



SmartClothes

SmartClothes, organiza tu ropa de forma inteligente

Nombre Estudiante: Sergio Lozano del Río

Máster universitario en Desarrollo de Aplicaciones para Dispositivos Móviles

Nombre Consultor: Pau Dominkovics Coll

Profesor/a responsable de la asignatura: Carles Garrigues Olivella

14/03/2018



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)
Copyright © 2018 SERGIO LOZANO DEL RÍO.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

© Sergio Lozano del Río

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	SmartClothes, organiza tu ropa de forma inteligente
Nombre del autor:	Sergio Lozano Del Río
Nombre del consultor:	Pau Dominkovics Coll
Fecha de entrega (mm/aaaa):	06/2018
Titulación:	<i>Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles</i>

Resumen del Trabajo:

Este proyecto está orientado al desarrollo de una aplicación para dispositivos android, que organice de forma inteligente la ropa de sus usuarios. Basado en el concepto de IOT (Internet Of Things), este proyecto busca conectar ropa de diario que utilizan sus usuarios, para así poder asesorarlos en función del tiempo o la disponibilidad dentro de su armario. Las diferentes prendas se comunican mediante el uso de tecnología NFC, que permite identificar cada una de manera independiente. Una de las bases de este proyecto es poder ayudar a personas discapacitadas, en especial a personas con problemas visuales o ceguera, es por este motivo que la aplicación permite la comunicación con el asistente personal de google que se comunica con los usuarios mediante la conversión de texto en voz.

Abstract:

This project is based on the development of an application for android devices, which intelligently organize the clothes of its users. Base on the concept of IOT (Internet Of Things), this project is looking to connect the daily wear that their users used to wear frequently, in order to be able to advise according to the weather or availability inside their wardrobes. The different garments communicate through the use of NFC technology, that allows you to identify each one independently. One of the bases of this project is to be able to help people with disabilities, especially people with visual problems or blindness, it is for this reason that the application allows communication with the google personal assistant that communicates with its users through the conversion of text to speech.

Palabras clave:

Invidentes, IOT, NFC, Ropa, Asistente personal, SmartClothes

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	5
1.3 Enfoque y método seguido.....	7
1.4 Planificación del Trabajo.....	9
1.5 Breve resumen de productos obtenidos.....	13
1.6 Recursos necesarios.....	14
1.7 Breve descripción de los otros capítulos de la memoria.....	15
2. Diseño de la aplicación.....	16
2.1 Análisis.....	16
2.1.1 Recopilación de datos.....	16
2.1.2 Ámbito de difusión.....	16
2.1.2 Contextos de uso.....	17
2.2 Diseño conceptual.....	19
2.2.1 Perfiles de usuario y posibles escenarios.....	19
2.2.2 Descripción de los diferentes casos de uso.....	23
2.2.3 Diagrama de flujo.....	33
2.3 Arquitectura y base de datos.....	35
2.3.1 Diseño de la base de datos.....	35
2.3.2 Diagrama de clases.....	37
2.3.3 Arquitectura del sistema.....	40
2.4 Prototipado.....	42
2.4.1 Líneas de diseño.....	42
2.4.2 Prototipo de baja fidelidad.....	45
2.4.2 Prototipo de alta fidelidad.....	47
2.5 Evaluación del prototipo.....	53
3. Desarrollo.....	55
3.1 Firebase.....	56
3.2 Gestión de los datos.....	57
3.3 Microprocesador esp8266.....	60
3.4 Desarrollo Android.....	64

3.4.1 Inicio de sesión	65
3.4.2 MainActivity	67
3.4.3 AddClothes.....	67
3.4.4 AccountActivities	69
3.4.5 ClosetActivity.....	69
3.5 Librerías utilizadas	71
3.7 Problemas durante el desarrollo.....	73
4 Pruebas.....	75
5. Conclusiones y líneas futuras	77
6. Glosario.....	80
7. Bibliografía	82
8. Anexos	84
Anexo 1: Diagrama Gantt de la planificación del trabajo.....	85
Anexo 2: Planificación por horas llevada a cabo en Excel.	86

Lista de figuras

Ilustración 1: Muestra de varios asistentes personales	2
Ilustración 2: Aplicación de la marca Vuplex	4
Ilustración 3: Primera fase; Archivo Gantt	9
Ilustración 4: Tabla de horas invertidas en la primera fase	10
Ilustración 5: Segunda fase; Archivo Gantt	10
Ilustración 6: Tabla de horas invertidas en la segunda fase.....	10
Ilustración 7: Tercera fase; Archivo Gantt	11
Ilustración 8: Tabla de horas invertidas en la tercera fase	11
Ilustración 9: Cuarta fase; Archivo Gantt.....	12
Ilustración 10: Tabla de horas invertidas en la cuarta fase	12
Ilustración 11: Ficha persona I	20
Ilustración 12: Ficha persona II	21
Ilustración 13: Ficha persona III	22
Ilustración 14: Árbol de navegación.	33
Ilustración 15: Diagrama de la Base de datos.....	35
Ilustración 16: Diagrama de clases	37
Ilustración 17: Arquitectura del sistema.....	40

Ilustración 18: Guía de diseño para logotipo	42
Ilustración 19: Prototipo de alta fidelidad; sesión de usuarios.....	47
Ilustración 20: Prototipo de alta fidelidad; Main Activity	48
Ilustración 21: Prototipo de alta fidelidad; sección cloth detect	48
Ilustración 22:Prototipo de alta fidelidad; sección My closet	49
Ilustración 23:Prototipo alta fidelidad; outfit edit y cloth status detail.....	50
Ilustración 24:Prototipo de alta fidelidad; sección de Settings	51
Ilustración 25:Prototipo de alta fidelidad; Settings: Name, Email, Password	52
Ilustración 26: Pantallas de prototipo descartadas.....	53
Ilustración 27:Esquema prendas de la base de datos.....	58
Ilustración 28: Esquema usuarios de la base de datos.	59
Ilustración 29: Microprocesador esp8266	60
Ilustración 30: Código petición REST desde microprocesador.	61
Ilustración 31: Cloud Function microprocesador	62
Ilustración 32: Notificación "alert"	64
Ilustración 33:Pantallas "SesionActivity"	66
Ilustración 34:Pantallas "MainActivity"	67
Ilustración 35: Cambio de valor hexadecimal a decimal en Kotlin.....	68
Ilustración 36: Panatalla "add clothes" tras analizar una prenda.....	69
Ilustración 37: Pantallas "ClosetActivity"	70
Ilustración 38: Grafica de consumo de recursos de la aplicación.....	75
Ilustración 39:Pruebas exitosas	76
Ilustración 40: Planificación en gantt.....	85
Ilustración 41: Planificación de horas en excel.....	86

1. Introducción

1.1 Contexto y justificación del Trabajo

En los últimos años la tecnología ha ayudado enormemente a personas con diferentes discapacidades, facilitando tareas de la vida cotidiana, que, sin la ayuda de ciertos dispositivos o aplicaciones, serían verdaderamente complicadas de realizar. Las tecnologías de asistencia y rehabilitación permiten ayudar a las personas con discapacidades a funcionar con más facilidad en la vida cotidiana y también facilitar el cuidado de una persona discapacitada a quienes se encargan de esta tarea. El término "tecnología de rehabilitación" a veces se usa para referirse a dispositivos que se utilizan para ayudar a las personas a recuperar funciones luego de una enfermedad o lesión. Las "tecnologías de asistencia" podrían ser algo tan simple como una lupa para mejorar la visión o tan complejo como sistemas de comunicación computarizados.¹

Tras analizar varios testimonios de personas con discapacidad visual, se llega a la conclusión de que la tecnología existente no es lo suficientemente eficaz como para resolver el problema de vestuario patente en miles de personas con deficiencias visuales.

“La ropa me la organizo yo, si me rompen el orden en el armario es cuando tengo problemas”, cuenta Juan Antonio.²

Aunque existen detectores de colores o aplicaciones que detectan objetos mediante el uso de la cámara, no es una tecnología cien por cien fiable o bien no aporta la suficiente información de la prenda u objeto en cuestión.

¹ <https://www1.nichd.nih.gov/espanol/salud/temas/rehabtech/Pages/default.aspx>

² <https://longituedeonda.com/index.php/el-dia-a-dia-de-un-invidente/>

Aprovechando el concepto del IOT (Internet Of Things)³, es posible plantear una aplicación que sea capaz de entender cuáles son los objetos que rodean la vida de una persona, aportando datos y creando múltiples funcionalidades entre ellos. Orientando este concepto a personas con discapacidad visual, es posible recrear su entorno y ayudarles a realizar tareas de forma mucho más ágil e independiente.



Ilustración 1: Muestra de varios asistentes personales

A día de hoy, los asistentes personales como, por ejemplo, Google Assistant, Siri o Alexa, se han ido estableciendo poco a poco en la vida cotidiana de sus usuarios. Actualmente, existen multitud de aplicaciones para dispositivos móviles que se nutren de estos asistentes, facilitando su uso en situaciones donde los usuarios tienen las manos ocupadas o no pueden interactuar con una pantalla. Sin embargo, y a pesar de que grandes empresas están promocionando cada vez más el uso de estos

³ IOT (Internet Of Things): Se trata de una nueva tecnología de acceso a Internet. Mediante el Internet de las cosas, los objetos se reconocen entre sí, obteniendo un comportamiento de inteligencia conjunta, haciendo o habilitando decisiones relacionadas.

dispositivos, son pocas las aplicaciones que se centran en personas con discapacidades, por lo que se trata de un campo aún por explotar.

Cada año estamos un paso más cerca de las prendas inteligentes y, es que, cada vez son más las empresas de la industria de la moda, que centran sus recursos en vincular la tecnología y la ropa. El “boom” de los “wearables” es cada vez mayor, pero ¿es realmente la solución definitiva? El uso de la tecnología NFC⁴ (Near Field Communication) y RFID⁵ (Radio Frequency Identification) es cada vez mayor y no tardará en llegar a la industria de la moda de manera estandarizada, eliminando intermediarios incómodos como, por ejemplo, pulseras inteligentes.

Actualmente, existen varias empresas que ofrecen una aplicación que conecta accesorios de ropa o prendas con una aplicación móvil, sin embargo, fuera del ámbito wearable, como por ejemplo puede ser la marca Fitbit⁶, es complicado encontrar empresas que realicen prendas inteligentes. El caso de éxito más representativo respecto a la creación y uso de prendas inteligentes es [Vuplex](https://www.vulpes-electronics.com/)⁷, una empresa dedicada a la venta de prendas de invierno, enfocadas a realizar deporte en la montaña, pero también de uso diario como gorros, guantes o plantillas para los zapatos. Combinando los productos que ofrece Vuplex con su aplicación móvil, es posible monitorizar la temperatura en sus prendas.

⁴ NFC: Del término inglés Near Field Communication, es una tecnología de comunicación inalámbrica de corto alcance basado radio frecuencia.

⁵ RFID: Es un sistema de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas, tarjetas o transpondedores RFID. El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto

⁶ <https://www.fitbit.com/es/home>

⁷ <https://www.vulpes-electronics.com/>

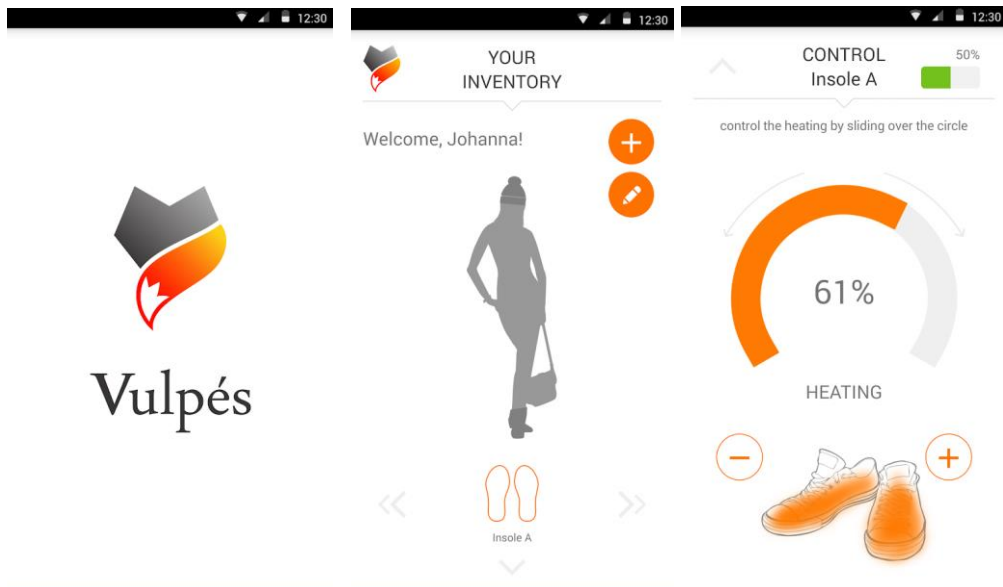


Ilustración 2: Aplicación de la marca Vuplex

Google, en colaboración con la marca de ropa Levis han creado un proyecto conjunto llamado Jacquard⁸ una chaqueta, capaz de comunicarse con tu Smartphone, aún no ha salido al mercado, pero, como vemos, la implementación de la tecnología en la ropa de uso diario está cada vez más cerca. Esta chaqueta permite comunicarte con otras personas, hacer uso de la navegación GPS o cambiar de música mientras andas de una forma mucho más cómoda, sin tener que hacer uso de un teléfono móvil.

Este proyecto pretende facilitar el día a día de una persona invidente, proporcionándole información sobre su vestuario e indumentaria de uso habitual.

⁸ <https://atap.google.com/jacquard/levi/>

1.2 Objetivos del Trabajo

El principal objetivo del proyecto es la comunicación directa entre los usuarios de la aplicación y la ropa que usan de forma diaria, que facilite el proceso de organización y vestimenta a cualquier usuario, facilitado esta tarea especialmente a las personas invidentes.

Requerimientos funcionales:

- Registrarse y autenticarse dentro de la aplicación.
- Añadir las prendas con chip NFC a la aplicación.
- Reconocimiento de las prendas mediante el lector NFC.
- Poder realizar preguntas adecuadas al sistema y que este sea capaz de responder correctamente mediante “speech text”.
- Hacer que el sistema sea capaz de asesorar en función del “outfit” propuesto y el tiempo meteorológico.
- Poder indicar si la prenda se encuentra lavada, en proceso de lavado o en el armario.

Requerimientos no funcionales:

- Desarrollar una aplicación nativa para dispositivos android.
- Priorizar la accesibilidad para personas invidentes.
- Comprobar la viabilidad y el coste de la tecnología NFC y RFID en la ropa.

- Conectar un lector NFC a una red Wi-Fi mediante un chip Wi-Fi ESP8266.
- Desarrollar el sistema de autenticación de usuarios y registro de datos mediante el uso de la plataforma Firebase.
- Uso del servicio Dialogflow⁹ para el reconocimiento de voz y creación de una interfaz inteligente, capaz de comunicarse con el asistente de voz google assistant.
- Para conectar el servicio de Dialogflow con Firebase, es necesario la creación de un webhook¹⁰ vinculado a varias funciones en google cloud functions.
- Uso del Api [Darksky](#) para la revisión del tiempo meteorológico, geo localizado.

La idea principal es conseguir conectar un microcontrolador y lector de tarjetas y chips RFID y NFC a la aplicación desarrollada, creando un entorno que permita gestionar los estados de todas las prendas NFC de las que disponga el usuario, ofreciendo un ecosistema funcional tanto a personas invidentes como a personas sin ningún tipo de defecto visual que únicamente busquen una mejor organización en su armario y vestimenta diaria.

⁹ Dialogflow: Plataforma de desarrollo orientado a la interacción humano-computadora, cuya tecnología permite generar conversaciones de forma natural.

¹⁰ Los webhooks son un método de alteración del funcionamiento de una página o aplicación web, que utilizando un callback personalizado puede modificar el contenido de la página.

1.3 Enfoque y método seguido

El proyecto busca desarrollar una tecnología de asistencia que permita a personas invidentes de todo el mundo realizar, de forma más sencilla, una tarea tan común como la de vestirse, que en manos de una persona discapacitada puede volverse realmente compleja.

Al mismo tiempo que el proyecto busca la innovación social y ayudar a un colectivo concreto, se pretende validar el coste y la eficiencia de la tecnología NFC en las prendas de uso cotidiano. De forma teórica y, partiendo del modelo de negocio startup¹¹, se plantea crear una línea de ropa económica que incorporen chips NFC. En conjunto con la aplicación desarrollada se pretende acceder al mercado de prendas inteligentes y wearables comprobando la viabilidad del producto creado.

A pesar de que existen múltiples aplicaciones para personas invidentes, ninguna se centra exclusivamente en prendas de ropa, lo que deja abiertas múltiples opciones a la hora de establecer el desarrollo de la aplicación. Por este motivo, dentro de la estrategia a seguir, se empleará una metodología de diseño centrado en el usuario y el modelo ágil scrum¹², ya que, en especial este último, permite iterar en el desarrollo una vez se detectan posibles problemas, al realizar pruebas y detectar posibles fallos. La metodología de diseño centrada en el usuario, permite desarrollar, buscando solventar las necesidades concretas de sus

¹¹ El término inglés “startup” hace referencia a una compañía emergente que busca emprender o montar un nuevo negocio. Normalmente este tipo de compañías busca desarrollare un modelo de negocio efectivo validando primero su producto y creando un modelo de negocio escalable.

¹² Desarrollo ágil que se caracteriza por adoptar una estrategia de desarrollo incremental, buscar la calidad del producto basándose en el conocimiento táctico de las personas en equipos auto organizados y el solapamiento de diferentes fases del desarrollo en lugar de realizar una tras otra en un ciclo secuencial.

usuarios finales, dadas las características del proyecto y su público objetivo.

Se ha optado por crear una aplicación de android nativa, ya que se pueden explotar mejor las funcionalidades que ofrece el sistema operativo android como, por ejemplo, la geolocalización, la vinculación directa con el asistente de google y que permite, la lectura de tarjetas NFC desde la propia aplicación. Además, el desarrollo nativo tiene muchas ventajas ya que, al tratarse de aplicaciones diseñadas directamente para el software del terminal, tiene un rendimiento optimizado, que garantiza buenos tiempos de reacción de cara a los usuarios.

[Ver apartado 2.3.3 Arquitectura del sistema](#)

1.4 Planificación del Trabajo

El proyecto está dividido en cuatro fases, cuyos hitos coinciden con las entregas establecidas por las PEC de la asignatura. Una vez establecidos los objetivos del proyecto se han ido planteando las tareas dentro de un diagrama de Gantt y, se han repartido los tiempos en días, en función de la dificultad de la tarea y la experiencia con tareas similares dada en el pasado. También se ha creado un archivo Excel para establecer un número concreto de horas, para llevar a cabo cada tarea. El número de horas totales se ha deducido en base al número de créditos ECTS establecido para la asignatura, lo que equivale a un total de 300h. Es probable que el tiempo invertido a lo largo del proyecto sea mayor, pero se ha creído conveniente mantener este número, ya que, de esta forma, si surge algún imprevisto no contemplado en la planificación, se puede reaccionar a tiempo y actuar en consecuencia sobre la planificación establecida. El tiempo medio de dedicación al proyecto es de 4h entre semana y 3h los fines de semana.

A continuación, se explica el reparto de tareas y sub-tareas en las distintas fases del proyecto:

Primera fase – Plan de trabajo:

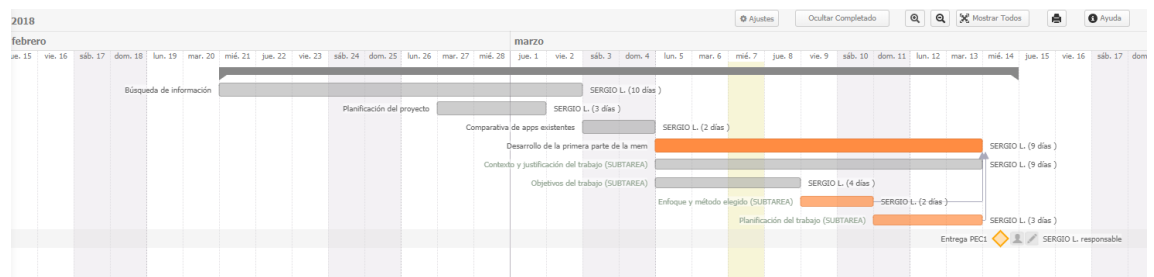


Ilustración 3: Primera fase; Archivo Gantt

	A	B	C
1	Tareas	Horas	Nº Horas restantes
2	Busqueda de información	6	294
3	Comparativa de apps	4	290
4	Planificación del proyecto	8	282
5	Desarrollo de la primera parte de la memoria	12	270

Ilustración 4: Tabla de horas invertidas en la primera fase

La primera fase del proyecto consiste en marcar los objetivos del proyecto, realizar una búsqueda de información, hacer una comparativa de apps similares a la idea propuesta, preparar la planificación del proyecto y comenzar a estructurar los primeros apartados de la memoria.

Segunda fase - Diseño:

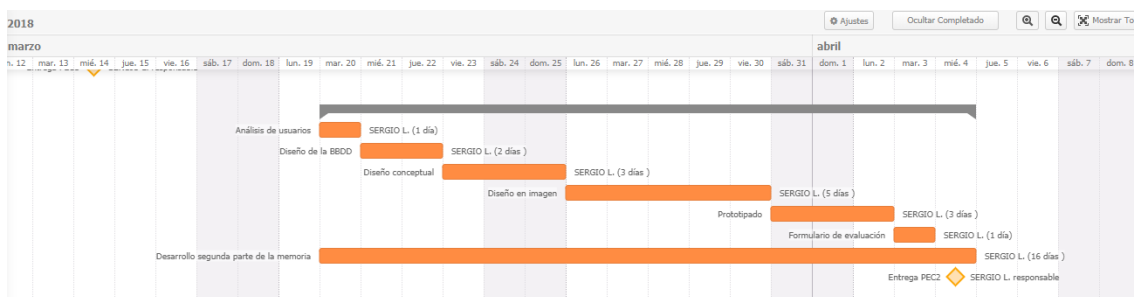


Ilustración 5: Segunda fase; Archivo Gantt

6	Análisis de usuarios	4	266
7	Diseño de la BBDD	6	260
8	Diseño conceptual	12	248
9	Diseño en imagen	20	228
10	Prototipado	16	212
11	Formulario de evaluación	2	210
12	Desarrollo segunda parte de la memoria	12	198

Ilustración 6: Tabla de horas invertidas en la segunda fase

La segunda fase del proyecto está orientada al diseño de la aplicación, identificando a los posibles usuarios del producto, desarrollado el diseño gráfico de las pantallas que componen la aplicación de forma conceptual (incluyendo los diagramas de flujo de la aplicación), para posteriormente crear, un diseño de alta fidelidad. Tras el diseño de alta fidelidad se

desarrolla el prototipo, al finalizar este, se proporciona un formulario de evaluación a varios usuarios. El objetivo del formulario es encontrar posibles fallos de usabilidad dentro del prototipo y evitar arrastrarlos en la versión final de la aplicación. Esta fase incluye también el diseño de la base de datos que esta implementada en la aplicación. La memoria se actualiza progresivamente conforme se realizan avances en las diferentes tareas.

Tercera fase – Implementación:

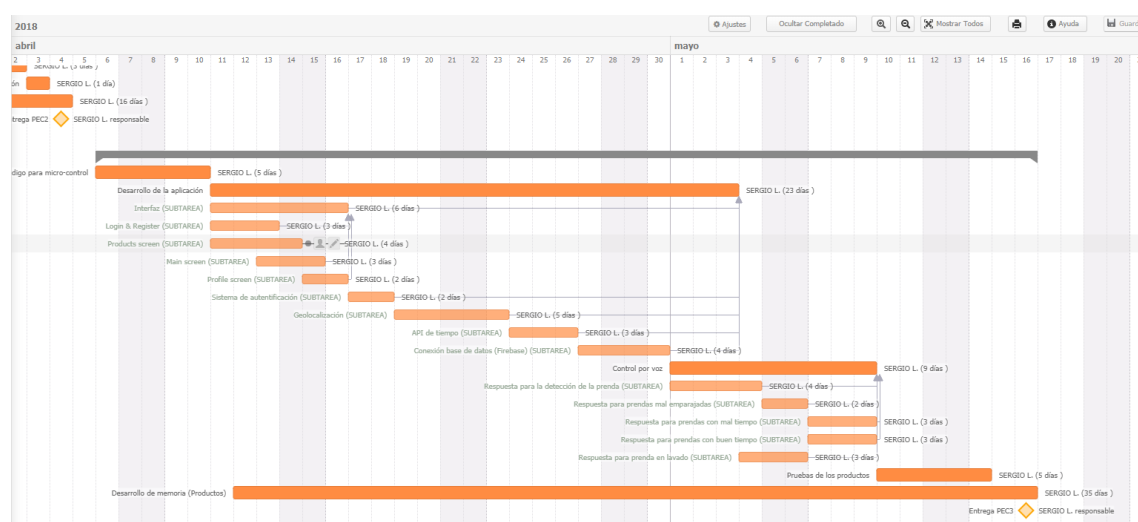


Ilustración 7: Tercera fase; Archivo Gantt

Desarrollo del código para micro-controladores	12	186
Desarrollo de la aplicación	72	114
Control por voz	40	74
Pruebas de productos	4	70
Desarrollo de la memoria (Productos)	16	54

Ilustración 8: Tabla de horas invertidas en la tercera fase

Durante la tercera fase, se da comienzo la producción de los productos, en esta fase se invierten la mayoría de horas preestablecidas y se dedican especialmente al desarrollo completo de la aplicación y el control por voz. Puesto que para este proyecto se van a emplear micro-controladores, es necesario establecer una tarea que implique la

resolución de un código para sus diferentes funciones. Tras desarrollar los productos, se realizan las correspondientes pruebas y se anotan los fallos detectados, para poder corregirlos en la siguiente fase del proyecto. La memoria, del mismo modo que en la fase anterior, se va actualizando progresivamente, al mismo tiempo que se realiza la producción de los productos.

Cuarta fase - Entrega final:

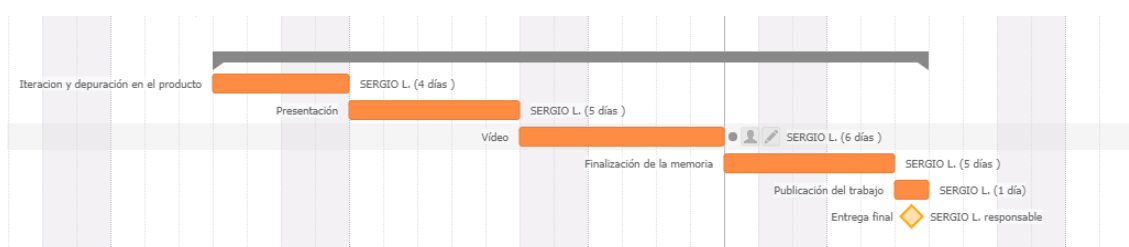


Ilustración 9: Cuarta fase; Archivo Gantt

18	Iteración y depuración en el producto	8	46
19	Presentación	12	34
20	Video	16	18
21	Finalización de la memoria	14	4
22	Publicación del trabajo	4	0

Ilustración 10: Tabla de horas invertidas en la cuarta fase

Tras la fase de producción quedan, 54 horas para finalizar el proyecto, en este apartado, se procede a la iteración sobre la aplicación y la depuración de los productos obtenidos. Por último, el resto del tiempo se dedica a la creación de la presentación del proyecto, un video explicativo del proyecto y finalizar la memoria.

Nota: Para mejor visualización de la planificación en Gantt y Excel ver anexos 1 y 2.

1.5 Breve resumen de productos obtenidos

Al finalizar el proyecto, se obtendrán los siguientes productos:

- Un lector NFC capaz de comunicarse con la base de datos de Firebase.
- Una aplicación para la gestión de las prendas con chip NFC, capaz de mostrar el estado de cada prenda y detectar fallos y asesorar a la hora de vestirse, facilitando esta acción a personas invidentes.
- Un sistema de control por voz que pueda ser utilizado mediante el asistente de google.
- Presentación para la defensa del proyecto.
- Video presentación del proyecto.
- Memoria que documente el proceso de creación y desarrollo en cada fase del proyecto.

Al final del proyecto se habrá desarrollado un ecosistema capaz de controlar el estado de las prendas mediante micro-controladores y tarjetas de lectura NFC/RFID que, en combinación con la aplicación, permitirán a los usuarios gestionar sus prendas de uso diario.

1.6 Recursos necesarios

Los recursos de hardware necesarios para este proyecto son:

- Dos microcontroladores entre los que se incluyen un ESP8266 y un lector NFC.
- Pegatinas NFC.
- Ordenador personal para el desarrollo.
- Dispositivo android con una versión de Android 4.4 (Lollipop) o superior.
- Cámara de video.
- Dispositivo Asistente personal Google Home (recurso opcional), a partir de la versión 4.4 de android es posible utilizar google assistant, en cualquier dispositivo móvil que incluya esta versión.

Los recursos de software y servicios necesarios para este proyecto son:

- IDE de programación Android Studio.
- Servicios de Firebase y Google Cloud.
- Servicio Dialogflow.
- Adobe Photoshop.
- Adobe Premiere.
- Adobe Illustrator.
- OBS Studio.
- JustInMind (Prototipo).
- Microsoft Word.
- Microsoft Excel.

1.7 Breve descripción de los otros capítulos de la memoria

En futuras secciones de este documento, se tratará en profundidad la fase de diseño, el análisis de usuarios, el diseño conceptual, el diseño gráfico, y la creación de un prototipo de alta fidelidad para la aplicación.

Después del apartado de diseño, se describe el proceso de desarrollo de la aplicación, donde se tratan las principales clases y funciones que, emplean la aplicación, los micro-controladores y el control por voz del asistente de google.

En los últimos capítulos, se comenta la viabilidad del producto, las conclusiones obtenidas tras la realización del proyecto, las fuentes que se han consultado, los anexos del documento y el glosario.

2. Diseño de la aplicación

2.1 Análisis

2.1.1 Recopilación de datos

Tras contextualizar y plantear los requerimientos de la aplicación, es posible realizar un estudio de los grupos de usuarios que accederán y darán uso de las distintas funcionalidades de la aplicación. Para la realización del estudio se han obviado aquellos usuarios que no tienen acceso a un dispositivo android en su versión 4.4 (KitKat), el cual representa el 90,1% de los dispositivos android registrados, también se discriminan aquellas personas que por restricciones de edad no tienen relación directa con la aplicación desarrollada.

La aplicación va dirigida de igual forma a personas invidentes y usuarios sin ningún tipo de defecto visual, por lo que existen dos grupos de usuarios con perfiles y contextos de uso muy diferenciados. Para el análisis de ambos grupos de usuarios, se ha realizado un estudio de tipo cualitativo. Este tipo de estudio permite conocer tanto el hardware, como el software, y los diferentes contextos de uso habitual que los usuarios utilizan para resolver el problema de gestión de prendas, el cual se quiere solventar de una forma más eficaz gracias al ecosistema propuesto. Para realizar con éxito el estudio, se han buscado diferentes testimonios de usuarios clave, estableciendo así cuáles son las necesidades reales de los usuarios potenciales de la aplicación.

2.1.2 Ámbito de difusión

El ámbito de difusión de la aplicación tiene dos vertientes, por una parte, se encuentra el componente de ayuda social a personas con discapacidad visual, mientras que, por otra parte, en combinación con las nuevas tecnologías RFID y NFC puede ser una aplicación de gran valor comercial en el sector de la moda. Por estos motivos la aplicación

tiene que ser intuitiva, accesible y simple, a cualquier perfil de usuario y que, de esta forma, sea capaz de atraer a nuevos usuarios debido a su componente innovador y social.

A medio y largo plazo es posible que sea necesario crear una campaña de difusión a través de redes sociales, para dar a conocer la aplicación mediante la ayuda de “influencers”¹³. También es interesante poder plantear un B2B¹⁴ con alguna cadena de ropa, donde además poder realizar tests más avanzados de la aplicación y colaborar en la difusión de la aplicación junto con la marca de ropa que apoye la iniciativa.

2.1.2 Contextos de uso

En este apartado es donde entran en juego los sistemas de detección inalámbricos RFID y NFC que permiten reconocer las prendas junto con la aplicación. Actualmente la aplicación ofrece tres estados, orientados a la ropa del usuario; disponibilidad en el armario, ropa lavándose y ropa sucia. Para actualizar estos estados es necesario conectar la ropa a la aplicación, esto solo puede lograrse combinando detectores y lectores de chips RFID y NFC. De forma hipotética, el objetivo es crear una línea de productos que ofrezca la posibilidad de leer y mandar los estados de la ropa a la aplicación, para ello sería necesario una línea de ropa con chips NFC y lectores RFID, que permitan al usuario registrar el estado de cada prenda de forma cómoda sin tener que ir una a una, a corta distancia.

¹³ La mercadotecnia influyente o marketing influyente es una forma de publicidad que ha surgido a partir de una variedad de prácticas y estudios recientes, enfocada más a los individuos que al mercado objetivo en su conjunto.

¹⁴ Negocio a negocio (del inglés business-to-business o B2B) hace referencia a las transacciones comerciales entre empresas, es decir, a aquellas que típicamente se establecen entre un fabricante y el distribuidor de un producto.

Debido al precio actual de los lectores RFID y NFC más potentes el mejor caso de uso doméstico que se le puede dar a la aplicación, es para personas con discapacidad visual, que quieren mejorar la accesibilidad en su vivienda, ya que, aunque no es un coste inalcanzable, sí que supone un desembolso relevante para el consumidor medio. No obstante, la domótica poco a poco se va involucrando cada vez más en el día a día de las personas y esto inevitablemente afectará de una forma u otra al mundo de la moda.

Gracias al internet de las cosas podemos controlar lo que queramos, cuando queramos y donde queramos, lo que nos deja un contexto de uso muy amplio, sin embargo, aquí se especifican tres casos posibles:

1. En casa, una persona invidente elige sus prendas de ropa de su armario, y se dispone a salir de paseo, en ese momento el detector RFID de su armario ha detectado que prendas ha extraído, una vez vestido le pregunta mediante comando de voz a la aplicación si su conjunto es correcto, al cual la aplicación responde que todo es correcto, pero que recuerde que hoy lloverá y no debe olvidar abrigarse.
2. Desde el trabajo, es posible consultar si el traje para una cena formal está en el armario o no se encuentra limpio.
3. En su domicilio, una persona desorganizada busca la forma de tener mayor control sobre su armario, apoyándose en una aplicación que registre ropa de forma eficiente.

Teniendo como premisa estos casos es necesario, considerar un diseño simple y accesible, que además se maneje de forma rápida y fluida, para que los usuarios puedan consultar la aplicación sin frustraciones ni esperas innecesarias.

2.2 Diseño conceptual

2.2.1 Perfiles de usuario y posibles escenarios

Al ofrecer una aplicación relacionado con un producto de difusión tan generalista como es el caso de la ropa de uso diario, existe un rango muy amplio de edades, géneros, ingresos y sobre todo un nivel de conocimiento informático y tecnológico muy variable. Sin embargo, y teniendo como precedente las cadenas de ropa como H&M, Zara o G.A.P.¹⁵, las cuales ofrecen un catálogo muy amplio a un público muy general; se define para la aplicación el rango de edad de los usuarios entre 25 y 55 años, teniendo en cuenta que la mayoría de usuarios con discapacidad visual grave tienen un perfil de edad superior a los 50 años¹⁶, sin este dato, la edad del público objetivo quedaría reducido a 25-40 años. Se puede identificar una mayor tendencia de descarga de aplicaciones de moda en el género femenino, ya que muestran una mayor preocupación por este tema concreto, por lo que la aplicación se centra en usuarios potenciales mayormente de sexo femenino, con un rango de edad de 18 a 55 años, que tengan cierto interés por la moda.

A continuación, se exponen tres fichas de usuarios donde se definen los perfiles más comunes de la aplicación y los posibles escenarios de uso:

¹⁵ Fernando del Pozo, "El perfil del consumidor es la clave", <http://www.portafolio.co/economia/finanzas/perfil-consumidor-clave-fernando-pozo-experto-internacionalizacion-marcas-199576>

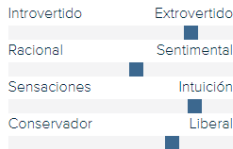
¹⁶ OMS, "Ceguera y discapacidad visual", Nota descriptiva, <http://www.who.int/mediacentre/factsheets/fs282/es/>



"Siempre en movimiento."

Edad: 28
 Trabajo: Diseñador web
 Familia: En pareja
 Localización: Madrid, España
 Carácter: Activo

Personalidad



- Atlético
- Geek
- Viajero
- Desorganizado

Retos

- Probar nuevas programas y aplicaciones.
- Perder menos tiempo buscando ropa en su armario.
- Encontrar la forma de organizar y lavar su ropa de forma eficiente.

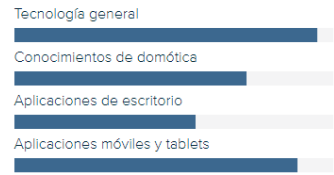
Frustraciones

- Perder tiempo en combinando o buscando ropa en su armario.
- Mala organización en su vestimenta.
- Miedo a llegar tarde al trabajo.

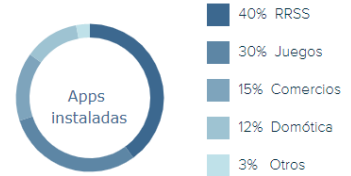
Biografía

Javier es un diseñador web, apasionado de la tecnología. Es un apasionado de la domótica y siempre busca la forma de vincular sus electrodomésticos y aparatos electrónicos a su smartphone. Sin embargo, todavía no ha encontrado la forma de conectar algo tan común como puede ser su vestuario. Le gusta vestir de forma casual, pero a la moda y si es posible con algunas marcas influyentes.

Motivación



Aplicaciones instaladas



Marcas / Influencias



Canales de comunicación

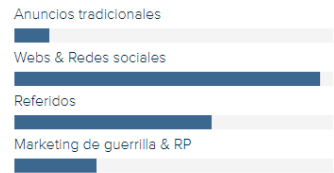


Ilustración 11: Ficha persona I

Escenario de uso:

Javier tiene una reunión de trabajo a primera hora de la mañana, con unos clientes importantes para su empresa. Preocupado por la impresión que debe causar, al llegar a casa busca una camisa elegante que hace varios meses que no se pone, sin embargo, no la encuentra por ninguna parte, tras buscar por todo su armario decide ir al cesto de la ropa sucia y descubre que esta se encuentra en el fondo del mismo. Por desgracia ya no podrá utilizarla porque, aunque consiga poner una lavadora, la camisa no se secará a tiempo. Frustrado piensa como es posible que pueda controlar a distancia la temperatura de su hogar, pero no tener ningún control sobre una simple camisa.

Tommy Edison
Xtensio



"This is how I see the world"

Edad: 50
Trabajo: Blind Film Critic, YouTuber, Radio Personality, and Public Speaker
Familia: En pareja
Localización: Los Angeles, CA
Carácter: Apasionado

Invidente
Geek
Atrevido

Retos

- Ofrecer un nuevo punto de vista sobre personas ciegas a todo el mundo.
- Ser completamente independiente.
- Poder organizarse rápidamente sin necesidad de ayuda.

Frustraciones

- Pedir la ayuda de otras personas a la hora de vestirse o comprar ropa.
- No poder encontrar algo.
- No poder adaptarse bien a ciertas situaciones que salen de su control.

Motivación

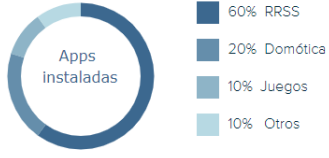
Tecnología general

Conocimientos de domótica

Aplicaciones de escritorio

Aplicaciones móviles y tablets

Aplicaciones instaladas



- 60% RRSS
- 20% Domótica
- 10% Juegos
- 10% Otros

Marcas / Influencias

Canales de comunicación

Anuncios tradicionales

Webs & Redes sociales

Referidos

Marketing de guerrilla & RP

Personalidad

Introvertido

Extrovertido

Racional

Sentimental

Sensaciones

Intuición

Conservador

Liberal

Biografía

Tommy es una persona ciega desde que nació actualmente produce videos para su canal de youtube que cuenta con más de 600mil subscribers. Se siente completamente adaptado al mundo que le rodea, pero debido a su discapacidad siempre existe alguna situación que sale de su control. Siempre que puede trata de ser independiente y valerse por si mismo apoyándose de nuevas tecnologías. En su casa tiene múltiples dispositivos adaptados a su discapacidad que le ayudan en el día día.

Ilustración 12: Ficha persona II

Escenario de uso:

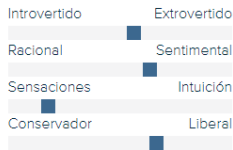
La madre de Tommy ha ido a su casa a dejarle unos pantalones nuevos que le acaba de comprar, sin darse cuenta los deja en un lugar erróneo del armario, lo que provoca que la cadena de orden que tiene establecida James a la hora de escoger su ropa se rompa y tenga que verificar prenda por prenda que estas se encuentran en un orden correcto.



"Me gusta la moda, compartirla y vivirla"

Edad: 26
 Trabajo: Community manager
 Familia: Soltera
 Localización: Madrid, España
 Carácter: Coqueta

Personalidad



Moda Geek Apasionada

Retos

- Compartir su pasión por la moda con todos sus conocidos.
- Estar siempre actualizada de las últimas tendencias en moda.
- Saber que tiene en el armario.

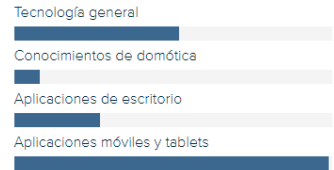
Frustraciones

- No conseguir conjuntar vestimenta porque no está disponible ese día.
- Pensar cada mañana que ponerse y perder excesivo tiempo.

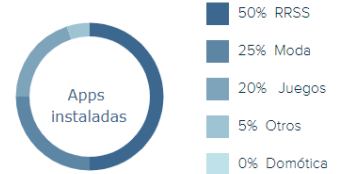
Biografía

Sara es una apasionada de la moda, le encanta ir a sus tiendas favoritas y comprarse lo último en tendencias. En su móvil ya tiene instaladas varias aplicaciones que le permiten crearse sus outfits por la noche y al día siguiente conjuntarse rápidamente, pero en muchas ocasiones se lamenta de no haber puesto su ropa a lavar a tiempo.

Motivación



Aplicaciones instaladas



Marcas / Influencias



Canales de comunicación

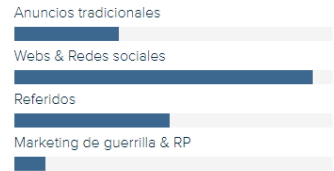


Ilustración 13: Ficha persona III

Escenario de uso:

Sara lleva meses buscando la aplicación perfecta para combinar su armario y su Smartphone en uno solo. La mayoría de aplicaciones son de un carácter más lúdico que práctico, ya que le resulta muy tedioso tener que fotografiar cada prenda una a una. Además, aparte de poder compartir sus conjuntos con otras personas no les encuentra una verdadera utilidad ya que muchas veces los outfits que se prepara para la mañana siguiente han de cambiarse debido al cambiante tiempo meteorológico de su ciudad.

2.2.2 Descripción de los diferentes casos de uso

El diagrama de flujo de la aplicación se muestra en la [figura 14](#) (Árbol de navegación), a continuación, se explican los diferentes casos de uso en cada pantalla:

Caso de uso	Registro
Descripción	El usuario descarga la aplicación e inicia sesión por primera vez, por lo que debe registrarse.
Actor/es	Usuario estándar
Precondiciones	No se cuenta con ningún registro previo del usuario
Postcondiciones	El usuario se registra en la aplicación.
Escenario principal	<ol style="list-style-type: none">1. El usuario se descarga la aplicación por primera vez.2. Al iniciar la aplicación el sistema le indica que debe registrarse, en la pantalla de <i>Register</i>.3. El usuario se registra introduciendo los datos pedidos, quedando guardados en la base de datos.

Caso de uso	Sesión
Descripción	El usuario ya dispone de una cuenta en la aplicación. En este caso se verifica si dicha sesión esta iniciada.
Actor/es	Usuario estándar
Precondiciones	El usuario dispone de una cuenta de usuario.
Postcondiciones	Verificación del estado de sesión del usuario.
Escenario principal	<ol style="list-style-type: none"> 1. La sesión del usuario no se encuentra activa por lo que debe iniciarla, desde la pantalla de <i>Login</i>. 2. El estado de la sesión del usuario es activo, por lo que pasa directamente al main activity de la aplicación.

Caso de uso	Recuperación de contraseña
Descripción	El usuario ha olvidado su contraseña y quiere recuperarla.
Actor/es	Usuario estándar
Precondiciones	El usuario dispone de una cuenta, pero no puede acceder a ella.
Postcondiciones	Se envía un email de recuperación de contraseña al usuario.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario introduce un email valido y previamente registrado, en la pantalla de <i>Password Reset</i> 2. El usuario presiona el botón de recuperar contraseña. 3. El sistema un email con instrucciones para recuperar la contraseña.

Caso de uso	Comando de voz
Descripción	El usuario realiza una pregunta o afirmación al sistema.
Actor/es	Usuario estándar
Precondiciones	El usuario dispone de una sesión activa.
Postcondiciones	El sistema responde al usuario con una respuesta adecuada a su pregunta o afirmación.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario activa el botón central de la <i>Main Activity</i>, el cual activa el micrófono del dispositivo. 2. El usuario realiza una petición al sistema mediante un comando de voz. 3. El sistema aporta nueva información o actúa en consecuencia al comando de voz dado.

Caso de uso	Añadir prenda
Descripción	El usuario quiere añadir una prenda a su armario virtual.
Actor/es	Usuario estándar
Precondiciones	El usuario dispone de una prenda con chip NFC/RFID.
Postcondiciones	La prenda se añade al armario virtual del usuario.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario tiene una prenda con un chip NFC/RFID registrado en la base datos. 2. El usuario acerca la prenda a su dispositivo móvil con la pantalla <i>Add Cloth</i> activa. 3. Se detecta el chip NFC/RFID de la prenda de ropa. 4. Se deriva al usuario a la pantalla de <i>Cloth Detail</i> donde seleccionando el botón de añadir prenda, está queda guardada en el armario virtual del usuario.

Caso de uso	Eliminar prenda
Descripción	El usuario se dispone a eliminar una prenda de su armario virtual.
Actor/es	Usuario estándar
Precondiciones	El usuario dispone de una o más prendas registradas en su armario virtual.
Postcondiciones	La prenda se elimina del armario virtual del usuario.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona una prenda desde la pantalla de <i>My Clothes</i>. 2. Se muestra la prenda en la pantalla de <i>Cloth Detail</i>. 3. El usuario marca la opción de eliminar prenda, para desvincular dicha prenda de su armario.

Caso de uso	Activar/Desactivar outfit
Descripción	El usuario puede activar o desactivar un outfit para que el sistema le haga recomendaciones y le asesore sobre el mismo.
Actor/es	Usuario estándar
Precondiciones	El usuario dispone de una o más prendas registradas en su armario virtual y ha creado al menos un outfit en la pantalla de <i>Outfits</i>
Postcondiciones	Se activa o desactiva un outfit creado por el usuario.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario marca o desmarca un outfit en la pantalla de <i>Outfits</i>.

Caso de uso	Crear un outfit
Descripción	El usuario puede seleccionar una o varias prendas de su armario virtual para crear un nuevo outfit.
Actor/es	Usuario estándar
Precondiciones	El usuario dispone de una o varias prendas en su armario virtual.
Postcondiciones	Se registra un nuevo outfit para el usuario.
Escenario principal	<ol style="list-style-type: none"> 1. Desde la pantalla de <i>Outfits Editor</i> el usuario indica un nuevo nombre para su outfit. 2. Se seleccionan las prendas que conforman el outfit desde la pantalla <i>Pick Cloth</i>. 3. El usuario guarda el outfit desde el botón de guardar outfit.

Caso de uso	Editar outfit
Descripción	El usuario desea modificar uno de los outfits que ya ha creado
Actor/es	Usuario estándar
Precondiciones	El usuario ha creado al menos un outfit.
Postcondiciones	Se modifican los valores del outfit elegido por el usuario.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona el outfit que quiere modificar desde la pantalla de <i>Outfits</i> 2. Desde la pantalla de <i>Outfit Editor</i> el usuario modifica sus prendas y valores del outfit. 3. El usuario presiona el botón de guardar outfit para registrar los valores modificados.

Caso de uso	Eliminar outfit
Descripción	El usuario desea eliminar uno de los outfits que ya ha creado
Actor/es	Usuario estándar
Precondiciones	El usuario ha creado al menos un outfit.
Postcondiciones	Se elimina el outfit elegido por el usuario.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona el outfit que quiere eliminar desde la pantalla de <i>Outfits</i> 2. Desde la pantalla de <i>Outfit Editor</i> el usuario selecciona el botón de eliminar outfit.

Caso de uso	Añadir/modificar imagen de perfil
Descripción	Se añade o se cambia la imagen de perfil del usuario activo.
Actor/es	Usuario estándar
Precondiciones	La sesión del usuario se encuentra activa.
Postcondiciones	Se añade o se modifica la imagen de perfil del usuario.
Escenario principal	<ol style="list-style-type: none"> 1. Se selecciona el botón de modificar imagen de perfil desde la pantalla de <i>Main Activity</i> o <i>Settings</i> 2. Se abre la galería de imágenes del dispositivo y el usuario selecciona una imagen de perfil. 3. Se modifica la imagen de perfil.

Caso de uso	Modificar el nombre del usuario
Descripción	Cambia el nombre del usuario.
Actor/es	Usuario estándar
Precondiciones	La sesión del usuario se encuentra activa en la aplicación.
Postcondiciones	Se añade o modifica el nombre del usuario.
Escenario principal	1. El usuario añade o modifica su nombre de usuario, desde la pantalla <i>Name</i> .

Caso de uso	Modificar el email del usuario
Descripción	Se modifica el email vinculado a la cuenta del usuario.
Actor/es	Usuario estándar
Precondiciones	El usuario está registrado y tiene la sesión activa.
Postcondiciones	Se modifica el email del usuario en la base de datos.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario introduce su contraseña y su nuevo email dos veces en la pantalla <i>Email</i>. 2. El usuario selecciona el botón de guardar cambios. 3. Se modifica el email en la base de datos de la aplicación y se cierra la sesión del usuario.

Caso de uso	Cambio de contraseña
Descripción	Se modifica la contraseña de acceso a la cuenta del usuario.
Actor/es	Usuario estándar
Precondiciones	El usuario está registrado y tiene la sesión activa.
Postcondiciones	Se modifica la contraseña del usuario en la base de datos.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario introduce su antigua contraseña, la nueva contraseña y su confirmación en la pantalla de <i>Account Security</i>. 2. El usuario selecciona el botón de guardar cambios. 3. Se modifica el email en la base de datos de la aplicación y se cierra la sesión del usuario.

Caso de uso	Cerrar sesión
Descripción	El usuario termina la sesión dentro de la aplicación.
Actor/es	Usuario estándar
Precondiciones	El usuario se encuentra con la sesión de usuario activa.
Postcondiciones	Se cierra la sesión del usuario activo.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona el botón de cerrar sesión, desde la pantalla <i>Settings</i>. 2. Se cierra la sesión de usuario y este es llevado a la pantalla de <i>Login</i>.

Caso de uso	Eliminar cuenta
Descripción	Se elimina la cuenta del usuario de la base de datos.
Actor/es	Usuario estándar
Precondiciones	El usuario está registrado y tiene la sesión activa.
Postcondiciones	Se elimina la cuenta del usuario.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario selecciona el botón de borrar cuenta, desde la pantalla <i>Manage Account</i>. 2. El usuario introduce su contraseña en formato pop-up. 3. Si la contraseña es correcta se elimina la cuenta del usuario y se deriva al mismo a la pantalla de <i>Login</i>.

2.2.3 Diagrama de flujo

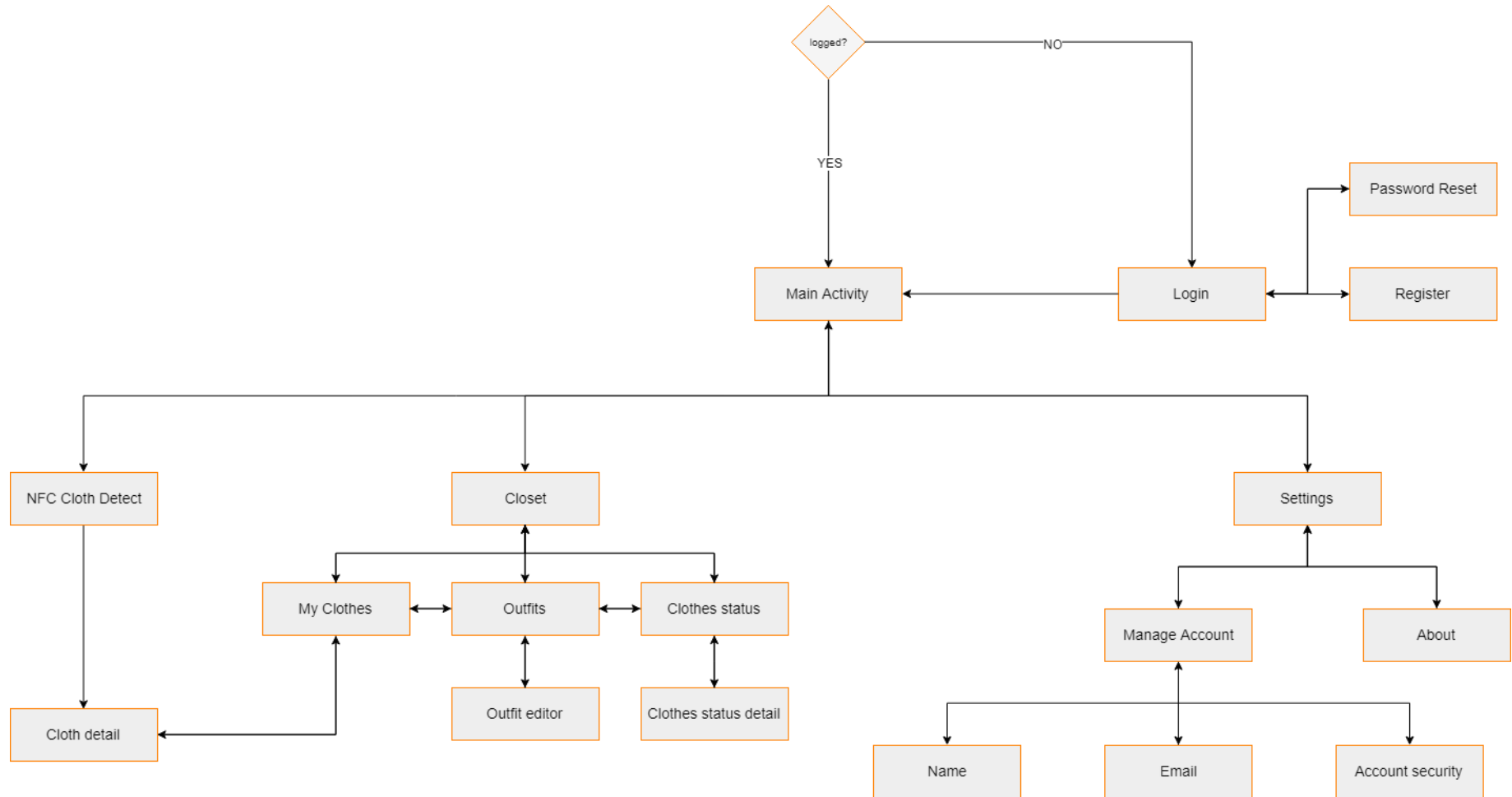


Ilustración 14: Árbol de navegación.

La aplicación se divide en tres secciones, la primera sección (comenzando por la izquierda) es *NFC Cloth Detect*, esta sección se encarga de gestionar las prendas del usuario vía NFC permitiéndole incluir nuevas prendas a su armario virtual. La segunda sección es la sección de *Closet* donde los usuarios pueden ver sus prendas añadidas, crear o editar sus outfits y ver su estado. Por último, la sección de *Settings* que permite al usuario gestionar su cuenta. Hay que tener en cuenta que el sistema verifica si el usuario está o no logeado dentro de la aplicación, si el usuario no cuenta con una sesión iniciada es llevado a la pantalla de login de la aplicación, donde puede gestionar su inicio de sesión. Una vez el usuario cuenta con una sesión abierta es llevado al *Main Activity* de la aplicación.

El tipo de navegación escogida para la aplicación no es lineal, es decir, no se dispone de un menú para viajar entre secciones y pantallas en cada una de ellas, sino que es necesario pasar por el *Main Activity* de la aplicación para llegar de una sección a otra. Esto se debe a que los usuarios pasarán el mayor tiempo de uso de la aplicación en el *Main Activity*, por lo que el resto de funcionalidades pasan a un segundo plano, evitando de esta forma tener pantallas excesivamente recargadas de opciones y menús complejos.

2.3 Arquitectura y base de datos

2.3.1 Diseño de la base de datos

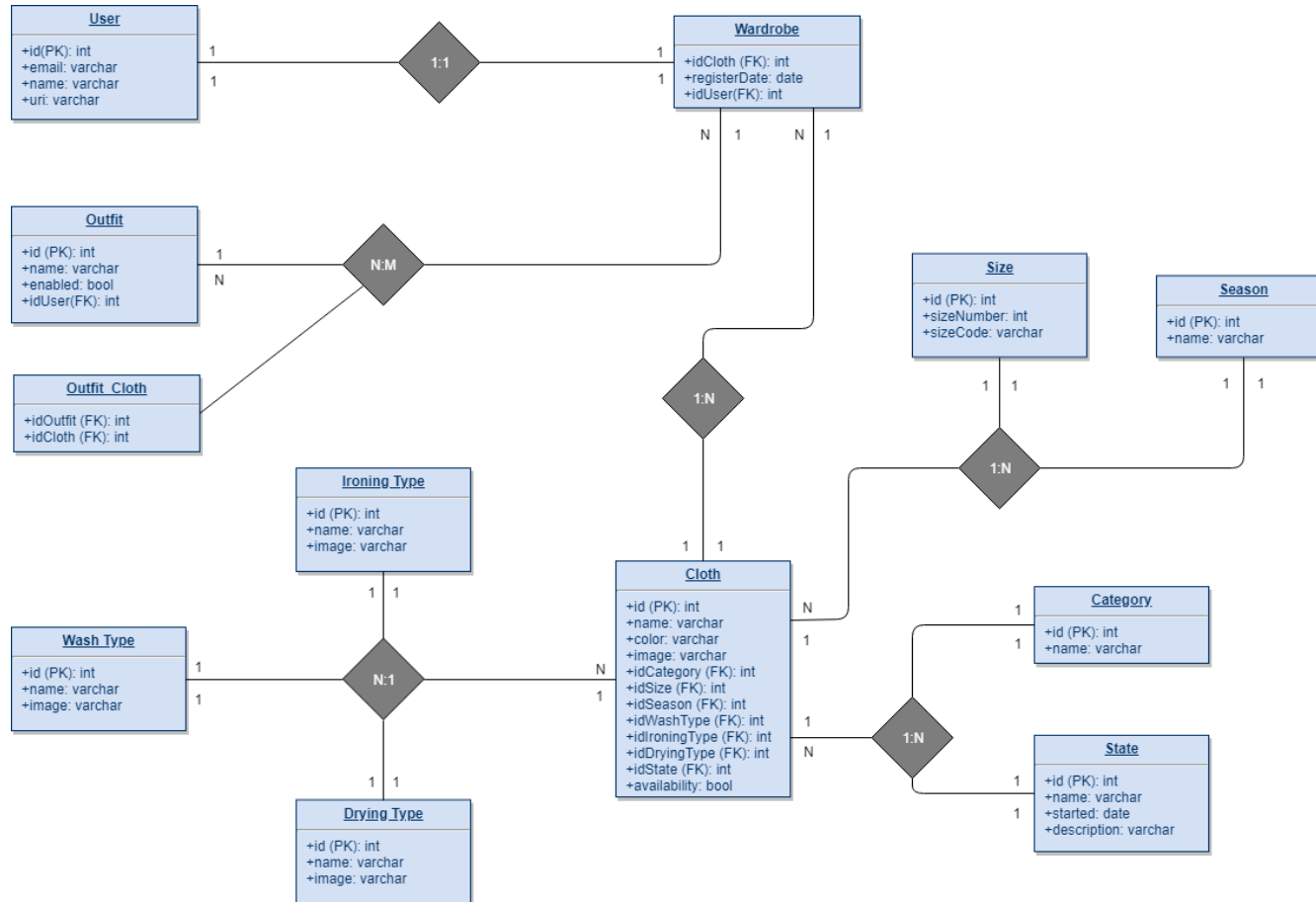


Ilustración 15: Diagrama de la Base de datos

Aunque el diseño de la base de datos sea un modelo relacional SQL, el uso de Firebase como base de datos obliga a utilizar un modelo noSql, que basa su almacenamiento de datos en un árbol JSON alojado en la nube. Sin embargo, el diagrama de la base de datos se puede utilizar para predefinir las claves y atributos de cada rama.

La distribución de la base de datos se plantea de forma que los usuarios tengan un armario donde almacenen las prendas que registran mediante NFC. Todas las prendas se encuentran ya disponibles en la sección cloths de la base de datos y los usuarios pueden añadirlas a su armario virtual. La gestión de los outfits que realizan los usuarios viene dada por las prendas registradas en su armario y su gestión propia. Las prendas, ya establecidas en la base de datos dentro del apartado cloth, disponen de varias tablas vinculadas, que aportan información detallada de los estados, categorías y tipos de limpieza que tienen las prendas.

2.3.2 Diagrama de clases

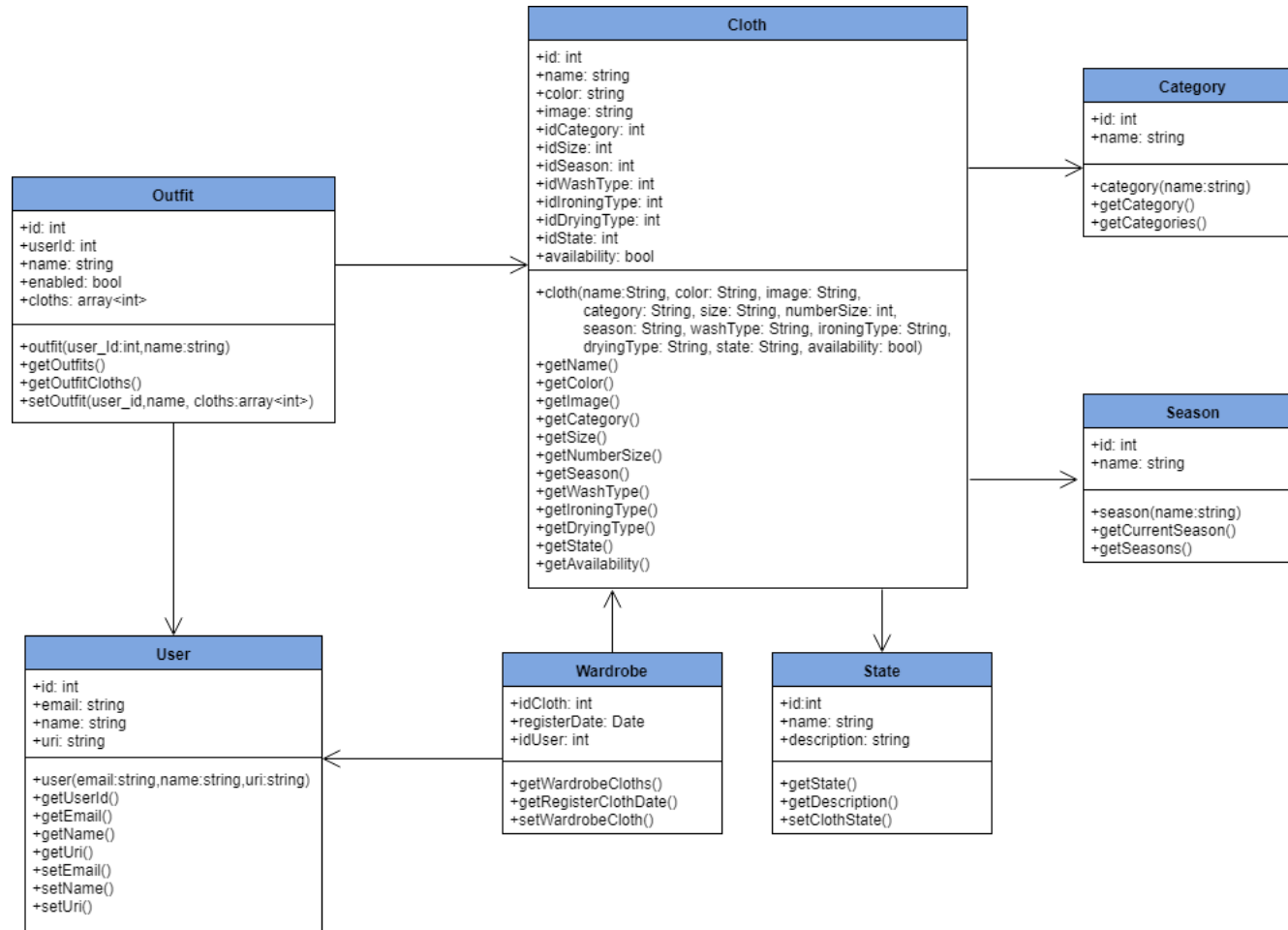


Ilustración 16: Diagrama de clases

Además de las clases expuestas en la ilustración 16, se han de tener en cuenta las llamadas al API de tiempo metereológico y Dialogflow, ya que forman parte de la implementación de clases dentro de la aplicación.

En el API de [Darksky](#) utilizaremos la llamada currently pasando la geo posición del usuario mediante un método GET y la url del api de Darksky.

En Dialogflow el sistema de llamadas se vuelve algo más complejo ya que se han de crear llamadas en función de lo que especifique el usuario, esto se resuelve entrenando varios intents¹⁷, que permiten reconocer palabras clave que emplea el usuario en las oraciones que formula.

Aunque en el proceso de desarrollo de la aplicación se pueden modificar, a continuación, se exponen los intents necesarios para la correcta implementación de Dialogflow:

Intent	Palabras clave
¿Voy bien conjuntado ?	“conjuntado”, “vestido”, “combinado”
¿Qué me pongo ?	“pongo”, “llevar”, “utilizo”
¿Hace buen tiempo ?	“tiempo”, “temperatura”
¿Tengo algún outfit preparado?	“outfit”, “conjunto”, “trajes”
¿Hay alguna lavadora puesta ?	“lavadora”, “maquina”, “puesta”, “encendida”, “activada”
¿Qué ropa está en el cubo de la ropa sucia ?	“sucia”, “sin lavar”, “cubo”, “cesto”, “ropa”, “prendas”
¿Hay ropa secándose ?	“ropa”, “prendas”, “secándose”, “tendidas”, “sol”, “secadora”

¹⁷ Un intent representa el mapeado a través de las palabras que dicta el usuario y la acción que debe ser tomada por el software.

Estas son las palabras clave, que Dialogflow utiliza para responder de una forma u otra al usuario y activar determinadas funciones que modifican la base de datos. Al tratarse de un proceso de entrenamiento y una tecnología innovadora, es difícil prever con certeza todos los casos, pero se ha tratado cada posible interpretación del usuario en cada pantalla.

2.3.3 Arquitectura del sistema

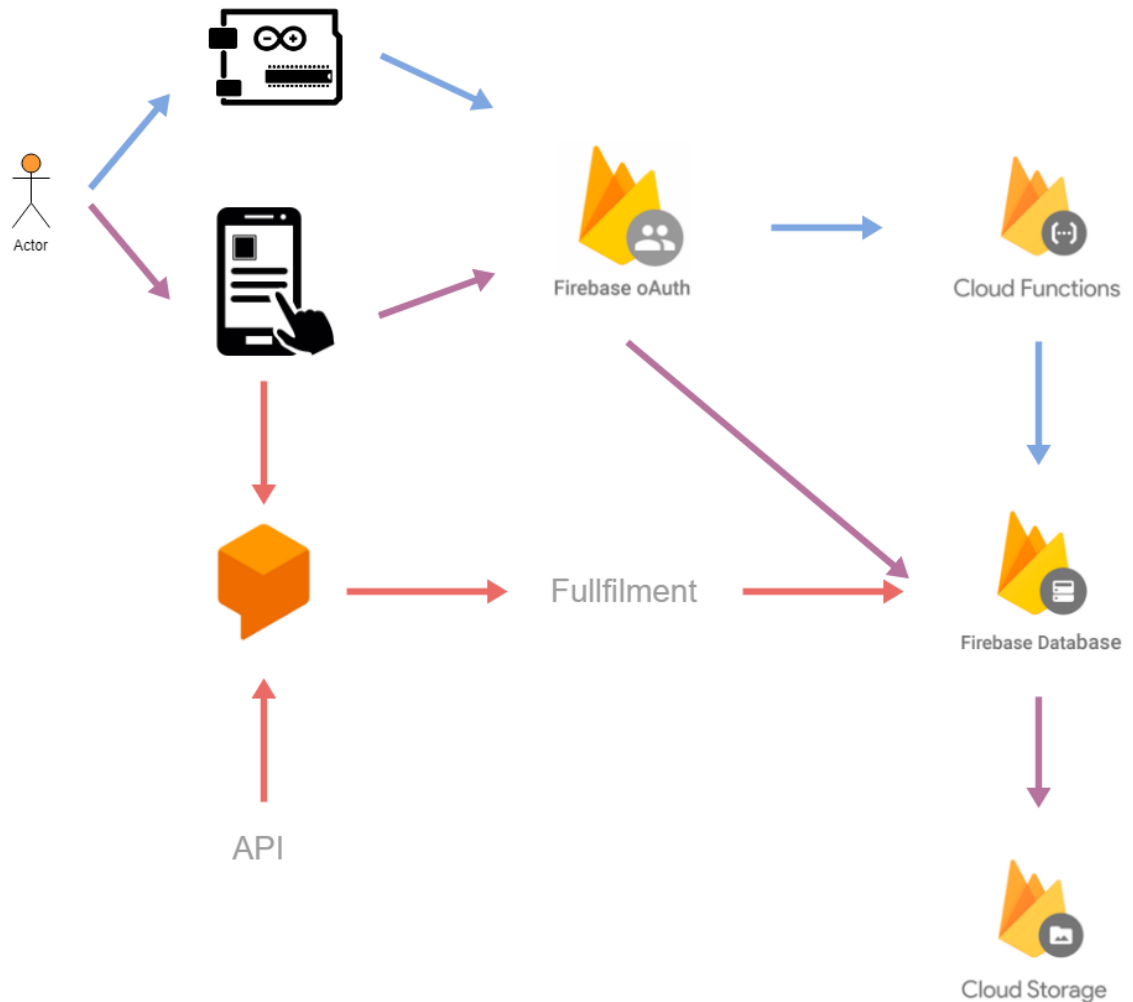


Ilustración 17: Arquitectura del sistema

Para el sistema cloud de la aplicación se utiliza la plataforma Firebase, ya que sus servicios proporcionan todo lo necesario para el correcto desarrollo y funcionamiento del proyecto. Los comandos de voz se gestionan mediante Dialogflow plataforma que a su vez se beneficia de una API para la gestión del tiempo atmosférico. A través de un fullfilment se modifican los parámetros de la base datos.

La arquitectura del sistema plantea un escenario en el que el usuario puede interactuar con su dispositivo android y los microprocesadores de forma completamente independiente. El objetivo es que las prendas se

auto-gestionen mediante los microprocesadores, para ello estos se conectan utilizando el método de autenticación OAuth, una vez se identifica al usuario, las funciones establecidas dentro del apartado de Cloud Functions se encargan de modificar los valores en la base de datos, evitando cargar pesadas funciones en los microprocesadores. Para que el usuario pueda ver y modificar sus datos en la aplicación ha de establecer de igual manera que los microcontroladores, una conexión mediante OAuth.

Por último, se utiliza el servicio Cloud Storage para guardar las imágenes que utiliza la aplicación y de esta forma evitar ocupar espacio de forma innecesaria en el dispositivo del usuario.

Hay que tener en cuenta que por motivos de planificación el microcontrolador utilizado para este proyecto contiene en su código el método de autenticación del usuario, sin embargo, en el apartado de líneas futuras de este documento se establece que la conexión entre microcontrolador y aplicación móvil debe quedar implementada para obtener un método de autenticación de usuarios mucho más eficaz.

2.4 Prototipado

2.4.1 Líneas de diseño

Para la creación del prototipo de alta fidelidad se ha utilizado el programa [justinmind](#), que permite la creación de pantallas mediante el uso de widgets personalizables. También se ha empleado adobe Photoshop y adobe Illustrator para la edición de imágenes y el diseño del logo de la aplicación.

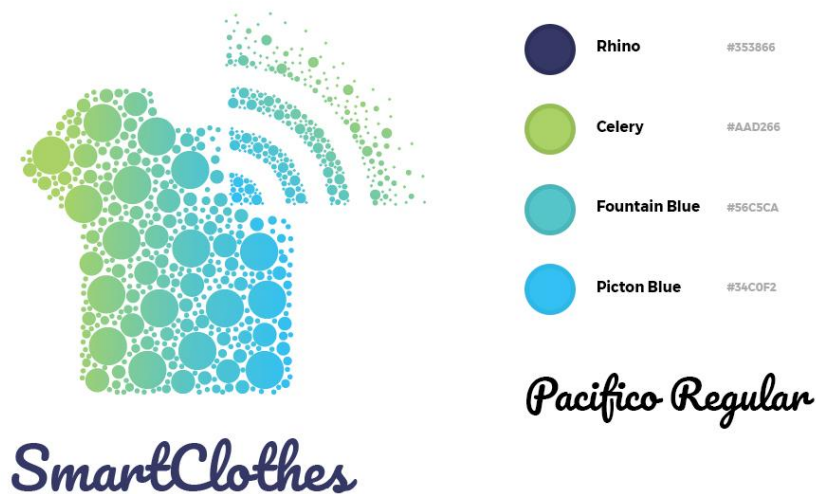









Ilustración 18: Guía de diseño para logotipo

Para definir el estilo visual de la aplicación se han seguido las líneas de diseño de Material Design establecidas por Google, ya que al tratarse de una aplicación exclusiva para dispositivos android, hace que esta normativa de diseño sea la idónea a la hora de crear un diseño familiar para los usuarios de este tipo de dispositivo. Para aportar un carácter más personalizado a la aplicación se han pautado los colores, tipografías, y elementos básicos en una guía de diseño propia, evitando así la homogeneidad con el resto de aplicaciones que siguen las líneas de diseño de forma estricta.

PALETA DE COLORES

	Rhino	#353866
	Wistful	#AAABD3
	Lilac	#CBA6C3
	Zircon	#F8FAFF
	Athens Gray	#E9EAEE
	Gray	#808080
	Tundora	#434343

TIPOGRAFÍAS

Display4 112sp

Fuente: Roboto Light / Color: #353866

Display3 56sp

Fuente: Roboto Regular / Color: #353866

Display2 45sp

Fuente: Roboto Regular / Color: #353866

Display1 35sp

Fuente: Roboto Regular / Color: #353866

Headline 24sp

Fuente: Roboto Regular / Color: #F8FAFF

Title 20sp

Fuente: Roboto Medium / Color: #F8FAFF

Subheading 16sp

Fuente: Source Sans Pro Regular / Color: #F8FAFF

Body2 14sp

Fuente: Source Sans Pro Semi-Bold / Color: #434343

Body 14sp

Fuente: Source Sans Pro Regular / Color: #808080

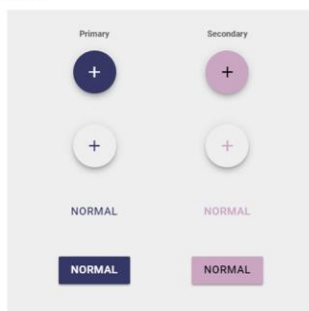
Caption 12sp

Fuente: Source Sans Pro Regular / Color: #808080

Button (ALL CAPS)14sp

Fuente: Source Sans Pro Regular / Colors: #F8FAFF #353866

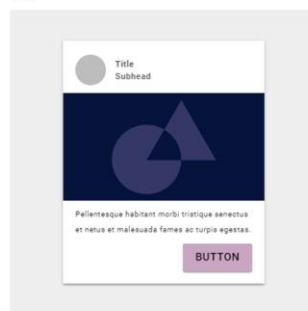
Buttons



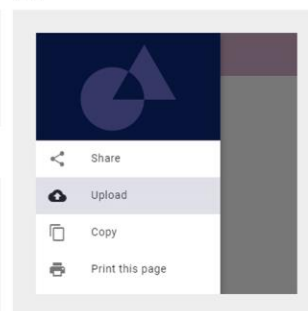
Switches and sliders



Cards



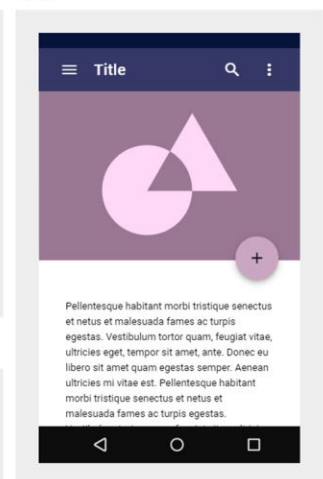
Menu



Text Field



Tool Bar



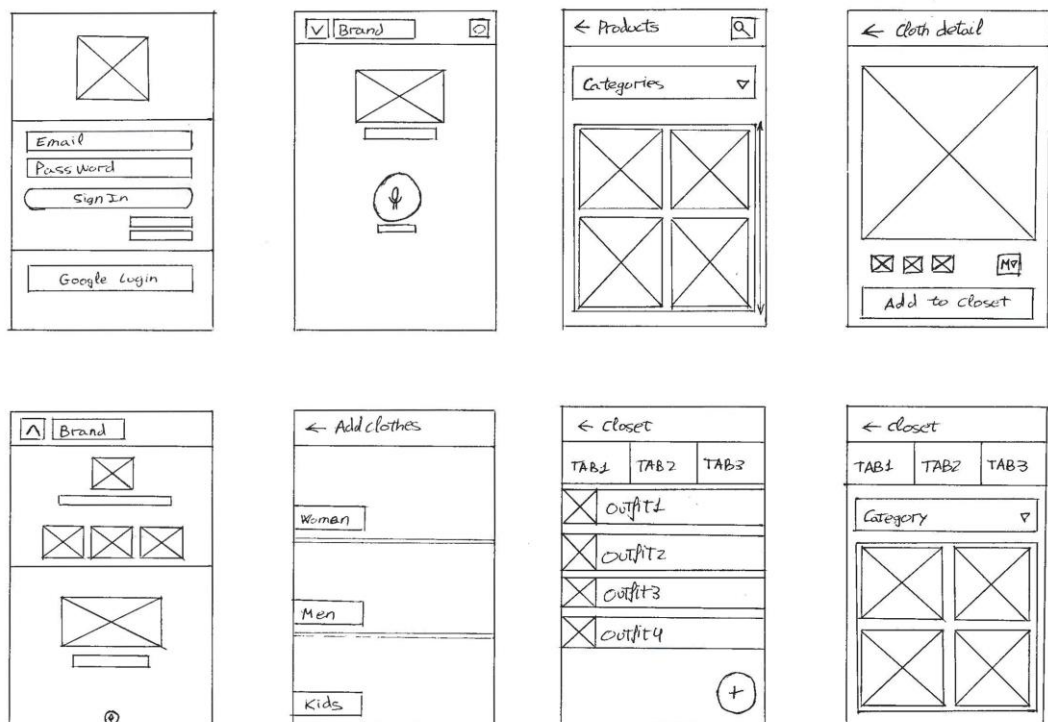
La paleta de colores escogida se basa en colores fríos que se encuentran entre el azul real y el lavanda que en combinación con varias tonalidades de grises y blancos, hace que el diseño refleje amplitud y seriedad. Al disponer de tantas opciones dentro de la aplicación se ha tratado de espaciar el máximo posible dichas opciones buscando la simplificación de las mismas, se quiere, de esta forma dar la sensación de una aplicación simple, pero con múltiples opciones.

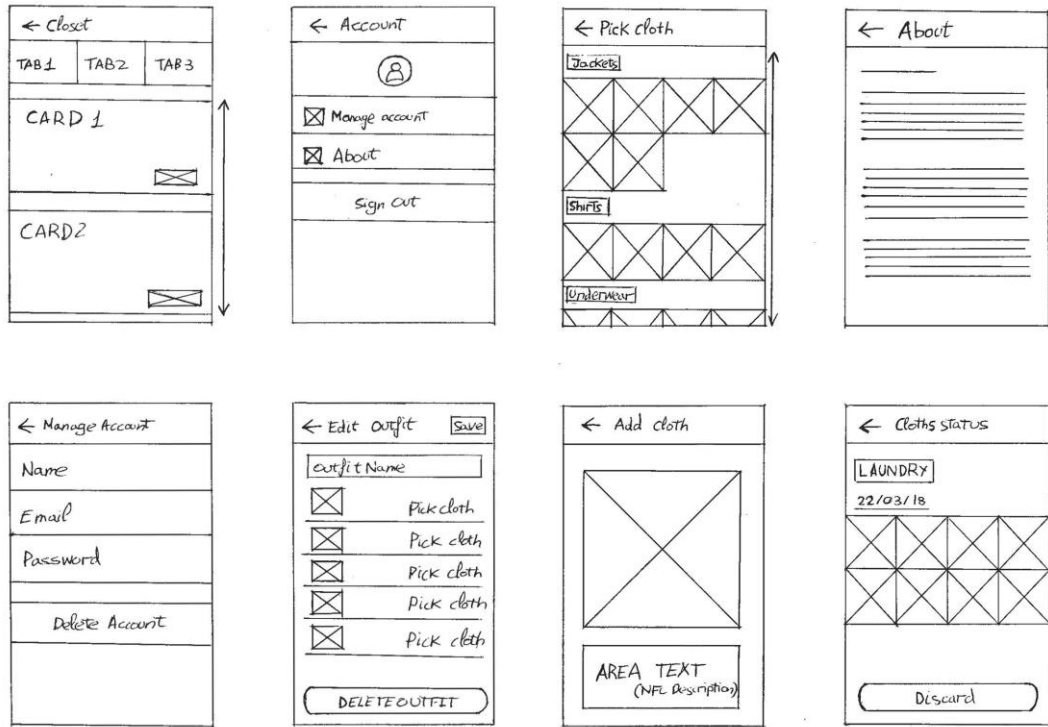
Las tipografías escogidas son Pacifico Regular en el caso del texto de la marca, Roboto Regular y Medium para títulos y textos más grandes dentro de la aplicación y Source Sans Pro para los cuerpos del texto de la aplicación y botones. A excepción de la tipografía Pacifico que se trata de una tipografía “handmade”¹⁸ que aporta elegancia y personalidad a la marca de la aplicación, las tipografías más utilizadas son Sen Serif (Sin serifa). Estas tipografías permiten que todos los elementos de la aplicación puedan leerse correctamente en diversos tamaños y ofrecen un diseño moderno y simplificado.

¹⁸ Handmade: Término anglosajón que hace referencia a un elemento hecho a mano.

2.4.2 Prototipo de baja fidelidad

Los prototipos de baja fidelidad se han realizado a mano, para poder tratar de obtener el máximo número de ideas posibles, en un tiempo óptimo. Tras varias pruebas y establecer ciertos criterios en cada pantalla, se han obtenido los siguientes prototipos de baja fidelidad, sin embargo, no todas las pantallas que se exponen en las imágenes han terminado implementándose en el prototipo de alta fidelidad o en la versión final de la aplicación, ya que algunas han sido modificadas o descartada.





Estos prototipos han sido diseñados tomando como referencias varias aplicaciones de tiendas de ropa como h&m o zara, google photos y habitica (app para la gestión de hábitos), entre otras.

2.4.2 Prototipo de alta fidelidad

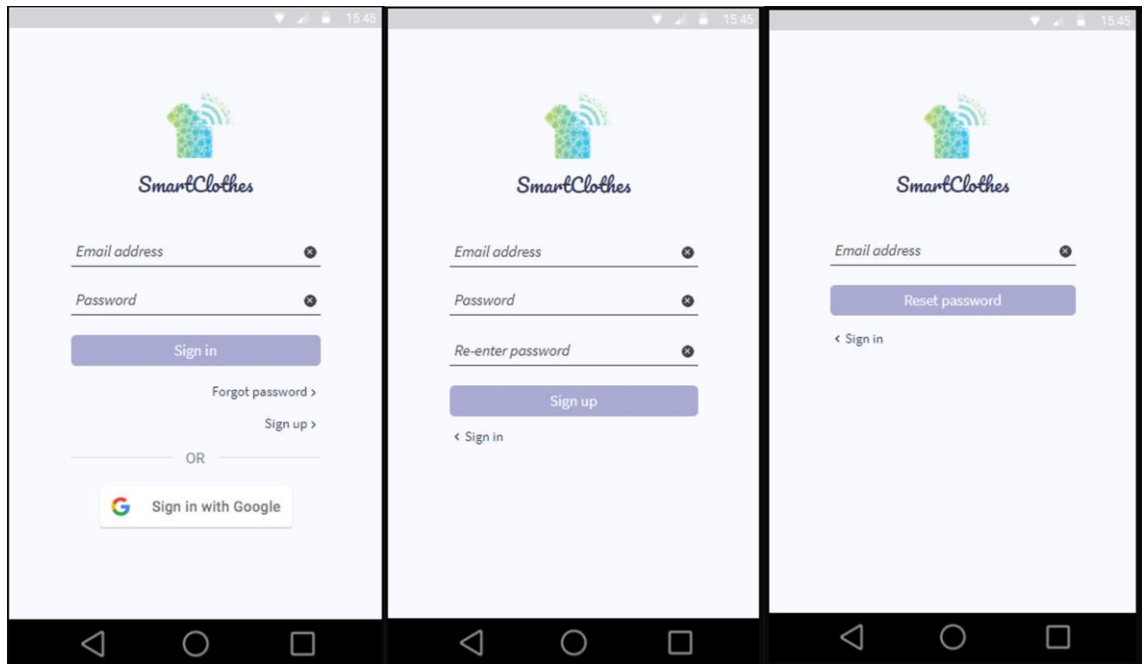


Ilustración 19: Prototipo de alta fidelidad; sesión de usuarios.

Pantallas de sesión: Esta pantalla gestiona la sesión de los usuarios, si el usuario no tiene cuenta se le enviará a la pantalla de registro y si ha olvidado su contraseña podrá recuperarla introduciendo su email en la pantalla de recuperación de contraseña. Estas pantallas no son necesariamente actividades diferentes, sino que conforman una misma actividad.

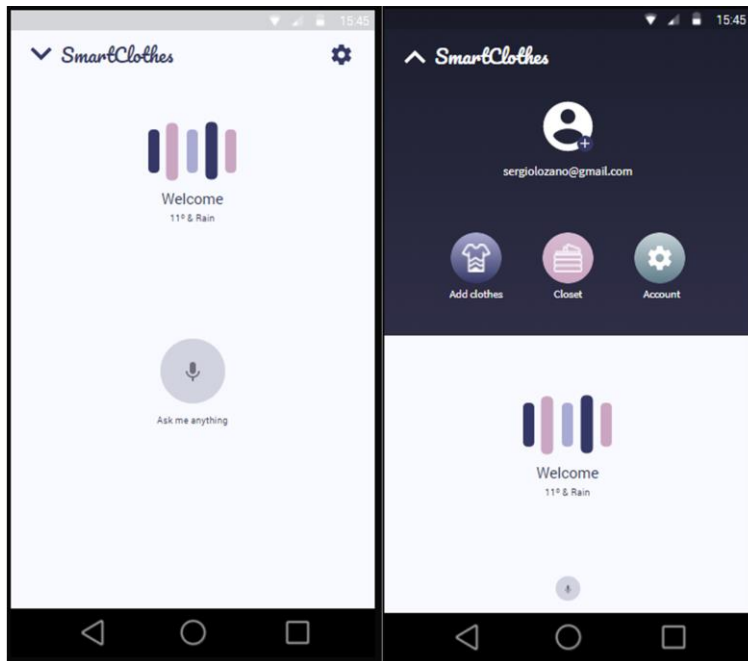


Ilustración 20: Prototipo de alta fidelidad; Main Activity

Pantalla Main Activity: Esta pantalla consiste en un menú principal para la aplicación, donde los usuarios pueden acceder al resto de secciones la aplicación. Esta pantalla está pensada para que el usuario emplee los comandos de voz que le asesorarán a la hora de vestirse en su día a día.

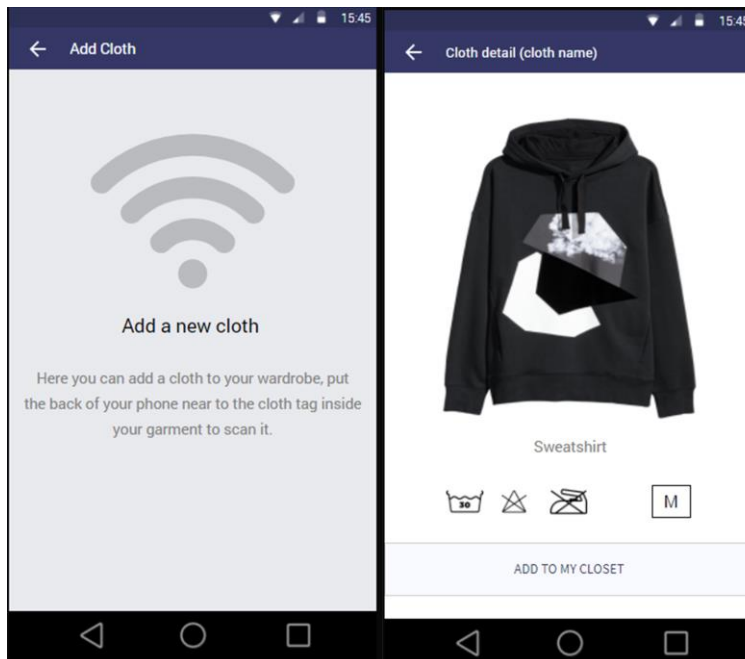


Ilustración 21: Prototipo de alta fidelidad; sección cloth detect

Pantallas Add Cloth y Cloth Detail: Estas pantallas consisten en un añadir las prendas del usuario vía NFC/RFID a su armario virtual. La primera pantalla se utiliza para detectar la etiqueta NFC/RFID mientras que la segunda, una vez reconocida la prenda, muestra los detalles de la misma.

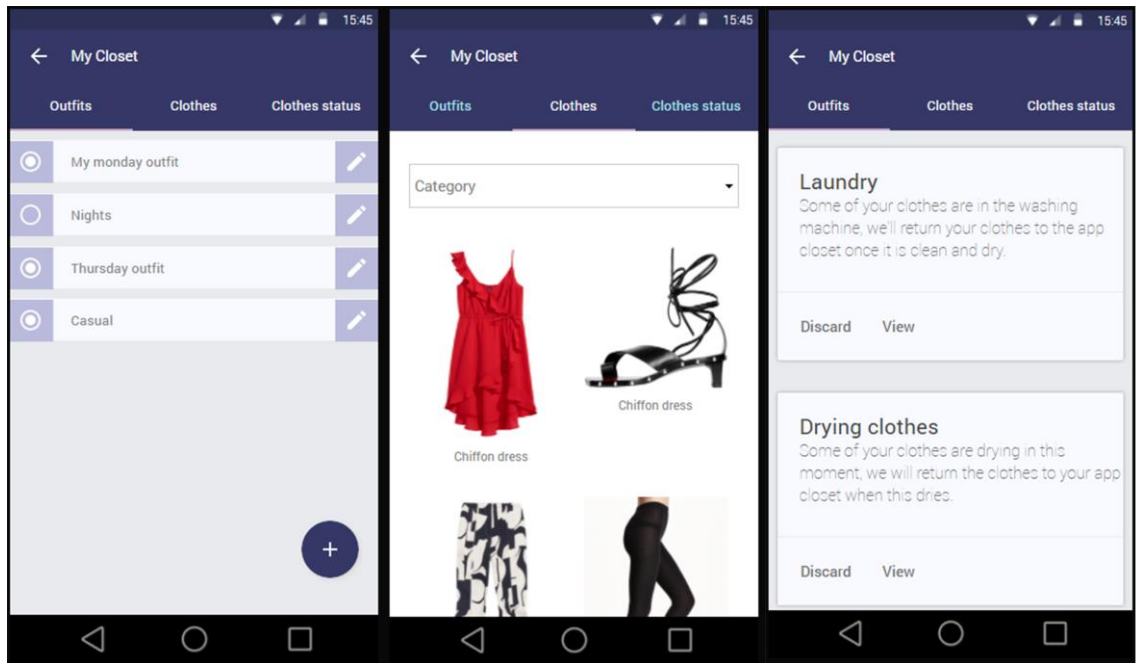


Ilustración 22:Prototipo de alta fidelidad; sección My closet

Pantallas My Closet: Estas pantallas están dispuestas con un Tabs Menu para facilitar la navegación. En esta sección el usuario administra sus outfits, prendas, y el estado de las prendas.

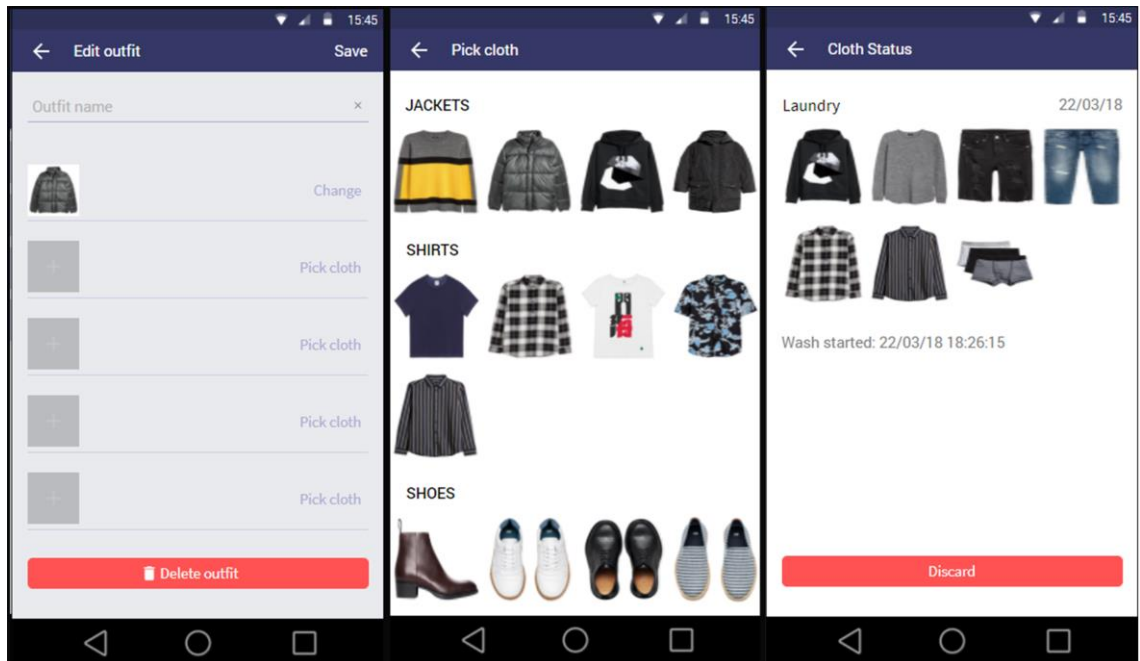


Ilustración 23:Prototipo alta fidelidad; outfit edit y cloth status detail

Pantallas Edit Outfit, Pick Cloth, Cloth Status Dteail: La primera pantalla sirve para crear, editar o eliminar outfits, seleccionando multiples prendas y asignándoles un nombre. La pantalla de Pick Cloth se activa cuando el usuario quiere seleccionar una prenda en la primera pantalla. Por último, la pantalla Cloth Status Detail que descende de la pantalla Clothes Status, aquí se puede visualizar si una prenda está fuera de servicio, lavándose o secándose, es decir el estado en el que se encuentra la prenda.

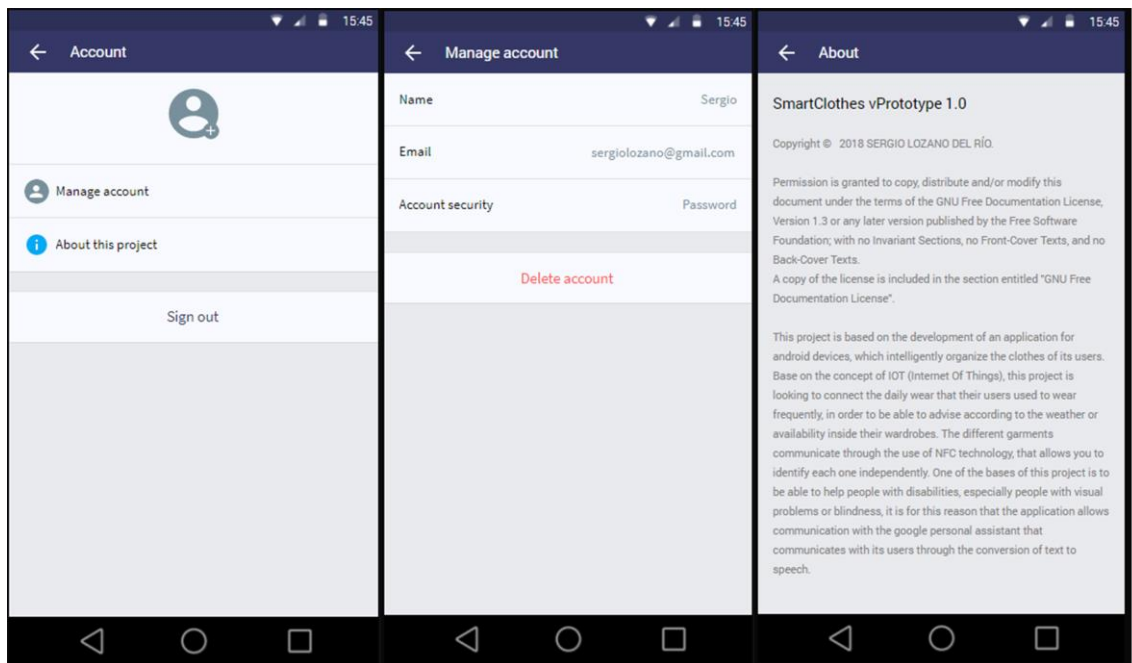


Ilustración 24: Prototipo de alta fidelidad; sección de Settings

Pantallas Settings: En este apartado el usuario puede modificar los parámetros vinculados a su cuenta de usuario como, por ejemplo, el nombre, email o su contraseña. Estas pantallas son un paso intermedio entre los campos del formulario para modificar los valores y el menú principal de la aplicación. De esta forma el usuario no se abruma al ver muchos campos para modificar valores.

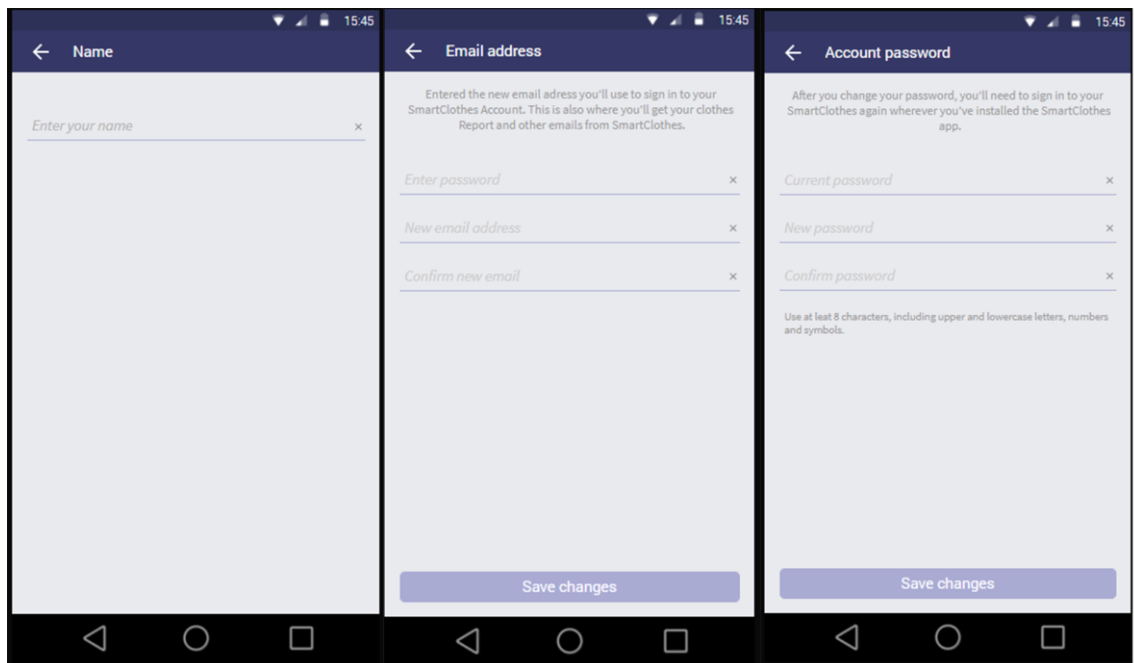


Ilustración 25:Prototipo de alta fidelidad; Settings: Name, Email, Password

Pantallas Settings (Forms): Estas pantallas cuentan con los campos para que los usuarios modifiquen los datos de su cuenta, la primera pantalla sirve para establecer el nombre del usuario, que por defecto se encontrará en blanco si el usuario no lo modifica. En la pantalla de Email adress se modifica el email vinculado a la cuenta del usuario, para poder modificar este parámetro el usuario debe introducir su contraseña, tras realizar el cambio se cierra la sesión del usuario. En último lugar se encuentra la pantalla Account Password donde el usuario modifica la contraseña de su cuenta, de igual manera que ocurría en la pantalla de Email Address, tras realizar los cambios se cierra la sesión del usuario y se envía a este a la pantalla de login de la aplicación.

2.5 Evaluación del prototipo

El prototipo se ha evaluado a medida que iba evolucionando, es decir conforme se iban obteniendo las pantallas, estas se mostraban a un público muy cerrado y concreto para evaluar la futura usabilidad de la aplicación y tener de forma segura que cada pantalla no iba a suponer ningún reto para ningún tipo de usuario. Se realizó un formulario¹⁹ en inglés y español, para evaluar las necesidades de múltiples usuarios y ver sus necesidades reales, y posteriormente, realizar los cambios en el prototipo.

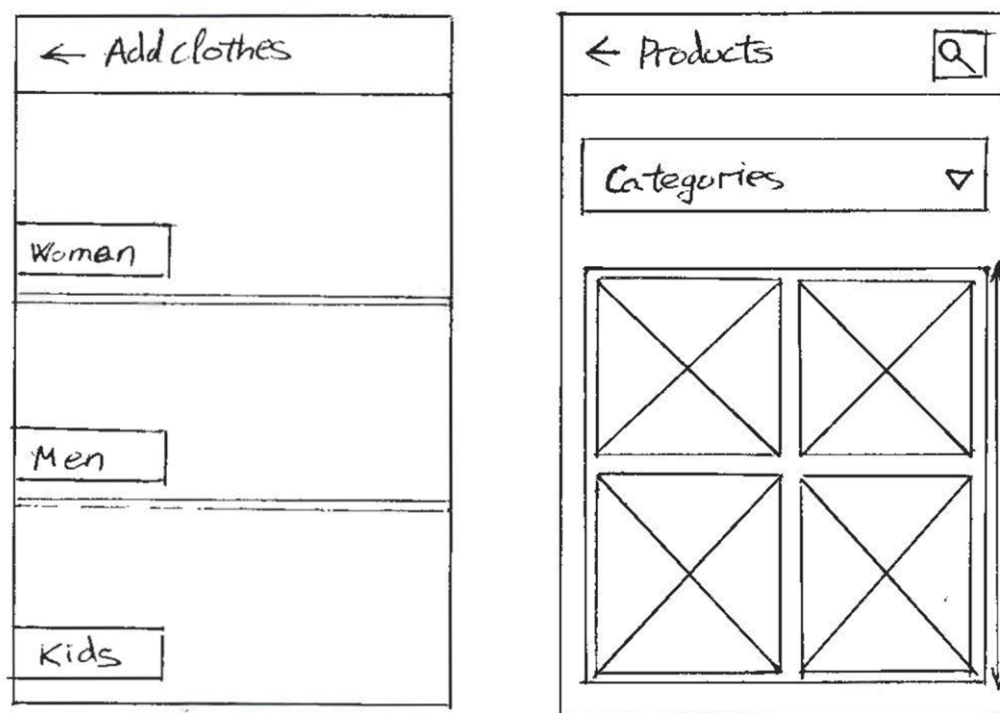


Ilustración 26: Pantallas de prototipo descartadas

Al inicio del proyecto, no se tuvo en cuenta el uso de la tecnología NFC a través del dispositivo android, sin embargo, tras investigar el funcionamiento de esta tecnología, se decidió implementar y se descartó

¹⁹ <https://goo.gl/forms/f8jXvXKzOPuf4v2t2>

la idea de un selector de prendas dentro de la aplicación a modo de tienda interna.

En las pantallas finales de la aplicación, existen algunos elementos que no se han podido implementar en el prototipo de alta fidelidad, por ejemplo, hay que tener en cuenta que se debe añadir una barra de carga para dispositivos más lentos o problemas con la conexión a internet, el cambio de colores a la hora de seleccionar un elemento o escribir sobre un campo, así como alertas al usuario, son elementos que no se han podido introducir en el prototipo, pero que hay que tener en cuenta para garantizar una buena experiencia de usuario.

Aunque, el prototipo de alta fidelidad proporciona una visión muy ajustada al estilo visual que tendrán todas las pantallas finales de la aplicación, estas no se encuentran exentas de cambios, ya que se trata de un proceso de creación iterativo, y en el momento que se detecte algún fallo en el diseño o sea necesario implementar cambios, las pantallas pueden sufrir ligeras modificaciones.

3. Desarrollo

El desarrollo se ha realizado utilizando varios editores y lenguajes de programación, ya que debido a las características del proyecto era necesario utilizar el lenguaje de programación que mejor se adecuaba a cada apartado del desarrollo del proyecto.

Para la aplicación móvil al tratarse de una aplicación nativa de android se ha desarrollado utilizando Android Studio, un entorno de trabajo gratuito y oficial para el desarrollo de aplicaciones Android. Al comienzo del desarrollo apareció la duda de que lenguaje de programación utilizar Java o Kotlin. Ambos lenguajes tenían sus ventajas e inconvenientes, por una parte, Java es un lenguaje muy extendido y consolidado, del cual además, ya se tenían conocimientos previos, sin embargo, tras leer varios artículos sobre el uso de Kotlin²⁰ y sus ventajas, el proyecto se decantó por este último, y es que, aunque al principio, el desarrollo iba a ser mucho más lento, Kotlin es un lenguaje muy simplificado y potente, que además ya cuenta con una versión muy sólida para el desarrollo de aplicaciones Android y perfectamente integrado con el entorno de trabajo, lo que se traduce a medio y largo plazo en un aumento de la productividad.

Para la programación de los microprocesadores se ha utilizado el IDE de programación de Arduino Uno, cuyo lenguaje principal está basado en C++, en la sección [microprocesador esp8266](#) se muestra con mayor detalle el proceso de desarrollo para el funcionamiento de los microprocesadores.

Por último, se ha utilizado JavaScript para el desarrollo de Cloud Functions en Firebase, las cuales, nos permiten extender el backend de la aplicación e identificar por sistema OAuth2 cada placa NFC del

²⁰ <https://devexperto.com/12-razones-usar-kotlin-android/>

microprocesador y asegurar correctamente la base datos mediante las reglas de Firebase en función de cada usuario.

3.1 Firebase

Firebase, como ya se ha comentado anteriormente, es una plataforma ideal para el desarrollo de este proyecto, ya que, debido a las limitaciones de tiempo y equipo, es posible ahorrar muchísimo tiempo en el desarrollo del backend si empleamos varios servicios de Firebase.

- **oAuth2:** Disponemos de autenticación de usuarios mediante el sistema oAuth2, que a su vez utiliza el sistema de tokens JWT²¹, sistema de uso muy extendido que utiliza el estándar RFC 7519 para la representación de llamadas seguras entre dos dispositivos o peticiones cliente-servidor. En el caso de la aplicación desarrollada los métodos de inicio de sesión utilizados son por usuario y contraseña, junto con el servicio de identificación de Google que ofrece un servicio de autenticación más cómoda y rápida a los usuarios.
- **Real Time Database:** Base de datos en tiempo real y en formato JSON que nos permite acceder a todos los datos de forma rápida y cómoda desde cualquier dispositivo o aplicación. Lo cual resulta idóneo para el entorno creado ya que tanto el microprocesador como la aplicación pueden modificar y acceder a todos los datos rápidamente.
- **Storage:** Cloud Storage, permite almacenar las imágenes de perfil de los usuarios y crear una base de datos de imágenes para la ropa y el etiquetado NFC, esto es posible combinando el servicio de Real Time Database con el de Cloud Storage.

²¹ JWT: JSON Web Tokens, sistema utilizado para autenticación de usuarios y realización de peticiones seguras a una base de datos.

- **Cloud Functions:** Funciones que actúan dentro del servidor y que permiten llamadas HTTPS con los métodos GET, POST, PUT, DELETE y OPTIONS a cada una de ellas. Sin el uso de este servicio la autenticación de usuarios en los microprocesadores habría resultado muy compleja ya no existen librerías oficiales de Firebase para este tipo de dispositivos. En adición DialogFlow tiene una buena sincronía con Firebase, mediante el uso de fulfillment. También es posible modificar todos los datos o detectar cambios en la base de datos y actuar en consecuencia a ello.

Como punto adicional Firebase permite manejar la recuperación de contraseñas, cambios de contraseña, cambios de correo electrónico, mediante el uso de plantillas, que se envían correos electrónicos a los usuarios. Además de ser un servicio muy escalable y con servicios adicionales como, por ejemplo, analíticas de las aplicaciones, servicios de mantenimiento y test, cloud messaging...

3.2 Gestión de los datos

Desde el comienzo del proyecto se tenía claro que los datos iban a ser gestionados por múltiples usuarios y que cada usuario iba a disponer de datos propios en función de múltiples variables como, por ejemplo, los diferentes estados de las prendas o la disposición de las mismas. En adición estos datos se gestionan desde múltiples dispositivos, lo que hace necesario que la estructura sea lo más clara y precisa posible.

Las prendas de ropa, se gestionan en base a un sistema de etiquetado NFC cuyo identificador es único, de esta forma nos aseguramos de que cuando un usuario acceda a una prenda, la aplicación se encargue de buscar la única pieza de ropa que coincida con su identificador, mostrándole las características de esta prenda. Actualmente este apartado de la base de datos es inmutable, es decir, los usuarios no

pueden modificar los datos o añadir nuevas prendas, si en fases futuras del proyecto se quiere permitir la modificación de los datos será necesario incluir un sistema que verifique que el identificador de la tarjeta NFC del usuario es único y si fuese necesario sobrescribir la tarjeta del usuario haciendo que su identificador sea único.

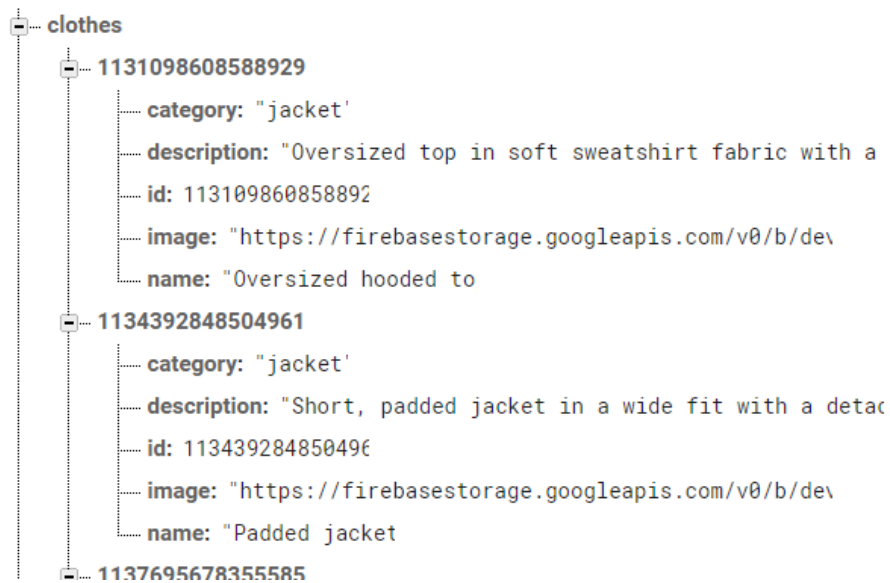


Ilustración 27: Esquema prendas de la base de datos.

Aunque Firebase se encargue de la autenticación de los usuarios, siempre es interesante tener registrados estos mismos usuarios en la base de datos, de esta forma tenemos un registro completo de cada usuario con todos los datos que nos interesan y si es necesario añadir nuevos datos que Firebase no pueda gestionar. Los usuarios disponen de un UID (identificador único de usuario), y estos se guardan en la base de datos siguiendo el siguiente esquema:

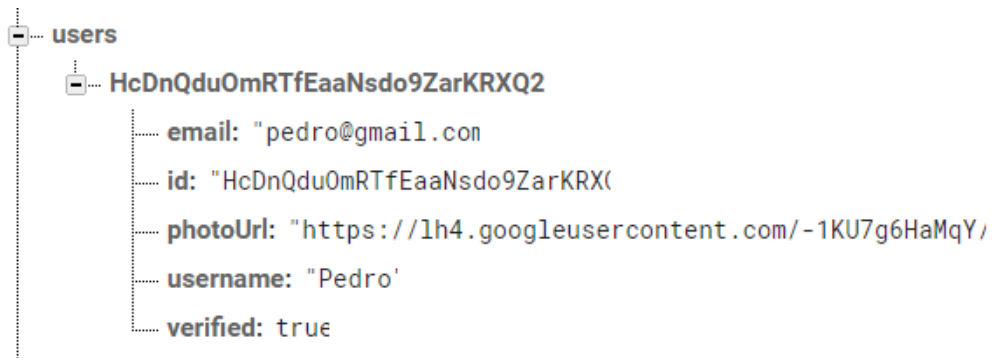


Ilustración 28: Esquema usuarios de la base de datos.

Una vez ya están gestionados tanto los usuarios como las prendas, se pueden crear armarios virtuales para cada usuario. Cuando un usuario registra una prenda como suya en la aplicación, esta queda registrada como prenda del usuario en su armario virtual, guardando el identificador de la prenda y su estado actual. Los armarios se gestionan en el apartado “wardrobe” de la base de datos, donde mediante el UID de usuario se crea un apartado único para cada usuario, donde se guardan las prendas y los “outfits” de cada uno de los usuarios. Este apartado sólo aparecerá en la base de datos si el usuario registra una prenda. Los dispositivos NFC de los usuarios modifican este apartado de la base de datos, concretamente, el estado de cada prenda del usuario, verificando primero el UID del usuario que emplea el dispositivo y accediendo únicamente a sus datos mediante la función desarrollada en el apartado cloud functions de Firebase.

3.3 Microprocesador esp8266

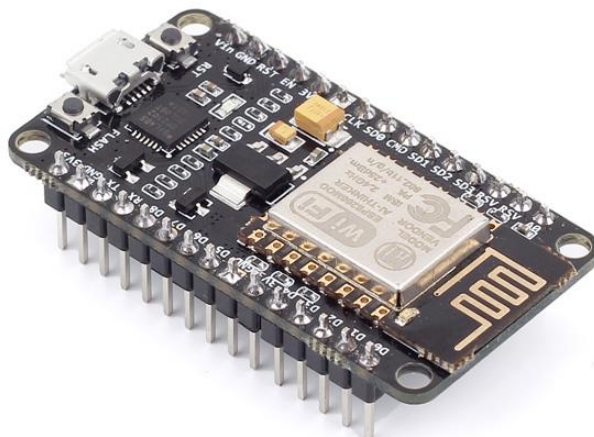


Ilustración 29: Microprocesador esp8266

El modelo de microprocesador esp8266 permite la conexión a redes WiFi, lo que hace que este proyecto enfocado en el IoT sea posible. Este microprocesador junto con una placa NFC permite crear registros de las prendas en la base de datos de Firebase. El cambio del estado de cada prenda que tiene el usuario en su armario es muy importante ya que es lo que define la aplicación y permite modificar los datos de cada prenda de una forma rápida a los usuarios. Este microprocesador nos permite crear una red de objetos físicos en la nube, y gracias a su pequeño tamaño y bajo coste de producción es posible utilizarlo en múltiples situaciones y entornos. En este proyecto el microprocesador se conecta a la red WiFi del domicilio y accede a la base de datos de Firebase cuando el usuario registra una prenda a través del lector NFC conectado a los pines del microprocesador. El estado de la prenda se modifica en la prenda registrada en el armario virtual del usuario, esto es posible gracias a una solicitud HTTPS en la que el microprocesador realiza una solicitud en la cual se indica el estado, la prenda a modificar y el UID del usuario que realizó la acción.

```

..
18 if (content) //change UID of the card that you want to give access
19 {
20
21     String tagId = content ;
22
23     Serial.println(tagId);
24     if (WiFi.status() == WL_CONNECTED) { //Check WiFi connection status
25
26         String url = "https://us-central1-dev-smartcloths.cloudfunctions.net/arduinoData?";
27         //String tagId = "7898479879";
28         String dataUrl = url + "uid=" + USER_UID + "&tagId=" + tagId + "&status=" + CLOTH_STATUS;
29
30         Serial.println(dataUrl);
31
32         char host[] = "us-central1-dev-smartcloths.cloudfunctions.net";
33
34         if (client.connect(host, 443) ) {
35             Serial.println("connected");
36
37             String URL = dataUrl;
38
39             Serial.println(URL);
40
41             client.println("GET " + URL + " HTTP/1.1");
42             client.print("Host: ");
43             client.println(host);
44             client.println("User-Agent: arduino/1.0");
45             client.println("");
46
47         } else {
48
49             Serial.println("Error in WiFi connection");
50
51         }

```

Ilustración 30: Código petición REST desde microprocesador.

Gracias a estos datos una función implementada en Firebase Cloud Functions se encarga de gestionar estos datos y modificarlos en la sección correcta de la base de datos mediante el método `admin.database` de Firebase.

```

const functions = require('firebase-functions');
const admin = require('firebase-admin');
admin.initializeApp(functions.config().firebase);

exports.arduinoData = functions.https.onRequest((req, res) => {
  const uid = req.query.uid;
  var tagId = req.query.tagId;
  const status = req.query.status;

  tagId = hexToDec(tagId);

  admin.database().ref('/wardrobe').child(uid).child('clothes').child(tagId).set({
    id: tagId,
    status: status
  }).then(snapshot =>{
    res.send("Data added");
  })
});

function hexToDec(s) {

  function add(x, y) {
    var c = 0, r = [];
    var x = x.split('').map(Number);
    var y = y.split('').map(Number);
    while(x.length || y.length) {
      var s = (x.pop() || 0) + (y.pop() || 0) + c;
      r.unshift(s < 10 ? s : s - 10);
      c = s < 10 ? 0 : 1;
    }
    if(c) r.unshift(c);
    return r.join('');
  }

  var dec = '0';
  s.split('').forEach(function(chr) {
    var n = parseInt(chr, 16);
    for(var t = 8; t; t >>= 1) {
      dec = add(dec, dec);
      if(n & t) dec = add(dec, '1');
    }
  });
  return dec;
}

```

Ilustración 31: Cloud Function microprocesador

Por motivos de tiempo y estructura del proyecto el acceso WiFi y el UID de usuario, se ha implementado directamente en el código del microprocesador, sin embargo, para una versión de producto final es necesario implementar un sistema mediante el cual el microprocesador del usuario pueda identificar su clave de acceso WiFi y el UID de la cuenta de usuario de la aplicación. Una forma de hacerlo podría consistir

en el paso de información mediante el uso de NFC móvil o conectándose por WiFi al microprocesador, haciendo que este haga de servidor y el dispositivo móvil del usuario de cliente.

3.4 Desarrollo Android

El desarrollo en android se ha realizado teniendo en cuenta cada pantalla y funcionalidad establecidas en la fase de diseño de la aplicación. Se ha tratado de ser lo más fiel posible al diseño establecido realizando pequeños cambios o implementaciones si por motivos de usabilidad se consideraba necesario. También se han utilizado indicadores de carga de datos, para notificar al usuario cuando estamos accediendo a la base de datos o haciendo una carga que pueda demorarse. También se ha implementado un sistema de alertas al usuario para confirmar ciertas acciones o notificar al usuario problemas cuando una acción no se ha podido realizar.

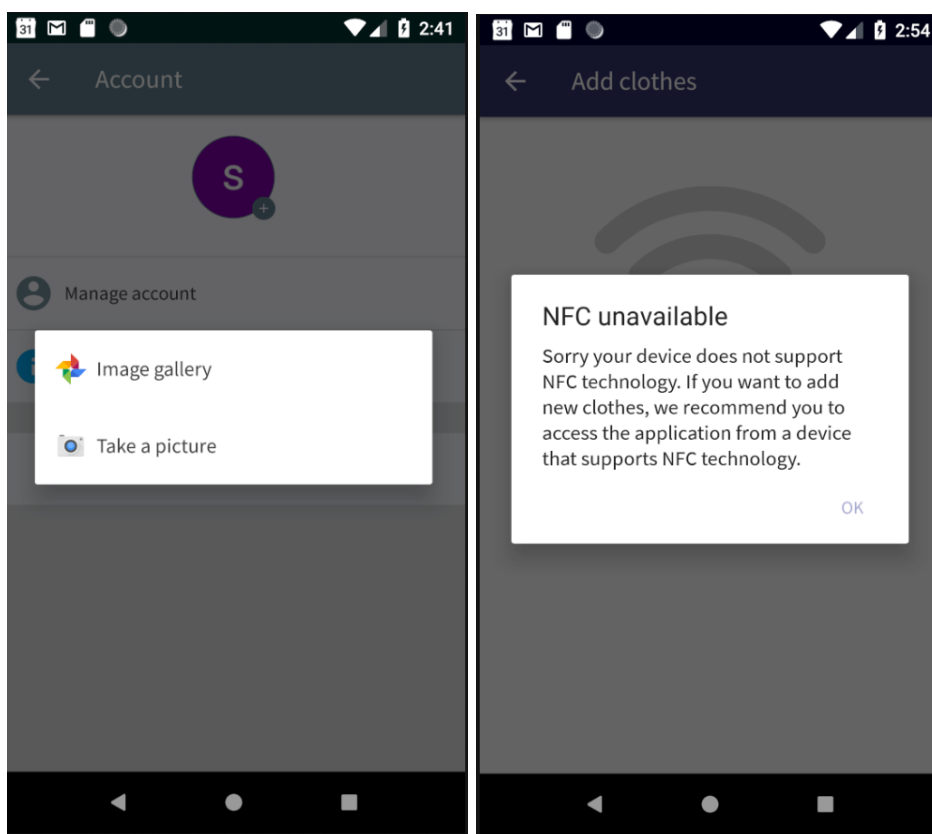


Ilustración 32: Notificación "alert"

Se ha tratado todo el código del proyecto de la forma más escalable posible, ya que, de esta forma si se quiere, por ejemplo, realizar un cambio en la estructura de la base de datos se puede realizar

cambiando las clases de objeto ubicados en la carpeta DataModel del proyecto. Todos los valores string de la aplicación se encuentran documentados en el archivo strings.xml de la aplicación, de esta forma si en un futuro se quiere traducir la aplicación a varios idiomas será posible hacerlo rápidamente.

Las imágenes e iconos del proyecto, en la medida de lo posible, han sido convertidos a elementos vectorizados xml que evitan cargas pesadas, las imágenes más grandes se han almacenado en el cloud storage de Firebase para no tener que realizar una actualización de la aplicación cada vez que se quiera añadir nuevas imágenes y evitar sobretodo que el empaquetado de la aplicación sea excesivamente pesado.

3.4.1 Inicio de sesión

Cuando el usuario instala la aplicación por primera vez, ha de identificarse en la pantalla SesionActivity, pantalla compuesta por varios fragments que incluyen el login y registro de usuarios, así como también recuperación de contraseña. El registro de los usuarios puede hacerse mediante registro estándar (usuario y contraseña) o inicio de sesión google, utilizando este como proveedor de autenticación. Si el usuario se ha registrado y ha iniciado sesión se le redirige directamente al MainActivity, de esta forma evitamos que el usuario tenga que identificarse cada vez que se abre la aplicación.

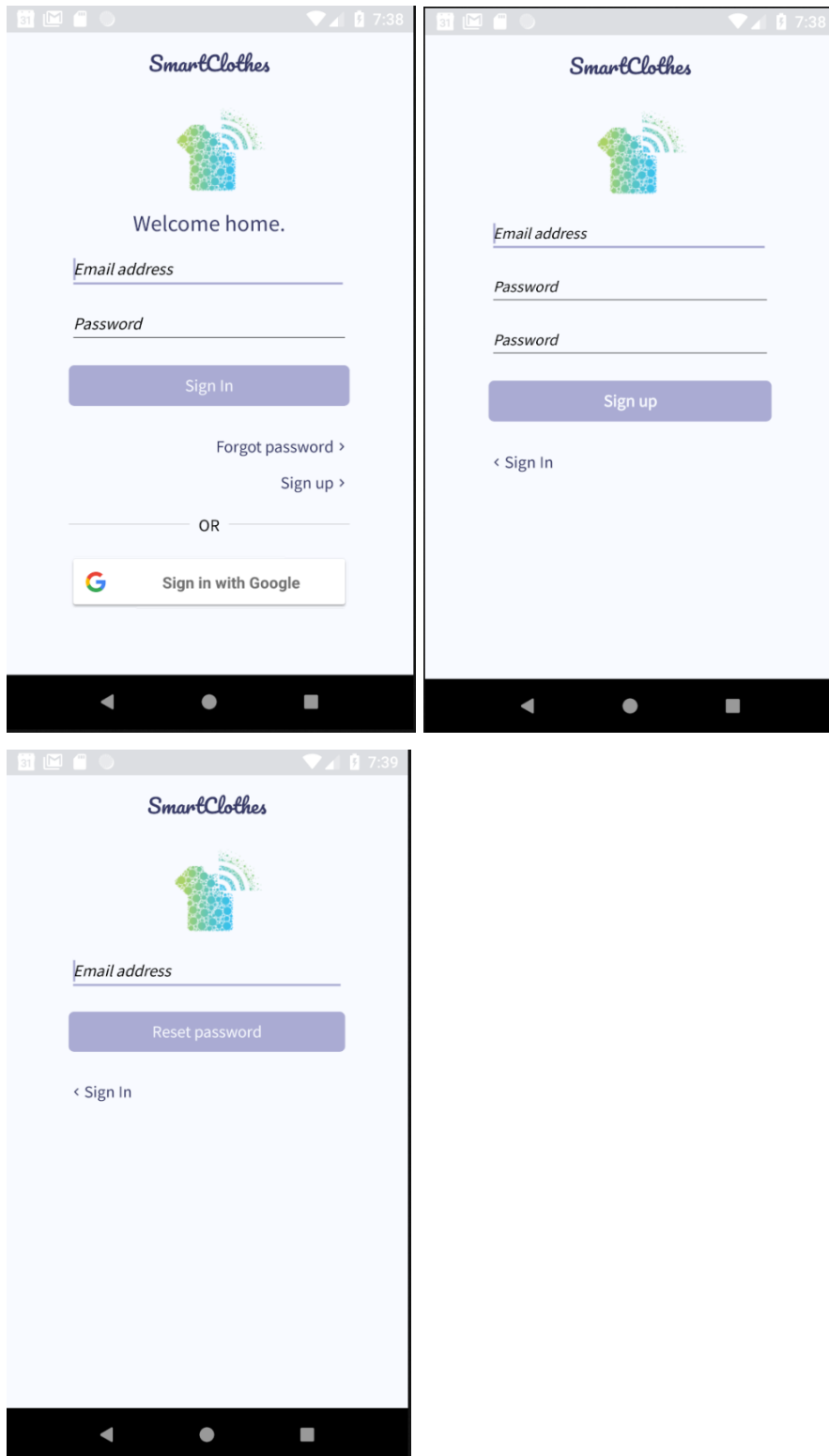


Ilustración 33: Pantallas "SesionActivity"

3.4.2 MainActivity

El "main activity" de la aplicación es el centro de la aplicación aquí se encuentra el menú principal y una de las funcionalidades más importantes, que es el hecho de poder comunicarte con el API de DialogFlow. Este API permite recibir un string/frase del usuario y reconoce ciertas palabras clave (previamente establecidas) y ofrece una respuesta acorde. A nivel de usabilidad e interacción interfaz – usuario aporta incontables posibilidades ya que el usuario puede realizar múltiples preguntas y gracias al uso de Fullfilment y Cloud Functions podemos devolver información personal del usuario.

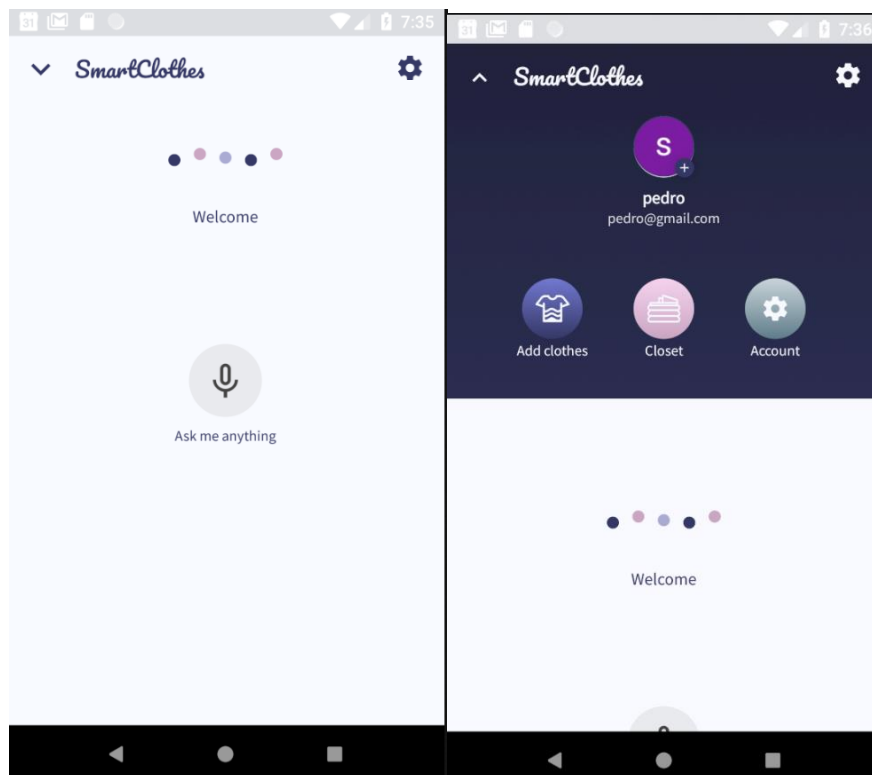


Ilustración 34: Pantallas "MainActivity"

3.4.3 AddClothes

En esta pantalla los usuarios pueden agregar prendas a su armario virtual. Estas prendas deben de disponer de una pegatina NFC cuya tecnología sea MIFARE Ultralight Type C, descartando otras tecnologías NFC, debido a su tamaño o poca viabilidad con el proyecto. Este tipo de

chips NFC son muy pequeños y ligeros y se pueden colocar fácilmente en la ropa en forma de pegatina o etiqueta por un coste muy bajo. Cuando el usuario acerca una prenda NFC al dispositivo la aplicación detecta la etiqueta mediante el método `onTagDiscover` de android, este método devuelve un valor hexadecimal que se convierte en un valor decimal.

```
class NfcUtils {
    companion object {
        private val HEX_CHARS_ARRAY = "0123456789ABCDEF".toCharArray()
        fun toHex(byteArray: ByteArray) : String {
            val result = StringBuffer()

            byteArray.forEach { it: Byte
                val octet = it.toInt()
                val firstIndex = (octet and 0xF0).ushr( bitCount: 4)
                val secondIndex = octet and 0x0F
                result.append(HEX_CHARS_ARRAY[firstIndex])
                result.append(HEX_CHARS_ARRAY[secondIndex])
            }

            return result.toString()
        }

        fun toDec(s:String):Long {
            val bigInt = BigInteger(s, radix: 16)
            return bigInt.toLong()
        }
    }
}
```

Ilustración 35: Cambio de valor hexadecimal a decimal en Kotlin

Una vez convertido el valor hexadecimal recibido, se compara con la base de datos en Firebase y recuperamos los datos de la prenda que el usuario quiere añadir a su armario virtual. Cuando se contrasta el id de la etiqueta NFC con el mismo id de la prenda en la base de datos y se reciben todos los datos que se quieren mostrar se permite al usuario guardar y ver la prenda en detalle.

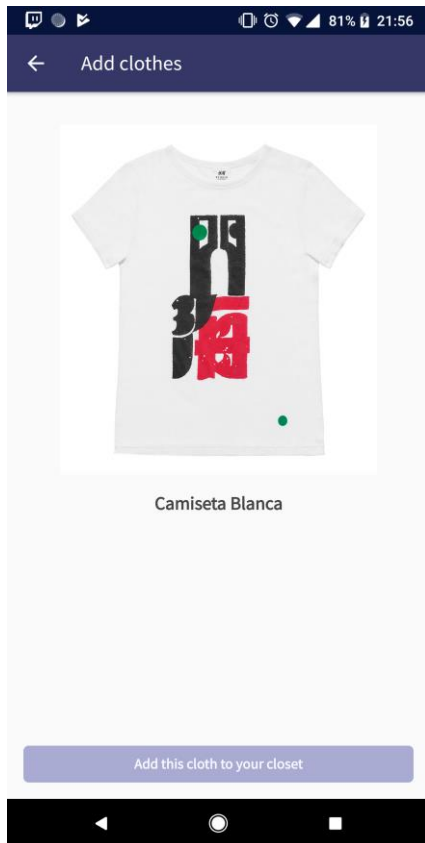


Ilustración 36: Pantalla “add clothes” tras analizar una prenda

3.4.4 AccountActivities

Esta sección de la aplicación se compone por varias actividades que permiten a los usuarios gestionar su cuenta y modificar sus datos de autenticación de Firebase. En este apartado se pueden modificar todos los datos del usuario, como por ejemplo, el nombre, el email o la contraseña. También es posible cambiar la imagen de perfil del usuario, utilizando la galería o la cámara del dispositivo. Si el usuario quiere cambiar de cuenta, cerrar su sesión o eliminar la cuenta, puede hacerlo en este apartado de la aplicación.

3.4.5 ClosetActivity

Otra de las secciones más importantes de la aplicación, compuesta de 3 fragments y varias actividades subsecuentes. La sección closet de la

aplicación muestra a los usuarios datos clave de su armario virtual. Tras registrar las prendas mediante el sistema de registro NFC, [descrito en la sección anterior](#), el usuario puede encontrar todas las prendas de su armario y crear diferentes outfits. En esta sección predomina el uso de recyclerviews, los cuales permiten mostrar listas de datos de forma dinamica. Para algunos de los recyclers views implementados, se ha utilizado la librería de Firebase UI, que contiene metodos y funciones prestablecidos que facilitan las llamadas a la base de datos y rellena el recycler view dentro del propio método de forma asincrona.

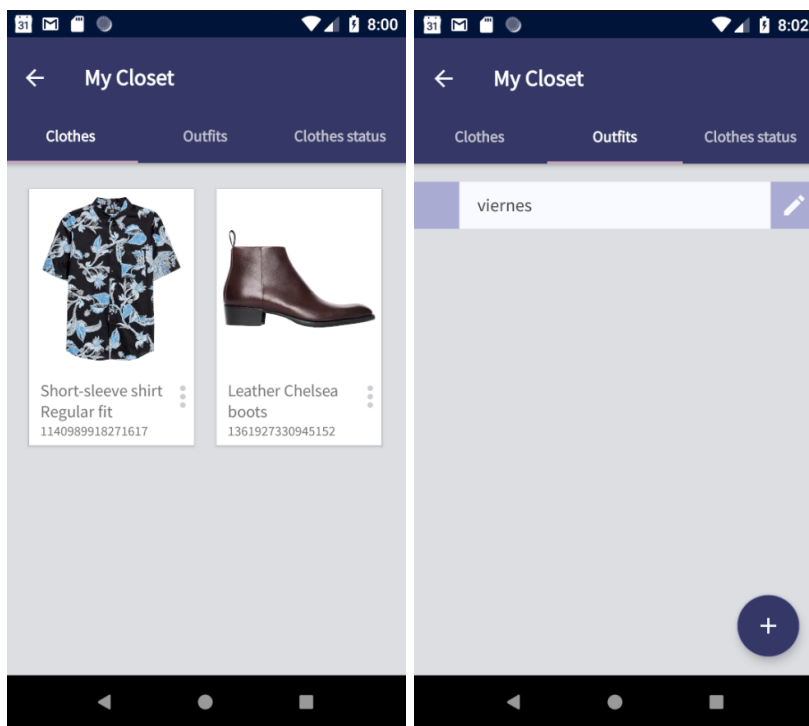


Ilustración 37: Pantallas "ClosetActivity"

3.5 Librerías utilizadas

- **Librerías de Firebase:**

```
//firebase
implementation 'com.google.firebase:firebase-core:12.0.1'
implementation 'com.google.firebase:firebase-firestore:12.0.1'
implementation 'com.google.firebase:firebase-auth:12.0.1'
implementation 'com.google.firebase:firebase-database:12.0.1'
implementation 'com.firebaseui:firebase-ui-auth:3.3.0'
implementation 'com.firebaseui:firebase-ui-database:3.3.0'
```

Sdk oficial de Firebase para android que permite implementar llamadas a todos los servicios de firebase.

- **Librerías de interfaz:**

```
//UI dependencies
implementation 'com.github.zagum:SpeechRecognitionView:1.2.2'
implementation 'com.squareup.picasso:picasso:2.5.2'
implementation 'com.mikhaellopez:circularimageview:3.0.2'
implementation 'com.github.bumptech.glide:glide:4.7.1'
annotationProcessor 'com.github.bumptech.glide:compiler:4.7.1'
```

Múltiples librerías que permiten modificar el aspecto de la aplicación de una forma rápida y eficiente.

Se ha utilizado SpeechRecognitionView para crear la interfaz de reconocimiento de voz que en combinación con DialogFlow facilitan al usuario información sobre su armario.

Squareup picasso y glide se utilizan para cargar imágenes que tenemos en forma de link URI desde una base de datos a la aplicación, ahorrando tener que implementar un servicio Http y decodificar la imagen.

CircularImageView permite crear imágenes circulares rápidamente, además de poder modificar diversos parámetros del borde de la imagen. Se ha utilizado esta librería para obtener un diseño similar al planteado en la fase de diseño.

- **Librerías de DialogFlow:**

Las librerías de DialogFlow permiten la conexión directa mediante el uso de método y funciones propias, la conexión con su API.

Aunque no se trate de una librería en específico, destacar que el control de versiones se ha realizado utilizando el GIT de android studio y guardando las versiones en [Gitlab](#).

3.7 Problemas durante el desarrollo

El mayor problema que se presentó en la fase de desarrollo, fue el hecho de autenticar a los usuarios en la base de datos desde el microprocesador, ya que no se dispone de una librería firebase oficial para microprocesadores. El IoT es un concepto único y emergente en la actualidad, por lo que todavía no existe mucha información acerca de como crear un entorno óptimo para conectar microprocesadores en la nube. Tras realizar múltiples búsquedas en internet y leer múltiples documentos sobre google cloud e IoT, se descubrió que la forma más optima de conectar dispositivos era mediante el uso de una Raspberry Pi, sin embargo, esto aumentaría el coste de producción del proyecto en exceso, ya que cada Raspberry Pi tiene un precio aproximado de 70 euros frente a los 3 o 4 euros que cuesta un microprocesador esp8266. Puesto que iba a ser complicado desarrollar un código eficiente para autenticar a usuarios dentro de la memoria del microprocesador se decidió delegar el sistema de autenticación de usuarios al backend de firebase y que el arduino unicamente realizase una solicitud HTTPS.

Durante la fase de testeo se detectaron multiples “crashes”²² que hubo que solventar a medida que desarrollaba la aplicación. Al tratarse un proyecto con múltiples pantallas es difícil encontrar todos los fallos de programación, sin embargo son errores menores que se pueden solventar facilmente en las fases de iteración del proyecto.

Debido a estos problemas hubo que re-estructurar la planificación eliminando las acciones y consultas al API metereológico, ya que a pesar de ser una función verdaderamente atractiva para la aplicación, se catalogo como prescindible, debido a que únicamente ofrece valores añadidos sobre el hecho de gestionar el outfit del usuario, sin embargo

²² Crash: Cuando se produce cierre repentino o sin consentimiento del usuario debido a un fallo en la programación.

en futuras versiones la aplicación es una función a priorizar en el desarrollo.

4 Pruebas

El testeo de la aplicación se ha llevado a cabo con una prueba de robo en Firebase Test Lab la cual ha dado valores positivos en el rendimiento. Este test demuestra que la aplicación no consume excesivos recursos de memoria interna del dispositivo ni de transmisión de datos en red.

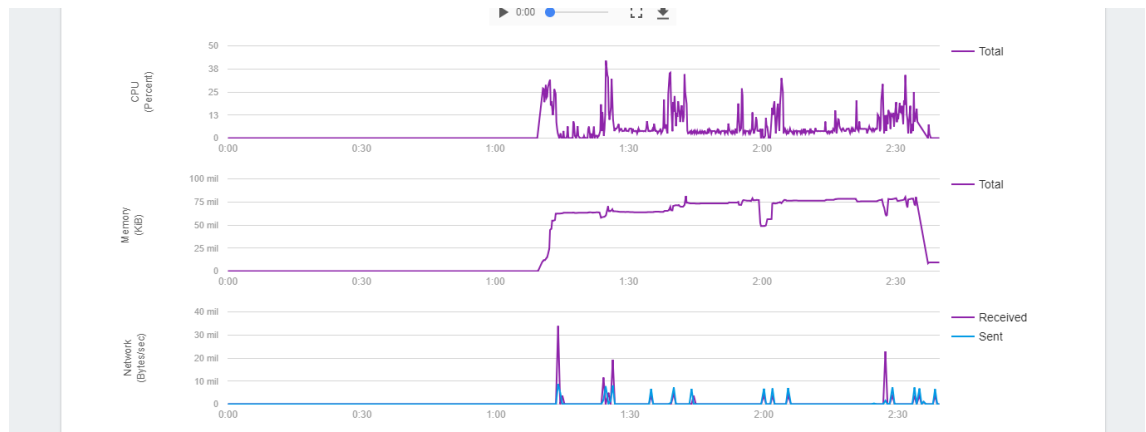


Ilustración 38: Grafica de consumo de recursos de la aplicación.

La primera prueba de robo realizada notifico un error en la pantalla de OutfitForm, ya que el usuario podía guardar un outfit sin ninguna prenda ni nombre, lo que provocaba un error a la hora de recuperar los datos de la clase en el recycler view. Este error se solvento de manera rápida creando una alerta que impedía a los usuarios registrar un outfit vacío.

Para la siguiente prueba realizada se han utilizado dos dispositivos un Pixel con API level 27 y un Galaxy J2 con API level 23, además se ha creado un archivo “Record Robo Script” con un inicio de sesión pre configurado para que se puedan emplear datos de tarjetas NFC, de esta forma es posible testear la aplicación con un armario lleno de diferentes prendas.

✓ Prueba Robo, hace 21 minutos ⓘ

Fallida	Correctos	Omitidos	No concluyentes
0	2	0	0

[VER CLÚSTERES DE CAPTURAS DE PANTALLA →](#)

Ejecución de prueba	Duración	Configuración regional	Orientación	Incidencias
✓ Pixel, nivel de API 27	5 min 46 s	inglés (Estados Unidos)	Retrato	–
✓ Galaxy J2 Prime SM-G532M, nivel de API 23	5 min 48 s	inglés (Estados Unidos)	Retrato	–

Ilustración 39: Pruebas exitosas

Las pruebas en ambos dispositivos se efectuaron sin ninguna incidencia y reportaron valores de rendimiento relativamente bajos.

5. Conclusiones y líneas futuras

El desarrollo de una aplicación móvil no solo consiste en la creación y desarrollo de las diferentes pantallas de la aplicación, sino que requiere de un constante crecimiento, mantenimiento y testeado para comprobar que todo funcione perfectamente una vez la aplicación se encuentra abierta a los usuarios.

Hay que tener siempre muy en cuenta los tiempos de desarrollo, prevenir percances tanto de programación como ajenos a esta, ya que el retraso en cualquiera de las fases puede desencadenar un retraso en el desarrollo de todo el proyecto.

Tras haber concluido el desarrollo y probado la aplicación, se ha llegado a la conclusión de que el uso de un sistema NFC para el registro y gestión de prendas es rápido, económico y eficaz. Con tan solo acercar una prenda al dispositivo móvil, es posible gestionar todo el armario de un usuario de una forma mucho más cómoda que empleando por ejemplo descriptores de imagen que identifiquen cada prenda, además de ofrecer un rendimiento mucho más optimizado.

Ahora que las bases del proyecto han sido asentadas, es necesario ir mejorando poco a poco la aplicación desde la interacción con DialogFlow, la cual no se ha podido implementar por completo. También es necesario inculcar en los usuarios el uso de prendas NFC por lo que un tutorial, sería indispensable de cara a buscar nuevos usuarios.

De carácter más personal me gustaría finalizar por completo el proyecto, comprobando si hay interés en la sociedad por el uso de un armario y prendas "inteligentes", continuar desarrollando el ecosistema planteado y experimentar nuevas formas de gestionar los datos que pueden ofrecer las prendas y como los usuarios las utilizan. También añadir que he aprendido mucho sobre el desarrollo de aplicaciones Android con Kotlin

y su vinculación a los servicios de Firebase, sin los cuales este proyecto habría tenido un desarrollo muchísimo más lento.

Si bien es cierto que el IoT es algo que está muy cerca de ser realidad, en mi opinión aún quedan algunos años hasta que se normalice tiempo suficiente para perfeccionar el ecosistema planteado y buscar la forma de satisfacer las necesidades de los futuros usuarios.

6. Glosario

IOT (Internet Of Things): Se trata de una nueva tecnología de acceso a Internet. Mediante el Internet de las cosas, los objetos se reconocen entre sí, obteniendo un comportamiento de inteligencia conjunta, haciendo o habilitando decisiones relacionadas.

Dialogflow: Plataforma de desarrollo orientado a la interacción humano-computadora, cuya tecnología permite generar conversaciones de forma natural.

NFC: Del término inglés Near Field Communication, es una tecnología de comunicación inalámbrica de corto alcance basado radio frecuencia.

RFID: Es un sistema de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas, tarjetas o transpondedores RFID. El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto

Webhook: Es un método de alteración del funcionamiento de una página o aplicación web, que utilizando un callback personalizado puede modificar el contenido de la página.

Startup: El término inglés “startup” hace referencia a una compañía emergente que busca emprender o montar un nuevo negocio. Normalmente este tipo de compañías busca desarrollare un modelo de negocio efectivo validando primero su producto y creando un modelo de negocio escalable.

Scrum: Desarrollo ágil que se caracteriza por adoptar una estrategia de desarrollo incremental, buscar la calidad del producto basándose en el conocimiento táctico de las personas en equipos auto organizados y el

solapamiento de diferentes fases del desarrollo en lugar de realizar una tras otra en un ciclo secuencial.

Influencers: La mercadotecnia influyente o marketing influyente es una forma de publicidad que ha surgido a partir de una variedad de prácticas y estudios recientes, enfocada más a los individuos que al mercado objetivo en su conjunto. Identifica a las personas que tienen influencia sobre los compradores potenciales y las actividades de mercadotecnia orientadas en torno a estas personas influyentes.

B2B: Negocio a negocio (del inglés business-to-business o B2B) hace referencia a las transacciones comerciales entre empresas, es decir, a aquellas que típicamente se establecen entre un fabricante y el distribuidor de un producto.

Intents: Un intent representa el mapeado a través de las palabras que dicta el usuario y la acción que debe ser tomada por el software.

7. Bibliografía

1. Eunice Kennedy Shiver National Institute of Child Health and Human Development, “*Tecnología de asistencia y rehabilitación*”, <https://www1.nichd.nih.gov/espanol/salud/temas/rehabtech/Pages/default.aspx>. Consultado Febrero 2018.
2. M. Condoluci, G. Araniti, T. Mahmoodi and M. Dohler, “*Enabling the IoT Machine Age With 5G: Machine-Type Multicast Services for Innovative Real-Time Applications*”, [IEEE Access](#), vol. 4, pp. 5555-5569, 2016. doi: 10.1109/ACCESS.2016.2573678. Consultado Febrero 2018.
3. Longitud, Fuente original: eldigital.es, “*El día a día de un invidente*”, <https://longitudeonda.com/index.php/el-dia-a-dia-de-un-invidente/>. Consultado Febrero 2018.
4. Michael Sawh, “*The best smart clothing*”, <https://www.wearable.com/smart-clothing/best-smart-clothing>, September 26, 2017. Consultado Febrero 2018.
5. “*Electronic Cloths E-commerce*”, <https://www.vulpes-electronics.com/>. Consultado Febrero 2018.
6. “*Dialogflow definition*”, <https://en.wikipedia.org/wiki/Dialogflow>. Consultado Marzo 2018.
7. “*NFC Forum Technical Specifications*”, <https://nfc-forum.org/our-work/specifications-and-application-documents/specifications/nfc-forum-technical-specifications/>. Consultado Marzo 2018.

8. Jeff Lindsay, “*Web hooks to revolutionize the web*”
<http://progrium.com/blog/2007/05/03/web-hooks-to-revolutionize-the-web/>. Consultado Marzo 2018.
9. “Jacquard by Google”, <https://atap.google.com/jacquard/levi/>. Consultado Marzo 2018.
10. Tommy Edison “*How Blind People Choose Their Clothes*”,
<https://www.youtube.com/watch?v=upxt3qbPII0>. Consultado Marzo 2018.
11. <https://about.hm.com/en/terms-and-conditions.html> Marzo 2018
12. <https://www.flaticon.com/packs/laundry-guide> Marzo 2018
13. OMS, “*Ceguera y discapacidad visual*”, Nota descriptiva, Octubre de 2017, <http://www.who.int/mediacentre/factsheets/fs282/es/>. Consultado Marzo 2018.

8. Anexos

Anexo1: Diagrama Gantt de la planificación del trabajo. **54**

Anexo 2: Planificación por horas llevada a cabo en Excel..... **55**

Anexo 1: Diagrama Gantt de la planificación del trabajo.

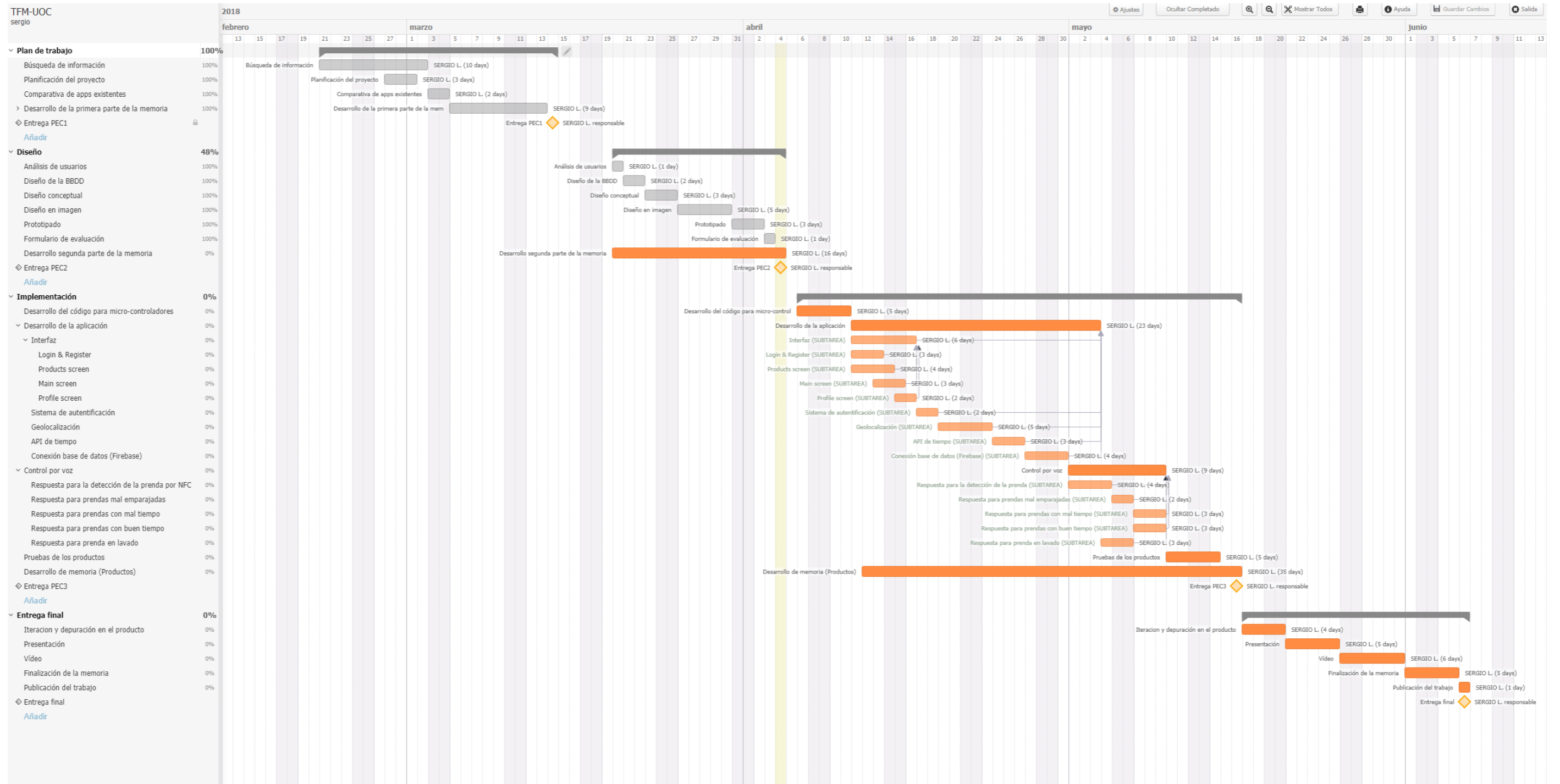


Ilustración 40: Planificación en gantt

Anexo 2: Planificación por horas llevada a cabo en Excel.

	A	B	C	D	E	F
1	Tareas	Horas	N° Horas restantes		N° Horas totales	N° Horas restantes
2	Busqueda de información	6	294		300	0
3	Comparativa de apps	4	290			
4	Planificación del proyecto	8	282			
5	Desarrollo de la primera parte de la memoria*	12	270			
6	Análisis de usuarios	4	266			
7	Diseño de la BBDD	6	260			
8	Diseño conceptual	12	248			
9	Diseño en imagen	20	228			
10	Prototipado	16	212			
11	Formulario de evaluación	2	210			
12	Desarrollo segunda parte de la memoria	12	198			
13	Desarrollo del código para micro-controladores	12	186			
14	Desarrollo de la aplicación*	72	114			
15	Control por voz*	40	74			
16	Pruebas de productos	4	70			
17	Desarrollo de la memoria (Productos)	16	54			
18	Iteración y depuración en el producto	8	46			
19	Presentación	12	34			
20	Vídeo	16	18			
21	Finalización de la memoria	14	4			
22	Publicación del trabajo	4	0			
23						
24						
25						
26						
27	Subtareas	Horas	N° Horas restantes			
28	Desarrollo primera parte de la memoria	12	12			
29	Contexto y justificación del trabajo	5	7			
30	Objetivos del trabajo	2	5			
31	Enfoque y metodo elegido	3	2			
32	Descripción de la planificación	2	0			
33	Desarrollo de la aplicación	72	72			
34	Interfaz	30	42			
35	Login & Register	4	26			
36	Products screen	8	18			
37	Main screen	12	6			
38	Profile screen	6	0			
39	Sistema de autenticación	6	36			
40	Geolocalización	16	20			
41	API Tiempo	14	6			
42	Conexión a la base de datos (Firebase)	6	0			
43	Control por voz	40	40			
44	Respuesta para la detección de una prenda por NFC	10	30			
45	Respuesta para prendas mal emparejadas	4	26			
46	Respuesta para prendas con mal tiempo	8	18			
47	Respuesta para prendas con buen tiempo	8	10			
48	Respuesta para prendas en lavado	6	4			

Ilustración 41: Planificación de horas en excel

