

CYCLEAPP

**APLICACIÓN PARA COMPARTIR
RUTAS DE BICI Y MONITORIZACIÓN
DE MANTENIMIENTO**



**VNiVERSiDAD
D SALAMANCA**
Julio 2021

AUTOR

Ignacio Arroyo Crespo

TUTORES

André Sales Mendes

Gabriel Villarrubia Gonzalez

Juan Francisco de Paz Santana

CERTIFICADO DE LOS TUTORES.

D. André Sales Mendez, D. Gabriel Villarubia González y D. Juan Francisco de Paz Santana profesores del Departamento de Informática y automática de la universidad de Salamanca.

CERTIFICAN:

Qué el trabajo titulado “Cycleapp aplicación para compartir rutas en bici y monitorización de mantenimiento” ha sido realizado por Ignacio Arroyo Crespo con DNI 70920385T constituyendo así la memoria del trabajo final de grado en la titulación Grado de Ingeniería Informática de esta universidad.

Y para que así conste a todos los efectos.

En Salamanca, a 7 Julio de 2021

RESUMEN

Este proyecto de fin de grado consiste en la realización de una aplicación móvil que permita llevar el seguimiento del desgaste de los componentes de las bicicletas de los usuarios. Asimismo, permitirá compartir las rutas seguidas mediante redes sociales como pueden ser WhatsApp o Instagram.

También se ha implementado la posibilidad de modificar los componentes de la bicicleta para que, si se le ha realizado un mantenimiento a un componente específico en un taller, se le pueda indicar a la aplicación, y la durabilidad de ese módulo se reestablecerá por completo; lo mismo ocurrirá si en lugar de haber realizado un mantenimiento, directamente se ha sustituido uno de los componentes por otro.

Además, los usuarios de la aplicación tendrán a su disposición una sección de Amigos donde podrán encontrar a sus compañeros de ruta y solicitarles ser amigos en la aplicación. Esto permitirá visualizar tanto la bicicleta que ellos están usando, como las rutas que han realizado, además de poder ver los kilómetros totales recorridos y así tener un incentivo para fomentar la competitividad.

En la sección de seguimiento, se nos indicará la ruta que estamos recorriendo mediante GPS, así como la velocidad actual, velocidad media, velocidad máxima y distancia total. Una vez finalizado el seguimiento, se nos indicará el desgaste que ha sufrido cada componente en base a lo que ya tenía desgastado. El recorrido de todas estas rutas queda almacenado junto a sus parámetros para tener la posibilidad de volver a verlo y compartirlo.

Finalmente, nos encontramos con la sección de estadísticas, que recopila información de todas las rutas realizadas y nos proporciona datos como pueden ser distancia total recorrida, tiempo total de uso de la aplicación en marcha, velocidad máxima absoluta etc.

SUMMARY

In this end grade project, I have developed a mobile application which let us to follow the wear of the different components of a bike. Also, it allows to share the information of the routes that we have followed by social networks like Whatsapp or Instagram.

I have implemented the possibility of modifying the bike components. For example, if you make the maintenance of a specific component, the application will restore the durability of this piece. On the other hand, if you swap one component for another different, the app will allow save this information and the new wear will be in this new component.

In the friend section, you will have the possibility of search your friends and make a friend request. If they accept that request, you will have the possibility of watch their current bike and the routes that they have followed. Also, you will be able to watch the number of kilometres that they have made, and it will let you compete with them.

Talking about the tracing section, the application will follow you on your routes tracing the path you have kept. At the end of the route, the application will display the wear of the bike components and the whole traced route you have done. The app calculates some parameters per route like current speed, max speed, total distance... You will be able to share this information with your friends.

To sum up, I have developed a stats section where the application shows you global information of your routes for example total distance, total time spent in the application or maximum speed

CONTENIDO

Certificado de los tutores.	2
RESUMEN.....	4
Summary	5
1.Introducción	2
2.Objetivos.....	5
2.1 Objetivos del sistema	5
2.2 Objetivos personales.....	7
3.Conceptos teóricos	7
3.1 GPS	8
3.2 Herramientas CASE	10
4.Técnicas y herramientas utilizadas	10
4.1 Técnicas usadas en el desarrollo	10
4.1.1 Android Studio	10
4.1.2 Base de datos Firebase	11
4.2 Herramientas case	11
4.2.1 REM.....	11
4.2.2 Visual Paradigm.....	12
4.2.3 Microsoft Project	13
4.2.4 Ezestimate.....	14

5. Aspectos relevantes del desarrollo	15
5.1 Estimación del esfuerzo	16
5.3. Planificación temporal	18
5.4. Especificación de requisitos	19
5.5 Análisis de requisitos.....	20
5.5.1 Modelo de dominio.....	20
5.5.2Realización de casos de uso del análisis	21
5.5.3 Clases de análisis	21
5.5.4 Vista arquitectónica.....	22
5.6 Diseño del sistema	22
5.6.1 Patrón arquitectónico MVC	22
5.6.2 Subsistemas	23
5.6.3 Clases de diseño	24
5.6.4 Realización de casos de uso del diseño	25
5.6.5 Modelo de despliegue	26
6 Implementación y pruebas	26
8. Funcionalidad del sistema.....	29
9.Conclusiones y líneas de trabajo futuras.....	40
10. Bibliografía.....	43

ÍNDICE DE ILUSTRACIONES

Figura 1. Petición de localización y comprobación de servicios	8
Figura 2. Constructor de la notificación.....	9
Figura 3. Ruta activa.....	9
Figura 4.Estructura firebase.....	11
Figura 5. Herramienta REM	12
Figura 6. Herramienta Visual Paradigm	13
Figura 7. Herramienta Microsoft Project.....	14
Figura 8. Herramienta Microsoft.....	14
Figura 9. Fases del proceso unificado	16
Figura 10. Casos de uso y actores del sistema.....	17
Figura 11. Resultados de la estimación obtenida.....	17
Figura 12. Tareas del diagrama de Gantt.....	18
Figura 17. Diagrama de gestión de usuarios.....	21
Figura 18. Diagrama de Registrar usuario	21
Figura 20. Patrón MVC	23
Figura 21. Modelo de diseño.....	24
Figura 22. Paquete vista	25
Figura 23. Diagrama de secuencia de diseño	26
Figura 24. Modelo de despliegue	26
Figura 25. Strava Figura 26. Salud Huawei Figura 27 CycleApp.....	28

Figura 30. Garaje.....	31
Figura 31. Desgaste componentes	32
Figura 32. Estilo bicicleta	33
Figura 33. Reparación componentes	34
Figura 34. Buscar amigos y enviar solicitud	35
Figura 35. Notificación petición de amistad.....	35
Figura 36. Solicitud recibida y aceptación de amistad.....	36
Figura 37. Lista de amigos.....	37
Figura 38. Información disponible de amigos	38
Figura 39. Ruta en curso Figura 40. Finalización ruta	39

1.Introducción

El presente documento recoge la memoria del Trabajo de Fin de Grado titulado “Cycleapp aplicación para compartir rutas en bici y monitorización de mantenimiento” en la titulación de Grado en Ingeniería Informática. Ha sido realizado por el alumno Ignacio Arroyo Crespo durante el curso 2020-2021 y dirigido por D.André Sales Méndez, D. Gabriel Villarrubia González y D. Juan Francisco de Paz Santana.

En la actualidad, según estudios de la DGT a fecha del 3 de febrero del 2020, la bicicleta alcanza los 20 millones de usuarios, aumentando con notoriedad el uso de la misma con el paso del tiempo y disminuyendo además la brecha de genero un 16% respecto el 2015. Todo esto sumado a la situación actual con la COVID-19 hace que la O.P.S y O.M.S certifiquen el que el uso de la bicicleta sea una actividad “casi obligatoria” para mantenernos sanos y saludables, además de cuidar el medio ambiente.

Todo esto, sumado a que siempre me he considerado un chico deportista que aprovecha cualquier momento libre para salir a hacer una ruta con la bici, y que como buen ciclista he probado distintas aplicaciones de rutas en bicicleta y mantenimiento de estas, considero que puede ser un trabajo bastante útil enfocarme en lo que les falta a las otras aplicaciones y desarrollar una propia.

Por todo lo dicho con anterioridad, el objetivo principal del proyecto va a ser el desarrollar una aplicación móvil en Android, en la cual cualquier usuario que se registre pueda monitorizar de forma sencilla su bicicleta y obtener información útil sobre las rutas que realiza, ya que es algo que me ha faltado en otras aplicaciones, además, también se gestionarán datos como los recambios de piezas, posibilidad de compartir rutas... entre otras posibilidades que explicaremos a lo largo de la memoria.

En este documento, por lo tanto, se explicarán los aspectos más importantes en el desarrollo del proyecto, por lo cual, la estructura de este documento será:

- Objetivos
 - Objetivos del sistema
 - Objetivos de usuarios
 - Exploración y búsqueda de usuarios
 - Competitividad entre usuarios
 - Portabilidad
 - Usabilidad
 - Privacidad de datos
 - Objetivos personales
- Conceptos teóricos
- Técnicas y herramientas usadas en la aplicación
 - Técnicas usadas en el desarrollo
 - Android Studio
 - Base de datos Firebase
 - Herramientas CASE
 - REM
 - Visual Paradigm
 - Microsoft Project
 - EzEstimate
- Aspectos relevantes del desarrollo
 - Estimación del esfuerzo
 - Planificación temporal
 - Especificación de requisitos
 - Participantes

- Objetivos del sistema
- Requisitos de información
- Requisitos funcionales
- Análisis de requisitos
 - Modelo de dominio
 - Realización de los casos de uso del análisis
 - Vista arquitectónica del modelo de análisis
- Diseño del sistema
 - Patrón MVC
 - Subsistemas
 - Clases de diseño
 - Vista arquitectónica del modelo de diseño
 - Casos de uso del diseño
 - Modelo de despliegue
- Implementación
- Pruebas
- Funcionalidad del sistema
- Conclusiones y líneas de trabajo futuras
- Referencias

Además de estar complementado por los anexos:

- **Anexo I: Planificación temporal:** Se basa en la estimación de la duración del proyecto y una planificación temporal con las distintas tareas para llevar a cabo

- **Anexo II: Especificación de requisitos:** Contiene la especificación de los requisitos del sistema
- **Anexo III Análisis de requisitos:** Documentación sobre la fase de análisis de los requisitos
- **Anexo IV: Diseño del sistema software:** Documentación sobre la fase del diseño del sistema
- **Anexo V: Documentación técnica:** Documento que se otorga para ayudar a la comprensión del código y del proyecto desarrollado
- **Anexo VI: Manual de usuario** documento usado para explicar de forma detallada todas las funcionalidades del sistema para que el usuario pueda utilizarlo sin ningún problema

2.OBJETIVOS

En esta sección, se va a exponer los objetivos que debe cumplir el sistema a desarrollar para su posterior validación, así como aquellos que me he propuesto a nivel personal con este proyecto como he comentado anteriormente.

2.1 Objetivos del sistema

El objetivo principal de la aplicación va a ser desarrollar un software orientado a promover el uso de la bicicleta, a hacer rutas en equipo y poder observar rápidamente cuanto uso llevan tus piezas y cuando las tienes que reemplazar.

Para ello, el sistema va a tener que ser capaz de detectar:

- ✓ **Estado del entorno:** Debe ser capaz de detectar el desgaste de las piezas de la bicicleta, y situar la posición del usuario en tiempo real durante la ruta.
- ✓ **Interacción con el entorno:** La aplicación debe ser capaz de calcular la velocidad a la que circula el usuario
- ✓ **Presentar información útil sobre el usuario y la bicicleta:** Se mostrarán una serie de estadísticas sobre el usuario y las rutas realizadas.

- ✓ **Alta, modificación y eliminación de usuarios:** En los que la aplicación controlará que distintos usuarios puedan crearse una cuenta, darla de baja o modificar sus datos personales, foto de perfil....
- ✓ **Añadir, ver y eliminar amigos:** En los cuales dentro de la aplicación se dará la posibilidad de añadir amigos en la aplicación a través de la búsqueda del Nick, atributo de cada usuario que será único dentro de la aplicación, también estará toda la gestión de solicitudes de amistad y su eliminación.
- ✓ **Creación, visualización, compartición de rutas:** En las cuales los distintos usuarios cada vez que salgan con su bicicleta podrán crear una ruta y luego observar todos los parámetros del desgaste que han realizado, también podrán compartir las rutas a través de redes sociales.
- ✓ **Visualizar información del desarrollador:** Se ha creado un apartado donde se puede visualizar la información del desarrollador y la versión de Android para la que está desarrollada la aplicación
- ✓ **Visualización de estadísticas:** El usuario podrá visualizar información relativa a la distancia total que ha recorrido desde la aplicación, velocidades medias, máximas, distancias...
- ✓ **Gestión de usuarios:** El sistema deberá permitir la creación de usuarios con sus respectivos perfiles donde ingresaran el tipo de bicicleta que tienen e información básica de usuario, estos datos, se podrán modificar en cualquier momento.
- ✓ **Exploración y búsqueda de usuarios:** El sistema facilitará que los distintos usuarios se puedan encontrar con otros usuarios permitiéndoles realizar búsquedas a través de su Nick en la aplicación, permitiendo así observar las rutas, bicicletas... de cada uno de ellos.
- ✓ **Competitividad entre usuarios:** La aplicación ofrecerá la posibilidad de visualizar la cantidad de kilómetros totales, tanto tuyos como de las personas que tengas como amigo, de forma que se podrá competir entre distintos usuarios y promoviendo así que utilicen más la aplicación
- ✓ **Portabilidad:** Debido a que se trata de una aplicación pensada para usarla con la bicicleta, se ha prestado especial atención a que se use en una plataforma Android, aunque posteriormente se puede ampliar a la visualización de rutas online a través de una página web.

- ✓ **Usabilidad** El objetivo principal es la satisfacción del usuario mientras usa la aplicación, tanto a nivel persona-ordenador como en tiempos de carga, control de errores, que se intentarán minimizar lo máximo posible.
- ✓ **Privacidad de datos** Se establece una privacidad de datos de forma que ningún usuario podrá acceder a los datos de otro en ningún momento, estando encriptadas las contraseñas y gestionadas por la base de datos

2.2 Objetivos personales

Como he comentado antes, el objetivo principal de esta aplicación va a ser desarrollar una aplicación totalmente práctica destinada para aquellos que disfrutan de un hobby tan extendido como el de mountainbike, para ello se desarrollará como si en un futuro fuera a salir a producción, pero se tendrá en cuenta las limitaciones propias debido a que es un trabajo de ámbito académico que explicaremos más adelante.

Como usuario de distintas aplicaciones de bicicleta, he echado en falta que posean un apartado para registrar la bici, recuento total de kms hechos, mantenimiento de la bicicleta... y creo que si se unifica lo que ya tenemos en las distintas aplicaciones más lo que voy a desarrollar, se puede realizar una aplicación bastante útil y profesional.

Uno de los objetivos también a los que me quería enfrentar era poner a prueba mi capacidad de resolver distintos problemas como programador empezando un desarrollo de un proyecto desde cero siguiendo las directrices el proceso unificado y teniendo en cuenta que cada decisión que tome va a generar un impacto positivo o negativo en el resto de las fases del proyecto.

3. Conceptos teóricos

En este apartado, hemos decidido profundizar más en algunos conceptos que hemos usado a lo largo del proyecto para facilitar su comprensión.

3.1 GPS

Es a lo que se define como un sistema de posicionamiento global, siendo un sistema que permite posicionar cualquier objeto con una precisión casi exacta, que normalmente varía unos escasos metros.

Hay distintos tipos de GPS, pero nosotros como desarrolladores nos vamos a centrar en el segmento de usuario, que consiste en que un equipo, en este caso cualquier dispositivo cuyo sistema operativo sea Android, va a ser un receptor de GPS que recibe señales a través de los distintos satélites GPS, y procesará la posición y la hora en la que está el objeto en un tiempo y momento determinados.

A través de este servicio de ubicación, vamos a poder gestionar toda la información relativa a rutas en tiempo real, estando la aplicación en primer y segundo plano, que vamos a explicar a continuación.

A través de las figuras 1 y 2, vamos a ser capaces de realizar una petición de localización a través de nuestra aplicación para que nuestro dispositivo móvil nos indique unos parámetros y así reproducirlos en el mapa, y además también hemos incorporado un servicio de forma que, aunque la aplicación no se encuentre en primer plano, si está en ruta, siga enviando peticiones de localización e ir dibujando la ruta que está haciendo el usuario en ese momento.

```
LocationRequest locationRequest = new LocationRequest();
locationRequest.setInterval(7000);
locationRequest.setFastestInterval(7000);
locationRequest.setSmallestDisplacement(12);
locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);

if (ActivityCompat.checkSelfPermission(context, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
    ActivityCompat.checkSelfPermission(context, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
    return;
}
LocationServices.getFusedLocationProviderClient(context).requestLocationUpdates(locationRequest, locationCallback, Looper.getMainLooper());
startForeground(Constants.LOCATION_SERVICE_ID, builder.build());
```

Figura 1. Petición de localización y comprobación de servicios

```

String channelId = "location_notification_channel";
NotificationManager notificationManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
Intent resultIntent = new Intent();
PendingIntent pendingIntent = PendingIntent.getActivity(getApplicationContext(), requestCode, resultIntent, PendingIntent.FLAG_UPDATE_CURRENT);
NotificationCompat.Builder builder = new NotificationCompat.Builder(getApplicationContext(), channelId);
builder.setSmallIcon(R.mipmap.ic_launcher);
builder.setContentTitle("Location service");
builder.setDefaults(NotificationCompat.DEFAULT_ALL);
builder.setContentText("Running");
builder.setContentIntent(pendingIntent);
builder.setAutoCancel(false);
builder.setPriority(NotificationCompat.PRIORITY_MAX);

```

Figura 2. Constructor de la notificación

De forma que podremos obtener algo similar a la siguiente imagen:

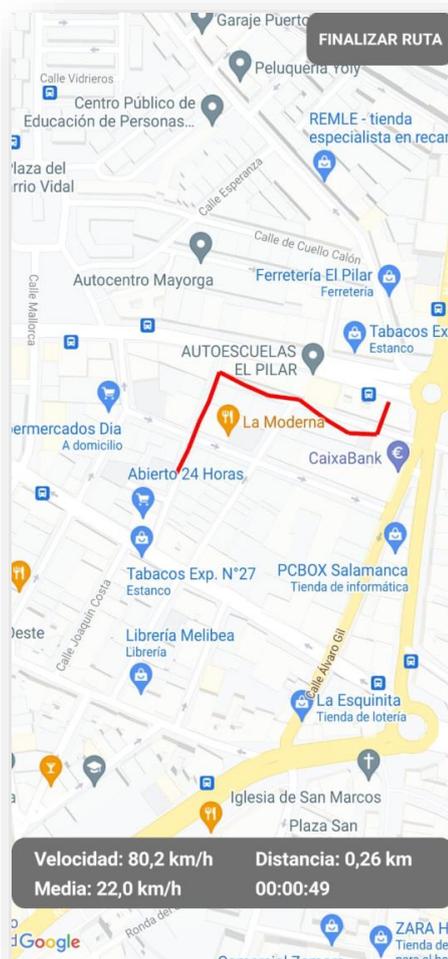


Figura 3. Ruta activa

3.2 Herramientas CASE

Por otro lado, a través de las herramientas CASE se definen como diversas aplicaciones informáticas destinadas a aumentar el balance en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y dinero.

Sus objetivos principalmente son mejorar la productividad y la calidad del software, mejorar la planificación de un proyecto, ayudar a la reutilización, facilitar el uso a través de metodologías, automatizar el desarrollo a través de documentación, generación de código, pruebas de errores... entre otras.

Cualquier herramienta CASE va a estar compuesta por lo tanto de:

- **Repositorio** Diccionario en el que se almacenan los elementos creados
- **Metamodelo:** Marco para la definición de técnicas y metodologías soportadas
- **Carga o descarga de datos**
- **Comprobación de errores:** Conjunto de facilidades para el análisis de la aplicación donde podremos comprobar la integridad, consistencia de esquemas...
- **Interfaz de usuario:** Editor de texto y herramientas de diseño gráfico.

4. Técnicas y herramientas utilizadas

4.1 Técnicas usadas en el desarrollo

4.1.1 Android Studio

Android Studio es un entorno de desarrollo integrado para la plataforma Android, reemplazando así a eclipse como IDE oficial para el desarrollo de este lenguaje, es compatible con todos los sistemas operativos

4.1.2 Base de datos Firebase

Es una base de datos alojada en la nube donde se almacena la información relativa a los clientes de múltiples plataformas que comparten instancia de Realtime Database. El funcionamiento de esta base de datos se basa en que los datos se estructuran en forma de árbol, de la siguiente manera:

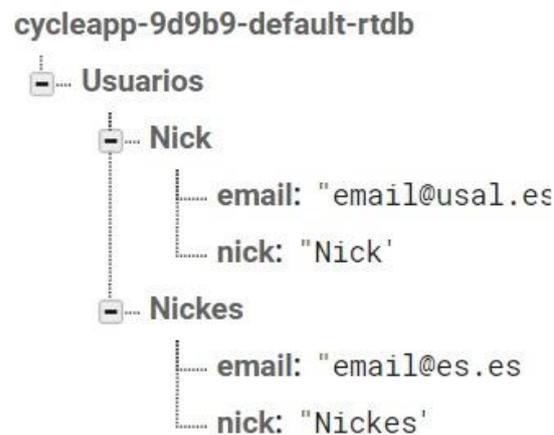


Figura 4. Estructura firebase

4.2 Herramientas case

Las herramientas CASE se definen como un conjunto de herramientas que se encuentran dentro de la Ingeniería del software asistida por ordenador (Computer Aided Software Engineering) que se utiliza para facilitar el desarrollo de aplicaciones de software de calidad, estructuradas y fáciles de mantener. Dentro de estas herramientas, vamos a usar las siguientes.

4.2.1 REM

REM (Requirements Management) se define como una herramienta de gestión de requisitos diseñada para soportar la fase de Ingeniería de requisitos de un producto de desarrollo. Sigue la metodología de Duran y Bernárdez proporcionando plantillas para la documentación de los distintos elementos que componen la elicitación de requisitos.

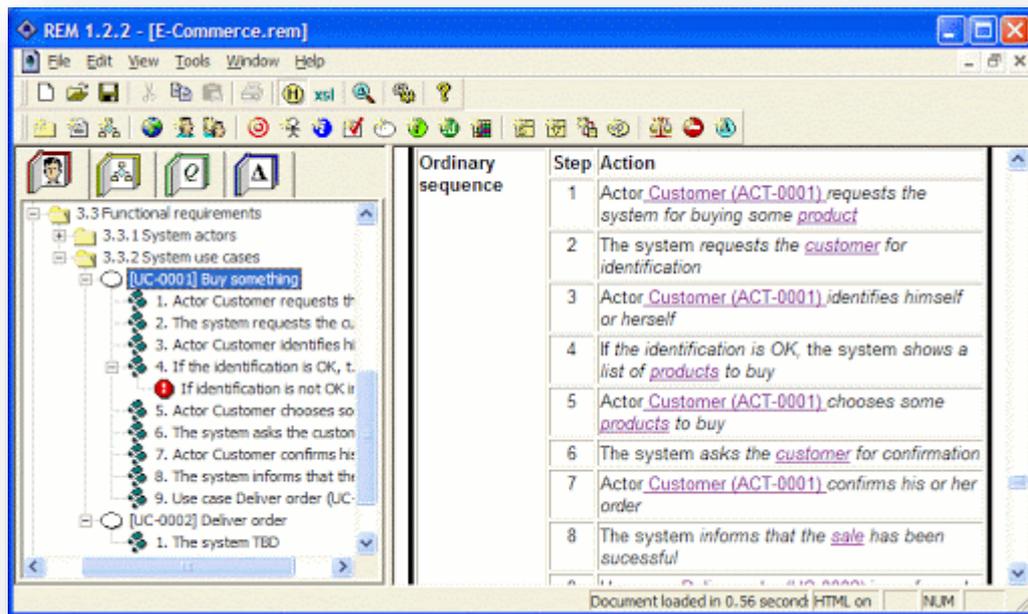


Figura 5. Herramienta REM

4.2.2 Visual Paradigm

Visual Paradigm se define como una herramienta CASE de UML que permite la creación de diagramas de modelado UML como son los diagramas de clase, de casos de uso, de actividad usados a lo largo del proyecto

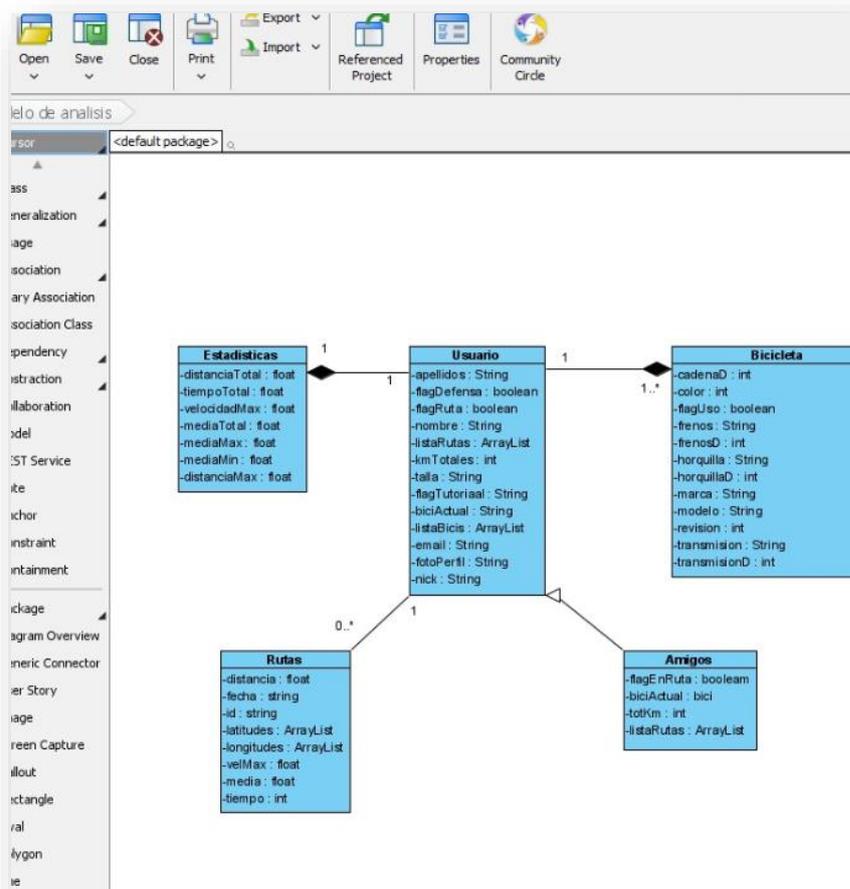


Figura 6. Herramienta Visual Paradigm

4.2.3 Microsoft Project

Microsoft Project es un software de administración de proyectos desarrollado y comercializado por Microsoft cuya función principal es facilitar a los administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, seguimiento de progreso, administración de presupuesto y análisis de cargas de trabajo. En mi caso, ha sido usada para la realización de una planificación temporal.

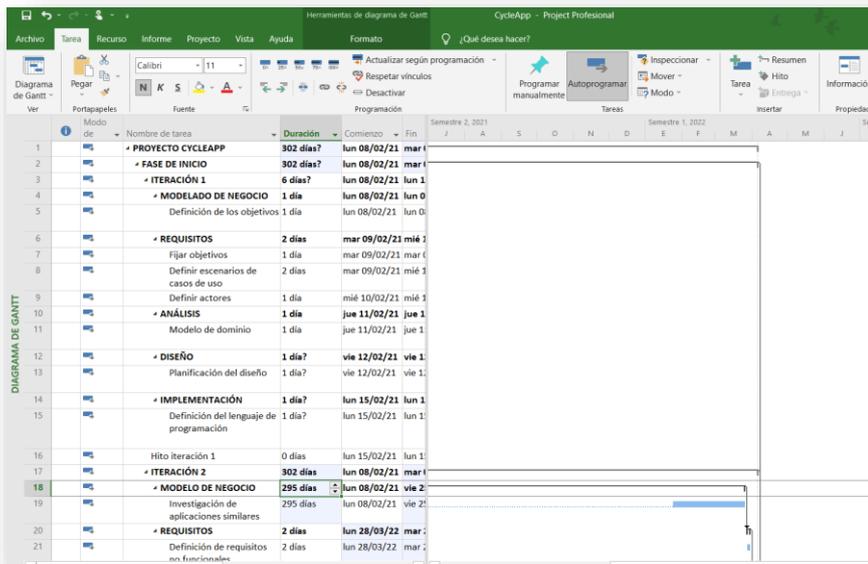


Figura 7. Herramienta Microsoft Project

4.2.4 Ezeestimate

EzEstimate es una herramienta de desarrollo que permite realizar una estimación del esfuerzo del proyecto a través de la complejidad de los casos de uso, permitiendo realizar una estimación de tiempo de un proyecto, dándote una idea de lo que tardarías en desarrollar el sistema desde que empiezas a hacerlo, hasta que sale al mercado.

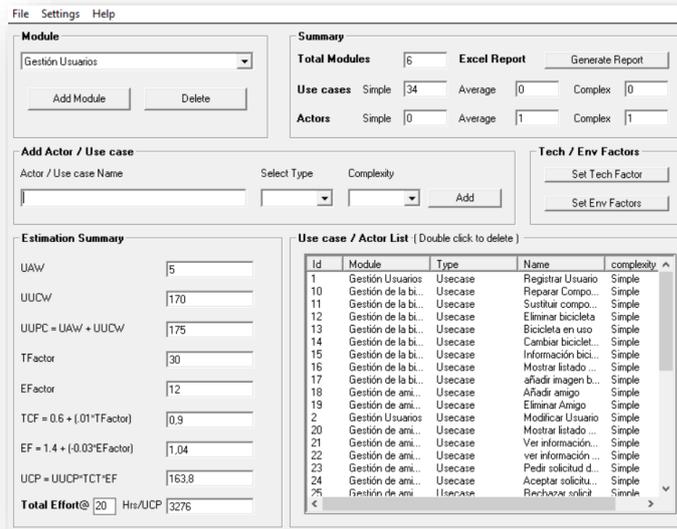


Figura 8. Herramienta Microsoft

5. Aspectos relevantes del desarrollo

En este apartado vamos a especificar las partes más importantes y representativas de las distintas fases del desarrollo del proyecto.

Para ello, nuestro marco de trabajo se ha basado en el proceso unificado, cuyas características principales se basan en:

- **Iterativo e incremental:** Se basa en cuatro fases principales, que son el Inicio, la elaboración, construcción y transición. Cada una de estas cuatro fases, a su vez se van a dividir en una serie de iteraciones, y estas a su vez tendrán como resultado un hito, es decir, un producto que va mejorando, según se va avanzando en el proceso, añadiendo mejores en el producto.
- **Dirigido por caso de uso:** Los casos de uso se usan para capturar los requisitos funcionales y definir los contenidos en cada iteración. La idea principal, es que en cada hito se vayan realizando unos casos de uso llegando a todas las disciplinas como pueden ser el diseño, la implementación, las pruebas...
- **Centrado en la arquitectura:** Como sabemos, el proceso unificado cuenta con múltiples modelos y vistas que definen la arquitectura del software, para ello, vamos a partir de una arquitectura inicial, la cual iremos refinando y mejorando hasta obtener la aplicación final deseada.

Para ello, vamos a contar con varias fases, que estas a su vez se dividen en iteraciones. Las fases principales que vamos a encontrar en todo proceso unificado son:

- Inicio: En dicha fase, definiremos la estrategia de negocio, presentando un modelo, visión, deseos de los usuarios, su coste, viabilidad del producto...
- Elaboración: Obtendremos una visión más refinada sobre el proyecto a realizar, donde veremos modificaciones de requisitos y ajustes en las estimaciones.
- Construcción: Se lleva a cabo el proceso de desarrollo del proyecto
- Transacción: Se define como la etapa final, en la que todo el proyecto debe estar listo para ser entregado, probado y puesto a punto para que los usuarios lo puedan utilizar sin ningún tipo de problema, siendo el momento de pensar den futuras actualizaciones si así se desea

En la siguiente figura, podemos ver de forma gráfica las fases y como según avanzamos en el tiempo y las iteraciones van tomando importancia unas fases u otras, además de las disciplinas que se van usando en cada etapa.

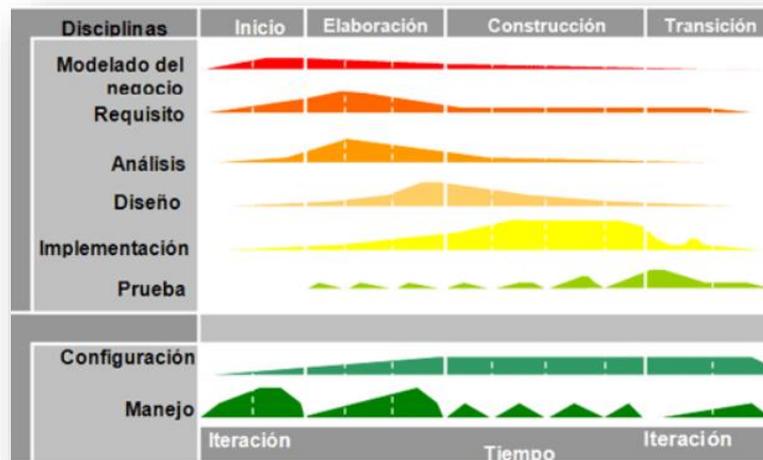


Figura 9. Fases del proceso unificado

Otro aspecto relevante, ha sido la planificación temporal, siendo una de las primeras tareas que llevé a cabo, con esta planificación, ha sido posible conocer el tiempo aproximado que se emplearía en cada una de las tareas, y saber la fecha final de finalización del proyecto.

5.1 Estimación del esfuerzo

Para ello, por un lado, realizamos una estimación de esfuerzo a través de la herramienta EzEstimate anteriormente descrita, en la cual, el principal objetivo era calcular la estimación del esfuerzo a través de los casos de uso de nuestro sistema, conociendo esta información, y cuanto nos abarcará el tiempo de desarrollo, podremos distribuir el tiempo para cada una de las tareas del proyecto.

Para conocer más información acerca de la estimación realizada para CycleApp, se puede consultar el anexo I- Plan de proyecto que va inclusive con esta memoria.

Para la estimación de esfuerzo, metimos los casos de uso con la complejidad de cada uno de ellos, y los actores que intervienen en nuestro sistema como vemos en la figura 9

Id	Module	Type	Name	complexity
1	Gestión Usuarios	Usecase	Registrar Usuario	Simple
10	Gestión de la bi...	Usecase	Reparar Compo...	Simple
11	Gestión de la bi...	Usecase	Sustituir compo...	Simple
12	Gestión de la bi...	Usecase	Eliminar bicicleta	Simple
13	Gestión de la bi...	Usecase	Bicicleta en uso	Simple
14	Gestión de la bi...	Usecase	Cambiar biciclet...	Simple
15	Gestión de la bi...	Usecase	Información bici...	Simple
16	Gestión de la bi...	Usecase	Mostrar listado ...	Simple
17	Gestión de la bi...	Usecase	añadir imagen b...	Simple
18	Gestión de ami...	Usecase	Añadir amigo	Simple
19	Gestión de ami...	Usecase	Eliminar Amigo	Simple
2	Gestión Usuarios	Usecase	Modificar Usuario	Simple
20	Gestión de ami...	Usecase	Mostrar listado ...	Simple
21	Gestión de ami...	Usecase	Ver información...	Simple
22	Gestión de ami...	Usecase	ver información ...	Simple
23	Gestión de ami...	Usecase	Pedir solicitud d...	Simple
24	Gestión de ami...	Usecase	Aceptar sollicitu...	Simple
25	Gestión de ami...	Usecase	Rechazar sollicit...	Simple

Figura 10. Casos de uso y actores del sistema

Introduciremos todos los parámetros necesarios en la aplicación como puede ser la complejidad técnica y la de entorno y donde nos resulta:

Estimation Summary	
UAW	5
UUCW	170
UUPC = UAW + UUCW	175
TFactor	30
EFactor	12
TCF = 0.6 + (.01*TFactor)	0,9
EF = 1.4 + (-0.03*EFactor)	1,04
UCP = UUCP*TCF*EF	163,8
Total Effort@ 6 Hrs/UCP	982,8

Figura 11. Resultados de la estimación obtenida

Lo cual nos indica que esos 982.8 h/UCP son 122,85 días, siendo aproximadamente 4 meses su tiempo de realización, tiempo que abarca totalmente hasta la presentación del trabajo final de grado.

5.3. Planificación temporal

Fue realizada tras conocer el resultado de la estimación del apartado anterior. Para la realización de este apartado, en primer lugar, debimos realizar un esquema con las distintas tareas que teníamos que realizar y conocer o tener una ligera idea del tiempo que deberíamos emplear en cada una de ellas. Debido a que hemos usado el proceso unificado durante toda la fase de desarrollo del proyecto como hemos indicado en el [punto anterior](#), el diagrama de Gantt resultante con las distintas tareas y realizado a través de Microsoft Project es el siguiente:

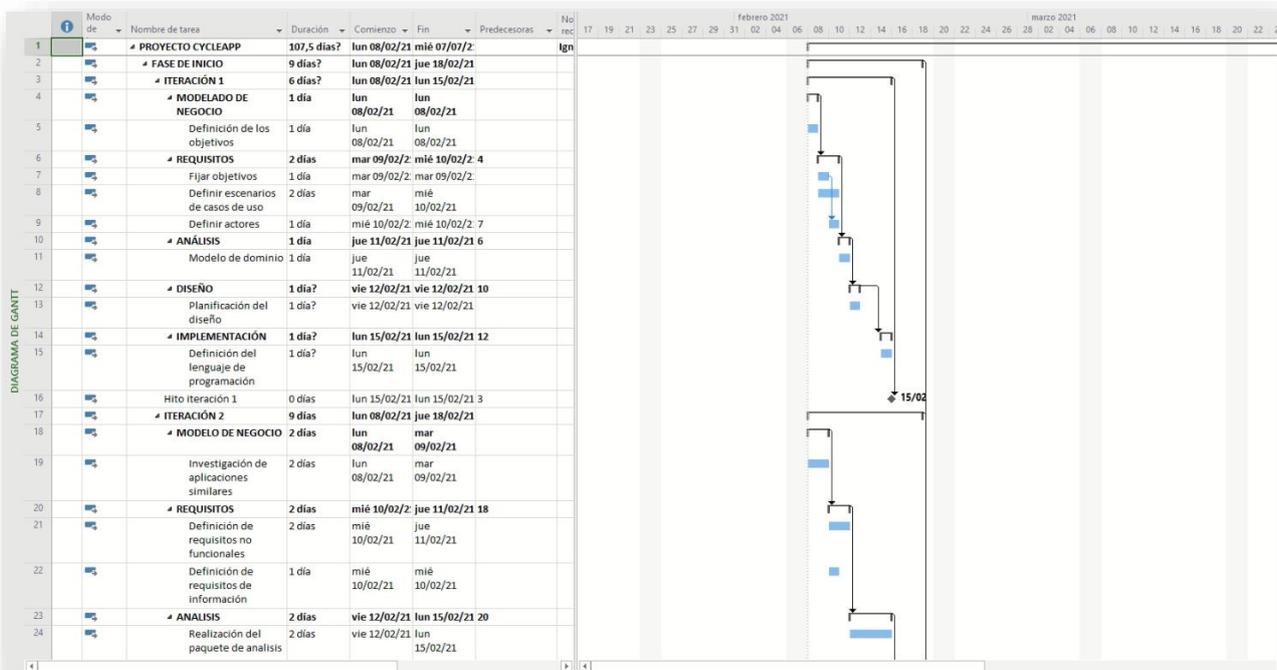


Figura 12. Tareas del diagrama de Gantt

Al igual que en el anterior punto, para conocer más información acerca de la planificación realizada para CycleApp, se puede consultar el anexo I- Plan de proyecto que va inclusive con esta memoria.

5.4. Especificación de requisitos

En este apartado los objetivos a los que nos hemos enfrentado han sido el control de usuarios, control de desgaste de bicicletas, control de la información de las rutas y amigos, así como el control de la gestión de la ubicación, tanto en primer como en segundo plano.

Respecto a la especificación de requisitos, se ha seguido la metodología de Durán y Bernárdez, en la que se ha especificado los participantes del proyecto, así como los requisitos funcionales, no funcionales y de información.

Para manejar de manera mas sencilla todas las funcionalidades de nuestro sistema, se han dividido en paquetes donde se van a resumir las principales funcionalidades de Cycleapp: gestión de rutas, de estadísticas, de bicicletas, de amigos y de información. Para entender más fácilmente todo lo descrito en este punto, podemos visualizar la figura 13 donde se ve uno de los diagramas de casos de uso de la aplicación.

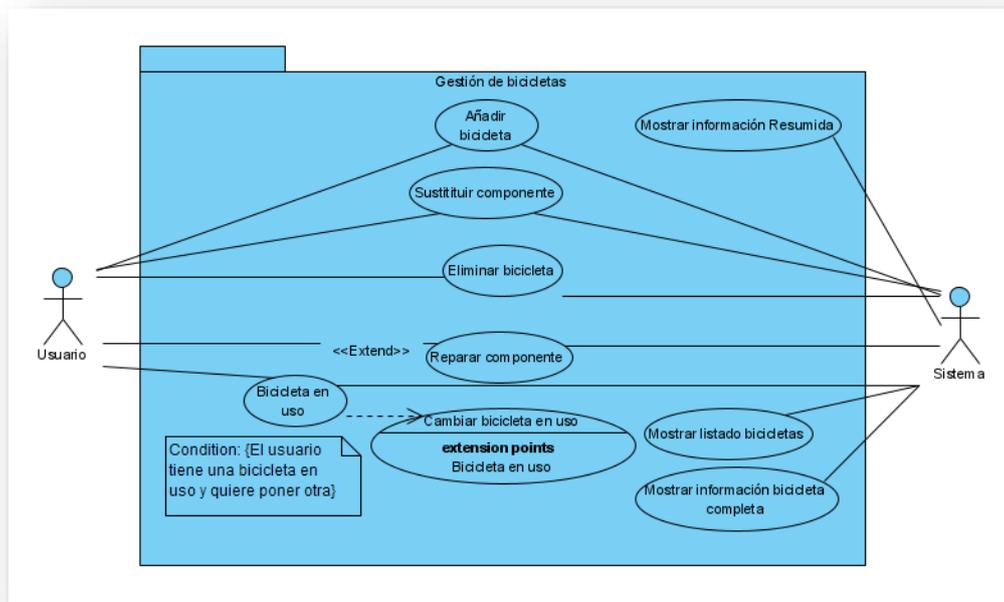


Figura 13. Diagrama gestión de bicicletas

Toda esta información se encuentra más detallada en el Anexo II: Especificación de requisitos software.

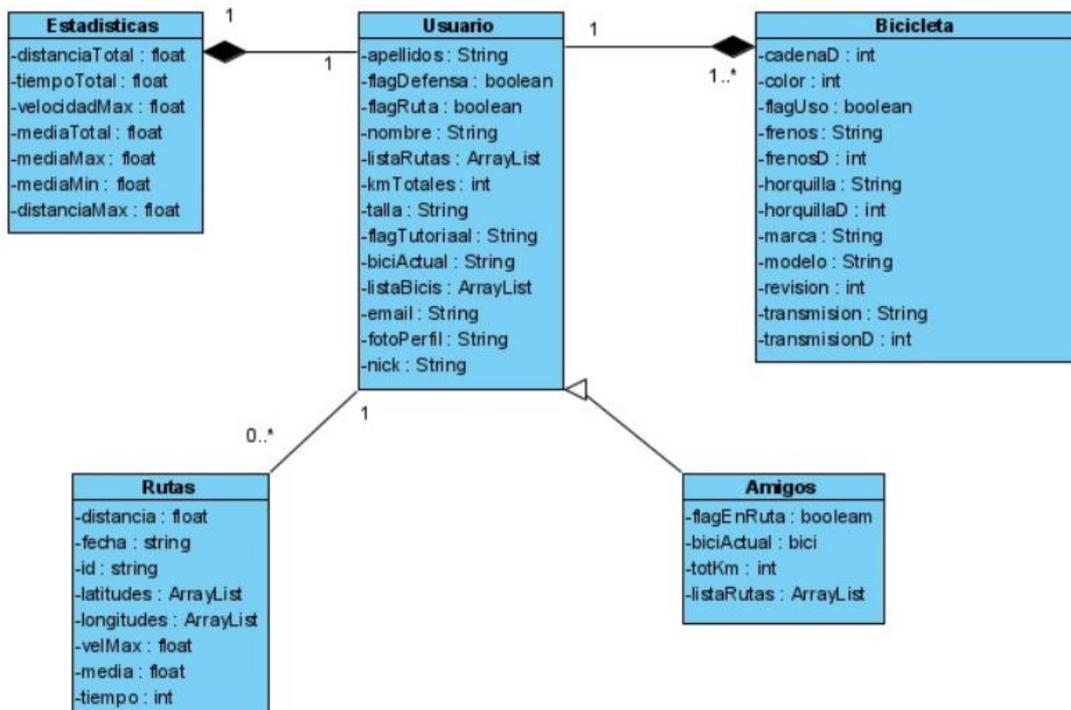
5.5 Análisis de requisitos

El análisis de requisitos lleva a cabo un análisis de los requisitos recogidos en el [apartado anterior](#). Para ello se han determinado los apartados principales pertenecientes a este punto.

Al igual que en otros puntos, para más información, se puede visualizar el documento Anexo III- Análisis de requisitos inclusive a esta memoria

5.5.1 Modelo de dominio

El modelo de dominio es una representación de las clases conceptuales que representan lo importante de nuestra aplicación para la vida real, recogiendo así las necesidades de almacenamiento y gestión de la información en el sistema, para ello,



hemos decidido determinar aquellas entidades más importantes en las cuales son importante la gestión que hagamos de ellas, como podemos ver en la figura 18

Figura 17. Diagrama de gestión de usuarios

5.5.2 Realización de casos de uso del análisis

En este apartado se han incluido los distintos diagramas de secuencia donde se ve claramente las iteraciones y mensajes entre los distintos objetos del sistema.

En la figura 18, podremos visualizar el diagrama de secuencia en el caso de registrar usuario:

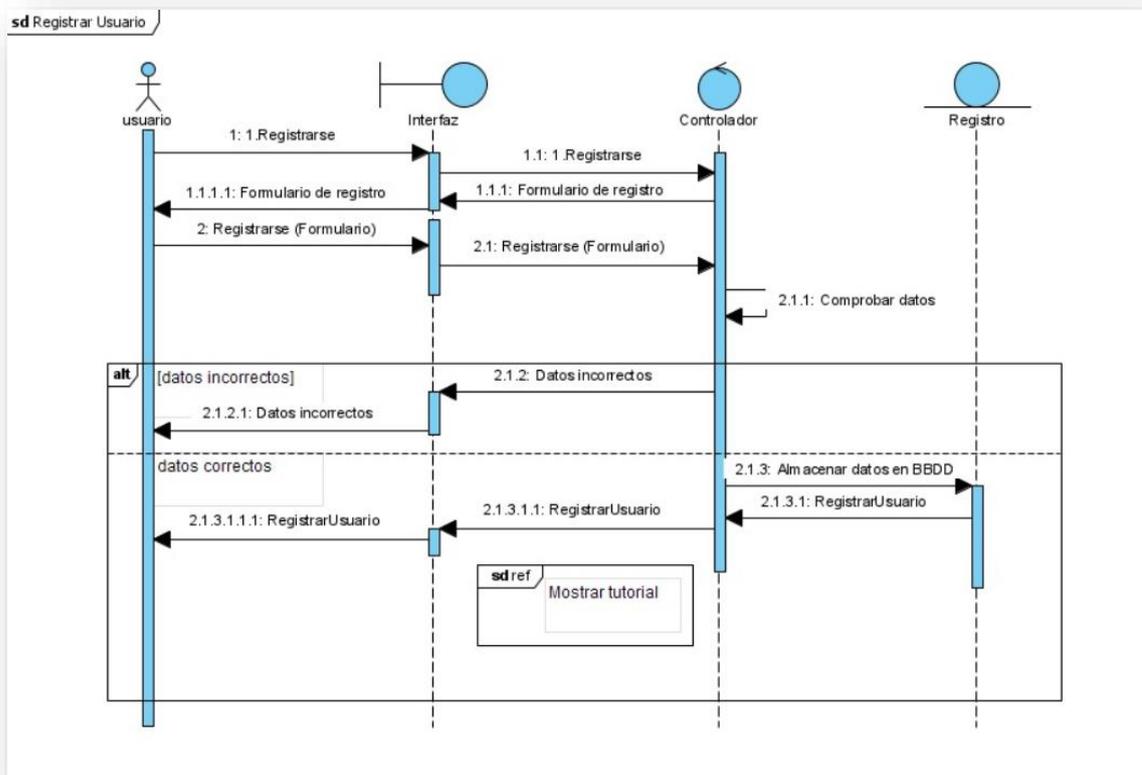


Figura 18. Diagrama de Registrar usuario

5.5.3 Clases de análisis

Dentro de este apartado hemos mostrado los diagramas de comunicación para observar mejor como se comunican entre ellos, estos diagramas se dividen en distintos paquetes que componen el sistema.

5.5.4 Vista arquitectónica

La vista arquitectónica se usa para situar cada clase dentro de la capa correspondiente dentro del modelo vista controlador explicado también en el Anexo III.

El resultado final será la figura 19, la cual será la propuesta para la arquitectura final de CycleApp dividida por paquetes, en el caso de gestión de usuarios resulta:

5.6 Diseño del sistema

La fase de diseño se centra en el dominio de la solución, es decir, haremos una aproximación sobre la implementación, es decir, tanto el nombre de las clases, métodos y atributos que se mostrarán a lo largo del documento, serán una aproximación cercana a la implementación, teniendo bastante similitud en cuanto a nombres.

La información detallada de la fase de diseño del sistema se puede encontrar en el anexo IV- Diseño del sistema software inclusive en esta memoria.

5.6.1 Patrón arquitectónico MVC

El patrón MVC se define como un patrón cuya arquitectura software separa los datos de la aplicación, la interfaz del usuario y la lógica de control, centrándose así de forma independiente en las tecnologías y entornos del proyecto.

La separación de Cycleapp entonces, se va a dividir en tres grupos.

- **Modelo:** Contiene una representación sobre los datos del sistema que queremos almacenar.
- **Vista:** Representación visual de los datos para el usuario.
- **Controlador:** Actúa como intermediario entre el modelo y la vista, gestionando el flujo de información entre ambos.

Obteniendo un flujo similar al de la figura 20

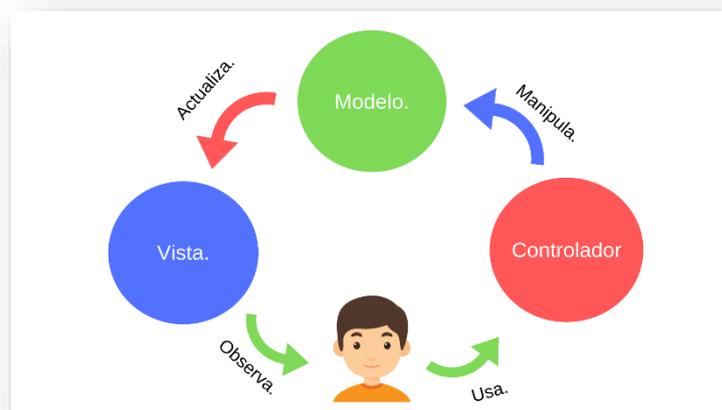


Figura 20. Patrón MVC

Dentro de nuestra aplicación Android, hemos hecho una separación donde nuestro modelo será nuestra base de datos, en este caso, Firebase, el controlador las clases .java de nuestro sistema y finalmente el diseño los ficheros .xml que son la interfaz del proyecto.

De forma que la vista enviará eventos al controlador y este los gestionará, invocando distintas llamadas al modelo.

5.6.2 Subsistemas

En esta sección se realiza la descomposición del sistema de diseño en subsistemas para que sea más manejable basándonos en la arquitectura nombrada con anterioridad.

- **Aplicación Android:** Hace referencia a nuestra aplicación Android, donde se gestionará toda la lógica de la aplicación donde encontraremos los distintos controladores de cada paquete de CycleApp.
- **Firestore:** Base de datos donde se almacena cualquier información referente a la aplicación

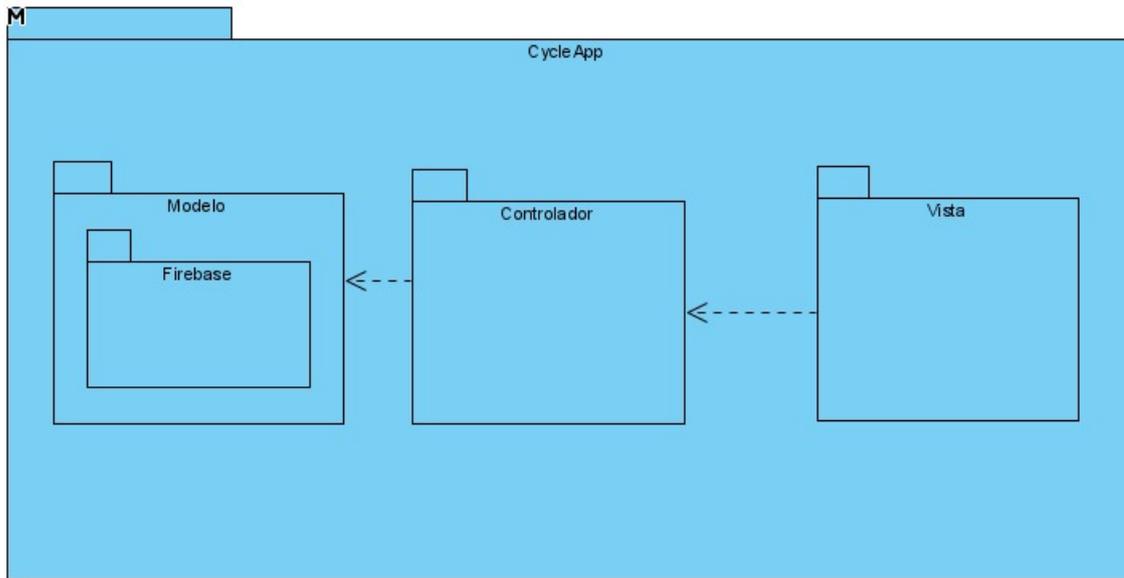


Figura 21. Modelo de diseño

5.6.3 Clases de diseño

En este apartado, hemos especificado aquellos componentes de nuestro sistema, las clases de cada paquete y los métodos de cada una de ellas. Para entenderlo mejor, vamos a adjuntar un diagrama que hace referencia al patrón MVC que como hemos explicado con anterioridad, es el que hemos usado. Dentro de cada paquete de modelo, vista y controlador, hemos puesto las clases correspondientes a cada una de ellas, como podemos visualizar en la siguiente figura 22

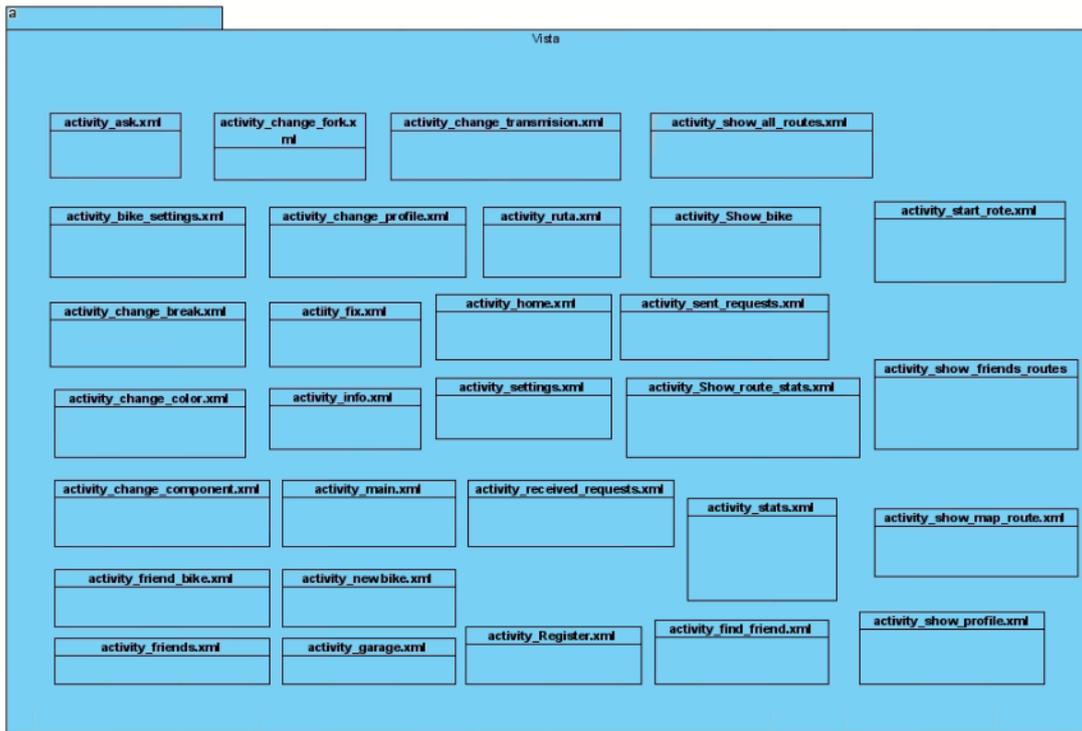


Figura 22. Paquete vista

5.6.4 Realización de casos de uso del diseño

Dentro de este apartado, hemos detallado el intercambio de mensajes entre los distintos objetos de nuestra aplicación mediante los diagramas de secuencia de diseño, para ello, podemos visualizar la figura 25 en el caso de sustituir componente

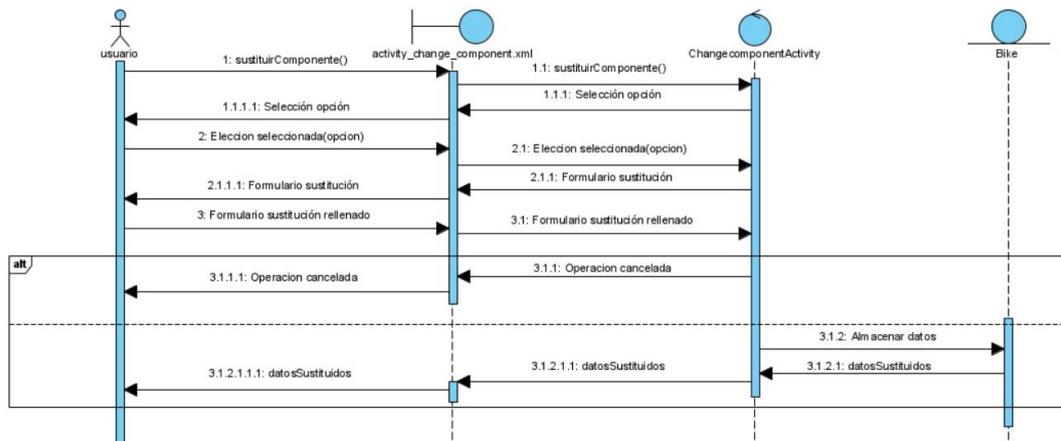


Figura 23. Diagrama de secuencia de diseño

5.6.5 Modelo de despliegue

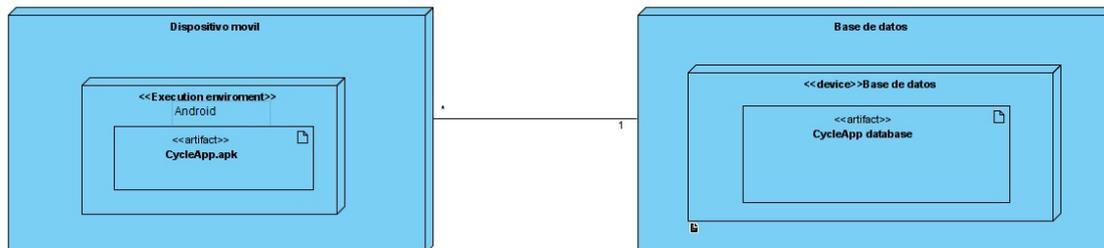


Figura 24. Modelo de despliegue

Como vemos en la imagen superior, tenemos principalmente dos nodos que vamos a explicar con mayor detenimiento a continuación:

- **Dispositivo móvil:** Dispositivo cuyo entorno es Android y donde se encuentra la aplicación a desarrollar. Mediante esta aplicación, los usuarios podrán acceder a cualquier acción detallada con anterioridad en este anexo.
- **Base de datos:** Accederemos a nuestra base de datos actualizada, donde almacenaremos la información relevante para esta, en nuestro caso, la base de datos que usamos es Firebase

6 IMPLEMENTACIÓN Y PRUEBAS

Dentro de este apartado se ha realizado la codificación del sistema tomando como base los resultados obtenidos en la fase de diseño desarrollados en el punto anterior.

El primer proceso que seguí fue la implantación de un sistema de gestión de usuarios con Firebase. Esto consistió en asociar mi cuenta de Firebase con la de Android Studio y no dio mayor problema. El proceso de Loguin es casi automático puesto que hay un

par de funciones que permiten loguear a los usuarios con mail y contraseña de manera sencilla.

A continuación, me encargué de construir todo el concepto de bicicleta con todos sus parámetros, y posteriormente desarrollé la asociación de un listado de bicicletas para cada usuario registrado. El proceso no fue complicado, porque una vez entendida la jerarquía y la estructura que sigue Firebase, simplemente es añadir cada atributo donde corresponde. El proceso de mostrar este listado de bicicletas asociado a cada usuario, si que se complicó algo más, puesto que eran valores dinámicos que no tenían siempre un valor concreto (un usuario puede añadir, eliminar y desgastar una bicicleta, por lo que cada vez que se acceda a la opción de listar las bicicletas habrá distintas cosas que mostrar). Para solucionar esto, utilicé un ScrollLayout cuyos children estuvieran generados por código, así a la hora de recorrer la base de datos, se mostraban en pantalla tantas bicicletas como hubiera en la base de datos. Esto es bastante complicado puesto que la manera de representar formas y datos en pantalla, si no son objetos fijos que los puedes arrastrar con el ratón y ponerlos donde quieras, lo tienes que hacer con valores escritos por código (márgenes, paddings, tamaños...).

La idea de tener amigos era sencilla, pero no tanto de implementar, puesto que para cada usuario había que generar un listado de amigos, de peticiones emitidas y de peticiones recibidas, junto a todas las comprobaciones de no poder enviar solicitudes a alguien que ya se la has mandado, no poder mandar solicitudes a alguien que ya es tu amigo. Por ejemplo, el proceso interno de aceptar una solicitud consiste en eliminar la solicitud de la lista de solicitudes emitidas del usuario que ha hecho la petición de amistad, eliminar la solicitud de la lista de solicitudes recibidas del usuario que ha recibido la petición de amistad, añadir a la lista de amigos del usuario actual, al usuario que ha emitido la petición y añadir a la lista de amigos del usuario que ha hecho la petición al usuario que le ha solicitado; así que fácil no fue.

Una vez hecho esto, acceder a la bicicleta actual de un amigo no resultó complicado puesto que se pudo reutilizar bastante código.

Respecto a todo lo relacionado con el GPS y las rutas hubo varias semanas de trabajo. Lo primero que hice fue ver varios tutoriales de como implementar un mapa en la aplicación, y una vez aprendido esto, la API de Google Maps me proporcionó una clave que exporté a mi proyecto y automáticamente me aparecía un mapamundi con toda la información que necesitaba. Más tarde traté de programar la simulación de una ruta, solo las líneas sin ningún parámetro más; es decir, que simplemente pulsando con el dedo en un punto, el mapa me dirigiera con una línea roja hacia él, y así las veces que pulsara, fabricando una ruta. Una vez hecho esto, simplemente calculé varios parámetros (velocidad, tiempo, distancia) que se generaban en esas rutas simuladas y los valores eran bastante aceptables. Una vez hecho esto, simplemente había que modificar la manera de obtener esta localización, recordamos que hasta ahora lo hacía pulsando con los dedos por donde iría yo con mi bici teóricamente, esto me demoró unos cuantos días puesto que la precisión no era nada buena y me marcaba puntos que distaban mucho entre sí. Finalmente logré encontrar un proveedor de GPS aceptable, activé los flags de precisión y logré que se asemejaran bastante las rutas a las generadas por otras aplicaciones de seguimiento:

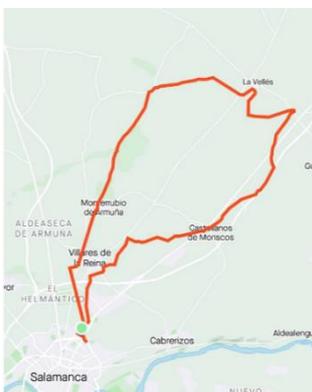


Figura 25. Strava

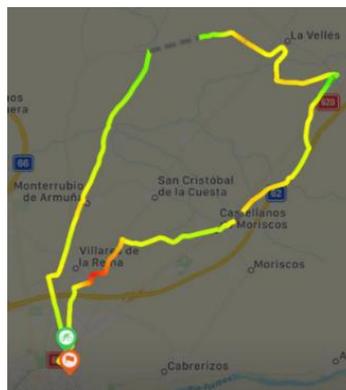


Figura 26. Salud Huawei

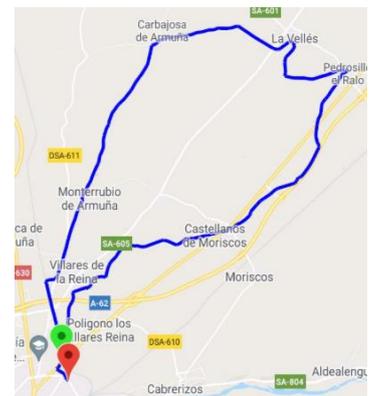


Figura 27 CycleApp

Finalmente introduje los parámetros que calculaban las estadísticas y ya quedó implementada completamente la sección de rutas. Para darle mayor usabilidad, y para los ciclistas que lo deseen, implementé un servicio que mantiene el seguimiento de la aplicación en segundo plano, por lo que pueden usarla cuando está el móvil bloqueado (pero personalmente prefiero ir mirando el mapa para saber la ruta por la que voy).

Para terminar, implementé los procesos que funcionan con agentes externos de la aplicación, como puede ser cambiar la foto de perfil por una de la galería, enviar las

rutas por redes sociales, o cambiar la imagen de las bicicletas, donde no tuve mayor problema.

Las pruebas han sido fundamentales en este tipo de aplicación debido a que nos han permitido comprobar que el cálculo de todo tipo de variables como estadísticas, desgastes, la trazada de cada una de las rutas es correcto.

En primer lugar, cada vez que se añadía una funcionalidad, se probaba de forma individual, y una vez probadas todas estas, se ha probado la aplicación de forma global teniendo integrados todos los componentes necesarios.

8. Funcionalidad del sistema

Dentro de este apartado vamos a mostrar de forma resumida los aspectos más relevantes de la funcionalidad de CycleApp. Para más información, se puede visualizar el anexo VI Manual de usuario donde se explica cada una de las funcionalidades que puede desarrollar el usuario.

Se ha implementado la opción de que tanto cuando un usuario vea su perfil como cuando quiera ver la información de desgaste de su bici, pueda añadir una imagen tanto de él mismo, como de la bicicleta que usa, dándole así más personalización a la aplicación y no proponiéndole únicamente unas imágenes por defecto, como ocurre en el caso de la bicicleta, elementos que podemos ver en las figuras 28 y 29 y 32

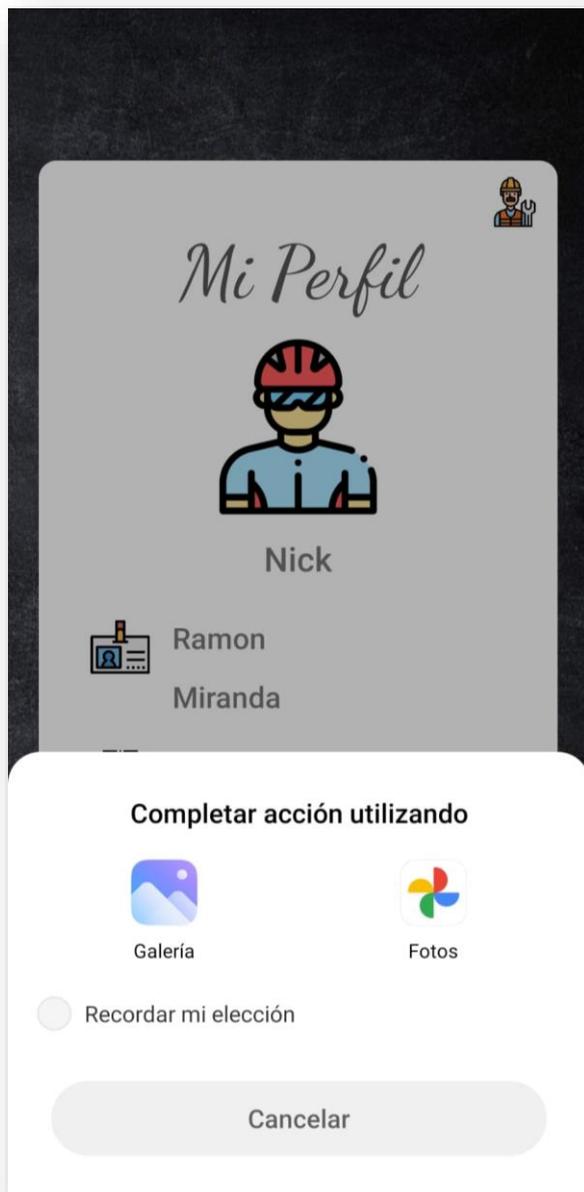


Figura 28. Cambiar imagen

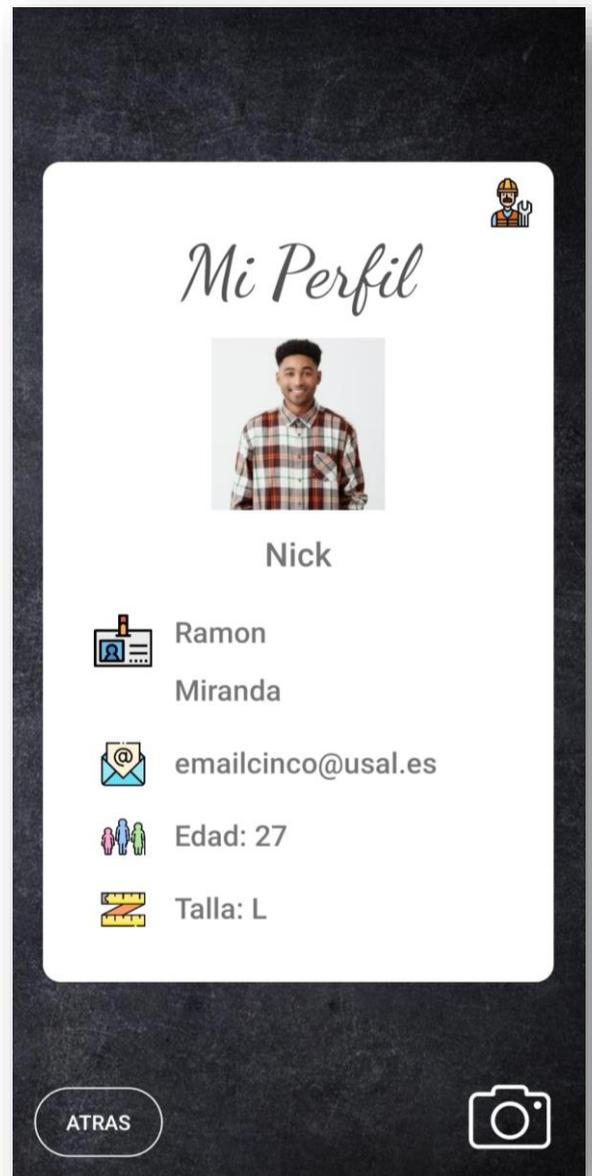


Figura 29. Perfil Usuario

Otro aspecto importante, es que un usuario disponga de un apartado de garaje donde pueda registrar más de una bici y que dependiendo de las rutas que vaya a realizar o cual le apetezca usar, pueda elegir entre una u otra, de ahí que cada bicicleta tenga un

botón en rojo o en verde dependiendo de si está operativa o no, como podemos visualizar en la figura 30 y el desgaste de los distintos componentes en la figura 31

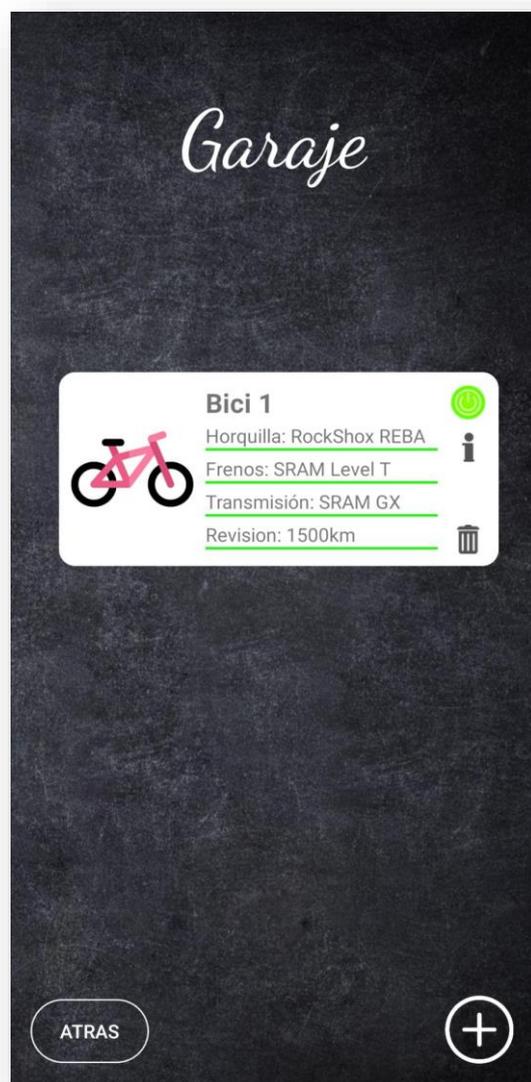


Figura 30. Garaje

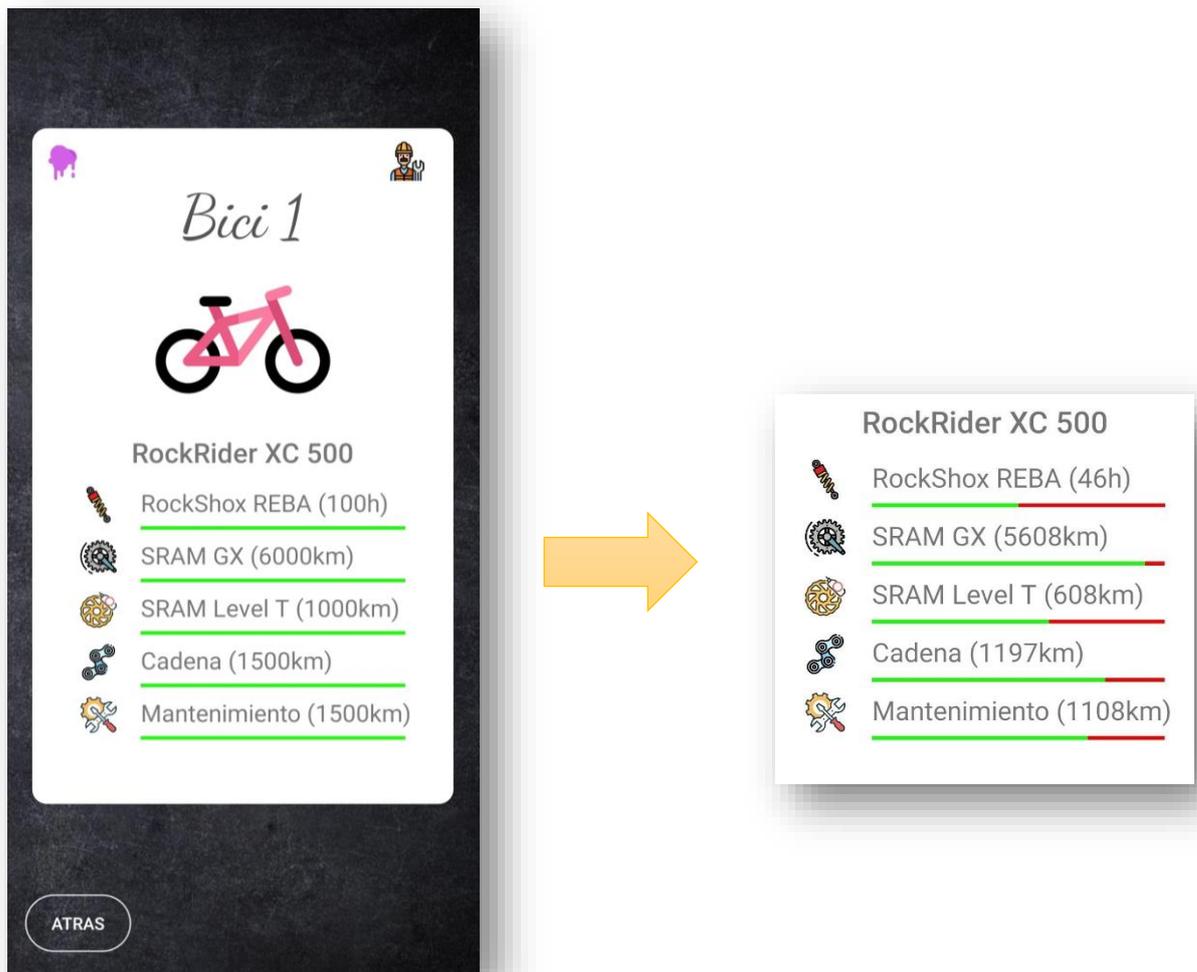


Figura 31. Desgaste componentes

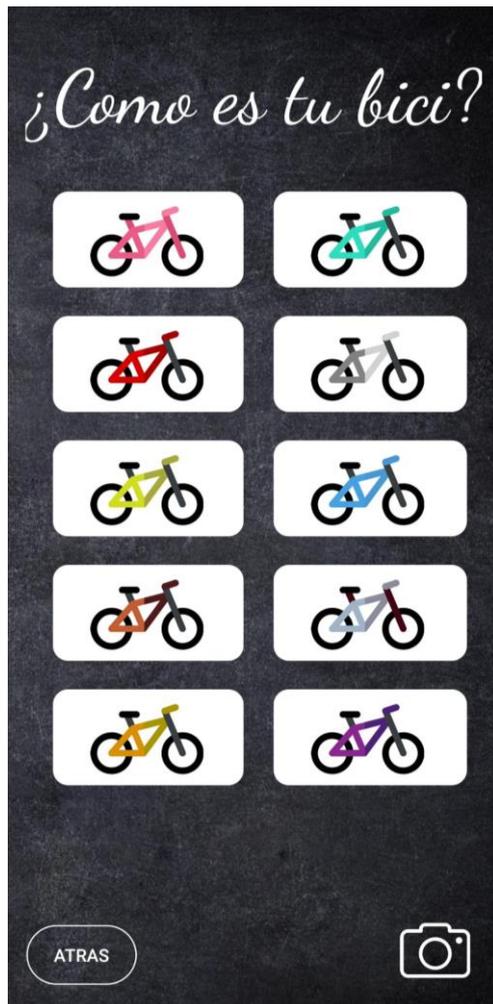


Figura 32. Estilo bicicleta

También es reseñable la gestión de sustitución reparación de componentes, donde dependiendo de lo que haga, como podemos ver en la figura 33

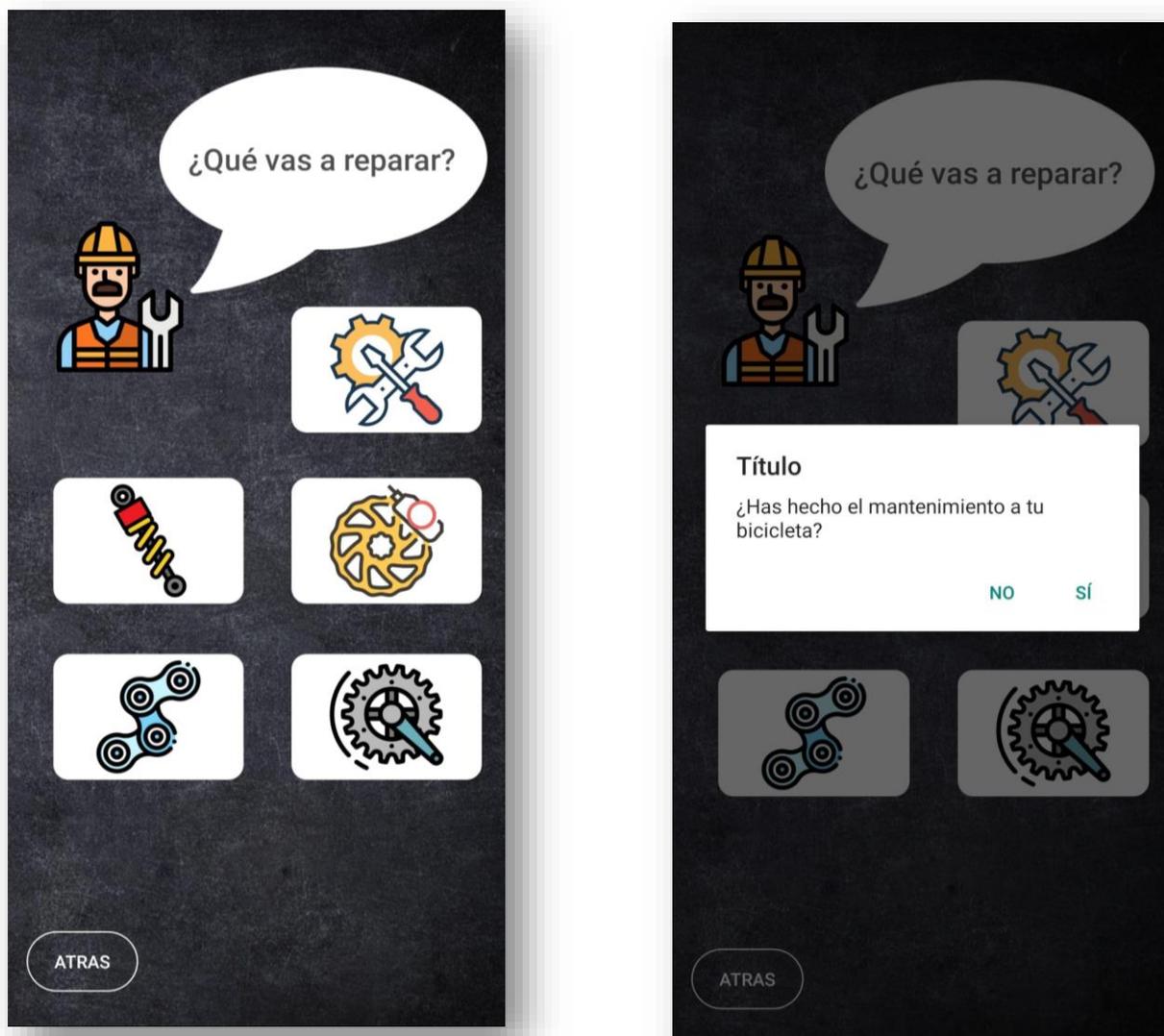


Figura 33. Reparación componentes

También me pareció interesante destacar la funcionalidad del caso de amigos, donde si tenemos dos usuarios y uno de ellos envía petición al otro, a este segundo le llegará notificación de la petición de amistad del primero, y al cual, podrá aceptar o rechazar y en caso de aceptarle, ver información del usuario y de sus rutas y bicicleta como veremos en las figuras del 34 a la 38



Figura 34. Buscar amigos y enviar solicitud



Figura 35. Notificación petición de amistad



Figura 36. Solicitud recibida y aceptación de amistad



Figura 37. Lista de amigos

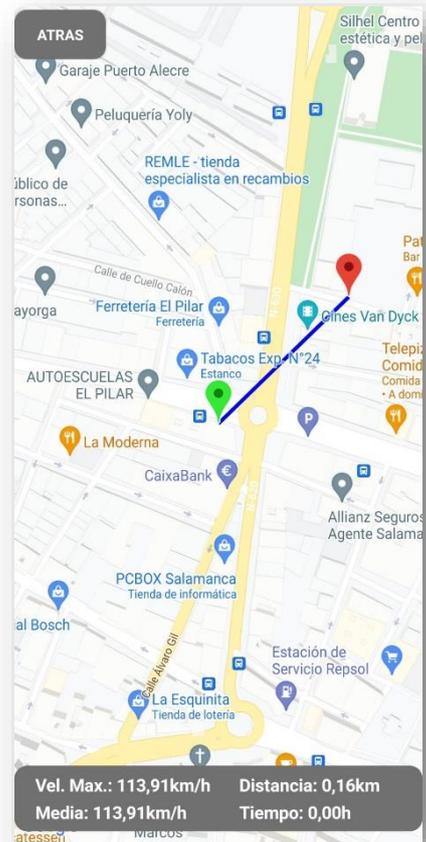
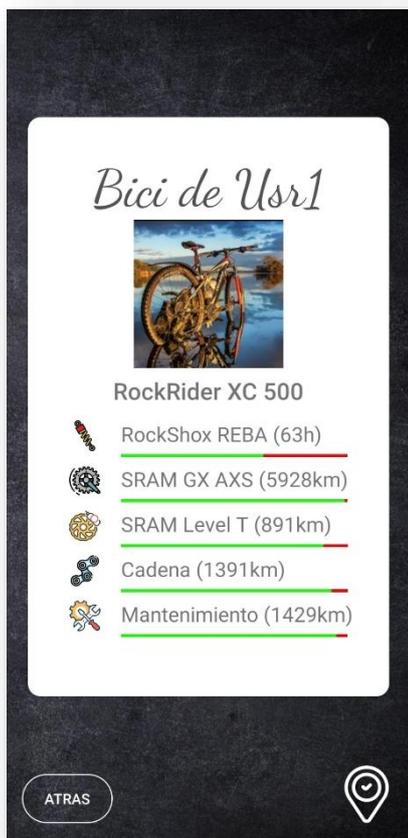


Figura 38. Información disponible de amigos

Aunque hay bastantes funcionalidades implementadas me parecido reseñable destacar es sobre todo la gestión de rutas, donde el usuario puede seguir a tiempo real la ruta que está haciendo y ver como los parámetros van variando y el hecho de poderla visualizar después como podemos ver en las figuras 39 y 40

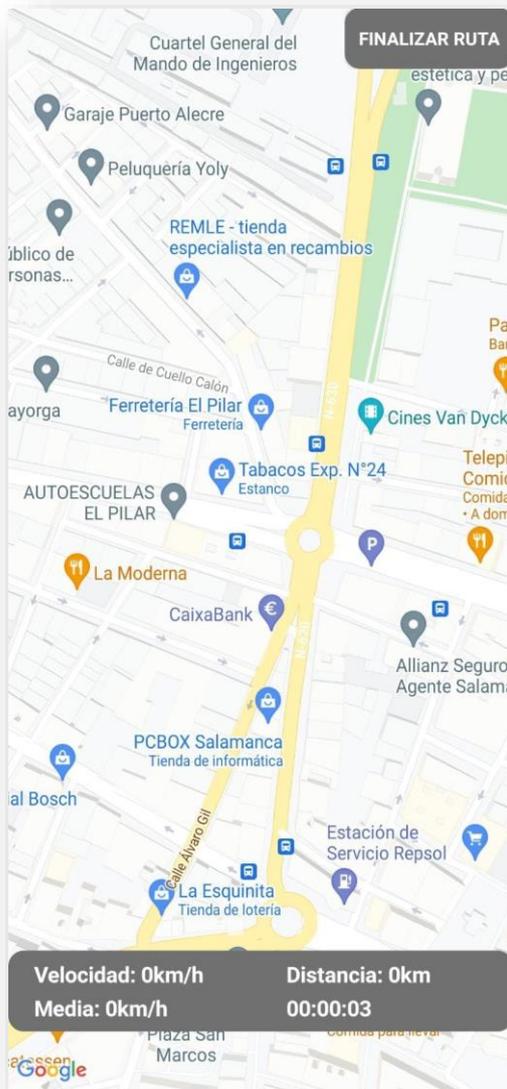


Figura 39. Ruta en curso

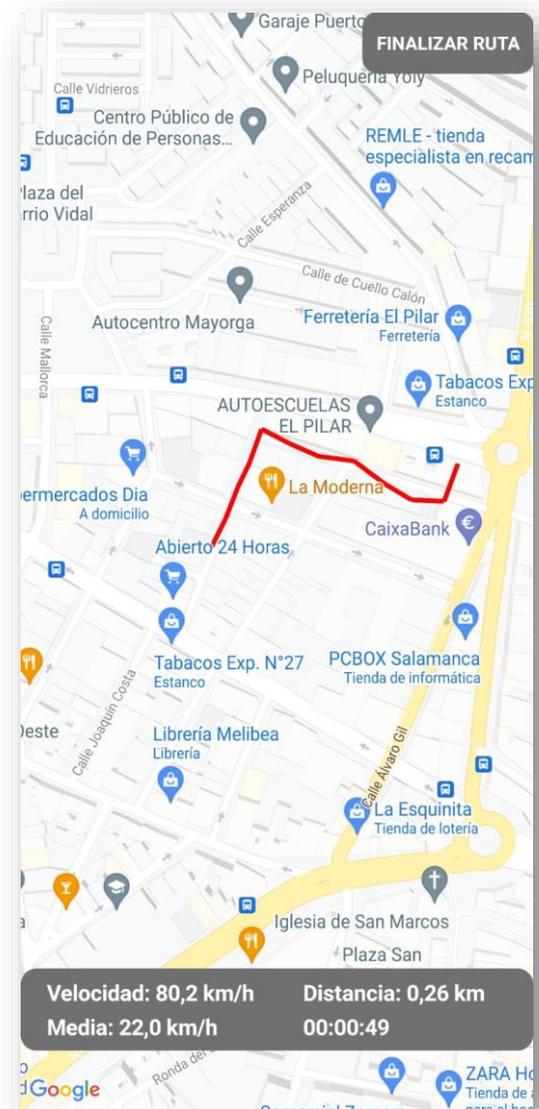


Figura 40. Finalización ruta

9. Conclusiones y líneas de trabajo futuras

Una vez terminada la aplicación, se puede decir que se han conseguido cumplir todos los objetivos que hemos explicado a lo largo de los anexos y que vamos a recordar en este apartado.

- **Es capaz de conocer y almacenar el estado del entorno**, puesto que tiene información acerca del desgaste de cada componente, así como el nuevo desgaste generado.
- **Interacciona con el entorno ya** que es capaz de calcular parámetros de las rutas como pueden ser velocidad máxima, velocidad actual, distancia, tiempo invertido...
- **Presentar información útil sobre el usuario y la bicicleta** ya que, como hemos nombrado ya, el sistema es capaz de mostrar todos los datos de los usuarios, bicicletas, estadísticas, teniendo apartados específicos en la App para ello.
- **Alta, modificación y eliminación de usuarios:** En los que la aplicación controla satisfactoriamente que distintos usuarios puedan crearse una cuenta, darla de baja o modificar sus datos personales, foto de perfil...
- **Añadir, ver y eliminar amigos:** El usuario tiene la posibilidad de añadir amigos en la aplicación a través de la búsqueda del Nick, atributo de cada usuario que será único dentro de la aplicación, también estará toda la gestión de solicitudes de amistad y su eliminación.
- **Creación, visualización, compartición de rutas:** En las cuales los distintos usuarios cada vez que salgan con su bicicleta pueden crear una ruta y luego observar todos los parámetros del desgaste que han realizado, también podrán compartir las rutas a través de redes sociales.
- **Visualizar información del desarrollador:** Se ha creado un apartado donde se puede visualizar la información del desarrollador y la versión de Android para la que está desarrollada la aplicación
- **Visualización de estadísticas:** El usuario puede visualizar información relativa a la distancia total que ha recorrido desde la aplicación, velocidades medias, máximas, distancias...

- **Gestión de usuarios:** El sistema permite la creación de usuarios con sus respectivos perfiles donde ingresaran el tipo de bicicleta que tienen e información básica de usuario, estos datos, se podrán modificar en cualquier momento.
- **Exploración y búsqueda de usuarios:** El sistema facilitará que los distintos usuarios se puedan encontrar con otros usuarios permitiéndoles realizar búsquedas a través de su Nick en la aplicación, permitiendo así observar las rutas, bicicletas... de cada uno de ellos.
- **Competitividad entre usuarios:** La aplicación ofrecerá la posibilidad de visualizar la cantidad de kilómetros totales, tanto tuyos como de las personas que tengas como amigo, de forma que se podrá competir entre distintos usuarios y promoviendo así que utilicen más la aplicación
- **Portabilidad:** Se ha cumplido ya que, al tratar con bicicletas, se ha prestado especial atención a que se use en una plataforma Android,
- **Usabilidad** Creemos que es lo más usable posible para el usuario
- **Privacidad de datos** Se establece una privacidad de datos de forma que ningún usuario podrá acceder a los datos de otro en ningún momento, estando encriptadas las contraseñas y gestionadas por la base de datos

Para concluir, me gustaría decir que el hecho de tratar un tema que me gusta ha sido de gran ayuda a la hora de enfrentarme a este proyecto, puesto que para hacer las pruebas no me quedaba otra que coger mi bicicleta e irme a comprobar que todos los parámetros funcionaban correctamente. También considero que ha sido una gran elección desarrollar en Android mi TFG puesto que como se ha visto estos últimos años, es un entorno que cada vez está creciendo más con más desarrolladores y cada vez más aplicaciones de todo lo que nos podamos imaginar.

Y como implementaciones futuras, se tendría pensado aumentar la cantidad de componentes a los que se les hace mantenimiento, aun sabiendo que se han elegido los más representativos para mantener (frenos, horquilla, transmisión y cadena), está claro que en el mundo de las bicicletas hay gente que cuida hasta el más mínimo detalle de la suya, pudiendo añadir así componentes como bielas, desgaste de las cámaras, ruedas... Así mismo tampoco estaría de más ofrecerle al usuario, a la hora de incorporar un nuevo componente a la bicicleta, otorgarle la posibilidad de añadir una pieza usada, es decir, con algo de desgaste inicial.

Otro punto con el que se podría seguir sería añadir más tipos de bicicleta, puesto que CycleApp “solo” nos permite llevar el seguimiento de mtb rígidas (como sistema de suspensión solo una horquilla), pudiendo implementarse así ideas de bicicletas dobles, bicicletas eléctricas (con sus respectivos mantenimientos de todo el sistema eléctrico), o incluso bicicletas de carretera y de paseo.

Finalmente, el último punto que a priori podría mejorarse sería fomentar la comunicación interna en la sección de amigos. Esto podría hacerse implementando un chat interno en el que puedas compartir tus rutas (ya no solo por RRSS, sino por la propia app), poder poner comentarios, y poder tener una lista de rutas favoritas.

10. Bibliografía

✚ Introducción:

- Revista DGT Aumento del uso de la bicicleta <https://revista.dgt.es/es/noticias/nacional/2020/02FEBRERO/0203-Barometro-de-la-bicicleta-2019.shtml> [Accedido: XX-XX-XXXX]
- Uso bicicleta en tiempos Covid Organización Panamericana de la salud y Organización Mundial de la salud. <https://www.paho.org/es/noticias/18-5-2020-tiempos-covid-19-uso-bicicleta-es-imperativo-para-mantenernos-sanos-saludables> [Accedido: 22-03-2021]