Legislación Policial

Nombre Estudiante:
Alejandro Suárez Ocampo

Nombre Consultor/a: Eduard Martin Lineros
Profesor/a responsable de la asignatura: Carles Garrigues Olivella

13/12/2017

A) Creative Commons:

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial 3.0 España de Creative Commons

B) GNU Free Documentation License (GNU FDL)

Copyright © 2017 Alejandro Suárez Ocampo.

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

C) Copyright

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

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Legislación Policial*
Nombre del autor:	*Alejandro Suárez Ocampo*
Nombre del consultor/a:	*Eduard Martin Lineros*
Nombre del PRA:	*Carles Garrigues Olivella*
Fecha de entrega (mm/aaaa):	12/2017
Titulación:	Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles
Idioma del trabajo:	*Castellano*
Palabras clave	Seguridad, Emergencias, Legislación, Ordenanzas, Policía, Tráfico

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.*

La aplicación permite la consulta de legislación relacionada con la seguridad y las emergencias, haciendo uso de filtros de búsqueda para facilitar y agilizar el acceso a la información requerida.

También permite realizar un importante feedback con el desarrollador, de forma que cualquier usuario pueda reportar cualquier incidencia, duda o aclaración relacionada con cualquier artículo de cualquier ley o reglamento incluido en la misma.

Al hacer uso de firebase como base de datos, la actualización de los datos consultados podrá hacerse de forma transparente para el usuario, y sin necesidad de actualizar la aplicación, de forma que cualquier cambio en los datos será visible al momento.

Por otro lado, al desarrollar la aplicación haciendo uso del framework IONIC, permite que esta sea multiplataforma.

Abstract (in English, 250 words or less):

The app allows to search legislation related with security and emergencies by using search filters in order to make easier the access to the information required.

It also allows to make an important feedback with the developer in order to enable the users to report incidences, doubts or explanations related with any article of any law or regulations included in the legislation.

By making use of the firebase as a date base, the update of the information consulted can be made in a transparent way by the user, without the need to update the app. This means that any change in the data will be visible to the user straight away.

Finally, by developing the app making use of the framework IONIC allows it to be multiplatform.

Índice

1. Introducción **1**

1.1 Contexto y justificación del Trabajo **1**

1.2 Objetivos del Trabajo **3**

1.3 Enfoque y método seguido **4**

1.4 Planificación del Trabajo **5**

1.5 Breve sumario de productos obtenidos **5**

1.6 Breve descripción de los otros capítulos de la memoria **5**

1.6.1 Fundamentos teóricos: 6

1.6.2 El diseño: 6

1.6.3 La implementación 6

1.6.4 El testeo de la aplicación 7

2. Usuarios y contexto de uso **7**

2.1 Requisitos de usuario **7**

2.2 Contexto de uso **8**

3. Diseño conceptual **9**

3.1 Diagrama de casos de uso **10**

3.2 Diagrama de flujo del Usuario **10**

3.2 Diagrama de flujo del Administrador **13**

4. Prototipado **15**

5. Arquitectura de la aplicación **23**

5.1 Elementos de la vista **26**

5.2 Controladores **31**

5.3 Base de datos **33**

6. Desarrollo **35**

6.1 LawListPage **35**

6.2 ItemToShowPage **36**

6.3 CalendarComponent **36**

6.4 LawRowComponent **37**

6.5 TitleComponent **37**

6.6 PkAutoviaComponent **37**
6.7 PhoneBookRowComponent **38**
6.8 FirebaseProvider **38**
6.9 ModelProvider **39**
6.10 HideHeaderDirective **40**
6.11 Plugins externos **40**
6.11.1 Plugin Firebase 40
6.11.2 Plugin CallNumber 41
7. Pruebas **41**
8. Conclusiones **44**
9. Glosario **47**
10. Bibliografía **48**
11. Anexos **50**

1. Introducción

1.1 Contexto y justificación del Trabajo

En el año 2005 comienza mi carrera profesional con el empleo de policía en el ayuntamiento de Las Palmas de Gran Canaria, y desde ese momento he desempeñado las funciones propias de mi cargo. Durante todo este tiempo he observado las deficiencias tecnológicas que adolece el servicio de Policía Local.

Los más de 10 años de profesión, unido a la experiencia y conocimientos adquiridos a lo largo de mi carrera académica, me permite tener un enfoque ideal sobre las posibilidades de mejora para el sector de la seguridad y las emergencias.

Uno de los aspectos más destacados a mejorar es la consulta de la legislación aplicable a la tarea policial, ya que, la consulta de ésta es una de las acciones que con más frecuencia se repite en el día a día del desempeño profesional y que compromete la eficacia de la labor del agente debido a la lentitud en el acceso a esta información, incidiendo negativamente en la relación con el ciudadano, debido a que la resolución de la acción policial puede verse dilatada a la espera de la consulta de la legislación aplicable al incidente en cuestión.

En la actualidad, dicho problema se resuelve consultando manuales en papel o haciendo búsquedas en internet desde el teléfono móvil, con la correspondiente pérdida de tiempo que ello supone.

Actualmente, el Policía Local que realiza el servicio en la vía pública está dotado de la legislación de uso diario en formato papel. Dicha legislación es consultada por éste, si fuese necesario, para resolver la incidencia que se le presenta. Para ello el agente debe consultar durante varios minutos de una forma arcaica y cuanto menos tediosa que articulo debe aplicar según la ordenanza o ley que se entiende infringida. También hay que tener en cuenta un aspecto de suma importancia que puede conllevar a una errónea aplicación de la ley con un grave perjuicio al ciudadano, y no es otra que las actualizaciones de la legislación aplicable. Como todos conocemos, las leyes están sometidas a continuos cambios y con manuales en formato papel esto no es posible,

a no ser que estemos todo el día consultado las posibles modificaciones. Con una aplicación multiplataforma, es obvio que este tema estaría resuelto, ya que la legislación estaría permanentemente actualizada.

Otros de los aspectos importantes para un Agente es lo tedioso que resulta el tener que llevar carpetas en la mano con toda la legislación, manuales, etc. ya que, a la hora de intervenir, este material en vez de un recurso para la resolución de la incidencia se puede convertir en un inconveniente, ya que no todos los servicios a resolver son pacíficos.

También se observa que muchos policías utilizan sus propios Smartphone para la consulta de la legislación vía internet, con lo cual, parece que el cambio de mentalidad para modificar la dinámica de la consulta en papel a la consulta de un posible aplicativo utilizando la telefonía móvil no parece que sea un obstáculo a salvar.

Para poder afianzar la idea del desarrollo de una aplicación multiplataforma he medido los tiempos de respuesta haciendo uso de la actual forma de búsqueda de información, oscilando los mismos entre 7 y 10 minutos según la destreza del agente a la hora de buscar y el medio empleado. Con la aplicación, este tiempo se vería reducido a segundos, ya que solo se emplearía el tiempo que tarda la aplicación en cargar más el que tarde el agente en introducir la palabra a buscar.

Lo que se espera con esta aplicación es economizar el tiempo que emplea un agente a la hora de intervenir, reduciendo el tiempo que el ciudadano tiene que esperar, facilitando el acceso a una información en constante actualización y evolución gracias a la colaboración y el aporte de los usuarios.

Por otro lado, se ha realizado un estudio en los diferentes mercados de aplicaciones, observando que hay varias apps que ofrecen la búsqueda de legislación de forma muy parecida a la anteriormente relatada, pero dicha legislación se limita a leyes de ámbito nacional, y en ningún caso ofrecen la posibilidad de consultar legislación municipal, la cual es muy importante en el desempeño de la labor que realiza un Policía Local a diario.

Por tanto, queda justificado realizar un estudio de diseño, desarrollo e implantación de una aplicación multiplataforma, que mejorará el tiempo de resolución de las incidencias que deben ser atendidas por la

Policía Local por los motivos que arriba he indicado y que a continuación resumo:

1º) Mejora del tiempo de respuesta.
2º) Actualización de la legislación aplicable.
3º) Reduce carga física al agente al llevar menos documentación.
3º) Implantación relativamente sencilla.

Dicha aplicación cubrirá las siguientes necesidades:

1. Permitir la consulta de la legislación policial (reglamentos, ordenanzas, leyes...), con filtros de búsqueda que permita un rápido acceso a la información solicitada.
2. Poder enviar cualquier tipo de consulta o sugerencia, que será tratada debidamente por parte del administrador.
3. Posibilidad de añadir el calendario laboral adaptado a las peculiaridades de cada municipio.
4. Permitirá la consulta de una agenda telefónica orientada a la labor que desempeñan

En la aplicación se podrá encontrar, inicialmente, legislación relacionada en materia de tráfico y derecho penal, así como muchas de las ordenanzas del municipio de Las Palmas de Gran Canaria. No obstante, se podrán añadir todas aquellas que fuesen necesarias.

Para el desarrollo de la aplicación se emplearán los conocimientos adquiridos en el Master sobre el framework Ionic. Además, se hará uso de firebase como base de datos, lo que permite una rápida actualización de los datos de consulta sin necesidad de tener que actualizar la aplicación y de forma muy rápida, completamente transparente para el usuario.

1.2 Objetivos del Trabajo

El presente trabajo pretende cubrir objetivos tanto académicos como laborales, siendo los mismos los siguiente:

1. Crear una aplicación multidispositivo que permita la consulta de información por parte de los policías y cualquier ciudadano en general.

2. Recibir sugerencias y/o actualizaciones de la información contenida en la aplicación.
3. Asentar y mejorar los conocimientos adquiridos durante el Master, sobre todo en lo relacionado con el framework Ionic.
4. Introducir el uso de Firebase como base de datos de la aplicación.
5. Ser lo más accesible posible para que la información pueda ser consultada en cualquier momento y desde cualquier dispositivo.
6. Integrar los conocimientos adquiridos durante mi etapa académica junto a mi experiencia laboral para desarrollar una aplicación que mejore el día a día del trabajo de los miembros de las fuerzas y cuerpos de seguridad, y al alcance de cualquier ciudadano

1.3 Enfoque y método seguido

Para el desarrollo de este trabajo he decidido crear un producto nuevo, donde pueda estructurar desde el principio todo el proyecto, enfocándolo a posibles actualizaciones y empleando los conocimientos y metodologías adquiridos durante el Master.

Al tratarse de un producto creado desde el principio por mí, me va a permitir adaptarlo a las necesidades del sector al que va enfocado, pudiendo actualizarlo en cualquier momento para adaptarlo a dichas necesidades.

Respecto a la metodología a seguir, he realizado un pequeño análisis inicial, y teniendo en cuenta que los requisitos de la aplicación están completamente definidos y apenas van a cambiar, he decidido emplear un ciclo de vida clásico o en cascada.

En la descripción de las tareas se adjunta como anexo de esta memoria en un diagrama de Gantt, en un fichero llamado "*Planificación de Tareas - PEC 1.pdf*", puede verse como empiezo con la fase de requisitos, luego paso al diseño, sigo con la implementación y acabo con la verificación, los test. El mantenimiento de la aplicación, junto a posibles actualizaciones, quedan fuera del alcance de este proyecto. Dichos trabajos se llevarían a cabo como trabajo futuro si se decidiese publicar la aplicación.

1.4 Planificación del Trabajo

Para la realización de este trabajo se van a necesitar los siguientes recursos:

1. Ordenador personal con el correspondiente software para el desarrollo del trabajo:
    - Para programar usaré el IDE WebStorm [1].
    - Como repositorio del proyecto usaré Git [2] y SourceTree [3] para visualizar y administrar el repositorio mediante la sencilla interfaz de usuario, que alojaré en GitLab.
    - Para pruebas usaré mi teléfono personal, siendo el mismo un Huawei P8 (Android).
    - Para el diseño de las pantallas me apoyaré de la página web "*NinjaMock*" [4].
    - Para la creación de los distintos diagramas UML, usaré el StarUML [5].

Para poder cumplir con los distintos objetivos marcados en este proyecto se hace necesaria un plan de trabajo donde se especifiquen las tareas a realizar y su duración. Es por ello que se ha elaborado un diagrama de Gantt con las tareas a realizar, indicando su duración, fecha de inicio y fin prevista para cumplir con los objetivos marcados. Dicho diagrama se adjunta como anexo en un fichero llamado "*Planificación de Tareas - PEC 1.pdf*".

1.5 Breve sumario de productos obtenidos

Como resultado de este trabajo se obtendrá un proyecto realizado en Ionic, que podrá ser compilado para obtener una aplicación en cualquier plataforma que se desee.

1.6 Breve descripción de los otros capítulos de la memoria

Durante los siguientes capítulos se hablará con mayor profundidad de los siguientes temas:

1.6.1 Fundamentos teóricos:

a. Se explicarán los distintos aspectos relacionados con la aplicación que permitan comprender el alcance y necesidad de la misma.
b. Se identificarán las características, necesidades y objetivos de los usuarios.

1.6.2 El diseño:

Es la primera vez que desarrollo un proyecto basado en Ionic [6], ya que, hasta el momento, solo lo había realizado en Java o Android, por lo que debo adaptar el diseño del mismo a las características propias de dicho framework.

En esta etapa se realizará el diseño de la aplicación, definiendo la estructura de los datos. Hay que tener en cuenta que, al desarrollar la aplicación haciendo uso del framework Ionic, la estructura de la misma no se basa en clases y entidades como en la mayoría de los lenguajes de programación, sino en páginas, componentes y provider (o servicios). Para ello, se realizará el correspondiente diagrama que refleje cada una de las páginas utilizadas, los componentes y servicios que necesita, su utilidad y necesidad en la aplicación.

Para realizar el diseño de la aplicación se tendrán en cuenta los criterios de usabilidad que permitan al usuario no solo obtener la información que necesitan, sino hacerlo de una forma práctica y cómoda.

Por otro lado, para tener una idea de cómo quedará la aplicación una vez desarrollada, se realizarán los diseños de las distintas pantallas necesarias para la utilización de la aplicación, valorando la usabilidad y simplicidad de las mismas, explicando los distintos diseños planteados y la decisión del diseño final.

1.6.3 La implementación

Para llevar a cabo la implementación de la aplicación es muy importante contar con un buen entorno de trabajo. Para ello, usaré Git

como gestor de versiones, de forma que me permita ir guardando los cambios realizados, pudiendo volver atrás en cualquier momento que considere necesario.

Por otro lado, un buen IDE facilita la programación. Yo me he decidido a usar el WebStorm, aprovechando la licencia que me da el ser estudiante de la UOC. Gracias a él podré programar de forma rápida y eficiente, al mismo tiempo que me permite probar en tiempo real la evolución del proyecto.

1.6.4 El testeo de la aplicación

Para comprobar el correcto funcionamiento de la aplicación es imprescindible realizar los correspondientes test que validen los objetivos marcados al inicio del proyecto.

A medida que se van desarrollando nuevas funcionalidades, estas se irán probando en distintos dispositivos para verificar su correcto funcionamiento.

El contar con un IDE como WebStorm facilita bastante las cosas, ya que permite ver, en tiempo real, las nuevas funcionalidades o cambios realizados.

2. Usuarios y contexto de uso

2.1 Requisitos de usuario

En esta etapa nos centraremos en investigar los usuarios de la aplicación, recoger requisitos [7] que ayudarán a conocer los usuarios y definir perfiles.

Las personas entrevistadas son compañeros de trabajo, tanto de la Policía Local de Las Palmas de Gran Canaria, como de otros cuerpos policiales. Por otro lado, también entrevisté a personas ajenas a dichos cuerpos, mostrando bastante interés por la aplicación.

Se pueden distinguir dos tipos de requisitos atendiendo a su naturaleza. Por un lado, tenemos los requisitos funcionales, que son aquellos que describen lo que el sistema debe hacer. Por otro lado, los no funcionales, que describen las propiedades del propio sistema

A continuación, se enumeran los requisitos funcionales detectados:

- Los usuarios necesitan una consulta rápida, con filtros.
- Los usuarios necesitan buscar entre un amplio abanico de leyes y ordenanzas.
- Los usuarios necesitan localizar e identificar el punto kilométrico de una vía concreta en que se encuentran interviniendo.
- Los usuarios necesitan poder consultar el calendario laboral.
- Los usuarios necesitan consultar una agenda telefónica orientada a la labor que desempeñan.
- Los usuarios necesitan poder enviar sugerencias al administrador.

Respecto a los no funcionales, se han detectado los siguientes:

- Los usuarios deberán poder acceder a la aplicación en cualquier momento y desde cualquier dispositivo con conexión a internet.
- El sistema debe recoger las sugerencias enviadas por los usuarios para que puedan ser tratadas por el administrador.
- El sistema actualizará la información consultadas por los usuarios de forma transparente para ellos.
- El administrador gestionará la base de datos en Firebase.

2.2 Contexto de uso

En la actualidad, cuando un agente de la autoridad se encuentra interviniendo en la vía pública, no siempre tiene presente qué artículo de qué ley hay que aplicar a cada caso. Para ello, se recurren a manuales impresos, documentos guardados en teléfonos móviles o consultando directamente en internet.

El tiempo empleado en la consulta de dicha información hace que la persona con la que se está interviniendo tenga que esperar un tiempo que, con las herramientas adecuadas, sería mucho menor.

En la actualidad existen varias aplicaciones relacionadas con consultas de legislación, pero no siempre están disponibles para todas las plataformas y la legislación que incluye es estatal, en ningún caso municipal.

Son varios los escenarios donde puede hacerse uso de la aplicación.

Los agentes de la autoridad intervienen con bastante frecuencia con ciudadanos que comenten diversos tipos de infracciones, tanto en el ámbito relacionado con la seguridad vial como en materia penal o administrativa.

La aplicación puede ser usada no solo para intervenir con ciudadanos que comenten infracciones, sino como consulta para la realización de un informe policial, una diligencia en comisaria o simplemente a modo de consulta por cualquier persona interesada.

Es por ello que esta aplicación pretende agilizar este tipo de consultas.

3. Diseño conceptual

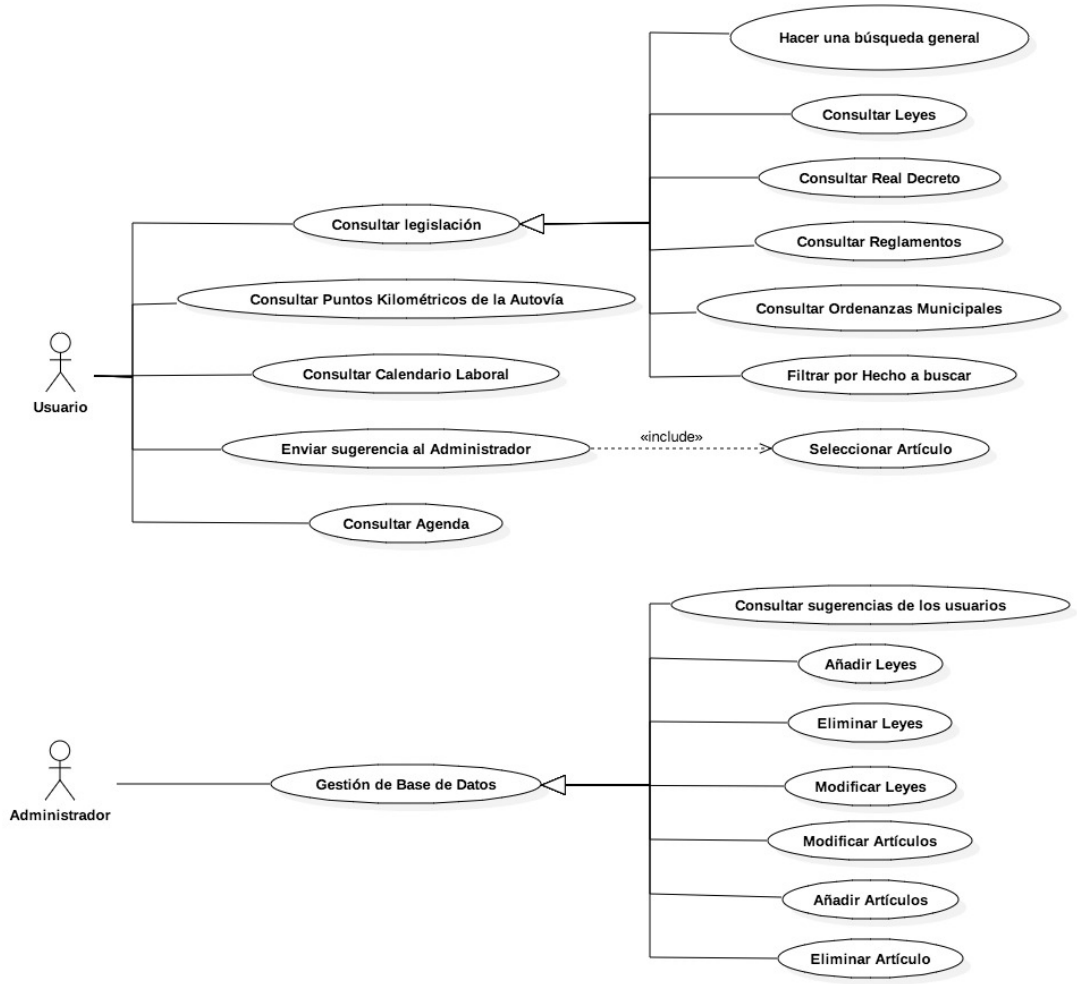
Para tener una idea global de lo que la aplicación puede hacer, y tras la fase de análisis de requisitos, se procede a la realización del correspondiente diagrama UML de casos de uso.

Por un lado, tenemos al *Usuario*, que podrá consultar toda la legislación contenida en la aplicación, así como otros datos de interés.

Por otro lado, tenemos al *Administrador*. Este actor no se verá reflejado en la aplicación móvil, ya que será el encargado de gestionar todo lo relacionado con la base de datos almacenada en Firebase y, por lo tanto, su trabajo se realizará desde "fuera" de la aplicación, pero su cometido es vital para el correcto funcionamiento de la misma.

En la siguiente imagen puede observarse los casos [8] de uso para cada actor.

3.1 Diagrama de casos de uso



3.2 Diagrama de flujo del Usuario

Caso de uso	Consultar Legislación
Descripción	Se podrá consultar los artículos de cualquier ley, real decreto y/o ordenanza contenida en la aplicación
Actores	Usuario
Precondiciones	Tener un smartphone con conexión a internet y la aplicación instalada
Flujo principal	
1	El usuario accede a la aplicación

2	El usuario accede a la ley de la que desea realizar la búsqueda
3	El usuario ve todos los artículos de la ley seleccionada, pudiendo introducir el hecho a buscar
Flujo secundario	
1	El usuario accede a la aplicación
2	El usuario accede a la búsqueda general
3	El usuario ve todos los artículos de la ley seleccionada, pudiendo introducir el hecho a buscar
Postcondiciones	El usuario accede a la información solicitada
Caso de uso	Consultar Puntos Kilométricos de la Autovía
Descripción	Se podrá consultar el punto kilométrico de la autovía en que estamos visualizando imágenes correspondientes a diferentes puntos de la citada vía
Actores	Usuario
Precondiciones	Tener un smartphone con conexión a internet y la aplicación instalada
Flujo principal	
1	El usuario accede a la aplicación
2	El usuario accede al listado de imágenes con los distintos puntos kilométrico de la vía
Postcondiciones	El usuario obtiene el punto kilométrico de donde se encuentra
Caso de uso	Consultar Calendario laboral
Descripción	Se podrá consultar el calendario laboral correspondiente a cada plantilla
Actores	Usuario
Precondiciones	Tener un smartphone con conexión a internet y la aplicación instalada
Flujo principal	
1	El usuario accede a la aplicación
2	El usuario accede al calendario laboral

Postcondiciones	El usuario obtiene la información del calendario laboral correspondiente
Caso de uso	Enviar sugerencia al administrador
Descripción	Se podrá enviar sugerencias sobre un determinado artículo al administrador
Actores	Usuario
Precondiciones	Tener un smartphone con conexión a internet y la aplicación instalada
Flujo principal	
1	El usuario accede a la aplicación
2	El usuario accede al artículo correspondiente
3	El usuario selecciona dicho artículo
4	El usuario introduce el texto a enviar y su correo electrónico
Postcondiciones	El usuario envía una sugerencia sobre un artículo concreto al administrador, adjuntando su correo por el posible feedback
Caso de uso	Consultar Agenda
Descripción	Se podrá consultar la agenda de la aplicación, desde la que se podrá realizar una llamada directamente
Actores	Usuario
Precondiciones	Tener un smartphone con conexión a internet y la aplicación instalada
Flujo principal	
1	El usuario accede a la aplicación
2	El usuario accede a la agenda
3	El usuario busca lo que desee, pudiendo llamar directamente si lo necesita
Postcondiciones	El usuario obtiene el nº de teléfono que estaba buscando, pudiendo llamar directamente si lo considera necesario

3.2 Diagrama de flujo del Administrador

Caso de uso	Consultar sugerencias de los usuarios
Descripción	Se podrá consultar las sugerencias enviadas por los usuarios
Actores	Administrador
Precondiciones	Tener un dispositivo con conexión a internet
Flujo principal	
1	El administrador accede a la base de datos
2	El administrador lee las sugerencias
Postcondiciones	El administrador tiene conocimiento de las sugerencias enviadas por los usuarios
Caso de uso	Añadir leyes
Descripción	Se podrán añadir nuevas leyes a la base de datos
Actores	Administrador
Precondiciones	Tener un dispositivo con conexión a internet
Flujo principal	
1	El administrador accede a la base de datos
2	El administrador inserta la ley deseada
Postcondiciones	La base de datos tiene una nueva ley para ser consultada por los usuarios
Caso de uso	Eliminar leyes
Descripción	Se podrán eliminar nuevas leyes a la base de datos
Actores	Administrador
Precondiciones	Tener un dispositivo con conexión a internet
Flujo principal	
1	El administrador accede a la base de datos
2	El administrador elimina la ley deseada

Postcondiciones	La base de datos ya no tiene la ley eliminada
Caso de uso	Modificar leyes
Descripción	Se podrán modificar leyes a la base de datos
Actores	Administrador
Precondiciones	Tener un dispositivo con conexión a internet
Flujo principal	
1	El administrador accede a la base de datos
2	El administrador accede a la ley deseada
3	El administrador modifica la citada ley
Postcondiciones	Se ha modificado una ley en la base de datos
Caso de uso	Añadir artículos
Descripción	Se podrán añadir nuevos artículos a las leyes contenidas en la base de datos
Actores	Administrador
Precondiciones	Tener un dispositivo con conexión a internet
Flujo principal	
1	El administrador accede a la base de datos
2	El administrador accede a la ley deseada
3	El administrador añade el articulo deseado
Postcondiciones	Se ha añadido un nuevo artículo a una ley
Caso de uso	Eliminar artículos
Descripción	Se podrán eliminar artículos a las leyes contenidas en la base de datos
Actores	Administrador
Precondiciones	Tener un dispositivo con conexión a internet
Flujo principal	
1	El administrador accede a la base de datos
2	El administrador accede a la ley deseada

3	El administrador elimina el articulo deseado
Postcondiciones	Se ha eliminado un artículo a una ley
Caso de uso	Modificar artículos
Descripción	Se podrán modificar los artículos de las leyes contenidas en la base de datos
Actores	Administrador
Precondiciones	Tener un dispositivo con conexión a internet
Flujo principal	
1	El administrador accede a la base de datos
2	El administrador accede a la ley deseada
3	El administrador modifica el artículo deseado
Postcondiciones	Se ha modificado un artículo a una ley

4. Prototipado

Tras la captura de requisitos y la creación de los distintos diagramas, he procedido a realizar el prototipo de cada una de las pantallas de la aplicación, intentando buscar la forma más rápida e intuitiva de buscar la información solicitada, sin descuidar los detalles visuales y atendiendo a criterios de usabilidad.

Tras una primera implementación, se ha dado a probar entre distintos potenciales usuarios, y tras recibir propuestas de mejora, se ha procedido a cambiar diferentes aspectos a los establecidos inicialmente, los cuales se explican a continuación:

1. **Colores**: Se ha decidido cambiar el blanco que se tenía inicialmente por diferentes tonalidades de azul (según se trate del título, subtítulo y fondo de la app) por su similitud a los colores del uniforme policial, resultando más atractivo visualmente.
2. **Grupo de botones para filtrar:** Se ha hecho uso del componente ***ion-segment*** proporcionado por ***Ionic*** para poder elegir el motivo por el que se desea filtrar un ítem en la búsqueda, pudiendo elegir

entre artículo, hecho, cuantía o puntos a detraer del permiso de conducir.

3. **<u>Cabecera oculta</u>:** se ha optado por esconder la cabecera de la aplicación cuando se hace scroll en la página que muestra los detalles de una. Esto permite un mejor aprovechamiento de la pantalla para poder mostrar mayor cantidad de información

A continuación, se muestra cada una de las pantallas de la aplicación, explicando su necesidad e importancia.

<u>Pantalla principal</u>: Da la bienvenida al usuario y muestra las principales opciones que tiene.

La primera opción nos envía hacia la pantalla de búsqueda general, desde donde podemos buscar cualquier artículo de las distintas leyes contenidas en la aplicación.

La segunda opción, hacia el listado de opciones disponibles para el citado municipio, entre los que se encuentra una agenda, un calendario laboral, el conjunto de ordenanzas municipales relacionadas con el citado municipio y los puntos kilométricos de la autovía.

Y la última opción, hacia un listado de leyes y decretos, y la tercera, al conjunto de ordenanzas pertenecientes al Ayuntamiento de Las Palmas de Gran Canaria.

Se ha decidido este diseño porque permite separar la legislación de un municipio concreto de la del resto, permitiendo además la adición de nuevos municipios de forma rápida, dinámica y transparente para el usuario.

<u>Búsqueda General</u>: permite al usuario realizar una búsqueda en la totalidad de la legislación contenida en la aplicación. De esta forma, aunque no se sepa exactamente en qué ley se encuentra lo que estamos buscando, se podrá buscar rápidamente.

Esta búsqueda es más general que la realizada dentro de una ley, ya que mostrará los resultados obtenidos buscando en la totalidad de las leyes introducidas en la aplicación.

Para optimizar la búsqueda, se ha añadido un componente que permite elegir, fácil y rápidamente, entre los diferentes campos en que se desea filtrar la búsqueda.

Como resultado de buscar una palabra en concreto, la aplicación filtra por dicha palabra, mostrando todas las ocurrencias encontradas.

Como puede apreciarse, indica en que ley ha encontrado la palabra buscada, así como cuál es su artículo y, si procede, su cuantía económica y puntos a detraer del permiso de conducir.

Resto de legislación: Muestra al usuario el listado de leyes, reglamentos y decretos disponibles para consultar. En definitiva, esta sección contiene toda la legislación que no es de índole municipal.

Permite al usuario ver que contiene la aplicación, pudiendo seleccionar la que desee.

Las Palmas de G.C.: En esta pantalla se muestran las diferentes opciones a elegir para el municipio seleccionado. Para este caso, el ayuntamiento seleccionado posee una agenda con teléfonos de interés, el calendario laboral correspondiente a la policía local del municipio, el conjunto de ordenanzas municipales y los puntos kilométricos de la autovía.

Ordenanzas Municipales: En esta pantalla podremos observar toda la legislación de índole municipal, que básicamente son ordenanzas municipales.

Permite al usuario ver que leyes municipales contiene la aplicación, pudiendo seleccionar la que desee.

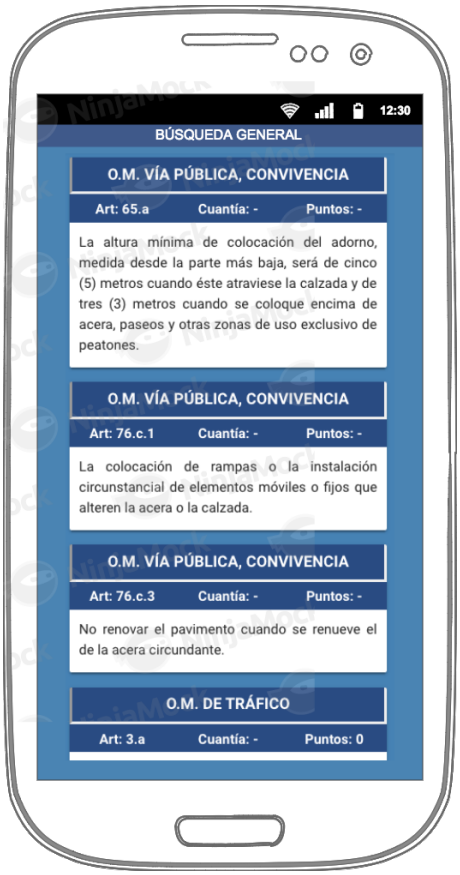
Pantalla Ley: muestra la información referente a la ley seleccionada, en este caso, la ordenanza municipal de tráfico.

Inicialmente muestra todos los artículos y, en su caso, su cuantía y punto a detraer, así como el hecho en sí. Esta pantalla permite la consulta de la ley en su totalidad, sin filtros.

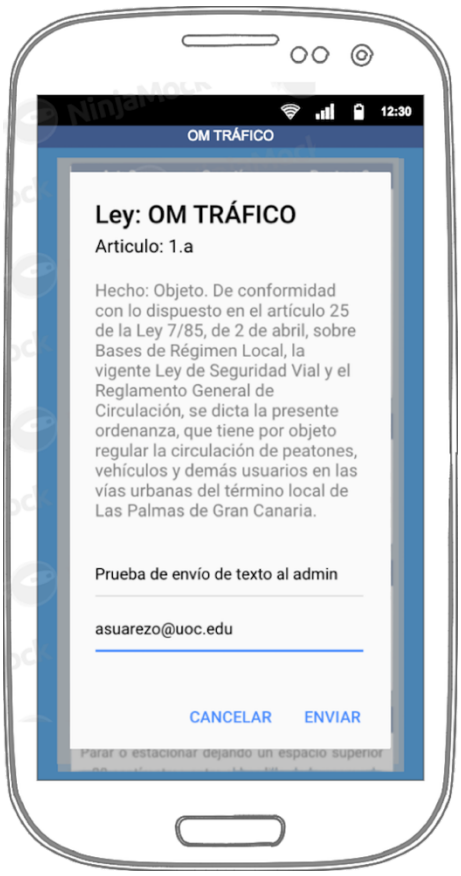
Como en el caso de la pantalla de "*Búsqueda General*" incorpora un filtro muy útil que permite al usuario buscar por los diferentes ítems de un artículo.

En este caso se ha decidido filtrar por la palabra "*acera*", por lo que la aplicación mostrará todos aquellos artículos que contenga dicha palabra dentro de la citada ley.

La utilización del filtro de búsqueda permite al usuario encontrar lo que está buscando de una forma bastante rápida y eficaz



<u>Ocultación título</u>: En la pantalla que muestra el detalle de una ley, de la agenda, calendario o punto kilométrico, si hacemos scroll para sobre la pantalla para ir desplazando los ítems, el título de la app se esconde para mostrar más información en la pantalla. El mismo vuelve a aparecer si hacemos el scroll inverso.



Envío de sugerencia: en el caso de querer enviar una sugerencia sobre un artículo, bien porque se haya derogado, se haya cambiado o por cualquier otra causa, el usuario solo tiene que pulsar sobre el citado artículo y aparecerá una pantalla con la información del mismo, un campo para introducir el texto que se desee enviar al administrador y un correo electrónico para el posible feedback.

Una vez se pulse en "ENVIAR", se enviará a la base de datos de Firebase la citada sugerencia.

Punto Kilométrico Autovía: muestra al usuario una relación de puntos kilométricos de la citada vía asociados a una imagen del mismo, para que sepa en qué lugar, aproximadamente, se encuentra.

Esta pantalla es muy importante, ya que se trata de una vía muy transitada y en la que se interviene con bastante frecuencia, y es muy difícil saberse de memoria todos y cada uno de los puntos kilométricos de la misma.

De esta forma, el usuario sabrá de forma rápida donde se encuentra.

<u>Agenda</u>: la aplicación facilita al usuario una agenda con muchos números de teléfonos de interés, centrada, sobre todo, en servicios municipales del ayuntamiento de Las Palmas de Gran Canaria.

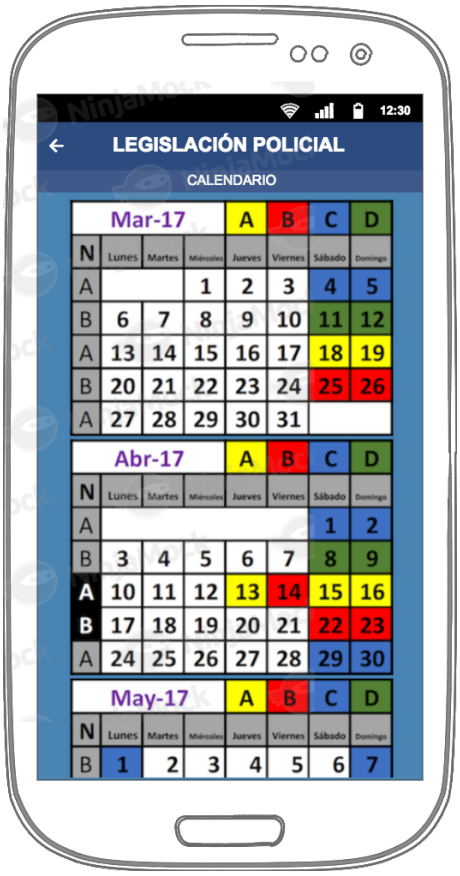
Inicialmente, la aplicación muestra la totalidad de los números, pero se puede filtrar por el nombre del servicio buscado.

Si el usuario pulsa sobre el número de teléfono, le permite realizar una llamada telefónica al nº pulsado.

Esta agenda puede crecer de forma rápida, dinámica y transparente para el usuario.

En esta imagen se puede observar como la aplicación hace un filtrado de un número buscado por el usuario, en este caso, busca "publi".

Como en el caso anterior, si el usuario pulsa sobre el número de teléfono, le permite realizar una llamada telefónica al nº pulsado.



Calendario: por último, la aplicación muestra el calendario laboral perteneciente a la policía local del Ayuntamiento de Las Palmas de Gran Canaria.

Gracias a esta pantalla, cualquier miembro del cuerpo puedo saber de forma rápida que festivos y fines de semana le toca trabajar.

Aunque inicialmente la aplicación parte sólo con este calendario, se pueden añadir todos lo que sean necesarios de forma rápida, dinámica y transparente para el usuario.

5. Arquitectura de la aplicación

En la siguiente imagen se puede apreciar el funcionamiento global de la aplicación. Se parte de un administrador, que gestiona la base de datos que almacena la información consultada por los usuarios a través de la aplicación.

Como ya hemos comentado anteriormente, Ionic 2 es un framework para el desarrollo de aplicaciones híbridas. Se caracteriza fundamentalmente porque usa por debajo Angular 2 juntos a una gran cantidad de componentes, facilitando mucho el desarrollo de las aplicaciones multiplataforma.

Esta herramienta nos permite crear aplicaciones de forma rápida, con poca inversión económica y para distintas plataformas con un mismo código fuente.

A diferencia de otros lenguajes de programación como Java, donde las entidades y clases quedan claramente diferenciadas, en Ionic es diferente. En este caso se trabaja con páginas, componentes, directivas y provider (o servicios), por lo que, en vez de presentar el típico diagrama UML con las clases y entidades, reflejaré cada una de las páginas utilizadas, los componentes y servicios que necesita, su utilidad y necesidad en la aplicación.

Los proyectos realizados en Ionic tienen la siguiente estructura de carpetas [9]:

- src:
    - Es la carpeta que contiene la mayor parte del trabajo con Ionic, como el código que viene dado al iniciar el proyecto

o los archivos que escribirás al desarrollar la aplicación para poder.
    - Es en este directorio donde encontraremos las carpetas que representaran la vista de la aplicación (pages, components y directivas), así como los controladores de la misma (providers).

- www:
    - Se encuentran los archivos que serán convertidos a archivos web desde la carpeta src a la carpeta www, y que se podrán visualizar correctamente en el navegador.

- plugins:
    - Contiene todos los plugins nativos, bien sean propios de Ionic o de terceros

- platforms:
    - Se encuentran los archivos necesarios para dar soporta a cada plataforma con la que se desee trabajar (Android, IOS…)

- resources:
    - En ella se guardan los iconos de la aplicación, así como de la pantalla splash, que es la que aparece mientras se carga la aplicación.

- node_modules:
    - Contiene las dependencias necesarias para el correcto funcionamiento de la aplicación, como las librerías en otros lenguajes de programación. Dichas dependencias vienen registradas en el fichero de configuración "package.json".

Para que la aplicación sea práctica para los usuarios y tenga éxito, es de vital importancia tener la base de datos de consulta actualizada en todo momento, y que dichas actualizaciones no supongan un problema para los usuarios. Es por ello que se ha decidido utilizar como base de datos firebase [10], lo que permite mantener la información actualizada en todo momento de forma transparente para el usuario.

El acceso a la base de datos es realizado por el controlador correspondiente de la aplicación, que será el encargado de gestionar

toda la lógica de negocio entre la aplicación y firebase. Dicho controlador será explicado con mayor detalle más adelante.

Para el correcto desarrollo del proyecto y su posterior mantenimiento, es necesario hacer uso de un patrón que permita llevarlo a cabo. El patrón utilizado en este proyecto es el Vista-Controlador.

Por un lado, tenemos la vista, que será representada por una serie de páginas (***pages*** en Ionic) que mostrarán información al usuario, algunas de las cuales contendrán componentes y/o directivas que ayuden a completar la vista.

Por otro lado, tendremos los controladores (provider en Ionic), que serán los encargados de gestionar toda la lógica de negocio de la aplicación.

5.1 Elementos de la vista

El elemento encargado de mostrar las distintas páginas que forman la vista de la aplicación es una ***"page"*** con sus respectivos componentes. La aplicación cuenta con dos, siendo las siguientes:

- **law-list-page**:

Esta vista se encargará de las siguientes funciones:

- **constructor:** inicializa todos los elementos necesarios de la vista para que pueda ser visualizada correctamente.
- **ionViewDidLoad:** muestra un mensaje mientras se cargan todas las variables necesarias.
- **initializeItems:** inicializa los distintos parámetros de la página.
- **initializeItemsToShow:** inicializa los primeros 20 elementos que se van a mostrar, para evitar cargar en memoria la totalidad de la lista, optimizando el consumo de recursos.
- **getFirebaseData:** inicializa las variables relacionadas con la conexión a firebase
- **goToKindOfLaw:** permite navegar hacia la correspondiente página cuando se pulsa sobre algún ítem de la lista.

En líneas generales, esta página muestra el listado de ítem disponibles a visualizar, pudiendo ser leyes, decretos, ordenanzas, agenda etc. En las siguientes pantallas se ve la información mostrada por la citada página.

Se ha decidido este diseño ya que los elementos son mostrados en forma de una típica lista, resultando bastante fácil identificar cada uno de los posibles ítems a seleccionar, la forma de interactuar con ellos y el posible resultado al pulsar sobre cualquiera de ellos.

Por otro lado, tenemos la ***"page"*** que muestra el detalle de los ítems contenidos en una ley, en la agenda etc.

- **Ítem-to-show-page:**

Esta vista se encargará de las siguientes funciones:

- **constructor:** inicializa todos los elementos necesarios de la vista para que pueda ser visualizada correctamente.
- **initializeItems:** inicializa los distintos parámetros de la página.
- **initializeItemsToShow:** inicializa los primeros 20 elementos que se van a mostrar, para evitar cargar en memoria la totalidad de la lista, optimizando el consumo de recursos.

- **<u>getFirebaseData</u>:** inicializa las variables relacionadas con la conexión a firebase.
- **<u>showItems</u>:** permite determinar el tipo de componente que se carga en base al ítem seleccionado en la página "*law-list-page*".

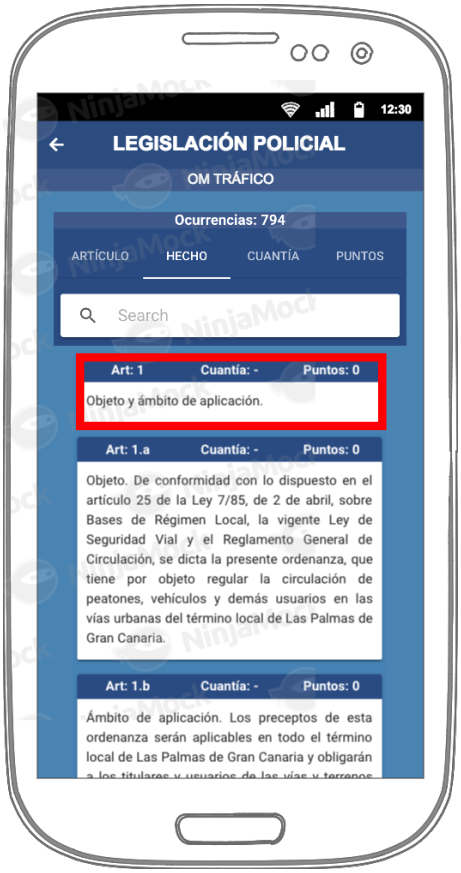
Esta página motrará la relación de artículos contenidos en una ley, los nº de teléfono de la agenda, la información correspondiente al calendario laboral seleccionado o de los puntos kilométricos de la autovía elegida.

Con este diseño se accede de forma rápida y clara a la información detallada de cada uno de los ítems, permitiendo una lectura ordenada de los mismos, lo que facilita el acceso a la información.

Hace uso de los siguientes componentes según la información a mostrar:

<u>TitleComponent:</u> permite agregar un subtítulo a las diferentes pantallas que lo necesiten, indicando en todo momento en que sección de la misma nos encontramos.

Este componente, a diferente del título de la aplicación, no se esconde cuando realizamos scroll en la lista de ítems.



<u>LawRowComponent</u>: muestra la información de cada artículo en la página correspondiente.

Este componente permite la gestión de los ítems a mostrar en cada una de las leyes de forma rápida y sencilla.

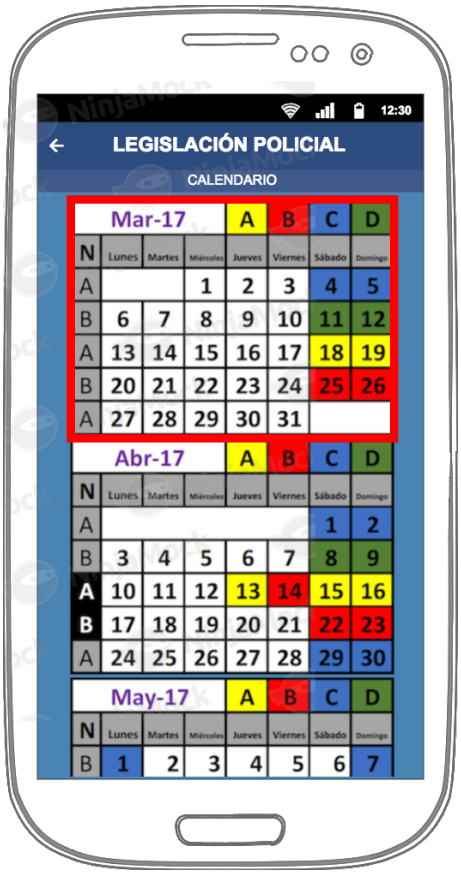
Modificando este componente, el cambio se ve reflejado de manera inmediata en cualquier paree de la aplicación que haga uso del mismo, lo que facilita su mantenimiento y actualización

<u>PhoneBookRowComponent</u>: muestra la información de cada nº de teléfono, permitiendo hacer una llamada si se pulsa sobre dicho número.

Este componente facilita la gestión de la agenda, ya que permita modificarla con facilidad y de forma transparente para el usuario.

<u>PkAutoviaComponent</u>: permite la visualización de los datos referentes a los puntos kilométricos de la citada vía.

Este componente nos permite gestionar los puntos kilométricos de la autovía, pudiendo agregarlos o eliminarlos de forma rápida y sencilla.



<u>CalendarComponent</u>: este componente permite la visualización del calendario laboral correspondiente, dividido por meses.

Permite la gestión del calendario de forma personalizada, pudiendo cambiar cualquier aspecto de forma muy rápida.

En las dos últimas pantallas se muestra información que está contenida en diferentes imágenes, por lo que se ha decidido un diseño en el que cada imagen ocupe casi la totalidad de la pantalla, de forma que esta sea accesible para todo tipo de usuarios.

Por otro lado, en la pantalla que muestra el detalle de una ley, de la agenda, calendario o punto kilométrico, se hace uso de la directiva ***"HideHeaderDirective"*** para esconder el título de la app si hacemos. De esta forma se puede mostrar más información en la pantalla. El mismo vuelve a aparecer si hacemos el scroll inverso.

5.2 Controladores

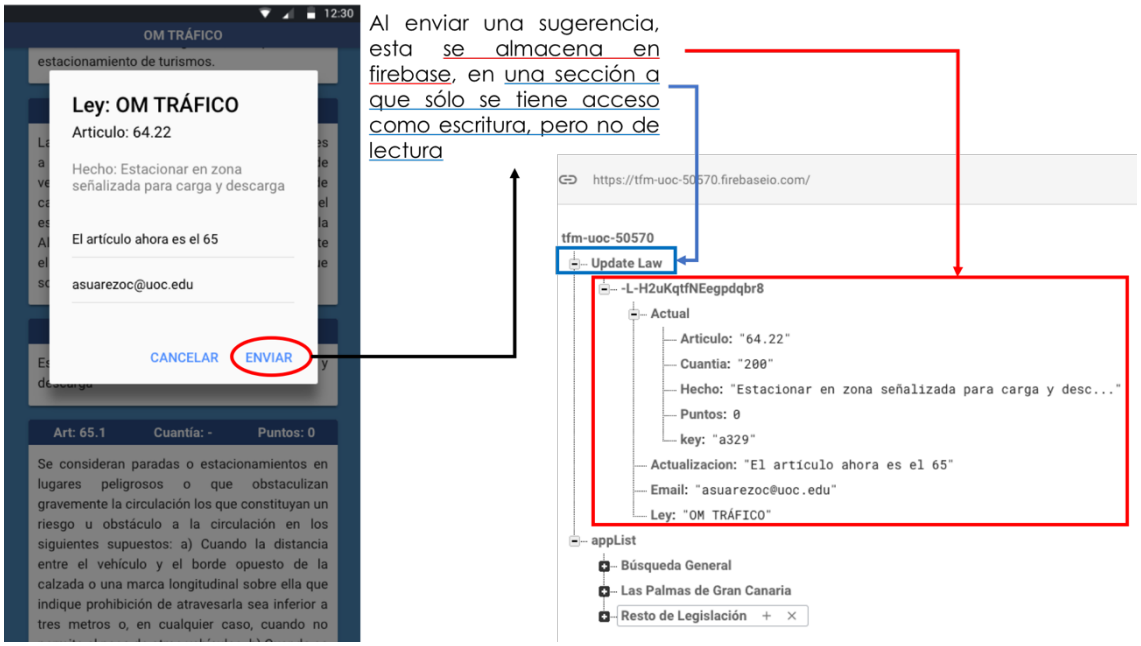
Todas las vistas son manejadas por sus respectivos controladores o provider. En este proyecto necesitaremos dos, uno que gestione la lógica de negocio de la base de datos, y otro manejará la información que se mostrará en cada una de las vistas:

- **firebase-provider**: será el encargado de gestionar la información relacionada con la base de datos y enviársela a la vista correspondiente. Para ello, la aplicación ha sido configurada para permitir el acceso a la base de datos de firebase [10], permitiendo al controlador el acceso a la misma.

Para el acceso a firebase se han establecido una serie de reglas [11] *que permiten el acceso de sólo lectura en general, pero de sólo escritura para la parte donde los usuarios enviarán sus sugerencias al administrador. De esta forma, se garantiza que los usuarios sólo podrán ver la información que el administrador decida en cada momento.*

El componente será el encargado de realizar las siguientes tareas:

- **getDatabaseRef:** Se conectará a firebase para obtener una instancia de la base de datos.
- **getUrlList:** Obtendrá la url de acceso a firebase de cada una de las pantallas que se mostrarán en la aplicación.
- **getList:** Facilitará una lista con todos los elementos a mostrar dada una url de acceso.
- **updateLaw:** este método, unido al método ***"presentPrompt"***, guardará en la base de datos las sugerencias enviadas por los usuarios para que puedan ser valoradas por el administrador, tal y como se muestra en la siguiente imagen:



- **model-provider**: este controlador se encargará de procesar el modelo de datos de la aplicación para permitir que se muestre la información correspondiente de cada una de las vistas.

Este componente será el encargado de realizar las siguientes tareas:

- **goToKindOfLaw:** Permitirá la navegación entre las distintas pantallas de la aplicación dada una url de acceso a firebase.

- **doInfinite:** Carga el componente ***"ion-infinite-scroll"*** [12] de Ionic que permite realizar "*scroll infinito*" en aquellas páginas que contengan una lista muy grande de ítems, de forma que los mismo se vayan cargando en bloques de "*n*" elementos. De esta forma, se optimiza el consumo de recursos y agiliza la presentación de los elementos en la pantalla.
- **isLastPageReached:** La anterior función se utiliza conjuntamente con esta para averiguar si se ha cargado el último elemento de la lista para dejar de hacer scroll.
- **filterItems:** permite filtrar los elementos de una lista en base a la palabra insertada en el campo de búsqueda, y teniendo en cuenta el ítem por el que se desea filtrar.
- **cleanText:** esta función se encarga de "*limpiar*" el texto a la hora de filtrar los elementos de búsqueda, de forma que no discrimine si una palabra se ha escrito con tilde o no, por ejemplo, si se filtra por la palabra "*movil*", la aplicación mostrará todos los ítems que contengan las palabras "*movil*" y "*móvil*", ofreciendo una búsqueda más versátil.
- **showPreloader:** muestra un mensaje informativo al usuario mientras se cargan datos en la lista de ítems que se están mostrando en la pantalla cuando se hace scroll.
- **validateEmail**: cuando se quiere enviar información sobre algún artículo al administrador, hay que rellenar dos campos. Uno es el texto que se desea enviar y el otro un email de contacto para el posible feedback con el usuario. Esta función comprueba que los datos introducidos en el correo tienen el formato de correo electrónico.
- **showMessage**: muestra un mensaje informativo en el centro de la pantalla de 2 segundos de duración.

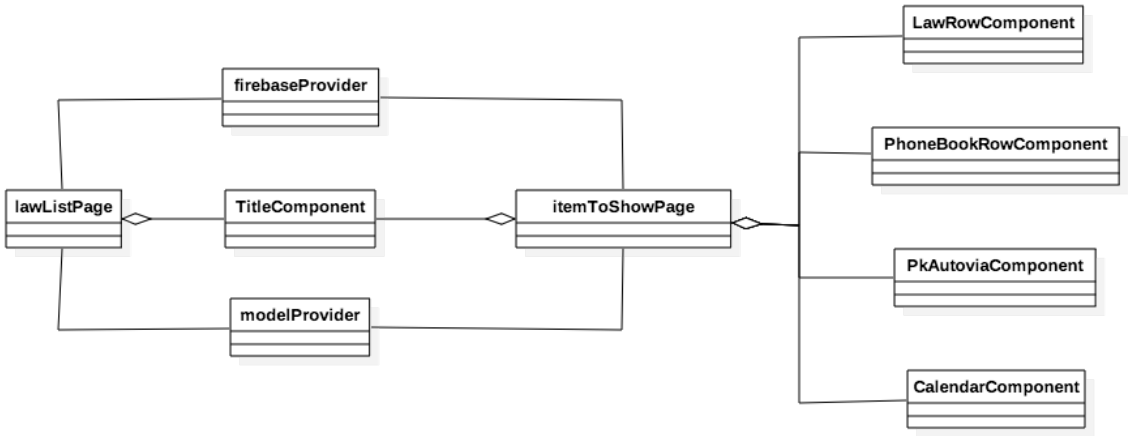
5.3 Base de datos

Para el almacenaje y consulta de la información contenida en la aplicación se ha utilizado firebase por los siguientes motivos [13]:

- **Rapidez:** permite detectar errores rápidamente, minimizando el tiempo empleado en la creación de la base de datos, pudiendo almacenar todo en la Nube, testear la app o configurarla de manera remota.

- **Infraestructura:** Gracias a las API contenidas en el SDK, firebase se encarga de todo, evitando perder tiempo en tener que desarrollar una infraestructura.
- **Multiplataforma:** Permite crear aplicaciones para diferentes tipos de plataformas con API integradas a SDK individuales para Android, IOS y JavaScript.
- **Inicio gratuito:** La mayoría de las funciones de Firebase son de uso gratuito para cualquier tamaño.
- **Escalable:** Si la app es un éxito, firebase se encarga de escalar por ti.
- **Soporte gratuito:** Google ofrece soporte gratuito por correo para todos los desarrolladores. Además, firebase y expertos en desarrollo de Google participan activamente en comunidades en línea como Stack Overflow y GitHub.

A continuación se muestra, de forma gráfica, la estructura de la aplicación, donde pueden verse las dos páginas principales controladas por los correspondientes controladores o providers, así como los componentes asociados a cada una.



Se ha decidido por esta arquitectura para facilitar tanto el desarrollo de la aplicación como el mantenimiento y actualización de la misma. De esta forma, se pueden añadir y quitar leyes de forma rápida y dinámica, sin necesidad de tener que publicar actualizaciones en los determinados stores, y de forma completamente transparente para el usuario.

6. Desarrollo

Para la implementación de esta fase se han desarrollado una serie de diagramas UML que han servido de guía durante toda la fase de implementación de la aplicación.

Las funcionalidades de cada uno de los elementos mostrados en los diagramas han sido explicadas detalladamente en los puntos 5.1 y 5.2 de esta memoria.

Los citados diagramas se muestran a continuación:

6.1 LawListPage

LawListPage
-law: FirebaseListObservable -items: Array -lawList: Array -lawTitle: String -urlList: string -totalUrlList: String -total: int -loader: Loading
constructor(FirebaseProvider, NavParams, ModelProvider, LoadingController) ionViewDidLoad() -initializeItems() -initializeTotalItemsToShow(law) -getFirebaseData() +goToKindOfLaw(listOfArticlesToShow, dataTitle, lawTitle

6.2 ItemToShowPage

ItemToShowPage
-lawList: Array -items: Array -lawTitle: String -kindOfLaw: String -phoneBookItemShow: String -lawItemShow: String -showCalendar: String -showPkAutovia: String -total: int -searchControl: FormControl
constructor(NavParams) -initializeItems() -initializeTotalItemsToShow(law) -getFirebaseData() +showItem(itemToShow)

6.3 CalendarComponent

CalendarComponent
Items: Input() lawList: Input() -page: int
constructor(ModelProvider) +doInfinite(infiniteScroll) +isLastPageReached()

6.4 LawRowComponent

LawRowComponent
items: Input() lawList: Input() ocurrences: Input() lawTitle: Input() -page: int -total: int -searchTerm: String -itemToFilter: String -checkup: String -totalFilterItemToShow: Array
constructor(ModelProvider, FirebaseProvider) +logChosen() +filterItems -initializeTotalItemsToShow(law) +doInfinite(infiniteScroll) +isLastPageReached() +updateLaw (itemToUpdate, lawTitle)

6.5 TitleComponent

TitleComponent
lawTitle: Input()
constructor()

6.6 PkAutoviaComponent

PkAutoviaComponent
items: Input() lawList: Input() -page: int
constructor(ModelProvider) +doInfinite(infiniteScroll) +isLastPageReached()

6.7 PhoneBookRowComponent

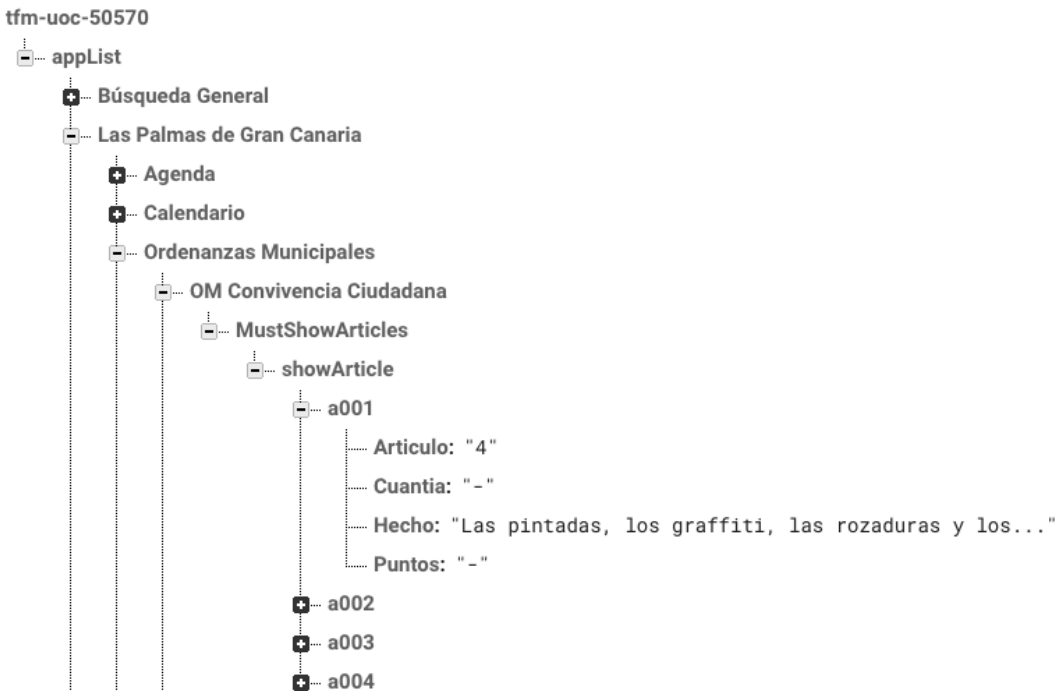
PhoneBookComponent
items: Input() lawList: Input() -page: int -total: int -searchTerm: String -itemToFilter: String -checkup: String -totalFilterItemToShow: Array
constructor(CallNumber, ModelProvider) +makeACall(phone) +filterItems() -initializeTotalItemsToShow(law) +doInfinite(infiniteScroll) +isLastPageReached()

6.8 FirebaseProvider

FirebaseProvider
navCtrl: NavController database: firebase.database.Reference
constructor(AngularFireDatabase, App, AlertController, ModelProvider) -getDatabaseRef() +getUrlList(urlList) +getList(urlList) -updateLaw (itemToUpdate, lawTitle) -presentPrompt(itemToUpdate, lawTitle, changeData)

Para la consulta de los diferentes aspectos contenidos en la aplicación se hace uso de firebase como base de datos. Se ha decidido por dicha plataforma por la facilidad de uso, acceso y actualización de la misma, pudiendo realizarse desde cualquier lugar y dispositivo con acceso a internet, permitiendo tener la aplicación actualizada en todo momento y de forma completamente transparente para el usuario.

El contenido de la base de datos ha sido desarrollado en un único fichero json [14] que ha sido subido a firebase, y cuyo desglose parcial puede observarse en la siguiente imagen:



6.9 ModelProvider

ModelProvider
navCtrl: NavController
constructor(App, ToastController) +goToKindOfLaw(listOfArticlesToShow, urlList, title, kindOfLaw) +doInfinite(page, items, law, infiniteScroll) +isLastPageReached (page, loadedItemList) +filterItems(searchTerm, items, checkup) -itemCleandedToLowerCase(item, checkupClened) -searchTermCleanedToLowerCase (searchTerm) -cleanText(input) +showPreloader(loading, loader) +validateEmail(email) +showMessage(message)

6.10 HideHeaderDirective

HideHeaderDirective
header: @Input headerHeight; scrollContent;
+onContentScroll(event)

6.11 Plugins externos

Para el correcto funcionamiento de la aplicación se han tenido que instalar dos plugins externos, uno que permita la conexión y gestión de la base de datos en firebase, y otro que permita realizar llamadas desde el teléfono móvil.

6.11.1 Plugin Firebase [15]

Para la utilización de firebase en el proyecto han de instalarse las correspondientes dependencias:

$ ionic cordova plugin add cordova-plugin-firebase
$ npm install --save @ionic-native/firebase
$ npm install angularfire2 firebase --save

Para el acceso a la base de datos se ha agregado toda la información necesaria en una variable de entorno en el proyecto, dentro de la siguiente ruta: "*/src/environments/environment.ts*". La siguiente imagen muestra dichos datos:

```
environments.ts
import * as firebase from "firebase";

// Initialize Firebase
var config = {
  apiKey: "[redacted]",
  authDomain: [redacted],
  databaseURL: [redacted],
  projectId: "[redacted]",
  storageBucket: "[redacted]",
  messagingSenderId: "[redacted]"
};

firebase.initializeApp(config);
```

6.11.2 Plugin CallNumber [16]

Este plugin nos permite poder realizar llamadas desde el teléfono móvil. Para ello, han de ejecutarse las siguientes instrucciones:

```
$ ionic cordova plugin add call-number
$ npm install --save @ionic-native/call-number
```

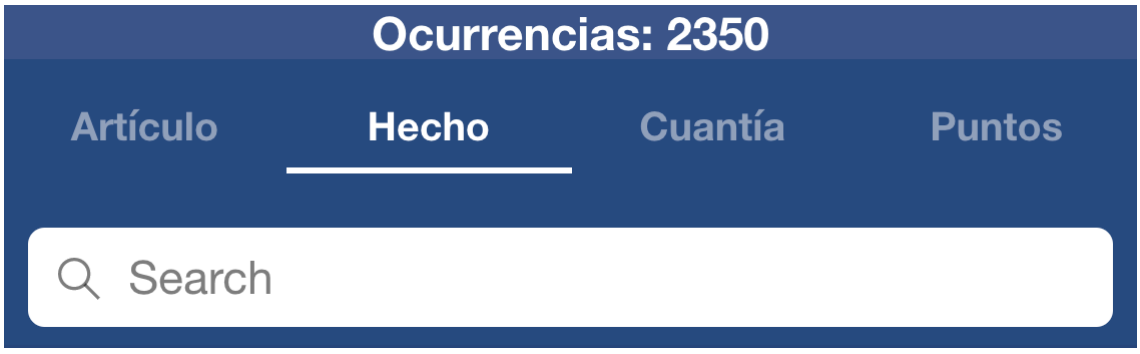
7. Pruebas

Para comprobar el correcto funcionamiento de la aplicación se han ido realizando pruebas de usuario a medida que se iban desarrollando nuevas funcionalidades.

Una vez completada la primera versión de la aplicación, se ha dado a probar a un total de 20 personas relacionas con los servicios de seguridad y emergencias y ajenos a ellos.

Tras la obtención y valoración de los resultados, se llegaron a las siguientes conclusiones:

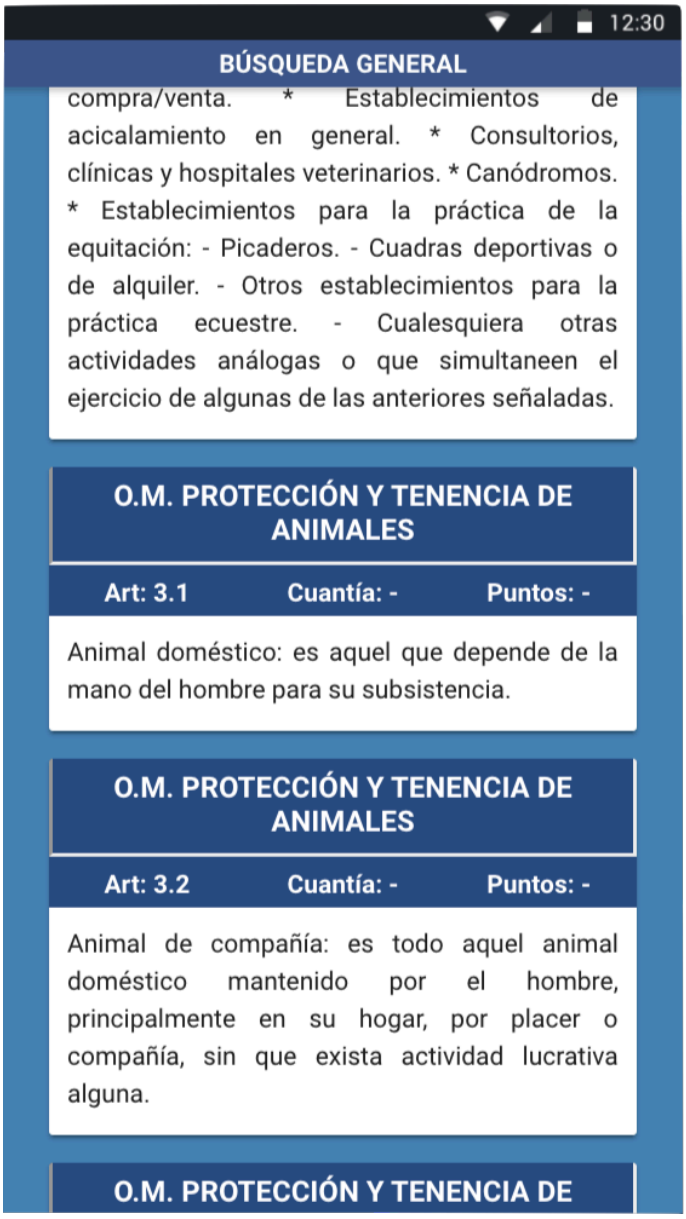
- Inicialmente se contaba con una búsqueda reducida en opciones, permitiendo únicamente filtrar por el "*hecho*" del artículo o el nombre del contacto telefónico correspondiente. Tras observar que 19 de los 20 usuarios sugirieron incorporar algún sistema que permitiera elegir el tipo de ítem por el que buscar (artículo, hecho, cuantía o puntos), se procedió a añadir un componente de Ionic llamado "*ion-segment-button*", permitiendo al usuario cambiar entre los distintos tipos de ítems de filtrado con solo pulsar sobre ellos, tal y como se observa en la siguiente imagen:



- La aplicación contaba con un diseño de fondo blanco, pero se decidió cambiar a fondo azul tras valorar como positiva la sugerencia de la mayoría de los usuarios, por ser este un color relacionado con el color de la uniformidad de los cuerpos policiales. En la siguiente imagen puede observase lo comentado anteriormente:

- Al navegar por la agenda o el interior de los artículos contenidos en una ley, la mayoría de los usuarios echaban en falta la funcionalidad incluida en muchas aplicaciones que oculta la cabecera. Por ese motivo, se decidió añadir una directiva *(HideBarDirective)* en la aplicación que llevara a cabo tal efecto,

permitiendo disponer de mayor cantidad de información en la pantalla, tal y como se muestra en la siguiente imagen:



- Cuando se navegaba entre las distintas pantallas, la aplicación cargaba la totalidad de los componentes y elementos en memoria. Esto provocaba un retardo en la carga de páginas con una gran cantidad de elementos como es el caso de la agenda, con 828 elementos, o la página de "*Búsqueda General*", con 2350. Esto repercutía negativamente en el manejo de la aplicación, recibiendo bastantes críticas por ello. Por ese motivo se decidió cambiar la forma en la que se cargaban los elementos. Ahora, al entrar en una página con gran cantidad de elementos a cargar, estos se cargan en bloques de 20 a medida que se va haciendo

scroll, teniendo un tiempo de carga prácticamente imperceptible para el usuario, consiguiendo aumentar bastante el rendimiento de la aplicación.

Los resultados obtenidos en las pruebas comentadas anteriormente han permitido desarrollar una versión mejorada de lo que se tenía pensado inicialmente.

Como bien es sabido, el mundo de las aplicaciones móviles está en constante cambio y evolución, motivo por el que esta aplicación ha sido diseñada para adaptarse fácilmente a los cambios y necesidades futuras.

8. Conclusiones

Durante el desarrollo del proyecto he podido ir adquiriendo experiencia en aspectos que, hasta el momento, eran desconocidos para mí, las cuales se comentan a continuación.

Por un lado, durante el transcurso del proyecto mis nociones básicas sobre Framework Ionic han mejorado, dando como resultado un desarrollo que creo puede ayudar de forma significativa en las labores diarias de los agentes de la Policía. Aun así, no se descarta una mejora del producto en un futuro inmediato.

La utilización de Firebase como base de datos supuso un reto, porque aun consultando bibliografía al respecto, era un campo desconocido para mí, pudiendo afirmar que la curva de aprendizaje resultó bastante sencilla. Además, sus funcionalidades ayudan de una forma significativa a simplificar la puesta en marcha del proyecto.

Es un hecho que todo nuevo proyecto debe contener una valoración de los usuarios del aplicativo a diseñar para mejorar su usabilidad y generar empatía con el proyecto, redundando en la mejora del producto final. Aplicando estos conceptos, este proyecto me ha servido para tener mi primera experiencia directa con los usuarios, que han contribuido a obtener un feedback de mejora con respecto a la idea inicial del mismo.

Para el desarrollo del proyecto se ha diseñado un plan ejecución que ha sido seguido para tratar de conseguir los objetivos marcados inicialmente.

En líneas generales, se han conseguido los objetivos marcados, obteniendo una aplicación multidispositivo capaz de:

- Consultar información por parte de los agentes de la Policía y ciudadanos en general.
- Permitir el envío de sugerencias y/o actualizaciones de la información contenida en la aplicación.
- Hacer uso de Firebase como base de datos.
- Ser lo más accesible posible para que la información pueda ser consultada en cualquier momento y desde cualquier dispositivo.

También me ha permitido la integración de los conocimientos adquiridos durante mi etapa académica fusionándola con mi experiencia laboral, y la aportación de los conocimientos adquiridos durante el Master, sobre todo en lo relacionado con el framework Ionic.

La puesta en práctica de los mecanismos aprendidos durante mi etapa académica me ha servido para llevar un control del timing del desarrollo en que se encuentra el proyecto. La utilización de un ciclo de vida clásico o en cascada me ha permitido ir avanzando poco a poco por las diferentes etapas del mismo, llevando a cabo una revisión al final de cada una, pudiendo determinar si el proyecto está listo para avanzar a la siguiente etapa.

Respecto a la duración del proyecto, considero que los tiempos marcados inicialmente fueron los correctos, si bien es cierto que en un proyecto de desarrollo software siempre surgen imprevistos que alteran la línea de trabajo establecida, como la actualización de algún plugin, librería o recurso de terceros.

Hay que tener en cuenta que, en el desarrollo de un proyecto como este, donde la tecnología está en constante evolución, es imposible profundizar en cada uno de los aspectos que aborda. Me hubiese gustado añadir más funcionalidades y hacer uso de las últimas actualizaciones de cada uno de los plugins instalados, pero hacerlo hubiese supuesto poner en riesgo la duración y resultado del proyecto, por lo que queda como propuesta de mejora.

Quisiera comentar que la ayuda recibida por parte de todo el equipo docente de la UOC ha sido importante durante toda la fase de desarrollo del proyecto, ya que me han resuelto dudas no solo de tipo técnico, sino de tipo formal.

Como cualquier otra aplicación, se puede incluir modificaciones y/o actualizaciones para adaptarlas a las necesidades de los usuarios. Como propuestas de mejoras futuras, cabe destacar las siguientes:

- Añadir un calendario que permita interactuar con él, permitiendo añadir eventos al mismo.
- Añadir una sección de "*Artículos favoritos*" o "*Artículos destacados*", dónde cada usuario pueda seleccionar los artículos que considere necesarios para tener un acceso inmediato desde la pantalla principal.
- Permitir almacenar la información de firebase en el sistema, de forma que pueda ser consultada sin necesidad de tener acceso a internet. Para mantener la aplicación actualizada, se comprobaría si se ha cambiado algo en el momento de abrirla. De ser así, se procedería a su actualización en el sistema, y sino, la información se mostraría haciendo uso de los datos almacenados en "*local*".
- Añadir un apartado donde los usuarios puedan alojar órdenes, manuales o cualquier tipo de documento que puedan considerar de especial relevancia para el servicio.

Por último, destacar la usabilidad del aplicativo desarrollado, observándose que los usuarios no han tenido dificultades en la instalación y manejo de éste, por lo que espero que pueda servir de gran ayuda y mejorar el día a día de aquellas personas que puedan necesitarla

9. Glosario

GIT: es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente

IDE: en ingles Integrated Development Environment. Es un entorno de desarrollo integrado o entorno de desarrollo interactivo. Es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

JSON: es un formato de texto ligero para el intercambio de datos (JavaScript Object Notation).

UML: Es un lenguaje unificado de modelado, empelado para visualizar, construir y documentar un proyecto (en inglés, Unified Modeling Language). Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

API: Interfaz de programación de aplicaciones (Application Programming Interface). Es un conjunto de funcionalidades ofrecidos por una biblioteca o librería para ser utilizado por otro software como una capa de abstracción.

SDK: Kit de desarrollo de software (software development kit). Se trata de un conjunto de herramientas de desarrollo de software que le permite al programador o desarrollador crear una aplicación informática para un sistema concreto.

10. Bibliografía

Para el desarrollo de este proyecto se han empleado los recursos académicos suministrados por la UOC durante toda la fase de formación del máster, así como las páginas de referencia de Firebase y del framework de Ionic.

Página web de referencia de firebase:

https://firebase.google.com/?hl=es-419

Página web de referencia del framework Ionic:

https://ionicframework.com/

Número	Referencia	Información adicional	Fecha
[1]	https://www.jetbrains.com/help/webstorm/meet-webstorm.html	Utilización del IDE WebStorm	nov-17
[2]	https://about.gitlab.com/	Uso de repositorio Git	nov-17
[3]	https://es.atlassian.com/software/sourcetree	Manejo de SourceTree	nov-17
[4]	https://ninjamock.com/	Creación de prototipo	oct-17
[5]	http://staruml.io/	Creación de diagramas UML	oct-17
[6]	https://ionicframework.com/	Framework Ionic	nov-17
[7]	https://msdn.microsoft.com/es-es/library/dd409376.aspx	Requisitos y perfiles de usuario	sept-17
[8]	https://msdn.microsoft.com/es-es/library/dd409432.aspx	Diagrama de casos de uso	sept-17
[9]	https://ionicframework.com/docs/intro/tutorial/project-structure/	Estructura del Framework Ionic	oct-17
[10]	https://firebase.google.com/docs/web/setup?authuser=0	Agregar firebase al proyecto	nov-17
[11]	https://firebase.google.com/docs/database/security/quickstart?hl=es-419	Reglas de acceso a Firebase	nov-17

[12]	https://ionicframework.com/docs/api/components/infinite-scroll/InfiniteScroll/	Scroll infinito en la pantalla de listado de ítems	nov-17
[13]	https://firebase.google.com/?hl=es-419	Motivos por los que usar Firebase	sept-17
[14]	https://www.w3schools.com/js/js_json_intro.asp	Ficheros JSON	sep-17
[15]	https://ionicframework.com/docs/native/firebase/	Plugin de Firebase	nov-17
[16]	https://ionicframework.com/docs/native/call-number/	Plugin Call Number para poder llamar por teléfono	nov-17

11. Anexos

Un diagrama de Gantt, en un fichero llamado "*Planificación de Tareas - PEC 1.pdf*".

Manual explicativo con las instrucciones de compilación, en un fichero llamado "*Instrucciones de compilación.pdf*".

Aplicación empaquetada lista para su instalación en dispositivos Android, con el nombre "*Legislación Policial.apk*".

Manual de usuario donde se explica el uso de la aplicación, en un fichero llamado "*Manual de usuario.pdf*".

***NOTA**: Aunque el desarrollo en Ionic permita exportar la aplicación para ser usada en diferentes plataformas, solo que adjunta la aplicación para Android, ya que la exportación en IOS requiere de una licencia de desarrollador de Apple que no se posee en el momento de realizar este proyecto.*