



**Universidad de Valladolid**

**ESCUELA DE INGENIERÍA INFORMÁTICA (SEGOVIA)**

**Grado en Ingeniería Informática de Servicios y Aplicaciones**

**MyClothes: aplicación Android para la gestión  
inteligente de la ropa de tu armario**

***Alumna: Virginia Jimeno Calle***

***Tutores: Aníbal Bregón Bregón***

***Miguel Ángel Martínez Prieto***



# Resumen

---

Este proyecto consiste en el desarrollo de una aplicación para dispositivos móviles, con sistema Android, para el control y gestión del vestuario particular de cada usuario, en el que, se recopilan las imágenes de la ropa y el calzado, pudiendo ser clasificadas por varios métodos.

La aplicación permite la visualización simultánea de estas prendas, lo cual facilita la creación de conjuntos, los cuales se pueden almacenar en un calendario, de la propia aplicación, asignándolos a una fecha determinada, lo cual evitaría en gran manera la repetición de conjuntos en días específicos. También se facilita la sugerencia de un conjunto determinado para una fecha en concreto, en base a anteriores elecciones del usuario y la climatología actual.

Con esto se pretende tener registrada toda la ropa y calzado del usuario, y verlo de una forma rápida y sencilla, en cualquier lugar donde se encuentre, a través de su dispositivo móvil. Por lo que sería muy útil también a la hora de salir de compras, para ver lo que se tiene y lo que se puede necesitar.

Por lo que se piensa que esta aplicación puede ser muy interesante, debido al auge de la moda y de los centros comerciales en la actualidad, así como de la venta de dispositivos móviles, de los que casi nadie hoy en día carece.



# Índice de contenido

---

Capítulo 1. Planteamiento inicial .....	13
1.1. Introducción .....	15
1.2. Motivación .....	16
1.3. Objetivos y alcance del sistema .....	17
1.3.1. Alcance.....	17
1.3.2. Objetivos .....	17
1.3.3. Reglas de negocio .....	17
1.4. ¿Por qué Android? .....	18
1.4.1. Razones para su elección .....	18
1.4.2. Ventajas.....	20
1.4.3. Inconvenientes .....	21
1.5. Estado del arte .....	21
Capítulo 2. Planificación y presupuesto.....	27
2.1. Planificación.....	29
2.2. Presupuesto .....	32
2.2.1. Estimación de costes .....	32
2.2.2. Estimación mediante Puntos de Función .....	34
2.2.3. Estimación mediante COCOMO .....	36
2.2.4. Estimación mediante Casos de Uso .....	37
2.3. Costes finales .....	41
Capítulo 3. Análisis.....	45
3.1. Características principales.....	47
3.1.1. Árbol de características .....	47
3.2. Identificación de los actores del sistema.....	48
3.3. Requisitos de usuario .....	48
3.3.1. Relación de los Casos de Uso con actores .....	48
3.3.2. Listado de Casos de Uso .....	49
3.3.3. Diagrama de Casos de Uso .....	49
3.3.4. Especificación de los Casos de Uso .....	51
3.4. Requisitos funcionales .....	68
3.5. Requisitos de información .....	68

3.5.1. Modelo conceptual .....	69
3.5.2. Diccionario de datos.....	70
3.6. Requisitos de interfaces externas .....	71
3.7. Requisitos no funcionales .....	71
3.8. Requisitos de internacionalización y localización .....	72
Capítulo 4. Diseño.....	73
4.1. Tecnologías .....	75
4.2. Arquitectura lógica.....	75
4.3. Arquitectura física.....	77
4.4. Clases de diseño .....	77
4.5. Diagramas de secuencia .....	78
4.6. Factores que determinan la recomendación de conjuntos.....	85
4.7. Diseño de la interfaz .....	88
Capítulo 5. Implementación y pruebas .....	93
5.1. Introducción a Android .....	95
5.1.1. Componentes de una aplicación Android .....	96
5.1.2. Estructura de un proyecto Android .....	97
5.2. Estructura del proyecto .....	98
5.3. Aspectos relevantes de implementación .....	100
5.4. Descripción de las clases y los métodos .....	101
5.5. Pruebas .....	114
Capítulo 6. Manuales .....	123
6.1. Manual de instalación .....	125
6.2. Manual de usuario.....	128
6.2.1. Gestión de prendas .....	128
6.2.2. Gestión de conjuntos.....	133
6.2.3. Gestión del calendario.....	137
6.2.4. Gestión de las recomendaciones .....	138
Capítulo 7. Futuras mejoras y conclusiones.....	141
7.1. Mejoras futuras .....	143
7.2. Conclusiones personales .....	144
Capítulo 8. Referencias .....	145
8.1. Bibliografía .....	146

8.2 Referencias web .....	146
ANEXO I. Contenido del CD-ROM.....	147
ANEXO II. Diagrama de clases .....	151



# Índice de tablas

---

Tabla 1. Versiones Android. ....	20
Tabla 2. Estimación mano de obra. ....	32
Tabla 3. Estimación del coste del hardware. ....	33
Tabla 4. Estimación del coste del software. ....	33
Tabla 5. Estimación del coste total. ....	34
Tabla 6. Grados de complejidad de PF. ....	34
Tabla 7. Valores de complejidad de PF. ....	35
Tabla 8. Factores de complejidad. ....	35
Tabla 9. Tipos de COCOMO. ....	36
Tabla 10. Conductores de coste de COCOMO. ....	36
Tabla 11. Interacción método de Casos de Uso. ....	37
Tabla 12. Transacciones método de Casos de Uso. ....	38
Tabla 13. Complejidades método de Casos de Uso. ....	38
Tabla 14. Factores Técnicos método de Casos de Uso. ....	39
Tabla 15. Factores de Entorno método de Casos de Uso. ....	40
Tabla 16. Coste real de la mano de obra. ....	41
Tabla 17. Coste real del hardware. ....	42
Tabla 18. Coste real del software. ....	42
Tabla 19. Coste real total. ....	43
Tabla 20. Especificación CU-01. ....	51
Tabla 21. Especificación CU-02. ....	52
Tabla 22. Especificación CU-03. ....	53
Tabla 23. Especificación CU-04. ....	54
Tabla 24. Especificación CU-05. ....	55
Tabla 25. Especificación CU-06. ....	56
Tabla 26. Especificación CU-07. ....	57
Tabla 27. Especificación CU-08. ....	58
Tabla 28. Especificación CU-09. ....	59
Tabla 29. Especificación CU-10. ....	60
Tabla 30. Especificación CU-11. ....	61
Tabla 31. Especificación CU-12. ....	62
Tabla 32. Especificación CU-13. ....	63
Tabla 33. Especificación CU-14. ....	64
Tabla 34. Especificación CU-15. ....	65
Tabla 35. Especificación CU-16. ....	66
Tabla 36. Especificación CU-17. ....	67
Tabla 37. Diccionario de datos. ....	70
Tabla 38. Ponderaciones de Estación. ....	85
Tabla 39. Ponderaciones de Estilo. ....	86
Tabla 40. Ponderaciones de Clima. ....	87
Tabla 41. Diseño de la interfaz inicio.xml. ....	88

Tabla 42. Diseño de la interfaz armario.xml.....	89
Tabla 43. Diseño de la interfaz probador.xml.....	90
Tabla 44. Diseño de la interfaz conjuntos.xml.....	91
Tabla 45. Caso de prueba CU-01.....	115
Tabla 46. Caso de prueba CU-06.....	116
Tabla 47. Caso de prueba CU-07.....	117
Tabla 48. Caso de prueba CU-08.....	117
Tabla 49. Caso de prueba CU-09.....	118
Tabla 50. Caso de prueba CU-12.....	119
Tabla 51. Caso de prueba CU-13.....	119
Tabla 52. Caso de prueba CU-14.....	120
Tabla 53. Caso de prueba CU-15.....	121
Tabla 54. Caso de prueba CU-16.....	121
Tabla 55. Caso de prueba CU-17.....	122

# Índice de ilustraciones

---

Ilustración 1. Sistemas operativos móviles más utilizados. ....	19
Ilustración 2. Versiones Android más utilizadas. ....	19
Ilustración 3. Pantalla principal de la aplicación MiRopa 1.0. ....	22
Ilustración 4. Otras pantallas de la aplicación MiRopa 1.0. ....	22
Ilustración 5. Pantallas de la aplicación MyDressing. ....	23
Ilustración 6. Pantallas de la aplicación Stylicious. ....	24
Ilustración 7. Pantallas de la aplicación Personal Closet. ....	25
Ilustración 8. Pantallas de la aplicación Mix ME. ....	26
Ilustración 9. Planificación de las iteraciones con el Método Incremental. ....	30
Ilustración 10. Gráfico de la planificación de las iteraciones con el Método Incremental. ....	31
Ilustración 11. Diagrama del árbol de características. ....	48
Ilustración 12. Diagrama de Casos de Uso. ....	50
Ilustración 13. Diagrama de Entidad-Relación. ....	69
Ilustración 14. Diagrama de la arquitectura lógica. ....	76
Ilustración 15. Diagrama de la arquitectura física. ....	77
Ilustración 16. Diagrama de clases de diseño. ....	78
Ilustración 17. Diagrama de secuencia del CU-01. ....	79
Ilustración 18. Diagrama de secuencia del CU-04. ....	80
Ilustración 19. Diagrama de secuencia del CU-05. ....	80
Ilustración 20. Diagrama de secuencia del CU-06. ....	81
Ilustración 21. Diagrama de secuencia del CU-07. ....	81
Ilustración 22. Diagrama de secuencia del CU-08. ....	82
Ilustración 23. Diagrama de secuencia del CU-09. ....	82
Ilustración 24. Diagrama de secuencia del CU-14. ....	83
Ilustración 25. Diagrama de secuencia del CU-15. ....	83
Ilustración 26. Diagrama de secuencia del CU-16. ....	84
Ilustración 27. Diagrama de secuencia del CU-17. ....	84
Ilustración 28. Arquitectura de Android. ....	95
Ilustración 29. Antes y después del probador. ....	100
Ilustración 30. Explicación de la instalación en un dispositivo móvil. ....	125
Ilustración 31. Explicación de la instalación a través del proyecto 1. ....	126
Ilustración 32. Explicación de la instalación a través del proyecto 2. ....	126
Ilustración 33. Explicación de la instalación a través del proyecto 3. ....	127
Ilustración 34. Explicación de la instalación a través del proyecto 4. ....	127
Ilustración 35. Manual de usuario: Agregar prenda 1. ....	128
Ilustración 36. Manual de usuario: Agregar prenda 2. ....	129
Ilustración 37. Manual de usuario: Agregar prenda 3. ....	129
Ilustración 38. Manual de usuario: Agregar prenda 4. ....	130
Ilustración 39. Manual de usuario: Agregar prenda 5. ....	130
Ilustración 40. Manual de usuario: Visualizar prenda 1. ....	131
Ilustración 41. Manual de usuario: Visualizar prenda 2. ....	132

Ilustración 42. Manual de usuario: Visualizar prenda 3. ....	132
Ilustración 43. Manual de usuario: Modificar prenda. ....	133
Ilustración 44. Manual de usuario: Probar conjunto. ....	134
Ilustración 45. Manual de usuario: Crear conjunto. ....	134
Ilustración 46. Manual de usuario: Visualizar categorías conjuntos. ....	135
Ilustración 47. Manual de usuario: Visualizar listado de conjuntos. ....	136
Ilustración 48. Manual de usuario: Visualizar conjunto. ....	136
Ilustración 49. Manual de usuario: Modificar conjunto. ....	137
Ilustración 50. Manual de usuario: Calendario. ....	137
Ilustración 51. Manual de usuario: Asignar conjunto en el calendario. ....	138
Ilustración 52. Manual de usuario: Pedir recomendación. ....	139





# Capítulo 1

## Planteamiento inicial

---



## 1.1. Introducción

Las tecnologías de la información y de la comunicación han venido haciéndose imprescindibles en la vida cotidiana de las personas en los últimos años. Hasta hace poco, las personas vivían la mayor parte de su vida ajenas a todas las facilidades que aportan estas tecnologías. En la actualidad, cualquiera puede sacarles provecho y, por lo tanto, pasar el día a día de forma más cómoda y sencilla, aportando un gran valor, e incluso su uso también da lugar al entretenimiento.

El elemento más representativo de las nuevas tecnologías es sin duda Internet, ya que supone un salto cualitativo de gran magnitud, cambiando y redefiniendo los modos de conocer y relacionarse del ser humano. Gracias a ello y junto con la digitalización de contenidos, se ha evolucionado pasando de páginas web estáticas, en su gran mayoría compuestas sólo de texto que aportaban poco valor añadido, a páginas web cada vez mejor construidas, dinámicas y eficientes, que daban lugar a una gran interactividad usuario-máquina. Aun así, esto impedía su uso frecuente, porque para estar conectado era necesario disponer de un ordenador de sobremesa, del cual no se podía hacer uso fuera de su lugar de instalación. Por eso, cuando se produjo el salto a dispositivos móviles, de cada vez menor tamaño, y con acceso a cualquier tipo de información, estos se empezaron a utilizar de manera más frecuente y habitual. Y esto fue debido a la comodidad que ofrecen en su desplazamiento, ya que se pueden llevar a cualquier punto geográfico, y utilizarlos en cualquier momento y lugar. También gracias a su bajo coste, y a que están extraordinariamente extendidos en la sociedad. Por todo ello, se ha derivado en una gran evolución en este tipo de tecnologías en muy poco tiempo<sup>1</sup>, y de hecho sigue creciendo hasta niveles impensables.

El continuo uso que se hace actualmente de los móviles, permite que cada vez se cree un mayor número de aplicaciones de lo más variadas, entre las que se pueden encontrar desde aplicaciones destinadas únicamente al entretenimiento hasta aplicaciones que pueden hacer el día a día más sencillo, porque son capaces de aportar casi infinitas posibilidades. A pesar del gran volumen de aplicaciones que existen actualmente, siguen surgiendo otras que se pueden convertir en un “amigo inseparable”. Todo el mundo ha pensado alguna vez que sería genial la existencia de una aplicación para hacer algo concreto. Pero la realidad es que el desarrollo de estas aplicaciones resulta bastante complicado, debido a que son muchos los factores a tener en cuenta, y que deben ser escrupulosamente razonados. Por ejemplo, existen muchas aplicaciones cuya idea es buena, pero no se plasman en la utilidad práctica a la que pudieran dar lugar.

Aunque, en principio, se puede prescindir sin mucha dificultad de los dispositivos móviles y sus aplicaciones, es indudable el partido que se les puede sacar, ya que cada vez más, se lleva una forma de vida a gran velocidad, en la que la gente trabaja y está constantemente ocupada, y por eso, es necesaria una mayor organización. Por lo tanto, aquella tecnología que sea capaz de ahorrar tiempo en la vida cotidiana de las personas, es la clave del progreso.

---

<sup>1</sup> <http://www.economiadigital.es/es/notices/2015/04/-como-ha-evolucionado-el-telefono-movil-tras-42-anos-de-vida-68911.php>

Debido a todo esto, porque es el presente y también el futuro, se pretende contribuir realizando la aplicación móvil objeto de este Trabajo Final de Grado.

## 1.2. Motivación

El mundo de la moda es un sector que se encuentra en continua expansión, nunca decae, y se está convirtiendo, cada vez más, en uno de los negocios más rentables. Esto ha venido propiciado por la constante apertura de tiendas y centros comerciales ofertando artículos a precios cada vez más económicos y por la venta online que permite realizar compras de la forma más rápida y sencilla, sin moverse de casa, lo que permite a la gente tener mucha más variedad de ropa. Además el surgimiento de “bloggers” o personas que se dedican a hacerse fotos y publicar en las redes sociales las prendas de ropa que llevan puestas, les hace incluso poder vivir de ello, ya que resultan campañas de marketing muy rentables para las marcas. Es, precisamente, porque son capaces de mover masas, por lo que cada vez más gente se preocupa por su forma de vestir.

La contrapartida de tener cada vez más ropa, hace que las personas no sean conscientes de lo que tienen y no consigan verlo de una forma rápida y organizada, y mucho menos en cualquier lugar en el que se encuentren, como fuera de su domicilio, lo que puede derivar en la compra de artículos que realmente no se necesitan, o incluso, ya se poseen. Entonces, sería muy interesante poder controlar aquella ropa que se tiene, para así evitar compras compulsivas o repetitivas, olvidos de prendas que no se recuerda que se tienen, realizar posibles combinaciones de las prendas de una forma rápida y clara, y llevar un registro de lo utilizado en un acontecimiento determinado, para evitar su repetición.

Debido a esto, se llega a la conclusión de que no hay nada mejor que juntar sectores tan importantes como el sector de la moda y el sector tecnológico, ya que pueden formar la combinación perfecta para lograr hacer un adecuado uso de la ropa que se tiene a través de las nuevas tecnologías.

Esta posibilidad se encuentra en la aplicación que se presenta, la cual está dirigida a cualquier persona que disponga de un dispositivo móvil inteligente y quiera organizar la ropa de su armario y administrarla de una manera eficaz.

## 1.3. Objetivos y alcance del sistema

### 1.3.1. Alcance

Se podrá acceder a esta aplicación a través de un dispositivo móvil Android en el rango de versiones desde la 4.0 (Ice Cream Sandwich) a la 4.4 (KitKat). Además se puede tener completa disponibilidad de la aplicación sin la necesidad de estar conectado a Internet.

### 1.3.2. Objetivos

- El **primer objetivo** es que se permita realizar un inventario de las prendas de ropa del usuario, junto con sus características más relevantes, y queden registradas en el dispositivo móvil.
- El **segundo objetivo** que se pretende es poder combinar unas prendas con otras con el fin de formar conjuntos.
- El **tercer objetivo** es que tanto las prendas como los conjuntos puedan estar clasificados en categorías para tenerlos organizados.

Además sería muy interesante tener disponible un calendario al que se le pueda añadir conjuntos en determinadas fechas, y que se muestre a simple vista.

Por último, se establece como objetivo que la aplicación ofrezca recomendaciones personalizadas de conjuntos al usuario, en base a anteriores conjuntos elegidos por él.

### 1.3.3. Reglas de negocio

Una vez establecidos el alcance y los objetivos a conseguir, se procede a describir unas reglas básicas a tener en cuenta que definan su ámbito de aplicación.

- **RN-01:** Se pueden crear cuatro tipos de prendas, parte de arriba, parte de abajo, una sola pieza (como pueden ser vestidos y monos), y el calzado.
- **RN-02:** Para formar un conjunto es necesario tener al menos dos prendas (una prenda de una sola pieza y calzado, o de otro modo, parte de arriba, parte de abajo y calzado).

- **RN-03:** En una determinada fecha del calendario solo se puede asignar un conjunto.
- **RN-04:** La recomendación de conjunto al usuario se hará en base a la estación, ocasión del conjunto y el clima.

## 1.4. ¿Por qué Android?

Android es un sistema operativo basado en el núcleo de Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tablets; y también para relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por Android Inc., empresa que en 2005 fue comprada por Google. Aunque no fue hasta 2008 cuando se popularizó, gracias a la unión al proyecto de Open Handset Alliance, un consorcio formado por 48 empresas de desarrollo de hardware, software y telecomunicaciones, que decidieron promocionar el software libre, pero ha sido Google quien ha publicado la mayor parte del código fuente del sistema operativo.

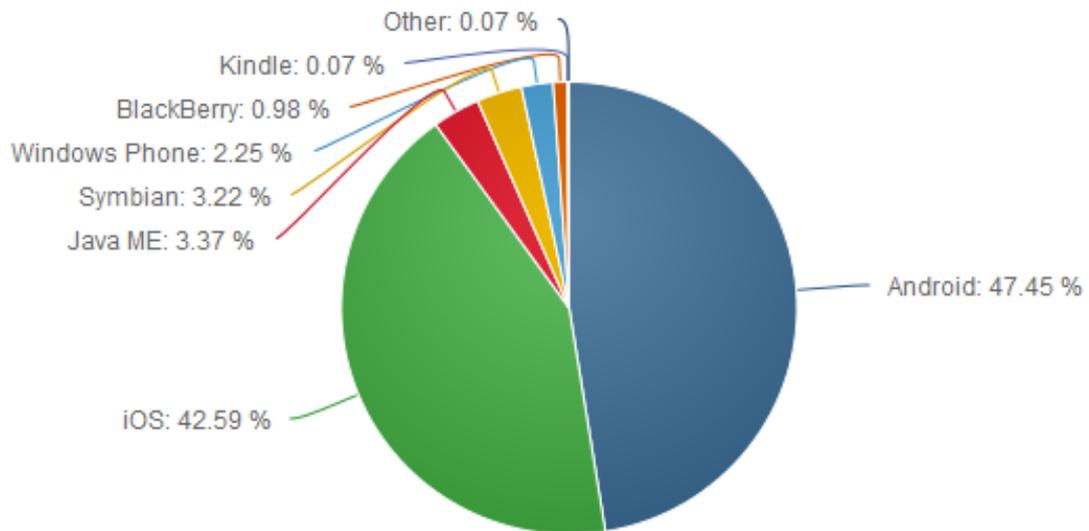
Tiene una gran comunidad de desarrolladores que crean aplicaciones para extender la funcionalidad de los dispositivos. Se ha superado el millón de aplicaciones disponibles para la tienda de aplicaciones oficial de Android: Google Play<sup>2</sup>, sin tener en cuenta aplicaciones de otras tiendas no oficiales para Android como la tienda de aplicaciones Samsung Apps de Samsung, Slideme de Java y Amazon Appstore. Google Play es la tienda de aplicaciones en línea administrada por Google, aunque existe la posibilidad de obtener software externamente.

### 1.4.1. Razones para su elección

En definitiva, ¿por qué elegir Android para el desarrollo de la aplicación? Principalmente porque es el sistema operativo móvil más usado actualmente. Se muestra un gráfico actual con el porcentaje de sistemas más usados:

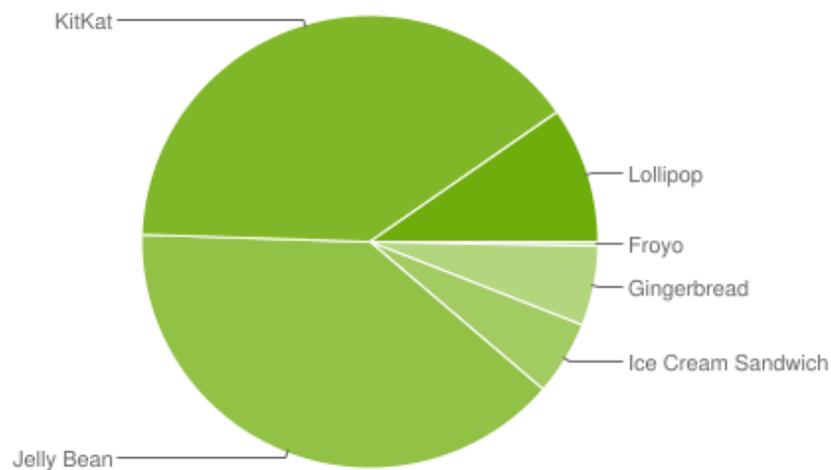
---

<sup>2</sup> <http://www.elandroidelibre.com/2015/01/google-play-supera-la-appstore-en-cantidad-de-aplicaciones-y-desarrolladores.html>



**Ilustración 1. Sistemas operativos móviles más utilizados.**

Además el rango de versiones elegido para que esté disponible la aplicación es usado por más del 80% de dispositivos Android actuales.



**Ilustración 2. Versiones Android más utilizadas.**

La Ilustración 1 y la Ilustración 2 son imágenes sacadas de la página oficial de Android. Esta página actualiza las estadísticas periódicamente.

Versión	Nombre	API
2.2	Froyo	8
2.3.3- 2.3.7	Gingerbread	10
4.0.3- 4.0.4	Ice Cream Sandwich	15
4.1.x	Jelly Bean	16
4.2.x		17
4.3		18
4.4	KitKat	19
5.0	Lollipop	21
5.1		22

**Tabla 1. Versiones Android.**

## 1.4.2. Ventajas

1. El código de Android es **abierto**: Google liberó Android bajo licencia Apache.
2. Existe una gran cantidad de aplicaciones disponibles, dos tercios de las cuales son gratuitas. Además la libertad de código permite adaptar Android a bastantes otros dispositivos además de teléfonos móviles. Está implantado en tablets, GPS, relojes, microondas, etc.
3. Es muy personalizable, permite modificar la apariencia del sistema operativo, añadir widgets, etc.
4. El sistema Android es capaz de hacer funcionar a la vez varias aplicaciones y además se encarga de gestionarlas, dejarlas en modo suspensión si no se utilizan e incluso cerrarlas si llevan un periodo determinado de inactividad.
5. Libertad para instalar aplicaciones. Permite instalar aplicaciones aunque sean de origen desconocido, y no las restringe tanto como otros sistemas operativos.

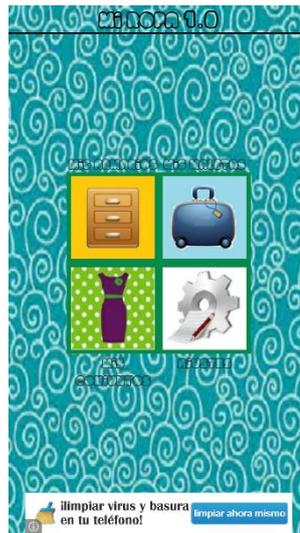
### 1.4.3. Inconvenientes

1. A pesar de ser una ventaja el ser un sistema multitarea, el hecho de tener varias aplicaciones abiertas hacen que el consumo de la batería aumente y por otro lado Android no siempre cierra todas las aplicaciones así que hace falta tener una aplicación que cierre las aplicaciones abiertas. En el Market de Android hay una buena cantidad de aplicaciones para este fin, así que el problema es solucionable.
2. La batería en un dispositivo móvil Android se agota muy rápido. Utilizando las aplicaciones de manera moderada ésta puede llegar a durar más, pero para un usuario que usa mucho sus aplicaciones, la batería se consume en exceso, lo que se puede solucionar con algunas aplicaciones para optimizarla.
3. La necesidad de instalar aplicaciones externas para solucionar problemas de uso normal.
4. Android se encuentra muy fragmentado, lo que provoca problemas de incompatibilidad con algunas aplicaciones del Market que funcionan en determinadas versiones de Android.
5. Resulta poco intuitivo a la hora de configurar y personalizar el dispositivo.
6. La libertad para instalar cualquier tipo de aplicación puede derivar en problemas de seguridad en el dispositivo.

## 1.5. Estado del arte

En los dos últimos años ha surgido un gran número de aplicaciones comparables a la desarrollada en este proyecto que, sin embargo, ya sea por su funcionalidad o por su estética, no cumplen las expectativas. Las siguientes apps que se muestran se pueden conseguir en el Android Market.

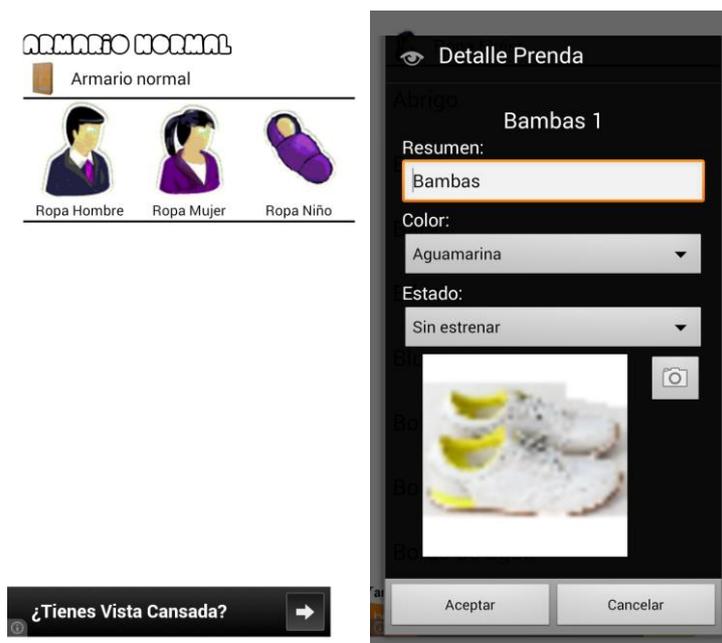
# MiRopa 1.0



**Ilustración 3. Pantalla principal de la aplicación MiRopa 1.0.**

Se trata de una aplicación que almacena prendas y conjuntos. Además se pueden crear maletas de ropa. Está enfocada a ser una aplicación para varias personas, como por ejemplo, los miembros de una familia. El diseño de la interfaz no es el más adecuado, resulta una aplicación poco intuitiva, y además contiene publicidad molesta. Puede ser complicada de usar para un usuario medio.

En la Ilustración 3, se muestra la pantalla principal de la aplicación. En la primera imagen de la Ilustración 4, se muestra la elección del tipo de armario a crear, y la segunda imagen es la creación de una nueva prenda.

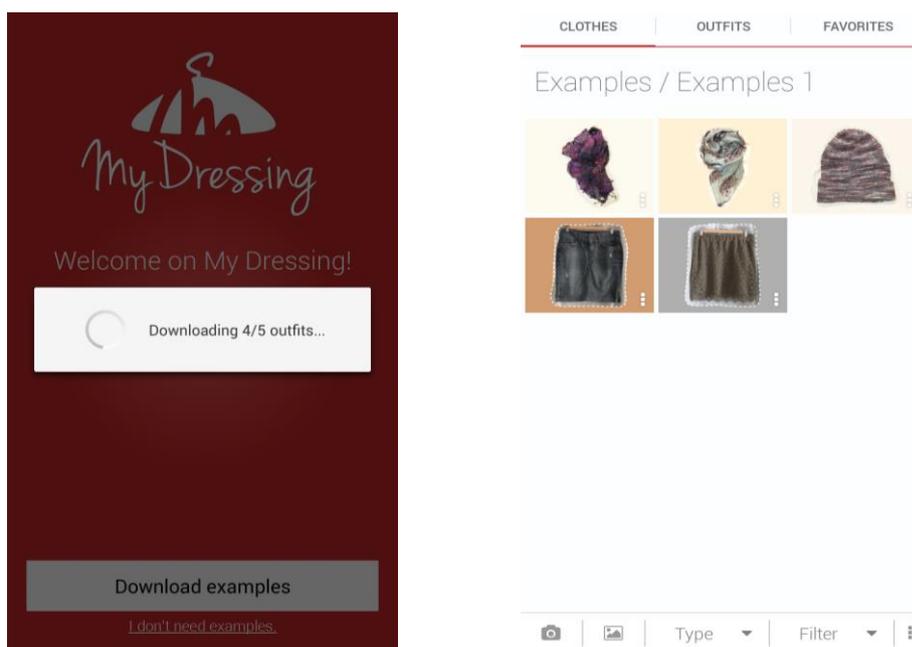


**Ilustración 4. Otras pantallas de la aplicación MiRopa 1.0.**

## MyDressing

Se trata de una aplicación que permite compartir las imágenes de prendas y conjuntos. No está basada en su gestión ni organización, se enfoca más en la creación de imágenes de las prendas y de los conjuntos. Además, los datos relacionados con ellos no son muy relevantes. Es una aplicación que emplea bastante tiempo en iniciarse y está únicamente en Inglés. La parte positiva es que permite recortar las prendas rápidamente sin una forma geométrica definida.

En la Ilustración 5, se muestra el cargado inicial de la aplicación, y una vista general de las prendas.



**Ilustración 5. Pantallas de la aplicación MyDressing.**

## Stylicious

Se trata de una aplicación que ofrece ropa de diferentes marcas. También se puede compartir información de conjuntos con el objetivo de inspirar a otros usuarios y vender ropa. Emplea mucho tiempo para el cargado de las prendas. Sin embargo, resulta intuitiva y sencilla de usar.

En la Ilustración 6, se muestra una imagen de la ayuda que se muestra la primera vez que se usa la aplicación, y la pantalla principal.

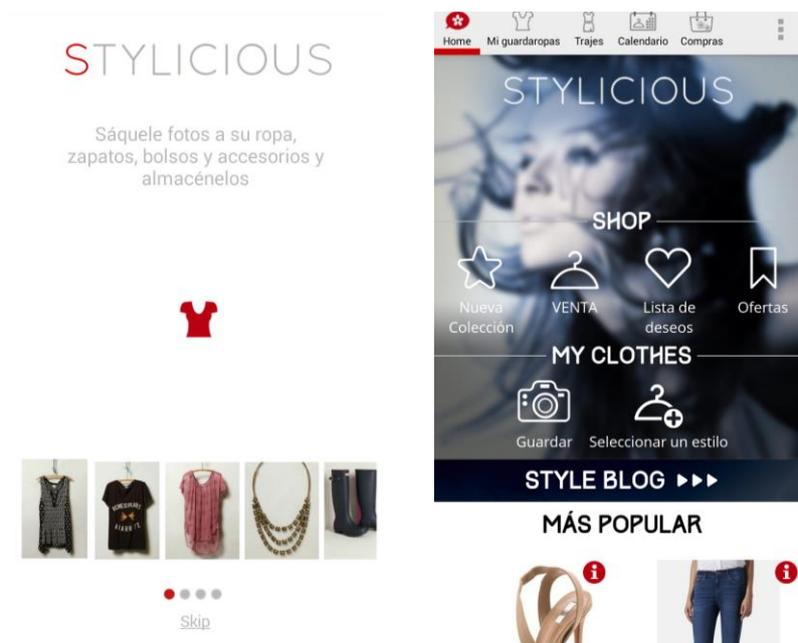


Ilustración 6. Pantallas de la aplicación Stylicious.

## Personal Closet

Es una aplicación que permite realizar muchas operaciones con las prendas y los conjuntos, lo cual puede ser un arma de doble filo porque si se le añade un diseño de interfaz que no es adecuado, resulta poco intuitiva en su uso. Tiene elementos que podrían ser interesantes pero que todos juntos hacen la aplicación un poco caótica. Contiene demasiada publicidad.

En la Ilustración 7, en la primera imagen se presenta un formulario que es mostrado en el inicio de la aplicación. En la segunda se puede ver un listado de las prendas.

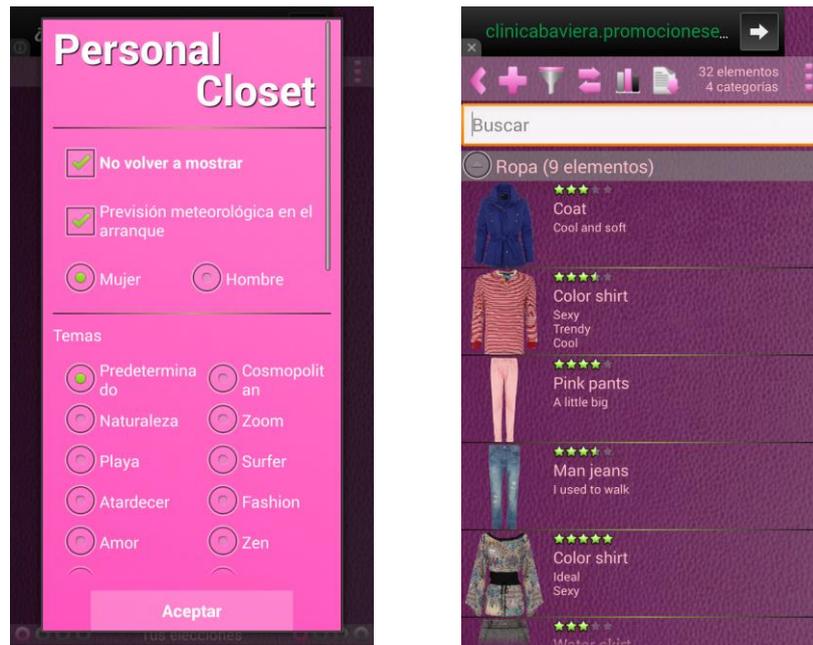


Ilustración 7. Pantallas de la aplicación Personal Closet.

## Mix ME

Resulta interesante en cuanto a la combinación de prendas se refiere, ya que divide las imágenes en tipos de prendas como partes de arriba y partes de abajo, pero es muy limitada ya que no aporta mayor funcionalidad que la creación de imágenes con los separadores de tipos de prendas.

Se muestra lo comentado en la Ilustración 8.

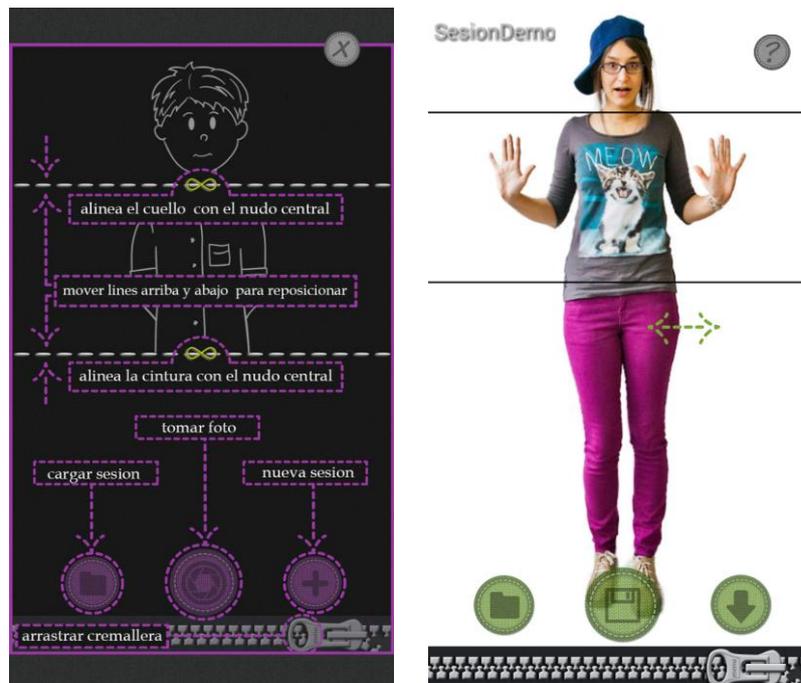


Ilustración 8. Pantallas de la aplicación Mix ME.

# Capítulo 2

## Planificación y presupuesto

---



## 2.1. Planificación

A la hora de llevar a cabo correctamente la planificación de un proyecto se debe tener en cuenta: la metodología que se va a utilizar, el tiempo disponible y recursos asignados, los objetivos y resultados de cada iteración, y tener también presente las restricciones de presupuesto.

En cuanto a la metodología a usar, en este caso se llevará a cabo un ciclo de vida basado en el Modelo Incremental, ya que no se van a saber todos los requisitos necesarios desde el principio, por lo que es necesario realizar varias iteraciones, permitiendo así tener un producto funcional en cada iteración. De esta manera, los primeros incrementos satisfacen los requisitos más críticos, servirán como prototipo y ayudarán en la tarea de detectar los posteriores requisitos.

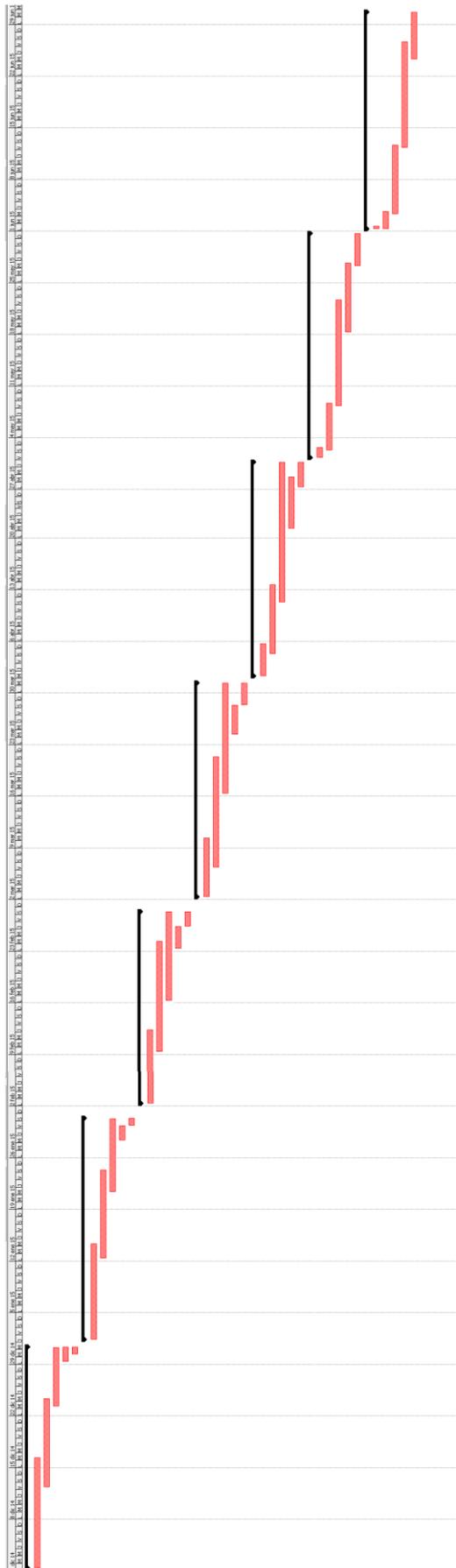
Cada iteración es mensual, y en cada una de estas iteraciones o incrementos se realizan las fases de *Análisis*, *Diseño*, *Implementación*, *Prueba* y *Despliegue*. En cada una de estas fases se realiza lo siguiente:

- **Análisis:** es el proceso en el que se intenta descubrir qué es lo que realmente se necesita y se llega a una comprensión adecuada de los requerimientos del sistema (las características que el sistema debe poseer).
- **Diseño:** fase en la que se describe requisitos de la arquitectura de la aplicación y la definición precisa de cada subconjunto de la aplicación, como la estructura del software, los datos y las interfaces entre componentes.
- **Implementación:** es la fase que se realiza una vez se conoce qué funciones debe desempeñar el sistema de información (análisis) y se ha decidido cómo se va a organizar sus distintos componentes (diseño). En ella se procede a desarrollar el sistema diseñado mediante el uso de las herramientas adecuadas.
- **Pruebas:** se prueba cada subconjunto de la aplicación creado para garantizar que el desarrollo ha sido realizado de acuerdo con las especificaciones.
- **Despliegue:** la etapa de despliegue implica instalación, personalización, y testeo.

En lo referido a la planificación temporal, que como se ha explicado está basada en incrementos, se llega a una estimación de iteraciones mensuales, que se encuentra representada en la siguiente tabla. También se puede observar la planificación de forma más visual en el diagrama de Gantt presentado en la Ilustración 10.

	📅	Nombre	Duración	Inicio	Terminado
1		<b>Iteración 1</b>	<b>30 days?</b>	<b>1/12/14 9:00</b>	<b>31/12/14 9:00</b>
2	📅	Análisis	15 days?	1/12/14 9:00	16/12/14 9:00
3	📅	Diseño	12 days?	12/12/14 9:00	24/12/14 9:00
4	📅	Implementación	8 days?	23/12/14 9:00	31/12/14 9:00
5	📅	Prueba	2 days?	29/12/14 9:00	31/12/14 9:00
6	📅	Despliegue	1 day?	30/12/14 9:00	31/12/14 9:00
7		<b>Iteración 2</b>	<b>30 days?</b>	<b>1/01/15 9:00</b>	<b>31/01/15 9:00</b>
8	📅	Análisis	13 days?	1/01/15 9:00	14/01/15 9:00
9	📅	Diseño	12 days?	12/01/15 9:00	24/01/15 9:00
10	📅	Implementación	10 days?	21/01/15 9:00	31/01/15 9:00
11	📅	Prueba	2 days?	28/01/15 9:00	30/01/15 9:00
12	📅	Despliegue	1 day?	30/01/15 9:00	31/01/15 9:00
13		<b>Iteración 3</b>	<b>26 days?</b>	<b>2/02/15 9:00</b>	<b>28/02/15 9:00</b>
14	📅	Análisis	10 days?	2/02/15 9:00	12/02/15 9:00
15	📅	Diseño	15 days?	9/02/15 9:00	24/02/15 9:00
16	📅	Implementación	12 days?	16/02/15 9:00	28/02/15 9:00
17	📅	Prueba	3 days?	23/02/15 9:00	26/02/15 9:00
18	📅	Despliegue	2 days?	26/02/15 9:00	28/02/15 9:00
19		<b>Iteración 4</b>	<b>29 days?</b>	<b>2/03/15 9:00</b>	<b>31/03/15 9:00</b>
20	📅	Análisis	8 days?	2/03/15 9:00	10/03/15 9:00
21	📅	Diseño	15 days?	6/03/15 9:00	21/03/15 9:00
22	📅	Implementación	15 days?	16/03/15 9:00	31/03/15 9:00
23	📅	Prueba	4 days?	24/03/15 9:00	28/03/15 9:00
24	📅	Despliegue	3 days?	28/03/15 9:00	31/03/15 9:00
25		<b>Iteración 5</b>	<b>30 days?</b>	<b>1/04/15 8:00</b>	<b>30/04/15 17:00</b>
26	📅	Análisis	5 days?	1/04/15 8:00	5/04/15 17:00
27	📅	Diseño	10 days?	4/04/15 8:00	13/04/15 17:00
28	📅	Implementación	20 days?	11/04/15 8:00	30/04/15 17:00
29	📅	Prueba	8 days?	21/04/15 8:00	28/04/15 17:00
30	📅	Despliegue	4 days?	27/04/15 8:00	30/04/15 17:00
31		<b>Iteración 6</b>	<b>31 days?</b>	<b>1/05/15 8:00</b>	<b>31/05/15 17:00</b>
32	📅	Análisis	2 days?	1/05/15 8:00	2/05/15 17:00
33	📅	Diseño	7 days?	2/05/15 8:00	8/05/15 17:00
34	📅	Implementación	15 days?	8/05/15 8:00	22/05/15 17:00
35	📅	Prueba	10 days?	18/05/15 8:00	27/05/15 17:00
36	📅	Despliegue	5 days?	27/05/15 8:00	31/05/15 17:00
37		<b>Iteración 7</b>	<b>30 days?</b>	<b>1/06/15 8:00</b>	<b>30/06/15 17:00</b>
38	📅	Análisis	1 day?	1/06/15 8:00	1/06/15 17:00
39	📅	Diseño	3 days?	1/06/15 8:00	3/06/15 17:00
40	📅	Implementación	10 days?	3/06/15 8:00	12/06/15 17:00
41	📅	Prueba	15 days?	12/06/15 8:00	26/06/15 17:00
42	📅	Despliegue	7 days?	24/06/15 8:00	30/06/15 17:00

**Ilustración 9. Planificación de las iteraciones con el Método Incremental.**



**Ilustración 10. Gráfico de la planificación de las iteraciones con el Método Incremental.**

## 2.2. Presupuesto

Las siguientes estimaciones se realizan a través de varios métodos con el objetivo de acotar los límites entre los que se va a encontrar el coste final y que se acerquen lo máximo al resultado final.

### 2.2.1. Estimación de costes

Para desarrollar el proyecto se necesitarán medios de Hardware y Software, cuyo coste proporcional al uso que se le dará en el proyecto hay que introducirlo en el presupuesto. Naturalmente, también se deberá incluir el coste de los recursos humanos utilizados.

Según el diagrama de Gantt obtenido anteriormente, para llevar a cabo las distintas actividades (teniendo en cuenta que se planea trabajar 25 horas a la semana) se van a necesitar las siguientes horas:

- **Análisis:** 173,25 horas.
- **Diseño:** 241,5 horas.
- **Implementación:** 325,5 horas.
- **Pruebas:** 203 horas.
- **Despliegue:** 63 horas.

Además se van a tener varias categorías de trabajo según las actividades, por lo que se tendrá:

- 1) Un **analista** para las actividades de Análisis y Diseño, sin contar el trabajo de documentar: 314,75 horas. Cobra 18 € la hora.
- 2) Un **documentalista** que se encargue de realizar todo lo relacionado con la recopilación de datos y redacción de la memoria: 100 horas. Cobra 10 € la hora.
- 3) Un **programador** para las actividades de Implementación, Pruebas y Despliegue: 591,5 horas. Cobra 15 € la hora.

**Horas totales estimadas = 1006,25 horas**

	TIEMPO	COSTE
Ingeniero Informática trabajando como <b>Analista</b>	314,75 horas	18 €/hora
Ingeniero Informática trabajando como <b>Programador</b>	591,5 horas	15 €/hora
Ingeniero Informática trabajando como <b>Documentalista</b>	100 horas	10 €/hora
<b>TOTAL</b> (1 persona)	1006,25 horas	15538€

**Tabla 2. Estimación mano de obra.**

**Presupuesto de Mano de Obra** = horas Analista \* precio Analista + horas Documentalista \* precio Documentalista + horas Programador \* precio Programador = 314,75\*18 + 100\*10 + 591,5\*15 = 5665,5 + 1000 + 8872,5 = 15 538 €

Hardware	Uso (%)	Coste total (€)	COSTE (€)
Un ordenador	10%	600	60
Conexión a Internet	19%	70	13,61
Terminal Android	40%	280	112
Impresora	10%	200	20
Material de oficina	80%	50	40
<b>TOTAL</b>			245,61

**Tabla 3. Estimación del coste del hardware.**

**Presupuesto Hardware = 245,61 €**

Entre los materiales software se encuentran los siguientes en forma de tabla:

Software	Uso (%)	Coste total (€)	COSTE (€)
Windows 8	30%	120	36
Eclipse	90%	0	0
Java 8	60%	0	0
SDK Android	70%	0	0
Día	10%	0	0
Procesador de texto	100%	0	0
Adobe Photoshop CS6	100%	37,95	37,95
<b>TOTAL</b>			73,95

**Tabla 4. Estimación del coste del software.**

**Presupuesto Software = 73,95 €**

	COSTE
Hardware	245,61 €
Software	73,95 €
Mano de obra	15538 €
<b><u>TOTAL</u></b>	15857,56 €

**Tabla 5. Estimación del coste total.**

**Presupuesto Total** = coste de la mano de obra + coste hardware + coste software = 15538 + 245,61 + 73,95 = **15 857,56 €**

## 2.2.2. Estimación mediante Puntos de Función

Este método se basa en una métrica que cuantifica la funcionalidad al construir la aplicación. Los parámetros que sirven para evaluar dicha funcionalidad son:

- **Número de entradas:** Datos que el usuario aporta al sistema (nombres de ficheros, menús de selección).
- **Número de salidas:** Datos que el sistema aporta al usuario (informes, mensajes).
- **Número de ficheros lógicos internos:** Ficheros o bases de datos internos del sistema.
- **Número de ficheros externos:** Ficheros o bases de datos externos al sistema.
- **Número de consultas externas:** Entradas que requieren de una respuesta por parte del sistema.

Para hallar los PFNA se debe contar el número de elementos de cada clase.

Cada elemento de cada clase debe ser clasificado según su grado de complejidad (alta, media o baja). Los criterios para evaluar la complejidad de los elementos de cálculo:

Ficheros lógicos externos e internos				Salidas y consultas				Entradas			
Registros elementales	Datos elementales			Tipos de ficheros	Datos elementales			Tipos de ficheros	Datos elementales		
	1-9	20-50	>51		1-9	6-19	>20		1-4	5-15	>16
1	Baja	Baja	Media	0-1	Baja	Baja	Media	0-1	Baja	Baja	Media
2-5	Baja	Media	Alta	2-3	Baja	Media	Alta	2-3	Baja	Media	Alta
>6	Media	Alta	Alta	>4	Media	Alta	Alta	>3	Media	Alta	Alta

**Tabla 6. Grados de complejidad de PF.**

	Complejidad baja	Complejidad media	Complejidad alta	TOTAL
<b>Entradas</b>	4x3	2x4	0x6	20
<b>Salidas</b>	14x4	0x5	0x7	56
<b>Consultas</b>	19x3	0x4	0x6	57
<b>Ficheros Internos</b>	1x7	0x10	0x15	7
<b>Ficheros Externos</b>	1x5	0x7	0x10	5

**Tabla 7. Valores de complejidad de PF.**

**PFNA (Puntos de Función No Ajustados) = 145**

Factores de complejidad (FC)	0-5
Comunicación de datos	4
Rendimiento	5
Frecuencia de transacciones	4
Requisitos de manejo del usuario final	2
Procesos complejos	3
Facilidad de mantenimiento	3
Instalación en múltiples lugares	1
Funciones distribuidas	0
Gran carga de trabajo	3
Entrada on-line de datos	0
Actualizaciones on-line	0
Utilización con otros sistemas	2
Facilidad de operación	2
Facilidad de cambio	2

**Tabla 8. Factores de complejidad.**

**$\Sigma FC$  (Sumatorio de los Factores de Complejidad) = 31**

**FA (Factor de Ajuste) =  $(0,01 * \Sigma FC) + 0,65 = 0,96$**

**PF (Puntos de Función) = PFNA \* FA = 145 \* 0,96 = 139,2**

Si un punto de función equivale a 53 líneas de código en Java:

**$139,2 * 53 = 7377,6 \text{ LDC} \approx 7,4 \text{ KLDC}$**

### 2.2.3. Estimación mediante COCOMO

Para realizar el modelado algorítmico de costes se va a llevar a cabo un **COCOMO semi-acoplado**, debido a que tiene requisitos restrictivos, existen presiones de tiempo y no existe la suficiente experiencia previa.

PROYECTO SOFTWARE	a	b	c	d
Orgánico	2,4	1,05	2,5	0,38
Semi-acoplado	3,0	1,12	2,5	0,35
Empotrado	3,2	1,2	2,5	0,32

Tabla 9. Tipos de COCOMO.

$$\text{Esfuerzo nominal} = a \cdot (\text{KLDC})^b = 3 \cdot 7,4^{1,12} = 28,22$$

CONDUCTORES DE COSTE	VALORACIÓN
Fiabilidad requerida del software	1,15
Tamaño de la base de datos	1,00
Complejidad del software	1,15
Restricciones de tiempo de ejecución	1,00
Restricciones de memoria	1,21
Volatilidad de la máquina virtual	1,00
Tiempo de respuesta requerido	1,07
Capacidad de los analistas	1,00
Experiencia con el tipo de aplicación	1,13
Experiencia con el hardware	1,00
Experiencia con el lenguaje de programación	1,00
Capacidad de los programadores	1,00
Prácticas de programación modernas	0,91
Utilización de herramientas software	0,91
Restricciones en la planificación temporal del desarrollo	1,04

Tabla 10. Conductores de coste de COCOMO.

**Esfuerzo** = Esfuerzo nominal \* Valor de los factores conductores del coste =  $28,22 * (1,15 * 1,00 * 1,15 * 1,00 * 1,21 * 1,00 * 1,07 * 1,00 * 1,13 * 1,00 * 1,00 * 1,00 * 0,91 * 0,91 * 1,04) = 28,22 * 1,66 = 46,85$  **personas-mes**

**Tiempo de desarrollo** =  $c * E^d = 2.5 * 46,85^{0,35} = 9,60$  meses

**Nº medio de personas** =  $E/TD = 46,85/9,60 \approx 4$  personas para realizar el trabajo en 9 meses.

## 2.2.4. Estimación mediante Casos de Uso

El método de Puntos de Casos de Uso requiere de experiencia previa, pero realizando este método se pretende familiarizarse con él.

Lo primero que se debe realizar es una clasificación de cada interacción entre actor y caso de uso según su complejidad y asignar un peso en función de ésta. Para poder clasificar la complejidad de los actores se debe analizar la interacción de éste con el sistema que se va a desarrollar.

TIPO DE INTERACCIÓN	PESO ASIGNADO
<b>Simple</b> (a través de API)	1
<b>Media</b> (a través de protocolo)	2
<b>Compleja</b> (a través de interfaz gráfica)	3

**Tabla 11. Interacción método de Casos de Uso.**

Debido a que solo se tiene un actor en el sistema y que su interacción con los Casos de Uso (especificados en el capítulo de Análisis) se dan a través de una interfaz gráfica, se le asigna un peso correspondiente con una interacción compleja.

Peso de los actores sin ajustar (**UAW**) =  $1 * 3 = 3$

Después se calcula la complejidad de cada caso de uso según el número de transacciones o pasos del mismo, siendo la asignación de pesos la que se muestra en la tabla siguiente.

Nº DE TRANSACCIONES DEL CASO DE USO	TIPO	PESO
menor o igual que 3	<b>Simple</b>	5
mayor o igual que 4 y menor que 7	<b>Medio</b>	10
mayor o igual que 7	<b>Complejo</b>	15

**Tabla 12. Transacciones método de Casos de Uso.**

CASO DE USO	COMPLEJIDAD
CU-01	15
CU-02	10
CU-03	10
CU-04	5
CU-05	5
CU-06	15
CU-07	10
CU-08	5
CU-09	10
CU-10	5
CU-11	5
CU-12	15
CU-13	10
CU-14	5
CU-15	10
CU-16	10
CU-17	10
<b>TOTAL</b>	<b>155</b>

**Tabla 13. Complejidades método de Casos de Uso.**

Peso de los Casos de Uso sin ajustar (**UUCW**) = **155**

Con ello ya se puede calcular los Puntos Casos de Uso No Ajustados (**UUCP**) del sistema.

$$\mathbf{UUCP} = 3 + 155 = \mathbf{158}$$

Lo siguiente es realizar el cálculo de los Factores Técnicos (TCF). A cada uno de los Factores Técnicos de la tabla siguiente se le asigna un valor de influencia en el proyecto.

FACTOR	DESCRIPCIÓN	PESO	INFLUENCIA	TOTAL
R1	Sistema Distribuido	2	0	0
R2	Objetivos de rendimiento	1	4	4
R3	Eficiencia respecto al usuario final	1	4	4
R4	Procesamiento complejo	1	3	3
R5	Código reutilizable	1	3	3
R6	Instalación sencilla	0,5	5	2,5
R7	Fácil utilización	0,5	4	2
R8	Portabilidad	2	2	4
R9	Fácil de cambiar	1	3	3
R10	Uso Concurrente	1	3	3
R11	Características de seguridad	1	2	2
R12	Accesible por terceros	1	0	0
R13	Se requiere formación especial	1	0	0
				<b>30,5</b>

**Tabla 14. Factores Técnicos método de Casos de Uso.**

Obtenidos los grados de influencia se multiplican por el peso de cada factor y con la siguiente fórmula se calcula el Factor Técnico que aplica:

$$\mathbf{TCF} = 0,6 + (0,01 \cdot \sum_{i=1}^{i=13} R_i)$$

$$\mathbf{TCF} = 0,6 + (0,01 * 30,5) = \mathbf{0,905}$$

Después se procede de manera similar con los Factores de Entorno.

FACTOR	DESCRIPCIÓN	PESO	INFLUENCIA	TOTAL
R1	Familiaridad con el modelo de proyecto utilizado	1,5	2	3
R2	Experiencia en la aplicación	0,5	1	0,5
R3	Experiencia en orientación a objetos	1,0	3	3
R4	Capacidades de análisis	0,5	3	1,5
R5	Motivación	1,0	5	5
R6	Estabilidad de los requerimientos	2,0	3	6
R7	Trabajadores a tiempo parcial	-1,0	1	-1
R8	Dificultad del lenguaje de programación	-1,0	2	-2
				<b>16</b>

**Tabla 15. Factores de Entorno método de Casos de Uso.**

$$EF = 1,4 - (0,03 \cdot \sum_{i=1}^{i=8} R_i)$$

$$EF = 1,4 - (0,03 * 16) = \mathbf{0,92}$$

Y así se tiene los puntos de Casos de Uso ajustados:

$$UCP = UUCP * TCF * EF = 158 * 0,905 * 0,92 = \mathbf{131,55}$$

Una vez obtenido el número de Puntos Casos de Uso, si se quiere obtener el esfuerzo necesario para llevarlos a cabo en el método se provee de un factor de productividad. El autor propone un valor de 20 horas/persona.

$$E = 131,55 UCP * 20 \text{ horas/hombre} = \mathbf{2631 \text{ horas/hombre}}$$

## 2.3. Costes finales

Las horas reales empleadas en las distintas actividades han sido aproximadamente las siguientes:

- **Análisis:** 100 horas.
- **Diseño:** 90 horas.
- **Implementación:** 350 horas.
- **Pruebas:** 130 horas.
- **Despliegue:** 30 horas.

Ahora se procede a dividir las horas entre las distintas categorías de personas que trabajan el proyecto.

- 1) Un **analista** para las actividades de Análisis y Diseño, sin contar el trabajo de documentar: 130 horas. Cobra 18 € la hora.
- 2) Un **documentalista** que se encargue de realizar todo lo relacionado con la recopilación de datos y redacción de la memoria: 60 horas. Cobra 10 € la hora.
- 3) Un **programador** para las actividades de Implementación, Pruebas y Despliegue: 510 horas. Cobra 15 € la hora.

**Horas totales ≈ 700 horas**

	Tiempo	Coste
Ingeniero Informática trabajando como <b>Analista</b>	130 horas	18 €/hora
Ingeniero Informática trabajando como <b>Programador</b>	510 horas	15 €/hora
Ingeniero Informática trabajando como <b>Documentalista</b>	60 horas	10 €/hora
<b>TOTAL</b> (1 persona)	700 horas	10590€

**Tabla 16. Coste real de la mano de obra.**

**Presupuesto de Mano de Obra** = horas Analista \* precio Analista + horas Documentalista \* precio Documentalista + horas Programador \* precio Programador =  $130 \cdot 18 + 60 \cdot 10 + 510 \cdot 15 = 2340 + 600 + 7650 = 10\,590 \text{ €}$

Hardware	Uso (%)	Coste total (€)	COSTE (€)
Un ordenador	10%	600	60
Conexión a Internet	19%	70	13,61
Terminal Android	40%	280	112
Impresora	10%	200	20
Material de oficina	80%	50	40
<b>TOTAL</b>			245,61

**Tabla 17. Coste real del hardware.**

**Costes Hardware = 245,61 €**

Entre los materiales software se encuentran los siguientes en forma de tabla:

Software	Uso (%)	Coste total (€)	COSTE (€)
Windows 8	30%	120	36
Eclipse	90%	0	0
Java 8	60%	0	0
SDK Android	70%	0	0
ORMLite	60%	0	0
Día	10%	0	0
Draw.io	80%	0	0
Procesador de texto	100%	0	0
Adobe Photoshop CS6	100%	37,95	37,95
<b>TOTAL</b>			73,95

**Tabla 18. Coste real del software.**

**Costes Software = 73,95 €**

	<b>COSTE</b>
Hardware	245,61 €
Software	73,95 €
Mano de obra	10590 €
<b><u>TOTAL</u></b>	<b>10909,56 €</b>

**Tabla 19. Coste real total.**

**Costes Totales** = coste de la mano de obra + coste hardware + coste software = 10590 + 245,61 + 73,95 = **10909,56 €**

En realidad, se ha necesitado menos tiempo de desarrollo que lo que se estimaba inicialmente, lo que ha hecho que el coste de mano de obra, y en consecuencia, el coste total disminuya considerablemente. El número de líneas de código totales también ha resultado menor a las estimadas mediante Puntos de Función. Y el número de personas necesarias y el tiempo de desarrollo también ha sido menor que lo estimado mediante COCOMO.



# Capítulo 3

## Análisis

---



## 3.1. Características principales

En este apartado se lista las características principales del software a desarrollar. Para dar con las características del sistema se debe pensar cómo los usuarios lo utilizarán y así, determinar tanto la falta de alguna característica, como la posibilidad de estar considerando características innecesarias.

### **C-01:** Gestión de prendas:

**C-01.1:** Módulo Armario. Es el módulo encargado de mostrar las prendas, modificarlas y eliminarlas.

### **C-02:** Gestión de conjuntos:

**C-02.1:** Módulo Conjuntos. Es el módulo encargado de mostrar los conjuntos, modificarlos y eliminarlos.

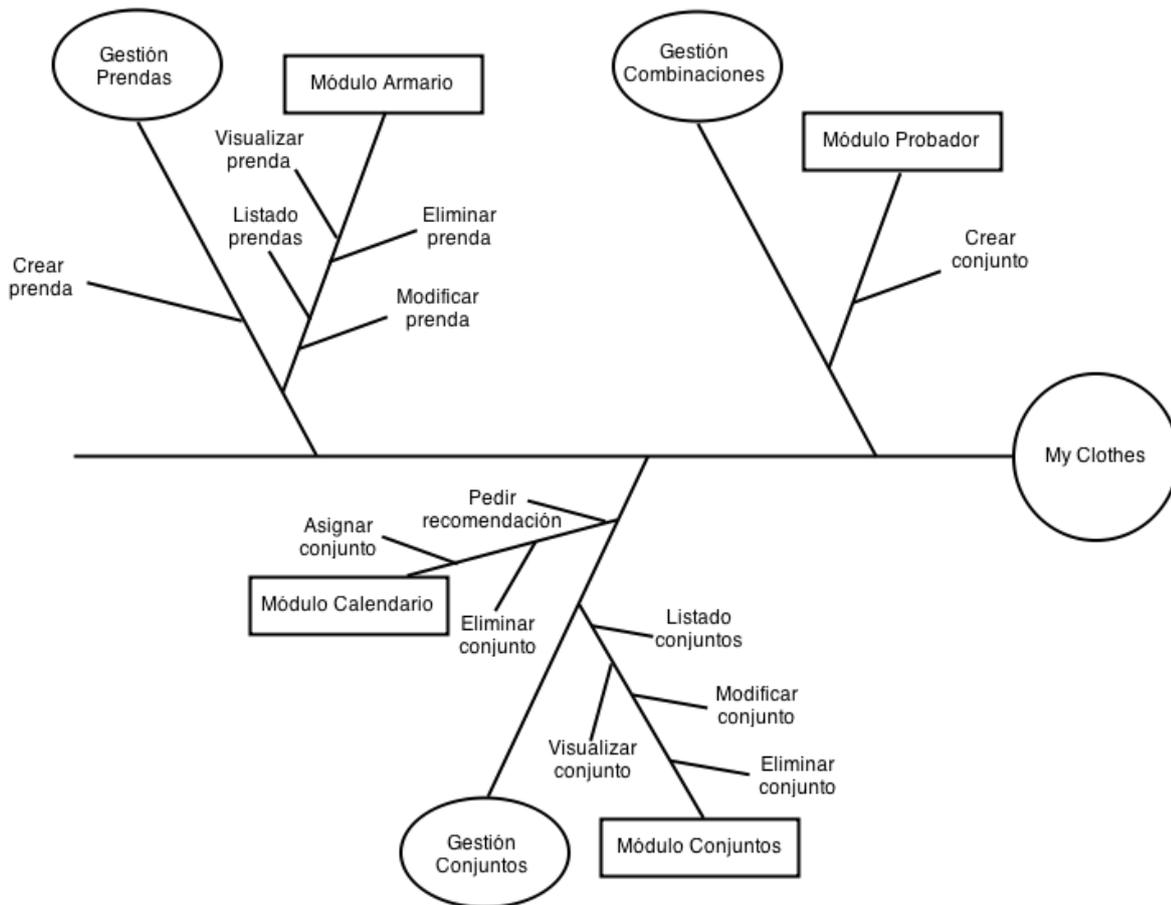
**C-02.2:** Módulo Calendario. Es el módulo encargado de asignar y mostrar los conjuntos en sus correspondientes fechas. También permite su eliminación.

### **C-03:** Gestión de combinaciones:

**C-03.1:** Módulo Probador. Es el módulo encargado de mostrar simultáneamente las prendas, de tal forma que puedan combinarse para formar conjuntos.

### 3.1.1. Árbol de características

El siguiente árbol de características plantea una forma efectiva para la descripción de las características del sistema.



**Ilustración 11. Diagrama del árbol de características.**

## 3.2. Identificación de los actores del sistema

Existe un actor principal, que es el **Usuario** que va a hacer uso de la aplicación. Además existe el actor **Forecast** que es el servicio externo que aporta la climatología.

## 3.3. Requisitos de usuario

En esta sección se van a exponer los requisitos de usuario modelándolos en forma de Casos de Uso, viéndose el sistema desde la perspectiva de los usuarios que interactúan con él.

### 3.3.1. Relación de los Casos de Uso con actores

El actor **Usuario** es aquel que puede realizar todos los Casos de Uso utilizando su dispositivo móvil de uso personal.

El actor **Forecast** se relaciona con los **CU-15** y **CU-17**, ya que son los que necesitan del servicio de climatología. Estos casos de uso serán mostrados en el siguiente apartado.

### 3.3.2. Listado de Casos de Uso

Se recogen los Casos de Uso del sistema para posteriormente realizar su especificación. El listado de estos Casos de Uso obtenido es el siguiente:

- CU-01.** Crear prenda.
  - CU-02.** Capturar foto.
  - CU-03.** Seleccionar foto galería.
- CU-04.** Visualizar armario.
  - CU-05.** Visualizar prenda.
    - CU-06.** Modificar datos prenda.
    - CU-07.** Eliminar prenda.
- CU-08.** Visualizar probador.
  - CU-09.** Seleccionar conjunto.
- CU-10.** Visualizar conjuntos.
  - CU-11.** Visualizar conjunto.
    - CU-12.** Modificar datos conjunto.
    - CU-13.** Eliminar conjunto.
- CU-14.** Visualizar calendario.
  - CU-15.** Asignar conjunto.
  - CU-16.** Eliminar conjunto.
- CU-17.** Pedir recomendación.

### 3.3.3. Diagrama de Casos de Uso

En el siguiente diagrama se muestra la interacción del actor con los Casos de Uso.



**Ilustración 12. Diagrama de Casos de Uso.**

### 3.3.4. Especificación de los Casos de Uso

CU - 01		CREAR PRENDA
<b>Versión</b>	1.0	
<b>Actor principal</b>	Usuario	
<b>Requisitos asociados</b>	<p><b>RF-01:</b> El sistema mostrará las opciones de cámara o galería.</p> <p><b>RF-04:</b> El sistema mostrará la imagen seleccionada a recortar.</p> <p><b>RF-05:</b> El sistema recortará la imagen elegida con la forma elegida.</p> <p><b>RF-06:</b> El sistema mostrará los campos a rellenar de la prenda.</p> <p><b>RF-07:</b> El sistema almacenará de forma asíncrona la imagen junto con su información.</p>	
<b>Descripción</b>	El usuario crea una prenda.	
<b>Precondición</b>	-	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario solicita la creación de una prenda.
	2	El sistema da a elegir al usuario entre el CU-02 o el CU-03.
	3	El usuario realiza el CU-02 o el CU-03.
	4	El sistema solicita ajustar la imagen.
	5	El usuario completa los datos de la prenda.
	6	El sistema recoge la información y la guarda.
<b>Postcondición</b>	Creación de una nueva prenda.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Fallo en la recogida de la imagen.
	4	Fallo en el recortado.
	5	Datos incorrectos.
	6	Fallo en el guardado de los datos de la prenda.
<b>Frecuencia</b>	Muy alta.	
<b>Importancia</b>	Imprescindible.	
<b>Comentarios</b>	Es necesario realizar este caso de uso para poder realizar alguno de los posteriores casos de uso.	

**Tabla 20. Especificación CU-01.**

<b>CU - 02</b>	<b>CAPTURAR IMAGEN</b>	
<b>Versión</b>	1.0	
<b>Actor principal</b>	Usuario	
<b>Requisitos asociados</b>	<b>RF-02:</b> El sistema accederá a la cámara del dispositivo.	
<b>Descripción</b>	El usuario realiza la captura de la foto de la prenda.	
<b>Precondición</b>	Solicitar la creación de una nueva prenda.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario solicita la captura de la foto.
	2	El sistema realiza la captura.
	3	El usuario confirma la foto capturada.
<b>Postcondición</b>	Se procede a la solicitud de recorte de la imagen.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	1	Fallo al acceder a la cámara.
	2	Fallo en la captura.
<b>Frecuencia</b>	Alta.	
<b>Importancia</b>	Alta.	
<b>Comentarios</b>	-	

**Tabla 21. Especificación CU-02.**

CU - 03		SELECCIONAR IMAGEN GALERÍA
<b>Versión</b>	1.0	
<b>Actor principal</b>	Usuario	
<b>Requisitos asociados</b>	<b>RF-03:</b> El sistema mostrará la galería del dispositivo.	
<b>Descripción</b>	El usuario elige la imagen de la prenda deseada desde la galería del dispositivo.	
<b>Precondición</b>	Solicitar la creación de una nueva prenda.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario solicita la visualización de su galería de imágenes.
	2	El sistema muestra la galería de imágenes.
	3	El usuario selecciona la imagen y confirma su selección.
<b>Postcondición</b>	Se procede a la solicitud de recorte de la imagen.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	1	No existen imágenes.
<b>Frecuencia</b>	Alta.	
<b>Importancia</b>	Alta.	
<b>Comentarios</b>	-	

**Tabla 22. Especificación CU-03.**

CU - 04	VISUALIZAR ARMARIO	
<b>Versión</b>	1.0	
<b>Actor principal</b>	Usuario	
<b>Requisitos asociados</b>	<b>RF-08:</b> El sistema mostrará las categorías en las que están organizadas las prendas.	
<b>Descripción</b>	El usuario entra en el módulo que permite la gestión de las prendas.	
<b>Precondición</b>	-	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario solicita la visualización del Armario.
	2	El sistema muestra el Armario.
	3	El usuario solicita la visualización mediante una clasificación específica y la visualización de un listado de prendas mediante la categoría seleccionada.
<b>Postcondición</b>	-	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	1	Fallo al cargar las imágenes de las prendas.
<b>Frecuencia</b>	Alta.	
<b>Importancia</b>	Alta.	
<b>Comentarios</b>	-	

**Tabla 23. Especificación CU-04.**

CU - 05	VISUALIZAR PRENDA	
<b>Versión</b>	1.0	
<b>Actor principal</b>	Usuario	
<b>Requisitos asociados</b>	<b>RF-09:</b> El sistema permitirá visualizar la prenda y su información.	
<b>Descripción</b>	El usuario visualiza la información y la imagen de la prenda seleccionada.	
<b>Precondición</b>	Seleccionar la prenda a visualizar.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario realiza la selección de la prenda a visualizar.
	2	El sistema muestra la información relativa a la prenda seleccionada.
<b>Postcondición</b>	-	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	1	Fallo al cargar los datos de la prenda.
<b>Frecuencia</b>	Muy alta.	
<b>Importancia</b>	Alta.	
<b>Comentarios</b>	-	

**Tabla 24. Especificación CU-05.**

CU - 06		MODIFICAR DATOS PRENDA	
<b>Versión</b>	1.0		
<b>Actor principal</b>	Usuario		
<b>Requisitos asociados</b>	<p><b>RF-10:</b> El sistema mostrará los datos actuales de la prenda y permitirá su modificación.</p> <p><b>RF-11:</b> El sistema actualizará la prenda con los nuevos datos.</p>		
<b>Descripción</b>	El usuario modifica la información de interés sobre la prenda.		
<b>Precondición</b>	Seleccionar la prenda a modificar.		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El usuario solicita la modificación de la prenda visualizada.	
	2	El sistema permite la modificación de los datos.	
	3	El usuario modifica los datos deseados.	
	4	El usuario confirma su modificación.	
<b>Postcondición</b>	Datos modificados.		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	1	La prenda no existe.	
	2	Datos incorrectos.	
	3	Error al actualizar.	
<b>Frecuencia</b>	Media.		
<b>Importancia</b>	Media.		
<b>Comentarios</b>	-		

**Tabla 25. Especificación CU-06.**

<b>CU - 07</b>		<b>ELIMINAR PRENDA</b>	
<b>Versión</b>	1.0		
<b>Actor principal</b>	Usuario		
<b>Requisitos asociados</b>	<b>RF-12:</b> El sistema permitirá eliminar la prenda.		
<b>Descripción</b>	El usuario elimina la prenda seleccionada.		
<b>Precondición</b>	Seleccionar la prenda a eliminar.		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El usuario solicita la eliminación de la prenda.	
	2	El usuario confirma la eliminación de la prenda.	
	3	El sistema elimina la prenda.	
<b>Postcondición</b>	Eliminación de la prenda.		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	1	La prenda no existe.	
	2	Error al eliminar.	
<b>Frecuencia</b>	Baja.		
<b>Importancia</b>	Baja.		
<b>Comentarios</b>	-		

**Tabla 26. Especificación CU-07.**

CU - 08		VISUALIZAR PROBADOR	
<b>Versión</b>	1.0		
<b>Actor principal</b>	Usuario		
<b>Requisitos asociados</b>	<b>RF-13:</b> El sistema permitirá combinar prendas.		
<b>Descripción</b>	El usuario puede combinar unas prendas con otras para formar un conjunto.		
<b>Precondición</b>	Prendas existentes.		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El usuario solicita la visualización del Probador.	
	2	El usuario combina distintas prendas.	
<b>Postcondición</b>			
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	1	Fallo al combinar tipos de prendas.	
<b>Frecuencia</b>	Alta.		
<b>Importancia</b>	Muy alta.		
<b>Comentarios</b>	-		

**Tabla 27. Especificación CU-08.**

CU - 09		SELECCIONAR CONJUNTO	
<b>Versión</b>	1.0		
<b>Actor principal</b>	Usuario		
<b>Requisitos asociados</b>	<b>RF-14:</b> El sistema permitirá crear conjuntos. <b>RF-15:</b> El sistema permitirá rellenar información sobre el conjunto. <b>RF-16:</b> El sistema permitirá guardar el conjunto.		
<b>Descripción</b>	El usuario crea un nuevo conjunto.		
<b>Precondición</b>	Realizar CU-08.		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El usuario solicita la creación de un nuevo conjunto.	
	2	El sistema muestra los datos a rellenar.	
	3	El usuario escribe los datos del conjunto.	
	4	El sistema recoge la información y la guarda.	
<b>Postcondición</b>	Creación de un nuevo conjunto.		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	1	Datos incorrectos.	
	2	Fallo en el guardado de la información del conjunto.	
<b>Frecuencia</b>	Alta.		
<b>Importancia</b>	Muy alta.		
<b>Comentarios</b>	-		

**Tabla 28. Especificación CU-09.**

CU - 10	VISUALIZAR CONJUNTOS	
<b>Versión</b>	1.0	
<b>Actor principal</b>	Usuario	
<b>Requisitos asociados</b>	<b>RF-17:</b> El sistema mostrará las categorías en las que están organizados los conjuntos.	
<b>Descripción</b>	El usuario entra en el módulo que permite la gestión de los conjuntos.	
<b>Precondición</b>	-	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario solicita la visualización de Conjuntos.
	2	El sistema muestra Conjuntos.
	3	El usuario solicita la visualización de Conjuntos mediante una clasificación específica, y la visualización de un listado de los conjuntos mediante una categoría concreta.
<b>Postcondición</b>	-	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	1	Fallo al cargar las imágenes de los conjuntos.
<b>Frecuencia</b>	Alta.	
<b>Importancia</b>	Alta.	
<b>Comentarios</b>	-	

Tabla 29. Especificación CU-10.

<b>CU - 11</b>	<b>VISUALIZAR CONJUNTO</b>	
<b>Versión</b>	1.0	
<b>Actor principal</b>	Usuario	
<b>Requisitos asociados</b>	<b>RF-18:</b> El sistema mostrará la información del conjunto.	
<b>Descripción</b>	El usuario visualiza la información del conjunto seleccionado.	
<b>Precondición</b>	Seleccionar el conjunto a visualizar.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario solicita la selección del conjunto a visualizar.
	2	El sistema muestra la información relativa al conjunto seleccionado.
<b>Postcondición</b>	-	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	1	Fallo al cargar los datos del conjunto.
<b>Frecuencia</b>	Muy alta.	
<b>Importancia</b>	Alta.	
<b>Comentarios</b>		

**Tabla 30. Especificación CU-11.**

CU - 12		MODIFICAR DATOS CONJUNTO	
<b>Versión</b>	1.0		
<b>Actor principal</b>	Usuario		
<b>Requisitos asociados</b>	<p><b>RF-19:</b> El sistema permitirá modificar los datos del conjunto.</p> <p><b>RF-20:</b> El sistema actualizará el conjunto con los nuevos datos.</p>		
<b>Descripción</b>	El usuario modifica la información de interés sobre el conjunto.		
<b>Precondición</b>	Seleccionar el conjunto a modificar.		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El usuario solicita la modificación del conjunto visualizado.	
	2	El sistema permite la modificación de los datos.	
	3	El usuario modifica los datos deseados.	
	4	El usuario confirma la modificación.	
<b>Postcondición</b>	Datos del conjunto modificados.		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	1	El conjunto no existe.	
	2	Datos incorrectos.	
	3	Error al actualizar.	
<b>Frecuencia</b>	Media.		
<b>Importancia</b>	Baja.		
<b>Comentarios</b>	-		

**Tabla 31. Especificación CU-12.**

<b>CU - 13</b>		<b>ELIMINAR CONJUNTO</b>	
<b>Versión</b>	1.0		
<b>Actor principal</b>	Usuario		
<b>Requisitos asociados</b>	<b>RF-21:</b> El sistema permitirá la eliminación del conjunto.		
<b>Descripción</b>	El usuario elimina el conjunto seleccionado.		
<b>Precondición</b>	Seleccionar el conjunto a eliminar.		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El usuario solicita la eliminación del conjunto.	
	2	El usuario confirma la eliminación del conjunto.	
	3	El sistema elimina el conjunto.	
<b>Postcondición</b>	Eliminación del conjunto.		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	1	El conjunto no existe.	
	2	Error al eliminar.	
<b>Frecuencia</b>	Baja.		
<b>Importancia</b>	Baja.		
<b>Comentarios</b>	-		

**Tabla 32. Especificación CU-13.**

<b>CU - 14</b>	<b>VISUALIZAR CALENDARIO</b>	
<b>Versión</b>	1.0	
<b>Actor principal</b>	Usuario	
<b>Requisitos asociados</b>	<b>RF-22:</b> El sistema mostrará el calendario y conjuntos asignados a sus días.	
<b>Descripción</b>	El usuario visualizará el calendario.	
<b>Precondición</b>	-	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario solicita la visualización del Calendario.
	2	El sistema muestra el calendario con los conjuntos asignados en su fecha determinada.
<b>Postcondición</b>	-	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	1	Fallo al cargar las imágenes de los conjuntos.
<b>Frecuencia</b>	Alta.	
<b>Importancia</b>	Alta.	
<b>Comentarios</b>	-	

**Tabla 33. Especificación CU-14.**

CU - 15		ASIGNAR CONJUNTO	
<b>Versión</b>	1.0		
<b>Actores</b>	Usuario y Forecast		
<b>Requisitos asociados</b>	<p><b>RF-23:</b> El sistema permitirá seleccionar el clima.</p> <p><b>RF-24:</b> El sistema mostrará los conjuntos que se pueden asignar.</p> <p><b>RF-25:</b> El sistema vinculará el conjunto asignado al día seleccionado.</p>		
<b>Descripción</b>	El usuario asigna un conjunto al día del calendario seleccionado.		
<b>Precondición</b>	Visualizar calendario.		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El usuario selecciona el día.	
	2	Forecast recoge el clima.	
	2	El sistema muestra el clima.	
	3	El usuario selecciona el clima.	
	4	El sistema muestra el listado de los conjuntos.	
	5	El usuario selecciona el conjunto a asignar.	
<b>Postcondición</b>	Vincular día con conjunto.		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	1	El día ya tiene conjunto asignado.	
	4	Fallo al cargar las imágenes de los conjuntos.	
<b>Frecuencia</b>	Alta.		
<b>Importancia</b>	Alta.		
<b>Comentarios</b>	-		

Tabla 34. Especificación CU-15.

CU - 16		ELIMINAR CONJUNTO	
<b>Versión</b>	1.0		
<b>Actor principal</b>	Usuario		
<b>Requisitos asociados</b>	<b>RF-26:</b> El sistema permitirá eliminar conjuntos a un día del calendario.		
<b>Descripción</b>	El usuario elimina un conjunto del día del calendario seleccionado.		
<b>Precondición</b>	Conjunto asignado.		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El usuario solicita la eliminación del conjunto en dicho día.	
	2	El usuario confirma su eliminación.	
	3	El sistema elimina el conjunto del mencionado día.	
<b>Postcondición</b>	Conjunto eliminado.		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	1	El día no tiene conjunto asignado.	
<b>Frecuencia</b>	Baja.		
<b>Importancia</b>	Baja.		
<b>Comentarios</b>	-		

**Tabla 35. Especificación CU-16.**

CU - 17	PEDIR RECOMENDACIÓN	
<b>Versión</b>	1.0	
<b>Actores</b>	Usuario y Forecast	
<b>Requisitos asociados</b>	<p><b>RF-27:</b> El sistema selecciona los parámetros del conjunto a recomendar.</p> <p><b>RF-28:</b> El sistema calculará el mejor conjunto a recomendar.</p> <p><b>RF-18:</b> El sistema mostrará la información del conjunto.</p>	
<b>Descripción</b>	El usuario solicita que se le recomiende un conjunto en base a la estación, estilo y clima.	
<b>Precondición</b>	Conjuntos asignados al calendario.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario solicita una recomendación.
	2	Forecast recoge el clima.
	3	El sistema muestra el clima.
	4	El usuario selecciona el clima, la estación y el estilo.
	5	El sistema obtiene un conjunto a recomendar.
<b>Postcondición</b>	Mostrar conjunto recomendado.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	1	No existen conjuntos asignados al calendario.
	4	No existe ningún conjunto que cumpla las condiciones.
<b>Frecuencia</b>	Baja.	
<b>Importancia</b>	Baja.	
<b>Comentarios</b>	-	

Tabla 36. Especificación CU-17.

## 3.4. Requisitos funcionales

- **RF-01:** El sistema mostrará las opciones de cámara o galería.
- **RF-02:** El sistema accederá a la cámara del dispositivo.
- **RF-03:** El sistema accederá a la galería del dispositivo.
- **RF-04:** El sistema mostrará la imagen seleccionada a recortar.
- **RF-05:** El sistema recortará la imagen elegida con la forma elegida.
- **RF-06:** El sistema mostrará los campos a rellenar de la prenda.
- **RF-07:** El sistema almacenará de forma asíncrona la imagen junto con su información.
- **RF-08:** El sistema mostrará las categorías en las que están organizadas las prendas.
- **RF-09:** El sistema permitirá visualizar la prenda y su información.
- **RF-10:** El sistema mostrará los datos actuales de la prenda y permitirá su modificación.
- **RF-11:** El sistema actualizará la prenda con los nuevos datos.
- **RF-12:** El sistema permitirá eliminar la prenda.
- **RF-13:** El sistema permitirá combinar prendas.
- **RF-14:** El sistema permitirá crear conjuntos.
- **RF-15:** El sistema permitirá rellenar información sobre el conjunto.
- **RF-16:** El sistema permitirá guardar el conjunto.
- **RF-17:** El sistema mostrará las categorías en las que se clasifican los conjuntos.
- **RF-18:** El sistema mostrará la información del conjunto.
- **RF-19:** El sistema permitirá modificar los datos del conjunto.
- **RF-20:** El sistema actualizará el conjunto con los nuevos datos.
- **RF-21:** El sistema permitirá la eliminación del conjunto.
- **RF-22:** El sistema mostrará el calendario y conjuntos asignados a sus días.
- **RF-23:** El sistema permitirá seleccionar el clima.
- **RF-24:** El sistema mostrará los conjuntos que se pueden asignar.
- **RF-25:** El sistema vinculará el conjunto asignado al día seleccionado.
- **RF-26:** El sistema permitirá eliminar conjuntos a un día del calendario.
- **RF-27:** El sistema selecciona los parámetros del conjunto a recomendar.
- **RF-28:** El sistema calculará el mejor conjunto a recomendar.

## 3.5. Requisitos de información

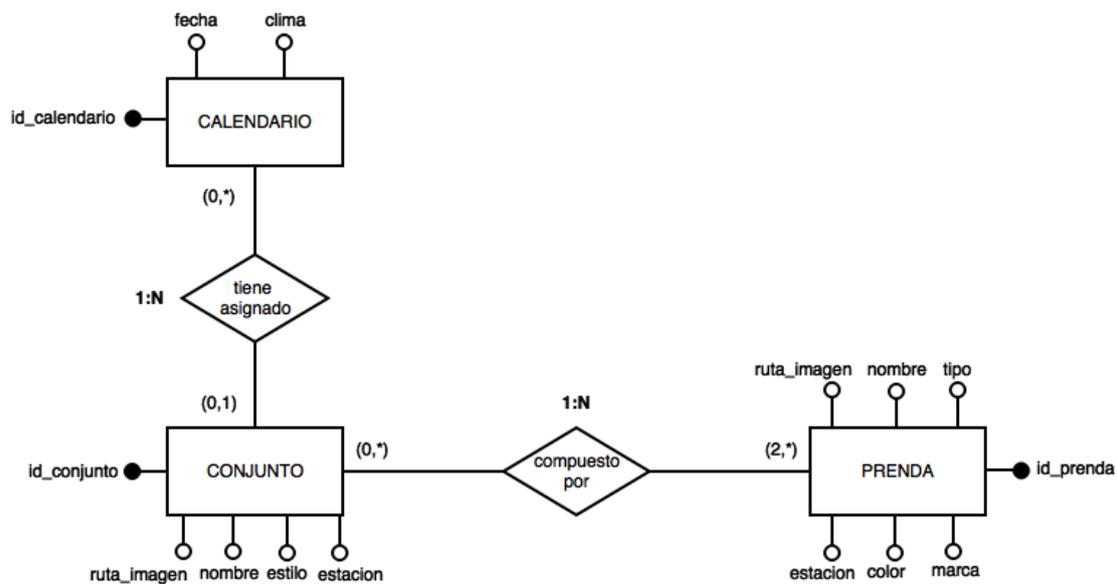
Los requisitos de información describen la información que debe almacenar y gestionar el sistema para dar soporte a los procesos de negocio.

- **RINF-01:** En la creación de una nueva prenda, el sistema debe registrar la imagen recortada, junto con otros datos como el identificador de la prenda, un nombre para que el usuario la identifique, su tipo (si es parte de arriba, de abajo, de una pieza o calzado), la estación en la que se usa, su color y marca.

- **RINF-02:** Al crear un nuevo conjunto, el sistema registrará la imagen formada por las imágenes de las prendas, junto con otros datos como el identificador del conjunto, los identificadores de las prendas por las que está compuesto, un nombre para que el usuario lo identifique, su estilo para qué tipo de ocasión está creado, y la estación en la que se usa.
- **RINF-03:** Al asignar un conjunto a una fecha del calendario, el sistema debe registrar el identificador del calendario, también el identificador del conjunto, la fecha en la que se le asigna, y el clima de dicho día.

### 3.5.1. Modelo conceptual

Para ilustrar los diferentes tipos de entidades y relaciones que se pueden dar, se crea el modelo Entidad-Relación, ya que este modelo se utiliza como una descripción conceptual de la base de datos.



**Ilustración 13. Diagrama de Entidad-Relación.**

Las suposiciones en las que se basa el modelo conceptual son principalmente las reglas de negocio **RN-02** y **RN-03**, que han sido anteriormente descritas en el capítulo 1.

## 3.5.2. Diccionario de datos

El diccionario de datos describe de forma detallada las entidades de datos utilizadas en el sistema.

ENTIDAD	ATRIBUTO	VALOR	DESCRIPCION
<b>prenda:</b> Cada una de las partes que componen la vestimenta de una persona.	<i>id_prenda</i>	Entero positivo.	Identificador único de la prenda.
	<i>ruta_imagen</i>	Cadena de caracteres.	Ruta donde se encuentra la imagen de la prenda.
	<i>nombre</i>	Cadena de caracteres.	Nombre con el que identifica el usuario a la prenda.
	<i>tipo</i>	Enumerado.	Prenda de parte de arriba, parte de abajo o calzado.
	<i>estación</i>	Enumerado.	Estación del año en el que se usa la prenda, puede ser: primavera, verano, otoño o invierno.
	<i>color</i>	Enumerado.	La gama de color en la que se encuentra la prenda entre los colores preestablecidos.
	<i>marca</i>	Cadena de caracteres.	Lugar o compañía que vende la prenda.
<b>conjunto:</b> Juego de vestir formado por varias prendas.	<i>id_conjunto</i>	Entero positivo.	Identificador único del conjunto.
	<i>ruta_imagen</i>	Cadena de caracteres.	Ruta en la que se encuentra la imagen del conjunto.
	<i>nombre</i>	Cadena de caracteres.	Nombre con el que identifica el usuario al conjunto.
	<i>estilo</i>	Enumerado.	Tipo de ocasión para el que se suele usar el conjunto entre los preestablecidos.
	<i>estacion</i>	Enumerado.	Estación del año en el que se usa el conjunto, puede ser: primavera, verano, otoño o invierno.
<b>calendario:</b> Distribución de conjuntos en distintas fechas a lo largo de varios años.	<i>id_calendario</i>	Entero positivo.	Identificador único del conjunto en el calendario.
	<i>fecha</i>	Fecha.	Día al que se le asigna un conjunto.
	<i>clima</i>	Enumerado.	Condiciones atmosféricas que caracterizan un día entre los preestablecidos.

**Tabla 37. Diccionario de datos.**

## 3.6. Requisitos de interfaces externas

- **RIE-01:** El sistema se relacionará con el servicio del clima a través de la API de Forecast.

## 3.7. Requisitos no funcionales

- **RNF-01:** El sistema debe funcionar para más del 80% de dispositivos con sistema operativo Android.
- **RNF-02:** El sistema deberá resultar sencillo de usar y se podrá acceder rápidamente a cada función del sistema, se estima como rápido menos de 1 segundo.
- **RNF-03:** El sistema deberá ser capaz de capturar todos los posibles fallos y tratarlos adecuadamente.
- **RNF-04:** El software debe ser lo más modular posible con el objetivo de que sea reusable.
- **RNF-05:** El sistema debe ser eficiente a la hora de realizar consultas y hacer uso de sus recursos con el objetivo de tener un alto rendimiento, que el usuario no espere más de 1 segundo en ver resultados.
- **RNF-06:** El sistema debe estar disponible las 24 horas del día, 7 días de la semana y debe poder utilizarse toda su funcionalidad sin tener acceso a Internet.
- **RNF-07:** Los requisitos hardware necesarios para el sistema deben ser fácilmente alcanzables, como un dispositivo móvil inteligente que tenga un sistema operativo del que disponga la mayor parte de usuarios de estos dispositivos<sup>3</sup> (Android), y una memoria interna de unos 6 MB disponibles o, en su defecto, almacenamiento externo con el mencionado espacio disponible.
- **RNF-08:** Debe ser fácil de mantener para poder añadir nuevas funcionalidades sin que exista detrimento de la calidad o del funcionamiento ya alcanzado.
- **RNF-09:** El sistema protegerá contra la inserción, modificación o borrado no autorizado de datos.
- **RNF-10:** Si el usuario no quiere introducir el clima manualmente, se necesitará de conexión a Internet, para hacer uso del servicio que obtiene el clima de la zona donde se encuentra el dispositivo automáticamente.

---

<sup>3</sup> <http://www.ditrendia.es/wp-content/uploads/2014/07/Ditrendia-Informe-Mobile-en-Espa%C3%B1a-y-en-el-Mundo.pdf>

## 3.8. Requisitos de internacionalización y localización

- **RIL-01:** El sistema deberá soportar los idiomas Español e Inglés.

# Capítulo 4

## Diseño

---



La etapa de diseño permite refinar el modelo del análisis, hasta obtener un diseño del sistema adecuado, considerando los requisitos no funcionales y restricciones del entorno, para después pasar a la implementación. Además esta etapa, permite tener una idea más clara del sistema, ya que elimina parte de la abstracción de la etapa anterior.

## 4.1. Tecnologías

El Android SDK y el JDK de Java son necesarios para realizar la programación Android en Java, que se realizará en el entorno de desarrollo de Eclipse.

Debido a que la aplicación es para dispositivos móviles hay que tener en cuenta sus limitaciones y se han tomado las siguientes decisiones:

- La **persistencia** se almacenará en el **propio dispositivo** lo que conlleva a espacio limitado, acceso rápido y alto coste en consultas complejas. La ventaja, es que funciona sin tener conexión a Internet y se puede tener completa **disponibilidad**. Para centralizar las tareas de persistencia (mejor mantenimiento) se usará OrmLite, que es un ORM básico para aplicaciones Java. Se integra con SQLite. Sirve para manipular la Base de Datos. Entre sus beneficios se encuentra la separación de la lógica de aplicación (conveniente para el modelo por capas), gestión de persistencia y evolución o migración de la persistencia sin cambiar esta lógica de aplicación.
- Si un dispositivo cuenta con **almacenamiento externo** se almacenará allí los datos de la aplicación, para que dispositivos con pocas capacidades de almacenamiento internas puedan guardarlo.
- Debido a las limitaciones de rendimiento de estos dispositivos al llevar a cabo ciertas operaciones se utilizará procesamiento **en segundo plano**.

## 4.2. Arquitectura lógica

La arquitectura es sencilla, y se ha usado el patrón de diseño estructurado en capas. Este tipo de arquitecturas permiten por un lado dividir los equipos de trabajo más fácilmente, y por otro desarrollar componentes o librerías que puedan ser reutilizadas en un futuro para otros proyectos.

Con esta arquitectura el software se divide en 3 niveles diferentes (ver Ilustración 13):

- 1) **Capa de presentación**, es la que proporciona la interfaz de usuario de la aplicación. Se comunica con la capa de negocio.
- 2) **Capa de negocio** (lógica del dominio), es aquella que realiza las reglas de negocio y se encuentra en el nivel intermedio que comunica con las capas de presentación y de datos. Contiene los procesos a realizar con la información recibida desde las anteriores capas.
- 3) **Capa de acceso a datos**, es la capa que permite acceder a las fuentes de datos. Allí se encuentran los datos que persisten. Se comunica con la capa de negocio facilitando los datos o recibéndolos para realizar las operaciones básicas, llamadas CRUD (Create-Read-Update and Delete), que se realizan sobre cualquier fuente de datos.

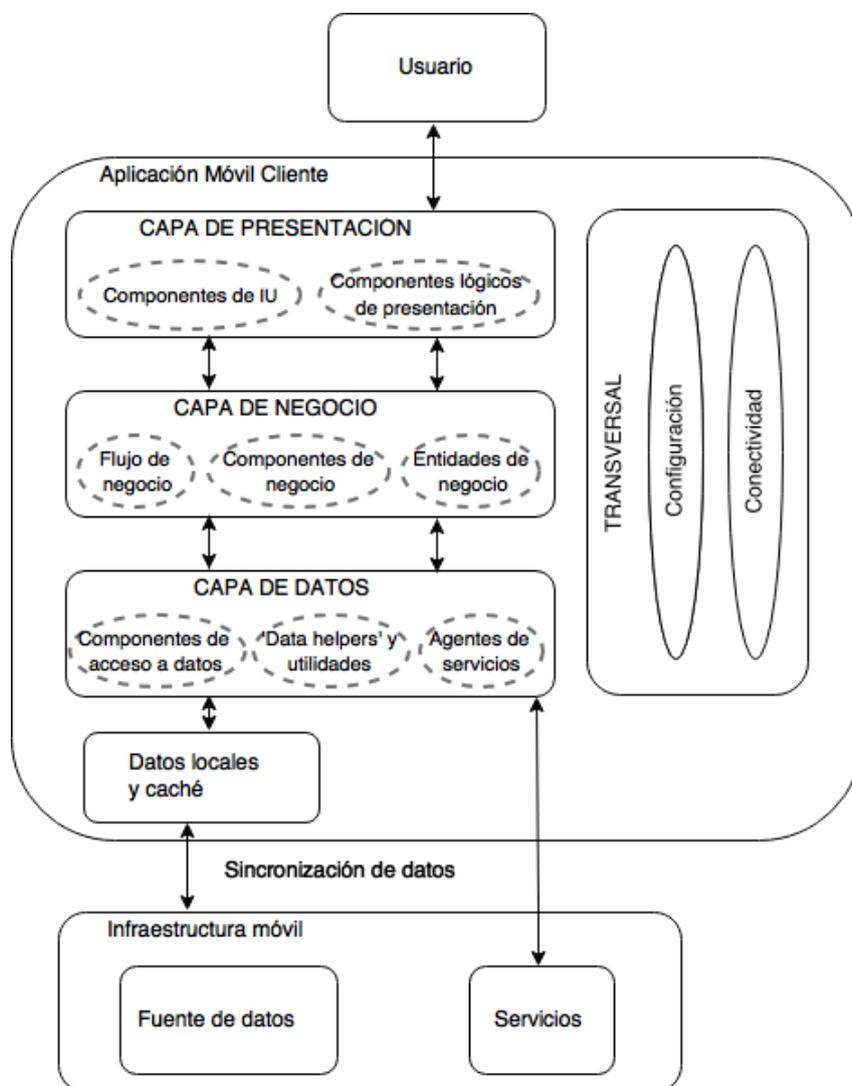


Ilustración 14. Diagrama de la arquitectura lógica.

## 4.3. Arquitectura física

La arquitectura física en su gran mayoría se limita al dispositivo móvil, lo que permite que esté siempre disponible, y se evitan latencias al no realizar peticiones externas. Pero también puede dar problemas, ya que si se da un fallo en el propio dispositivo que daña los datos, estos no se podrán recuperar debido a ser, el terminal, el único soporte.

Por otro lado se ha añadido una pequeña funcionalidad para solicitar datos a un servicio externo, el clima. Aunque, de todas formas, si no se dispone de conexión a Internet se puede introducir manualmente.

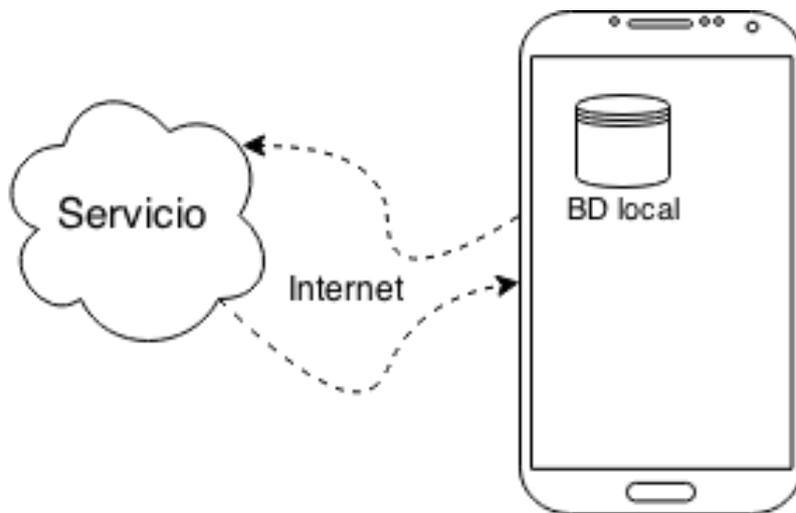


Ilustración 15. Diagrama de la arquitectura física.

## 4.4. Clases de diseño

Este tipo de diagrama describe la estructura del sistema, mostrando las clases, con sus atributos y métodos correspondientes, y las relaciones entre ellas. El diagrama de clases de diseño es elaborado para tener en cuenta los detalles concretos de la implementación del sistema a desarrollar.

El diagrama ampliado se encuentra en el ANEXO II.

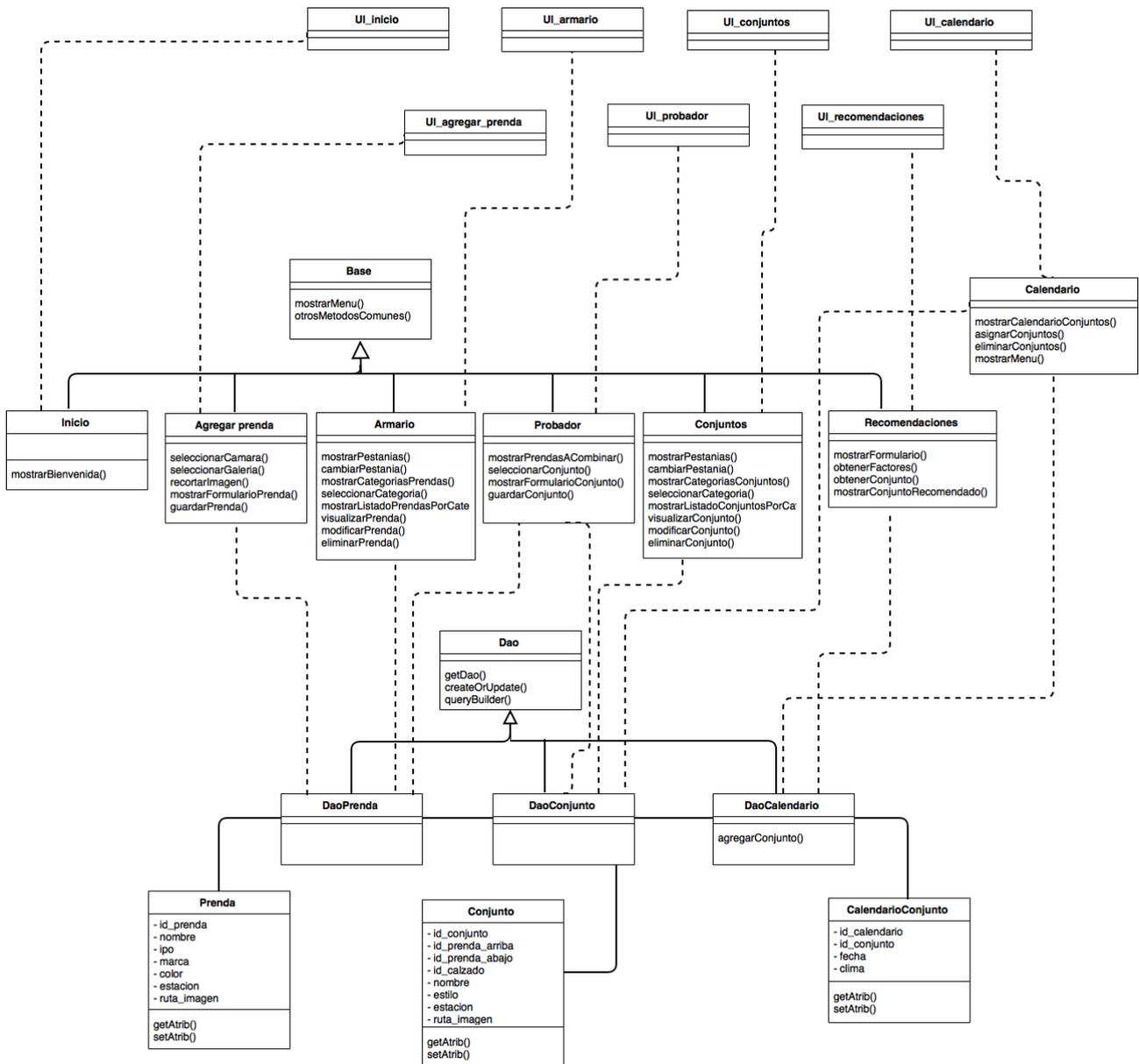
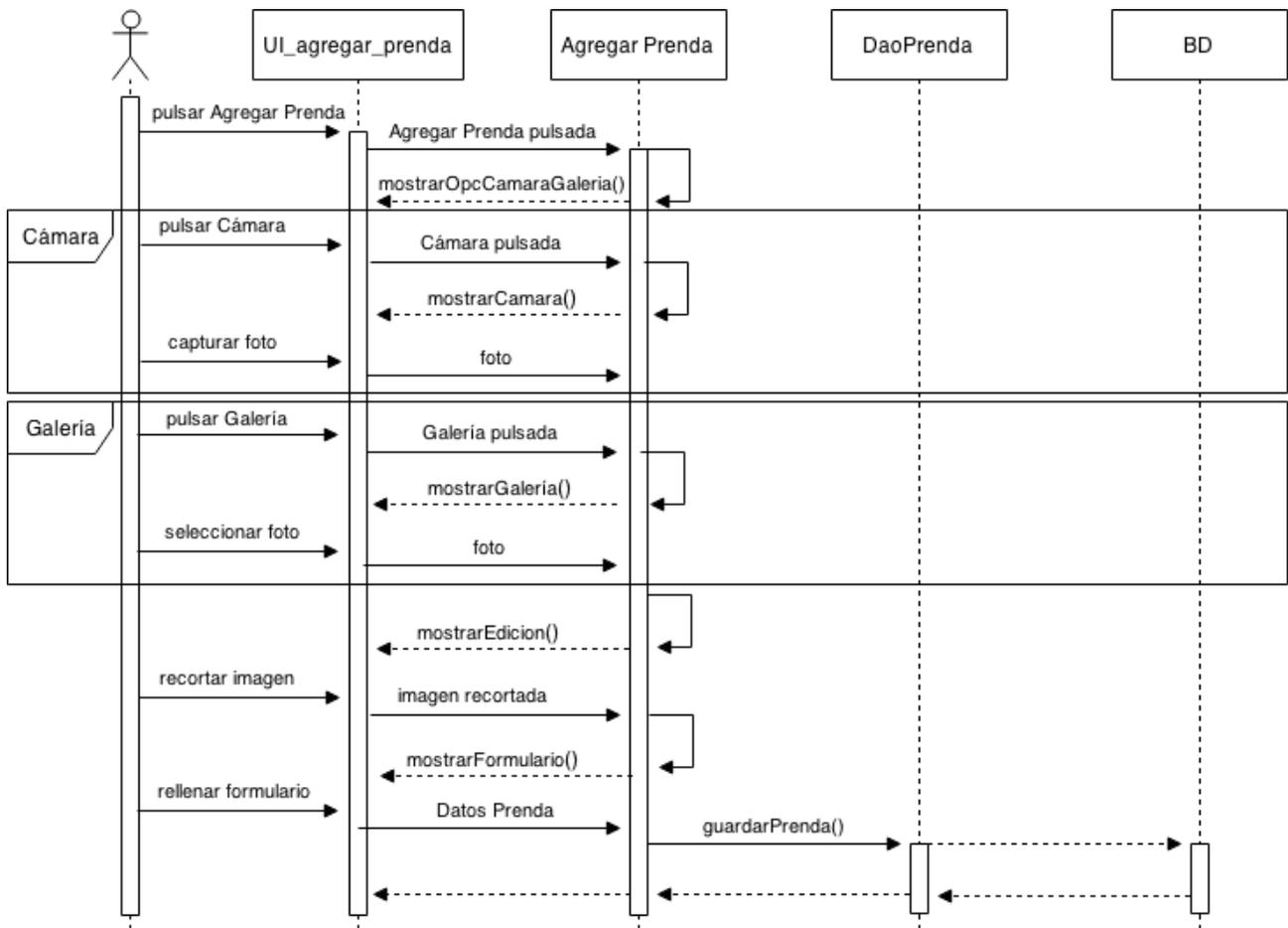


Ilustración 16. Diagrama de clases de diseño.

## 4.5. Diagramas de secuencia

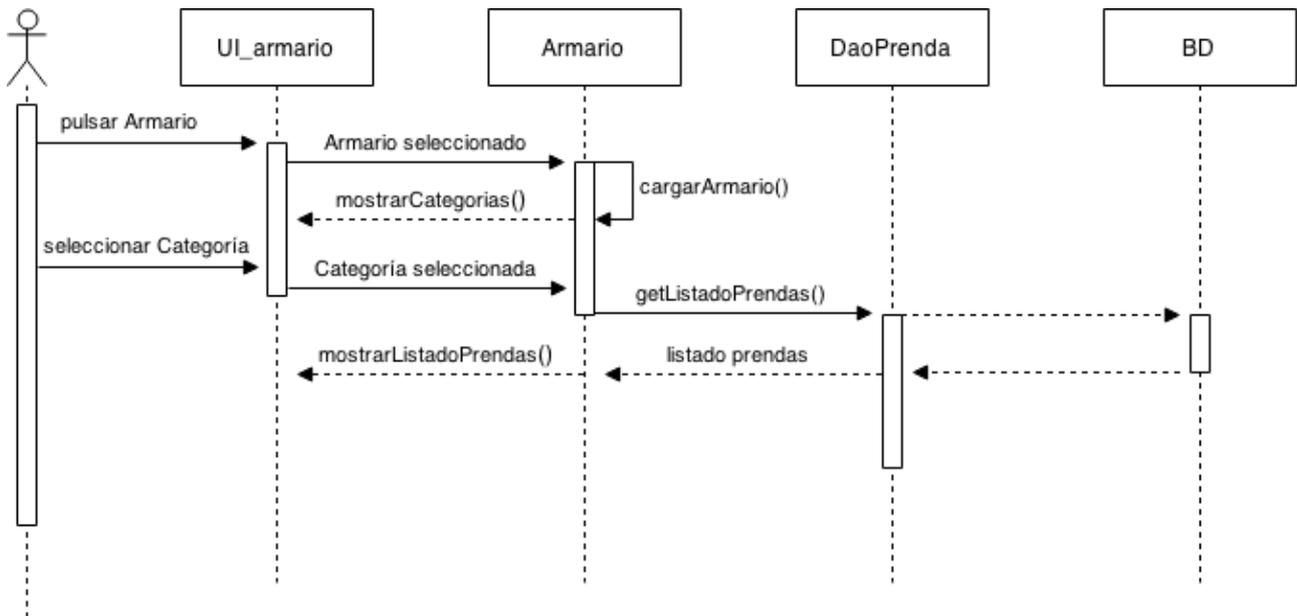
Los diagramas de secuencia muestran cómo se comunican las clases o módulos que forman parte de la aplicación y la secuencia de mensajes que realizan entre ellos durante un escenario concreto. A continuación se mostrarán los diagramas de secuencia de los casos de uso, que se especificaron en el apartado de análisis, más relevantes:

- Diagrama de secuencia del **CU-01**: Crear Prenda. Contiene a su vez el **CU-02**: Capturar Imagen y el **CU-03**: Seleccionar Imagen Galería.



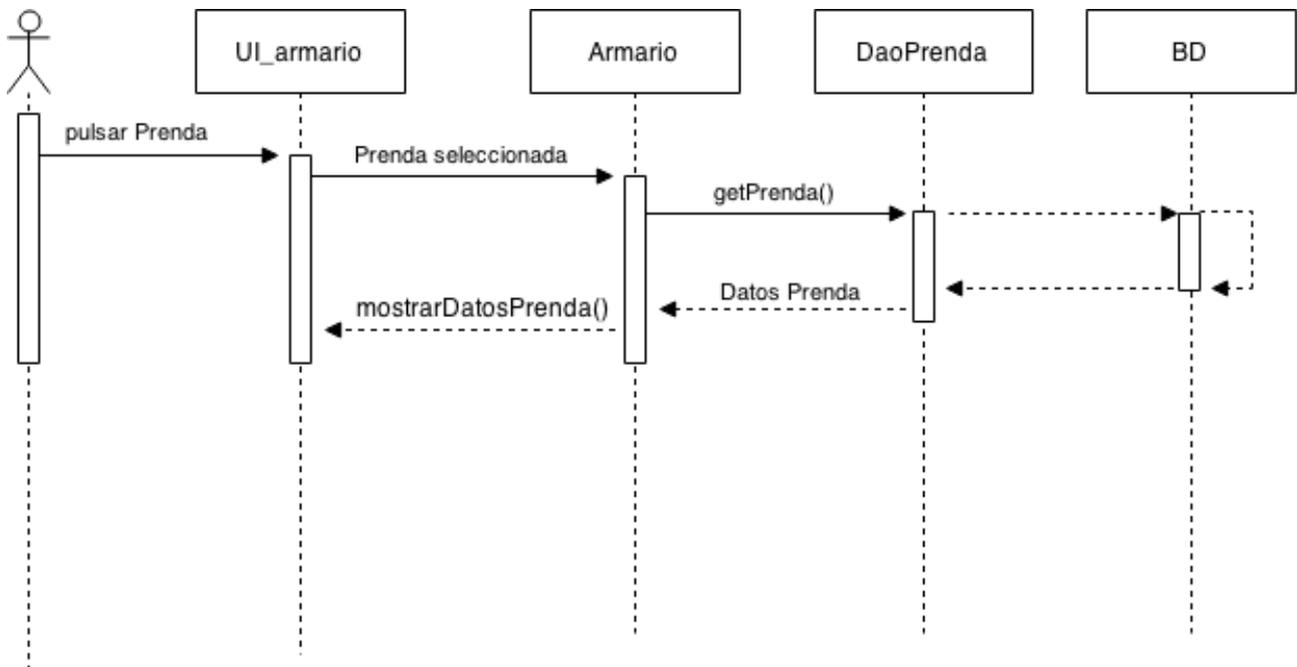
**Ilustración 17. Diagrama de secuencia del CU-01.**

- Diagrama de secuencia del **CU-04**: Visualizar Armario.



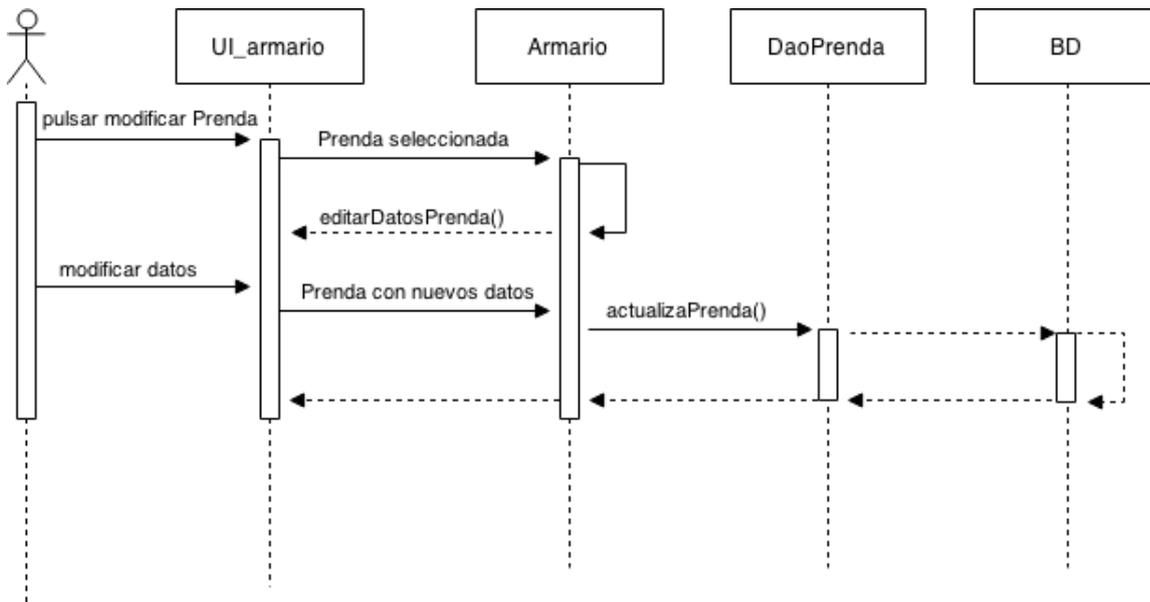
**Ilustración 18. Diagrama de secuencia del CU-04.**

- Diagrama de secuencia del **CU-05**: Visualizar Prenda.



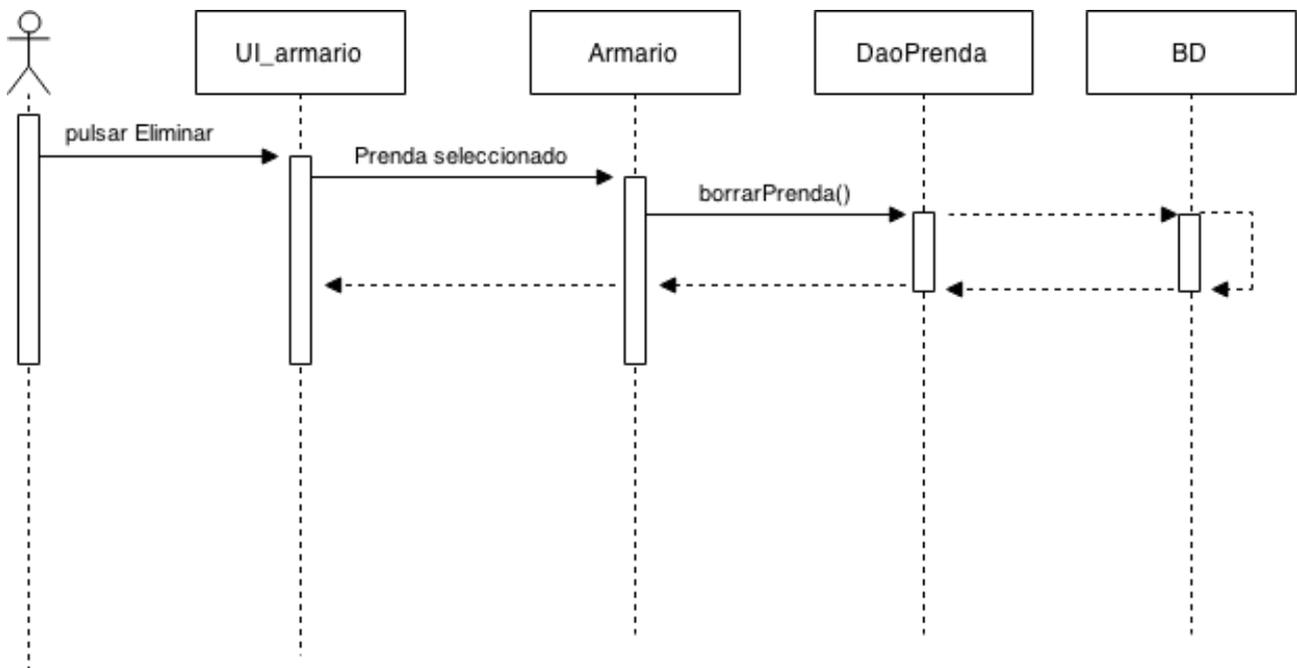
**Ilustración 19. Diagrama de secuencia del CU-05.**

- Diagrama de secuencia del **CU-06**: Modificar Datos Prenda.



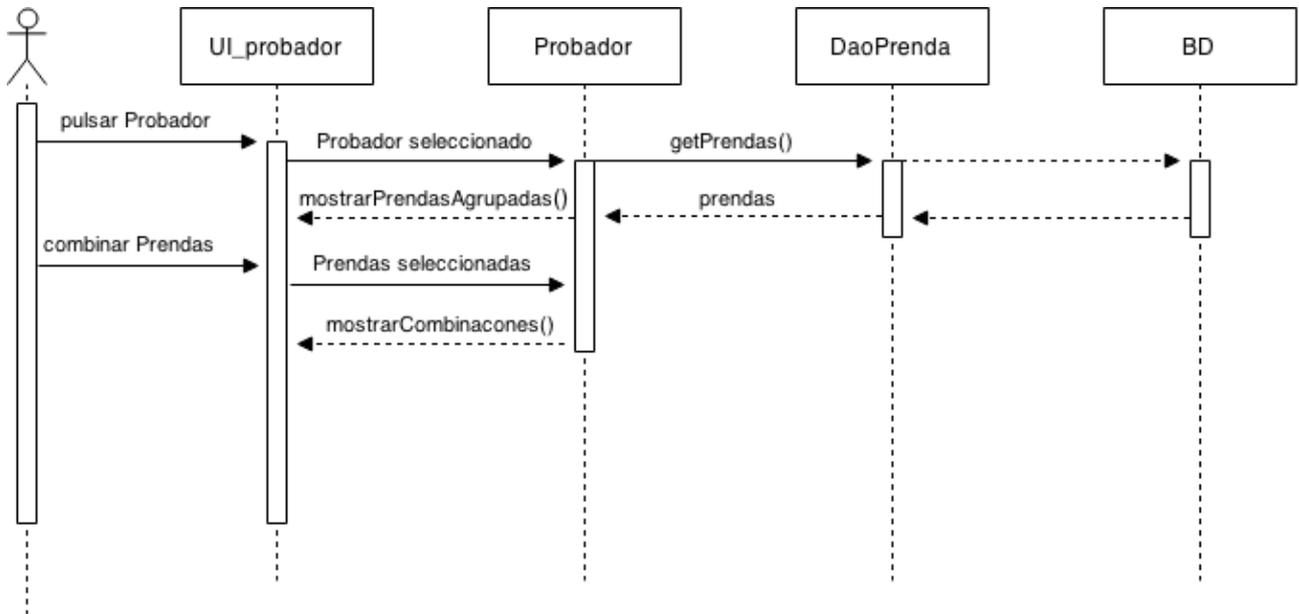
**Ilustración 20. Diagrama de secuencia del CU-06.**

- Diagrama de secuencia del **CU-07**: Eliminar Prenda.



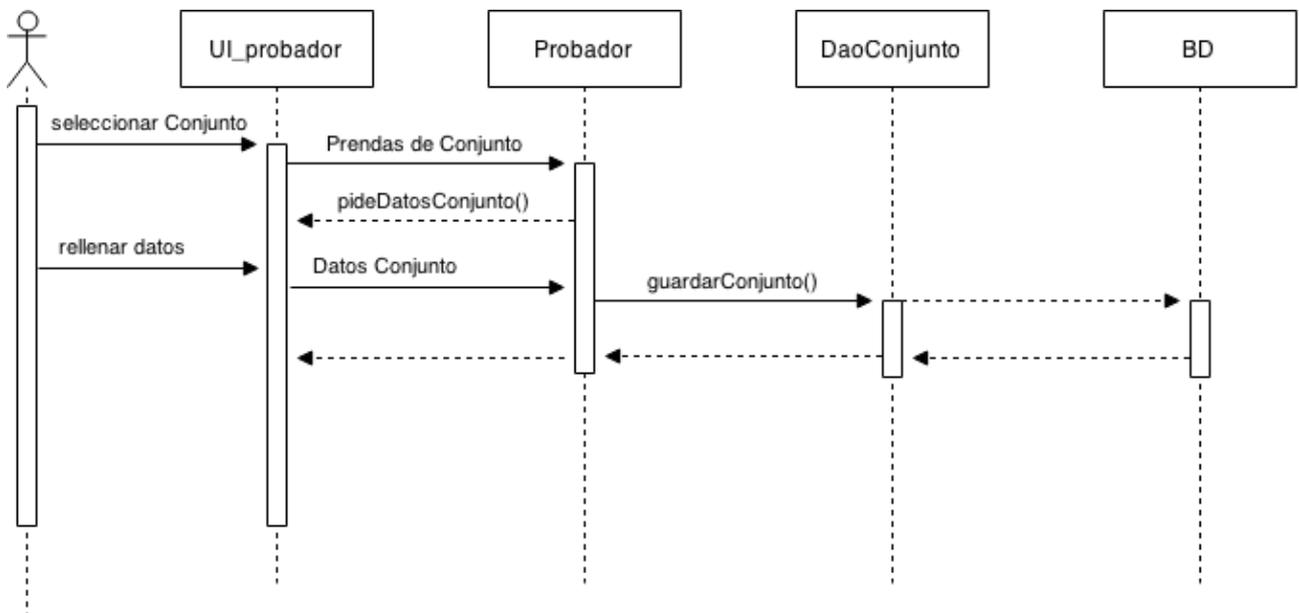
**Ilustración 21. Diagrama de secuencia del CU-07.**

- Diagrama de secuencia del **CU-08**: Visualizar Probador.



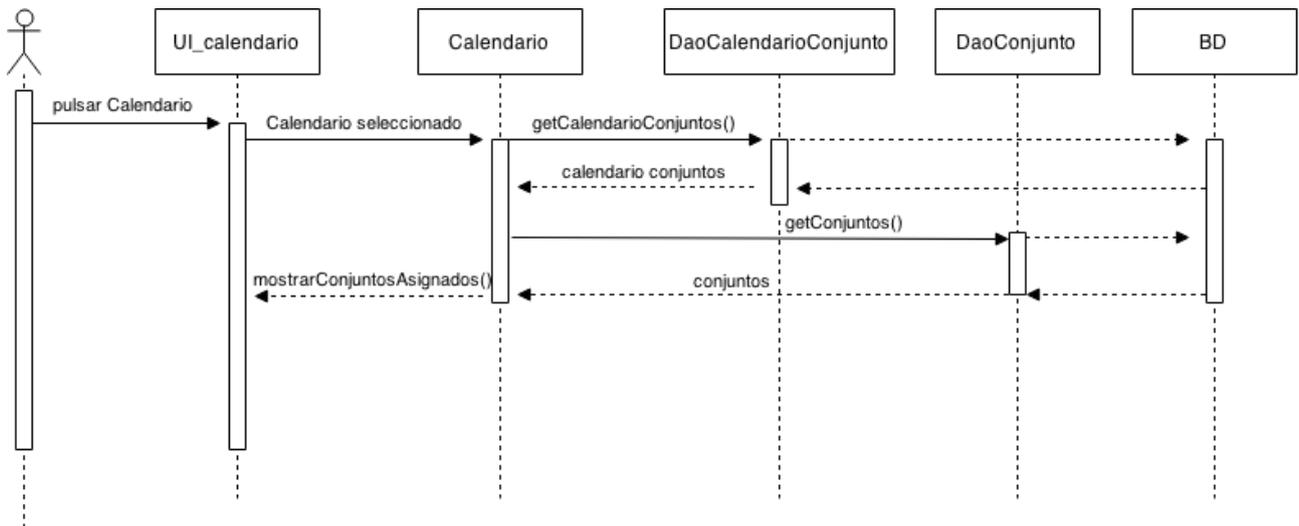
**Ilustración 22.** Diagrama de secuencia del CU-08.

- Diagrama de secuencia del **CU-09**: Seleccionar Conjunto.



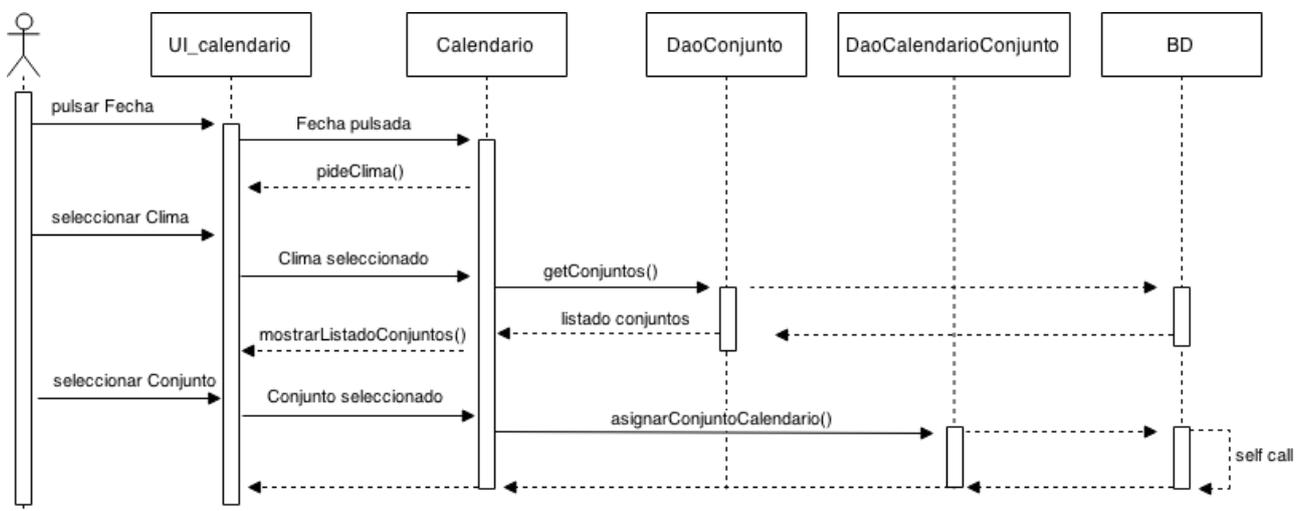
**Ilustración 23.** Diagrama de secuencia del CU-09.

- Diagrama de secuencia del **CU-14**: Visualizar Calendario.



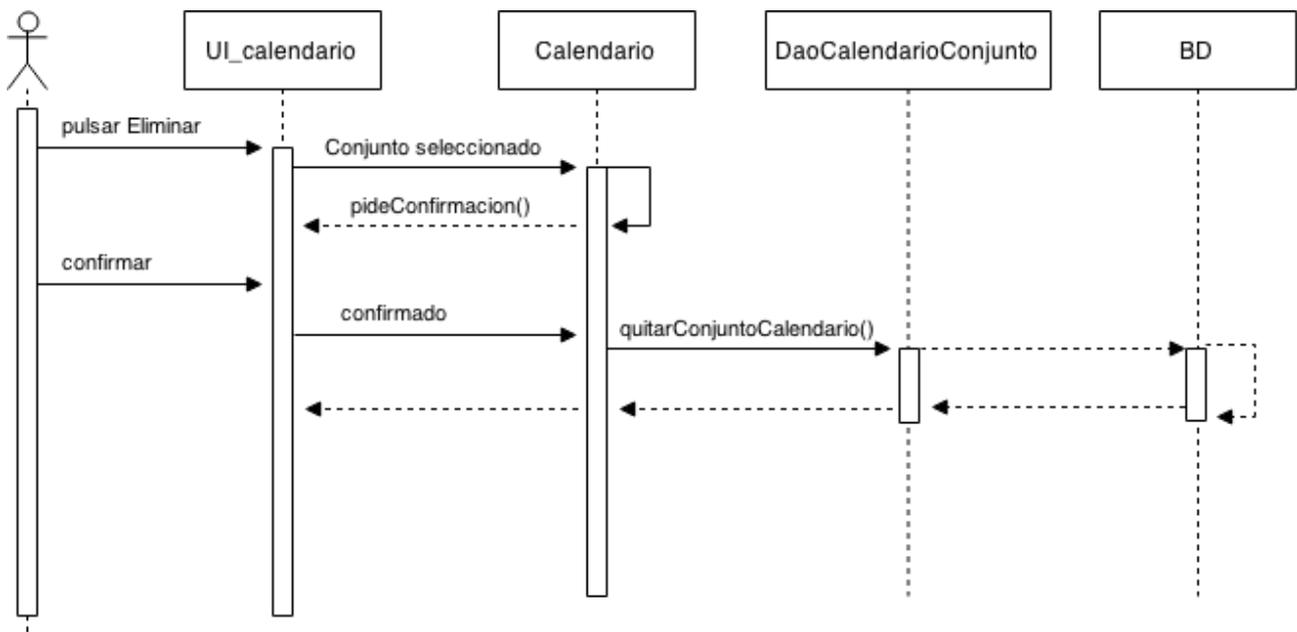
**Ilustración 24. Diagrama de secuencia del CU-14.**

- Diagrama de secuencia del **CU-15**: Asignar Conjunto.



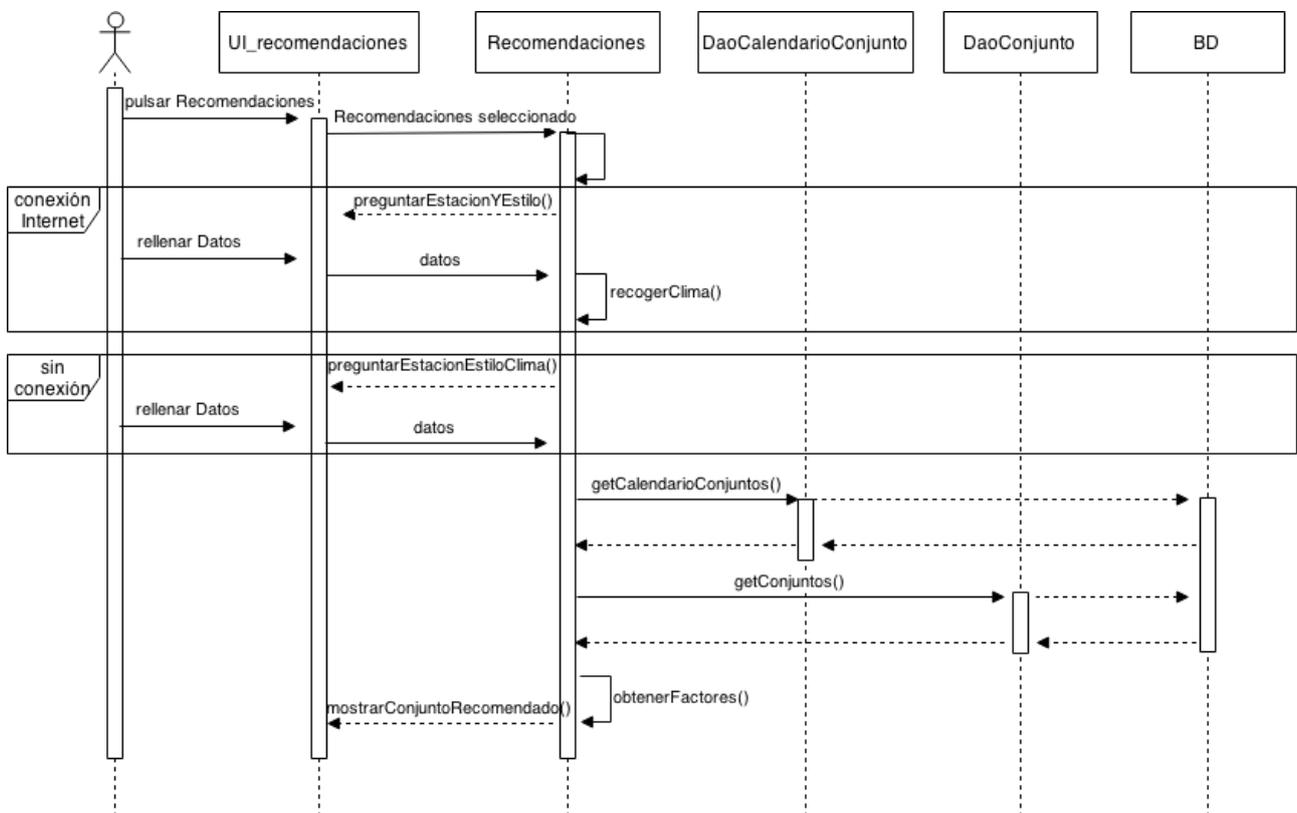
**Ilustración 25. Diagrama de secuencia del CU-15.**

- Diagrama de secuencia del **CU-16**: Eliminar Conjunto.



**Ilustración 26.** Diagrama de secuencia del CU-16.

- Diagrama de secuencia del **CU-17**: Pedir Recomendación.



**Ilustración 27.** Diagrama de secuencia del CU-17.

## 4.6. Factores que determinan la recomendación de conjuntos

A la hora de realizar recomendaciones de conjuntos al usuario en base a los conjuntos previamente almacenados en el calendario, se ha de tener en cuenta: los atributos propios del conjunto, su estilo y su estación, pero también se ha de considerar el valor del clima del conjunto en un día determinado del calendario. Gracias a esos tres valores, el sistema obtiene el conjunto a recomendar, que es el que tiene los valores más cercanos posibles a los que pide el usuario.

Para llegar a una ponderación adecuada, se han establecido las siguientes prioridades a estos valores:

El estilo (**0,5**) sería el que tiene mayor importancia, debido a que hoy en día se suele utilizar algunos conjuntos sin que influya mucho la época del año en el que se encuentre, y la mayoría de la gente discierne entre que vestir en un determinado ambiente, y que no. Por ejemplo, no ponerse un chándal para ir a una fiesta.

La estación (**0,3**) sería la siguiente de mayor importancia, debido a que está más orientada a las temperaturas de una época del año, y se suele priorizar más, por ejemplo, que hay buena temperatura aunque existan nubes, por lo que el clima estaría en último lugar (**0,2**).

Para obtener el factor de estas características, se ha valorado la similitud entre los valores que pueden darse, también de 0 a 1. Se muestra un ejemplo de cada uno junto a una tabla con todas las ponderaciones dadas.

- Obtener factor de **estación**: Si la estación seleccionada es otoño, el factor de similitud obtenido con un conjunto de primavera es 0,7.

	Primavera	Verano	Otoño	Invierno
Primavera	1	0,5	0,7	0,25
Verano	0,5	1	0,5	0
Otoño	0,7	0,5	1	0,5
Invierno	0,25	0	0,5	1

Tabla 38. Ponderaciones de Estación.

- Obtener factor de **estilo**: Si el estilo seleccionado es para trabajar, el factor de similitud obtenido con un conjunto de fiesta es 0.

	Casual	Formal	Deporte	Trabajo	Fiesta	Otro
Casual	1	0	0	0	0	0
Formal	0	1	0	0	0	0
Deporte	0	0	1	0	0	0
Trabajo	0	0	0	1	0	0
Fiesta	0	0	0	0	1	0
Otro	0	0	0	0	0	1

**Tabla 39. Ponderaciones de Estilo.**

- Obtener factor de **clima**: Si el clima seleccionado es lluvia, el factor de similitud obtenido con un conjunto usado en un día nublado es 0,8.

	Día despejado	Noche despejada	Lluvioso	Nieve	Aguanieve	Ventoso	Niebla	Nublado	Día parcialmente nublado	Noche parcialmente nublada	Granizo	Tormentoso	Tornado
Día despejado	1	0,9	0	0	0	0,3	0	0	0,6	0,5	0	0	0
Noche despejada	0,9	1	0	0	0	0,3	0	0	0,5	0,6	0	0	0
Lluvioso	0	0	1	0,9	0,9	0,9	0,8	0,8	0	0	0,9	0,7	0,8
Nieve	0	0	0,9	1	0,9	0,9	0,7	0,4	0	0	0,9	0	0,8
Aguanieve	0	0	0,9	0,9	1	0,9	0,8	0,8	0	0	0,9	0,7	0
Ventoso	0,3	0,3	0,9	0,9	0,9	1	0	0,7	0,6	0,5	0,5	0,6	0,9
Niebla	0	0	0,8	0,7	0,8	0	1	0,6	0	0	0,8	0	0
Nublado	0	0	0,8	0,4	0,8	0,7	0,6	1	0,8	0,8	0,6	0,8	0,9
Día parcialmente nublado	0,6	0,5	0	0	0	0,6	0	0,8	1	0,9	0	0	0
Noche parcialmente nublada	0,5	0,6	0	0	0	0,5	0	0,8	0,9	1	0	0,3	0,3
Granizo	0	0	0,9	0,9	0,9	0,5	0,8	0,6	0	0	1	0,7	0,8
Tormentoso	0	0	0,7	0	0,7	0,9	0	0,8	0,3	0,3	0,7	1	0,8
Tornado	0	0	0,8	0,8	0	0,9	0	0,6	0,3	0,3	0,7	0,8	1

**Tabla 40. Ponderaciones de Clima.**

Estos valores se eligen según el parecido de vestuario entre un clima y otro, no entre la similitud del propio clima. Además han sido tomados en base a un criterio personal, y se pueden llegar a modificar, si así se estima, de forma sencilla.

## 4.7. Diseño de la interfaz

El aspecto elegido para la aplicación tiene que ser un diseño sencillo (debido el reducido tamaño de los dispositivos móviles) y que resulte intuitivo y fácil de usar para cualquier persona.

En las siguientes tablas se presenta la estructura y distribución de los elementos de las pantallas principales que forman la aplicación.

Nombre	inicio.xml
Descripción	Pantalla principal que muestra un mensaje de bienvenida y una barra de menú en la que se puede acceder a los distintos módulos. Esta barra está presente en las siguientes interfaces.
Activación	Al acceder a la aplicación.
Boceto	
Eventos	agregarPrenda(), armarioActivity(), probadorActivity(), conjuntosActivity(), calendarioActivity(), recomendacionesActivity()

Tabla 41. Diseño de la interfaz inicio.xml.

<b>Nombre</b>	armario.xml/conjuntos.xml
<b>Descripción</b>	Pantalla en la que se pueden seleccionar varias pestañas para visualizar las prendas/conjuntos en base a distintas categorías. Está compuesta por una barra deslizable de pestañas para acceder a ellas de una forma más sencilla.
<b>Activación</b>	Al pulsar el botón de Armario/Conjuntos.
<b>Boceto</b>	
<b>Eventos</b>	agregarPrenda(), armarioActivity(), probadorActivity(), conjuntosActivity(), calendarioActivity(), recomendacionesActivity(), seleccionarCategoria()

**Tabla 42. Diseño de la interfaz armario.xml.**

<b>Nombre</b>	<b>probador.xml</b>
<b>Descripción</b>	Pantalla en la que se puede visualizar la combinación de las prendas y crear un nuevo conjunto.
<b>Activación</b>	Al pulsar el botón de espejo.
<b>Boceto</b>	
<b>Eventos</b>	agregarPrenda(), armarioActivity(), probadorActivity(), conjuntosActivity(), calendarioActivity(), recomendacionesActivity(), seleccionarConjunto()

**Tabla 43. Diseño de la interfaz probador.xml.**

<b>Nombre</b>	agregarprenda.xml/agregarconjuntos.xml
<b>Descripción</b>	Pantalla en la que se puede visualizar la imagen de la prenda/conjunto y los datos referentes a ella.
<b>Activación</b>	Al pulsar el botón de Agregar Prenda/Seleccionar Conjunto.
<b>Boceto</b>	
<b>Eventos</b>	agregarPrenda(), armarioActivity(), probadorActivity(), conjuntosActivity(), calendarioActivity(), recomendacionesActivity(), guardarPrenda(), guardarConjunto()

**Tabla 44. Diseño de la interfaz conjuntos.xml.**



# Capítulo 5

## Implementación y pruebas

---



Una vez se tienen claros los aspectos principales sobre el análisis y el diseño, se puede dar paso a la implementación del sistema, pues se debe tener el conocimiento necesario para llevarlo a cabo.

## 5.1. Introducción a Android

Como ya se ha mencionado en capítulos anteriores, Android es un sistema operativo y una plataforma software, basado en Linux para dispositivos móviles. Android permite programar en Java aplicaciones sobre una máquina virtual Dalvik (una variación de la máquina de Java con compilación en tiempo de ejecución). Cada aplicación es independiente, se ejecuta en su propio proceso de Linux y cada proceso tiene su propia máquina virtual. Las aplicaciones son empaquetadas en Android Package (.apk).

Dado que Android está basado en el núcleo de Linux, tiene acceso a sus recursos, pudiendo gestionarlo, gracias a que se encuentra en una capa por encima del Kernel, accediendo así a recursos como los controladores de pantalla, cámara, memoria flash... En la siguiente imagen se muestran las capas que conforman el sistema operativo Android.

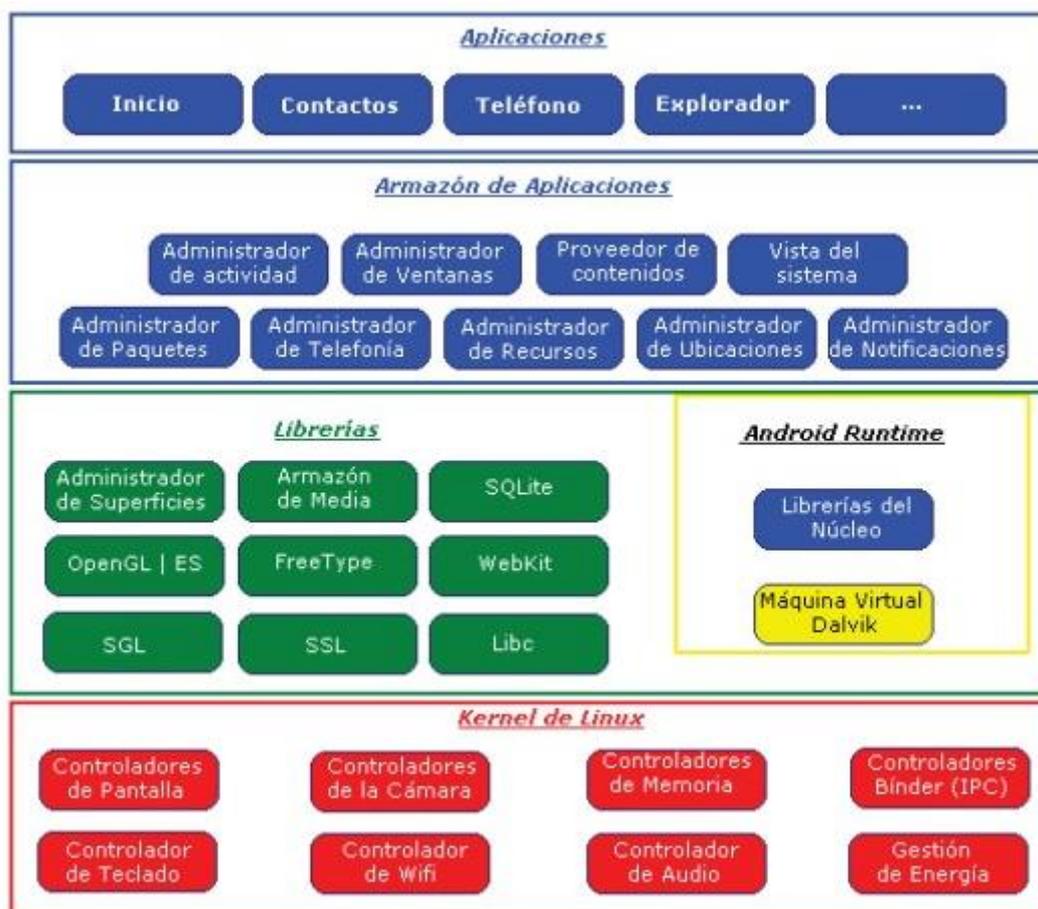


Ilustración 28. Arquitectura de Android.

En la imagen se distinguen claramente cada una de las capas: la que forma parte del propio Kernel de Linux, donde Android puede acceder a diferentes controladores, las librerías creadas para el desarrollo de aplicaciones Android, la capa que organiza los diferentes administradores de recursos, y por último, la capa de las aplicaciones a las que tiene acceso.

Una de las características principales del diseño en Android es la reutilización de componentes entre las aplicaciones, es decir, dos aplicaciones diferentes pueden utilizar un mismo componente, aunque esté en otra aplicación, para así, evitar la repetición innecesaria de código, y la consiguiente ocupación de espacio.

### 5.1.1. Componentes de una aplicación Android

Los componentes son los elementos básicos con los que se construyen el proyecto, son los siguientes:

**Activity:** Las actividades (activities) representan es el componente principal encargado de mostrar al usuario la interfaz gráfica de una aplicación Android., es decir, una actividad sería el equivalente a una ventana, y es el medio de comunicación entre la aplicación y el usuario. Los elementos que se muestran en ella deben ser definidos en el fichero xml que llevan asociado (que se guarda en ./res/layout) para poder ser tratados en la clase MainActivity.class, que hereda de la clase Activity. Dentro del fichero xml asociado a la actividad, se definen los elementos tales como ubicación de los elementos en la pantalla (layouts), botones, textos, checkbox, etc.

**View:** Las vistas (view) son los componentes básicos con los que se construye la interfaz gráfica de la aplicación, análoga por ejemplo a los controles de Java o .NET. Inicialmente, Android pone a nuestra disposición una gran cantidad de controles básicos, como cuadros de texto, botones, listas desplegables o imágenes, aunque también existe la posibilidad de extender la funcionalidad de estos controles básicos o crear nuestros propios controles personalizados.

**Services:** Los servicios son componentes sin interfaz gráfica que se ejecutan en segundo plano. En concepto, son similares a los servicios presentes en cualquier otro sistema operativo. Los servicios pueden realizar cualquier tipo de acciones, por ejemplo actualizar datos, lanzar notificaciones, o incluso mostrar elementos visuales (p.ej. actividades) si se necesita en algún momento la interacción con del usuario.

**Content Provider:** Un content provider es el mecanismo que se ha definido en Android para compartir datos entre aplicaciones. Mediante estos componentes es posible compartir determinados datos de nuestra aplicación sin mostrar detalles sobre su almacenamiento interno, su estructura, o su implementación. De la misma forma, nuestra aplicación podrá acceder a los datos de otra a través de los content provider que se hayan definido.

**Broadcast Receiver:** Un broadcast receiver es un componente destinado a detectar y reaccionar ante determinados mensajes o eventos globales generados por el sistema (por ejemplo: "Batería baja", "SMS recibido", "Tarjeta SD insertada",...) o por otras aplicaciones (cualquier aplicación puede generar mensajes broadcast, es decir, no dirigidos a una aplicación concreta sino a cualquiera que quiera escucharlo).

**Widget:** Los widgets son elementos visuales, normalmente interactivos, que pueden mostrarse en la pantalla principal (home screen) del dispositivo Android y recibir actualizaciones periódicas. Permiten mostrar información de la aplicación al usuario directamente sobre la pantalla principal.

**Intent:** Un intent es el elemento básico de comunicación entre los distintos componentes Android que hemos descrito anteriormente. Se pueden entender como los mensajes o peticiones que son enviados entre los distintos componentes de una aplicación o entre distintas aplicaciones. Mediante un intent se puede mostrar una actividad desde cualquier otra, iniciar un servicio, enviar un mensaje broadcast, iniciar otra aplicación, etc.

## 5.1.2. Estructura de un proyecto Android

Carpeta **/src/** Esta carpeta contendrá todo el código fuente de la aplicación, código de la interfaz gráfica, clases auxiliares, etc. Inicialmente, Eclipse creará por nosotros el código básico de la pantalla (Activity) principal de la aplicación (MainActivity en el ejemplo), y siempre bajo la estructura del paquete java definido.

Carpeta **/res/** Contiene todos los ficheros de recursos necesarios para el proyecto: imágenes, vídeos, cadenas de texto, etc. **/res/layout/** Contiene los ficheros de definición XML de las diferentes pantallas de la interfaz gráfica.

Carpeta **/gen/** Contiene una serie de elementos de código generados automáticamente al compilar el proyecto. Cada vez que generamos nuestro proyecto, la maquinaria de compilación de Android genera por nosotros una serie de ficheros fuente java dirigidos al control de los recursos de la aplicación. Dado que estos ficheros se generan automáticamente tras cada compilación del

proyecto es importante que no se modifiquen manualmente bajo ninguna circunstancia.

Carpeta **/assets/** Contiene todos los demás ficheros auxiliares necesarios para la aplicación (y que se incluirán en su propio paquete), como por ejemplo ficheros de configuración, de datos, etc. La diferencia entre los recursos incluidos en la carpeta **/res** y los incluidos en la carpeta **/assets/** es que para los primeros se generará un ID en la clase **R** y se deberá acceder a ellos con los diferentes métodos de acceso a recursos. Para los segundos sin embargo no se generarán ID y se podrá acceder a ellos por su ruta como a cualquier otro fichero del sistema. Usaremos uno u otro según las necesidades de nuestra aplicación.

Carpeta **/bin/** Contiene los elementos compilados de la aplicación y otros ficheros auxiliares. Cabe destacar el fichero con extensión ".apk", que es el ejecutable de la aplicación que se instalará en el dispositivo.

Carpeta **/libs/** Contendrá las librerías auxiliares, normalmente en formato ".jar" que utilizemos en nuestra aplicación Android.

Fichero **AndroidManifest.xml** Contiene la definición en XML de los aspectos principales de la aplicación, como por ejemplo su identificación, sus componentes, las librerías auxiliares utilizadas, o los permisos necesarios para su ejecución.

## 5.2. Estructura del proyecto

El código implementado está dividido en los siguientes paquetes y subpaquetes para que quede más organizado:

### 1) **src**

Es el lugar donde se encuentran los .java implementados. Se encuentra organizado en otros subpaquetes:

→ **com.virjc.myclothes**: Contiene los Activities principales de los que se compone la aplicación.

→ **com.virjc.myclothes.fragmentos**: Contiene todo el código que aporta de mayor funcionalidad a los Activities, y está representado mediante fragmentos.

→ **com.virjc.myclothes.BD**: Contiene todo el código referente a la Base de Datos.

→ **com.virjc.myclothes.utils**: En este paquete se encuentran aquellos elementos que optimizan y complementan el código de los paquetes mencionados anteriormente.

## 2) res

Es el lugar en el que se guardan los recursos de la aplicación.

→ **drawable**: Contiene los recursos de los que hace uso la aplicación como por ejemplo imágenes, que a su vez pueden ser divididos en otros paquetes según la resolución de pantalla que tenga el dispositivo.

→ **layout**: Contiene los siguientes ficheros .xml de la parte de la interfaz gráfica, aunque con el objetivo de personalizar más estos elementos, en muchos casos se ha llegado a su especificación en los .java ya que en el layout existen mayores restricciones.

→ **menu**: Lugar donde se encuentran los items del ActionBar.

→ **values**: Se encuentran los strings por defecto de la aplicación.

→ **values-en**: Está hecho para la internacionalización. Idioma: Inglés.

→ **values-es**: Está hecho para la internacionalización. Idioma: Español.

## 3) AndroidManifest

Es un archivo .xml que contiene la definición de los aspectos principales de la aplicación, como su identificación, sus componentes, las librerías auxiliares (AndroidCrop, AndroidImageSlider, Caldroid y Forecast) así como unos permisos para hacer uso de la cámara, escribir en la tarjeta SD del dispositivo, y también para acceder a Internet y al GPS de los que hace uso del servicio que extrae el clima.

## 5.3. Aspectos relevantes de implementación

Como se podrá leer más detalladamente en la descripción de las clases implementadas, hay varias características técnicas a destacar que aportan gran valor a los atributos de calidad mencionados en el capítulo de análisis, y merecen ser mencionadas en un apartado específico para ello.

En primer lugar, para cualquier operación de guardado como la creación de prendas y conjuntos, su modificación o eliminación, se ha utilizado una clase dinámica, que al recibir una tarea (o varias) cualesquiera que sean los parámetros dados, ejecute dicha tarea **en un hilo nuevo**. De esta manera, se consigue mayor sensación de fluidez, y no se bloqueará la pantalla al realizar dichas operaciones, otorgando mayor eficiencia y dando mayor libertad al usuario para realizar otras acciones.

En segundo lugar, se han utilizado **proyectos-librería**, para llevar a cabo funcionalidades como el recortado de la prenda (AndroidCrop), el calendario personalizable (Caldroid), el servicio para obtener el clima (Forecast) del que se puede hacer uso gratuito hasta 1.000 veces al día, y una galería de imágenes (AndroidImageSlider) que se utilizó como alternativa al obsoleto widget Gallery de Android, que daba numerosos problemas para que funcionase de la manera que se quería. La Ilustración 27 muestra la diferencia en el uso de ambos widgets.



**Ilustración 29. Antes y después del probador.**

Además, se han utilizado **Fragments** en gran parte de la aplicación en lugar de Activities para que no quede el efecto de que se cambia de pantalla para elementos pertenecientes a un mismo módulo. Por ende, la combinación de fragmentos y pestañas permiten transmitir una agradable sensación en el uso sencillo de la aplicación. Por contra, el uso de estos fragmentos ha hecho más complicada la integración con las distintas librerías.

Para llevar a cabo las operaciones de la Base de Datos se ha utilizado un modelo de mapeo objeto-relacional llamado **ORMLite**, que ofrece el uso de un patrón DAO básico, del que además se puede extender para crear un DAO con funciones propias. En este caso se tienen tres: DaoPrenda, DaoConjunto y DaoCalendarioConjunto, que hacen uso a su vez, de sus respectivas clases Prenda, Conjunto y CalendarioConjunto.

También se ha llevado a cabo una **paginación** de los elementos, en lugares como el listado de prendas y conjuntos, debido a los problemas de memoria que existían, de modo que cuando el usuario utiliza el scroll, se cargan los siguientes elementos a mostrar. Debido además, por esta limitación de memoria, se han redimensionado las imágenes, ya que si se han tomado fotografías con cámaras de mucha calidad es ilógico guardarlas tal cual si realmente no se va a hacer zoom de las imágenes, mientras que se visualicen de forma óptima en el espacio establecido.

Se ha creado además un método para **eliminar de la pila los activities** previos a los dos últimos utilizados, junto con un método para que cuando se pulse una vez atrás vuelva al activity anterior, y si se pulsa otra vez más, vuelva al MainActivity. Si se llega a pulsar una tercera vez, hará que se salga de la aplicación.

Por último, la aplicación es funcional en dos idiomas, **Inglés** y **Español**.

## 5.4. Descripción de las clases y los métodos

En este apartado se desarrolla la descripción de cada una de las clases que forman este proyecto. Dichas clases se encuentran enumeradas por orden alfabético dentro del paquete correspondiente.

Para una especificación más detallada, se debe consultar el código de la aplicación que a su vez se encuentra debidamente comentado.

→ [com.virjc.myclothes](http://com.virjc.myclothes):

### - **AgregarPrendaActivity**

Es el activity que va a contener los fragmentos relacionados con la adición de una prenda. Estos fragmentos están almacenados en su respectivo paquete y son:

FragmentoBotonesCamaraGaleria y FragmentoModificarFormulario.

Contiene el siguiente método:

- 1) Se crea la vista del activity, que está vacío, y se le añade el FragmentoBotonesCamaraGaleria, que contiene el acceso a la cámara y galería del dispositivo para crear la prenda.

## - **ArmarioActivity**

Este activity es un módulo principal de la aplicación, en él se declara una barra deslizable de pestañas, además debajo se pueden dar lugar los fragmentos siguientes: FragmentoPTipoPrendas, FragmentoPEstaciones, FragmentoPColores, FragmentoPTodas, FragmentoListadoPrendas, FragmentoVisualizarDatosPrenda, y FragmentoModificarFormularioPrenda.

Los métodos de la clase hacen lo siguiente:

- 1) Se crea la vista de la actividad con su barra de pestañas, y se asigna la pestaña que va a aparecer por defecto con su fragmento.
- 2) Se crea la barra contenedora de las pestañas y se añaden.
- 3) Se obtiene el fragmento con el id indicado, y se realiza el reemplazo de los fragmentos al seleccionar otra pestaña.

## - **BaseActivity**

Es el activity que sirve de base para el resto, ya que tiene varios métodos básicos en común. Hereda del activity que da acceso al OrmLite.

## - **CalendarioActivity**

Este Activity difiere de los demás en que no hereda de BaseActivity. Tiene que heredar de FragmentActivity, y BaseActivity no extiende de este, por lo que es necesario tener una clase intermediaria que tenga acceso al OrmLite con los FragmentActivity.

Los métodos de la clase hacen lo siguiente:

- 1) Si ya se había usado el calendario se carga el FragmentoCalendario con su estado anterior. Si no es así, se carga de cero.
- 2) Si se pulsa en un día concreto que no tiene conjunto asignado a dicho día, se pasa a una pantalla que pide el clima de dicho día, y posteriormente, a un listado de los conjuntos para que se elija uno asociado a dicho día.
- 3) En caso de que tenga conjunto asignado el día seleccionado, se pasa a la pantalla que muestra el conjunto con sus datos.

4) Al pulsar durante un tiempo, se borra el conjunto asignado.

5) Se guarda el estado actual de Caldroid.

#### - **ConjuntosActivity**

Este activity es un módulo principal de la aplicación, en él se declara una barra deslizable de pestañas, además en la parte inferior se pueden encontrar los fragmentos siguientes:

FragmentoCEstaciones.

FragmentoCEstilo.

FragmentoCTodos.

FragmentoListadoConjuntos.

FragmentoVisualizarDatosConjunto.

FragmentoModificarFormularioConjunto.

1) Se crea la vista de la actividad con su barra de pestañas, y se asigna la pestaña que va a aparecer por defecto con su fragmento.

2) Se crea la barra contenedora de las pestañas y se añaden.

3) Se obtiene el fragmento con el id indicado, y se realiza el reemplazo de los fragmentos al seleccionar otra pestaña.

#### - **MainActivity**

Es el activity principal de la aplicación y está declarado en el Manifest como el primero en mostrarse al acceder a la aplicación. Lo único que hace es mostrar su layout y tener implementado el listener de un botón para agregar una nueva prenda.

#### - **ProbadorActivity**

Es el Activity donde se crea la vista de la actividad y se asigna el FragmentoProbarConjunto.

#### - **RecomendacionesActivity**

Se crea la vista de la actividad y se le asigna el FragmentoRecomendacionesSeleccionar.

→ com.virjc.myclothes.BD:

- **CalendarioConjunto**

Es la clase que especifica la tabla de CalendarioConjunto con sus atributos (id\_calendario, id\_conjunto, fecha, clima).

- **Conjunto**

Es la clase que especifica la tabla de Conjuntos con sus atributos (id\_conjunto, nombre, estacion, estilo, id\_prenda\_arriba, id\_prenda\_abajo, id\_calzado, ruta\_imagen).

- **DaoCalendarioConjunto**

Interfaz que extiende del Dao genérico y lo especializa a CalendarioConjunto.

- **DaoCalendarioConjuntoImpl**

Implementación de DaoCalendarioConjunto.

- **DaoConjunto**

Interfaz que extiende del Dao genérico y lo especializa a Conjunto.

- **DaoConjuntoImpl**

Implementación de DaoConjunto.

- **DaoLoader**

Caché para que cuando se cree un Dao de una tabla no se vuelva a crear cada vez que se necesite porque no va a cambiar.

- **DaoPrenda**

Interfaz que extiende del Dao genérico y lo especializa a Prenda.

- **DaoPrendaImpl**

Implementación de DaoPrenda.

- **DBHelper**

Hereda del Helper que tiene ORMLite, y en el se definen los elementos propios de la BD, como las tablas que se van a tener, la versión que hay que cambiarla manualmente, y el nombre de la BD.

- **Prenda**

Es la clase que especifica la tabla de Prendas con sus atributos (id\_prenda, nombre, tipo, marca, color, estacion, ruta\_imagen).

→ com.virjc.myclothes.fragmentos:

Todos los fragmentos tienen los métodos para “inflar” la vista de su layout y también para destruir la vista una vez no se usa con el fin de ahorrar memoria.

- **FragmentoAgregarConjunto**

Es el fragmento que se utiliza para crear un nuevo conjunto. Su constructor recibe las prendas por las que se va a constituir un conjunto. Esta clase realiza los siguientes métodos:

- 1) Muestra los distintos campos de la vista.
- 2) Crea un canvas para formar la imagen del conjunto a partir de las imágenes, de cada una de las prendas que lo componen, redimensionadas.
- 3) Crea una nueva tarea para guardar el conjunto que hace lo siguiente: crea el dao del conjunto, y en un nuevo hilo se ejecuta la tarea de creación del nuevo conjunto con los datos obtenidos anteriormente y queda vinculado a los identificadores individuales de las prendas que lo componen. En este caso, viene muy bien realizar el guardado en segundo plano, ya que después de este método se realiza un cambio de actividad y si no se realiza en hilos distintos, da lugar a problemas.

- **FragmentoBotonesCamaraGaleria**

Es el fragmento que contiene las acciones de los botones que van a la cámara o a la galería de imágenes del dispositivo. Después se procede al recortado de la imagen. Es uno de los pasos a realizar para crear una nueva prenda. Esta clase realiza los siguientes métodos:

- 1) Muestra los distintos campos de la vista, se añaden los listeners a los botones, y se asignan las acciones de cada botón.
- 2) Una vez recogida la foto ya sea desde la cámara o desde la galería, se procede a recortarla haciendo uso de la librería `AndroidCrop`,
- 3) Una vez ha sido recortada la imagen, se reemplaza el fragmento por el `FragmentoModificarFormularioPrenda` que contiene el formulario de la prenda a rellenar por el usuario.

- **FragmentoCalendario**

Se trata del fragmento del calendario adaptado que contiene `CalendarioActivity`.

- **FragmentoCalendarioListadoConjuntos**

Es el fragmento que muestra un listado de los conjuntos para asignar en un día del calendario. Esta clase realiza los siguientes métodos:

- 1) Adapta la lista de los conjuntos en un grid.
- 2) Al seleccionar un conjunto, se asocia dicho conjunto con el día del calendario y el clima, que han sido seleccionados anteriormente.

- **FragmentoCalendarioSeleccionarClima**

Es el fragmento que muestra el clima de la zona en la que se encuentra el dispositivo, y que permite al usuario cambiarlo manualmente para, de esta forma, vincularlo con el conjunto y así poderle realizar recomendaciones de conjuntos en un futuro. Su constructor recibe la fecha y la consulta.

- 1) Una vez creada la vista, se localiza la zona del dispositivo y se pide el clima al servicio, el cual se muestra por pantalla.
- 2) Se recoge el clima, que puede venir del servicio o puede haberlo modificado el usuario para la fecha seleccionada y el conjunto que está por seleccionar.

- **FragmentoCalendarioVisualizarConjunto**

Es el fragmento que muestra el conjunto del día seleccionado junto con su información relativa.

- 1) Una vez creada la vista del fragmento, se asigna a los distintos TextView su correspondiente dato del conjunto en cuestión.

- **FragmentoCEstaciones**

Es el fragmento que ofrece visualizar los conjuntos según la estación del año en la que se hayan clasificado.

- 1) Una vez creada la vista, se adaptan los iconos de las categorías en un Grid, gracias a ConjuntoImageAdapter.
- 2) Cuando se selecciona una de las categorías se realiza el reemplazo de los fragmentos, y se construye la consulta para pasársela a FragmentoListadoConjuntos.

- **FragmentoCEstilo**

Es igual que el anterior, salvo que ofrece visualizar los conjuntos según su estilo u ocasión en la que se lleven.

- **FragmentoCTodos**

Es el fragmento que permite visualizar todos los conjuntos creados.

- **FragmentoListadoConjuntos**

Se trata del fragmento que muestra un listado de los conjuntos según un parámetro concreto. El constructor recibe el tipo de consulta a realizar, que ha sido generada anteriormente.

- 1) Una vez creada la vista, se adapta la lista de los conjuntos en un Grid.
- 2) Cuando se selecciona un conjunto, se realiza el reemplazo del fragmento actual por el FragmentoVisualizarDatosConjunto.

## - **FragmentoListadoPrendas**

Se trata del fragmento que muestra un listado de las prendas según un parámetro concreto. El constructor recibe el tipo de consulta a realizar, que ha sido generada anteriormente.

- 1) Una vez creada la vista, se adapta la lista de las prendas en un Grid.
- 2) Cuando se selecciona un conjunto, se realiza el reemplazo del fragmento actual por el FragmentoVisualizarDatosPrenda.

## - **FragmentoModificarFormularioConjunto**

Este fragmento permite modificar los datos de un conjunto. El constructor recibe el Conjunto a modificar.

- 1) Se muestran los distintos campos de la vista y se rellenan los campos con los datos previos del conjunto.
- 2) Una vez escritos los datos por el usuario, se capturan y se modifican en el conjunto.
- 3) Se crea el dao del conjunto, y en un nuevo hilo se ejecuta la tarea de actualización de los datos del conjunto. Es muy conveniente realizar el guardado en segundo plano ya que después de este método se realiza un cambio de actividad y si no se realiza en diferentes hilos, da lugar a problemas.

## - **FragmentoModificarFormularioPrenda**

Este fragmento permite crear una prenda o modificar una ya existente. Uno de los constructores recibe la imagen de la prenda a crear, y otro el objeto Prenda a modificar.

- 1) Se muestran los distintos campos de la vista. Si la prenda ya existía se rellenan los campos con sus datos previos.
- 2) Se definen las acciones a realizar al pulsar cada botón.
- 3) Una vez escritos los datos por el usuario, se capturan y se ponen en la prenda, ya sea creada por primera vez, o modificar una ya existente.
- 4) Se crea el dao de la prenda, y en un nuevo hilo se ejecuta la tarea de creación de la nueva prenda con los datos obtenidos anteriormente, o la actualización de los

datos de la prenda que ya existía. Resulta muy conveniente realizar el guardado en segundo plano, ya que después de este método se realiza un cambio de actividad y si no se realiza en diferentes hilos, da lugar a problemas.

#### - **FragmentoPColores**

Es el fragmento que ofrece visualizar las prendas según su color.

- 1) Una vez creada la vista, se adaptan los iconos de las categorías en un Grid, gracias a PrendaImageAdapter.
- 2) Cuando se selecciona una de las categorías se realiza el reemplazo de los fragmentos, y se construye la consulta para pasársela a FragmentoListadoPrendas.

#### - **FragmentoPEstaciones**

Es igual que el anterior, salvo que ofrece visualizar las prendas según la estación del año en la que se hayan clasificado.

#### - **FragmentoProbarConjunto**

Es el fragmento que permite combinar distintas prendas y crear un nuevo conjunto a partir de una de estas combinaciones.

- 1) Se implementan los tres métodos abstractos de Task.
- 2) Se realizan las consultas para obtener las prendas clasificadas según su tipo, ya que se distribuirán en la vista en base a ello.
- 3) Una vez, realizada las consultas, se distribuyen las prendas de tal forma que queden primero las partes de arriba, segundo las partes de abajo, y abajo de la pantalla, el calzado. Se mostrará una prenda por fila en forma de Slider. Además cada fila no debe terminar nunca de mostrarse, será como un ciclo infinito, si se acaban de mostrar las de una fila, se vuelve a empezar desde el principio.
- 4) Se inicializan todas las prendas seleccionadas.
- 5) Si la prenda a mostrar en la parte de arriba es del tipo de una sola pieza, se oculta la parte de abajo.
- 6) Si la prenda a mostrar en la parte de abajo es del tipo de una sola pieza, se oculta la parte de arriba.

- **FragmentoPTipoPrendas**

Es igual que FragmentoPEstaciones, salvo que ofrece visualizar las prendas según el tipo de prenda que sea: parte de arriba, parte de abajo, una pieza o calzado.

- **FragmentoPTodas**

Es el fragmento que permite visualizar todas las prendas creadas.

- **FragmentoRecomendacionesSeleccionar**

Es el fragmento que recomienda al usuario un conjunto en base a unas características previamente seleccionadas por el usuario.

- 1) Una vez creada la vista, se localiza el lugar del dispositivo para facilitárselo al servicio que da el clima del lugar y se muestra por pantalla.
- 2) Se le permitirá al usuario cambiar el clima del conjunto a buscar. También elegirá su estación y el estilo u ocasión del conjunto entre los preestablecidos en el sistema.
- 3) Una vez elegidos el clima, estilo y la estación, se buscan los conjuntos del calendario-conjuntos y se obtiene el factor de similitud de cada uno de ellos en cuanto al clima, estación y estilo.
- 4) Si se encuentra un conjunto de mismas características 100%, se muestra, si no, se elige aquel que se acerque más a lo establecido, y se muestra.

- **FragmentoVisualizarDatosConjunto**

Este fragmento se limita a mostrar la información del conjunto y dos botones por si se desea modificar los datos o eliminar el conjunto de la Base de Datos.

- 1) Una vez creada la vista del fragmento, se asigna a los distintos TextView su correspondiente dato del conjunto en cuestión.
- 2) Si se pulsa el botón de modificar, se realiza el reemplazo de fragmentos por el FragmentoModificarFormularioConjunto.
- 3) Si se pulsa el botón de eliminar, se le solicita al usuario la confirmación para eliminar el conjunto, y si es así, se borra de la base de datos y también, la imagen del conjunto en su carpeta contenedora.

## - **FragmentoVisualizarDatosPrenda**

Es el fragmento que muestra la información de la prenda y dos botones por si se desea modificar los datos o eliminar la prenda de la Base de Datos.

- 1) Una vez creada la vista del fragmento, se asigna a los distintos TextView su correspondiente dato de la prenda en cuestión.
- 2) Si se pulsa el botón de modificar, se realiza el reemplazo de fragmentos por el FragmentoModificarFormularioPrenda.
- 3) Si se pulsa el botón de eliminar, se le solicita al usuario la confirmación para eliminar la prenda, y si es así, se borra de la base de datos y también, la imagen de la prenda en el directorio.

→ com.virjc.myclothes.utils:

## - **CalendarioCustomAdapter**

Adapta la vista de los conjuntos asignados en sus respectivos días del calendario.

- 1) Se asignan las imágenes de cada conjunto en sus respectivas celdas.
- 2) Se asignan los estilos y las formas de mostrar los elementos en el calendario para customizarlo.

## - **ConjuntoImageAdapter**

Clase para adaptar el listado de conjuntos a la vista. Este listado se encuentra paginado, de manera que cuando el usuario baja el scroll se cargan los siguientes ImageView.

- 1) Se pagan las imágenes de los conjuntos.
- 2) Se crea un nuevo ImageView por cada item referenciado en el Adapter.

- **DynamicAsynTask**

Es la clase que ejecuta en un hilo aparte el conjunto de tareas que se desean realizar en segundo plano. Para ello hace uso de la clase Task. Primer parámetro: Tipo de los parámetros que se le pasan a la tarea asíncrona. Segundo parámetro: Tipo que controla el progreso de la tarea. Tercer parámetro: Tipo del resultado que devuelva la tarea.

- 1) Ejecuta las tareas en un hilo aparte. Cuando termina, devuelve un resultado con el tipo especificado en el tercer parámetro de la AsyncTask.
- 2) Para comunicar que se han hecho progresos en la tarea, se invoca el método publishProgress con un parámetro que sera del tipo especificado en el segundo parámetro de la clase.
- 3) Hay un método que se ejecuta si la tarea es cancelada. Cancela las tareas iterativamente.

- **Geolocation**

Es la clase encargada de obtener las coordenadas de latitud y longitud del dispositivo a partir del GPS o de la red.

- **ImageAdapter**

Es la clase que se encarga de adaptar las imágenes de las categorías a la vista de los fragmentos.

- **OrmLiteBaseFragmentActivity**

Este Activity sirve de base al CalendarioActivity para que tenga acceso al OrmLite con el FragmentActivity que necesita Caldroid, debido a que BaseActivity no hereda de FragmentActivity.

- **PrendaImageAdapter**

Clase para adaptar el listado de prendas a la vista. Este listado se encuentra paginado, de manera que cuando el usuario baja el scroll se cargan los siguientes ImageView.

- 1) Se paginan las imágenes de las prendas.
- 2) Se crea un nuevo ImageView por cada item referenciado en el Adapter.

#### - **Task**

Es una clase abstracta que contiene métodos abstractos que necesitan implementarse al crear objetos de ella. Estos métodos especifican lo que tiene que hacer la tarea en un hilo aparte , y lo que pasa una vez ejecutada o si es cancelada.

#### - **Utils**

Clase que contiene métodos de uso general de la aplicación. Son los siguientes:

- 1) Método que indica lo que hace cada icono del menú del ActionBar, lo usan todos los activities.
- 2) Métodos que realizan el intercambio de fragmentos.
- 3) Método a utilizar cada vez que se usen Intents, para eliminar Activities. Haciendo esto, se evita tener una larga pila de Activities en memoria.
- 4) Método para que al dar una vez para atrás vuelva al activity anterior, y si se da otra vez, vuelva al MainActivity. Si se pulsa una tercera vez que salga de la aplicación.
- 5) Método para obtener un algoritmo como el MD5, por ejemplo. Devuelve la instancia del algoritmo.
- 6) Método para obtener la representación hexadecimal del algoritmo MD5 aplicado a un conjunto de datos binarios.
- 7) Método para comprobar si está disponible la tarjeta SD para lectura y escritura.
- 8) Método para obtener el directorio de la aplicación.
- 9) Métodos para pasar bitmap a array de bytes y viceversa.
- 10) Métodos para escalar las dimensiones de una imagen (ya sea de forma proporcionada o no proporcionada) y para obtener sus dimensiones sin cargar una imagen en memoria.

- **Weather**

Es la clase que se utiliza para obtener los tipos de clima que puede dar el servicio Forecast. Se realiza la petición del servicio a través de una APIKey que facilitan al registrarse en la página. Esta APIKey está limitada a 1.000 llamadas gratis cada día. El servicio obtiene el clima en el lugar que se le pasa por coordenadas de latitud y longitud.

## 5.5. Pruebas

Como se indicó al principio del trabajo, la metodología utilizada en el desarrollo de la aplicación ha sido el modelo iterativo, donde la aplicación se va desarrollando en versiones incrementales, produciendo con cada iteración una versión más completa del sistema. Así que con cada iteración se han realizado una serie de test unitarios antes de continuar con la siguiente.

Se entiende que los casos en los que únicamente se visualizan datos sin que sea necesaria la interacción con el usuario, no son casos problemáticos siempre que el almacenamiento de esos datos se haya realizado correctamente, por lo que no es necesario la descripción de algunos de esos casos de prueba.

<b>Caso de prueba</b>	<b>CREAR PRENDA</b>
<b>Propósito</b>	Crear una nueva prenda en la Base de Datos.
<b>Prerrequisitos</b>	-
<b>Datos de entrada</b>	<ul style="list-style-type: none"> <li>• Imagen de la prenda.</li> <li>• Nombre.</li> <li>• Tipo.</li> <li>• Estación.</li> <li>• Marca</li> <li>• Color.</li> </ul>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Se comprueba que se ha obtenido una imagen desde la cámara o galería.</li> <li>2. Se comprueba que se ha realizado el recortado de la imagen correctamente.</li> <li>3. Se comprueba que ningún campo está vacío.</li> <li>4. Se comprueba que los datos son correctos.</li> <li>5. Se comprueba el correcto guardado en la Base de Datos.</li> </ol>
<b>Resultado esperado</b>	La prenda se guarda en la Base de Datos con los datos correctos, y se informa de ello.
<b>Resultado obtenido</b>	La prenda queda almacenada correctamente en la Base de Datos y se informa al usuario de ello.

**Tabla 45. Caso de prueba CU-01.**

<b>Caso de prueba</b>	<b>MODIFICAR DATOS PRENDA</b>
<b>Propósito</b>	Modificar una prenda.
<b>Prerrequisitos</b>	La prenda debe estar creada.
<b>Datos de entrada</b>	<ul style="list-style-type: none"> <li>• Nombre.</li> <li>• Tipo.</li> <li>• Estación.</li> <li>• Marca</li> <li>• Color.</li> </ul>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Se comprueba que la prenda existe.</li> <li>2. Se muestran los datos correctamente.</li> <li>3. Se comprueba la modificación de los datos es correcta y no hay campos vacíos.</li> <li>4. Se comprueba la correcta actualización en la Base de Datos.</li> </ol>
<b>Resultado esperado</b>	La prenda se actualiza en la Base de Datos con los datos correctos, y se informa de ello.
<b>Resultado obtenido</b>	La prenda queda actualizada correctamente en la Base de Datos y se informa al usuario de ello.

**Tabla 46. Caso de prueba CU-06.**

Caso de prueba	ELIMINAR PRENDA
<b>Propósito</b>	Eliminar una prenda.
<b>Prerrequisitos</b>	La prenda debe existir.
<b>Datos de entrada</b>	<ul style="list-style-type: none"> <li>• Confirmación del usuario</li> </ul>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Se comprueba la confirmación del usuario.</li> <li>2. Se comprueba que la prenda existe.</li> <li>3. Se eliminan los datos de la prenda en la Base de Datos.</li> <li>4. Se comprueba la eliminación de la imagen de la prenda en su carpeta contenedora.</li> </ol>
<b>Resultado esperado</b>	La prenda desaparece de la Base de Datos y se informa al usuario.
<b>Resultado obtenido</b>	La prenda queda eliminada correctamente en la Base de Datos y se informa al usuario de ello.

**Tabla 47. Caso de prueba CU-07.**

Caso de prueba	VISUALIZAR PROBADOR
<b>Propósito</b>	Visualizar conjuntos.
<b>Prerrequisitos</b>	Prendas existentes.
<b>Datos de entrada</b>	<ul style="list-style-type: none"> <li>• Imágenes de las prendas que lo forman.</li> </ul>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Se comprueba que existen prendas para poder formar conjuntos.</li> <li>2. Se comprueba que se reciben las imágenes de las prendas en el lugar que corresponde.</li> <li>3. Se comprueba que se desplazan correctamente las prendas.</li> </ol>
<b>Resultado esperado</b>	Se visualizan de manera adecuada las partes de arriba, partes de abajo, una sola pieza y calzado y se desplazan correctamente.
<b>Resultado obtenido</b>	Se visualizan de manera adecuada las partes de arriba, partes de abajo, una sola pieza y calzado y se desplazan correctamente.

**Tabla 48. Caso de prueba CU-08.**

<b>Caso de prueba</b>	<b>SELECCIONAR CONJUNTO</b>
<b>Propósito</b>	Crear nuevos conjuntos.
<b>Prerrequisitos</b>	Prendas existentes.
<b>Datos de entrada</b>	<ul style="list-style-type: none"> <li>• Prendas que lo forman.</li> <li>• Nombre.</li> <li>• Estilo.</li> <li>• Estación.</li> </ul>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Se comprueba que se reciben las prendas que lo componen.</li> <li>2. Se comprueba que ningún campo está vacío.</li> <li>3. Se comprueba que los datos son correctos.</li> <li>1. Se comprueba el correcto guardado en la Base de Datos.</li> </ol>
<b>Resultado esperado</b>	El conjunto se guarda en la Base de Datos con los datos correctos, y se informa de ello.
<b>Resultado obtenido</b>	El conjunto queda almacenado correctamente en la Base de Datos y se informa al usuario de ello.

**Tabla 49. Caso de prueba CU-09.**

Caso de prueba	MODIFICAR DATOS CONJUNTO
<b>Propósito</b>	Modificar un conjunto
<b>Prerrequisitos</b>	El conjunto debe estar creado.
<b>Datos de entrada</b>	<ul style="list-style-type: none"> <li>• Nombre.</li> <li>• Estilo.</li> <li>• Estación.</li> </ul>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Se comprueba que el conjunto existe.</li> <li>2. Se muestran los datos correctamente.</li> <li>3. Se comprueba la modificación de los datos es correcta y no hay campos vacíos.</li> <li>4. Se comprueba la correcta actualización en la Base de Datos.</li> </ol>
<b>Resultado esperado</b>	El conjunto se actualiza en la Base de Datos con los datos correctos, y se informa de ello.
<b>Resultado obtenido</b>	El conjunto queda actualizado correctamente en la Base de Datos y se informa al usuario de ello.

**Tabla 50. Caso de prueba CU-12.**

Caso de prueba	ELIMINAR CONJUNTO
<b>Propósito</b>	Eliminar un conjunto.
<b>Prerrequisitos</b>	El conjunto debe existir.
<b>Datos de entrada</b>	<ul style="list-style-type: none"> <li>• Confirmación del usuario.</li> </ul>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Se comprueba la confirmación del usuario.</li> <li>2. Se comprueba que el conjunto existe.</li> <li>3. Se eliminan los datos del conjunto en la Base de Datos.</li> <li>4. Se comprueba la eliminación de la imagen del conjunto en su carpeta contenedora.</li> </ol>
<b>Resultado esperado</b>	El conjunto desaparece de la Base de Datos y se informa al usuario.
<b>Resultado obtenido</b>	El conjunto queda eliminado correctamente en la Base de Datos y se informa al usuario de ello.

**Tabla 51. Caso de prueba CU-13.**

<b>Caso de prueba</b>	<b>VISUALIZAR CALENDARIO</b>
<b>Propósito</b>	Visualizar los conjuntos asignados a fechas concretas en un calendario.
<b>Prerrequisitos</b>	-
<b>Datos de entrada</b>	-
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Se comprueba que hay conjuntos asignados al calendario.</li> <li>2. Se recoge la imagen del conjunto.</li> <li>3. Se comprueba que se muestra correctamente la imagen del conjunto en la celda de la fecha correspondiente.</li> </ol>
<b>Resultado esperado</b>	Las imágenes de los conjuntos se muestran correctamente en las celdas de las fechas correspondientes.
<b>Resultado obtenido</b>	Las imágenes de los conjuntos se muestran correctamente en las celdas de las fechas correspondientes.

**Tabla 52. Caso de prueba CU-14.**

Caso de prueba	ASIGNAR CONJUNTO
<b>Propósito</b>	Asignar un conjunto a una fecha del calendario.
<b>Prerrequisitos</b>	Existe el conjunto y fecha seleccionada.
<b>Datos de entrada</b>	<ul style="list-style-type: none"> <li>• Clima.</li> <li>• Conjunto.</li> </ul>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Se comprueba que no hay conjuntos ya asignados en dicha fecha.</li> <li>2. Si no se puede conseguir el clima del servicio el usuario lo podrá elegir.</li> <li>3. Se comprueba que se ha dado un clima válido.</li> <li>4. Se comprueba que se ha elegido un conjunto correctamente.</li> <li>5. Se comprueba que se ha guardado la asignación del conjunto al calendario.</li> </ol>
<b>Resultado esperado</b>	El conjunto es asignado a la fecha del calendario y se guarda en la Base de Datos.
<b>Resultado obtenido</b>	El conjunto es asignado a la fecha del calendario y se guarda en la Base de Datos.

**Tabla 53. Caso de prueba CU-15.**

Caso de prueba	ELIMINAR CONJUNTO
<b>Propósito</b>	Eliminar un conjunto de la fecha del calendario.
<b>Prerrequisitos</b>	El conjunto del calendario debe existir.
<b>Datos de entrada</b>	<ul style="list-style-type: none"> <li>• Confirmación del usuario.</li> </ul>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Se comprueba la confirmación del usuario.</li> <li>2. Se comprueba que el conjunto del calendario existe.</li> <li>3. Se eliminan los datos del conjunto del calendario en la Base de Datos.</li> </ol>
<b>Resultado esperado</b>	El conjunto desaparece del calendario de la Base de Datos y se informa al usuario.
<b>Resultado obtenido</b>	El conjunto queda eliminado correctamente del calendario en la Base de Datos y se informa al usuario de ello.

**Tabla 54. Caso de prueba CU-16.**

<b>Caso de prueba</b>	<b>PEDIR RECOMENDACIÓN</b>
<b>Propósito</b>	Asignar un conjunto a una fecha del calendario.
<b>Prerrequisitos</b>	Existe el conjunto y fecha seleccionada.
<b>Datos de entrada</b>	<ul style="list-style-type: none"> <li>• Clima.</li> <li>• Estilo.</li> <li>• Estación.</li> </ul>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Se comprueba que existen conjuntos en el calendario.</li> <li>2. Se comprueba que se ha recibido el clima del servicio.</li> <li>3. Se comprueba que se han recibido parámetros válidos tanto del servicio como del usuario.</li> <li>4. Se comprueba que se devuelve un conjunto a recomendar.</li> </ol>
<b>Resultado esperado</b>	Un conjunto es mostrado al usuario como recomendación.
<b>Resultado obtenido</b>	Un conjunto es mostrado al usuario como recomendación.

**Tabla 55. Caso de prueba CU-17.**

# Capítulo 6

## Manuales

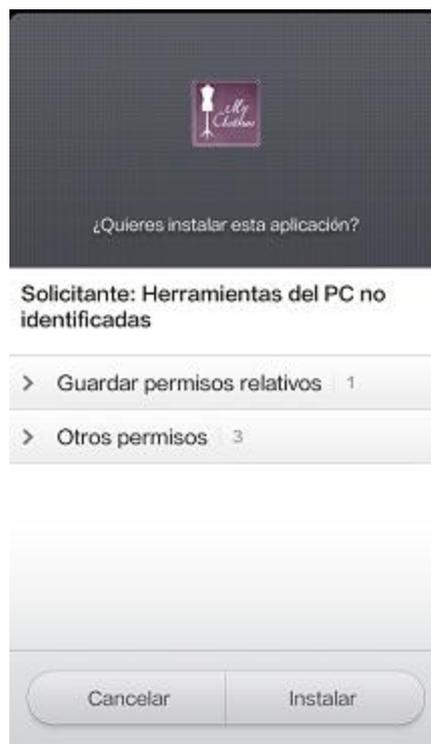
---



## 6.1. Manual de instalación

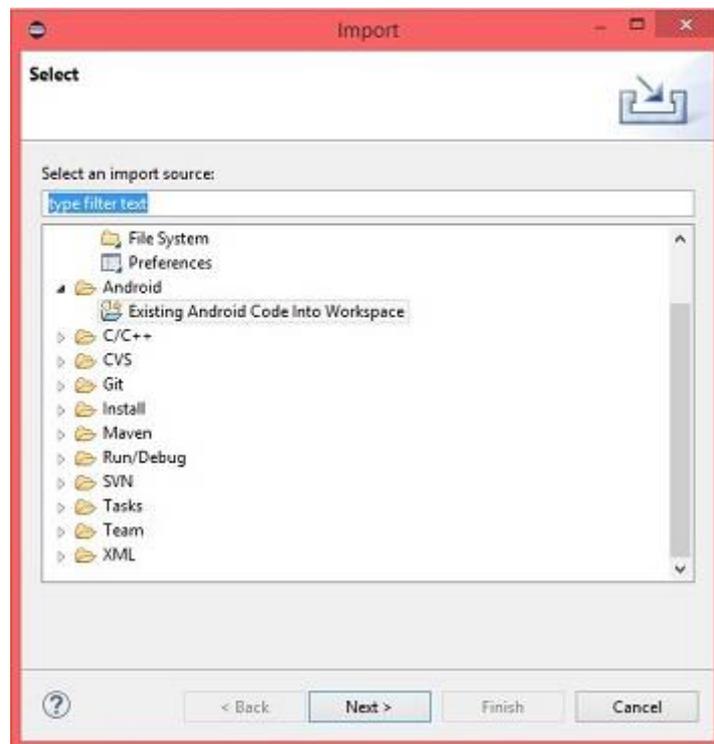
Al tratarse de una aplicación con base de datos en local, sólo es necesario instalar la app en un entorno Android. Esto puede realizarse a través de:

1. **Un dispositivo físico**, se descarga desde el dispositivo el .apk y se instala normalmente.

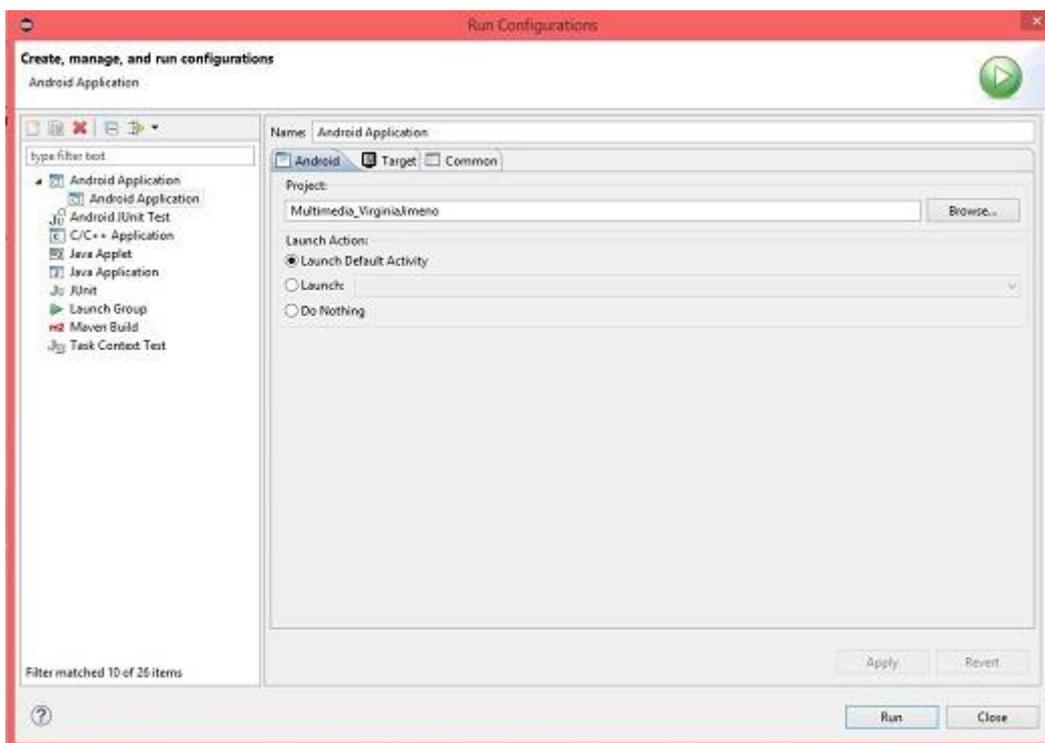


**Ilustración 30. Explicación de la instalación en un dispositivo móvil.**

2. **Una máquina virtual**, importando el proyecto en un IDE, como por ejemplo, Eclipse y ejecutándolo con los AVD que ofrece. También se puede con emuladores virtuales que existen en la web como por ejemplo: <https://www.manymo.com/> o simuladores como GenyMotion.



**Ilustración 31. Explicación de la instalación a través del proyecto 1.**



**Ilustración 32. Explicación de la instalación a través del proyecto 2.**

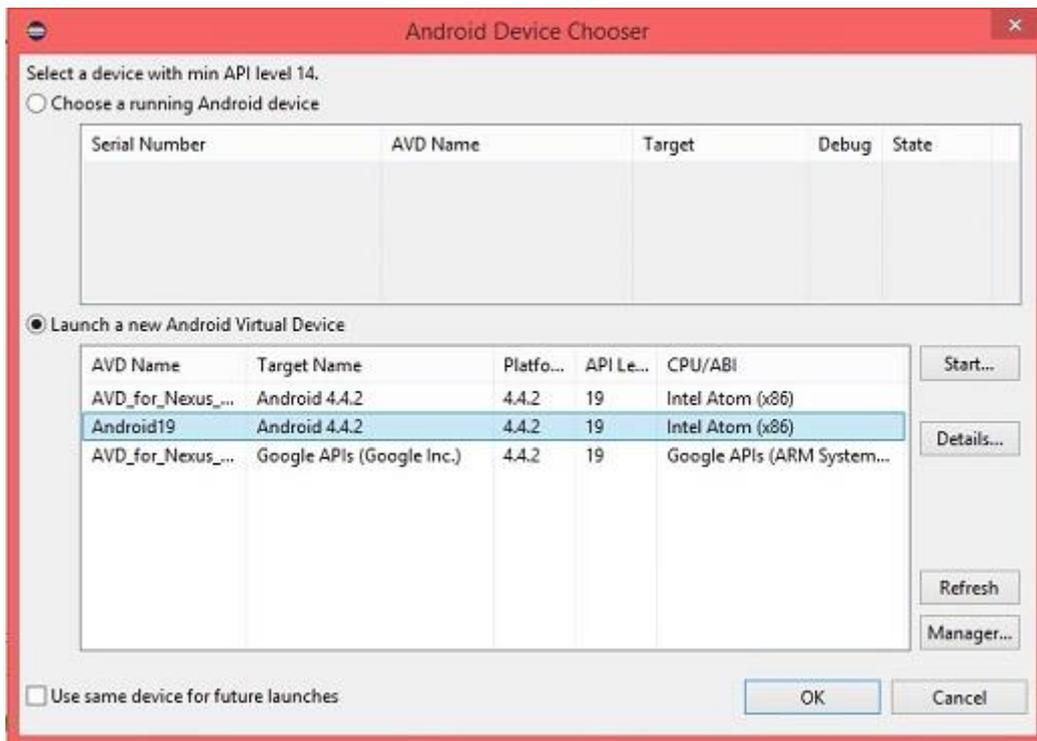


Ilustración 33. Explicación de la instalación a través del proyecto 3.

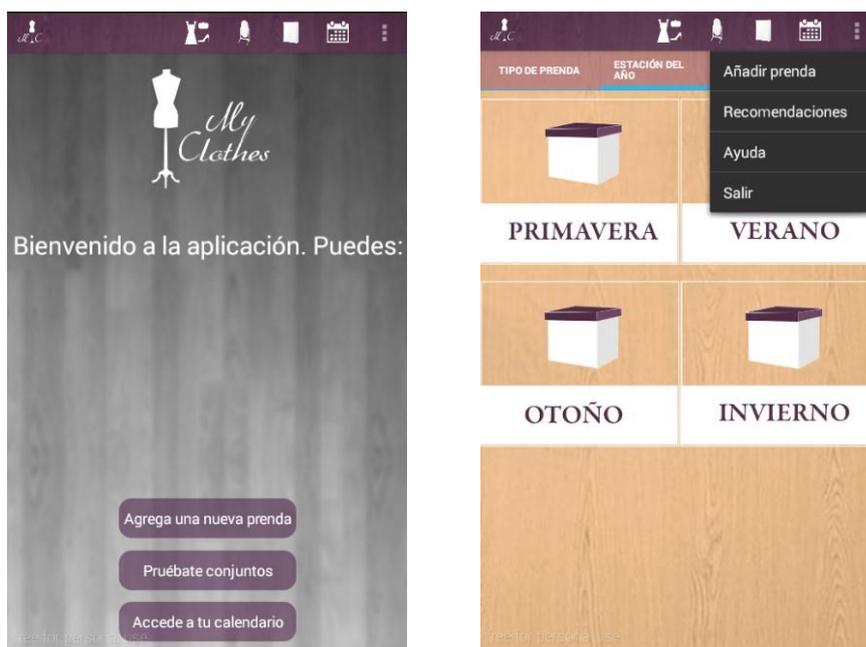


Ilustración 34. Explicación de la instalación a través del proyecto 4.

## 6.2. Manual de usuario

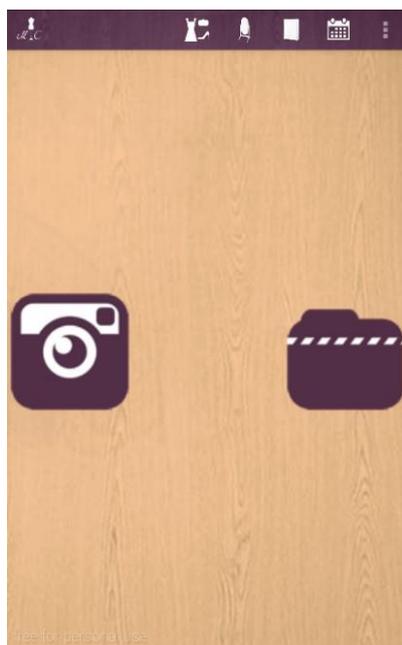
### 6.2.1. Gestión de prendas

#### ➤ Agregar una prenda de ropa



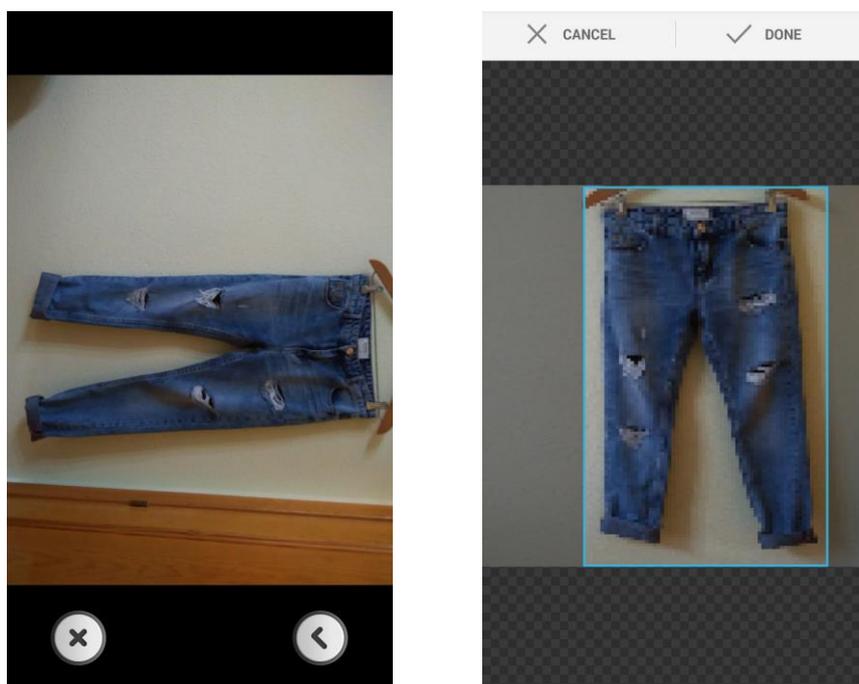
**Ilustración 35. Manual de usuario: Agregar prenda 1.**

Se podrá seleccionar el botón desde la pantalla principal “**Agrega una nueva prenda**”, o desde cualquier parte en la que se encuentre, seleccionando los tres puntos que se encuentran en la parte superior derecha y “**Añadir prenda**”. Entonces se visualizará la siguiente pantalla, que indica que se puede seleccionar una foto desde la cámara o desde la galería:



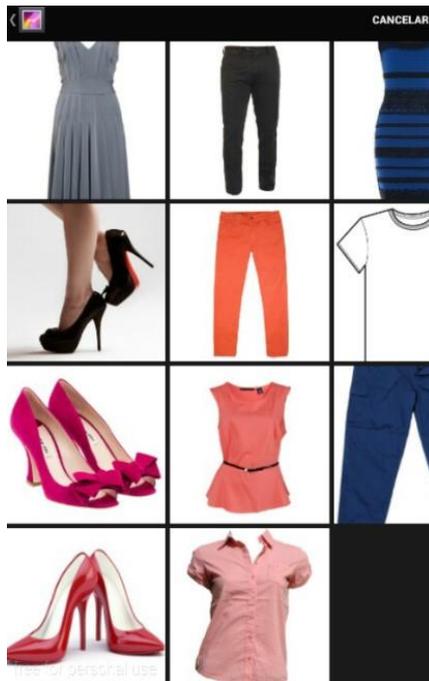
**Ilustración 36. Manual de usuario: Agregar prenda 2.**

Al seleccionar tomar la foto desde la cámara, se realiza su captura y luego se da la opción de recortar la imagen para que quede más centrada la prenda.



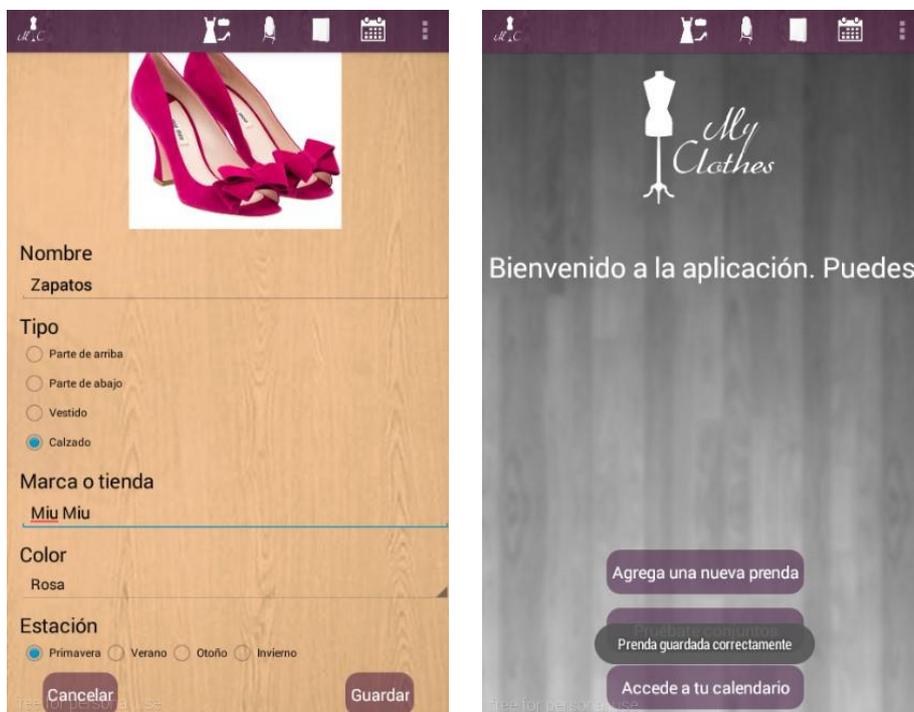
**Ilustración 37. Manual de usuario: Agregar prenda 3.**

En el caso de la galería es igual, primero se selecciona la foto y luego se recorta.



**Ilustración 38. Manual de usuario: Agregar prenda 4.**

Una vez recortada la imagen se mostrará un formulario para que se cumplimente con los datos de la prenda. Al pulsar en guardar se vuelve a la pantalla principal y se indica si ha sido guardado correctamente.



**Ilustración 39. Manual de usuario: Agregar prenda 5.**

➤ **Visualizar prendas por categorías**

Desde cualquier parte de la aplicación se puede acceder al módulo armario con el tercer icono de la barra superior. Al acceder a él, por defecto, sale la clasificación de las prendas según Tipo de Prenda, pero se puede seleccionar otra categoría deslizando la barra de pestañas.



**Ilustración 40. Manual de usuario: Visualizar prenda 1.**

Al seleccionar una de las categorías ya sea según Tipo de Prenda, Estación del Año, Color o Todos, se visualiza un listado con las prendas de dicha categoría (en este caso se ha seleccionado según la categoría de vestidos):



Ilustración 41. Manual de usuario: Visualizar prenda 2.

➤ **Visualizar la información de una prenda, modificarla y eliminarla**

Al seleccionar una prenda desde cualquier categoría, aparece la información relativa a ella:

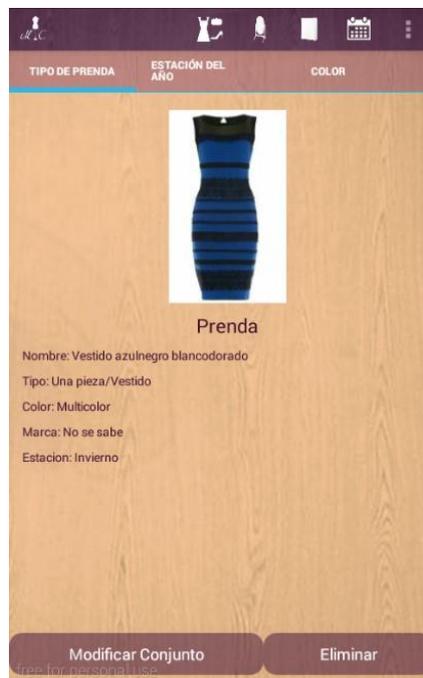


Ilustración 42. Manual de usuario: Visualizar prenda 3.

Si se desea eliminar la prenda se pulsa en “Eliminar” y después únicamente será necesario confirmar su eliminación. Si se quiere modificar los datos de la prenda, se muestra el formulario con los datos actuales para ser modificados.



**Ilustración 43. Manual de usuario: Modificar prenda.**

## 6.2.2. Gestión de conjuntos

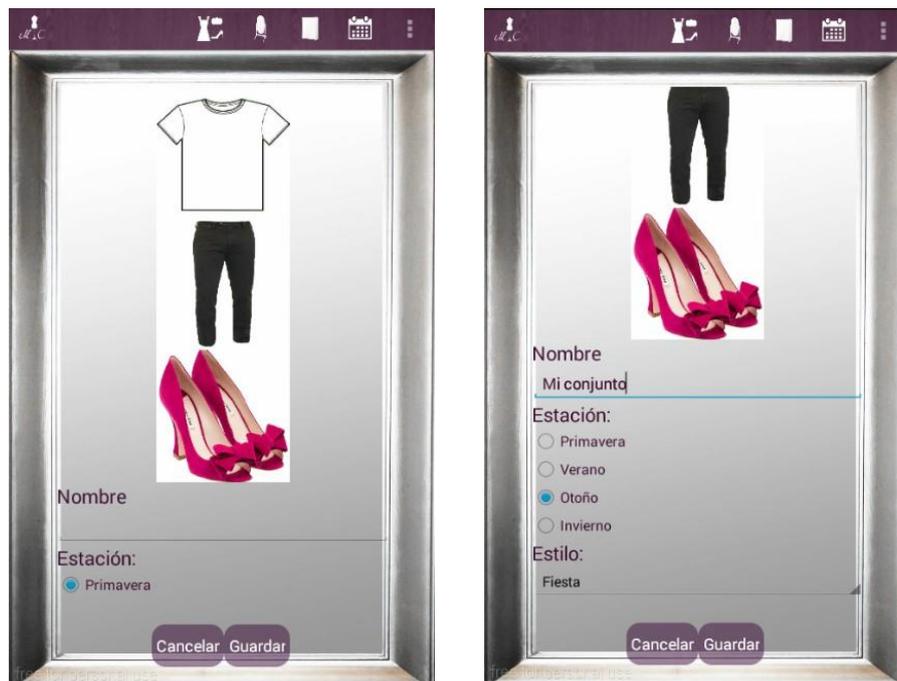
### ➤ Crear un nuevo conjunto

El usuario puede seleccionar el botón desde la pantalla principal “**Pruébate un conjunto**”, o desde cualquier parte en la que se encuentre, seleccionando el segundo icono de la barra superior.



**Ilustración 44. Manual de usuario: Probar conjunto.**

Al seleccionar un conjunto se pasa al formulario del conjunto.



**Ilustración 45. Manual de usuario: Crear conjunto.**

Al pulsar en guardar se vuelve a la pantalla principal y se indica si ha sido guardado correctamente.

### ➤ Visualizar conjuntos por categorías

Desde cualquier parte de la aplicación se puede acceder al módulo conjuntos con el primer icono de la barra superior. Al acceder a este módulo, por defecto, sale la clasificación de los conjuntos según su Estilo, pero se puede seleccionar otra categoría deslizando la barra de pestañas.



**Ilustración 46. Manual de usuario: Visualizar categorías conjuntos.**

Al seleccionar una de las categorías ya sea según Estilo, Estación o ver Todos, se visualiza un listado de los conjuntos de dicha categoría (en este caso se ha seleccionado según la categoría de otoño):



**Ilustración 47. Manual de usuario: Visualizar listado de conjuntos.**

➤ **Visualizar la información de un conjunto, modificarlo y eliminarlo**

Al seleccionar un conjunto desde cualquier categoría, aparece la información relativa a él:



**Ilustración 48. Manual de usuario: Visualizar conjunto.**

Si se desea eliminar el conjunto, tan sólo será necesario pulsar en “Eliminar” y confirmar su eliminación. Si se quiere modificar los datos del conjunto, se muestra el formulario con los datos actuales para ser modificados.



**Ilustración 49. Manual de usuario: Modificar conjunto.**

### 6.2.3. Gestión del calendario

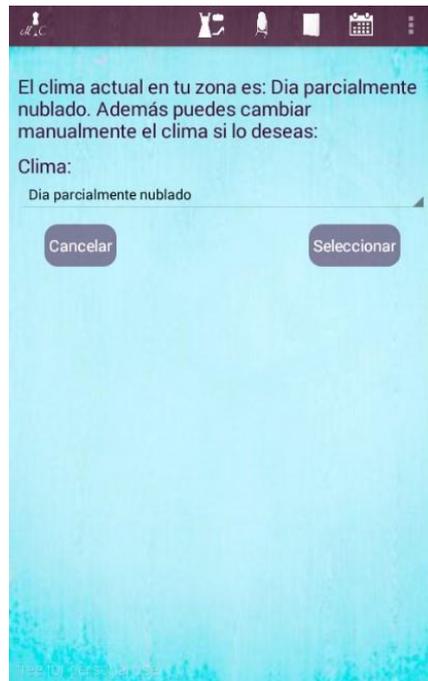
#### ➤ Asignar conjunto a una fecha determinada

Se accede al calendario pulsando en el cuarto botón de la barra superior.



**Ilustración 50. Manual de usuario: Calendario.**

Para asignar un conjunto se pulsa en el día al que se le quiere asignar un conjunto.



**Ilustración 51. Manual de usuario: Asignar conjunto en el calendario.**

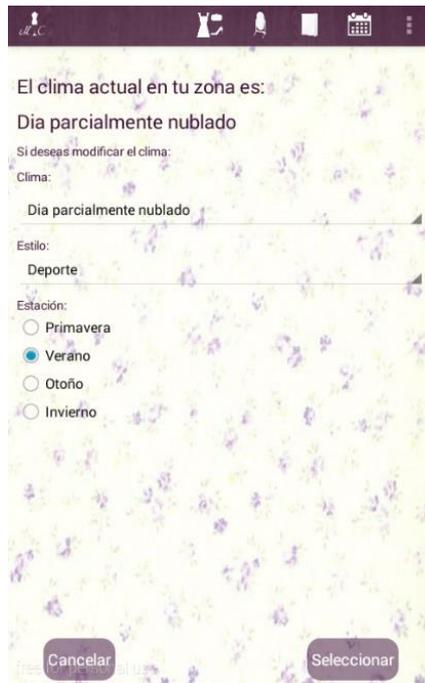
Se seleccionará el clima, que se le asigna al día que contiene el conjunto, y después el conjunto a añadir.

Para quitar el conjunto de una fecha determinada, se le debe hacer un pulsado largo y confirmar su eliminación de dicho día.

## 6.2.4. Gestión de las recomendaciones

### ➤ **Pedir recomendación**

Para solicitar a la aplicación la recomendación de un conjunto, se deberá pulsar en los tres puntos superiores y después en “Recomendaciones”.



**Ilustración 52. Manual de usuario: Pedir recomendación.**

Se seleccionará el clima, la estación y el estilo del conjunto que se quiere, y la aplicación mostrará los datos del conjunto que se le recomienda al usuario.



# Capítulo 7

## Futuras mejoras y conclusiones

---



## 7.1. Mejoras futuras

Una mejora podría consistir en aplicar la técnica de la segmentación al tratamiento de imágenes, siendo ésta un proceso de asignación de una etiqueta a cada píxel de la imagen, de forma que los píxeles que compartan la misma etiqueta también tendrán ciertas características visuales similares. Con esto se conseguiría que en la fotografía de una prenda de ropa se viera solamente dicha prenda, quitando el fondo y el resto de elementos que no son necesarios.

También se podría añadir más tipos de prendas como, por ejemplo, abrigos y accesorios, y más categorías a la hora de clasificar. Y, tal vez pudiera tener cierta utilidad compartir las imágenes de los conjuntos a través de las redes sociales.

Además, almacenar las prendas y conjuntos en servidores externos y que se puedan acceder a ellos a través de la web, daría mayor amplitud a la aplicación.

Así mismo, sería interesante expandir la aplicación a iOS, ya que en dicha plataforma existen bastantes usuarios potenciales.

En definitiva es una aplicación que tiene posibilidades para expandirse a otras opciones.

## 7.2. Conclusiones personales

Desde hace unos años rondaba por mi cabeza la idea de hacer una aplicación para teléfonos móviles que sirviera para el control del propio vestuario, no a todas las personas les puede parecer interesante, se lo comenté a mis amigas y amigos y les pareció bien, en mi casa mis padres me apoyaron sin ninguna duda, me decían que hiciera sobre todo algo que a mí me gustara y así disfrutaría del trabajo y aprendería mucho en el proceso.

Al principio no tenía nada claro, empecé con las asignaturas de Multimedia y Plataformas Software Móviles, con el apoyo de los profesores de estas asignaturas en este proyecto.

Hubo muchas preguntas, dudas de todo tipo, afortunadamente hoy día a través de Internet tenemos acceso a un gran volumen de información, pero que hay que cribar, seleccionar y asimilar, tampoco es tan fácil no está todo lo que se necesita en cualquier momento, hay que recurrir también a libros o intercambio de conversaciones con otras personas. Además hay que destacar que programar Android en Java con el IDE Eclipse daba lugar a numerosos fallos con las librerías y APIs, ya que este entorno de desarrollo no resulta robusto porque se colapsaba frecuentemente, lo que añadía todavía una mayor dificultad. En definitiva, ha sido necesario emplear muchas, muchas horas, pero yo seguía mi idea inicial y se iban abriendo caminos, posibilidades y realmente era interesante, gracias a ese interés es por lo que no te rindes y sobre todo si vas viendo resultados.

Han sido varios meses (casi un año) y al final todo llega, por fin he terminado mi trabajo fin de grado completamente satisfecha, tanto por haberlo acabado como por lo que he aprendido a lo largo del proceso.

No sé si este programa tendrá una gran aceptación entre las personas en general, pero yo me siento muy feliz en su creación.

Gracias a mis amistades y familia por apoyarme en todo momento, a Aníbal y Miguel Ángel (tutores de este trabajo), así como al resto del profesorado por transmitirme sus conocimientos en estos cuatro años.

# Capítulo 8

## Referencias

---

## 8.1. Bibliografía

K. Wiegers, J.Beatty. “*Software Requirements*”. Microsoft Press, 3rd edition, 2013.

Joan Ribas Lequerica. “*Manual imprescindible de desarrollo de aplicaciones para Android*”. Madrid: Anaya Multimedia, edición 2013.

W. Frank Ableson, Robi Sen, Chris King. “*Android: guía para desarrolladores*”. Anaya Multimedia, 2a edición, 2011.

P.Havaladar, G.Medioni. “*Multimedia Systems: Algorithms, Standards, and Industry Practices*”. Cengage Learning; 1st edition, 2009.

## 8.2 Referencias web

*Android Developer Reference*. Disponible en: <http://developer.android.com/index.html> [Última consulta: 24/06/2015]

*Java SE*. Disponible en: <http://www.oracle.com/technetwork/java/javase/overview/index.html> [Última consulta: 30/06/2015]

*ORMLite*. Disponible en: <http://ormlite.com/> [Última consulta: 24/06/2015]

*StackOverFlow*. Disponible en: <http://stackoverflow.com/> [Última consulta: 30/06/2015]

*40deFiebre*. Disponible en: <http://www.40defiebre.com/elementos-buena-interfaz-usuario/> [Última consulta: 30/06/2015]

# ANEXO I

## Contenido del CD-ROM

---



El CD-ROM tiene la siguiente estructura:

- **Software:** Contiene el .apk para instalar la aplicación en un dispositivo móvil. También contiene el código de la aplicación.
- **Diagramas:** En esta carpeta se encuentran todos los diagramas expuestos en este proyecto.
- **Documentación:** Contiene la versión digital de esta memoria en .pdf.



## ANEXO II

### Diagrama de clases

---

# Diagrama de clases en formato A3.

