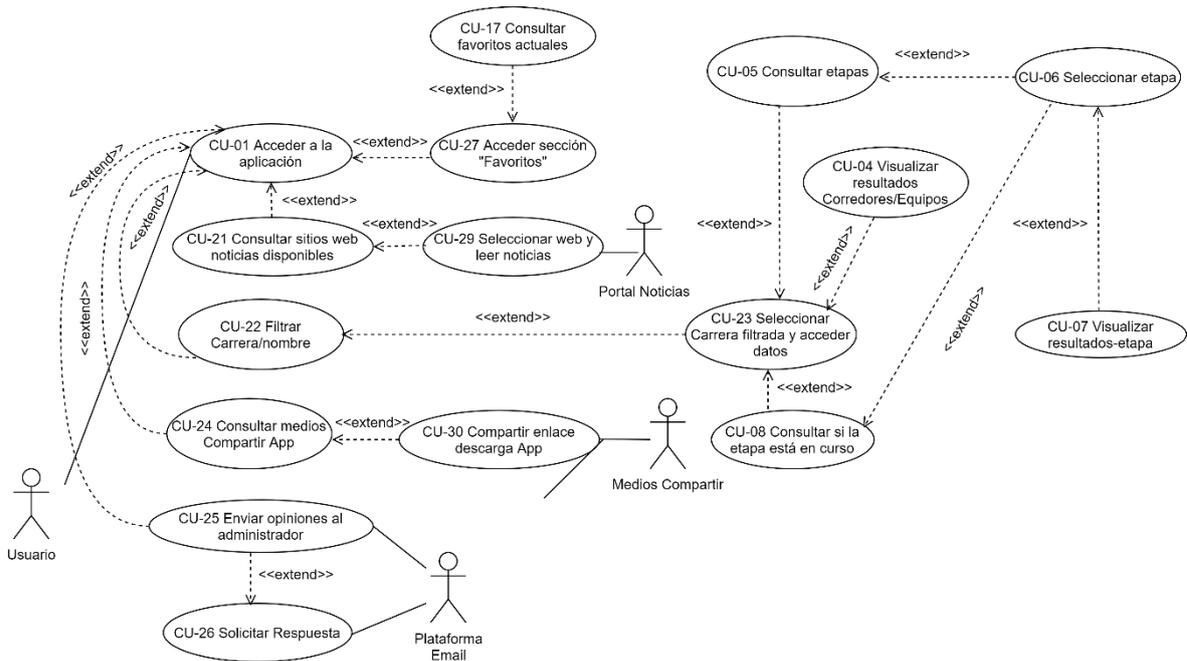


Ilustración 6 Diagrama de Casos de uso 2

Para facilitar la comprensión y visualización del diagrama de casos de uso se ha optado por dividir la representación en dos diagramas. Por esta razón, algunos de los casos aparecen duplicados ya que se encuentran en los dos diagramas.

4.2.2 Especificación de casos de uso

Una vez definidos y representados los requisitos de usuario, es necesario describir y analizar de manera más exhaustiva la totalidad de los requisitos. Dicho análisis se realiza a través de las tablas de especificación que se presentan a continuación (tablas 26 hasta la 55):

Identificador	CU-01 Acceder a la pantalla principal de la aplicación
Descripción	Caso de uso correspondiente al RU-01
Actores relacionados	Usuario
Casos de Uso relacionados	CU-02, CU-09, CU-10, CU-19, CU-21, CU-22, CU-24, CU-25, CU-27
Precondiciones	El actor se encontrará en la pantalla de bienvenida de la aplicación.
Flujo Principal	<ol style="list-style-type: none"> 1. El usuario selecciona el botón de acceder presente en la pantalla. 2. El sistema redirige al usuario a la página de inicio de la aplicación. 3. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	1.a Si el usuario ejecuta la acción "retroceder" del dispositivo el caso de uso finaliza de manera errónea.
Postcondiciones	La pantalla inicial de la aplicación se muestra satisfactoriamente.
Importancia	Muy Alta

Tabla 26 Caso de uso "Acceder a la pantalla principal de la aplicación"

Identificador	CU-02 Visualizar lista carreras fecha seleccionada
Descripción	Caso de uso correspondiente al RU-02
Actores relacionados	Usuario, Cloud Firestore
Casos de Uso relacionados	CU-03
Precondiciones	1. El usuario habrá realizado el CU-01.
Flujo Principal	<ol style="list-style-type: none"> 1. El usuario escoge y confirma una fecha en el Calendario situado en la parte superior. 2. El sistema actualiza la página con la fecha seleccionada por el usuario. 3. El usuario selecciona el botón "Mostrar Eventos-Temporada 2020 o 2021". 4. El sistema recupera y muestra las carreras que se disputan en el mes/año correspondiente a la fecha seleccionada.

	5. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	1.a. Si el usuario no confirma la fecha y ejecuta la acción “retroceder” en el dispositivo el sistema se mantiene en la vista actual. 1.a.1 El caso de uso continúa en el punto 3 del flujo principal. 3.b Si el usuario no selecciona el botón “Mostrar Eventos” el caso de uso finaliza de manera errónea.
Postcondiciones	El listado de carreras se muestra satisfactoriamente.
Importancia	Alta

Tabla 27 Caso de uso "Visualizar Lista Carreras"

Identificador	CU-03 Visualizar datos Carrera
Descripción	Caso de uso correspondiente al RU-03
Actores relacionados	Usuario, Cloud Firestore
Casos de Uso relacionados	CU-04, CU-05-CU-08
Precondiciones	1. El usuario habrá realizado el CU-02.
Flujo Principal	1. El usuario selecciona una determinada carrera de las mostradas en el listado para acceder a sus datos. 2. El sistema redirige al usuario a una nueva página denominada “Detalles Carrera” para consultar sus datos. 3. El sistema recupera y muestra los datos relativos a la carrera seleccionada. 4. El sistema habilitará la posibilidad de realizar el CU-04. 5. El sistema habilitará la posibilidad de realizar el CU-05. 6. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	4.a Si el usuario selecciona el botón “Equipos” o “Resultados” el sistema realiza el CU-04. 4.a.1 El caso de uso continúa en el punto 5 del flujo principal. 5.b Si el usuario selecciona el botón “Detalles” el sistema realiza el CU-05.
Postcondiciones	Los datos de la Carrera consultada se muestran satisfactoriamente.
Importancia	Alta

Tabla 28 Caso de uso " Visualizar datos Carrera"

Identificador	CU-04 Visualizar resultados Corredores y/o equipos
Descripción	Caso de uso correspondiente al RU-04
Actores relacionados	Usuario, Cloud Firestore
Casos de Uso relacionados	CU-03
Precondiciones	1. El usuario habrá realizado el CU-03.
Flujo Principal	1. El sistema recupera y muestra la clasificación correspondiente en la carrera consultada. 2. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	1.a Si la clasificación de Corredores o Equipos no se encuentra disponible el sistema informa al usuario mediante un mensaje por pantalla. 1.a.1 El sistema se mantiene en la página actual de "Detalles de Carrera".
Postcondiciones	El sistema se mantiene en la página actual de "Detalles de Carrera" mostrando el listado de la clasificación seleccionada.
Importancia	Alta

Tabla 29 Caso de uso "Visualizar resultados Corredores y/o Equipos"

Identificador	CU-05 Consultar etapas
Descripción	Caso de uso correspondiente al RU-05
Actores relacionados	Usuario, Cloud Firestore
Casos de Uso relacionados	CU-06
Precondiciones	1. El usuario habrá realizado el CU-03 o CU-22. 2. La Carrera consultada deberá ser del tipo "Vuelta por etapas".
Flujo Principal	1. El sistema recupera y muestra el listado con las etapas que forman parte de la carrera consultada. 2. El sistema habilitará la posibilidad de realizar el CU-06 . 3. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	2.a Si el usuario pulsa sobre alguna de las etapas mostradas en el listado el sistema realiza el CU-06.
Postcondiciones	El listado con las etapas pertenecientes a la carrera consultada se muestra satisfactoriamente.
Importancia	Alta

Tabla 30 Caso de uso "Consultar etapas"

Identificador	CU-06 Seleccionar etapa
Descripción	Caso de uso correspondiente al RU-06
Actores relacionados	Usuario, Cloud Firestore
Casos de Uso relacionados	CU-07
Precondiciones	1. El usuario habrá realizado el CU-05.
Flujo Principal	1. El sistema recupera y muestra los datos pertenecientes a la etapa consultada. 2. El sistema habilitará la posibilidad de realizar el CU-07 . 3. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	2.a Si el usuario selecciona el botón "Equipos" o "Resultados" el sistema realiza el CU-07.
Postcondiciones	Los datos de la etapa se muestran satisfactoriamente.
Importancia	Alta

Tabla 31 Caso de uso "Seleccionar Etapa"

Identificador	CU-07 Visualizar resultados (Corredores y/o equipos) etapa
Descripción	Caso de uso correspondiente al RU-07
Actores relacionados	Usuario, Cloud Firestore
Casos de Uso relacionados	
Precondiciones	1. El usuario habrá realizado el CU-06.
Flujo Principal	1. El sistema recupera y muestra la clasificación correspondiente en la etapa consultada. 2. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	1.a Si la clasificación de Corredores o Equipos no se encuentra disponible el sistema informa al usuario mediante un mensaje por pantalla. 1.a.1 El sistema se mantiene en la página actual de "Detalles de Etapa".
Postcondiciones	El sistema se mantiene en la página actual de "Detalles de Etapa" mostrando el listado de la clasificación seleccionada.

Tabla 32 Caso de uso "Visualizar resultados etapa"

Identificador	CU-08 Consultar si la etapa está en curso.
Descripción	Caso de uso correspondiente al RU-08
Actores relacionados	Usuario, Cloud Firestore
Casos de Uso relacionados	CU-06
Precondiciones	<ol style="list-style-type: none"> 1. El usuario habrá realizado el CU-03 o CU-23. 2. La fecha seleccionada en el Calendario de la página principal se corresponderá con la fecha actual.
Flujo Principal	<ol style="list-style-type: none"> 1. El sistema recupera si existe en la carrera consultada alguna etapa que se esté celebrando en la fecha actual y habilita el botón "Etapa en Curso". 2. El usuario selecciona el botón "Etapa en Curso" presente en la página actual. 3. El sistema muestra la etapa perteneciente a la carrera consultada que se está disputando en la fecha actual. 4. El sistema habilitará la posibilidad de realizar el CU-06. 5. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	<ol style="list-style-type: none"> 3.a Si en la Carrera consultada no existe ninguna etapa que se esté celebrando en la fecha actual el sistema no habilita el botón "Etapa en Curso". 3.a.1 El sistema se mantiene en la página actual de "Detalles de Etapa". 4.b Si el usuario pulsa sobre la etapa mostrada el sistema realiza el CU-06.
Postcondiciones	El sistema se mantiene en la página actual de "Detalles de Etapa" con el listado de la etapa en curso.
Importancia	Media

Tabla 33 Caso de uso "Consultar si la etapa está en curso"

Identificador	CU-09 Consultar Clasificación Mundial Corredores
Descripción	Caso de uso correspondiente al RU-09
Actores relacionados	Usuario, Cloud Firestore
Casos de Uso relacionados	
Precondiciones	1. El usuario habrá realizado el CU-01.
Flujo Principal	1. El usuario selecciona el botón "Clasificación" situado en la barra de

	navegación inferior presente en la página principal de la aplicación. 2. El sistema recupera y muestra el listado con la Clasificación Mundial de los Corredores en el momento actual junto a sus respectivos datos. 3. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	
Postcondiciones	La Clasificación Mundial de los Corredores se muestra satisfactoriamente.
Importancia	Alta

Tabla 34 Caso de uso "Consultar Clasificación Mundial Corredores"

Identificador	CU-10 Visualizar Lista Equipos
Descripción	Caso de uso correspondiente al RU-10
Actores relacionados	Usuario, Cloud Firestore
Casos de Uso relacionados	CU-11, CU-13
Precondiciones	1. El usuario habrá realizado el CU-01.
Flujo Principal	1. El usuario despliega el menú lateral de la aplicación a través del botón situado en la esquina superior izquierda y accede a la sección "Equipos". 2. El sistema habilita la posibilidad de elegir por pantalla la temporada sobre la que se desea consultar la lista de equipos. 3. El usuario escoge la temporada sobre la que desea consultar. 4. El sistema recupera y muestra el listado con los equipos inscritos en la temporada escogida por el usuario. 5. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	
Postcondiciones	El Listado con los Equipos se muestra satisfactoriamente.
Importancia	Alta

Tabla 35 Caso de uso "Visualizar Lista Equipos"

Identificador	CU-11 Seleccionar Equipo y ver datos
Descripción	Caso de uso correspondiente al RU-11
Actores relacionados	Usuario, Cloud Firestore
Casos de Uso relacionados	CU-12
Precondiciones	1. El usuario habrá realizado el CU-10.
Flujo Principal	<ol style="list-style-type: none"> 1. El usuario selecciona un determinado equipo de los presentes en pantalla para acceder a sus datos. 2. El sistema redirige al usuario a nueva página denominada "Detalles Equipo". 3. El sistema recupera y muestra por pantalla los datos pertenecientes al equipo consultado. 4. El sistema habilita la posibilidad de realizar el CU-12. 5. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	4.a Si el usuario selecciona el botón "Plantilla de Corredores" el sistema realiza el CU-12.
Postcondiciones	Los datos del equipo seleccionado se muestran satisfactoriamente.
Importancia	Media

Tabla 36 Caso de uso "Seleccionar Equipo y ver sus datos"

Identificador	CU-12 Consultar Corredores de Equipo
Descripción	Caso de uso correspondiente al RU-12
Actores relacionados	Usuario, Cloud Firestore
Casos de Uso relacionados	CU-14, CU-16
Precondiciones	1. El usuario habrá realizado el CU-11 o CU-18.
Flujo Principal	<ol style="list-style-type: none"> 1. El sistema recupera y muestra por pantalla el listado con los corredores que forman parte del equipo consultado. 2. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	
Postcondiciones	El listado de Corredores del Equipo se muestra satisfactoriamente.
Importancia	Media

Tabla 37 Caso de uso "Consultar Corredores de equipo"

Identificador	CU-13 Seleccionar Equipo Favorito
Descripción	Caso de uso correspondiente al RU-13
Actores relacionados	Usuario
Casos de Uso relacionados	CU-17, CU-28
Precondiciones	1. El actor habrá realizado el CU-10.
Flujo Principal	<p>1. El usuario selecciona el botón con el icono de un corazón asociado al equipo que desee de los mostrados en el listado de Equipos.</p> <p>2. El sistema redirige al usuario a una nueva página denominada “Mis Favoritos.”</p> <p>3. El sistema habilita la posibilidad de realizar el CU-17.</p> <p>4. El sistema habilita la posibilidad de realizar el CU-28.</p> <p>5. El caso de uso finaliza satisfactoriamente.</p>
Flujo Alternativo	<p>3.a Si el usuario selecciona el botón “Consultar” el sistema realiza el CU-17.</p> <p>3.a.1 El caso de uso continua en el punto 4 del flujo principal.</p> <p>4.b Si el usuario selecciona el botón “Guardar” el sistema realiza el CU-28.</p>
Postcondiciones	El sistema se mantiene en la página “Mis Favoritos”.
Importancia	Media

Tabla 38 Caso de uso "Seleccionar Equipo favorito"

Identificador	CU-14 Seleccionar Corredor y ver datos
Descripción	Caso de uso correspondiente al RU-14
Actores relacionados	Usuario, Cloud Firestore
Casos de Uso relacionados	CU-15
Precondiciones	1. El usuario habrá realizado el CU-12.
Flujo Principal	<p>1. El usuario selecciona el corredor que desea consultar de los mostrados en el listado del equipo.</p> <p>2. El sistema redirige al usuario a una nueva página denominada “Detalles Corredor.”</p> <p>3. El sistema recupera y muestra por pantalla los datos del corredor consultado.</p> <p>4. El sistema habilita la posibilidad de realizar el CU-15.</p>

	5. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	4.a Si el usuario selecciona el botón "Consultar sitio web" el sistema realiza el CU-15.
Postcondiciones	El sistema se mantiene en la página "Detalles Corredor".
Importancia	Media

Tabla 39 Caso de uso "Seleccionar Corredor y ver sus datos"

Identificador	CU-15 Acceder sitio web corredor
Descripción	Caso de uso correspondiente al RU-15
Actores relacionados	Usuario, Navegador Internet
Casos de Uso relacionados	
Precondiciones	1. El actor habrá realizado el CU-14, CU-20 o CU-18.
Flujo Principal	1. El sistema despliega el navegador del dispositivo electrónico y muestra al usuario el sitio web del corredor consultado. 2. El usuario puede navegar por cualquier sección que desee del sitio web. 3. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	
Postcondiciones	1. El sitio web se muestra satisfactoriamente en el navegador del dispositivo del usuario. 2. La aplicación web se mantiene activa.
Importancia	Media

Tabla 40 Caso de uso "Acceder sitio web corredor"

Identificador	CU-16 Seleccionar Corredor Favorito
Descripción	Caso de uso correspondiente al RU-16
Actores relacionados	Usuario
Casos de Uso relacionados	CU-17, CU-28
Precondiciones	1. El actor habrá realizado el CU-12 o CU-19.
Flujo Principal	1. El usuario selecciona el botón con el icono de un corazón asociado al corredor que desee de los mostrados en el listado de Corredores. 2. El sistema redirige al usuario a una nueva página denominada "Mis Favoritos".

	<p>3. El sistema habilita la posibilidad de realizar el CU-17.</p> <p>4. El sistema habilita la posibilidad de realizar el CU-28.</p> <p>5. El caso de uso finaliza satisfactoriamente.</p>
Flujo Alternativo	<p>3.a Si el usuario selecciona el botón "Consultar" el sistema realiza el CU-17.</p> <p>3.a.1 El caso de uso continua en el punto 4 del flujo principal.</p> <p>4.b Si el usuario selecciona el botón "Guardar" el sistema realiza el CU-28.</p>
Postcondiciones	El sistema se mantiene en la página "Mis Favoritos".
Importancia	Media

Tabla 41 Caso de uso "Seleccionar Corredor favorito"

Identificador	CU-17 Consultar favoritos actuales
Descripción	Caso de uso correspondiente al RU-17
Actores relacionados	Usuario
Casos de Uso relacionados	CU-18
Precondiciones	1. El actor habrá realizado el CU-13 o CU-16.
Flujo Principal	<p>1. El sistema carga el archivo de preferencias de la aplicación y muestra los favoritos actuales guardados (corredor y/o equipo).</p> <p>2. El sistema habilita la posibilidad de realizar el CU-18.</p> <p>3. El caso de uso finaliza satisfactoriamente.</p>
Flujo Alternativo	2.a Si el usuario pulsa sobre cualquiera de los dos elementos favoritos presentes en pantalla el sistema realiza el CU-18.
Postcondiciones	Los favoritos guardados se muestran satisfactoriamente y el sistema se mantiene en la página "Mis Favoritos".
Importancia	Media

Tabla 42 Caso de uso "Consultar favoritos actuales"

Identificador	CU-18 Acceder datos de elementos favoritos
Descripción	Caso de uso correspondiente al RU-18
Actores relacionados	Usuario, Cloud Firestore
Casos de Uso relacionados	CU-12, CU-15
Precondiciones	1. El usuario habrá realizado el CU-17
Flujo Principal	<p>1. El usuario pulsa sobre alguno de los elementos mostrados como favoritos (equipo o ciclista).</p> <p>2. El sistema redirige al usuario a la página “Detalles Equipo” o “Detalles Corredor” en función del elemento seleccionado.</p> <p>3. El caso de uso finaliza satisfactoriamente.</p>
Flujo Alternativo	<p>2.a Si el usuario se encuentra en la página “Detalles Equipo” el sistema habilita la posibilidad de realizar el CU-12.</p> <p>2.a.1 Si el usuario selecciona el botón “Plantilla de Corredores” el sistema realiza el CU-12.</p> <p>2.b Si el usuario se encuentra en la página “Detalles Corredor” el sistema habilita la posibilidad de realizar el CU-15.</p> <p>2.b.1 Si el usuario selecciona el botón “Consultar Sitio web” el sistema realiza el CU-15.</p>
Postcondiciones	El usuario es redirigido a la página correspondiente de manera satisfactoria.
Importancia	Media

Tabla 43 Caso de uso "Acceder datos de elementos favoritos"

Identificador	CU-19 Filtrar Corredor por nombre
Descripción	Caso de uso correspondiente al RU-19
Actores relacionados	Usuario, Cloud Firestore
Casos de Uso relacionados	CU-16, CU-20
Precondiciones	1. El usuario habrá realizado el CU-01
Flujo Principal	<p>1. El usuario despliega el menú lateral de la aplicación a través del botón situado en la esquina superior izquierda y accede a la sección “Corredores”.</p> <p>2. El sistema habilita un cuadro de texto para que el usuario pueda escribir el</p>

	<p>texto por el que desea realizar la búsqueda de corredores.</p> <p>3. El sistema recupera y muestra los corredores que cumplan coincidencias con el texto introducido por el usuario.</p> <p>4. El caso de uso finaliza satisfactoriamente.</p>
Flujo Alternativo	
Postcondiciones	El listado con los Corredores que cumplen las condiciones de filtrado se muestra satisfactoriamente.
Importancia	Alta

Tabla 44 Caso de uso "Filtrar Corredor por nombre"

Identificador	CU-20 Seleccionar Corredor filtrado y acceder datos
Descripción	Caso de uso correspondiente al RU-20
Actores relacionados	Usuario, Cloud Firestore
Casos de Uso relacionados	CU-16, CU-20
Precondiciones	1. El usuario habrá realizado el CU-19
Flujo Principal	<p>1. El usuario selecciona algún corredor de los mostrados por pantalla.</p> <p>2. El sistema redirige al usuario a la página "Detalles Corredor"</p> <p>3. El sistema recupera y muestra los datos correspondientes al corredor consultado.</p> <p>4. El sistema habilita la posibilidad de realizar el CU-15.</p> <p>5. El caso de uso finaliza satisfactoriamente.</p>
Flujo Alternativo	4.a Si el usuario selecciona el botón "Consultar sitio web" el sistema realiza el CU-15.
Postcondiciones	Los datos del corredor se muestran satisfactoriamente.
Importancia	Media

Tabla 45 Caso de uso "Seleccionar Corredor filtrado y ver sus datos"

Identificador	CU-21 Consultar sitios web noticias disponibles
Descripción	Caso de uso correspondiente al RU-21
Actores relacionados	Usuario
Casos de Uso relacionados	CU-29
Precondiciones	1. El usuario habrá realizado el CU-01
Flujo Principal	<ol style="list-style-type: none"> 1. El usuario despliega el menú lateral de la aplicación a través del botón situado en la esquina superior izquierda y accede a la sección "Noticias". 2. El sistema muestra por pantalla los iconos de los sitios web a los que puede acceder el usuario. 3. El sistema habilita la posibilidad de realizar el CU-29. 4. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	4.a Si el usuario pulsa sobre cualquiera de los iconos asociados a cada sitio web el sistema realiza el CU-29.
Postcondiciones	Los sitios web disponibles se muestran por pantalla satisfactoriamente.
Importancia	Media

Tabla 46 Caso de uso "Consultar sitios web de noticias disponibles"

Identificador	CU-22 Filtrar Carrera por nombre
Descripción	Caso de uso correspondiente al RU-22
Actores relacionados	Usuario, Cloud Firestore
Casos de Uso relacionados	CU-23
Precondiciones	1. El usuario habrá realizado el CU-01
Flujo Principal	<ol style="list-style-type: none"> 1. El usuario despliega el menú lateral de la aplicación a través del botón situado en la esquina superior izquierda y accede a la sección "Carreras". 2. El sistema habilita un cuadro de texto para que el usuario pueda escribir el texto por el que desea realizar la búsqueda de carreras. 3. El sistema recupera y muestra las carreras que cumplan coincidencias con el texto introducido por el usuario. 4. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	

Postcondiciones	El listado con los Carreras que cumplen las condiciones de filtrado se muestra satisfactoriamente.
Importancia	Alta

Tabla 47 Caso de uso "Filtrar Carrera por nombre"

Identificador	CU-23 Seleccionar Carrera y acceder datos
Descripción	Caso de uso correspondiente al RU-23
Actores relacionados	Usuario, Cloud Firestore
Casos de Uso relacionados	CU-04, CU-05, CU-08
Precondiciones	1. El usuario habrá realizado el CU-22.
Flujo Principal	<ol style="list-style-type: none"> 1. El usuario selecciona alguna de las carreras mostradas por pantalla. 2. El sistema redirige al usuario a la página "Detalles Carrera". 3. El sistema recupera y muestra los datos correspondientes a la carrera consultada. 4. El sistema habilitará la posibilidad de realizar el CU-04. 5. El sistema habilitará la posibilidad de realizar el CU-05. 6. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	<ol style="list-style-type: none"> 4.a Si el usuario selecciona el botón "Equipos" o "Resultados" el sistema realiza el CU-04. 4.a.1 El caso de uso continúa en el punto 5 del flujo principal. 5.b Si el usuario selecciona el botón "Detalles" el sistema realiza el CU-05.
Postcondiciones	Los datos de la carrera consultada se muestran satisfactoriamente.
Importancia	Media

Tabla 48 Caso de uso "Seleccionar Carrera y ver sus datos"

Identificador	CU-24 Consultar medios para compartir App
Descripción	Caso de uso correspondiente al RU-24
Actores relacionados	Usuario
Casos de Uso relacionados	CU-30
Precondiciones	1. El actor habrá realizado el CU-01
Flujo Principal	1. El usuario despliega el menú lateral de la aplicación a través del botón situado

	<p>en la esquina superior izquierda y accede a la sección “Compartir”.</p> <p>2. El sistema despliega por pantalla un cuadro de diálogo con los iconos de las aplicaciones disponibles para compartir la aplicación.</p> <p>3. El sistema habilita la posibilidad de realizar el CU-30.</p> <p>4. El caso de uso finaliza satisfactoriamente.</p>
Flujo Alternativo	3.a Si el usuario pulsa sobre cualquiera de los iconos asociados a cada aplicación el sistema realiza el CU-30.
Postcondiciones	Los medios para compartir la aplicación se muestran satisfactoriamente.
Importancia	Media

Tabla 49 Caso de uso "Consultar medios para compartir App"

Identificador	CU-25 Enviar opiniones, dudas
Descripción	Caso de uso correspondiente al RU-25
Actores relacionados	Usuario, Plataforma Email
Casos de Uso relacionados	CU-26
Precondiciones	1. El actor habrá realizado el CU-01
Flujo Principal	<p>1. El usuario despliega el menú lateral de la aplicación a través del botón situado en la esquina superior izquierda y accede a la sección “Contacto”.</p> <p>2. El sistema redirige al usuario a una nueva página denominada “envíanos tus comentarios”.</p> <p>3. El sistema muestra por pantalla los campos a rellenar por el usuario para enviar el comentario.</p> <p>4. El usuario completa los campos mínimos requeridos.</p> <p>5. El sistema habilita la posibilidad de realizar el CU-26 (Solicitar respuesta).</p> <p>6. El usuario pulsa sobre el botón “Enviar” presente en la pantalla para enviar el mensaje.</p> <p>7. El sistema se comunica con la “Plataforma Email” para establecer las condiciones del envío e incorporar el mensaje.</p> <p>8. El caso de uso finaliza satisfactoriamente.</p>

Flujo Alternativo	<p>5.a Si el usuario marca el campo para recibir un email de respuesta, el sistema realiza el CU-26.</p> <p>5.a.1 Si el usuario no marca el campo para recibir un email de respuesta, el sistema retorna al paso 6 del flujo principal.</p>
Postcondiciones	<ol style="list-style-type: none"> 1. El comentario se ha enviado satisfactoriamente al administrador del sistema. 2. El sistema muestra por pantalla un cuadro informativo con la confirmación del envío del mensaje.
Importancia	Media

Tabla 50 Caso de uso "Enviar comentarios"

Identificador	CU-26 Solicitar email de respuesta
Descripción	Caso de uso correspondiente al RU-26
Actores relacionados	Usuario, Plataforma Email
Casos de Uso relacionados	CU-26
Precondiciones	1. El actor habrá realizado el CU-25
Flujo Principal	<ol style="list-style-type: none"> 1. El sistema muestra por pantalla un nuevo campo para que el usuario introduzca la dirección de correo en la que desea recibir una respuesta. 2. El usuario introduce la información correspondiente y pulsa el botón "Enviar". 3. El sistema se comunica con la "Plataforma Email" para establecer las condiciones del envío e incorporar el mensaje. 4. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	
Postcondiciones	<ol style="list-style-type: none"> 1. El comentario con la información completa se ha enviado satisfactoriamente al administrador del sistema. 2. El sistema muestra por pantalla un cuadro informativo con la confirmación del envío del mensaje.
Importancia	Media

Tabla 51 Caso de uso "Solicitar email de respuesta"

Identificador	CU-27 Acceder sección Favoritos
Descripción	Caso de uso correspondiente al RU-27
Actores relacionados	Usuario
Casos de Uso relacionados	CU-17
Precondiciones	1. El actor habrá realizado el CU-01.
Flujo Principal	<ol style="list-style-type: none"> 1. El sistema usuario selecciona el icono "Favoritos" situado en la parte inferior de la pantalla. 2. El sistema redirige al usuario a la sección "Mis Favoritos". 3. El sistema habilita la posibilidad de realizar el CU-17. 4. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	3.a Si el usuario selecciona el botón "Consultar" el sistema realiza el CU-17.
Postcondiciones	La página "Mis Favoritos" se muestra satisfactoriamente".
Importancia	Media

Tabla 52 Caso de uso "Acceder sección Favoritos"

Identificador	CU-28 Guardar Favorito
Descripción	Caso de uso correspondiente al RU-28
Actores relacionados	Usuario
Casos de Uso relacionados	CU-18.
Precondiciones	1. El actor habrá realizado el CU-13 o CU-16.
Flujo Principal	<ol style="list-style-type: none"> 1. El sistema actualiza el archivo de preferencias de la aplicación con el nuevo elemento favorito añadido y lo muestra por pantalla. 2. El sistema muestra por pantalla un mensaje confirmación de que el elemento ha sido guardado como favorito correctamente. 3. El sistema habilita la posibilidad de realizar el CU-18. 4. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	3.a. Si el usuario pulsa sobre cualquiera de los dos elementos favoritos presentes en pantalla el sistema realiza el CU-18.
Postcondiciones	El sistema se mantiene en la página "Mis Favoritos" actualizada.
Importancia	Media

Tabla 53 Caso de uso "Guardar favorito"

Identificador	CU-29 Seleccionar web y leer noticias
Descripción	Caso de uso correspondiente al RU-29
Actores relacionados	Usuario, Portal Noticias
Casos de Uso relacionados	
Precondiciones	1. El usuario habrá realizado el CU-21.
Flujo Principal	<ol style="list-style-type: none"> 1. El sistema redirige al usuario a una nueva página denominada “Novedades” y carga el sitio web escogido por el usuario. 2. El usuario selecciona cualquier noticia para conocer más detalles. 3. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	
Postcondiciones	El sistema muestra la página “Novedades” satisfactoriamente.
Importancia	Media

Tabla 54 Caso de uso "Seleccionar web y leer noticias"

Identificador	CU-30 Compartir enlace descarga App
Descripción	Caso de uso correspondiente al RU-30.
Actores relacionados	Usuario, Medios Compartir
Casos de Uso relacionados	
Precondiciones	1. El usuario habrá realizado el CU-24.
Flujo Principal	<ol style="list-style-type: none"> 1. El sistema despliega la aplicación seleccionada por el usuario incluyendo el enlace de descarga de la aplicación a través de “Play Store”. 2. El usuario realiza el envío del enlace de descarga a la persona que desee. 3. El caso de uso finaliza satisfactoriamente.
Flujo Alternativo	
Postcondiciones	El sistema retorna a la vista anterior al momento de compartir la aplicación-
Importancia	Media

Tabla 55 Caso de uso "Compartir enlace descarga App"

4.3 Reglas de Negocio

Las reglas de negocio no son requisitos de software por sí mismas, pero si son requisitos que hay que tener en cuenta ya que todo proyecto debe estar sujeto a un conjunto de normas, mandatos o restricciones de carácter obligatorio. Estas reglas por lo tanto serán la base y afectan al resto de requisitos que encontramos en nuestro proyecto, ya que establecen ciertas limitaciones para algunos de ellos.

Por lo tanto, podemos definir las reglas de negocio como aquellas que establecen todas las regulaciones y políticas de negocio que deberá seguir “App4CyclingLovers” de acuerdo con lo establecido por las normativas vigentes en la actualidad. En nuestro caso se han localizado las siguientes:

- **RN-01:** App4CyclingLovers deberá cumplir la Ley Orgánica de Protección de datos en relación con la información publicada de corredores y equipos.
- **RN-02:** El usuario no tendrá que registrarse ni identificarse de ninguna manera para poder hacer uso de todas las funciones de la aplicación.
- **RN-03:** La aplicación únicamente permitirá visualizar la información, ni modificarla ni añadirla.
- **RN-04:** El usuario deberá mantener unos estándares de respeto a la hora de realizar comunicaciones con los encargados de soporte de la aplicación.
- **RN-05:** Los datos publicados al respecto de corredores, equipos y carreras constan de una veracidad y actualización plenas.
- **RN-06:** Los Resultados de las carreras que se vayan celebrando se incorporarán en la aplicación a partir de la finalización de esta.
- **RN-07:** Los datos relativos a la Clasificación del Ranking Mundial se actualizarán en la aplicación semanalmente (principalmente el lunes).
- **RN-08:** La Clasificación del Ranking Mundial incluirá a los 100 primeros ciclistas clasificados en cada momento.
- **RN-09:** El archivo de datos relativo a equipos, carreras y corredores incluye el perteneciente a las temporadas 2020 y 2021 exclusivamente.
- **RN-10:** El usuario puede seleccionar un corredor y/o un equipo como favorito únicamente al mismo tiempo.
- **RN-11:** El usuario puede seleccionar cualquier corredor como favorito independientemente de que no pertenezca al equipo almacenado como favorito en ese momento.
- **RN-12:** Los idiomas disponibles para la visualización de la aplicación son el español y el inglés (UK).
- **RN-13:** El idioma de visualización de la aplicación cambiará automáticamente en relación con el idioma predeterminado detectado en el dispositivo.
- **RN-14:** Los sitios web de noticias que el usuario puede seleccionar para consultar son: Ciclismo a fondo, Cycling News, y la web oficial de la Unión Ciclista Internacional.
- **RN-15:** Las Carreras que no se pudieran disputar por algún motivo no dispondrán de resultados para consultar.

- **RN-16:** Puede darse el caso de que algunas Carreras no dispongan de los resultados cosechados por los equipos participantes.
- **RN-17:** La aplicación permite consultar si una determinada carrera se está celebrando en el día actual.
- **RN-18:** Algunos corredores pueden aparecer en el sistema inscritos en una determinada temporada, pero en otra no debido a una posible retirada o al comienzo de su carrera profesional.
- **RN-19:** Todos los datos que aparecen en la aplicación son incorporados de manera externa por el encargado de administración de esta.
- **RN-20:** Las Carreras, equipos y Corredores únicamente incluyen los pertenecientes a la 1ª División del Ciclismo Mundial (“Categoría World Tour”).
- **RN-21:** La aplicación no va a obtener ningún dato de carácter personal del usuario, salvo que lo solicite al ponerse en contacto con el administrador de la aplicación.

4.4 Requisitos Funcionales

En este apartado se van a detallar las especificaciones que deberá cumplir el sistema para poder satisfacer los Requisitos de Usuario enunciados previamente en el apartado **4.2** de esta documentación. Por lo tanto, un requisito funcional define una función del sistema de software o sus componentes.

En primer lugar, se presentan algunos requisitos que son comunes a varios casos de uso, para así evitar una duplicidad de requisitos:

- **RF-01:** El sistema será capaz de cargar la información correspondiente de la base de datos y mostrar al usuario la información solicitada.
- **RF-02:** El sistema habilitará la opción de retroceder hacia la vista anterior accedida por el usuario.
- **RF-52:** El sistema permitirá al usuario desplegar el menú lateral para acceder a las diferentes secciones y funcionalidades del sistema.

A continuación, se presentan un listado de los requisitos funcionales mediante un identificador y una descripción de cada uno de ellos, organizados por el caso de uso al que pertenecen (tabla 56). En este sentido conviene aclarar que las alteraciones producidas en la nomenclatura de los identificadores de los requisitos se deben a que se han ido incorporando y eliminando requisitos a medida que se ha profundizado en el análisis del sistema.

Caso de uso	Identificador	Descripción
CU-01	RF-02, RF-52	
	RF-03	El sistema mostrará la pantalla de bienvenida de la aplicación.
	RF-04	El sistema ofrecerá la posibilidad de acceder a la aplicación.
	RF-05	El sistema mostrará por pantalla la página principal de la aplicación tras el acceso del usuario.
CU-02	RF-01, RF-02, RF-52	
	RF-06	El sistema habilitará el Selector de fecha en el Calendario por parte del usuario.
	RF-07	El sistema actualizará la pantalla principal con la fecha seleccionada por el usuario.
	RF-08	El sistema habilitará la opción de consultar el listado con las Carreras disponibles en la fecha seleccionada por el usuario.
CU-03	RF-01, RF-02	
	RF-09	El sistema habilitará la posibilidad de seleccionar una determinada carrera y acceder a sus datos.
CU-04	RF-01, RF-02	
	RF-10	El sistema habilitará la opción de consultar la clasificación de corredores y equipos de la carrera seleccionada.
	RF-48	El sistema mostrará por pantalla al usuario el mensaje informando de que los resultados de corredores y equipos de una determinada carrera o etapa no se encuentran disponibles por el momento.
CU-05	RF-01, RF-02	
	RF-11	El sistema habilitará la opción "Detalles" en caso de que la carrera seleccionada sea por etapas.
CU-06	RF-01, RF-02	
	RF-12	El sistema habilitará la posibilidad de seleccionar una determinada etapa y acceder a sus datos.
CU-07	RF-01, RF-02, RF-48	
	RF-13	El sistema habilitará la opción de consultar la clasificación de corredores y equipos de la etapa seleccionada.
CU-08	RF-01, RF-02	
	RF-14	El sistema habilitará la posibilidad de consultar si la carrera seleccionada consta de alguna etapa que se esté celebrando en la fecha actual.
	RF-15	El sistema ofrecerá la posibilidad de seleccionar la etapa en curso y acceder a sus datos.
CU-09	RF-01, RF-02, RF-52	
	RF-16	El sistema habilitará la posibilidad de acceder a consultar la Clasificación Mundial de Corredores en el

		momento actual desde el icono “Ranking” situado en la barra de navegación de la parte inferior.
CU-10	RF-01, RF-02, RF-52	
	RF-17	El sistema habilitará la posibilidad de acceder a la lista de Equipos desde el menú lateral de la aplicación.
	RF-18	El sistema permitirá seleccionar al usuario la temporada de la que desea consultar la lista de equipos. (2020 o 2021)
CU-11	RF-01, RF-02	
	RF-19	El sistema habilitará la posibilidad de seleccionar un determinado equipo/corredor y acceder a sus datos.
CU-12	RF-01, RF-02	
	RF-20	El sistema habilitará la opción de consultar la lista de corredores que forman parte del equipo seleccionado.
CU-13	RF-02	
	RF-21	El sistema habilitará la opción de marcar un determinado equipo o corredor como favorito.
	RF-22	El sistema te redirigirá a la página de tus favoritos.
	RF-23	El sistema ofrecerá la posibilidad de consultar el equipo/corredor almacenado como favorito antes de actualizar los cambios en los favoritos.
	RF-24	El sistema habilitará la posibilidad de guardar como favorito el equipo/corredor seleccionado por el usuario. (Realizar CU-28)
CU-14	RF-01, RF-02, RF-19	
CU-15	RF-02	
	RF-28	El sistema habilitará la opción de acceder al sitio web del corredor consultado por el usuario.
	RF-29	El sistema desplegará el navegador del dispositivo electrónico y mostrará el sitio web del corredor seleccionado.
CU-16	RF-21, RF-22, RF-23, RF-24, RF-25, RF-26, RF-27	
CU-17	RF-23, RF-02	
	RF-30	El sistema cargará el archivo de preferencias y mostrará al usuario los favoritos almacenados en ese momento.
CU-18	RF-01, RF-02, RF-19, RF-20, RF-28, RF-29,	
CU-19	RF-01, RF-02, RF-52	
	RF-31	El sistema habilitará la posibilidad de acceder a la búsqueda de corredores por su nombre desde el menú lateral de la aplicación.
	RF-32	El sistema mostrará un cuadro de texto al usuario para que incluya el texto a través del cual quiere realizar un determinado filtrado (de carreras o corredores).
CU-20	RF-01, RF-02, RF-19	
CU-21	RF-02, RF-52	

	RF-33	El sistema habilitará la posibilidad de acceder a la sección de “Noticias” desde el menú lateral de la aplicación.
	RF-34	El sistema mostrará los sitios web de noticias disponibles para consultar.
	RF-35	El sistema habilitará la posibilidad de realizar el CU-29 .
CU-22	RF-02, RF-32, RF-52	
CU-22	RF-37	El sistema habilitará la posibilidad de acceder al filtrado de Carreras por su nombre desde el menú lateral de la aplicación.
CU-23	RF-01, RF-02, RF-09, RF-10, RF-11, RF-12, RF-13, RF-14, RF-15	
CU-24	RF-02	
	RF-38	El sistema mostrará al usuario las vías disponibles para poder compartir la aplicación a otros usuarios.
	RF-39	El sistema habilitará la posibilidad de realizar el CU-30 .
CU-25	RF-02, RF-44, RF-45	
	RF-40	El sistema habilitará la opción de poder contactar con el soporte de la aplicación desde el menú lateral de la aplicación.
	RF-41	El sistema mostrará por pantalla al usuario los campos a introducir para el envío del mensaje al administrador de la aplicación.
CU-26	RF-02	
	RF-42	El sistema ofrecerá la posibilidad de solicitar un email de respuesta por parte del administrador de la aplicación de manera externa.
	RF-43	El sistema mostrará al usuario el campo para introducir la dirección de correo electrónico a la que desea recibir la respuesta a la petición enviada.
	RF-44	El sistema realizará el envío del comentario desarrollado por el usuario a la cuenta gestionada por el administrador de la aplicación.
	RF-45	El sistema mostrará por pantalla al usuario el mensaje de confirmación del envío de su comentario.
CU-27	RF-02	
	RF-47	El sistema habilitará la posibilidad de acceder a la sección “Mis Favoritos” a través de la selección del icono “Favoritos” de la barra de navegación situada en la parte inferior de la pantalla.
CU-28	RF-02	
	RF-25	El sistema modificará el archivo de preferencias para actualizar los cambios introducidos por el usuario en los favoritos.
	RF-26	El sistema mostrará por pantalla al usuario el mensaje de “Favorito Guardado correctamente”.

	RF-27	El sistema cargará el archivo de preferencias y mostrará al usuario el nuevo favorito añadido.
CU-29	RF-02	
	RF-49	El sistema desplegará el sitio web de noticias seleccionado por el usuario.
	RF-50	El sistema habilitará la posibilidad de seleccionar una noticia para su lectura.
CU-30	RF-02	
	RF-51	El sistema desplegará la “vía” seleccionada por el usuario para compartir la aplicación, incluyendo el enlace de descarga de esta.

Tabla 56 Requisitos Funcionales asociados a los casos de uso

4.5 Requisitos No Funcionales

4.5.1 Requisitos de Interfaz Externa

Este tipo de requisitos definen las interfaces o componentes de carácter externo con los que contará el sistema. En nuestro caso como ya he descrito en la documentación, nuestro sistema se comunicará con los siguientes servicios:

- **RIE-01:** El sistema se comunicará con el servicio “Cloud Firestore” (de Firebase) cada vez que el usuario solicite una información, para recuperarla y posteriormente mostrársela al usuario.
- **RIE-02:** El sistema se conecta con la herramienta solicitada por el usuario para compartir el enlace de descarga de la aplicación a través de PlayStore (por ejemplo, con WhatsApp o Gmail).
- **RIE-03:** El sistema se comunicará con el navegador del dispositivo electrónico para desplegar el sitio web de un determinado corredor.

4.5.2 Atributos de Calidad

Este tipo de requisitos revisten una importancia capital, ya que nos permitirán discernir si el sistema cumple con todos los estándares de calidad y también comprobar si el rendimiento del sistema es el correcto. Estos atributos serán medibles y cuantificables para determinar si el sistema cumple con la especificación.

A continuación, se agrupan los diferentes requisitos agrupados por la categoría del atributo:

- **AC-01: Seguridad:** la comunicación entre el cliente y el servidor se realiza mediante el protocolo de comunicación HTTPS.
- **AC-02: Robustez:** el sistema se mantendrá en activo ante situaciones de caída.
- **AC-03: Robustez:** el sistema se mantendrá estable ante situaciones inusuales (excesivo número de usuarios, importante tráfico de datos).
- **AC-04: Rendimiento:** el sistema responderá en un tiempo inferior a cinco segundos.

- **AC-05: Disponibilidad:** el sistema deberá estar disponible 24/7 para que el usuario pueda acceder.
- **AC-06: Escalabilidad:** la aplicación será compatible con cualquier dispositivo que disponga de sistema operativo Android 4.1 o superior.
- **AC-07: Usabilidad:** la aplicación será compatible con los dispositivos conectados a internet y que dispongan de las características de software mínimas requeridas.
- **AC-08: Usabilidad:** los usuarios únicamente disponen de permiso de lectura de los datos almacenados en la base de datos.
- **AC-09: Usabilidad:** el sistema mostrará al usuario los mensajes pertinentes en caso de no aparecer los datos o por necesidad de alguna confirmación.
- **AC-10: Usabilidad:** el usuario será capaz de acceder de manera sencilla e intuitiva a los datos y opciones que le interesen.
- **AC-11: Accesibilidad:** la aplicación será adaptable a los diferentes tamaños de dispositivos presentes en el mercado para su correcta visualización.
- **AC-12: Accesibilidad:** la aplicación será compatible con el idioma del teléfono español e inglés (UK).
- **AC-13: Eficiencia:** el sistema deberá ser capaz de soportar la presencia de numerosos usuarios al mismo tiempo.
- **AC-14: Implementación:** el desarrollo de la aplicación se realizará a través del entorno de programación Android Studio 4.0 y con lenguaje de programación Java.

4.6 Requisitos de Información

Los requisitos de información describen todos los aspectos relacionados con los datos creados, gestionados o emitidos por el sistema. En el caso de nuestro proyecto, el sistema simplemente se va a encargar de mostrar los datos recuperados de la base de datos, es decir no va a almacenar ningún conjunto de datos.

Los datos requeridos por la aplicación son:

- **RI-01:** El sistema se encargará de mostrar los datos relativos a las Carreras de categoría World Tour que se disputan en las temporadas 2020 y 2021.
- **RI-02:** El sistema se encargará de mostrar los datos relativos a los Equipos inscritos en las temporadas 2020 y 2021.
- **RI-03:** El sistema se encargará de mostrar los datos relativos a los Corredores que forman parte de cada uno de los equipos en las temporadas 2020 y 2021.
- **RI-04:** El sistema se encargará de mostrar los datos relativos a los resultados de las Carreras de categoría “World Tour” que se disputan en las temporadas 2020 y 2021.
- **RI-05:** El sistema se encargará de mostrar los datos relativos a la Clasificación Mundial de Corredores (según la UCI) en el instante actual.

Una vez definidos los requisitos de información con los que contará el sistema, faltaría por realizar una especificación más detallada de estos, y describir el modelo lógico de datos seguido en nuestro proyecto. Estas dos cuestiones se abordan con profundidad en los apartados **5.3 y 5.4** de esta Documentación.

5.DISEÑO

5.1 Arquitectura

5.1.1 Arquitectura Lógica

La arquitectura lógica consiste en un conjunto de patrones y abstracciones coherentes que proporcionan un marco definido y claro para interactuar con el código fuente del software.

Para elaborar el diseño de la arquitectura lógica de la aplicación hay que plantearse diferentes cuestiones, como por ejemplo elegir el número de capas de las que va a constar la aplicación. En nuestro caso hemos elegido una arquitectura de 3 capas (presentación, lógica y persistencia) que sea flexible y que permita granja de servidores web, granja de servidores de aplicaciones o un cluster de base de datos (en caso de ser necesarios).

Para diseñar la capa de presentación hay que tener en cuenta el tipo de aplicación (en este caso móvil), elegir la tecnología de la interfaz de usuario (alberga las vistas que se muestran al usuario), usar patrones de diseño (Modelo-Vista-Controlador) y plantearnos si necesitamos añadir componentes de proceso. En nuestro caso como el usuario no va a introducir ningún tipo de información al sistema, no habrá que realizar restricciones de seguridad en este sentido, pero si será necesario realizar un control de las excepciones para que la aplicación no deje de funcionar y establecer las comunicaciones con otras aplicaciones que requiera.

Respecto a la capa lógica de negocio, buscábamos implantar las tareas de negocio y guardar el estado de las operaciones de la aplicación, y los pasos que hemos seguido para su diseño han sido:

1. Crear un diseño de alto nivel de la capa de negocio.
2. Diseñar los componentes de negocio: implementan la lógica de negocio
3. Diseñar los componentes de entidad: estructuras de datos para el intercambio de información entre las capas.
4. Diseñar las interfaces para los servicios de la aplicación (comunicación, seguridad).
5. Diseñar los componentes de flujo de trabajo: transferencia de datos entre los diferentes componentes.

Por último, nos faltaría el diseño de la capa de persistencia, con el que queremos garantizar la persistencia de los datos en cualquier acceso a la base de datos desde nuestra aplicación. En nuestro caso se realiza a través del acceso a una base de datos situada en la “nube” (Cloud Firestore). Esta capa además incluye los servicios que interactúan de cualquier

forma con el sistema (en este caso, el API JavaMail para las comunicaciones del usuario con el administrador del sistema y los servicios de Google).

A continuación, se representa la arquitectura lógica asociada a nuestro sistema (ilustración 7):

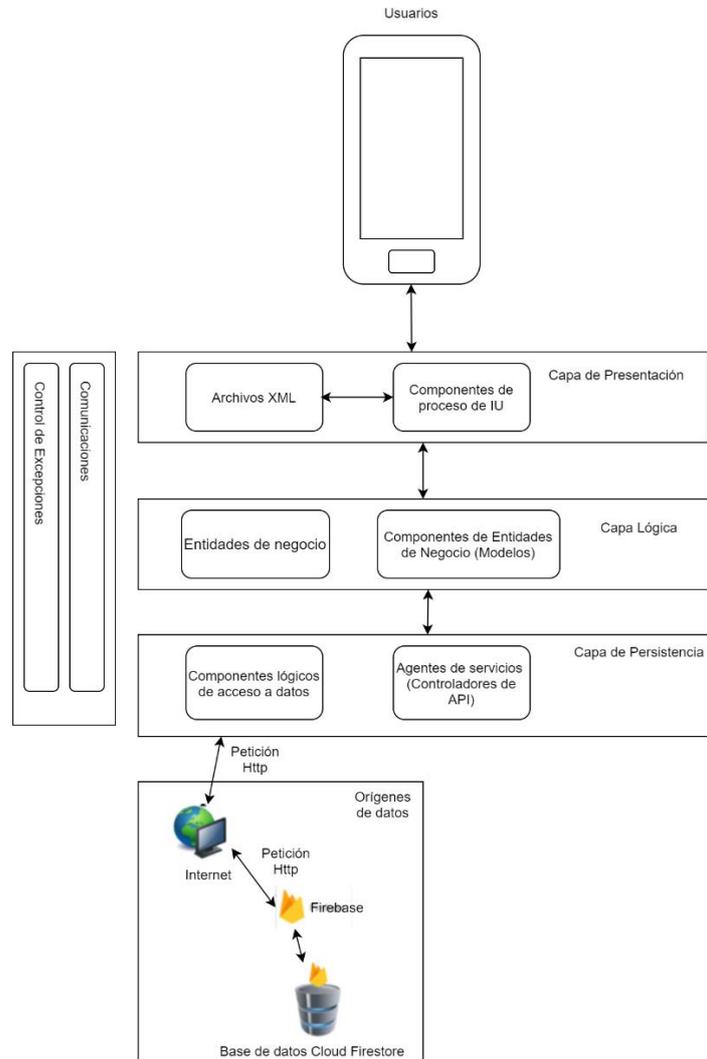


Ilustración 7 Arquitectura lógica de la aplicación

5.1.2 Arquitectura física

Se encarga de representar la forma en que se distribuye nuestra aplicación a los usuarios finales, en la que se denotan los actores y el medio a través del cual se hace llegar la aplicación al computador y/o dispositivo del usuario.

Lo primero que debemos tener en cuenta para el diseño de nuestra arquitectura física son qué atributos de calidad queremos satisfacer a través de ella, por lo que en nuestro sistema nos hemos decantado por un modelo cliente-servidor (2 capas) en función de los objetivos y características de nuestra aplicación.

Este tipo de arquitectura cuenta con las siguientes ventajas:

- Seguridad
- Centralización de los datos
- Tecnologías más maduras y robustas

En nuestro caso el componente del lado del cliente será el dispositivo electrónico usado por el usuario para acceder a la aplicación y en el lado del servidor se encontrarán el componente de Firebase para acceder a la base de datos, junto a los servicios de Google y el API para la gestión de las comunicaciones con el administrador.

En este punto el esquema de la arquitectura física de nuestro proyecto quedaría de la siguiente manera (ilustración 8):

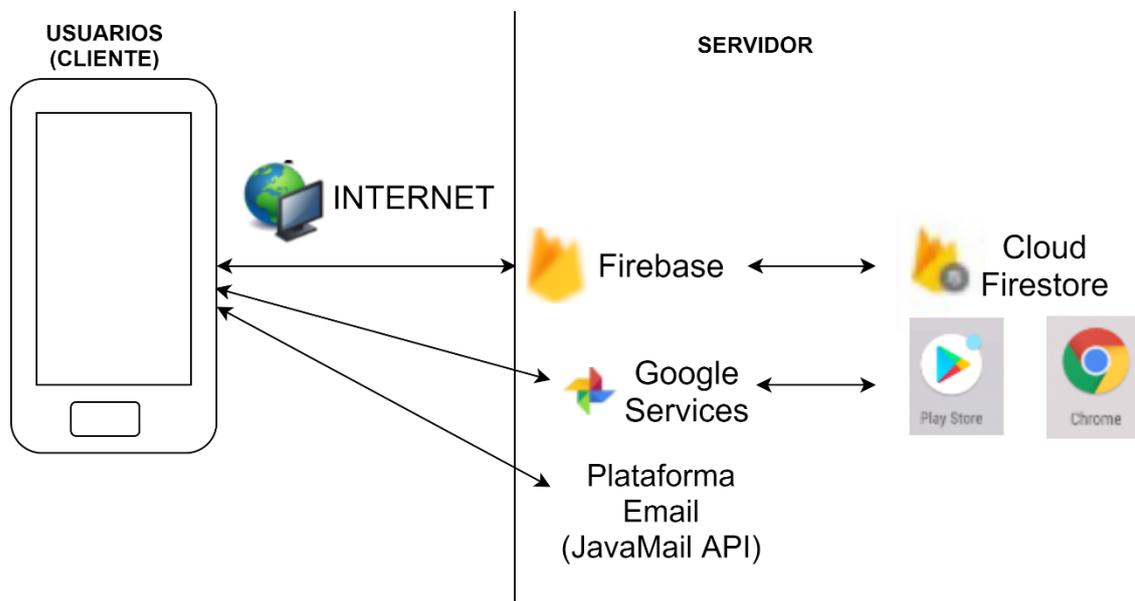


Ilustración 8 Arquitectura física de la aplicación

5.2 Modelos de Diseño

En este apartado se adjuntarán diferentes diagramas y representaciones que ayudarán a profundizar en el diseño final empleado en nuestra herramienta, tales como diagramas de clase, diagramas de secuencia, etc.

5.2.1 Diagrama de Clases

Representa la parte central del diseño, ya que permite representar las relaciones existentes entre las clases que involucran al sistema. Describe:

- **Clases:** contienen toda la información de un tipo de Objeto, junto a sus atributos (características) y métodos.
- **Relaciones entre las clases:** relaciones de uso, asociación entre las clases, relaciones de herencia.

Por lo tanto, la representación gráfica del diagrama de clases de la capa lógica de nuestro proyecto quedaría de la siguiente manera (ilustración 9):

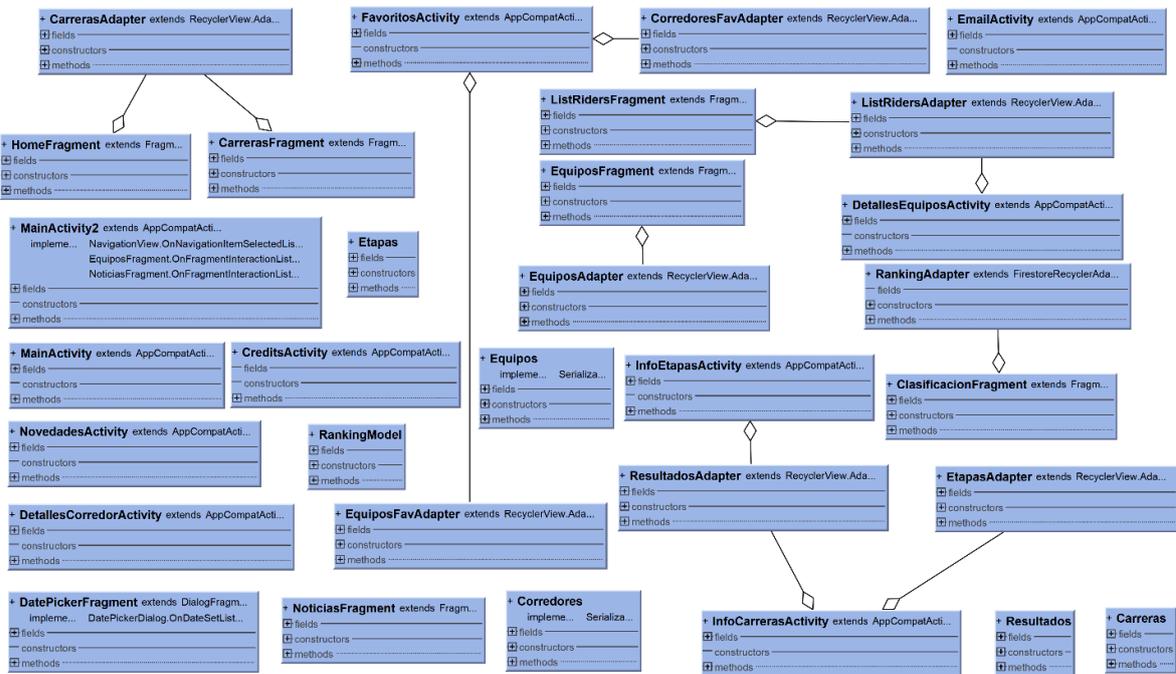


Ilustración 9 Diagrama de Clases

Además, también se adjuntará el archivo con el diagrama de clases completo (generado por el entorno de programación) en los “entregables” del proyecto para poder analizar convenientemente los distintos métodos, y atributos entre las clases.

5.2.2 Diagramas de Secuencia

Los diagramas de secuencia son un tipo de diagrama de interacción, ya que describen cómo, y en qué orden un grupo de objetos funcionan en conjunto. Se centran específicamente en líneas de vida o en los procesos y objetos que coexisten simultáneamente, y los mensajes intercambiados entre ellos para ejecutar una función antes de que la línea de vida termine.

En nuestro proyecto nos hemos centrado en representar como ejemplo tres funcionalidades independientes entre sí, ya que algunas de las funcionalidades del sistema se desarrollan de manera similar.

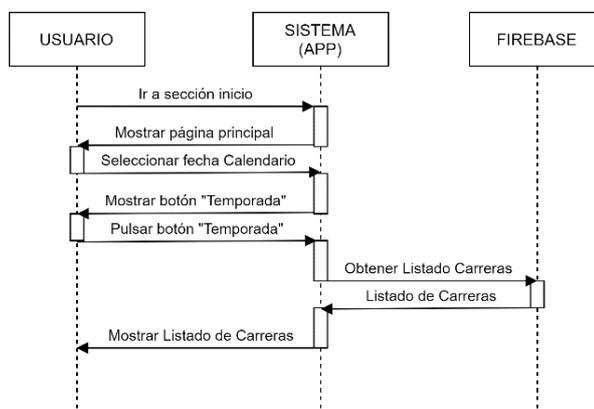


Ilustración 10 Diagrama de Secuencia "Visualizar Listado de Carreras en una fecha".

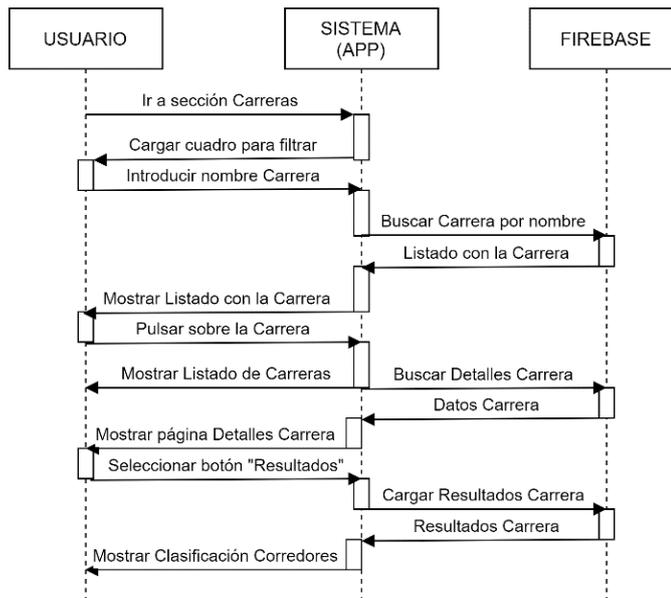


Ilustración 11 Diagrama de Secuencia "Consultar resultados de los corredores en una carrera"

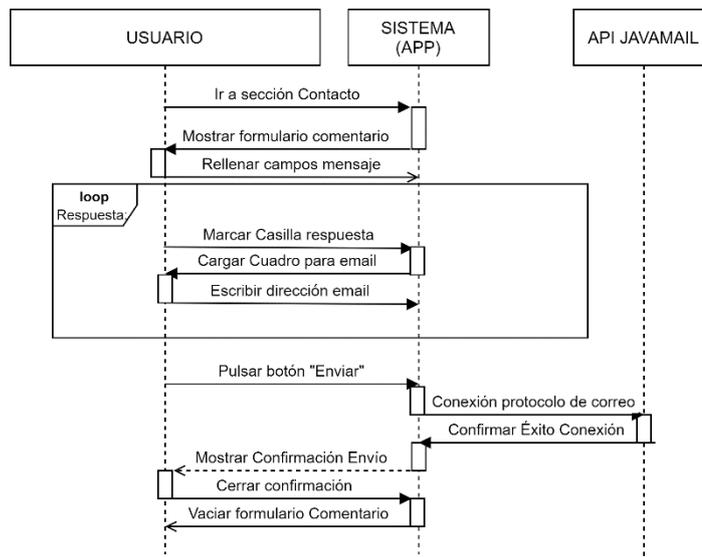


Ilustración 12 Diagrama de Secuencia "Enviar comentario al administrador del sistema"

5.2.3 Modelo Lógico de datos

En primer lugar, antes de describir la estructura de la base de datos seguida en nuestro proyecto, es necesario precisar la tipología de base de datos utilizada:

- **NoSQL:** Son los mecanismos para el almacenamiento y recuperación de datos que difieren, en aspectos importantes, del modelo clásico de relaciones entre entidades (tablas) existente en los sistemas de gestión de bases de datos relacionales, siendo el más destacado el que no usan SQL como lenguaje principal de consulta.

Los motivos para decantarnos por este modelo son la posibilidad de disponer de un esquema de base de datos no homogéneo, y la capacidad de soportar picos elevados de uso del sistema por parte de los usuarios.

Dentro de este modelo existen diferentes tipos, pero nosotros hemos utilizado uno basado en los documentos debido a sus características principales:

- Almacenan la información como un documento.
- Utilizan estructuras simples como JSON, BSON o XML.
- Permite realizar búsquedas por clave-valor y consultas más avanzadas sobre el contenido del documento.
- La naturaleza flexible, semiestructurada y jerárquica permite que evolucionen según las necesidades de las aplicaciones.

El servicio utilizado por nuestro proyecto para proporcionarnos dicho esquema de base de datos es **“Cloud Firestore”** perteneciente a la plataforma **Firestore**. A diferencia de las SQL, no hay ni tablas ni filas, simplemente almacena los diferentes campos en documentos (pares clave-valor), que se organizan en colecciones.

Los detalles de implementación relativos a este servicio se describen con más profundidad en el apartado **6.4 “Servidores de bases de datos utilizados”** de esta Documentación.

En este punto, la estructura de la base de datos utilizada en nuestro proyecto se refleja de la siguiente forma (ilustración 13):

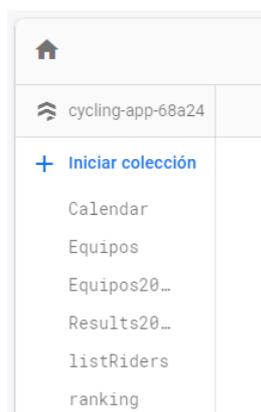


Ilustración 13 Estructura Base de datos “Cloud Firestore”

- Estas son las diferentes colecciones de datos que contienen la información a mostrar al usuario.
- Cada una de las colecciones cuenta con un número heterogéneo de documentos, los cuales cuentan con un identificador único y almacenan tipologías variables de datos.
- Sólo determinados documentos cuentan con alguna subcolección asociada a ellos.
- Las características anteriores muestran una estructura flexible y claramente jerarquizada.

Por ejemplo, al profundizar en los siguientes niveles de la estructura se observan los siguientes casos:

- Documentos sin ninguna subcolección anidada (ilustración 14)

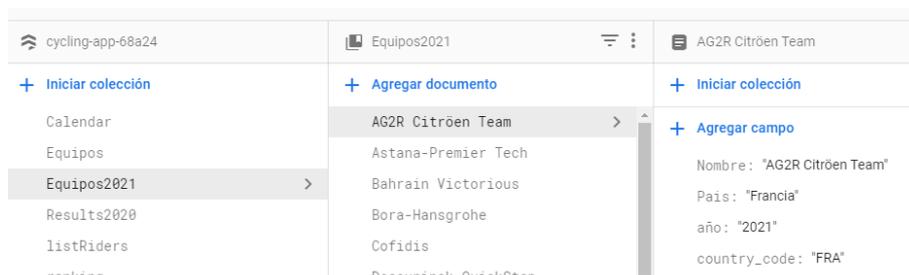


Ilustración 14 Estructura Documentos en base de datos

- Documento con subcolección anidada (ilustración 15)

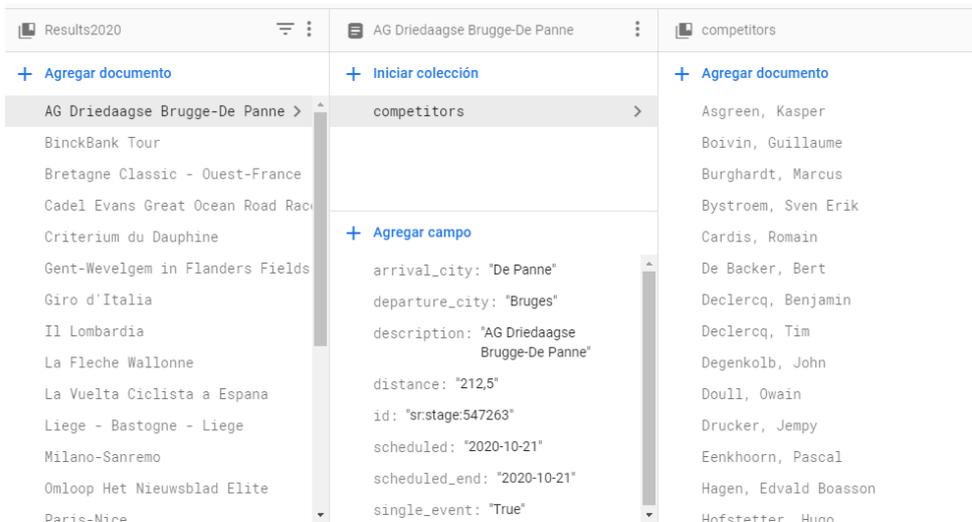


Ilustración 15 Estructura sub-colecciones en base de datos

La estructura seguida en cada una de las colecciones sigue el modelo representado en las imágenes anteriores.

Para finalizar faltaría por describir el procedimiento seguido para compatibilizar este modelo de datos con los modelos de datos utilizados en la implementación del sistema. La ventaja de “Cloud Firestore” es que interpreta estos modelos de datos de manera automática, y simplemente bastaría con utilizar objetos personalizados para cada uno de los conjuntos de datos almacenados en cada una de las colecciones. Un ejemplo de objeto personalizado es (ilustración 16):

- Conjunto de datos almacenados en una carrera del Calendario

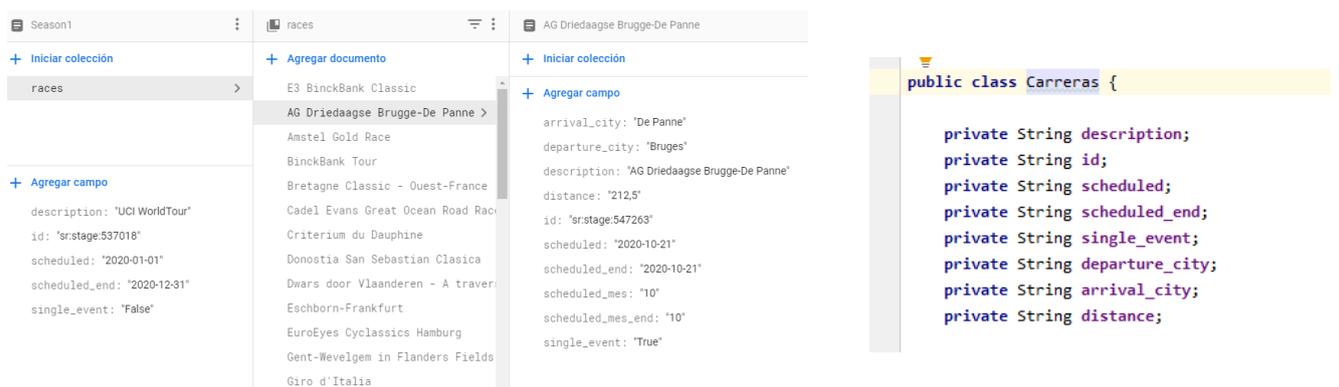


Ilustración 16 Equivalencia modelo datos Cloud Firestore- modelo de datos programación

5.2.4 Diccionario de datos

En este apartado se van a especificar los requisitos de información definidos en el punto 4.6 de esta Documentación. Cada uno de ellos está asociado a una colección o subcolección de las descritas en el modelo lógico de datos (tabla 57).

Identificador	Nombre
RI-C1	Carreras
RI-C2	Equipos
RI-C3	Corredores
RI-C4	Resultados Carreras
RI-C1	Clasificación Corredores

Tabla 57 Colecciones de Datos

RI-C1	Carreras					
Definición	Las Carreras describen a las competiciones ciclistas que forman parte del calendario de una temporada y en la que participan corredores y equipos.					
Consideraciones	Los elementos incluidos forman parte de las temporadas 2020 y 2021.					
Atributos						
ID	Nombre	Descripción	Domino	Unique	Null	Notas
RI-C1.1	description	El nombre de la carrera.	String	No	No	PK
RI-C1.2	Id	El identificador de la carrera.	String	Si	No	
RI-C1.3	Scheduled	La fecha de comienzo de la carrera.	String	No	No	Formato: yyyy/mm/dd
RI-C1.4	Scheduled_end	La fecha de finalización de la carrera.	String	No	No	
RI-C1.5	Scheduled_mes	El mes correspondiente a la celebración de la carrera.	String	No	No	
RI-C1.6	Single_event	El tipo de la carrera.	String	No	No	
RI-C1.7	Arrival_city	La ciudad en la que acaba la carrera.	String	No	Si	Este campo estará vacío en caso de no haberse presentado el recorrido.

RI-C1.8	Departure_city	La ciudad en la que inicia una carrera.	String	No	Si	Este campo estará vacío en caso de no haberse presentado el recorrido.
RI-C1.9	Distance	La cantidad de kms que engloba a la carrera.	String	No	Si	Este campo estará vacío en caso de no haberse presentado el recorrido.

Tabla 58 Diccionario Datos-Colección "Carreras"

RI-C2	Equipos					
Definición	Los Equipos describen a los equipos habilitados para participar en las carreras que se disputan en una determinada temporada.					
Consideraciones	Los elementos incluidos forman parte de las temporadas 2020 y/o 2021.					
Atributos						
ID	Nombre	Descripción	Dominio	Unique	Null	Notas
RI-C2.1	Nombre	El nombre del equipo.	String	No	No	PK
RI-C2.2	Pais	El país del equipo.	String	No	No	
RI-C2.3	año	La temporada en la que participa.	String	No	No	
RI-C2.4	Country_code	El código del país al que pertenece el equipo.	String	No	No	Formato: XXX

Tabla 59 Diccionario Datos- Colección "Equipos"

RI-C3	Corredores					
Definición	Los Corredores describen a cada una de las personas que forman parte de la plantilla de cada uno de los equipos inscritos.					
Consideraciones	Los elementos incluidos forman parte de las temporadas 2020 y/o 2021.					
Atributos						
ID	Nombre	Descripción	Domini o	Unique	Null	Notas
RI-C3.1	name	El nombre del corredor.	String	No	No	PK

RI-C3.2	age	La edad del corredor.	String	No	No	
RI-C3.3	temporada	La temporada en la que participa.	String	No	No	
RI-C3.4	Country_code	El código del país al que pertenece el equipo.	String	No	No	Formato: xx o XXX
RI-C3.5	puntos	La cantidad total de puntos acumulada en su trayectoria.	String	No	Si	Este campo será nulo en caso de que el corredor sea de la temporada 2021.
RI-C3.6	team	El equipo al que pertenece el corredor en el año concreto.	String	No	No	

Tabla 60 Diccionario- Datos Colección "Corredores"

RI-C4	Resultados de Carreras					
Definición	Los Resultados describen a cada uno de los participantes en una determinada carrera, ya sean corredores o equipos.					
Consideraciones	Los elementos incluidos forman parte de las temporadas 2020 y/o 2021 que se han disputado.					
Atributos						
ID	Nombre	Descripción	Dominio	Unique	Null	Notas
RI-C4.1	name	El nombre del corredor o equipo.	String	Si	no	PK
RI-C4.2	id	El identificador del corredor o equipo.	String	Si	No	
RI-C4.3	nationality	La nacionalidad del corredor o equipo.	String	No	No	
RI-C4.4	Country_code	El código del país al que pertenece el	String	No	No	Formato: XXX

		corredor o equipo.				
RI-C4.5	time	El tiempo empleado por el corredor o equipo en esa carrera.	String	No	Si	
RI-C4.6	Time_ranking	La posición ocupada en la clasificación de la carrera.	Int	Si	No	

Tabla 61 Diccionario Datos- Colección "Resultados de Carreras"

RI-C5	Clasificación de Corredores					
Definición	La clasificación describe a cada uno de los corredores que aparecen en el ranking de mejores clasificados en una determinada fecha.					
Consideraciones	La clasificación de Corredores se actualiza semana a semana. La clasificación incluye a los 100 primeros corredores.					
Atributos						
ID	Nombre	Descripción	Dominio	Único	Null	Notas
RI-C5.1	name	El nombre del corredor.	String	Si	No	PK
RI-C5.2	points	Los puntos del corredor acumulados en la temporada.	Int	No	No	
RI-C5.3	prev	La posición anterior que ocupaba en el ranking.	Int	Si	No	
RI-C5.4	rank	La posición actual que ocupa en el ranking.	Int	Si	No	
RI-C5.5	fecha	La fecha en la que se ha actualizado la clasificación.	String	Si	No	Formato: yyyy/mm/dd
RI-C5.6	team	El equipo al que pertenece el corredor.	String	No	No	

Tabla 62 Colección "Clasificación Mundial de Corredores"

5.3 Diseño de Interfaces

En primer lugar, es necesario precisar que debido a que en nuestro sistema la versión mínima del API de Android escogida es la 16 (compatible con casi la totalidad de dispositivos disponibles en el mercado) el estilo y el diseño del tema principal de la aplicación (seguido en todas las vistas) se ha proporcionado gracias a la librería “**Theme.AppCompat.NoActionBar**”. La inclusión de esta librería nos ha permitido emplear la normativa de diseño “**Material Design**” enfocada en mejorar la interfaz gráfica de Android que fue introducida a partir de la versión 20 del API de Android.

Por otro lado, también es conveniente describir los principales componentes de diseño empleados en cada una de las vistas de la aplicación:

- **Navigation Drawer:** es un elemento de interfaz definido por Material Design consistente en el típico menú lateral deslizante desde la izquierda y utilizado ampliamente como el principal elemento de navegación de las aplicaciones móviles. Suele encontrarse en la pantalla principal de la app y contar con un botón para desplegarlo, aunque también puede ser abierto deslizando el contenido de la pantalla desde el extremo izquierdo. A veces se suele complementar con el menú inferior Bottom Navigation.
- **CoordinatorLayout:** es un elemento que aglutina las diferentes vistas secundarias presentes en el diseño para una visualización eficiente.
- **LinearLayout:** es uno de los diseños más simples y más empleado. Simplemente establece los componentes visuales uno junto al otro, ya sea horizontal o verticalmente.
- **RecyclerView:** facilita que se muestren de manera eficiente grandes conjuntos de datos. Tú proporcionas los datos y defines el aspecto de cada elemento, y la biblioteca RecyclerView creará los elementos de forma dinámica cuando se los necesite.
- **WebView:** es un componente del sistema operativo Android, implementado dentro de las aplicaciones propias del sistema. Se trata de la app que permite que las aplicaciones puedan mostrar contenido web sin necesidad de que abramos un navegador externo.
- **Toolbar:** es un elemento presente en la parte superior de las aplicaciones de Android, que incluye el título de la aplicación y las herramientas requeridas en la misma.

5.3.1 Interfaces

En este apartado se van a describir las diferentes interfaces (pantallas) que se muestran al usuario en cada una de las funcionalidades de la aplicación. En este punto es necesario aclarar, que, aunque todas las vistas adopten el tema principal fijado, algunas de ellas adoptan criterios propios a la hora de mostrar o no en pantalla ciertos elementos (véase tablas 63 hasta 79).

Por otro lado, es necesario precisar que mediante este diseño de interfaces se pretende lograr un entorno intuitivo y fácilmente usable para el usuario. Además, se ha realizado un diseño adaptativo en cada una de las vistas (archivos XML), sobre las que se han aplicado

las características y estilos a los distintos elementos para garantizar una correcta visualización en todos los dispositivos.

En consecuencia, nuestro sistema cuenta con una pantalla principal, que adoptará dos mecanismos de diseño en función de la funcionalidad accedida:

- Se desplegará sobre la propia pantalla principal el contenido a mostrar al usuario.
- Se cargará por pantalla una nueva página a la cual será redirigido el propio usuario.

Pantalla Inicial	
Descripción	Pantalla que se muestra al iniciar la ejecución de la aplicación.
Activación	Al iniciar la aplicación en el dispositivo.
Eventos	<ul style="list-style-type: none"> • Acceso a la página principal de la aplicación pulsando el botón "Acceder" presente en la pantalla. • Finalizar ejecución de la aplicación al seleccionar el botón back del dispositivo.
Diseño	

Tabla 63 Interfaz "Pantalla Inicial"

Inicio/Pantalla Principal	
Descripción	<p>Pantalla que se muestra una vez el usuario ha accedido a la aplicación. Consta de:</p> <ul style="list-style-type: none"> • Toolbar con el título de la app y el Selector de fecha. • Button para desplegar el listado de Carreras. • BottomNavigationView para la selección de opciones.

	<ul style="list-style-type: none"> • RecyclerView para mostrar el Listado de carreras.
Activación	<ol style="list-style-type: none"> 1. Pantalla inicial de la aplicación 2. Icono "Inicio" de la barra de navegación inferior.
Eventos	<ul style="list-style-type: none"> • Acceso al menú lateral mediante el icono en la esquina superior izquierda o deslizando. • Desplegar interfaz con el selector de fecha al pulsar en el icono "Calendario" situado en la esquina superior derecha. • Consultar el Listado de Carreras disponibles en la fecha escogida pulsando el botón "Mostrar Eventos-Temporada". • Acceso a la información de una carrera (pulsando sobre ella). • Acceso a la sección "Favoritos" a través del icono de la barra de navegación inferior. • Acceso a la sección "Clasificación" a través del icono de la barra de navegación inferior. • Acceso a la vista anterior a través del botón "back" del dispositivo.
Diseño	

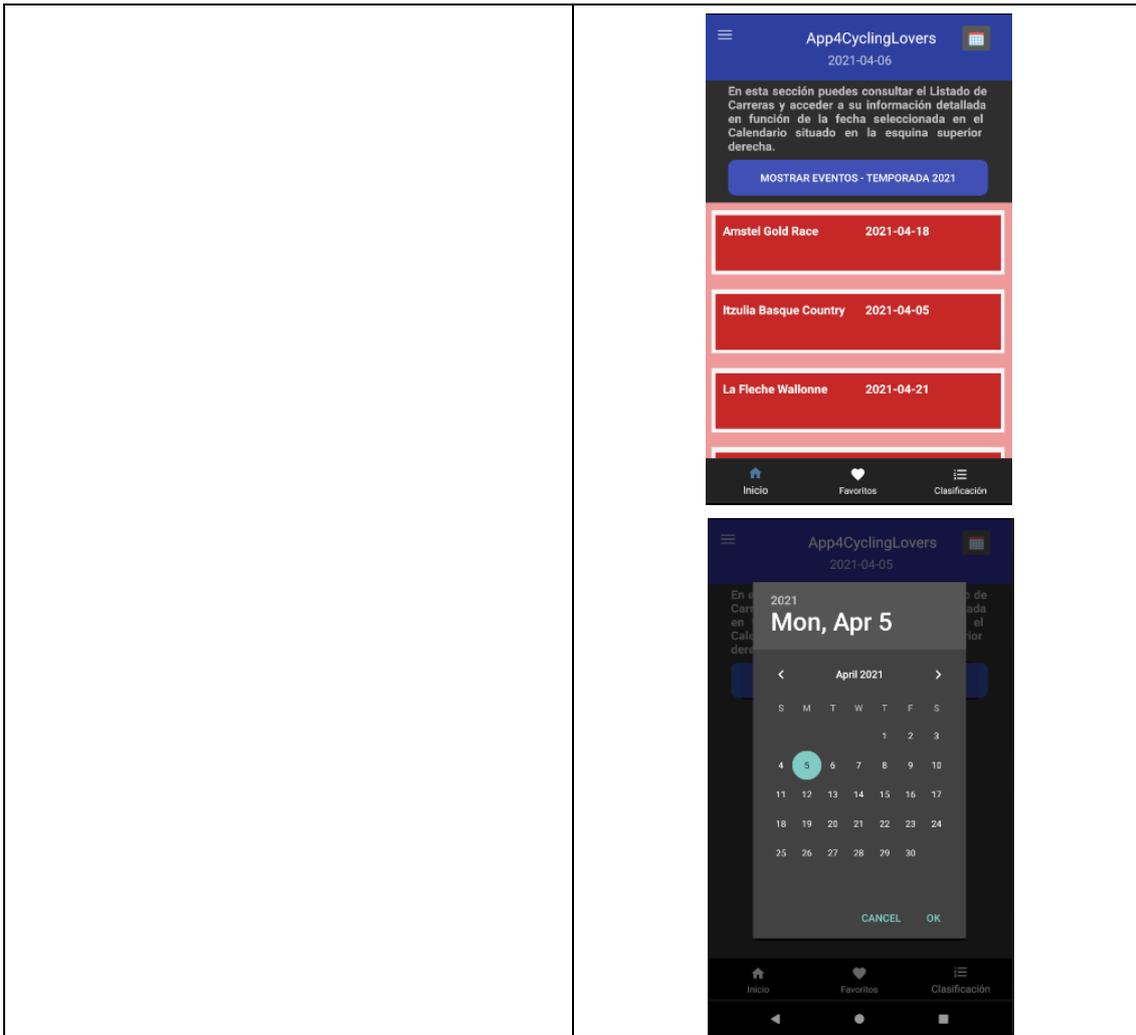


Tabla 64 Interfaz "Pantalla Principal"

Pantalla Detalles Carrera	
<p>Descripción</p>	<ul style="list-style-type: none"> • Pantalla que permite al usuario consultar los detalles de una determinada carrera, incluida su clasificación. Consta de: • Toolbar con el título de la pantalla en la que se encuentra el usuario. • TextView para mostrar los datos de la carrera (fechas, tipo, ciudad de salida, etc) • Button para desplegar el listado de las etapas pertenecientes a una carrera. (Sólo en caso de que sea "Vuelta por Etapas").

	<ul style="list-style-type: none"> • Dos Button para desplegar el listado con los resultados de los corredores o equipos. • Button para desplegar el listado con la etapa que se está disputando en la fecha actual. (Sólo en caso de que sea “Vuelta por etapas” y coincidan las fechas. • RecyclerView con cada uno de los elementos a mostrar (Etapas o Resultados).
Activación	<ol style="list-style-type: none"> 1. Pulsando sobre cualquiera de las carreras mostradas en la vista, ya sea desde la pantalla principal o desde la sección “Carreras”.
Eventos	<ul style="list-style-type: none"> • Acceso al listado de las etapas que forman parte de la carrera consultada haciendo “click” en el botón “Detalles” presente en la vista. • Acceso a los resultados de los corredores haciendo “click” en el botón “Resultados” presente en la vista. • Acceso a los resultados de los equipos haciendo “click” en el botón “Equipos” presente en la vista. • Acceso al listado de la etapa que se está desarrollando en la fecha actual haciendo “click” en el botón “Etapa en Curso”. • Acceso al contenido de cualquier etapa al seleccionar cualquiera de ellas en el listado mostrado. • Acceso a la vista anterior a través del botón “back” del dispositivo.

Diseño



Tabla 65 Interfaz "Detalles Carrera"

Pantalla Detalles Etapa	
Descripción	<p>Pantalla que permite al usuario consultar los detalles de una determinada etapa, incluida su clasificación. Consta de:</p> <ul style="list-style-type: none"> • Toolbar con el título de la pantalla en la que se encuentra el usuario. • TextView para mostrar los datos de la etapa (fechas, tipo, ciudad de salida, etc) • Dos Button para desplegar listado con los resultados de los corredores o equipos. • RecyclerView con los Listados de los Resultados a mostrar.
Activación	<ol style="list-style-type: none"> 1. Pulsando sobre cualquiera de las etapas mostradas en la vista, una vez que se ha accedido a la pantalla "Detalles Carrera".
Eventos	<ul style="list-style-type: none"> • Acceso a los resultados de los corredores haciendo "click" en el botón "Resultados" presente en la vista. • Acceso a los resultados de los equipos haciendo "click" en el botón "Equipos" presente en la vista. • Acceso a la vista anterior a través del botón "back" del dispositivo.
Diseño	 <p>The screenshot shows a mobile application interface for 'Detalles Etapa'. At the top is a blue header with the title 'Detalles Etapa'. Below the header, the following information is displayed: 'Nombre: Stage 3', 'Fecha Inicio: 2020-08-31', 'Fecha Fin: 2020-08-31', 'Tipo: Etapa de Tour de France', 'Ciudad de Salida: Nice', 'Ciudad de Llegada: Nice', and 'Distancia (km): 198'. Below this information are two blue buttons: 'RESULTADOS' and 'EQUIPOS'. Underneath the buttons is a list of teams with their times and positions. The first team is 'Movistar Team' with a time of 262:14.58 and position 1. The second team is 'Team Jumbo - Visma' with a time of +00:18.31 and position 2. The third team is 'Team' with a time of +00:57.10 and position 3. The list is styled with red and white text on a dark background.</p>

Tabla 66 Interfaz "Detalles Etapa"

Sección Clasificación	
Descripción	<p>Pantalla que se muestra al acceder a la “Sección Clasificación” y muestra la clasificación mundial de los corredores en el momento actual. Consta de:</p> <ul style="list-style-type: none"> • Toolbar con el título de la aplicación. • TextView con los títulos de los campos a mostrar. • RecyclerView con la información de cada uno de los elementos mostrados.
Activación	Al pulsar sobre el icono “Clasificación” presente en la pantalla principal de la aplicación.
Eventos	<ul style="list-style-type: none"> • Acceso a la vista anterior a través del botón “back” del dispositivo. • Acceso al menú lateral mediante el icono en la esquina superior izquierda o deslizando hacia la derecha.
Diseño	

Tabla 67 Interfaz "Sección Clasificación"

Sección Favoritos (Acceso Pantalla Principal)	
Descripción	<p>Pantalla que se muestra al acceder a la “Sección Favoritos” y permite consultar los elementos almacenados como favoritos en el dispositivo por parte del usuario. Consta de:</p>

	<ul style="list-style-type: none"> • Toolbar con el título de la pantalla en la que se encuentra el usuario. • Button para desplegar los RecyclerView (de Corredor y Equipo) con los elementos guardados.
Activación	Al pulsar sobre el icono “Favoritos” presente en la pantalla principal de la aplicación.
Eventos	<ul style="list-style-type: none"> • Acceso a la vista anterior a través del botón “back” del dispositivo. • Consultar los elementos guardados como favoritos en este momento. • Acceder al contenido del elemento favorito, pulsando sobre él.
Diseño	 <p>The image shows two screenshots of a mobile application screen titled "Mis Favoritos". The top screenshot displays a blue header with the title "Mis Favoritos", a paragraph of text: "En esta sección puedes consultar el Equipo y el Corredor almacenado como favorito, y actualizarlo desde sus respectivas secciones en el menú principal.", and a blue button labeled "CONSULTAR". The bottom screenshot shows the same screen with two items listed below the button. The first item is under the heading "Equipo" and is "Bora-Hansgrohe 2020". The second item is under the heading "Ciclista" and is "VENTURINI CLÉMENT 2020". Both items are highlighted with a red border.</p>

Tabla 68 Interfaz "Sección Favoritos-Acceso Pantalla Principal"

Sección Favoritos (Acceso Selección Elemento Favorito)	
Descripción	<ul style="list-style-type: none"> • Pantalla que permite consultar los elementos favoritos almacenados por el usuario en el dispositivo y modificarlos por el nuevo elemento seleccionado. Consta de: • Toolbar con el título de la pantalla en la que se encuentra el usuario. • Button para desplegar los RecyclerView (de Corredor y Equipo) con los elementos guardados. • Button para guardar los cambios en los favoritos y desplegar el RecyclerView (corredor o equipo) con el elemento actualizado.
Activación	Al pulsar sobre el icono “corazón” situado a la derecha del elemento “corredor” o “equipo” desplegado en la vista.
Eventos	<ul style="list-style-type: none"> • Acceso a la pantalla inicial a través del botón “back” del dispositivo. • Consultar los elementos guardados como favoritos en este momento haciendo “click” en el botón “Consultar”. • Acceder al contenido del elemento favorito, pulsando sobre él. • Modificar el elemento favorito correspondiente (corredor o equipo) por el nuevo escogido por el usuario pulsando sobre el botón “Guardar”.

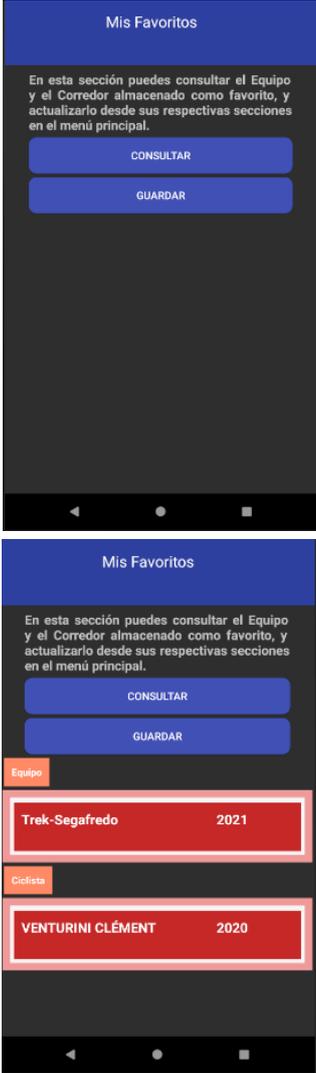
<p>Diseño</p>	
----------------------	---

Tabla 69 Interfaz "Sección Favoritos-Aceso Seleccion Item favorito"

<p>Menú Lateral</p>	
<p>Descripción</p>	<p>Pantalla a través de la cual el usuario puede acceder a las diferentes "Secciones" y funcionalidades proporcionadas por la aplicación.</p>
<p>Activación</p>	<ol style="list-style-type: none"> 1. Pulsando el icono ("tres rayas") presente en la esquina superior izquierda de la barra superior. 2. Deslizando hacia la derecha sobre la vista.
<p>Eventos</p>	<ul style="list-style-type: none"> • Acceso a la vista anterior a través del botón "back" del dispositivo. • Cierre del menú lateral deslizando hacia la izquierda de la vista o pulsando en otro punto de la pantalla.

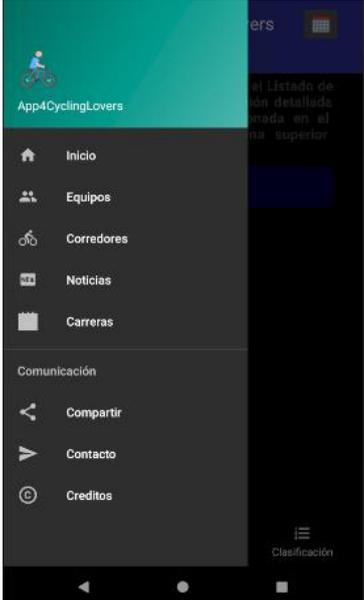
	<ul style="list-style-type: none"> • Acceso a las diferentes funcionalidades (“Secciones”) de la aplicación pulsando sobre ellas.
Diseño	

Tabla 70 Interfaz "Menú Lateral"

Sección Corredores	
Descripción	<p>Pantalla a través de la cual el usuario puede buscar cualquier corredor filtrando por su nombre. Consta de:</p> <ul style="list-style-type: none"> • Toolbar con el título de la aplicación. • Campo EditText para que el usuario introduzca el texto para el filtrado de corredores. • RecyclerView con el Listado de los corredores que cumplen las condiciones de búsqueda introducidas por el usuario.
Activación	<p>Desde el menú lateral, seleccionando la opción “Corredores”.</p>
Eventos	<ul style="list-style-type: none"> • Acceso a la vista anterior a través del botón “back” del dispositivo. • Acceso al menú lateral (procedimientos mencionados anteriormente). • Búsqueda de un determinado corredor a través de la introducción de texto en el cuadro mostrado.

	<ul style="list-style-type: none"> • Acceder al contenido de los corredores mostrados, pulsando sobre cualquiera de ellos. • Seleccionar un determinado corredor para guardar como favorito pulsando sobre el icono de “corazón” asociado a cada elemento.
Diseño	 <p>The screenshot shows the 'App4CyclingLovers' interface. At the top, there is a blue header with the app name and the date '2021-04-05'. Below the header, a text box explains that users can search for cyclists by name in a search bar at the bottom. The main content area is titled 'MADS' and displays a list of three cyclist entries, each with a red background and white text. The entries are: 'WÜRTZ SCHMIDT MADS 2020', 'PEDERSEN MADS 2021', and 'PEDERSEN MADS 2020'. Each entry has a white heart icon on the right side, indicating a favorite function. The bottom of the screen shows the standard Android navigation bar.</p>

Tabla 71 Interfaz "Sección Corredores"

Sección Carreras	
Descripción	<p>Pantalla a través de la cual el usuario puede buscar cualquier carrera filtrando por su nombre. Consta de:</p> <ul style="list-style-type: none"> • Toolbar con el título de la aplicación. • Campo EditText para que el usuario introduzca el texto para el filtrado de carreras. • RecyclerView con el Listado de las carreras que cumplen las condiciones de búsqueda introducidas por el usuario.
Activación	Desde el menú lateral, seleccionando la opción “Carreras”.
Eventos	<ul style="list-style-type: none"> • Acceso a la vista anterior a través del botón “back” del dispositivo. • Acceso al menú lateral (procedimientos mencionados anteriormente).

	<ul style="list-style-type: none"> • Búsqueda de una determinada carrera a través de la introducción de texto en el cuadro mostrado. • Acceder al contenido de las carreras mostradas, pulsando sobre cualquiera de ellas.
Diseño	

Tabla 72 Interfaz "Sección Carreras"

Sección Noticias	
Descripción	<p>Pantalla a través de la cual el usuario puede consultar los sitios web disponibles para acceder a consultar las últimas noticias. Consta de:</p> <ul style="list-style-type: none"> • Toolbar con el título de la aplicación. • Tres ImageButton con el icono representativo de cada sitio web disponible para que el usuario acceda a cualquiera de ellos.
Activación	Desde el menú lateral, seleccionando la opción "Noticias".
Eventos	<ul style="list-style-type: none"> • Acceso a la vista anterior a través del botón "back" del dispositivo. • Acceso al menú lateral (procedimientos mencionados anteriormente). • Acceso a una nueva pantalla ("Novedades") que desplegará un determinado sitio web haciendo

	“click” en cualquiera de los iconos presentes en la vista.
Diseño	 <p>The screenshot shows a mobile application interface. At the top, there is a blue header with the text 'App4CyclingLovers' and the date '2021-04-05'. Below the header, there is a dark grey area with the text 'CLICK EN LA IMAGEN PARA DESPLEGAR SITIO WEB' in pink. Underneath, there are three news items, each with a logo: 'Ciclismo a fondo' (yellow and black), 'cyclingnews' (purple), and 'UCI' (white with a rainbow stripe).</p>

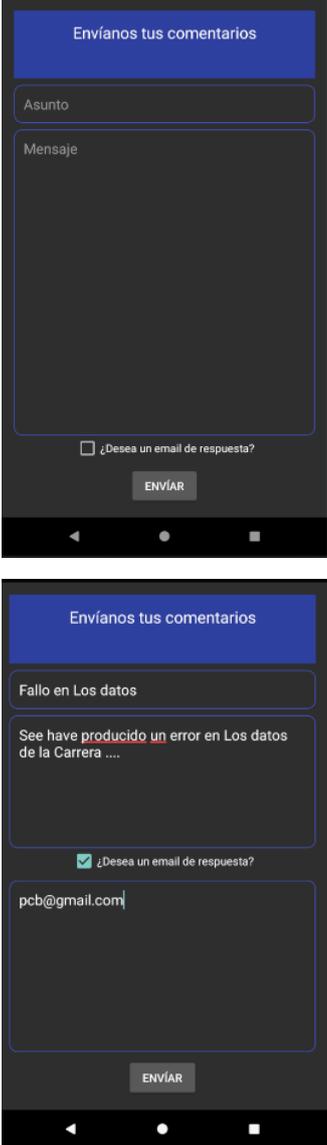
Tabla 73 Interfaz "Sección Noticias"

Pantalla Novedades	
Descripción	<p>Pantalla a través de la cual el usuario puede navegar por un determinado sitio web y consultar sus noticias. Consta de:</p> <ul style="list-style-type: none"> • Toolbar con el título de la pantalla en la que se encuentra el usuario. • WebView que carga la vista con el respectivo sitio web de noticias sin salir de la aplicación.
Activación	<p>Desde la sección “Noticias”, al pulsar sobre cualquiera de los “ImageButton” presentes.</p>
Eventos	<ul style="list-style-type: none"> • Acceso a la vista anterior a través del botón “back” del dispositivo. • Acceso al contenido más detallado de una noticia pulsando sobre ella.

<p>Diseño</p>	
----------------------	--

Tabla 74 Interfaz "Pantalla Novedades"

<p>Sección Contacto</p>	
<p>Descripción</p>	<p>Pantalla a través de la cual el usuario puede enviar cualquier comentario o duda al administrador de la aplicación. Consta de:</p> <ul style="list-style-type: none"> • Toolbar con el título de la pantalla en la que se encuentra el usuario. • Campo EditText para que el usuario introduzca el motivo del mensaje. • Campo EditText para que el usuario redacte el contenido del mensaje. • Checkbox para que el usuario seleccione si desea recibir una respuesta individualizada. • CampoEditText para que el usuario introduzca el correo al que desea recibir la respuesta. Sólo se despliega en caso de marcar la casilla anterior. • Button para que el usuario efectúe el envío del mensaje.
<p>Activación</p>	<p>Desde el menú lateral, seleccionando la opción "Contacto".</p>
<p>Eventos</p>	<ul style="list-style-type: none"> • Acceso a la vista anterior a través del botón "back" del dispositivo.

	<ul style="list-style-type: none">• Envío del comentario redactado al administrador a través del “click” sobre el botón “Enviar”.• Solicitar respuesta a través de la selección en la casilla planteada para ello.• Limpiar el formulario de mensaje pulsando sobre el botón “ok” en la alerta de confirmación desplegada en pantalla.
Diseño	

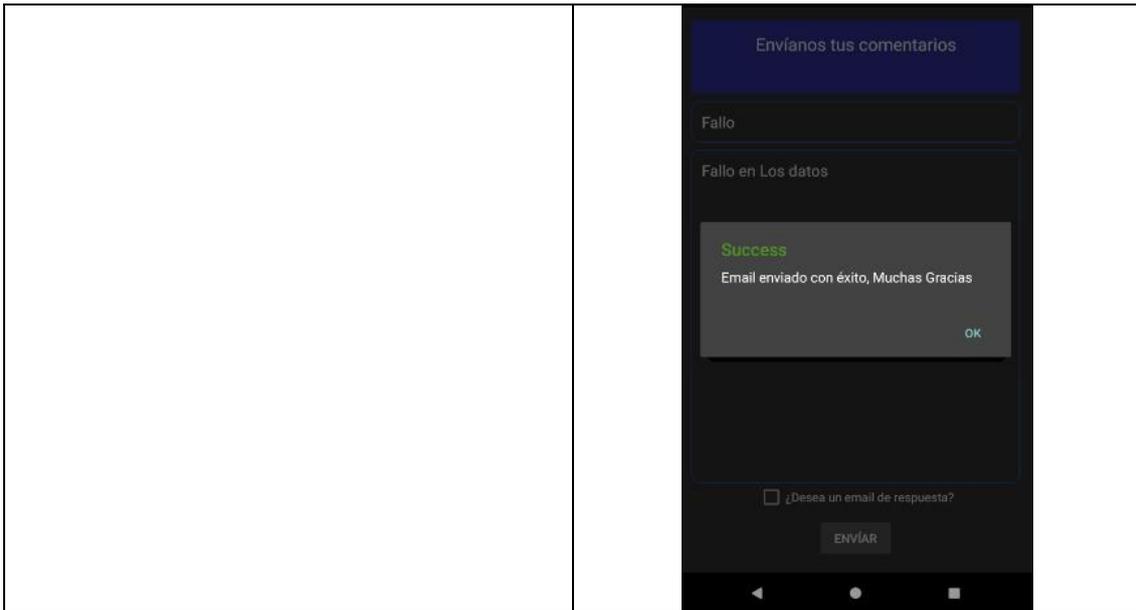


Tabla 75 Interfaz "Sección Contacto"

Sección Créditos	
Descripción	<p>Pantalla que muestra al usuario la información correspondiente a los créditos de la aplicación (procedencia datos, iconos). Consta de:</p> <ul style="list-style-type: none"> • Toolbar con el título de la pantalla en la que se encuentra el usuario. • TextView con la información relativa a los orígenes de los datos e imágenes.
Activación	Desde el menú lateral, seleccionando la opción "Creditos".
Eventos	<ul style="list-style-type: none"> • Acceso a la vista anterior a través del botón "back" del dispositivo.

<p>Diseño</p>	
----------------------	--

Tabla 76 Interfaz "Sección Créditos"

<p>Sección Compartir</p>	
<p>Descripción</p>	<p>Pantalla que se despliega y muestra al usuario los medios disponibles para poder compartir la aplicación con otro usuario.</p>
<p>Activación</p>	<p>Desde el menú lateral, seleccionando la opción "Compartir".</p>
<p>Eventos</p>	<ul style="list-style-type: none"> • Desplegar en el dispositivo el medio escogido por el usuario para compartir el enlace de descarga de la aplicación. En este caso no se finaliza la aplicación, y una vez completado el evento se regresa a la vista anterior. • Acceso a la vista anterior a través del botón "back" del dispositivo o pulsando fuera de la pantalla desplegada.

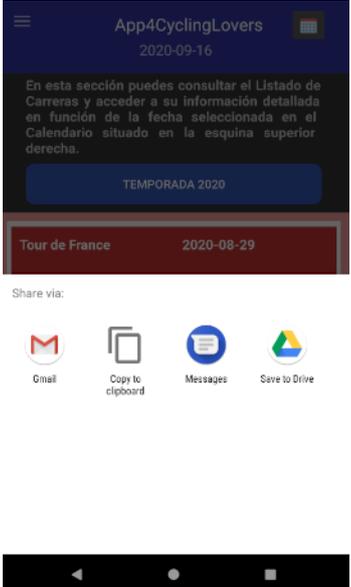
<p>Diseño</p>	 <p>The screenshot shows the 'App4CyclingLovers' app interface. At the top, it displays the app name and the date '2020-09-16'. Below this, there is a text block explaining that users can consult a list of races and access detailed information based on the date selected in the calendar. A blue button labeled 'TEMPORADA 2020' is visible. Below the button, a red banner displays 'Tour de France' and the date '2020-08-29'. A 'Share via:' section is open, showing icons for Gmail, Copy to clipboard, Messages, and Save to Drive. The bottom of the screen shows the Android navigation bar.</p>
----------------------	--

Tabla 77 Interfaz "Sección Compartir App"

<p>Pantalla Detalles Equipo</p>	
<p>Descripción</p>	<p>Pantalla que muestra al usuario la información correspondiente a un determinado equipo. Consta de:</p> <ul style="list-style-type: none"> • Toolbar con el título de la pantalla en la que se encuentra el usuario. • Tres campos TextView que muestran los datos relativos al nombre, temporada y país. • ImageView que muestra la bandera representativa al país en el que se encuentra la sede del equipo. • Button para que el usuario pueda desplegar la lista de corredores del equipo. • RecyclerView que muestra el Listado de Corredores que forman parte del equipo.
<p>Activación</p>	<ol style="list-style-type: none"> 1. Desde la sección "Equipos", pulsando sobre cualquiera de los elementos. 2. Desde la sección "Favoritos", pulsando sobre el equipo almacenado como favorito.
<p>Eventos</p>	<ul style="list-style-type: none"> • Acceso a la vista anterior a través del botón "back" del dispositivo.

	<ul style="list-style-type: none"> • Consultar la plantilla de Corredores del equipo al pulsar sobre el botón presente en la vista. • Acceso a la pantalla “Detalles Corredor” al pulsar sobre cualquiera de los corredores desplegados en el listado. • Seleccionar un determinado corredor para guardar como favorito pulsando sobre el icono de “corazón” asociado a cada elemento mostrado.
<p>Diseño</p>	 <p>The screenshot shows a mobile application interface titled 'Detalles Equipo'. It displays the following information: 'Nombre: Trek-Segafredo', 'Temporada Equipo: 2021', and 'País de Sede: Estados Unidos' with a US flag icon. Below this is a blue button labeled 'PLANTILLA DE CORREDORES'. Underneath the button is a list of three runners, each in a red card with a heart icon for favoriting: 'BERNARD JULIEN 2021', 'BRAMBILLA GIANLUCA 2021', and 'CICCONE GIULIO 2021'.</p>

Tabla 78 Interfaz "Detalles Equipo"

<p>Pantalla Detalles Corredor</p>	
<p>Descripción</p>	<p>Pantalla que muestra al usuario la información correspondiente a un determinado corredor. Consta de:</p> <ul style="list-style-type: none"> • Toolbar con el título de la pantalla en la que se encuentra el usuario. • Cinco campos TextView que muestran los datos relativos al nombre, equipo, edad, puntos y país de nacimiento. • ImageView que muestra la bandera representativa al país de nacimiento del corredor.

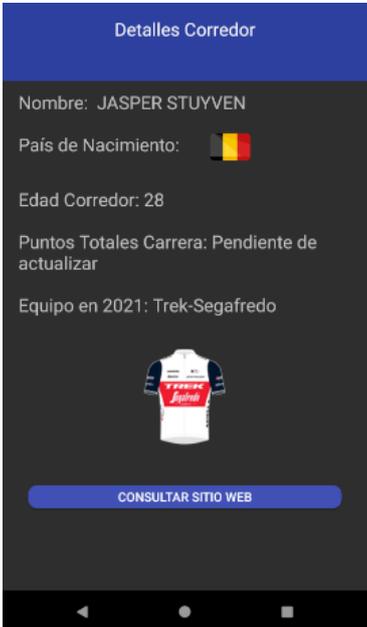
	<ul style="list-style-type: none"> • ImageView que muestra la equipación correspondiente al equipo del corredor. • Button para que el usuario pueda acceder al sitio web del corredor consultado.
Activación	<ol style="list-style-type: none"> 1. Desde la sección “Corredores”, pulsando sobre cualquiera de los elementos mostrados. 2. Desde la pantalla “Detalles Equipo”, pulsando sobre cualquiera de los corredores del equipo mostrados. 3. Desde la sección “Favoritos”, pulsando sobre el corredor almacenado como favorito.
Eventos	<ul style="list-style-type: none"> • Acceso a la vista anterior a través del botón “back” del dispositivo. • Acceso al sitio web del corredor consultado, pulsando sobre el botón indicativo presente en la pantalla. En este caso se despliega el navegador del dispositivo y se muestra el sitio web correspondiente, sin finalizar la aplicación.
Diseño	 <p>The screenshot shows a mobile application interface titled "Detalles Corredor". The background is dark grey. At the top, there is a blue header with the text "Detalles Corredor". Below the header, the following information is displayed in white text: "Nombre: JASPER STUYVEN", "País de Nacimiento:" followed by a small flag of Belgium, "Edad Corredor: 28", "Puntos Totales Carrera: Pendiente de actualizar", and "Equipo en 2021: Trek-Segafredo". Below this text is a small image of a white cycling jersey with blue and red accents. At the bottom of the screen, there is a blue button with the text "CONSULTAR SITIO WEB". The Android navigation bar is visible at the very bottom.</p>

Tabla 79 Interfaz "Detalles Corredor"

6. IMPLEMENTACIÓN

6.1 Requerimientos Hardware y Software

En este apartado se reflejan los requerimientos mínimos con los que debe contar cualquier dispositivo Android para poder utilizar la aplicación de manera eficiente.

6.1.1 Software

- Sistema Operativo: Android 4.1 o superior.
- Otros: Vías para compartir aplicación (ejemplo: Whatsapp o Gmail).
- Otros: Navegador de Internet.

6.1.2 Hardware

- RAM: 2 GB
- Velocidad CPU: Quad Core 1.2 GHz.
- Memoria Interna: 16 GB.
- Conexión: Wi-Fi y 4G (servicio de datos) para poder disponer de acceso a Internet en cualquier momento.
- Compatibilidad con recursos de Google.

6.2 Herramientas empleadas

6.2.1 Herramientas para el desarrollo móvil.

- **Android Studio:** entorno de programación oficial para la realización de cualquier software pensado para ejecutarse en dispositivos Android. Se basa en la tecnología IntelliJ IDEA y consta de diferentes herramientas de desarrollo como:
 - Editor libre de Código
 - Sistema de compilación basado en Gradle.
 - Integración con Github, para el control de versiones.
 - Emuladores, rápida y ampliamente configurables, que permiten al usuario probar el software desarrollado.
 - Compatibilidad con los servicios de Google.
- **Firebase:** plataforma de servicios con el objetivo de ayudar en el proceso de desarrollo de aplicaciones, tanto web como móviles. Cada servicio responde a una demanda concreta, por lo que en nuestro proyecto hemos empleado el servicio “Cloud Firestore”, que sirve para suministrar a nuestra aplicación una base de datos en tiempo real con localización en la “nube”. Se trata de una base de datos “NoSQL” capaz de almacenar gran cantidad de información, y fácilmente integrable con el entorno de programación Android Studio. Además, goza de gran escalabilidad para poder soportar una demanda creciente de usuarios.
- **JavaMail API:** es un paquete opcional (extensión estándar) para leer, componer, y enviar mensajes electrónicos directamente desde la aplicación. Básicamente, permite enviar y recibir mensajes independientemente del

proveedor/protocolo. Simplemente es necesario incorporar las librerías y dependencias necesarias en el proyecto de Android Studio.

6.2.2 Herramientas de Soporte

- **Microsoft Word:** herramienta de Microsoft utilizada para editar y procesar textos durante el proyecto. En este caso se ha utilizado el equipamiento de “Word 365” para administrar y gestionar las diferentes versiones elaboradas de la documentación.
- **Diagrams.net:** es una aplicación web gratuita y de código abierto que te permite crear una gran variedad de diagramas desde cualquier navegador web. En nuestro proyecto hemos utilizado esta herramienta a través de la plataforma “Google Drive”, ya que además nos permitía almacenar los distintos diagramas creados en la “nube”.
- **GitHub:** es una compañía sin ánimo de lucro que ofrece un servicio de hosting de repositorios almacenados en la nube. En nuestro proyecto se ha empleado para salvaguardar las distintas versiones desarrolladas de nuestro sistema.
- **Adobe Reader:** herramienta perteneciente a la compañía Adobe Systems que permite la creación y gestión de archivos en formato PDF. En nuestro caso se ha utilizado para obtener una visualización y lectura más eficiente de los archivos de texto.
- **PyCharm:** es un entorno de programación en lenguaje Python, que en el caso de nuestro proyecto se ha empleado para generar los scripts necesarios para realizar el “scrapping” sobre los datos de los sitios web y para realizar las llamadas al API “SportsRadar”. También se ha empleado para conectarse con la base de datos “Cloud Firestore” y volcar los respectivos datos en su localización correspondiente.
- **SportsRadar API:** es un API de origen americano que nos ofrece la posibilidad de realizar diversas solicitudes para obtener acceso a contenido de numerosos deportes. En el caso de nuestro proyecto hemos utilizado el API correspondiente al ciclismo, el cual nos proporciona acceso al calendario de las carreras que se disputan en cada temporada, junto con los resultados de estas.

6.3 Tecnologías empleadas

- **Java:** es un lenguaje de programación que se utiliza para desarrollar sistemas de software capaces de ejecutarse en distintas plataformas. En nuestro caso ha servido para implementar toda la lógica de negocio de la aplicación.
- **Python:** es un lenguaje de programación versátil, multiplataforma y multiparadigma que destaca por su código legible y limpio. En nuestro proyecto se ha empleado de manera externa para la gestión de los datos y su posterior carga en la base de datos, cuya finalidad es la visualización posterior por parte del usuario.

- **XML (Extensible Markup Language):** es un conjunto de reglas para codificar documentos que contienen información estructurada, y que así pueda ser compatible con los dispositivos electrónicos. En nuestro proyecto estos archivos representan las vistas que se van a mostrar al usuario.

6.4 Consideraciones de Implementación

En esta sección se van a reflejar las consideraciones previas y cuestiones clave que se han seguido para el desarrollo de nuestro sistema.

En primer lugar, la aplicación debe ser capaz de cumplir todas las funcionalidades y características que se habían planteado al principio del proyecto, en las cuales el objetivo final (de la mayoría) radica en la visualización de diversa información por parte de los usuarios. Por lo tanto, la aplicación deberá ser capaz de acceder a la información correspondiente que se encuentra almacenada en la base de datos, a la cual podremos acceder gracias al servicio “Cloud Firestore” proporcionado por la plataforma Firebase. El funcionamiento de este servicio se explica con detalle en el apartado “**6.5 Servidores de bases de datos utilizados**” de esta documentación.

Las funcionalidades que emplean el mecanismo anterior son:

- Gestión de Carreras
- Gestión Equipos
- Gestión Corredor
- Gestión Ranking

Por otro lado, se encuentran el resto de las funcionalidades que emplean un mecanismo independiente cada una de ellas:

- **Portal de Noticias:** en este caso el sistema simplemente accede a un sitio web para consultar noticias (sin salir de la aplicación), así que no requiere de nada especial más que la conexión con el sitio web empleando el protocolo de comunicación segura HTTPS.
- **Compartir aplicación:** en este caso el usuario puede compartir con otros usuarios el enlace de descarga de la aplicación (desde PlayStore) a través de las diferentes vías que disponga en su dispositivo (por ejemplo, Whatsapp o Gmail). La única consideración de implementación que hay que tener en cuenta es que el sistema debe estar subido al servicio de Google “Play Store” anterioridad para que el enlace de descarga funcione adecuadamente.
- **Envío Comentarios:** en este caso el usuario puede enviar comentarios, dudas o sugerencias al respecto de la aplicación al encargado de soporte de esta. En este caso hay que mencionar que la aplicación utiliza el API “JavaMail” que posibilita el envío de dicho mensaje (sin salir de la aplicación) y gestiona la recepción de este en la dirección electrónica especificada por el administrador de manera intrínseca. En nuestro sistema el proveedor de correo electrónico utilizado es el de Gmail.

- **Gestión de Favoritos:** en este caso el funcionamiento consiste en que al momento de iniciar la aplicación por primera vez el usuario no dispone de ningún corredor ni equipo como favorito. Una vez el usuario haya guardado como favorito un equipo y/o corredor en su aplicación, cada vez que acceda a la sección “Favoritos” desde la aplicación dichos elementos estarán disponibles. El mecanismo de implementación simplemente consiste en que al guardar un determinado elemento como favorito se genera un archivo de preferencias (a través del API SharedPreferences) en la propia aplicación, que se recupera cada vez que el usuario accede a dicha sección.

Por último, faltaría por mencionar los detalles de implementación correspondientes a la manera de obtener y cargar los datos de la aplicación en la base de datos. Todas estas cuestiones se explican de manera exhaustiva en la sección “**6.6 Técnicas de obtención de datos**” que aparecerá más adelante.

6.5 Servidores de bases de datos utilizados

Como ya he mencionado anteriormente, para todas las cuestiones referentes a la gestión y administración de los datos se ha empleado el servicio “**Cloud Firestore**” (Firebase).

Gracias a sus características favorables (ya mencionadas en la sección “**6.2 Herramientas utilizadas**”) y al tipo de datos que se van a almacenar en nuestro proyecto se ha optado por esta opción, ya que nos proporciona gran flexibilidad e independencia entre los conjuntos de datos.

Por lo tanto, en el caso de nuestro proyecto (que va a almacenar datos relativos a: carreras, clasificación de corredores, resultados de carreras, equipos y corredores) la estructura en la que distribuye los datos este servicio nos facilita mucho las cosas.

En este servicio los datos se almacenan en estructuras denominadas “**colecciones**”, en las cuales cada registro diferente se corresponde con un “**documento**”, y es ahí donde se almacenan sus respectivos campos. De igual manera, cada documento puede almacenar a su vez su propia “**sub-colección**”, lo cual nos ha permitido distribuir nuestros datos de la manera que mejor nos convenía e interesaba en cada caso.

Pero el motivo principal por el que me he decantado por este sistema es su ubicación en la nube, ya que me otorgaba la posibilidad de actualizar los datos en la base de datos y que los usuarios de la aplicación tengan constancia en tiempo real de dichas modificaciones.

Primordialmente, la función principal de la aplicación “**App4CyclingLovers**” es la de aportar visualización de contenido al usuario, así que la facilidad y sencillez que nos ofrece este servicio para garantizarla es incomparable con ningún otro procedimiento.

Una vez explicado el sistema de base de datos empleado, es necesario detallar todo el proceso de integración de nuestro proyecto con este servicio. En este caso la integración con Android Studio se consigue con gran sencillez a través de una serie de pasos, los cuales son:

- Creación del proyecto dentro de la plataforma Firebase.
- Integración del SDK de Cloud Firestore en la aplicación.
- Procedimientos de recuperación de datos.
- Configuración de reglas de seguridad en Cloud Firestore.

A continuación, vamos a detallar el proceso seguido en cada uno de ellos.

1. Creación del proyecto en Firebase

- Accedemos a la plataforma desde el navegador a través de <https://firebase.google.com/?hl=es>
- Una vez allí seleccionamos la opción “Ir a la consola”.
- Accedemos con nuestras credenciales de autenticación.
- En este punto se ofrece la posibilidad de añadir un nuevo proyecto, o acceder a uno ya existente. En nuestro caso ya disponemos del nuestro (ilustración 17).

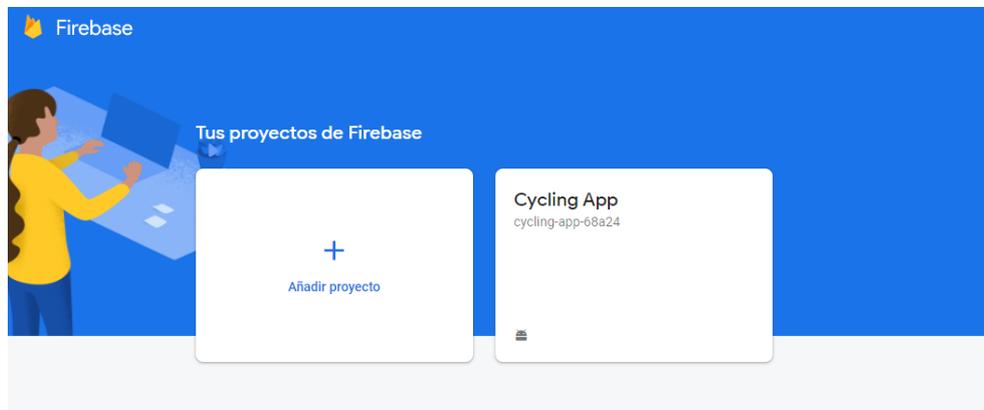


Ilustración 17 Proyectos Cuenta Firebase

- En cambio, si se quiere añadir un nuevo proyecto, escogemos la opción de “Añadir proyecto” en la cual deberemos establecer un nombre para el proyecto y otras cuestiones generales para poder completar el proceso.



Ilustración 18 Creación Nuevo Proyecto Firebase 1



Ilustración 19 Creación Nuevo Proyecto Firebase 2

- Suponemos que ya nos encontramos dentro de la consola de nuestro proyecto, en la cual se puede acceder a cualquiera de los servicios que ofrece la plataforma desde un menú lateral, cuyos resultados se hacen visibles en la pantalla principal. En nuestro caso, el servicio que nos interesa es el de Cloud Firestore.

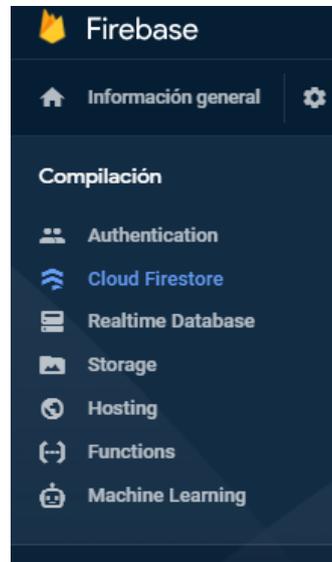


Ilustración 20 Servicios disponibles Firebase

2. Integración del SDK de Cloud Firestore en la aplicación

- Para comenzar, seleccionamos el icono de Android desde la pantalla principal de la plataforma para iniciar el proceso.



Ilustración 21 Integración Cloud Firestore en la aplicación (1)

- En primer lugar debemos establecer el nombre del paquete de Android de nuestra aplicación (se puede encontrar en el archivo build.gradle de nuestro proyecto en Android Studio), junto a unos campos opcionales que en nuestro caso preferimos dejar sin contenido.

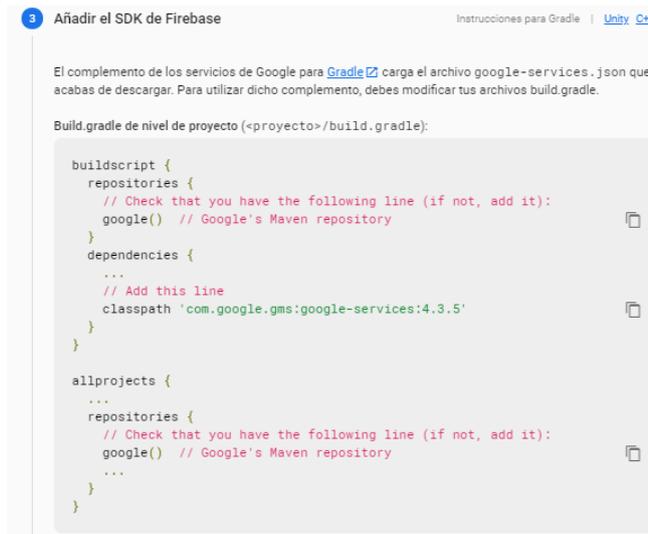


Ilustración 24 Integración Cloud Firestore en la aplicación (4.1)

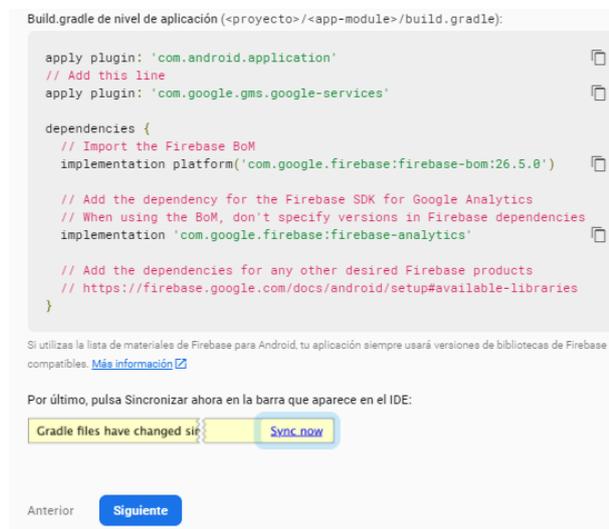


Ilustración 25 Integración Cloud Firestore en la aplicación (4.2)

3. Procedimientos de recuperación de datos

En este punto, ya disponemos de nuestra aplicación perfectamente vinculada con la plataforma Firebase, y con el servicio Cloud Firestore por extensión, por lo tanto, vamos a describir la manera en que obtenemos los datos que nos interesan en nuestra aplicación.

Lo primero que hay que hacer es crear una instancia de nuestra base de datos en todas las funcionalidades que requieran de sus servicios. Esto se deberá incluir en los métodos onCreate() y onCreateView() en función de si se trata de un Activity o un Fragment.

La instancia se crea de la siguiente manera:

- **Private FirebaseFirestore miFirestore = FirebaseFirestore.getInstance();**

A partir de este momento, ya se puede interactuar en cualquier momento con la base de datos.

Para las distintas funcionalidades de nuestro proyecto, hemos empleado diversos métodos a la hora de recuperar un conjunto de datos u otro, en función de las necesidades requeridas. En este punto es necesario precisar qué el fundamento básico de recuperación reside en la creación de objetos personalizados capaces de almacenar la tipología variable de los elementos almacenados en la nube. Por lo que en la mayoría de los casos el mecanismo utilizado ha sido construir una Lista de Objetos ("ArrayList") que almacene los elementos buscados y recuperados por **Firestore** desde una colección específica de la base de datos para posteriormente mostrárselos al usuario.

El ejemplo de un método genérico de recuperación sería el siguiente:

```
private void llenarCorredoresList(){
    corredoresList= new ArrayList<>();
    miFirestore = FirebaseFirestore.getInstance();
    //IMPLEMENTACIÓN PARA CARGAR TODA LA LISTA DE CORREDORES DE LA BASE
    DE DATOS EN LA ARRAYLIST
    CollectionReference list= miFirestore.collection("listRiders");
    list.get().addOnSuccessListener(queryDocumentSnapshots -> {
        if(!queryDocumentSnapshots.isEmpty()){
            for (QueryDocumentSnapshot riders: queryDocumentSnapshots){
                Corredores c= riders.toObject(Corredores.class);
                corredoresList.add(c);
            }
        }
    });
}
```

En este caso, se recupera toda la lista de corredores almacenados en la colección "listRiders" de la base de datos (cada uno de los Corredores corresponde a un objeto diferente recuperado desde Firestore), que se utilizará en la opción de "**Búsqueda de un Corredor**" para que el usuario realice un filtrado por el nombre del corredor.

Por otro lado, también se emplean métodos de búsqueda de ciertos elementos en función de unas condiciones que deben cumplir. Por ejemplo, sería el caso de recuperar las carreras que se disputan en un determinado mes:

```
Query q2 = miFirestore.collection("Calendar").document("Season2")
    .collection("races")
    .whereEqualTo("scheduled_mes",mes_fecha);

//AÑADIMOS LAS CARRERAS QUE CUMPLE LA CONSULTA ANTERIOR A LA ARRAYLIST DE
CARRERAS PARA MOSTRARLAS
q2.get().addOnSuccessListener(queryDocumentSnapshots -> {
```

```

if(!queryDocumentSnapshots.isEmpty()){
    for (QueryDocumentSnapshot riders: queryDocumentSnapshots){
        Carreras c= riders.toObject(Carreras.class);
        carrerasList.add(c);
    }
}
});
}

```

En este caso, el funcionamiento es similar al método anterior, la diferencia radica en que primero se busca en la colección de la base de datos (“Calendar”) todos los elementos que cumplen la condición evaluada con la sentencia “whereEqualTo”, para únicamente añadir dichos elementos a la lista de objetos que se va a mostrar al usuario.

El proceso de recuperación de todos los conjuntos de datos se basa en los procedimientos anteriormente descritos, por lo que resulta muy sencillo de gestionar y de administrar desde nuestra aplicación.

4. Configuración de reglas de seguridad en Cloud Firestore

Uno de los puntos más importantes para tener en cuenta con los datos almacenados en la base de datos de nuestro proyecto, radica en garantizar la completa seguridad de estos para que no puedan ser manipulados ni alterados indebidamente. En este caso este servicio cuenta con la ventaja de que se pueden establecer las reglas de seguridad que estimemos oportunas desde la propia ventana de Cloud Firestore, en la pestaña de “Reglas”.

En nuestro caso, como simplemente queremos permitir la lectura de los datos por parte de los usuarios de la aplicación y no disponemos de sistemas de autenticación establecemos las siguientes restricciones:

```

rules_version = '2';

service cloud.firestore {

    match /databases/{database}/documents {

        match /{document=**} {

            allow read: if true;

            allow write: if false;

        }

    }
}

```

6.6 Técnicas de Obtención de Datos

A raíz de lo descrito anteriormente, se deduce que la aplicación requiere de gran cantidad de información de ciclismo que sea completamente veraz y actual en todo momento. Por esta razón, la obtención de estos datos se convierte en la piedra angular de este proyecto y por ello se va a describir con gran lujo de detalles en esta sección.

En primer lugar, hay que precisar que la opción más recomendable en este tipo de casos es recurrir a un API ya existente que nos proporcione los datos requeridos para la aplicación. Para el caso de nuestro proyecto se encontró un API conocido como **“CyclingV2”** perteneciente a la compañía **“SportsRadar”**, pero no nos otorgaba acceso a la totalidad de las características y datos deseados. Por lo tanto, hemos tenido que recurrir a una técnica complementaria conocida como **“web scraping”** para disponer de los conjuntos de datos que nos faltaban.

Esta técnica consiste en realizar un raspado a un determinado sitio web (es decir leer sus textos) para extraer y almacenar parte de sus datos para analizarlos o utilizarlos en otra parte.

Existen distintas maneras de realizar el “scraping”, de manera automática (utilizando un software o algoritmo especializado para analizar varias páginas web para extraer información) o de manera manual, que es la opción por la que me he decantado en este proyecto.

El proceso manual requiere de un trabajo más laborioso e individualizado, ya que es necesario elaborar los mecanismos para acceder a un determinado sitio web y a un conjunto de datos concreto de manera específica y manualmente. Sin embargo, en el caso de este proyecto nos hemos decantado por esta metodología porque solamente se necesitaba acceder a datos específicos de un determinado sitio web por separado y no extraer grandes conjuntos de datos de forma concurrente.

Por último, antes de detallar en profundidad el procedimiento seguido, es importante precisar que esta técnica no siempre es recomendable (se encuentra en los límites de la legalidad), por lo que antes de realizarla es necesario tener en cuenta los derechos de propiedad intelectual de los sitios web analizados y el uso que se pretende dar a la información obtenida. Por lo tanto, para poder realizar esta técnica de manera ética se necesita que los datos recabados estén disponibles públicamente en la web sobre la que la realizas y no se puede comercializar el material obtenido como propio.

En el caso de nuestro proyecto, la web escogida ha sido **“worldcyclingstats.com”**, así que debemos informar de la procedencia de los datos mostrados en nuestra aplicación.

Una vez explicados los aspectos teóricos de la metodología empleada vamos a detallar el proceso seguido en la obtención de cada uno de los datos:

1. El lenguaje de programación empleado ha sido Python, a través del desarrollador **PyCharm**. Por lo que creamos un nuevo archivo “.py” y comenzamos a desarrollar el script para analizar el sitio web y extraer la información.

2. Incluimos la biblioteca **“Beautiful Soup”** de Python que nos permite analizar el documento HTML del sitio web, y así poder extraer la información de los elementos que nos interesaban.
3. Incluimos la biblioteca **“requests”** de Python para facilitar las peticiones HTTP sobre la página indicada y así poder obtener la información.
4. En función del tipo de elemento HTML que albergue el campo que queremos obtener, (por ejemplo <table>) buscamos la manera para acceder a cada etiqueta y extraer su valor para disponer de él en nuestro programa.
5. A continuación, se incluyen diferentes scripts para ejemplificar el proceso de análisis y extracción de datos.

```

2 import requests
3 import pandas as pd
4 from bs4 import BeautifulSoup
5
6 url = 'https://www.worldcyclingstats.com/teams/'
7 page = requests.get(url)
8 soup = BeautifulSoup(page.content, 'html.parser')
9
10 #Equipos
11 eq = soup.find_all('h3', class_='article-name')
12
13 equipos = list()
14 count = 0
15 for i in eq:
16     if count < 19:
17         equipos.append(i.text)
18     else:
19         break
20     count += 1
21
22 print(equipos, len(equipos))
23
24 df = pd.DataFrame({'Nombre': equipos}, index=list(range(1, 20)))
25 print(df)

```

Ilustración 26 Procedimiento Extracción Datos de " Equipos inscritos en la temporada"

En la ilustración 26 se puede observar claramente cada uno de los pasos comentados anteriormente. En este caso accede a las etiquetas <h3> que contienen cada uno de los nombres de los equipos (en función del valor de su atributo), y los incorpora a una lista para su posterior visualización. Dicha visualización se lleva a cabo gracias a la inclusión de la biblioteca “pandas” que genera un “Dataframe” (ilustración 27) con la lista obtenida.

	Nombre
1	AG2R Citroën Team
2	Astana-Premier Tech
3	Bahrain Victorious
4	Bora-Hansgrohe
5	Cofidis
6	Deceuninck-QuickStep
7	EF Education-Nippo
8	Groupama-FDJ
9	Ineos Grenadiers
10	Intermarché-Wanty-Gobert Matériaux
11	Israel Start-Up Nation
12	Lotto Soudal

Ilustración 27 Lista de Equipos en la Temporada 2021

```

1 import requests
2 from bs4 import BeautifulSoup
3
4 url = ''
5 count = 0 #CONTADOR PARA EVALUAR LA POSICIÓN DE CADA EQUIPO A INSERTAR
6
7 #Variables para almacenar los campos obtenidos de cada uno de los corredores
8 m = {}
9 corredores = []
10 while count < 19:
11     if count == 0:
12         url = 'https://www.worldcyclingstats.com/es/equipo/trek-segafredo'
13         r = requests.get(url, headers={"User-Agent": "Mozilla/5.0"})
14         soup = BeautifulSoup(r.text, 'html.parser')
15         co = soup.find('table', class_='large')
16         rows = co.find_all('tr')
17         for country in rows:
18             country_code = country.find('td', class_='code').text
19             edad = country.find_all('td', class_='')[2].text
20             nombre = country.find_all('td', class_='')[1].text.strip()
21             nombre_replace = nombre.replace(",",".")

```

Ilustración 28 Procedimiento Extracción Datos de " Corredores pertenecientes a un Equipo"

```

31 # CÓDIGO PARA MODIFICAR LA CADENA RECIBIDA CON EL NOMBRE DE CADA CORREDOR
32 # ANTES DE SUBIRLO A LA BASE DE DATOS
33 i = 0
34 def reemplazar_cadena(cadena, i):
35     while i < len(cadena):
36         if cadena[i].isupper():
37             cadena = cadena.replace(cadena[i], cadena[i].swapcase())
38         if (cadena[i] == 'o'):
39             cadena = cadena.replace(cadena[i], '0')
40         if (cadena[i] == 'c'):
41             cadena = cadena.replace(cadena[i], 'C')
42         if (cadena[i] == 'a'):
43             cadena = cadena.replace(cadena[i], 'A')
44         if (cadena[i] == 'C'):
45             cadena = cadena.replace(cadena[i], 'c')
46         if (cadena[i] == 'a'):
47             cadena = cadena.replace(cadena[i], 'A')
48         if (cadena[i] == 'e'):
49             cadena = cadena.replace(cadena[i], 'E')
50         if (cadena[i] == 's'):
51             cadena = cadena.replace(cadena[i], 'S')
52         if (cadena[i] == 'S'):
53             cadena = cadena.replace(cadena[i], 's')
54         i += 1
55     return cadena

```

Ilustración 29 Procedimiento Modificación Campos Extraídos (1)

```

56 # CÓDIGO PARA REEMPLAZAR LOS CARÁCTERES QUE CONTENGAN TILDE POR EL MISMO SIN TILDE
57 def normalize(s):
58     replacements = (
59         ("á", "a"),
60         ("é", "e"),
61         ("í", "i"),
62         ("ó", "o"),
63         ("ú", "u"),
64     )
65     for a, b in replacements:
66         s = s.replace(a, b).replace(a.upper(), b.upper())
67     return s
68 m['Nacionalidad'] = country_code
69 m['edad'] = edad
70 m['nombre'] = normalize(reemplazar_cadena(nombre_replace, i))
71 corredores.append(m.copy())

```

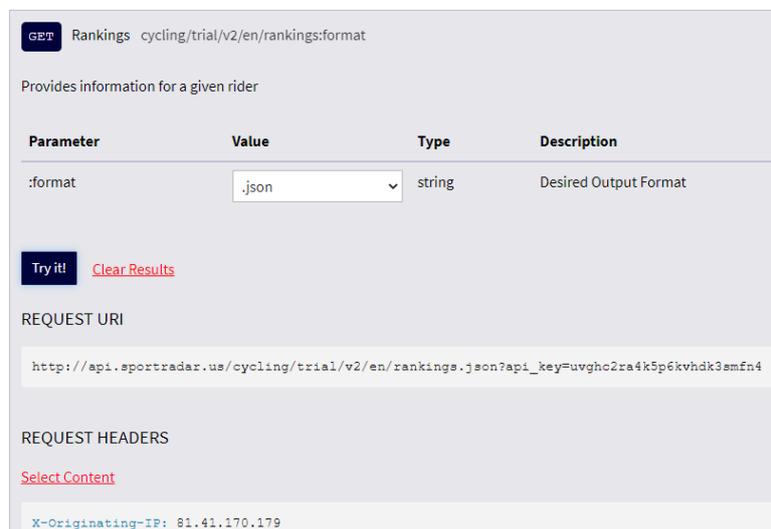
Ilustración 30 Procedimiento Modificación Campos Extraídos (2)

En la ilustración 28 se aprecia que el procedimiento seguido es el mismo, pero en este caso se busca extraer los campos almacenados en una tabla. El funcionamiento del código consiste en obtener todas las filas que se encuentran en la tabla específica, ya que cada una de ellas contiene un corredor perteneciente al equipo accedido junto a sus respectivos campos (país, edad, nombre). Una vez obtenidas, simplemente es necesario extraer los valores requeridos y almacenarlos en la posición correspondiente de una lista conjunta. En este caso además se incluyen dos métodos para modificar el valor de uno de los campos obtenidos (nombre) en función de nuestros intereses (ilustraciones 29 y 30).

En este punto podemos considerar que gracias a los ejemplos anteriores la extracción de datos mediante la técnica del “**web scraping**” queda convenientemente explicada, ya que han sido los conjuntos de datos de los que nos hemos nutrido para nuestro proyecto (Equipos y Corredores).

Pero por otro lado nos faltaría detallar el proceso de obtención de la información restante que se va a presentar a los usuarios de la aplicación, para el cual hemos empleado el API mencionado al comienzo de este apartado. Siguiendo el modelo anterior, en primer lugar, vamos a enumerar los pasos seguidos en este nuevo proceso:

1. Registro en la web www.sportsradar.com y solicitar acceso a la prueba gratuita de su API de ciclismo (“**Cycling V2**”) para poder acceder a los diferentes conjuntos de datos que ofrecen.
2. Realizar peticiones “GET” desde la propia consola de ejecución del API para comprobar los resultados que proporcionan en cada una de las categorías de datos (Calendario de Carreras, Información de Equipos, etc) (ilustraciones 31 y 32).



GET Rankings cycling/trial/v2/en/rankings:format

Provides information for a given rider

Parameter	Value	Type	Description
:format	.json	string	Desired Output Format

Try it! [Clear Results](#)

REQUEST URI

```
http://api.sportsradar.us/cycling/trial/v2/en/rankings.json?api_key=uvghc2ra4k5p6kvhdk3emfn4
```

REQUEST HEADERS

[Select Content](#)

```
X-Originating-IP: 81.41.170.179
```

Ilustración 31 Ejemplo Solicitud API "Sportsradar" (1)

```
RESPONSE BODY
Select Content

"rankings": [{
  "type_id": 6,
  "name": "Individual World Ranking",
  "year": 2021,
  "week": 10,
  "competitor_rankings": [{
    "rank": 1,
    "points": 4237,
    "ranking_movement": 0,
    "competitor": {
      "id": "sr:competitor:241198",
      "name": "Roglic, Primož",
      "abbreviation": "ROG"
    }
  }, {
    "rank": 2,
    "points": 3200,
    "ranking_movement": 0,
    "competitor": {
      "id": "sr:competitor:507514",
      "name": "Pogacar, Tadej",

```

Ilustración 32 Ejemplo Solicitud API "Sportsradar" (2)

3. Realizar la conexión del API con el desarrollador PyCharm, ya que ofrece la posibilidad de realizar las peticiones de manera sencilla a través de diferentes lenguajes y en nuestro caso como ya estábamos trabajando con Python nos resultaba más cómodo. Bastaría con incluir las siguientes sentencias en nuestro código en Python y adaptarlas a los valores requeridos en cada una de las categorías (ilustración 33):

```
import http.client

conn = http.client.HTTPSConnection("api.sportsradar.us")

conn.request("GET", "/cycling/trial/v2/en/rankings.xml?api_key={your_api_key}")

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Ilustración 33 Integración API "Sportsradar" con entorno "Pycharm".

4. Por último, faltaría por manejar la información proporcionada por el API, que puede ser suministrada a través de un archivo XML o en formato JSON. Este último es el modelo escogido ya que resultaba sencillo poder leerlo y extraer los campos que nos interesaban en cada caso.

Una vez explicado el procedimiento seguido, faltaría por presentar de una manera más visual los diferentes ejemplos desarrollados para obtener la información necesaria y así poder completar nuestra base de datos. Sin embargo, no hay que olvidar que se tratan de unos ejemplos ilustrativos para alcanzar una mayor comprensión, y no de los programas completos desarrollados para extraer y trabajar con los datos obtenidos.

```
1 import http.client
2 import json
3
4 conn = http.client.HTTPSConnection("api.sportradar.us")
5
6 conn.request("GET", "/cycling/trial/v2/en/sport_events/sr:stage:537016/schedule.json?api_key=uyghc2ra4k5p6kvhdK3smfn4")
7
8 res = conn.getresponse()
9 data = res.read()
10 #print(data.decode("utf-8"))
11
12 jsonToPython = json.loads(data)
13 position = 1
14
15 for client in jsonToPython['stages']:
16     print('id:', client['id'])
17     print('description:', client['description'])
18     print('scheduled:', client['scheduled'])
19     print('scheduled_end:', client['scheduled_end'])
```

Ilustración 34 Procedimiento Extracción Datos de "Calendario de Carreras de la Temporada"

En la (ilustración 34) se aprecia el sencillo funcionamiento de este método, simplemente bastaría con introducir en la petición "GET" la ruta proporcionada por el API (con los datos relativos a una determinada categoría y nuestra clave de acceso) y leer el archivo JSON que nos suministra para acceder a los campos que contienen cada una de las carreras de la temporada y poder trabajar con ellos como nos interese.

```
1 import http.client
2 import json
3
4 conn = http.client.HTTPSConnection("api.sportradar.us")
5
6 conn.request("GET", "/cycling/trial/v2/en/rankings.json?api_key=uyghc2ra4k5p6kvhdK3smfn4")
7
8 res = conn.getresponse()
9 data = res.read()
10 #print(data.decode("utf-8"))
11 jsonToPython = json.loads(data)
12 for client in jsonToPython['rankings']:
13     print('id:', client['type_id'])
14     print('name:', client['name'])
15     print('week:', client['week'])
16     print('year:', client['year'])
17     print('')
18     #print('rank:', client['competitor_rankings'])
19     for client in client['competitor_rankings']:
20         print('rank:', client['rank'])
21         print('points:', client['points'])
22         print('rank_move:', client['ranking_movement'])
23     print('name:', client['competitor']['name'].replace(","," "))
24     nombre_corredor = client['competitor']['name'].replace(","," ")
```

Ilustración 35 Procedimiento Extracción Datos de "Ranking Mundial de Corredores"

En este caso (ilustración 35), el funcionamiento es el mismo, pero para la categoría de datos específica. Simplemente incluye dos niveles de acceso en el documento JSON para obtener un conjunto de datos anidados.

```

1 import http.client
2 import json
3
4 conn = http.client.HTTPSConnection("api.sportradar.us")
5 # RESULTADOS CARRERA GLOBAL POR ETAPAS
6 conn.request("GET", "/cycling/trial/v2/en/sport_events/sr:stage:547421/summary.json?api_key=uyqhc2ra4k5p6kvhdK3smfn4")
7
8 res = conn.getresponse()
9 data = res.read()
10
11 print(data.decode("utf-8"))
12 jsonToPython = json.loads(data)
13 print('id:', jsonToPython['stage']['id'])
14 print('description:', jsonToPython['stage']['description'])
15 print('id:', jsonToPython['stage']['scheduled'])
16 print('id:', jsonToPython['stage']['scheduled_end'])
17 print('id:', jsonToPython['stage']['single_event'])
18 print('id:', jsonToPython['stage']['departure_city'])
19 print('id:', jsonToPython['stage']['arrival_city'])
20 print('id:', jsonToPython['stage']['distance'])
21 print('')
22

```

Ilustración 36 Procedimiento Extracción Datos de "Resultados de una carrera" (1)

```

23 count = 0
24 for client in jsonToPython['stage']['competitors']:
25     print('')
26     print('id:', client['id'])
27     print('id:', client['name'])
28     print('id:', client['country_code'])
29     print('id:', client['nationality'])
30     enlace = client['name']
31     if count < 133:
32         print('time:', client['result']['time'])
33         print('time_ranking:', client['result']['time_ranking'])
34         count = count+1
35
36 for client in jsonToPython['stage']['teams']:
37     print('')
38     print('id:', client['id'])
39     print('id:', client['name'])
40     print('id:', client['nationality'])
41     print('id:', client['country_code'])
42     print('team_time:', client['result']['team_time'])
43     print('team_time_ranking:', client['result']['team_time_ranking'])

```

Ilustración 38 Procedimiento Extracción Datos de "Resultados de una carrera" (2)

```

id: sr:competitor:123856
name: Bystroem, Sven Erik
nationality: Norway
country_code: NOR
time: +00:00.57
time_ranking: 11

id: sr:competitor:27620
name: Devenyns, Dries
nationality: Belgium
country_code: BEL
time: +00:00.57
time_ranking: 12

```

Ilustración 37 Procedimiento Extracción Datos de "Resultados de una carrera" (3)

En las ilustraciones anteriores (ilustraciones 36, 37 y 38) se pueden observar también cada uno de los pasos seguidos (y parte de la salida por pantalla), en este caso bastaría con incluir el identificador de una determinada carrera (obtenido de un listado del API) y el API ya se encarga de proporcionar los datos relativos a esa carrera, junto con la clasificación ocupada por los diferentes corredores y equipos.

Para finalizar, es necesario detallar el procedimiento seguido con todo el conjunto de datos obtenidos mediante estas dos técnicas para almacenarlos en la base de datos del proyecto ("Cloud Firestore") y que así puedan estar disponibles para los usuarios de nuestra aplicación:

1. Integración SDK de Cloud Firestore con el desarrollador de Python utilizado para obtener los datos (Pycharm).

Para ello simplemente hay que dirigirse al “Administrador SDK de Firebase” desde la consola del proyecto (**Configuración del proyecto>Cuentas de servicio**) y seleccionar el lenguaje de programación escogido para la configuración.

A continuación, se debe generar una nueva clave privada (archivo “.json”) e incluir dicho archivo en el desarrollador **Pycharm** junto a las sentencias indicadas para finalizar la integración (ilustración 39).

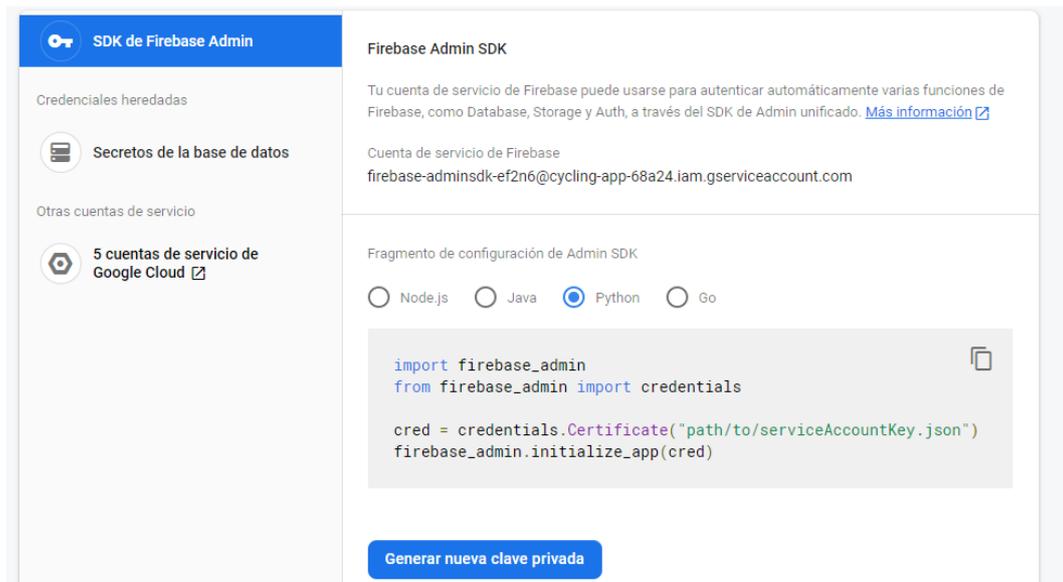


Ilustración 39 Integración Cloud Firestore con entorno Pycharm (1)

2. Generación de la base de datos con los datos extraídos.

Una vez que se ha logrado la integración y conexión con el servidor de la base de datos del proyecto, simplemente es necesario cargar los conjuntos de datos obtenidos en la base de datos. En este sentido la gran ventaja de “Cloud Firestore” radica en que no es necesario definir la estructura de datos a seguir con anterioridad, ya que “**Cloud Firestore**” crea colecciones y documentos de forma implícita la primera vez que agregas datos al documento. Para alcanzar una mayor comprensión del procedimiento mencionado se adjuntan algunos ejemplos utilizados para completar la base de datos del proyecto:

```

30     import firebase_admin
31     from firebase_admin import credentials, firestore
32
33     cred = credentials.Certificate("./serviceAccountKey.json")
34     firebase_admin.initialize_app(cred)
35     db = firestore.client()
36     position = 1
37     for corredor in corredores:
38         doc_ref = db.collection(u'listRiders').document(
39             u'corredor' + str(position))
40         doc_ref.set({
41             u'country_code': corredor['Nacionalidad'],
42             u'name': corredor['nombre'],
43             u'age': corredor['edad'],
44             u'team': 'Trek-Segafredo',
45             u'temporada': '2021'
46         })
47         int(position)
48         position = position + 1

```

Ilustración 40 Inserción en Cloud Firestore de "Corredores pertenecientes a un Equipo"

```

37     doc_ref = db.collection(u'rankingUCI').document(u'+str(client['rank']))
38     doc_ref.set({
39         u'rank': client['rank'],
40         u'prev': client['ranking_movement'],
41         u'name': nombre_corredor,
42         u'points': client['points'],
43         u'fecha': str(fecha)
44     })

```

Ilustración 41 Inserción en Cloud Firestore de "Ranking Mundial de Corredores"

En las figuras anteriores (ilustraciones 40 y 41) se aprecia como gracias a la inclusión de ciertas librerías y sentencias se puede generar una nueva colección en la base de datos (con el nombre que queramos), en la cual se van a agregar nuevos documentos independientes (cada uno con un identificador distinto) que contienen los campos obtenidos anteriormente. Dichos campos se definen a través del método **“.set()”**.

Este es el procedimiento empleado para la carga de todos los datos extraídos, por lo que en este punto doy por finalizada la descripción de todo el proceso de obtención y almacenamiento de los datos ofrecidos a los usuarios de la aplicación.

6.7 Organización Interna del Proyecto

En esta sección se describe la estructura organizativa interna del proyecto (en el entorno de programación), es decir la distribución de los archivos en sus respectivas carpetas, para facilitar la comprensión y localización de la implementación de cada una de las funcionalidades.

La estructura principal de la aplicación es la siguiente:

- **/App4CyclingLovers/app/src**: directorio principal, que a su vez está dividido en subdirectorios.
- **/App4CyclingLovers/app/src/androidTest** y **../Test**: archivos encargados de pruebas de test.
- **/App4CyclingLovers/app/src/main**: archivos con el código fuente y recursos de la aplicación.
- **/App4CyclingLovers/app/src/main/java/com.example.rodrigo.app4cyclinglovers**: archivos “.java” responsables de la lógica de negocio de la aplicación. A su vez los diferentes archivos están organizados en subdirectorios en función del objetivo de su implementación:
 - **/App4CyclingLovers/app/src/main/java/com.example.rodrigo.app4cyclinglovers/model**: contiene los archivos de soporte de las diferentes estructuras de datos utilizadas.
 - **/App4CyclingLovers/app/src/main/java/com.example.rodrigo.app4cyclinglovers/Adapters**: contiene los archivos responsables de generar las vistas (listados) a partir de los datos recibidos de la base de datos.
 - **/App4CyclingLovers/app/src/main/java/com.example.rodrigo.app4cyclinglovers/fragment**: contiene los archivos pertenecientes a una determinada actividad que se encargan de ampliar parte de la lógica utilizada para la navegación entre pantallas.
 - **/App4CyclingLovers/app/src/main/java/com.example.rodrigo.app4cyclinglovers/Activity**: contiene los archivos relativos a cada una de las actividades independientes presentes en la aplicación.
- **/App4CyclingLovers/app/src/main/java/AndroidManifest.xml**: archivo dónde se establecen los aspectos técnicos principales de la aplicación (actividades, estilos...)
- **/App4CyclingLovers/app/src/main/res**: contiene los archivos .xml en los cuales se definen cada una de las vistas de la aplicación (carpeta layout). Además, también existen diversas carpetas que contienen los numerosos tipos de recursos empleados en la aplicación (animaciones, colores, estilos, imágenes, iconos...).

7.PRUEBAS

7.1 Pruebas de Caja Blanca

El objetivo de este tipo de pruebas es validar las funciones internas de los módulos o subprogramas que se van a probar.

El funcionamiento consiste en coger porciones de código y dividirlos en bloques, es decir cada bloque se trata como un proceso independiente y que se realiza de manera secuencial. Con esto se consigue rastrear todos los flujos de actuación que puede adoptar un programa para asegurarnos de que realiza las acciones de la forma esperada.

En nuestro proyecto las pruebas realizadas durante todo el proceso de implementación han sido constantes, de las cuales se pueden mencionar:

- Comprobación de las peticiones a la base de datos en tiempo de ejecución.
- Comprobación del funcionamiento del sistema sin conexión a internet.
- Comprobación de todas las funcionalidades mostradas al usuario.
- Comprobación del funcionamiento del sistema en caso de que los datos a mostrar no se encuentren disponibles.
- Comprobación de la adecuación de las interfaces relativas a las diferentes funcionalidades.
- Comprobación de la similitud de los datos recuperados de la base de datos y los mostrados al usuario.
- Comprobación de la conexión y comunicación del sistema con los agentes externos participantes del sistema (herramientas complementarias).
- Comprobación del correcto funcionamiento de los archivos de preferencia de la aplicación almacenados en el propio dispositivo.
- Comprobación de la comunicación entre el sistema y el proveedor de correo electrónico utilizado para que el usuario envíe sus comentarios al administrador de la aplicación.

7.2 Pruebas de Caja Negra

El objetivo de este tipo de pruebas es validar las respuestas o salidas que el sistema produce al introducirle unas determinadas entradas. En ninguna de ellas se mira el funcionamiento interno del sistema, así que pueden ser realizadas por usuarios finales de la aplicación.

En el caso de este proyecto, el procedimiento empleado ha sido suministrar el sistema a dos personas interesadas en este tipo de herramientas para que se encargaran de ejecutar y probar todas las funcionalidades y posteriormente comunicar los resultados obtenidos. Sin embargo, hay que tener en cuenta que la función principal del sistema es mostrar información al usuario, así que no se producen grandes entradas de información por parte de este.

A continuación, se especifican algunos de los resultados más destacados (tablas 80 hasta 89):

PCN-01: Selección Parámetros Fecha del Calendario	
Propósito	Comprobar que el sistema actualiza la pantalla para consultar las carreras con la fecha seleccionada por el usuario en el calendario desplegado.
Prerequisitos	Ninguno
Datos de entrada	Año:2020 Fecha: 18 de septiembre
Resultado esperado	Al pulsar "Aceptar" el sistema cierra el Calendario y la pantalla principal se actualiza correctamente con la fecha seleccionada.
Resultado obtenido	Correcto

Tabla 80 Prueba de Caja Negra: "Seleccionar fecha Calendario"

PCN-02: Buscador de Corredores	
Propósito	Comprobar que el filtro de búsqueda de corredores funciona adecuadamente.
Prerequisitos	Ninguno, aunque para una carga rápida y eficiente de los datos es preciso estar conectado a Internet.
Datos de entrada	Filtro de Búsqueda: Stuyven
Resultado esperado	El sistema despliega un listado con los elementos que cumplen las condiciones introducidas en el buscador.
Resultado obtenido	Correcto

Tabla 81 Prueba de Caja Negra: "Buscador de Corredores"

PCN-03: Buscador de Carreras	
Propósito	Comprobar que el filtro de búsqueda de carreras funciona adecuadamente.
Prerequisitos	Ninguno, aunque para una carga rápida y eficiente de los datos es preciso estar conectado a Internet.
Datos de entrada	Filtro de Búsqueda: Tour
Resultado esperado	El sistema despliega un listado con los elementos que cumplen las condiciones introducidas en el buscador.
Resultado obtenido	Correcto

Tabla 82 Prueba de Caja Negra: "Buscador de Carreras"

PCN-04: Envío de comentarios	
Propósito	Comprobar que el sistema permite redactar un nuevo mensaje para enviar al administrador de la aplicación.
Prerequisitos	El dispositivo deberá estar conectado a internet.
Datos de entrada	Asunto: Fallo en los datos de la Carrera Mensaje: Descripción del problema Casilla de respuesta: Marcar en caso de querer recibir una respuesta personalizada. Dirección de Correo: email al que deseas recibir la respuesta (ej: pcb@gmail.com).
Resultado esperado	Al pulsar "Enviar" el sistema debe mostrar una alerta informando del éxito en el envío del mensaje, el cual debe ser recibido correctamente en la cuenta del administrador.
Resultado obtenido	Correcto

Tabla 83 Prueba de Caja Negra: "Envío de comentarios"

PCN-05: Acceso a sitio web Noticias	
Propósito	Comprobar que el sistema muestra correctamente el sitio web de noticias seleccionado por el usuario.
Prerequisitos	El dispositivo deberá estar conectado a Internet.
Datos de entrada	Selección "Icono" sitio web
Resultado esperado	La aplicación redirige al usuario a una nueva pantalla que muestra el sitio web que ha seleccionado.
Resultado obtenido	Correcto

Tabla 84 Prueba de Caja Negra: "Acceso a web de noticias"

PCN-06: Compartir Aplicación	
Propósito	Comprobar que el sistema muestra correctamente los medios disponibles para compartir la aplicación con otros usuarios.
Prerequisitos	El dispositivo deberá estar conectado a Internet. El dispositivo deberá contar con aplicaciones complementarias para comunicarse con otros usuarios (WhatsApp, Gmail...).

Datos de entrada	Selección "Icono" aplicación para compartir.
Resultado esperado	La aplicación despliega en el dispositivo la nueva aplicación que ha seleccionado el usuario, e incorpora en ella el enlace de descarga de la aplicación. Una vez enviado el sistema debe mantenerse en el estado anterior.
Resultado obtenido	Correcto, aunque como todavía la aplicación no está subida en la tienda oficial de Android el enlace de descarga no funciona adecuadamente.

Tabla 85 Prueba de Caja Negra: "Compartir Aplicación"

PCN-07: Almacenar Elemento Favorito	
Propósito	Comprobar que el sistema guarda correctamente los archivos de preferencias de la aplicación con los favoritos actualizados por el usuario.
Prerequisitos	Ninguno
Datos de entrada	Selección del icono "corazón" asociado al elemento visualizado en pantalla.
Resultado esperado	La aplicación redirige al usuario a una nueva pantalla que ofrece la posibilidad de consultar los elementos favoritos actuales, y modificar el elemento correspondiente (corredor o equipo) por el favorito seleccionado. Una vez guardado, se muestra en pantalla el mensaje de confirmación y el favorito actualizado.
Resultado obtenido	Correcto

Tabla 86 Prueba de Caja Negra: "Almacenar elemento favorito"

PCN-08: Consultar Listado Equipos	
Propósito	Comprobar que el sistema muestra correctamente el listado con los equipos pertenecientes a una temporada.
Prerequisitos	Ninguno, aunque para una carga rápida y eficiente de los datos es preciso estar conectado a Internet.
Datos de entrada	Selección botón "Temporada".
Resultado esperado	La aplicación muestra un listado al usuario con los nombres de los equipos

	que participan en las carreras de la temporada escogida.
Resultado obtenido	Correcto

Tabla 87 Prueba de Caja Negra: "Consultar listado Equipos"

PCN-09: Acceso a sitio web del Corredor	
Propósito	Comprobar que el sistema despliega automáticamente el navegador del dispositivo con el sitio web cargado del corredor seleccionado por el usuario.
Prerequisitos	El dispositivo deberá estar conectado a Internet.
Datos de entrada	Selección botón "Consultar sitio web".
Resultado esperado	La aplicación despliega el navegador del dispositivo con el sitio web del corredor abierto.
Resultado obtenido	Correcto, aunque en algunas ocasiones pueden producirse fallos al cargar la URL con el nombre del corredor, debido a incompatibilidades entre los datos almacenados del corredor y el proveedor del sitio web.

Tabla 88 Prueba de Caja Negra: "Acceder web del corredor"

PCN-10: Acceso a Detalles Carrera	
Propósito	Comprobar que el sistema muestra al usuario los datos correctos de la carrera que ha seleccionado.
Prerequisitos	Ninguno, aunque para una carga rápida y eficiente de los datos es preciso estar conectado a Internet.
Datos de entrada	Seleccionar una determinada carrera de las desplegadas en pantalla.
Resultado esperado	La aplicación redirige al usuario a una nueva página que contiene todos los datos posibles a conocer de la carrera accedida.
Resultado obtenido	Correcto, aunque en el caso de que la carrera no se haya disputado es posible que algunos datos no se muestren por pantalla.

Tabla 89 Prueba de Caja Negra: "Acceso a Detalles Carrera"

Cómo ya se ha mencionado anteriormente, las pruebas descritas representan sólo una parte de la totalidad de las realizadas, ya que el funcionamiento de las características no incluidas es similar, por lo que las pruebas y resultados obtenidos no aportan nada nuevo.

8.MANUAL DE INSTALACIÓN

En esta sección se van a enumerar las pautas y pasos a seguir en el proceso de descarga e instalación del sistema (es necesario disponer del espacio de almacenamiento suficiente en el dispositivo), para así disponer de todas las funcionalidades que ofrece al alcance de la mano.

En este sentido se van a considerar dos posibles escenarios para tener acceso a la descarga de la aplicación:

- La aplicación se encuentra convenientemente subida en la tienda oficial de “Android” para su posterior descarga gratuita.
- La aplicación no se encuentra todavía subida en la tienda oficial de “Android” para su descarga gratuita.

Una vez se ha completado el proceso de instalación, la facilidad de uso del sistema no requiere ningún trabajo complementario de aprendizaje por parte del usuario.

8.1 Instalación a través de la Tienda Oficial

1. En este escenario el acceso a la descarga de la aplicación se puede conseguir de manera muy sencilla. Simplemente es necesario iniciar el servicio “**Play Store**” presente en todos los dispositivos con sistema operativo “Android” y acceder con tus credenciales de acceso de Google.

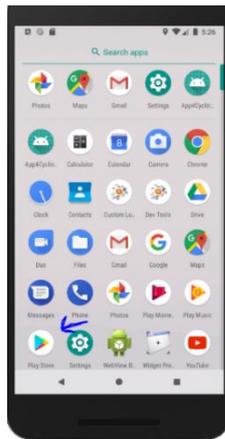


Ilustración 43
Instalación App desde
Tienda Oficial (1)

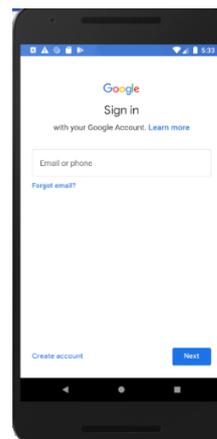


Ilustración 42
Instalación App desde
Tienda Oficial (2)

- Una vez dentro simplemente será necesario introducir el nombre de la aplicación **“App4CyclingLovers”** en el buscador presente en la parte superior, y seleccionar la aplicación deseada de la lista desplegada. (En este caso como todavía no se encuentra subida adjuntamos el ejemplo con otra aplicación distinta).

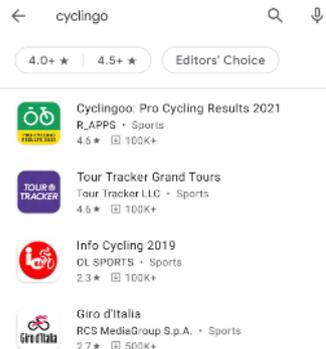


Ilustración 44 Instalación App desde Tienda Oficial (3)

- Una vez escogida la aplicación, simplemente hay que seleccionar el botón **“Instalar”** presente en la pantalla y al cabo de unos segundos la aplicación se habrá descargado correctamente en nuestro dispositivo. Para poder disfrutar de todas sus funcionalidades únicamente hay que localizar el icono representativo de nuestra aplicación e iniciarla.

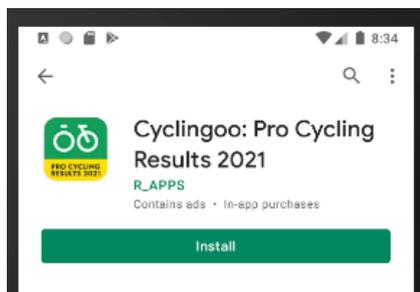


Ilustración 46 Instalación App desde Tienda Oficial (4)

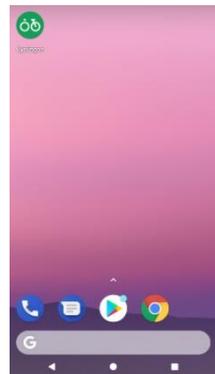


Ilustración 45 Instalación App desde Tienda Oficial (5)

8.2 Instalación a través del Archivo de Ejecución

- En este escenario, en primer lugar, hay que activar los permisos necesarios para instalar en el dispositivo aplicaciones de origen desconocido (todas las que no provienen de la tienda oficial “Android”). Esto se realiza desde los Ajustes del dispositivo, en la sección “Seguridad”.
- Introducir el archivo .apk de la aplicación en el dispositivo:
 - A través de un cable USB conectado al ordenador que contiene dicho archivo.
 - Descargando el archivo que nos han enviado directamente en nuestro teléfono.
 - Descargando el archivo que puede estar ubicado en un repositorio en la nube, al que nos han compartido el acceso.

3. Una vez disponemos del archivo de ejecución, simplemente hay que iniciarle para comenzar el proceso de instalación (Si el medio a través del que estamos iniciando la ejecución todavía no tiene los permisos de instalación para app desconocidas habilitados nos lo volverá a solicitar).

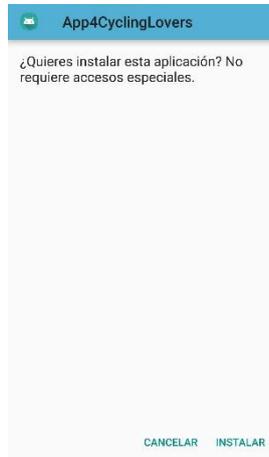


Ilustración 47 Instalación App a través de archivo ejecución (1)

4. Al cabo de un rato, la instalación se habrá completado y podremos abrir la aplicación en nuestro dispositivo para disfrutar de todas sus características. Es posible que durante la utilización de alguna funcionalidad nos soliciten permisos para acceder a otras herramientas del dispositivo (por ejemplo, el navegador de Internet).

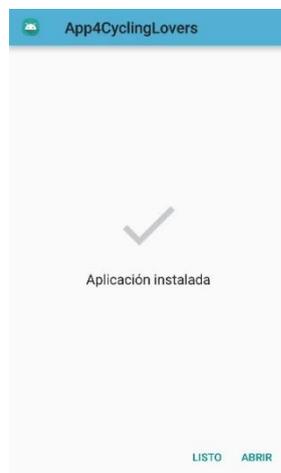


Ilustración 49 Instalación App a través de archivo ejecución (2)



Ilustración 48 Instalación App a través de archivo ejecución (3)

9.CONCLUSIONES Y POSIBLES MEJORAS

9.1 Conclusiones

En primer lugar, la cuestión más destacada que me gustaría relatar es que la falta de conocimiento en el desarrollo de sistemas en el entorno de programación “Android” me ha ocasionado un trabajo de aprendizaje más grande del esperado en un principio. Todo esto engloba la cantidad de errores y problemas que han ido surgiendo, los cuales me han producido gran frustración porque parecía que no había manera de solucionarlos. Sin embargo, gracias a todas las fuentes de información relativas a este tipo de desarrollos presentes en el mercado y a medida que el proyecto ha ido avanzando las dificultades se han ido solventando.

Por otro lado, habría que destacar la novedad que ha supuesto para mis los mecanismos empleados en todo el proceso de obtención y administración de los datos de la aplicación (**Scraping y Cloud Firestore**). Con anterioridad ni siquiera había oído hablar de los conceptos mencionados, por lo que la etapa de comprensión y su posterior uso eficiente han supuesto un gran reto para mí. Pero en este sentido me alegro de haber escogido trabajar con elementos desconocidos y laboriosos, para así poder ampliar el rango de conocimientos y capacidades adquiridas durante el periodo de formación.

Cambiando de dinámica y dejando a un lado los aspectos más técnicos y de aprendizaje, también sería importante destacar los cambios emocionales (además de la situación mundial provocada por la pandemia del virus “COVID 19”) experimentados durante todo el proceso de desarrollo. Dichas situaciones han provocado que no pudiera seguir un progreso lineal en cada uno de los meses de trabajo, ocasionando a su vez un incremento en los plazos de finalización planificados en un principio. En este sentido, otra cuestión que quiero destacar es que al principio del proyecto decidí dedicarme exclusivamente al desarrollo del proyecto sin ningún tipo de ocupación laboral, y a la larga he comprobado que fue una decisión correcta.

Por último, me gustaría concluir que, a pesar de los aspectos negativos mencionados la sensación general final de este proyecto es muy positiva, ya que me ha permitido generar un gran producto perteneciente al “ámbito” del ciclismo (del que yo soy un gran aficionado) y al mismo tiempo ampliar mis conocimientos y experiencias. Esta última cuestión es la que me genera mayor satisfacción, ya que al plantearme este proyecto tenía muchas dudas sobre si iba a ser capaz de desarrollar un proyecto de tales características y en una tecnología tan desconocida para mí. Por lo que el hecho de haberlo podido concluir en una distribución temporal asumible me hace sentirme muy orgulloso de mi mismo, y poder mirar hacia el futuro laboral con mayor optimismo.

9.2 Posibles mejoras

A medida que se ha ido avanzando en el desarrollo del proyecto, y el producto iba dotándose de mayor virtud, se nos iban planteando nuevas ideas o sugerencias que podrían resultar muy satisfactorias para el sistema. Sin embargo, para esta primera versión se ha priorizado avanzar en el desarrollo de la funcionalidad mínima planteada al principio del proyecto antes de entretenerse en una determinada característica que pudiera retrasar los plazos de entrega del proyecto. Por lo tanto, lamentándolo mucho dichas ideas tuvieron que ser desechadas.

A continuación, se describen algunas de las mejoras posibles que se pueden incorporar a la aplicación en próximas versiones del producto:

- Incorporación de más corredores y equipos como favoritos por parte del usuario.
- Ampliación del archivo histórico de carreras, equipos y corredores.
- Incorporación de notificaciones de los resultados obtenidos por los corredores y equipos escogidos por el usuario.
- Incorporación de notificaciones en tiempo real del comienzo o finalización de un evento.
- Incorporación de mapas con los recorridos por los que transcurren las carreras.
- Acceso a retransmisión o comentarios de las carreras que dispongan de dicho soporte.
- Establecer un chat de comunicación entre diferentes usuarios de la aplicación para opinar sobre el desarrollo y resultados de las carreras.

10. REFERENCIAS

En esta sección se van a enumerar y precisar las diferentes fuentes de información consultadas y empleadas durante la realización de este proyecto.

En primer lugar, hay que mencionar los materiales proporcionados (y desarrollados) en las diferentes asignaturas cursadas en el Grado de Ingeniería Informática de Servicios y aplicaciones como pueden ser:

- Modelado del Software y Sistemas de Información.
- Plataformas Software Empresarial.
- Gestión de Proyectos Basados en las TI.
- Aplicaciones móviles.
- Gestión de Proyectos basados en las TI.
- Administración de bases de datos.
- Protocolos y Comunicaciones Seguras (Python).

Por otro lado, hay que considerar el resto de las fuentes de información utilizadas para el desarrollo del sistema, las cuales se van a distribuir en tres categorías bien diferenciadas:

- **Bibliografía:** conjunto de referencias sobre las publicaciones consultadas, por ejemplo, las publicaciones disponibles en la biblioteca de la Universidad de Valladolid.
- **Webgrafía:** artículos de información consultados en los diferentes sitios web accedidos. En este apartado no se incluye la totalidad de los sitios web consultados, ya que algunos se engloban dentro de la misma fuente de información, por lo que se ha optado por mencionar los sitios web principales en cada caso.
- **Materiales Audiovisuales:** contenido audiovisual disponible en Internet, empleado para familiarizarme con el entorno de programación y comprender las diferentes cuestiones de implementación aplicadas en el proyecto.

10.1 Bibliografía

- **Desarrollo de aplicación Android:** Inmaculada Cárcel Puig. Edición: 2020. Editor: Universitat de Valencia. ISBN: 9788491332862.
- **Desarrollo de aplicaciones Android con Java:** Miguel Ángel Lozano Ortega, Antonio Javier Gallego Sanchez. Edición: 2017. Editor: Grupo Editorial Ra-Ma. ISBN: 9788499647005
- **Website Scraping with Python Using BeautifulSoup and Scrapy:** Hajba Gábor László. Edición: 1st edition 2018. Editor: Apress (Berkeley) ISBN: 1-4842-3925-3
- **Firestore, trabajar en la nube:** Jesús Tomás (coautor). Edición: 2018. Editor: Marcombo. ISBN: 978-84-267-2660-5

10.2 Webgrafía

- **Stack Overflow:** plataforma pública utilizada como punto de encuentro para todos los desarrolladores, que incluye innumerables “preguntas y respuestas” relativas a cualquier cuestión y dificultad surgida durante el proceso de implementación y codificación. Último acceso: 13/04/2021
Disponible en: <https://stackoverflow.com/>
- **Android Developers:** sitio oficial para el desarrollo de aplicaciones Android.
Último acceso: 05/04/2021
Disponible en: <https://developer.android.com/>
- **Coding in Flow:** Último acceso: 12/04/2021
Disponible en: <https://codinginflow.com/>
- **GitHub:** repositorio de código en el que pueden encontrarse proyectos públicos con características similares al nuestro. Último acceso: 12/04/2021
Disponible en: <https://github.com/>
- **Cloud Firestore (Firebase):** documentación relativa a todas las cuestiones de integración e implementación del servicio en nuestro sistema.
Último acceso: 12/04/2021
Disponible en: <https://firebase.google.com/docs/firestore>
- **UvaDoc “Repositotio Documental de la Universidad de Valladolid”:** documentación relativa a la plataforma Firebase y el servicio Cloud Firestore.
Disponible en: <http://uvadoc.uva.es/handle/10324/45258>
Último acceso: 12/04/2021.
- **Flip Android:** Último acceso: 19/04/2021
Disponible en: <http://www.flipandroid.com/>
- **Academia Android:** Último acceso: 12/04/2021
Disponible en: <https://academiaandroid.com/>
- **Desarrollador Android:** Último acceso: 12/04/2021
Disponible en: <https://desarrollador-android.com/>
- **Tutoriales Programación Ya:** Último acceso: 19/04/2021
Disponible en: <https://www.tutorialesprogramacionya.com/>
- **Develou:** Último acceso: 12/04/2021.
Disponible en: <https://www.develou.com/>.
- **Envatotuts:** sitio web utilizado para la realización de la técnica de raspado web.
Último acceso: 12/04/2021
Disponible en: <https://code.tutsplus.com/es/tutorials/scraping-webpages-in-python-with-beautiful-soup-search-and-dom-modification--cms-28276>
- **W3schools.com:** Último acceso: 12/04/2021
Disponible en: <https://www.w3schools.com/>
- **Python para principiantes:** Último acceso: 12/04/2021
Disponible en: <https://uniwebsidad.com/libros/python>
- **Danielme.com Programación Android, Spring y JEE:**
Último acceso: 12/04/2021
Disponible en: <https://danielme.com/2015/07/14/disen-android-actionbar-con-toolbar/>

- **Flaticon:** sitio web utilizado para obtener las imágenes e iconos requeridos. Último acceso: 19/04/2021. Disponible en: <https://www.flaticon.es/>
- **Programación y Más: “Android Listas Dinámicas usando RecyclerView”.**
Último acceso: 12/04/2021
Disponible en: <https://programacionymas.com/blog/listas-dinamicas-android-usando-recycler-view-card-view>
- **Desarrollo Libre: “¿Cómo mostrar páginas web con WebView en Android?”.**
Último acceso: 12/04/2021
Disponible en: <https://www.desarrollolibre.net/blog/android/como-mostrar-paginas-web-con-webview-en-android#.YHR22OgzblU>
- **SportsRadar:** documentación relativa al API de ciclismo utilizado para obtener datos reales. Último acceso: 22/04/2021
Disponible en: https://developer.sportradar.com/docs/read/baseline_sports_coverage/Cycling_v2

10.3 Contenidos Audiovisuales

- **Youtube: Canal Cristian Henao “Curso Android desde Cero”:**
Último acceso: 15/04/2021
Disponible en: <https://www.youtube.com/watch?v=2b7bXOiOA38&list=PLAg6Lv5BbjdvIclQdVg4ROZnfuuQcqXB>
- **Youtube: Canal Programación y Más “Tutoriales Firebase” (1) y “Tutoriales sobre Desarrollo Android” (2):**
Último acceso: 21/04/2021
Disponibles en: <https://www.youtube.com/watch?v=2b7bXOiOA38&list=PLAg6Lv5BbjdvIclQdVg4ROZnfuuQcqXB> (1)
<https://www.youtube.com/watch?v=wnlAOBS0cqA&list=PLzSFZWTjelblOdVDOsT9hKRI1jsnv> REN (2)
- **Youtube: Canal Coding in Flow:**
Último acceso: 16/04/2021
Disponible en: https://www.youtube.com/channel/UC_Fh8kvtkVPkeihBs42jGcA
- **Youtube: Canal Android Coding:**
Último acceso: 19/04/2021
Disponible en: <https://www.youtube.com/channel/UCUIF5MImktJLDWdKe5oTdJQ>
- **Discord: Canal Coding in Flow:**
Último acceso: 19/04/2021
Disponible en: <https://discord.com/channels/581483136093913128/597356915219103745>
- **Youtube: Web Scrapping “HTML Tables with Python”**
Último acceso: 02/04/2021
Disponible en: <https://www.youtube.com/watch?v=15f4JhJ8SiQ>

- **Youtube: Canal Firebase:**
Último acceso: 21/04/2021
Disponible en: https://www.youtube.com/channel/UCP4bf6IHJJQehibu6ai_cg
- **Youtube: “Android Studio Tutorial-Cloud Firestore:**
Último acceso: 16/04/2021
Disponible en:
<https://www.youtube.com/watch?v=KW-6egvVock>

