Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Aplicación móvil para visualizar, monitorizar y recoger contenedores de basura

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Daniel Álvarez Vallejo

Tutor: Carlos Herrero Cucó

Curso 2020-2021

Resumen

Actualmente, muchas veces se encuentran bolsas de basura fuera de los propios contendores que impiden poder caminar cómodamente por la acera o introducir correctamente cada residuo en su correspondiente contenedor.

Esto suele estar causado por diferentes situaciones, como, por ejemplo, quien vaya a tirar la basura desconozca si el contenedor más cercano de su casa está lleno o no y, cuando llegue al contenedor, se encuentre con la obligación de dejar la bolsa fuera porque está a rebosar.

Por otra parte, la recogida de basura de los contenedores suele ser realizada por barrios, por lo que un contenedor lleno puede mantenerse así durante uno o dos días, llegando a tener un nivel de acumulación de basura perjudicial para las calles y para el medio ambiente.

Para solucionar el primer problema, se propone una aplicación móvil con la que los usuarios tengan total conocimiento de dónde se encuentra cada tipo de contenedor y el porcentaje de basura que contiene cada uno de ellos. Con esto, se evita la acumulación perjudicial mencionada anteriormente.

Y para el segundo, en una aplicación independiente a la primera, se les ofrece a los recogedores de basura un sistema de creación de rutas automáticas para recorrer los puntos donde los porcentajes de residuos sean altos.

En esta solución se utilizan herramientas propias de Google. Para la parte de la interfaz de usuario se utiliza el lenguaje de programación Dart junto con su *framework* Flutter, y para la parte de la gestión de datos se hace uso también de Dart, pero con el apoyo de Firebase. También se incluye un sistema basado en IOT que incluye en cada contenedor los componentes necesarios para comunicar la situación de dicho contenedor en el sistema.

Palabras clave: Gestión de residuos; Ética; Compromiso Social; Optimización; Gestión de la información; IOT.

Abstract

Nowadays, many times garbage bags are found outside the bins themselves, making it impossible to walk comfortably on the sidewalk or to correctly introduce each waste in its corresponding container.

This is usually caused by different situations, such as, for example, the person who goes to throw the garbage does not know if the nearest container to his house is full or not and, when he arrives at the container, he is forced to leave the bag outside because it is overflowing.

On the other hand, the collection of garbage from the containers is usually done by neighborhoods, so that a full container can remain full for one or two days, reaching a level of accumulation of garbage that is harmful to the streets and the environment.

To solve the first problem, we propose a mobile application with which users have full knowledge of where each type of container is located and the percentage of garbage contained in each of them. With this, the aforementioned harmful accumulation is avoided.

And for the second, in an application independent of the first, garbage collectors are offered a system for creating automatic routes to go through the points where the percentages of waste are high.

This solution uses Google's own tools. For the user interface part, the Dart programming language is used together with its framework Flutter, and for the data management part, Dart is also used, but with the support of Firebase. An IOT-based system is also included that includes in each container the necessary components to communicate the status of said container in the system.

Keywords: Waste management; Ethics; Social Commitment; Optimization; Information management; IOT.

Resum

Actualment, moltes vegades es troben bosses de fem fora dels propis contenidors que impedeixen poder caminar còmodament per la vorera o introduir correctament cada residu en el seu corresponent contenidor.

Això sol estar causat per diferents situacions, com, per exemple, qui vaja a tirar el fem desconega si el contenidor més pròxim de la seua casa està ple o no i, quan arribe al contenidor, es trobe amb l'obligació de deixar la bossa fóra perquè està a vessar.

D'altra banda, la recollida de fem dels contenidors sol ser realitzada per barris, per la qual cosa un contenidor ple pot mantenir-se així durant un o dos dies, arribant a tenir un nivell d'acumulació de fem perjudicial per als carrers i per al medi ambient.
Per a solucionar el primer problema, es proposa una aplicació mòbil amb la qual els usuaris tinguen total coneixement d'on es troba cada tipus de contenidor i el percentatge de fem que conté cadascun d'ells. Amb això, s'evita l'acumulació perjudicial esmentada anteriorment.

I per al segon, en una aplicació independent a la primera, se'ls ofereix als recollidors de fem un sistema de creació de rutes automàtiques per a recórrer els punts on els percentatges de residus siguen alts.

En aquesta solució s'utilitzen eines pròpies de Google. Per a la part de la interfície d'usuari s'utilitza el llenguatge de programació Dart juntament amb el seu *framework* Flutter, i per a la part de la gestió de dades es fa ús també de Dart, però amb el suport de Firebase. També s'inclou un sistema basat en IOT que inclou en cada contenidor els components necessaris per a comunicar la situació d'aquest contenidor en el sistema.

Paraules clau: Gestió de residus; Ètica; Compromís Social; Optimització; Gestió de la informació; IOT.

Tabla de contenidos

Tabla de ilustraciones

1. Introducción

Hoy en día, cuando llega el momento de bajar a tirar la basura, es habitual encontrarse bolsas de residuos, colchones, cartones, plásticos… fuera de sus respectivos contenedores. Esto impide caminar cómodamente por la acera o introducir correctamente los desperdicios en su contenedor correspondiente.

Existen varios escenarios que conducen a la situación mencionada anteriormente. Por ejemplo, cuando se va a bajar la basura, se desconoce si el contenedor (normalmente el más cercano) está lleno o no. Suele pasar que existe esa "obligación" de dejar la bolsa fuera porque el contenedor está a rebosar, y así sucesivamente hasta bloquear la vía pública, además de dañar el medio ambiente. También puede ocurrir que, la recogida de la basura, al ser realizada por barrios o distritos, un mismo contenedor puede llegar a mantenerse lleno hasta uno o dos días.

Actualmente, se tienen al alcance de la mano medios tecnológicos con los que se puede navegar por el mapa de cualquier territorio y otros que permiten detectar objetos a corta y larga distancia. Estos últimos son reconocidos como sensores ultrasónicos o de ultrasonidos, y se encargan de emitir sonidos y medir el tiempo que tardan las señales en regresar.

En efecto, se pueden unir las ventajas que ofrecen los servicios de geolocalización con los datos que proporcionan este tipo de sensores. De esta necesidad, surgen las aplicaciones Tot Net y Tot Net Recogedores para dispositivos móviles que soporten el sistema operativo Android. El título del trabajo es genérico, es decir, la aplicación está compuesta por otras dos diferenciadas para usuarios y recogedores.

La primera aplicación permite al usuario visitar el mapa territorial de Burjasot, municipio y ciudad de la Comunidad Valenciana, en España. En este se ofrece la localización de la gran mayoría de los contenedores existentes, coloreados de manera diferente según el tipo de residuo que almacenen. De este modo, cuando el usuario lo desee, puede obtener la siguiente información de cada contenedor accediendo a él directamente desde el mapa: la calle en la cual está ubicado, la cantidad de residuos que contiene, posibles sugerencias, la opción de generar una ruta hasta él o si está o no marcado como favorito.

Por otra parte, en Tot Net Recogedores se cuenta con la misma información que en la anterior aplicación a excepción del sistema de favoritos, pero se pueden generar rutas con paradas. Esas paradas son contenedores con altos porcentajes de residuos que están listos para recoger.

Finalmente, en los próximos apartados de forma resumida, se explican tanto los objetivos marcados como los requisitos especificados de cada aplicación, además de la arquitectura en la que se sostienen, su implantación y las correspondientes pruebas para verificar el correcto funcionamiento de ambos sistemas.

1.1. Motivación

Lamentablemente, es una realidad ir andando por la calle y cambiarse a la otra acera o incluso bajarse a la carretera porque se encuentran deshechos y bolsas de basura que obstaculizan el camino.

España generó 132,1 millones de toneladas de residuos en 2017, el último año del que se tienen datos, siendo así el país de la Unión Europea que vierte una mayor cantidad de residuos a vertederos. Centrándose en lo más cercano a la sociedad, aproximadamente un 17,1% de esos residuos provienen de los hogares. Esta cifra corresponde a la última Memoria Anual de Generación y Gestión de Residuos de Competencia Municipal publicada por el Ministerio de Transición Ecológica [1].

132,1 millones de toneladas es una cantidad estratosférica e inimaginable. Como se ha dicho en el párrafo anterior, el 17,1% corresponde a 22,5 millones de toneladas de basura que se vierten en los contenedores de las calles. Si se cuenta con que una tonelada es un bloque de 1m x 1m x 1m, y que, por ejemplo, el campo de Mestalla mide 105x68 m^2, si se apilase la basura sobre el campo de futbol (suponiendo que no se desmorona y está compactada) se tendría una torre de 3.151,2 metros de altura…

Por ello, es necesaria una aplicación móvil al alcance de cualquiera que ayude a localizar aquellos contenedores ideales para tirar la basura y, además, otra para apoyar a los recogedores de basura ofreciéndoles rutas efectivas para evitar acumulaciones de deshechos en la vía pública.

1.1.1. Motivación personal

En la medida de lo posible, siempre he intentado realizar aquellas acciones que, por alguna razón, ayudan al bienestar del medio ambiente. Ahora que poseo conocimientos suficientes como para construir una solución, no veo mejor momento para ponerlos en marcha.

Por tanto, y viendo las cifras mencionadas en el anterior apartado, creo que no hay mejor manera posible de ayudar a mejorar esta situación que aprovechar los medios tecnológicos con los que contamos y aplicar todo lo que he aprendido durante esta etapa.

1.2. Objetivos

Como se ha comentado anteriormente en la introducción, el principal objetivo de esta aplicación es dar a conocer a los usuarios la posición y el tipo de cada uno de los contenedores ubicados en el territorio de Burjasot. Además, también se debe informar sobre el porcentaje de residuos que contiene cada uno haciendo uso de un sensor ultrasónico.

Por otra parte, los recogedores tienen una segunda aplicación independiente con funcionalidades similares. El objetivo principal de esta es aumentar la eficiencia con la que se recogen los contenedores.

En resumen, las dos aplicaciones deben cumplir los siguientes puntos básicos:

- Mostrar ubicación actual en el mapa.
- Mostrar mapa territorial de Burjasot.
- Mostrar contenedores de cada tipo en el mapa.
- Mostrar para cada contenedor, el porcentaje ocupado por residuos.
- Generar una ruta para un contenedor seleccionado.

La primera aplicación, en especial, debe cumplir los siguientes:

- Registrar usuarios con nombre de usuario, correo y contraseña.
- Registrar usuarios a través de Google.
- Consultar perfil.
- Cambiar nombre de usuario.
- Cambiar contraseña (sólo si la cuenta no es Google).
- Cambiar foto de perfil.

- Cerrar sesión.
- Marcar/desmarcar como favorito cualquier contenedor.
- Consultar los contenedores favoritos.

Por último, la aplicación de los recogedores tiene que cumplir especialmente el siguiente objetivo:

- Generar una ruta por los contenedores con porcentajes de llenado más altos.

1.3. Estructura

Hasta este punto, se ha visto una pequeña introducción que explica a grandes rasgos qué hace cada aplicación, los motivos por los cuales se han desarrollado y en qué contexto, y cuáles son los objetivos que deben cumplir.

En los próximos apartados, se comenta el estado del arte o contexto tecnológico en el que se han desarrollado los sistemas, además de presentar las aplicaciones que se han escogido como competencia para estas.

Posteriormente se expone un análisis del problema, se explican las técnicas utilizadas para establecer los requisitos y tras ello, se muestran los casos de uso definidos para cada aplicación.

En el siguiente capítulo se detalla cómo se ha llevado a cabo el diseño de la solución de ambos servicios, la arquitectura en la que se dividen, sus componentes esenciales, y, por último, la tecnología utilizada.

Más adelante, con las aplicaciones ya implementadas, se realiza la puesta en marcha de estas para probarlas y obtener resultados. Como en todo trabajo, se presentan las pruebas que se han hecho para verificar que las soluciones finales funcionan correctamente.

En la siguiente sección, se exponen las conclusiones en las que se pone de manifiesto si se han alcanzado todos los objetivos planteados y si se han desarrollado satisfactoriamente. Como subapartado, se analiza la relación de los estudios realizados con el trabajo presentado y después se redactan las posibles mejoras que se pueden aplicar en un futuro.

Finalmente, se adjunta un listado de referencias para verificar las fuentes originales en las que se basa el trabajo, seguido de un glosario en el que aparecen palabras específicas o acrónimos del área que pueden ser poco conocidos.

2. Estado del arte

En este capítulo se ofrece un vistazo general por el contexto en el cual ha sido desarrollado este proyecto, centrándose en los dispositivos móviles que utilizan Android como sistema operativo.

Por una parte, se habla sobre las características más importantes de Android y la arquitectura sobre la que se sostiene. Seguidamente, se comenta la situación actual del mercado en cuanto a la compra de dispositivos Android y iOS, argumentando por qué se ha elegido el primero. También se realiza una pequeña introducción sobre las características más importantes que poseen tanto la placa de desarrollo como el sensor utilizado en este proyecto.

Y, por otra parte, se describen las aplicaciones ya existentes en la tienda de Google o trabajos relacionados que suponen una amenaza para el trabajo desarrollado, siempre hablando en términos de competencia.

2.1. Android

Principalmente, Android es un sistema operativo móvil basado en el núcleo de Linux y otros softwares de código abierto. Fue diseñado para dispositivos móviles con pantalla táctil, como tabletas, teléfonos, relojes inteligentes…

Fue desarrollado por Android Inc. y adquirido por Google en 2005. El código central de Android se conoce principalmente como Android Open Source Project (AOSP) y tiene licencia de Apache. Además, el sistema operativo fue uno de los objetivos de las agencias de inteligencia internacionales, según documentos secretos filtrados en 2013 y 2014.

Léxicamente, Android (androide en español) y Nexus One aluden a la novela de Philip K. Dick "¿Sueñan los androides con ovejas eléctricas?", que luego fue adoptada al cine como Blade Runner[1]. Tanto el libro como la película se centran en un grupo de robots conocidos como "replicadores" del modelo Nexus 6.

Otra curiosidad es el nombre que tienen las distintas versiones de Android hasta la 9.0. Hasta esta versión, utilizan dulces para definir sus nombres, como por ejemplo Pie, Icecream o KitKat.

[1] Película de ciencia ficción estadounidense dirigida por Ridley Scott, estrenada en 1982.

A partir de la 10.0 (esta inclusive) simplemente se utilizan los números de la versión como nombre.

2.1.1. Características principales

De las características que poseen los dispositivos que cuentan con este sistema operativo, se pueden destacar las siguientes:

· **Diseño de dispositivo**: la plataforma es adaptable a pantallas de mayor resolución, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D basada en las especificaciones de la OpenGL ES 2.0 y diseño de teléfonos tradicionales.

· **Conectividad**: Android soporta tecnologías como GSM/EDGE, UMTS, Bluetooth, Wi-Fi, LTE, NFC, WiMAX…

· **Navegador web**: el navegador web incluido en Android está basado en el motor de renderizado de código abierto WebKit, emparejado con el motor JavaScript V8 de Google Chrome.

· **Entorno de desarrollo**: inicialmente el entorno de desarrollo integrado (IDE) utilizado era Eclipse con el plugin de Herramientas de Desarrollo de Android (ADT). Ahora se considera como entorno oficial Android Studio, descargable desde la página oficial de desarrolladores de Android.

· **Google Play**: es un catálogo de aplicaciones gratuitas o de pago en el que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC.

· **Multitarea**: es la manera que tiene Android de manejar los hilos y poder así realizar varias tareas de manera simultánea.

· ***Tethering***: permite usar el dispositivo como punto de acceso alámbrico o inalámbrico.

· **Paquetes**: la aplicación final es una APK, paquete en el que está todo el contenido de esta. Además, este paquete es el propio instalador de la aplicación.

· **Multiplataforma**: puede adaptarse a diferentes tamaños de pantalla o hacer consultas específicas sobre hardware.

· **Principio de mínimo privilegio**: cada aplicación sólo tiene acceso a los componentes que necesita para llevar a cabo su trabajo.

Por otra parte, las aplicaciones Android están formadas por una serie de componentes donde cada uno de ellos es un punto de entrada por el que el sistema o un usuario ingresan a una aplicación.

Las aplicaciones Android tienen 4 tipos de componentes [2]:

· **Actividades**: es el punto de entrada de interacción con el usuario y representa una pantalla individual con una interfaz.

· **Servicios**: son componentes que se ejecutan en segundo plano y no proporcionan una interfaz de usuario. Dos posibles ejemplos pueden ser el reproductor de música del propio dispositivo o la sincronización de datos.

· **Receptores de emisiones**: permiten que el sistema entregue eventos a la aplicación fuera de un flujo de usuario habitual, es decir, las aplicaciones pueden responder a anuncios de emisión del sistema u otras aplicaciones. Por ejemplo, programar una alarma futura. Estos componentes tampoco exhiben una interfaz de usuario, aunque pueden crear notificaciones en la barra de estado que alerten al usuario cuando se produzca un evento de emisión.

· **Proveedores de contenido**: se encargan de administrar un conjunto compartido de datos de la aplicación que se pueden almacenar en el sistema de archivos, en la web, en una base de datos SQLite o en cualquier otra ubicación de almacenamiento persistente a la que tenga acceso una aplicación.

2.1.2. Arquitectura

El sistema operativo Android es una pila de componentes de software que se divide en cinco secciones y cuatro capas principales, como se muestra a continuación en el diagrama de arquitectura [3].

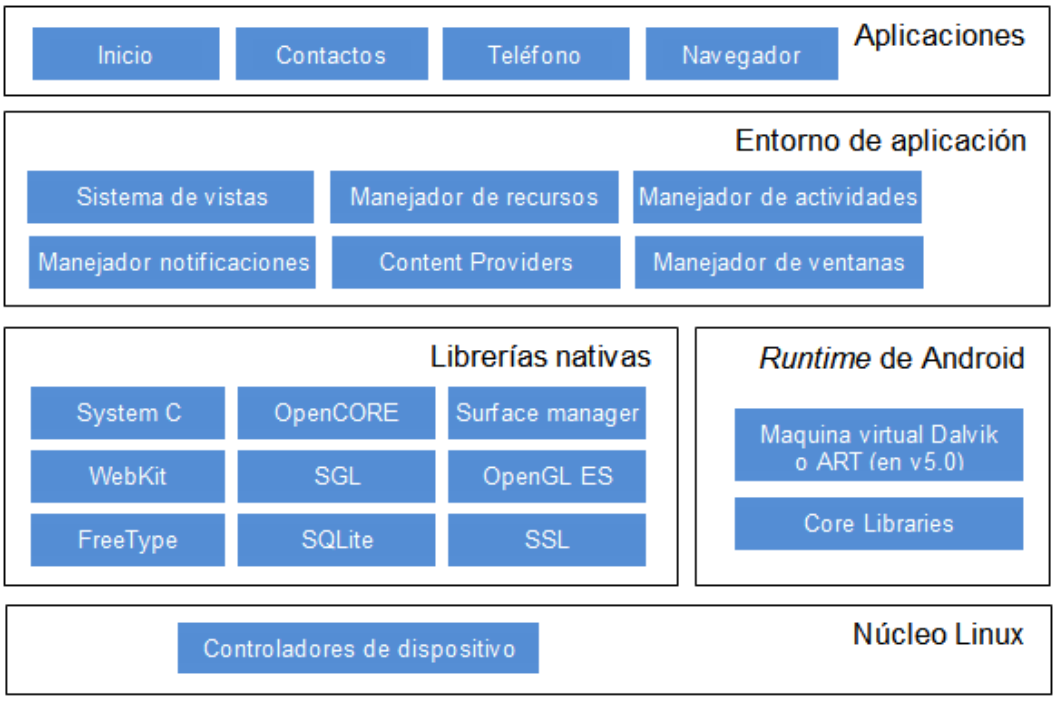

Figura 1. Arquitectura del sistema Android

El nivel de **aplicaciones** está formado por un conjunto de servicios instalados en una máquina Android. Todas ellas han de correr en la máquina virtual Dalvik para garantizar la seguridad del sistema. Normalmente las aplicaciones Android están escritas en Java o Kotlin, para desarrollar este tipo de aplicaciones se puede utilizar el SDK de Android.

El **marco de trabajo de aplicaciones** o también llamado **entorno de aplicación**, proporciona una plataforma de desarrollo libre para aplicaciones con gran riqueza e innovaciones. Esta capa se diseñó para simplificar la reutilización de componentes.

Para que los desarrollos vayan más allá, se puede hacer uso de las **librerías** nativas de Android que están compiladas en código nativo del procesador. Además, muchas de ellas utilizan proyectos de código abierto. Algunas de estas librerías son:

· **Surface Manager**: se encarga de manejar el acceso al subsistema de representación gráfica en 2D y 3D.

· **FreeType**: ofrece fuentes en bitmap y renderizado vertical.

· **SQLite**: motor potente y ligero de bases de datos relacionales disponible para todas las aplicaciones.

· **SSL**: librería que proporciona servicios de encriptación Secure Socket Layer (capa de conexión segura).

Dadas las limitaciones de los dispositivos donde ha de correr Android, Google decidió crear la máquina virtual Dalvik mencionada anteriormente, para que así las aplicaciones respondieran mejor y tuvieran un ***runtime*** similar al que ofrece la máquina virtual de Java.

Esta nueva máquina está basada en registros y utiliza los archivos *.dex* ejecutables de Dalvik (un formato optimizado para ahorrar memoria). Pero podría decirse que lo más importante es que cada aplicación se ejecuta en su propio proceso de Linux utilizando su propia versión de la máquina virtual Dalvik. Esto mejora considerablemente la seguridad del sistema.

Por último, el núcleo de Android incluye el sistema operativo Linux. Esta capa proporciona servicios como seguridad, administración de memoria, multiproceso, pilas de protocolos y compatibilidad con controladores de dispositivos. También actúa como una capa de abstracción entre el hardware y el software, por lo que esta es la única capa que depende del hardware.

2.1.3. Mercado actual de dispositivos móviles

El panorama de los smartphones se ha convertido en un duopolio en los últimos años, después de que Android y iOS desplazaran a lo desconocido a marcas como Windows Phone o BlackBerry.

Según datos de la consultora de tecnología IDC[2], los dispositivos que corren con el sistema operativo Android representaron algo más del 86% de las unidades distribuidas en 2019, y casi el 14% restante la marca Apple.

Para ver el cambio de tendencia en el mercado, simplemente hay que echar la vista atrás y ver que en 2010, la cuota de mercado combinada de Android y iOS no llegaba al 40%, compartiendo el resto del mercado con Windows Phone, BlackBerry y otros sistemas operativos.

Actualmente en 2021 y como demuestra la siguiente imagen de la página web Statcounter GlobalStats[3], Android, con un 72,19%, sigue reinando el mercado tecnológico de smartphones,

[2] Principal proveedor mundial de inteligencia de mercado, servicios de consultoría y eventos para los mercados de tecnología de la información, telecomunicaciones y tecnología de consumo.

[3] Servicio de análisis web, https://gs.statcounter.com/

por lo que esto se puede considerar un motivo directo por el cual este trabajo va dirigido a dispositivos Android.

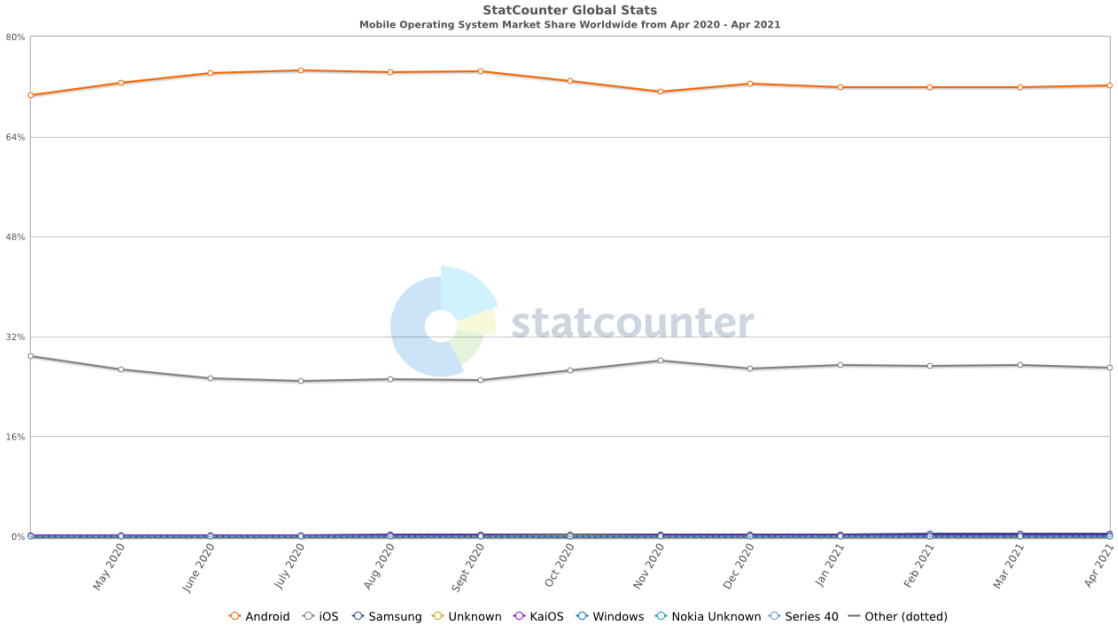

Figura 2. Comparativa de mercado: venta dispositivos móviles

2.2. ESP8266 y NodeMCU

El ESP8266 [4] es un chip Wi-Fi económico con un bloque TCP/IP y un microcontrolador confeccionado por la empresa china Espressif. Este módulo permite que otros microcontroladores se conecten a una red inalámbrica Wi-Fi y use comandos al estilo Hayes[4] para establecer una conexión TCP/IP simple.

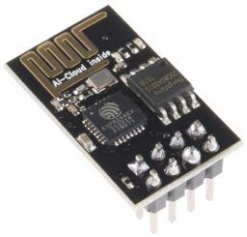

Figura 3. Chip ESP8266

[4] Lenguaje desarrollado por la compañía Hayes Communications que prácticamente se convirtió en estándar abierto de comandos para configurar y parametrizar módems.

Poco después de aparecer el ESP8266 apareció NodeMCU [4][5][6], una de las placas de desarrollo más populares basada en este chip y uno de los pilares fundamentales en los que se ha basado el ecosistema del ESP8266 durante su evolución.

Al principio se utilizaba mucho el lenguaje script Lua para programar en esta placa, ya que permitía la conexión y programación de NodeMCU más sencilla que las herramientas oficiales proporcionadas por Espressif. A pesar de que la programación en Lua tenía aspectos interesantes, no es un lenguaje tan extendido como C++ o Python, por lo que la importancia de Lua comenzó a descender.

Figura 4. Placa NodeMCU

La placa de desarrollo NodeMCU está basada concretamente en el ESP12E y expone las funcionalidades y la capacidad de este. Pero, además, añade las siguientes ventajas propias de placas de desarrollo:

- **Puerto micro USB y conversor Serie-USB.**
- **Programación sencilla a través del Micro-USB.**
- **Alimentación a través del USB.**
- **Terminales (pines) para facilitar la conexión.**
- **LED y botón de reinicio integrados.**

Por último, y el principal objetivo por el cual se ha escogido esta placa de desarrollo, es la facilidad con la que se puede instalar un sensor ultrasónico y ponerlo en funcionamiento en escasas líneas de código. En el siguiente apartado se comentan aspectos más profundamente sobre este tipo de sensores.

2.3. Sensores ultrasónicos

Los sensores de ultrasonidos o sensores ultrasónicos [7][8][9] son detectores de proximidad que trabajan libres de roces mecánicos y que detectan objetos a distancias que van desde pocos

centímetros hasta varios metros. El sensor emite un sonido y mide el tiempo que la señal tarda en regresar.

Debido a que estos sensores requieren un medio de propagación, solo funcionan en presencia de aire y pueden detectar objetos de diversas formas, colores, superficies y materiales. Puede ser líquido, sólido o polvoriento, pero debe constituir una pantalla acústica. Además, el sensor funciona con tiempo de eco, es decir, se evalúa el intervalo de tiempo entre el pulso transmitido y el eco.

Como siempre, existen pros y contras. Como este tipo de sensor no requiere contacto físico con el objeto, puede detectar objetos frágiles como pintura fresca. También detecta todos los materiales en el mismo rango, independientemente del color, sin ajustes ni factores de corrección.

El problema que presentan estos dispositivos son las zonas ciegas y las falsas alarmas. La zona ciega es la zona comprendida entre el lado sensible del detector y el alcance mínimo en el que ningún objeto puede detectarse de forma fiable.

En este trabajo se ha utilizado el sensor de distancia de ultrasonidos HC-SR05 [5], un modelo actualizado del popular HC-SR04. De las características que tiene, hay que destacar el rango de medición, que va desde los 2 centímetros hasta los 4,5 metros, por lo que se puede medir la cantidad de residuos que hay en el interior de un contenedor de manera precisa.

Figura 5. Sensor de distancia de ultrasonido HC-SR05

2.4. Competencia

Debido a la combinación tecnológica (bases de datos, sensores, navegación…) que se debe realizar para poder desarrollar una aplicación de este tipo, es complicado encontrar aplicaciones que se dediquen a monitorizar contenedores u ofrecer datos de ellos a los usuarios. Aun así, existen algunas que se acercan al propósito de este trabajo o utilizan parte de la tecnología utilizada con un fin parecido.

Con el objetivo de poder hacer una comparativa real, se ha realizado una búsqueda profunda tanto en la tienda de Google Play Store como en el repositorio de trabajos de la Universidad, llamado RiuNet. En la tienda de Google, la única aplicación que se ha desarrollado en el mismo contexto que este trabajo ha sido "Smart waste monitoring". Por otro lado, en la plataforma de RiuNet, se ha encontrado un proyecto titulado "Optimización del sistema de recogida de basuras", de Enrique Navarro Bañó y Nuno Hepanhol. Este servicio se centra en mejorar la eficiencia en la recogida de los contenedores, trazando rutas por las áreas donde se encuentran los contenedores más llenos. Se ha intentado establecer contacto con los respectivos autores tratando de obtener más información, ya que el acceso a la memoria del proyecto es privado, pero no se ha obtenido respuesta.

Por tanto, se ha descargado la primera aplicación de la tienda de Google.

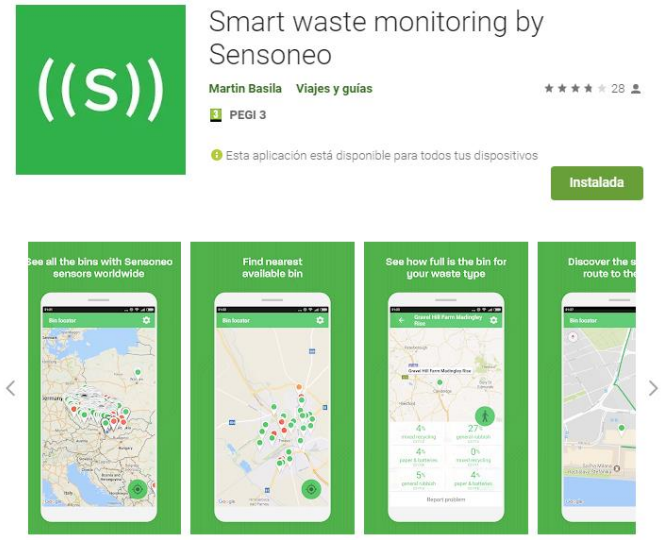

Figura 6. Aplicación Smart waste monitorin

En primer lugar, hay que destacar que la aplicación, a pesar de tener pocas descargas, su última actualización fue el 10 de junio, por lo que se puede confirmar que cuenta con soporte y no está abandonada.

Una vez descargada y utilizada, se ha podido comprobar que cuenta con los aspectos básicos para el desarrollo de una aplicación de este tipo, ya que posee ubicación en tiempo real en el mapa, la posibilidad de ver los contenedores y sus porcentajes de llenado según el tipo de residuo.

Figura 7. Pantalla del mapa de la aplicación Smart waste monitoring

Figura 8. Pantalla de un contenedor del mapa de la aplicación Smart waste monitoring

Por otra parte, a pesar de tener la opción de poder navegar a un contenedor marcado, se ha intentado probar, pero no ha funcionado, simplemente la aplicación no responde. Otro aspecto negativo es el tiempo de procesado, es decir, cada vez que se cambia de pantalla, la aplicación carga el mapa y los marcadores, en lugar de dejarlo en segundo plano como se hace en las soluciones planteadas de este trabajo.

En cuanto al mapa y a la ubicación de contenedores, no hay ninguno marcado en la Comunidad Valenciana, de hecho, sólo hay dos contenedores marcados en España, por lo que no supone un problema. Además, cuenta con un filtro con el que se puede seleccionar qué tipo de contenedores mostrar en el mapa.

Figura 9. Pantalla del mapa de España de la aplicación Smart waste monitoring

En el menú del mapa existen varias opciones. Una de ellas es “Detalles del Contenedor”, donde se puede escanear un contenedor mediante código de barras. También cuenta con “Ajustes” para personalizar la información que muestran los contenedores y existe un apartado llamado “Acerca de”, donde aparece información de la aplicación.

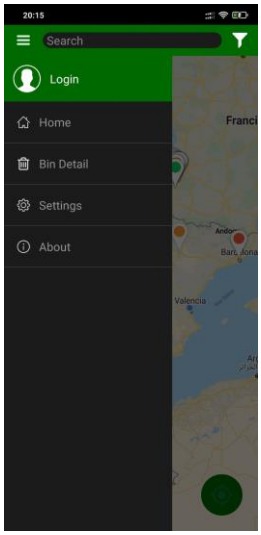

Figura 10. Pantalla del menú de la aplicación Smart waste monitoring

Cuenta también con un apartado de inicio de sesión, pero no se puede utilizar ya que no existe ningún método para registrarse. Además, no existe ningún sistema que haga más eficaz la recogida de esos contenedores, por lo que el desarrollo de la segunda aplicación está libre de competencia alguna.

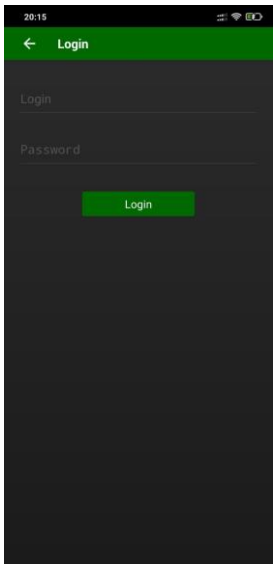

Figura 11. Pantalla de inicio de sesión de la aplicación Smart waste monitoring

3. Análisis del problema

Este capítulo se considera un apartado imprescindible en el proceso de desarrollo de una aplicación, ya que es necesario entender al usuario para concretar la funcionalidad del sistema y conocer el contexto en el que se encuentra el problema. Principalmente se establecen los requisitos de cada aplicación, tanto funcionales como no funcionales, y posteriormente se definen los casos de uso establecidos.

3.1. Requisitos del sistema

A continuación, se muestra un listado de requisitos para cada aplicación, cuyo objetivo es definir las necesidades y condiciones que deben poseer para que tengan la funcionalidad esperada por el usuario.

En primer lugar, se describen los requisitos funcionales, es decir, aquellas características demandadas por el usuario cuyo objetivo es cumplir con el comportamiento deseado del sistema.

Aplicación Tot Net

Requisitos de usuario (UR):

- **UR-R01**: registrar un nuevo usuario a partir de un nombre de usuario, un correo y una contraseña.
- **UR-R02**: registrar un nuevo usuario a partir de una cuenta Google existente.
- **UR-R03**: iniciar sesión con una cuenta estándar.
- **UR-R04**: iniciar sesión con una cuenta de Google.
- **UR-R05**: modificar nombre de usuario.
- **UR-R06**: modificar foto de perfil.
- **UR-R07**: modificar contraseña, únicamente si la cuenta es estándar.
- **UR-R08**: consultar listado de contenedores marcados como favoritos.
- **UR-R09**: recuperar sesión iniciada previamente (Inicio de sesión automático).

· **UR-R10**: cerrar sesión.

Requisitos de mapa (MR):

· **MR-R01**: mostrar mapa.

· **MR-R02**: navegar por el mapa.

· **MR-R03**: mostrar la ubicación actual.

· **MR-R04**: centrar posición en la ubicación actual.

· **MR-R05**: mostrar los diferentes tipos de contenedores en el mapa.

· **MR-R06**: acceder a la información de cada contenedor desde su marcador en el mapa.

Requisitos de contenedor (CR):

· **CR-R01**: marcar/desmarcar un contenedor como favorito.

· **CR-R02**: mostrar la calle en la que se encuentra el contenedor.

· **CR-R03**: mostrar el tipo de residuo que almacena el contenedor.

· **CR-R04**: mostrar la cantidad de residuos que tiene el contenedor.

· **CR-R05**: mostrar una sugerencia de acudir a otro contenedor si el mismo está muy lleno.

· **CR-R06**: permitir trazar una ruta hacia el contenedor.

Requisitos de ruta (RR):

· **RR-R01**: abrir ruta en Google Maps.

· **RR-R02**: mostrar trayecto al contenedor seleccionado.

· **RR-R03**: salir de la ruta creada y volver a la aplicación.

Aplicación Tot Net Recogedores

Requisitos de mapa (MR):

· **MR-R01**: mostrar mapa.

- **MR-R02**: navegar por el mapa.
- **MR-R03**: mostrar la ubicación actual.
- **MR-R04**: centrar posición en la ubicación actual.
- **MR-R05**: mostrar los diferentes tipos de contenedores en el mapa.
- **MR-R06**: acceder a la información de cada contenedor desde su marcador en el mapa.
- **MR-R07**: generar ruta por los contenedores más llenos.

Requisitos de contenedor (CR):

- **CR-R01**: mostrar la calle en la que se encuentra el contenedor.
- **CR-R02**: mostrar el tipo de residuo que almacena el contenedor.
- **CR-R03**: mostrar la cantidad de residuos que tiene el contenedor.
- **CR-R04**: mostrar una sugerencia de acudir a otro contenedor si el mismo está muy lleno.
- **CR-R05**: permitir trazar una ruta hacia el contenedor.

Requisitos de ruta (RR):

- **RR-R01**: abrir ruta en Google Maps.
- **RR-R02**: mostrar trayecto al contenedor seleccionado.
- **RR-R03**: salir de la ruta creada y volver a la aplicación.

Y, en segundo lugar, se definen los requisitos no funcionales para ambas aplicaciones, aquellos que hacen referencia a restricciones o condiciones que debe tener los sistemas, sin llegar a necesitar desarrollar una funcionalidad específica para su cumplimiento:

- **Seguridad**: la configuración del sistema sólo puede ser modificada por un administrador.
- **Rendimiento**: la aplicación debe ser ejecutada de manera fluida y agradable para el usuario.

· **Disponibilidad**: la aplicación debe ser funcional siempre y cuando se establezca una conexión a internet.

· **Integridad**: el modelo de datos debe estar relacionado para mantener la integridad de los datos.

· **Accesibilidad**: la aplicación debe ser legible y tiene que seguir los estándares de accesibilidad establecidos por Google.

· **Estabilidad**: en caso de ocurrir algún error, la aplicación debe ser capaz de gestionarlo y avisar de manera clara y concisa.

· **Mantenimiento**: la aplicación debe ser actualizada y mantenida en la medida de lo posible.

· **Optimización**: el consumo de datos y batería debe controlarse de manera eficaz, eliminando aquellos procesos que no tengan importancia.

· **Idioma**: el idioma principal de la aplicación es el castellano; ES-es.

· **Compatibilidad**: la aplicación debe poder ejecutarse correctamente en cualquier versión Android superior a la 5.0, esta inclusive.

3.2. Casos de uso

Los diagramas de casos de uso es una de las herramientas más utilizadas en la fase de análisis de un proyecto software. Gracias a ellos se puede describir la interacción entre el usuario y el sistema, es decir, las actividades que realizan estos en cada una de las aplicaciones para llevar a cabo algún proceso.

Los actores de la primera aplicación son el usuario no registrado y el usuario registrado, y el de la segunda aplicación simplemente es el recogedor. En ambas aplicaciones la ruta actúa como actor externo, ya que el desarrollo no pertenece a este trabajo.

En primer lugar, se definen los casos de uso de la primera aplicación (Tot Net). Para poder utilizarla, el usuario debe registrarse si no lo está. Para ello tiene dos opciones: registrar una cuenta estándar o registrarse a través de una cuenta Google ya existente. A continuación, se describen los casos de uso relacionados mediante escenarios.

CU1 – Registrar cuenta estándar

Referencia:	1
Nombre:	Registrar cuenta estándar
Descripción:	El usuario abre la aplicación y accede a la pantalla principal. En esta elige la opción de registrar una cuenta normal y rellena los campos requeridos. Posteriormente el sistema verifica que los datos han sido introducidos correctamente y registra el usuario
Actor:	Usuario no registrado
Relaciones:	
Precondición:	

CU2 – Registrarse a través de Google

Referencia:	2
Nombre:	Registrarse a través de Google
Descripción:	El usuario abre la aplicación y accede a la pantalla principal. En esta elige la opción de continuar con Google y elige una cuenta existente en el dispositivo. Posteriormente el sistema verifica la existencia de esa cuenta y registra el usuario
Actor:	Usuario no registrado
Relaciones:	
Precondición:	

CU3 – Iniciar sesión

Referencia:	3
Nombre:	Iniciar sesión
Descripción:	El usuario abre la aplicación y accede a la pantalla principal. Introduce un correo y una contraseña y presiona “Iniciar sesión”. El sistema revisa los datos introducidos y una vez validados, el usuario inicia sesión
Actor:	Usuario registrado
Relaciones:	Si el usuario ha iniciado sesión previamente, se recupera esa sesión y se inicia sesión automáticamente
Precondición:	

CU4 – Iniciar sesión a través de Google

Referencia:	4
Nombre:	Iniciar sesión a través de Google
Descripción:	El usuario abre la aplicación y accede a la pantalla principal. En esta elige la opción de continuar con Google y el sistema inicia sesión con la cuenta de Google registrada anteriormente
Actor:	Usuario registrado
Relaciones:	Si el usuario ha iniciado sesión previamente, se recupera esa sesión y se inicia sesión automáticamente
Precondición:	

CU5 – Iniciar sesión automáticamente

Referencia:	5
Nombre:	Iniciar sesión automáticamente
Descripción:	El usuario ya ha iniciado sesión recientemente, por lo que el sistema recupera esa sesión y la inicia automáticamente una vez el usuario abre la aplicación
Actor:	Usuario registrado
Relaciones:	
Precondición:	

Estos casos de uso se representan junto con sus actores en el siguiente diagrama de casos de uso:

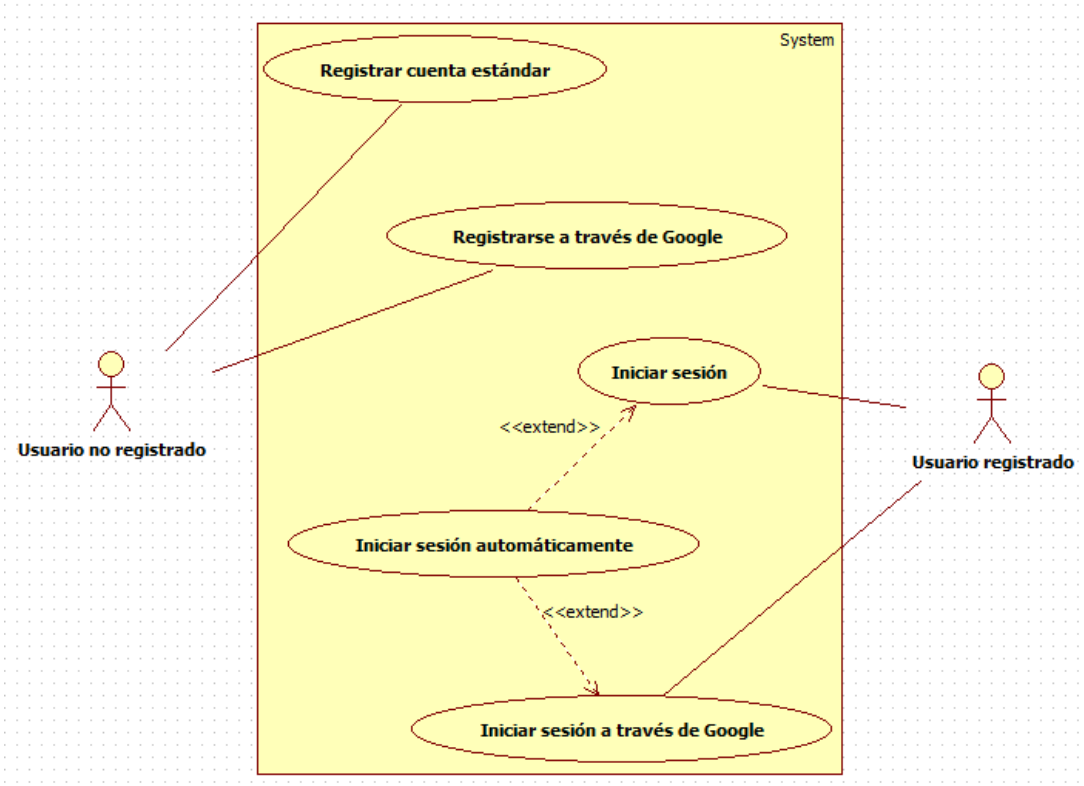

Figura 12. Casos de uso registro/inicio de sesión de la aplicación Tot Net

Una vez accedido a la aplicación tras iniciar sesión, el usuario cuenta con más opciones con las que interactuar con el sistema. Una de ellas es centrar su posición en el mapa:

CU6 – Mostrar ubicación actual

Referencia:	6
Nombre:	Mostrar ubicación actual
Descripción:	Una vez el usuario se encuentra en la pantalla del mapa, presiona el botón situado en la parte inferior derecha de la pantalla y centra su posición en el mapa
Actor:	Usuario registrado
Relaciones:	
Precondición:	

También puede navegar por el mapa para visualizar los contenedores de la zona:

CU7 – Navegar por el mapa

Referencia:	7
Nombre:	Navegar por el mapa
Descripción:	En la pantalla del mapa, el usuario navega por el mapa arrastrando la pantalla. También puede acercar y alejar la cámara
Actor:	Usuario registrado
Relaciones:	
Precondición:	

En el mismo mapa, tiene la posibilidad de ver la información de cada contenedor:

CU8 – Consultar información de contenedor

Referencia:	8
Nombre:	Consultar información de contenedor
Descripción:	En la pantalla del mapa, el usuario presiona el marcador de un contenedor y se muestra la información relativa a este contenedor
Actor:	Usuario registrado
Relaciones:	
Precondición:	

Y ese mismo contenedor lo puede marcar o desmarcar como favorito:

CU9 – Marcar contenedor como favorito

Referencia:	9
Nombre:	Marcar contenedor como favorito
Descripción:	En la pantalla del contenedor, el usuario marca el icono de favorito y el sistema guarda el contenedor en la lista de contenedores favoritos del usuario
Actor:	Usuario registrado
Relaciones:	
Precondición:	El contenedor no debe ser favorito

CU10 – Desmarcar contenedor como favorito

Referencia:	10
Nombre:	Desmarcar contenedor como favorito
Descripción:	En la pantalla del contenedor, el usuario desmarca el icono de favorito y el sistema borra el contenedor de la lista de contenedores favoritos del usuario
Actor:	Usuario registrado
Relaciones:	
Precondición:	El contenedor debe ser favorito

Además, en la misma pantalla del contenedor puede generar una ruta en Google Maps para poder llegar a él:

CU11 – Solicitar ruta hacia contenedor

Referencia:	11
Nombre:	Solicitar ruta hacia contenedor
Descripción:	En la pantalla del contenedor, el usuario presiona el botón “Ir” y el sistema procesa la solicitud de ruta
Actor:	Usuario registrado
Relaciones:	
Precondición:	El contenedor tiene un porcentaje ideal

Por otra parte, en la misma pantalla del mapa puede abrir un menú:

CU12 – Abrir menú

Referencia:	12
Nombre:	Abrir menú
Descripción:	En la pantalla del mapa, el usuario presiona el botón situado en la parte superior izquierda y abre el menú principal
Actor:	Usuario registrado
Relaciones:	
Precondición:	

Dentro del menú existen las siguientes opciones:

CU13 – Consultar perfil

Referencia:	13
Nombre:	Consultar perfil
Descripción:	Con el menú principal abierto, el usuario presiona “Mi perfil” y accede a visualizar su foto de perfil, su usuario y su correo electrónico
Actor:	Usuario registrado
Relaciones:	
Precondición:	

CU14 – Consultar contenedores favoritos

Referencia:	14
Nombre:	Consultar contenedores favoritos
Descripción:	Con el menú principal abierto, el usuario presiona ”Favoritos” y visualiza los contenedores que tiene marcados como favoritos en ese momento
Actor:	Usuario registrado
Relaciones:	
Precondición:	

CU15 – Cerrar sesión

Referencia:	15
Nombre:	Cerrar sesión
Descripción:	Con el menú principal abierto, el usuario presiona ”Cerrar sesión” y el sistema cierra la sesión actual, devolviendo al usuario a la pantalla principal de registro/inicio de sesión
Actor:	Usuario registrado
Relaciones:	
Precondición:	Tener una sesión iniciada

Dentro del perfil, el usuario puede realizar varias modificaciones:

CU16 – Cambiar foto de perfil

Referencia:	16
Nombre:	Cambiar foto de perfil
Descripción:	En la pantalla del perfil, el usuario presiona su avatar y elige su nueva foto de perfil desde los archivos del dispositivo
Actor:	Usuario registrado
Relaciones:	
Precondición:	

CU17 – Cambiar nombre de usuario

Referencia:	17
Nombre:	Cambiar nombre de usuario
Descripción:	En la pantalla del perfil, el usuario presiona ”Cambiar nombre de usuario” y rellena el campo con el nuevo nombre de usuario
Actor:	Usuario registrado
Relaciones:	
Precondición:	

CU18 – Cambiar contraseña

Referencia:	18
Nombre:	Cambiar contraseña
Descripción:	En la pantalla del perfil, el usuario presiona ”Cambiar contraseña” y rellena los campos de contraseña actual y nueva contraseña
Actor:	Usuario registrado
Relaciones:	
Precondición:	La cuenta debe ser estándar, no de Google

Estos casos de uso se representan junto con su actor en el siguiente diagrama de casos de uso:

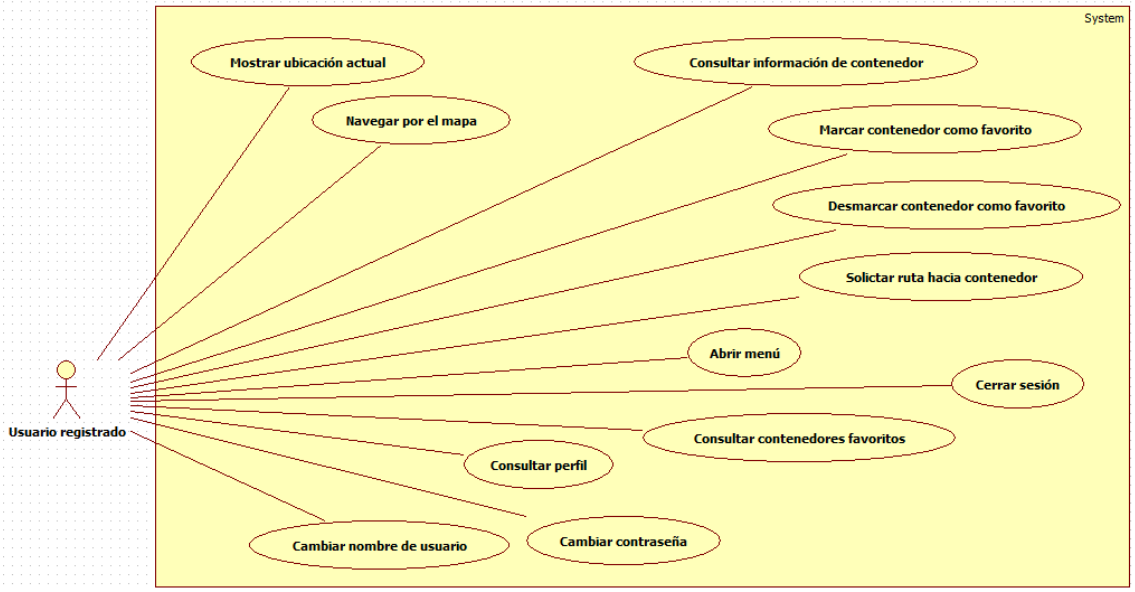

Figura 13. Casos de uso del usuario registrado de la aplicación Tot Net

En segundo y último lugar, se definen los casos de uso de la segunda aplicación (Tot Net Recogedores). A esta aplicación solamente tienen acceso los recogedores de basura, es decir, solamente ellos pueden descargar esta aplicación.

Como el acceso es restringido, los usuarios recogedores no necesitan registrarse o iniciar sesión, por lo que los escenarios en los que pueden interactuar con el sistema son muy parecidos a los de la primera aplicación, pero con algún ligero cambio.

En la pantalla del mapa se encuentran los siguientes casos de uso:

CU1 – Mostrar ubicación actual

Referencia:	1
Nombre:	Mostrar ubicación actual
Descripción:	Una vez el usuario se encuentra en la pantalla del mapa, presiona el botón situado en la parte inferior derecha de la pantalla y centra su posición en el mapa
Actor:	Recogedor
Relaciones:	
Precondición:	

CU2 – Navegar por el mapa

Referencia:	2
Nombre:	Navegar por el mapa
Descripción:	En la pantalla del mapa, el usuario navega por el mapa arrastrando la pantalla. También puede acercar y alejar la cámara
Actor:	Recogedor
Relaciones:	
Precondición:	

CU3 – Consultar información de contenedor

Referencia:	3
Nombre:	Consultar información de contenedor
Descripción:	En la pantalla del mapa, el usuario presiona el marcador de un contenedor y se muestra la información relativa a este contenedor
Actor:	Recogedor
Relaciones:	
Precondición:	

En la misma pantalla del contenedor, el recogedor puede generar una ruta en Google Maps para poder llegar a él:

CU4 – Solicitar ruta hacia contenedor

Referencia:	4
Nombre:	Solicitar ruta hacia contenedor
Descripción:	En la pantalla del contenedor, el usuario presiona el botón “Ir” y el sistema procesa la solicitud de ruta
Actor:	Recogedor
Relaciones:	
Precondición:	

La única diferencia respecto a la primera aplicación es la siguiente. En la pantalla del mapa, aparece en la parte central superior de la interfaz un botón con el que es posible generar una ruta

con paradas. Así, el recogedor puede realizar una ruta eficaz por los contenedores más llenos de la zona:

CU5 – Generar ruta con paradas

Referencia:	5
Nombre:	Generar ruta con paradas
Descripción:	En la pantalla del mapa, el usuario presiona el botón "Generar ruta" y el sistema envía la consulta a Google Maps, la cual devuelve una ruta con paradas en distintos contenedores
Actor:	Recogedor
Relaciones:	
Precondición:	

Por tanto, el diagrama de casos de uso de la segunda aplicación queda así:

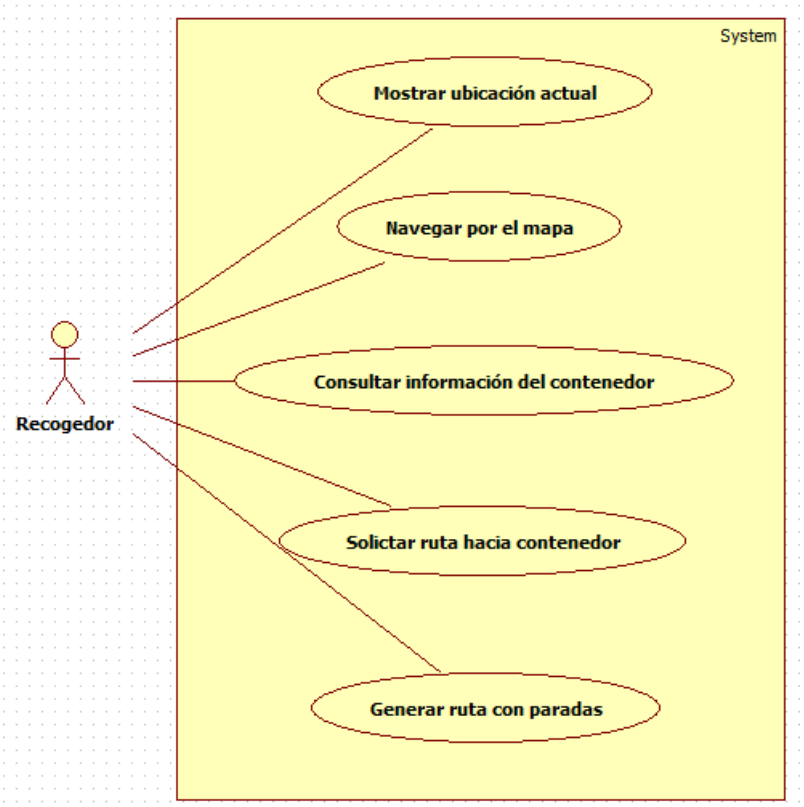

Figura 14. Casos de uso del usuario recogedor de la aplicación Tot Net Recogedores

4. Diseño de la solución

En este apartado se comenta el diseño del proyecto, tanto de la aplicación Tot Net como de la aplicación Tot Net Recogedores. Al principio se muestra la arquitectura de cada uno de los sistemas, seguidamente se incide en el diseño de cada una de las aplicaciones, estableciendo una división por capas y, por último, se especifica la tecnología utilizada en ambos servicios.

4.1. Arquitectura del sistema

A continuación, se exponen cada uno de los esquemas que resume de forma muy general la arquitectura que tiene cada sistema, desde la aplicación y la placa NodeMCU con el sensor ultrasónico hasta la base de datos donde se almacenan tanto los usuarios registrados como la información de cada contenedor. Como se puede observar en la siguiente imagen, la aplicación Tot Net se queda dividida en dos bloques diferenciados.

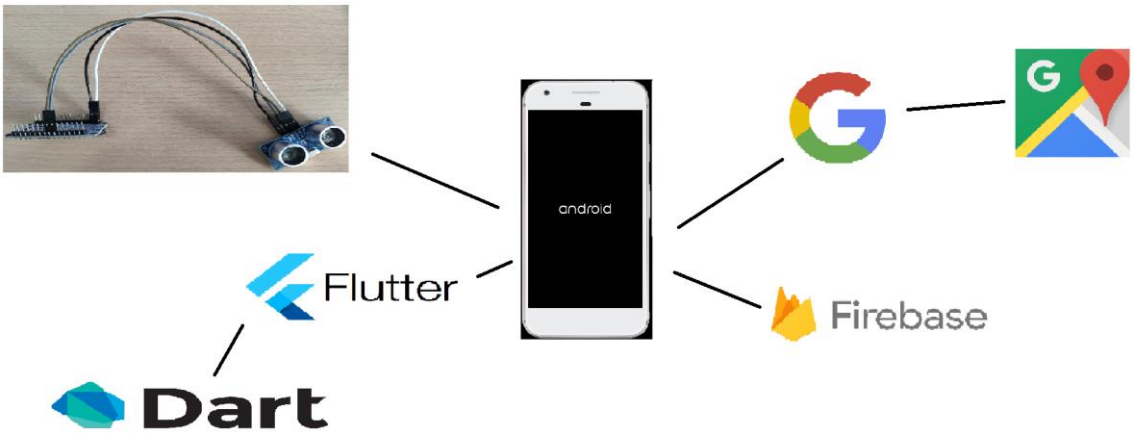

Figura 15. Estructura del sistema de la aplicación Tot Net

En la parte de la derecha se encuentra el servidor de Firebase, este realiza las funciones tanto de registro como de inicio de sesión, y, además, lleva a cabo la gestión de la información de cada contenedor y el almacenaje de los datos recogidos por el sensor de ultrasonidos.

También se hace uso de las API de Google y de Google Maps, desde la primera es posible el registro automático siempre que el usuario lo haya autorizado. De esta forma, se está accediendo a la aplicación vía *oAuth* – autorización abierta – sin necesidad de hacer uso del registro estándar de la aplicación. Con la API de Google Maps se puede manejar todo lo relacionado con los mapas en la aplicación, así como la geolocalización instantánea a través de la señal GPS o de internet. Además, permite posicionar sobre el mapa la localización de cada contenedor y al

seleccionar uno, ofrece la posibilidad de abrir la aplicación nativa de Google Maps para obtener el trayecto desde la posición actual hasta él.

Por otro lado, en la parte izquierda, está el *framework* de Flutter, que utiliza el lenguaje Dart, con el cual se ha desarrollado todo este proyecto, y el sensor ultrasónico con la placa NodeMCU para medir los residuos que contiene un contenedor. Esta placa también está conectada al servidor de Firebase, ya que se encarga de enviar periódicamente los datos recogidos en el contenedor.

El esquema de la aplicación Tot Net Recogedores es similar al mostrado arriba, únicamente desaparece la API de Google para el registro de usuarios, ya que esta aplicación no necesita de autentificación porque sólo tienen acceso a ella los recogedores de basura. Por lo que el resultado final del esquema es el mostrado en la siguiente figura.

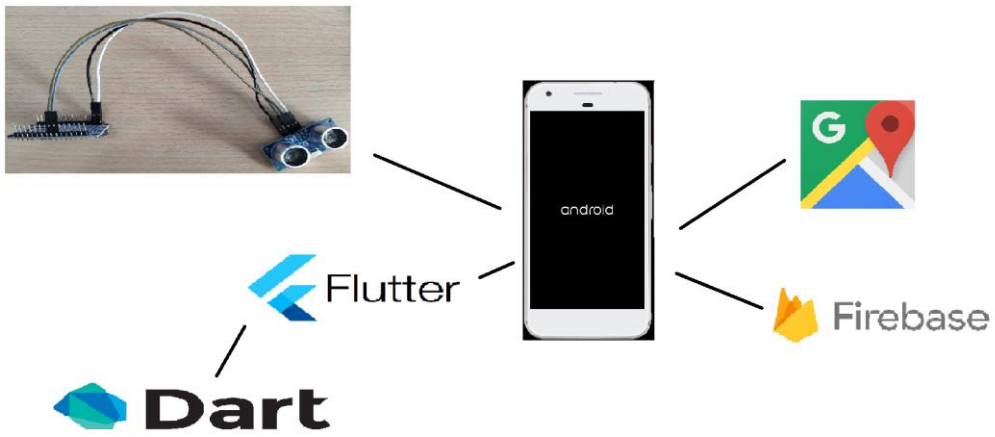

Figura 16. Estructura del sistema de la aplicación Tot Net Recogedores

En las dos aplicaciones se definen, tanto en el lado del servidor como del cliente Android, los modelos necesarios para el correcto funcionamiento de cada una, así como la creación de los controladores que se encargan de realizar las acciones necesarias para conseguir que cada sistema tenga la funcionalidad deseada.

Concretamente, los modelos son aquellas representaciones de los objetos creadas en la base de datos y los controladores gestionan la lógica de la aplicación, cumpliendo así cada una de las peticiones realizadas por el usuario. Hay que destacar también que, gracias a estos, se hace posible la comunicación entre los distintos sistemas que interactúan con cada una de las aplicaciones. Por último, la interfaz del usuario está definida por vistas, gracias a ellas los usuarios pueden interactuar con el dispositivo y realizar cada una de las funcionalidades ofrecidas.

4.2. Diseño detallado

Para tener un producto final completo y preciso es imprescindible establecer un diseño detallado sobre el que se sostenga el mismo. Para ello, se han descompuesto ambas aplicaciones en tres capas [10], cada una de ellas con un comportamiento y un objetivo diferente.

4.2.1. Capa de presentación

Para establecer un diseño de la capa de presentación cercano a la versión final de ambos productos, se ha utilizado la herramienta Adobe XD[5], la cual permite realizar un prototipado del sistema, sencillo pero completo, gracias a la cantidad de elementos que se pueden agregar. Además, permite simular la navegación entre las distintas interfaces de usuario diseñadas.

Es normal que haya diferencias respecto al resultado final de cada una de las aplicaciones, ya que como bien se ha mencionado anteriormente, son prototipos y el objetivo de estos es definir a grandes rasgos el aspecto que debe tener cada pantalla, simplemente para tener una idea básica y seguir unos patrones de diseño.

Primero se muestran las pantallas que debe tener la aplicación Tot Net. Cuando se abre la aplicación, la primera pantalla que aparece es la de inicio de sesión. Desde esta pantalla, el usuario puede iniciar sesión con una cuenta estándar rellenando los campos de correo y contraseña, con una cuenta existente de Google o puede dirigirse a la pantalla de registro para crear una nueva cuenta estándar. En esta última es necesario un nombre de usuario, un correo electrónico y una contraseña.

[5] https://www.adobe.com/products/xd.html

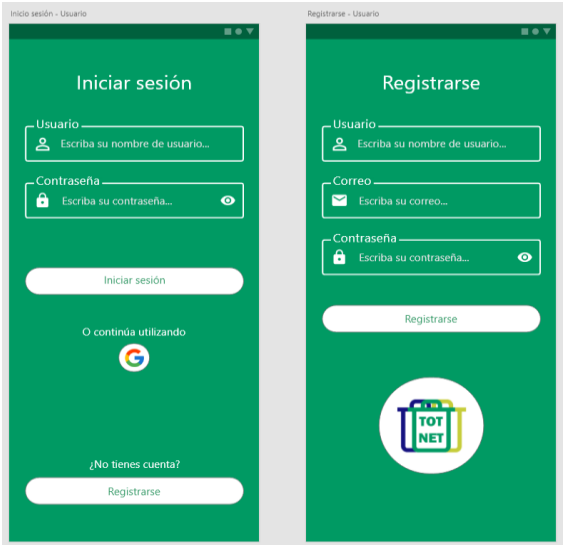

Figura 17. Pantallas de inicio de sesión y registro de la aplicación Tot Net

Una vez se ha iniciado sesión, el usuario puede centrar su posición en el mapa y navegar por él para visualizar los diferentes tipos de contenedores que hay en su zona.

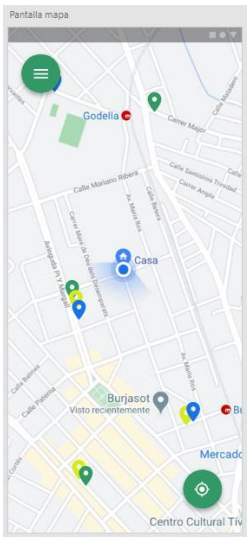

Figura 18. Pantalla del mapa de la aplicación Tot Net

Si presiona sobre un contenedor, puede consultar la información más relevante del mismo, como por ejemplo en qué calle se encuentra, de qué tipo es, cuántos residuos contiene, si es recomendable o no tirar la basura en ese contenedor… También tiene la opción de marcar ese contenedor como favorito o incluso generar una ruta en Google Maps para poder llegar a él.

Figura 19. Pantallas de los contendores de la aplicación Tot Net

En la misma pantalla del mapa, el usuario tiene disponible el botón del menú, el cual proporciona acceso al resto de funcionalidades de la aplicación. En primer lugar, puede modificar su perfil, cambiando la foto, el nombre de usuario e incluso la contraseña si la cuenta no es de Google.

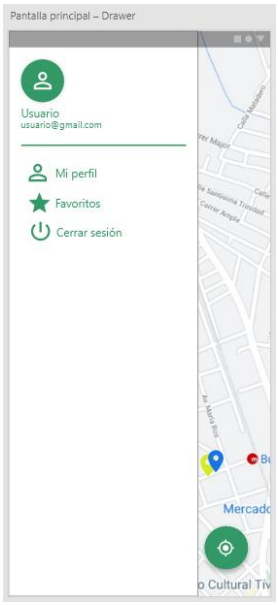

Figura 20. Pantalla del menú de la aplicación Tot Net

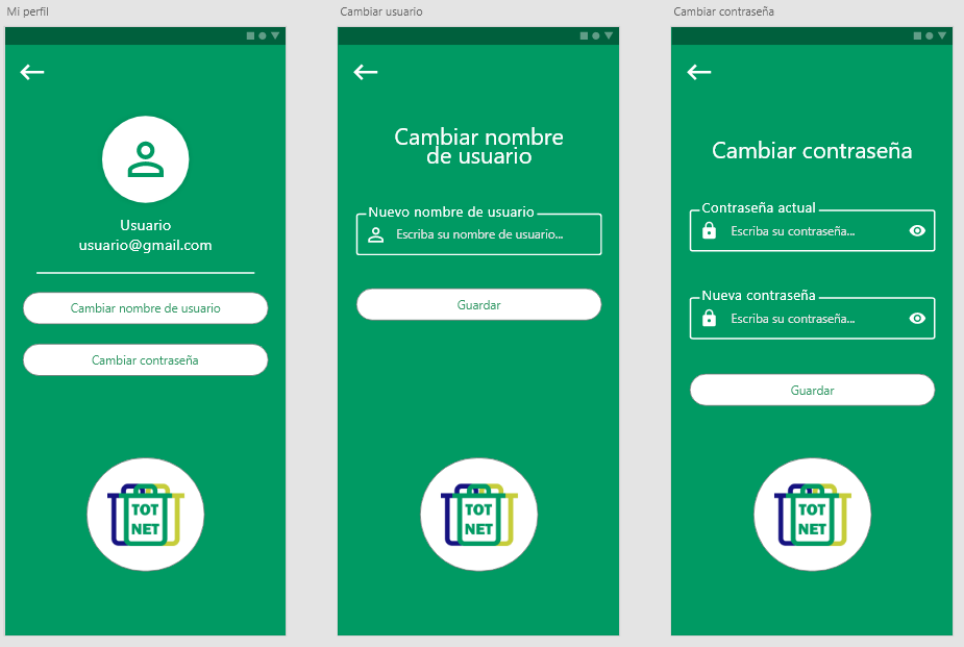

Figura 21. Pantalla del perfil de la aplicación Tot Net

En segundo lugar, puede consultar el listado de contenedores que tiene marcados como favoritos.

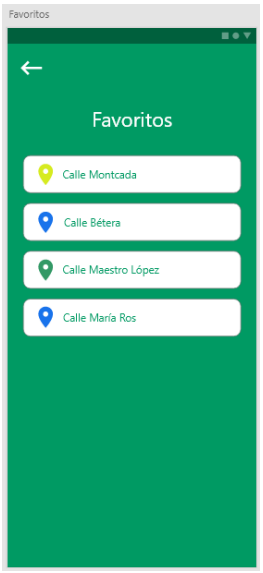

Figura 22. Pantalla del listado de contenedores favoritos de la aplicación Tot Net

Por último, en el menú, el usuario tiene la opción de cerrar sesión y volver a la pantalla principal de la aplicación.

A continuación, se muestran las pantallas por las que está compuesta la segunda aplicación, Tot Net Recogedores. Esta no tiene ni pantalla de inicio de sesión ni menú, aunque posee un

botón en la pantalla del mapa para generar una ruta en Google Maps por los contenedores más llenos.

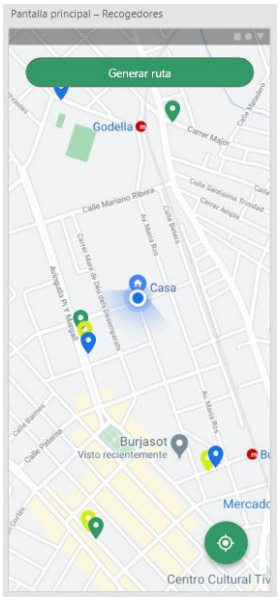

Figura 23. Pantalla del mapa de la aplicación Tot Net Recogedores

E igual que en la aplicación Tot Net, se puede presionar sobre un contenedor para visualizar su información y generar un trayecto hasta él, aunque no se puede marcar como favorito.

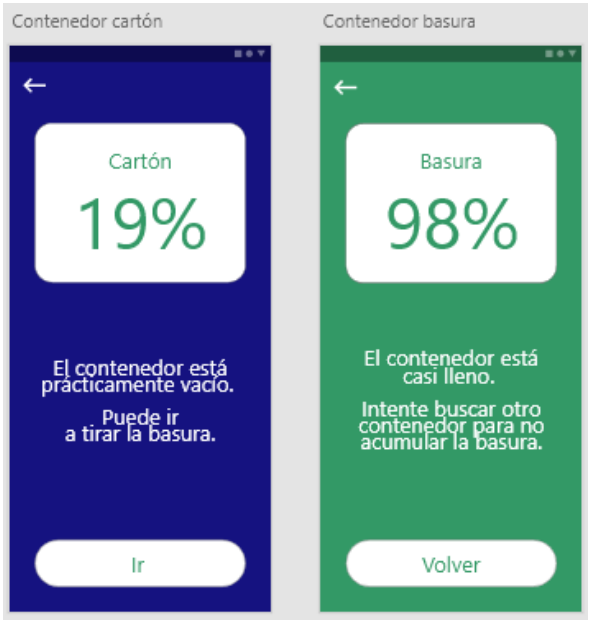

Figura 24. Pantallas de los contenedores en la aplicación Tot Net Recogedores

4.2.2. Capa lógica

En esta capa, también llamada capa de negocio, se definen los flujos que sigue cada uno de los procesos para el correcto funcionamiento de los sistemas. Aquí se recibe la solicitud del usuario y se envía la respuesta procesada. Además, se comunica con la capa de presentación para recibir solicitudes, mostrar resultados, comunicarse con la capa de datos y solicitar al gestor de esta que guarde o recupere la información.

Básicamente, para que quede una capa lógica completa, se deben cubrir cada una de las funcionalidades definidas en los casos de uso del apartado 3.2. En primer lugar, se explican de manera más detallada las acciones que el sistema de la aplicación Tot Net debe realizar para llevar a cabo cada proceso. Posteriormente se describen del mismo modo las de la aplicación Tot Net Recogedores.

Aplicación Tot Net

· **CU1 – Registrar cuenta estándar**: el sistema debe recoger los datos introducidos por el usuario en la pantalla de registro, comprobar que son válidos, crear el usuario, enviarlo al servidor de Firebase e iniciar sesión.

· **CU2 - Registrarse a través de Google**: si el usuario decide registrar una cuenta mediante una existente de Google, aparece la pantalla de autorización propia de Google. En caso de que se acepte y se elija una cuenta, el sistema sigue el mismo proceso que en el anterior caso de uso.

· **CU3 – Iniciar sesión**: el sistema debe recoger los datos introducidos en la pantalla de inicio de sesión, comprobar que son válidos realizando una consulta al servidor de Firebase e iniciar sesión.

· **CU4 – Iniciar sesión a través de Google**: si el usuario elige iniciar sesión con una cuenta de Google, el sistema también realiza una consulta a Firebase para obtener la cuenta elegida en el registro e inicia sesión.

· **CU5 – Iniciar sesión automáticamente**: si el usuario ha iniciado sesión recientemente, ya sea con una cuenta estándar o con una cuenta de Google, el sistema debe recuperar la sesión previa e iniciar sesión con ella automáticamente.

· **CU6 – Mostrar ubicación actual**: la aplicación tiene que centrar la posición del usuario en el mapa cuando se presione el botón indicado.

· **CU7 – Navegar por el mapa**: el sistema debe permitir al usuario arrastrar el mapa, acercar o alejar la cámara y visualizar los contenedores mediante marcadores.

· **CU8 – Consultar información de contenedor**: al presionar el marcador de un contenedor en el mapa, la aplicación debe mostrar una pantalla con la información relevante de ese contenedor, es decir, el nombre de la calle donde se encuentra, la cantidad de residuos que contiene realizando una consulta al servidor de Firebase donde se encuentran los datos almacenados por el sensor, el tipo de contenedor, si es recomendable o no acudir a ese contenedor o si está o no marcado como favorito.

· **CU9 – Marcar contenedor como favorito**: cuando el usuario marque como favorito un contenedor, el sistema debe añadirlo a su lista de favoritos y almacenarlo en el servidor de Firebase.

· **CU10 – Desmarcar contenedor como favorito**: igual que en el anterior caso de uso, pero se debe eliminar el contenedor desmarcado como favorito tanto del listado como de la base de datos de Firebase.

· **CU11 – Solicitar ruta hacia contenedor**: el sistema tiene que realizar una consulta vía HTTPS a la API de Google para generar una ruta hacia el contenedor seleccionado y abrirla en la aplicación nativa de Google Maps.

· **CU12 – Abrir menú**: la aplicación abre un menú lateral en la parte izquierda cuando el usuario presione el botón correspondiente. En este menú deben aparecer las secciones de “Mi perfil”, “Favoritos” y “Cerrar sesión”.

· **CU13 – Consultar perfil**: si el usuario accede a la sección “Mi perfil” del menú, el sistema muestra una pantalla con su foto de perfil, nombre de usuario y correo electrónico.

· **CU14 – Consultar contenedores favoritos**: cuando el usuario acceda a la sección de “Favoritos”, la aplicación debe recuperar ese listado haciendo una consulta al servidor de Firebase. Además, muestra de manera más resumida la información de cada contenedor.

· **CU15 – Cerrar sesión**: el sistema cierra la sesión actual del usuario y lo devuelve a la pantalla de registro/inicio de sesión.

· **CU16 – Cambiar foto de perfil**: la aplicación permite al usuario cambiar su foto de perfil eligiendo una imagen desde el gestor de archivos de su dispositivo. Para ello, el sistema debe solicitar los permisos adecuados. Una vez se haya cambiado la foto de perfil, la aplicación la almacena en la base de datos.

· **CU17 – Cambiar nombre de usuario**: al rellenar el campo con el nuevo nombre de usuario, el sistema lo almacena en la base de datos realizando una petición a Firebase.

· **CU18 – Cambiar contraseña**: para poder modificar la contraseña, el usuario debe introducir la contraseña actual y la nueva contraseña. Si la contraseña actual es correcta y la nueva contraseña es válida (tras realizar la consulta en la base de datos y la verificación correspondiente), la aplicación notifica a Firebase de ese cambio.

Aplicación Tot Net Recogedores

· **CU1 – Mostrar ubicación actual**: la aplicación tiene que centrar la posición del recogedor en el mapa cuando se presione el botón indicado.

· **CU2 – Navegar por el mapa**: el sistema debe permitir al recogedor arrastrar el mapa, acercar o alejar la cámara y visualizar los contenedores mediante marcadores.

· **CU3 – Consultar información del contenedor**: al presionar el marcador de un contenedor en el mapa, la aplicación debe mostrar una pantalla con la información relevante de ese contenedor, es decir, el nombre de la calle donde se encuentra, la cantidad de residuos que contiene realizando una consulta al servidor de Firebase donde se encuentran los datos almacenados por el sensor, el tipo de contenedor y si es recomendable o no acudir a ese contenedor.

· **CU4 – Solicitar ruta hacia contenedor**: el sistema tiene que realizar una consulta vía HTTPS a la API de Google para generar una ruta hacia el contenedor seleccionado y abrirla en la aplicación de Google Maps.

· **CU5 – Generar ruta con paradas**: cuando el recogedor presione el botón de “Generar ruta” en la parte central superior de la pantalla del mapa, el sistema debe realizar una consulta vía HTTPS a la API de Google Maps para trazar una ruta por los contenedores más llenos.

4.2.3. Capa de persistencia

A la hora de realizar el diseño de esta capa, se debe tener en cuenta tanto la parte del servidor como la parte del cliente de cada una de las aplicaciones. En el servidor existen dos tipos de entidades, el usuario y el contenedor. Tanto la aplicación Tot Net como Tot Net Recogedores, acceden al mismo servidor de Firebase, pero sólo la primera accede a los usuarios, ya que la segunda, como se dijo, no contempla el registro de usuarios.

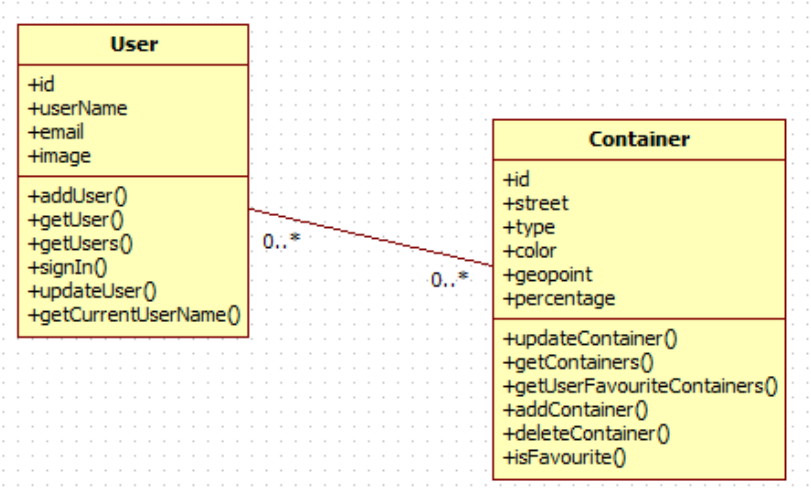

Figura 25. Entidades del servidor de Firebase

Como se puede observar en la imagen anterior, el usuario está compuesto por los siguientes campos:

- **Id**: conjunto de caracteres y números que sirven para identificar al usuario.
- **UserName**: nombre del usuario.
- **Email**: correo electrónico del usuario.
- **Image**: imagen de perfil del usuario.

Y el contenedor tiene los siguientes:

- **Id**: conjunto de caracteres y números que sirven para identificar al contenedor.
- **Street**: calle en la que se ubica el contenedor.
- **Type**: tipo de residuos que contiene el contenedor.
- **Color**: color que tiene el marcador del contenedor en el mapa.

· **Geopoint**: latitud y longitud del punto donde se encuentra el contenedor.

· **Percentage**: porcentaje de cantidad de residuos que tiene el contenedor.

Por otra parte, en el lado del cliente están las clases y métodos, ya que Flutter y Dart permiten una programación orientada a objetos. La estructura de ambas aplicaciones, definida con diagramas de clases, queda reflejada en la siguiente imagen.

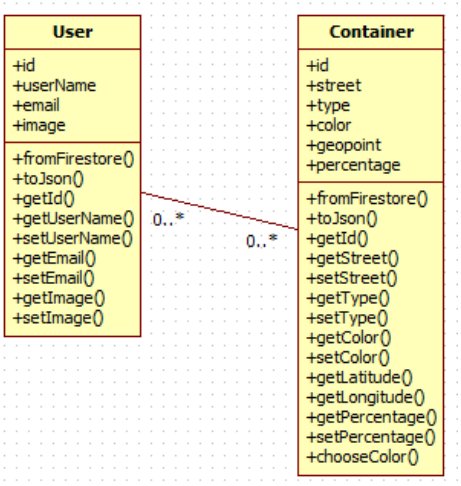

Figura 26. Entidades de la aplicación Tot Net

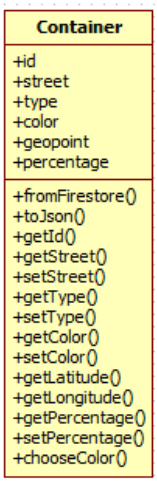

Figura 27. Entidad de la aplicación Tot Net Recogedores

Al diseñar estas tres capas, se asegura que el sistema tenga una estructura bien definida y consistente. Se han establecido desde las vistas que va a tener cada pantalla, hasta los modelos que van a ser instanciados en la base de datos y los flujos que tiene cada uno de los procesos.

4.3. Tecnologías utilizadas

En este apartado se comentan las tecnologías que se han utilizado en el trabajo, desde entornos para programar las diferentes funcionalidades y herramientas para diseñar los bocetos, hasta los lenguajes de programación utilizados tanto en las aplicaciones como en la implementación del sensor ultrasonidos.

4.3.1. Herramientas de diseño

En primer lugar, se exponen las diferentes herramientas que se han utilizado para diseñar tanto los bocetos presentados en el apartado 4.2 en la capa de presentación, como los casos de uso y diagramas de clases diseñados para representar las interacciones del usuario con el sistema y las entidades de ambas aplicaciones.

4.3.1.1. Adobe XD

Adobe XD es un editor de gráficos vectoriales desarrollado y publicado por Adobe Inc. para diseñar y crear prototipos de interfaces de usuario para sitios web y dispositivos móviles. Se puede utilizar en Mac OS y Windows y se utiliza especialmente en las vistas generadas en la sección 4.2.1 de la capa de presentación.

Esta herramienta posee muchas características interesantes, pero a continuación se destacan las más importantes para este trabajo:

· **Animaciones**: permite añadir animaciones e interactuar con los diseños realizados, además de simular la navegación entre los diferentes prototipos creados.

· **Componentes reutilizables**: una vez creado un componente, se puede reutilizar y seguir modificando el diseño continuamente.

· **Kits de interfaz de usuario**: Adobe XD ofrece paquetes con diseños ya hechos de Apple Design, Google Material Design, IBM, Microsoft… Incluso se puede compartir, obtener y modificar vistas enteras de otros usuarios gracias a la comunidad que tiene Adobe.

· ***Plug-ins***: gracias a ellos, se automatizan las tareas más pesadas evitando procesos repetitivos. Además, existen algunos que permiten exportar el diseño creado directamente a código, ahorrando una gran cantidad de tiempo en el desarrollo de las interfaces de usuario.

4.3.1.2. StarUML

StarUML[6] es un modelador de software sofisticado destinado a respaldar el modelado ágil y conciso. Esta herramienta se ha utilizado principalmente para el diseño de los casos de uso y diagramas de clases.

Esta herramienta asegura diseñar diagramas y metamodelos estándar compatibles con UML 2.x: clases, objetos, caso de usos, componentes, diagramas de estado, actividades, flujos de información, diagramas de perfil… Además, soporta el modelado con diagramas SysML: requisitos, definición de bloques, diagramas paramétricos…

Otra de las características más importantes es el desarrollo basado en modelos, los datos de modelado se almacenan en un formato JSON muy simple, por lo que se pueden usar fácilmente para generar código personalizado mediante plantillas definidas por el usuario. Además, permite generar código a partir de los diagramas para varios lenguajes de programación, incluidos Java, C#, C++ y Python a través de extensiones de código abierto.

Cabe destacar que cuenta con clientes muy importantes del mundo tecnológico, como Google, Apple, IBM, Facebook, Amazon, Samsung… Por lo que la calidad en los diseños que se crean está asegurada.

4.3.2. Entornos de desarrollo

En este subapartado se describen los diferentes entornos de programación que se utilizan en el trabajo, tanto para desarrollar las aplicaciones como para implementar el código del sensor ultrasónico y la placa de desarrollo.

[6] https://staruml.io/

4.3.2.1. Visual Studio Code

Para la implementación en ambas aplicaciones de las tres capas definidas en el diseño se ha utilizado el IDE de Visual Studio Code[7].

Con IntelliSense, además del autocompletado y resaltado de la sintaxis, se proporcionan terminaciones inteligentes basadas en tipos de variables, definiciones de funciones y módulos importados.

También viene con comandos de Git integrados, lo que facilita la revisión de diferencias con anteriores cambios, realizar confirmaciones directamente desde el editor, consultar el historial de versiones que tiene el repositorio…

Pero, lo más importante, son las extensiones que permite instalar este entorno. A continuación, se nombran las utilizadas en este trabajo:

- **Awesome Flutter Snippets**: es una colección de clases y métodos de Flutter de uso común. Aumenta la velocidad de desarrollo al eliminar la mayor parte del código estándar asociado a la creación de un widget.

- **Dart**: agrega soporte para el lenguaje de programación Dart. Proporciona herramientas para editar, refactorizar, ejecutar y recargar de manera efectiva las aplicaciones móviles Flutter.

- **Flutter**: agrega soporte para editar, refactorizar, ejecutar y recargar de manera efectiva las aplicaciones móviles Flutter, así como soporte para el lenguaje de programación Dart.

- **GitHub**: integra la mayoría de las funcionalidades de GitHub. Permite, entre otras muchas cosas, conectar un repositorio con el entorno de desarrollo, subir cambios, crear ramas, sincronizar directorios… Es decir, se puede gestionar el repositorio de GitHub al completo.

- **Pubspec Assist**: permite agregar fácilmente dependencias al archivo pubspec.yaml (donde se encuentran las librerías instaladas de Flutter) del proyecto Dart y Flutter.

[7] Editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS, https://code.visualstudio.com/

4.3.2.2. Arduino IDE

El entorno de desarrollo integrado de Arduino[8] es una aplicación multiplataforma que está escrita en el lenguaje de programación Java. Se utiliza para crear y cargar programas en placas compatibles con Arduino, pero también se puede utilizar en placas de desarrollo de terceros.

Este IDE utiliza reglas especiales de estructura de código para admitir los lenguajes C y C++. También ofrece una biblioteca de software para proyectos de cableado que proporciona muchos procesos de E/S comunes.

Para tener código funcional se requieren dos reglas básicas:

1. Definir una función *setUp()* donde se preparan las variables, los pines del sensor que se utilizan y las credenciales del servidor de base de datos de Firebase donde se almacenan los datos recogidos por el mismo sensor.
2. Definir otra función *loop()* que es la encargada de ejecutarse cada periodo de tiempo, recoger los datos que emite el sensor ultrasonidos y almacenarlos en la base de datos.

Estas dos funciones se compilan y vinculan con un apéndice de programa *main()* en un ciclo con el GNU *toolchain*, que también se incluye.

Por último, las bibliotecas que se han utilizado para este trabajo son las siguientes:

· **WiFi**: permite que una placa de desarrollo se conecte a Internet. Puede utilizarse como servidor que acepta conexiones entrantes o como cliente que realiza conexiones salientes. La biblioteca admite el cifrado WEP y WPA2.

· **Firebase ESP8266 Client**: esta biblioteca ofrece las operaciones más confiables para leer, almacenar, actualizar, eliminar y restaurar los datos de la base de datos Firebase Realtime.

4.3.3. Lenguajes de programación

Para desarrollar este trabajo, se ha utilizado el *framework* de Flutter junto con el lenguaje de programación Dart que lo respalda, y C++ para programar la funcionalidad del sensor

[8] https://www.arduino.cc/en/guide/windows

ultrasonidos y la placa de desarrollo NodeMCU. En los siguientes apartados se especifican con más detalle estos lenguajes.

4.3.3.1. Flutter y Dart

Flutter es un kit de desarrollo software de código abierto creado por Google que se utiliza para crear aplicaciones multiplataforma para Android, iOS, Linux, Mac, Windows y web a partir de una única base de código.

Las aplicaciones de Flutter están escritas en el lenguaje Dart y utilizan muchas de las funciones más avanzadas del lenguaje. Además, Flutter se ejecuta en una máquina virtual Dart, que cuenta con un motor de ejecución *Just In Time*, esto quiere decir que permite inyectar modificaciones en el código de una aplicación y apreciar esos cambios en tiempo de ejecución.

El motor de Flutter, escrito principalmente en C++, proporciona soporte de renderizado de bajo nivel utilizando la biblioteca de gráficos Skia[9] de Google. Además, interactúa con SDK específicos de la plataforma, como los proporcionados por Android y iOS. Este motor implementa las bibliotecas centrales de Flutter, que incluyen animación y gráficos, E/S de archivos y redes, soporte de accesibilidad, arquitectura de complementos y una serie de herramientas de compilación de Dart.

Por otra parte, uno de los aspectos más interesantes de este *framework* son los widgets. Flutter utiliza una variedad de widgets, que son la arquitectura del marco de Flutter, para ofrecer una aplicación completamente funcional. Este marco contiene dos conjuntos de widgets que se ajustan a lenguajes de diseño específicos: los widgets de Material Design implementan el diseño de Google y los widgets de Cupertino implementan las pautas de la interfaz humana iOS de Apple.

En cuanto al lenguaje de programación Dart, es de código abierto y estructurado, también desarrollado por Google. Dart está pensado para resolver algunos problemas de JavaScript, ya que pretende ser una herramienta sencilla para proyectos más grandes y ofrecer una mejor seguridad.

[9] Biblioteca de gráficos 2D de código abierto que proporciona API comunes que funcionan en una variedad de plataformas de hardware y software.

Por último, Dart-sdk es el kit de desarrollo de software de Dart. Incluye todas las bibliotecas de este lenguaje, como *dart:core* y *dart:hmtl*, así como algunas herramientas de línea de comandos muy útiles como el compilador Dart-JavaScript y las máquinas virtuales Dart.

4.3.3.2. C++

Este lenguaje se ha utilizado principalmente para programar el código del sensor en la placa de desarrollo NodeMCU. Es uno de los más veteranos, diseñado en 1979. Desde la perspectiva de los lenguajes orientados a objetos, C++ es un leguaje híbrido. Esto se debe a que tiene como objetivo ampliar el mecanismo mediante el cual C puede manipular objetos.

Desde entonces, se han agregado características de programación comunes a la programación estructurada y los paradigmas de programación orientada a objetos. Por lo tanto, C++ es conocido también como un lenguaje de programación de múltiples paradigmas.

Las características más importantes de C++ son las siguientes:

· Permite la agrupación de instrucciones.

· Lenguaje muy didáctico, con este lenguaje se puede aprender mucho de otros lenguajes con gran facilidad.

· Es portátil y tiene un gran número de compiladores en diferentes plataformas y sistemas operativos.

· Permite la separación de un programa en módulos que admiten compilación independiente.

· Es un lenguaje de alto nivel.

4.3.4. API de Google Maps

Gracias a las URL de Maps [12], se puede crear una URL universal multiplataforma para iniciar Google Maps y realizar búsquedas, obtener direcciones y rutas navegables. La sintaxis de la URL es la misma independientemente de la plataforma de uso.

Concretamente, en ambas aplicaciones se utiliza la función de direcciones que ofrece Google Maps, con esta se puede solicitar direcciones y trazar una ruta con los parámetros adecuados. El aspecto básico de la URL es el siguiente:

- `https://www.google.com/maps/dir/?api=1&parameters`

La API de direcciones muestra la ruta entre dos o más puntos especificados en el mapa, así como la distancia y el tiempo de viaje. Por consiguiente, los parámetros posibles que se pueden adjuntar a la URL básica mencionada anteriormente son los siguientes:

- **`origin`**: define el punto de partida desde el que se muestran las direcciones. De forma predeterminada, la ubicación de inicio más relevante, como la ubicación del usuario si está disponible. El valor puede ser el nombre de un lugar, la dirección o las coordenadas de latitud/longitud separadas por comas.
- **`destination`**: define el punto final de las direcciones. El valor, igual que en el origen, puede ser el nombre de un lugar, la dirección o las coordenadas de latitud/longitud separadas por comas.
- **`travelmode`**: define el método de viaje. Las opciones son *driving*, *walking*, *bicycling* o *transit*. Si no se especifica ninguno, se muestra uno o más de los modos más relevantes para la ruta especificada y/o las preferencias del usuario.
- **`dir_action=navigate`**: inicia la navegación paso a paso o la vista previa de la ruta al destino especificado, en función de si el origen está disponible. Si el usuario no especifica un origen o el origen está cerca de la ubicación actual del usuario, el mapa inicia la navegación paso a paso.
- **`waypoints`**: especifica uno o más lugares intermedios a través de los cuales enrutar direcciones entre *origin* y *destination*. La cantidad de *waypoints* permitidos va desde los tres en navegadores móviles hasta nueve en caso contrario. Cada punto de ruta puede ser el nombre de un lugar, la dirección o las coordenadas de latitud/longitud separadas por comas.

Una vez claros los parámetros que permiten las URL, esta es la consulta que se realiza cada vez que un usuario genera una ruta hacia un contenedor:

origin	destination	travelmode	dir_action	waypoints
-	container.latitude + container.longitude	walking	navigate	-

Por otro lado, en la aplicación de Tot Net Recogedores, las consultas que se realizan cuando un recogedor genera una ruta hacia un contenedor o solicita generar una ruta con paradas son las siguientes:

Hacia un contenedor:

origin	destination	travelmode	dir_action	waypoints
-	container.latitude + container.longitude	driving	navigate	-

Ruta con paradas:

origin	destination	travelmode	dir_action	waypoints
-	position.latitude + position.longitude (position = ubicación actual, ya que la ruta debe ser circular)	driving	navigate	waypoints (lista de contenedores ordenada de más llenos a menos llenos)

4.3.5. Firebase

Firebase [13] es una plataforma en la nube que se integra con Google Cloud Platform. Utiliza un conjunto de herramientas para crear y sincronizar proyectos de alta calidad. Tiene las siguientes ventajas:

- Sincronizar los datos de los proyectos sin tener que administrar conexiones o escribir lógica de sincronización compleja.

- Se integra fácilmente en aplicaciones móviles.

- Utiliza la infraestructura y el ajuste de escalado automático de Google para todo tipo de aplicaciones.

- Crea proyectos sin necesidad de un servidor.

Esta plataforma cuenta con una serie de servicios muy extensos, por lo que se van a describir los utilizados en este trabajo. Estos servicios se dividen en dos, aquellos que se centran en el análisis de uso de las aplicaciones y los que dan un salto de calidad a las aplicaciones, es decir, los centrados en el desarrollo.

En los servicios de ***analytics***, se encuentra la aplicación gratuita de **Firebase Analytics**, la cual proporciona una visión profunda sobre el uso de la aplicación por parte de los usuarios. Así se puede ver si la aplicación rinde según lo esperado y si los usuarios están o no satisfechos con los cambios que se van realizando.

En cuanto a los servicios de **desarrollo**, una variedad de funciones, especialmente la detección de errores y las pruebas, ayudan a crear mejores aplicaciones y reducir el tiempo de optimización y desarrollo. Lo más destacado es la capacidad de almacenar toda la información en la nube, distribuirla y organizarla.

Dentro de estos servicios de desarrollo, está **Firebase Auth**. Este servicio permite autenticar a los usuarios utilizando únicamente código del lado del cliente. Incluye los proveedores de inicio de sesión de Facebook, GitHub, Twitter, Google, Yahoo! y Microsoft, así como los métodos clásicos de inicio de sesión mediante correo electrónico y contraseña. Con esto, Firebase proporciona una forma simple, eficiente y segura de administrar usuarios, por lo que el desarrollador no tiene que preocuparse por extender los métodos de autenticación existentes.

Por otra parte, también se ha utilizado el servicio de **Realtime Database** para almacenar los datos recogidos por el sensor ultrasonidos. Firebase proporciona un *back-end* en formato JSON en tiempo real, así como una base de datos organizada y una API que permite almacenar información de la aplicación en sincronización con la nube de Firebase. La sincronización en tiempo real de esta base de datos brinda a los usuarios acceso en todo momento a la información de cada uno de los contenedores desde cualquier dispositivo, almacenada en la nube cada vez que los resultados del sensor se envían. Al mismo tiempo, notifica al resto de los dispositivos.

Este trabajo también cuenta con **Firebase Storage**, servicio que permite realizar cargas y descargas seguras de archivos, sin importar la calidad de la red. Concretamente se utiliza en la primera aplicación, Tot Net, para almacenar de manera segura las fotos de perfiles de cada usuario.

Por último, y uno de los servicios más importantes, es el de **Firebase Cloud Firestore**. Al igual que Realtime Database, es una base de datos NoSQL, aunque existen algunas diferencias. Se organiza en forma de documentos agrupados en colecciones, y en ellos se pueden incluir tanto campos de diversos tipos como otras colecciones. En este proyecto existen dos colecciones, la de contenedores y la de usuarios, además esta última tiene otra colección con los contenedores favoritos de cada usuario.

5. Implantación

Para poner en marcha cada una de las aplicaciones, es necesario realizar la instalación del APK de cada una. Para generar ese paquete de aplicaciones de Android, se ha de firmar ambas aplicaciones y realizar la configuración correcta de los archivos, la cual se va a explicar a continuación [11]. Cabe destacar que el proceso es el mismo para las dos aplicaciones, por lo que se explica el proceso de instalación una vez.

En primer lugar, se debe crear un almacén de claves, para ello se introduce el siguiente comando en la consola de una máquina Windows:

```
keytool -genkey -v -keystore c:\Users\Usuario\key.jks -storetype JKS -keyalg RSA -keysize 2048 -validity 10000 -alias key
```

Tras ejecutar este comando, se debe introducir una contraseña tanto para el almacén como para la llave generada.

El siguiente paso es hacer referencia a este almacén de claves desde la aplicación. Por tanto, en el nivel de directorio *[project]/android/key.properties* se crea el archivo *key.properties*. El contenido tiene que ser el siguiente:

```
storePassword=<contraseña del almacén del paso anterior>
keyPassword=<contraseña de la llave del paso anterior>
keyAlias=<alias para la llave>
storeFile=/Users/Usuario/key.jks (directorio donde se encuentra la llave)
```

Posteriormente, hay que configurar el inicio de sesión en *gradle* para usar la llave al crear la aplicación. Por lo que en el archivo *[project]/android/app/build.gradle* se agrega la información del almacén de llaves del archivo de propiedades anterior:

```
def keystoreProperties = new Properties()
    def keystorePropertiesFile = rootProject.file('key.properties')
    if (keystorePropertiesFile.exists()) {
        keystoreProperties.load(new
FileInputStream(keystorePropertiesFile))
    }
```

Se reemplaza también el bloque *buildTypes* de este archivo por el siguiente:

```
signingConfigs {
        release {
            keyAlias keystoreProperties['keyAlias']
            keyPassword keystoreProperties['keyPassword']
            storeFile        keystoreProperties['storeFile']        ?
file(keystoreProperties['storeFile']) : null
            storePassword keystoreProperties['storePassword']
        }
}
buildTypes {
    release {
        signingConfig signingConfigs.release
    }
}
```

También se debe verificar que el manifiesto, ubicado en *[project]/android/app/src/main*, contiene el valor `android:label` que refleja el nombre final de la aplicación y el permiso `android.permission.INTERNET` para darle acceso a internet a la misma:

```
<uses-permission android:name="android.permission.INTERNET"/>
```

Una vez realizados todos los pasos anteriores, ya se puede crear el APK. Cuando se ejecute el comando `flutter build apk` (en el directorio raíz del proyecto), se tiene como resultado un APK que contiene el código de la aplicación compilado para todas las ABI de destino.

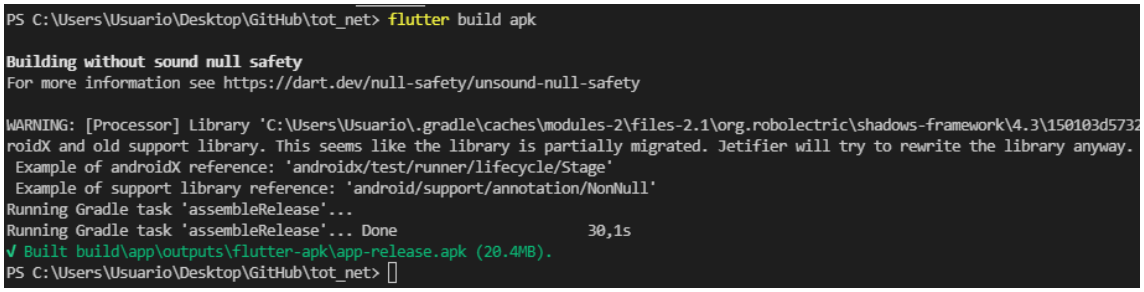

Figura 28. Construcción del APK de la aplicación Tot Net

Finalmente, para instalar este APK, se conecta el dispositivo a la máquina en la que se haya realizado todo este proceso, se abre una consola y se lanza el comando `flutter install`.

```
PS C:\Users\Usuario\Desktop\GitHub\tot_net> flutter install
Installing app.apk to MI 8...
Installing build\app\outputs\flutter-apk\app.apk...                5,2s
PS C:\Users\Usuario\Desktop\GitHub\tot_net>
```

Figura 29. Instalación del APK de la aplicación Tot Net en un dispositivo real

Esta secuencia de firma, construcción e instalación, como se ha comentado anteriormente, es idéntico para instalar la segunda aplicación. Como resultado al realizar este proceso en ambas aplicaciones es la obtención de dos accesos directos en el dispositivo.

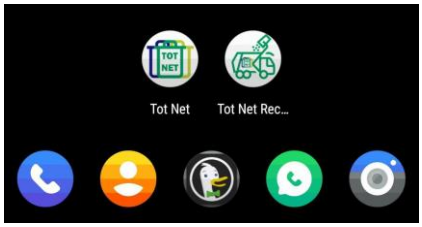

Figura 30. Accesos directos de la aplicación Tot Net y Tot Net Recogedores

Al abrirlas, se puede comprobar que se accede correctamente a la pantalla principal de ambas aplicaciones.

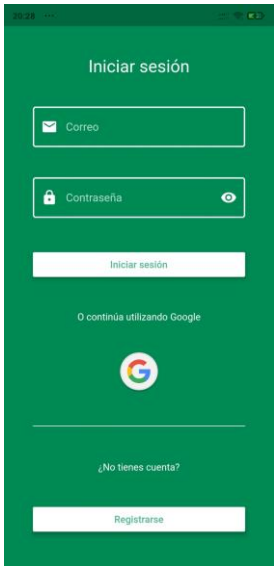

Figura 31. Pantalla principal de la aplicación Tot Net

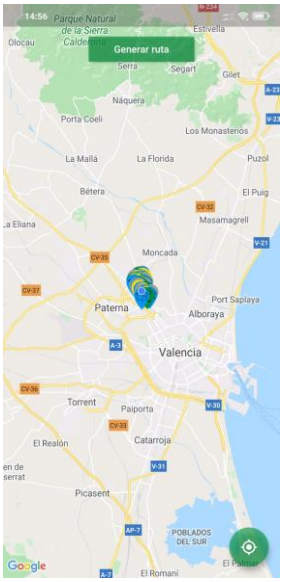

Figura 32. Pantalla principal de la aplicación Tot Net Recogedores

En el siguiente apartado se detallan las pruebas que se han realizado sobre estas dos aplicaciones instaladas en el dispositivo real.

6. Pruebas

La fase de realización de pruebas es indispensable en el desarrollo de software. Gracias a ellas, se puede comprobar que el sistema realiza lo que el usuario espera y, lo más importante, que se cumple con todo lo especificado en la fase de especificación de requisitos y el usuario puede interactuar como se ha indicado en los casos de uso en el análisis del problema.

En primer lugar, se comienzan a realizar las pruebas en la aplicación Tot Net. La primera de ellas es verificar si el registro de una cuenta estándar es correcto. Para ello, al abrir la aplicación, se pulsa el botón de registrarse en la parte inferior central de la pantalla y se rellenan los datos correspondientes, en este caso usuario, correo electrónico y contraseña.

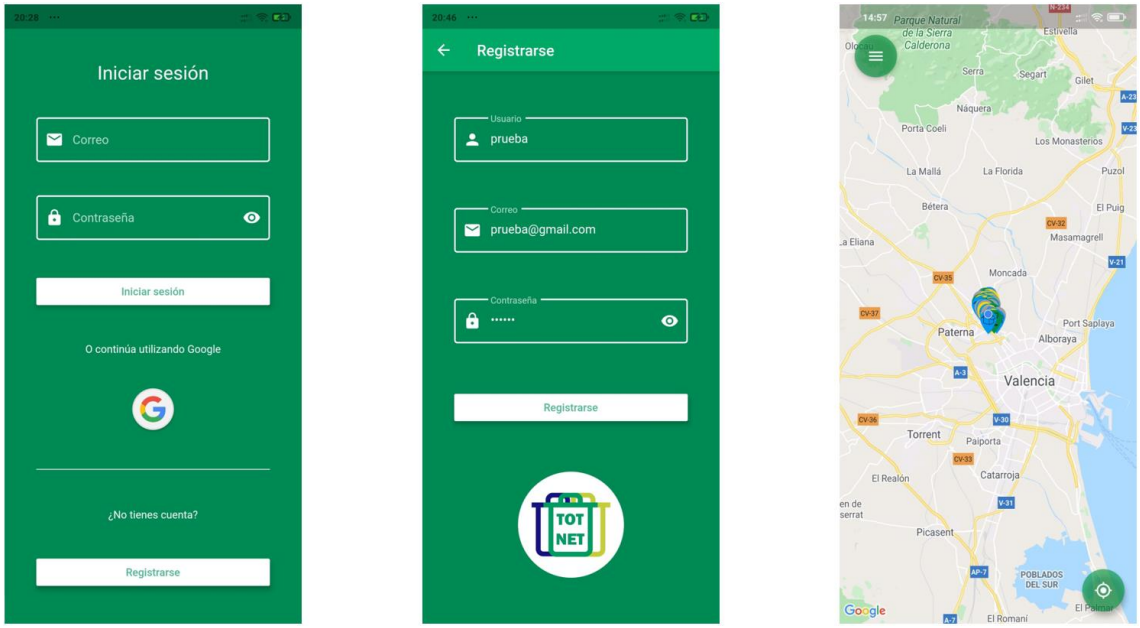

Figura 33. Proceso de registro de cuenta estándar en la aplicación Tot Net

Al introducir los datos correctamente, la aplicación redirige al usuario a la pantalla del mapa en la que se puede ejecutar el resto de las funcionalidades. Además, en el servidor de Firebase, se puede observar en las siguientes imágenes que el usuario se ha registrado correctamente y se ha almacenado en la base de datos con la información precisa.

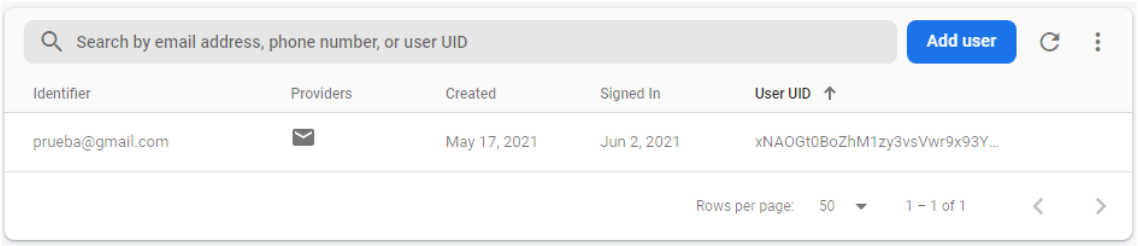

Figura 34. Cuenta estándar de usuario registrada en Firebase

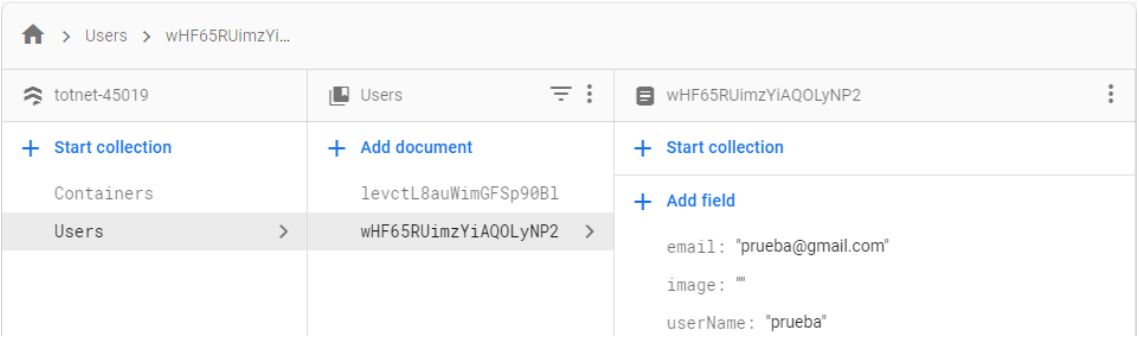

Figura 35. Usuario almacenado en la base de datos de Firebase

Por tanto, se verifica la correcta implementación del requisito UR-R01.

Para comprobar que el registro de un nuevo usuario a partir de una cuenta Google existente es correcto, en lugar de pulsar el botón de registrarse en la pantalla principal, se pulsa el icono de Google. Seguidamente aparece el proveedor de Google para elegir una cuenta. Se elige una y la aplicación redirige al usuario, como antes, a la pantalla del mapa.

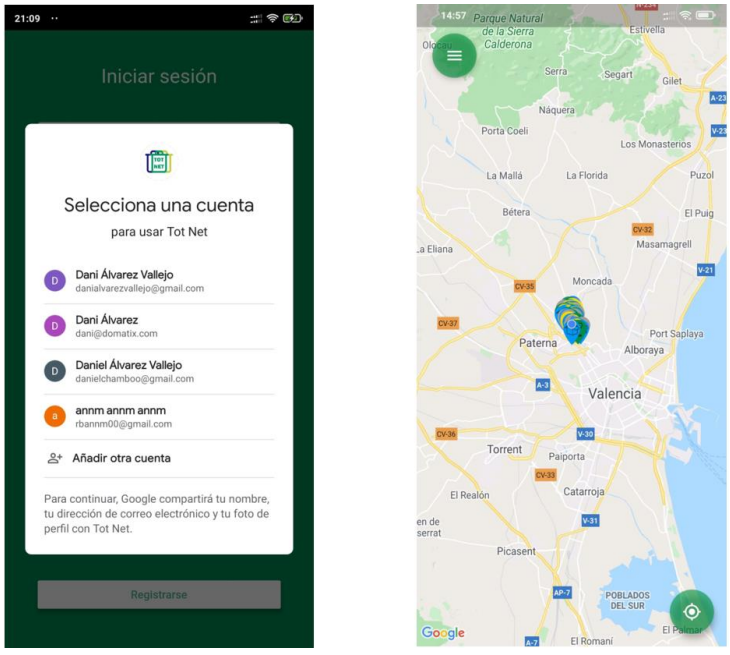

Figura 36. Proceso de registro con cuenta de Google existente en la aplicación Tot Net

E igual que en el caso anterior, en el servidor de Firebase se observa que la cuenta y el usuario se han creado correctamente.

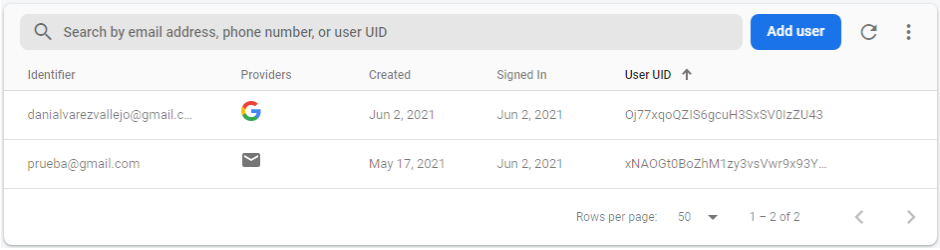

Figura 37. Cuenta de usuario de Google registrada en Firebase

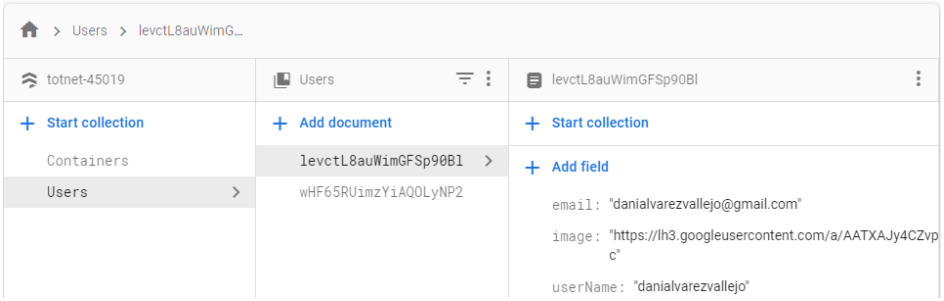

Figura 38. Usuario de Google almacenado en la base de datos de Firebase

También se puede verificar que el requisito UR-R02 ha sido satisfecho.

Una vez creados ambos tipos de cuentas, se prueba el funcionamiento de inicio de sesión. Por una parte, para iniciar sesión con la cuenta estándar, simplemente se rellenan los campos de correo y contraseña de la pantalla principal como se muestra en la siguiente figura. Tras ello, la aplicación redirige al usuario a la pantalla del mapa.

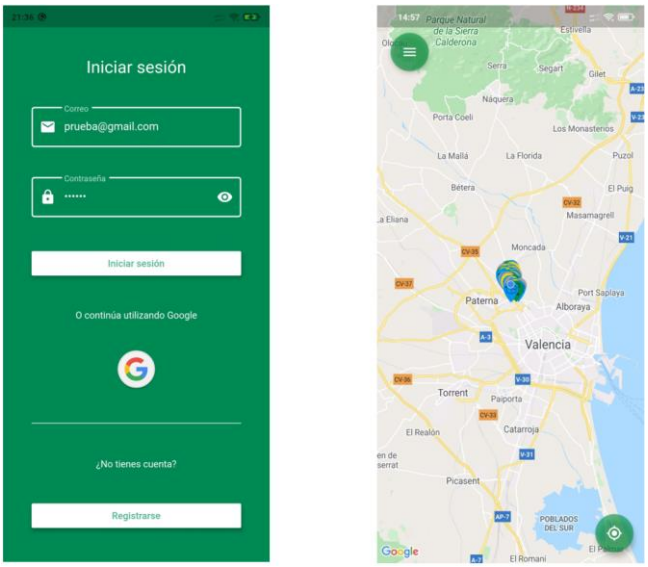

Figura 39. Proceso de inicio de sesión con cuenta estándar en la aplicación Tot Net

Por otra parte, para iniciar sesión con una cuenta de Google, hay que pulsar el botón con el símbolo de Google como se ha hecho anteriormente. El sistema inicia sesión con la cuenta de Google que se ha elegido en el registro y la aplicación lleva al usuario a la pantalla del mapa. Por tanto, se verifican los requisitos UR-R03 y UR-R04.

Después de iniciar sesión y cargar el mapa, se puede navegar por él, acercar y alejar la vista y además se muestra la ubicación actual del usuario, la cual se puede centrar pulsando el botón de la parte inferior derecha de la pantalla. De esta manera, se cumplen los requisitos MR-R01, MR-R02, MR-R03 y MR-R04.

Si se acerca la vista sobre la zona de Burjasot, como se aprecia en la siguiente figura, se observa que se pueden distinguir los diferentes tipos de contenedores por su color y si se pulsa sobre alguno, se puede consultar la información más relevante del mismo, validando los requisitos MR-R05 y MR-R06.

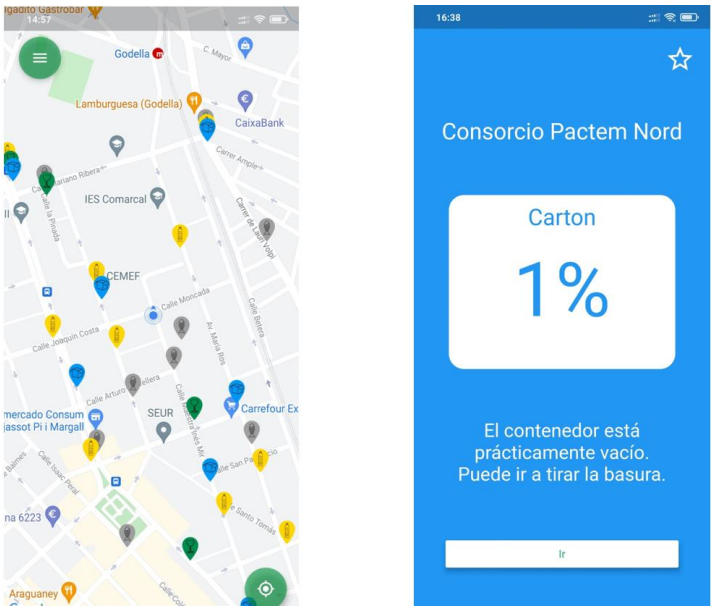

Figura 40. Pantalla del mapa y del contenedor de la aplicación Tot Net

En la misma pantalla del contenedor se visualiza la calle en la que se encuentra, el tipo de residuo que almacena, la cantidad que tiene en ese momento o si es o no recomendable ir hasta él, satisfaciendo los requisitos CR-R02, CR-R03, CR-R04 y CR-R05. Si el contenedor tiene pocos residuos, se encuentra disponible el botón que permite generar una ruta hacia él y seguir el trayecto en Google Maps. Se puede cancelar la ruta en cualquier momento y volver a la aplicación. Así, estarían correctamente implementados los requisitos CR-R06, RR-R01, RR-R02 y RR-R03.

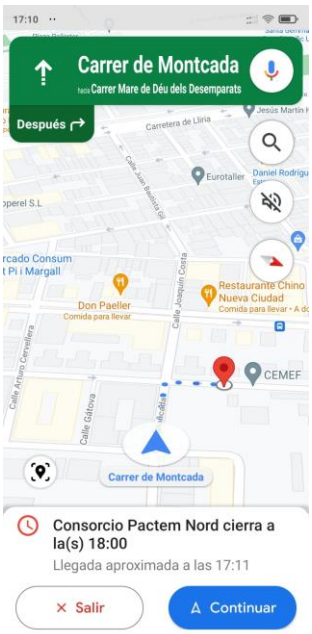

Figura 41. Ruta hacia un contenedor en Google Maps desde la aplicación Tot Net

Volviendo a la pantalla del mapa, se puede acceder al menú donde se encuentran el resto de las funcionalidades presionando el botón de la parte superior izquierda.

En primer lugar, el usuario puede acceder a su perfil y modificar tanto el nombre de usuario como la foto de perfil y contraseña. Para cambiar la foto de perfil, se presiona el avatar y se otorgan permisos de almacenamiento a la aplicación. Posteriormente se elige una imagen y la aplicación redirige al usuario al perfil con la nueva foto cargada.

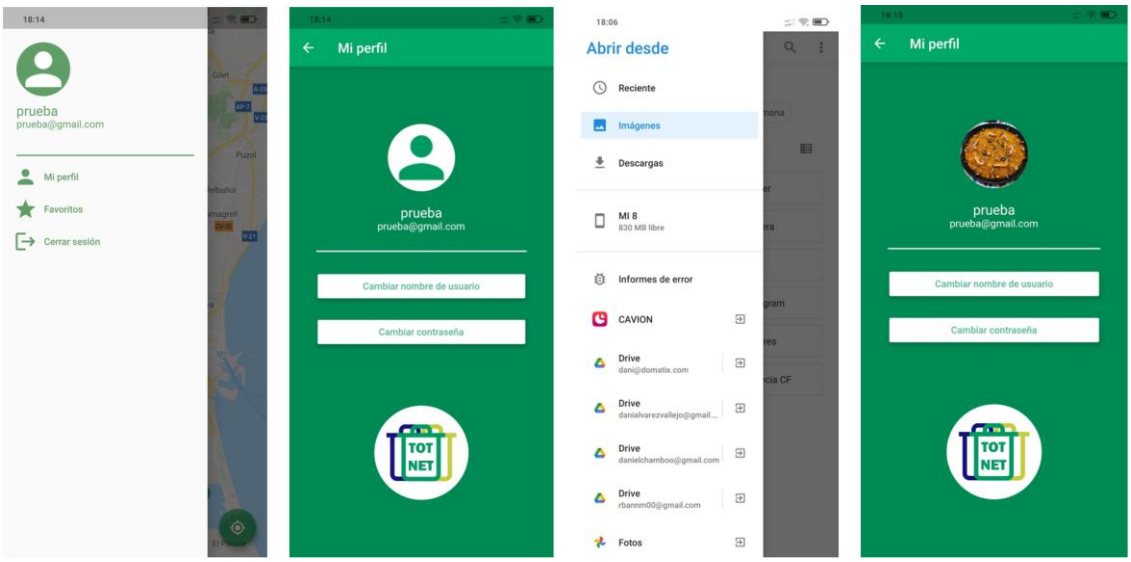

Figura 42. Proceso de cambio de foto de perfil en la aplicación Tot Net

Para modificar el nombre de usuario, se presiona el primer botón y se rellena el campo que se pide.

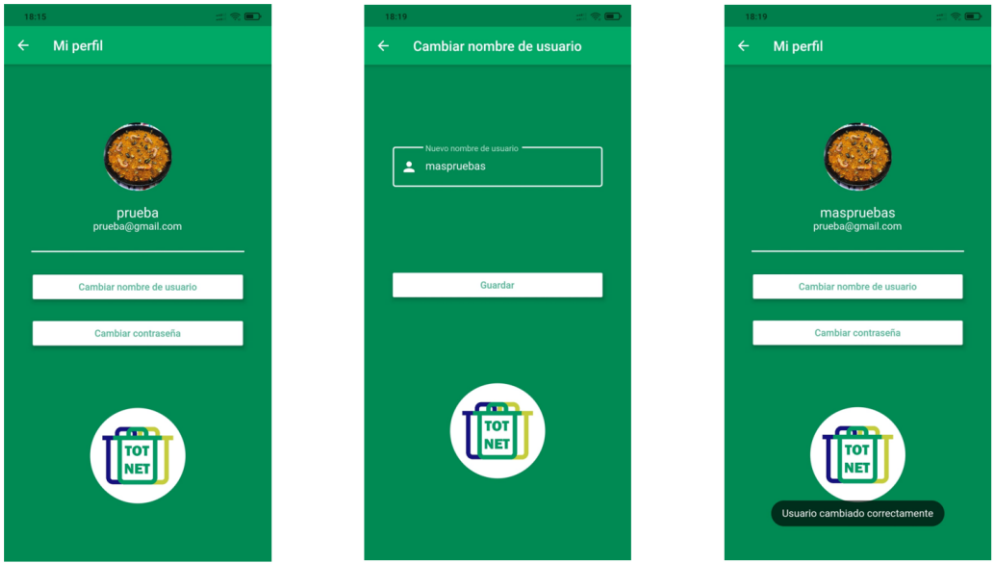

Figura 43. Proceso de cambio de nombre de usuario en la aplicación Tot Net

Y de igual manera si se quiere realizar el cambio de contraseña. Para ello, se presiona el segundo botón y se rellenan los dos campos que se piden, la contraseña actual y la nueva que se desea tener. Pero, si se accede con una cuenta de Google, no se puede cambiar la contraseña ya que se debería hacer desde Google. Esto último se puede observar en la última pantalla de la siguiente imagen.

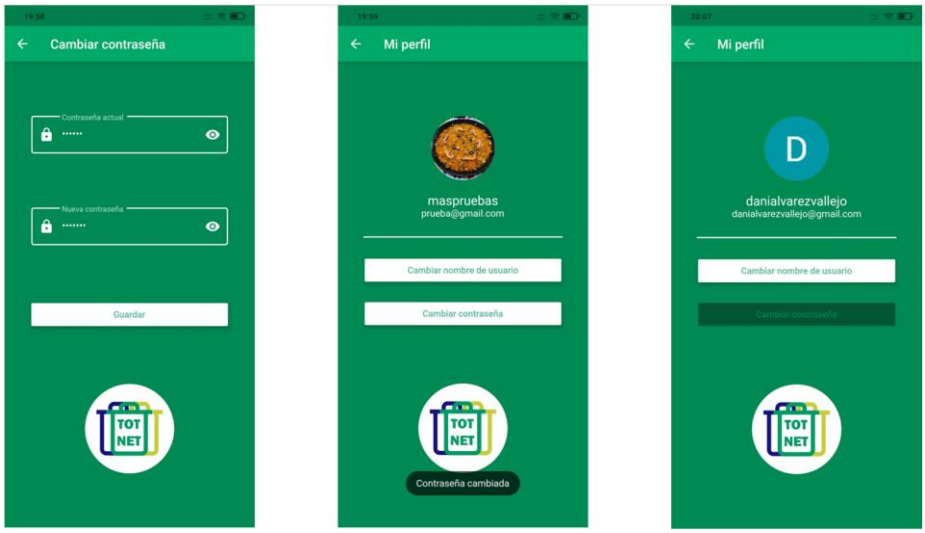

Figura 44. Cambio de contraseña cuenta estándar/cuenta Google en la aplicación Tot Net

Tras la realización de estas pruebas, se verifica el cumplimiento de los requisitos UR-R05, UR-R06 y UR-R07.

De vuelta al menú de la pantalla del mapa, se observa que existe un apartado de favoritos, donde se encuentran aquellos contenedores que se han marcado como tal. Para ello, se procede a marcar tres contenedores cualesquiera y se accede a los favoritos.

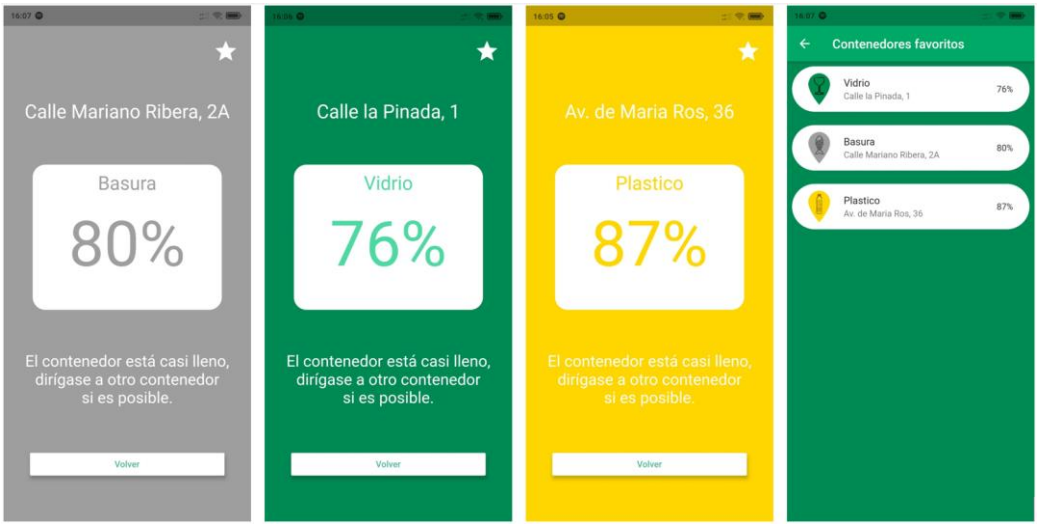

Figura 45. Marcar y consultar contenedores favoritos en la aplicación Tot Net

Como se observa, los contenedores marcados se han añadido al listado de favoritos. Además, se puede consultar la información más relevante de cada uno de ellos desde el mismo listado y también, al presionar sobre ellos, se accede a la pantalla del propio contenedor.

Por otra parte, si se desmarca como favorito uno de los contenedores anteriores, lista se actualiza correctamente en la siguiente imagen.

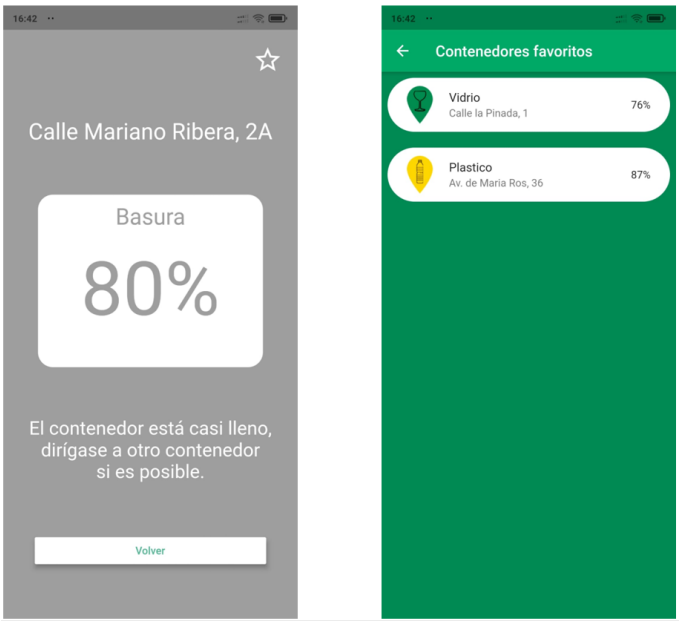

Figura 46. Desmarcar y consultar contenedores favoritos en la aplicación Tot Net

Por lo tanto, se confirma que la implementación de los requisitos UR-R08 y CR-R01 es correcta.

Si se vuelve a la pantalla principal del mapa y se cierra la aplicación, el sistema guarda la sesión actual de manera definida en el tiempo. Al abrir de vuelta la aplicación, se retoma la sesión anterior, verificando la correcta implementación del requisito UR-R09.

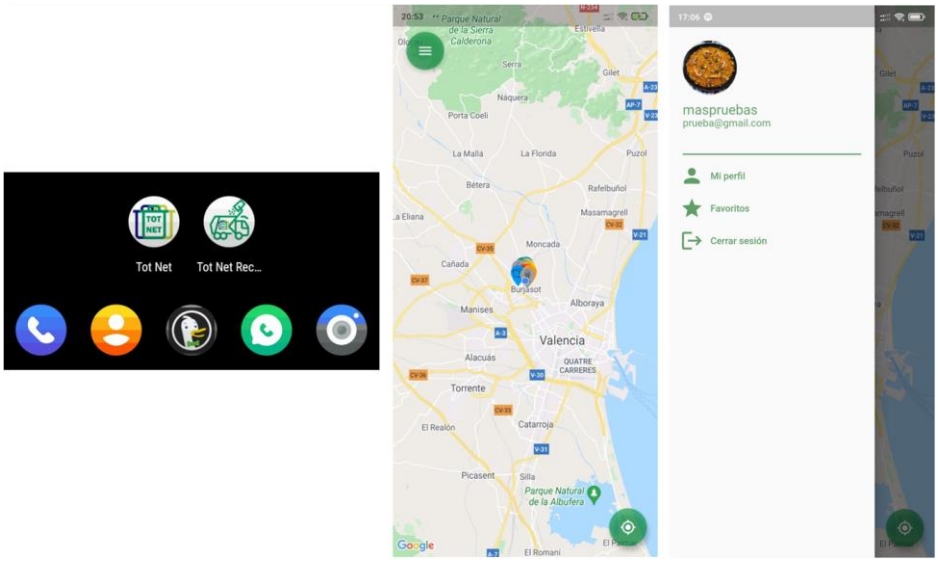

Figura 47. Inicio de sesión automático en la aplicación Tot Net

Por último, en el menú está la posibilidad de cerrar la sesión actual. Si se pulsa esa opción, el sistema envía al usuario a la pantalla principal de la aplicación donde se puede registrar una cuenta o iniciar sesión con otra, como se muestra a continuación.

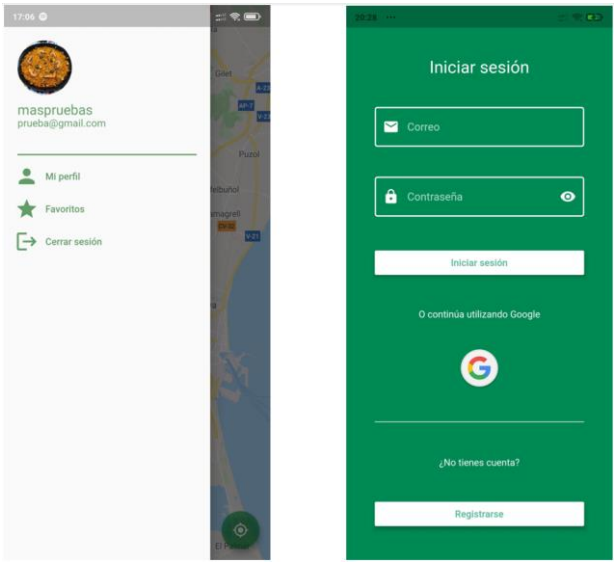

Figura 48. Proceso de cierre de sesión en la aplicación Tot Net

A continuación, se exponen las pruebas realizadas en la aplicación Tot Net Recogedores.

Después de abrir la aplicación y cargar el mapa, al igual que en la primera aplicación, se puede navegar por él, acercar y alejar la vista y se muestra la ubicación actual, la cual se puede centrar pulsando el botón de la parte inferior derecha de la pantalla. Por tanto, se cumplen los requisitos MR-R01, MR-R02, MR-R03 y MR-R04. Para los requisitos MR-R05 y MR-R06 se han realizado las mismas pruebas mencionadas anteriormente, si se acerca la vista sobre la zona de Burjasot, como se aprecia en la primera imagen de la siguiente figura, se observa que se pueden distinguir los diferentes tipos de contenedores según el color de estos y, si se pulsa sobre alguno, está disponible la información cada uno de ellos.

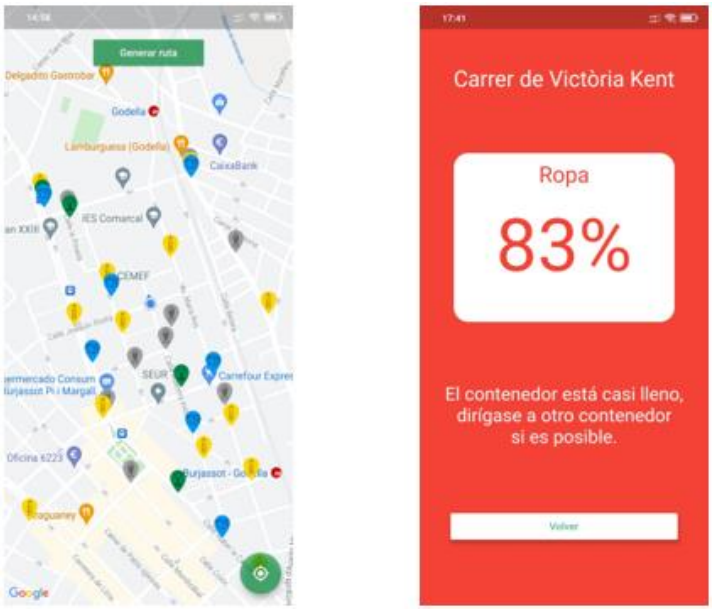

Figura 49. Pantalla del mapa y del contenedor en la aplicación Tot Net Recogedores

En la misma pantalla del contenedor se encuentra la misma información que en Tot Net, se puede consultar la calle en la que se encuentra, el tipo de residuo que almacena, la cantidad que tiene en ese momento o si es o no recomendable ir hasta él, satisfaciendo los requisitos CR-R01, CR-R02, CR-R03 y CR-R04. Si el contenedor tiene pocos residuos, está disponible el botón que permite generar una ruta hacia él y seguir el trayecto en Google Maps. Se puede cancelar la ruta en cualquier momento y volver a la aplicación. Así, estarían correctamente implementados los requisitos CR-R05, RR-R01, RR-R02 y RR-R03.

Figura 50. Ruta a un contenedor en Google Maps desde Tot Net Recogedores

Por último, se ha probado la generación de una ruta por los contenedores existentes con mayor porcentaje de residuos. Para ello, en la pantalla del mapa, se presiona el botón “Generar ruta” y se espera a que Google Maps reciba la petición y cree la ruta esperada.

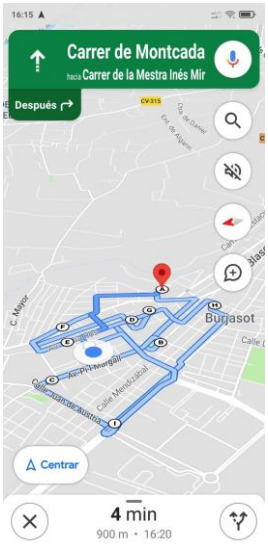

Figura 51. Ruta con paradas en Google Maps desde Tot Net Recogedores

Como se puede observar, la ruta cuenta con una serie de paradas ordenadas de forma alfabética. Esas letras hacen referencia a cada uno de los contenedores con mayor cantidad de residuos.

7. Conclusiones

Las aplicaciones móviles han estado presentes en los dispositivos personales desde hace aproximadamente más de diez años, y lo seguirán estando durante muchos años más, ya que es la industria que demuestra un crecimiento en el mercado cada vez más importante año tras año.

Gracias a ellas, todo aquel que posea un dispositivo móvil en sus manos en la actualidad, es capaz de hacer muchas cosas que antes era imposible de imaginar, teniendo acceso a innumerables bienes y servicios proporcionados por proveedores expertos en el sector. Por ello, se puede asegurar que todo este mundo de las aplicaciones móviles es una de las revoluciones tecnológicas más grandes de todo el tiempo.

En este proyecto, se ofrece a la actual sociedad una solución para poder cuidar las calles haciendo uso de un dispositivo móvil, consultando la cantidad de residuos que tienen los contenedores de alrededor y evitando así acumular basura y llenar las aceras de residuos indeseados.

Partiendo de los objetivos descritos al principio del proyecto, se puede concluir que se han alcanzado todos y cada uno de ellos. A continuación, se resume en qué consistían:

- ✓ Desarrollar dos aplicaciones, una para ciudadanos y otra para recogedores, que permita visualizar y navegar por un mapa, además de ofrecer la ubicación de los diferentes tipos de contenedores, consultar la información de cada uno y generar trayectos hacia ellos.
- ✓ Para la aplicación de los ciudadanos, en especial, permitir el registro mediante cuentas estándar o cuentas de Google existentes, consultar o modificar el perfil de la cuenta como se desee y cerrar la sesión que esté activa. Además de poder gestionar y visitar aquellos contenedores marcados como favoritos.
- ✓ Para la aplicación de los recogedores, permitir generar una ruta con paradas por los contenedores con mayor porcentaje de residuos, mejorando la eficiencia de recogida y la limpieza de las calles.

Por otra parte, aunque en la carrera no se haya utilizado Flutter ni Dart, el autor de este trabajo ya tenía una experiencia personal previa suficiente para poder realizar este trabajo al completo. Por lo que no ha sido necesario aprender nada desde cero durante el desarrollo del proyecto.

En el caso de la implantación del sensor ultrasonidos, sí que ha hecho falta aprender e investigar el funcionamiento de la placa de desarrollo, saber cuáles son los pines funcionales y comprender cómo desarrollar el fragmento de código C necesario para poner el sensor en marcha. Gracias a la gran cantidad de tutoriales que existen en internet, tampoco ha sido obligatorio invertir tiempo en exceso en el aprendizaje de este tipo de tecnología.

Gracias a este trabajo, se ha conseguido diseñar e implementar un producto software desde cero. Empezando por crear una idea adaptada al contexto social actual, siguiendo por el análisis y especificación de los requisitos del sistema que el usuario desea y acabando por el desarrollo e implantación del producto final.

7.1. Relación del trabajo desarrollado con los estudios cursados

Durante el proceso de desarrollo de este trabajo, se han puesto en práctica muchos conceptos aprendidos a lo largo de la carrera. Entre todas las asignaturas impartidas, las que más influencia han tenido en este proyecto han sido AER (Análisis de Especificación de Requisitos) y PIN (Proyecto de Ingeniería del software).

En el caso de la asignatura de AER, se aprendieron los siguientes aspectos fundamentales para la consecución de este trabajo:

- Entender las necesidades de los usuarios y los clientes.
- Comprender el contexto en el que se utiliza el sistema software.
- Identificar, analizar, negociar y documentar los requisitos de los usuarios y del sistema.
- Validar que los requisitos documentados se corresponden con los requisitos negociados.
- Gestionar la evolución de los requisitos.
- Utilizar herramientas comerciales para gestionar los requisitos.

En cuanto a PIN, gracias a ella se ha conseguido:

- Aplicar técnicas para la evaluación de una técnica de negocio.

· Aplicar una metodología ágil para el desarrollo de un producto.

· Gestionar un proyecto de desarrollo de software prestando atención a su planificación y seguimiento.

En cuanto a las competencias transversales adquiridas durante este grado y que se han aplicado en este proyecto, se destaca la capacidad de **aplicar a la práctica los recursos personales** necesarios para alcanzar los objetivos que se desean, estableciendo un proceso bien definido y argumentando las soluciones y/o acciones adoptadas. También ha sido importante la capacidad de **transformar un problema complejo en uno más sencillo y argumentar ese proceso**, empleando y respaldando un método eficiente de **resolución de problemas**. Por último, y destacando esta competencia sobre las demás, se ha sido capaz de **identificar amenazas y oportunidades** de mejora para proponer ideas y planteamientos originales que aporten valor económico, social y medioambiental, utilizando adecuadamente estrategias y / o técnicas creativas.

8. Trabajos futuros

Este trabajo, al haber sido realizado de manera individual y si ningún tipo de colaboración con ningún proveedor ni desarrollador, ha sido construido a pequeña escala. Es decir, solamente se han ubicado marcadores de contenedores en el territorio de Burjasot y únicamente se cuenta con un sensor ultrasonidos para un contenedor. Por ello, como siguiente movimiento, sería ideal buscar una empresa que estuviera de acuerdo en financiar la continuación del proyecto, para así colocar más sensores en el resto de los contenedores y proveerlos de señal de internet.

Para conocer el tipo y la ubicación de los contenedores, el autor de este trabajo se puso en contacto con el ayuntamiento de Burjasot para preguntar sobre cómo podrían facilitar dicha información. Desgraciadamente, el contrato urbanístico que recoge todos estos datos estaba y sigue estando en proceso de redacción, por lo que esa información no es pública. También se preguntó por el pasado contrato, pero desconocían dónde se encontraba. Por tanto, se decidió obtener esa información realizando rutas y paseos por las diferentes zonas del pueblo. Entonces, como mejora importante a realizar en el futuro, se debe preguntar por el nuevo contrato y recoger la información precisa de los contenedores para plasmarla en nuestra aplicación.

Centrándose en la ubicación de los contenedores, también se ha pensado en agrupar los marcadores que estén muy cercanos entre sí, para evitar saturar el mapa una vez se cuente con más localizaciones. Por tanto, poco a poco se irían mostrando individualmente cada uno de los contenedores que componen el grupo conforme se acerca la visión del mapa. Otra posibilidad que se ha contemplado para evitar esta saturación es la creación de un filtro en la pantalla del mapa, en el que el usuario puede marcar qué tipo de contenedores desea que se visualicen.

Como se ha dicho al principio, sólo se cuenta con un sensor funcional. Este sensor ha sido probado en pequeños recipientes y en basuras medianas / grandes, pero no en un contenedor real, por lo que esto sería un gran trabajo para realizar y realmente necesario para el futuro. Cabe destacar que el sensor elegido está preparado para trabajar en dichos contenedores, ya que como se ha dicho en la descripción de esta tecnología, tiene un rango de medición que alcanza los cuatro metros y medio.

En cuanto a la primera aplicación, los usuarios únicamente pueden gestionar los contenedores que tienen ellos marcados como favoritos. En ese aspecto, se ha contemplado la posibilidad de poder compartir esos contenedores con otros usuarios, para conocer la ubicación de estos o si se necesita saber dónde se encuentra un contenedor especial que no está por la

zona, por ejemplo, uno de aceite, de ropa o de pilas. Debido a la complejidad del desarrollo y a la gran inversión de tiempo que requiere, se ha propuesto como mejora para el futuro.

Por último, como gran futura mejora, se ha pensado en desarrollar un motor propio de generación de rutas para no depender de un proveedor externo y aplicar las mejoras que se deseen de manera directa. Así, por ejemplo, los recogedores no tendrían un límite de nueve paradas cuando generan una ruta entre distintos contenedores.

9. Referencias

[1] A. PILAR, Samuel. *Desbordados por la basura* [en línea]. 2020 [Consulta: 01-02-2021]. Disponible en: https://www.rtve.es/noticias/20200222/desbordados-basura/2003029.shtml

[2] DISCA – ETS de Ingeniería Informática (UPV). *Tema 3: Ingeniería inversa de aplicaciones Android* [en línea]. Valencia: 43 p [Consulta: 14-02-2021]. Disponible en: https://poliformat.upv.es/access/content/group/GRA_14100_2020/Teor%C3%ADa/Tema%203. parte1.pdf

[3] Universidad Politécnica de Valencia. *Arquitectura de Android* [en línea]. Valencia: 2017 [Consulta: 17-02-2021]. Disponible en: http://www.androidcurso.com/index.php/99

[4] DEL VALLE HERNÁNDEZ, Luís. *ESP8266 todo lo que necesitas saber del módulo WiFi para Arduino* [en línea]. 2016. Capítulo 93 [Consulta: 22-02-2021]. Disponible en: https://programarfacil.com/podcast/esp8266-wifi-coste-arduino/

[5] Solectro. *NodeMcu WiFi ESP8266 ESP-12E CP2102 Placa de Desarrollo Inalámbrica* [en línea]. [Consulta: 01-03-2021]. Disponible en: https://solectroshop.com/es/modulos-wifi/82-esp8266-esp12e-placa-de-desarrollo-inalambrico-nodemcu-lua-cp2102-wifi.html#description

[5] Solectro. *HC-SR05 Sensor de Distancia de Ultrasonido* [en línea]. [Consulta: 01-03-2021]. Disponible en: https://solectroshop.com/es/sensores-de-distancia-proximidad-ir-y-ultrasonidos/1196-hc-sr05-sensor-de-distancia-de-ultrasonido.html#description

[6] LLAMAS, Luís. *NodeMCU, la popular placa de desarrollo con ESP8266* [en línea]. 2018 [Consulta: 10-03-2021]. Disponible en: https://www.luisllamas.es/esp8266-nodemcu/

[7] GANDHI, Mayur. *Qué es un sensor ultrasónico y para qué sirve* [en línea]. [Consulta: 23-03-2021]. Disponible en: https://www.autycom.com/que-es-un-sensor-ultrasonico-y-para-que-sirve/

[8] Keyence. *¿Qué es un sensor ultrasónico?* [en línea]. [Consulta: 03-04-2021]. Disponible en: https://www.keyence.com.mx/ss/products/sensor/sensorbasics/ultrasonic/info/

[9] WONG, Jessie. *Cómo funcionan los sensores ultrasónicos* [en línea]. 2017 [Consulta: 15-04-2021]. Disponible en: https://www.bjultrasonic.com/es/how-do-ultrasonic-sensors-work/

[10] RJ Code Advance. *Arquitectura en Capas – Análisis completo + Tradicional vs Modernas, DDD, DIP (Cap 5)* [en línea]. 2019. Capítulo 5 [Consulta: 03-05-2021]. Disponible en: https://rjcodeadvance.com/patrones-de-software-arquitectura-en-capas-analisis-completo-ejemplo-ddd-parte-5/

[11] Flutter. *Build and release an Android app* [en línea]. [Consulta: 26-05-2021]. Disponible en: https://flutter.dev/docs/deployment/android#enabling-material-components

[12] Google. *Maps URLs* [en línea]. 2021 [Consulta: 05-06-2021]. Disponible en: https://developers.google.com/maps/documentation/urls/get-started#directions-action

[13] LÓPEZ, Sara. *Firebase: qué es, para qué sirve, funcionalidades y ventajas* [en línea]. Madrid: 2020 [Consulta: 08-06-2021]. Disponible en: https://www.digital55.com/desarrollo-tecnologia/que-es-firebase-funcionalidades-ventajas-conclusiones/

10. Glosario

· **AOSP**: acrónimo de Android Open Source Project o "Proyecto de código abierto de Android" en español. Es el código fuente del sistema operativo Android.

· **VGA**: acrónimo de Video Graphics Array o "Matriz de gráficos de vídeo" en español. Es un tipo de puerto estándar para dispositivos de video como monitores, proyectores y televisores, desarrollado por IBM e introducido en 1987.

· **GSM/EDGE**: EDGE son las siglas de Enhanced Data Rates for GSM Evolution, también conocida como Enhanced GPRS (EGPRS) o "GPRS Mejorado". Se trata de una tecnología de telefonía móvil que actúa como puente entre las redes 2G y 3G.

· **UMTS**: abreviatura de Universal Mobile Telecommunications System o "Sistema Universal de Telecomunicaciones Móviles" en español. La tecnología GPRS (GSM evolutiva) en sí misma no se pudo desarrollar para proporcionar servicios de tercera generación, por lo que UMTS, sucesora de GPRS, se desarrolló con el fin de arreglar este problema y ofrecer finalmente estos servicios de tercera generación a los dispositivos móviles.

· **LTE**: acrónimo de Long Term Evolution. Es el estándar para la comunicación inalámbrica de datos de alta velocidad dedicada a teléfonos móviles y terminales de datos.

· **NFC**: abreviatura de *Near-field communication* o "comunicación de campo cercano" en español. Se trata de una tecnología de comunicación inalámbrica de corto alcance y alta frecuencia que permite el intercambio de datos entre dispositivos.

· **WiMAX**: siglas de Worldwide Interoperability for Microwave Access (interoperabilidad mundial para acceso por microondas). Es un estándar de transferencia de datos que utiliza ondas de radio con una frecuencia de 2,5 a 5,8 GHz y puede tener un alcance de hasta 70 km.

· **APK**: acrónimo de Android Application Package. Es un paquete para el sistema operativo Android.

· **API**: acrónimo de Application Programming Interface o "Interfaz de Programación de Aplicaciones" en español. Se trata de un conjunto de subrutinas, funciones y procedimientos que se proporciona a una biblioteca determinada para su uso como capa de abstracción en otro software.

· **SysML**: siglas de Systems Modeling Language. Es un lenguaje de especificación de sistemas ampliado de UML 2.0, y desde el 19 de septiembre de 2007 un estándar de la OMG.

· **WEP**: acrónimo de Wired Equivalent Privacy o "Privacidad equivalente a cableado" en español. Es un protocolo de red inalámbrica que puede cifrar y transmitir información según el estándar IEEE 802.11.

· **WPA2**: abreviatura de Wi-Fi Protected Access 2 o "Acceso Wi-Fi protegido 2" en español. Es un sistema de protección para redes inalámbricas (Wi-Fi), creado para corregir las deficiencias de los sistemas anteriores en el nuevo estándar 802.11i.

· **SDK**: acrónimo de Kit de Desarrollo de Software. Es una colección de herramientas de desarrollo de software en paquetes instalables.

· **ABI**: siglas de Application Binary Interface o "Interfaz Binaria de Aplicación" en español. Es la interfaz entre dos módulos de programa. Uno de ellos suele ser una biblioteca o sistema operativo a nivel de lenguaje de máquina.