



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA
MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN

**BirdWatchingApp: App sobre avistamiento de
aves**

Alumno/a: D. José Luis Sánchez Paredes



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA
MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN

BirdWatchingApp: App sobre avistamiento de aves

Alumno/a: D. José Luis Sánchez Paredes

Tutora: D^a. Margarita Gonzalo Tasis

Le dedico este trabajo de fin de grado a mi familia.

Agradecimientos

A mi familia,

mi padre, mi madre, mi tía, mis hermanos y a mi abuela,
por estar ahí siempre que necesito ayuda y darme ánimos
porque a su lado puedo hacer cualquier cosa.

A mis amigos,

por apoyarme en todo y saber distraerme cuando estoy agobiado.

A mis compañeros de carrera,

porque cuando tenía un problema siempre tenían una solución que darme, así como consejos para mejorar.

A mi tutora Margarita,

porque sin su ayuda esto no hubiera sido posible.

Resumen

Este proyecto trata de realizar una aplicación para el sistema operativo Android, dado que es el más extendido entre los dispositivos móviles. Esta aplicación será de ayuda y soporte a aquellas personas amantes de la naturaleza y en especial de las aves, en la búsqueda de las mismas.

Con este trabajo se pretende consultar las distintas aves que hay en Castilla y León así como las características sobre ellas y sus fotos. Se podrá diferenciar las aves por distintos criterios como área, nombre común, nombre científico o temporada del año en la que está en ciertos lugares. Además se podrá guardar la información necesaria para el avistamiento de aves y ver los avistamientos del resto de usuarios. Otra funcionalidad es la de cumplir retos para impulsar el uso de la aplicación.

Abstract

This project is the Bachelor Final Project for Computer Engineering of the University of Valladolid. The project consists in the creation of a mobile application for Android that helps all the bird's lovers users while they go sighting.

With this project, it will be possible to search any kind of bird in Castilla y León and their characteristics, as well as photos. It will be possible to distinguish the birds by: area, common name, scientific name or the season of the year. Moreover, it will be possible to storage their sightings in their smartphone and see others users sightings. In addition, the app provides challenges to users, so that they can get achievements after they reach them.

Índice general

Agradecimientos	III
Resumen	V
Abstract	VII
Lista de figuras	XIX
Lista de tablas	XXII
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.2.1. Objetivos generales	2
1.2.2. Objetivos específicos	2
1.3. Estructura de la memoria	2
2. Estudio de aplicaciones similares	5
2.1. Birds Check	5
2.2. eBird	9
2.3. Merlin	12
2.4. Conclusiones	14
	IX

3. Planificación	15
3.1. Estimación temporal	15
3.2. Iteraciones	16
3.3. Análisis de riesgos	16
3.3.1. Riesgos	17
3.4. Calendarización	19
3.5. Hitos	21
3.6. Recursos	21
3.6.1. Hardware	21
3.6.2. Software	22
3.7. Organización del proyecto	22
3.7.1. Estructura interna de la organización	22
3.7.2. Roles y sus responsabilidades	23
3.8. Estimación de costes	24
3.8.1. Coste	24
3.9. Seguimiento	25
3.9.1. Hitos	26
4. Análisis	27
4.1. Stakeholders	27
4.2. Requisitos	28
4.2.1. Requisitos funcionales	28
4.2.2. Requisitos no funcionales	33
4.3. Casos de uso	35
4.3.1. Diagrama de casos de uso	35
4.3.2. Matriz de requisitos / Casos de uso	36
4.3.3. Especificación de los casos de uso	37

4.4. Modelo de Dominio	43
4.5. Diagramas de actividad	45
5. Arquitectura y diseño	57
5.1. Arquitectura	57
5.1.1. Arquitectura del software	57
5.1.2. Ciclo de vida de Actividades y Fragmentos	58
5.2. Diseño	60
5.2.1. Diseño de la base de datos	60
5.2.2. Principios de diseño	62
5.2.3. Guías de diseño	62
5.2.4. Diagramas de interacción	63
5.2.5. Diagramas de clases de diseño	74
5.2.6. Prototipo de bajo coste	76
5.2.7. Mejoras con respecto al prototipo	82
6. Implementación	85
6.1. Aplicación android	85
6.1.1. Tecnología empleada	85
6.1.2. Entorno de desarrollo	86
6.1.3. Versiones de implementación	86
6.1.4. Decisiones de implementación	86
6.2. API	87
6.2.1. Tecnología empleada	87
6.2.2. Entorno de desarrollo	88
6.2.3. Recursos y bibliotecas externas utilizadas	88
6.2.4. Entorno	88

7. Pruebas	89
7.1. Alcance de las pruebas	89
7.2. Requisitos de las pruebas	89
7.3. Tipos de pruebas	90
7.3.1. Pruebas de caja blanca	90
7.3.2. Pruebas de caja negra	90
7.4. Resultados	91
8. Conclusiones	97
8.1. Líneas futuras	97
A. Manual de usuario	99
A.1. Introducción	99
A.2. Funcionalidades	99
A.2.1. Registrarse	99
A.2.2. Acceder	100
A.2.3. Ver aves disponibles y ordenarlas por distintos criterios	101
A.2.4. Ver la descripción de un ave	104
A.2.5. Crear avistamiento	104
A.2.6. Consultar tus avistamientos	105
A.2.7. Actualizar un avistamiento	106
A.2.8. Borrar avistamiento	106
A.2.9. Consultar los avistamientos de todos los usuarios	106
A.2.10. Consultar los retos	107
A.2.11. Consultar mis logros	108
A.2.12. Actualizar los datos del usuario	109
A.2.13. Borrar mi usuario	109

B. Manual de instalación	111
C. Contenidos del CD-ROM	113
Bibliografía	115
Webgrafía	120

Índice de figuras

2.1. Birds Check - Pantalla inicial y Menús	6
2.2. Birds Check - Retos y Logros	6
2.3. Birds Check - Ejemplo de características de un ave	7
2.4. eBird - Pantalla principal y opciones disponibles	9
2.5. eBird - Creación de una nueva lista	10
2.6. Merlin - Selección de área geográfica	12
3.1. Estimación temporal	20
3.2. Diagrama de Gantt	20
3.3. Estructura interna de la organización	23
4.1. UVa	27
4.2. Tutora	28
4.3. Alumno	28
4.4. Ver aves disponibles	29
4.5. Consultar aves por criterios	29
4.6. Añadir avistamiento	29
4.7. Consultar descripción de un ave	30
4.8. Consultar retos	30
4.9. Consultar mis avistamientos	30

4.10. Consultar logros	31
4.11. Consultar avistamientos de todos los usuarios	31
4.12. Modificar un avistamiento propio	31
4.13. Eliminar un avistamiento propio	32
4.14. Registrarse	32
4.15. Acceder a la aplicación	32
4.16. Actualizar datos del usuario	33
4.17. Borrar mi usuario	33
4.18. Tiempo de arranque	34
4.19. Tiempo de refresco	34
4.20. Versión de Android	34
4.21. Diagrama de casos de uso	35
4.22. Matriz de requisitos-casos de uso	36
4.23. Consultar aves por distintos criterios	37
4.24. Consultar la descripción de un ave	37
4.25. Crear avistamiento	38
4.26. Consultar los retos	38
4.27. Borrar mi usuario	39
4.28. Actualizar datos de mi usuario	39
4.29. Consultar logros	40
4.30. Consultar mis avistamientos	40
4.31. Modificar avistamiento propio	41
4.32. Eliminar avistamiento propio	41
4.33. Registrarse	42
4.34. Acceder a la aplicación	42
4.35. Consultar todos los avistamientos	43

4.36. Modelo de dominio	44
4.37. Diagrama de actividad del caso de uso CU01	45
4.38. Diagrama de actividad del caso de uso CU02	46
4.39. Diagrama de actividad del caso de uso CU03	47
4.40. Diagrama de actividad del caso de uso CU04	48
4.41. Diagrama de actividad del caso de uso CU05	49
4.42. Diagrama de actividad del caso de uso CU06	50
4.43. Diagrama de actividad del caso de uso CU07	51
4.44. Diagrama de actividad del caso de uso CU08	51
4.45. Diagrama de actividad del caso de uso CU09	52
4.46. Diagrama de actividad del caso de uso CU10	53
4.47. Diagrama de actividad del caso de uso CU11	54
4.48. Diagrama de actividad del caso de uso CU12	55
4.49. Diagrama de actividad del caso de uso CU13	56
5.1. Modelo Vista Controlador	58
5.2. Ciclo de vida de una actividad	59
5.3. Ciclo de vida de un fragmento mientras su actividad esta activa	60
5.4. Diagrama Entidad-Relación	61
5.5. Diagrama de secuencia 1	64
5.6. Diagrama de secuencia 2	65
5.7. Diagrama de secuencia 12	65
5.8. Diagrama de secuencia 3	66
5.9. Diagrama de secuencia 4	67
5.10. Diagrama de secuencia 5	68
5.11. Diagrama de secuencia 6	68
5.12. Diagrama de secuencia 7	69

5.13. Diagrama de secuencia 8	70
5.14. Diagrama de secuencia 9	71
5.15. Diagrama de secuencia 10	72
5.16. Diagrama de secuencia 11	72
5.17. Diagrama de secuencia 13	73
5.18. Diagrama del modelo	74
5.19. Diagrama del presentador	74
5.20. Diagrama de la vista - Actividades	75
5.21. Diagrama de la vista - Fragmentos	75
5.22. Prototipo: Pantalla inicial para acceder y registro	76
5.23. Prototipo: Pantalla del menú principal	77
5.24. Prototipo: Pantalla detalle de un ave	78
5.25. Prototipo: Pantalla para introducir un nuevo avistamiento	79
5.26. Prototipo: Pantalla de los avistamientos de todos los usuarios	79
5.27. Prototipo: Pantalla de mis avistamientos	80
5.28. Prototipo: Pantalla de mis logros	81
5.29. Prototipo: Pantalla de los retos	81
5.30. Prototipo: Pantalla de ajustes de usuario	82
A.1. Registrarse en la aplicación	100
A.2. Acceder a la aplicación	100
A.3. Menú principal	101
A.4. Menú para ordenar las aves	102
A.5. Ordenadas por nombre	102
A.6. Ordenadas por áreas	103
A.7. Ordenadas por temporada	103
A.8. Descripción de un ave	104

A.9. Crear un nuevo avistamiento	105
A.10.Mis avistamientos	105
A.11.Avistamiento	106
A.12.Avistamientos de todos los usuarios	107
A.13.Retos	107
A.14.Mis logros	108
A.15.Ajustes	109
B.1. Instalar aplicaciones de terceros	112

Índice de tablas

3.1. Fases	15
3.2. Estimación semanal	16
3.3. Iteraciones	16
3.4. Roles y responsabilidades	23
3.5. Asignación de roles	24
7.1. Prueba 01- Registrarse relleno los datos	91
7.2. Prueba 02- Registrarse sin rellenar todos los campos	91
7.3. Prueba 03- Acceder a la aplicación con datos correctos	91
7.4. Prueba 04- Acceder a la aplicación con datos incorrectos	92
7.5. Prueba 05- Ver aves por nombre común	92
7.6. Prueba 06- Ver aves por nombre científico	92
7.7. Prueba 07- Ver aves por área	92
7.8. Prueba 08- Ver aves por temporada	93
7.9. Prueba 09- Ver aves por otro criterio	93
7.10. Prueba 10- Ver la descripción de un ave	93
7.11. Prueba 11- Consultar sus avistamientos	93
7.12. Prueba 12- Consultar sus avistamientos	93
7.13. Prueba 13- Crear avistamientos relleno los campos	94
7.14. Prueba 14- Crear avistamientos sin rellenar todos los campos	94

7.15. Prueba 15- Crear avistamientos con datos erróneos que no existen en la aplicación	94
7.16. Prueba 16- Modificar avistamiento propio	94
7.17. Prueba 17- Modificar avistamiento ajeno	94
7.18. Prueba 18- Eliminar avistamiento propio	95
7.19. Prueba 19- Eliminar avistamiento ajeno	95
7.20. Prueba 20- Consultar los avistamientos de todos los usuarios	95
7.21. Prueba 21- Consultar los retos	95
7.22. Prueba 22- Consultar mis logros	95
7.23. Prueba 23- Borrar usuario	96
7.24. Prueba 24- Borrar otro usuario	96
7.25. Prueba 25- Modificar usuario	96
7.26. Prueba 26- Modificar otro usuario	96

Capítulo 1

Introducción

En esta sección se presentarán los conceptos recogidos en la memoria, la motivación de haber escogido este TFG, los objetivos de éste y la estructura de esta memoria.

1.1. Motivación

Los móviles hoy en día son un instrumento indispensable de cualquier persona para ayudarle en múltiples tareas: desde servirle para hacer fotos, escuchar música, hasta para acceder al correo electrónico o pagar con el móvil. Las aplicaciones del móvil son numerosas, porque pueden ser aplicadas en cualquier tipo de actividad y para cualquier tipo de usuario. Por ello en la disciplina de la ingeniería informática es importante conocer cómo se construyen y crean aplicaciones para cualquier tipo de teléfono móvil, ya sea con sistema operativo iOS o Android.

Las aplicaciones que más se demandan son aquellas que tienen que ver con el aspecto lúdico de la población: juegos, entretenimiento, música o hobbies. Un hobby que está en auge sobre todo para un amplio espectro de la población y el turismo, es el avistamiento de aves. Las personas que tienen este hobby necesitan aplicaciones sencillas y eficaces que les ayuden a llevar el seguimiento de las aves avistadas.

Este TFG trata de dar respuesta a esa necesidad.

1.2. Objetivos

1.2.1. Objetivos generales

El trabajo de fin de grado consiste en cómo elaborar un proyecto que resuma la máxima cantidad de competencias adquiridas durante los años de estudio en este grado. En el trabajo fin de grado se aplicarán e integrarán las habilidades, conocimientos y actitudes adquiridos durante el transcurso de estos años. Los objetivos generales de este trabajo fin de grado son:

- Profundizar sobre las aplicaciones *Android* y poder diseñar soluciones a medida a las necesidades encontradas.
- Realizar un estudio que integre la búsqueda, análisis y organización de la información necesaria para realizar este Trabajo de Fin de Grado.
- Trabajar en el proyecto según los distintos roles necesarios, saber identificarlos y realizar sus tareas, tales como jefe de proyecto, analista, programador, etc.
- Aplicación de métodos de desarrollo de producto y de trabajo correctos para la planificación de un proyecto.
- Elaborar y defender una presentación de este trabajo ante un tribunal de profesores de la Universidad de Valladolid.

1.2.2. Objetivos específicos

Además de lograr y mostrar esa serie de competencias, existen unos objetivos particulares referentes a la temática de este trabajo como son:

- Construir una aplicación móvil para Android.
- Poder mostrar diferentes características de las aves de la zona.
- Mostrar imágenes de las aves.
- Poder mostrar las aves por distintos criterios.
- Poder guardar los avistamientos del usuario.
- Poder ver los avistamientos de otros usuarios.

1.3. Estructura de la memoria

Aquí se muestra el esquema general de la memoria realizada, describiendo brevemente su contenido:

- **Estudio de aplicaciones similares:** Análisis de aplicaciones similares que existen actualmente en el mercado, Google Play. Incluyendo una descripción, opiniones a favor y en contra, así como una conclusión de cada una de ellas y una general.
- **Planificación:** descomposición del proyecto en fases e iteraciones, que serán estimadas en tiempo y recursos para realizar una buena planificación. Además de incluirse el análisis de riesgos y el seguimiento en este apartado.
- **Análisis:** Descripción de la fase de análisis de la Ingeniería de Software
- **Arquitectura y diseño:** Detalles más relevantes de la arquitectura y diseño de la aplicación.
- **Implementación:** Detalle de la fase de implementación de la Ingeniería de Software.
- **Pruebas:** Pruebas realizadas de la aplicación.
- **Conclusiones:** Conclusiones obtenidas de la realización de este proyecto y su resultado. Así como líneas futuras de trabajo.
- **Apéndices:** Información complementaria relativa al proyecto que puede resultar de interés para el lector y en ocasiones provee una visión más extensa de algunos de los temas tratados. Se incluyen los manuales de instalación y usuario.
- **Bibliografía:** Referencias usadas durante este proyecto tanto de libros como enlaces a páginas web.

Toda la documentación generada en el proceso de desarrollo, así como el código fuente se encuentra disponible en el CD adjunto, cuya estructura se describe en el Apéndice C

Capítulo 2

Estudio de aplicaciones similares

En este capítulo se describe el conjunto más destacado de aplicaciones android encontradas, cuyo objetivo principal o secundario es mostrar al usuario, información sobre las aves y almacenar los posibles avistamientos en una zona determinada.

2.1. Birds Check

La primera aplicación relevante es la de *Birds Check*. La información siguiente ha sido obtenida de Google Play o de la propia aplicación.

- **Nombre de la aplicación:** Birds Check
- **Nombre del desarrollador:** BirdsCheck
- **Valoración en Google Play:** 4,7 de 5 estrellas
- **Número de descargas:** 5000

A continuación algunas capturas de pantalla para mostrar las principales funciones de la aplicación.

2.1. BIRDS CHECK

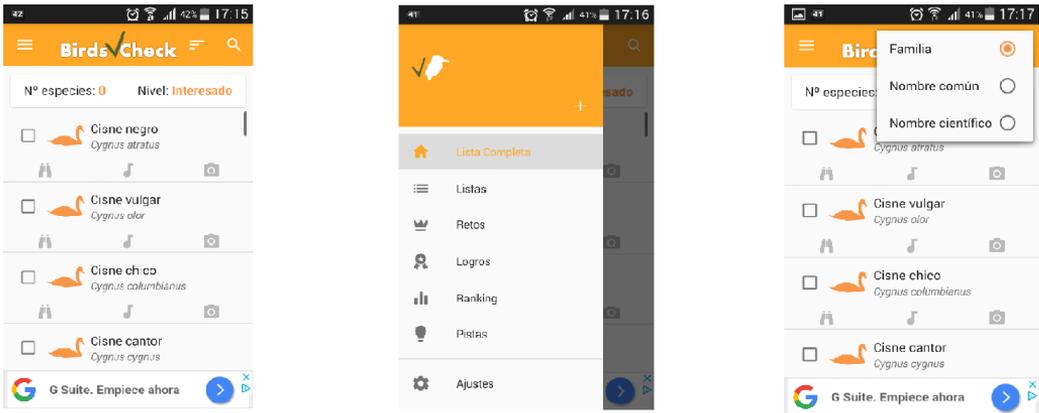


Figura 2.1: Birds Check - Pantalla inicial y Menús

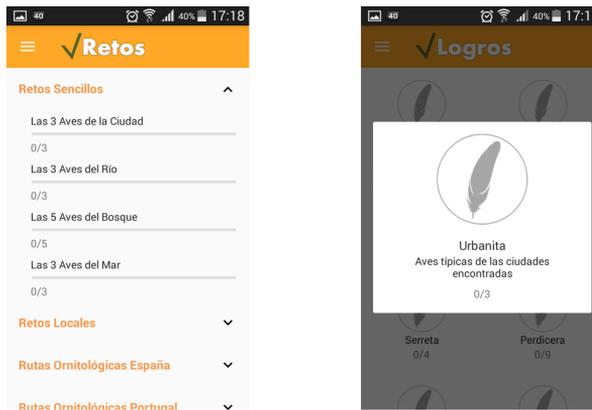


Figura 2.2: Birds Check - Retos y Logros

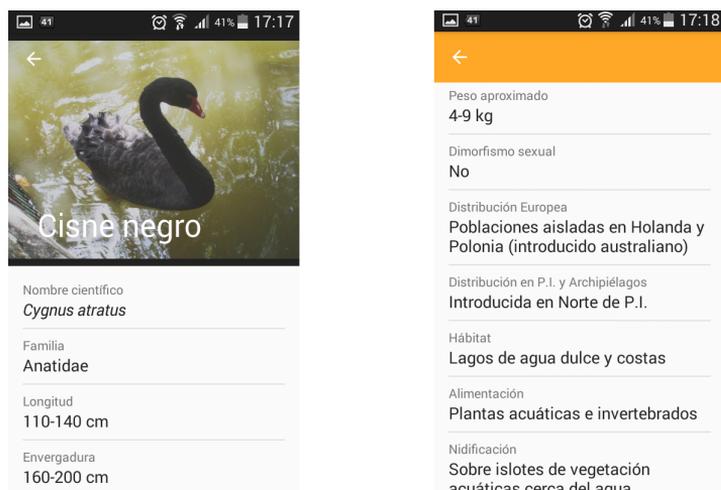


Figura 2.3: Birds Check - Ejemplo de características de un ave

■ **Funcionalidades:**

● **Versión gratuita:**

- Registrar tus avistamientos.
- Consultar las fichas de tus aves favoritas, con su nombre común, nombre científico, familia y otras características.
- Ordenar la lista por familia, nombre común o científico.
- Retos para el usuario.
- Logros dependiendo de que retos has superado y posibilidad de subir de nivel en el ranking.
- Crear listas.
- Centrado en España y Portugal.
- Acceder al ranking de usuarios.
- Consultar consejos para ver nuevas especies de aves.
- Contiene anuncios.

● **Versión de pago:**

- Mismas características que la gratuita.
- No tiene publicidad.

- **Precio:** la aplicación es gratuita ya que contiene anuncios. Existe una versión de pago Birds Check Pro, que cuesta 1,99 euros.

■ **Opiniones de los usuarios de Google Play**

● **Opiniones a favor**

- Los retos: puedes elegir una lista de aves y verlas todas como objetivo.

- Sencilla de utilizar.
- Permite crear listas a tu gusto de tu zona.
- Permite ponerte retos creados por la aplicación y subir de niveles como un juego.

● Opiniones en contra

- No puedes crear tus propios retos.
- En la lista de aves aparece una silueta en vez de una imagen.
- Se puede filtrar por muy pocos criterios de búsqueda, solo por: familia, nombre común y nombre científico. No se puede filtrar por ejemplo por: estación del año o zona geográfica.
- La aplicación funciona sin internet y la base de datos es local a cada móvil, y si cambias de móvil pierdes tus datos.
- No puedes poner comentarios ni imágenes a tu avistamiento.

■ Conclusiones

- La aplicación es sencilla, es decir, que es fácil de aprender y aplica los criterios de usabilidad.
- Los retos son un aliciente para los usuarios ya que pueden ponerse metas.
- Así mismo, los logros y niveles que obtiene el usuario al completar retos son una recompensa que influye en el usuario para que use más la aplicación.
- Aunque hay un ranking global, la aplicación tiene una base de datos de avistamientos solo disponible para el propio usuario, por lo tanto, no hay casi interacción entre usuarios.

2.2. eBird

Esta aplicación es la más significativa de acuerdo a un experto ornitólogo de SEO/Birdlife. Es la más popular entre expertos y tiene además, una buena posición en los resultados de la búsqueda para aquellos usuarios noveles.

- **Nombre de la aplicación:** eBird
- **Nombre del desarrollador:** Cornell Lab of Ornithology
- **Valoración en Google Play:** 4,3 de 5 estrellas
- **Número de descargas:** 100.000

A continuación se muestran unas capturas de pantallas de la aplicación:



Figura 2.4: eBird - Pantalla principal y opciones disponibles

En la figura 2.4 vemos la pantalla inicial, después de haberte registrado y haber confirmado tu cuenta en tu correo electrónico.

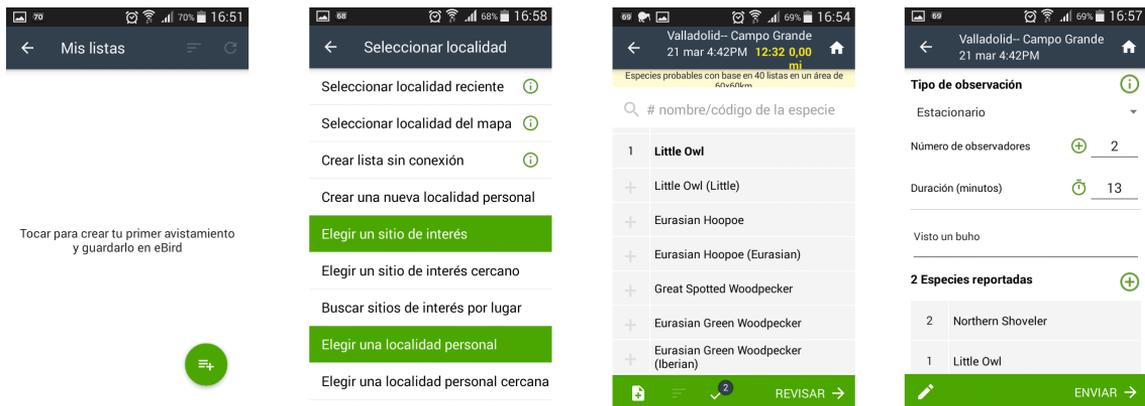


Figura 2.5: eBird - Creación de una nueva lista

En la figura 2.5 vemos el proceso de crear un avistamiento: primero pulsamos en botón de nueva lista, bien en la pantalla principal o en el apartado de *Mis Listas*. Después elegimos la ubicación y el tiempo de inicio del avistamiento. Una vez hecho esto, empieza a correr un contador de tiempo y te aparece la lista de aves para que señales cuales has visto. Y al final, todo eso se guarda en tu avistamiento, que se agrega a tus listas.

■ Funcionalidades:

- Ingresar observaciones de aves desde cualquier lugar del mundo.
- Taxonomía mundial completa basada en la Lista de Clements de las Aves del mundo.
- Nombres comunes disponibles en idiomas locales, en cada país esta disponible en su idioma.
- Lista personalizada para tu ubicación y época del año, mostrando las especies más probables basados en datos de eBird.
- Alertas para las especies raras durante la entrada de datos y en la lista.
- Funcionalidad para *añadir especies* y números durante la pajareada.
- Ubicación por GPS para asegurar la precisión de la localidad.
- Herramienta de mapa que permite la selección de lugares de interés de eBird.
- Funcionalidad completa fuera de línea, lo que permite su uso en lugares sin Internet.
- Funcionalidad de crear tus listas.
- Toda la app está traducida al alemán, mandarín, francés, español, portugués (Brasil) y turco.
- Gratuita, no tiene versión de pago.

■ Precio: Gratuito.

■ **Opiniones de los usuarios de Google Play**

● **Opiniones a favor**

- Imprescindible para los observadores de pájaros.
- Te permite introducir los datos de las aves que ves en el mismo lugar dónde los ves.
- Te mide la distancia que recorres durante el avistamiento.
- Muy útil para guardas tus listas.

● **Opiniones en contra**

- No es fácil de aprender a usar.
- Te obliga a registrarte. En ocasiones, se ha podido observar que esta función de registro falla.
- Hay problemas en sincronizar las listas de la aplicación web y a la del móvil.
- No permite cancelar un avistamiento, tienes que acabarlo, después rellenarlo y luego ya borrarlo.

■ **Conclusiones**

- Es una buena opción para guardar tus listas de avistamientos.
- No tiene facilidad de aprendizaje, pues tiene bastantes opciones y a veces es confuso que elección tienes que escoger.
- La opción de poder guardar tu lista de avistamientos fuera de línea, sin Internet, es una muy buena funcionalidad.
- La opción de determinar tu ubicación bien por GPS o bien introduciéndola directamente, es muy precisa y útil.

2.3. Merlin

Esta aplicación se anuncia como una guía de pájaros y reconocimiento de estos.

- **Nombre de la aplicación:** Merlin Bird ID de Cornell Lab
- **Nombre del desarrollador:** Cornell Lab of Ornithology
- **Valoración en Google Play:** 4,1 de 5 estrellas
- **Número de descargas:** 100.000

A continuación se muestran unas capturas de pantalla de la aplicación:

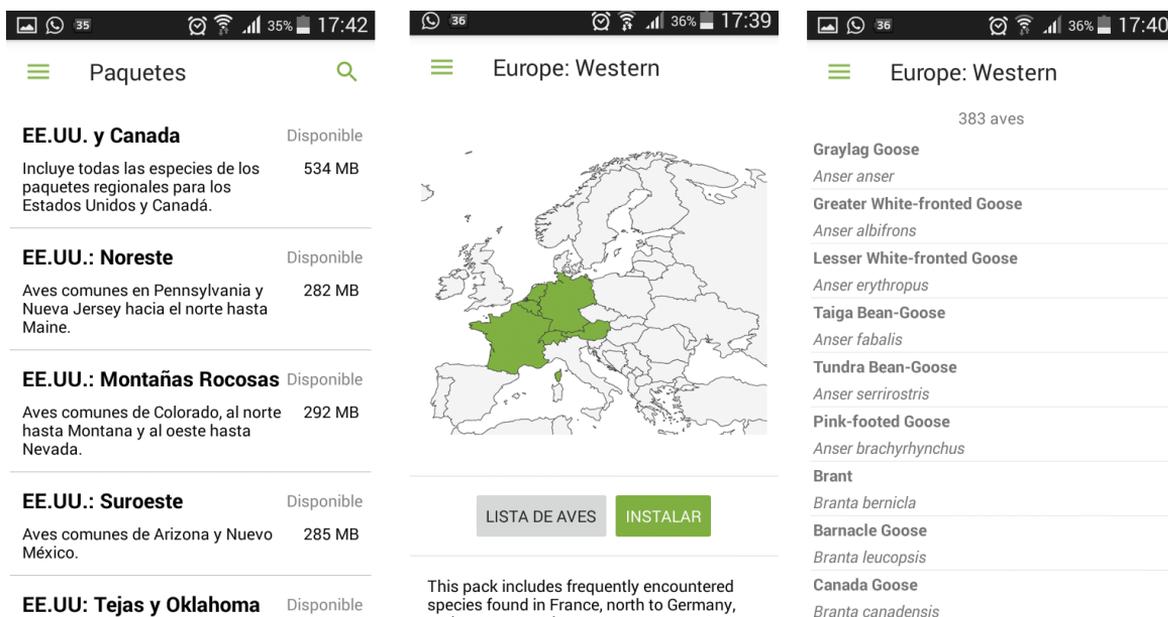


Figura 2.6: Merlin - Selección de área geográfica

Como se puede ver, no hay opción de elegir la zona de España. Por eso no se ha podido probar esta aplicación en la práctica.

■ **Funcionalidades:**

- Permite identificar un ave mediante cinco preguntas o mediante una fotografía.
- Puedes subir una foto indicando el lugar y la hora para identificarla.
- Por cada ave tiene varias fotos y sonidos asociados.
- Tiene una descripción de cada ave y sus principales características.
- Te da consejos sobre cómo encontrar las aves.
- Gratuita, no tiene versión de pago.

■ **Precio:** Gratuito.

■ **Opiniones de los usuarios de Google Play**

● **Opiniones a favor**

- Buena aplicación para identificar aves ya sea por foto o por sonido.
- Bastante precisa.
- Gran base de datos.
- Fácil de usar y fiable.

● **Opiniones en contra**

- Falta poner los nombres científicos.
- La aplicación ocupa lo normal para cualquier aplicación, pero el almacenamiento requerido aumenta drásticamente con cada paquete de mapas que te descargas, casi 200 MB por cada paquete.
- Falta poder ordenar por localidad, pues al final es lo que te interesa, saber las aves de la zona dónde te encuentras.
- No puedes organizar ninguna lista de tus aves favoritas, ya vistas u otras posibilidades.

■ **Conclusiones**

- Según las opiniones, es bastante buena identificando aves mediante una fotografía. Asimismo, te ayuda a reconocerlas mostrándote sonidos de dichas aves.
- Centrado solo en unas zonas específicas de cada continente: por ejemplo, en Europa solo te deja elegir entre Reino Unido, Francia a Alemania y Escandinavia, dejando de lado el resto de países. Como no se centra en España los usuarios de aquí no podrán usarla, a no ser que viajen al extranjero. Por muy buenas funcionalidades que tenga, no es aplicable a nuestro territorio.
- Tienes que descargarte el paquete de aves dependiendo de la zona que quieras. Esto quiere decir que, ocupa muchos megabytes en el dispositivo, esto limita la cantidad de dispositivos que pueden descargarla.
- Tiene una base de datos muy extensa y completa, lo cual es bueno, aunque sacrificas el espacio en el dispositivo.

2.4. Conclusiones

Se han mostrado una serie de aplicaciones que ya existen y que cubren la necesidad de los aficionados a las aves. Aunque el principal problema es que hay muchas aplicaciones pero pocas centradas en España y por lo tanto que sean de utilidad para los usuarios aquí en nuestra zona geográfica.

De las aplicaciones que son aplicables a España, muchas se centran en los sonidos de las aves. De las pocas en las que hay imágenes y características de las aves, se pueden mejorar bastante. Por ejemplo, las principales funciones que se echan en falta, es la de tener una base de datos general para todos los usuarios y compartir avistamientos entre los usuarios. Otra funcionalidad que falta es la de poder subir las fotos de aves del usuario a esa base de datos.

Por lo tanto este proyecto se centra en mejorar las funcionalidades mencionadas anteriormente, a parte de implementar otras funcionalidades básicas e imprescindibles para una aplicación de este tipo.

Capítulo 3

Planificación

A continuación se detalla la planificación temporal del proyecto. Se ha elegido de modelo de proceso el Proceso Unificado, así pues, la planificación y las fases están muy bien definidas, quedando sólo por definir la cantidad de iteraciones que se van a realizar en cada fase.

3.1. Estimación temporal

Como el proyecto tiene una dificultad media, se ha considerado las siguientes fases e iteraciones dentro de cada fase, con una estimación temporal que se muestra en la siguiente tabla 3.1:

Dominio	Inicio	Elaboración	Construcción	Transición
Esfuerzo	5 %	20 %	65 %	10 %
Tiempo	10 %	30 %	50 %	10 %

Tabla 3.1: Fases

- Fase de **inicio**: es la primera fase del proyecto. Consiste en planear el proyecto, definir los objetivos y ver el alcance del mismo.
- Fase de **elaboración**: es la segunda fase en la que se centra en analizar las tecnologías a emplear y la arquitectura que se va a usar.
- Fase de **construcción**: es la tercera fase en la que se crea el propio producto con los objetivos definidos anteriormente.
- Fase de **transición**: es la última fase en la que se entrega el producto y se lidia con los problemas de implementar el producto en el mercado.

3.2. ITERACIONES

El proyecto se ha empezado el 5 de Febrero, coincidiendo con el principio del cuatrimestre y el final se tiene planificado para el 27 de Junio. Por lo tanto se tienen casi 5 meses, 20 semanas aproximadamente, para acabar el proyecto.

Por lo tanto, con las fases anteriormente mencionadas y sus porcentajes, se obtiene la estimación semanal de la tabla :

Fase	Duración
Inicio	2
Elaboración	6
Construcción	10
Transición	2

Tabla 3.2: Estimación semanal

3.2. Iteraciones

Según la dificultad normal del proyecto, el tiempo disponible y la experiencia, se han elegido las siguientes iteraciones en cada fase. Se han elegido hacer dos iteraciones en la fase de elaboración y construcción porque son las más largas y complejas. Todo esto está recogido en la tabla 3.3:

Inicio	Elaboración	Construcción	Transición	Total
1	2	2	1	6

Tabla 3.3: Iteraciones

3.3. Análisis de riesgos

Según el PM-BOK (1), *“Un riesgo es un evento o una condición inciertos que, si ocurren, tienen un efecto positivo o negativo sobre los objetivos del proyecto”*. Está caracterizado por la probabilidad de que ocurra y el tamaño de la pérdida que generaría en el caso de que ocurriera. Los riesgos se pueden clasificar en las siguientes categorías:

- **Riesgos del proyecto:** afectan a la planificación temporal, al coste y a la calidad del proyecto.
- **Riesgos técnicos:** amenazan a la planificación temporal y calidad del producto software a desarrollar.
- **Riesgos de negocio:** amenazan la viabilidad del sistema a construir. Tiene varios tipos:

- **Riesgos de mercado:** desarrollar un producto excelente pero que no tiene salida o a nadie le interesa.
- **Riesgos estratégicos:** desarrollar un producto que no encaje con la estrategia comercial de la empresa.
- **Riesgos de ventas:** construir un producto que no se sepa cómo venderlo.
- **Riesgos de dirección:** perder el apoyo de la dirección por cambio de personal o de enfoque.
- **Riesgos de presupuesto:** no se mantienen los recursos asignados tanto en presupuesto como en personal asignado.

El proceso de la **gestión de los riesgos** está dividido en las siguientes etapas:

- Identificación de riesgos
- Análisis de riesgos
- Control de riesgos
- Monitorización de riesgos

A continuación se detalla cómo se ha realizado esta gestión de riesgos . Se han elegido los siguientes aspectos a tratar de cada riesgo:

- Problema
- Probabilidad de ocurrencia
- Impacto
- Valoración
- Plan de actuación

La escala de valores de la probabilidad de ocurrencia y el impacto es nominal, teniendo cinco valores, yendo de menor a mayor valor *muy baja, baja, media, alta, muy alta* la probabilidad y de *muy bajo, bajo, medio, alto, muy alto* el impacto. Cuanta mayor sea la valoración más peligroso es el riesgo. Esta valoración sirve para ver si se puede asumir el riesgo (nivel muy bajo o bajo) o se puede controlar (nivel medio) o si se tiene que mitigar por que el nivel es alto o muy alto y esto es inadmisibile.

3.3.1. Riesgos

Estos han sido los riesgos detectados en el análisis de riesgos hecho para este proyecto, y la probabilidad de aparición, el impacto y el plan de actuación de cada uno.

Para ningún riesgo hay que ejecutar ninguna medida preventiva o de mitigación, ya que son de valoración media. Por lo tanto se aceptan y se hace un seguimientos de ellos.

1. Error al hacer la estimación temporal

- **Clasificación:** riesgo de proyecto a nivel organizativo.
- **Problema:** estimación temporal errónea, dado que se estima de manera optimista y no se cumplen los plazos e hitos fijados.
- **Probabilidad de ocurrencia:** media.
- **Impacto:** medio.
- **Valoración:** media.
- **Plan de actuación:** se volverá a planear el proyecto con el tiempo restante, intentando alcanzar todos los objetivos. Aunque se repriorizarán las tareas y objetivos.

2. Retraso en el proyecto

- **Clasificación:** riesgo de proceso, en particular de gestión, aunque puede ser un riesgo de proyecto.
- **Problema:** retraso en el proyecto por falta de disponibilidad para trabajar en él por razones de educativas, laborales o personales.
- **Probabilidad de ocurrencia:** media.
- **Impacto:** medio.
- **Valoración:** media.
- **Plan de actuación:** se volverá a planear el proyecto para adaptarse al nuevo tiempo restante.

3. No disponibilidad de recursos

- **Clasificación:** riesgo de proyecto.
- **Problema:** no se cuenta con algún recurso de hardware por rotura o fallo en él, como puede ser el ordenador.
- **Probabilidad de ocurrencia:** bajo.
- **Impacto:** alto.
- **Valoración:** media.
- **Plan de actuación:** se buscará, tan pronto como sea posible, un recurso alternativo que pueda sustituir las prestaciones del anterior, con lo que se puede seguir con el proyecto.

4. Falta de los conocimientos necesarios

- **Clasificación:** riesgo técnico.
- **Problema:** no se cuenta con los conocimientos necesarios para desarrollar algún apartado del proyecto, y por consecuente se generan errores.
- **Probabilidad de ocurrencia:** medio.

- **Impacto:** medio.
- **Valoración:** media.
- **Plan de actuación:** se debe buscar la raíz del problema y buscar documentación que sirva de apoyo y permita seguir con el proyecto.

5. Proceso de diseño pobre

- **Clasificación:** riesgo de proceso.
- **Problema:** el diseño puede no ser correcto debido a la poca experiencia del alumno.
- **Probabilidad de ocurrencia:** medio.
- **Impacto:** medio.
- **Valoración:** media.
- **Plan de actuación:** se podría añadir una iteración a mayores en la fase de diseño para subsanar este defecto de diseño.

Hay que hacer un seguimiento para todos los riesgos, teniendo en cuenta los principales indicadores como son la planificación y los hitos. Se debe controlar que estos indicadores se cumplan en el plazo fijado.

3.4. Calendarización

A continuación, se pueden ver dos figuras en las que se puede observar al detalle la estimación temporal de cada una de las fases del proyecto, incluyendo también la redacción de esta memoria.

3.4. CALENDARIZACIÓN



Nombre	Fecha d...	Fecha de ...
● Fase de inicio	5/02/18	16/02/18
● Iteración 1	5/02/18	16/02/18
● Fase de elaboración	19/02/18	30/03/18
● Iteración 1	19/02/18	9/03/18
● Iteración 2	12/03/18	30/03/18
● Fase de construcción	2/04/18	8/06/18
● Iteración 1	2/04/18	4/05/18
● Iteración 2	7/05/18	8/06/18
● Fase de transición	11/06/18	22/06/18
★ Iteración 1	11/06/18	22/06/18
● Redacción de la memoria	5/02/18	29/06/18

Figura 3.1: Estimación temporal

Con las estimaciones temporales por fases e iteraciones anteriores, se ha confeccionado el siguiente Diagrama de Gantt:

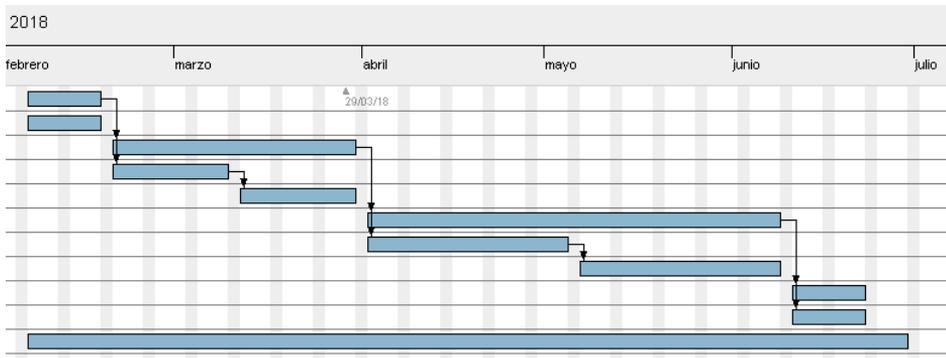


Figura 3.2: Diagrama de Gantt

3.5. Hitos

- Análisis de aplicaciones similares: 9/03/2018
- Prototipo de bajo coste: 30/03/2018
- Prototipo de baja funcionalidad: 9/05/2018
- Versión final de la aplicación: 8/06/2018
- Memoria acabada: 27/06/2018

3.6. Recursos

A continuación se pasa a describir brevemente todos los recursos necesarios para realizar el proyecto, además del programador que lo va a desarrollar.

3.6.1. Hardware

De hardware el único recurso que se necesitará es un ordenador portátil. Aunque opcionalmente se usará un dispositivo móvil para probar la aplicación.

Portátil

Es un recurso indispensable para la realización de este proyecto. Tiene las siguientes características:

- Fabricante: Lenovo
- Procesador: Intel Core i7-6500U CPU @ 2.50GH-2.94GHz
- RAM: 8,00 GBs
- Disco duro: 1TB
- Sistemas Operativos: Windows 10 / Ubuntu 16.4

Dispositivo móvil

Aunque podría valer con el emulador de Android para ejecutar y probar la aplicación, se usará un dispositivo móvil físico con estas características:

- Modelo: Samsung Note 2
- Procesador: 1.6 GHz quad-core Cortex-A9

- RAM: 1,00 GB
- Disco duro: 4GB + 16 GB (tarjeta SD)
- Sistema operativo: 4.4.2 (KitKat)

3.6.2. Software

Los recursos software serán varios, unos solo disponibles en el sistema operativo Windows y otros solo para sistemas Linux (Ubuntu en este caso), mientras que otros recursos funcionan en cualquier sistema operativo.

- **Windows 10:** sistema operativo instalado en el dispositivo de fábrica. No supone un coste adicional.
- **Ubuntu 16.4:** sistema operativo de libre distribución que se usará para la instalación de algunos programas.
- **Gantt Project:** Es una aplicación fácil de usar para entornos Windows y MacOs. Podemos definir jerarquía de tareas y dependencias, diagramas de Gantt, informes en PDF y HTML, importación y exportación de o desde MS Project y gráfico de carga de recursos. En este proyecto lo usamos para la gestión del mismo y planificaciones de recursos, y con licencia GPL.
- **REM 1.2.2:** software para la especificación de requisitos. Gratuita también.
- **Astah Professional:** herramienta de modelado UML 2. La escuela suministra la licencia, así que no supone ningún coste.
- **Android Studio:** recurso muy necesario para el desarrollo de la aplicación móvil para Android. Se trata de un entorno de desarrollo integrado que suministra Google para el desarrollo en Android, siendo totalmente gratuita. Se usará la versión 2.3.3, aunque esta disponible la 3.0.
- **Sharelatex:** software que mediante un cliente web puedes acceder a tus proyectos y editarlos con el procesador de textos Latex. Se usará para redactar este documento.
- **Visual Studio Code:** software para programar código en casi cualquier lenguaje de programación. Usado para programar la API.

3.7. Organización del proyecto

3.7.1. Estructura interna de la organización

A continuación se muestra en la figura 3.3 la estructura de la organización, mostrando los diferentes roles dentro de ella.

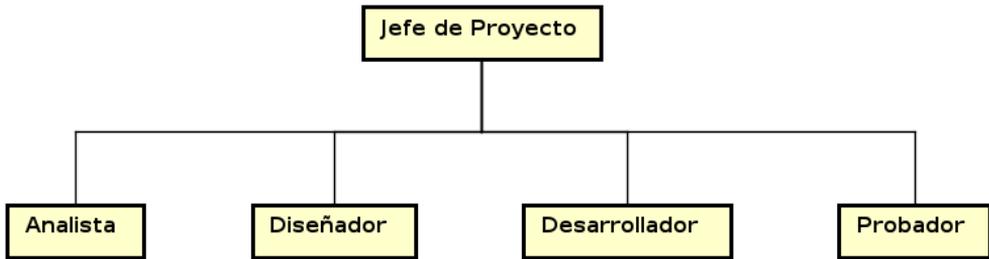


Figura 3.3: Estructura interna de la organización

3.7.2. Roles y sus responsabilidades

Las responsabilidades de cada rol dentro del equipo de desarrollo son:

Roles	Responsabilidades
Jefe de proyecto	Es la persona encargada de la planificación, ejecución y control del proyecto. También es encargado de recordar y cumplir los plazos establecidos para cumplir los objetivos
Analista	Encargado del análisis del sistema, obteniendo los requisitos y validándolos con las necesidades del cliente
Diseñador	Encargado de definir la arquitectura del sistema y de la base de datos
Desarrollador	Implementa los diseños hechos anteriormente, cumpliendo los requisitos obtenidos
Probador	Realiza las pruebas sobre el sistema, con el objetivo de recavar la información necesaria para verificar si se cumplen los objetivos, y si encuentra fallos, poder arreglarlos

Tabla 3.4: Roles y responsabilidades

La asignación de roles por persona es la que aparece en la tabla 3.5:

	Inicio	Elaboración	Construcción	Transición
José Luis	Analista	Analista y Diseñador	Desarrollador	Probador
José Luis	Jefe de proyecto			
Margarita	Gestor de proyecto			

Tabla 3.5: Asignación de roles

3.8. Estimación de costes

En la estimación de costes de este proyecto se tiene que tener en cuenta que el alumno que lo desarrolla es un estudiante de Grado en Ingeniería Informática, mención en Tecnologías de la Información, y por lo tanto, no existirá ningún coste real, pues no va a recibir ningún salario por el trabajo hecho. Y como consecuencia, tampoco habrá ningún coste indirecto.

A pesar de ello, se hará una estimación del coste en función del tiempo trabajado (en número de horas hombre) y suponiendo una remuneración de **11 €** por hora como graduado en Ingeniería Informática, sin entrar a especificar distintos sueldos para los distintos roles desempeñados.

3.8.1. Coste

El coste asociado al software es nulo, pues todo el software necesario o lo proporciona la Universidad, o es gratuito. El coste de los materiales hardware usados es el siguiente:

- **Ordenador:** El coste aproximado del ordenador utilizado es de 650 €. Y se estima unos 5 años de vida útil, y un uso de 300 días al año. Usándolo de media 8h/día. El porcentaje del precio del ordenador usado para este proyecto es de:

$$650€ * 300h/5años * 300días * 8h = 16,25€$$

- **Teléfono móvil:** El coste aproximado del dispositivo móvil utilizado es de 200 €. Usándolo de media 8 horas al día, durante 365 días al año durante sus 2 años de vida útil:

$$200€ * 300h/2años * 365días * 8h = 10,27€$$

El coste del personal necesario para llevar a cabo este proyecto es el siguiente. Como se ha planificado anteriormente, se estima una cantidad de **300 horas** para este trabajo. Llevado a cabo por una única persona. Por lo tanto el coste de los recursos humanos es de:

- 300 horas por 11€/h hace un total de 3300 €.

A lo que hay que sumarle el gasto producido del material usado y el lugar donde se ha realizado el proyecto. Así como el gasto que supondría darse de alta como autónomo y pagar los impuestos correspondientes.

- A un precio medio de 0.1155€/KWh y un consumo de 50 Vatios con el portátil encendido y usándose. El total de consumo eléctrico que sale es de $50\text{Wh} * 300\text{h} * 0.1155\text{€/}1000\text{Wh} = 1,7\text{€}$.
- Suponiendo el gasto en electricidad de la luz de la habitación como otros 50Vatios la hora. Tenemos otro 1,7€.
- Habría que pagar el alquiler del tiempo destinado al proyecto. $300/24=12,5$ días de alquiler, tendría que sumarse al coste. Suponiendo un gasto al mes de 200€/mes incluyendo la calefacción por una habitación. El coste a sumar es de:
 $200 * 12,5\text{días}/30\text{días} = 83,3\text{€}$
- El gasto en una conexión a Internet. Con un gasto medio de 50€al mes. Teniendo que pagar $300/24=12,5$ días de Internet. El coste a sumar es de:
 $50 * 12,5\text{días}/30\text{días} = 20,8\text{€}$
- Si contamos también que nos tenemos que dar de alta al ser autónomos. Son 50 euros al mes, siendo una rebaja especial por ser nuevos autónomos. Por lo tanto, por unos 5 meses de trabajo que durará el proyecto, sería un total de 250€.

Por lo que se puede ver, el gasto en electricidad es bastante pequeño, tanto el del ordenador como el de la luz. El gasto que supone el alquiler y la calefacción es más significativo, pero se trata de solo una habitación el alquiler presupuestado, si fuera un local independiente, habría que pagar bastante más.

Por lo tanto, el coste total sería de:

$$10,25 + 10,27 + 3300 + 1,7 + 1,7 + 83,3 + 20,8 + 250 = \mathbf{3678,02\text{€}}$$

3.9. Seguimiento

Este apartado se ha completado después de realizar el proyecto, para mejorar la planificación mediante la comparativa con los datos reales.

La planificación, como se ha visto anteriormente, se ha hecho por fases e iteraciones. Además, se fijaron hitos para tener un control del seguimiento. A continuación se detalla este seguimiento:

- En la primera fase, la de **inicio**, se calculó dos semanas para su realización, y se cumplió el plazo, excepto pequeños cambios que se hicieron después.
- En la fase de **elaboración** se invirtieron 7 semanas, una más de lo planificado, debido a causas personales, no se pudo dedicar el tiempo necesario. Además de que el análisis de la aplicación llevó un coste mayor del esperado. Al margen de eso, se desarrolló sin problemas, elaborando todo el análisis, exceptuando algún cambio durante la fase de construcción.

- La fase de **construcción** empezó con una semana de retraso, pero eso no fue problema para desarrollar toda la aplicación y la API. En la primera iteración se elaboró la API, pues se consideró que sin datos, la aplicación no era muy útil y lo primero era cómo obtenerlos. En la segunda iteración se desarrolló la aplicación con todos sus elementos y la conexión con la base de datos a través de la API. Esta fase en general, ha sido la más costosa de recursos y tiempo, pues se ha tenido que aprender la tecnología con la que se trabajaba, usarla y saber cómo se hacía lo que se quería implementar.
- En la última fase, la de **transición**, se acabó de rematar la memoria y verificar que todo estaba bien. Para poder revisar que estaba todo correcto y en formato se utilizó 30 horas más, por lo que hubo retraso respecto a la fecha final del proyecto.

La redacción de la memoria se llevó a cabo durante todo el proyecto. La cantidad de horas dedicadas ha sido de 3 horas a la semana aproximadamente, pero durante las fases de inicio, elaboración y transición se aumentó el número de horas dedicado a la memoria para añadir todo lo que se iba haciendo en el mismo momento.

3.9.1. Hitos

- Análisis de aplicaciones similares: 9/03/2018
Este hito se cumplió adecuadamente.
- Prototipo de bajo coste: 30/03/2018
El prototipo se retrasó 15horas (5 días), pues no dio tiempo a realizarlo antes por motivos personales.
- Prototipo de baja funcionalidad: 9/05/2018
Este prototipo también se retrasó, debido a la demora anterior y que hubo dificultades técnicas para desarrollarlo.
- Versión final de la aplicación: 8/06/2018
La versión final de la aplicación se completó a tiempo, una vez aprendida la tecnología el desarrollo fue rápido.
- Memoria acabada: 27/06/2018
Este hito se retrasó 30 horas (10 días) debido a correcciones que había que hacer en la memoria. Por lo que la fecha final definitiva fue la del 10/07/2018.

Capítulo 4

Análisis

El objetivo de este capítulo es el de obtener toda la información necesaria para desarrollar la aplicación, siguiendo las fases de la ingeniería del software, ésta corresponde con la fase de análisis.

4.1. Stakeholders

Los *stakeholders* son cualquier persona u organismo interesado en el desarrollo de un proyecto. También son las personas afectadas por él. En este caso tenemos :

Organización	Universidad de Valladolid
Dirección	Plaza Santa Cruz, nº8
Teléfono	983 423 242
Fax	983 423 234
Comentarios	Ninguno

Figura 4.1: UVa

Participante	Margarita Gonzalo Tasis
Organización	Universidad de Valladolid
Rol	Tutor TFG
Es desarrollador	No
Es cliente	Sí
Es usuario	Sí
Comentarios	Ninguno

Figura 4.2: Tutora

Participante	José Luis Sánchez Paredes
Organización	Universidad de Valladolid
Rol	Jefe de proyecto, analista, desarrollador, diseñador, probador
Es desarrollador	Sí
Es cliente	Sí
Es usuario	Sí
Comentarios	Ninguno

Figura 4.3: Alumno

4.2. Requisitos

Aquí se muestran los requisitos obtenidos de la etapa de elicitación de requisitos de la Ingeniería de Software. Divididos en funcionales y no funcionales. En nuestro caso, no se ha necesitado ningún requisito de información.

Alguno de los requisitos funcionales se obtiene del llamado CRUD: Create, Read, Update y Delete, lo cual quiere significar: crear, leer, actualizar y borrar. El usuario tiene que ser capaz de realizar las operaciones anteriores sobre sus datos. Este término se suele aplicar a las acciones básicas que se pueden ejecutar en la base de datos.

4.2.1. Requisitos funcionales

Son servicios que el sistema tiene que cumplir.

FRQ-0001	Ver aves disponibles
Versión	1.0 (13/02/2018)
Autores	• José Luis Sánchez Paredes
Fuentes	• Margarita Gonzalo Tasis
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir que el usuario vea todas las aves contenidas en la aplicación</i>
Importancia	importante
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	media
Comentarios	Ninguno

Figura 4.4: Ver aves disponibles

FRQ-0002	Consultar aves por criterios
Versión	1.0 (19/02/2018)
Autores	• José Luis Sánchez Paredes
Fuentes	• Margarita Gonzalo Tasis
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir al usuario visualizar las aves por nombre común, nombre científico, región o época del año.</i>
Importancia	importante
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	baja
Comentarios	Ninguno

Figura 4.5: Consultar aves por criterios

FRQ-0003	Añadir avistamiento
Versión	1.0 (19/02/2018)
Autores	• José Luis Sánchez Paredes
Fuentes	• Margarita Gonzalo Tasis
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir al usuario agregar un nuevo avistamiento hecho por él</i>
Importancia	importante
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	baja
Comentarios	Ninguno

Figura 4.6: Añadir avistamiento

4.2. REQUISITOS

FRQ-0004	Descripción de cada ave
Versión	1.0 (19/02/2018)
Autores	<ul style="list-style-type: none">• José Luis Sánchez Paredes
Fuentes	<ul style="list-style-type: none">• Margarita Gonzalo Tasis
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir al usuario consultar una descripción de cada ave</i>
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	baja
Comentarios	Ninguno

Figura 4.7: Consultar descripción de un ave

FRQ-0005	Retos
Versión	1.0 (19/02/2018)
Autores	<ul style="list-style-type: none">• José Luis Sánchez Paredes
Fuentes	<ul style="list-style-type: none">• Margarita Gonzalo Tasis
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir al usuario consultar los retos disponibles</i>
Importancia	importante
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	baja
Comentarios	Ninguno

Figura 4.8: Consultar retos

FRQ-0006	Ver mis avistamientos
Versión	1.0 (19/02/2018)
Autores	<ul style="list-style-type: none">• José Luis Sánchez Paredes
Fuentes	<ul style="list-style-type: none">• Margarita Gonzalo Tasis
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir al usuario ver sus avistamientos creados anteriormente</i>
Importancia	importante
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	baja
Comentarios	Ninguno

Figura 4.9: Consultar mis avistamientos

FRQ-0007	Logros
Versión	1.0 (19/02/2018)
Autores	• José Luis Sánchez Paredes
Fuentes	• Margarita Gonzalo Tasis
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir al usuario recibir logros segun los retos que supere</i>
Importancia	importante
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	baja
Comentarios	Ninguno

Figura 4.10: Consultar logros

FRQ-0008	Avistamientos de otros usuarios
Versión	1.0 (19/02/2018)
Autores	• José Luis Sánchez Paredes
Fuentes	• Margarita Gonzalo Tasis
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir al usuario ver los avistamientos de otros usuarios</i>
Importancia	importante
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	baja
Comentarios	Ninguno

Figura 4.11: Consultar avistamientos de todos los usuarios

FRQ-0009	Modificar avistamiento
Versión	1.0 (21/02/2018)
Autores	• José Luis Sánchez Paredes
Fuentes	• Margarita Gonzalo Tasis
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir al usuario modificar un avistamiento propio</i>
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	baja
Comentarios	Ninguno

Figura 4.12: Modificar un avistamiento propio

4.2. REQUISITOS

FRQ-0010	Eliminar un avistamiento propio
Versión	1.0 (21/02/2018)
Autores	<ul style="list-style-type: none">• José Luis Sánchez Paredes
Fuentes	<ul style="list-style-type: none">• Margarita Gonzalo Tasis
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir al usuario eliminar un avistamiento propio</i>
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	baja
Comentarios	Ninguno

Figura 4.13: Eliminar un avistamiento propio

FRQ-0011	Registrarse
Versión	1.0 (23/02/2018)
Autores	<ul style="list-style-type: none">• José Luis Sánchez Paredes
Fuentes	<ul style="list-style-type: none">• Margarita Gonzalo Tasis
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir al usuario registrarse</i>
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	baja
Comentarios	Ninguno

Figura 4.14: Registrarse

FRQ-0012	Acceder a la aplicación
Versión	1.0 (23/02/2018)
Autores	<ul style="list-style-type: none">• José Luis Sánchez Paredes
Fuentes	<ul style="list-style-type: none">• Margarita Gonzalo Tasis
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir al usuario acceder a la aplicación con su usuario y contraseña</i>
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	baja
Comentarios	Ninguno

Figura 4.15: Acceder a la aplicación

FRQ-0013	Actualizar datos del usuario
Versión	1.0 (23/02/2018)
Autores	• José Luis Sánchez Paredes
Fuentes	• Margarita Gonzalo Tasis
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir al usuario modificar sus datos personales</i>
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	media
Comentarios	Ninguno

Figura 4.16: Actualizar datos del usuario

FRQ-0014	Borrar mi usuario
Versión	1.0 (23/02/2018)
Autores	• José Luis Sánchez Paredes
Fuentes	• Margarita Gonzalo Tasis
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir al usuario borrar su cuenta</i>
Importancia	importante
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	media
Comentarios	Ninguno

Figura 4.17: Borrar mi usuario

4.2.2. Requisitos no funcionales

Son restricciones que afectan a los servicios o funciones del sistema, tales como restricción de tiempo, sobre el proceso de desarrollo, estándares...

4.2. REQUISITOS

NFR-0001	Tiempo de arranque
Versión	1.0 (19/02/2018)
Autores	<ul style="list-style-type: none">• José Luis Sánchez Paredes
Fuentes	<ul style="list-style-type: none">• Margarita Gonzalo Tasis
Dependencias	Ninguno
Descripción	El sistema deberá <i>arrancar en menos de 15 segundos</i>
Importancia	importante
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	baja
Comentarios	Ninguno

Figura 4.18: Tiempo de arranque

NFR-0002	Tiempo de refresco
Versión	1.0 (19/02/2018)
Autores	<ul style="list-style-type: none">• José Luis Sánchez Paredes
Fuentes	<ul style="list-style-type: none">• Margarita Gonzalo Tasis
Dependencias	Ninguno
Descripción	El sistema deberá <i>refrescarse entre pantallas con un tiempo menor de 10 segundos</i>
Importancia	importante
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	baja
Comentarios	Ninguno

Figura 4.19: Tiempo de refresco

NFR-0003	Versión Android
Versión	1.0 (19/02/2018)
Autores	<ul style="list-style-type: none">• José Luis Sánchez Paredes
Fuentes	<ul style="list-style-type: none">• Margarita Gonzalo Tasis
Dependencias	Ninguno
Descripción	El sistema deberá <i>poder ejecutarse en sistemas Android con mínimo de 4.4(KitKat) API 19.0</i>
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	baja
Comentarios	Ninguno

Figura 4.20: Versión de Android

4.3. Casos de uso

Un caso de uso es un conjunto de escenarios (con éxito o no) relacionados por el resultado que el actor espera obtener. Están muy relacionados con los requisitos, pues son la implementación de éstos.

4.3.1. Diagrama de casos de uso

En la figura 4.21 se ve el diagrama de casos de uso que ha resultado de los requisitos obtenidos.

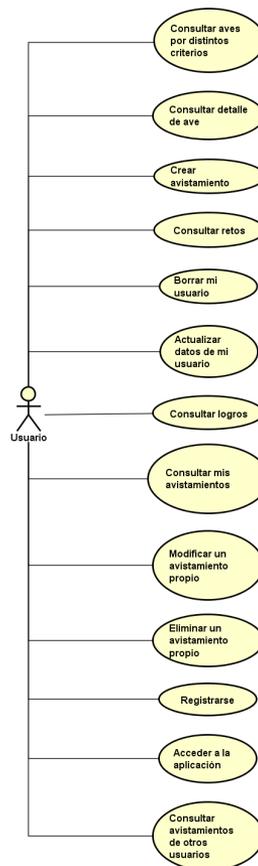


Figura 4.21: Diagrama de casos de uso

4.3.3. Especificación de los casos de uso

UC-0001	Consultar las aves por distintos criterios	
Versión	1.0 (19/02/2018)	
Autores	<ul style="list-style-type: none"> • José Luis Sánchez Paredes 	
Fuentes	<ul style="list-style-type: none"> • Margarita Gonzalo Tasis 	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>pulse el boton de ordenar</i>	
Precondición	Ha accedido con su usuario y contraseña	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) pulsa el boton ordenar y elige el criterio por el que ordenar
	2	El sistema ordena según el criterio elegido y muestra los datos al usuario
Postcondición	Se muestran las aves ordenadas	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	PD	
Importancia	vital	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	baja	
Comentarios	Ninguno	

Figura 4.23: Consultar aves por distintos criterios

UC-0002	Consultar la descripción de un ave	
Versión	1.0 (19/02/2018)	
Autores	<ul style="list-style-type: none"> • José Luis Sánchez Paredes 	
Fuentes	<ul style="list-style-type: none"> • Margarita Gonzalo Tasis 	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>pulse el nombre de un ave</i>	
Precondición	Aparece la lista de todas las aves	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) pulsa el nombre de un ave de la lista
	2	El sistema carga los datos del ave seleccionada y se los muestra al usuario
Postcondición	Aparece la descripción del ave elegida	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	PD	
Importancia	vital	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	baja	
Comentarios	Ninguno	

Figura 4.24: Consultar la descripción de un ave

4.3. CASOS DE USO

UC-0003	Crear avistamiento	
Versión	1.0 (19/02/2018)	
Autores	• José Luis Sánchez Paredes	
Fuentes	• Margarita Gonzalo Tasis	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>pulsa la opción de crear un avistamiento</i>	
Precondición	Ha accedido con su usuario y contraseña	
Secuencia normal	Paso Acción	
	1	El actor Usuario (ACT-0001) selecciona la opción de crear un nuevo avistamiento
	2	El sistema pide al usuario mediante unos campos que rellene los datos relacionados con el nuevo avistamiento
	3	El actor Usuario (ACT-0001) introduce los datos pedidos y acepta la operación
	4	El sistema guarda los datos y los envía a la base de datos. Avisa al usuario de que la operación se ha llevado a cabo con éxito
Postcondición	Se ha añadido un nuevo avistamiento	
Excepciones	Paso Acción	
	3	Si introduce mal los datos, el actor Usuario (ACT-0001) tiene que repetir la acción de rellenar los datos, a continuación este caso de uso continúa
Rendimiento	Paso Tiempo máximo	
	-	-
Frecuencia esperada	PD	
Importancia	vital	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	baja	
Comentarios	Ninguno	

Figura 4.25: Crear avistamiento

UC-0004	Consultar los retos	
Versión	1.0 (19/02/2018)	
Autores	• José Luis Sánchez Paredes	
Fuentes	• Margarita Gonzalo Tasis	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>pulsa el menú y selecciona el botón retos</i>	
Precondición	Ha accedido con su usuario y contraseña	
Secuencia normal	Paso Acción	
	1	El actor Usuario (ACT-0001) pulsa el menú
	2	El sistema muestra las opciones del menú
	3	El actor Usuario (ACT-0001) selecciona la opción de retos
	4	El sistema muestra los retos disponibles
Postcondición	Se muestran los retos	
Excepciones	Paso Acción	
	3	Si pulsa otra opción, el actor Usuario (ACT-0001) irá a otro caso de uso que no es éste, a continuación este caso de uso queda sin efecto
Rendimiento	Paso Tiempo máximo	
	-	-
Frecuencia esperada	PD	
Importancia	vital	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	baja	
Comentarios	Ninguno	

Figura 4.26: Consultar los retos

UC-0005	Borrar mi usuario	
Versión	1.0 (19/02/2018)	
Autores	<ul style="list-style-type: none"> • José Luis Sánchez Paredes 	
Fuentes	<ul style="list-style-type: none"> • Margarita Gonzalo Tasis 	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>pulsa boton borrar cuenta</i>	
Precondición	Se visualiza la pantalla de ajustes	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) <i>selecciona borrar cuenta</i>
	2	El sistema <i>borra el usuario y le avisa de que ya ha borrado la cuenta</i>
Postcondición	El usuario esta borrado	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	PD	
Importancia	vital	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	baja	
Comentarios	Ninguno	

Figura 4.27: Borrar mi usuario

UC-0006	Actualizar datos de mi usuario	
Versión	1.0 (19/02/2018)	
Autores	<ul style="list-style-type: none"> • José Luis Sánchez Paredes 	
Fuentes	<ul style="list-style-type: none"> • Margarita Gonzalo Tasis 	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>pulsa el botón de actualizar</i>	
Precondición	se encuentra en la pantalla de ajustes	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) <i>pulsa el botón actualizar</i>
	2	El sistema <i>actualiza en la base de datos los nuevos datos y avisa al usuario de que se ha actualizado</i>
Postcondición	el usuario tiene los nuevos datos	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	PD	
Importancia	vital	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	baja	
Comentarios	Ninguno	

Figura 4.28: Actualizar datos de mi usuario

4.3. CASOS DE USO

UC-0007	Consultar logros	
Versión	1.0 (19/02/2018)	
Autores	<ul style="list-style-type: none"> • José Luis Sánchez Paredes 	
Fuentes	<ul style="list-style-type: none"> • Margarita Gonzalo Tasis 	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>pulsa el menú y elige la opción logros</i>	
Precondición	Ha accedido con su usuario y contraseña	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) <i>pulsa el menú</i>
	2	El sistema <i>muestra las opciones del menú</i>
	3	El actor Usuario (ACT-0001) <i>selecciona la opción de logros</i>
	4	El sistema <i>muestra los logros alcanzados</i>
Postcondición	Ve los logros	
Excepciones	Paso	Acción
	3	Si <i>pulsa otra opción</i> , el actor Usuario (ACT-0001) <i>irá a otro caso de uso que no es éste</i> , a continuación este caso de uso <i>queda sin efecto</i>
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	PD	
Importancia	vital	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	baja	
Comentarios	Ninguno	

Figura 4.29: Consultar logros

UC-0008	Consultar mis avistamientos	
Versión	1.0 (19/02/2018)	
Autores	<ul style="list-style-type: none"> • José Luis Sánchez Paredes 	
Fuentes	<ul style="list-style-type: none"> • Margarita Gonzalo Tasis 	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>pulsa el menú y selecciona el botón mis avistamientos</i>	
Precondición	Ha accedido con su usuario y contraseña	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) <i>pulsa el menú</i>
	2	El sistema <i>muestra las opciones del menú</i>
	3	El actor Usuario (ACT-0001) <i>selecciona la opción de mis avistamientos</i>
	4	El sistema <i>muestra todos los avistamientos de todos los usuarios</i>
Postcondición	Ve todos sus avistamientos	
Excepciones	Paso	Acción
	3	Si <i>pulsa otra opción</i> , el actor Usuario (ACT-0001) <i>irá a otro caso de uso que no es éste</i> , a continuación este caso de uso <i>queda sin efecto</i>
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	PD	
Importancia	vital	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	baja	
Comentarios	Ninguno	

Figura 4.30: Consultar mis avistamientos

UC-0009	Modificar avistamiento propio	
Versión	1.0 (21/02/2018)	
Autores	<ul style="list-style-type: none"> • José Luis Sánchez Paredes 	
Fuentes	<ul style="list-style-type: none"> • Margarita Gonzalo Tasis 	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>pulsa un avistamiento propio</i>	
Precondición	Está dentro de la opción de mis avistamientos	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) selecciona el avistamiento que quiere modificar
	2	El sistema muestra los detalles de ese avistamiento
	3	El actor Usuario (ACT-0001) modifica los campos que quiere y los guarda
	4	El sistema guarda los cambios y vuelve a la vista anterior
Postcondición	El sistema muestra los avistamientos propios con el cambio	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	PD	
Importancia	vital	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	baja	
Comentarios	Ninguno	

Figura 4.31: Modificar avistamiento propio

UC-0010	Eliminar un avistamiento propio	
Versión	1.0 (21/02/2018)	
Autores	<ul style="list-style-type: none"> • José Luis Sánchez Paredes 	
Fuentes	<ul style="list-style-type: none"> • Margarita Gonzalo Tasis 	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>pulsa un avistamiento propio</i>	
Precondición	Está dentro de la opción de mis avistamientos	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) selecciona el avistamiento que quiere eliminar
	2	El sistema muestra los datos de ese avistamiento
	3	El actor Usuario (ACT-0001) pulsa la opción de eliminar avistamiento
	4	El sistema elimina el avistamiento y vuelve a la vista anterior
Postcondición	El sistema borra el avistamiento y muestra el resto	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	PD	
Importancia	vital	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	baja	
Comentarios	Ninguno	

Figura 4.32: Eliminar avistamiento propio

4.3. CASOS DE USO

UC-0011	Registrarse										
Versión	1.0 (23/02/2018)										
Autores	• José Luis Sánchez Paredes										
Fuentes	• Margarita Gonzalo Tasis										
Dependencias	Ninguno										
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>pulsa el boton de registrarse</i>										
Precondición	ninguna										
Secuencia normal	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El actor Usuario (ACT-0001) <i>pulsa el botón de registrarse</i></td> </tr> <tr> <td>2</td> <td>El sistema <i>muestra la vista de registro y pide los datos necesarios</i></td> </tr> <tr> <td>3</td> <td>El actor Usuario (ACT-0001) <i>rellena los datos</i></td> </tr> <tr> <td>4</td> <td>El sistema <i>guarda los datos y vuelve a la vista de acceder a la aplicación</i></td> </tr> </tbody> </table>	Paso	Acción	1	El actor Usuario (ACT-0001) <i>pulsa el botón de registrarse</i>	2	El sistema <i>muestra la vista de registro y pide los datos necesarios</i>	3	El actor Usuario (ACT-0001) <i>rellena los datos</i>	4	El sistema <i>guarda los datos y vuelve a la vista de acceder a la aplicación</i>
Paso	Acción										
1	El actor Usuario (ACT-0001) <i>pulsa el botón de registrarse</i>										
2	El sistema <i>muestra la vista de registro y pide los datos necesarios</i>										
3	El actor Usuario (ACT-0001) <i>rellena los datos</i>										
4	El sistema <i>guarda los datos y vuelve a la vista de acceder a la aplicación</i>										
Postcondición	el usuario esta registrado										
Excepciones	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>3</td> <td><i>Si no rellena todos los campos , el sistema vuelve a pedir los datos, a continuación este caso de uso continúa</i></td> </tr> </tbody> </table>	Paso	Acción	3	<i>Si no rellena todos los campos , el sistema vuelve a pedir los datos, a continuación este caso de uso continúa</i>						
Paso	Acción										
3	<i>Si no rellena todos los campos , el sistema vuelve a pedir los datos, a continuación este caso de uso continúa</i>										
Rendimiento	<table border="1"> <thead> <tr> <th>Paso</th> <th>Tiempo máximo</th> </tr> </thead> <tbody> <tr> <td>-</td> <td>-</td> </tr> </tbody> </table>	Paso	Tiempo máximo	-	-						
Paso	Tiempo máximo										
-	-										
Frecuencia esperada	PD										
Importancia	vital										
Urgencia	inmediatamente										
Estado	en construcción										
Estabilidad	baja										
Comentarios	Ninguno										

Figura 4.33: Registrarse

UC-0012	Acceder a la aplicación										
Versión	1.0 (23/02/2018)										
Autores	• José Luis Sánchez Paredes										
Fuentes	• Margarita Gonzalo Tasis										
Dependencias	Ninguno										
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>inicia la aplicación</i>										
Precondición	ninguna										
Secuencia normal	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El actor Usuario (ACT-0001) <i>inicia la aplicación</i></td> </tr> <tr> <td>2</td> <td>El sistema <i>muestra los campos a rellenar para acceder a la aplicación</i></td> </tr> <tr> <td>3</td> <td>El actor Usuario (ACT-0001) <i>rellena los campos y pulsa entrar</i></td> </tr> <tr> <td>4</td> <td>El sistema <i>verinfica los datos y le muestra la lista de las aves</i></td> </tr> </tbody> </table>	Paso	Acción	1	El actor Usuario (ACT-0001) <i>inicia la aplicación</i>	2	El sistema <i>muestra los campos a rellenar para acceder a la aplicación</i>	3	El actor Usuario (ACT-0001) <i>rellena los campos y pulsa entrar</i>	4	El sistema <i>verinfica los datos y le muestra la lista de las aves</i>
Paso	Acción										
1	El actor Usuario (ACT-0001) <i>inicia la aplicación</i>										
2	El sistema <i>muestra los campos a rellenar para acceder a la aplicación</i>										
3	El actor Usuario (ACT-0001) <i>rellena los campos y pulsa entrar</i>										
4	El sistema <i>verinfica los datos y le muestra la lista de las aves</i>										
Postcondición	El usuario está dentro de la aplicación										
Excepciones	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>3</td> <td><i>Si introduce mal los datos o no está registrado, el sistema vuelve a pedir los datos correctos, a continuación este caso de uso continúa</i></td> </tr> </tbody> </table>	Paso	Acción	3	<i>Si introduce mal los datos o no está registrado, el sistema vuelve a pedir los datos correctos, a continuación este caso de uso continúa</i>						
Paso	Acción										
3	<i>Si introduce mal los datos o no está registrado, el sistema vuelve a pedir los datos correctos, a continuación este caso de uso continúa</i>										
Rendimiento	<table border="1"> <thead> <tr> <th>Paso</th> <th>Tiempo máximo</th> </tr> </thead> <tbody> <tr> <td>-</td> <td>-</td> </tr> </tbody> </table>	Paso	Tiempo máximo	-	-						
Paso	Tiempo máximo										
-	-										
Frecuencia esperada	PD										
Importancia	vital										
Urgencia	inmediatamente										
Estado	en construcción										
Estabilidad	baja										
Comentarios	Ninguno										

Figura 4.34: Acceder a la aplicación

UC-0013	Consultar todos los avistamientos sean suyos o no	
Versión	1.0 (23/02/2018)	
Autores	<ul style="list-style-type: none"> • José Luis Sánchez Paredes 	
Fuentes	<ul style="list-style-type: none"> • Margarita Gonzalo Tasis 	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>pulsa el menú y pulsa el botón de todos los avistamientos</i>	
Precondición	Ha accedido con su usuario y contraseña	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) pulsa el menú
	2	El sistema muestra las opciones del menú
	3	El actor Usuario (ACT-0001) selecciona la opción de todos los avistamientos
	4	El sistema muestra todos los avistamientos de todos los usuarios
Postcondición	Ve todos los avistamientos de otros usuarios y suyos	
Excepciones	Paso	Acción
	3	Si <i>pulsa otra opción</i> , el actor Usuario (ACT-0001) irá a otro caso de uso que no es éste, a continuación este caso de uso queda sin efecto
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	PD	
Importancia	vital	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	baja	
Comentarios	Ninguno	

Figura 4.35: Consultar todos los avistamientos

4.4. Modelo de Dominio

En la figura 4.36 de esta sección se muestra el diagrama de clases que representa el modelo de dominio con todas las clases afectadas en este proyecto, así como sus atributos. Las operaciones no se muestran para mayor claridad; en el apartado de diseño se tratará con más profundidad.

Las clases del modelo de dominio son:

- **Area:** Clase que recoge todos los posibles valores del área geográfica en las que puede ser añadido un avistamiento, o un usuario. En este proyecto, como se centra en España, en especial en Castilla y León. Así pues, contendrá básicamente las posibles provincias en las que se pueden hacer avistamientos.
- **Ecosystem:** Clase de tipo *enum* que recoge todos los posibles valores de los ecosistemas en los que pueden verse las aves.
- **Season:** Clase que recoge todos los posibles valores de las estaciones o temporadas en las que pueden verse determinadas aves. Tiene mucha importancia sobretudo en las aves migratorias.

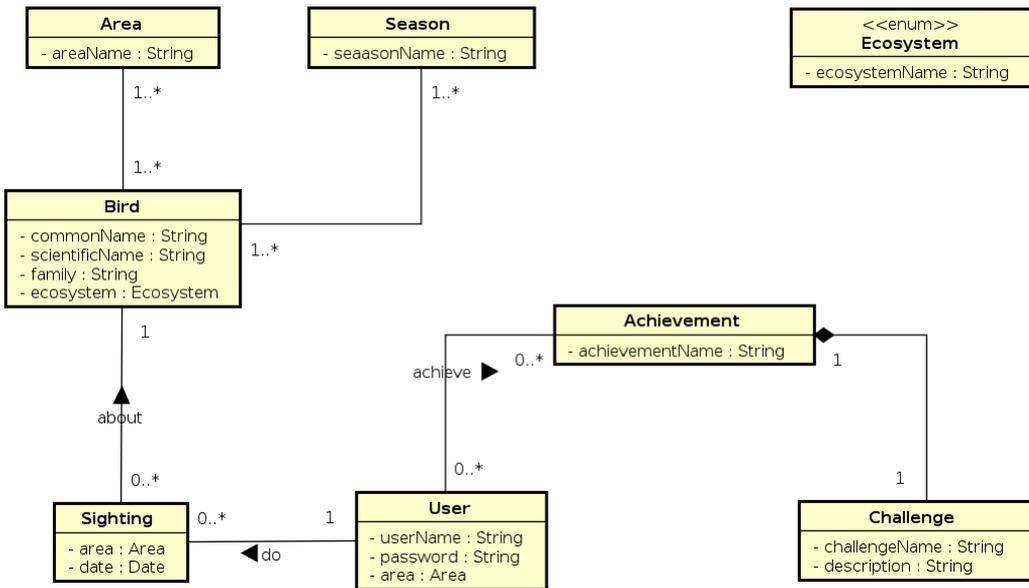


Figura 4.36: Modelo de dominio

- **Bird:** Clase que rige a los objetos de tipo *Bird* o pájaro. Se compone de los atributos: nombre común, nombre científico, familia y ecosistema.
- **User:** Clase que rige a los objetos de tipo *User* o usuario. Se compone de los atributos: nombre del usuario, contraseña y área.
- **Sighting:** Clase que rige a los objetos de tipo *Sighting* o avistamiento. Se compone de los atributos: nombre común del ave vista, área donde se ha visto el ave y la fecha cuando se ha visto el ave.
- **Challenge:** Clase que rige a los objetos de tipo *Challenge* o reto. Se compone de los atributos: nombre del reto y descripción del reto.
- **Achievement:** Clase que rige a los objetos de tipo *Achievement* o logro. Se compone de los atributos: nombre del logro.

4.5. Diagramas de actividad

A continuación se muestran los diagramas de actividad de cada caso de uso, en los que se puede ver el flujo dentro de los mismos.

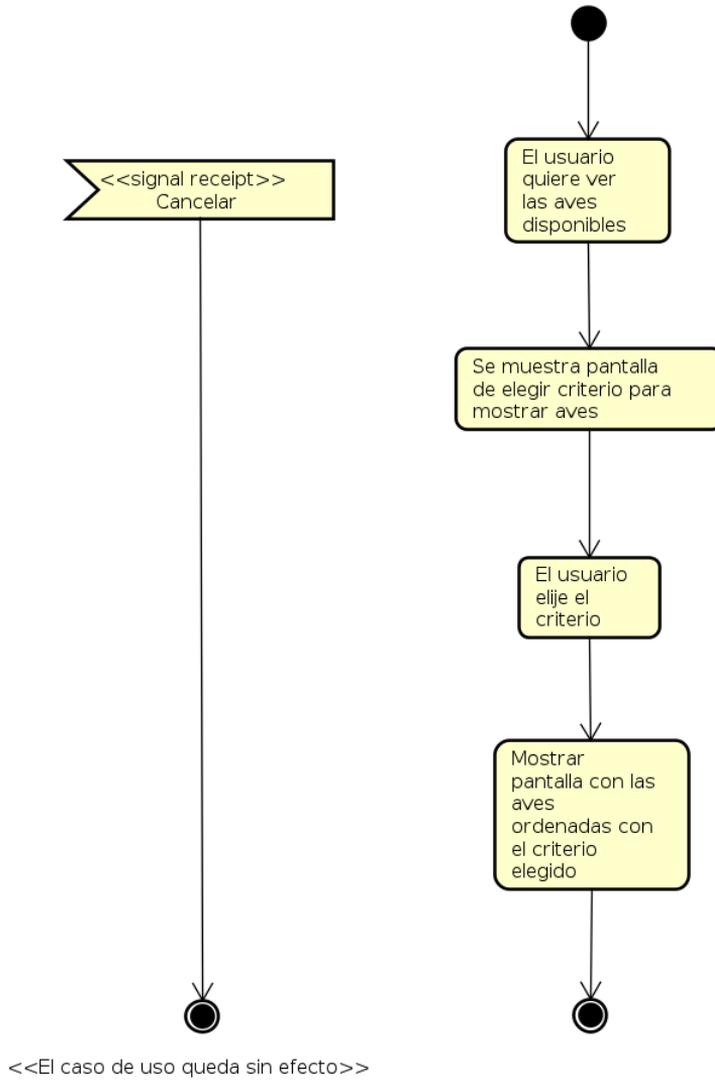


Figura 4.37: Diagrama de actividad del caso de uso CU01

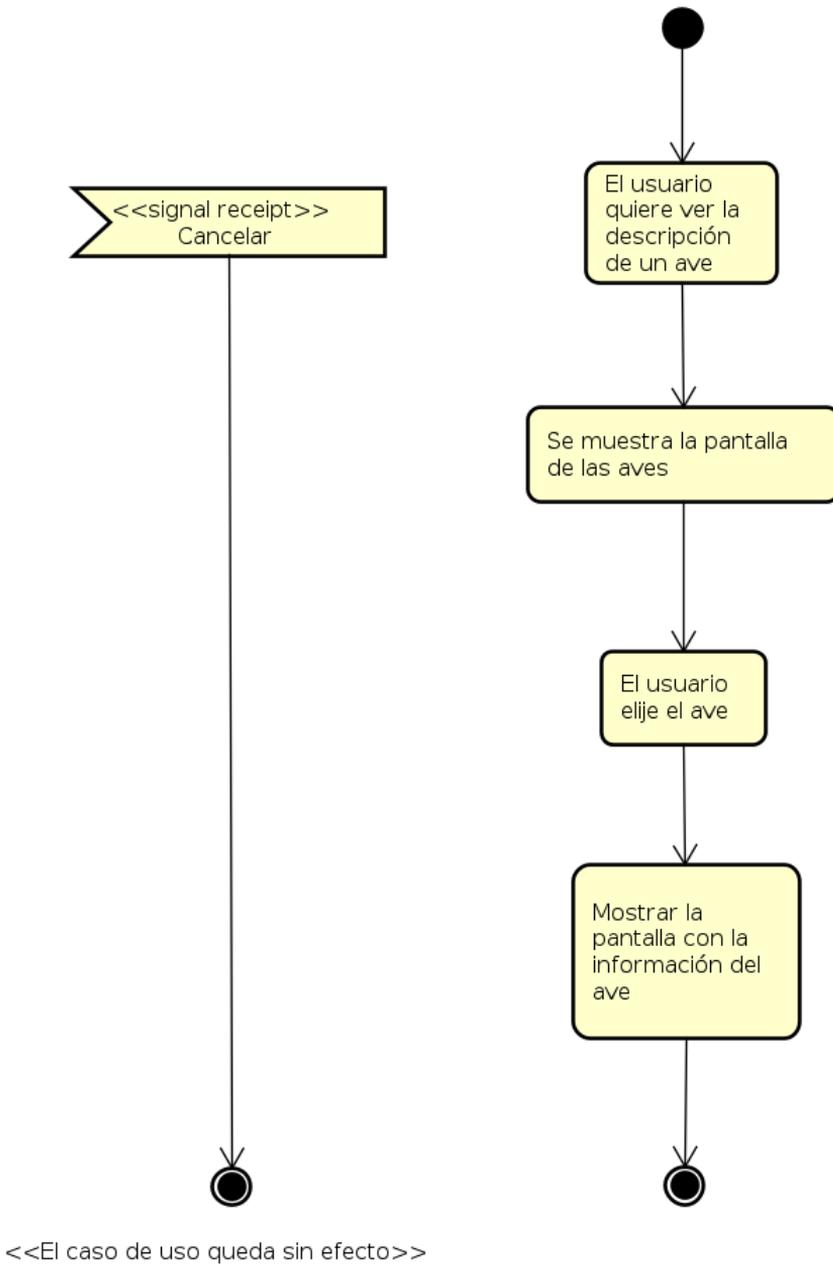
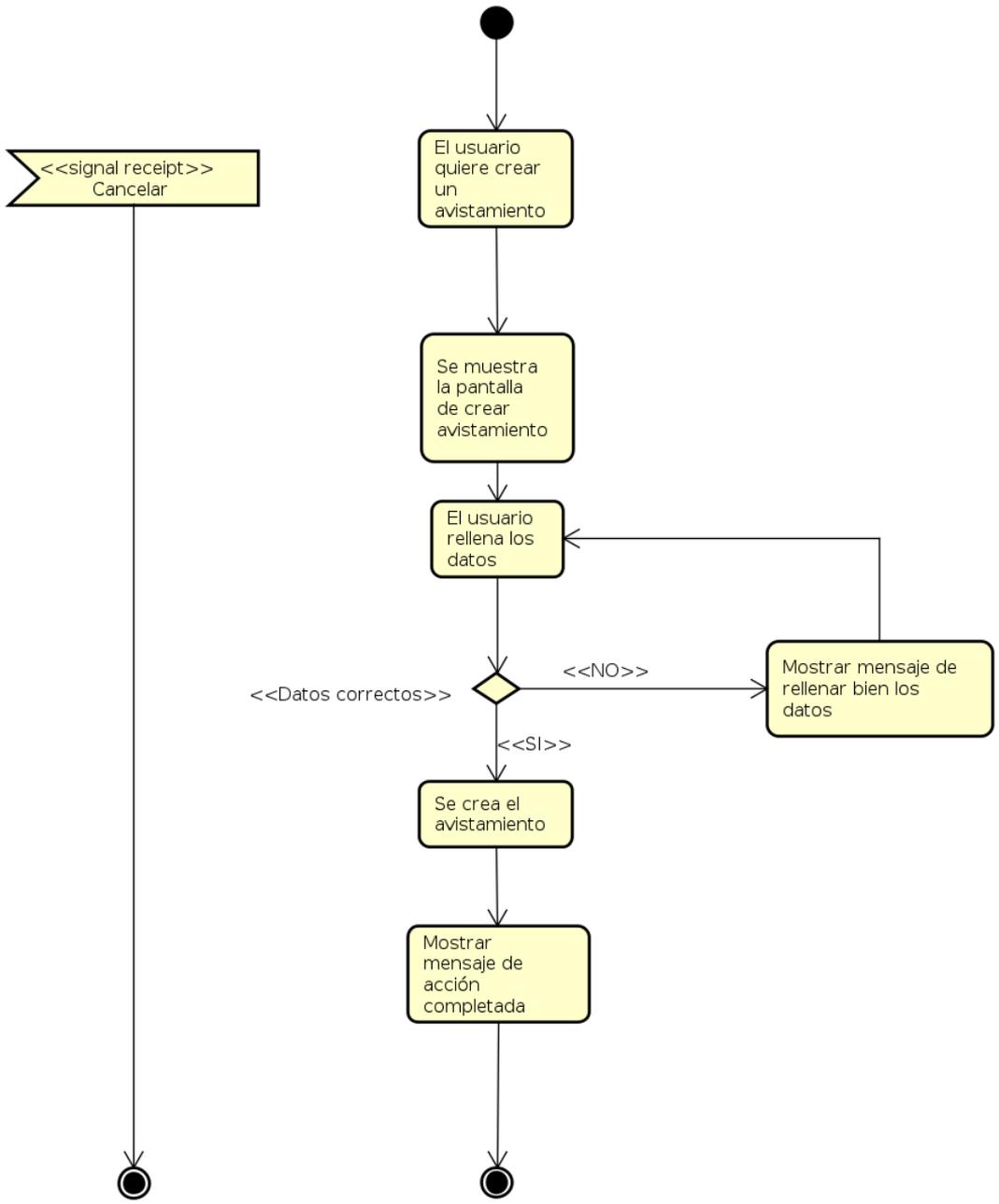


Figura 4.38: Diagrama de actividad del caso de uso CU02



<<El caso de uso queda sin efecto>>

Figura 4.39: Diagrama de actividad del caso de uso CU03

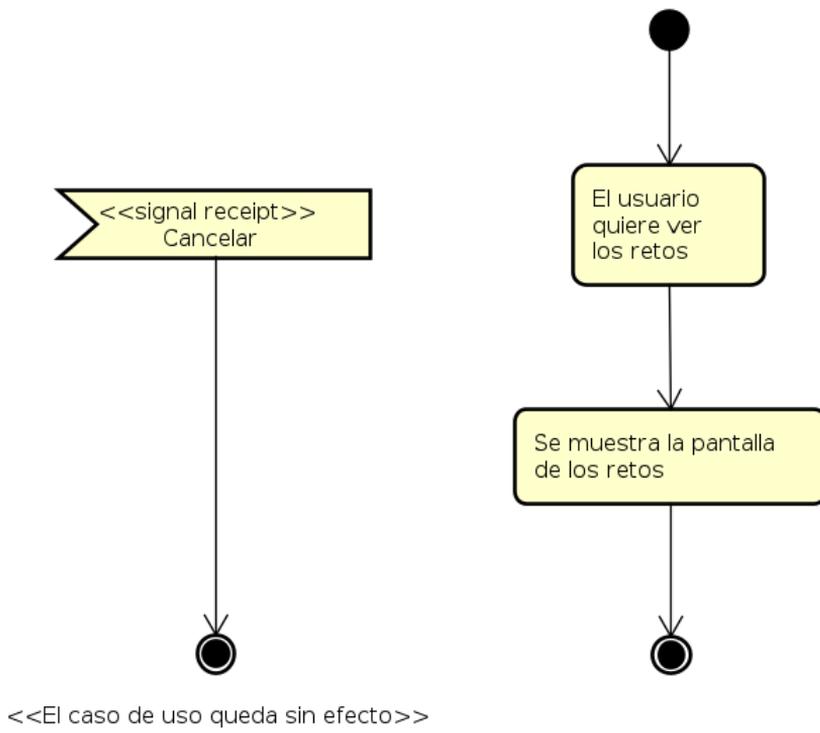


Figura 4.40: Diagrama de actividad del caso de uso CU04

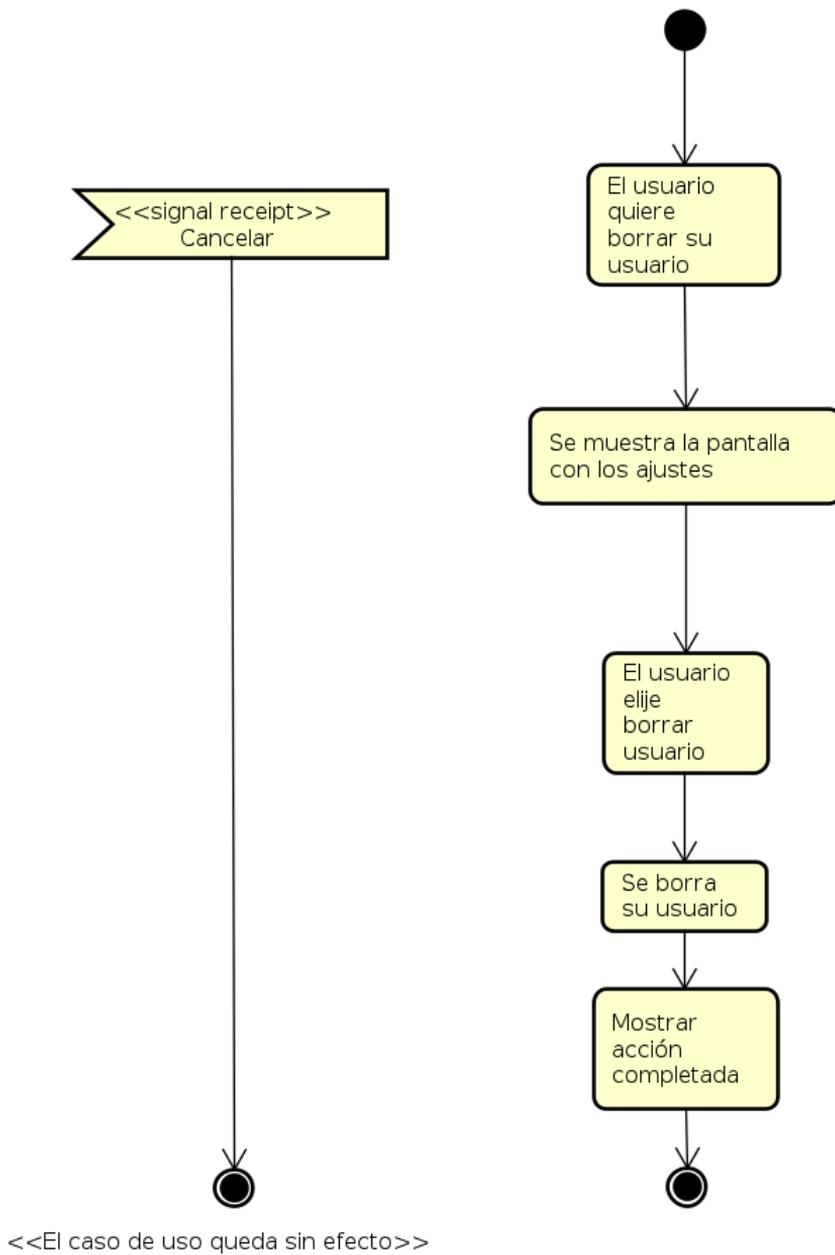
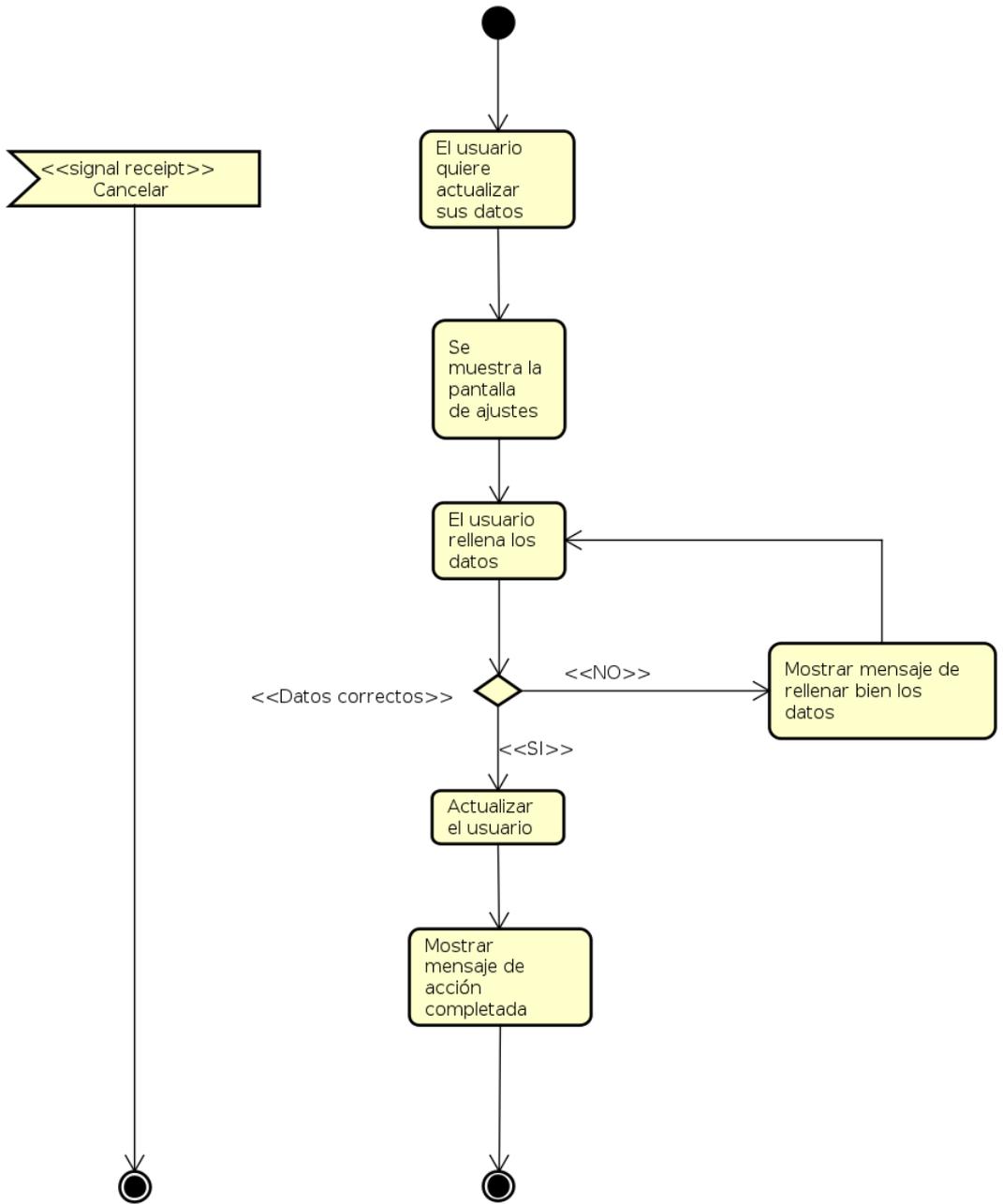


Figura 4.41: Diagrama de actividad del caso de uso CU05



<<El caso de uso queda sin efecto>>

Figura 4.42: Diagrama de actividad del caso de uso CU06

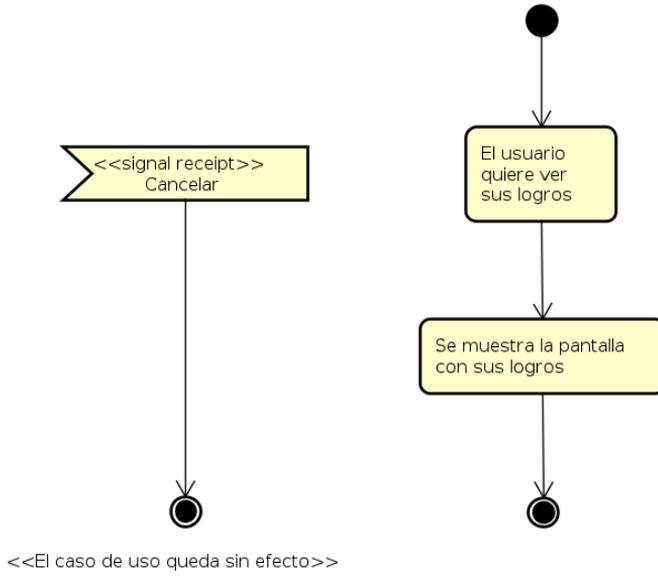


Figura 4.43: Diagrama de actividad del caso de uso CU07

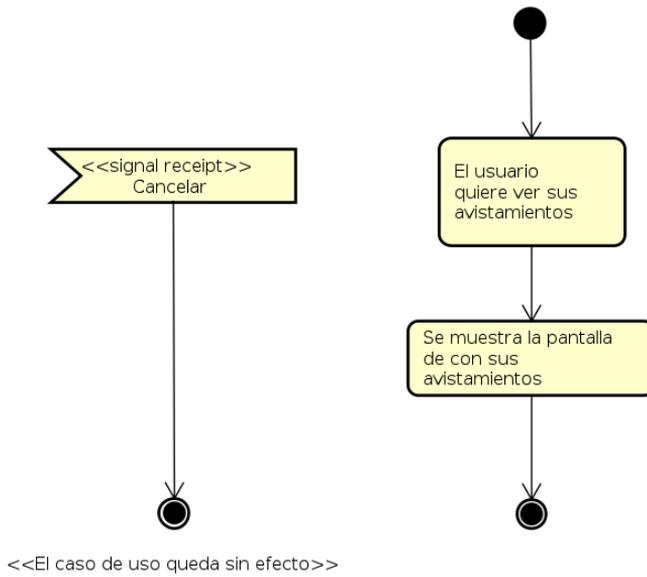


Figura 4.44: Diagrama de actividad del caso de uso CU08

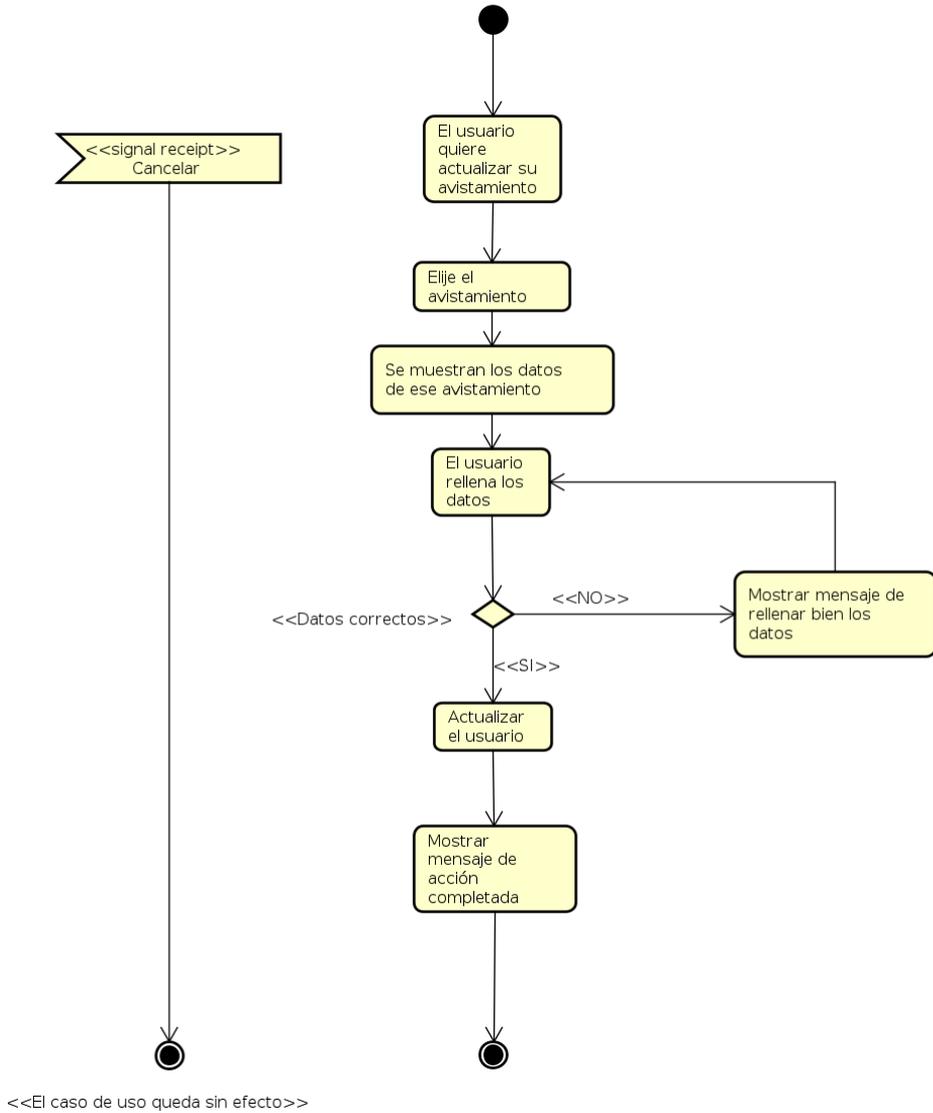


Figura 4.45: Diagrama de actividad del caso de uso CU09

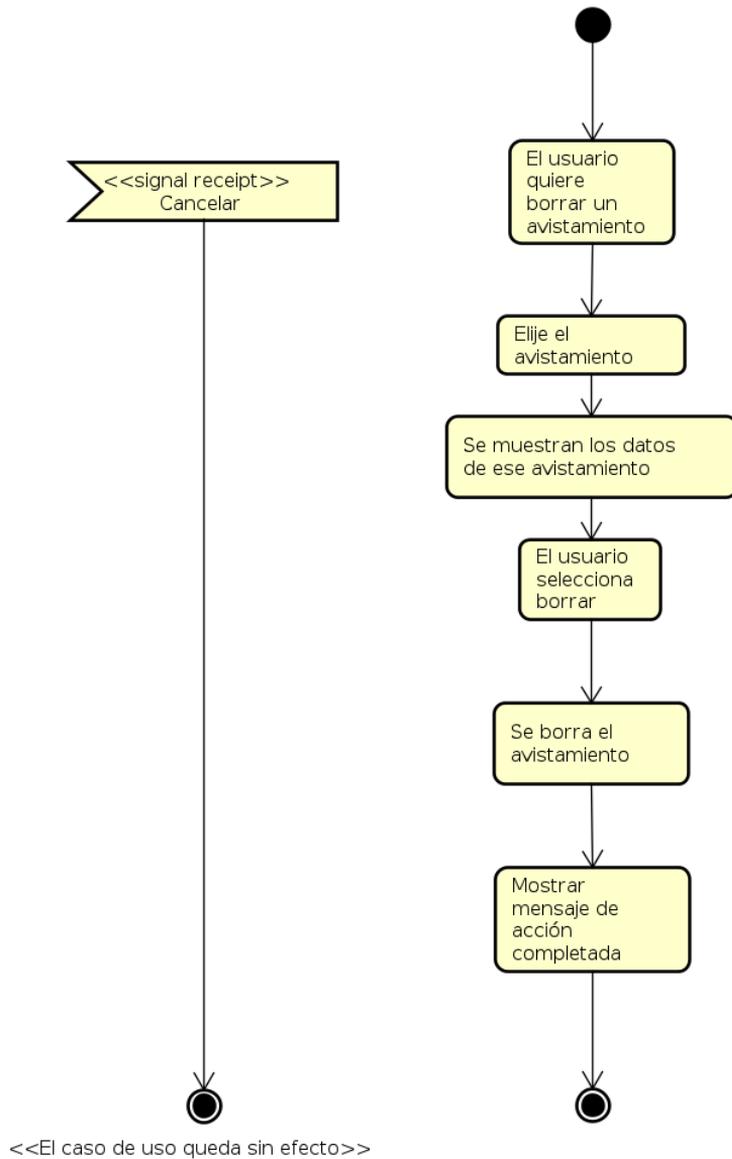


Figura 4.46: Diagrama de actividad del caso de uso CU10

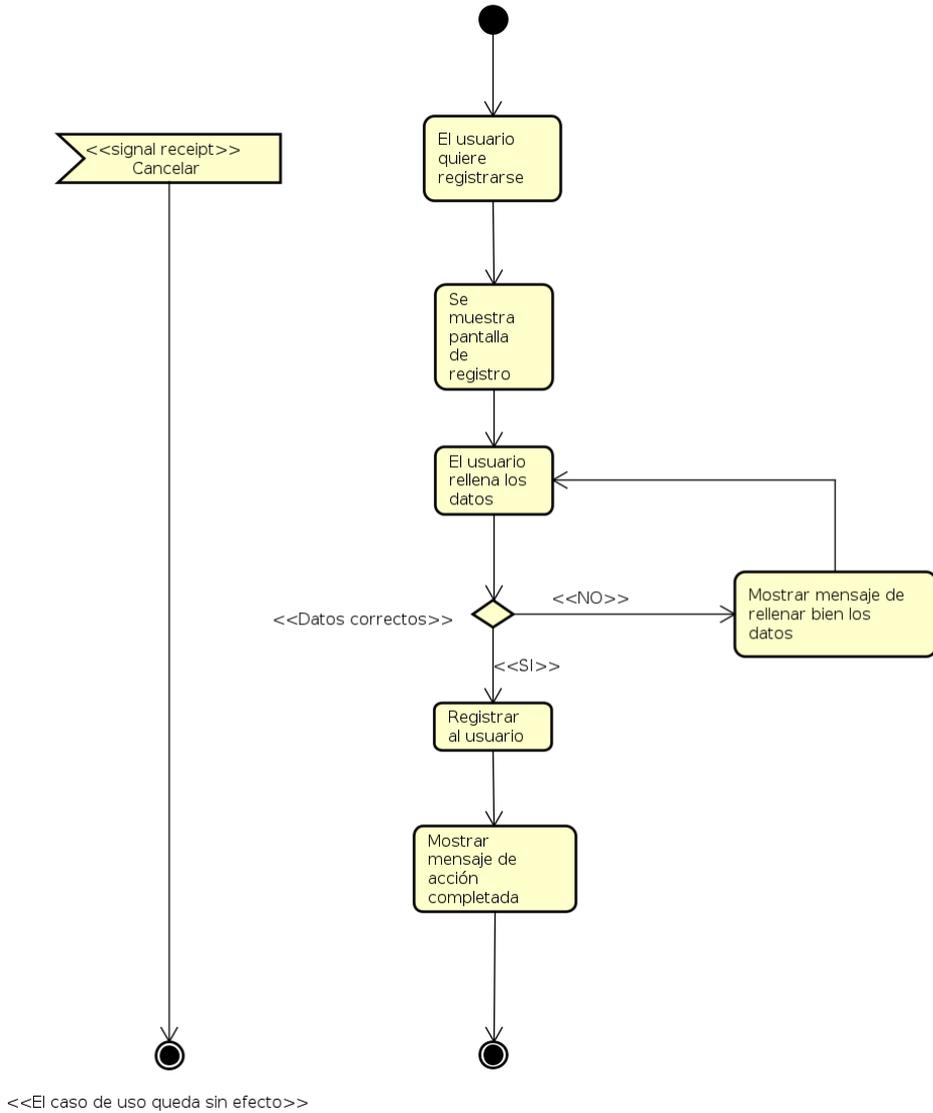


Figura 4.47: Diagrama de actividad del caso de uso CU11

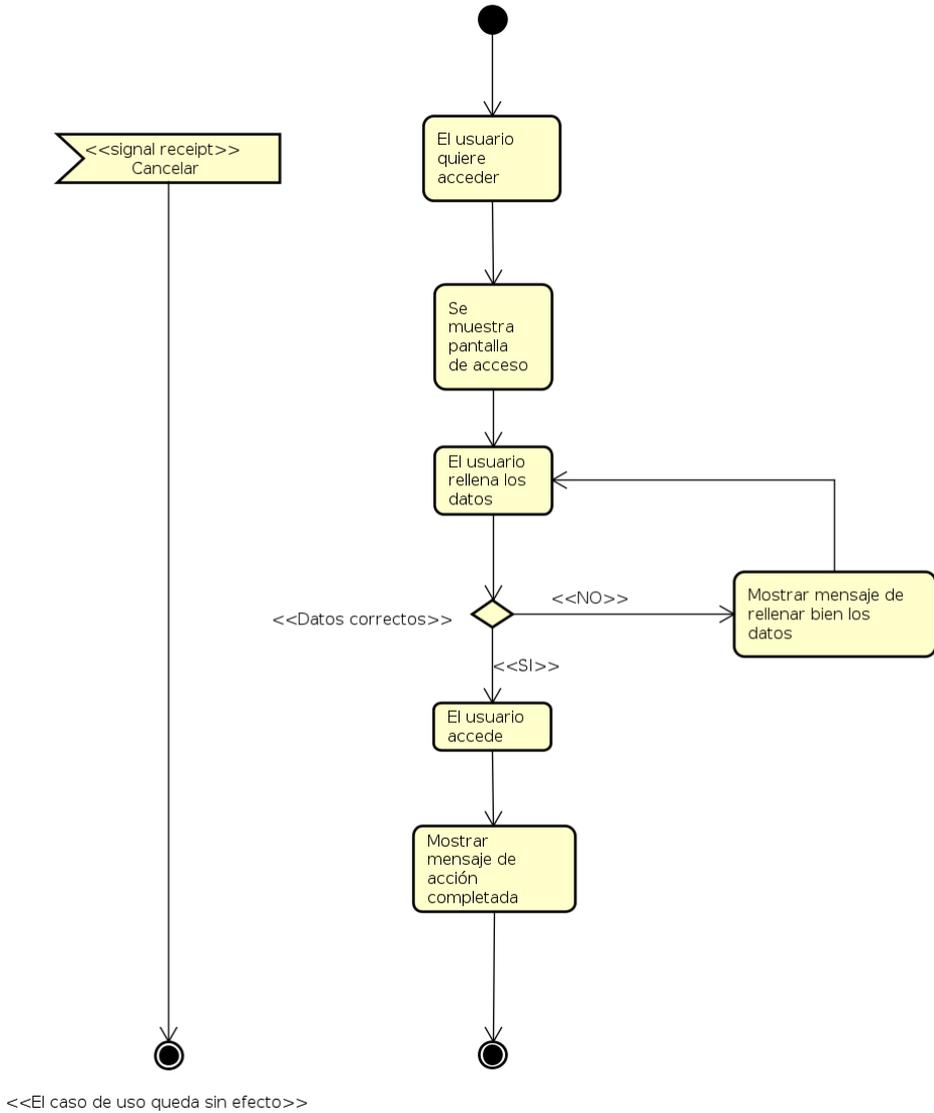


Figura 4.48: Diagrama de actividad del caso de uso CU12

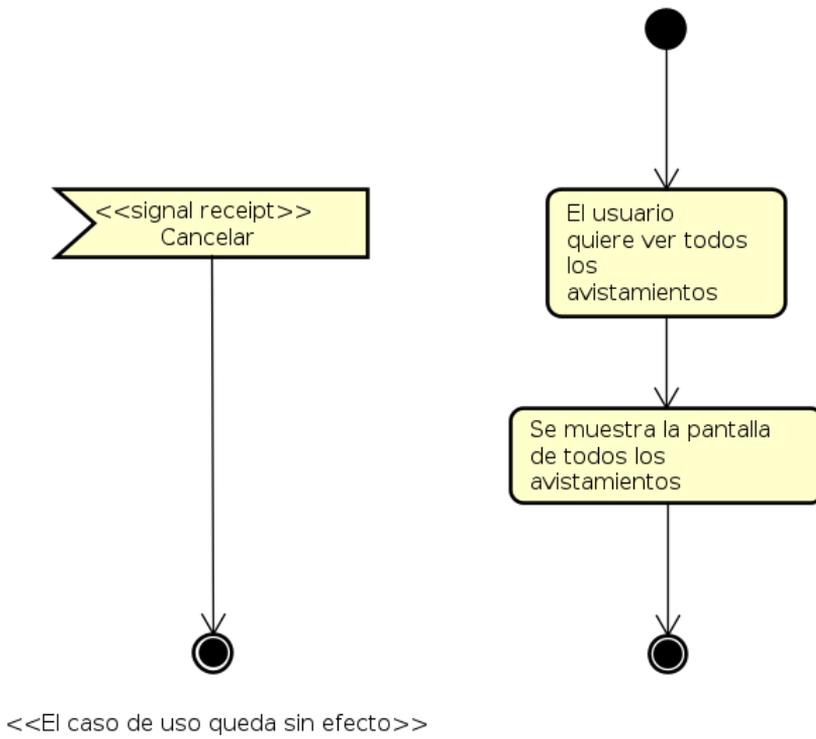


Figura 4.49: Diagrama de actividad del caso de uso CU13

Capítulo 5

Arquitectura y diseño

En este capítulo se detalla la arquitectura utilizada, así como las tareas a realizar en la etapa de Diseño en el proceso de Ingeniería del Software.

5.1. Arquitectura

5.1.1. Arquitectura del software

Lo primero que hay que definir es la arquitectura que se va a seguir. En este caso se va a seguir el patrón arquitectónico MVP(Modelo Vista Presentador) que se basa en el modelo MVC (Modelo Vista Controlador). A continuación veremos las diferencias entre ambos.

En el MVC, la vista avisa al modelo de cualquier cambio producido en la interfaz para que éste se mantenga actualizado con respecto al estado del sistema. La información puede pasarse en la propia notificación, o después de la notificación, la vista puede consultar el modelo directamente para obtener los datos actualizados. Por el contrario, en el MVP, la vista no sabe nada sobre el modelo y la función del presentador es la de mediar entre ambos, enlazando los datos con la vista.

En el modelo MVC activo, la vista tiende a tener más lógica porque es responsable de manejar las notificaciones del modelo y de procesar los datos. En el modelo MVP, esa lógica se encuentra en el presentador, haciendo a la vista sin una funcionalidad clara ni control. Su único cometido es representar la información que el presentador le ha proporcionado.

El patrón MVP, en contra al MVC, es más sencillo de implementar en las aplicaciones para móviles.

Al aplicar este patrón tenemos las tres capas siguientes:

- **Presentador:** se encargará de manejar la comunicación entre la vista y el modelo. Recupera los datos del modelo y se los devuelve a la vista formateados. También, decide qué ocurre cuando se interactúa con la vista lo que supone una diferencia con el patrón MVC. Estará implementado por clases java.
- **Vista:** se suele implementar mediante una Activity aunque puede ser un Fragment o una View que contendrá una referencia al Presentador. La función de la vista será llamar a un método del Presentador cada vez que se realice una acción sobre la interfaz. En resumen, mostrar la información y manejar los eventos de entrada y salida del usuario. En nuestro caso, se implementará con actividades y fragmentos.
- **Modelo:** será la puerta de enlace a la capa de dominio o lógica de negocio, es decir, el proveedor de los datos que queremos mostrar en la vista. Será implementado mediante clases java.

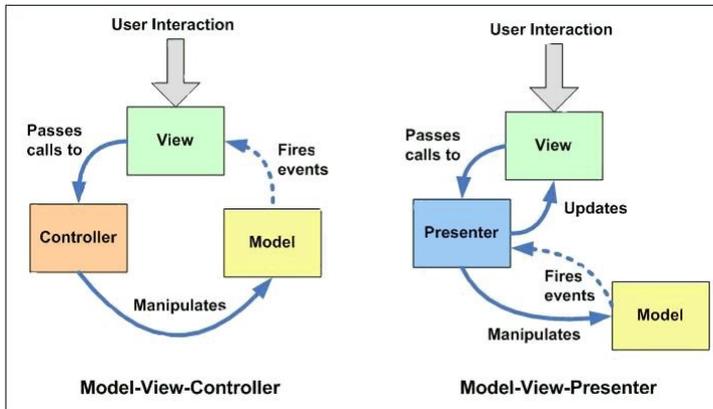


Figura 5.1: Modelo Vista Controlador

En la figura 5.1 se puede ver claramente la diferencia entre MVC y MVP.

5.1.2. Ciclo de vida de Actividades y Fragmentos

Android se basa en las *Activities* (actividades) y en los *Fragments* (fragmentos) para guardar toda la lógica de la aplicación. Y dadas las interacciones entre el usuario y su dispositivo, se disparan una serie de eventos que afectan a las Actividades y Fragmentos. Y por lo tanto se compone el siguiente ciclo de vida:

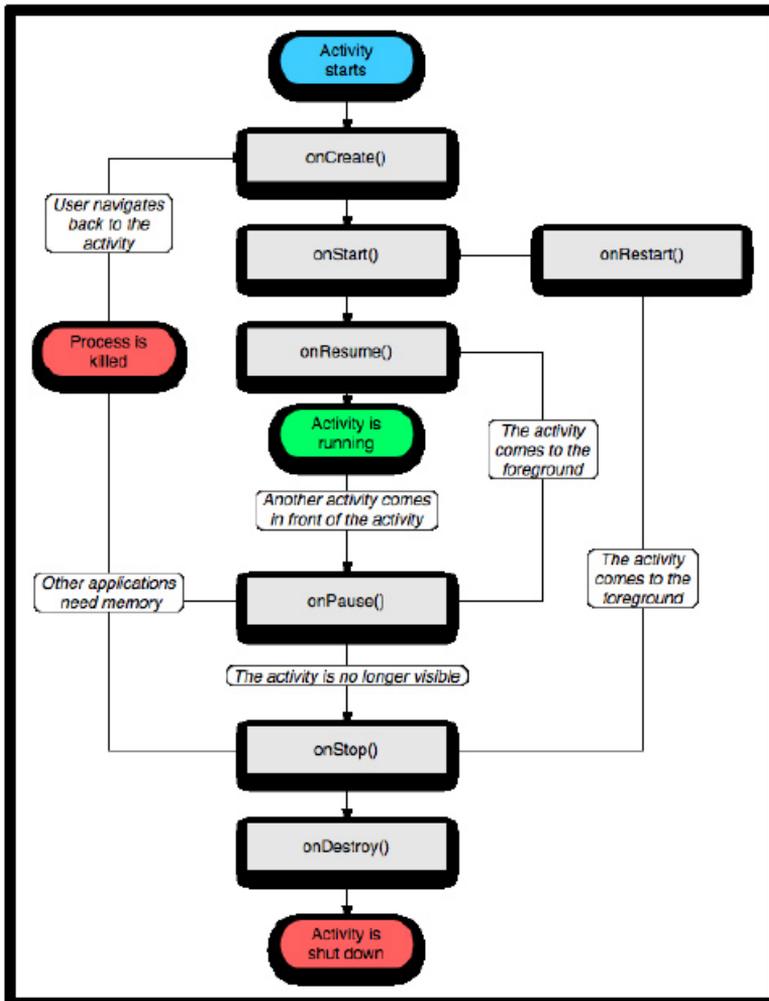


Figura 5.2: Ciclo de vida de una actividad

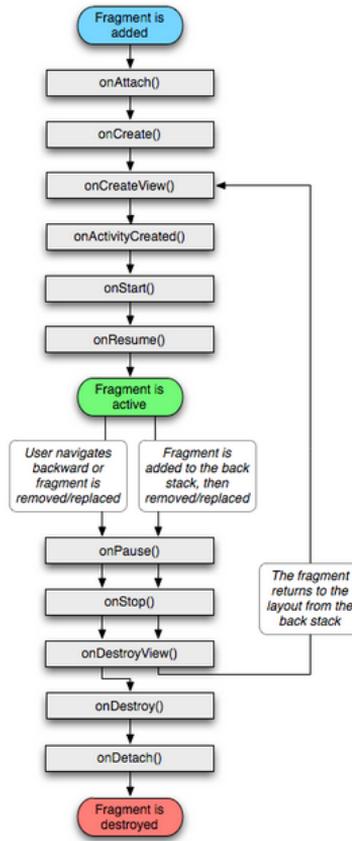


Figura 5.3: Ciclo de vida de un fragmento mientras su actividad esta activa

5.2. Diseño

5.2.1. Diseño de la base de datos

Para la gestión de la persistencia de la aplicación se ha elegido el sistema gestor de bases de datos MySQL que actúa sobre una base de datos relacional necesaria para compartir la información entre todos los usuarios sobre los avistamientos de cada usuario.

El diagrama entidad-relación 5.4 hace referencia a cómo esta construida y diseñada la base de datos, con sus tablas y relaciones entre ellas.

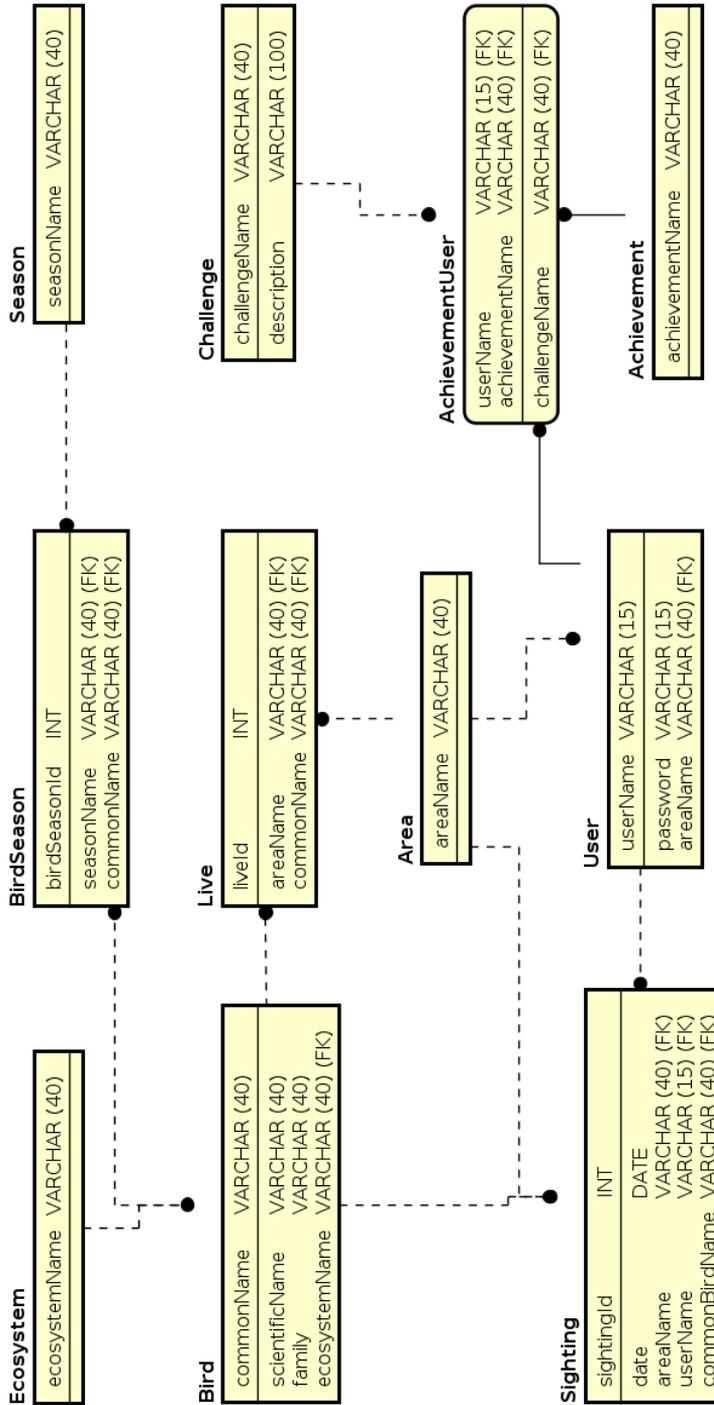


Figura 5.4: Diagrama Entidad-Relación

5.2.2. Principios de diseño

Para que la aplicación tenga una buena interfaz de usuario y una buena usabilidad, se van a tener en cuenta estos principios de diseño:

- **Sencillez:** Emplear acciones, iconos, palabras, y controles “naturales”. Las tareas complejas deben ser divididas en tareas más simples.
- **Estructura:** Organizar la IU (Interfaz de Usuario) de forma significativa. Las características relacionadas deben aparecer juntas y viceversa. La IU debe reflejar las ideas del usuario sobre el dominio y sobre cómo realizar las tareas
- **Consistencia:** Importancia de la uniformidad en la apariencia, disposición y comportamiento de los elementos de una IU. El usuario se hace un modelo mental de cómo funciona una IU a partir de su experiencia y espera que el resto de la IU se comporte igual
- **Tolerancia:** Se centra en la necesidad de prevenir los errores del usuario.

5.2.3. Guías de diseño

Para conseguir esos principios de diseño. Se tendrá en cuenta la guía de diseño de **Facilidad de aprendizaje**, pues es la principal guía de diseño que nos interesa para este proyecto. Se basa en el atributo de usabilidad:

- **Facilidad de aprendizaje:** Se refiere a la facilidad para aprender a usar un sistema. Hay que tener en cuenta la facilidad para aprender lo más básico y la facilidad para aprender todas las funcionalidades. Suelen responder a las preguntas: ¿Es posible para el usuario deducir cómo usar un sistema explorando la interfaz y probando diferentes acciones? ¿Cómo de difícil será aprender toda la funcionalidad de esta manera?

Se ha implementado en todas las funcionalidades del sistema, intentando que sean lo más intuitivas posibles, mostrando iconos o textos que se relacionen con la acción que realizan. También se tienen que juntar las funcionalidades parecidas pues es lógico que lo que se parece esté junto. Todo esto para facilitar el uso y aprendizaje de la aplicación.

También se tendrá en cuenta la guía **Facilidad de recuerdo**. Esta guía está muy relacionada con la anterior. Pues al ser fácil de aprender, es fácil de recordar. Se basa en el atributo de usabilidad:

- **Facilidad de recuerdo:** Se refiere a la facilidad para aprender a usar un sistema. Hay que tener en cuenta la facilidad para aprender lo más básico y la facilidad para aprender todas las funcionalidades. Suele responder a las preguntas: ¿Es posible para el usuario deducir cómo usar un sistema explorando la interfaz y probando diferentes acciones? ¿Cómo de difícil será aprender toda la funcionalidad de esta manera?

Se aplicará desarrollando las funcionalidades parecidas con unos pasos a seguir comunes o muy parecidos. Así como una interfaz igual o parecida para cada pantalla de la aplicación, o para cada pantalla de mostrar listas de los mismos datos.

5.2.4. Diagramas de interacción

Los diagramas de secuencia siguientes muestran como el usuario interactúa con el sistema, y dentro del sistema, se ve como interactúan las clases entre ellas.

Cada caso de uso tiene asociado un diagrama de secuencia correlativamente en orden de aparición.

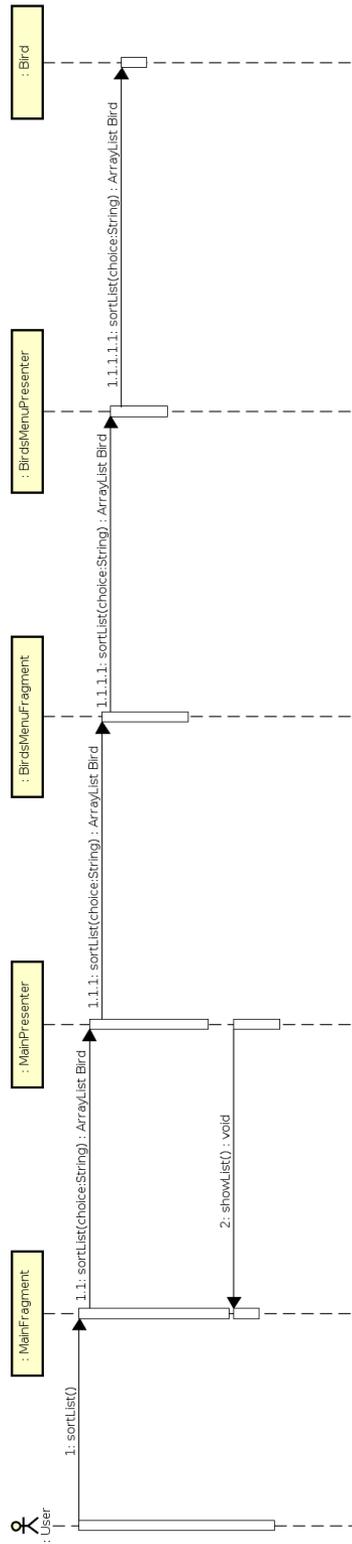


Figura 5.5: Diagrama de secuencia 1

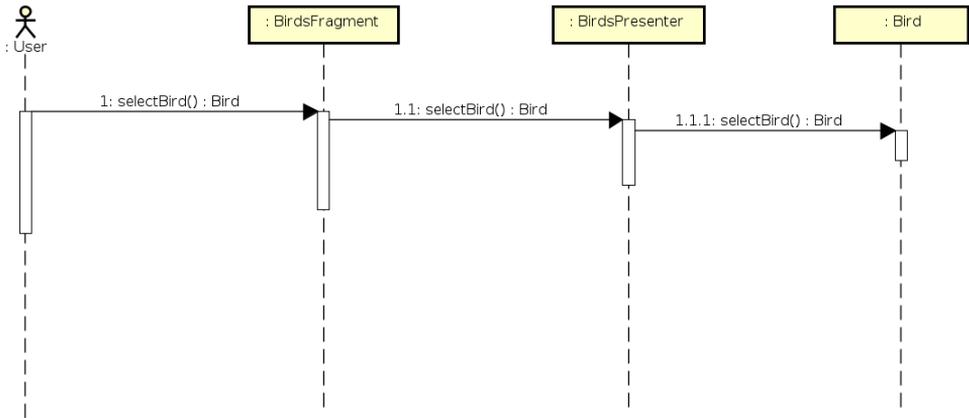


Figura 5.6: Diagrama de secuencia 2

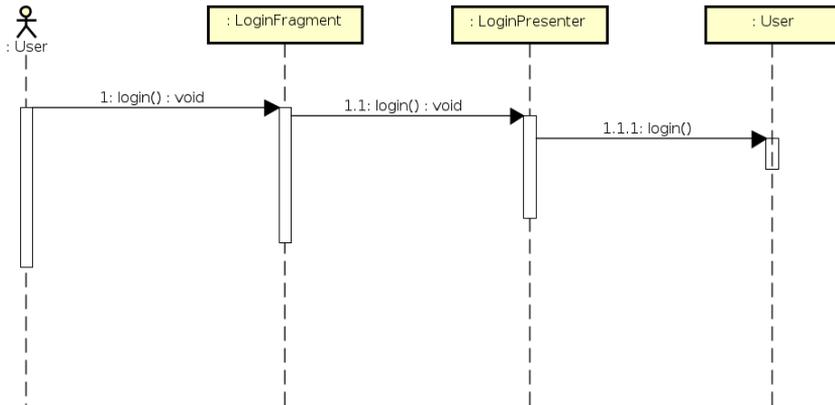


Figura 5.7: Diagrama de secuencia 12

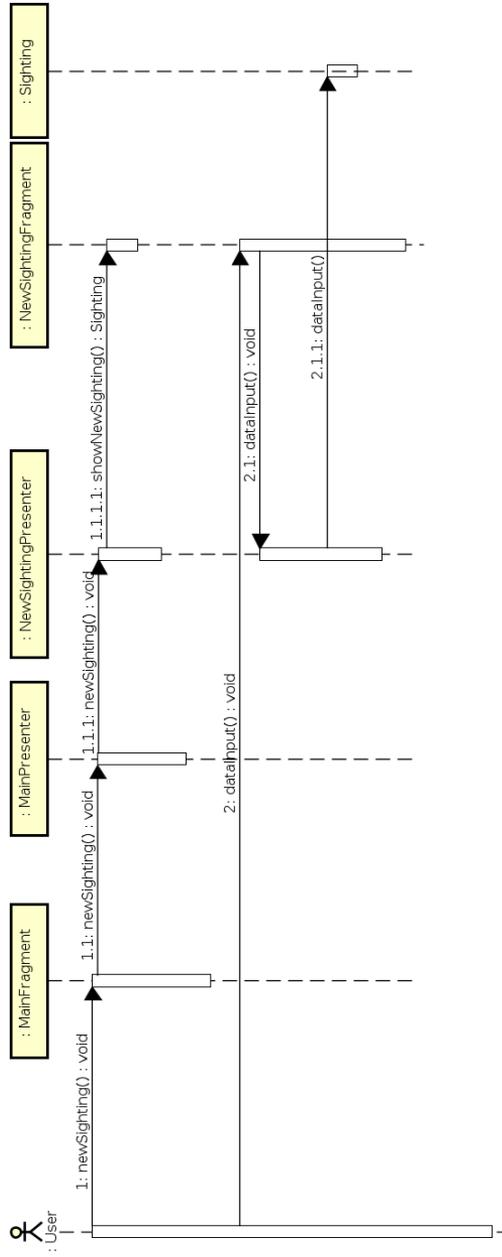


Figura 5.8: Diagrama de secuencia 3

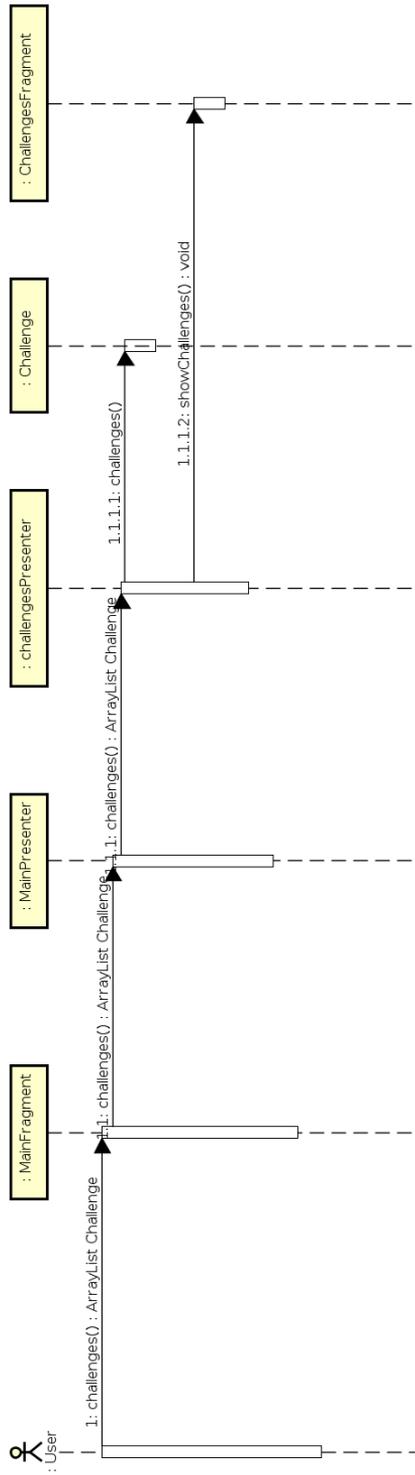


Figura 5.9: Diagrama de secuencia 4

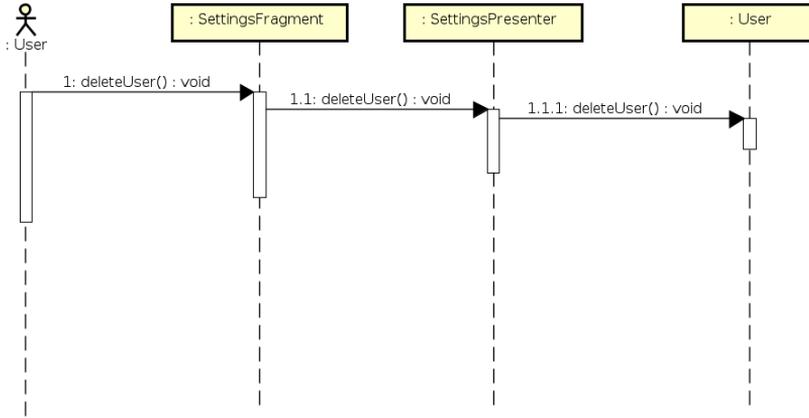


Figura 5.10: Diagrama de secuencia 5

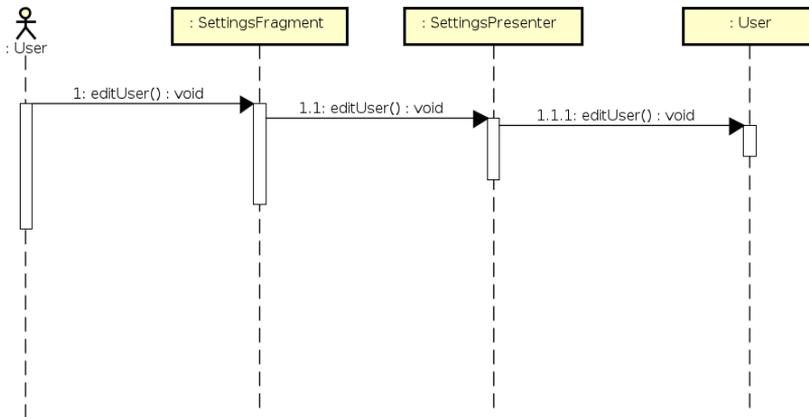


Figura 5.11: Diagrama de secuencia 6

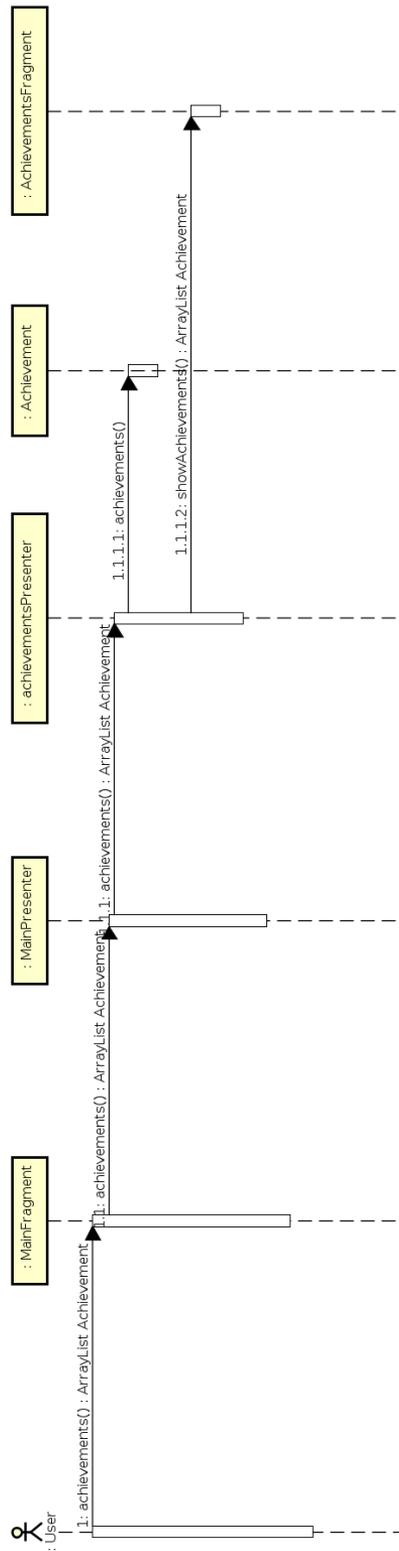


Figura 5.12: Diagrama de secuencia 7

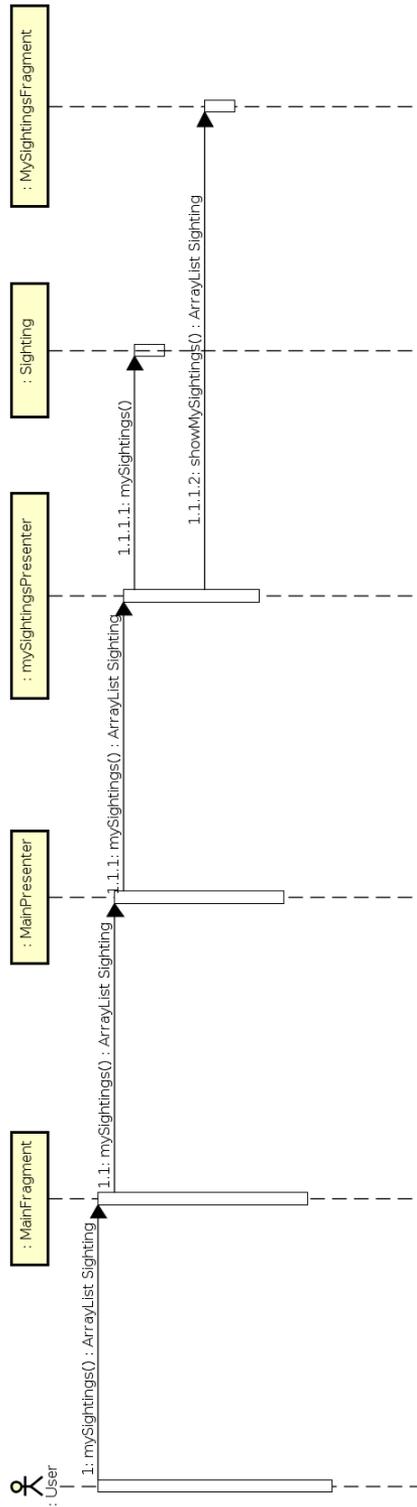


Figura 5.13: Diagrama de secuencia 8

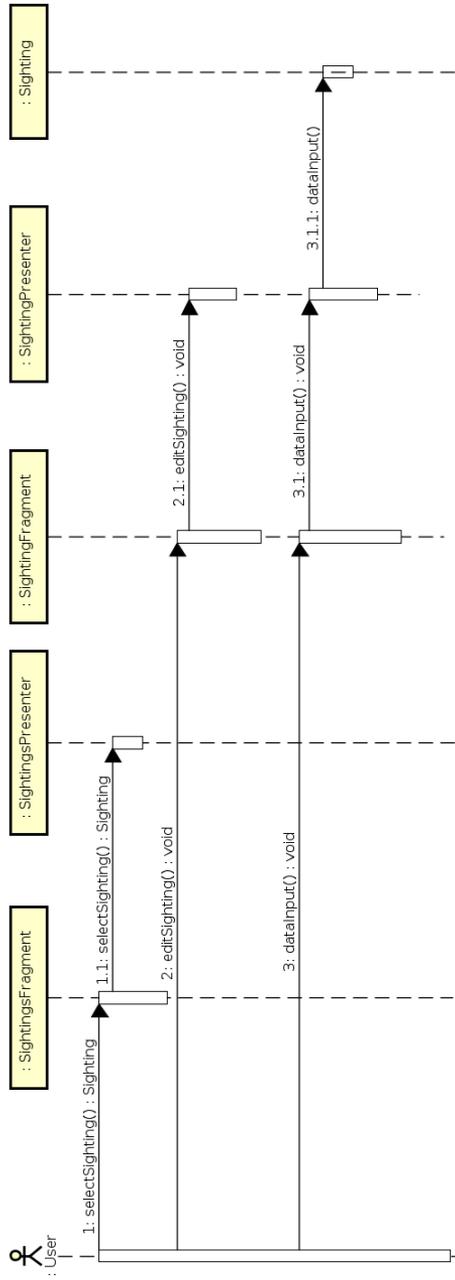


Figura 5.14: Diagrama de secuencia 9

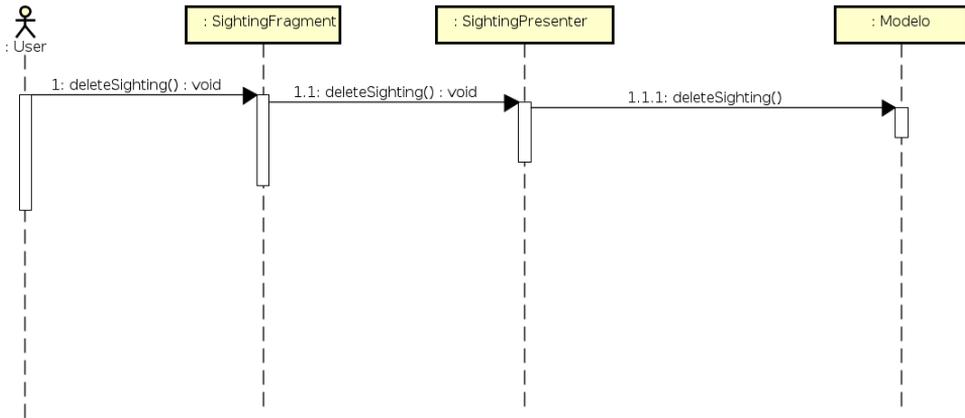


Figura 5.15: Diagrama de secuencia 10

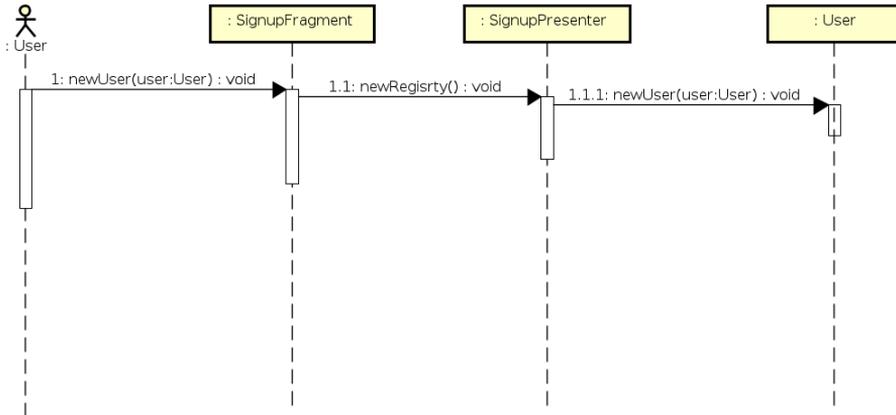


Figura 5.16: Diagrama de secuencia 11

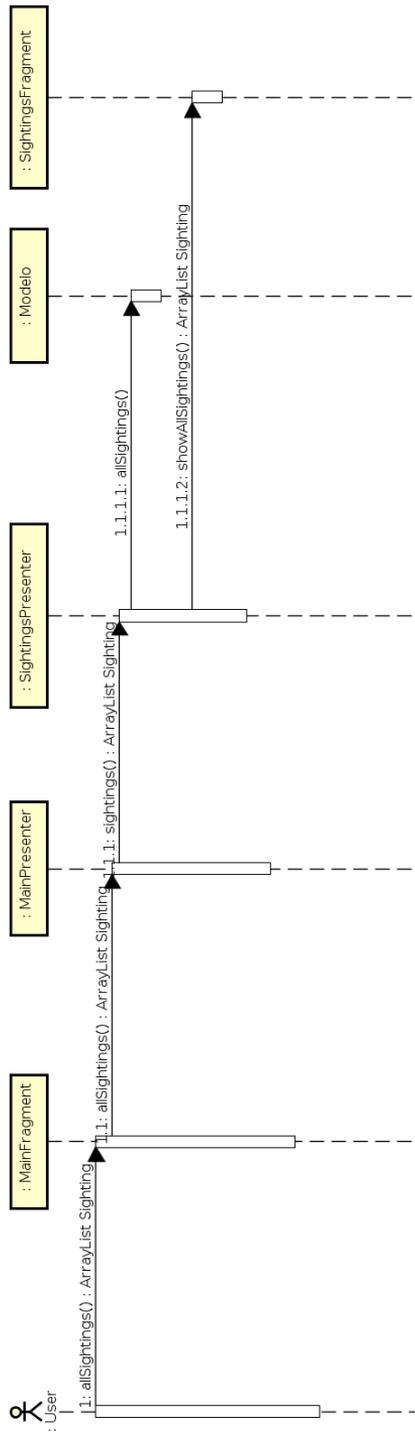


Figura 5.17: Diagrama de secuencia 13

5.2.5. Diagramas de clases de diseño

A continuación se muestran los diagramas de clases de diseño; para mejor organización, se han dividido en tres diagramas: modelo, vista y presentador. Se han omitido los atributos de las clases, ya que se mostraron en la sección de análisis, y dificultarían la lectura.

Como hemos implementado MVP, el flujo de comunicación es el siguiente: las actividades cargan los fragmentos y juntos forman la vista, éstos comunican las interacciones a sus respectivos presentadores y ellos con las clases del modelo para obtener los datos.

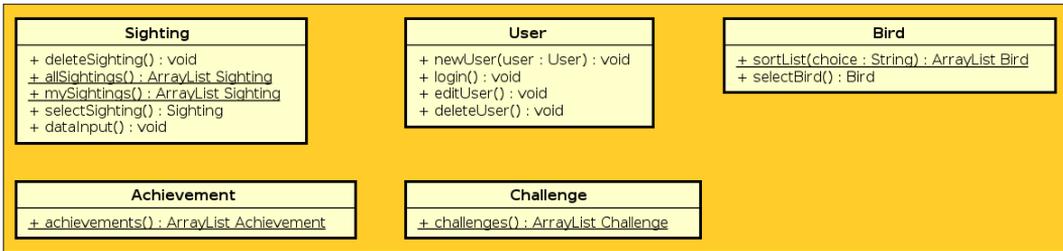


Figura 5.18: Diagrama del modelo

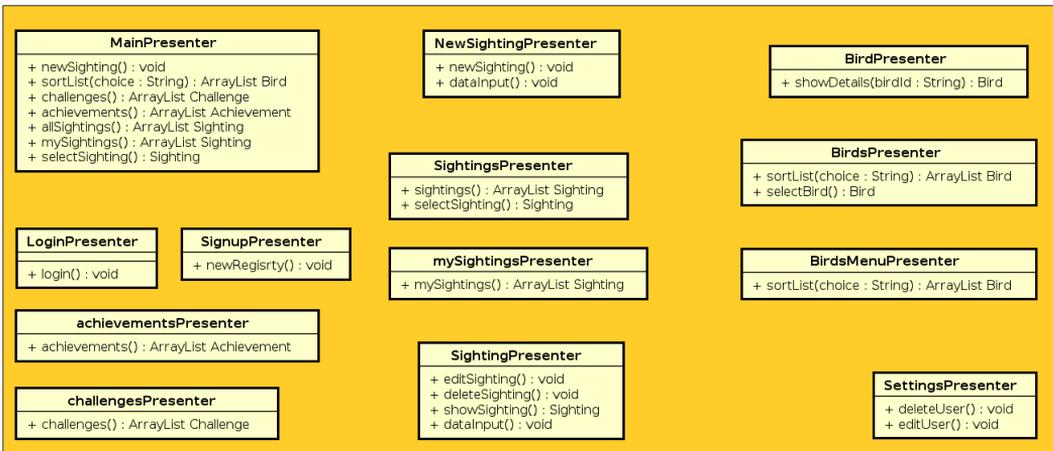


Figura 5.19: Diagrama del presentador

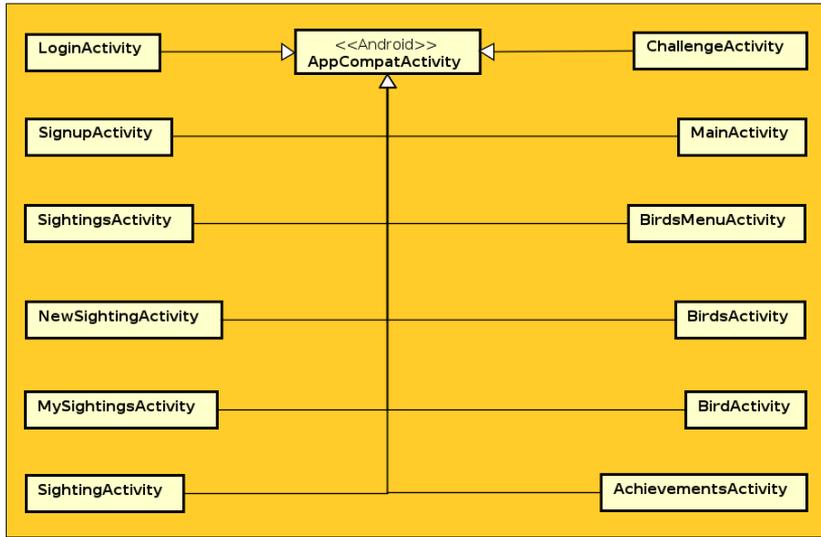


Figura 5.20: Diagrama de la vista - Actividades

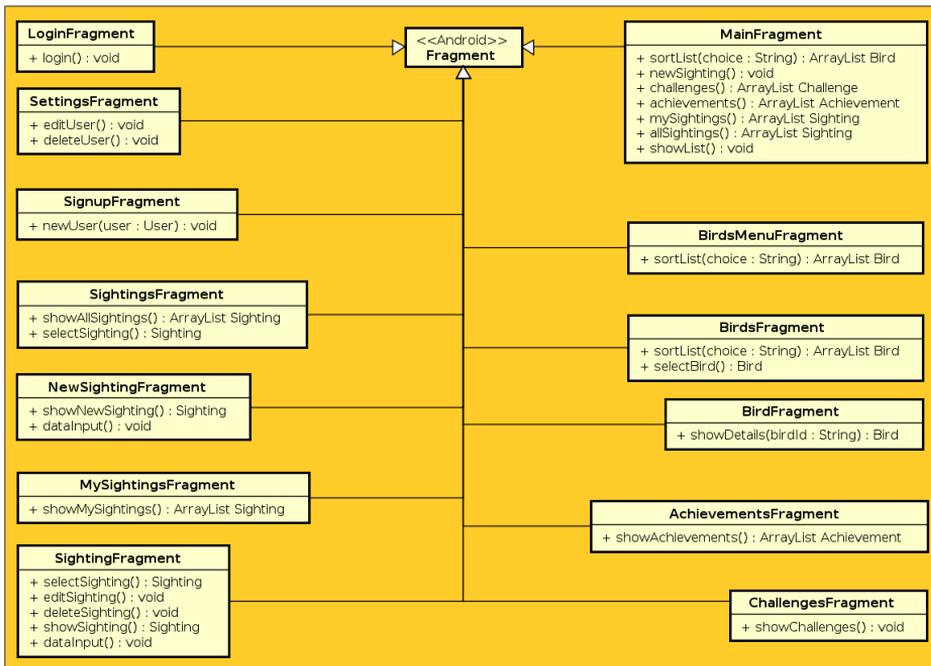


Figura 5.21: Diagrama de la vista - Fragmentos

5.2.6. Prototipo de bajo coste

En este apartado se muestran los bocetos iniciales de la aplicación. Se ha realizado una serie de bocetos dentro de un prototipo de bajo coste. Este prototipo tiene como objetivo evaluar la usabilidad de la interfaz antes del inicio del desarrollo de la aplicación. Se ha aplicado el atributo de usabilidad de facilidad de aprendizaje como primer atributo a tener en cuenta con el fin de mejorar la experiencia de usuario, así como el de facilidad de recuerdo.

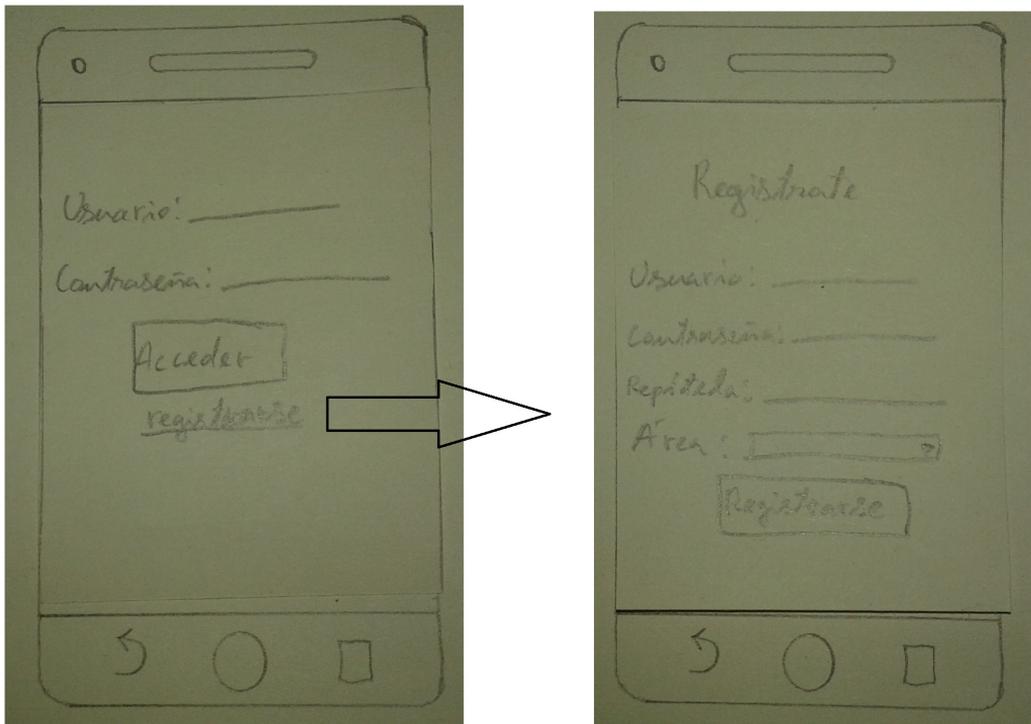


Figura 5.22: Prototipo: Pantalla inicial para acceder y registro

En la figura 5.22 se puede ver a la izquierda la primera pantalla que mostrará la aplicación, que consiste en un formulario para introducir el usuario y la contraseña para poder entrar al resto de funcionalidades. La primera vez no tendrá ningún usuario creado, así que tendrá que pulsar el enlace *Registrarse* para poder crearse la cuenta. Entonces, saldrá la pantalla de la derecha, en la que se deberá rellenar con el nombre de usuario, la contraseña (2 veces para verificar y evitar los errores del usuario, y así aplicar el principio de diseño: Tolerancia) y el área donde resides. Finalmente se pulsará el botón de *Registrarse* para completar la acción.

En la figura 5.23 podemos ver la pantalla principal de la aplicación, donde se encuentra el menú principal, pantalla de la derecha, donde aparecen las acciones principales: Avistamientos de todos los usuarios, mis avistamientos, mis logros, retos y ajustes.

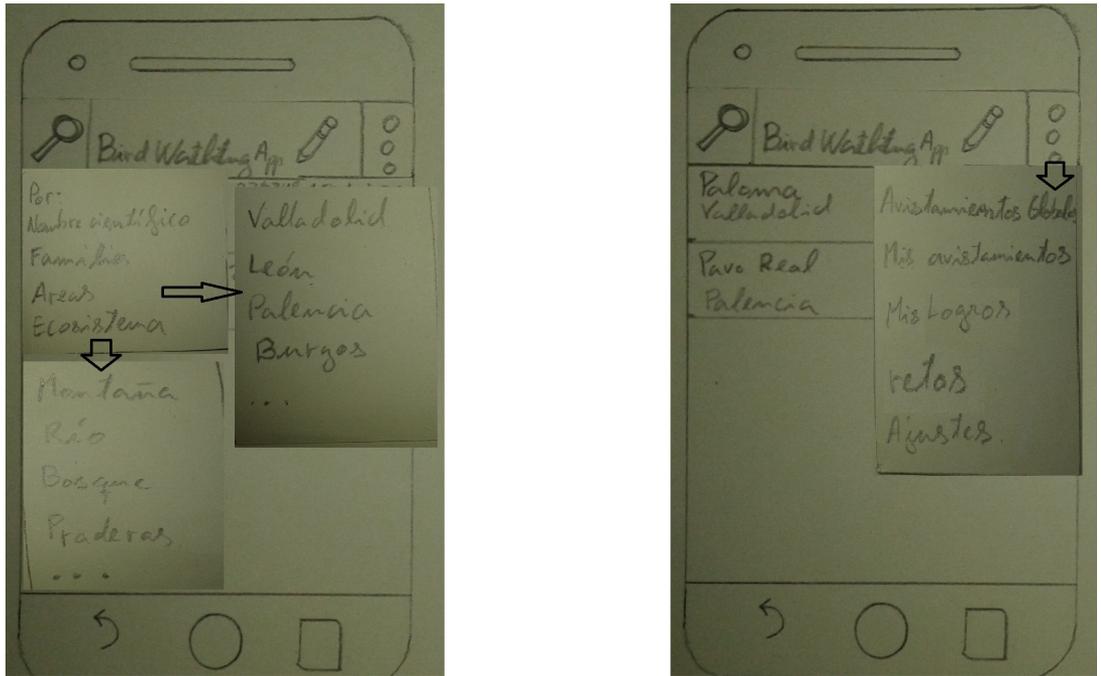


Figura 5.23: Prototipo: Pantalla del menú principal

En la pantalla de la izquierda se muestra lo que pasa cuando se pulsa el icono de la Lupa, que como el principio de diseño Facilidad de aprendizaje dice, es un buen icono ya que asocia normalmente la acción buscar con una Lupa. Seguidamente aparecen las opciones de ordenar por: nombre científico, familia, área y ecosistema. Luego, al pulsar en *área* o *ecosistema* se muestra los distintos valores para ordenar.

5.2. DISEÑO

En la figura 5.24 se puede ver que cuando se pulsa sobre un ave de la lista de aves, el sistema te lleva a otra pantalla con el detalle de las características y fotos sobre ese ave.

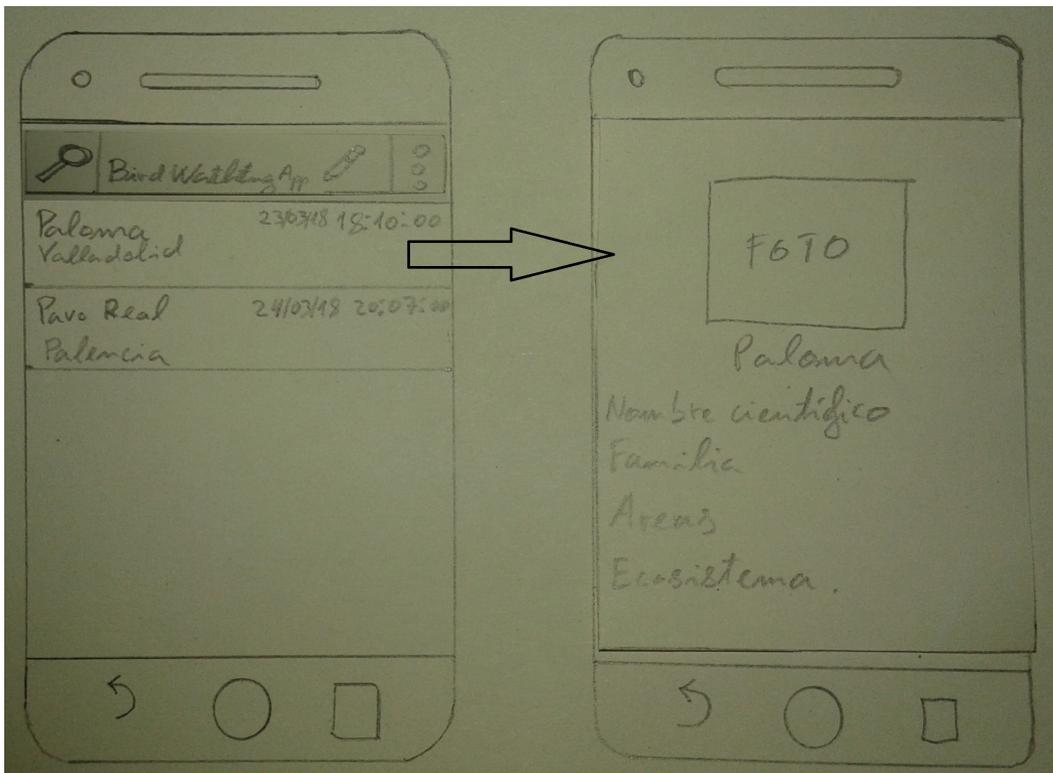


Figura 5.24: Prototipo: Pantalla detalle de un ave

En la figura 5.25 se ve que, cuando se pulsa el botón con forma de lápiz, te lleva a crear un nuevo avistamiento. Esto no era muy intuitivo, y se cambió en la interfaz final. Se sustituyó por un signo más, desde la pantalla de mis avistamientos.

En la figura 5.26 se observa que si se pulsa el menú superior derecho y luego a la opción *Avistamientos globales* el sistema te lleva a la pantalla de la derecha, donde se muestran todos los avistamientos de todos los usuarios de la aplicación, ordenados por la fecha, mostrando la más reciente la primera.

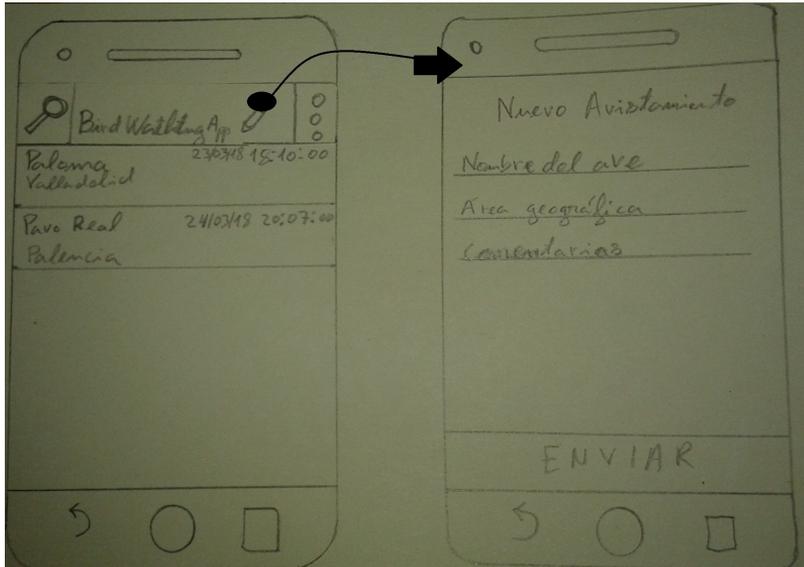


Figura 5.25: Prototipo: Pantalla para introducir un nuevo avistamiento

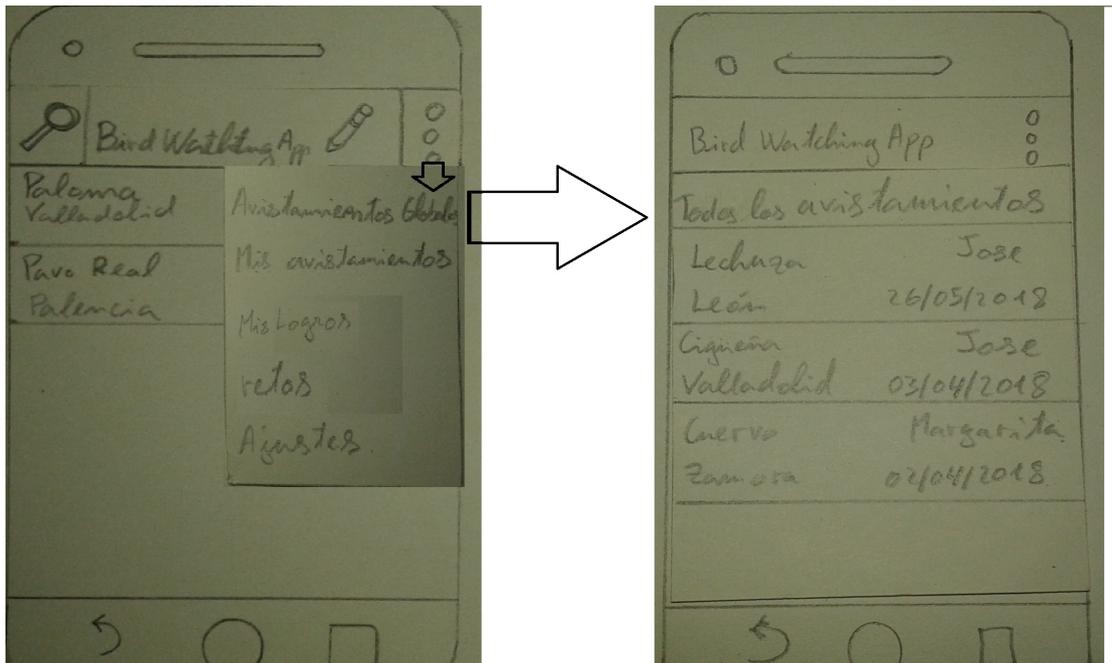


Figura 5.26: Prototipo: Pantalla de los avistamientos de todos los usuarios

5.2. DISEÑO

En la figura 5.27 se puede ver, parecida a la anterior, lo que pasa cuando pulsa el botón *Mis avistamientos*: Se muestra los avistamientos del usuario. No se muestra el nombre del usuario del avistamiento por ejemplo, porque es información redundante.

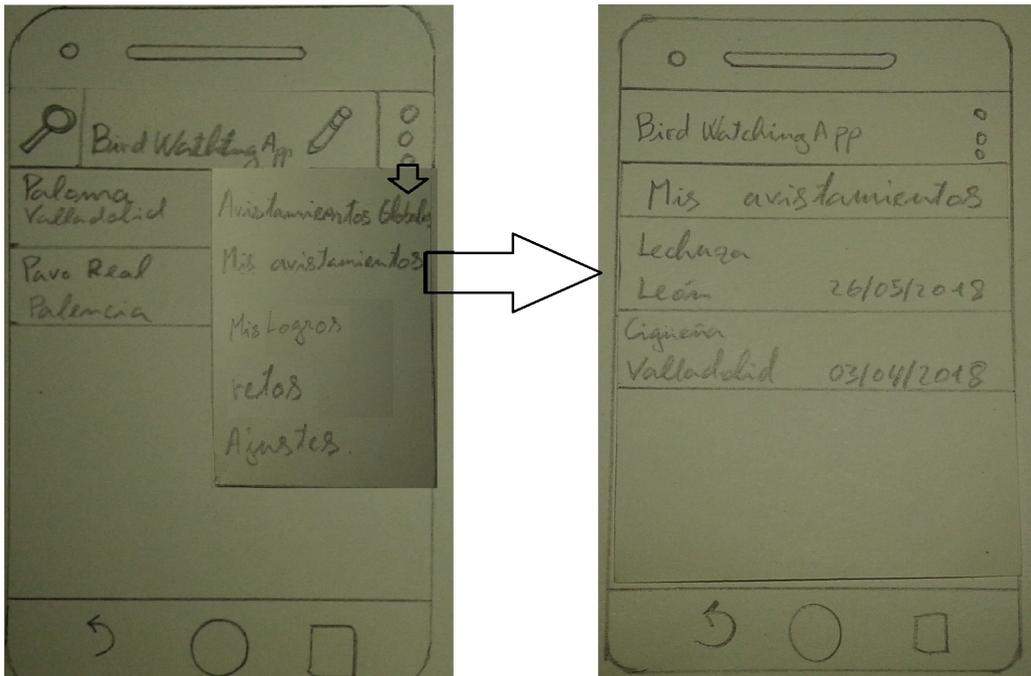


Figura 5.27: Prototipo: Pantalla de mis avistamientos

En la figura 5.28 se muestran los logros del usuario. En la interfaz final se añadió también la descripción del reto superado.

En la figura 5.29 se muestran todos los retos disponibles en la aplicación para que el usuario los vea como metas.

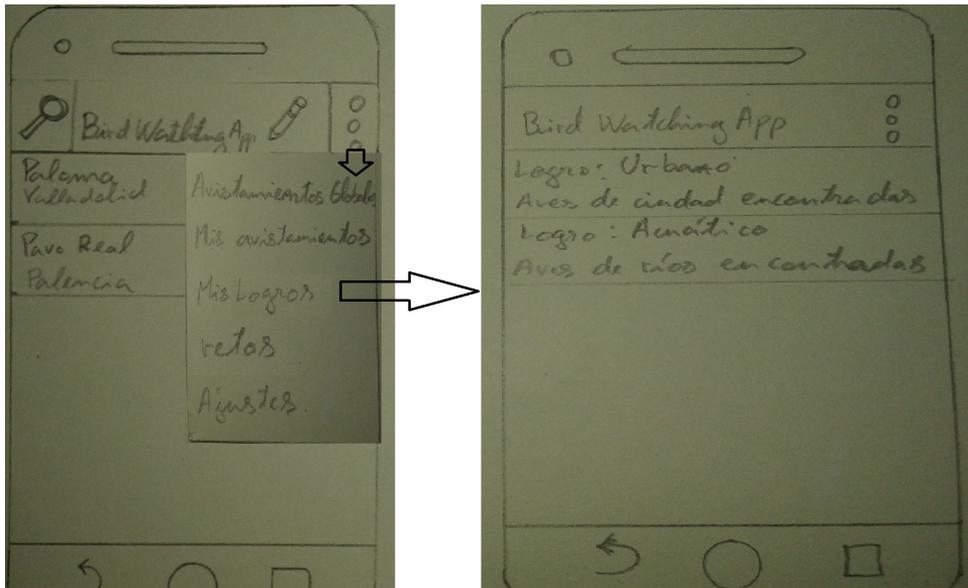


Figura 5.28: Prototipo: Pantalla de mis logros

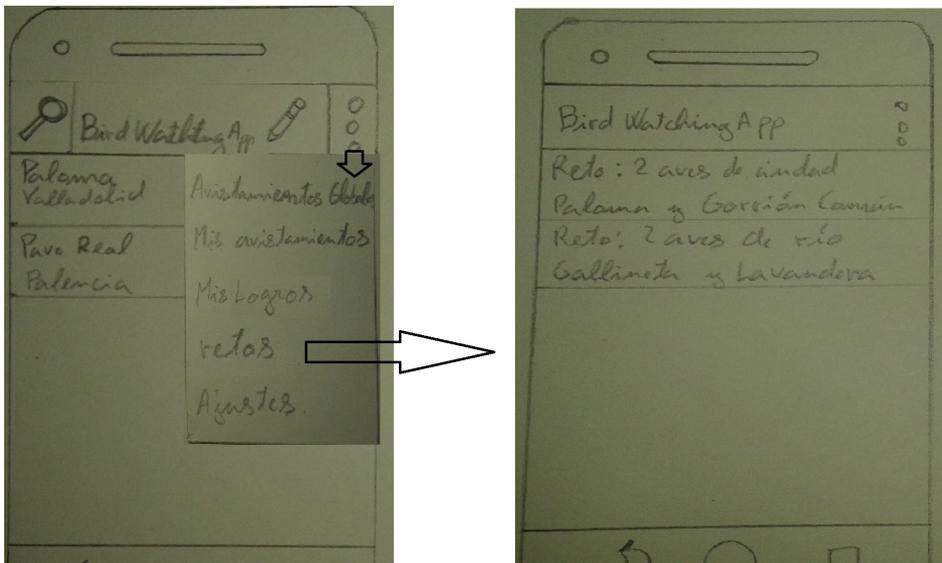


Figura 5.29: Prototipo: Pantalla de los retos

En la figura 5.30 se pueden ver los ajustes del usuario, es decir, la pantalla donde puede el usuario cambiar sus datos como: contraseña y área, o incluso borrar su cuenta. Todo esto para garantizar las propiedades CRUD(Create Read Update y Delete).

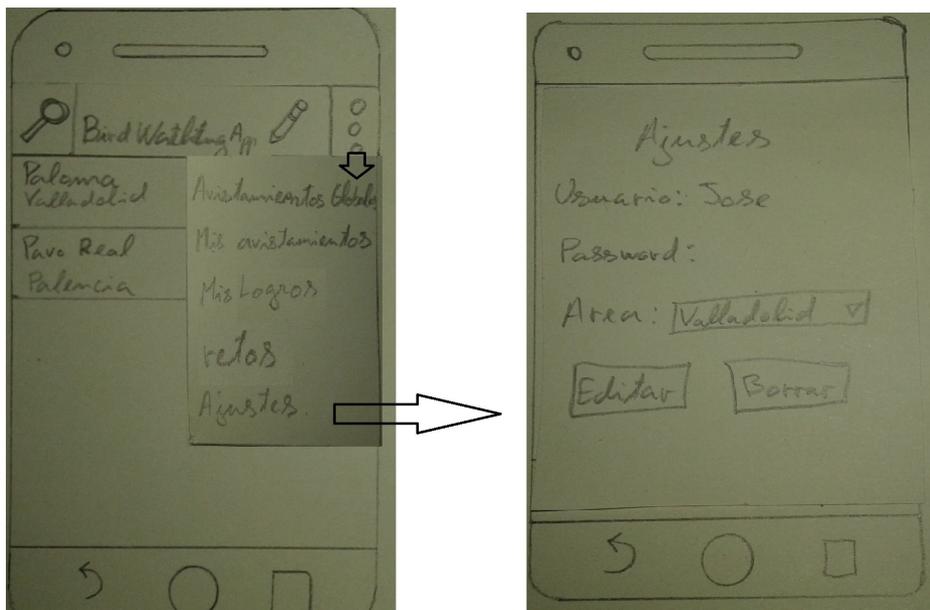


Figura 5.30: Prototipo: Pantalla de ajustes de usuario

5.2.7. Mejoras con respecto al prototipo

Debido a que el anterior era solo un prototipo de bajo coste, en la aplicación final ha habido algunos pequeños cambios en la interfaz de usuario, todos ellos para mejorar la usabilidad.

En las primeras pantallas (5.22) de acceso y registro, no ha habido ningún cambio, más que se añadió una imagen de fondo.

Con respecto a la pantalla del menú principal (5.23), se procedió a realizar varios cambios. Lo primero que se modificó fue la lista de aves por una lista de funcionalidades. Los menús que había desplegable se redujeron considerablemente, pasando todas las opciones de elegir cómo mostrar las aves a otra pantalla nueva con esas opciones.

Las pantallas donde se muestran las aves y su detalle (5.24) no cambió mucho, solo se añadió el fondo y se añadieron otras dos fotos al final del detalle de un ave.

En la pantalla de crear un nuevo avistamiento (5.25) se modificó el icono del lápiz a un signo de sumar (+) para indicar que se quiere añadir un avistamiento.

En la pantalla de mostrar todos los avistamientos (5.26) no cambió más que el lugar para acceder a ellos, que se puso directamente en el menú principal en el medio de la pantalla.

En la pantalla de mostrar mis avistamientos (5.27) se modificó el lugar para acceder a ellos, que se puso directamente en el menú principal en el medio de la pantalla.

En la pantalla de mostrar mis logros (5.28) se cambiaron el orden de los retos y mis logros.

En la pantalla de mostrar los retos (5.29) y en la pantalla de mostrar mis ajustes (5.28) no se realizó ningún cambio.

Capítulo 6

Implementación

Para desarrollar este proyecto ha sido necesario crear una API para la comunicación entre los usuarios y la base de datos y ésta con otros usuarios. La API ofrece la posibilidad de obtener e introducir los datos en la base de datos. Luego ya se ha creado la aplicación Android para mostrar los datos de los usuarios.

Después de tener ambos elementos, desde la aplicación se han hecho las llamadas HTTP correspondientes a la API para obtener los datos a mostrar y así completar la comunicación entre varios usuarios.

Todo el código fuente tanto de la aplicación como de la API se encuentran adjuntados en el CD.

6.1. Aplicación android

6.1.1. Tecnología empleada

La plataforma de desarrollo de la aplicación ha sido **Android**, a pesar de ser un requisito, nos da las siguientes ventajas:

- Es un sistema operativo de código abierto, lo que significa que no supone un coste adicional usarlo.
- Al ser de código abierto, hay mucha más documentación accesible, así como múltiples ejemplos.
- Es el sistema operativo mayoritario en los dispositivos móviles actualmente.
- Android usa el lenguaje de programación Java, el cual se conoce y no requiere de coste adicional de aprendizaje.

6.1.2. Entorno de desarrollo

Para el desarrollo de la aplicación móvil se ha optado por utilizar **Android Studio**, el entorno de desarrollo oficial para aplicaciones Android.

Entre sus diversas características *Android Studio* ofrece: renderización en tiempo real, soporte para la construcción basada en Gradle, herramientas para detectar problemas de rendimiento, usabilidad y compatibilidad, entre otros.

6.1.3. Versiones de implementación

La implementación de la aplicación *Bird Watching App* se ha realizado en tres fases:

- Versión 1.0: Desarrollo de la aplicación sin datos de la base de datos. Solo comunicación entre pantallas y tareas.
- Versión 2.0: Desarrollo de la aplicación con datos de la base de datos, cargándolos en las distintas pantallas como listas o pantallas detalle.
- Versión Final: Aplicación con todas las funcionalidades, datos de la base de datos e imágenes añadidas. Así como la versión final de la interfaz de usuario.

Toda la aplicación ha sido desarrollada en Android. Con un total de 24 clases java y 13 actividades y fragmentos.

6.1.4. Decisiones de implementación

Versión de Android soportadas

En el Requisito no funcional tercero se especifica el nivel de API mínimo para la aplicación, API 19 (Versión 4.4 KitKat). Aunque el nivel objetivo ha sido la API 26.

Teniendo en cuenta la cuota de dispositivos móviles que usan Android. Con este nivel mínimo se alcanza una cuota de mercado del 90,04 %, a fecha del inicio de este proyecto, un ratio más que suficiente. Aparte que es la versión del sistema instalado en el móvil disponible para probarla.

Recursos y bibliotecas externas utilizadas

Las bibliotecas siguientes se han usado para poder desarrollar esta aplicación:

- **Android RecyclerView:** nos permite visualizar listas de datos separando los elementos individualmente. Usada para mostrar ordenadamente las listas de aves, avistamientos, retos y logros.
- **Android CardView:** nos permite trabajar con tarjetas para visualizar los datos. En nuestro caso, estas tarjetas están integradas en el RecyclerView para mostrar un elemento de la lista.
- **Picasso:** nos permite obtener imágenes estáticas de forma fácil y eficiente, cacheando las imágenes automáticamente. Usado para cargar las imágenes de las aves.
- **Okhttp:** nos permite realizar las peticiones HTTP de forma fácil. Usadas para hacer las llamadas a la API y así poder obtener los datos.

Permisos requeridos por la aplicación

Un posible inconveniente de esta aplicación es que necesitas de una conexión a Internet en todo momento. Pero actualmente es mucho más común que todo dispositivo acceda a Internet, así que no es un problema grave.

Así que, el único permiso requerido por la aplicación es el de Internet. Necesario para obtener los datos y estar actualizado con respecto a los avistamientos.

6.2. API

Para poder llevar a cabo toda la comunicación entre la base de datos y la aplicación se ha elegido hacer una API (Application Programming Interface) para ofrecer los datos de la base de datos de una forma estructurada y segura. Se ha desarrollado en la tecnología **Nodejs**.

6.2.1. Tecnología empleada

Para el desarrollo de la API, se ha utilizado la tecnología **Nodejs** que es un entorno de *JavaScript* para ejecutar en momento de ejecución. Al escribirse en *JavaScript* tiene las mismas características y ventajas:

- Es de código abierto y no supone un coste adicional.
- Tipado dinámico, el tipo esta asociado al valor y no a la variable.
- Lenguaje Interpretado, no es compilado.
- Permite el uso de promesas; una promesa es el manejo del resultado obtenido de una operación asíncrona.

- Lenguaje ampliamente usado.
- Multitud de ejemplos y manuales actuales.

6.2.2. Entorno de desarrollo

Para el desarrollo de la API se ha utilizado *Visual Studio*. Que es un IDE (Integrated Development Environment) de Microsoft para varios lenguajes de programación entre ellos: NodeJS, C++, C, JAVA, Visual Basic, Python... Esta herramienta esta orientada a ayudarte a desarrollar código para aplicaciones web o servicios web. Usando un código de colores para las palabras clave y permitiéndote ver la estructura de carpetas de tu proyecto.

6.2.3. Recursos y bibliotecas externas utilizadas

- **Express:** Para el soporte con la base de datos MySQL se ha usado la biblioteca *Express* de nodejs.
- **Sequelizejs:** Y para la interacción con la base de datos se ha usado *Sequalizejs*, que es un ORM (Object-Relational Mapping) para enlazar los objetos relacionales de la base de datos con los datos que hay que devolver mediante la API.
- **Forever:** para poder tener el servicio activo en el servidor en todo momento se ha usado *Forever*.

6.2.4. Entorno

Todo esto ha sido desarrollado e implementado en una máquina virtual de la Escuela de Ingeniería Informática de Valladolid. Que, con una arquitectura Cliente-Servidor, la máquina virtual actúa como servidor y los clientes son los móviles que ejecutan la aplicación Android. Fluyendo toda la comunicación por internet.

Capítulo 7

Pruebas

Con este plan de pruebas lo que se quiere conseguir es :

- Analizar los requisitos y su validez
- Identificar los elementos que se deben probar y la información que se debe de obtener de ellos.
- Describir la estrategia de pruebas para obtener un correcto funcionamiento de la aplicación.

7.1. Alcance de las pruebas

En la realización de este apartado se han realizado todos los casos de uso descritos anteriormente. Por lo tanto solo habrá un grupo de pruebas, relacionadas con el usuario y el uso de sus funcionalidades.

7.2. Requisitos de las pruebas

A continuación se muestra la lista de aquellos elementos (requisitos funcionales y no funcionales y casos de uso) que serán los objetivos de las pruebas.

- El usuario puede registrarse en la aplicación.
- El usuario puede acceder a la aplicación.
- El usuario puede ver las aves disponibles en la aplicación y ordenarlas por distintos criterios.

- El usuario puede ver la descripción de un ave.
- El usuario puede consultar sus avistamientos.
- El usuario puede crear avistamientos.
- El usuario puede modificar un avistamiento propio.
- El usuario puede eliminar un avistamiento propio.
- El usuario puede consultar los avistamientos de todos los usuarios.
- El usuario puede consultar los retos de la aplicación.
- El usuario puede consultar sus logros obtenidos.
- El usuario puede borrar su usuario.
- El usuario puede modificar su usuario.

7.3. Tipos de pruebas

7.3.1. Pruebas de caja blanca

Las pruebas de caja blanca son las relacionadas con el código fuente de la aplicación, son las pruebas que prueban las funciones y algoritmos del programa para ver el buen funcionamiento de éste.

Estas pruebas se han realizado durante el desarrollo de la aplicación, durante la fase de construcción. Cada vez que se completaba una función se comprobaba con una serie de pruebas los posibles errores y las posibles soluciones a ellos.

7.3.2. Pruebas de caja negra

Las pruebas de caja negra son las que dejan de lado el código fuente, y se centran en las funcionalidades del sistema. Sobre todo en los casos de uso. Aunque también comprueba la buena interacción entre el usuario y la aplicación.

Pruebas de funcionalidad

Las primeras pruebas que deben realizarse con el usuario son éstas y se comprueba que al haber una interacción entre el usuario y el sistema, la salida es la esperada por el usuario o no.

Pruebas de interfaz de usuario

Estas pruebas tienen como objetivo verificar la buena navegación entre pantallas y la buena usabilidad de los botones y acciones de la aplicación.

Pruebas de datos e integridad de la base de datos

Estas pruebas se realizan para comprobar la consistencia de la base de datos y evitar repetición en los datos persistentes. Todas estas pruebas implican la inserción, lectura, actualización y borrado (CRUD) correctas de la base de datos.

7.4. Resultados

A continuación se muestran los resultados de todas las pruebas realizadas.

Registrarse	
Prueba-01	Registrarse rellenando todos los datos
Versión	1.0
Descripción	El usuario se registra rellenando todos los datos correctamente
Resultado esperado	El usuario se crea correctamente
Valoración	Correcto

Tabla 7.1: Prueba 01- Registrarse rellenando los datos

Registrarse	
Prueba-02	Registrarse datos incompletos
Versión	1.0
Descripción	El usuario se registra sin rellenar ninguno o alguno de los campos
Resultado esperado	El sistema le dice que le faltan datos de rellenar y no hace nada
Valoración	Correcto

Tabla 7.2: Prueba 02- Registrarse sin rellenar todos los campos

Acceder a la aplicación	
Prueba-03	Acceder a la aplicación con datos correctos
Versión	1.0
Descripción	El usuario intenta acceder a la aplicación con un usuario y contraseña correctos
Resultado esperado	El sistema le deja acceder a la aplicación
Valoración	Correcto

Tabla 7.3: Prueba 03- Acceder a la aplicación con datos correctos

Acceder a la aplicación	
Prueba-04	Acceder a la aplicación con usuario incorrecto
Versión	1.0
Descripción	El usuario intenta acceder a la aplicación con un usuario y contraseña vacíos o no correctos
Resultado esperado	El sistema le dice que las credenciales no son correctas o vacías y espera una nueva entrada de datos
Valoración	Correcto

Tabla 7.4: Prueba 04- Acceder a la aplicación con datos incorrectos

Ver aves disponibles y ordenarlas por distintos criterios	
Prueba-05	Ver aves por nombre común
Versión	1.0
Descripción	El usuario quiere ver las aves ordenadas por nombre común
Resultado esperado	El sistema le devuelve la lista de aves ordenadas por nombre común
Valoración	Correcto

Tabla 7.5: Prueba 05- Ver aves por nombre común

Ver aves disponibles y ordenarlas por distintos criterios	
Prueba-06	Ver aves por nombre científico
Versión	1.0
Descripción	El usuario quiere ver las aves ordenadas por nombre científico
Resultado esperado	El sistema le devuelve la lista de aves ordenadas por nombre científico
Valoración	Correcto

Tabla 7.6: Prueba 06- Ver aves por nombre científico

Ver aves disponibles y ordenarlas por distintos criterios	
Prueba-07	Ver aves por área
Versión	1.0
Descripción	El usuario quiere ver las aves de solo un área
Resultado esperado	El sistema le muestra las aves de ese área
Valoración	Correcto

Tabla 7.7: Prueba 07- Ver aves por área

Ver aves disponibles y ordenarlas por distintos criterios	
Prueba-08	Ver aves por temporada
Versión	1.0
Descripción	El usuario quiere ver las aves que estén un una determinada temporada del año
Resultado esperado	El sistema le muestra las aves de esa temporada del año
Valoración	Correcto

Tabla 7.8: Prueba 08- Ver aves por temporada

Ver aves disponibles y ordenarlas por distintos criterios	
Prueba-09	Ver aves por otro criterio
Versión	1.0
Descripción	El usuario intenta ver las aves ordenadas por otro criterio
Resultado esperado	El sistema no le deja ordenar por otro criterio
Valoración	Correcto

Tabla 7.9: Prueba 09- Ver aves por otro criterio

Ver la descripción de un ave	
Prueba-10	Ver la descripción de un ave
Versión	1.0
Descripción	El usuario quiere ver la descripción de un ave
Resultado esperado	El sistema le muestra la descripción de ese ave
Valoración	Correcto

Tabla 7.10: Prueba 10- Ver la descripción de un ave

Consultar los avistamientos del usuario	
Prueba-11	Consultar sus avistamientos
Versión	1.0
Descripción	El usuario quiere ver sus avistamientos
Resultado esperado	El sistema le muestra sus avistamientos y solo esos
Valoración	Incorrecto, se muestra la hora 2 horas atrasada

Tabla 7.11: Prueba 11- Consultar sus avistamientos

Consultar los avistamientos del usuario	
Prueba-12	Consultar sus avistamientos
Versión	2.0
Descripción	El usuario quiere ver sus avistamientos
Resultado esperado	El sistema le muestra sus avistamientos y solo esos
Valoración	Correcto, se arregló la hora

Tabla 7.12: Prueba 12- Consultar sus avistamientos

7.4. RESULTADOS

Crear avistamientos	
Prueba-13	Crear avistamientos rellenando todos los campos
Versión	1.0
Descripción	El usuario quiere crear avistamientos
Resultado esperado	El sistema recoge los datos y le indica que ha creado un avistamiento
Valoración	Correcto

Tabla 7.13: Prueba 13- Crear avistamientos rellenando todos los campos

Crear avistamientos	
Prueba-14	Crear avistamientos sin rellenar todos los campos
Versión	1.0
Descripción	El usuario no rellena ninguno o alguno de los campos al crear un avistamiento
Resultado esperado	El sistema le indica que le faltan datos por rellenar
Valoración	Correcto

Tabla 7.14: Prueba 14- Crear avistamientos sin rellenar todos los campos

Crear avistamientos	
Prueba-15	Crear avistamientos con datos erróneos que no existen en la aplicación
Versión	1.0
Descripción	El usuario quiere crear un avistamiento con datos erróneos, como un ave o área que no esta en el sistema
Resultado esperado	El sistema no se lo permite
Valoración	Correcto

Tabla 7.15: Prueba 15- Crear avistamientos con datos erróneos que no existen en la aplicación

Modificar avistamiento propio	
Prueba-16	Modificar avistamiento propio
Versión	1.0
Descripción	El usuario quiere modificar un avistamiento propio
Resultado esperado	El sistema se lo permite y cambia los datos
Valoración	Correcto

Tabla 7.16: Prueba 16- Modificar avistamiento propio

Modificar avistamiento propio	
Prueba-17	Modificar avistamiento ajeno
Versión	1.0
Descripción	El usuario intenta modificar un avistamiento ajeno
Resultado esperado	El sistema no se lo permite
Valoración	Correcto

Tabla 7.17: Prueba 17- Modificar avistamiento ajeno

Eliminar avistamiento propio	
Prueba-18	Eliminar avistamiento propio
Versión	1.0
Descripción	El usuario quiere eliminar un avistamiento propio
Resultado esperado	El sistema le elimina el avistamiento seleccionado y vuelve atrás
Valoración	Incorrecto, no vuelve atrás

Tabla 7.18: Prueba 18- Eliminar avistamiento propio

Eliminar avistamiento propio	
Prueba-19	Eliminar avistamiento ajeno
Versión	1.0
Descripción	El usuario intenta eliminar un avistamiento ajeno
Resultado esperado	El sistema no se lo permite
Valoración	Correcto

Tabla 7.19: Prueba 19- Eliminar avistamiento ajeno

Consultar los avistamientos de todos los usuarios	
Prueba-20	Consultar los avistamientos de todos los usuarios
Versión	1.0
Descripción	El usuario quiere ver los avistamientos de todos los usuarios
Resultado esperado	El sistema le muestra todos los avistamientos de todos los usuarios
Valoración	Correcto

Tabla 7.20: Prueba 20- Consultar los avistamientos de todos los usuarios

Consultar los retos	
Prueba-21	Consultar los retos
Versión	1.0
Descripción	El usuario quiere ver los retos de la aplicación
Resultado esperado	El sistema le muestra los retos de la aplicación
Valoración	Correcto

Tabla 7.21: Prueba 21- Consultar los retos

Consultar mis logros	
Prueba-22	Consultar mis logros
Versión	1.0
Descripción	El usuario quiere ver sus logros
Resultado esperado	El sistema le muestra todos sus logros
Valoración	Correcto

Tabla 7.22: Prueba 22- Consultar mis logros

Borrar usuario	
Prueba-23	Borrar usuario
Versión	1.0
Descripción	El usuario quiere borrar el usuario
Resultado esperado	El sistema le borra el usuario y sale de la sesión
Valoración	Incorrecto, no se le cierra la sesión

Tabla 7.23: Prueba 23- Borrar usuario

Borrar usuario	
Prueba-24	Borrar otro usuario
Versión	1.0
Descripción	El usuario intenta borrar otro usuario que no es el suyo
Resultado esperado	El sistema no se lo permite
Valoración	Correcto

Tabla 7.24: Prueba 24- Borrar otro usuario

Modificar usuario	
Prueba-25	Modificar usuario
Versión	1.0
Descripción	El usuario quiere modificar datos de su usuario
Resultado esperado	El sistema le recoge los nuevos datos y se los cambia
Valoración	Correcto

Tabla 7.25: Prueba 25- Modificar usuario

Modificar usuario	
Prueba-26	Modificar otro usuario
Versión	1.0
Descripción	El usuario intenta modificar datos de otro usuario
Resultado esperado	El sistema no se lo permite
Valoración	Correcto

Tabla 7.26: Prueba 26- Modificar otro usuario

Capítulo 8

Conclusiones

Después de haber realizado este trabajo de fin de grado se puede ver que se han cumplido los objetivos fijados. También me ha hecho apreciar la buena planificación y me ha enfrentado a un proyecto a medio-largo plazo, con los retos que ello supone.

De este proyecto se ha podido apreciar que al tener que aprender una nueva tecnología se tarda más de lo esperado. Así como que al planificar se puede pasar por alto alguna tarea y luego tener que dedicarle tiempo, el cual no se tenía planeado y puede retrasar esa fase.

Se ha realizado la aplicación propuesta implementando el modelo arquitectónico MVP, el cual ha sido muy útil para separar la lógica de la aplicación y organizar el proyecto por dentro. Aprender este modelo no llevó mucho tiempo ya que se conocía con anterioridad el modelo MVC.

La funcionalidad más compleja ha sido la de compartir avistamientos entre todos los usuarios. Pues debido a esta función se ha tenido que utilizar un servidor centralizado y todo lo que ello conlleva: la API, la base de datos, etc.

Aunque la parte más importante ha sido la de mostrar la información de las aves, pues al fin y al cabo es lo que busca el usuario. Para ello se han tenido que utilizar imágenes y se ha tenido que pensar como mostrar mucha información en el menor espacio posible.

8.1. Líneas futuras

Como líneas futuras de trabajo se plantean las siguientes mejoras de funcionalidades:

- Incrementar el número de especies en la base de datos con alguna base de datos pública.
- Para ampliar el público objetivo a usuarios experimentados, sería recomendable añadir un buscador dentro de la lista de las aves para poder buscar por nombres de especies.

8.1. LÍNEAS FUTURAS

- Ampliar el rango de área a la totalidad de la Península Ibérica.
- Traducción a otros idiomas con la finalidad de que el turismo ornitológico extranjero se incremente.
- Utilizar la gamificación para el aprendizaje en los usuarios noveles.

Apéndice A

Manual de usuario

A.1. Introducción

Este apartado del documento mostrará todas las funcionalidades de la aplicación y cómo el usuario tiene que usarlas.

A.2. Funcionalidades

A.2.1. Registrarse

Para registrarse el usuario nada mas abrir la aplicación, lo que tiene que hacer es pulsar en el enlace *¿Todavía no tienes una cuenta? Créate una.* Después le llevará a la pantalla donde tendrá que rellenar los datos necesarios para crearse la cuenta: nombre de usuario, contraseña (2 veces) y el área (provincia) al que pertenece.

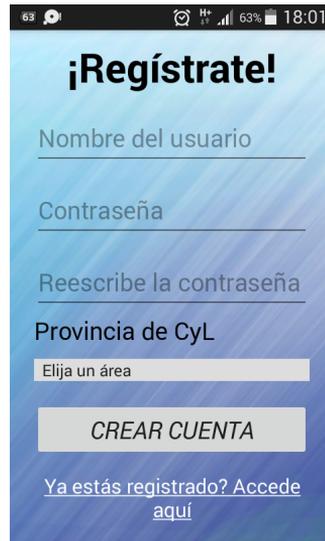


Figura A.1: Registrarse en la aplicación

A.2.2. Acceder

Después de crearte una cuenta, el usuario podrá acceder a la aplicación y resto de funcionalidades. En la pantalla de inicio tendrá que introducir su nombre de usuario y contraseña, y ya accederá a la aplicación.

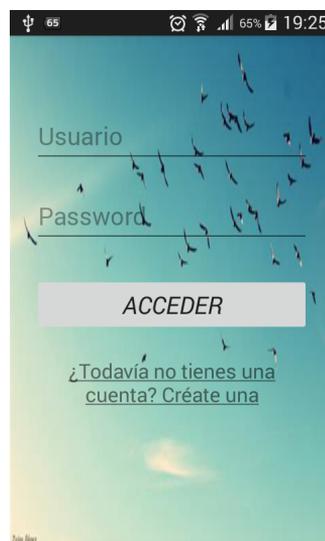


Figura A.2: Acceder a la aplicación

Al acceder le aparecerá la siguiente pantalla con el menú principal y el submenú superior derecho. Donde se puede acceder al resto de funcionalidades que trataremos más adelante.



(a) Menú principal

(b) Menú superior derecho, desplegable

Figura A.3: Menú principal

A.2.3. Ver aves disponibles y ordenarlas por distintos criterios

Una vez haya accedido a la aplicación, se le mostrarán varias opciones. Una de ellas es la de *Búsqueda de aves*, que es la que nos interesa en este caso. Una vez pulsada, nos llevará a otra pantalla donde podrá elegir por qué criterio de búsqueda quiere ordenar las aves: por nombre común, por nombre científico, por provincia o por temporada.

Cada opción te lleva a una de las siguientes pantallas con la lista de las aves ordenadas por el criterio elegido.

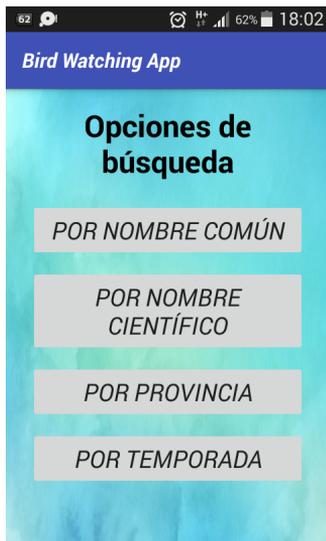
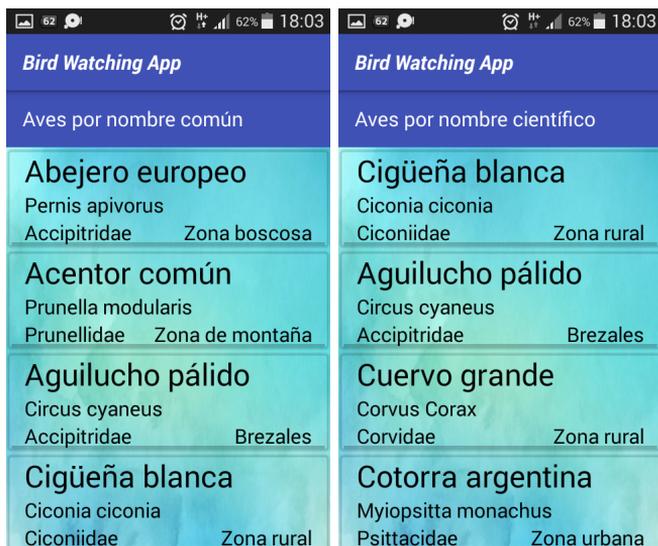
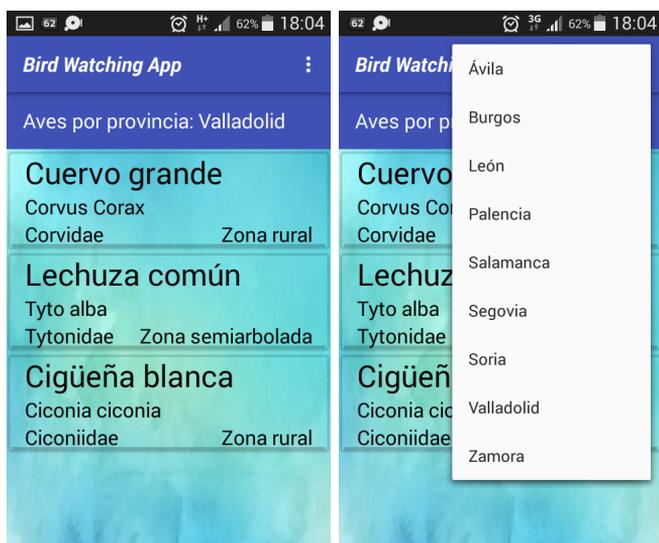


Figura A.4: Menú para ordenar las aves



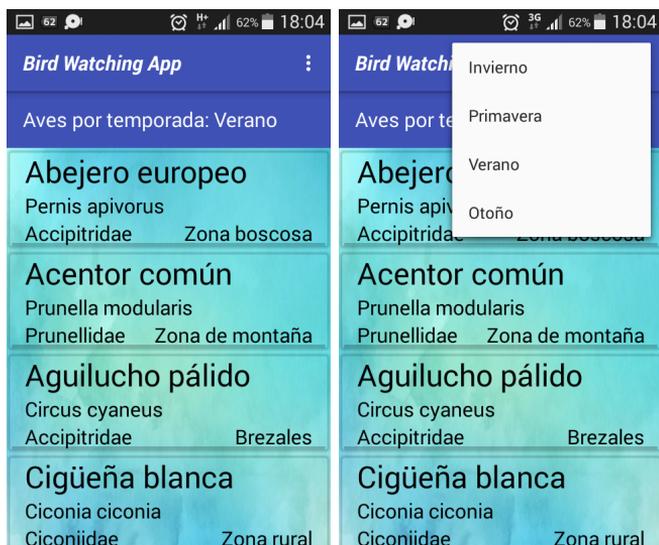
(a) Aves ordenadas por nombre común (b) Aves ordenadas por nombre científico

Figura A.5: Ordenadas por nombre



(a) Aves ordenadas por área (b) Aves ordenadas por área - menú

Figura A.6: Ordenadas por áreas



(a) Aves ordenadas por temporada (b) Aves ordenadas por temporada - menú

Figura A.7: Ordenadas por temporada

A.2.4. Ver la descripción de un ave

Para ver el detalle de un ave, el usuario tiene que haber abierto alguna lista con las aves disponibles. Después puede seleccionar cualquiera, y se le abrirá una pantalla con toda la información de ese ave: nombre común, nombre científico, familia a la que pertenece, estaciones del año en las que se puede ver en las distintas áreas de Castilla y León (las dos primeras iniciales de la provincia), dichas áreas y tres fotografías sobre ese ave.

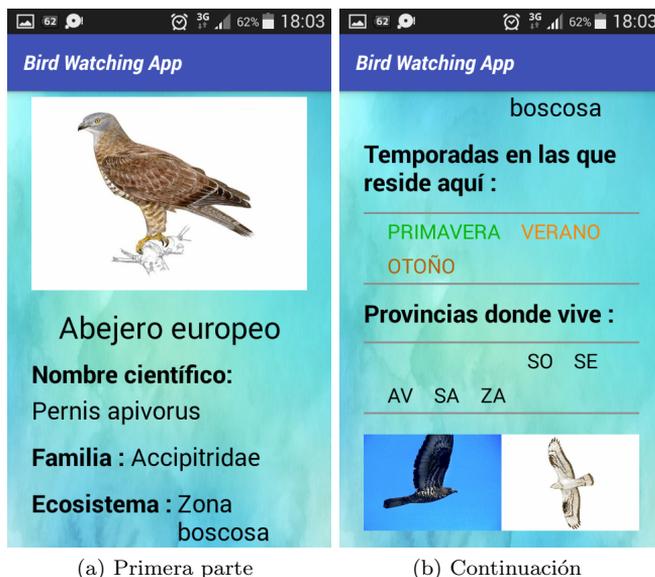


Figura A.8: Descripción de un ave

A.2.5. Crear avistamiento

Para crear un avistamiento el usuario tiene dos caminos: puede pulsar el botón del menú principal o pulsar al signo de más en la pantalla de *Mis avistamientos*. Una vez que pulse alguno de esos botones, le llevará a una pantalla donde tendrá que rellenar qué ave ha visto y en qué provincia. Después, cuando pulse *Crear un nuevo avistamiento*, el sistema guardará el avistamiento y se lo comunicará, y luego puede añadir más o volver a otra pantalla.



Figura A.9: Crear un nuevo avistamiento

A.2.6. Consultar tus avistamientos

Para ver los avistamientos del usuario, tiene que pulsar la opción de *Mis avistamientos* en el menú principal. Después le llevará a la pantalla donde se mostrarán todos los avistamientos hechos por él, ordenados de más reciente a más antiguo.



Figura A.10: Mis avistamientos

A.2.7. Actualizar un avistamiento

Una vez dentro de *Mis avistamientos*, el usuario puede pulsar sobre cualquiera de sus avistamientos y le llevará a otra pantalla con la información de ese avistamiento, así como una opción para actualizar, en forma de lapicero arriba a la derecha. Cuando pulse, se le pondrán visibles los campos para actualizar el avistamiento.



Figura A.11: Avistamiento

A.2.8. Borrar avistamiento

En la misma pantalla en la que actualizas el avistamiento (A.11), esta la opción de borrarlo, en forma de X arriba a la derecha, o en el botón *Borrar*. Una vez que pulse cualquiera de esas opciones, se borrará el avistamiento seleccionado.

A.2.9. Consultar los avistamientos de todos los usuarios

Para ver todos los avistamientos, tiene que pulsar la opción de *Avistamientos de todos los usuarios* en el menú principal. Después le llevará a la pantalla donde se mostrarán todos los avistamientos hechos por todos, ordenados de más recientes a más antiguos.

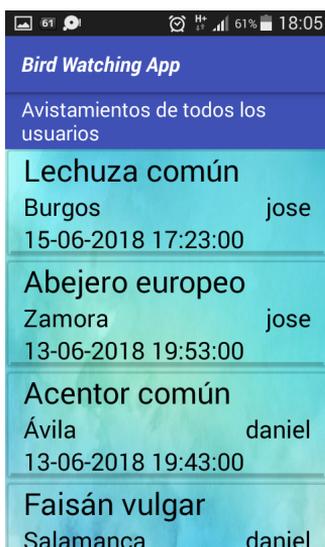


Figura A.12: Avistamientos de todos los usuarios

A.2.10. Consultar los retos

Para ver los retos de la aplicación, el usuario tiene que pulsar la opción *Retos*, en el menú desplegable de arriba a la derecha en la pantalla principal, después de acceder a la aplicación. Después se le mostrarán los retos.

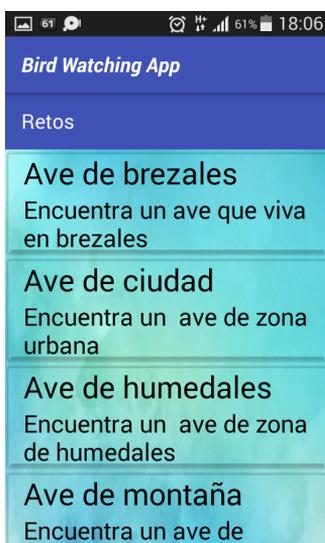


Figura A.13: Retos

A.2.11. Consultar mis logros

Para ver los logros obtenidos, el usuario tiene que pulsar la opción *Mis logros*, en el menú desplegable de arriba a la derecha en la pantalla principal, después de acceder a la aplicación. Después se le mostrarán los logros obtenidos por él, los cuales se le asignarán automáticamente después de añadir los avistamientos correspondientes.

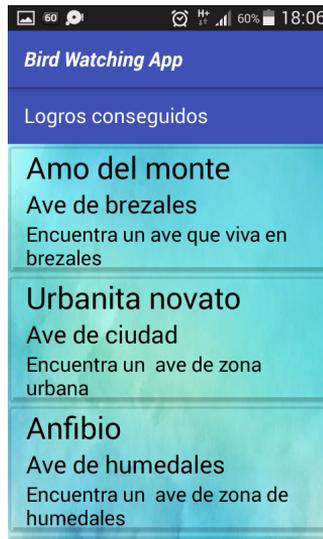


Figura A.14: Mis logros

A.2.12. Actualizar los datos del usuario

Para actualizar los datos, tiene que pulsar la opción *Ajustes*, en el menú desplegable de arriba a la derecha en la pantalla principal, después de acceder a la aplicación. Después se le mostrará la pantalla con los datos del usuario, con un botón en forma de lápiz para editar su contraseña o área.



Figura A.15: Ajustes

A.2.13. Borrar mi usuario

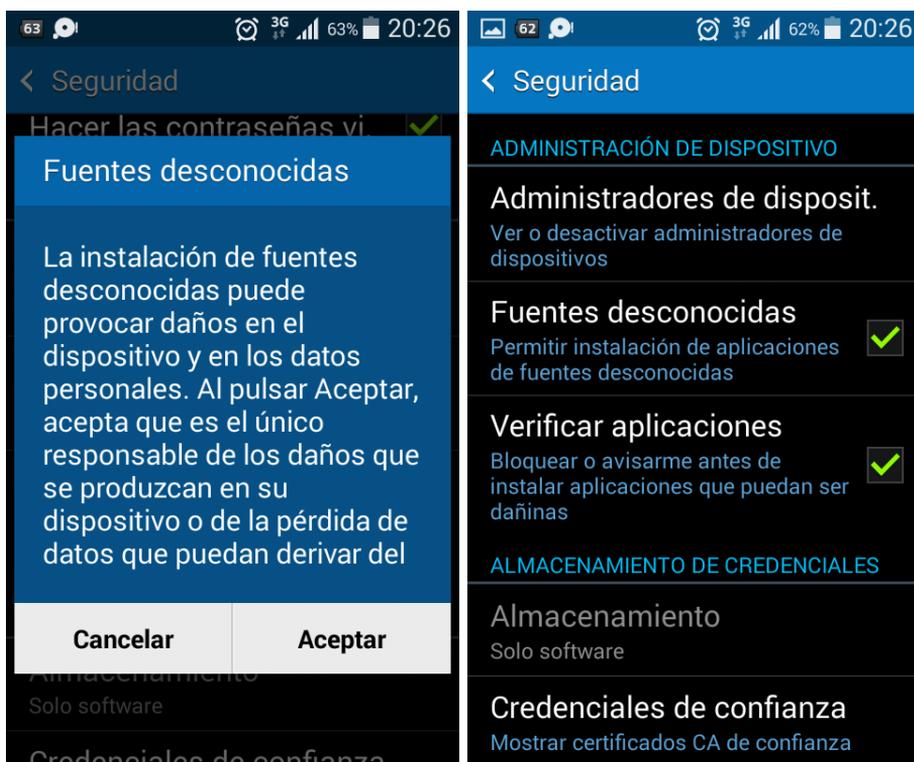
Para borrar el usuario, la opción está en la misma pantalla *Ajustes* (A.15) de antes. Solo que esta vez tiene que pulsar la opción *Borrar usuario*, y entonces se le borrará la cuenta.

Apéndice B

Manual de instalación

En este apéndice se muestra cómo se pueden instalar aplicaciones de terceros, que es el caso de nuestra aplicación.

1. Lo primero que debemos de hacer es acceder a los *ajustes* de nuestro teléfono.
2. Después, puede cambiar de un dispositivo a otro, entramos en el apartado *Seguridad*, en otros dispositivos el apartado es *Aplicaciones*.
3. Y finalmente seleccionamos la opción de *Fuentes desconocidas*, que permite la instalación de aplicaciones de fuentes desconocidas. Nos saldrá un cuadro de texto informándonos del riesgo.



(a) Diálogo de aceptación del riesgo

(b) Opción marcada

Figura B.1: Instalar aplicaciones de terceros

4. Después de esto, se procederá a la instalación de la aplicación Android en el dispositivo.

Apéndice C

Contenidos del CD-ROM

El contenido del CD adjunto a este TFG es el siguiente:

- Código fuente.
- APK (fichero de instalación) de la aplicación.
- Memoria.pdf.

Bibliografía

- [1] Project Management Institute *A guide to the project management body of knowledge (PMBOK guide)* 2000, Newtown Square, Pennsylvania.
- [2] Martin Fowler *Patterns of Enterprise Application Architecture* Addison-Wesley Professional. 2002, Boston.

Webgrafía

- [1] PMBOK Guide and Standard. [En línea] Disponible en: <https://www.pmi.org/pmbok-guide-standards>[Accedido el 26 de Marzo de 2018]
- [2] Escuela de Ingeniería Informática. [En línea] Disponible en: <https://www.inf.uva.es/>[Accedido el 5 de Febrero de 2018]
- [3] Universidad de Valladolid. [En línea] Disponible en: <https://www.uva.es/>[Accedido el 5 de Febrero de 2018]
- [4] Iberzal «Gestión de proyectos: 10 alternativas a MS Project».[En línea] Disponible en: <http://iberzal.com/gestion-de-proyectos-10-alternativas-a-ms-project/> [Accedido el 26 de Febrero de 2018]
- [5] Gantt Project «Gantt Project Free Desktop Project Management App». [En línea] Disponible en: <http://www.ganttproject.biz/> [Accedido el 26 de Febrero de 2018]
- [6] Modelo-Vista-Presentador.[En línea] Disponible en: <http://www.developps.com/es/noticias/modelo-vista-presentador-mvp-en-android>[Accedido el 26 de Febrero de 2018]
- [7] How to adopt model-view-presenter on Android [En línea] Disponible en: <https://code.tutsplus.com/tutorials/how-to-adopt-model-view-presenter-on-android-cms-26206>[Accedido el 26 de Febrero de 2018]
- [8] Hermosa Programación «Crear un Web service para Android con mysql, php y JSON».[En línea] Disponible en: <http://www.hermosaprogramacion.com/2015/05/crear-un-webservice-para-android-con-mysql-php-y-json/>[Accedido el 8 de Marzo de 2018]
- [9] Clouding, Apache vs Nginx. [En línea] Disponibe en: <https://clouding.io/blog/apache-nginx/> [Accedido el 5 de Marzo de 2018]
- [10] Digital Ocean. Cómo instalar nginx en Ubuntu 16.04.[En línea] Disponible en: <https://www.digitalocean.com/community/tutorials/como-instalar-nginx-en-ubuntu-16-04-es> [Accedido el 6 de Marzo de 2018]
- [11] Digitl Ocean. Cómo configurar Nginx como Web Server y Proxy server. [En línea] Disponible en: <https://www.digitalocean.com/community/tutorials/how-to-configure-nginx-as-a-web-server-and-reverse-proxy-for-apache-on-one-ubuntu-16-04-server>[Accedido el 23 de Abril de 2018]

- [12] How to install Apache Tomcat 8 on Ubuntu [En línea] Disponible en: <https://www.digitalocean.com/community/tutorials/how-to-install-apache-tomcat-8-on-ubuntu-16-04>[Accedido el 8 de Marzo de 2018]
- [13] Apache Tomcat 8.0.50 [En línea] Disponible en: <https://tomcat.apache.org/download-80.cgi#8.0.50>[Accedido el 8 de Marzo de 2018]
- [14] Nginx as a reverse proxy to Apache Tomcat [En línea] Disponible en: <https://devtidbits.com/2015/12/08/nginx-as-a-reverse-proxy-to-apache-tomcat/>[Accedido el 9 de Marzo de 2018]
- [15] Ciclo de vida de una actividad en Android.[En línea] Disponible en: <http://www.terminalesandroid.com/Ciclo-de-vida-de-una-actividad-en-android>[Accedido el 9 de Abril de 2018]
- [16] Activity y Fragment. [En línea] Disponible en: <https://academiaandroid.com/activity-y-fragments/>[Accedido el 9 de Abril de 2018]
- [17] Installing Nodejs. [En línea] Disponible en: <https://nodejs.org/en/download/package-manager/>[Accedido el 20 de Abril de 2018]
- [18] Object-Relational Mapping Sequelizejs. [En línea] Disponible en: <http://docs.sequelizejs.com/>[Accedido el 1 de Mayo de 2018]
- [19] Como instalar Postman en Ubuntu. [En línea] Disponible en: <https://lernilo.com/post/postman-instalar-ubuntu/>[Accedido el 3 de Mayo de 2018]
- [20] Creating Web API Using Node.js And MySQL . [En línea] Disponible en: <https://www.c-sharpcorner.com/article/creating-web-api-using-node-js-and-mysql/>[Accedido el 3 de Mayo de 2018]
- [21] Swagger Editor. [En línea] Disponible en: <https://editor.swagger.io/>[Accedido el 4 de Mayo de 2018]
- [22] Github/antoniolg- MVP Android Example. [En línea] Disponible en: <https://github.com/antoniolg/androidmvp>[Accedido el 5 de Mayo de 2018]
- [23] Model-View-Presenter In Android Part1 . [En línea] Disponible en: <http://www.tinmegali.com/en/2016/03/04/model-view-presenter-android-part-1/>[Accedido el 5 de Mayo de 2018]
- [24] Pixabay- Imágenes gratis . [En línea] Disponible en: <https://pixabay.com/>[Accedido el 10 de Mayo de 2018]
- [25] Create a List with RecyclerView . [En línea] Disponible en: <https://developer.android.com/guide/topics/ui/layout/recyclerviewjava> [Accedido el 12 de Mayo de 2018]
- [26] Getting started with RecyclerView and CardView . [En línea] Disponible en: <https://code.tutsplus.com/tutorials/getting-started-with-recyclerview-and-cardview-on-android-cms-23465>[Accedido el 12 de Mayo de 2018]

- [27] Android RecyclerView Example. [En línea] Disponible en: <http://stacktips.com/tutorials/android/android-recyclerview-example> [Accedido el 13 de Mayo de 2018]
- [28] AsyncTask - Android developers. [En línea] Disponible en: <https://developer.android.com/reference/android/os/AsyncTask> [Accedido el 13 de Mayo de 2018]
- [29] Android spinner example. [En línea] Disponible en: <https://www.mkymong.com/android/android-spinner-drop-down-list-example/> [Accedido el 15 de Mayo de 2018]
- [30] Parsing JSON Object in java. [En línea] Disponible en: <https://stackoverflow.com/questions/5015844/parsing-json-object-in-java> [Accedido el 17 de Mayo de 2018]
- [31] Tabla de colores HTML. [En línea] Disponible en: <https://html-color-codes.info/codigos-de-colores-hexadecimales/> [Accedido el 19 de Mayo de 2018]
- [32] Android examples - RecyclerViews. [En línea] Disponible en: <https://www.android-examples.com/?s=recyclerview> [Accedido el 19 de Mayo de 2018]
- [33] How to use putExtra() and getExtra(). [En línea] Disponible en: <https://stackoverflow.com/questions/5265913/how-to-use-putextra-and-getextra-for-string-data> [Accedido el 19 de Mayo de 2018]
- [34] How display an alertDialog. [En línea] Disponible en: <https://stackoverflow.com/questions/2115758/how-do-i-display-an-alert-dialog-on-android> [Accedido el 20 de Mayo de 2018]
- [35] Tell Git not to track a file any more. [En línea] Disponible en: <https://alvinalexander.com/git/git-rm-file-git-repository-dont-track-accidentally-added> [Accedido el de Mayo de 2018]
- [36] How to add Action bar options in menu. [En línea] Disponible en: <https://stackoverflow.com/questions/18714322/how-to-add-action-bar-options-menu-in-android-fragments> [Accedido el 21 de Mayo de 2018]
- [37] SEO Birdlife. [En línea] Disponible en: <https://www.seo.org/> [Accedido el 22 de Mayo de 2018]
- [38] Table in Android. [En línea] Disponible en: <https://developer.android.com/guide/topics/ui/layout/grid> [Accedido el 25 de Mayo de 2018]
- [39] Is there an easy way to strike through text in a app widget?. [En línea] Disponible en: <https://stackoverflow.com/questions/3881553/is-there-an-easy-way-to-strike-through-text-in-an-app-widget> [Accedido el 25 de Mayo de 2018]
- [40] App bar. [En línea] Disponible en: <https://developer.android.com/training/appbar/setting-up> [Accedido el 26 de Mayo de 2018]

- [41] Using the App Toolbar. [En línea] Disponible en: <https://guides.codepath.com/android/Using-the-App-ToolBar> [Accedido el 26 de Mayo de 2018]
- [42] How to convert a String to an Int in Java. [En línea] Disponible en: <https://stackoverflow.com/questions/5585779/how-do-i-convert-a-string-to-an-int-in-java> [Accedido el 28 de Mayo de 2018]
- [43] Picasso - Image Downloading library. [En línea] Disponible en: <http://square.github.io/picasso/> [Accedido el 4 de Junio de 2018]
- [44] Pixabay - Texture. [En línea] Disponible en: <https://pixabay.com/es/textura-de-fondo-suave-azul-luz-1668079/> [Accedido el 4 de Junio de 2018]
- [45] Java: unparseable date exception. [En línea] Disponible en: <https://stackoverflow.com/questions/2009207/java-unparseable-date-exception> [Accedido el 4 de Junio de 2018]
- [46] Express - servicio de archivos estáticos. [En línea] Disponible en: <http://expressjs.com/es/starter/static-files.html> [Accedido el 5 de Junio de 2018]
- [47] Android - Two imagesViews side by side. [En línea] Disponible en: <https://stackoverflow.com/questions/6220354/android-two-imageviews-side-by-side> [Accedido el 5 de Junio de 2018]
- [48] Gradle sync failed couldn't find constraint-layout:1.0.1. [En línea] Disponible en: <https://stackoverflow.com/questions/37992187/gradle-sync-failed-could-not-find-constraint-layout1-0-0-alpha2> [Accedido el 6 de Junio de 2018]
- [49] How to revert a git repository to a previous commit?. [En línea] Disponible en: <https://stackoverflow.com/questions/4114095/how-to-revert-git-repository-to-a-previous-commit> [Accedido el 6 de Junio de 2018]
- [50] Caracteres reservados en LaTeX. [En línea] Disponible en: <http://tiburondealambre.blogspot.com/2012/03/caracteres-reservados-de-latex.html> [Accedido el 10 de Junio de 2018]
- [51] Tablas en LaTeX. [En línea] Disponible en: <https://latexlive.files.wordpress.com/2009/04/tablas.pdf> [Accedido el 12 de Junio de 2018]
- [52] UTC DATE in sequelizejs. [En línea] Disponible en: <https://stackoverflow.com/questions/17976459/utc-date-in-sequelizejs> [Accedido el 14 de Junio de 2018]
- [53] Java, sumar o restar horas. [En línea] Disponible en: <http://desarrollando.com/blog/java-sumar-restar-horas-dias-fecha/> [Accedido el 14 de Junio de 2018]
- [54] Latex, Múltiples imágenes juntas. [En línea] Disponible en: <http://www.elmundoenbits.com/2012/05/latex-multiples-imagenes-juntas.html> [Accedido el 16 de Junio de 2018]