



Escuela Técnica Superior de Ingeniería Informática
Grado de Ingeniería Informática – Ingeniería del Software

Curso 2019/2020

CLARIFY

Departamento: Lenguajes y Sistemas Informáticos

Breve descripción

Creación de una aplicación Android que facilita a las personas con discapacidad visual la creación de listas de la compra, identificación de productos y gestión de inventario

Autor

Álvaro de la Flor Bonilla

Tutor

José Antonio Parejo Maestre

RESUMEN

La accesibilidad es un derecho, de suma importancia, toda la sociedad debería velar por su cumplimiento y mejora. Sin embargo, existen situaciones cotidianas donde su desarrollo es prácticamente imposible, como puede ser el simple hecho de hacer la compra por parte de una persona con discapacidad visual en establecimientos físicos.

Incluso superada la odisea de realizar la compra (ya sea recurriendo a la compra online o a terceras personas) los problemas siguen existiendo: en la mayoría de los casos los productos carecen de elementos básicos de información tan importantes como su fecha de caducidad o composición alimentaria adaptados para personas con discapacidad visual.

Eludir casos como los expuestos anteriormente, tanto por parte de las cadenas de supermercados, como por los fabricantes, limita los derechos de este sector de la población como consumidores.

En este proyecto queremos plantear una solución que sea lo suficientemente sencilla tanto en aplicación como en costes para facilitar una rápida implantación. En concreto se propone aplicar un etiquetado NFC a todos los productos en su costado, de tal forma que mediante el uso de una aplicación móvil el usuario sea capaz de saber qué tiene en sus manos (precio, descripción, etc.) con tan solo acercar el móvil a esa zona del producto. Este producto podrá ser almacenado (y eliminado llegado el caso) si el usuario lo desea en una sección de la aplicación (almacén individual), permitiendo así mantener un inventario completo de todas sus adquisiciones y el estado actual de su despensa.

Adicionalmente, con el fin de convertir la aplicación en algo más atractivo para usuarios habituales de Android, que carecen de discapacidad visual, ha sido añadida la funcionalidad de listas de compras individuales y compartidas. Esta característica facilita la tarea de elaborar una lista de compra, permitiendo añadir, eliminar y marcar como seleccionados los diferentes productos que el usuario indique, con la posibilidad de compartirlo con otro usuario si lo desea, todo ello manteniendo actualizaciones en tiempo real entre las distintas cuentas.

ABSTRACT

Accessibility is a right, of the utmost importance. The whole of society must ensure that it is fulfilled and improved. However, there are daily situations where its development is practically impossible, such as the simple act of shopping for a visually impaired person in physical establishments.

Even after the odyssey of making the purchase (either by resorting to online shopping or to third parties) the problem still exists. In most cases products lack basic information elements as important as their expiration date or food composition adapted to make them available to people with visual impairment.

Ignoring the situations described above, both by supermarket chains and by manufacturers, limits the rights of this sector of the population that they should be able to enjoy as consumers.

In this project we want to propose a solution that is simple enough in both application and cost to allow for rapid implementation. Specifically, we propose an NFC labeling of all products on their side, so that by using a mobile application the user is able to know what he has in his hands (price, description, etc.) just by bringing the mobile to that area of the product. This product can be stored (and removed if necessary) if the user wishes in a section of the application (individual warehouse), thus allowing to maintain a complete inventory of all their purchases.

Additionally, in order to make the application more attractive to regular Android users, who are not visually impaired, the functionality of individual and shared shopping lists has been added. This feature facilitates the task of creating a shopping list, allowing to add, remove and mark as selected the different products that the user indicates, with the possibility of sharing it with another user if desired, all while maintaining real-time updates between different accounts.

ÍNDICE DEL PROYECTO

1	Introducción.....	10
1.1	Análisis de lo existente	10
1.2	Objetivos y solución propuesta.....	17
1.3	Estructura de la memoria.....	18
2	Análisis tecnológico.....	20
2.1	Lenguajes, plataforma y librerías	20
2.1.1	Android.....	20
2.1.2	Java	20
2.1.3	Python.....	21
2.1.4	Selenium.....	22
2.1.5	BeautifulSoup	22
2.2	Bases de datos	22
2.2.1	Firebase Realtime Database.....	22
2.2.2	Realm.....	23
2.3	Experiencia e Interfaz de usuario.....	24
2.3.1	XML.....	24
2.3.2	Material Design.....	24
2.3.3	Google TalkBack	24
2.3.4	Android Speech Recognizer.....	25
2.4	Herramientas adicionales	25
2.4.1	Android Studio	25
2.4.2	Git y GitHub	26
2.4.3	Jira	26
2.4.4	Toggl	26
2.4.5	Google Drive.....	27
2.4.6	Office 365 (Word)	27
2.4.7	Google Meet	27
3	Planificación	28
3.1	Estimación temporal del proyecto	28
3.2	Metodología	28

3.3	Descripción de las iteraciones	31
3.3.1	Sprint 1: Inicio del proyecto	31
3.3.2	Sprint 2: Primera funcionalidad	34
3.3.3	Sprint 3: Primera aplicación funcional.....	36
3.3.4	Sprint 4: Almacén de productos y pivotación de la aplicación	38
3.3.5	Sprint 5: Lista de la compra funcional.....	40
3.3.6	Sprint 6: Funcionalidad compartida.....	42
3.3.7	Sprint 7: Accesibilidad de la aplicación.....	45
3.3.8	Sprint 8: Corrección de errores	46
3.4	Backlog del proyecto	48
3.5	Burn-up global del proyecto.....	50
3.6	Presupuesto del proyecto	51
3.6.1	Costes de personal.....	51
3.6.2	Costes de material.....	51
3.6.3	Costes indirectos	52
3.6.4	Beneficio industrial	52
4	Análisis y diseño	53
4.1	Análisis	53
4.1.1	Casos de uso	53
4.1.2	Matriz de trazabilidad	58
4.1.3	Mockups.....	60
4.1.4	Diagramas de dominio.....	63
4.2	Diseño	64
4.2.1	Arquitectura	64
4.2.2	Diagrama de componentes	65
5	Manual de Usuario.....	67
5.1	Gestión de usuarios	67
5.2	Gestión de lista de la compra	70
5.3	Gestión de lista de la compra compartida	80
5.4	Gestión de identificación.....	83
5.5	Gestión de productos.....	86

5.6	Gestión de notificaciones.....	88
5.7	Gestión de control por voz.....	89
6	Pruebas	95
6.1	Pruebas funcionales	95
6.1.1	Sprint 1.....	95
6.1.2	Sprint 2.....	96
6.1.3	Sprint 3.....	96
6.1.4	Sprint 4.....	97
6.1.5	Sprint 5.....	97
6.1.6	Sprint 6.....	97
6.1.7	Sprint 7.....	99
6.1.8	Sprint 8.....	99
6.2	Pruebas de rendimiento.....	99
6.2.1	Reposo sin actividad	100
6.2.2	Reposo con actividad	100
6.2.3	Primer plano con actividad	102
7	Conclusiones.....	104
7.1	Objetivos iniciales	104
7.2	Futuras mejoras	105
7.3	Reflexión personal	106
8	Referencias	107

ÍNDICE DE ILUSTRACIONES

Ilustración 1 - Baliza Beepcons y su app oficial [5]	11
Ilustración 2 - Interfaz básica de Lazzus [8]	12
Ilustración 3 - Ejemplo de uso [10].....	13
Ilustración 4 - Ejemplo de uso de TapTapSee.....	14
Ilustración 5 - Interfaz de Bring!	15
Ilustración 6 - Interfaz básica de Listonic.....	16
Ilustración 7 Framework Sprint [33]	29
Ilustración 8 - Rendimiento Sprint 1	33
Ilustración 9 - Rendimiento temporal Sprint 1	33
Ilustración 10 - Rendimiento Sprint 2.....	35
Ilustración 11 - Rendimiento temporal Sprint 2.....	35
Ilustración 12 - Rendimiento Sprint 3	37
Ilustración 13 - Rendimiento temporal Sprint 3.....	38
Ilustración 14 - Rendimiento Sprint 4	39
Ilustración 15 - Rendimiento Temporal Sprint 4	40
Ilustración 16 - Rendimiento Sprint 5	41
Ilustración 17 - Rendimiento Temporal Sprint 5	42
Ilustración 18 - Rendimiento Sprint 6	44
Ilustración 19 - Rendimiento Temporal Sprint 6	44
Ilustración 20 – Rendimiento Sprint 7	45
Ilustración 21 – Rendimiento temporal Sprint 7	46
Ilustración 22 – Rendimiento Sprint 8	47
Ilustración 23 – Rendimiento temporal Sprint 8	47
Ilustración 24 - Pantalla de Sign-up y Log-in	60
Ilustración 25 - Vista principal	61
Ilustración 26 - Vista de Identificación	62
Ilustración 27 – Diagrama de dominio Firebase private	63
Ilustración 28 – Diagrama de dominio Firebase public.....	63
Ilustración 29 – Diagrama de dominio Realm.....	64
Ilustración 30 – Diagrama de arquitectura.....	65

Ilustración 31 – Diagrama de componentes	65
Ilustración 32 – Vista de bienvenida	68
Ilustración 33 – Selección de cuenta	68
Ilustración 34 – Creación de una nueva cuenta	69
Ilustración 35 – Vista home	69
Ilustración 36 – Vista lateral habilitada	69
Ilustración 37 – Vista home completa	70
Ilustración 38 – Ejemplo de marcar y desmarcar todo	71
Ilustración 39 – Mensaje informativo 1	72
Ilustración 40 – Proceso de vaciar la lista.....	72
Ilustración 41 – Mensaje informativo 2	73
Ilustración 42 – Usuarios con acceso a la lista	73
Ilustración 43 – Proceso cancelar lista	74
Ilustración 44 - Mensaje informativo 3	74
Ilustración 45 - Abrir lista de la compra.....	75
Ilustración 46 – Añadir producto a lista de la compra 1.....	76
Ilustración 47 – Añadir producto a la cesta 2.....	76
Ilustración 48 – Eliminar producto de la cesta	77
Ilustración 49 – Marcar un producto y ocultación del panel superior	78
Ilustración 50 – Borrar todo mediante acción rápida	78
Ilustración 51 – Añadir acceso a un usuario	79
Ilustración 52 – Errores al invitar a un usuario	79
Ilustración 53 - Proceso de eliminar invitación	80
Ilustración 54 - Proceso de aceptar o rechazar invitación.....	81
Ilustración 55 – Lista compartida	82
Ilustración 56 – Ejemplo de lista compartida	83
Ilustración 57 – Activar identificación.....	84
Ilustración 58 – Proceso de identificar un producto.....	85
Ilustración 59 – Detección de NFC desactivado.....	85
Ilustración 60 – Dispositivo sin NFC	86
Ilustración 61 – Almacén de productos personales.....	86

Ilustración 62 – Mostrar almacén	87
Ilustración 63 – Eliminar todos los productos de un almacén	88
Ilustración 64 - Ejemplos de notificaciones	88
Ilustración 65 – Uso de TalkBack y comandos de voz	89
Ilustración 66 – Desmarcar todos los productos mediante comandos de voz	90
Ilustración 67 – Añadir producto a la lista	91
Ilustración 68 – Eliminar producto de la lista	92
Ilustración 69 – Marcar un producto en la lista	92
Ilustración 70 – Identificar un producto 1	93
Ilustración 71 – Abrir almacén.....	94
Ilustración 72 – Aviso de error de lectura	96
Ilustración 73 – Bug al aceptar petición solventado	98
Ilustración 74 - Bug al acceder a lista	98
Ilustración 75 - Rendimiento en reposo	100
Ilustración 76 - Consumo de memoria de rivales	100
Ilustración 77 – Rendimiento en reposo activo	101
Ilustración 78 – Rendimiento de la CPU en estado de reposo activo	101
Ilustración 79 – Rendimiento en primer plano con actividad.....	102
Ilustración 80 – Rendimiento de la CPU en primer plano activo	102
Ilustración 81 – Rendimiento de la conexión	103
Ilustración 82 – Consumo de energía	103

ÍNDICE DE TABLAS

Tabla 1 - Estimación temporal	28
Tabla 2 - Sprint 1 Backlog	32
Tabla 3 - Sprint 2 Backlog	34
Tabla 4 - Sprint 3 Backlog	36
Tabla 5 - Sprint 4 Backlog	39
Tabla 6 - Sprint 5 Backlog	41
Tabla 7 - Sprint 6 Backlog	43
Tabla 8 - Sprint 7 Backlog	45
Tabla 9 – Burn-Up global	50

1 INTRODUCCIÓN

Actualmente, el uso de dispositivos móviles se ha popularizado a tal nivel que las tareas habituales del día a día son realizadas, en un elevado porcentaje de los casos, a través de ellos (tales como el consumo multimedia, comunicación, pagos, compras, y un largo etcétera). Su uso se ha extendido y afianzado su posición de una manera muy sólida debido, en gran parte, a la capacidad de centralizar multitud de servicios en un solo dispositivo.

Como muestra de este aumento exponencial del uso de los móviles, y concretamente de los que utilizan la plataforma Android, puede señalarse el crecimiento de su plataforma aplicaciones, “Google Play Store”. En 2010, cuando el servicio aún se llamaba “Android Market” contaba con un total de 100.000 aplicaciones. En la actualidad, el número total de aplicaciones disponible roza los 2,6 millones de aplicaciones [1].

Sin embargo, existe un sector de la población que se ha visto claramente afectado por toda esta unificación de tareas en un único dispositivo. De todas las aplicaciones actuales, un número despreciable en comparación con el total cuentan con soporte específico para personas con discapacidad visual, a quiénes más que una ayuda les supone un obstáculo tener que realizarlo todo a través de una pantalla táctil. Esto es algo comprensible, puesto que dicha adaptación conlleva asumir un sobrecoste que, en la mayoría de los casos no deriva en una clara ventaja competitiva frente al resto de rivales. Sin embargo, esto no es una situación deseable, por lo que con este proyecto nos proponemos ayudar a paliarlo.

1.1 Análisis de lo existente

Para conocer el verdadero panorama en el que se encuentra la situación actual (respecto a la adaptación de tecnologías para personas con discapacidad visual) contactamos con la organización ONCE [2], en concreto con Adriana Pérez Barrones, Jefa del Departamento de Servicios Sociales para Afiliados a través de una reunión presencial en la Delegación Territorial de Sevilla (C/ Resolana, 30 - 41009).

Entre los datos aportados en la reunión, destaca que casi la totalidad de los afiliados utilizan dispositivos iOS (iPhones en su mayoría). El motivo de ello, casi en la totalidad de los casos, se debe a la posibilidad de poder usar VoiceOver (funcionalidad exclusiva de Apple) [3]. VoiceOver plantea un sistema bastante simple y amigable para el usuario donde la pantalla es leída por pasos y el usuario interactúa, en el caso del iPhone 8, a base de clics sobre el botón home.

Por otro lado, también existen afiliados que usan Android, sin embargo, son un porcentaje muy bajo en comparativa. En este caso, es utilizado Google TalkBack [4], sistema similar al de Apple. Quizás en la actualidad Google TalkBack se encuentra en un estado menos avanzado en comparación con el producto de Apple. Todo depende del apoyo de la comunidad de desarrolladores existente detrás de él y la influencia de las aplicaciones que requieran de su funcionamiento conjunto con este, facilitando así el interés en la mejora de la herramienta.

Respecto a aplicaciones independientes de los desarrolladores de las plataformas software con las que la comunidad de afiliados se desenvuelve, en la actualidad pueden destacarse un número importante de ellas.

Se comenzarán realizando un análisis de las que han sido desarrolladas para facilitar la identificación.

- **Beepcons**

Se trata de un sistema de uso muy parecido a las etiquetas NFC, pero en su lugar utilizan tecnología Bluetooth de corto alcance. Su finalidad es más cercana a ayudar al usuario a su localización respecto a su entorno en términos de distancia. Que hay en la zona, contextualización los elementos cercanos, etc. Es decir, similar a la función de “*GPS de interiores*”.

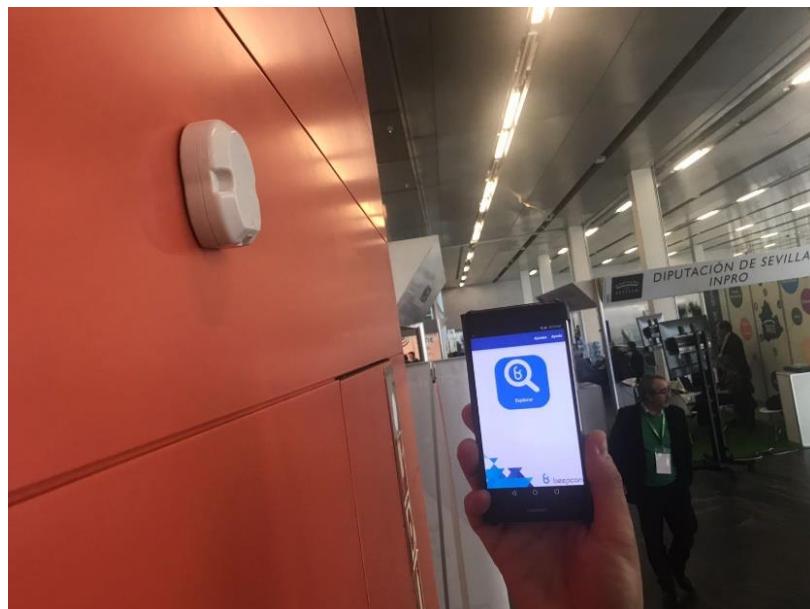


Ilustración 1 - Baliza Beepcons y su app oficial [5]

En la ilustración anterior puede apreciarse un ejemplo de cómo podría usarse. La aplicación entraría en contacto con la baliza y esta le devolvería información de interés con relación a su distancia entre ella y el propio dispositivo. Su situación actual, los elementos cercanos de referencia o recorrido a realizar para llegar a algún lugar pueden ser algunos de los ejemplos de la funcionalidad que ofrece.

En la actualidad, como se le indicó desde la ONCE al autor, su uso está muy poco extendido y se debe, en cierta medida, al alto coste que supone la instalación de balizas compatibles con este sistema. Sin embargo, el sistema se encuentra disponible tanto para plataformas Android como iOS.

Continuando con la disponibilidad del sistema, por desgracia, su implementación y uso efectivo se reduce a ciertos museos y edificios oficiales. Muy pocos centros de ocio apuestan por tecnologías de accesibilidad como esta. Como excepción, en Sevilla puede señalarse el centro comercial “*Nervión Plaza*” el cual ha instalado en toda su zona interior multitud de estos dispositivos [6].

En comparación con la aplicación de este proyecto, aunque su fin último es similar, ayudar a una persona con discapacidad visual, el problema que afronta el totalmente distinto.

Beepcons es utilizado como una plataforma de guiado y localización de punto a punto, no fue diseñada con el fin de poder identificar productos cercanos sino darle a conocer al usuario su situación actual y permitirle su desplazamiento entre distintos puntos sin la dependencia de una tercera persona.

- **Lazzus**

Su fin es muy similar al de *Beepcons*, sin embargo, en este caso la aplicación no necesita terceras herramientas como emisores bluetooth/NFC, sino que requiere tanto la ubicación del dispositivo como del contraste de datos entre Google Places y Open Street Data [7]. *Lazzus* depende del nivel y calidad de las actualizaciones de los sistemas de terceros anteriormente nombrados.

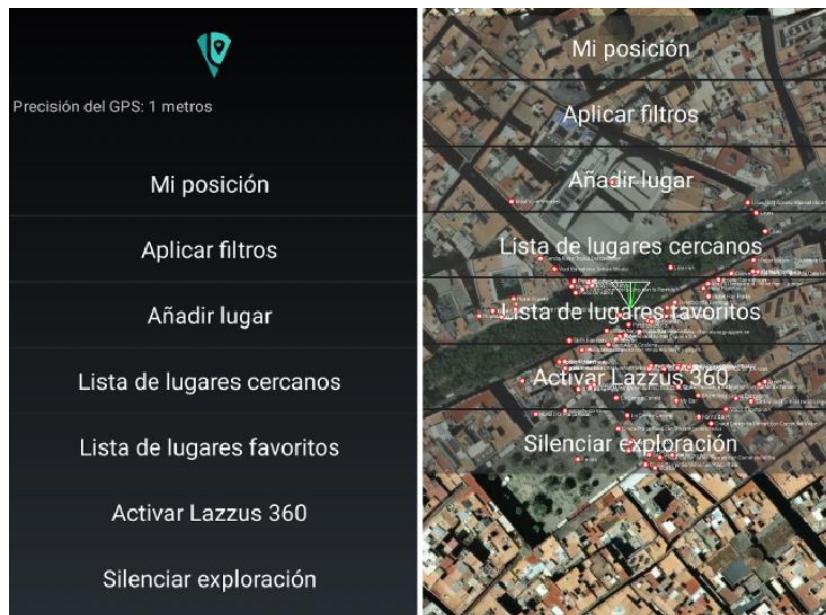


Ilustración 2 - Interfaz básica de Lazzus [8]

El autor del proyecto ha probado esta aplicación en profundidad, para comprobar el desempeño real de la misma, estableciendo las siguientes conclusiones.

Su funcionamiento es muy bueno en grandes ciudades, sin embargo, en pequeñas localidades el servicio se vuelve muy básico y prácticamente sin valor. Por ejemplo, en su campaña de ventas [9] (su uso no es gratuito) anuncian como elemento diferenciador las alertas de cruces y escaleras cercanas. Sin embargo, esta función no se encuentra disponible en la mayoría de las ocasiones, con excepción de grandes núcleos poblacionales donde la información suele actualizarse constantemente.

La dependencia de la calidad de la información de los servicios de terceros como se indicó al inicio supone que, en los casos de pueblos o núcleos de población más pequeños, la aplicación se comporte exactamente igual al guiado de Google Maps.

En definitiva, trata de imitar la funcionalidad de Beepcons con independencia de balizas y en espacios más abiertos, lo que conlleva a la necesidad de tener una enorme y amplia base de datos, así como mantener una información constantemente actualizada.

Respecto a su interfaz es un claro ejemplo de adaptación, ya que se encuentra preparada para poder ser utilizada con independencia visual.

- ***Be My Eyes***

Be My Eyes es mucho más cercana a la filosofía de este proyecto. Su forma de uso es muy simple, el usuario realiza una llamada en la aplicación y sitúa la cámara hacia el objeto que pretende identificar (o la situación en la que necesita ayuda) y es un colaborador el encargado de describir que se puede ver a través de la cámara del usuario que necesita ayuda.

Son evidentes ciertas limitaciones en el uso de esta aplicación como, por ejemplo, depender de que el usuario sea capaz de realizar una captura correcta de lo que realmente pretendía conocer. Aunque parezca bastante sencillo para un usuario normal, en ocasiones resulta realmente difícil encontrar y poder capturar de forma que pueda ser entendible la información de la que se necesita ayuda.

Puede ser utilizado el ejemplo de la fecha de caducidad. El usuario puede ser capaz de hacer un enfoque centrado y definido pero que sin embargo apunta a otra situación del producto que no muestre esa información de la que realmente se requería una aclaración.

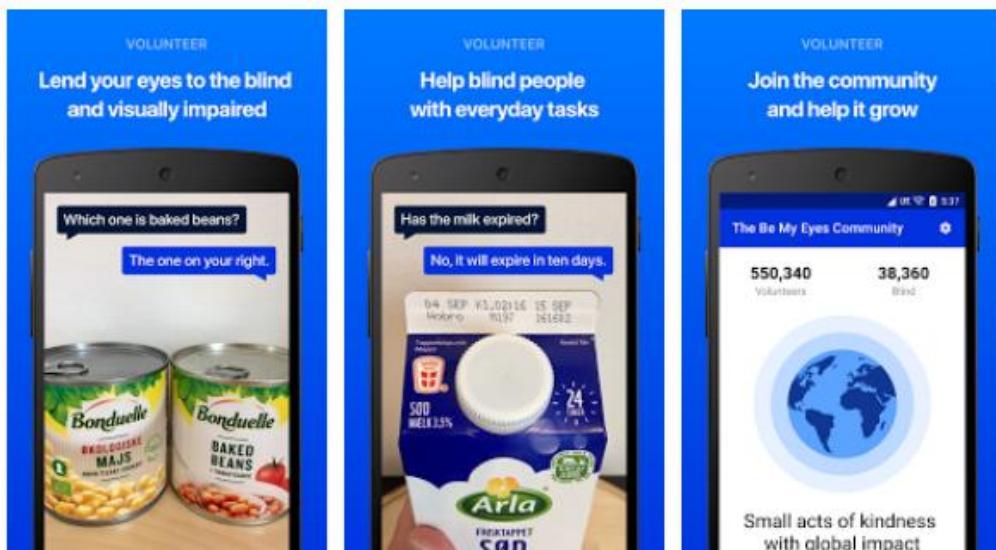


Ilustración 3 - Ejemplo de uso [10]

Continuando con el problema de la dependencia de un usuario colaborador puede añadirse que en la inmensa mayoría de las ocasiones la respuesta no es instantánea, por lo que el usuario tendrá que esperar a la respuesta de un voluntario. Además, pueden existir casos en los que el usuario se sienta incómodo de pedir ayuda a otra persona. El fin de la aplicación que se quiere desarrollar en este proyecto es conseguir una independencia total no sujeta a otros actores que deberían entrar en acción en el contexto de la aplicación.

Clarify es independiente de voluntarios externos, la etiqueta NFC que se utiliza le proporciona toda la información necesaria al usuario, bastaría con acercar su dispositivo a un costado y la información se le mostraría de manera inmediata.

A pesar de todo esto, al autor le gustaría reconocer la fantástica propuesta de esta aplicación, ya que no solo se limita al reconocimiento del producto, sino que de igual forma que las dos primeras aplicaciones que han sido comentadas en este bloque introductorio, puede ser una excelente herramienta que ayude al usuario a conocer la situación en términos de localización en la que se encuentra.

Por último, respecto a términos de accesibilidad, se encuentra totalmente adaptada ya que se reduce a pulsar un botón y entrar en contacto con una persona.

- **TapTapSee**

Es una aplicación diseñada específicamente para la mejora de la accesibilidad de personas con discapacidad visual.

En esta ocasión se repite la funcionalidad de la aplicación anterior (“Be My Eyes”), es decir, el usuario desconoce un objeto del que pretende obtener información, realiza una foto y la aplicación le devuelve todo lo que es capaz de extraer a partir de esta foto.

Como diferencia a *Be My Eyes* esta aplicación no necesita de un voluntario externo para procesar la información, sino que utiliza la inteligencia artificial para capturar toda la información posible a partir de la captura.

Tiene un desempeño realmente bueno, es capaz de diferenciar multitud de objetos, con algo de retardo, pero no falla en casi ninguna ocasión. Sin embargo, la información dada es muy general y, aunque útil, en muchos casos se queda un poco corta. Como casos de ejemplo donde no es tan exacta como podría esperarse puede señalarse la descripción de medicamentos (no es capaz de decir el nombre de este) o la fecha de caducidad de los productos.

Otro de los fallos que ha podido detectar el autor es que la aplicación tiene errores de traducción, utilizando una mezcla de español e inglés en la información que devuelve.



Ilustración 4 - Ejemplo de uso de TapTapSee

En la ilustración anterior puede verse un ejemplo de cómo se usaría. El usuario apuntaría al objeto que desea identificar y la aplicación en la parte inferior le informaría que es lo que muestra la imagen, así como también repetiría esta descripción mediante una función de dictado por voz.

Hasta aquí la comparativa sobre las aplicaciones utilizadas como identificadoras de objetos a continuación se realizará un pequeño resumen de las aplicaciones con más descargas en la plataforma Android para la realización de listas de la compra.

- ***Bring! Lista de la compra***

Es una aplicación muy trabajada a nivel visual que, al igual que la aplicación que ha sido desarrollada en este proyecto, permite realizar listas de la compra compartidas.



Ilustración 5 - Interfaz de Bring!

A pesar de su enfoque tan visual, pueden localizarse algunos errores bastante importantes, la mayoría relativos a la intuitividad de la aplicación.

Un primer ejemplo podría ser borrar un elemento de la lista. Para realizarlo hay que entrar dentro de las opciones del propio producto, cerrar el teclado y activar la pestaña ajustes, una vez dentro puede activarse la opción de eliminarlo y posteriormente tendría que confirmarse la acción de borrado.

Otro concepto que se considera un error desde el punto de vista del autor es la imposibilidad de borrar el acceso a una persona que ha sido invitada a una lista. En esta aplicación no existe el rol administrador por lo tanto una vez enviada una invitación esta no puede modificarse. Si el usuario la acepta tiene acceso completo a la lista y puede enviar nuevas invitaciones.

Por otro lado, tampoco se informa del estado de las invitaciones que han sido enviadas ni tampoco los usuarios que cuentan con acceso a la lista en ninguno de los apartados. Además, el proceso para conocer las personas que cuentan con acceso a la lista en cuestión es bastante difícil de acceder por la cantidad de pasos a realizar.

En relación a apartados de accesibilidad “*Bring!*” no cuenta con ninguna opción referente a ello en ninguno de sus apartados, se realiza todo a base de múltiples toques en la pantalla. Además, el intento de dar un toque más visual a la aplicación es excesivo en algunos casos. Véase el caso de que la imagen predefinida de los productos es considerablemente mucho más grande que el nombre del producto en sí. Incluso en los casos en los que la aplicación no es capaz de encontrar una buena imagen y utiliza la inicial del producto que hemos introducido.

Esta aplicación se encuentra disponible en todas las plataformas (Android, iOS y PC).

- ***Listonic***

En esta ocasión, a diferencia de la anterior, la aplicación se centra más en la funcionalidad que en lo visual.

Permite elaborar una lista de la compra a base de añadir y eliminar productos, así como habilita de una forma muy visual el marcado de un elemento que ha sido seleccionado por algún usuario.

Por otro lado, “*Listonic*” también permite compartir la lista con otro usuario y eliminarle el acceso a esta si el propietario lo considera oportuno, cosa que en la aplicación anterior no podía realizarse.

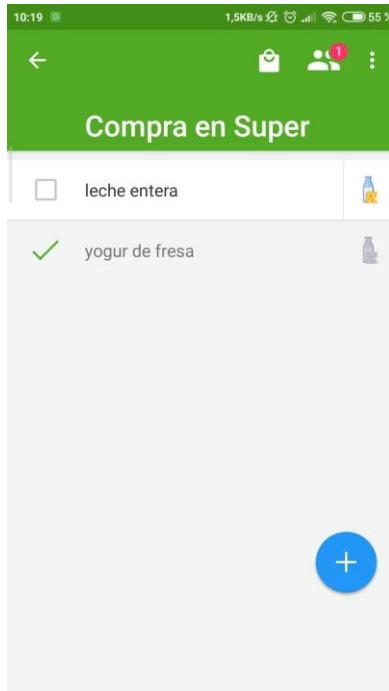


Ilustración 6 - Interfaz básica de Listonic

A pesar de no ser mucho, “*Listonic*” si dispone de una opción muy interesante de mejora de la accesibilidad, como es el dictado por voz de producto a añadir. Una vez el usuario se encuentre en la pantalla de añadir nuevos productos se le proporciona, si desea activarlo, la opción de introducir el nombre de los productos de forma dictada.

Sin embargo, la tarea de añadir un producto es un poco engorrosa en la metodología de uso. Casi todo en esta aplicación excede en un toque más de lo que el usuario normal está acostumbrado a realizar; por ejemplo, en la acción de añadir un producto el usuario

tiene que pulsar como mínimo tres veces la pantalla (sin contar la escritura del nombre del producto en sí).

Por otro lado, un gran defecto que se ha identificado es la lentitud de las actualizaciones del sistema en listas compartidas donde se puede llegar a contabilizar tiempos de espera de entre 20 y 30 segundos desde que un usuario realiza una acción y esta puede verse reflejada en el otro dispositivo.

1.2 Objetivos y solución propuesta

¿Cómo podría realizarse una aplicación realmente útil para facilitar el día a día de una persona con discapacidad visual que a la vez sea lo bastante atractiva para el público general?

El objetivo de este proyecto es dar una solución a la pregunta anterior y para ello se ha diseñado una aplicación destinada a ser usada habitualmente por una persona con discapacidad visual siendo una de sus principales funciones la identificación de cualquier producto en sí y las peculiaridades que lo caracterizan (modelo, color, marca, etcétera). El usuario solo deberá acercar su teléfono a un costado del producto y toda la información se le detallará al instante.

En concreto se plantea establecer una etiqueta NFC estandarizada que contenga el ID del producto y, a través de este, entrar en contacto con un servidor propio. Se le proporcionará información importante y actualizada referente al producto tales como el modelo, la marca, el precio, el color, etc. Todo aquel dato que resulte interesante para distinguir al producto e importante para el consumidor.

Además, para agilizar la gestión de inventario, todos estos productos que han sido identificados por la aplicación tienen la posibilidad de ser añadidos a un gestor personal si el usuario lo desea, donde será capaz de administrar todas las compras que ha realizado y ser consciente de que posee realmente. Por tanto, se aborda un problema que va más allá de la realización de la compra en sí.

Este proyecto pretende ser el asistente de identificación en el día a día del usuario, una vía de independencia para evitar tener que recurrir a una tercera persona en casos que pueden llegar a ser tan cotidianos como saber de qué color es la prenda que portamos o la talla de un pantalón que queremos adquirir.

Sin embargo, como es evidente, ninguna gran cadena comercial asume en la actualidad como viable el aumento de costes en el uso de etiquetas NFC sin algún beneficio a cambio. Es por ello por lo que uno de los grandes retos de este proyecto ha sido lograr encontrar un punto intermedio entre accesibilidad de usuarios y ventaja competitiva dentro del mercado, y para ello la idea que se ha planteado ha sido la gamificación del proceso de compra desarrollando para ello la funcionalidad de lista de compra personal y compartida.

Así, el usuario de la aplicación puede elaborar su lista de la compra de forma manual rellenando un simple campo de texto, eliminándolos e incluso marcándolos como seleccionados mientras realiza la propia compra. Además, si lo considera oportuno puede invitar a nuevas personas a esta lista. Ambas pueden añadir, borrar productos en tiempo real e incluso marcarlos como seleccionados mientras realizan la compra.

El punto de enlace entre la gamificación y la fidelización comienza en la adición de productos a la lista. El usuario puede navegar entre los productos con los que cuenta en su almacén en ese momento y localizar los que va a necesitar renovar/comprar próximamente. El usuario antes de eliminar el producto lo añade a su lista y sabe con exactitud qué producto tiene que comprar y en el establecimiento en que se encuentra, dándole así un valor extra de fidelización de clientes hacia los puntos de venta.

Abarcados los puntos anteriores, la aplicación que se ha desarrollado pretende considerar los siguientes apartados, distinguidos en función del tipo de usuario.

En primer lugar, para los usuarios con discapacidad visual, nuestro proyecto propone como principal objetivo **ayudarles**, y para ello se establecen los siguientes subobjetivos. Por otro lado, el número que aparece entre paréntesis será utilizado como un identificador para la tabla de trazabilidad que será expuesta en el apartado 4.1.2.

- (1) La independencia en tareas del día a día cotidianas, como por ejemplo conocer el color de una prenda antes de usarla o identificar un producto antes de realizar la compra.
- (2) Estar completamente adaptada a su uso conjunto con TalkBack [11], el cual es gestor de accesibilidad Android basado en el dictado de pantalla por toques.
- (3) El uso a través de comandos de voz fáciles e intuitivos en toda la aplicación.

Seguidamente, para todos los tipos de usuario se establece un objetivo **facilitador**, el cual se constituye a partir de los siguientes subobjetivos:

- (4) Una herramienta utilizada para realizar listas de productos para comprar.
- (5) Capacidad para compartir las listas entre los usuarios registrados en la aplicación.
- (6) La identificación de productos mediante el uso de etiquetas NFC.
- (7) La posibilidad de añadir o eliminar productos que han sido identificados a los distintos almacenes personales.
- (8) Actualizaciones en tiempo real de todos los servicios de la aplicación, como la interacción entre varios usuarios sobre una misma lista (agregación o eliminación de productos y marcarlos como seleccionados), modificaciones de los almacenes, etc.
- (9) Una interfaz sencilla y agradable al uso.

Finalmente, queda establecido como objetivo **fidelizar la clientela** de establecimientos de venta, tal y como se nombró anteriormente.

1.3 Estructura de la memoria

El contenido de esta memoria se enmarca en la siguiente estructura:

1. Introducción

Contextualización del problema que tratamos de abordar y análisis de la situación que ofrece el panorama actual a nivel de aplicaciones.

2. Análisis tecnológico

Se realizará una explicación en detalle de la tecnología utilizada, es decir, lenguajes usados en el proyecto, software adicional, así como herramientas de apoyo utilizadas durante el desarrollo.

3. Metodología y planificación

Este apartado recoge la especificación y detalle de la metodología que ha sido utilizada durante el desarrollo del proyecto, la forma y estrategia que se ha optado para la planificación de las distintas tareas.

4. Análisis y diseño

En esta parte se detallarán los requisitos del sistema, las pruebas de usuario y los distintos mockups que han sido diseñados.

5. Manual de usuario

Se explicará cómo se usa la aplicación y como se utilizan las funcionalidades que han sido planteadas en el sistema.

6. Pruebas

Muestra de las pruebas que se han realizado para validar la funcionalidad planteada.

7. Conclusiones

Opinión final del autor del desempeño del trabajo y valor generado durante el desarrollado del proyecto.

2 ANÁLISIS TECNOLÓGICO

En esta sección se detallará un análisis de toda la tecnología utilizada durante el proyecto, tanto para el desarrollo de la aplicación, como para la planificación y gestión de este.

La estructura de este bloque queda dividida en cuatro apartados. Se comenzará en primer lugar con el análisis de los lenguajes, plataformas y principales librerías auxiliares utilizadas. Posteriormente se comentarán las tecnologías de bases de datos de este proyecto, continuando con el apartado de experiencia de usuario. Finalmente se cerrará esta sección con la descripción de las herramientas adicionales utilizadas para la gestión y desarrollo.

2.1 Lenguajes, plataforma y librerías

Serán detallados tanto los lenguajes utilizados como las plataformas elegidas y herramientas auxiliares.

2.1.1 Android

Desde el primer momento del desarrollo del proyecto se decidió que se debería trabajar con los dispositivos móviles debido a su portabilidad y que comparten el día a día de la totalidad de las personas actualmente. Surge a partir de este momento la duda entre desarrollar una aplicación para dispositivos Android o iOS ya que su estructura es completamente distinta.

Como anteriormente se indicó, desde la sede de la ONCE se indicó que el gran porcentaje de los afiliados utilizan iPhone como dispositivo principal en caso de utilizar un smartphone en su día a día.

Como consecuencia, el desarrollo de una aplicación para dispositivos iPhone conlleva programar en el lenguaje Swift el cual en la actualidad solo puede utilizarse de forma funcional si se posee un dispositivo con macOS [12] (además contar con un dispositivo iPhone para realizar pruebas, en añadido).

La adquisición ambos dispositivos fue totalmente descartada por parte del autor ya que el coste asociado a esas compras era demasiado alto.

Tomada la decisión de rechazar la programación en Swift se decidió migrar todo el plan a la plataforma Android.

En cuanto a Android, se fundó como una compañía independiente en 2003 y fue adquirida por Google en 2005 y en 2008 HTC lanzó el primer dispositivo con Android OS [13]. Las aplicaciones Android son desarrolladas en Java y recientemente también permiten el uso de Kotlin.

Según datos del 2019, en el mundo existen 2.5 billones de dispositivos activos que utilizan Android [14]. Respecto al mercado español, el 85% de los dispositivos utilizan la plataforma Android, frente algo más del 14% que utiliza iOS [15].

2.1.2 Java

Cuando se plantea el desarrollo de una aplicación Android, hasta hace unos años la única opción válida consistía en elegir Java como lenguaje de programación, sin

embargo, en 2012 la compañía JetBrains [16] lanzó (en formato de código abierto) un nuevo lenguaje llamado Kotlin [17], el cual en cierta medida es una modificación de Java. En 2017, Google anunció soporte oficial para Kotlin en Android Studio IDE.

Mostrados los puntos anteriores se tuvo que decidir entre la elección de uno de estos dos lenguajes, siendo Java finalmente el escogido, principalmente, por los motivos que serán expuestos a continuación:

- **Período de aprendizaje**

Durante toda la formación académica del autor, ha sido Java el lenguaje base utilizado en casi la totalidad de las asignaturas. La elección de Kotlin, a pesar de mostrar una curva de aprendizaje bastante corta según multitud de reseñas hubiera supuesto un añadido en tiempo que no podía ser asumido en ese momento.

- **Afinidad del programador**

Java es el lenguaje en el que se siente más cómodo el autor, en el que presupone una fluidez que le costaría alcanzar si se hubiera decantado por la elección de Kotlin.

Se partió con un desconocimiento total de la programación en Android, por lo que allanar el proceso en cierta medida partiendo de un lenguaje conocido facilitaría el transcurso del proyecto, evitando así tener que comenzar totalmente de cero y teniendo unas ciertas bases sobre las que apoyar el resto del desarrollo.

- **Mayor comunidad**

A pesar del gran crecimiento en el que se encuentra Kotlin, Java presenta un mayor respaldo a nivel de comunidad de usuarios que le dan soporte. Se posee un mayor abanico de posibilidades para la solución de problemas en el caso de que aparecieran.

Finalmente, respecto a Java, es un lenguaje de programación orientado a objetos el siendo actualmente es uno de los más utilizados. Su desarrollo estuvo liderado por James Gosling en Sun Microsystem [18]. En la actualidad sigue recibiendo actualizaciones por parte de Oracle y de la comunidad de software libre.

2.1.3 Python

Con el fin de hacer aún más reales los datos utilizados en el proyecto se decidió obtener una base de información bastante sólida. Para ello, sería necesario contar con un conjunto bastante grande de datos sobre productos actuales vendidos al público general.

Establecido lo anterior y como forma más sencilla de realizarlo se optó por crear un proceso de scrapping sobre el portal de venta online de la compañía Mercadona [19].

Para ello ha sido utilizado Python, ya que además de que es el lenguaje más utilizado para este fin, el autor posee conocimientos previos sobre el mismo, útiles para esta tarea. El proceso de desarrollo de toda esta funcionalidad será detallado posteriormente.

En lo que respecta a Python, es un lenguaje de programación de alto nivel, interpretado y orientado a objetos [20].

2.1.4 Selenium

Una de las herramientas para el tratamiento de datos que ha sido utilizada para la realización del proceso de scrapping ha sido Selenium.

Inicialmente Selenium fue desarrollado como un entorno de pruebas que se utiliza para comprobar si el software que se desarrolla funciona correctamente. Sin embargo, es una herramienta bastante potente, que permite hacer que el navegador funcione tal y como si lo estuviese manejando un humano. Facilitando la tarea de extracción de datos de las distintas webs.

2.1.5 BeautifulSoup

Como complemento a la tecnología anterior ha sido utilizada esta herramienta. BeautifulSoup se emplea para analizar documentos HTML, crea un árbol con toda la información de documento sobre el que podemos iterar y tratar los datos a partir de las distintas etiquetas.

2.2 Bases de datos

La aplicación desarrollada en este proyecto utiliza de forma dual dos bases de datos para gestión del contenido.

2.2.1 Firebase Realtime Database

Debido a las características de la aplicación era necesario encontrar un servicio que permitiera mantener la persistencia de los datos en la nube en tiempo real. Firebase, al ser un producto propio de Google, así como por la propia recomendación del tutor de proyecto, se postuló como la mejor opción a elegir.

En cuanto a esta base de datos, de tecnología NoSQL [21], señalaremos que permite una sincronización en tiempo real con la aplicación de forma bidireccional, es decir, los datos que se modifican en la base de datos afectan y se ven reflejados directamente sobre los dispositivos. Esta última característica encaja a la perfección con la intención del proyecto, por ejemplo, en el caso de mantener actividades compartidas en tiempo real entre dos usuarios.

A pesar de todas las ventajas ofrecidas por Firebase Realtime Database, se ha optado por mantener una doble base de datos con el uso de Realm (el cual será desarrollado en el siguiente apartado) para el trabajo de datos a nivel local.

El motivo de la utilización de esta metodología se justifica para facilitar que el uso de toda la aplicación sea funcional sin necesidad de depender de una conexión a internet manteniendo así la persistencia de los datos sin contacto con la red en los casos donde, por las características de Realm y la lógica de negocio de nuestra aplicación, se produciría un bloqueo.

Por todo lo anterior, la estructura de la aplicación queda diseñada de tal forma que el núcleo principal de la aplicación trabaja con Realm mientras que un proceso secundario mantiene la constancia de los datos a nivel de Firebase.

2.2.2 Realm

Como base de datos local se ha decidido utilizar Realm [22] en su versión 6.0.2, la más actualizada en el momento de comenzar este proyecto.

Realm es una base de datos NoSQL concebida para su uso en aplicaciones móviles tanto en Android como en iOS. Se caracteriza por su simplicidad a la hora de realizar consultas y actualización de datos. Muestra de esta simplicidad véase las siguientes dos capturas extraídas del portal Medium [23] en las que se representa una consulta básica realizada a una base de datos SQLite (siendo esta una de las más utilizadas en la actualidad en el panorama Android) y seguidamente la consulta homóloga utilizando esta vez una base de datos Realm. La diferencia en términos de simplificación es evidente.

Consulta básica, utilizando una base de datos **SQLite** en Android:

```
@Override
public void getFavorites(int page, GetFavouritesCallBack callback) {
    List<Favourites> favourites = new ArrayList<>();
    try {
        Cursor cursor = mDb.query(
            FavouritesDBContract.FavouritesEntry.TABLE_NAME, null,
            null, null, null, null, null);
        if (cursor != null && cursor.getCount() > 0) {
            while (cursor.moveToNext()) {
                String title =
                    cursor.getString(cursor.getColumnIndexOrThrow(FavouritesDBContract.FavouritesEntry.COLUMN_NAME_TITLE));
                String imagePath =
                    cursor.getString(cursor.getColumnIndexOrThrow(FavouritesDBContract.FavouritesEntry.COLUMN_NAME_IMAGE_URL));
                Favourites favourite = new Favourites(title, imagePath,
                    null);
                favourites.add(favourite);
            }
        }
    } catch (IllegalStateException e) {
        callback.onDataNotAvailable();
    }
}
```

Consulta básica homóloga a la anterior, utilizando una base de datos **Realm** en Android:

```
public List<Favourites> getFavorites() {
    Realm realm = getRealm();
    List<Favourites> favourites = realm.where(Favourites.class).findAll();
    favourites = realm.copyFromRealm(favourites);
    realm.close();
    return favourites;
}
```

Respecto al rendimiento, este mismo portal asegura lo siguiente:

“Realm también es mejor que SQLite cuando hablamos de velocidad. Y este hecho fue probado por la investigación publicada en el sitio web oficial de Realm. Se estableció que Realm es 10 veces más rápido que el SQLite en bruto. Por ejemplo, el Realm permite insertar 20 mil registros en un segundo. SQLite permite insertar sólo 9 mil registros.”

Son estos dos grandes bloques, simplificación y eficiencia los que hicieron decantarse al autor por el uso de esta tecnología.

2.3 Experiencia e Interfaz de usuario

Para el desarrollo de las vistas visibles por el usuario podemos, destacar el uso de XML y Material Design, así como Google TalkBack y Android Speech Recognizer para el manejo de la aplicación del proyecto a través de comandos de voz.

2.3.1 XML

Extensible Markup Language (o XML) es un lenguaje marcado muy parecido a HTML con la diferencia de que las etiquetas en XML no están predefinidas, deben estructurarse en el propio desarrollo. XML puede ser interpretado exactamente como quiera el desarrollador.

Android aprovecha esta característica utilizándola como elemento diferenciador para separar todo el proceso de la lógica de negocio con las vistas de usuarios. Se establecen un subconjunto básico de etiquetas como “*LinearLayout*”, “*RelativeLayout*”, “*Button*” y un sinfín más. A parte de las predefinidas por el sistema Android, pueden ser creadas por el propio usuario, de hecho, existen una gran cantidad de librerías externas creadas por la comunidad que han diseñado increíbles funcionalidades, bastando simplemente con importarlas para un uso libre en el caso de que su licencia lo permita.

2.3.2 Material Design

Material Design se trata de un patrón y filosofía de diseño cuyo primer objetivo fue Android, avanzando posteriormente a los distintos sistemas web y plataformas. Fue anunciado por Google en 2014 y comenzó a extenderlo a todos sus productos a partir de esa fecha, reemplazando de esta manera al antiguo patrón llamado “*Holo*” [24].

Con el fin de estandarizar esta filosofía de diseño, Google ha hecho públicas una serie de librerías que facilitan el desarrollo de aplicaciones siguiendo sus patrones de diseño [25].

Desde el punto de vista de este proyecto se ha hecho un esfuerzo para que la aplicación que se ha desarrollado cumpla con los criterios establecidos por Google en relación con Material Design.

2.3.3 Google TalkBack

Como anteriormente fue comentado, TalkBack [4] es una herramienta nativa del sistema Android la cual es capaz de dictar al usuario la pantalla activa en ese momento e

interactuar con las distintas opciones mediante un doble toque en cualquier parte de la pantalla.

Su correcto funcionamiento depende de utilizar etiquetado descriptivo en el momento de construir las vistas y así como intentar reducir el número de interacciones que requieran el uso de gestos como deslizar (o al menos ofrecer alguna alternativa como un botón).

2.3.4 Android Speech Recognizer

SpeechRecognizer [26] es una librería propia de Google la cual, tras ser habilitada por el usuario, devuelve como una cadena de caracteres todo lo que ha sido capaz de transcribir a partir del audio.

Esta librería ha sido utilizada por el autor para construir un sistema de comandos de voz, que permiten reducir al mínimo la necesidad de interactuar con la aplicación mediante gestos y pulsaciones en la pantalla.

2.4 Herramientas adicionales

En la sección que será desarrollada a continuación se detallarán todas las herramientas que han sido utilizadas durante este proyecto. Desde aplicaciones para el desarrollo y la gestión del código hasta planificadores de tareas.

2.4.1 Android Studio

Android Studio es el entorno oficial de desarrollo para aplicaciones Android basado en el IDE IntelliJ IDEA, gratuito y de código libre. En concreto Android Studio incorpora de forma nativa GitHub como gestor de código integrado en el propio IDE de la aplicación, así como diversas funcionalidades de conectividad directa con las distintas aplicaciones de Google (Firebase, por ejemplo).

Por otro lado, Android Studio permite ejecutar el código y depurarlos tanto en un dispositivo Android externo, como en el propio emulador integrado por el entorno sin necesidad de contar con un dispositivo móvil real. El propio IDE permite realizar estas tareas sin necesidad de reiniciar la aplicación.

Además, cuenta con un gran número de plantillas prediseñadas que permiten trabajar sobre patrones estándar de una forma muy rápida [27].

Android Studio se basa en el uso de Gradle (“*build system*”), el cual es utilizado para administrar y automatizar la compilación. Además, permite definir un subconjunto de configuraciones de compilación personalizadas como, por ejemplo, para utilizar determinadas funcionalidades en el *Layout* (diseñado a través de nuestro archivo XML) a partir de librerías externas.

En cuanto a alternativas, tras analizar las diferentes opiniones de la comunidad Android, es cierto que existen posibles herramientas distintas a Android Studio, tales como Visual Studio Code o Eclipse, sin embargo, se comparte la decisión generalizada de que hasta el momento, Android Studio es la mejor plataforma como entorno de desarrollo por las distintas posibilidades proporcionadas respecto a funcionalidad como fue expuesto anteriormente, de las que carecen el resto de las herramientas de la competencia.

2.4.2 Git y GitHub

En primer lugar, Git es un software para el control de versiones para desarrolladores. Como principal ventaja permite realizar un seguimiento de todas las acciones que se han realizado sobre los archivos a través de la creación de repositorios los cuales permiten volver a una versión anterior en el caso de que se produzcan errores. También permite dividir el trabajo en distintas ramas de desarrollo y la fusión de varios archivos.

Las dos opciones barajadas por el autor antes de comenzar el proyecto, como herramientas de gestión de código remoto fueron GitHub y GitLab ya que, aparte de ser las plataformas referencias, se contaba con amplia experiencia en ambas.

Respecto a GitHub y GitLab son plataformas que incrementan la funcionalidad de Git en lo referente a que permiten mantener todo el repositorio que tenemos a nivel local en la nube, facilitando así su acceso desde cualquier sitio, además de que permite la colaboración entre varios usuarios.

Existe mucha controversia entre cuál de las dos plataformas es mejor, sin embargo, la decisión de optar por una de las dos (que en este caso fue GitHub) se debió a su mejor compatibilidad con el entorno de desarrollo, expresado en el apartado anterior.

Respecto a la configuración que fue realizada se optó por primer lugar por crear una organización compartida por el autor y el tutor de este proyecto. Sin embargo, al carecer de la posibilidad de establecer el repositorio de gestión de código como privado en una organización se decidió cambiar por la opción de un repositorio privado compartido entre el autor y el tutor del proyecto.

En cuanto a GitHub en particular fue adquirida por Microsoft en 2008 por 7.500 millones de dólares [28]. En la actualidad cuenta con funcionalidades muy interesantes como la creación de repositorios compartidos en lectura y edición simultánea, pull requests, forks y muchas otras funcionalidades añadidas bastante interesantes [29].

2.4.3 Jira

Debido a que ha sido utilizada una planificación ágil, como posteriormente será explicada en el apartado tres de este documento, se ha decidido usar Jira [30] como gestor de tareas.

Jira es un software online que permite organizar y gestionar tareas relativas a software que normalmente siguen una metodología ágil ya sea Kanban, Scrum u otras.

En un inicio el tutor del proyecto recomendó usar ZenHub, y su integración con GitHub como herramienta principal, sin embargo, el autor prefería la utilización de Jira por su mayor facilidad de uso y la cantidad de estadísticas y configuraciones que de forma nativa permite realizar. Fue aceptado el cambio y por tanto comenzó a utilizarse durante todo el desarrollo del proyecto.

2.4.4 Toggl

Toggl es una herramienta para el seguimiento de tiempo, especialmente enfocada para desarrolladores, que se integra fácilmente en multitud de dispositivos e incluso como extensión para navegadores.

Su interfaz es muy simple y posibilita su integración con Jira muy fácilmente. Basta con un clic para comenzar el conteo de tiempo y otro clic para pausarlo y/o finalizarlo.

Proporciona también un panel de administración donde se pueden agrupar las contabilizaciones temporales en función de proyectos o de los diferentes registros en una misma actividad.

Antes de comenzar el proyecto se valoró la opción de utilizar Clockify como alternativa, sin embargo, debido al uso de Jira se optó por su descarte ya que Toggl presenta una mejor integración.

2.4.5 Google Drive

Servicio de almacenamiento propio de la empresa Google y gratuito en su plan estándar con una cuota inicial máxima de 15 GB para cada cuenta. Permite alojar ficheros de todo tipo.

En lo que a nuestro proyecto respecta se ha utilizado para gestión de artefactos, documentación y referencias de nuestro proyecto. Todo aquel elemento diferente al código fuente de la aplicación ha sido almacenado en esta herramienta.

De igual forma, para las actas realizadas en las reuniones entre autor y tutor fue utilizada la herramienta Google Docs (editor de texto) presente en el propio Google Drive para dar constancias de estas, y permitirle al tutor por otro lado la posibilidad de almacenar y crear tareas pendientes para la próxima reunión.

2.4.6 Office 365 (Word)

Para la elaboración de la memoria ha sido utilizado la herramienta Word (que pertenece al paquete Office 365) el cual es gratuito dentro del plan que facilita la Universidad de Sevilla a los miembros de su comunidad educativa.

La razón por la cual se utiliza esta herramienta en lugar de “Google Docs” es debido a la versatilidad que ofrece esta aplicación en todo lo que rodea al estilo y organización del documento. De hecho, antes del inicio de la realización de la memoria se configuró una plantilla para facilitar la labor de la redacción de esta.

2.4.7 Google Meet

El desarrollo de este proyecto se vio envuelto en la expansión de la pandemia del COVID-19 [31]. A causa de la situación a la que se enfrentó nuestra sociedad fue necesario el anuncio del estado de alarma en nuestro país, y consecuentemente se estableció un período de confinamiento que, en el caso de la Universidad de Sevilla significó la cancelación del curso académico 2019 – 2020 de forma presencial, sustituido por una enseñanza online.

Es por este motivo por el que se decidió, de manera consensuada entre tutor y autor, la sustitución de las reuniones presenciales que se realizaban hasta ese momento por reuniones online a través de la plataforma Google Meet, la cual fue propuesta por el profesor y supuso un acierto ya que, por comprobaciones con diferentes herramientas utilizadas en otras tareas, resultó ser la que mejor rendimiento mostraba en comparación de otras aplicaciones similares como *Discord*, *Jitsi Meet* o *Skype*, las cuales requieren mayor ancho de banda del cual no se disponía en ese momento.

3 PLANIFICACIÓN

En esta sección se comentará el proceso que se ha seguido para planificar el proyecto, la metodología utilizada para desarrollarlo y el desglose del rendimiento que se ha mantenido en cada iteración.

3.1 Estimación temporal del proyecto

El proyecto que se presenta en este documento es la solución propuesta como TFG por parte del autor, lo cual conlleva tener una carga de 12 créditos ECTS según el plan de estudios establecido por la Escuela Técnica Superior de Ingeniería (ETSII) a la que pertenece el autor.

En la Universidad de Sevilla, un crédito ECTS equivale a unas 25 horas de trabajo del estudiante, por lo que la carga total de este proyecto equivale a un total de 300 horas, que en esta ocasión serán repartidas en tres bloques fundamentales, tal y como se muestra en la siguiente tabla. En ella además se realiza una comparación con el ajuste real en horas que ha tenido en concreto este proyecto.

BLOQUE	HORAS ESTIMADAS	HORAS REALES	DESVIACIÓN
DESARROLLO	240	243	3
MEMORIA	40	51	11
PRESENTACIÓN	20	<i>Por determinar</i>	<i>Por determinar</i>

Tabla 1 - Estimación temporal

3.2 Metodología

La idea esencial de este proyecto en un inicio fue la de involucrar en todo el proceso de desarrollo a los usuarios de la aplicación de tal forma que se pudiesen pulir los fallos de diseño que podrían haberse producido. De igual forma se pretendió adaptar e introducir toda su experiencia y conocimiento en el desarrollo del proyecto.

Establecidos estos pilares fundamentales se decidió, de manera consensuada con el tutor, el uso de una metodología ágil. En concreto será utilizado SCRUM, ya que ha sido la metodología más utilizada por el autor a lo largo de su formación académica en la escuela.

Respecto a Scrum fue concebida con la idea de cumplir con los criterios establecidos por el *Manifiesto Ágil* [32], el cual establece estos cuatro puntos:

- “*Individuos e interacciones sobre procesos y herramientas.*”
- “*Software funcionando sobre documentación exhaustiva.*”
- “*Colaboración con el cliente sobre negociación contractual.*”
- “*Respuesta ante el cambio sobre seguir un plan.*”

La metodología ha sido adaptada al contexto de este proyecto. Se expresarán las características de Scrum enmarcadas en las modificaciones que han sido realizadas

dadas las peculiaridades en las que se ha desarrollado este y que serán expuestas a continuación.

Para la explicación de toda esta metodología se apoyará la explicación en la siguiente imagen.

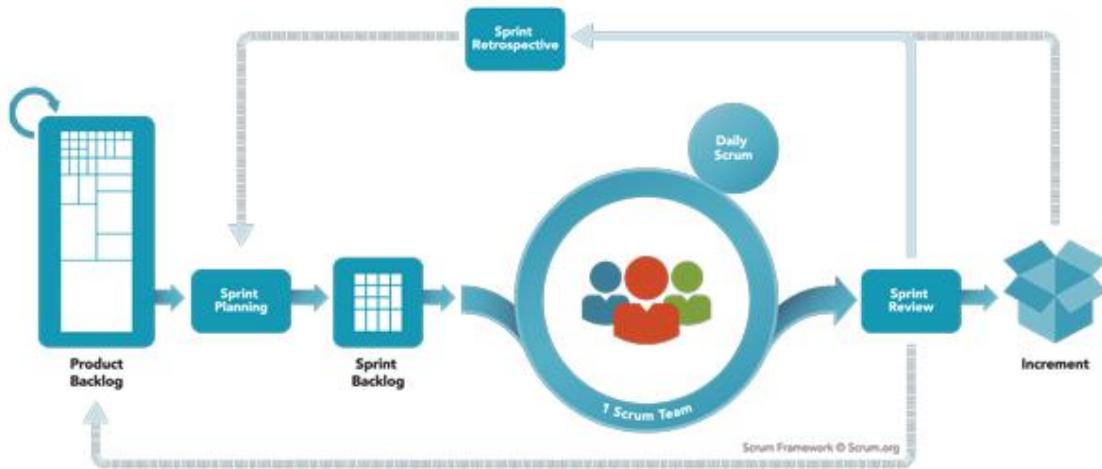


Ilustración 7 Framework Sprint [33]

En primer lugar, Scrum define tres roles principales los cuales son:

- ***Scrum Master***

Jefe de proyecto encargado de supervisar todo el trabajo realizado, cuyo objetivo es guiar a los distintos integrantes del desarrollo para la consecución de los objetivos. En el caso de este proyecto este rol es desempeñado por José Antonio Parejo Maestre.

- ***Product Owner***

Es el representante de la parte interesada. Es quién demanda los requisitos que se espera obtener y su función es la de vigilar y procurar que el equipo Scrum aporte valor al desarrollo que se está realizando, así como ser responsable de Product Backlog (pila de productos). En el caso de este proyecto este rol es desempeñado por José Antonio Parejo Maestre.

- ***Team***

Es el equipo de desarrolladores del proyecto incluyéndose en estos programadores, redactores de documentación, testers, analistas, etc. En este caso, al conformarse el equipo de trabajo por una sola persona este rol lo desempeña el autor, Álvaro de la Flor Bonilla.

Establecidos los tres roles principales que constituyen Scrum continuamos detallando cada una de las etapas y elementos que conforman esta metodología de desarrollo.

Scrum se basa en un proceso que es iterativo e incremental, basándose en un conjunto de sprints. Se entiende por sprint al pequeño proyecto cuya duración no puede extenderse más de un mes (normalmente entre dos y cuatro semanas) en el que el equipo desarrolla un incremento de valor respecto al producto final que se está construyendo, es decir, un software potencialmente entregable, funcional y que permite

ver al Product Owner el avance del proyecto. En concreto, para este proyecto se decidió que la duración de los sprints que se iban a llevar a cabo fuesen de 2 semanas.

Respecto al ciclo de vida que mantiene cada uno de estos sprints podemos señalar las siguientes etapas (que pueden apreciarse en la ilustración 7 expuesta anteriormente).

Antes de comenzar con la explicación de cada uno de los ciclos cabe puntualizar la definición de **Product Backlog** ya que será nombrado en repetidas ocasiones a lo largo del documento. *Product Backlog* representa toda la pila de tareas que se pretende realizar a lo largo de la vida del proyecto.

- **Sprint Planning**

Es la reunión del sprint que está próximo a comenzar, en la que participan Product Owner, Scrum Master y Team. Durante esta reunión el Product Owner decide los requisitos extraídos del Product Backlog que serán desarrollados durante el sprint que va a comenzar. El Team decide cuales son viables y descarta aquellos que no, conformando de esta forma el *Sprint Backlog*.

- **Ejecución del Sprint**

Una vez constituido el *Sprint Backlog* da inicio el comienzo del sprint.

Una de las modificaciones de la metodología Scrum que se ha realizado en este proyecto ha sido la eliminación del *Daily Scrum*. Esta acción consiste en una pequeña reunión diaria entre los distintos miembros del equipo de desarrollo para comentar el estado en el que se encuentra su actividad y los posibles problemas con los que se están encontrando. Al formarse en esta ocasión el equipo de trabajo por una única persona carece de lógica mantener esta acción y se ha optado por suprimirla.

- **Sprint Review**

Se conoce como *Sprint Review* a la reunión que ocurre al final de cada sprint. En esta reunión el equipo de desarrollo le realiza una demostración del producto actual al Product Owner de todos los requisitos que habían sido acordados en el *Sprint Backlog*. Es tarea de Product Owner realizar modificaciones en este momento sobre el *Product Backlog* en caso de que sea necesario.

Este evento representa una gran oportunidad para recibir un *feedback* sobre la situación real del desarrollo del producto.

Como resultado de esta reunión se obtienen dos hechos bastante importantes.

En primer lugar, se obtiene, como fue indicado anteriormente, un *Product Backlog* que ha sido revisado y organizado.

En segundo lugar, se confirma el incremento de valor de nuestro producto a partir del *feedback* que ofrece el cliente.

- **Sprint Retrospective**

Este es el último evento de una iteración en una metodología sprint antes de comenzar la siguiente. Consiste en que todo el equipo realice una reflexión tanto de sí mismo, como del desarrollo del equipo, intentando localizar posibles fallos que hayan afectado

al rendimiento individual, así como crear un plan de reacción ante tales problemas localizados.

El fin de este evento no es otro que tratar de localizar errores, solventarlos y lograr un mejor rendimiento para la siguiente interacción. Al tratarse en este lugar de un equipo compuesto de una única persona la tarea de autocrítica se dificultaba, sin embargo, el autor cree que la realización de una autorreflexión ha sido un punto positivo a lo largo de todo el desarrollo y le ha ayudado mejorar las estrategias planificación de tareas.

Debido a las características que conformaban este proyecto (conformado por un solo tutor y autor) tanto la *Sprint Planning* como la *Sprint Review* eran realizadas el mismo día y, como anteriormente fue indicado, los ciclos de cada sprint se repetían cada dos o tres semanas.

3.3 Descripción de las iteraciones

En esta sección serán explicadas cada una de las iteraciones que se han realizado a lo largo del proyecto cuyo núcleo fundamental se ha realizado en el período febrero - junio. Los sprints tuvieron una duración media de dos semanas, salvo algunas excepciones que fueron modificados a una.

Las reuniones Scrum fueron realizadas en el despacho del tutor, hasta el inicio del período de alarma en la que se modificó la estrategia seguida y se comenzaron a realizar reuniones en remoto a través de la plataforma Google Meet.

En total se planificaron 8 sprints, los cuales serán detallados uno a uno a continuación. En ellos se expondrán con exhaustividad los objetivos, las tareas planificadas, los resultados, el rendimiento de cada uno de ellos y los planes de mejora que se establecerían.

Antes de comenzar en análisis pormenorizado de cada sprint es importante señalar el siguiente dato.

Se estimó inicialmente que se dedicaría unas 240 horas de trabajo a tareas referentes al desarrollo del proyecto divididas en unos 8 sprints (15 semanas ya que el sprint inicial fue de una única semana), lo cual lleva a que, como ocurre en el caso de este proyecto, si se realiza un reparto de tareas equitativo a nivel de desempeño total entre las distintas iteraciones el tiempo máximo que puede emplearse en cada una de estas semanas es de 16 horas, es decir, 32 horas para un sprint estándar.

3.3.1 Sprint 1: Inicio del proyecto

Este sprint comenzó el día 17 de febrero de 2020 y finalizó el día 24 de febrero de 2020.

OBJETIVOS

En esta primera iteración el objetivo principal es conocer el alcance del proyecto, donde, debido a que se desconocía por completo el panorama actual y creyendo que posiblemente era la mejor opción a la que acudir, se concertó una cita con un miembro de la ONCE [2]. El autor de este proyecto fue atendido por la Jefa de Departamento de Servicios Sociales para Afiliados y le comentó cual era la verdadera situación actual, que problemas afectaban a sus afiliados y no estaban solucionados por ninguna tecnología.

Como meta, el autor se propuso intentar abarcar una casuística de casos en sus requisitos lo suficientemente completa y compacta para satisfacer los problemas que habían sido planteados. Además, como plan bastante optimista, se pretendió desarrollar una aplicación con interfaz básica como primer contacto con la tecnología.

Por otro lado, y hecho muy importante, se realizó en este sprint una primera familiarización con la tecnología ya que se desconocía por completo (a excepción del lenguaje a utilizar – Java) el entorno con el que se iba a trabajar.

TAREAS Y RESULTADOS

A diferencia de los sprints que se analizarán posteriormente, esta primera iteración tuvo un desarrollo corto, una semana. La causa se fundamenta en que fue un primer sprint de toma de contacto con la tecnología, así como las tareas propuestas en su Sprint Backlog dependía de una rápida aprobación del Product Owner para el buen funcionamiento del resto de los sprints.

Se propusieron cinco tareas las cuales quedan conformadas de la siguiente forma:

ID	NOMBRE	P.H	¿COMPLETADA?
#C10	Requisitos de cliente potencial	6	Sí
#C11	Crear plantilla para la memoria	3	Sí
#C20	Configurar repositorio y proyecto base (GitHub)	2	Sí
#C21	App con menú e interfaz básica	16	Sí
#C12	Mockups de la aplicación	4	NO

Tabla 2 - Sprint 1 Backlog

Como se aprecia en la tabla anterior todas las tareas fueron completadas a excepción de una.

Se consiguió, como se marcó en el objetivo del sprint, conseguir una información bastante detallada de la situación actual gracias a la fuente de calidad con la que se consiguió contactar.

También se elaboró una aplicación, aunque con un diseño muy básico pero que sirvió de gran ayuda para un primer contacto con Android.

RENDIMIENTO Y PLANES DE MEJORA

En el siguiente gráfico se muestra el rendimiento que se ha mantenido durante este sprint.

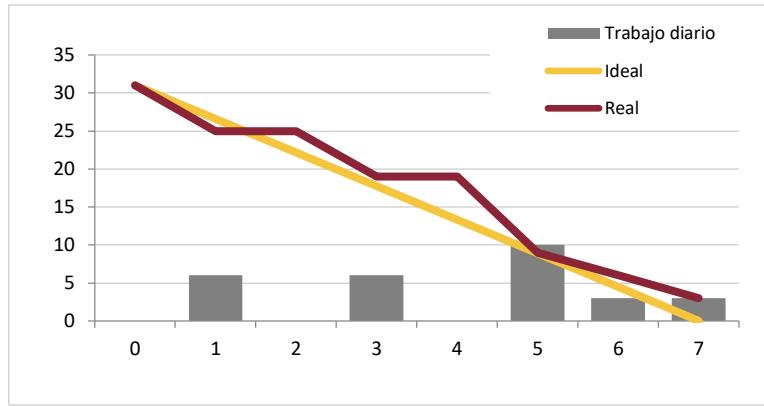


Ilustración 8 - Rendimiento Sprint 1

La línea amarilla representa el rendimiento ideal que se debería mantener, la cual, para este sprint constituido por 7 días naturales sería de 4.43 puntos de historia diarios (31 puntos de historia reales).

La línea roja representa el rendimiento real, la cual se completa con el gráfico inferior que indica los puntos de historia completados ese día.

Como puede observarse por una parte no se han completado todas las historias y por otra, el rendimiento a la hora de completarlas ha sido inferior al esperado (línea roja por encima de la línea amarilla).

Como causa podemos señalar que la adaptación a la tecnología fue mucho más dura de lo que el autor esperaba en un primer momento, la planificación inicial fue errónea y se debería haber asignado mucho más tiempo al estudio y conocimiento de esta.

4	● Clarify TFG	20:58:08
	App con menu e interfaz básica	12:53:59
	Crear plantilla para la memoria	1:12:43
	Mockups de la aplicación	3:34:26
	Requisitos de cliente potencial	3:17:00

Ilustración 9 - Rendimiento temporal Sprint 1

Respecto al análisis del registro temporal, según lo estimado, para un sprint de una semana el gasto máximo debería haber sido de unas 16 horas frente a las casi 21 que finalmente se consumieron (31,25% más de los esperado).

Claramente se identifica que la actividad de creación de la App básica supuso un bache en el desarrollo, justificado en la opinión del autor, por intentar comenzar a partir de una formación nula en la plataforma que se estaba utilizando.

Como planes de mejora se optó por plantear actividades más sencillas de abordar y que colaboraran en la familiarización con la tecnología. Para ello se estableció la regla de que ninguna tarea pudiese suponer más del 30% del total del gasto temporal de ese sprint.

3.3.2 Sprint 2: Primera funcionalidad

Este sprint comenzó el día 24 de febrero de 2020 y finalizó el día 9 de marzo de 2020.

OBJETIVOS

El plan de este sprint fue ambicioso desde su comienzo. Entre los puntos fundamentales constaban la investigación e implementación de funcionalidad con Firebase, tanto para el inicio de sesión como para el almacenamiento y consulta de datos. Además de lo anterior se añadió toda la funcionalidad relativa a la identificación de productos a través de NFC y su lectura mediante un asistente de voz.

Como meta se estableció que, a la finalización de este sprint, el usuario debería poder realizar “login” en la aplicación y el sistema ser capaz de almacenar de manera persistente información como datos del usuario, etiquetas públicas y registros de productos almacenados de forma privada tanto en local como en remoto.

Finalmente, además de lo anterior se debería poder leer una etiqueta NFC, mostrar información básica y ser detallada a través de un asistente de voz.

TAREAS Y RESULTADOS

ID	NOMBRE	P.H	¿COMPLETADA?
#C08	Lectura etiquetas NFC	17	SÍ
#C11	Registro de usuarios	6	SÍ*
#C20	Mockups de la aplicación	4	SÍ
#C21	Guardado de productos en almacén personal	14	NO
#C12	Documentar proyecto (Sprint 1 y 2)	5	NO
#C29	Inicio de sesión con Firebase	12	SÍ*
#C30	API de productos con Firebase	9	SÍ*
#C13	Asistente de voz	13	NO

Tabla 3 - Sprint 2 Backlog

Las tareas marcadas con “*” indican que se marcaron como finalizadas pero que, sin embargo, tras su análisis en las pruebas fueron detectados errores por parte del equipo de clientes pilotos los cuales serán detallados en el apartado de pruebas.

Tras el fracaso a nivel de resultados se decidió abortar el sprint y comenzar una nueva iteración con pleno conocimiento de la verdadera dificultad a la que se enfrentaba el proyecto en ese momento.

RENDIMIENTO Y PLANES DE MEJORA

En el siguiente gráfico se muestra el rendimiento que se mantuvo durante este sprint.

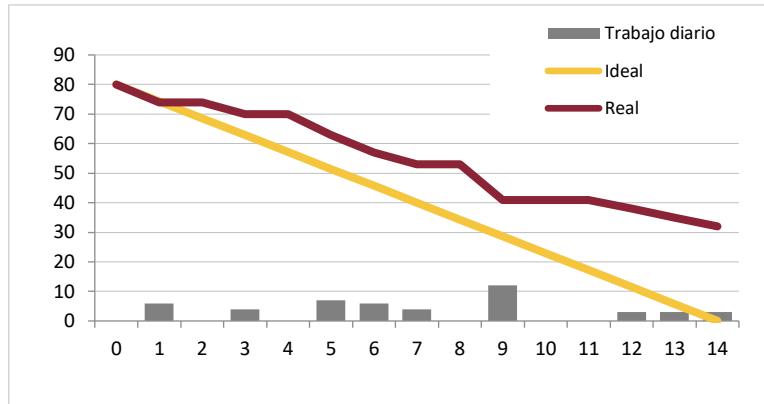


Ilustración 10 - Rendimiento Sprint 2

La línea amarilla representa el rendimiento ideal que se debería mantener, la cual, para este sprint constituido por 14 días naturales sería de 5.71 puntos de historia diarios (80 puntos de historia totales).

La línea roja representa el rendimiento real que se mantuvo en ese sprint. Por otro lado, el gráfico inferior representa los puntos de historia completados ese día.

En todo momento, el rendimiento real se mantuvo por encima del rendimiento ideal, causando que al momento de finalización del sprint quedaron por completar 32 puntos de historia de los 80 totales (40% del total sin completar). El rendimiento real de este sprint fue de 3.43 historias/día frente a las 5.71 historias/días que realmente se debería haber mantenido.

5	● Clarify TFG	30:36:32
	API de productos con Firebase	5:34:51
	Inicio de sesión con Firebase	2:20:06
	Lectura etiquetas NFC	15:13:18
	Mockups de la aplicación	6:14:10
	Registro de usuarios con Firebase	1:14:07

Ilustración 11 - Rendimiento temporal Sprint 2

Respecto a las medidas de tiempo empleado para este sprint, debido a que su duración fue de unas 2 semanas, se habrían de haber empleado según lo estimado unas 32 horas. Como puede observarse en la captura, se han empleado 30 horas y media lo que representa un 6.25% menos del desempeño total que se debería haber realizado.

Las restricciones que se habían tomado en el anterior sprint de que una actividad no supusiese más del 30% del total de horas del sprint, no se han cumplido.

Por otro lado, a pesar de que no se cumplieron las mejoras establecidas en el sprint anterior (y por tanto no puede ser analizado el éxito de esta), se realizó un

nuevo análisis, y en suma al problema de la sobreestimación de conocimientos se añadió el exceso de tamaño con el que se configuraban las tareas.

Como solución a ambos problemas se estableció la necesidad de atomizarlas, para evitar bloqueos en determinados apartados y facilitar la finalización de puntos de historia.

Como medida para comprobar si el plan de reacción funcionó se estableció que para el próximo sprint el rendimiento final finalizaría con una velocidad de 0.6 (puntos de historia) / día mejor a la ideal.

3.3.3 Sprint 3: Primera aplicación funcional

Este sprint comenzó el día 9 de marzo de 2020 y finalizó el día 23 de marzo de 2020.

OBJETIVOS

Durante el desarrollo de este sprint el autor intentó que en suma a las tareas que no fueron completadas durante el sprint anterior se añadieran el número acorde de nuevas, de tal forma que a finalización de este sprint el producto hubiese alcanzado un valor suficiente para comenzar a trabajar con los clientes pilotos.

Por tanto, como meta para este sprint se establece que se debe tener una aplicación que permita al usuario registrarse e iniciar sesión utilizando la pasarela de Google. En añadido, se debe tener una primera versión de la identificación de productos donde el usuario pueda conocer los datos de un producto acercando su teléfono a la etiqueta NFC. Este producto se podrá almacenar en su almacén privado y ser accesible y visible desde otro dispositivo si se inicia sesión con la misma cuenta en tiempo real.

Es importante destacar que el día 15 de marzo se decretó en España el estado de alarma y el consecuente confinamiento de la población, lo que supuso importantes cambios en el proyecto que se verán reflejados en el siguiente sprint.

Como consecuencia a lo anterior, las actividades de pilotajes tuvieron que ser canceladas por las dificultades que mostraban para el cliente con el que se había contactado instalar y usar esta aplicación sin ayuda presencial.

TAREAS Y RESULTADOS

ID	NOMBRE	P.H	¿COMPLETADA?
#C31	Registrarse con la cuenta de Google	6	SÍ
#C32	Iniciar sesión con la cuenta de Google	6	SÍ
#C34	Leer datos de productos escaneados (Firebase)	11	SÍ
#C33	Crear interfaz atractiva (detección de productos)	14	SÍ
#C23	Guardar producto escaneado (Firebase)	15	SÍ
#C35	Guardar producto escaneado (Realm)	15	SÍ
#C36	Rutina de sincronización (Firebase – Realm)	17	SÍ

Tabla 4 - Sprint 3 Backlog

Respecto al extracto de tareas anterior hay que recalcar que, aunque según las actividades “#C23” y “#C35” confirman el almacenamiento de un producto en el gestor personal del usuario es cierto que se realizaba, sin embargo, no era transparente al usuario ya que su interfaz gráfica aún no se había completado.

Además, en esta iteración se completó la tarea del sprint anterior que no había sido completada y que conllevaba consolidar el JSON utilizado para almacenar toda la información privada referente al usuario (**#C30 – API de productos Firebase**) ya que *compartían el mismo núcleo necesario para completar su desarrollo*.

RENDIMIENTO Y PLANES DE MEJORA

En el siguiente gráfico se muestra el rendimiento que se mantuvo durante este sprint.

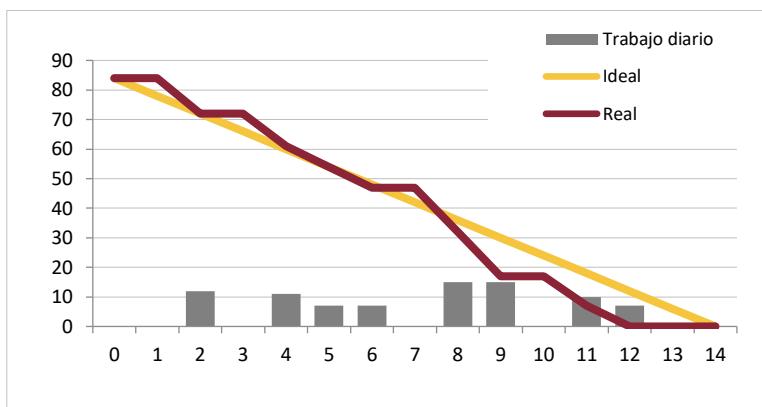


Ilustración 12 - Rendimiento Sprint 3

La línea amarilla representa el rendimiento ideal que se debería mantener, la cual, para este sprint constituido por 14 días naturales sería de 6 puntos de historia diarios (84 puntos de historia totales). La línea roja representa el rendimiento real que se mantuvo en sprint.

Por otro lado, el gráfico inferior representa los puntos de historia completados ese día.

Como objetivo en el sprint anterior se planteó establecer que el rendimiento fuera mejor en seis décimas al ideal.

En este sprint de 14 días y 84 puntos de historia el rendimiento ideal era de 6 (puntos de historia) / día. Finalmente, el proyecto se finalizó con dos días de antelación a lo esperado, lo que supuso que el rendimiento final fuese de 7 (puntos de historia) / día.

7	● Clarify TFG	30:50:25
	Crear interfaz atractiva (detección de productos)	2:20:56
	Guardar producto escaneado (Firebase)	6:07:37
	Guardar producto escaneado (Realm)	7:19:41
	Iniciar sesión con la cuenta de Google	1:36:07
	Leer datos de productos escaneados (Firebase)	7:51:35
	Registrarse con la cuenta de Google	1:22:07
	Rutina de sincronización (Firebase – Realm)	4:12:22

Ilustración 13 - Rendimiento temporal Sprint 3

En cuanto a desviaciones en tiempo, de las 32 horas que le correspondían a este sprint se consumieron casi 31, lo que supuso un decremento del 3.66% respecto a lo estimado inicialmente.

No se propuso ninguna fórmula de mejora para el siguiente sprint, simplemente mantener el rendimiento actual.

3.3.4 Sprint 4: Almacén de productos y pivotación de la aplicación

Este sprint comenzó el día 23 de marzo de 2020 y finalizó el día 6 de abril de 2020.

OBJETIVOS

Como se indicó en el comienzo del sprint anterior, debido a la situación del COVID-19, el plan que se había establecido para la aplicación resultó muy afectado, ya que por la casuística de la aplicación era muy importante la recomendación y el guiado durante el proceso de desarrollo de los clientes pilotos.

Para las primeras versiones de la aplicación, los usuarios pilotos indicaron que les era muy complicado la instalación del software necesario sin ayuda personal por lo que se cancelaría por el momento su participación en el proyecto.

Como respuesta ante esta situación, debido a que se optó por la utilización de una metodología ágil la cual facilita las modificaciones en función de las condiciones del proyecto, el autor decidió potenciar una funcionalidad que en un principio se postulaba a ser secundaria pero que sin embargo a partir de este momento pasaría a formar un importante núcleo en la aplicación.

Esta funcionalidad es la lista de la compra personal y compartida que será detallada con mayor exhaustividad en la sección de análisis y se comenzaría su desarrollo en el siguiente sprint.

Ante la necesidad contactar con nuevos usuarios pilotos, el autor decidió acudir a personas que vivían en su misma residencia, así podría realizar las revisiones de manera presencial. De igual forma se contactó con dos personas más (con conocimiento de la plataforma Android) que colaborarían de manera auxiliar.

TAREAS Y RESULTADOS

ID	NOMBRE	P.H	¿COMPLETADA?
#C42	Listado de productos (tiempo real)	18	Sí*
#C43	Paginación de productos	8	Sí*
#C24	Ver productos del almacén	9	Sí
#C37	Eliminar productos del almacén (Realm)	12	Sí
#C38	Eliminar productos del almacén (Firebase)	12	Sí
#C39	Eliminar almacén (Realm)	6	Sí
#C40	Eliminar almacén (Firebase)	6	Sí
#C41	Rutina de sincronización Realm-Firebase	18	NO

Tabla 5 - Sprint 4 Backlog

Las tareas marcadas como finalizadas y con la adición del símbolo “*” indican que el autor las marcó como finalizadas pero que, sin embargo, tras su análisis en las pruebas fueron detectados errores por parte del equipo de clientes pilotos los cuales serán detallados en el apartado de pruebas.

RENDIMIENTO Y PLANES DE MEJORA

En el siguiente gráfico se muestra el rendimiento que se mantuvo durante este sprint.

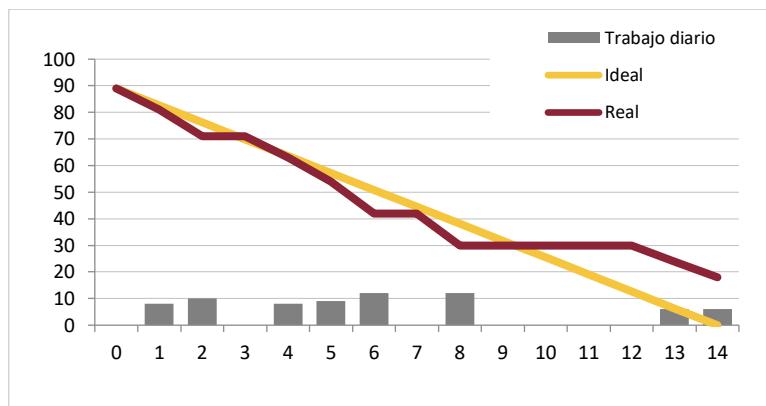


Ilustración 14 - Rendimiento Sprint 4

La línea amarilla representa el rendimiento ideal que se debería mantener, la cual, para este sprint constituido por 14 días naturales sería de 6,36 puntos de historia diarios (89 puntos de historia totales). La línea roja representa el rendimiento real que se mantuvo en sprint. Por otro lado, el gráfico inferior representa los puntos de historia completados ese día.

Con respecto a este sprint puede apreciarse que se mantuvo un rendimiento bastante semejante al ideal, sin embargo, apareció un parón al momento de comenzar a trabajar con la tarea #C41.

Para evitar un colapso total de este sprint el usuario decidió abortar esa tarea y continuar con las restantes que finalmente fueron finalizadas.

8	● Clarify TFG	37:06:22
Eliminar almacén (Firebase)	0:59:42	
Eliminar almacén (Realm)	1:15:17	
Eliminar productos del almacén (Firebase)	3:42:49	
Eliminar productos del almacén (Realm)	5:42:25	
Listado de productos (tiempo real)	8:10:18	
Paginación de productos	1:31:03	
Rutina de sincronización Realm-Firebase	12:54:29	
Ver productos del almacén	2:50:19	

Ilustración 15 - Rendimiento Temporal Sprint 4

Respecto a este sprint su desviación frente a la estimación ha sido de 5 horas, lo cual representa un 15.63% más de los esperado inicialmente.

Como dato que también puede extrapolarse de la gráfica de rendimiento puede ser la diferencia temporal consumida por la tarea en la que se produjo el bloqueo (#C41 – *Rutina de sincronización Realm-Firebase*) en relación con el resto de las actividades de este sprint.

En la retrospectiva realizada el autor señaló que un posible error que se había podido producir durante el sprint habría sido el intentar afrontar esta tarea sin ningún tipo de formación afianzada sobre la comunicación entre actividades. Además, se confió en que se afrontaría sin problemas, abandonando todo su desarrollo hacia el final del sprint.

Para evitar que este problema se convirtiera en recurrente (ya que sería una constante en nuestro desarrollo) se estableció como obligatoria una tarea de formación incluida en el desarrollo del próximo sprint.

3.3.5 Sprint 5: Lista de la compra funcional

Este sprint comenzó el día 6 de abril de 2020 y finalizó el día 20 de abril de 2020.

OBJETIVOS

Tal y como se indicó en los objetivos del sprint anterior, en esta iteración se llevará a cabo la implementación de gran parte de la nueva funcionalidad propuesta ante la situación excepcional del momento.

Como meta para este sprint (además de solventar los errores de funcionalidad anterior) es la de posibilitar al usuario que pueda crear su propia lista de la compra. Podrá añadir y eliminar productos en su aplicación de manera local y con copia de seguridad remota. Además, también podrá marcar un producto como seleccionado en la vista de listado.

Todas estas actualizaciones tendrán la característica de actualizarse a nivel de tiempo real entre los distintos dispositivos que mantenga una misma cuenta abierta de la aplicación en ellos.

TAREAS Y RESULTADOS

ID	NOMBRE	P.H	¿COMPLETADA?
#C44	Formación (comunicación – actualización entre actividades)	8	Sí
#C45	Bug actualización listados	12	Sí
#C46	Creación de lista personal de la compra	6	Sí
#C47	Listado de lista de la compra	9	Sí
#C48	Añadir producto a lista de la compra (Firebase)	15	Sí
#C49	Añadir producto a lista de la compra (Realm)	15	Sí
#C50	Eliminar producto de lista de la compra (Firebase)	12	Sí
#C51	Eliminar producto de lista de la compra (Realm)	12	Sí
#C53	Marcar producto como seleccionado (Firebase)	12	Sí
#C54	Marcar producto como seleccionado (Realm)	12	Sí
#C52	Rutina de sincronización Realm-Firebase	18	Sí

Tabla 6 - Sprint 5 Backlog

Todas las tareas de este sprint fueron completadas tal y como se establecieron inicialmente.

RENDIMIENTO Y PLANES DE MEJORA

En el siguiente gráfico se muestra el rendimiento que se mantuvo durante este sprint.

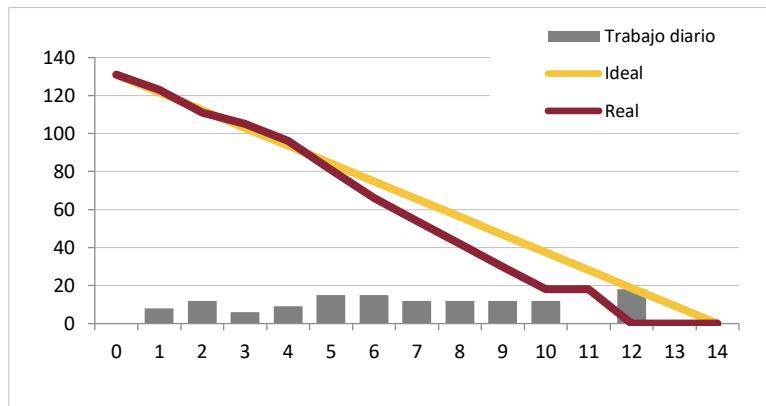


Ilustración 16 - Rendimiento Sprint 5

La línea amarilla representa el rendimiento ideal que se debería mantener, la cual, para este sprint constituido por 14 días naturales sería de 9,36 puntos de historia diarios (131 puntos de historia totales). La línea roja representa el rendimiento real que se mantuvo en sprint. Por otro lado, el gráfico inferior representa los puntos de historia completados ese día.

Respecto a este sprint es importante destacar que se consiguió mantener una velocidad bastante cercana a la idea en cuanto a la quema de historias de usuarios. La atomización de tareas parece que ha tenido un muy buen funcionamiento en este proyecto tras los resultados obtenidos.

11 ● Clarify TFG	38:23:09
Añadir producto a lista de la compra (Firebase)	6:42:12
Añadir producto a lista de la compra (Realm)	6:52:41
Bug actualización listados	2:19:11
Creación de lista personal de la compra	0:57:51
Eliminar producto de lista de la compra (Firebase)	5:30:12
Eliminar producto de lista de la compra (Realm)	5:44:19
Formación (comunicación – actualización entre actividades)	2:40:02
Listado de lista de la compra	0:41:09
Marcar producto como seleccionado (Firebase)	1:32:11
Marcar producto como seleccionado (Realm)	1:14:34
Rutina de sincronización Realm-Firebase	4:08:47

Ilustración 17 - Rendimiento Temporal Sprint 5

Pese a la cantidad de tareas que conformaban este sprint (con la correspondiente suma de puntos de historia) la desviación total ha sido de 6 horas y 20 minutos, es decir, un 19.7% de la estimación inicial temporal.

No se plantearon hacer reajustes a nivel temporal de la dinámica de consumo de tiempo que se están manteniendo en los últimos sprints ya que, si se consolidaba este ritmo, el proyecto podría finalizar (a nivel de desarrollo) antes, pudiendo dedicar más tiempo a otras secciones igual de importantes que la tecnológica como es la elaboración de la memoria.

3.3.6 Sprint 6: Funcionalidad compartida

Este sprint comenzó el día 20 de abril de 2020 y finalizó el día 11 de mayo de 2020.

OBJETIVOS

Como meta para este sprint se plantea finalizar toda la funcionalidad de la lista de la compra compartida entre usuarios.

El usuario debería tener la posibilidad de enviar invitaciones para acceder a su lista (y eliminar estas invitaciones también). Por otro lado, el usuario que recibe la invitación la aceptará o rechazará.

En la finalización de este sprint cualquier usuario podrá invitar a cualquier otro que se encuentre registrado en la aplicación. Una vez esta invitación este aceptada, ambos

usuarios compartirán una lista de la compra completa, actualizada en tiempo real entre todos los usuarios que posean acceso (inserción de productos, borrado de ellos, etc.).

TAREAS Y RESULTADOS

ID	NOMBRE	P.H	¿COMPLETADA?
#C58	Añadir invitación a usuario (Realm)	18	Sí*
#C56	Añadir invitación a usuario (Firebase)	12	Sí*
#C57	Eliminar invitación a usuario (Realm)	12	Sí*
#C59	Eliminar invitación a usuario (Firebase)	12	Sí*
#C60	Listar invitaciones a listas de la compra	12	Sí
#C61	Aceptar invitación a lista de la compra (Firebase)	18	Sí
#C62	Rechazar invitación a lista de la compra (Firebase)	18	Sí
#C63	Completar pantalla inicial	12	Sí
#C64	Añadir producto en lista de amigo	12	Sí
#C65	Borrar producto de lista de amigo	12	Sí
#C66	Marcar como seleccionado producto de lista de amigo	8	Sí

Tabla 7 - Sprint 6 Backlog

Las tareas marcadas como finalizadas y con la adición del símbolo “*” indican que el autor las marcó como finalizadas pero que, sin embargo, tras su análisis en las pruebas fueron detectados errores por parte del equipo de clientes pilotos los cuales serán detallados en el apartado de pruebas.

Esta iteración cuenta con mucha más dificultad que todas las anteriores. Debido a las nuevas circunstancias, el autor del proyecto disponía de mayor disponibilidad para el proyecto, por lo que decidió centrar esfuerzo y poder dar un gran impulso al desarrollo.

RENDIMIENTO Y PLANES DE MEJORA

En el siguiente gráfico se muestra el rendimiento que se mantuvo durante este sprint.

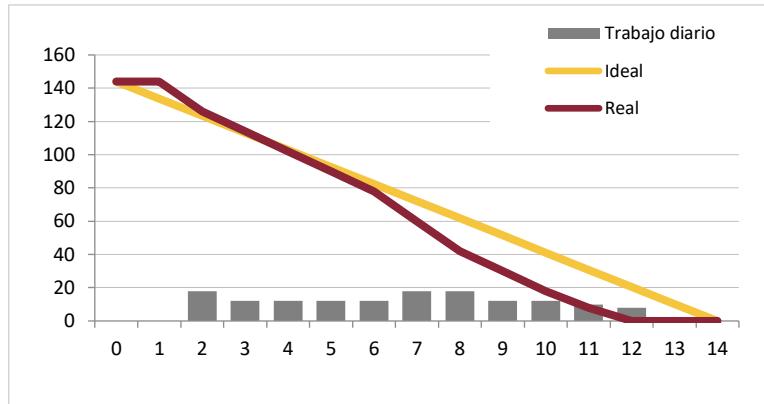


Ilustración 18 - Rendimiento Sprint 6

La línea amarilla representa el rendimiento ideal que se debería mantener, la cual, para este sprint constituido por 14 días naturales sería de 10.29 puntos de historia diarios (144 puntos de historia totales). La línea roja representa el rendimiento real que se mantuvo en sprint. Por otro lado, el gráfico inferior representa los puntos de historia completados ese día. El rendimiento final mostrado fue de 12 puntos de historia al día.

11	● Clarify TFG	39:52:04
	Aceptar invitación a lista de la compra (Firebase)	7:00:49
	Añadir invitación a usuario (Firebase)	6:14:31
	Añadir invitación a usuario (Realm)	6:16:18
	Añadir producto en lista de amigo	1:41:17
	Borrar producto de lista de amigo	1:27:18
	Completar pantalla inicial	0:49:06
	Eliminar invitación a usuario (Firebase)	2:02:02
	Eliminar invitación a usuario (Realm)	3:56:43
	Listar invitaciones a listas de la compra	2:19:31
	Marcar como seleccionado producto de lista de amigo	0:42:53
	Rechazar invitación a lista de la compra (Firebase)	7:21:36

Ilustración 19 - Rendimiento Temporal Sprint 6

Como se refleja en la ilustración anterior este sprint ha sido el que más desviación temporal ha tenido respecto al resto, pero a su vez el que más valor ha aportado al producto. En concreto la desviación de 7 horas corresponde a un 15.6% respecto al ideal estimado.

Durante esta iteración se consiguió mantener el rendimiento más cercano al ideal de todo el proyecto. Casi la totalidad del desarrollo de la iteración se mantuvo por debajo de la línea ideal, salvo por una leve descompensación al inicio del sprint.

3.3.7 Sprint 7: Accesibilidad de la aplicación

Este sprint comenzó el día 11 de mayo de 2020 y finalizó el día 25 de mayo de 2020.

OBJETIVOS

Para este sprint se espera que la aplicación sea totalmente accesible para personas con discapacidad visual, para ello se tendrá que facilitar una compatibilidad completa con el software Google TalkBack.

Además, como meta para este sprint la aplicación deberá permitir manejar toda la interfaz de usuario mediante comandos de voz.

TAREAS Y RESULTADOS

ID	NOMBRE	P.H	¿COMPLETADA?
#C67	Adaptación a Google TalkBack para listados	12	Sí
#C68	Adaptación a Google TalkBack para botones	14	Sí
#C69	Adaptación a Google TalkBack para textos	10	Sí
#C70	Comandos de voz en vistas generales	12	Sí
#C71	Comandos de voz en listados de compra	16	Sí*
#C72	Comandos de voz en procesos de identificación	12	Sí

Tabla 8 - Sprint 7 Backlog

Las tareas marcadas como finalizadas y con la adición del símbolo “*” indican que el autor las marcó como finalizadas pero que, sin embargo, tras su análisis en las pruebas fueron detectados errores por parte del equipo de clientes pilotos los cuales serán detallados en el apartado de pruebas.

Todas las tareas de este sprint fueron completadas tal y como se establecieron inicialmente.

RENDIMIENTO Y PLANES DE MEJORA.

En el siguiente gráfico se muestra el rendimiento que se mantuvo para este sprint.

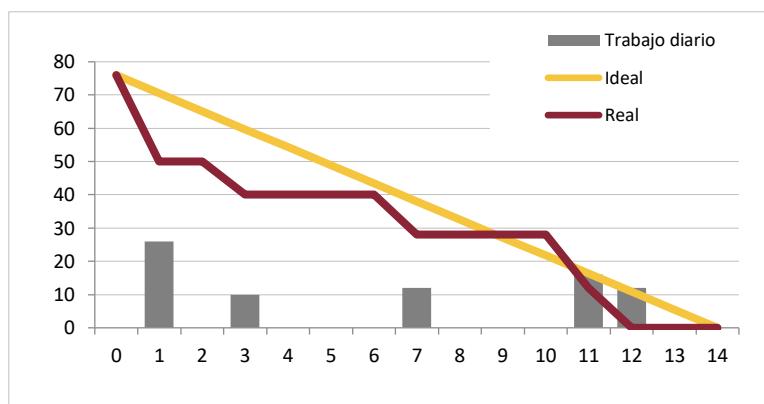


Ilustración 20 – Rendimiento Sprint 7

La línea amarilla representa el rendimiento ideal que se debería mantener, la cual, para este sprint constituido por 14 días naturales sería de 5.43 puntos de historia diarios (76 puntos de historia totales). La línea roja representa el rendimiento real que se mantuvo en sprint. Por otro lado, el gráfico inferior representa los puntos de historia completados ese día. El rendimiento final mostrado fue de 6.33 puntos de historia al día.

6	● Clarify TFG	23:35:51
Adaptación a Google TalkBack para botones	3:22:08	
Adaptación a Google TalkBack para listados	3:48:12	
Adaptación a Google TalkBack para textos	4:06:26	
Comandos de voz en listados de compra	5:12:21	
Comandos de voz en procesos de identificación	2:58:01	
Comandos de voz en vistas generales	4:08:43	

Ilustración 21 – Rendimiento temporal Sprint 7

Como se refleja en la ilustración anterior, la desviación temporal ha sido de unas 8,5 horas en positivo, es decir se han consumido 8.5 horas menos de los que se esperaba para un sprint de dos semanas (26,56% menos de horas de los esperado).

La razón de esta desviación se debe a que en sprints anteriores se ha adelantado mucha tarea.

3.3.8 Sprint 8: Corrección de errores

Este sprint comenzó el día 25 de mayo de 2020 y finalizó el día 8 de junio de 2020.

OBJETIVOS

Como objetivo para este sprint se espera localizar y solventar cualquier tipo de error que pueda existir en la aplicación, así como obtener datos reales de productos mediante un proceso de scrapping realizado a una web de comercio electrónico.

TAREAS Y RESULTADOS

ID	NOMBRE	P.H	¿COMPLETADA?
#C73	Test – Identificar producto	9	Sí
#C74	Test – Lista de la compra	18	Sí
#C75	Test – Lista de la compra (compartida)	19	Sí
#C76	Test – Notificaciones	9	Sí
#C77	Test – Comandos de voz	12	Sí
#C78	Test – Uso de TalkBack	9	Sí
#C79	Scraping a la web de Mercadona	10	Sí

Todas las tareas de este sprint fueron completadas tal y como se establecieron inicialmente.

RENDIMIENTO Y PLANES DE MEJORA

En el siguiente gráfico se muestra el rendimiento que se mantuvo para este sprint.

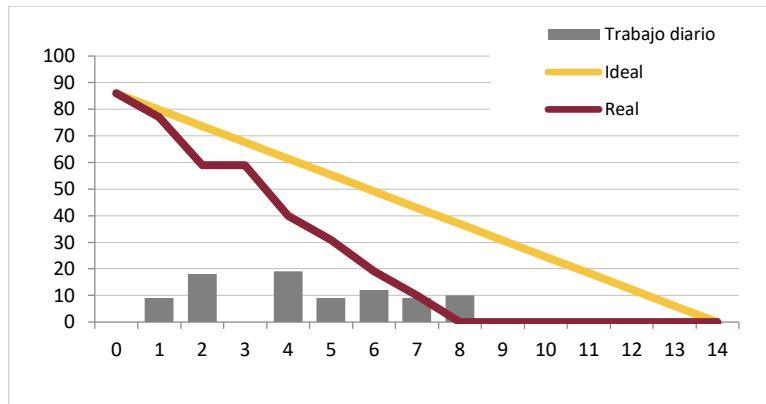


Ilustración 22 – Rendimiento Sprint 8

La línea amarilla representa el rendimiento ideal que se debería mantener, la cual, para este sprint constituido por 14 días naturales sería de 6.14 puntos de historia diarios (86 puntos de historia totales). La línea roja representa el rendimiento real que se mantuvo en sprint. Por otro lado, el gráfico inferior representa los puntos de historia completados ese día. El rendimiento final mostrado fue de 12.29 puntos de historia al día.

7	● Clarify TFG	21:50:05
	Scraping a la web de Mercadona	3:49:41
	Test – Comandos de voz	2:49:06
	Test – Identificar producto	3:17:36
	Test – Lista de la compra	3:25:13
	Test – Lista de la compra (compartida)	3:11:41
	Test – Notificaciones	2:46:32
	Test – Uso de TalkBack	2:30:16

Ilustración 23 – Rendimiento temporal Sprint 8

Como se refleja en la ilustración anterior, la desviación temporal ha sido de unas 10 horas en positivo, es decir se han consumido 10 horas menos de los que se esperaba para un sprint de dos semanas (31,25% menos de horas de los esperado).

La razón de esta desviación se debe a la misma causa que en anteriores iteraciones, se conocía por completo la tecnología y se había adelantado mucha tarea anteriormente.

Esta iteración fue la última. Todas las tareas del Product Backlog fueron completadas a tiempo y aceptadas como válidas cada una de ellas.

3.4 Backlog del proyecto

ID	NOMBRE	P.H	SPRINT
#C10	Requisitos de cliente potencial	6	1
#C11	Crear plantilla para la memoria	3	1
#C20	Configurar repositorio y proyecto base (GitHub)	2	1
#C21	App con menú e interfaz básica	16	1
#C12	Mockups de la aplicación	4	1
#C08	Lectura etiquetas NFC	17	2
#C11	Registro de usuarios	6	2
#C20	Mockups de la aplicación	4	2
#C21	Guardado de productos en almacén personal	14	2
#C12	Documentar proyecto (Sprint 1 y 2)	5	2
#C29	Inicio de sesión con Firebase	12	2
#C30	API de productos con Firebase	9	2
#C13	Asistente de voz	13	2
#C31	Registrarse con la cuenta de Google	6	3
#C32	Iniciar sesión con la cuenta de Google	6	3
#C34	Leer datos de productos escaneados (Firebase)	11	3
#C33	Crear interfaz atractiva (detección de productos)	14	3
#C23	Guardar producto escaneado (Firebase)	15	3
#C35	Guardar producto escaneado (Realm)	15	3
#C36	Rutina de sincronización (Firebase – Realm)	17	3
#C42	Listado de productos (tiempo real)	18	4
#C43	Paginación de productos	8	4
#C24	Ver productos del almacén	9	4
#C37	Eliminar productos del almacén (Realm)	12	4
#C38	Eliminar productos del almacén (Firebase)	12	4
#C39	Eliminar almacén (Realm)	6	4
#C40	Eliminar almacén (Firebase)	6	4
#C41	Rutina de sincronización Realm-Firebase	18	4

#C44	Formación (comunicación – actualización entre actividades)	8	5
#C45	Bug actualización listados	12	5
#C46	Creación de lista personal de la compra	6	5
#C47	Listado de lista de la compra	9	5
#C48	Añadir producto a lista de la compra (Firebase)	15	5
#C49	Añadir producto a lista de la compra (Realm)	15	5
#C50	Eliminar producto de lista de la compra (Firebase)	12	5
#C51	Eliminar producto de lista de la compra (Realm)	12	5
#C53	Marcar producto como seleccionado (Firebase)	12	5
#C54	Marcar producto como seleccionado (Realm)	12	5
#C52	Rutina de sincronización Realm-Firebase	18	5
#C58	Añadir invitación a usuario (Realm)	18	6
#C56	Añadir invitación a usuario (Firebase)	12	6
#C57	Eliminar invitación a usuario (Realm)	12	6
#C59	Eliminar invitación a usuario (Firebase)	12	6
#C60	Listar invitaciones a listas de la compra	12	6
#C61	Aceptar invitación a lista de la compra (Firebase)	18	6
#C62	Rechazar invitación a lista de la compra (Firebase)	18	6
#C63	Completar pantalla inicial	12	6
#C64	Añadir producto en lista de amigo	12	6
#C65	Borrar producto de lista de amigo	12	6
#C66	Marcar como seleccionado producto de lista de amigo	8	6
#C67	Adaptación a Google TalkBack para listados	12	7
#C68	Adaptación a Google TalkBack para botones	14	7
#C69	Adaptación a Google TalkBack para textos	10	7
#C70	Comandos de voz en vistas generales	12	7
#C71	Comandos de voz en listados de compra	16	7
#C72	Comandos de voz en procesos de identificación	12	7

#C73	Test – Identificar producto	9	8
#C74	Test – Lista de la compra	18	8
#C75	Test – Lista de la compra (compartida)	19	8
#C76	Test – Notificaciones	9	8
#C77	Test – Comandos de voz	12	8
#C78	Test – Uso de TalkBack	9	8
#C79	Scrapping a la web de Mercadona	10	8

3.5 Burn-up global del proyecto

En la siguiente tabla se muestra el rendimiento general del proyecto, en forma de “Burn-up” utilizando los puntos de historia completados en cada sprint.

La línea amarilla indica el número ideal de puntos de historia que se deberían haber completado en esa iteración (teniendo en cuenta que por cada iteración se acumulan los puntos de historia completados de las anteriores iteraciones) en el caso de que se hubiese seguido una distribución lineal (90,1 puntos de historia por cada sprint).

La línea roja representa el rendimiento real que se mantuvo a lo largo de las distintas iteraciones.

Las barras grises indican el número de puntos de historias completados en cada sprint. En total, en este proyecto, se realizaron 721 puntos de historia.

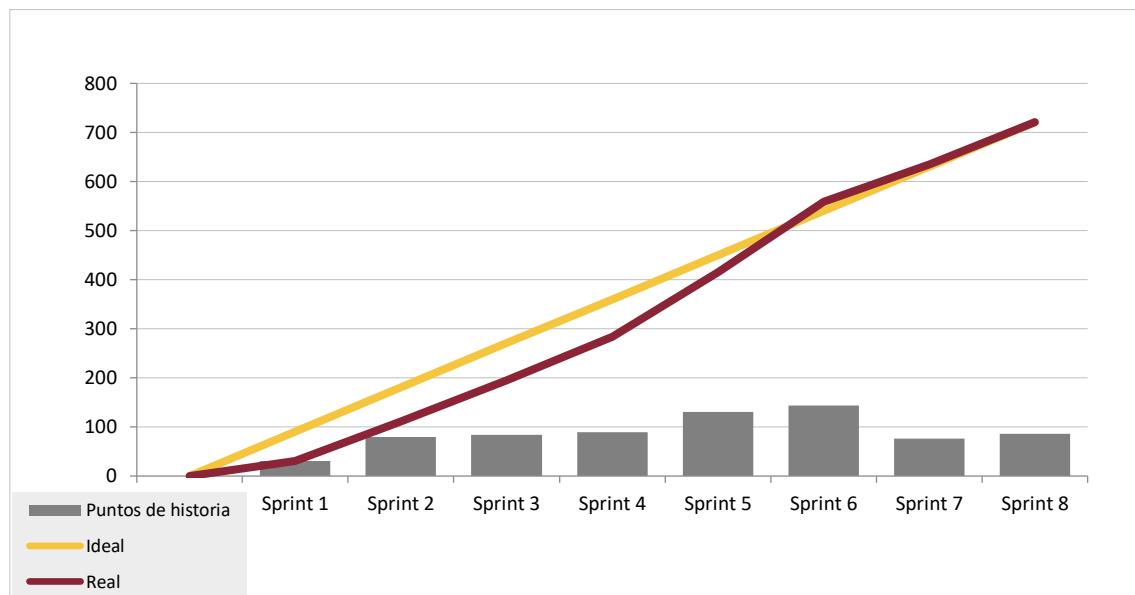


Tabla 9 – Burn-Up global

3.6 Presupuesto del proyecto

Resumen del proyecto	
Costes de personal	4.071,3 €
<i>Sueldo bruto</i>	3.105,5 €
<i>Seguridad Social</i>	965,8 €
Costes de material	182,8 €
Costes indirectos	510,5 €
Beneficio industrial	952,2 €
Total	5.716,8 €

3.6.1 Costes de personal

Según el portal Glassdoor [34] el sueldo bruto para un ingeniero software junior a tiempo completo en España es de unos 23000 euros al año. Teniendo en cuenta que en España se establecen 1.800 horas anuales en el “Convenio colectivo estatal de empresas de consultaría y estudios de mercado y de la opinión pública” [35] se tomará 12.78 € la hora como sueldo estándar para un desarrollador como el autor de este proyecto.

En total, en cuestiones de desarrollo, este proyecto ha consumido un total de 243 horas, así que el salario bruto es de unos 3.105,5 €.

De este salario bruto se ha de calcular los costes de la Seguridad Social de la compañía. Según las directrices establecidas por el portal de la seguridad social en España [36], el coste total que tendría que pagar la empresa es de 31,1% de los cuales 23,6% serían del tipo de cotización común, 6.7% es el porcentaje correspondiente a desempleo para contratos de una duración determinada a tiempo parcial, 0.2% corresponden a FOGASA y finalmente 0.6% a la formación profesional. Por tanto, asociado a este porcentaje la empresa pagaría 965,8 €.

El total de los gastos de personal del proyecto es de 4.071,3€.

3.6.2 Costes de material

Para este apartado se ha calculado la depreciación de los distintos dispositivos que han sido utilizados. En concreto, para este proyecto se han usado tres dispositivos:

- PC HP ay150ns + SSD Samsung EVO 8
- Teléfono móvil Xiaomi Mi 6 64 GB.
- Teléfono móvil Xiaomi Mi A1 64 GB.

Según queda establecido en el portal de la agencia tributaria, los equipos informáticos tienen un período de depreciación máximo de 4 años [37].

Para el primer caso, el ordenador portátil tuvo un coste inicial de 625 €, comprado hace tres años. Teniendo en cuenta que el proyecto ha tenido una duración de 5 meses, la amortización sería en este caso de:

$$\frac{625}{12 * 3} \times 5 = 86.8 \text{ €}$$

Para el segundo caso, el primer dispositivo móvil tuvo un coste de 360 €, comprado hace tres años. Teniendo en cuenta que el proyecto ha tenido una duración de 5 meses, la amortización sería en este caso de:

$$\frac{360}{12 * 3} \times 5 = 50 \text{ €}$$

Finalmente, en el caso del segundo dispositivo móvil tuvo un coste de 221€, comprado hace dos años. Teniendo en cuenta que el proyecto ha tenido una duración de 5 meses, la amortización sería en este caso de:

$$\frac{221}{12 * 2} \times 5 = 46 \text{ €}$$

Por tanto, la cifra total de sumas de costes en materiales asciende a los 182,8 €.

3.6.3 Costes indirectos

Debido a que no se conocen con exactitud el importe de los costes indirectos asociados al proyecto se ha considerado incluir como costes indirectos el 12% de los costes directos, por lo que el importe sería de:

$$(4.071,3 + 182,8) \times 0.12 = 510,5 \text{ €}$$

3.6.4 Beneficio industrial

En este proyecto se ha contemplado un beneficio industrial del 20%. Teniendo en cuenta que los gastos totales ascienden a 4767.6 € el beneficio industrial sería de:

$$4767.6 \times 0.2 = 952.2 \text{ €}$$

4 ANÁLISIS Y DISEÑO

Hasta este momento, ha sido detallada tanto la tecnología utilizada como la planificación y rendimiento mantenido a lo largo de todo el proyecto.

El objetivo de este bloque es mostrar el análisis realizado para el diseño de la aplicación desarrollada. Para ello a continuación se mostrarán una serie de diagramas, mockups y casos de usos que ayudarán al lector a entender el comportamiento y diseño de la aplicación de una forma más técnica.

4.1 Análisis

En primer lugar, se realizará un análisis de la aplicación. Para ello serán detallados los casos de uso, una serie de mockups de las distintas vistas del sistema que muestran su funcionalidad principal y finalmente el diagrama UML del dominio del sistema.

4.1.1 Casos de uso

En este bloque se describirá que podrán hacer y como podrán interactuar los usuarios con el sistema. Es decir, las solicitudes que los usuarios pueden realizar y como la aplicación responderá ante estas.

Antes de comenzar, para todos los casos de uso donde se utilice el superíndice “^{TR}” la funcionalidad debe trabajar en tiempo real, lo que quiere decir que si el usuario inicia su cuenta en otro dispositivo de manera simultánea los cambios tienen que reflejarse instantáneamente en ambos. En el caso de que se utilice el superíndice “^{TR-L}”, significa que además de utilizar la funcionalidad de tiempo real, todos los cambios quedan almacenados de manera persistente de forma local, con independencia a una conexión a internet. Por otro lado, el número que aparece entre paréntesis será utilizado como un identificador para la tabla de trazabilidad que será expuesta en el apartado 4.1.2.

En el caso de los listados, a pesar de no utilizar la nomenclatura anterior, para todos los casos en los que el usuario realice acciones en ellos (actualización, borrado o agregación de algún elemento) mantendrán actualizaciones en tiempo real, por lo que el usuario verá reflejado los cambios en ellos de manera instantánea, manteniendo una copia tanto local como en remoto. En el caso donde esta copia solo esté presente de manera remota se señalará con el correspondiente superíndice.

GESTIÓN DE USUARIOS

Respecto a los usuarios, en cuanto a su perfil en la aplicación, el sistema le permitirá realizar la siguiente funcionalidad.

- **Registro y Login (1):** el sistema permitirá tanto registrar como hacer “login” al usuario en la aplicación mediante la pasarela de Google.

El motivo de unificar ambas acciones fue recomendación directa por parte de la ONCE. Al utilizar un dispositivo Android, el usuario se ve obligado a contar, como mínimo, con una cuenta Gmail [38] por lo que la acción tanto de registrarse como de entrar al sistema se reduce a un clic y elegir la cuenta deseada.

GESTIÓN DE IDENTIFICACIÓN

- **Identificación de productos (2):** para todo aquel usuario registrado en el sistema que cuente con NFC en su dispositivo el sistema deberá devolver toda la información del producto disponible si este cuenta con una etiqueta compatible con la aplicación y la aplicación la escanea.
- **Activar compatibilidad (3):** el sistema deberá poder detectar si el dispositivo cuenta con el hardware necesario (NFC). En el caso de que este requisito se encuentre disponible pero este desactivado, el sistema se lo notificará al usuario y permitirá configurarlo de una forma sencilla.

GESTIÓN DE PRODUCTOS

- **Guardar productos^{TR-L} (4):** una vez identificado un producto por medio de la aplicación, deberá permitírselo al usuario poder guardarlo de manera privada en alguno de sus almacenes por defecto.

Como aclaración, con motivo de facilitar este proceso para un usuario con discapacidad visual, cada producto contará con un almacén por defecto en el que será guardado. La aplicación, por otro lado, deberá crear ese almacén en el caso de que no exista o añadirlo a uno existente en caso contrario. Con esta especificación se pretende reducir al mínimo los pasos que se deben ejecutar para realizar la acción de guardado.

- **Mostrar el listado de almacenes (5):** una vez guardado los productos, se generarán los distintos almacenes donde se guardarán. La aplicación debe permitir listar todos ellos de una manera sencilla y visual al usuario. Además, en función del tipo de productos que guarde cada almacén la aplicación le establecerá una foto por defecto. También se deberá mostrar información básica, como la última vez que fue editado ese almacén (por incorporar o borrar un elemento, por ejemplo) o el número total de elementos que almacena.
- **Mostrar un almacén (6):** se le ofrecerá la opción al usuario de poder mostrar el listado de todos los productos de un almacén (que ha guardado personalmente). Para cada producto listado del almacén, se mostrará información importante como una pequeña imagen, su modelo y marca.
- **Mostrar un producto (7):** tras listar los productos de un almacén el sistema debe poder mostrar de manera individual más información de alguno de ellos si el usuario lo selecciona.
- **Eliminar un producto^{TR-L} (8):** una vez mostrado un producto el usuario debe tener la opción de eliminarlo de ese almacén si así lo requiere. Esta acción debe ejecutarse en tiempo real.

- **Eliminar un almacén^{TR-L}(9):** en el caso de que un almacén no cuente con ningún producto (en el caso de que el usuario los haya eliminado todos), el usuario debe tener la posibilidad de eliminar dicho almacén del sistema.

GESTIÓN DE LISTA DE LA COMPRA

- **Crear lista de la compra (10):** cualquier usuario registrado en el sistema debe poseer una lista personal y privada (si lo desea) con capacidad ilimitada de añadir productos.
- **Añadir productos^{TR-L} (11):** el usuario podrá añadir a su lista de la compra tantos productos como desee, de una forma rápida e intuitiva.
- **Mostrar listado de productos (12):** una vez añadidos productos a la lista de la compra el sistema deberá listarlos de forma ordenada. Para cada producto de la lista, el sistema deberá indicar el usuario que ha añadido este producto a la lista.

En el caso de que el usuario que utiliza la aplicación sea el mismo que el usuario que añadió el producto se obviará esta información adicional para no saturar el listado.

- **Marcar un producto^{TR-L} (13):** para todos los productos almacenados en la lista, el usuario tendrá la posibilidad de marcar ese producto como seleccionado y que quede reflejado de una forma visual.
- **Borrar un producto^{TR-L} (14):** el usuario podrá eliminar un producto de la lista en el caso de que lo desee, ya se encuentre marcado o no.
- **Añadir acceso a lista^{TR-L} (15):** todas las listas del sistema pueden ser compartidas entre los usuarios registrados, a través del correo con el que configuraron su cuenta en la aplicación. Esta acción se llevará a cabo por medio de una invitación a través de la propia aplicación que el usuario receptor tendrá que aceptar o rechazar.

Para toda nueva invitación, el sistema deberá comprobar si anteriormente el usuario había enviado esta petición al mismo correo. En el caso de que exista esta invitación y su estado sea pendiente o aceptada se le notificará al usuario de que no puede volver a enviarla.

- **Listar los usuarios con acceso (16):** el sistema ofrecerá la posibilidad de mostrar un panel de todos los usuarios que cuentan con acceso a la cesta de la compra mediante un listado que muestre su foto de perfil, su nombre y correo además de una insignia indicativa que señale el estado de su invitación pudiendo ser esta aceptada o pendiente, todo ello de manera individualizada.
- **Eliminar acceso a lista^{TR-L} (17):** el usuario dueño de una lista tendrá la posibilidad de eliminar el acceso de esta a cualquier usuario, ya haya aceptado

la invitación o no. Una vez eliminado el acceso no podrá añadir, eliminar o ver la lista.

En el caso de que se encuentre viéndola (la lista de la que ha sido expulsado) se le redireccionará automáticamente fuera de esta, en concreto al panel de invitaciones.

GESTIÓN DE LA LISTA DE LA COMPRA COMPARTIDA

- **Listar invitaciones^{TR-L} (18):** el sistema mostrará un listado de todas las invitaciones que ha recibido el usuario. Por cada una de ellas se mostrará la imagen de perfil del usuario, su nombre, su correo, y el estado en el que se encuentra dicha invitación.
- **Aceptar invitación^{TR-L} (19):** para cada invitación mostrada al usuario en el listado, el sistema debe permitir aceptar la invitación.
- **Rechazar invitación^{TR-L} (20):** para cada invitación mostrada al usuario en el listado, el sistema debe permitir rechazarla. Esta posibilidad también estará permitida después de haber aceptado la invitación.
- **Mostrar lista de la compra^{TR} (21):** para cada invitación, en el caso de que se acepte, el sistema tiene que permitir ver al usuario un listado de todos los productos que posee la lista del usuario que le ha enviado la petición, este listado al igual que en el caso anterior, debe mostrar por cada producto el usuario que lo ha insertado en la lista.
- **Insertar producto en la lista^{TR} (22):** el usuario invitado tendrá la posibilidad de añadir a la lista a la que ha sido invitado tantos productos como desee, de igual forma que en el caso anterior.
- **Marcar producto en la lista^{TR} (23):** para todos los productos almacenados en la lista, el usuario tendrá la posibilidad de marcar ese producto como seleccionado y que quede reflejado de una forma visual, al igual que en el caso anterior.
- **Eliminar producto de la lista^{TR} (24):** el usuario invitado tendrá la posibilidad de eliminar cualquier producto de la lista que ha sido invitado, al igual que en el caso anterior.
- **Listar los usuarios con acceso^{TR} (25):** la aplicación ofrecerá la posibilidad de mostrar un panel en el que aparezcan todos los usuarios que cuentan con acceso a la cesta de la compra, mediante un listado que muestre su foto de perfil, su nombre y correo además de una insignia indicativa que señale el estado de su invitación pudiendo ser esta aceptada o pendiente, todo ello de forma individualizada. Además, se deberá mostrar de una forma clara y visual quien es el dueño y administrador de la lista.

GESTIÓN DE NOTIFICACIONES

Para las siguientes acciones, el sistema deberá alertar al usuario receptor de una acción mediante una notificación push Android [39].

Todas las notificaciones estarán compuestas del nombre del usuario que ha desencadenado la notificación, su imagen de perfil y un pequeño texto explicativo indicando un resumen de la acción en sí.

- **Nueva invitación (26):** tras ser invitado por otro usuario a su lista, la aplicación alertará al usuario de que ha recibido esta invitación mediante una notificación, donde se indicará al usuario receptor que le han invitado a una nueva lista. Si pulsa sobre ella el sistema debe mostrar la vista necesaria para aceptar o rechazar la invitación.

Como anteriormente se indicó la notificación deberá contener la información necesaria del usuario que le ha invitado.

- **Invitación aceptada (27):** si el usuario al que se le ha enviado una invitación a participar en una lista la acepta, se notificará al dueño de esta lista que la notificación ha sido aceptada. Si el usuario pulsa sobre la notificación el sistema le debe mostrar su propia lista.
- **Invitación rechazada (28):** si el usuario al que se le ha enviado una invitación a participar en una lista la rechaza, se notificará al dueño de esta lista que la notificación ha sido rechazada. Si el usuario pulsa sobre la notificación el sistema le debe mostrar la lista de invitaciones.
- **Invitación eliminada (29):** si el usuario al que se le envió una invitación a participar en una lista (y la aceptó) la elimina de su lista de invitaciones, se notificará al dueño de esta lista de que el usuario ha decidido dejar de participar. Si el usuario pulsa sobre la notificación el sistema le debe mostrar la lista de invitaciones.
- **Nuevo producto (30):** si un usuario invitado añade un producto, se le notificará de esta acción al usuario dueño de la lista.

GESTIÓN DE CONTROL DE VOZ

Para todos los casos de uso anteriormente desarrollados el sistema debe permitir los siguientes apartados.

- **Integración con Google TalkBack (31).** Para todas las vistas de la aplicación, el asistente de Android debe poder describir la vista en su totalidad de una forma simple e intuitiva.
- **Navegabilidad y usabilidad (32).** Para todos los casos de uso anteriormente nombrados, desde la identificación hasta eliminar una invitación, la aplicación debe permitir realizarlo todo mediante comandos de voz intuitivos y variados.

Como ejemplo de comando de voz variados el sistema añadirá un producto a la lista tanto usando “Por favor, añádeme una garrafa de aceite de 5 litros” como también con “Añadir una garrafa de aceite de 5 litros”.

4.1.2 Matriz de trazabilidad

En la columna con el nombre “Caso de uso” cada número identifica el caso de uso que se analiza en esa fila el cual se asocia al identificador establecido en el apartado 4.1.1 de este documento.

La columna “Objetivos” representas los objetivos iniciales establecidos en este proyecto los cuales, a su vez, quedan divididos en una serie de subobjetivos.

Se distinguen por tanto los siguientes objetivos principales:

- **APDV – Ayudar a personas con discapacidad visual**
- **FDD – Facilitador del día a día**
- **FC – Fidelizador de clientela**

Para cada uno de estos objetivos principales quedan detallados una serie de subobjetivos los cuales se encuentran diferenciados mediante un identificador en el apartado 1.2 de este documento.

Matriz de trazabilidad – Casos de uso y objetivos										Caso de uso	
Objetivos											
APDV			FDD						FC		
1	2	3	4	5	6	7	8	9	10		
	X	X						X		1	
X	X	X			X		X	X	X	2	
	X	X						X		3	
	X	X				X	X	X	X	4	
	X	X					X	X	X	5	
	X	X					X	X	X	6	
X	X	X						X	X	7	
	X	X				X	X	X	X	8	
	X	X					X	X		9	
X	X	X	X				X	X	X	10	
X	X	X	X				X	X	X	11	
X	X	X	X				X	X	X	12	
X	X	X	X				X	X		13	

X	X	X	X				X	X			14
	X	X	X	X			X	X	X		15
	X	X	X				X	X			16
	X	X	X				X	X			17
	X	X	X				X	X			18
	X	X	X	X			X	X	X		19
	X	X	X	X			X	X			20
	X	X	X	X			X	X			21
	X	X	X	X			X	X	X		22
	X	X	X	X			X	X	X		23
	X	X	X	X			X	X			24
	X	X	X	X			X	X			25
	X	X					X	X			26
	X	X					X	X			27
	X	X					X	X			28
	X	X					X	X			29
	X	X					X	X			30
	X	X									31
	X	X									32

4.1.3 Mockups

Al autor le gustaría destacar que en esta sección se mostrarán los mockups que se presentaron en la sede de la ONCE, para conocer la viabilidad del sistema que se planteó inicialmente. Se realizaron únicamente 3 maquetas, con las que mostrar una idea de que podría hacer el sistema.

Sin embargo, debido a la situación a la que se enfrentó el proyecto a causa de la pandemia del COVID-19 [40] la aplicación sufrió grandes cambios.



Ilustración 24 - Pantalla de Sign-up y Log-in

En esta primera maqueta se muestra lo que sería la pantalla de inicio nada más abrir la aplicación.

Tal y como puede apreciarse en este punto de la aplicación aún no se consideraba la funcionalidad del uso de lista de compras, por lo que no aparece en el resumen.

En cuanto a la vista en sí, contaría con un único botón, el cual activaría la pasarela de Google tanto para el registro como para iniciar sesión en la aplicación si el usuario ya se había registrado anteriormente. La existencia y configuración mediante un único botón facilitaría la navegabilidad para personas con discapacidad visual.

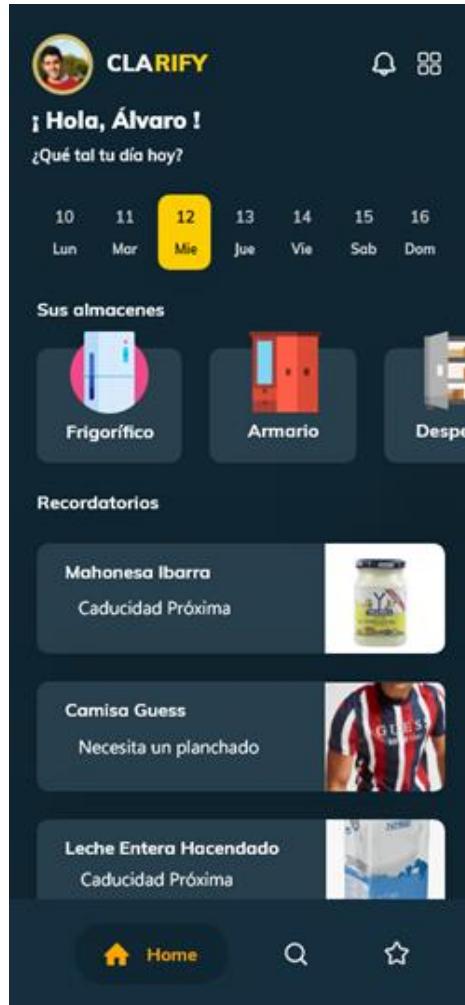


Ilustración 25 - Vista principal

Una vez el usuario inicie sesión en el sistema esta será la primera vista que se le mostrará (ilustración 25). En esta versión inicial se planteó la idea de configurar recordatorios sobre productos, pero posteriormente descartó por la complejidad que supondría para una persona con discapacidad visual su configuración para cada uno de los distintos productos de los que dispone.

Por otro lado, se muestra también la funcionalidad de almacenes de productos personales, que si ha sido desarrollada en la aplicación final de este proyecto. Respecto a esta primera versión ha sufrido algunos cambios, como su traslado a una vista única y separada por recomendación de los usuarios pilotos.

Por último, en la parte inferior se establece la barra de navegación que será visible en casi la totalidad de todas las vistas. En este caso, la versión final de la aplicación de este proyecto mantiene una estructura bastante similar a la que se presenta en esta maqueta.

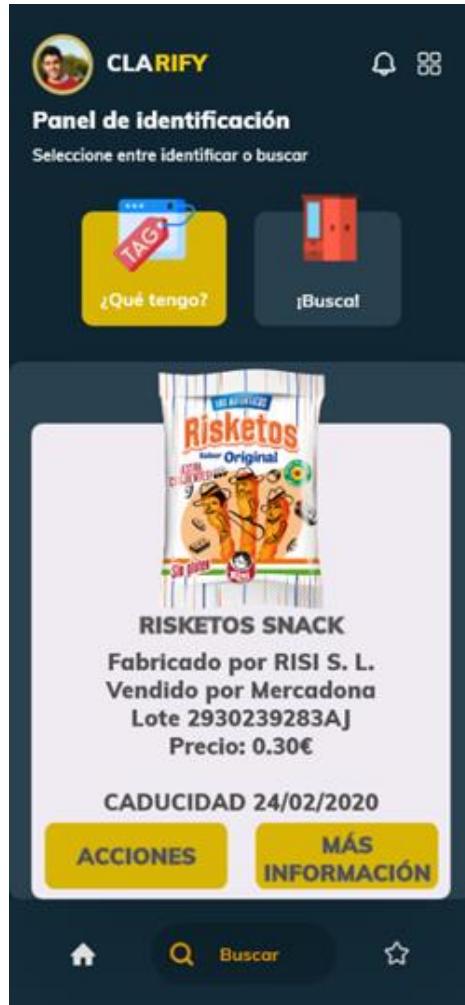


Ilustración 26 - Vista de Identificación

En esta ilustración se muestra la funcionalidad más interesante para los usuarios con discapacidad visual. Se planteó de tal manera que en primer lugar se mostraría un cuadro en blanco, el cual se completaría con información en el caso de que el usuario acercara su teléfono a un producto con una etiqueta NFC compatible y la aplicación fuese capaz de reconocerla.

Además de mostrar los datos con mayor interés del producto la aplicación le ofrecería la posibilidad al usuario de mostrar aún más información, así como una serie de acciones como, por ejemplo, añadir ese producto a un almacén personal y privado.

Por otro lado, el autor planteó en la sede de la ONCE un sistema de localización de un producto en concreto, pero se le recomendó que abandonará la idea tal y como la había planteado ya que resultaría un poco incómodo y engorroso para el usuario buscar un producto en concreto a través de un buscador y posteriormente realizar un barrido por una estantería al completo.

4.1.4 Diagramas de dominio

En este último apartado se mostrarán los diagramas de dominio del sistema. En concreto se realizarán dos distinciones.

En primer lugar, se mostrará el diagrama sobre el que se rige la aplicación en el servidor, es decir, sobre la base de datos Firebase Realtime Database. En concreto, al ser una base de datos NoSQL (no relacional) las relaciones entre clases se realizan por medio de un atributo id en la clase correspondiente, aun así se mostrará un pequeño diagrama en el que las líneas representan la unión entre las distintas clases.

Para facilitar la lectura de este diagrama, ya que la estructura del sistema (en la parte remota) se encuentra dividida en dos grandes apartados los cuales son parte privada y parte pública, se ha utilizado esta gran división para crear dos subdivisiones.

La parte privada está compuesta de los distintos usuarios (User):

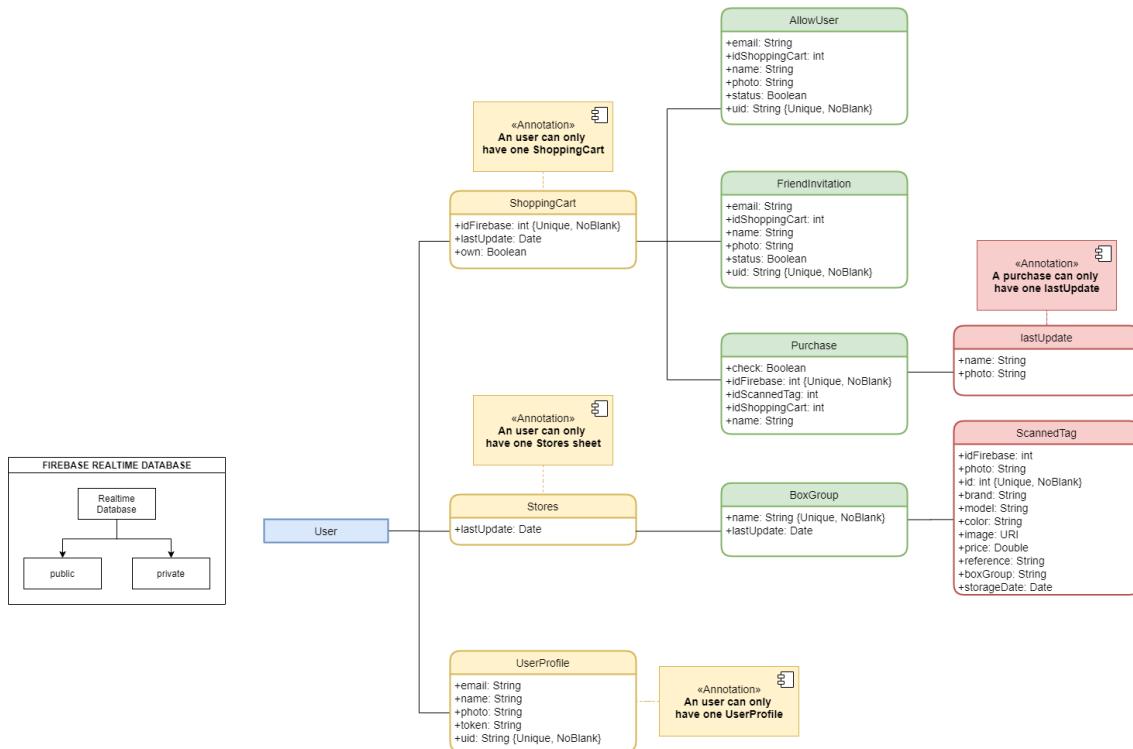


Ilustración 27 – Diagrama de dominio Firebase private

Por otro lado, la parte privada está conformada por el conjunto de etiquetas públicas.



Ilustración 28 – Diagrama de dominio Firebase public

En segundo lugar, se mostrará el diagrama de dominio que utiliza la aplicación a nivel local. Para su gestión se ha utilizado Realm, de nuevo una base de datos NoSQL.

En concreto, su dominio es totalmente homólogo al de Firebase, con la peculiaridad de que a todas las clases se le ha añadido un atributo de ID local, para poder permitir tareas de sincronización.

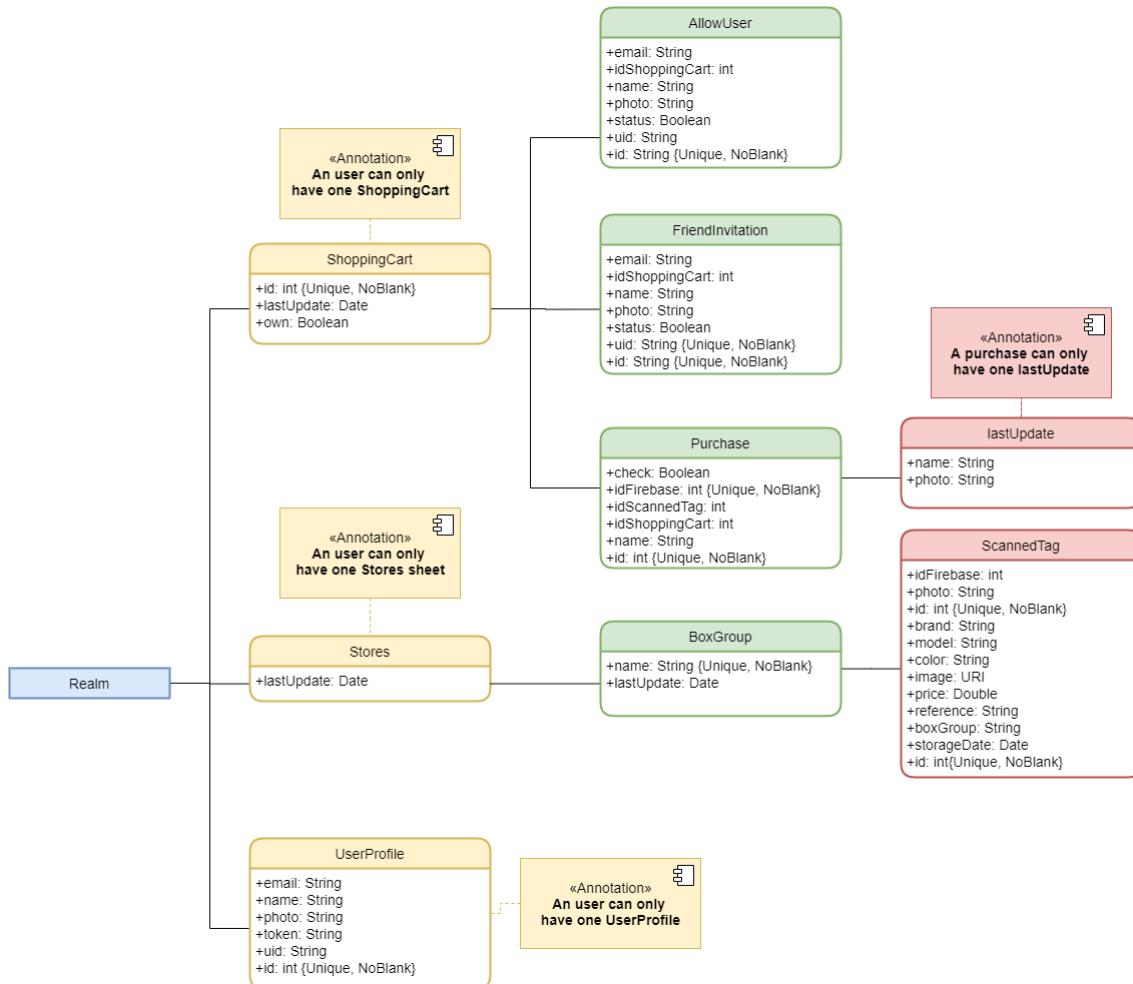


Ilustración 29 – Diagrama de dominio Realm

Todas las peculiaridades y la interconexión de las distintas clases serán explicadas en la sección de implementación de este documento.

4.2 Diseño

En esta sección del documento se tratará de dar una descripción del proyecto más detallada mediante los diferentes diagramas de componentes que conforman la aplicación Android.

4.2.1 Arquitectura

Para la aplicación móvil que se ha desarrollado, se ha utilizado una arquitectura de dos capas. Por un lado, se ha utilizado Android, plataforma contenedora tanto del back-end como del front-end de la aplicación y por otro lado un doble sistema de bases; local y remoto.

En la siguiente imagen puede verse el diagrama de esta arquitectura.

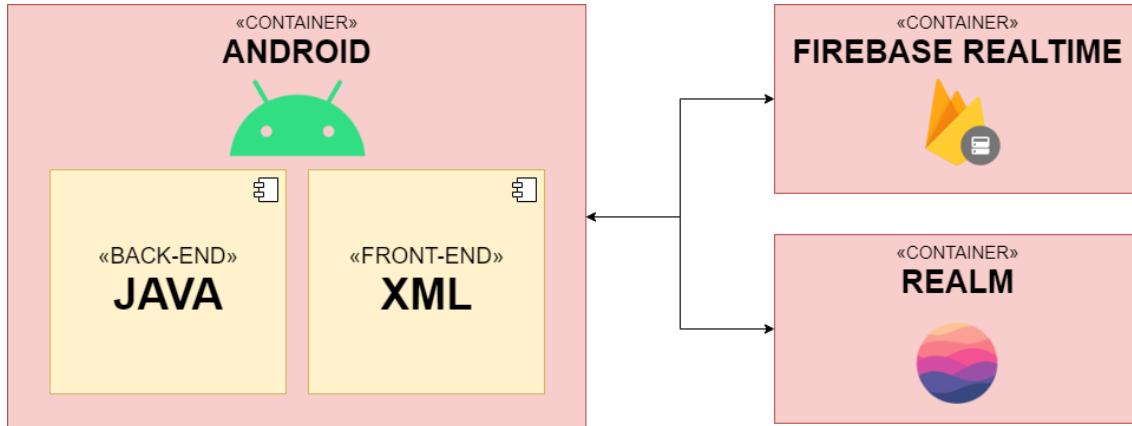


Ilustración 30 – Diagrama de arquitectura

Como puede observarse en el diagrama anterior, es en el cliente donde es realizada toda la lógica de negocio, tanto back-end como front-end.

Por otro lado, el cliente también será el encargado de realizar la comunicación con las distintas bases de datos para labores de consulta, inserción y actualización de datos en local, es decir, en Realm DB y su correspondiente sincronización con Firebase Realtime DB.

4.2.2 Diagrama de componentes

Finalmente será mostrado el diagrama de componentes, donde queda detallado cada una de las dependencias de nuestro proyecto con las librerías externas utilizadas. No se mostrarán todas, pero si las que han resultado de mayor interés para el desarrollo del proyecto.

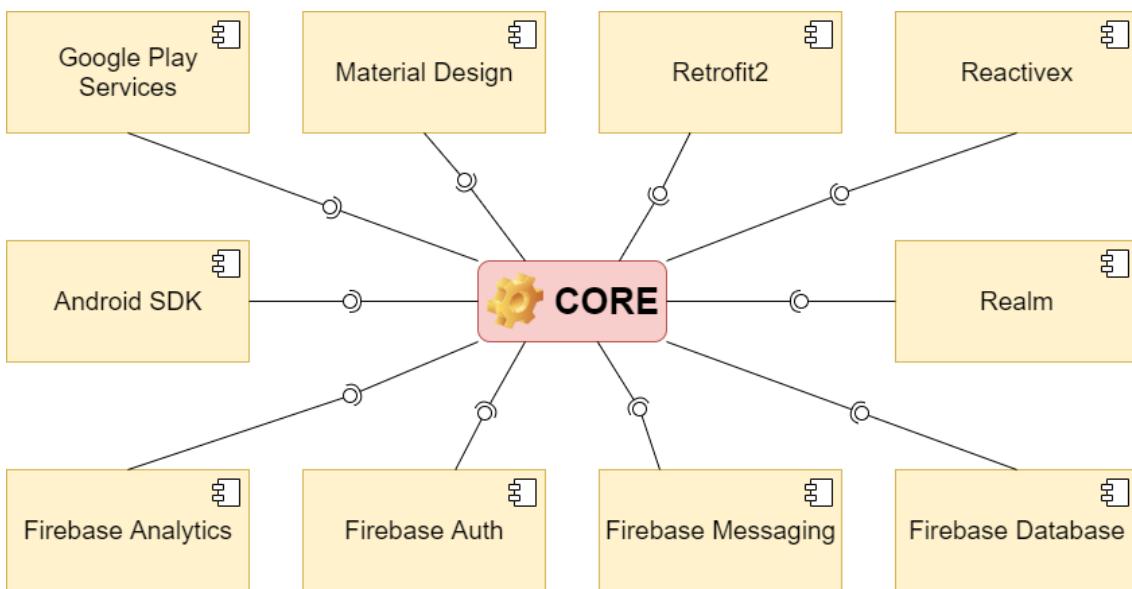


Ilustración 31 – Diagrama de componentes

Para cada una de ellas, se aporta un breve resumen de su funcionalidad.

- **Google Play Services:** provee acceso a la API general de Google.
- **Firebase Analytics:** gestión de acceso, notificaciones y análisis.
- **Firebase Auth:** control de registro y login en el sistema.

- **Firebase Database:** acceso a la base de datos remota.
- **Firebase Messaging:** para la gestión de notificaciones.
- **Android Material (design):** desarrollo de vistas.
- **Reactivex:** permite construir eventos asíncronos.
- **Retrofit2:** hace que sea relativamente más fácil recuperar y cargar JSON (u otros datos estructurados) a través de REST.
- **Realm:** gestión de la base de datos local.

Existen muchas más librerías que han sido utilizadas en este proyecto, en la mayoría de los casos para mejorar aspectos visuales. Otras muchas vienen incorporadas por defecto en la librería básica de Android, como el reconocimiento de voz. En este diagrama han sido incorporadas las principales librerías externas que más peso han tenido en este proyecto.

Para más información sobre las librerías utilizadas, puede consultar gradle del proyecto Android de la aplicación.

5 MANUAL DE USUARIO

El objetivo de este bloque tras haber mostrado todos los detalles de la planificación, análisis, desarrollo y diseño de la aplicación es mostrar el resultado final de este proyecto en funcionamiento. Para ello se proporcionará un manual de usuario en el que se realiza un recorrido completo por la funcionalidad de la aplicación, mostrando como usarla.

Para una mejor compresión de toda la funcionalidad, el autor se apoya en los bloques en los que se aglutinaron los distintos casos de usos expuestos en el apartado 4.1.1 de este documento, los cuales son:

- **Gestión de usuarios.** Todo lo que tenga que ver con el perfil del usuario en la aplicación en sí.
- **Gestión de lista de la compra.** Se muestra como añadir, eliminar y marcar distintos productos, así como la gestión de los distintos usuarios invitados en la lista.
- **Gestión de la lista de compra compartida.** Se indica los cambios respecto al funcionamiento de esta lista frente a la propia del usuario.
- **Gestión de identificación.** Toda aquella funcionalidad en la que se le permita a un usuario identificar un producto.
- **Gestión de productos.** Se entiende por esta gestión a toda la funcionalidad que permite consultar, actualizar o añadir productos al inventario privado del usuario.
- **Gestión de notificaciones.** Alertas fuera de la aplicación.
- **Gestión de control de voz.** Se recoge como realizar toda la funcionalidad anteriormente descrita mediante el uso de comandos de voz con el uso conjunto de Google TalkBack.

5.1 Gestión de usuarios

INICIO EN EL SISTEMA Y/O REGISTRO

Nada más abrir la aplicación, se le mostrará al usuario una pequeña vista inicial, siempre y cuando no exista ninguna sesión abierta, ya que el usuario tiene la posibilidad de elegir mantener su sesión activa cuando deja de usar la aplicación.

En esta vista solo se mostrará un único botón el cual activa la interfaz que permite realizar, en la misma vista, las funciones de inicio de sesión y registro en el sistema.

Por otro lado, puede observarse también el enlace a la web oficial, donde se mostrará información básica de este proyecto.

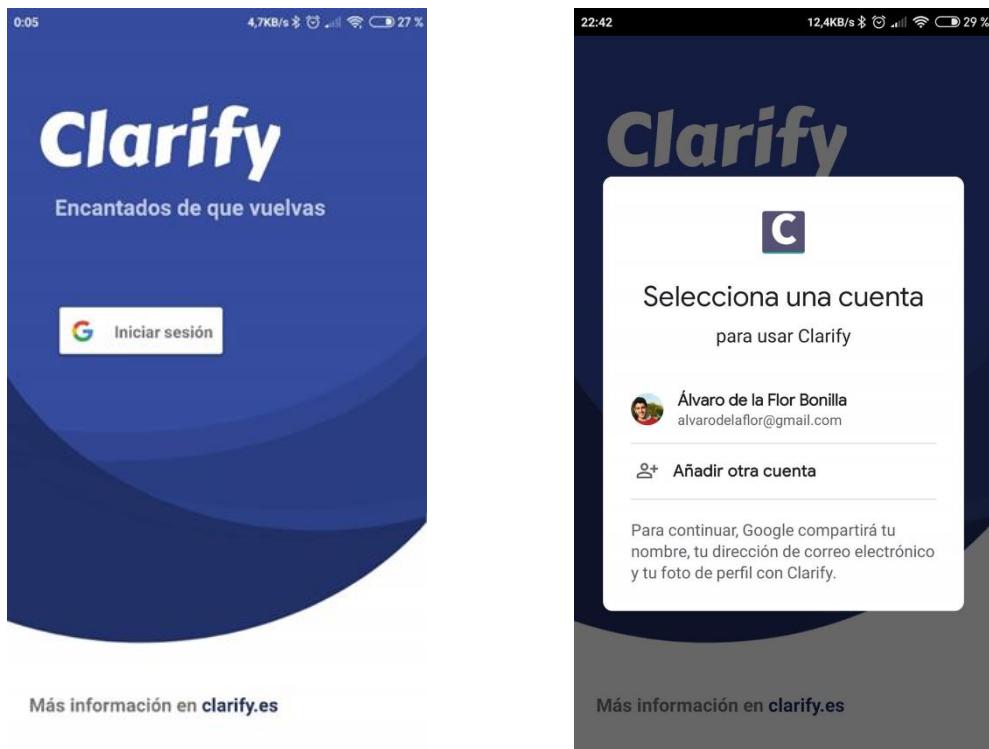


Ilustración 32 – Vista de bienvenida

Ilustración 33 – Selección de cuenta

En las dos imágenes anteriores se muestra el proceso que realizará el usuario de la aplicación. Tras pulsar el botón “*Iniciar sesión*” se abrirá la pasarela de autentificación de Google, la cual le solicitará al usuario que seleccione una cuenta para entrar en la aplicación. Bastaría con pulsar la cuenta deseada y automáticamente se iniciaría sesión en el sistema sin necesidad de escribir ninguna contraseña.

En el menú anterior por defecto se mostrarán las cuentas que se encuentran activas en el dispositivo, sin embargo, se ofrecerá al usuario la posibilidad crear una nueva cuenta.

Para ello pulsará en “*Añadir otra cuenta*” y se iniciará un proceso que añadirá una nueva cuenta al dispositivo (sincronización completa con los servicios de Google, no solo con nuestra aplicación).

En este proceso el usuario deberá indicar en primer lugar el correo con el que se quiere registrar, posteriormente la contraseña de este y finalmente, si las credenciales han sido correctas el dispositivo se sincronizará con esta cuenta. Al finalizar el proceso de sincronización, automáticamente se iniciará sesión en la aplicación con la nueva cuenta que hemos registrado, sin necesidad de volver al selector mostrado en la ilustración número 33.

A continuación, se puede ver una ilustración en el que aparece todo el proceso descrito.

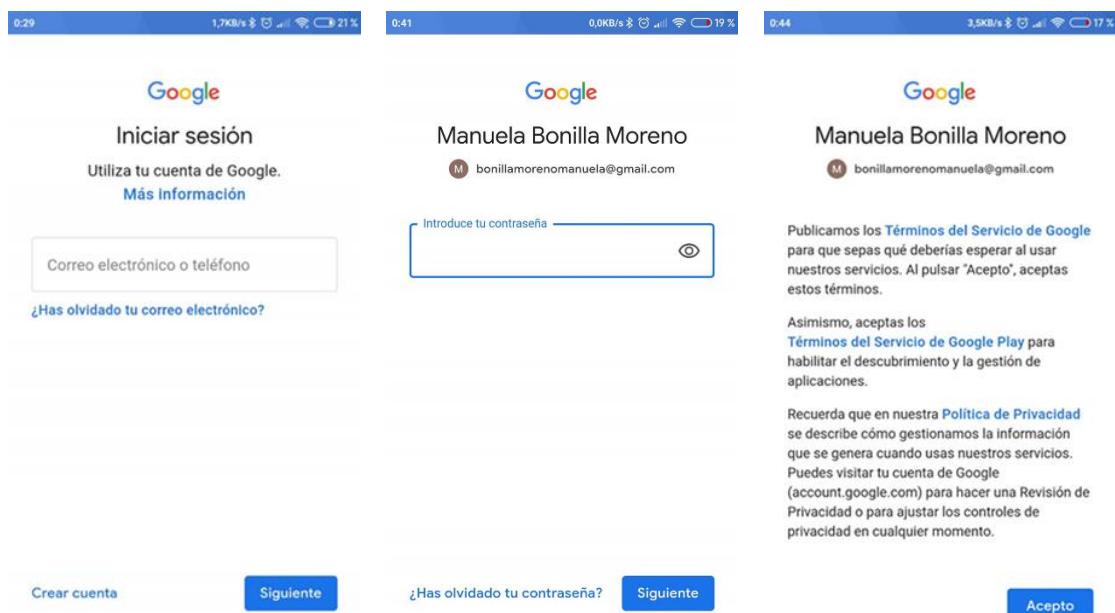


Ilustración 34 – Creación de una nueva cuenta

Tras el inicio de sesión, la aplicación lleva al usuario a la vista “Home”.

CERRAR SESIÓN



Ilustración 35 – Vista home

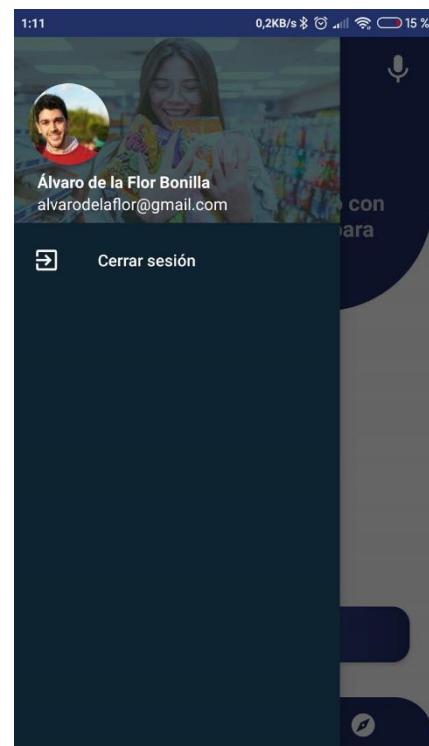


Ilustración 36 – Vista lateral habilitada

En la vista home además de la funcionalidad de la lista de la compra y la barra de navegación que será posteriormente explicada, se mostrará información básica del usuario como su foto de perfil y nombre.

Para cerrar sesión será tan sencillo como abrir el menú lateral y pulsar posteriormente en el botón de cerrar sesión. Acto seguido a esta acción la aplicación le mostrará al usuario de nuevo la vista de inicio de sesión de la ilustración número 32.

Existen dos formas de abrir el menú lateral, pulsando sobre la imagen de perfil o bien realizar el gesto de desplazar el dedo desde el borde izquierdo al derecho.

5.2 Gestión de lista de la compra

PANTALLA RESUMEN Y ACCIONES RÁPIDAS

Tal y como se refleja en la ilustración 35, nada más iniciar sesión en el sistema se le mostrará la vista home al usuario, la cual contiene dos ventanas principales. La primera de ellas muestra un resumen de la cesta personal del usuario, mientras que la segunda un conjunto de acciones rápidas, las cuales al encontrarse la lista de la compra completamente vacía no se encuentra ninguna activa.



Ilustración 37 – Vista home completa

En primer lugar, puede apreciarse que en el resumen de la cesta de la compra ya se observan datos relevantes como el número de productos que se han insertado en la lista, cuántos de ellos se han marcados (como posteriormente se verá, se ha añadido una funcionalidad que imita la acción de meter un producto en la cesta de la compra para diferenciarlos de los que aún no han sido introducidos en ella) y cuantas invitaciones de amigos para que el usuario pueda participar en su lista ha recibido, remarcando cuales de ellas aún no han sido aceptadas y se encuentran pendientes (en este caso dos, como indica el texto rojo superior “*2 nuevas*”). En el caso de que no haya ninguna invitación pendiente desaparecerá el texto rojo, reflejando únicamente un número exacto tal y como ocurre en las dos cifras de la izquierda.

No se debe confundir este número (que indica el número de invitaciones recibidas) con las invitaciones que el usuario puede enviar para que accedan a su lista, que como se verá posteriormente, se encuentra en un menú distinto.

Estos tres datos (número de productos, cuántos de ellos están marcados y el número de invitaciones) son actualizados en tiempo real, lo que quiere decir que, en primer lugar, mientras una tercera persona modifica la lista del usuario añadiendo, eliminando o marcando productos, esos cambios se verán reflejados inmediatamente en este pequeño resumen sin necesidad de que actualicemos la vista de forma manual.

Por otro lado, a diferencia de los casos anteriores, al tener productos (y usuario invitados) en la lista de la compra, los ajustes rápidos quedan activados. En concreto, todos los ajustes rápidos se encuentran activados en la ilustración 37. Se pasará a continuación a explicar cada uno de ellos.

- ***Marcar todos / Desmarcar todos***



Ilustración 38 – Ejemplo de marcar y desmarcar todo

En la ilustración anterior se muestra el ejemplo en el que en una lista de 8 productos (de los cuales 3 estaban marcados) se ha pulsado sobre el botón “*Marcar todos*”.

Como resultado, en la segunda imagen se muestra como el número de marcados es igual al número de productos (todos los productos han sido marcados) y el botón “*Marcar todos*”, se ha desactivado ya que al encontrarse todos marcados no tiene sentido su acción.

Por otro lado, el botón “*Desmarcar todo*” tiene una función homóloga a la anterior, pero en este caso desmarcando todos los productos de la cesta. Como se puede ver en la tercera imagen todos los productos han sido desmarcados y, al igual que en el caso anterior, el botón “*Desmarcar todos*” ha desaparecido, ya que no tienen sentido en el contexto de que todos los productos se encuentren desmarcados.

Además, cada una de estas dos funciones, al ser activadas se muestra una alerta informativa en forma de mensaje emergente que será de gran ayuda para usuarios con discapacidad visual, como se verá en el último apartado de este bloque.

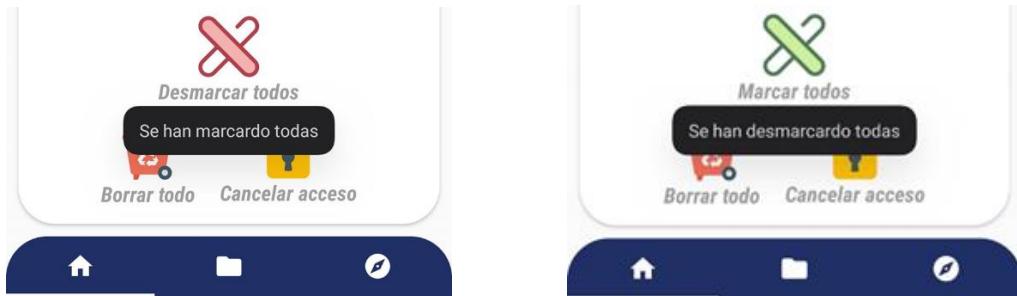


Ilustración 39 – Mensaje informativo 1

- Borrar todo**

El objetivo de este botón es vaciar toda la lista eliminando todos los productos y consecuentemente dejando el número de marcados también en cero.



Ilustración 40 – Proceso de vaciar la lista

En la imagen anterior se simula el proceso que tendría que realizar el usuario. Tras pulsar en el botón “Borrar todo” se abrirá una nueva ventana superpuesta de diálogo donde se le solicitará confirmación para realizar la acción. En el caso de pulsar en “No, cancelar” el usuario volverá a la primera vista de la sección anterior. Por el contrario, si pulsa sobre “Sí, borrar toda la lista” el diálogo se cerrará también, pero en esta ocasión todos los productos serán eliminados. Además, al no haber productos, los ajustes rápidos “Marcar todos” y “Desmarcar todos” se deshabilitan y desaparecen de esta vista.

Al igual que en la ocasión anterior, si finalmente se ejecuta la acción se le mostrará una alerta informativa en forma de mensaje emergente al usuario indicándole que ha eliminado todos los productos de su lista de la compra.

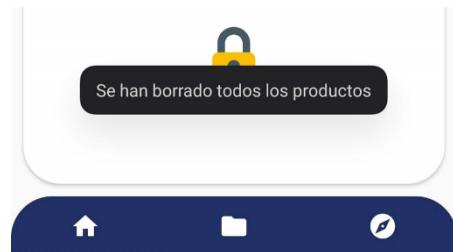


Ilustración 41 – Mensaje informativo 2

- **Cancelar acceso**

Para cerrar este bloque de pantalla inicial y ajustes rápidos concluiremos con esta acción. “Cancelar acceso” conlleva a que todas las invitaciones a participar en la lista de la compra personal del usuario serán canceladas y solo el usuario propietario podrá acceder y gestionar esta.

Cabe destacar que esta acción es dependiente de un proceso que aún no ha sido explicado, la acción de invitar a usuarios externos a una lista personal. En el ejemplo que se muestra a continuación la lista del usuario posee dos usuarios con acceso a ella.

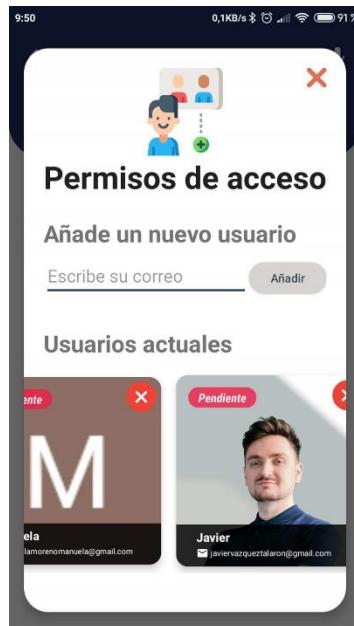


Ilustración 42 – Usuarios con acceso a la lista

Como puede apreciarse en la ilustración 42, el usuario actual ha creado dos invitaciones a su lista. En el caso de que quiera borrar todas las invitaciones de una forma rápida y desde la vista home, bastará con que pulse en el botón “Cancelar acceso” el cual iniciará un proceso muy similar al de borrar todos los productos de la lista.



Ilustración 43 – Proceso cancelar lista

Al usuario, tras pulsar en el botón “Cancelar acceso”, se le mostrará una ventana de diálogo de confirmación. Si pulsa sobre el botón “No, cancelar” el diálogo se cerrará y no se realizará ninguna acción. Por el contrario, si se pulsa sobre el botón “Sí, borrar todos los usuarios” se eliminarán todas las invitaciones que se han enviado, por lo que el usuario propietario de la lista será el único con acceso a ella a partir de ese momento.

En esta ocasión, al no existir productos en la lista, ningún ajuste rápido se encuentra disponible, por lo que se le mostrará un mensaje al usuario indicándole esto último.

Además, igual que en todos los casos anteriores, al realizarse de forma efectiva esta acción el usuario verá una alerta emergente informativa.



Ilustración 44 - Mensaje informativo 3

Cabe destacar, tras haber realizado todo este recorrido a lo largo de los ajustes rápidos, que la visibilidad o no de los distintos botones no depende de una tasa de refresco de algún método comprobador sino que, gracias a la característica de funcionalidad en tiempo real de esta aplicación, cada vez que se lanza un nuevo evento por parte del usuario dueño de la lista o por un usuario invitado, son los propios botones los que recalculan si se deben encontrar disponibles o no, de manera individual cada uno de ellos.

ABRIR LISTA DE LA COMPRA (O CARRITO)

Para realizar esta acción la aplicación permite hacerlo a través de 4 formas distintas.

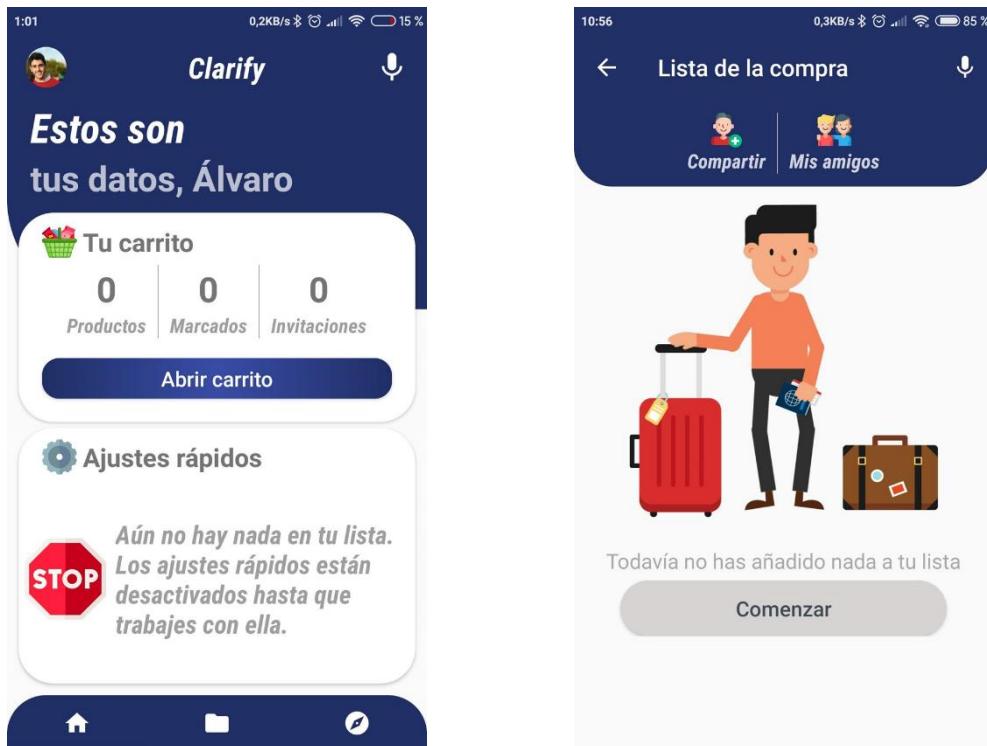


Ilustración 45 - Abrir lista de la compra

La ilustración anterior representa la acción de abrir el carrito, con las cuatro opciones indicadas:

- Pulsando sobre el botón “Abrir carrito”.
- Pulsando en el recuadro del número de productos (en este caso en “0 Productos”).
- Pulsando en el recuadro del número de marcados (en este caso en “0 Marcados”).
- Mediante comandos de voz (se explicará en su correspondiente apartado posteriormente).

Una vez quede desplegada la vista de la lista de la compra, en el caso de que no haya ningún producto en la lista se mostrará una vista personalizada indicándolo. En caso de que si haya productos se verá el listado de estos, como se indicará a continuación.

AÑADIR PRODUCTO A LA LISTA

Para explicar esta acción se diferenciará los casos en los que aún no hay ningún producto en la lista, como en la segunda parte de la ilustración 45 y los casos donde ya había productos en la lista.

Se comenzará por el caso en el que la lista se encuentra vacía.

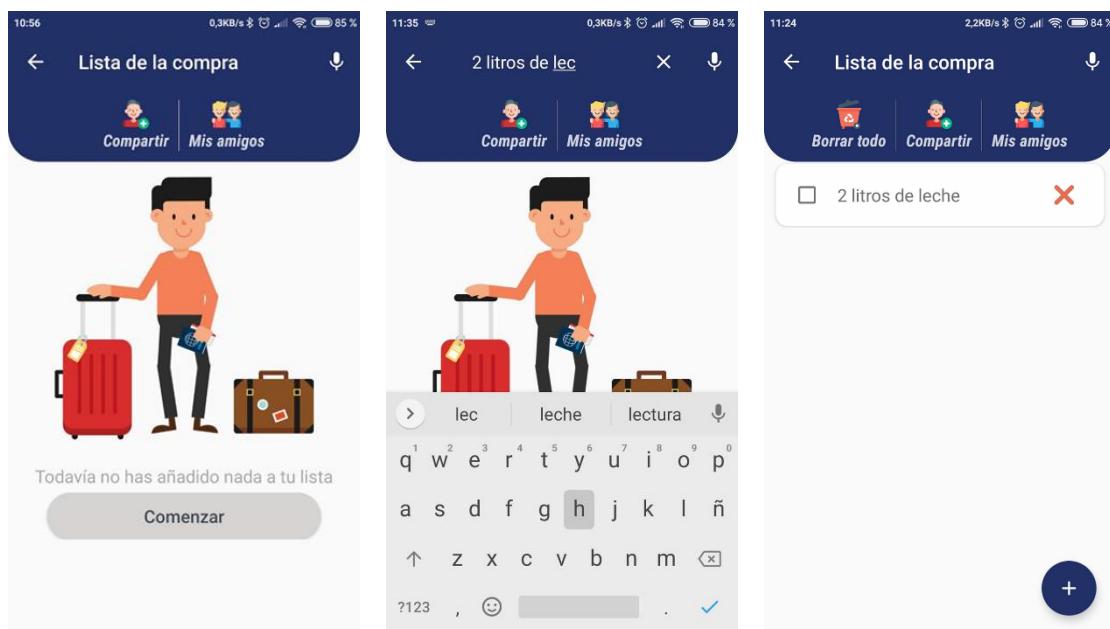


Ilustración 46 – Añadir producto a lista de la compra 1

En la imagen anterior se muestra el proceso que tendrá que seguir el usuario para añadir un producto en el caso de que la lista se encuentre vacía. En primer lugar, pulsará sobre el botón “Comenzar” y automáticamente se le mostrará el cuadro de texto donde escribirá el nombre del producto que desea añadir. Una vez termine, pulsará sobre el botón azul de validar (abajo del recuadro de texto a la derecha) y el producto se añadirá a la lista como aparece en la parte derecha de la ilustración 46. Si aún el usuario no ha escrito nada y abre el cuadro de texto se mostrará el texto aclaratorio “Nombre del producto” (en el siguiente caso ocurre un ejemplo de lo indicado).

En el caso de que la lista no estuviese vacía se seguirá el siguiente procedimiento, como el que se muestra en la siguiente ilustración.

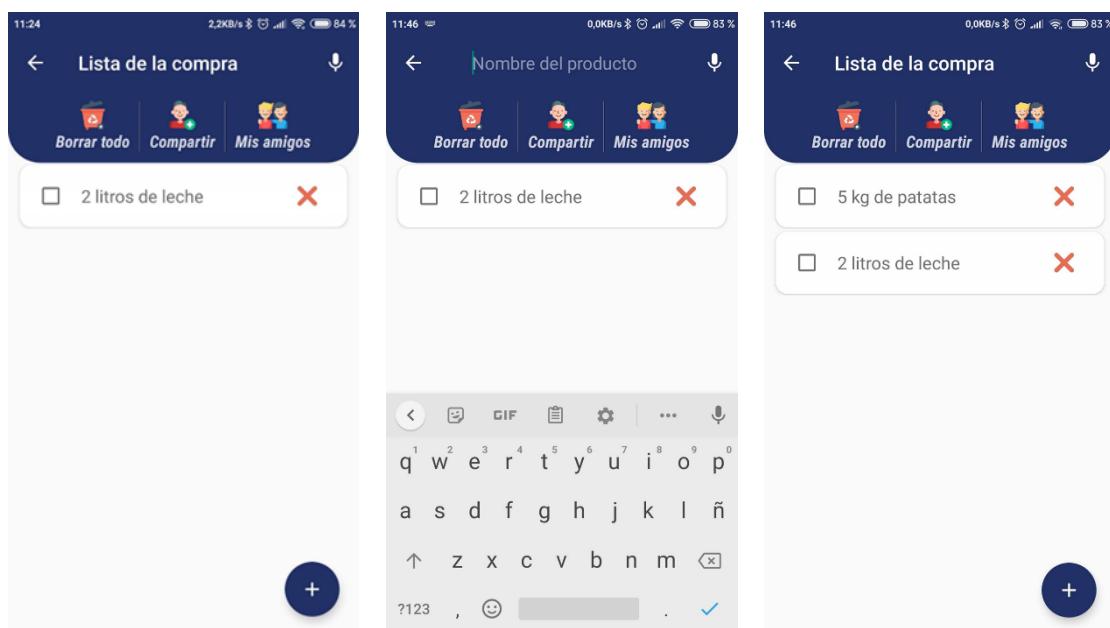


Ilustración 47 – Añadir producto a la cesta 2

En esta ocasión el usuario tendrá que pulsar sobre el botón circular azul con el signo blanco “+” en su interior. Seguidamente escribirá el nombre del producto que desea añadir y, como en el caso anterior le dará al botón de validar (en la misma posición que en el caso anterior). Automáticamente aparecerá el nuevo producto justo encima del anterior.

Apréciese también que cuando la lista pasa de cero a algún elemento, se habilita automáticamente la opción “*Borrar todo*”.

ELIMINAR UN PRODUCTO

La eliminación de un producto es un paso muy simple, únicamente requerirá que el usuario pulse en la cruz “X” roja y automáticamente el producto desaparecerá de la lista.

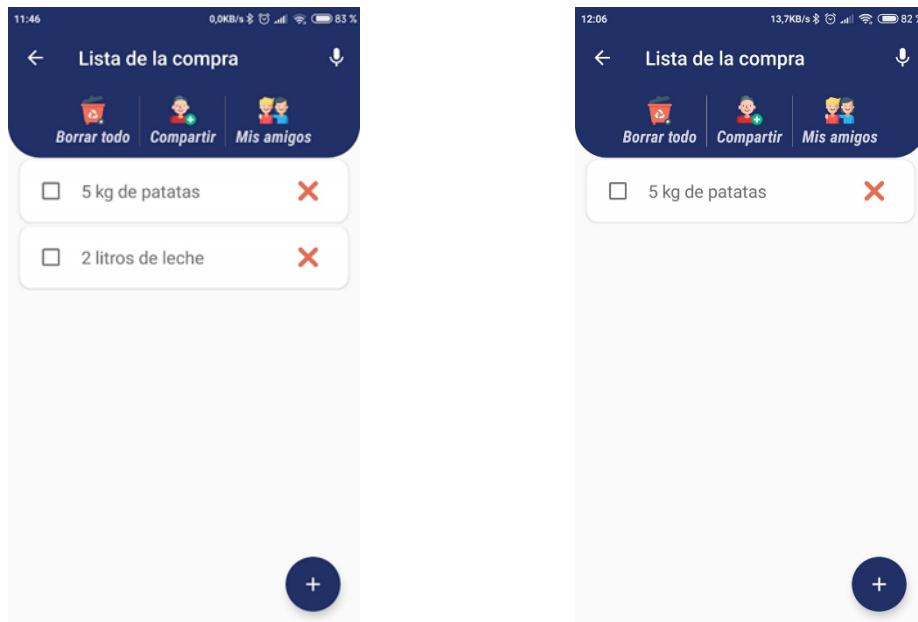


Ilustración 48 – Eliminar producto de la cesta

MARCAR UN PRODUCTO

Marcar un producto también sigue un procedimiento muy sencillo, bastará con pulsar sobre el cuadro de la izquierda y automáticamente se volverá verde. Además, el nombre del producto cambiará su aspecto y se mostrará tachado.

Llegados a este punto, al autor le gustaría señalar que la lista posee una funcionalidad de ocultación de herramientas superiores para mejorar la visibilidad, de tal forma que si existen elementos suficientes y el usuario hace scroll hacia abajo en la vista, el panel superior que contiene las acciones “*Borrar todo*”, “*Compartir*”, y “*Mis amigos*” se ocultará poco a poco hasta desaparecer completamente.

En la siguiente ilustración se muestra un ejemplo en el que un usuario ha marcado un producto “*Una sandía*” y posteriormente ha realizado scroll hacia abajo, ocultando de esa manera el panel de herramientas del panel superior y mostrando el producto que quedaba restante al final de la lista, en este caso “*5 kg de patatas*”.

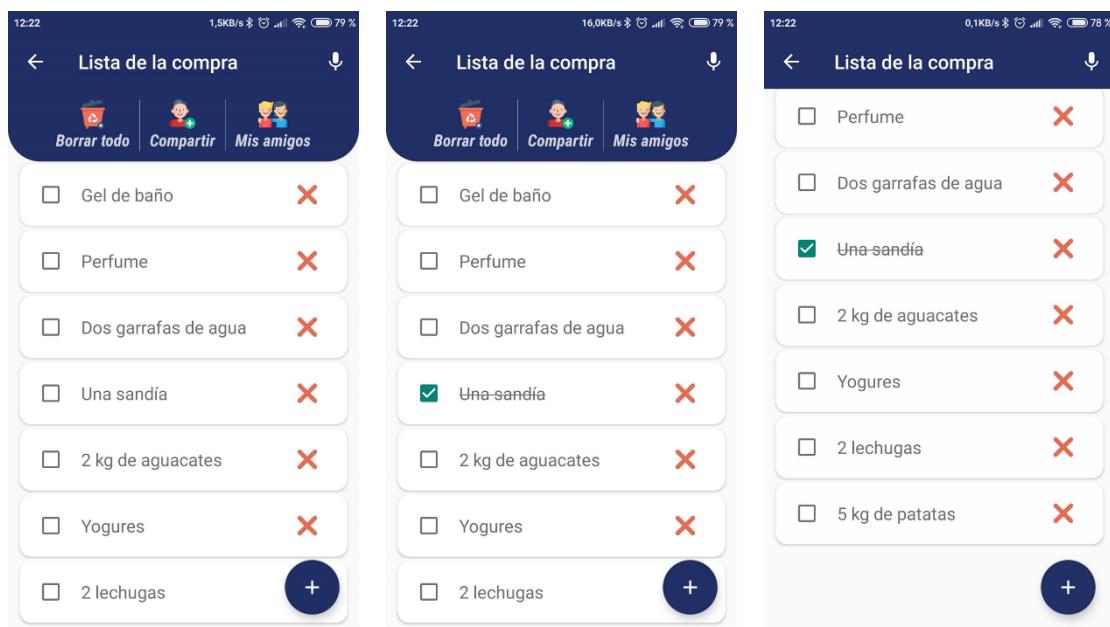


Ilustración 49 – Marcar un producto y ocultación del panel superior

BORRAR TODO – ACCIÓN RÁPIDA

Dentro de la propia lista de la compra se ofrece la misma funcionalidad de borrar todos los productos que fue mostrada anteriormente, activada mediante la pulsación en el botón “Borrar todo”. Se comporta exactamente igual, si el usuario pulsa sobre el botón, a continuación, se le mostrará un panel en el que tendrá que confirmar la acción si desea borrarlo todo. Si la acepta, la lista se vaciará por completo y desaparecerá el botón “Borrar todo” ya que la lista estará vacía.

La siguiente imagen ejemplifica la acción expuesta anteriormente.

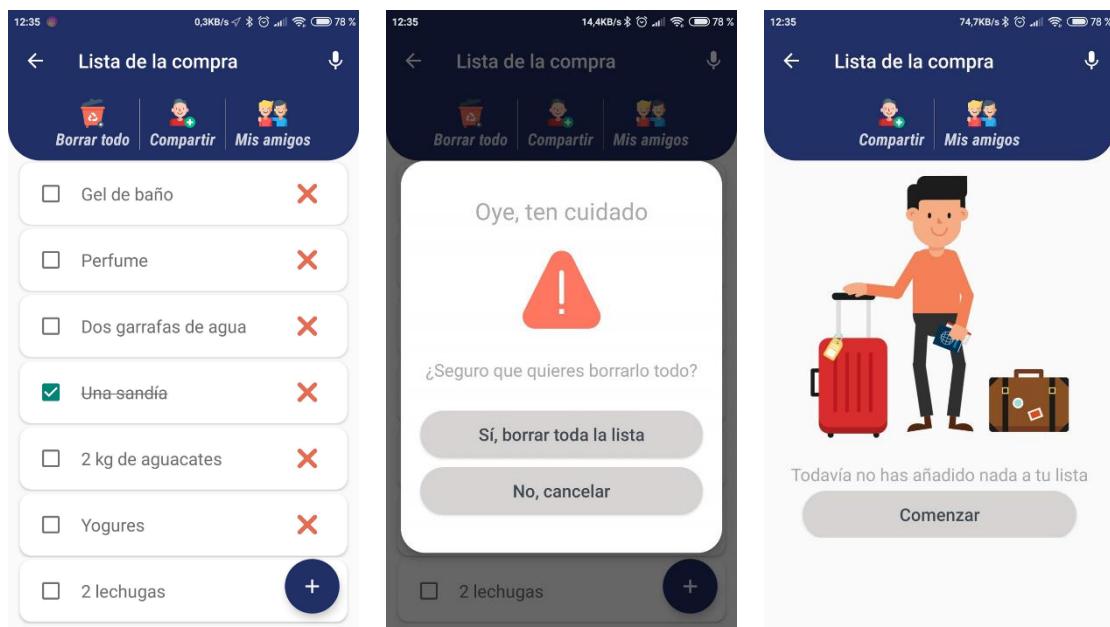


Ilustración 50 – Borrar todo mediante acción rápida

COMPARTIR LISTA – AÑADIR ACCESO

Si el usuario lo estima oportuno, se le ofrece la posibilidad de poder compartir su lista de la compra con otro usuario de la aplicación, para ello solo tendrá que realizar los pasos que se muestran en la siguiente ilustración.

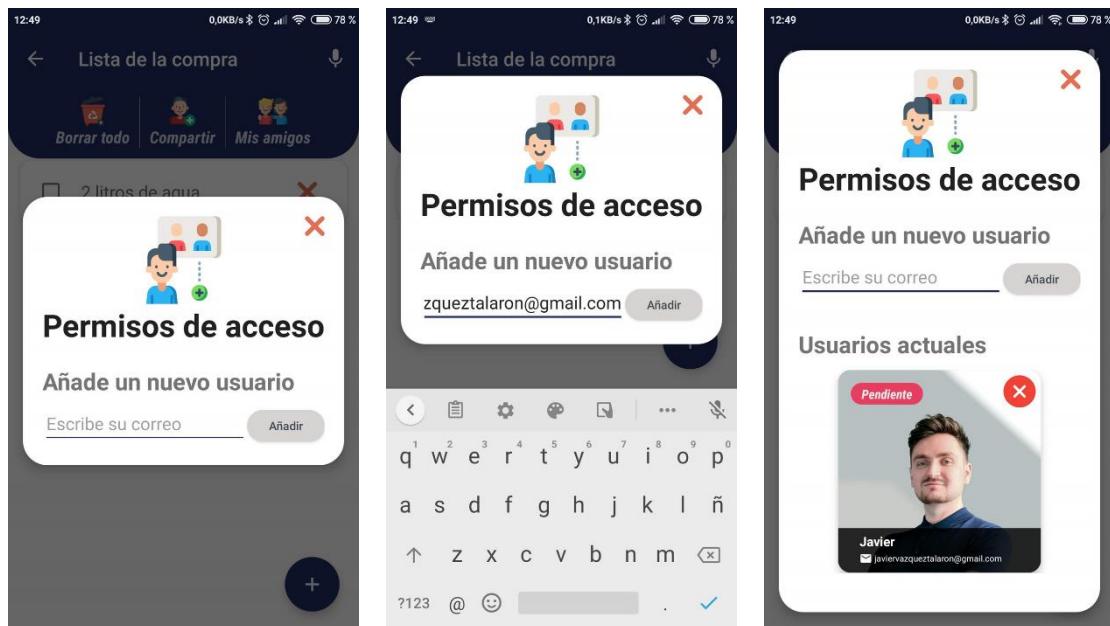


Ilustración 51 – Añadir acceso a un usuario

El usuario deberá pulsar sobre el botón “Compartir” tal y como aparece en la ilustración 48 y automáticamente le aparecerá una nueva ventana de diálogo, como en la primera imagen de la ilustración 51. Si desea cerrar este dialogo solo deberá pulsar en la cruz.

El usuario introducirá el correo de la persona a la que desea invitar y si esta se encuentra registrada en el sistema, en la parte inferior se creará una nueva vista con la imagen, el nombre y el correo del invitado, además de una insignia indicativa “Pendiente” en el caso de que el usuario aún no haya aceptado la invitación. Si ya existía algún invitado en la lista, simplemente se añadirá en nuevo cuadro de contenido a la derecha del último usuario invitado (lista horizontal desplazable).

Para todos los correos que introduzca el usuario, se le notificara con un mensaje emergente en la pantalla el resultado de la acción, ya que el correo puede no existir o el usuario puede introducir su propio correo por error.

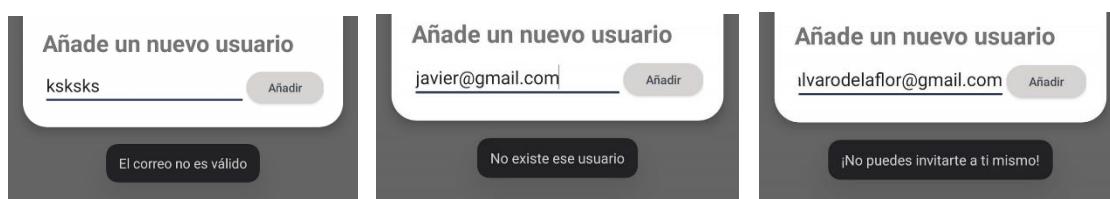


Ilustración 52 – Errores al invitar a un usuario

Cuando el correo si es válido y se realiza la invitación, además de realizar todo lo que se mostró anteriormente, también se creará un mensaje emergente como en la ilustración 52 para facilitar la labor de TalkBack (aplicación en modo adaptado).

COMPARTIR LISTA – ELIMINAR ACCESO

Si el usuario lo que desea es eliminar el acceso a todos y convertir su lista en privada puede realizarlo rápidamente desde la pantalla home pulsando en el botón “Cancelar acceso” y seguir los pasos detallados anteriormente.

Sin embargo, si lo que desea es cancelar el acceso a un usuario en concreto lo que deberá hacer es pulsar sobre el botón “Compartir”. *Seguidamente* se le abrirá una nueva ventana donde, en la parte inferior mediante desplazamientos horizontales deberá buscar el usuario al que le quiere retirar el acceso. Una vez localizado, pulsará en el botón superior derecho de la foto de perfil del usuario (círculo rojo redondo con una “X” blanca en su interior) y acto seguido se le borrará instantáneamente la invitación, desapareciendo del listado.

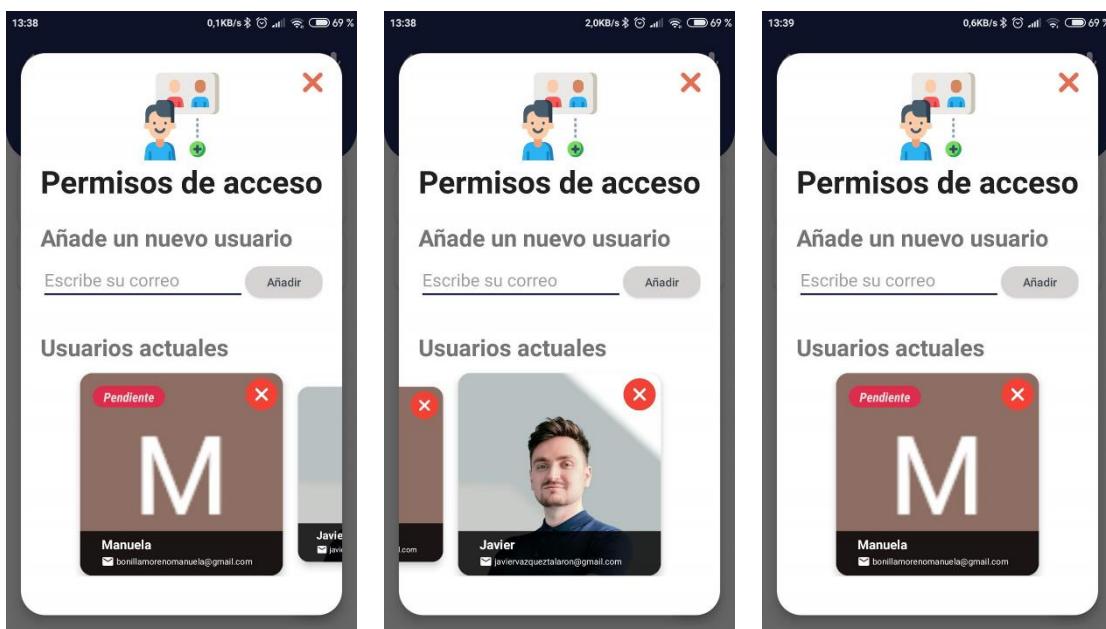


Ilustración 53 - Proceso de eliminar invitación

La ilustración anterior muestra el proceso que deberá seguir el usuario para eliminar una invitación.

5.3 Gestión de lista de la compra compartida

LISTAR INVITACIONES

Cuando el usuario envía una notificación es recibida por el usuario invitado, que tendrá que aceptarla o rechazarla en función de sus intereses.

Existen dos formas de acceder a la vista de las invitaciones:

- A través de la vista home, pulsando en la celda indicativa del número de invitaciones, por ejemplo, en el caso de la ilustración 43 habría que pulsar donde se indica “2 Invitaciones”.
- A través de la vista de la lista de la compra pulsando en el botón “Mis Amigos”, el cual se puede apreciar en la primera captura de la ilustración 50.

ACEPTAR O RECHAZAR INVITACIONES

Una vez en la vista de las invitaciones recibidas, para aceptar o rechazar una invitación habrá que realizar el siguiente procedimiento.

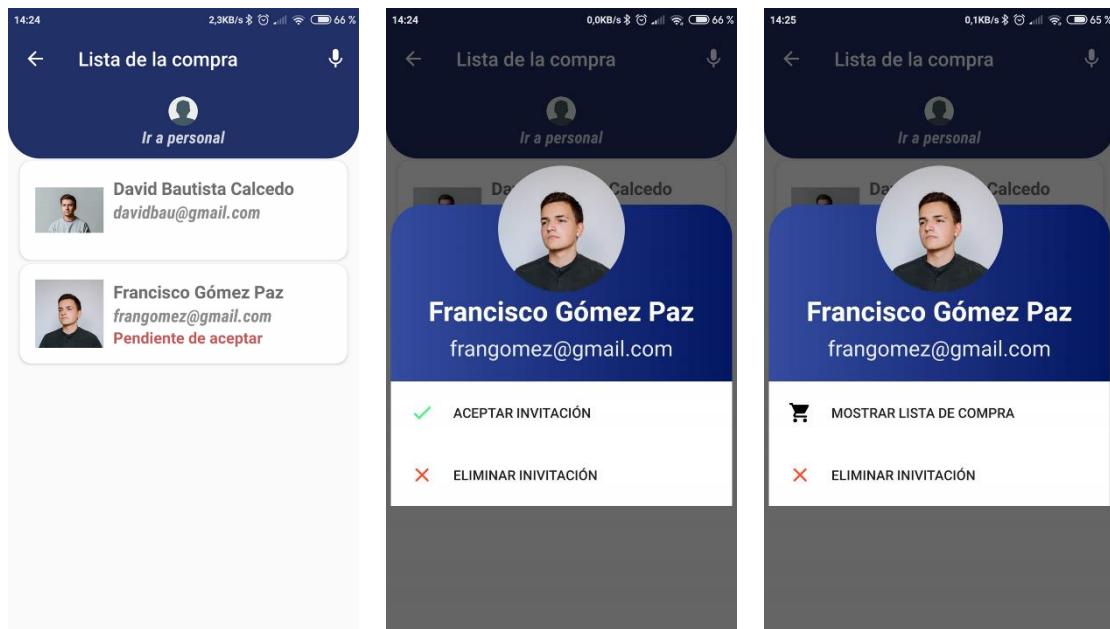


Ilustración 54 - Proceso de aceptar o rechazar invitación

En la vista, se mostrará un listado vertical con todas las invitaciones (en el caso de que haya muchas se habitará una funcionalidad scroll donde el cuadro que contiene el botón “Ir a personal” desaparecerá a medida que el usuario avanza). Una vez localizada la invitación deseada el usuario pulsará sobre ella y se le ofrecerá una nueva ventana de diálogo. En ella tendrá que seleccionar si aceptar la invitación pulsando sobre “ACEPTAR INVITACIÓN” o rechazarla pulsando sobre el botón “ELIMINAR INVITACIÓN”. En el caso de que sea rechazada se cerrará el diálogo y la invitación desaparecerá del listado. Por otro lado, en el caso de que sea aceptada el botón “ACEPTAR INVITACIÓN” desaparecerá y será sustituido por el botón “MOSTRAR LISTA DE COMPRA” el cual si es pulsado lleva a la lista de la compra personal del usuario que envía la petición.

En el caso de que el usuario acepte la invitación en el listado desaparecerá el texto rojo “Pendiente de aceptar” que se puede ver en la ilustración 54. Por otro lado, al usuario propietario de la lista le desaparecerá el logotipo “Pendiente” que aparece en la ilustración número 53.

Cabe destacar que cada vez que el usuario receptor gestiona una invitación (aceptar/rechazar/abrir lista) el sistema comprueba que esa invitación siga vigente, para proteger los casos en los que el usuario desactiva la conexión a internet para mantener activa la invitación.

Posteriormente, en la sección de pruebas se especificarán casos donde este sistema evita situaciones de conflicto adicionales.

Por último, respecto al listado de invitaciones, señalar que existe un botón “*Ir a personal*” el cual si es pulsado redirecciona al usuario a su lista de la compra personal, funcionalidad que fue detallada al completo en el punto 6.2.

USO DE LISTA DE LA COMPRA COMPARTIDA

Como anteriormente fue descrito, para abrir la lista de la compra (de una invitación) que el usuario desee, bastará con buscar la invitación correspondiente en el listado y posteriormente hacer clic en el botón del diálogo “*MOSTRAR LISTA DE COMPRA*”.

Una vez dentro, se ofrecerá una vista y como la siguiente.

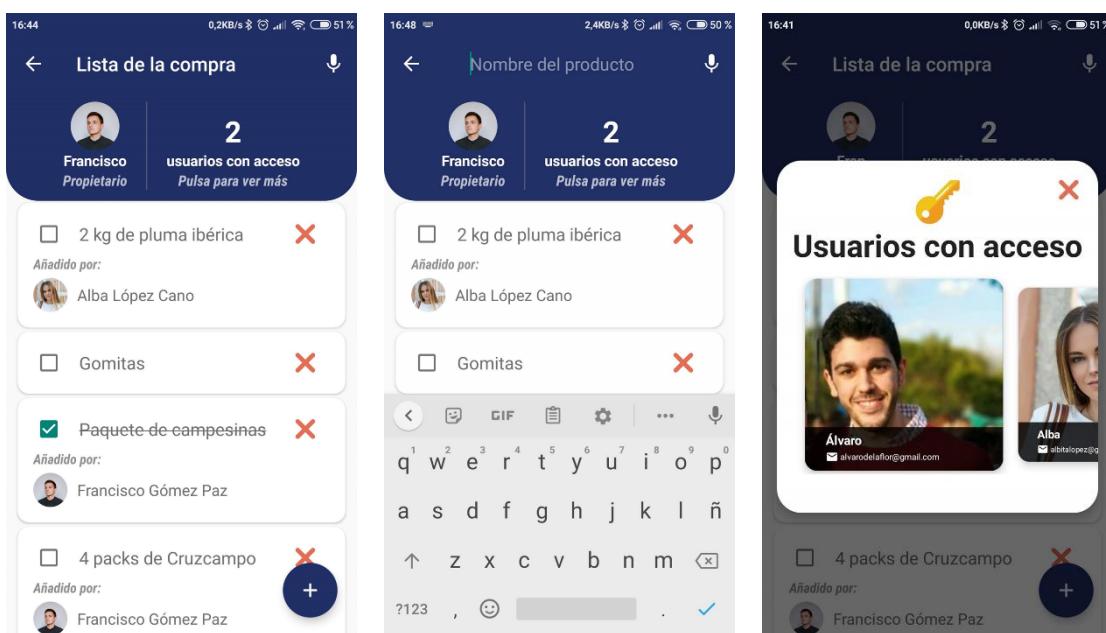


Ilustración 55 – Lista compartida

El comportamiento de la interfaz de una lista que le han compartido a un usuario es exactamente el mismo que el de una lista personal salvo por dos diferencias fundamentales.

La primera de ellas es que el encabezado superior cambia por completo, en esta ocasión solo se muestra la información del propietario y un botón para poder ver los usuarios que cuentan con acceso a la lista.

La segunda diferencia radica en que un usuario invitado no puede modificar las personas que disponen de acceso a esa lista, únicamente puede listar de manera horizontal los usuarios que tienen acceso a la misma, como aparece en la última captura de la ilustración 55.

Al autor le gustaría destacar (ya que en el punto 6.2 no fue nombrado) que tanto en la interfaz de visualización de una lista de la compra personal como en la de visualización de una lista a la que se accede mediante invitación, para cada producto que aparezca se añadirá una información adicional de quién es la persona que ha insertado ese producto en los casos en los que la persona que se encuentre “logueada” sea distinta a la que lo añadió.

Será exemplificado a través de la siguiente ilustración en el que se muestra la vista de la misma lista de la compra, desde la perspectiva del dueño de esta y desde la de un usuario invitado a ella.

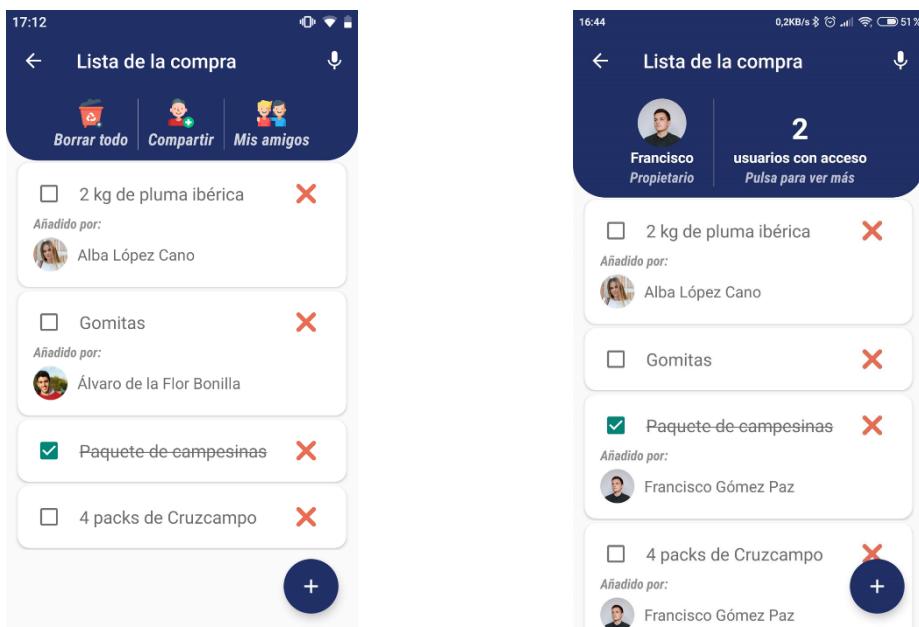


Ilustración 56 – Ejemplo de lista compartida

A la izquierda se muestra una captura de un dispositivo donde Francisco Gómez Paz, el dueño de la lista, ha iniciado sesión en la aplicación. Francisco ha añadido en la lista dos productos (“Paquete de campesinas” y “4 packs de Cruzcampo”). En su interfaz, ya que él ha sido quien ha añadido estos dos productos no le aparecerá ninguna información adicional más. Sin embargo, para los otros dos productos restantes, al haber sido añadidos por usuarios distintos a él, la aplicación le indicara a Francisco por cada producto de ellos que usuario lo ha añadido a la lista.

A la derecha se muestra una captura de un dispositivo donde Álvaro de la Flor Bonilla, usuario al que Francisco ha invitado a su lista, ha iniciado sesión en la aplicación. Como Álvaro ha sido quien ha añadido el producto “Gomitas” no se le muestra ninguna información adicional, sin embargo, para el resto de los productos la aplicación si le indica quién ha sido el usuario que los ha añadido a la lista.

Destacar una vez más que toda la funcionalidad que ha sido mostrada hasta el momento (así como la que queda por mostrar) trabaja en tiempo real, lo cual quiere decir por ejemplo en este caso que multitud de dispositivos pueden añadir, marcar o borrar productos simultáneamente mientras que las actualizaciones serán instantáneas entre todas las cuentas. Es decir, si Álvaro añade un producto, Francisco lo verá reflejado al momento en su lista.

Finalmente, la forma en la que se añaden, eliminan o marcan los productos no serán explicadas de nuevo, ya que su comportamiento es exactamente igual al detallado en el apartado 6.2 de este documento.

5.4 Gestión de identificación

Para este apartado será necesario utilizar por primera vez la barra navegación.

IDENTIFICAR PRODUCTOS – NFC ACTIVADO

En este bloque procederemos a explicar los pasos que tendrá que seguir un usuario en la aplicación para identificar un producto en el caso de que el NFC de su dispositivo se encuentre activado.

Suponiendo que el usuario se encuentre en la vista *HOME* deberá pulsar en la barra de navegación el ícono más a la derecha, el que se representa con una brújula. También podrá desplazarse hacia esa vista desplazando el dedo desde el borde de la pantalla hacia la izquierda dos veces, como si cada vista se tratara de una foto en la galería.

Una vez dentro de la vista de identificación (imagen central de la siguiente ilustración 57) deberá pulsar en el botón “*Escanear*” y automáticamente se le abrirá un diálogo indicando que ya puede acercar un dispositivo para que sea identificado. Si desea cancelar esta identificación solo deberá pulsar en el botón de cancelar.

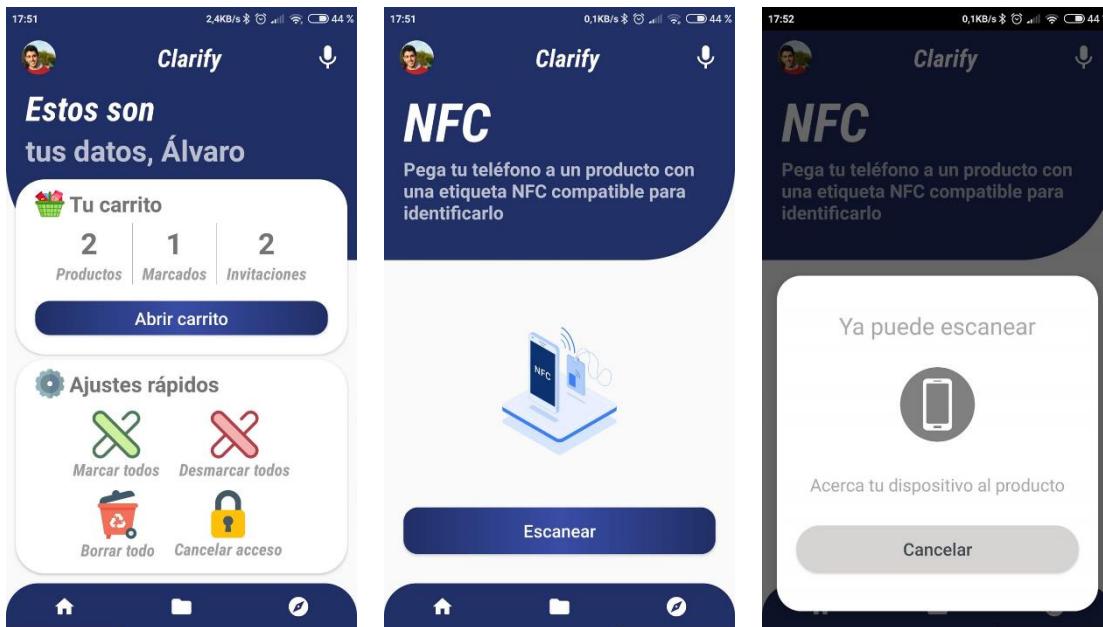


Ilustración 57 – Activar identificación

Una vez se encuentre el dispositivo preparado (tercera imagen de la ilustración anterior) bastará con que el usuario acerque su dispositivo a una etiqueta válida.

En el caso de que sea leída una etiqueta no válida o haya ocurrido algún error el sistema se lo notificará al usuario y le permitirá realizar una nueva lectura.

Si la etiqueta es válida el dialogo de la última ilustración se cerrará y aparecerá uno nuevo con la información más importante del producto. En esta nueva vista además se le permite realizar al usuario dos funciones principales.

- **Añadir el producto al almacén personal:** funcionalidad que será detallada en el próximo apartado.
- **Añadir el producto a la lista de la compra personal:** tal y como si un usuario la añadiera manualmente a su lista como se vio en el apartado 6.2. Como nombre utilizará el modelo del producto y la marca.

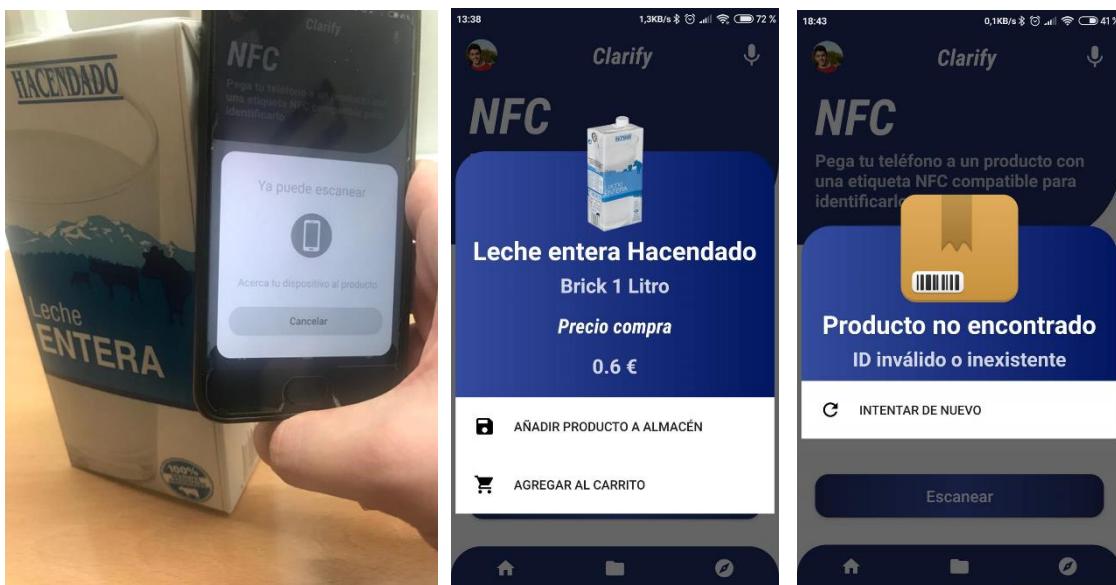


Ilustración 58 – Proceso de identificar un producto

Como se indicó anteriormente, para añadir un producto en el almacén personal bastará con pulsar en el botón “**AÑADIR PRODUCTO A ALMACÉN**”. Este producto se añadirá al almacén que establece la etiqueta por defecto, para facilitar este paso a las personas con discapacidad visual.

IDENTIFICAR PRODUCTOS – NFC ACTIVADO

Este caso representa la situación en la que el dispositivo del usuario dispone de sensor NFC, pero se encuentra desactivado por el usuario.

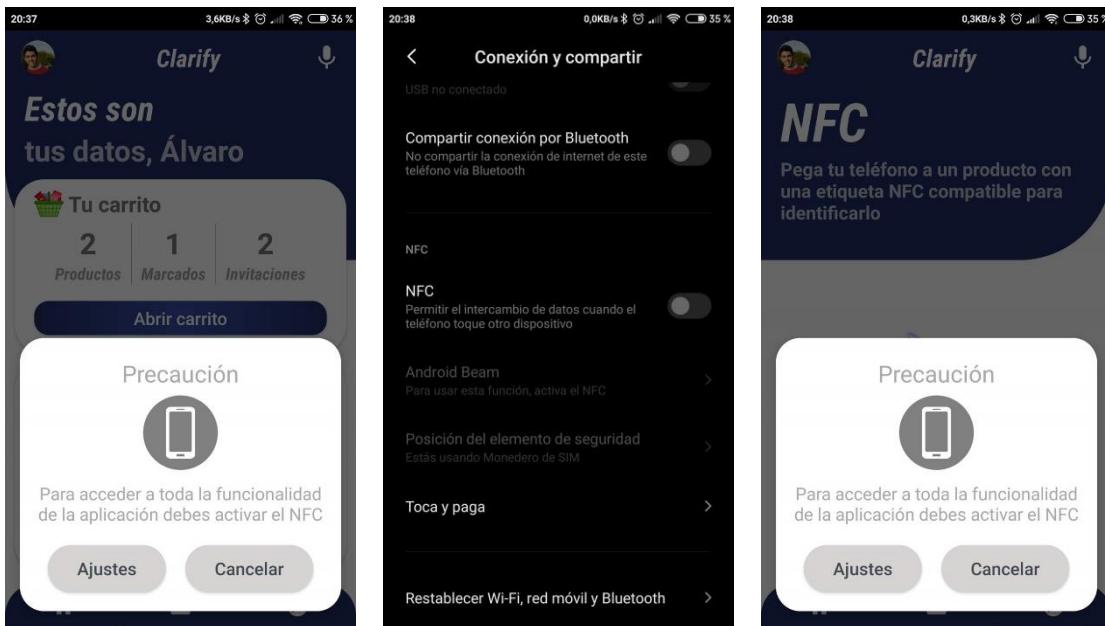


Ilustración 59 – Detección de NFC desactivado

Nada más abrir la aplicación, se le solicitará al usuario que active el sensor NFC. Pulsando en el botón “**ACTIVAR**” se llevará al usuario a la vista exacta de ajustes donde activar el NFC. Una vez activado podrá usar la aplicación con normalidad.

Por otro lado, si el usuario decide cancelar esta activación le dará al botón “CANCELAR”, pero este volverá a aparecer cada vez que intente usar alguna funcionalidad que requiera del sensor NFC (tercera captura de la ilustración 59).

IDENTIFICAR PRODUCTOS – DISPOSITIVO SIN NFC

En el caso de que el dispositivo no disponga de NFC, no podrá activar ninguna función que lo requiera. No aparecerá ningún mensaje como en el caso anterior, pero si intentará activar alguna función que necesite NFC aparecería una notificación emergente notificando el problema.



Ilustración 60 – Dispositivo sin NFC

5.5 Gestión de productos

En esta sección se abordará como tratar los productos una vez han sido añadidos al almacén personal.

MOSTRAR LISTADO

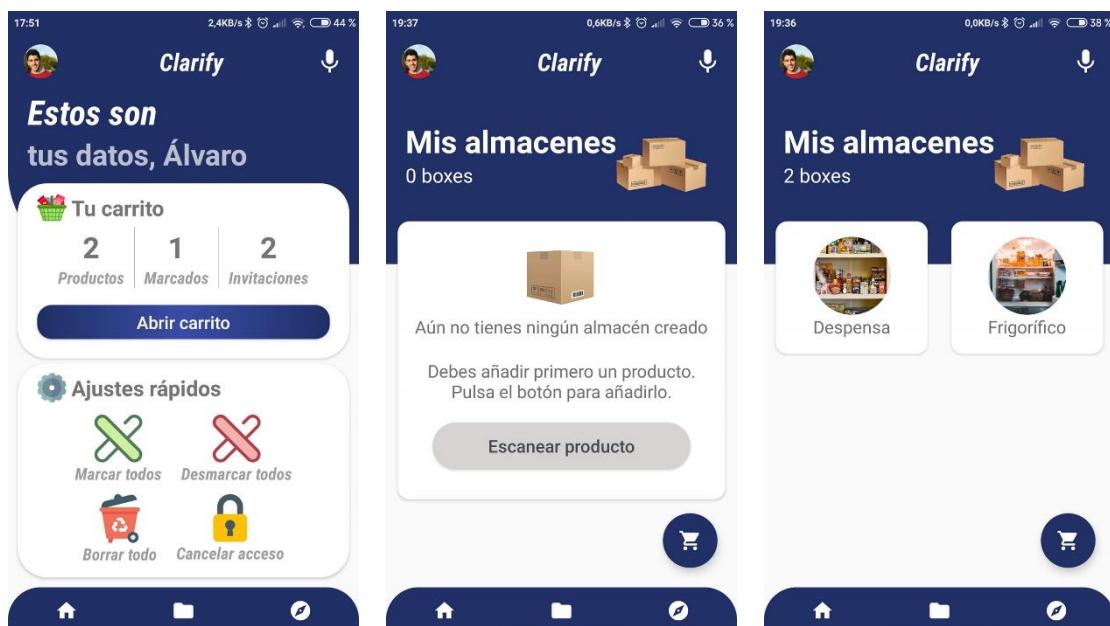


Ilustración 61 – Almacén de productos personales

Para acceder a esta vista el usuario tendrá que pulsar, en la barra de navegación, en el ícono de la vista que está representada con una carpeta, o bien desplazando el dedo del borde derecho al centro de la pantalla (si se encuentra en la vista home).

Existen dos posibles variantes de la vista principal de este bloque, en función de si existe algún almacén o no.

En el caso de que no exista ningún almacén creado, se le mostrará una vista explicativa al usuario de la situación como la imagen central de la ilustración 61. En esta habrá un

botón que enviará al usuario directamente a la vista de identificación de un producto, tal y como se mostró en el apartado 6.4.

Por otro lado, en el caso de que si exista al menos un almacén se mostrará una vista similar a la tercera imagen de la ilustración 61. Para abrir un almacén bastará con que el usuario pulse el cuadro del almacén que quiere abrir.

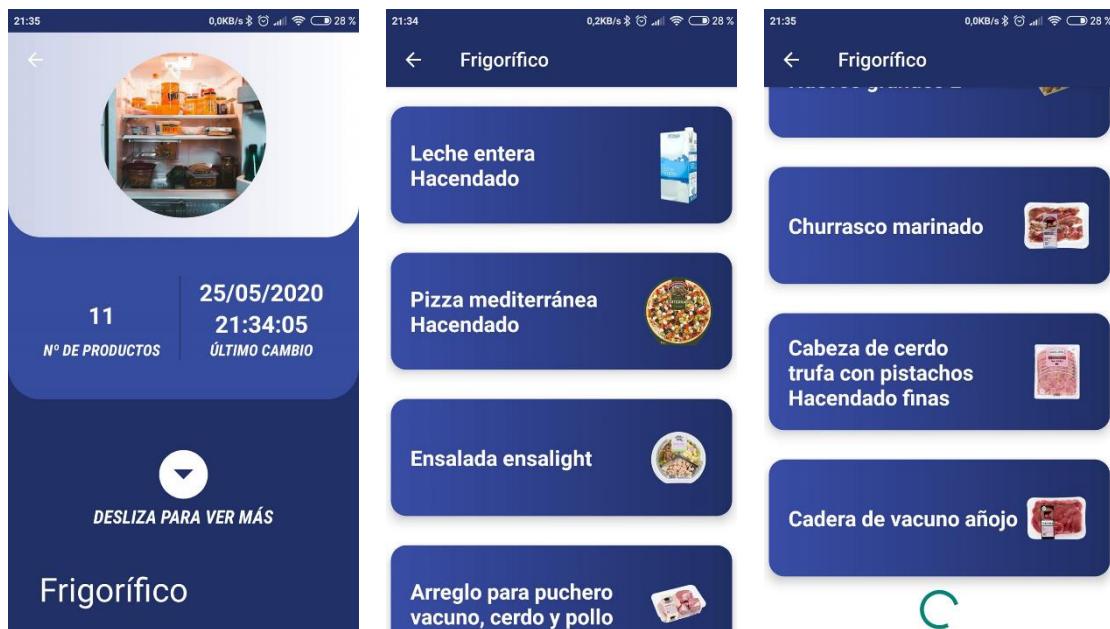


Ilustración 62 – Mostrar almacén

En la ilustración anterior se muestra el proceso para ver un almacén. En primer lugar, se le mostrará un resumen del almacén en sí (número de productos y fecha del último cambio).

Si desliza hacia arriba esta vista o pulsa sobre el botón “DESLIZA PARA VER MÁS”, se le mostrará el listado completo de todos los productos que se han guardado en este almacén. Este listado se encuentra paginado cada diez elementos, por lo que una vez llegado al final de este aparecerá un “span” como el de la tercera imagen de la ilustración 62.

ACCIONES SOBRE PRODUCTO

Respecto a los productos que ya se encuentran en alguno de los almacenes de los que dispone el usuario, si se pulsa sobre alguno de ellos se desplegará una ventana de diálogo que posibilita al usuario realizar varias acciones sobre este (véase siguiente ilustración, ilustración número 63). Estas dos opciones son:

- **Borrar:** el producto será eliminado de la aplicación y consecuentemente del almacén que lo contiene.
- **Agregar al carrito:** se añadirá un nuevo producto a la lista de la compra del usuario que tiene la sesión iniciada en el sistema. El nombre de este producto estará compuesto por el nombre completo, por ejemplo, para la ilustración 63 se añadirá un producto a la lista de la compra con el nombre “Cadera de vacuno añojo – Bandeja 530 g aprox.”.

En el caso de que el usuario elimine todos los productos le aparecerá una vista como se refleja en la imagen central de la ilustración 63. El usuario puede decidir dejar el almacén vacío para que se pueda ver la hora en la que eliminó el último producto. Por otro lado, si decide borrarlo pulsando en el botón “Borrar este almacén”, el almacén se eliminará y se trasladará al usuario a la vista del listado de todos los almacenes donde ya no aparecerá el almacén eliminado.

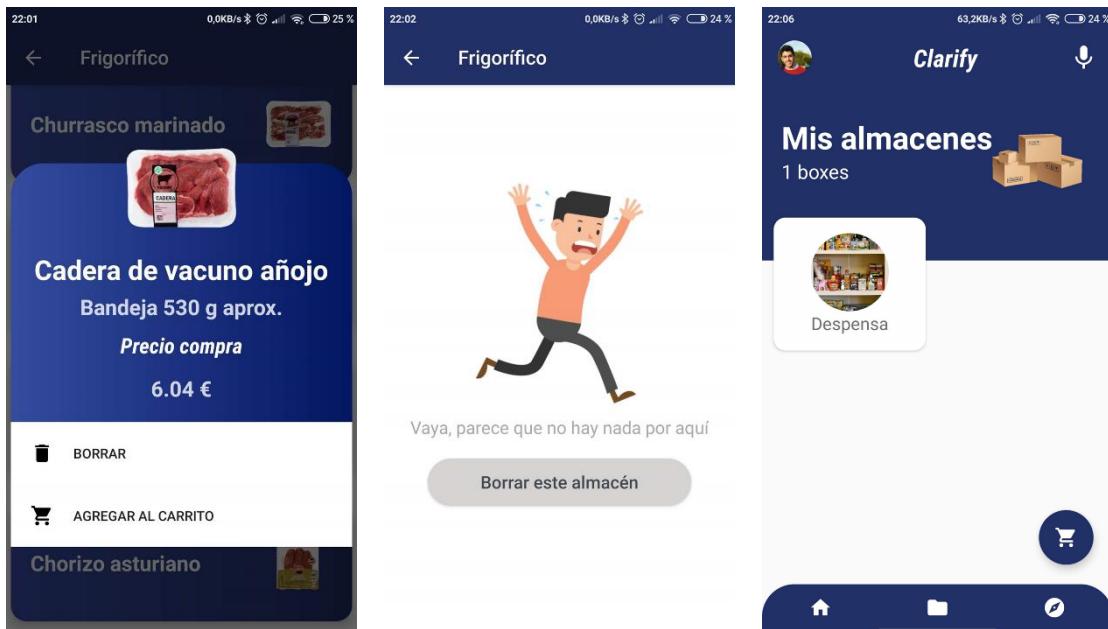


Ilustración 63 – Eliminar todos los productos de un almacén

Como se puede apreciar en la última imagen de la ilustración 63 se ha añadido otro botón de acceso directo a la lista de la compra desde esta vista.

5.6 Gestión de notificaciones

Para poder ofrecer a los usuarios alertas ante nuevos eventos en la aplicación, a pesar de que esta no se encuentre abierta en primer plano, se ha desarrollado un sistema de alertas push.

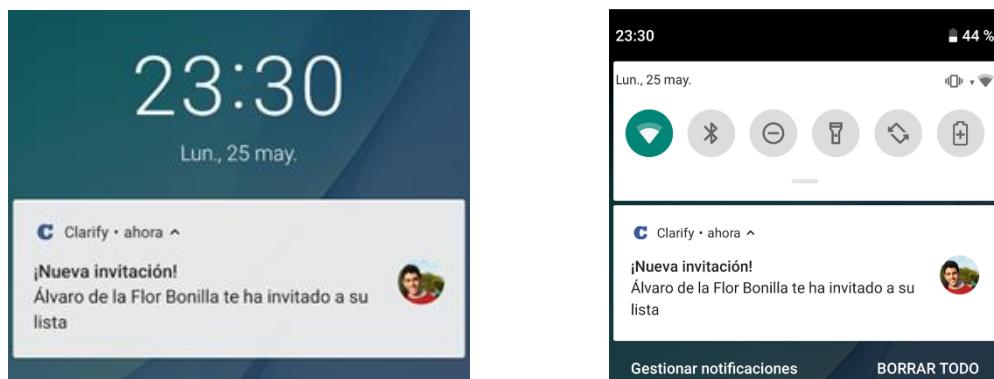


Ilustración 64 - Ejemplos de notificaciones

En la ilustración anterior se muestra un mismo ejemplo desde dos situaciones distintas. La primera representa como se recibiría con la pantalla de dispositivo bloqueada y la segunda con el dispositivo desbloqueado en la propia barra de notificaciones.

El usuario solo tendrá que pulsar dos veces sobre la notificación, esta desaparecerá y se le redireccionará posteriormente a la vista que requiere de la acción del usuario.

Por ejemplo, en la situación mostrada en la ilustración 64, la aplicación dirigirá al usuario al pulsar a la lista de invitaciones que ha recibido.

5.7 Gestión de control por voz

Toda la funcionalidad que se ha presentado anteriormente se encuentra completamente adaptada para su uso con TalkBack. Además, se ha añadido un botón de control por voz de tal forma que el usuario solo tendrá que pulsar sobre este, decir su orden en lenguaje natural y la aplicación lo realizará.

Todo el tutorial que será realizado en este apartado es preferible usarlo con la funcionalidad TalkBack activada, para ello, el lector debe dirigirse a (la localización puede variar dependiendo del dispositivo):

Ajustes > Ajustes Adicionales > Accesibilidad > TalkBack > Activar

El funcionamiento de TalkBack es muy sencillo. El usuario desplaza el dedo por la pantalla y el sistema lee la zona por la que se desplaza.

En el caso de los botones, por ejemplo, si el usuario desea activarlo (tras que TalkBack le haya leído el contenido e indicado que es un botón) si el usuario pulsa dos veces la pantalla en cualquier parte, no importa que sea no sobre el botón, la acción se realizará.

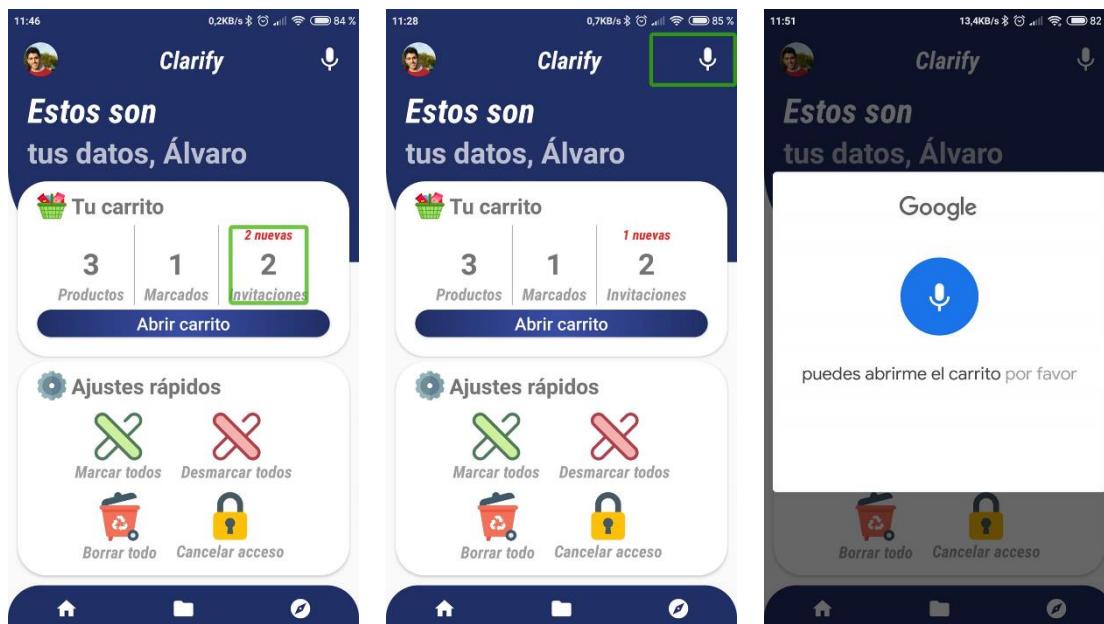


Ilustración 65 – Uso de TalkBack y comandos de voz

Utilícese la ilustración 65 para un ejemplo práctico. TalkBack señala con un recuadro verde todo lo que se dispone a leer lo que significa que para la primera captura el usuario ha pasado su dedo por la zona del recuadro verde y automáticamente el sistema le ha transcripto “2 nuevas, 2 invitaciones”. Si el usuario pulsa dos veces en cualquier parte de la pantalla se le dirigirá automáticamente al listado de invitaciones.

Un ejemplo más claro puede ser el botón “*Marcar todos*”. Si el usuario pasa su dedo por la zona en la que se encuentra este botón y tras la lectura de TalkBack realiza una doble pulsación en cualquier parte de la pantalla, sea cual sea, automáticamente se marcarán todos los productos de la cesta.

Además, en todas las vistas se ha añadido un botón representado con un micrófono que una vez pulsado activa los comandos de voz. Para activarlo se continúa con la dinámica de la ocasión anterior. El usuario pasa su dedo por la parte superior de la pantalla y cuando escuche “*Activar comandos de voz*” hará una doble pulsación en la pantalla (cualquier parte de ella es válida). Instantáneamente se abrirá un diálogo en el que se mostrará todo lo que el usuario dicta. El sistema es capaz de detectar cuando el usuario comienza y termina de hablar por lo que salvo las instrucciones anteriores no tendrá que realizar ningún toque en la pantalla adicional.

La segunda y tercera captura de la ilustración 65 representa como un usuario activa los comandos de voz y le pide al sistema que le muestre su carrito.

Todo el mecanismo de utilización de comandos de voz ha sido entrenado de tal forma que permite la utilización de un lenguaje natural. Para la acción que se representa en la tercera captura de la ilustración 65 son admitidos comandos como “*Quiero abrir mi lista de la compra*”, “*Ábreme mi lista*”, “*Quiero ver mis productos*”, etc.

En el caso de las acciones rápidas de la pantalla inicial se utilizarán para todos los casos una metodología similar. Utilicemos el caso de desmarcar todos los productos.

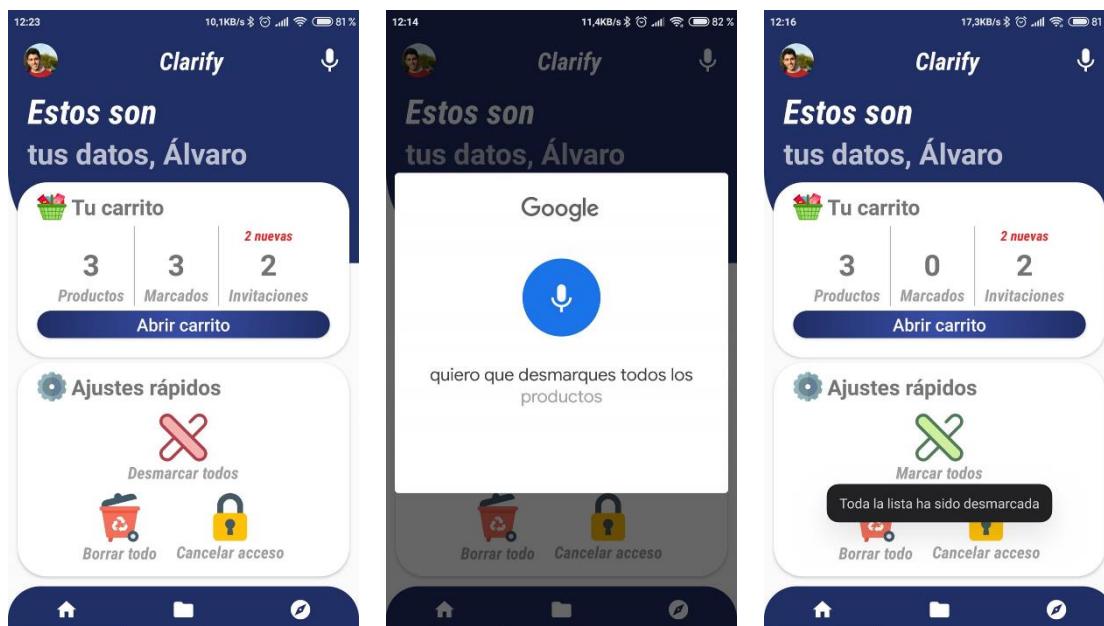


Ilustración 66 – Desmarcar todos los productos mediante comandos de voz

Como puede apreciarse en la ilustración anterior el usuario puede utilizar un lenguaje natural para ejecutar los comandos de voz. Una vez sean ejecutadas se notificará mediante una notificación emergente, que al tener activado TalkBack, será transcrita en voz alta por la aplicación. Para los casos “*Borrar todo*”, “*Marcar todo*” y “*Cancelar acceso*” se utiliza una dinámica homóloga.

LISTA DE LA COMPRA

Como anteriormente fue descrito, para abrir la lista de la compra bastará con que el usuario diga cualquier variante a “Abrir carrito”, “Ver productos”, “Ver marcados”, etc.

- **Listar los productos**

Para listar los productos el usuario simplemente pasará el dedo a lo largo de la lista. Por cada celda, el sistema le dictará su contenido y, además, le dirá su posición relativa en la lista. Por ejemplo, si en una lista de ocho elementos el primero de ellos es “Botella de agua” y se encuentra marcada, el sistema dirá:

“Botella de agua, marcada. Elemento en la posición 1 de 8”

Para hacer scroll en la pantalla, si se encuentra TalkBack activado habrá que realizar el mismo gesto que normalmente, pero utilizando dos dedos en lugar de uno.

- **Añadir productos**

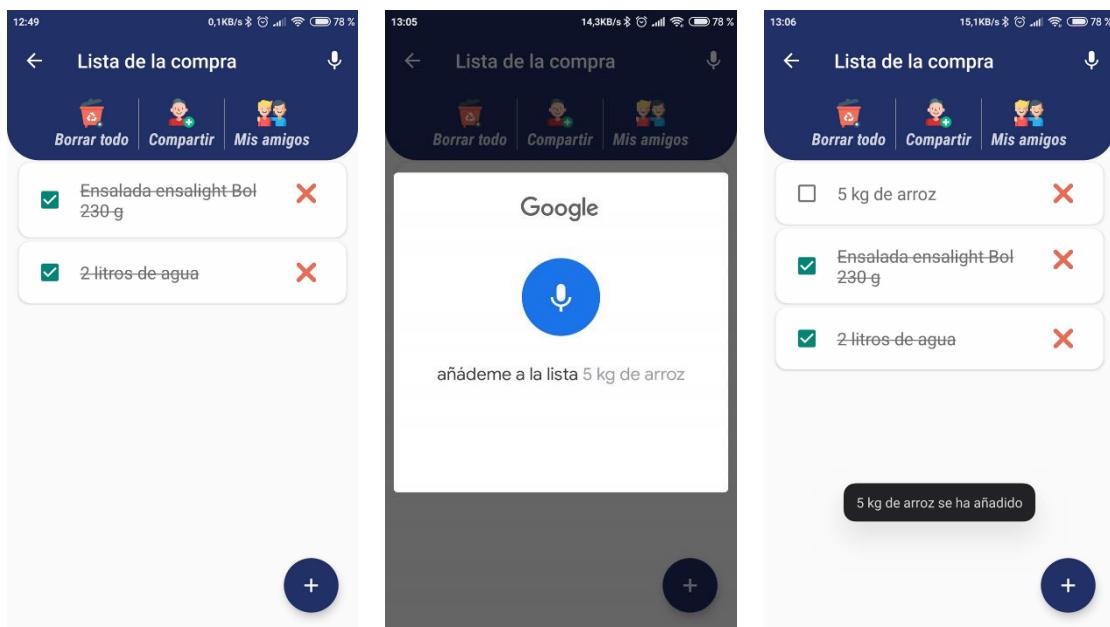


Ilustración 67 – Añadir producto a la lista

Para añadir un producto a la lista el usuario debe activar los comandos de voz como se indicó anteriormente y decir cualquier variación del verbo “añadir”. El sistema se encargará de reconocer de entre todo lo que ha dicho el usuario que es el producto y eliminar todo lo restante (en el caso del ejemplo de la ilustración 67 el sistema ha eliminado “añádeme a la lista”).

- **Eliminar un producto**

Para eliminar un producto, gracias a que TalkBack indica la posición de los elementos en una lista, se proporciona al usuario dos formas distintas que posibilitan realizar esta acción.

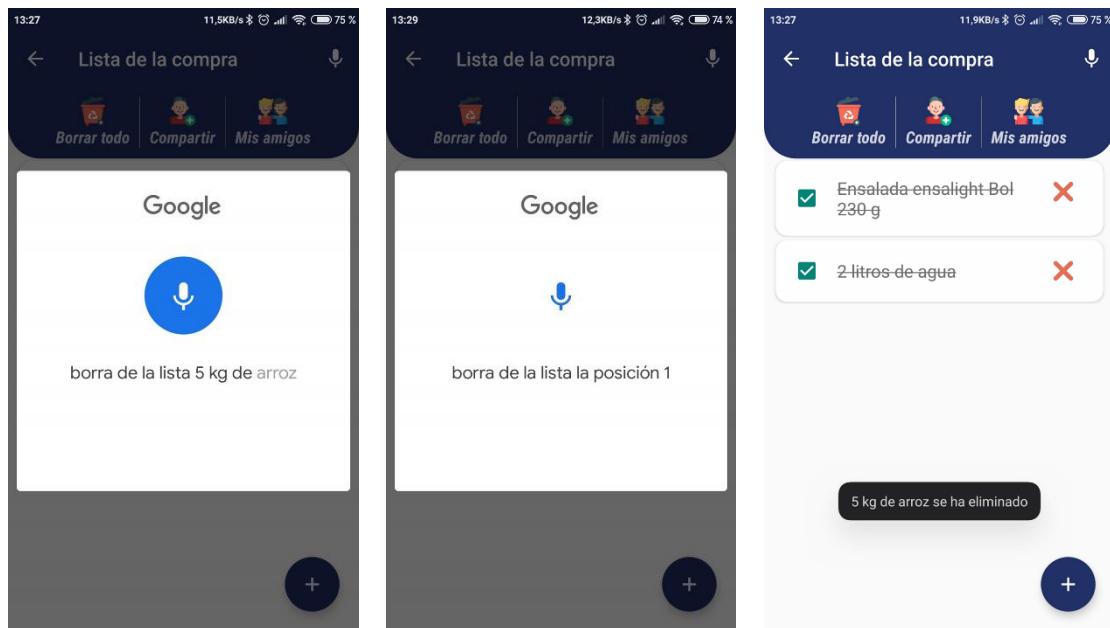


Ilustración 68 – Eliminar producto de la lista

La primera opción consiste en nombrar la orden borrar (cualquier variante de del verbo “borrar” es aceptado) y posteriormente el nombre del producto que se desea borrar. La aplicación buscará una coincidencia entre todos los productos y lo borrará. La segunda opción consiste en utilizar una vez más la orden borrar, pero en esta ocasión se indicará la posición que ocupa el elemento en la lista.

Para ambos casos, si el producto es eliminado se mostrará el nombre a través de un mensaje emergente en la pantalla.

- **Marcar/Desmarcar un producto**

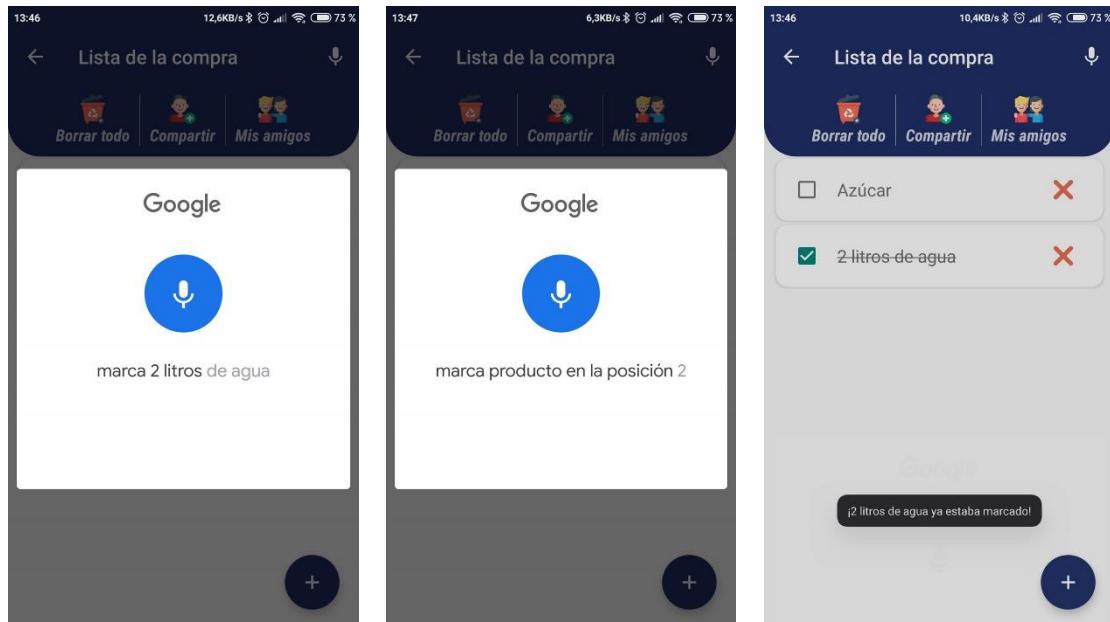


Ilustración 69 – Marcar un producto en la lista

Para marcar o desmarcar un producto se sigue exactamente el mismo procedimiento que para borrar un producto, pero en este caso los verbos a utilizar son “*marcar*” o “*desmarcar*” respectivamente, así como cualquiera de sus conjugaciones.

La opción de utilizar este modo indicando la posición también es válida.

Al igual que en el caso de borrar, si se ha producido algún error, se le notificará al usuario con una alerta en la pantalla. En la última captura de la ilustración 69 de muestra como se ha intentado marcar un producto que ya lo estaba.

Toda la funcionalidad del encabezado de la lista (borrar todo, compartir, listado de amigos y la propia lista compartida de alguna invitación) sigue la misma dinámica que ha sido explicada hasta el momento, es decir, lectura de la pantalla y doble clic para aceptar la opción.

IDENTIFICACIÓN DE PRODUCTOS

Para acceder a la vista de identificación mediante comandos de voz el usuario deberá realizar los siguientes pasos.

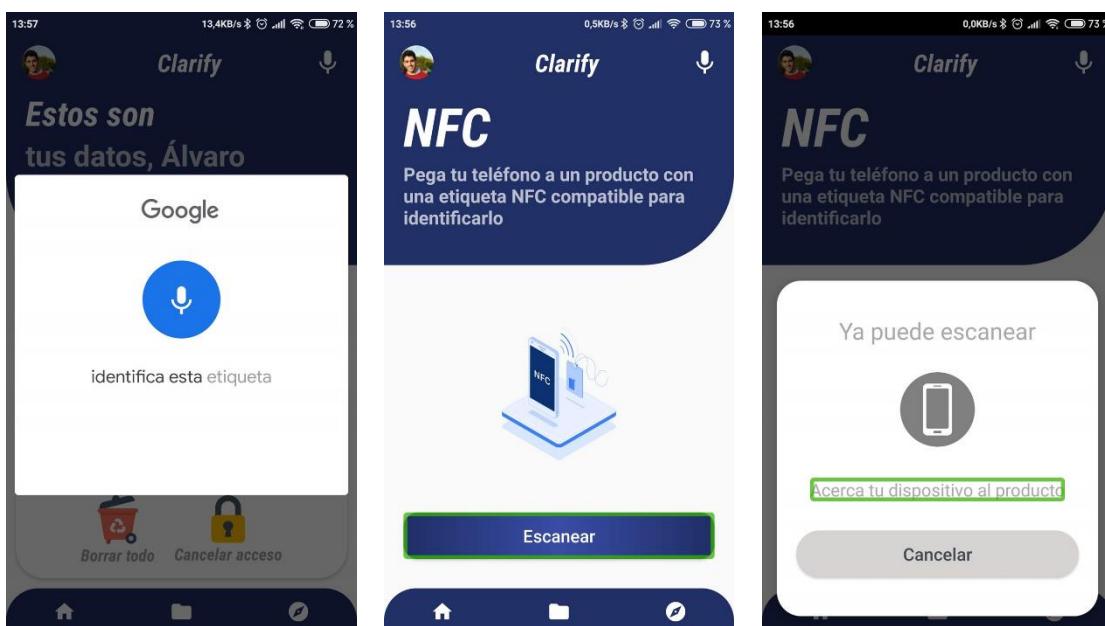


Ilustración 70 – Identificar un producto 1

Ordenar al sistema mediante comandos de voz que desea identificar una etiqueta o producto, automáticamente se abrirá la vista que permite realizar esta opción y TalkBack se dirigirá al botón que habilita la función de escanear. Bastará con hacer una doble pulsación en cualquier parte de la pantalla y el sistema se encontrará preparado para escanear una etiqueta.

Una vez una etiqueta ha sido escaneada satisfactoriamente el sistema leerá, como hemos indicado anteriormente todo lo que se encuentra en la pantalla y en este caso en específico, toda la información del producto que el usuario ha escaneado.

Como siempre, indicar que esta función de lectura solo se encuentra disponible si se ha activado TalkBack previamente.

GESTIÓN DE PRODUCTOS

Para abrir el listado de almacenes bastará con ordenar un comando del tipo “*Ábreme mis almacenes*”.

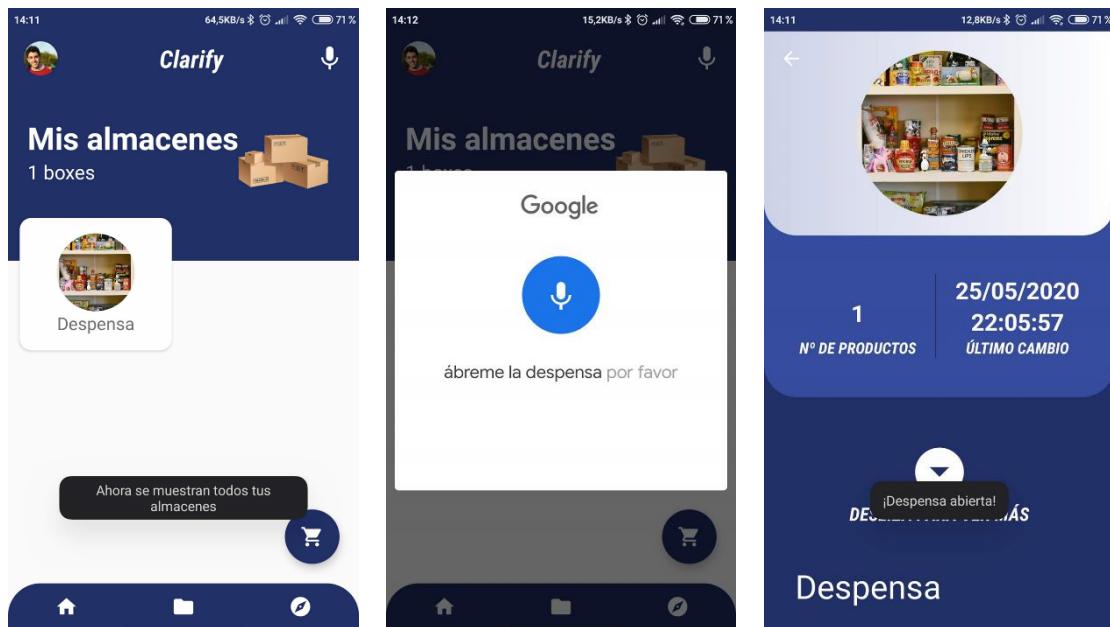


Ilustración 71 – Abrir almacén

Una vez abierto el listado, el usuario utilizará TalkBack para navegar por la lista. Cuando localice el almacén que desea abrir la única acción que debe realizar es activar las funciones de voz y nombrar el almacén que desea abrir. Automáticamente este aparecerá.

Una vez dentro del listado el usuario navegará por la aplicación exactamente igual que se ha explicado en ocasiones anteriores, con el uso de dictado por voz y doble toques para activar las distintas opciones, así como scroll a base de dos dedos en la pantalla.

6 PRUEBAS

En un proyecto software es habitual que aparezcan errores, los cuales suelen ir surgiendo conforme se avanza en el ciclo de vida del desarrollo. Sin embargo, pueden no llegarse a descubrir incluso ya lanzado el proyecto final.

Por ello, el proceso de pruebas debe conformar un pilar fundamental durante el transcurso de un producto software.

En este proyecto, se han llevado a cabo dos tipos de pruebas sobre la aplicación que se ha desarrollado: pruebas funcionales y pruebas de rendimiento.

6.1 Pruebas funcionales

El objetivo de las pruebas funcionales es el de proporcionar información sobre la calidad de la aplicación que se ha desarrollado. También pretende comprobar la funcionalidad del sistema, de tal forma que este devuelva los resultados esperados para cada uno de los casos diseñados.

En concreto en este proyecto, para las pruebas funcionales, el autor se ha centrado para su realización en las llamadas “**pruebas de aceptación del usuario**”. En ellas, se les solicitó a usuarios piloto que hiciesen un uso intensivo de la aplicación para que, una vez explicado el funcionamiento de esta y facilitarles un guion sobre el uso debían hacer.

Estos, por su parte, comunicaran los posibles errores detectados, o mejoras que en su opinión incrementarían el valor del producto. Este proceso se repitió en cada iteración.

Debido a las circunstancias derivadas del confinamiento, el plan de pilotaje inicial, consistente en un usuario con discapacidad visual, tuvo que ser anulado ya que le resultaba imposible la instalación y manejo de la aplicación en sus etapas iniciales sin una ayuda presencial.

Debido a este imprevisto se decidió variar el plan inicial y utilizar dos miembros externos que contaban con conocimiento de la tecnología Android. En concreto, por cada iteración (a partir del sprint 2), se entregó al usuario una versión de la aplicación en ese momento (en el caso del sprint 2 esta revisión fue presencial).

Por cada una de estas iteraciones, el autor le indicó a cada usuario piloto las nuevas funcionalidades añadidas, como podía interactuar con ellas y cuáles eran los resultados esperados, todo ello a través de una guía de revisión software. El usuario piloto, por su parte devolvería un nuevo informe al autor con los errores localizados si existiesen y las posibles recomendaciones de mejora si así lo estimase oportuno.

A continuación, por cada una de las diferentes iteraciones se detallarán cada uno de estos informes.

6.1.1 Sprint 1

Para el sprint 1 aún no se disponía de ninguna función lo bastante consolidada en la aplicación como para ser mostrada, por lo que se decidió postponer el plan de pilotaje para el segundo sprint.

6.1.2 Sprint 2

En el sprint 2 comenzó la primera ronda de pilotaje. Respecto a la funcionalidad que fue entregada, tal y como fue señalado en el punto 3.3.2 de este documento, los usuarios pilotos indicaron los siguientes puntos:

- **Registro e inicio de sesión**

Tanto en el caso de inicio de sesión, como en el de registro en el sistema, el usuario debía insertar sus credenciales usando campos de textos (y por tanto usar el teclado de escritura), es decir usuario y contraseña.

Fue el propio usuario piloto quien recomendó el uso de la pasarela de Google que había utilizado en otras aplicaciones. Con ella tanto el proceso de inicio de sesión como de registro se unificaban y solo era necesario pulsar un botón en la aplicación para acceder, en el caso de que se seleccione la cuenta por defecto.

- **API de productos con Firebase**

A pesar de que la funcionalidad de identificación aún no se había desarrollado, se le indicó al usuario la información estándar que iba a contener la etiqueta. Se le trasmitió al autor que sería interesante introducir algunos elementos adicionales como el precio del producto.

La propuesta fue aceptada y debido a que era un cambio menor y sencillo de solucionar, fue solventado ese mismo día. Tras mostrar el cambio, las mejoras fueron aceptadas por parte del cliente piloto.

6.1.3 Sprint 3

Durante esta iteración, que fue la primera tras el cambio de clientes pilotos (debido a las circunstancias del confinamiento), para el software entregado según la guía de revisión los clientes pilotos no localizaron ningún error.

Sin embargo, como mejora, indicaron que en el caso de que hubiese ocurrido un error en el momento de una lectura de una etiqueta NFC, o que el producto no fuese reconocible, debería aparecer un mensaje más descriptivo en lugar del que se estaba utilizando hasta ese momento. El cambio fue aceptado, lanzado una nueva versión al siguiente día de ser solicitado y enviada a los clientes pilotos. Dicha mejora la aceptaron satisfactoriamente los clientes piloto.

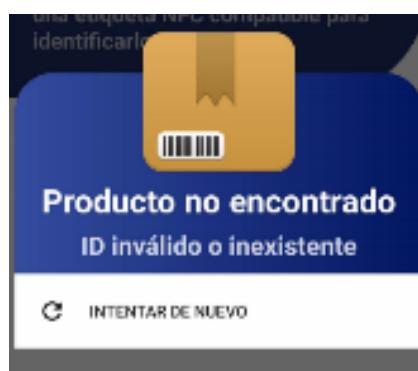


Ilustración 72 – Aviso de error de lectura

6.1.4 Sprint 4

Tal y como quedó indicado en el punto 3.3.4 se mostrarán a continuación los errores que fueron detectados por los clientes pilotos.

- **Listado de productos (tiempo real)**

La tarea fue marcada como finalizada, sin embargo, tras ser revisada por los clientes pilotos declararon que, si se hacía uso de dos dispositivos de manera simultánea y utilizando la función de añadir/eliminar productos de forma muy rápida, el listado no terminaba de actualizarse correctamente. Tras ser analizado el problema, se detectó que la rutina de actualizado no era correcta, y que además era un fallo no simple de solucionar. Al no ser una solución sencilla de solventar se decidió incorporar la tarea al *Sprint Backlog* de la siguiente iteración, para mostrarle la atención y detalle necesario.

Cabe destacar que a pesar de que en la tabla número 5 (Sprint 4 Backlog) se indica que la tarea de sincronización no se completó, la parte encargada de la sincronización de productos si se había finalizado.

- **Paginación de producto**

Los clientes pilotos indicaron que si se repetía la casuística del caso anterior la animación de la carga de productos entraba en un bucle infinito. El error fue analizado y su causa radicaba en el mismo problema del caso anterior por lo que fue solucionado de la misma forma.

6.1.5 Sprint 5

Durante este sprint no fue localizado ningún error por parte del equipo de clientes pilotos.

6.1.6 Sprint 6

Para este sprint no fue detectado ningún error por parte de los clientes pilotos, sin embargo, tras realizar pruebas individuales por parte del autor, si localizó uno muy importante relacionado con el acceso de listas de la compra a las que los usuarios son invitados.

Fueron localizados dos errores diferenciados, aunque ambos seguían la misma dinámica.

El primero de ellos consistía en la iteración entre dos sujetos, el cual será explicado por medio de dos usuarios ejemplo, nombrados como *usuario1* y *usuario2*. Suponga que *usuario1* envía una invitación a su lista de la compra a *usuario2*. *Usuario2* abre el diálogo que le permite aceptar o rechazar la invitación, pero no hace nada. Mientras *usuario2* mantiene abierto el diálogo de opciones, *usuario1* elimina su invitación, por lo que *usuario2* ya no debería poder aceptarla. Sin embargo, esta situación no estaba controlada y si *usuario2* (que tenía en ese momento la ventana de dialogo abierta) pulsaba sobre el botón de aceptar invitación esta era aceptada y le permitía el acceso.

El error fue controlado, si el usuario origen *usuario1* borra su invitación (como se puede apreciar en la captura central de la ilustración 73), el listado se actualizará y en el caso de que el usuario mantuviera el dialogo abierto, para cualquier acción que realice sobre esta (aceptar o rechazar) en primer lugar se comprobará el estado de la invitación. Como se puede ver a continuación, si existiera alguna incoherencia se le notificaría al usuario.

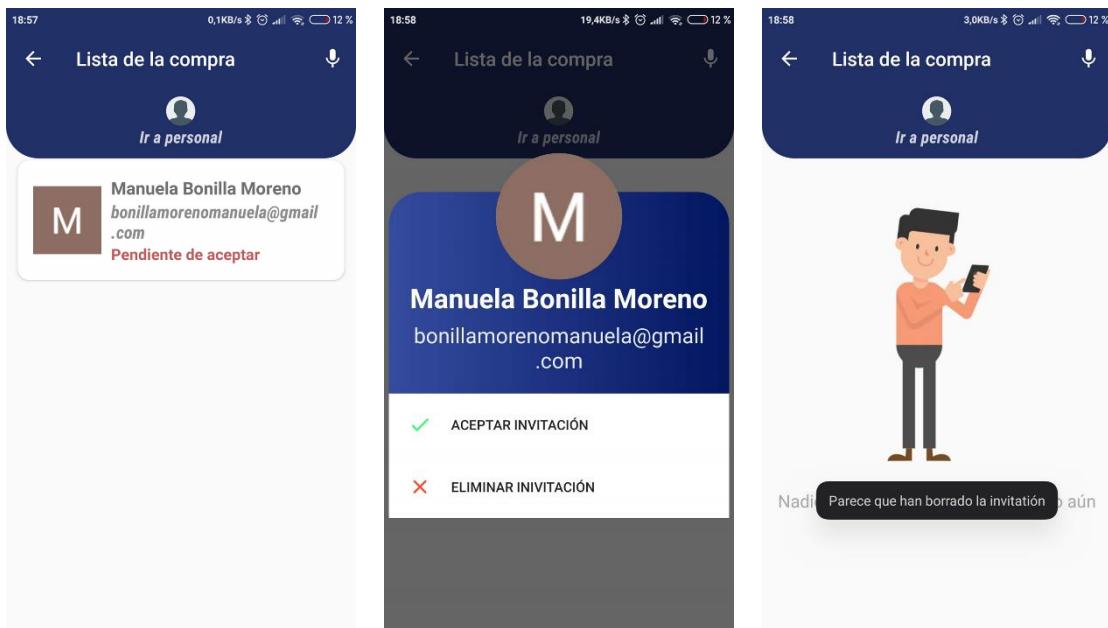


Ilustración 73 – Bug al aceptar petición solventado

El segundo caso se asemeja bastante al anterior. El error ocurría si *usuario2* abría el diálogo de acceso de la petición enviada por *usuario1*, mientras tanto *usuario1* eliminaba esta misma invitación (como se aprecia en la captura central de la siguiente ilustración, ya no aparece la invitación de Manuela). Una vez borrada la invitación, ya que *usuario2* mantenía abierta la ventana de diálogo aún disponía de todas las funcionalidades disponibles de la invitación, incluyendo el acceso a la propia lista. El error fue controlado y desde ese momento cada vez que se intenta acceder a un listado compartido se comprueba el estado de la invitación en remoto.

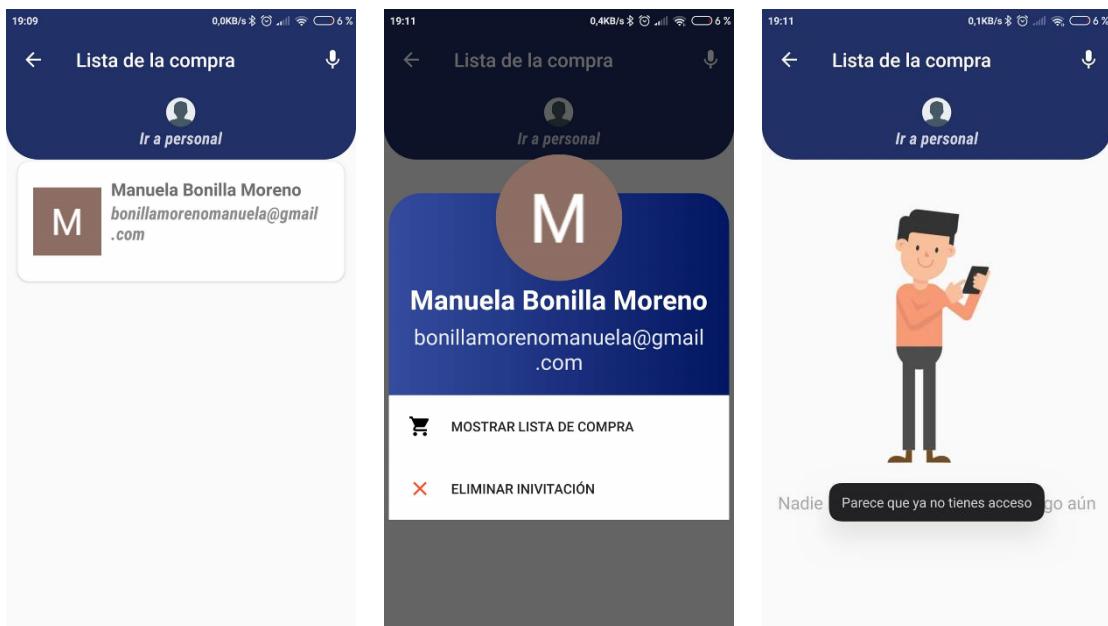


Ilustración 74 - Bug al acceder a la lista

Cabe destacar que para todos los casos de listados compartidos se ha implementado una funcionalidad que permite expulsar el usuario invitado (redireccionarlo al listado de

invitaciones) en el caso de que se encontrara visualizando la lista de la que el usuario propietario le hubiese eliminado el acceso.

6.1.7 Sprint 7

Tal y como puede apreciarse en la tabla resumen de tareas del apartado 3.3.7 se señaló la localización de errores, por parte del equipo de clientes pilotos en la tarea #C71 (“*Comandos de voz en listados de compra*”).

El error consistía en que resultaba difícil borrar un producto dictando su nombre al completo ya que a veces, una mínima modificación causaba el no reconocimiento del producto. Como ejemplo se señaló el caso de la palabra “*kilogramo*” la cual en ocasiones era reconocida tal cual, y en otra era recogida por el sistema como “*kg*”. Este error no solo se mantenía en el caso del borrado de producto, sino que también se extendía a la funcionalidad de marcado.

Como solución se planteó diseñar un nuevo sistema de borrado. La nueva metodología (no excluyente de la anterior) consiste en borrar y/o marcar un producto a través de su posición en la lista. De esta manera, ya que el sistema TalkBack indica la posición de un elemento en un listado, también podría ser utilizado por personas con discapacidad visual.

El funcionamiento de esta mejora queda detallado en el punto 6.7 de este documento, en concreto en el apartado (“*LISTA DE LA COMPRA – Marcar/Desmarcar un producto*”).

6.1.8 Sprint 8

Este sprint constituyó la última fase del desarrollo. En él se realizaron los test finales para la comprobación y búsqueda de posibles errores aún presentes.

Tras el análisis exhaustivo tal y como se establecieron en los distintos requisitos no fue localizado ningún error, situación que se presupone normal debido al plan de pilotaje que ha sido mantenido durante el transcurso de las diferentes iteraciones.

6.2 Pruebas de rendimiento

Para el desarrollo de este apartado ha sido utilizada la herramienta “*Android Profile [41]*” integrada en la propia interfaz de Android Studio. Dividiremos este bloque en tres apartados, dependiendo del estado de la aplicación en el dispositivo pudiendo ser reposo sin actividad, reposo con actividad y primer plano con actividad.

6.2.1 Reposo sin actividad

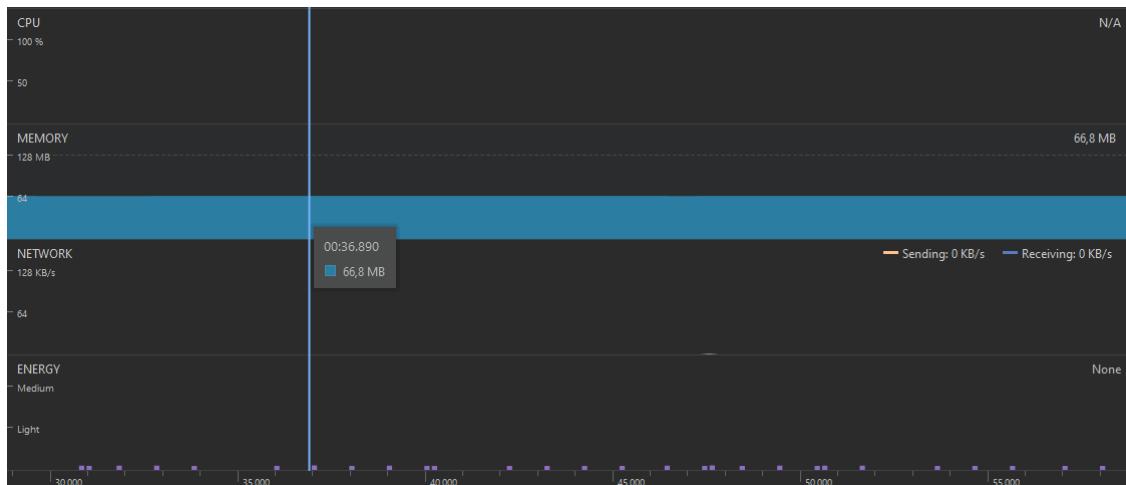


Ilustración 75 - Rendimiento en reposo

En la ilustración anterior se muestra el consumo de recursos por parte de la aplicación en estado completo de reposo sin actividad alguna.

Como se puede apreciar, respecto a la memoria se mantiene en un valor estándar de 66,8 MB. En cuanto a valores como CPU, conexión y energía los valores mostrados son prácticamente insignificantes.

Este valor de consumo de memoria, único significativo en este caso, ha sido comparado con dos de las aplicaciones iniciales con las que fue realizada la comparativa, las cuales son “Listonic” con un consumo de 169 MB y “Bring!” con un consumo de 113 MB.

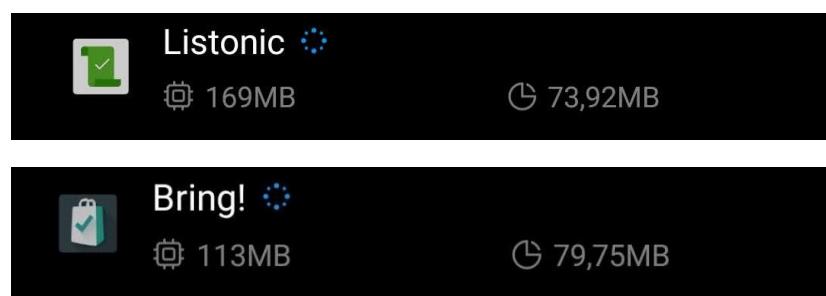


Ilustración 76 - Consumo de memoria de rivales

Por tanto, como mínimo, el consumo de la aplicación de este proyecto tiene un rendimiento 2,53 veces mejor que “Listonic” y 1,70 veces mejor que “Bring!”.

6.2.2 Reposo con actividad

Para esta simulación, y como forma de comprobar los límites de la aplicación, el test realizado ha consistido en comprobar el comportamiento entre dos dispositivos que comparten una lista de la compra (en ambos se inició la misma cuenta en la aplicación).

En concreto la prueba ha consistido en marcar y desmarcar todos los productos de una lista de 450 elementos, a la vez, utilizando la funcionalidad de acceso rápido “Marcar todos” durante un período de 2 minutos, repitiendo el proceso cada 2 segundos.

Destacar que el análisis que se muestra a continuación pertenece al dispositivo que sincroniza los datos, es decir, este dispositivo no es el que usa el usuario para marcar y desmarcar los productos. Solo se encarga de sincronizar los datos que recibe desde la base de datos remota y guardarlos a nivel local.

En definitiva, esta tarea consiste en el que dispositivo detecte que se ha producido una modificación, señalar los productos que han sido modificados y ejecutar los cambios a nivel local.

La siguiente ilustración muestra el estado de la CPU, memoria, batería y red durante este período.

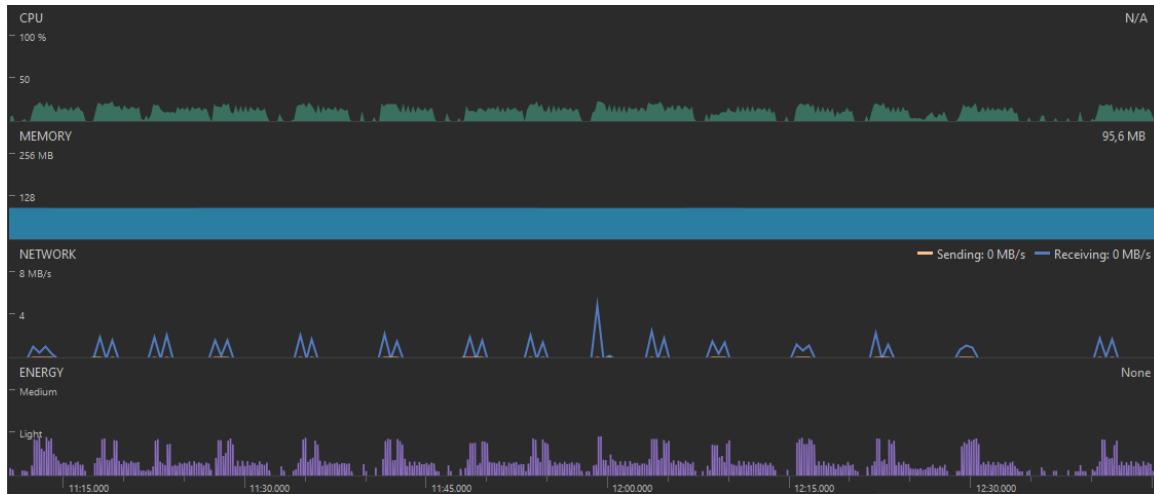


Ilustración 77 – Rendimiento en reposo activo

A diferencia del caso anterior (reposo sin actividad), se puede observar como el consumo de todos los recursos ha crecido, aunque sin llegar a ser excesivo en ningún momento.

Respecto a la batería en ningún momento se sobrepasa el nivel “Light” (*más bajo*), en la red el pico máximo es de 4.2 MB/s y la memoria se mantiene en un nivel estable de 104 MB.

El caso más excepcional, que puede apreciarse en esta ocasión, es el consumo de CPU como se muestra a continuación:

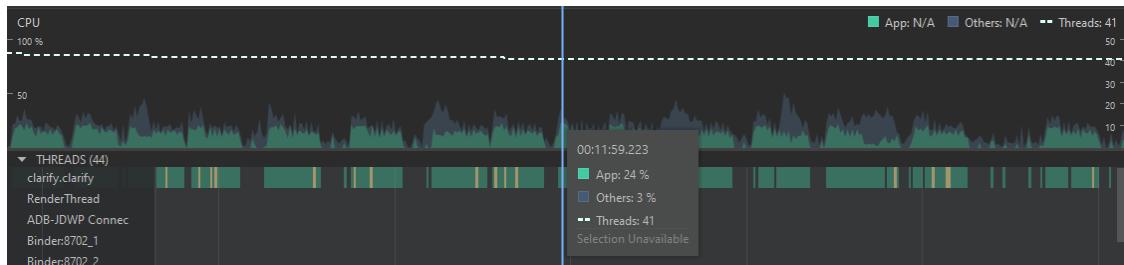


Ilustración 78 – Rendimiento de la CPU en estado de reposo activo

El hilo principal de la aplicación ha alcanzado el 24% del total de la capacidad máxima del procesador. Teniendo en cuenta que se supone que mediante esta prueba se ha alcanzado el mayor porcentaje que podría requerir la aplicación, se considera una cifra comprensible y adecuada.

6.2.3 Primer plano con actividad

Para esta prueba se repetirá el procedimiento detallado anteriormente, pero en este caso el dispositivo al que se le medirá el consumo de recursos también será el encargado de realizar la tarea de marcar y desmarcar todos ellos.

Al igual que en el caso anterior la lista está compuesta por 450 elementos y durante dos minutos serán marcados y desmarcados todos en intervalos de dos segundos por cada acción durante 2 minutos.

Finalmente, en este caso la tarea consistirá en marcar y desmarcar todos los productos a nivel local, detectar los cambios, guardar los cambios y actualizar los datos a nivel remoto en la base de datos externa.

La siguiente ilustración muestra el estado de la CPU, memoria, batería y red durante este período.

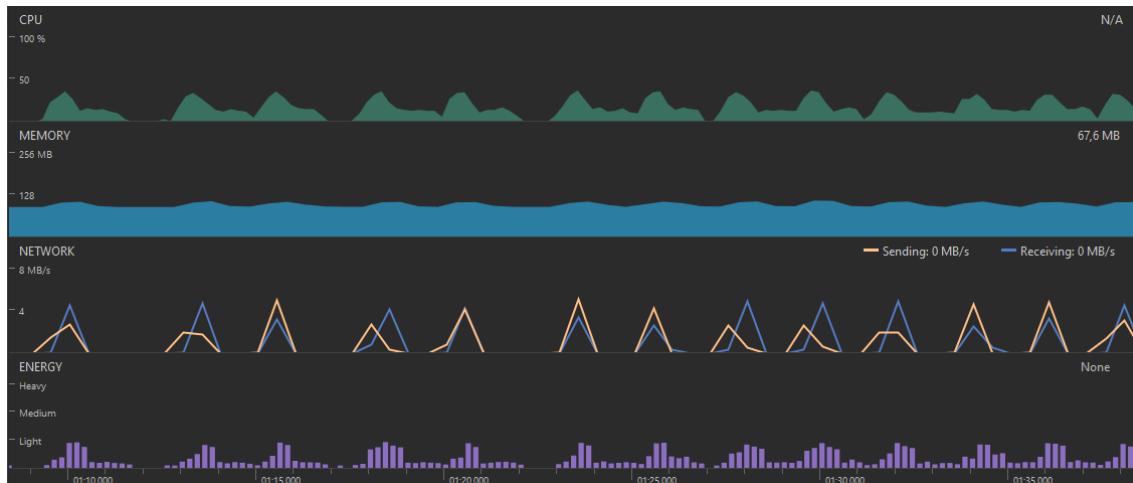


Ilustración 79 – Rendimiento en primer plano con actividad

En esta ocasión si se puede apreciar como el nivel de consumo de los distintos recursos ya es más alto que anteriores ocasiones.

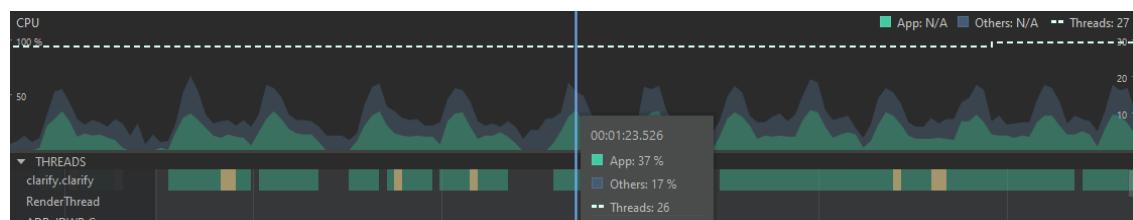


Ilustración 80 – Rendimiento de la CPU en primer plano activo

En la ilustración general de rendimiento aparece haber un consumo de la CPU cercano al 50%, sin embargo, la aplicación tal y como se puede ver en la captura anterior, en su pico máximo solo alcanza el 37%. La razón del aumento de consumo en este apartado frente a la situación de reposo activo se justifica en que en esta ocasión a la tarea de modificación de la base de datos local se añade la modificación de datos a nivel remoto, en Firebase. Suponiendo que se ha simulado una situación extrema y por tanto el límite de la aplicación, se entienden estos valores como normales dado este escenario.

Respecto al consumo de memoria, tal y como se muestra en la ilustración 79 ha sufrido variaciones, sin embargo, en todo momento se ha mantenido entorno a valores de 109 y 96 MB. Cifra inapreciable para un dispositivo de 6 GB de RAM.

En la siguiente imagen se muestra un extracto del consumo de red en el momento de mayor demanda por parte de la aplicación.

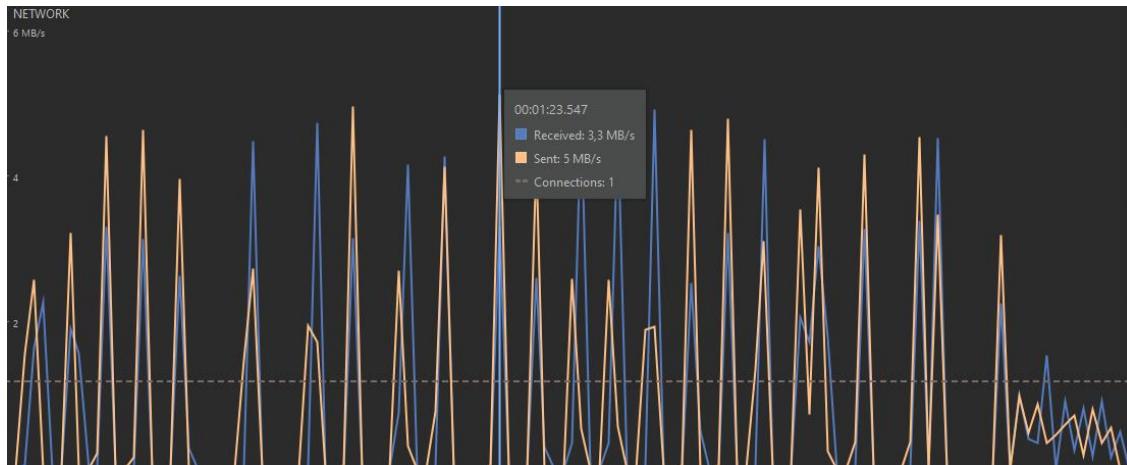


Ilustración 81 – Rendimiento de la conexión

A diferencia de los casos anteriores, en esta ocasión existe tráfico de datos tanto entrante como saliente. Los datos salientes corresponden a la sincronización de la base de datos remota cuando la tarea en local ha finalizado. Por otro lado, el tráfico entrante corresponde a que la aplicación ha detectado que se han producido modificaciones a nivel remoto, correspondiente a los cambios que acaban de realizarse. De igual forma, si existiese en la trama algún cambio no presente a nivel local se procesaría.

En esta ocasión el pico máximo alcanzado es de 8,3 MB/s, cifra que representa el 2.18% de la capacidad máxima del dispositivo (47.5 MB/s).

Por último, en la siguiente captura se muestra el consumo de energía de la aplicación:

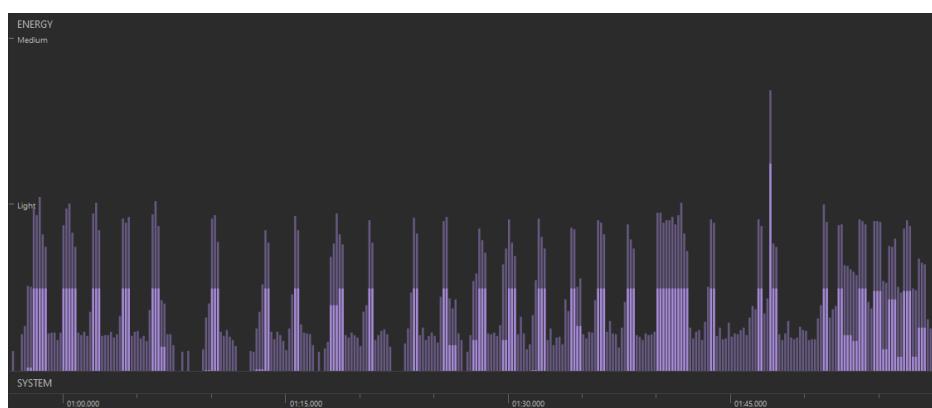


Ilustración 82 – Consumo de energía

En el pico máximo de consumo no se ha llegado a sobrepasar el límite marcado como "Medium", por lo que se comprende que, dado el tipo de prueba que se ha realizado, la gestión de gasto energético ha sido bastante buena.

7 CONCLUSIONES

En este bloque se pretende realizar un análisis de la consecución de los objetivos que fueron planteados inicialmente en el proyecto, establecer una serie de mejoras para el futuro y, por último, una reflexión personal del proyecto por parte del autor de este Trabajo de Fin de Grado.

7.1 Objetivos iniciales

El objetivo principal y que inicialmente se marcó como base en este TFG fue la creación de un proyecto que permitiese aumentar el nivel de independencia que poseen las personas con discapacidad visual en su día a día.

Para ello, se ha desarrollado una aplicación móvil que permite identificar todos los productos con los que se encuentre una persona con discapacidad visual, siempre que estos cuenten con una etiqueta NFC compatible con los estándares del proyecto desarrollado.

Por otro lado, este proyecto también pretende demostrar que la complejidad de adaptar cualquier aplicación (en este caso Android) a un entorno más accesible no es excesivo, incluso si ésta ya se encuentra en fase de producción.

De esta forma, se ha llegado a la conclusión que basta con seguir los dictámenes marcados por los distintos sistemas operativos en cuestiones de diseños accesibles, los cuales se basan en contenidos descriptivos de interfaces, que no difieren excesivamente de la dinámica que usualmente se utilizaría en un desarrollo tradicional. Tras toda la investigación realizada por el autor del proyecto, en la inmensa mayoría de las ocasiones, los propios sistemas operativos cuentan con mecanismos propios encargados de realizar el trabajo más complejo, proveyendo a los desarrolladores de una API completa para poder utilizarla, por lo que el trabajo extra se reduciría a hacer uso de esta.

Además, debido a la situación de confinamiento, y en suma a la imposibilidad de contactar de forma presencial con los clientes pilotos iniciales tal y como se había planeado para la función de identificación (acorde a la reunión que se realizó con la ONCE), se decidió potenciar aún más la funcionalidad de creación de listas de compra compartidas, función que permitiría abrir el abanico de posibles clientes pilotos que colaboraran a lo largo de las distintas fases del proyecto, aportando su “feedback” personal.

Esta funcionalidad, que en un primer momento parece alejar los objetivos iniciales de su dirección natural inicial, supuso todo un acierto, ya que permitió completar el ciclo del proceso de compra, incluyendo en este a todos los actores participantes, desde grandes cadenas de supermercados (mediante tareas de fidelización) hasta el propio comprador del producto.

Por tanto, se ha logrado alcanzar el objetivo planteado inicialmente, se ha creado una aplicación móvil que facilita el día a día de personas con discapacidad visual pero que, a su vez, queda enmarcada en una aplicación móvil usable y llamativa para un público general.

A partir del objetivo principal señalado anteriormente surgieron necesidades derivadas a medida que el proyecto se desarrollaba y que igualmente fueron completadas.

La mayoría de estos nuevos objetivos surgieron a partir del desconocimiento de la tecnología al comienzo de este proyecto.

El autor nunca había desarrollado ninguna aplicación Android, ni había trabajado con dispositivos NFC, por lo que desconocía por completo el nuevo entorno de trabajo (con la excepción de Java). Es por ello por lo que adquirir una formación consistente en estas tecnologías supuso una meta adicional a lo largo de cada etapa.

Respecto al código fuente desarrollado, se encuentra disponible en el siguiente repositorio: <https://github.com/alvarodelaflor/clarify>.

Finalmente, además de todo lo anterior, es importante añadir que todo el proyecto se ha desarrollado siguiendo las distintas fases de la ingeniería de software como son el análisis de requisitos, la planificación el diseño y el desarrollo. El objetivo fundamental respecto a este ámbito ha sido tratar de aplicar todos los conocimientos adquiridos por el autor durante su etapa de formación en el grado y ser consciente de las ventajas que supone aplicarlas de forma efectiva en el contexto de las metodologías estudiadas.

7.2 Futuras mejoras

Una posible mejora para este proyecto en un futuro podría ser ampliar la funcionalidad a otras plataformas como iOS o incluso un sistema web accesible desde el propio navegador y, por tanto, compatible con prácticamente cualquier dispositivo con acceso a internet.

Ambos casos suponen partir prácticamente desde 0 en un nuevo proceso de desarrollo, sin embargo, el nuevo alcance supondría extender la compatibilidad del proyecto a prácticamente la totalidad de dispositivos móviles. “Clarify” surgió como necesidad para un determinado sector de la población muy concreto y gracias a esta mejora se podría ayudar a un número aún más grande de personas, simplemente exportando la funcionalidad que se ha desarrollado en este proyecto.

Respecto a la aplicación desarrollada, podrían ser mejorados determinados aspectos, como la integración con “TalkBack”. En concreto, la funcionalidad de asistente de voz se puede seguir perfeccionando, para que tenga en cuenta todas las posibles situaciones existentes en el día a día para una persona con discapacidad visual, que no han podido ser consideradas en este caso, en parte, por la dificultad de encontrar un cliente acorde debido a la situación excepcional en la que ha transcurrido el proyecto. Con el estado confinamiento de la población tuvo que anularse el plan de pilotaje que había sido diseñado conjuntamente con una persona con discapacidad visual, ya que en este caso el contacto presencial era esencial.

Por otro lado, otra mejora podría ser la posibilidad de integrar servicios de terceros, como el propio Google Assistant, el cual permitiría interactuar con la aplicación sin necesidad de abrir esta, y desde cualquier dispositivo. Especialmente sería interesante su uso conjunto con productos como “Google Home Speaker [42]”. Este asistente de voz está especialmente diseñado para interactuar con él a base de comandos de voz, lo cual encaja a la perfección en la filosofía de este proyecto.

7.3 Reflexión personal

Este proyecto ha representado el primer contacto con la realidad respecto a cuáles son nuestros verdaderos conocimientos. La realización de un Trabajo Fin de Grado supone ser capaz de condensar y demostrar todos los conocimientos alcanzados durante cuatro años en un proyecto, con la ayuda en este camino de José Antonio Parejo (tutor).

Durante mi formación universitaria, he logrado comprender que el objetivo del grado realizado no es lograr que el estudiante alcance a conocer el máximo número lenguajes, metodologías y tecnologías, sino más bien conseguir un nivel de maduración y aptitud tales que le permitan dar otro enfoque al enfrentarse a los continuos y diversos problemas que surgen en todo ciclo software. Saber enfrentarse a los problemas de una forma metódica, ordenada y óptima.

La tecnología está viva, es cambiante y fluye en constante evolución. La finalización de este grado solo es el punto de partida para iniciar nuestra carrera en el mundo laboral, en el que la continua formación debe seguir siendo un pilar fundamental e imprescindible en cada paso que demos.

Desde el inicio de este proyecto pretendí enfocarlo como un reto personal, una demostración de mis aptitudes más que de mis conocimientos. Muestra de ello es el desconocimiento total de las tecnologías utilizadas; *Realm*, *Firebase* (bases de datos no relaciones) e incluso *Android* eran una completa incógnita al inicio.

Durante el desarrollo del proyecto he logrado alcanzar la autonomía y experiencia necesaria para enfrentarme a los problemas surgidos durante el desarrollo, conocer nuevas tecnologías y en definitiva adquirir las competencias básicas que se esperan de cualquier ingeniero del software.

Por otro lado, a pesar de que el desarrollo en iOS (sistema que se presupone más accesible para personas con discapacidad visual) fue descartado debido a mis dificultades técnicas, ya que no cuento con ningún dispositivo Apple. Considero que la aplicación que he desarrollado en este proyecto ha alcanzado un alto nivel de accesibilidad junto con una gran aplicabilidad, demostrando así que el uso de Android es igual de viable que iOS.

A pesar de que el proyecto fue duro en sus inicios, debido a las dificultades que se iban sucediendo, finalmente se han cumplido todos los objetivos planteados e incluso superado en gran medida las expectativas iniciales, tanto en el grado de desarrollo como en el público objetivo, ya que inicialmente el proyecto estaba limitado a un nicho de mercado muy concreto, pero finalmente se ha desarrollado una aplicación que además de cubrir ese nicho de mercado, también se puede hacer extensiva a la población en general que está familiarizada y habituada al uso de aplicaciones móviles para el desarrollo de sus actividades diarias. En definitiva, considero que a nivel personal ha sido un proyecto bastante enriquecedor, ya que he logrado afianzar aún más mis competencias a nivel de planificación y análisis, además de conocer tecnologías bastante interesantes que espero que solo sean el comienzo del futuro que me depara.

"The present is theirs; the future, for which I really worked, is mine."

Nikola Tesla

8 REFERENCIAS

[1]

J. Clement, 4 Mayo 2020. [En línea]. Available: <https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>.

[2]

ONCE, «Web ONCE,» [En línea]. Available: <https://www.once.es/>.

[3]

[En línea]. Available: <https://es.wikipedia.org/wiki/VoiceOver>.

[4]

Google, «TalkBack Support,» [En línea]. Available: <https://support.google.com/accessibility/android/answer/6283677?hl=es>.

[5]

Ilunion, «ilunion.com,» [En línea]. Available: <https://www.ilunion.com/es/comunicacion/actualidad/13032018/ilunion-tecnologia-y-accesibilidad-instala-sus-pioneros-beepcons-en>.

[6]

Diario de Sevilla, «<https://www.diariodesevilla.es/>,» 23 Enero 2020. [En línea]. Available: <https://www.diariodesevilla.es/>.

[7]

Lazzus, «Portal Lazzus,» [En línea]. Available: lazzus.com/es/.

[8]

EIEspañol, «[elespanol.com,](https://elespanol.com/)» [En línea]. Available: https://cronicaglobal.elespanol.com/creacion/vida-tecky/lazzus-el-google-maps-para-ciegos_88274_102.html.

[9]

neosentec, «YouTube,» 2018. [En línea]. Available: <https://www.youtube.com/watch?v=BvXCs9PtKYU>. [Último acceso: 22 05 2020].

[10]

AndroidRed, [En línea]. Available: <https://www.androidred.com/download-be-my-eyes-for-android/>.

[11]

Google LLC, «Google Support,» [En línea]. Available: <https://support.google.com/accessibility/android/answer/6283677?hl=es>. [Último acceso: 22 05 2020].

[12]

J. C. Fernández, «Applesfera,» 26 Marzo 2020. [En línea]. Available: <https://www.applesfera.com/desarrollo-de-software/swift-5-3-llegara-a-windows-otras-distros-linux>.

[13]

J. Callaham, «Android Authority,» 18 Agosto 2019. [En línea]. Available: <https://www.androidauthority.com/history-android-os-name-789433/>.

[14]

R. Brandom, "The Verge," 7 Mayo 2019. [Online]. Available: <https://www.theverge.com/2019/5/7/18528297/google-io-2019-android-devices-play-store-total-number-statistic-keynote>.

[15]

Nts-Solutions, 10 Febrero 2020. [En línea]. Available: <https://www.nts-solutions.com/blog/dispositivos-android.html>.

[16]

JetBrains, «JetBrains,» [En línea]. Available: <https://www.jetbrains.com/>.

[17]

«Kotlin,» [En línea]. Available: <https://kotlinlang.org/>.

[18]

«FreeJavaGuide,» [En línea]. Available: <https://www.freejavaguide.com/history.html>.

[19]

«Mercadona,» [En línea]. Available: <https://info.mercadona.es/es/conocenos>.

[20]

Python, [En línea]. Available: <https://www.python.org/>.

[21]

«Google Firebase,» [En línea]. Available: <https://firebase.google.com/docs/database?hl=es-419>.

[22]

«Realm,» [En línea]. Available: <https://realm.io/>.

[23]

Bitrise, 3 Agosto 2018. [En línea]. Available: <https://medium.com/bitrise/realm-vs-sqlite-which-database-is-better-for-android-apps-c751dc8b150c>.

[24]

M. J. Gutiérrez, «ElAndroidLibre,» 1 Enero 2017. [En línea]. Available: <https://elandroidelibre.elespanol.com/2017/01/de-holo-a-material-design-cual-será-el-siguiente-cambio-de-android.html>.

[25]

«Material Design,» [En línea]. Available: <https://material.io/design/introduction#principles>.

[26]

Google LLC, «SpeechRecognizer,» [En línea]. Available: <https://developer.android.com/reference/kotlin/android/speech/SpeechRecognizer>. [Último acceso: 2020 05 22].

[27]

DevMagazine, 29 Julio 2018. [En línea]. Available: <https://devmagazine.co/conoce-la-herramienta-android-studio-y-sus-caracteristicas/2168/>.

[28]

J. Pastor, «Xataka,» 4 Junio 2018. [En línea]. Available: <https://www.xataka.com/aplicaciones/oficial-microsoft-compra-github-7-500-millones-dolares>.

[29]

K. BROWN, 13 Noviembre 2019. [En línea]. Available: <https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/>.

[30]

«Jira,» [En línea]. Available: <https://www.atlassian.com/es/software/jira>.

[31]

Ministerio de Sanidad, 17 Marzo 2020. [En línea]. Available: https://www.mscbs.gob.es/profesionales/saludPublica/ccayes/alertasActual/nCov-China/documentos/20200417_ITCoronavirus.pdf.

[32]

Ward Cunningham, «agilemanifesto.org,» [En línea]. Available: <https://agilemanifesto.org/iso/es/manifesto.html>. [Último acceso: 22 Mayo 2020].

[33]

J. Francia, «scrum.org,» 25 Septiembre 2017. [En línea]. Available: <https://www.scrum.org/resources/blog/que-es-scrum>. [Último acceso: 22 Mayo 2020].

[34]

Glassdoor, «Glassdoor,» [En línea]. Available: https://www.glassdoor.es/Sueldos/junior-software-engineer-sueldo-SRCH_KO0,24.htm.

[35]

Gobierno de España, «boe.es,» [En línea]. Available: <https://www.boe.es/boe/dias/2018/03/06/pdfs/BOE-A-2018-3156.pdf>.

[36]

Ministerio de inclusión, seguridad social y migraciones, [En línea]. Available: <http://www.seg-social.es/wps/portal/wss/internet/Trabajadores/CotizacionRecaudacionTrabajadores/36537>.

[37]

Agencia Tributaria, [En línea]. Available: https://www.agenciatributaria.es/AEAT.internet/Inicio/_Segmentos_/_Empresas_y_profesionales/Empresas/Impuesto_sobre_Sociedades/Periodos_impositivos_a_partir_de_

1_1_2015/Base_imponible/Amortizacion/Tabla_de_coeficientes_de_amortizacion_lineal_.shtml.

[38]

Google LLC, «Gmail,» [En línea]. Available: <https://www.google.com/intl/es/gmail/about/>. [Último acceso: 22 Mayo 2020].

[39]

Google LLC, «Notifications - Android Developer,» [En línea]. Available: <https://developer.android.com/guide/topics/ui/notifiers/notifications?hl=es-419>. [Último acceso: 23 Mayo 2020].

[40]

Organización Mundial de la Salud, «who.int,» [En línea]. Available: <https://www.who.int/es/emergencies/diseases/novel-coronavirus-2019/advice-for-public/q-a-coronaviruses#:~:text=La%20COVID%2D19%20es,en%20diciembre%20de%202019>. [Último acceso: 23 Mayo 2020].

[41]

Google LLC, [En línea]. Available: <https://developer.android.com/studio/profile>.

[42]

Google LLC, [En línea]. Available: https://store.google.com/es/product/google_home_speaker.