



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Escuela de Ingeniería Informática



Aplicación para el seguimiento de repartos en una empresa de transportes para administradores, repartidores y clientes

PARTE 1

Tutor: Abraham Rodríguez Rodríguez

Autor: Miguel González Villa

Noviembre de 2017

Índice

1.	ESTADO ACTUAL Y OBJETIVOS INICIALES.....	1
1.1.	INTRODUCCIÓN.....	1
1.2.	DESCRIPCIÓN DE OBJETIVOS.....	1
1.3.	METODOLOGÍA.....	2
2.	JUSTIFICACIÓN DE LAS COMPETENCIAS ESPECÍFICAS CUBIERTAS.....	3
3.	APORTACIONES AL ENTORNO SOCIO-ECONÓMICO, TÉCNICO O CIENTÍFICO.....	5
4.	DESARROLLO	5
4.1.	ANÁLISIS DE APLICACIONES SIMILARES	6
4.2.	ANÁLISIS DE REQUISITOS.....	8
4.2.1.	<i>Definición de tareas</i>	9
4.3.	DIAGRAMA DE CASOS DE USO	13
4.4.	MOCKUPS.....	20
4.5.	MODELO DE LA BASE DE DATOS.....	29
4.6.	ALCANCE DE LA IMPLEMENTACIÓN.....	32
4.7.	REQUISITOS HARDWARE MÍNIMOS	33
4.8.	HERRAMIENTAS DE DESARROLLO	33
4.9.	DISEÑO E IMPLEMENTACIÓN.....	35
4.9.1.	<i>Arquitectura del software</i>	35
4.9.2.	<i>Desarrollo del prototipo</i>	37
4.9.3.	<i>Pruebas</i>	40
4.9.4.	<i>Acceso al código</i>	41
5.	ASPECTOS ECONÓMICOS Y TEMPORALES	41
6.	NORMATIVA Y LEGISLACIÓN	42
7.	INTEGRACIÓN CON LA PARTE 2	43
8.	CONCLUSIONES Y TRABAJOS FUTUROS	2
9.	FUENTES DE INFORMACIÓN	5
10.	ANEXOS.....	6
	ANEXO I: MANUAL DE USUARIO Y SOFTWARE DE LA APLICACIÓN DEL CLIENTE	6
	ANEXO II: MANUAL DE USUARIO Y SOFTWARE DE LA APLICACIÓN DEL REPARTIDOR	10

1. Estado actual y objetivos iniciales

1.1. Introducción

El objetivo inicial del proyecto era crear una serie de herramientas que pudieran ser útiles en el mundo laboral, de cara a obtener una experiencia más enriquecedora. Es por ello que, junto con la alumna María Guerra Suárez, se analizaron las ideas previas y se llegó a la conclusión de desarrollar una serie de aplicaciones para una hipotética empresa de repartos. Con el objetivo de tratar de mejorar el sistema de repartos de dicha empresa, nos centramos en la productividad y simplicidad de los usuarios.

Se identificaron tres agentes principales en el proceso de entrega de un paquete, el administrador del almacén, el repartidor y el cliente (receptor del paquete). El trabajo de los dos primeros podía agilizarse y simplificarse, mientras que el tercer agente requería una ampliación de la información que las compañías de repartos suelen proporcionar. Para lograr la movilidad necesaria en esta situación, se desarrollan tres aplicaciones móviles para cada uno de los agentes.

La primera parte de este proyecto se centrará en la aplicación destinada al repartidor, cuyo trabajo consiste en la capacidad de visualizar la información de un pedido asignado, validar su recogida y entrega y contactar con el receptor para su entrega entre otras tareas, y el cliente, que actuará como receptor de pedidos y podrá consultar sus encargos, estado de los mismos, localización del repartidor que los entregará y visualización de un historial con los encargos pasados.

1.2. Descripción de objetivos

Identificados los requisitos propuestos de la aplicación del repartidor y el cliente, además de los que el tutor ha creído convenientes añadir, se pretende desarrollar dos prototipos de aplicación móvil que ejemplifique el trabajo de desarrollo en un entorno real. Dichos prototipos pueden ampliarse hasta obtener una versión final con la debida financiación por parte de una empresa de transportes interesada.

Si bien se trata de dos prototipos de aplicación, y por ello presentan algunas limitaciones, se ha tratado de realizar un trabajo muy cercano a la realidad. Debido al carácter educativo del TFT, los principales objetivos propuestos por el estudiante durante el desarrollo del proyecto son los siguientes:

- Aprendizaje de una tecnología de reciente desarrollo como es *ionic 3* y *Angular 4*.
- Mejora en los conocimientos de diseño y desarrollo web (*HTML5* y *CSS3*) obtenidos hasta ahora.
- Adquirir habilidad en el uso del lenguaje de programación *Typescript*.
- Familiarizarse con el uso de librerías y API's móviles para implementar y aprovechar las funcionalidades y servicios de un dispositivo móvil.
- Aprender a desenvolverse con un servidor de almacenamiento remoto como *Firebase*.

Como se puede observar, los objetivos están muy centrados en adquirir los conocimientos necesarios para el desarrollo de aplicaciones en *Ionic 3*, un *framework* sustentado sobre *Angular 4*, *Typescript 2.5*, *HTML5* y *CSS3*. La decisión de usar un entorno de desarrollo que apenas tiene unos meses de vida se debe al deseo de trabajar desde un punto muy temprano en un *framework* que probablemente acapare el desarrollo de aplicaciones web en un futuro cercano, con las ventajas que ello proporciona a nivel laboral. Es notable la falta de documentación con respecto a desarrollar en otras plataformas como *Android* (que se sustenta en *Java*) o *iOS*, pero se compensa con una interfaz sencilla y la posibilidad de construir aplicaciones para ambos dispositivos anteriormente nombrados sin la necesidad de aprender a trabajar en cada uno por separado.

1.3. Metodología

Se ha utilizado una metodología convencional en cascada con retroalimentación para la realización de este trabajo. Se trata de una metodología que permite ordenar de manera rigurosa las fases del proceso del desarrollo del software. Cada etapa posee ciertos objetivos que se deben cumplir siguiendo un conjunto de actividades específicas para cada una. En la *Imagen 1* se puede observar las distintas fases del modelo en cascada.

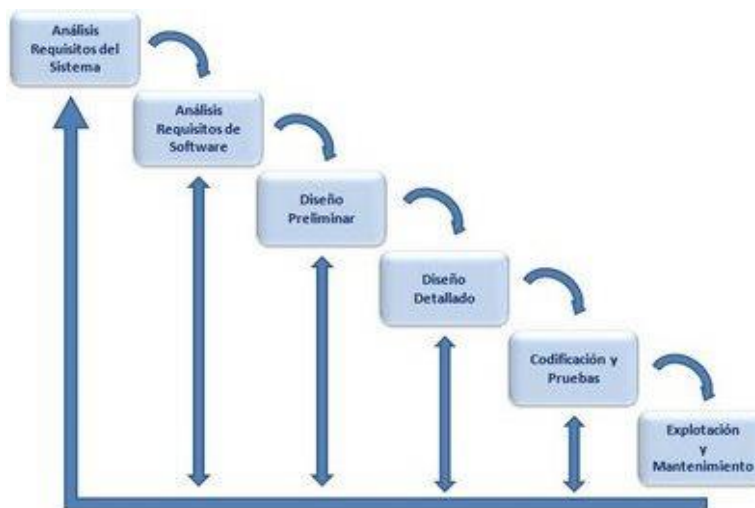


Imagen 1: Etapas de la metodología en cascada

En el TFT01 se especificaron las siguientes tareas a realizar para las dos aplicaciones:

Repartidor:

- Localización geográfica e indicaciones de ruta a seguir del punto de entrega de un pedido.
- Selección del pedido que será el siguiente en ser entregado.
- Verificación del código del pedido a entregar mediante escaneo de imagen.
- Posibilidad por parte del cliente de añadir incidencias al recibir un pedido.
- Proceso de entrega del pedido en el que el cliente firma su recepción.

Cliente:

- Posibilidad de registro en la aplicación.
- Visualización de los pedidos encargados y su estado actual.
- Visualización de la localización del repartidor cuando entregue un pedido asignado.

Aunque la mayoría de estas tareas se han implantado e incluso se ha ampliado a algunas más que no aparecen (como el historial de los pedidos entregados en el caso del cliente y del repartidor, y la posibilidad de realizar llamadas a los clientes en la aplicación del repartidor), hay una de ellas que se ha decidido modificar. Los pedidos se asignan al repartidor a través de la aplicación del administrador, siendo estos revisados con anterioridad.

2. Justificación de las competencias específicas cubiertas

Las competencias que cubre este proyecto comienzan con el aprendizaje de una nueva área de conocimiento y el desarrollo de la misma, además del despliegue del trabajo realizado. Todo ello sin descuidar la importancia del trabajo en equipo o el conocimiento de la legislación vigente. Pueden destacarse las siguientes competencias:

G1: *Poseer y comprender conocimientos en un área de estudio (Ingeniería Informática) que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.*

En este apartado se puede destacar el aprendizaje relativo a las nuevas tecnologías que forman el proyecto, como *ionic 3* o el refuerzo de los conocimientos en desarrollo web que se adquirieron a lo largo del periodo universitario.

G2: *Aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.*

El proyecto desarrollado es una buena prueba de la aplicación de los conocimientos adquiridos y la resolución de un problema planteado como es la mejora de la productividad en el sector de transportes, teniendo en cuenta especialmente la profesionalidad con la que se ha trabajado para obtener unas aplicaciones lo más cercanas posibles al mundo real.

N2: *Cooperar con otras personas y organizaciones en la realización eficaz de funciones y tareas propias de su perfil profesional, desarrollando una actitud reflexiva sobre sus propias competencias y conocimientos profesionales y una actitud comprensiva y empática hacia las competencias y conocimientos de otros profesionales.*

Al tratarse de un proyecto que se ha desarrollado en colaboración con otro alumno, la cooperación ha sido clave para resolver los problemas encontrados. Las tres aplicaciones que conforman la totalidad del proyecto deben su éxito al trabajo en equipo.

T7: Capacidad para conocer, comprender y aplicar la legislación necesaria durante el desarrollo de la profesión de Ingeniero Técnico en Informática y manejar especificaciones, reglamentos y normas de obligado cumplimiento.

Puesto que las aplicaciones pretenden mejorar el proceso de gestión y reparto de una empresa de transporte, era inevitable tratar con datos personales, tanto de los trabajadores de la empresa como los receptores de paquetes.

T101: Capacidad para comprender el entorno de una organización y sus necesidades en el ámbito de las tecnologías de la información y las comunicaciones.

El primer reto que se ha debido superar a la hora de realizar este proyecto es justamente esta competencia. Hasta que no se identificaron las necesidades de una empresa de transportes no se pudo desarrollar un proyecto que cumpliera dichas necesidades y garantizase un manejo fiable y seguro.

T102: Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, construir, gestionar, explotar y mantener las tecnologías de hardware, software y redes, dentro de los parámetros de coste y calidad adecuados.

La sincronización entre las aplicaciones que forman el proyecto es una necesidad que no puede evitarse, pero hay una pieza clave en el desarrollo y se trata del servidor de almacenamiento empleado. Dicho servidor se emplea como gestor de autenticación de los usuarios, gestor de la base de datos y como espacio de almacenamiento, por lo que la correcta configuración del mismo (además de la correcta ejecución de las operaciones de lectura y escritura realizadas desde las aplicaciones) son vitales para un funcionamiento idóneo.

T106: Capacidad de concebir sistemas, aplicaciones y servicios basados en tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil.

El desarrollo del proyecto ha permitido alcanzar esta competencia con facilidad, todo ello gracias al planteamiento móvil de las aplicaciones, pero también al desarrollo basado en tecnologías web que aporta Ionic 3.

TFG01: *Ejercicio original a realizar individualmente y presentar y defender ante un tribunal universitario, consistente en un proyecto en el ámbito de las tecnologías específicas de la Ingeniería en Informática de naturaleza profesional en el que se sinteticen e integren las competencias adquiridas en las enseñanzas.*

Todos los conocimientos adquiridos a lo largo del periodo universitario se condensan en un proyecto con grandes proyecciones a un cercano futuro profesional, proyecto que se encuentra detallado en una memoria final dividida en dos partes.

3. Aportaciones al entorno socio-económico, técnico o científico

Al tratarse de un proyecto orientado a una hipotética empresa de transportes, puede considerarse una excelente opción de mejora del servicio a un coste reducido para empresas de logística de tamaño pequeño, habituales en el entorno actual. La versatilidad para trabajar con *Ionic* supone una importante mejora a la hora de desarrollar aplicaciones destinadas al mundo empresarial, tanto soluciones de bajo coste como proyectos de mayor envergadura en colaboración con un equipo informático, sin importar la plataforma o el tipo de proyecto en el que se trabaje.

Dicho entorno en colaboración con un equipo se ha trabajado durante este proyecto al tratarse de trabajo conjunto, que tiene una vital consideración en el ámbito laboral. Además de ser importante el trabajo en equipo, también lo es la comunicación entre cliente y desarrollador. Se trata de un aspecto clave tener la capacidad de tratar adecuadamente con un cliente, ofrecerle un servicio adecuado y ser honesto a la hora de presentar el desarrollo actual, aspectos que se trabajaron en conjunto con el tutor del proyecto.

Finalmente, añadir que los conocimientos adquiridos durante el desarrollo del proyecto han logrado crear una serie de prototipos perfectamente funcionales, pero que, debido a carecer de un equipo de trabajo, la finalización de los mismos o la mejora del servicio son inalcanzables en este momento. Por otra parte, una empresa real que esté interesada en la adquisición de este servicio, podría sufragar los costes derivados, llegando a disponer de una versión final de las aplicaciones.

4. Desarrollo

El proyecto está formado por un conjunto de tres aplicaciones, pero en esta memoria solo se mostrará el desarrollo de la primera parte de dicho proyecto, fundamentada en las aplicaciones diseñadas para el repartidor y el cliente.

4.1. Análisis de aplicaciones similares

Antes de comenzar con el análisis de los requisitos del proyecto, es importante valorar las soluciones software existentes de forma que el desarrollo del prototipo pueda aportar algún valor tecnológico de interés y diferenciarse con funcionalidades nuevas que resulten de utilidad para la tarea concreta que se pretende desarrollar.

Los sistemas empleados en el campo de la distribución de paquetería se basan en los SGA, cuyas siglas se corresponden a “Software de Gestión de Almacén”. Estos sistemas se encargan de facilitar la gestión de almacenes y automatización de control de mercancía, además del registro de entradas y salidas de mercancía.

El primer sistema para repartidores analizado es “[OnERP](#)”, un software online desarrollado por *Web Evolutive Business Software S.L* que, basándose en el uso de módulos es capaz de cubrir las necesidades de cualquier empresa que quiera mejorar sus sistemas en los campos de gestión de recursos humanos, ventas, almacenamiento, contabilidad, gestión de proyectos, producción, calidad o seguridad entre otros. Se encuentra el módulo de almacén, un programa para gestión de almacenes que posee control de existencias y trazabilidad completa, así como control de expediciones con su correspondiente conexión con dispositivos móviles. Por tratarse de una solución *online*, es posible acceder desde cualquier dispositivo, ya sea un ordenador que se encuentre en el almacén o un *Smartphone*.

Por otra parte, analizamos la aplicación utilizada por Correos en sus nuevas *PDA*, que permite agilizar enormemente el trabajo de los repartidores. A parte de las funcionalidades más habituales en este tipo de sistemas, esta aplicación ofrece otras mucho más innovadoras como el rastreo de la firma biométrica, la captura de forma automática de los datos de un DNI mediante el código MZR, garantizando así quién firma la recepción del pedido (contrastando dicho DNI con una base de datos centralizada), y la posibilidad de realizar pagos de los servicios a través de tarjetas bancarias. Además, permite los servicios *Smart cities*, donde el repartidor podrá hacer y enviar fotografías con el objetivo de prestar servicios de utilidad pública. Gracias al GPS, se podrá ubicar de manera exacta dónde ha sido realizada las fotos y ser enviadas a entidades interesadas.

“[AfterShip](#)”, la primera de las aplicaciones para clientes analizadas, es una aplicación móvil creada por *AfterShip Limited* que soporta más de 370 servicios de mensajería y ofrece una interfaz atractiva y sencilla. Tiene un amplio abanico de funcionalidades como la detección automática de la agencia de transporte al introducir el código de seguimiento, avisos de los cambios del estado de los pedidos mediante notificaciones y escáner de código de barras entre otros. Además, cuenta con una versión para navegadores.

Otra de las aplicaciones para el seguimiento de envíos es “[Deliveries Package Tracker](#)”, creada por *oRRs*. Ofrece una interfaz muy intuitiva y posee un menú lateral deslizante desde donde se puede acceder con facilidad a todas las funcionalidades disponibles. Posee un diseño adaptable que deja a la vista una nueva columna de información en *tablets*. Es posible comprobar las novedades de los envíos de forma manual o automática, dispone de notificaciones *push* y posee una gran lista de agencias de mensajería. Finalmente, permite la visualización en un mapa de la ubicación de la compra y sistema propio de sincronización entre dispositivos.

La última de las aplicaciones destinadas a clientes es “[Correos Info](#)”, creada por *Correos*. Facilita la información sobre los envíos que has realizado al introducir el código de envío (o bien leyendo el código con la cámara del dispositivo), con la posibilidad de incorporar notas personalizadas a cada envío. Por otra parte la aplicación incluye la posibilidad de consultar códigos postales y buscar oficinas cercanas con su respectiva información.

Los sistemas anteriormente descritos cuentan con un equipo que trabaja en ellos, actualizando y proporcionando mantenimiento a las empresas o clientes que adquieran cualquiera de las aplicaciones, por lo que la calidad de dichos sistemas está garantizada, así como su fiabilidad. Para que el proyecto suponga una diferenciación en un campo como la logística es importante fijarse en los grandes aciertos de otras compañías de “software” a la hora de diseñar sus productos. Es por ello que, además de requerir funcionalidades de verificación de pedidos y establecimiento de la ruta más óptima al destino de entrega, un sistema dedicado a un repartidor deberá emplearse en un dispositivo portátil como un *SmartPhone*, además de requerir una interfaz sencilla, pero con las operaciones debidamente establecidas y en perfecta sincronización con la aplicación del administrador y en última instancia, con la del cliente. Para cumplir estas características se desarrollará la aplicación teniendo en mente dispositivos móviles. Por parte del cliente, en ocasiones es evidente la falta de información que se le proporciona en las aplicaciones dedicadas (ya sea para dispositivos móviles o en aplicaciones web). Por ello optamos por mostrar la mayor cantidad posible de información, individualizada a cada usuario y sin tener que introducir largos códigos de seguimiento, en ocasiones se trata de códigos alfanuméricos de hasta 20 caracteres. Para agilizar el uso de la aplicación y con intención de aprovechar el auge de las compras mediante dispositivos móviles, se ha desarrollado para *SmartPhone*, pero también sería posible desarrollar una versión para navegadores en base al trabajo realizado.

En la *Tabla 1* y *Tabla 2*, se muestra una lista de las características más relevantes para el proyecto de cada una de las aplicaciones analizadas.

APLICACIONES SIMILARES CLIENTE	CARACTERÍSTICAS MÁS IMPORTANTES
	- Notificaciones tanto dentro como fuera de la aplicación. - Versión para navegadores.
	- Posibilidad de añadir título al paquete. - Posibilidad de diferenciar los paquetes por categorías.
	- Posibilidad de consultar códigos postales y buscar oficinas cercanas.

Tabla 1: Características relevantes de las aplicaciones similares a la del cliente

APLICACIONES SIMILARES REPARTIDOR	CARACTERÍSTICAS MÁS IMPORTANTES
	- Versión para navegadores.
	- Captura de los datos del DNI mediante MZR. - Servicios <i>Smart Cities</i>

Tabla 2: Características relevantes de las aplicaciones similares a la del repartidor

4.2. Análisis de requisitos

El primer paso del desarrollo del proyecto consiste en realizar un análisis exhaustivo de las funcionalidades que son necesarias para la realización de un correcto trabajo por parte del repartidor y una experiencia satisfactoria por parte del cliente. Los requisitos funcionales del repartidor son:

1. Verificación del pedido a repartir mediante escaneo de imagen.
2. Visualización del punto de entrega del pedido seleccionado.
3. Posibilidad de comunicarse con el administrador a través de la aplicación.
4. Visualización de un historial de pedidos entregados.
5. Restricción de acceso a usuarios no identificados.
6. Posibilidad de contactar con el receptor del pedido mediante el servicio telefónico.
7. Verificación del pedido a entregar mediante escaneo de imagen.
8. Posibilidad de añadir incidencias u observaciones por parte del cliente.
9. Proceso de entrega del pedido en el que el cliente firma su recepción.

Los requisitos funcionales del cliente son:

1. Posibilidad de registro en la aplicación.
2. Visualización detallada de los pedidos encargados y su estado actual.
3. Visualización de la localización del repartidor cuando entregue un pedido asignado.
4. Restricción de acceso a usuarios no identificados.
5. Valoración de una entrega.
6. Visualización de un historial de pedidos recibidos.

Los requisitos no funcionales para ambas aplicaciones son aquellos que especifican características de funcionamiento. Entre ellos encontramos:

1. **Rendimiento:** Las aplicaciones requieren una velocidad de ejecución de operaciones lo suficientemente alta como para no perjudicar el flujo de trabajo del usuario que la emplee. El código empleado deberá ser ligero y rápido para funcionar con normalidad en la mayoría de dispositivos móviles del mercado.

2. **Estabilidad:** Es preciso que las aplicaciones no presenten errores durante su uso, puesto que podría causar problemas en el proceso de entrega de pedidos o la recepción de información por parte del cliente.
3. **Portabilidad:** Un requisito a destacar es que las aplicaciones serán empleadas en un “Smartphone”.
4. **Usabilidad:** Las aplicaciones presentarán una gran sencillez de uso y facilidad de acceso a cualquier punto de la misma en pocos pasos, lo que garantizará un estilo fluido e intuitivo.

4.2.1. Definición de tareas

Una vez definidas las funcionalidades, es necesario establecer una serie de requisitos para cada una, lo que desembocará en un listado de tareas que serán implementadas en las aplicaciones. Comenzamos por los requisitos de la aplicación del repartidor:

1. *Verificación del pedido a repartir mediante escaneo de imagen.*

El repartidor deberá ser capaz de verificar un pedido seleccionado mediante escaneo de un código de imagen como *QR* o de barras. Una vez verificado el repartidor procederá a su entrega. Para cumplir con esta funcionalidad, se han definido las siguientes tareas:

- Mostrar en la pantalla principal los pedidos que se encuentran asignados y en el almacén, con información breve sobre cada uno.
- Implementar la opción de escanear un pedido empleando la cámara del dispositivo.
- Posibilidad de introducir el código del pedido manualmente.
- El pedido se desplazará a la sección de “En reparto”, con su estado actualizado.

2. *Visualización del punto de entrega del pedido seleccionado.*

El repartidor deberá ser capaz de ver la dirección de entrega de un pedido en un mapa, además de disponer de la mejor ruta hasta ese punto. Se cumplirá esta funcionalidad a raíz de completar las siguientes tareas:

- Implementación del módulo del mapa y creación de la página para mostrarlo.
- El mapa con la localización del vehículo será accesible desde la página principal.
- Al pulsar sobre el marcador de posición, se desplegará la opción para generar una ruta óptima al punto de entrega.

3. Posibilidad de comunicarse con los repartidores a través de la aplicación.

El repartidor tendrá la posibilidad de comunicarse con el administrador en privado en cualquier momento sin salir de la aplicación. De esta forma podrá informar personalmente de los paquetes que han sido entregados, proporcionar alguna observación sobre algún pedido en concreto.

- En el menú lateral se mostrará un botón con el que acceder a una página nueva desde la que intercambiar mensajes con el repartidor deseado.

4. Posibilidad de contactar con el receptor del pedido mediante el servicio telefónico.

El repartidor podrá usar su *SmartPhone* para contactar con el receptor del pedido a través de un módulo implementado en la aplicación. Las tareas previstas son:

- Implementación del módulo que permite realizar llamadas telefónicas desde la aplicación.

5. Verificación del pedido a entregar mediante escaneo de imagen.

Del mismo modo que un pedido será verificado previamente a su proceso de reparto, también es necesario verificarlo en el instante previo a entregarlo. Este proceso se efectuará mediante un escaneo de imagen, tras lo que se procederá a la entrega del pedido y su firma. Las tareas necesarias son:

- Mostrar en la pantalla de pedidos “En reparto” los pedidos que se encuentran en reparto.
- Implementar la opción de escanear un pedido empleando la cámara del dispositivo.
- Posibilidad de introducir el código del pedido manualmente.
- Tras verificarlo, se mostrará la pantalla de firma de recepción del pedido por parte del cliente.

6. Posibilidad de añadir incidencias u observaciones por parte del cliente.

Durante el proceso de entrega y previo a que el cliente firme la recepción del pedido, el repartidor dispondrá al cliente la opción de añadir una incidencia u observación en caso de que hubiera algún fallo.

- Creación de una ventana emergente en el panel de firma para añadir incidencias.

7. Proceso de entrega del pedido en el que el cliente firma su recepción.

Para culminar el proceso de entrega, será necesaria la confirmación mediante firma por parte del cliente.

- Creación de una nueva página dedicada al panel para firmar.
- Implementación del módulo que gestiona el panel de firma.
- Actualización de la base de datos con la firma y datos relevantes como el estado, fecha de entrega, u observaciones si las hubiera entre otros.

8. Visualización de un historial de pedidos entregados

El trabajo diario del repartidor requiere también la posibilidad de consultar el historial de pedidos entregados, en caso de existir una reclamación por parte del cliente, una petición de consulta por parte de la empresa remitente o para la consulta y estadística posteriores. Las tareas a realizar son:

- Creación de una nueva página para mostrar el histórico.
- Introducción de un sistema de ordenación y filtrado por múltiples campos.

9. Restricción de acceso a usuarios no identificados.

Será necesario que el usuario, que en este caso será el repartidor, se identifique para poder acceder a la aplicación. De esta manera se controla el acceso de usuarios no autorizados.

- Sistema de autenticación al inicio de la aplicación basado en correo electrónico y clave.
- Sistema de cierre de sesión.

Requisitos de la aplicación del cliente:

1. Posibilidad de registro en la aplicación.

El cliente podrá usar su *SmartPhone* para registrarse en la aplicación. Siempre y cuando disponga de pedidos asignados al DNI con el que creó la cuenta, se mostrarán en la pantalla principal o el historial. Las tareas previstas son:

- Implementación del código de registro de usuarios mediante DNI, nombre y contraseña.

2. Visualización detallada de los pedidos encargados y su estado actual.

El cliente tendrá acceso a información detallada de los pedidos encargados, así como observar su estado actual y hora de llegada al almacén entre otros. Las tareas necesarias son:

- Mostrar en la pantalla principal los pedidos que no están entregados, indicando con un icono su estado.
- Posibilidad de acceder a los pedidos para obtener más información relevante.

3. Visualización de la localización del repartidor cuando entregue un pedido asignado.

Durante el proceso de reparto de un pedido, cuando el repartidor marque un pedido como siguiente en entrega, el cliente dispondrá de la localización en tiempo real del repartidor.

- Creación de una página para contener el mapa.
- Implementación del módulo del mapa y del código actualizable de la posición.

4. Visualización de un historial de pedidos recibidos.

Para el cliente será útil disponer de un historial con los pedidos recibidos, fechas de entrega, observaciones realizadas y otros campos importantes. Las tareas a realizar son:

- Creación de una nueva página para mostrar el histórico.
- Introducción de un sistema de ordenación y filtrado por múltiples campos.

5. Valoración de una entrega.

El cliente podría estar interesado en la opción de implicarse en el proceso de entrega de un paquete mediante la valoración de una entrega recibida. De esta forma, asignaría una nota a la entrega y el administrador podrá observar dichas valoraciones, contrastarlas con los repartidores que las entregaron y desarrollar un informe sobre la calidad del repartidor, gracias a la retroalimentación que provee el cliente. Las tareas a realizar son:

- Creación de una nueva sección en la visualización de un paquete desde el historial para valorar una entrega.
- Introducción de un sistema de valoración que será enviado al servidor.

6. Restricción de acceso a usuarios no identificados.

Será necesario que el usuario, que en este caso será el cliente, se identifique para poder acceder a la aplicación. De esta manera se controla el acceso de usuarios no autorizados.

- Sistema de autenticación al inicio de la aplicación basado en DNI y clave.
- Sistema de cierre de sesión.

4.3. Diagrama de casos de uso

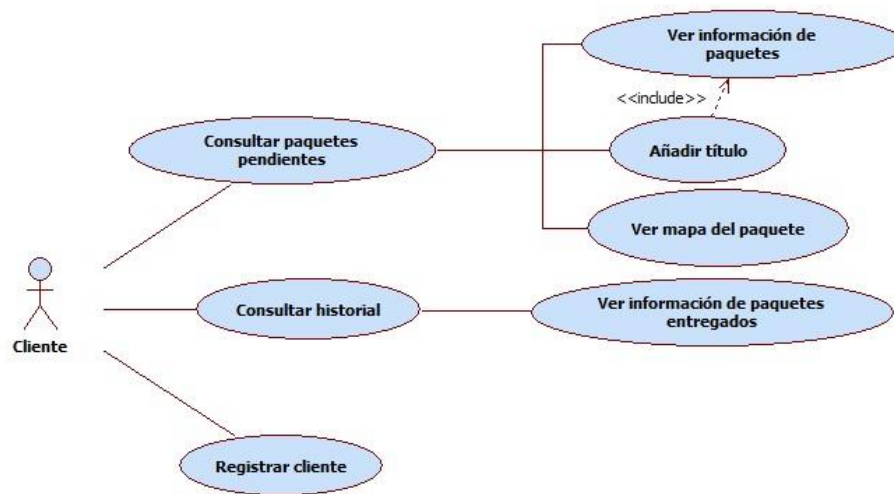


Imagen II: Diagrama de casos de uso para el Cliente

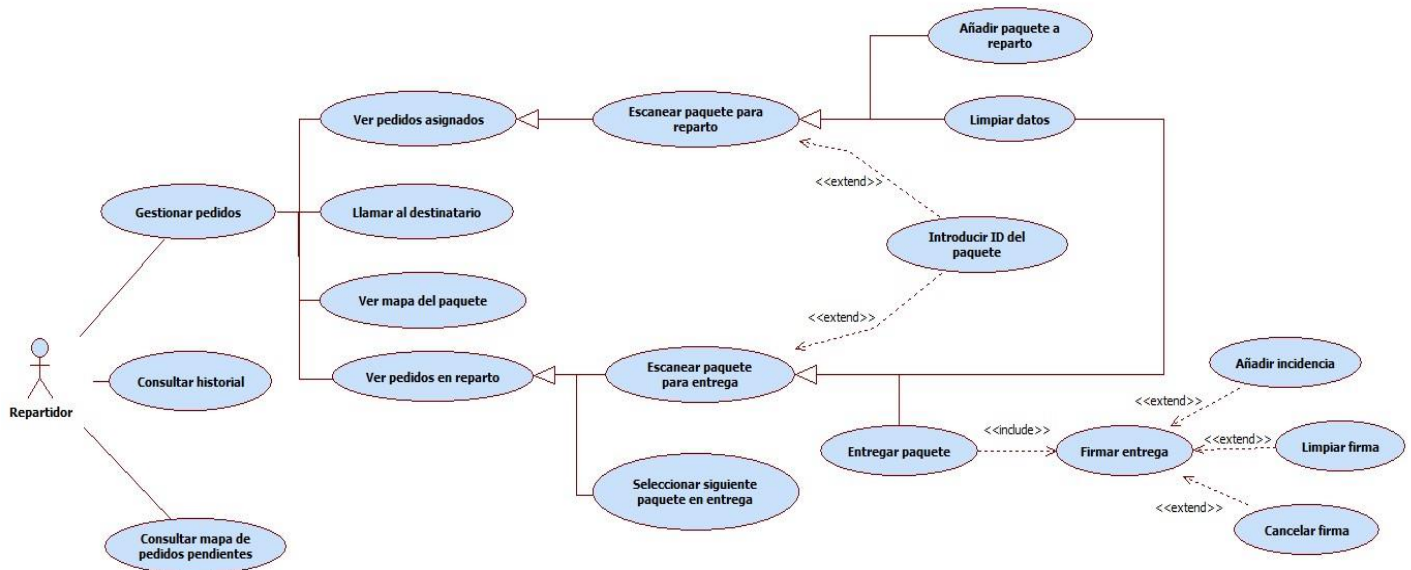


Imagen III: Diagrama de casos de uso para el Repartidor

A continuación, se definen cada uno de los casos de uso mostrados en la *Imagen II*.

CU-001	Consultar paquetes pendientes
Versión	1.0
Descripción	El cliente puede ver una lista de paquetes que aún no le han sido entregados.
Secuencia normal	1. Iniciar sesión en la aplicación.
Frecuencia	Alta
Importancia	Alta

CU-002	Ver información del paquete
Versión	1.0
Descripción	El cliente puede ver toda la información relevante de un paquete seleccionado.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Seleccionar un paquete en la página principal.
Frecuencia	Alta
Importancia	Alta

CU-003	Añadir título
Versión	1.0
Descripción	El usuario puede añadir un título a un paquete.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Seleccionar un paquete en la página principal. 3. Pulsar icono del lápiz. 4. Confirmar.
Frecuencia	Media
Importancia	Media

CU-004	Ver mapa del paquete
Versión	1.0
Descripción	El cliente puede ver la localización de su pedido en tiempo real en un mapa cuando va a ser entregado.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Seleccionar botón de mapa.
Frecuencia	Alta
Importancia	Alta

CU-005	Consultar historial
Versión	1.0
Descripción	El cliente puede consultar una lista de los paquetes que le han sido entregados.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Seleccionar “Historial de pedidos” en el menú lateral.
Frecuencia	Alta
Importancia	Alta

CU-006	Ver información de paquetes entregados
Versión	1.0
Descripción	Se muestra toda la información relevante de un paquete.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Seleccionar “Historial de pedidos” en el menú lateral. 3. Seleccionar un paquete de la lista.
Frecuencia	Media
Importancia	Alta

CU-007	Registrar cliente
Versión	1.0
Descripción	El usuario se puede registrar en la aplicación para luego acceder a ella.
Secuencia normal	1. Pulsar link “Regístrate” en la página de acceso. 2. Rellenar formulario de registro. 3. Confirmar.
Frecuencia	Alta
Importancia	Alta

De la misma forma, se detalla cada uno de los casos de uso que se reflejan en la *Imagen III*.

CU-001	Gestionar paquetes
Versión	1.0
Descripción	Es la parte de la aplicación que se encarga de los paquetes del repartidor.
Secuencia normal	1. Iniciar sesión en la aplicación.
Frecuencia	Alta
Importancia	Alta

CU-002	Ver pedidos asignados
Versión	1.0
Descripción	El repartidor puede ver los paquetes que le han sido asignados.
Secuencia normal	1. Iniciar sesión en la aplicación.
Frecuencia	Alta
Importancia	Alta

CU-003	Ver mapa del paquete
Versión	1.0
Descripción	Se muestra en un mapa la dirección donde debe ser entregado el paquete.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Seleccionar el botón del mapa de un paquete de la pestaña "Asignados" o "En reparto".
Frecuencia	Alta
Importancia	Alta

CU-004	Llamar al destinatario
Versión	1.0
Descripción	El repartidor puede hacer una llamada al destinatario.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Seleccionar el botón de llamada de un paquete de la pestaña "Asignados" o "En reparto".
Frecuencia	Alta
Importancia	Alta

CU-005	Escanear paquete para reparto
Versión	1.0
Descripción	El repartidor debe escanear el código del paquete antes de meterlo en el vehículo.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Pulsar el botón de escáner de un paquete de la pestaña "Asignados".
Frecuencia	Alta
Importancia	Alta

CU-006	Añadir paquete a reparto
Versión	1.0
Descripción	Una vez escaneado un paquete se puede añadir al reparto y posteriormente al vehículo.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Pulsar el botón de escáner de un paquete de la pestaña "Asignados". 3. Escanear el paquete. 4. Pulsar el botón "Añadir paquete".
Frecuencia	Alta
Importancia	Alta

CU-007	Limpiar datos
Versión	1.0
Descripción	Una vez escaneado un paquete se puede limpiar el código obtenido.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Pulsar el botón de escáner de un paquete de la pestaña "Asignados" o "En reparto". 3. Escanear el paquete. 4. Pulsar botón "Limpiar".
Frecuencia	Baja
Importancia	Media

CU-008	Ver pedidos en reparto
Versión	1.0
Descripción	El repartidor puede ver una lista de los paquetes que ha seleccionado para el reparto.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Seleccionar pestaña "En reparto".
Frecuencia	Alta
Importancia	Alta

CU-009	Escanear paquete para entrega
Versión	1.0
Descripción	El repartidor debe escanear un paquete antes de entregarlo al destinatario.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Pulsar el botón de escáner de un paquete de la pestaña "En reparto".
Frecuencia	Alta
Importancia	Alta

CU-010	Entregar paquete
Versión	1.0
Descripción	Una vez escaneado un paquete que será entregado, el repartidor puede confirmar la entrega.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Pulsar el botón de escáner de un paquete de la pestaña "En reparto". 3. Escanear el paquete. 4. Pulsar botón "Entregar paquete".
Frecuencia	Alta
Importancia	Alta

CU-011	Firmar entrega
Versión	1.0
Descripción	El destinatario deberá firmar para confirmar que ha recibido su pedido.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Pulsar el botón de escáner de un paquete de la pestaña "En reparto" y escanear. 3. Pulsar botón "Entregar paquete". 4. Firmar 5. Confirmar.
Frecuencia	Alta
Importancia	Alta

CU-012	Seleccionar siguiente paquete en entrega
Versión	1.0
Descripción	El repartidor debe seleccionar el siguiente paquete que será entregado.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Seleccionar la pestaña "En reparto". 3. Pulsar sobre un paquete.
Frecuencia	Alta
Importancia	Alta

CU-013	Consultar historial
Versión	1.0
Descripción	El repartidor puede ver una lista de todos los paquetes que ha entregado.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Seleccionar "Historial de pedidos" en el menú lateral.
Frecuencia	Alta
Importancia	Alta

CU-014	Consultar mapa de pedidos pendientes
Versión	1.0
Descripción	El repartidor puede consultar en un mapa la ubicación de entrega de los pedidos que le han sido asignados.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Seleccionar “Mapa de pedidos pendientes” en el menú.
Frecuencia	Alta
Importancia	Alta

CU-015	Introducir ID del paquete
Versión	1.0
Descripción	El repartidor tiene la opción de introducir el ID del paquete de forma manual cuando falla el escaneo.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Pulsar el botón de escáner de un paquete de la pestaña “En reparto” o “Asignados” y escanear. 3. Pulsar botón “Esc. manual”.
Frecuencia	Media
Importancia	Alta

CU-016	Añadir incidencia
Versión	1.0
Descripción	Opción de añadir una incidencia al paquete entregado.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Pulsar el botón de escáner de un paquete de la pestaña “En reparto” y escanear. 3. Pulsar botón “Entregar paquete”. 4. Pulsar botón “Añadir incidencia”. 5. Confirmar.
Frecuencia	Media
Importancia	Alta

CU-017	Cancelar firma
Versión	1.0
Descripción	El repartidor puede cancelar una firma.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Pulsar el botón de escáner de un paquete de la pestaña “En reparto” y escanear. 3. Pulsar botón “Entregar paquete”. 4. Pulsar botón “Cancelar”. 5. Confirmar.
Frecuencia	Baja
Importancia	Alta

CU-018	Limpiar firma
Versión	1.0
Descripción	El repartidor puede rectificar una firma.
Secuencia normal	1. Iniciar sesión en la aplicación. 2. Pulsar el botón de escáner de un paquete de la pestaña “En reparto” y escanear. 3. Pulsar botón “Entregar paquete”. 4. Pulsar botón “Limpiar”. 5. Confirmar.
Frecuencia	Media
Importancia	Alta

4.4. Mockups

Una vez analizados los casos de uso, era el momento de diseñar unas muestras de la interfaz correspondiente a cada una de las aplicaciones con el objetivo de construir una base para el desarrollo del prototipo.

La aplicación del cliente se centra en un diseño atractivo y manejable, de forma que el usuario pueda acceder a todas las funcionalidades ofrecidas en pocas transiciones. De esta manera, se consigue una aplicación rápida e intuitiva.

En la *Imagen IV* se muestra la primera página de la aplicación, donde el cliente deberá introducir sus datos de acceso.



Imagen IV: Página de autenticación

Por el contrario, si el usuario no tiene aún una cuenta en la aplicación, podrá obtenerla de manera sencilla a través de un formulario de registro (*Imagen V*).

El diagrama muestra una interfaz de usuario en un teléfono móvil. En la parte superior, hay una barra de navegación con un icono de retroceso a la izquierda y el título 'Nuevo registro'. Debajo de la barra, hay tres campos de entrada de texto etiquetados 'Nombre', 'DNI' y 'Contraseña'. Debajo de estos campos, hay un botón gris con el texto 'Acceder'.

Imagen V: Página de registro de clientes

Al tratarse de una aplicación puramente informativa, se ha pretendido que la información mostrada al usuario sea clara y concisa, satisfaciendo así las necesidades del usuario de una forma rápida y completa. Por ello, en la ventana principal (*Imagen VI*) se listan aquellos pedidos que están pendientes de entrega, mostrando el id del paquete y su remitente. Además, en la esquina superior derecha se puede ver un icono con el que el cliente podrá buscar un paquete de manera más directa. Por otro lado, dispondrá de un botón para acceder a un mapa en el que podrá ver en donde se encuentra su pedido cuando esté en ruta.



Imagen VI: Página principal de la aplicación

Al seleccionar cualquier paquete de la lista mostrada en la página principal, el cliente accederá a otra página (Imagen VII) con información más detallada de dicho paquete.



Imagen VII: Página de información del paquete

Seleccionando el icono del mapa en la página principal, el usuario accederá al mapa (Imagen VIII) de la localización de su pedido en tiempo real, una vez esté a punto de serle entregado.

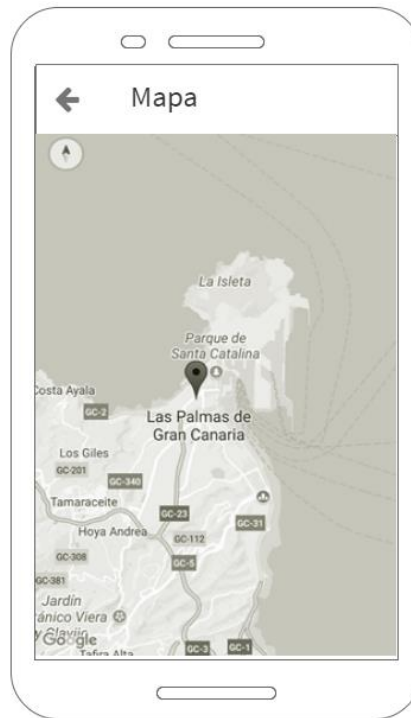


Imagen VIII: Página del mapa

Por último, mediante el menú lateral, el usuario podrá acceder a un historial de sus pedidos ya entregados de los que también podrá acceder a más información accediendo a ellos. En la *Imagen IX* se observa una tabla con los pedidos ordenados por fecha de entrega y una barra de búsqueda donde se podrán filtrar los paquetes por diferentes campos.



Imagen IX: Página del historial

En cuanto a la aplicación del repartidor, se optó por un diseño que mostrase los pedidos asignados y los que están seleccionados para el reparto en la ventana principal. El resto de páginas serían accesibles a través de un menú lateral. Reducir el número de transiciones entre páginas es muy importante para obtener una aplicación que se pueda operar con rapidez, razón por la que la página principal concentra la mayor parte de la zona de trabajo de la gestión del reparto de paquetes, siendo incluso posible controlar la mayor parte de la aplicación con una sola transición entre páginas. Como se puede ver en los diseños a continuación, el objetivo es una aplicación que presente una interfaz capaz de mejorar la productividad del repartidor.



Imagen X: Página de autenticación

En la *Imagen X* se observa la página de autenticación del repartidor, donde deberá introducir sus datos de acceso. Una vez se accede a la aplicación, se mostrará la página principal, que, por defecto, se situará en la pestaña “Asignados”, tal y como se aprecia en la *Imagen XI*.

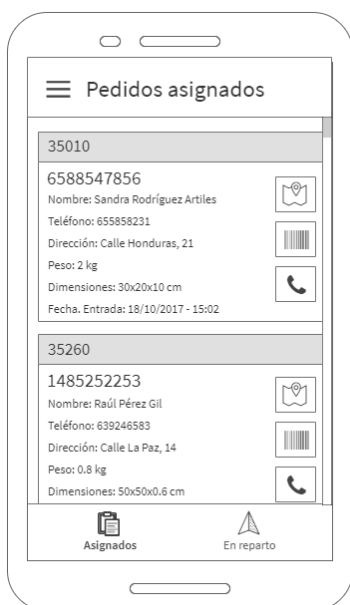


Imagen XI: Página principal

Si el repartidor selecciona la pestaña “En reparto” (*Imagen XII*), podrá observar una lista de aquellos paquetes que han sido elegidos para el reparto, ordenados por el código postal del destinatario, al igual que en la anterior pestaña.

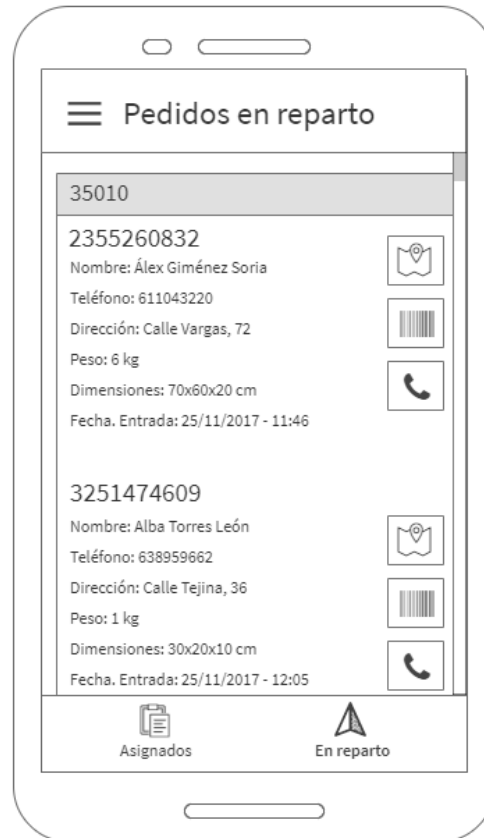


Imagen XII: Página principal, pestaña “En reparto”

En ambas pestañas se ofrecerá por cada paquete un botón con el icono de un mapa, un botón para realizar una llamada al destinatario y un botón de escaneo. El primer botón mencionado, mostrará al repartidor un mapa (*Imagen XIII*) con la ubicación donde deberá ser entregado dicho paquete. El siguiente botón que aparece escaneará un paquete y redireccionará al repartidor a la página de escaneo (*Imagen XIV*), donde tendrá ciertas funcionalidades dependiendo de en qué pestaña se encuentre el paquete escaneado.



Imagen XIII: Página del mapa del paquete

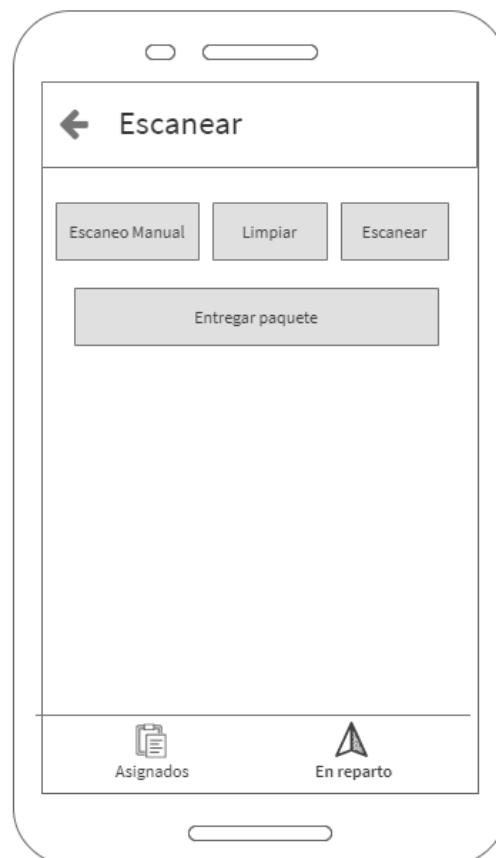


Imagen XIV: Página de escaneo de paquete

Cuando un paquete ha sido escaneado y se ha pulsado el botón “Entregar paquete”, el cliente deberá firmar para la confirmación de la entrega de su pedido. En la *Imagen XV* se observa el panel de firma donde el cliente podrá realizar la firma y finalizar la entrega.

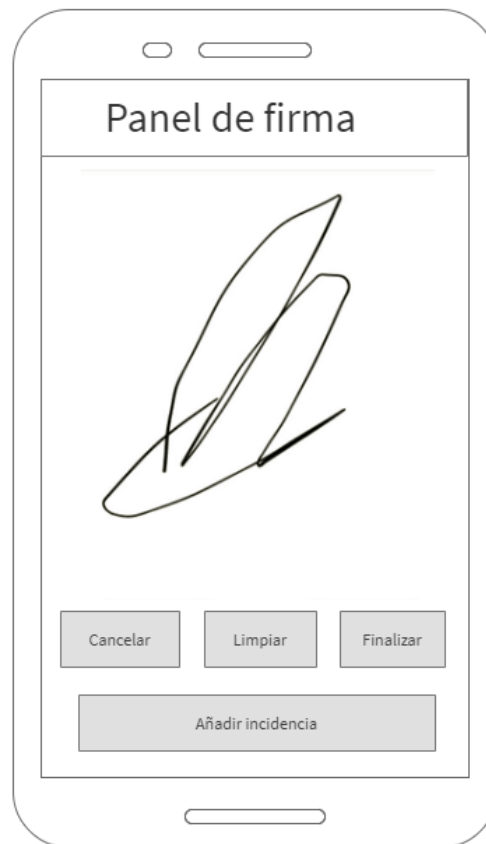


Imagen XV: Panel de firma

Accediendo al menú lateral, el repartidor podrá dirigirse al historial de pedidos entregados (*Imagen XVI*), donde podrá ver una lista de todos los paquetes que ha entregado ordenados por la fecha de la entrega, y a un mapa donde se mostrará la ubicación de entrega de todos los pedidos que le han sido asignados (*Imagen XVII*).



Imagen XVI: Página del historial



Imagen XVII: Mapa de pedidos asignados

A medida que se iban implementando todos los detalles de ambas aplicaciones especificados en estos bocetos, surgía la necesidad de hacer cambios en la estructura de la interfaz de ciertas páginas, tanto por la aparición de problemas como por la de nuevas ideas. Por ello, el resultado final difiere un poco con respecto a la idea inicial de los prototipos.

En el caso de la aplicación del repartidor se ha empleado una paleta de colores basado en el blanco, empleado en los fondos de las páginas, el rojo, empleado en las cabeceras de las páginas, y el azul para diversos elementos puntuales. El propósito es obtener una interfaz visualmente atractiva y sencilla ya que debe usarse durante una jornada completa de trabajo, por lo que un tono cálido como el rojo en conjunto con el blanco es perfecto para evitar el cansancio de la vista de los trabajadores que empleen las aplicaciones. El color azul se ha empleado para destacar secciones o resaltar algún elemento como separadores de elementos.

Por otro lado, la aplicación del cliente sustituye el color rojo de las cabeceras y otros elementos por el azul, con el objetivo de transmitir confianza, seriedad y profesionalidad al cliente, además de ofrecerle una interfaz atractiva y cómoda de manejar.

Para los textos se ha usado el color negro sobre fondo blanco en ambos prototipos, o texto blanco sobre fondo rojo o azul (como sucede en las cabeceras de secciones), con un tamaño de letra escalable a la longitud de la pantalla. Se ha empleado la negrita para resaltar los títulos de campos informativos de forma que los usuarios que empleen la aplicación encontrarán con facilidad los elementos que estén buscando.

4.5. Modelo de la base de datos

La base de datos empleada en este proyecto es la que ofrece *Firebase*, propiedad de *Google*. Cuenta con tres tablas diferentes, las cuales contienen toda la información necesaria de los diferentes repartidores, pedidos y vehículos. Para este proyecto sólo han sido necesarias dos de esas tablas: la de repartidores y la de pedidos.

Repartidores

- **UID:** Este campo es usado como identificador del repartidor.
- **coche:** Recoge la matrícula del vehículo asignado a ese repartidor.
- **disponibilidad:** Indica si un repartidor está en ruta o se encuentra en el almacén.
- **dni:** DNI del repartidor.
- **horaCapturaGPS:** Este campo almacena el momento en el que se obtuvo la última posición del repartidor.
- **imagen:** Es utilizado para almacenar la URL de la imagen, almacenada en *Firebase*.
- **latitud:** Recoge la coordenada de latitud de la posición de un repartidor. Útil para la geolocalización.
- **longitud:** Recoge la coordenada de longitud de la posición de un repartidor. Útil para la geolocalización.

- **nombre:** Nombre completo del repartidor.
- **numPedidos:** Número de paquetes asignados a ese repartidor.
- **sigDireccion:** Es la siguiente dirección de entrega a la que irá un repartidor.

Pedidos

- **codigoPostal:** Código Postal del destino del paquete.
- **destinatario:** Nombre del destinatario.
- **dimensiones:** Dimensiones del paquete en centímetros.
- **dirección:** Domicilio donde se debe entregar el paquete.
- **dni:** DNI del usuario que recibirá el paquete.
- **estado:** Indica si un paquete está asignado, en el almacén, entregado, en ruta, etc.
- **fechaEntradaAlmacen:** La fecha en la que ha llegado el paquete al almacén.
- **fechaEntrega:** Fecha en la que se ha entregado un paquete.
- **fechaEstimadaEntrega:** Fecha en la que espera entregarse el paquete.
- **firma:** Firma del cliente una vez haya recibido el paquete en su domicilio.
- **idPaquete:** Número de identificación de paquete.
- **idRepartidor:** Número de identificación del repartidor al que ha sido asignado el paquete.
- **latitud:** Coordenada de latitud de la dirección donde se debe entregar.
- **longitud:** Coordenada de longitud de la dirección donde se debe entregar.
- **nombre:** Nombre que puede asignar el cliente a su paquete para identificarlo mejor.
- **observaciones:** Sirve para indicar si un paquete tiene alguna consideración especial o algún error.
- **peso:** Peso del paquete.
- **remitente:** Entidad que desea enviar el paquete.
- **repartidor:** Nombre del repartidor al que ha sido asociado el paquete.
- **telf:** Teléfono de contacto del cliente que recibe el paquete.
- **urgente:** Sirve para indicar si un cliente ha solicitado envío urgente.
- **valoracionEntrega:** Campo dedicado a la nota que otorgue el cliente sobre el proceso de entrega de un paquete.

La base de datos que proporciona *Firestore* es conocida como *Realtime Database*, que nos ofrece una base de datos *NoSQL* alojada en la nube. Los datos se almacenan en formato *JSON* y se sincronizan con todos los clientes en tiempo real, permitiendo realizar operaciones básicas como subida de nuevos elementos, consulta de elementos, actualización de los existentes o borrado de los mismos. En la *Imagen XVIII* e *Imagen XIX* se pueden ver la estructura de las dos tablas empleadas.



Imagen XVIII: Tabla "repartidores"



Imagen XIX: Tabla "pedidos"

4.6. Alcance de la implementación

Debido a la ambición de este proyecto, pese a realizarse en conjunto, el número de horas necesarias para completarlo supera con creces las 300 horas máximas que establece el reglamento. Por ello, se ha optado por construir dos prototipos de las aplicaciones que tendrán la mayor parte de las funcionalidades descritas anteriormente, pero con algunas limitaciones.

Antes de comenzar describiendo las funcionalidades que se han omitido en el proyecto, hay que destacar que la implementación del servidor de almacenamiento y base de datos se ha realizado en la nube, aprovechando el servicio *Firebase* de *Google*. Esto permite disponer de la información en tiempo real en cualquier localización, a una velocidad bastante aceptable y con una tasa de transferencia adecuada para un proyecto de estas características. Es de suponer que una empresa que quisiera emplear las aplicaciones que forman este proyecto recurriría a un plan de datos con *Firebase* de mayor volumen o pondría en funcionamiento un servidor dedicado.

La principal funcionalidad que no se ha implementado es la de la comunicación entre administrador y repartidores, debido a las exigencias en la seguridad de las comunicaciones. A fin de proteger las conversaciones mantenidas, los mensajes deberán cifrarse de extremo a extremo, sin posibilidad de ser descifrados en su paso por el servidor. Este cifrado requiere una implementación tan elaborada en el software del emisor y el receptor que hace inviable su desarrollo en un proyecto con las limitaciones de infraestructura, tiempo y recursos humanos actuales.

Otra funcionalidad que no se ha implementado ha sido la de valoración por parte del cliente de paquetes recibidos. Esto es debido a que era prioritario lanzar un prototipo funcional antes de añadir más funciones, por lo que dicha función se descartó para el prototipo inicial para evitar superar el tiempo disponible y de entrega.

Entre las funcionalidades implementadas con limitaciones se encuentra la del servicio de registro para clientes. Este servicio podría mejorarse al añadir un sistema de recuperación de contraseñas en caso de perderla o la posibilidad de editar los datos personales de registro, pero esta operación requiere activar un módulo determinado del servidor para habilitar el envío de correos electrónicos a clientes, además de modificar la aplicación para que sea capaz de recibir la actualización, confirmaciones por parte del correo electrónico, etc. Lo antes mencionado supone un elevado esfuerzo y tiempo dedicado a las operaciones necesarias como para implementarlo en un prototipo.

Finalmente, si bien la aplicación se actualiza correctamente con los cambios producidos en la base de datos, completamente en tiempo real, encontramos la limitación de no disponer de notificaciones tanto para repartidores como clientes. El desarrollo de un sistema de notificaciones es complejo y, del mismo modo que la anterior funcionalidad limitada, es necesario un cambio en el servidor para permitir a este enviar notificaciones a los usuarios cuando se produzca algún cambio en los datos que tienen asociados. Debido a estos hechos, se optó por amplificar el sistema de actualización de datos en tiempo real, por el cual tanto repartidores como clientes recibirán cualquier cambio en un tiempo aproximado de milisegundos desde que se actualice en la base de datos, además de incorporar la posibilidad de actualizar ciertas páginas de las aplicaciones para recargar los datos existentes y solicitar nuevos datos si fuera necesario a la base de datos.

El resto de funcionalidades se han implementado completamente, si bien hay ligeras variaciones respecto al objetivo deseado, los requisitos se han cumplido satisfactoriamente.

4.7. Requisitos hardware mínimos

La versatilidad que ofrece *Ionic* garantiza su funcionamiento en cualquier dispositivo de tipo PC, pero es recomendable emplear un equipo más potente para sacar partido a esta tecnología. El equipo utilizado cuenta con un procesador Intel i7 a 2.60 GHz; 8 GB de memoria RAM; Windows 10 Pro y un disco duro de estado sólido (SSD) donde se ejecuta el código para obtener un mayor rendimiento a la hora de realizar las pruebas pertinentes.

4.8. Herramientas de desarrollo

La elección de las herramientas que se emplearon en el proyecto requirió un análisis previo de las tecnologías disponibles. En particular, *Ionic Framework* comenzaba a ser más común en el desarrollo de aplicaciones para dispositivos móviles (aunque sin demasiado éxito), pero su elección estaba condicionada al increíble crecimiento que ha experimentado *Angular 4*, tal y como se demuestra en la *Imagen XX*. En la misma imagen, encontramos que *Typescript*, ligado también a *Ionic*, está en crecimiento. A nivel profesional es importante considerar el aprendizaje obtenido durante el desarrollo del proyecto, puesto que el auge de las tecnologías móviles y por ello las herramientas de desarrollo son determinantes a la hora de contratar profesionales formados.

En un nivel más práctico, centrado en el desarrollo del proyecto y como se ha comentado en otros puntos de este informe, hemos elegido *Ionic Framework* por su versatilidad y capacidad para desarrollar un mismo código para navegadores, *Android* o *iOS*. La versión escogida, *Ionic 3*, nos permitía contar con las últimas actualizaciones, por lo que disfrutamos de una mejora en el tiempo de ejecución, mejoras de estabilidad y nuevas funciones como nuevas librerías nativas (entre las que encontramos el escáner o los mapas) o la más reciente función que permite realizar pruebas en diversos dispositivos enlazados con una simple orden en la consola. Por otra parte, *Visual Studio Code* mejora mucho nuestra productividad, gracias a los módulos que pueden añadirse para incrementar las funciones y agilizar el trabajo, que en conjunto con *Lodash*, esta mejora era evidente.

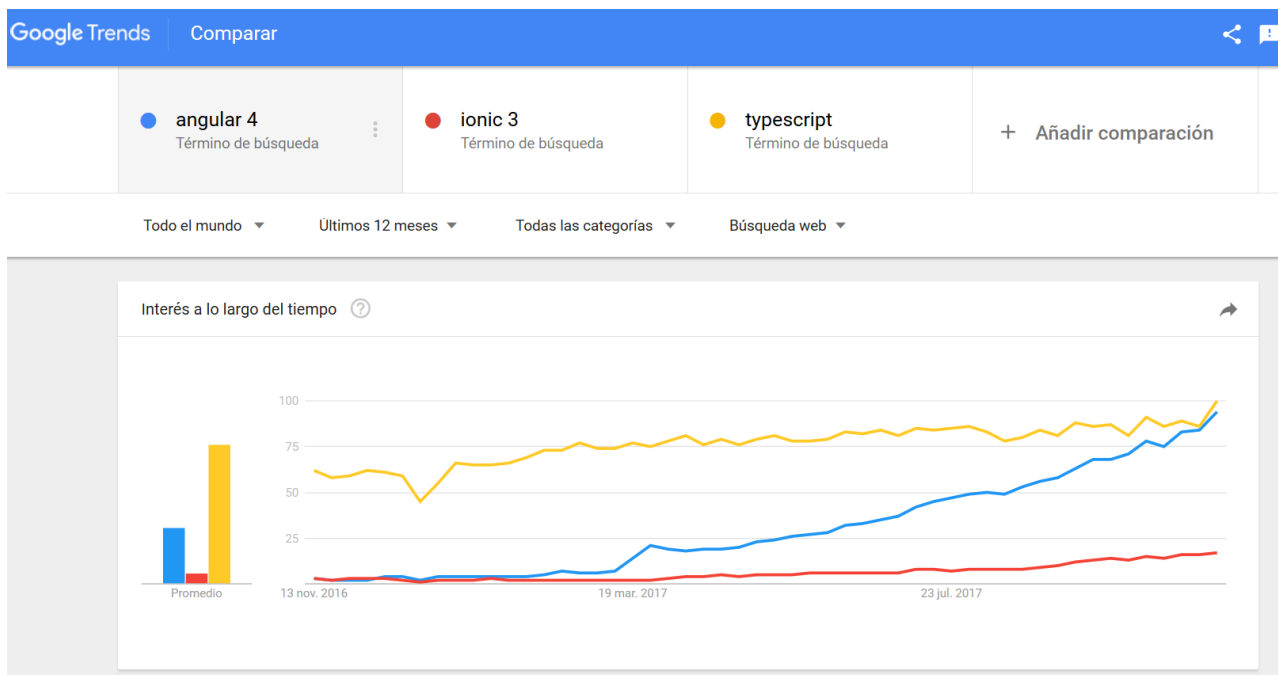


Imagen XX: Gráfico de la búsqueda de las tecnologías empleadas en Google

Las herramientas y librerías que se emplearán son:

Icono	Nombre	Descripción	Versión
	Ionic	Esta tecnología es un <i>framework</i> de código abierto empleado en el desarrollo de aplicaciones híbridas móviles. Creado sobre <i>AngularJS</i> y <i>Apache Cordova</i> , <i>Ionic</i> proporciona las herramientas y servicios necesarios para, mediante el uso de <i>HTML</i> y <i>CSS3</i> , desarrollar aplicaciones para móviles sin emplear el lenguaje nativo de cada plataforma.	3.7.1
	Angular	Es un <i>framework</i> de código abierto desarrollado por Google y es utilizado para crear y mantener aplicaciones web. Su principal objetivo es aumentar las aplicaciones basadas en navegador con capacidad de “Modelo Vista Controlador”. Se emplea en combinación con un entorno de ejecución como <i>NodeJS</i> .	4.4.5
TypeScript	TypeScript	Se trata de un lenguaje de programación de código abierto desarrollado por Microsoft. Es un conjunto de <i>JavaScript</i> , pero añadiendo tipado estático y objetos basados en clases.	2.5.3
	HTML5 y CSS3	En primer lugar, tenemos <i>HTML5</i> , un lenguaje de marcado que se encuentra en su quinta revisión. Este lenguaje es empleado en la “World Wide Web”. A continuación, tenemos <i>CSS3</i> , una hoja de estilo en cascada que se encuentra en su tercera revisión, siendo empleado para establecer el diseño visual de las páginas web e interfaces de usuario.	HTML5 + CSS3
	Cordova	<i>Apache Cordova</i> es un marco de desarrollo móvil de código abierto. Permite utilizar las tecnologías estándar web como <i>HTML5</i> , <i>CSS3</i> y <i>JavaScript</i> para desarrollo multiplataforma, evitando el lenguaje de desarrollo nativo de cada plataforma móvil.	7.0.1
	NodeJS	Se trata de un entorno de ejecución para <i>JavaScript</i> construido con el motor de <i>JavaScript V8</i> de <i>Chrome</i> . <i>NodeJS</i> usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. El ecosistema de paquetes de <i>Node.js</i> , “ <i>npm</i> ”, supone el conjunto más grande de librerías de código abierto en el mundo.	7.7.0
	Visual Studio Code	Editor de código empleado para la realización del proyecto, destaca por su sencillez de uso, personalización y capacidad de integrar extensiones que aumenten sus funcionalidades.	1.17.2
	Lodash	Empleado para facilitar el uso de <i>JavaScript</i> , emplea métodos para iterar listas, objetos y cadenas de texto, además de manipular valores y crear funciones complejas.	4.17.4

	Typings	Puesto que JavaScript por sí mismo no emplea tipos de datos, es necesario crear ficheros que describan los tipos empleados en nuestro código que pertenezcan a librerías externas. <i>Typings</i> se encarga de instalar los ficheros descriptivos necesarios de dichas librerías.	2.1.1
	AngularFire2	Proporciona una forma de comunicación entre <i>Angular 4</i> y <i>Firebase</i> . La creación de objetos observables, la sincronización de objetos en tiempo real o la autenticación de usuarios son sus grandes virtudes.	2.0.0-beta.8
	Google Maps	Esta librería nativa de <i>Ionic</i> ofrece una total integración de los componentes de <i>Google Maps</i> en <i>Angular 4</i> .	4.3.1
	Moments	La librería <i>Moment.json</i> permite realizar operaciones con fechas y horas en <i>JavaScript</i> .	2.18.1
	Background Geolocation	Este módulo nos proporciona la capacidad de obtener en tiempo real la localización del repartidor aunque su aplicación no se encuentre en ejecución (en primer plano) en ese momento.	4.3.0

4.9. Diseño e implementación

4.9.1. Arquitectura del software

La creación de un proyecto en *Ionic 3* genera un sistema de carpetas y ficheros, organizados en una estructura de árbol, que permite la organización del código. A continuación, se puede observar la utilidad de cada uno de estos elementos:

- **node_modules/** - En este directorio se encuentran todos los módulos necesarios para que *Ionic* funcione correctamente. La instalación del proyecto incluye estos módulos. A medida que se vayan instalando módulos, se irán almacenando en esta carpeta.
- **Platforms/** - Este directorio contiene las plataformas para las que se ha construido el proyecto. En sus respectivas carpetas se almacena una versión del proyecto actual para cada plataforma. En este caso, se encuentra la versión para *Android*.
- **plugins/** - Almacena los *plugins* que se han ido instalando a lo largo del desarrollo del proyecto.
- **resources/** - Contiene el icono de la aplicación, la imagen de carga y los iconos relativos a las versiones de *Android* e *iOS*.
- **src/** - En este directorio se encuentran las páginas de la aplicación, así como el estilo y el control de los módulos que la integran.
 - **app/** - Contiene los ficheros principales del proyecto, como el fichero que maneja la página principal de la aplicación y el menú lateral o el fichero que gestiona los módulos que se incluirán a la hora de ejecutar la aplicación.

- **assets/** - Este directorio es empleado para almacenar elementos por deseo del desarrollador, como las imágenes que desee insertar en la aplicación. En este caso, se ha utilizado para incluir iconos.
- **pages/** - Aquí se encuentran los directorios que contienen las páginas que forman la aplicación. Cada carpeta contiene un fichero con extensión “html”, otro con extensión “ts” y en ocasiones uno con “scss”.
 - Aplicación del cliente:
 - **home/** - El directorio contiene la página principal de la aplicación.
 - **login/** - Almacena la página de inicio de sesión.
 - **map/** - Este directorio contiene la página que muestra el mapa con la localización del repartidor cuando va a entregar el paquete.
 - **order/** - Contiene la página de información de los paquetes.
 - **registry/** - Se almacena la página del historial de la aplicación.
 - **signup/** - Es el directorio que contiene la página de registro de usuarios.
 - Aplicación del repartidor:
 - **delivery/** - Contiene la página de la pestaña “En reparto” de la página principal, donde se muestran todos los paquetes seleccionados para el reparto.
 - **drawpad/** - Almacena la página que gestiona el panel de firma.
 - **home/** - El directorio contiene la página principal de la aplicación, dividida en dos pestañas, “Asignados” y “En reparto”.
 - **login/** - Almacena la página de inicio de sesión.
 - **map/** - Este directorio contiene la página que muestra el mapa con la ubicación de destino de los paquetes.
 - **orderAssigned/** - Contiene la página de la pestaña “Asignados” de la página principal de la aplicación, donde se muestran los paquetes asignados a un repartidor.
 - **registry/** - Se almacena la página del historial de la aplicación.
 - **scan/** - El directorio contiene la página de escaneo de paquetes.
 - **pages.ts** - Este fichero contiene un índice de las clases creadas en cada página, lo que permite que, en caso de ser necesario, ser importado en una página que quiera trabajar con alguna otra página, tal como navegar hacia esa página. Está presente en ambas aplicaciones.
- **providers/** - Almacena el fichero “authdata.ts” (“auth-data.ts” en el caso del repartidor), que contiene la configuración de control de la acción de inicio y cierre de sesión. Además, en la aplicación del repartidor se encuentra también el directorio **location-tracker/** que contiene el fichero que se encarga de la localización en segundo plano.
- **theme/** - En este directorio se encuentra el fichero “variables.scss”, que contiene valores introducidos por el desarrollador sobre el estilo de la aplicación.
- **typings/** - Este directorio almacena los ficheros importados por la librería “Typings” de los módulos que requieren un tipo no propio de *TypeScript*.

- **www/** - Contiene el código necesario para la simulación de la aplicación en el navegador, sin necesidad de construir el proyecto.
- **.gitignore** – El fichero que contiene los archivos que no estarán controlados por *Git*.
- **config.xml** - Este fichero alberga información sobre la aplicación, tal como el nombre de la misma, descripción, iconos de la aplicación (para su visualización en un *Smartphone*), preferencias de uso y *plugins* a importar.
- **package.json** - El fichero contiene el núcleo de *Ionic*, albergando las dependencias necesarias para ejecutar el proyecto, las librerías instaladas y los *plugins* añadidos.

4.9.2. Desarrollo del prototipo

Después de analizar de forma exhaustiva el código de la aplicación, se han encontrado ciertas funcionalidades que resaltan con respecto a las demás debido a su importancia. Es el caso de la implementación del mapa, por ejemplo. De las numerosas formas existentes de implementar un mapa en una aplicación, se ha optado por recurrir a la librería de *Google Maps* para emplear la interfaz más sencilla, rápida y de menor código posible. De manera que, sólo es preciso inicializar el mapa en fichero *TypeScript*, con los datos correspondientes. Como se muestra en la *Imagen XX* a continuación (correspondiente al código de la aplicación del repartidor), al mapa se le añade una agrupación de marcadores a partir de la línea 50, pero el proceso para añadir un único marcador recurre a una sintaxis similar.

```

30  loadMap(){
31
32      this.mapElement = document.getElementById('map2');
33
34      var mapOptions2: GoogleMapOptions = {
35          camera: {
36              target: {
37                  lat: 27.9422467,
38                  lng: -15.598526
39              },
40              zoom: 10,
41              tilt: 10
42          }
43      };
44
45      this.map2 = this.googleMaps.create(this.mapElement, mapOptions2);
46
47      this.map2.one(GoogleMapsEvent.MAP_READY).then(() => {
48          console.log('Mapa listo!');
49
50          this.map2.addMarkerCluster({
51              boundsDraw: true,
52              markers: this.data,
53              icons: [
54                  {min: 2, max: 5, url: 'assets/marks/m1.png', anchor: {x: 16, y: 16}},
55                  {min: 5, max: 10, url: 'assets/marks/m2.png', anchor: {x: 16, y: 16}},
56                  {min: 10, max: 20, url: 'assets/marks/m3.png', anchor: {x: 24, y: 24}},
57                  {min: 20, url: 'assets/marks/m4.png', anchor: {x: 24, y: 24}}
58              ]
59          });
60      });
61  }

```

Imagen XXI: Código de inicialización del mapa

Otro aspecto que llama la atención es la utilización de la función “`ionViewDidLoad()`”, en la cual se extraen los datos de *Firebase* para luego ser mostrados. El propósito esencial de este método es ejecutarse varias veces, a diferencia del constructor que sólo se ejecuta una vez. Como los datos de *Firebase* varían, es necesario que se pueda ver esos cambios en la página. Usando este método, no es preciso refrescar la página en busca de actualizaciones, ya que se realizan automáticamente. Por ello, es empleado con frecuencia en las páginas de la aplicación. En la *Imagen XXI* se muestra un ejemplo de la función usada en la aplicación del cliente, para obtener y mostrar los datos de los pedidos asignados a él y que no hayan sido entregados.

```
home.page.ts x
36   ionViewDidLoad(){
37
38       let loader = this.loadingController.create({
39           content: 'Obteniendo datos...',
40           spinner: 'bubbles'
41       });
42
43       loader.present().then(() => {
44           this.angularFire.database.list('/pedidos').subscribe(data =>{
45               this.userData = _.chain(data)
46                   .filter(a => a.dni === this.userDni && a.estado !== "Entregado")
47                   .orderBy('fechaEntradaAlmacen')
48                   .value();
49
50               this.allOrders = this.userData;
51               loader.dismiss();
52           });
53       });
54   }
55 }
56 }
```

Imagen XXII: Función “ionViewDidLoad()” en la página principal del cliente

A continuación encontramos un fragmento de código correspondiente a la aplicación del repartidor, que se corresponde con el método para iniciar el proceso de geolocalización del repartidor mientras la aplicación se encuentra en segundo plano. Este método obtiene la posición geográfica del dispositivo en base a la velocidad actual. A mayor velocidad, el intervalo de cálculo de una nueva posición aumentará debido a que se recorre una mayor distancia en el tiempo, por lo que se calculará con más frecuencia la posición geográfica actual. Por otra parte, al disminuir la velocidad se reducirá el número de cálculos de la posición actual con el objetivo de optimizar el gasto de batería, puesto que al recorrerse menor distancia en el mismo tiempo, la necesidad de obtener la posición geográfica disminuye porque el margen de actualización es bastante escaso.


```
TS location-tracker.ts x
35 startTracking(){
36     // Background Tracking
37
38     let config = {
39         desiredAccuracy: 0,
40         stationaryRadius: 20,
41         distanceFilter: 10,
42         stopOnTerminate: false,
43         interval: 2000
44     };
45
46     this.backgroundGeolocation.configure(config).subscribe((location) => {
47
48         console.log('BackgroundGeolocation: ' + location.latitude + ', ' + location.longitude);
49
50         // Run update inside of Angular's zone
51         this.zone.run(() => {
52             this.lat = location.latitude;
53             this.lng = location.longitude;
54             this.horaCaptura = moment().format();
55             this.horaCapturaObserver.next(true);
56         });
57     }, (err) => {
58
59         console.log(err);
60     });
61
62     // Turn ON the background-geolocation system.
63     this.backgroundGeolocation.start();
64 }
65
66
67
68
```

Imagen XXII: Función que permite obtener la posición del repartidor pese a tener la aplicación cerrada

Por último, cabe destacar el control de la autenticación por parte de la librería *AngularFire* sobre la base de datos *Firebase*. *AngularFire* se encarga de comprobar si el usuario introducido está registrado en la parte de Autenticación de *Firebase* y si la contraseña coincide con el correo electrónico. De esta manera, el desarrollador no tiene que hacer demasiadas comprobaciones, simplemente envía los datos recogidos en el formulario para iniciar sesión y los envía a las funciones de *AngularFire*. En la imagen XXIII se observa el código de control para la autenticación del cliente, pero en el caso del repartidor es extremadamente similar.

```
TS authdata.ts x
11 constructor(public angularFire: AngularFire) {
12     angularFire.auth.subscribe(user => {
13         if (user){
14             this.userAuth = user.auth;
15         }
16     })
17 }
18
19 getCurrentUid(){
20     return this.angularFire.auth.getAuth().auth.uid;
21 }
22
23 getCurrentDni(){
24     let dni = this.angularFire.auth.getAuth().auth.email;
25     return dni;
26 }
27
28 loginUser(dni: string, pass: string): firebase.Promise<any> {
29     let string = dni + "@cliente.es"
30     return this.angularFire.auth.login({
31         email: string,
32         password: pass
33     });
34 }
35
36 logoutUser(): firebase.Promise<any> {
37     return this.angularFire.auth.logout();
38 }
39
40 }
```

Imagen XXIII: Código TypeScript de control de autenticación

4.9.3. Pruebas

Se ha realizado una batería de pruebas para comprobar el correcto funcionamiento de cada una de las aplicaciones. A medida que se han ido implementando las diferentes funcionalidades, se ha ido corroborando que éstas funcionasen bien. No obstante, una vez acabada por completo cada aplicación, se han repetido todas las pruebas con el objetivo de confirmar el perfecto funcionamiento de ambos prototipos. En la *Tabla 1* se muestran los resultados de las diferentes fases de prueba en la aplicación del cliente.

Funcionalidades	Primera fase	Segunda fase
Iniciar sesión en la aplicación	Correcto	Correcto
Registro de usuarios	Correcto	Correcto
Buscador en página principal	Correcto	Correcto
Panel lateral	Correcto	Correcto
Acceder a la página de información de paquetes	Correcto	Correcto
Añadir título a un paquete	Correcto	Correcto
Buscador en historial	Correcto	Correcto
Acceder al mapa de localización del repartidor	Correcto	Correcto
Cerrar sesión en la aplicación	Correcto	Correcto

Tabla 3: Resultado de las pruebas realizadas en la aplicación del cliente

Asimismo, en la *Tabla 2* se pueden observar los resultados obtenidos de la batería de pruebas realizada en la aplicación del repartidor.

Funcionalidades	Primera fase	Segunda fase
Iniciar sesión en la aplicación	Correcto	Correcto
Seleccionar siguiente pedido en entrega	Correcto	Correcto
Mapa de ubicación de destino de un paquete	Correcto	Correcto
Llamar al destinatario	Correcto	Correcto
Escanear paquetes	Correcto	Correcto
Panel lateral	Correcto	Correcto
Añadir un paquete al reparto	Correcto	Correcto
Gestión del escaneo de pedidos	Correcto	Correcto
Gestión del panel de firma	Correcto	Correcto
Añadir incidencia a un paquete	Correcto	Correcto
Buscador en historial	Correcto	Correcto
Acceder al mapa de pedidos pendientes	Correcto	Correcto
Cerrar sesión en la aplicación	Correcto	Correcto

Tabla 4: Dispositivos con los que se ha mostrado la aplicación

Por otro lado, se han mostrado ambas aplicaciones en diferentes pantallas para comprobar su diseño adaptable. En la *Tabla 3* se observan los diferentes dispositivos en los que ha sido mostrada:

Dispositivo	Pantalla	Resultado
Samsung Galaxy S5	5.1"	Correcto
Nexus 5X	5.2"	Correcto
Nexus 6P	5.7"	Correcto
Iphone 5	4"	Correcto
Iphone 6	4.7"	Correcto
Iphone 6 plus	5.5"	Correcto
Ipad	9.7"	Correcto
Ipad Pro	10.5"	Correcto

Tabla 3: Dispositivos con los que se han mostrado las aplicaciones

4.9.4. Acceso al código

Repositorio de *GitHub* donde se encuentra el código de la aplicación del cliente:

<https://github.com/Khondax/tft-i2-cliente>

Repositorio de *GitHub* donde se encuentra el código de la aplicación del repartidor:

<https://github.com/Khondax/tft-i2-repartidor>

5. Aspectos económicos y temporales

Las tres aplicaciones del proyecto han sido creadas con vistas a ser empleadas por una posible empresa de transporte de paquetería, de forma que dispongan de un grupo de aplicaciones para agilizar el trabajo del gestor del almacén, aumentar la transparencia con la que trabaja el repartidor y, finalmente, informar debidamente al receptor de un paquete.

Para poner en marcha este proyecto en un entorno realista, es necesario en primer lugar realizar un análisis exhaustivo de los requisitos de cada aplicación, su diseño y posterior codificación. Sin embargo, debido a nuestra falta de experiencia laboral y a que durante el desarrollo de nuestra carrera universitaria no hemos obtenido conocimientos al respecto, nos resultaría imposible estimar el número de horas necesarias para el desarrollo de la aplicación, así como tampoco podemos deducir el coste que conllevaría.

Sabemos que será necesario un servidor para almacenar la información que generen y controlen las aplicaciones. Actualmente, el prototipo está usando la versión gratuita de *Firebase*. No obstante, esto no sería suficiente en el caso de una aplicación totalmente funcional y en uso comercial. Sería preciso pues, usar una versión de pago de *Firebase*. Accediendo a su página oficial vemos que *Firebase* ofrece dos planes de pago: El plan *Flame* y el plan *Blaze*. Comparando ambos, observamos que el plan *Flame* es la opción más económica, pero podría resultar escaso a empresas de reparto de gran tamaño, mientras que el plan *Blaze*, con un precio más elevado, es totalmente personalizable y ofrece copias de seguridad automáticas.

Como es bien conocido, en el negocio de la informática el concepto de vender nuestra aplicación a una empresa y desvincularnos de ella va desapareciendo con el tiempo. A día de hoy lo habitual es ofrecer un servicio a varias empresas y encargarnos del mantenimiento de éste. Sin tener en cuenta la posible personalización que solicite la empresa contratante sobre la aplicación, la cual sólo se abonaría una vez, las empresas que quieran hacer uso de la aplicación estarán sujetas por contrato a una cuota mensual que incluirá el mantenimiento y futuras actualizaciones. El precio de esa cuota no se ha podido calcular por la misma razón expuesta anteriormente.

Tras lo anterior, hay que destacar también el valor añadido que proporciona la transparencia y acercamiento al público, gracias al uso de una aplicación para clientes diseñada para mostrar el máximo de información posible de una forma agradable e intuitiva, además de implicarlo en el proceso de reparto.

6. Normativa y legislación

Este proyecto se ha desarrollado con especial atención en la fiabilidad, seguridad y calidad de ella, conforme a los principios de desarrollo ético del software y la legislación vigente.

Es preciso tener en consideración una serie de medidas legales debido a que se manejarán datos de carácter personal. El proyecto debe cumplir con el Reglamento de Seguridad del Real Decreto 994/1999 de la Ley Orgánica de Protección de Datos (LOPD). Dicha ley tiene como objetivo garantizar y proteger, en lo concerniente al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, atendiendo especialmente a su honor e intimidad personal y familiar.

Dicha ley será aplicada a los datos de carácter personal registrados sobre cualquier tipo de soporte (informático o no). Están excluidos datos recogidos para uso doméstico y material clasificado por el estado que tratan sobre delincuencia organizada o terrorismo y a toda modalidad de uso posterior de estos datos por los sectores público y privado. Si la LOPD no estuviera en vigor, nuestros datos estarían en manos de la especulación.

La LOPD cataloga los ficheros según la naturaleza de los datos que contienen. Acorde con ello, se establecen diferentes niveles de seguridad que garanticen en mayor o menor medida la confidencialidad e integridad de dicha información. Los niveles en los que puede catalogarse un fichero son los siguientes:

- **Nivel alto:** Se considerarán de este nivel aquellos ficheros que contengan datos relativos a la ideología, afiliación social, religión, creencias, origen racial, salud o vida sexual. También forman parte de este nivel los ficheros recabados con fines policiales sin consentimiento de las personas afectadas o aquellos derivados de actos de violencia de género.
- **Nivel medio:** Dentro de este nivel se encuentran aquellos ficheros que contengan relativos a la comisión de infracciones administrativas o penales; aquellos cuyo funcionamiento se rija por el artículo 29 de la Ley Orgánica 15/1999 del 13 de diciembre; prestación de servicios de administraciones tributarias, relaciones con entidades financieras, gestorías o servicios comunes de seguridad social. Así mismo, formarán parte de este nivel aquellos ficheros que

sean responsables las mutuas de accidentes de trabajo y enfermedades profesionales de la Seguridad Social; aquellos que contengan un conjunto de datos de carácter personal que ofrezcan una definición de las características o de la personalidad de los ciudadanos y que permitan evaluar aspectos de la personalidad o del comportamiento de los mismos. Finalmente, este nivel incluye aquellos datos que permitan la localización de una persona.

- **Nivel básico:** Las medidas de seguridad que se han de adoptar en los ficheros que contengan datos de carácter personal están en función de la tipología de tales datos. Al operar de forma “acumulativa”, todos los ficheros o tratamientos de datos de carácter personal deberán adoptar las medidas de seguridad calificadas de nivel básico.

Este proyecto hace uso de varios datos de carácter personal de los repartidores de la empresa (como su ubicación exacta) y de los clientes (dirección, teléfono, nombre, DNI, etc.). Por ello, se cuenta con el fichero de datos de los repartidores, que requiere una protección de nivel medio, y el fichero de datos de los clientes, el cual requiere una protección de nivel básico.

A fin de cumplir la normativa vigente, los clientes tendrán a su disposición una serie de condiciones para las cuales se solicita expresamente su conformidad, relativas a la manipulación de sus datos y la debida protección por parte de la empresa.

7. Integración con la parte 2

En esta sección destacaremos la importante cohesión de las aplicaciones desarrolladas, tanto en este proyecto como en la parte 2, llevada a cabo por la estudiante María Guerra Suárez. La aplicación del administrador, a grandes rasgos, es la encargada de asignar pedidos a repartidores, que serán visibles por los mismos en su aplicación. Durante todo el proceso, el cliente observará los cambios que se suceden sobre su paquete, desde su llegada al almacén hasta su entrega.

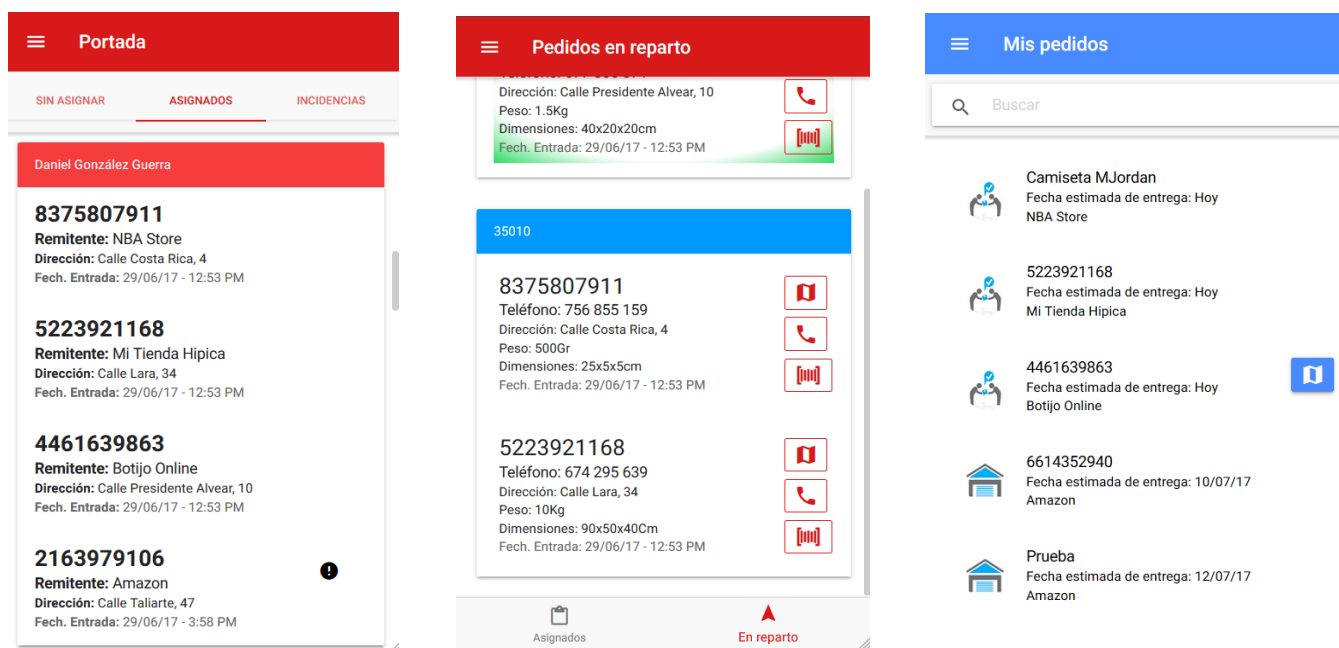


Imagen XXIV: Muestra de la interconexión entre aplicaciones

Como se puede observar en la *Imagen XXV*, un mismo pedido como es el que tiene el código 5223921168 se encuentra en las tres aplicaciones. El administrador, en la parte izquierda de la imagen, observa los pedidos asignados a repartidores, en concreto, el repartidor Daniel González Guerra. Dicho repartidor al acceder a su aplicación, representado en la parte central de la imagen, observa los pedidos que le han sido asignados y se encuentran en reparto. Finalmente, el cliente, representado en la sección derecha de la imagen, observa los pedidos que ha realizado, disponiendo de un icono distinto en función del estado en ese momento.

El mejor ejemplo de la interconexión entre las aplicaciones es el proceso completo de entrega de un pedido, desde la entrada en el almacén hasta la recepción del pedido. Este desarrollo comienza con la llegada del pedido 152876545258 al almacén, como se muestra en la *Imagen XXVI*.

← 152876545258

Información del destino	Información del paquete
Destino: Miguel Gonzalez Villa DNI: 44734470E Dirección: Avenida Mesa y Lopez 60, 8C Teléfono: 654321987	Remitente: PCComponentes Fecha de Entrada al Almacén: 13/11/17 11:45 AM Peso: 2.7kg Dimensiones: 20x20x10cm Repartidor: Sin asignar



Nombre: Daniel González Guerra
Número de paquetes asignados: 4
Disponibilidad: En ruta
Vehículo asignado: 1542-JPX

Imagen XXV: Captura del pedido recién llegado al almacén.

Posteriormente, el administrador asigna el pedido al repartidor Daniel González Guerra y dicho repartidor, en la *Imagen XXVII* observa dicho pedido en su aplicación y procede a su reparto al seleccionarlo tal y como se muestra en la *Imagen XXVIII*.

15:53 Pedidos asignados

35009

2163979106

Nombre: César Ortega Alonso
Teléfono: 651189138
Dirección: Calle Taliarte, 47
Peso: 7Kg
Dimensiones: 40x30x20cm
Fech. Entrada: 29/06/17 - 3:58 PM

35010

152876545258

Nombre: Miguel Gonzalez Villa
Teléfono: 654321987
Dirección: Avenida Mesa y Lopez 60, 8C
Peso: 2.7kg
Dimensiones: 20x20x10cm
Fech. Entrada: 13/11/17 - 11:45 AM

Asignados En reparto

15:54 Pedidos en reparto

35006

4461639863

Teléfono: 677 806 371
Dirección: Calle Presidente Alvear, 10
Peso: 1.5Kg
Dimensiones: 40x20x20cm
Fech. Entrada: 29/06/17 - 12:53 PM

35010

152876545258

Teléfono: 654321987
Dirección: Avenida Mesa y Lopez 60, 8C
Peso: 2.7kg
Dimensiones: 20x20x10cm
Fech. Entrada: 13/11/17 - 11:45 AM

8275807011

Asignados En reparto

Imagen XXVII: Captura del pedido marcado como siguiente en entrega.

Imagen XXVI: Captura del pedido en la aplicación del repartidor.

Para finalizar la entrega, el repartidor presenta al cliente el panel para firmar la recepción del pedido, tras lo cual finaliza su reparto, como se muestra en la *Imagen XXIX*. El administrador, como se muestra en la *Imagen XXXI*, puede observar el pedido entregado y la firma del cliente.

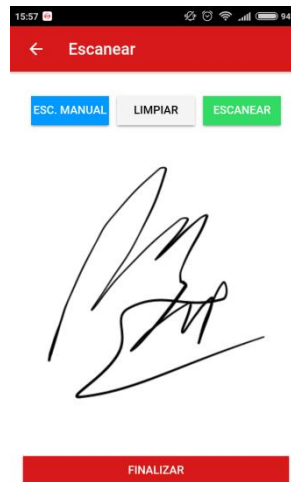
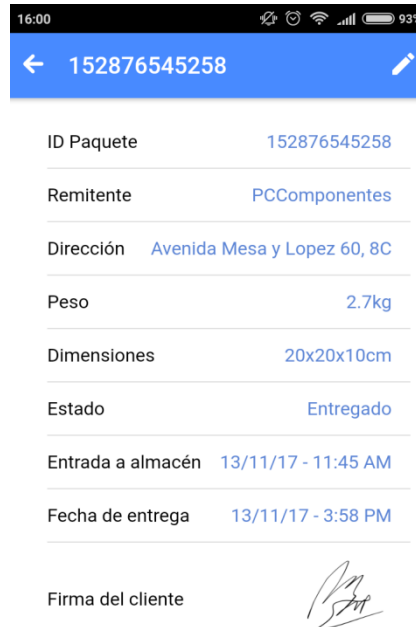


Imagen XXVIII: Captura de la firma de la recepción del pedido.

← 152876545258	
Información del destino Destino: Miguel Gonzalez Villa DNI: 44734470E Dirección: Avenida Mesa y Lopez 60, 8C Teléfono: 654321987	Información del paquete  Remitente: PCCComponentes Fecha de Entrada al Almacén: 13/11/17 11:45 AM Peso: 2.7kg Dimensiones: 20x20x10cm Repartidor: Daniel González Guerra
Información de la entrega Lugar de Entrega: Avenida Mesa y Lopez 60, 8C Fecha de Entrega: 13/11/17 3:58 PM Observaciones:	
	Firma del Cliente: 

Imagen XXIX: Captura del pedido finalizado en la aplicación del administrador.

Durante este proceso el cliente ha observado los cambios de estado del pedido, y en el historial de pedidos recibidos encuentra el pedido entregado y su recepción firmada, como se muestra en la *Imagen XXXII*.



ID Paquete	152876545258
Remitente	PCComponentes
Dirección	Avenida Mesa y Lopez 60, 8C
Peso	2.7kg
Dimensiones	20x20x10cm
Estado	Entregado
Entrada a almacén	13/11/17 - 11:45 AM
Fecha de entrega	13/11/17 - 3:58 PM
Firma del cliente	

Imagen XXX: Captura del pedido finalizado en la aplicación del cliente.

8. Conclusiones y trabajos futuros

Este trabajo se aproxima a su fin, y con ello, el continuo trabajo dedicado muestra sus frutos finalmente. Los tres prototipos (pero altamente funcionales) de las aplicaciones creadas cuentan con una gran calidad y avanzado desarrollo gracias a que la amplia mayoría de funcionalidades han sido implementadas. Por ello, podemos considerar superados los objetivos propuestos, así como es un hecho que se trata de un importante punto de partida en el campo de la programación de aplicaciones móviles.

Los comienzos suelen ser complicados, y un proyecto como este no iba a ser menos. Se han puesto a prueba nuestros conocimientos en desarrollo web, pero también nos hemos enfrentado a nuevos lenguajes de programación y tecnologías cuya vida en el mercado las convierten en desconocidas para la mayoría de desarrolladores, más aun para dos estudiantes de ingeniería informática sin documentación en sus manos, en ocasiones irrelevante (este proyecto comenzó como una idea mientras existía *Ionic 1* y ha finalizado a poco tiempo del lanzamiento de *Ionic 4*). Pese a ello, *Ionic* cuenta con una comunidad de programadores muy activa y cada vez estas herramientas adquieren más éxito.

Pese al duro trabajo y un inicio que requirió un par de cursos online sobre *Angular* y *TypeScript*, cuando los prototipos comenzaron a cobrar vida los ánimos se relajaron, programar no era tan tedioso, desarrollar no era tan complicado como parecía y hasta comprendíamos el funcionamiento de la base de datos. Pasó a ser un pasatiempo, el tiempo libre se convertía en una nueva funcionalidad y aunque cada nuevo reto se afrontaba con una mirada de incredulidad al tutor, no dejábamos de avanzar. Por increíble que parezca, llegamos a divertirnos realizando el proyecto. Las pequeñas sutilezas que entraña el código, que lo hacen más vivo y funcional surgían por la simple inercia que supone trabajar con una motivación diferente, lo que nos permitía avanzar con paso firme. Después de todo, escoger una plataforma como *ionic* para desarrollar el proyecto fue una buena elección.

Como se menciona en varias ocasiones a lo largo de este informe, este proyecto ha sido dividido en dos trabajos distintos debido a la envergadura con la que se planteó en un comienzo. La primera parte consistía en el desarrollo de la aplicación del administrador, siendo esta segunda parte la correspondiente a los repartidores y clientes (pese a que los títulos de los trabajos no expresen lo mismo debido a un error a la hora de presentar la solicitud del proyecto), por lo que desde el comienzo se requería una comunicación constante para que la interconexión entre aplicaciones no se derrumbara. El trabajo en paralelo fue crucial para garantizar un análisis de requisitos adecuados, diseño inicial consensuado e implementaciones de prototipos que funcionaran, a su vez, en paralelo. Aun así, las negociaciones se tornaron agresivas al tratar la forma del árbol de almacenamiento de la base de datos, llegando a su punto álgido cuando se alcanzó el extremo de añadir campos en la base de datos mientras el otro alumno realizaba pruebas. Esta es una de las muchas anécdotas que se formaron durante el desarrollo del proyecto, lo que es una prueba perfectamente válida de que el proyecto finalizó como una colaboración satisfactoria en las ocasiones que se requería y con magníficos resultados en los prototipos finales. Consideramos la experiencia obtenida durante las fases de trabajo en paralelo de un valor incalculable y que jugará un papel muy importante en el futuro laboral.

El proyecto en sí puede tener un futuro potencialmente activo, partiendo de una reconstrucción total en términos de objetivos, funcionalidades nuevas y un diseño acorde al momento en el que se retome su desarrollo, ampliando el trabajo realizado hasta el punto de obtener una aplicación perfectamente compatible y funcional en el ámbito empresarial. Si bien no hay datos concretos, no es tan descabellado suponer que las aplicaciones pueden mejorar el rendimiento y fluidez en el entorno de trabajo de aquellos que las empleen, lo que se traduce en beneficios debido al incremento de la productividad. Si bien el proyecto se ha centrado en un tema concreto como puede ser la distribución de paquetería, gracias a la versatilidad de *ionic* es posible desarrollar aplicaciones específicas en múltiples campos donde sea necesaria una herramienta potente y precisa o simplemente una renovación del anterior sistema. Por ejemplo, la gestión de flotas de vehículos o la integración del gremio de taxistas a la actualidad tecnológica, pero también una aplicación de comunicación para la comunidad universitaria orientada a incentivar el trabajo en equipo, desarrollada para múltiples plataformas (empleando chat de texto y voz, envío de ficheros, integración de encuestas y con un control de usuarios eficaz). La propia universidad podría adoptar dicha aplicación y ponerla a disposición de estudiantes, docentes e investigadores, facilitando la comunicación entre estudiantes del mismo o distinto curso e incluso se podría llegar a incentivar la colaboración entre facultades, dando un salto de calidad a la educación actual e incluso repercutiendo favorablemente sobre proyectos de investigación.

Un enorme aspecto positivo del proyecto es la amplitud de herramientas empleadas, lo cual es ideal para un futuro a corto plazo porque no nos encamina en una dirección determinada, sino que mantiene muchas puertas abiertas. El desarrollo web, de aplicaciones para navegador o dispositivos móviles, de programación en *Angular* o *Typescript*, todos son posibles puntos de entrada en un mercado en auge. Mientras tanto, lo que era una posibilidad de desarrollar un proyecto por cuenta propia comienza a ser un hecho palpable, traduciéndose en un currículum más extenso.

Para finalizar, a nivel personal este proyecto ha repercutido positivamente en mi forma de trabajar, especialmente a la hora de enfrentar un problema y orquestar una solución creativa para un error que hubiera echado abajo el proyecto. Hubo errores de esa magnitud, errores que nos impedían escribir siquiera una línea de código, pero eso no evitó que siguiera motivado día tras día hasta resolver cada inconveniente. Es por ello que afirmo que desarrollar en Ionic ha sido ameno y entretenido, lo que ayudó enormemente a mantener las ganas de seguir avanzando continuamente, consiguiendo que la, en teoría, estricta y formal labor de desarrollar un proyecto de final de carrera se convirtiera en un actividad apasionante. Es posible que este largo trayecto haya superado ligeramente las 300 horas estipuladas, pero hay veces que no puedes resistirte a un rompecabezas sin importar el tiempo que necesites para completarlo. Como diría Franklin D. Roosevelt, “Siempre que te pregunten si puedes hacer un trabajo, contesta que sí y ponte enseguida a aprender cómo se hace”.

9. Fuentes de información

Listado de enlaces consultados durante la realización del proyecto.

- **Preparación de Ionic**
 - Instalación de NodeJS: <https://nodejs.org/es/>
 - Guía de instalación de Ionic:
<http://ionicframework.com/docs/guide/installation.html>
 - Documentación de Ionic: <https://ionicframework.com/docs/>
 - GitHub de Ionic: <https://github.com/ionic-team/ionic/blob/master/CHANGELOG.md>
 - Documentación de Visual Studio Code: <https://code.visualstudio.com/docs/?dv=win>
 - Documentación de Firebase: <https://firebase.google.com/docs/>

- **Librerías y herramientas destacadas empleadas**
 - GitHub de AngularFire2: <https://github.com/angular/angularfire2>
 - GitHub de Google Maps: <https://github.com/mapsplugin/cordova-plugin-googlemaps>
 - Documentación de Moment.js: <http://momentjs.com/docs/>
 - Utilidad para generar códigos QR: <http://www.qrcode.es/es/generador-qr-code/>
 - Geolocalización en segundo plano: <https://github.com/mauron85/cordova-plugin-background-geolocation>

- **Otros**
 - Curso de Ionic 2, Angular 2 y TypeScript:
<https://www.pluralsight.com/courses/ionic2-angular2-typescript-mobile-apps>
 - Precios Firebase:
<https://firebase.google.com/pricing/?hl=es-419>
 - Blog oficial de Ionic: <http://blog.ionic.io/>

10. Anexos

ANEXO I: Manual de usuario y software de la aplicación del cliente

Esta aplicación está pensada para mostrar a nuestros clientes información de sus pedidos de manera atractiva y sencilla de utilizar. A continuación, se presenta el manual de usuario de esta aplicación, detallando cada una de las funcionalidades que ofrece.

En la *Ilustración 1* vemos la primera página que aparece al iniciar la aplicación. Debemos introducir nuestro DNI y la clave para acceder a ella.

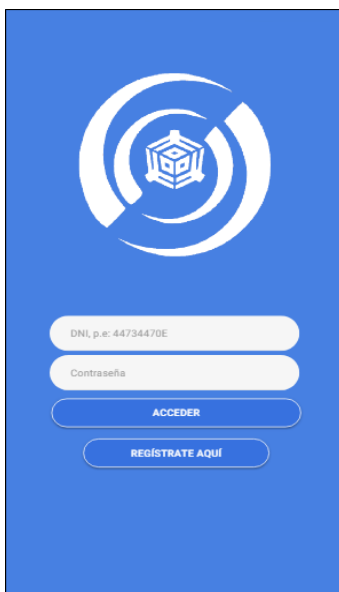


Ilustración 1: Página de acceso a la aplicación

En el caso de que aún no tengamos cuenta en la aplicación, podemos hacer clic en el botón “REGÍSTRATE AQUÍ” y rellenar el formulario de registro (*Ilustración 2*).

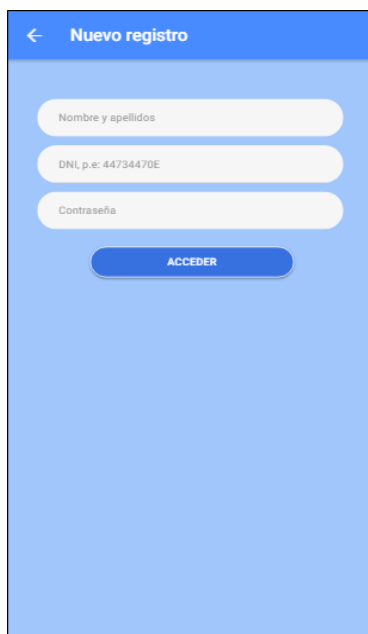


Ilustración 2: Página de registro

Una vez hemos accedido a la aplicación, nos encontramos con la página principal (*Ilustración 3*). En ella vemos una lista de los paquetes que tenemos pendientes de entrega. Observamos que hay dos tipos de iconos acompañando a la pequeña información que se muestra de cada paquete. El icono de la casita representa que el paquete se encuentra en el almacén y el otro icono nos dice que nuestro paquete está en reparto y se nos entregará ese mismo día.

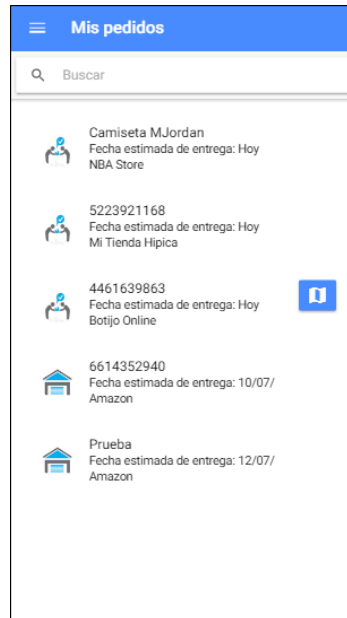


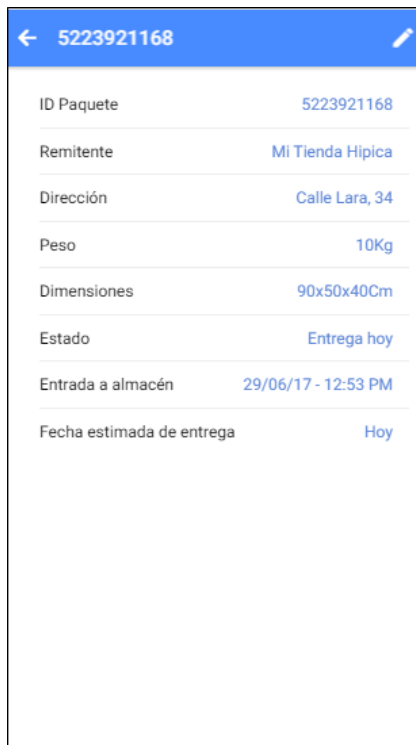
Ilustración 3: Página principal

La página principal ofrece una barra de búsqueda que nos permite filtrar nuestros paquetes pendientes por varios campos. Además, algunos paquetes pueden mostrar un botón con el icono de un mapa. Esto quiere decir que el paquete está próximo a ser entregado y, pulsando sobre él, podemos acceder a un mapa en el que veremos la localización en tiempo real del repartidor que nos trae el pedido (*Ilustración 4*).



Ilustración 4: Mapa

En la *Ilustración 5* vemos información más detallada de un pedido. Para acceder a esta página, debemos seleccionar uno de los paquetes de la lista de la página principal. En la esquina superior derecha nos encontramos un icono de un lápiz, el cual nos permite cambiar el título del paquete. Por ejemplo, cambiamos el ID “5223921168” por “Silla australiana Wintec Pro”. De esta manera, podremos identificar el paquete fácilmente.



← 5223921168	
ID Paquete	5223921168
Remitente	MI Tienda Hípica
Dirección	Calle Lara, 34
Peso	10Kg
Dimensiones	90x50x40Cm
Estado	Entrega hoy
Entrada a almacén	29/06/17 - 12:53 PM
Fecha estimada de entrega	Hoy

Ilustración 5: Página de información de paquetes

Accediendo al menú lateral de la aplicación podemos dirigirnos al historial de pedidos. En la *Ilustración 6* vemos la página del historial donde se muestra una lista de nuestros paquetes ya entregados, ordenados por la fecha de la entrega. Así mismo, también contamos con una barra de búsqueda y podemos acceder a más información de un paquete haciendo clic sobre él. Por último, podemos cerrar sesión en la aplicación accediendo al menú lateral.

← Pedidos entregados		
🔍 Buscar		
25/05/17		
ID del paquete	Remitente	Hora de entrega
521565218245	Ebay	10:32 AM
20/05/17		
ID del paquete	Remitente	Hora de entrega
152876545258	PCComponentes	3:18 PM

Ilustración 6: Página del historial

ANEXO II: Manual de usuario y software de la aplicación del repartidor

La aplicación del repartidor está orientada a proporcionar una mejora en la realización del trabajo de los repartidores de una empresa de transporte, ofreciendo una interfaz amigable, rápida y muy intuitiva.

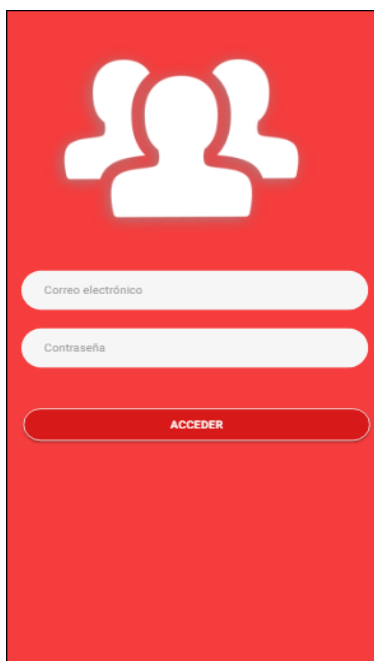


Ilustración 7: Página de acceso a la aplicación

En la *Ilustración 7*, vemos la página de acceso a la aplicación, donde los repartidores deben ingresar sus datos. Una vez el usuario está autenticado, es redirigido a la página principal de la aplicación. Esta página está dividida en dos pestañas. Por defecto, se muestra la pestaña “Asignados”, tal y como vemos en la *Ilustración 8*. En esta pestaña se muestran aquellos paquetes asignados al repartidor y todos los datos que pueden ser útiles para él. Por cada paquete, tenemos disponibles tres botones. El primer botón que aparece, , nos llevará a un mapa que muestra la dirección del destinatario. Con el segundo botón, , podemos realizar una llamada al destinatario del paquete y con el tercer botón, , podemos escanear los paquetes que queramos repartir. El icono con el símbolo de exclamación al principio del ID del paquete nos indica que ese pedido es urgente.

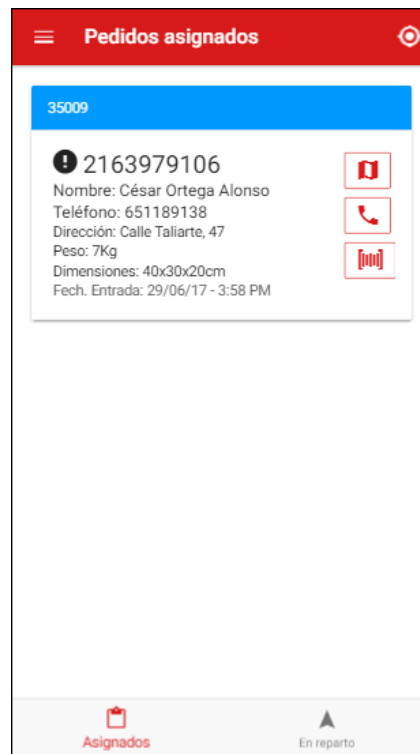


Ilustración 8: Página principal, pestaña “Asignados”

Si cambiamos de pestaña, podemos ver todos los paquetes elegidos para el reparto (*Ilustración 9*). Si seleccionamos sobre uno de ellos, éste cambiará de color indicando que es el siguiente paquete que será entregado.

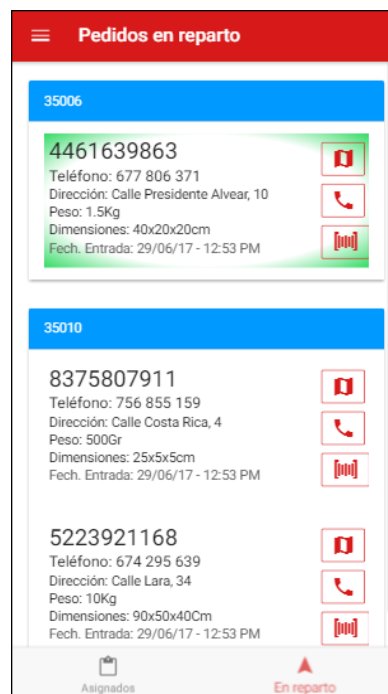


Ilustración 9: Página principal, pestaña “En reparto”

En esta página tenemos los mismos botones que en la anterior, con las mismas funcionalidades. En la *Ilustración 10* observamos el mapa al que accedemos mediante el primer botón mencionado anteriormente.



Ilustración 10: Mapa de paquetes

Cuando escaneamos un paquete, la aplicación nos redirecciona a la página que vemos en la *Ilustración 11*. En ella podemos introducir el código del paquete manualmente en caso de que el escaneo haya fallado, limpiar los datos o cancelar el escaneo. También aparece un botón que, según en la pestaña en la que estuviera el paquete escaneado, tendrá una funcionalidad u otra. En el caso de que el paquete estuviera en la pestaña asignados, el botón “Añadir paquete” que nos permitirá añadir dicho paquete al reparto. De esta manera, el paquete pasaría a estar en la pestaña “En reparto”.



Ilustración 11: Página de escaneo

Si, por el contrario, el paquete que hemos escaneado pertenecía a la pestaña “En reparto”, aparecerá el botón “Entregar Paquete” que nos permitirá entregar el paquete al destinatario. Una vez pulsado ese botón, la aplicación nos lleva a un panel donde el destinatario del pedido deberá firmar para finalizar la entrega con éxito (*Ilustración 12*). Esta página nos ofrece funcionalidades como limpiar la firma, cancelarla, añadir una incidencia si fuera necesario y finalizar la entrega.

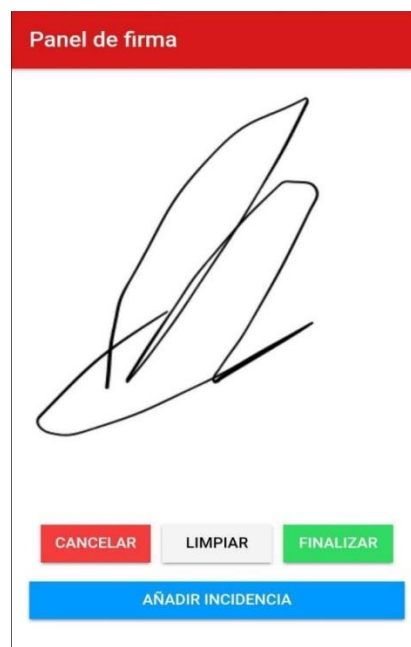
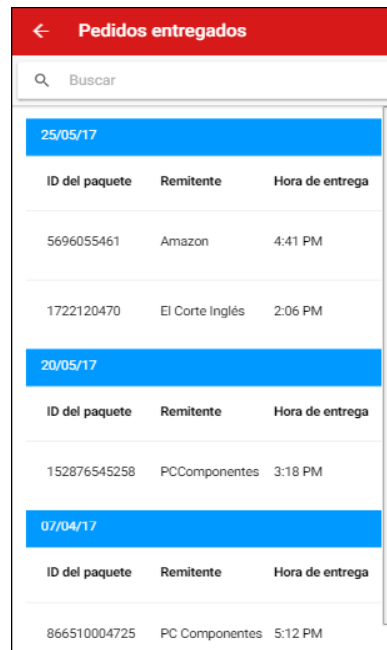


Ilustración 12: Panel de firma

Accediendo al menú lateral de la aplicación, podemos dirigirnos al historial de pedidos, donde encontraremos una lista de todos los pedidos entregados, ordenados por fecha de entrega

(Ilustración 13), además de una barra de búsqueda en la que podemos filtrar nuestros paquetes por diferentes campos.



25/05/17		
ID del paquete	Remitente	Hora de entrega
5696055461	Amazon	4:41 PM
1722120470	El Corte Inglés	2:06 PM

20/05/17		
ID del paquete	Remitente	Hora de entrega
152876545258	PCComponentes	3:18 PM

07/04/17		
ID del paquete	Remitente	Hora de entrega
866510004725	PC Componentes	5:12 PM

Ilustración 13: Página del historial

También en el menú lateral podemos encontrar la opción “Mapa de pedidos pendientes”, la cual nos llevará a un mapa donde se muestran todas las ubicaciones de destino de los paquetes asignados al repartidor (Ilustración 14). También podemos acceder a esta página mediante el botón de la parte superior derecha de la página principal, en la pestaña “Asignados”. Para terminar, podemos cerrar sesión mediante la opción “Cerrar Sesión” del menú lateral.

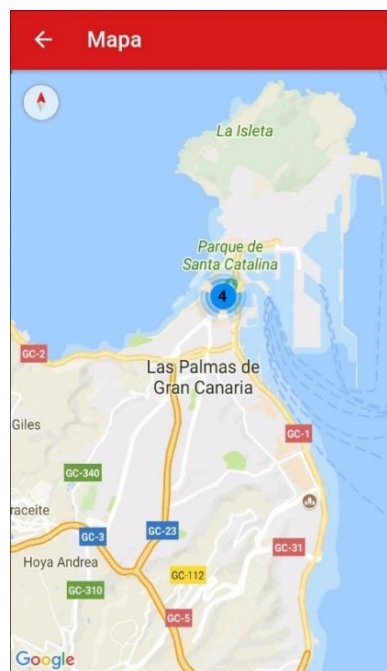


Ilustración 14: Mapa de paquetes asignados