

APLICACIÓN IOS TRAFIQUEANDO

PARTE 1: PANTALLA DE CARGA E ICONO

Para la pantalla de carga se ha diseñado la imagen que mostramos a continuación que muestra el logo de la aplicación junto con el nombre:



Para la carga de la “launcher image” y la “App icon” se ha utilizado un catálogo de imágenes en el proyecto: images.xcassets

PARTE 2: DESCARGA DE INCIDENCIAS

La aplicación presenta todas las incidencias que registra la ciudad de Zaragoza desde la página oficial del Ayuntamiento en su sección de Movilidad. Estas incidencias comprenden:

- Cortes de tráfico
- Cortes de agua
- Afecciones al transporte público
- Afecciones importantes de tranvía
- Afecciones importantes

Para la descarga del XML se ha utilizado la clase TRDescargaXM que utiliza la conectividad basada en las clases del framework Foundation. Para ello se ha hecho que la clase implemente:

- La petición a la URL que es la fuente de los feed mediante un objeto `NSURLRequest`.
- La ejecución de la petición anterior mediante un objeto `NSURLConnection` en el que se designa como delegado de conexión a la propia clase `TRDescargaXML`.

Como delegado de la conexión la clase implementa el protocolo `NSURLConnectionDataDelegate`.

Como delegada de la conexión `TRDescargaXML` se encarga:

- Establecer la conexión
- Recibir los datos en formato `NSData`
- Gestionar si se produce algún error
- Indicar que se terminó la descarga

Una vez finalizada la descarga la clase `TRDescargaXML` posee un objeto parseador al que llama para traducir el xml obtenido en datos válidos para su uso.

PARTE 3: PARSEO DEL XML DESCARGADO

Una vez que se ha descargado el xml sin problemas se procede a la traducción o parseo del mismo, para ello hemos creado la clase `TRParseoXML` que cumplirá el protocolo `NSXMLParserDelegate` y es delegada de sí misma.

Obtención de tipo de incidencia

La obtención del tipo de incidencia se ha hecho a través de la URL del feed. Si nos fijamos en los campos que forman la dirección, existe uno "id" fijándonos e investigando un poco se llega a la conclusión de que:

- Id=0→Corte de Agua
- Id=1→Corte de Tráfico
- Id=2→Afección Importante
- Id=3→Todas la incidencias
- Id=4→Afección importante de tranvía
- Id=5→Afección al transporte público

De esta manera extrayendo el campo id de la URL asociada al feed podemos obtener el tipo de incidencia.

Característica 1: Parseo y formato del campo fecha y hora

Para el parseo de la fecha se han seguido los siguientes pasos:

- Obtener la fecha en formato `NSString` del buffer del parseador.
- Crear un formateador con el formato en el que se obtuvo la fecha y hora y la zona horaria del usuario.
- Obtener la fecha en formato `NSDate` con el formateador anterior.

- Como la fecha y hora van a mostrarse en una etiqueta se ha creado una función que convierte la fecha NSDate en NSString con el formato dd/mm/aaa hh:mm:ss y la guarda en el feed:

-(NSString *)DarFormatoFecha:(NSDate *)fecha;

Característica 2: Parseo del campo descripción

El parseo del xml descargo lo lleva a cabo la clase llamada TRParserXML. El único problema que planteaba el parseo del xml era que el campo descripción de algunas incidencias contenían etiquetas html que daban formato al texto. Para ello en el parseador se ha implementado una función:

-(void)parseoDescripcion:(NSString *)string;

Esta función realiza las siguientes operaciones:

- Elimina cualquier etiqueta comprendida entre <..> o </..> que se encuentre en el texto pasado como argumento.
- Tras eliminar todas las etiquetas el texto nos quedan caracteres especiales del tipo:
 - (espacio en blanco)→Se sustituye por su equivalente espacio en blanco
 - á é í ó ú→Se sustituye por su equivalente á, é, í, ó, ú.
- Por último daríamos formato a los subtítulos aplicando un salto de línea al inicio de cada nuevo subtítulo y al final añadiríamos “:”. Los subtítulos que nos encontramos pueden ser:
 - Tramo comprendido
 - Desvío
 - Finalización
 - Observaciones

En caso de que el campo descripción apareciera vacío en el feed, la aplicación mostraría en su etiqueta el texto “Sin descripción”.

Característica 3: Parseo del campo latitud y longitud

Para el parseo de la latitud y la longitud que vienen juntas bajo la etiqueta "georss:point" en el xml descargado lo que hemos hecho ha sido:

- Obtenemos el contenido del buffer en formato NSString.
- Separamos en un array de NSString la latitud y la longitud.
- Convertimos la latitud y la longitud en FLoats.
- Guardamos los valores en el feed.

NOTA: El resto de campos del parseo no han presentado peculiaridades.

Una vez finalizado el proceso, el parseador comunica al controlador encargado del listado TRListadoController que terminó el proceso mediante notificaciones.

PARTE 4: ALERTAS POR ERROR PERSONALIZADAS

Tanto si se produce un error en la descarga como si sucede un error en el parseo del xml la app avisará al usuario mediante alertas. Estas alertas son personalizadas y su creación se realiza de forma dinámica, debido a esto se adaptan al contenido.

Por último comentar que el parseador y la clase de descarga avisarán al controlador del listado de que se produjo un error a través de notificaciones:

Alerta de Error en la descarga

Al aparecer la alerta el botón del filtro (ver mejoras en el Listado) desaparece y el botón de refresco (ver también en mejoras en el Listado) aparece inhabilitado.

Al aparecer la alerta el fondo de la aplicación se oscurece para mantener el foco de atención en el texto de la alerta. A su vez la alerta por error de descarga consta de:

- Título que indica que error se produjo durante la descarga
- Descripción del error.
- Pregunta al usuario de reintento en la descarga acompañado de dos botones "Cancelar" y "Aceptar".



Si el usuario pulsa el botón de “Aceptar” se reinicia todo el proceso de descarga del xml de la página del ayuntamiento.

Si por el contrario presiona “Cancelar” la alerta desaparece de la pantalla y veremos una imagen de error como fondo de la tabla que nos avisará de que la descarga falló.

Alerta de Error en el Parseo

Al igual que ocurría al aparecer la alerta de Error de Descarga, cuando aparece la alerta por Error en el Parseo, el botón del filtro (ver mejoras en el Listado) desaparece y el botón de refresco (ver también en mejoras en el Listado) aparece inhabilitado.

En cuanto al contenido de alerta es muy similar a la anterior. Tendrá el título indicando el tipo de alerta y la descripción del error.

La diferencia sustancial reside en que esta alerta posee únicamente un botón de aceptar y su objetivo es meramente informativo. Al pulsar el botón la alerta desaparecerá de la pantalla. A continuación la pantalla de fondo de la app nos indicará que ha habido un error en parseo y desbloqueará el botón de refresco para que volvamos a lanzar el proceso.



PARTE 4: LISTADO DE INCIDENCIAS

Indicador de Carga

El indicador de carga (señalado en amarillo en la imagen) sirve para indicar al usuario que la app está realizando alguna operación, ya sea descargando feeds o parseando los datos descargados. Sirve sobre todo para indicar al usuario que la aplicación no se ha quedado colgada sino que está ejecutando algún proceso. Cuando la app finaliza el icono de carga desaparece de la pantalla.



Listado sin incidencias

Si la descarga y el parseo de los feeds ha sido correcto se mostrará un listado de las incidencias. Si en la página oficial no hay incidencias la tabla mostrará la siguiente imagen.



Al no haber incidencias el botón correspondiente al filtro (véase Mejoras de la Navigation Bar) quedará inutilizado. Sin embargo el botón de refresco nos permitirá reiniciar el proceso de descarga y parseo y en caso de existir nuevas incidencias, actualizar el estado del listado mostrándolas por pantalla.

Listado con celdas personalizadas

Por el contrario si se registraron incidencias, el fondo de la tabla cambiará de imagen y las incidencias pasarán a mostrarse en la parte del listado. Para ello se ha desarrollado celdas personalizadas que cuentan con:

- Imagen que se corresponde con el icono que indica el tipo de incidencia: afección importante, afección importante de travía, afección al transporte público, corte de agua y corte de tráfico, respectivamente.



- La calle en la se ha producido la incidencia
- El tipo de incidencia en formato de texto

- La hora y la fecha en la que se registró dicha incidencia (indicar que no es la fecha en la que se llevará a cabo el incidencia como un corte de tráfico, sino la fecha en la que el operario colgó la noticia en la página del ayuntamiento que es la que nos viene reflejada en el xml descargado→Para ver información sobre la fecha en la que se llevará a cabo tendrá que redirigirse a la web oficial desde la sección de detalle)

A continuación mostramos un par de pantallazos de la parte del listado de la app:



Asimismo cuando se seleccione una de las celdas esta cambiará su fondo a un gris oscuro antes de pasar a mostrar la incidencia detalla.



Implementación del filtro

Si presionamos el botón correspondiente al filtro, que se encuentra en la parte izquierda de la Navigation Bar se desplegará un filtro. Este filtro nos permitirá seleccionar que tipo de incidencias queremos mostrar en el listado. A su vez el icono de desplegar el filtro cambiará por el texto “Aplicar” para hacer efectivo el filtro.

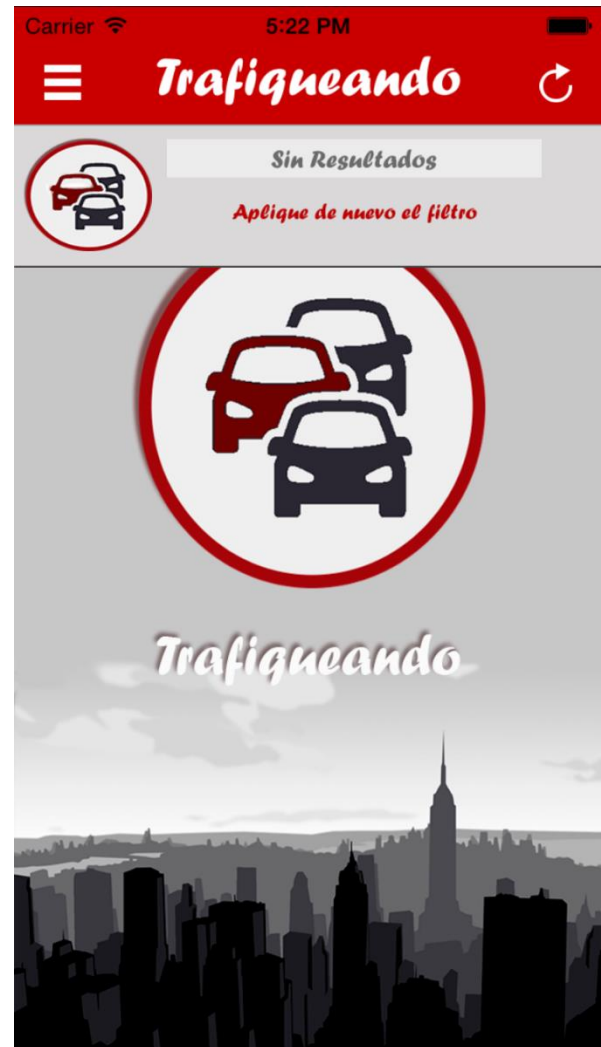
Para poder ver las opciones del filtro es necesario desplegar el menú pulsando el triangulito a la derecha del texto “Selecione...”. Una vez hecho esto el triangulito cambiará para poder ocultar el menú



Una vez seleccionado el tipo de filtro, presionando en opción, el texto “Seleccione...” será sustituido por la opción elegida y el menú se esconderá. Tras esto debemos presionar el botón de aplicar para que el filtro tenga efecto.



Si el filtro encuentra coincidencias las mostrará en el listado, por el contrario si no existen coincidencias aparecerá una celda que indicará que no ha habido resultados y que es necesario aplicar de nuevo el filtro.

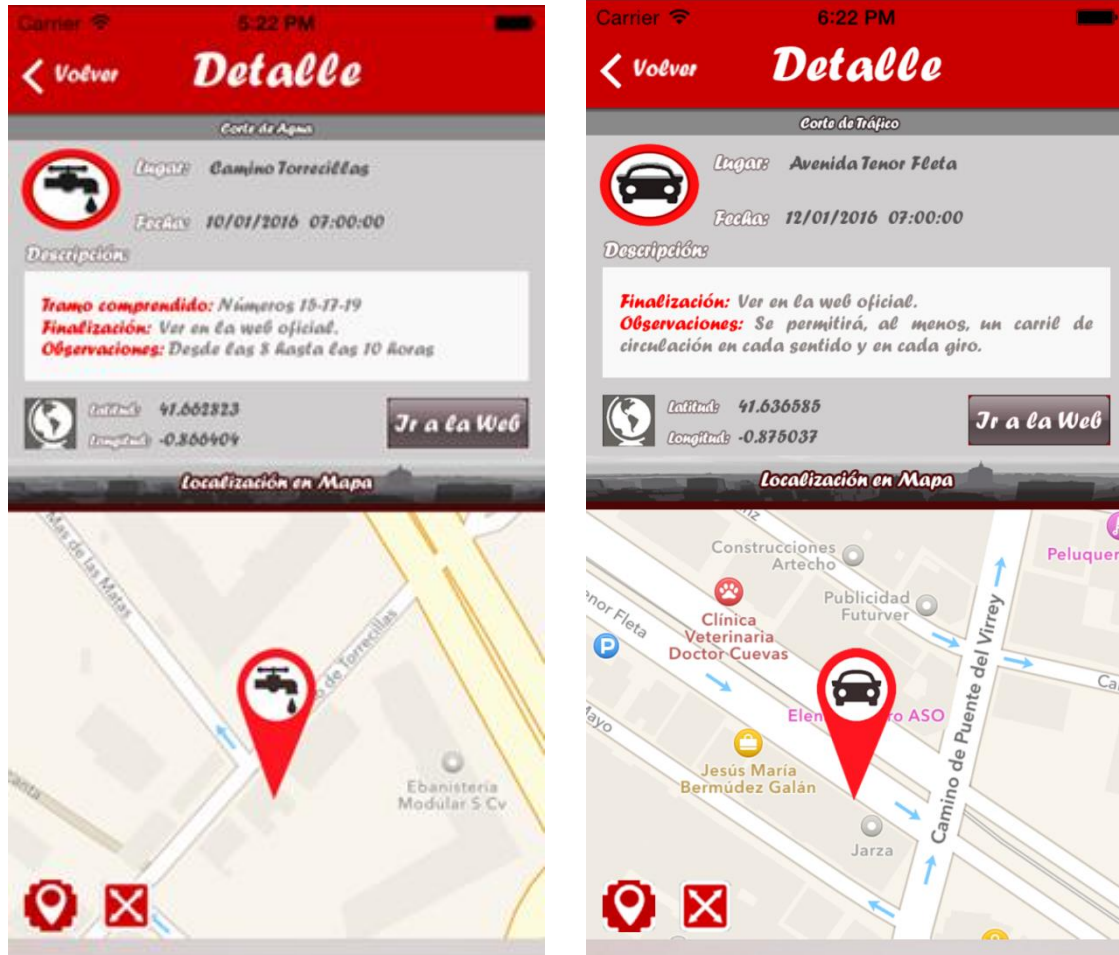


Implementación del botón de refresco

El botón de refresco nos permite realizar una actualización del contenido del listado de forma manual, de manera que al presionarlo el usuario estará reiniciando el proceso de descarga de los feeds de la página oficial y su correspondiente parseo.

PARTE 3: VISTA DETALLADA

Una vez seleccionada una incidencia en el listado pasaremos a verla en detalle:



La vista Detalle muestra:

- 1.-Tipo de incidencia mediante un texto que encabeza la vista
- 2.-Icono del tipo de incidencia
- 3.-Lugar del tipo de incidencia
- 4.-Fecha y hora de registro de la incidencia
- 5.-Descripción con datos de la incidencia
- 6.-Latitud y Longitud de en la que se ubica la incidencia
- 7.-Botón que enlaza con la URL de la incidencia
- 8.-Ubicación de la incidencia en el mapa

Inclusión de la vista Detalle en una ScrollView

Con motivo de mostrar todos los campos con cierto espacio y de forma más clara y visualmente más atractiva, esta vista se ha incluido dentro de una ScrollView que nos permite desplazarlos hacia abajo.

A la izquierda podemos ver la imagen sin aplicar el ScrollView y a la derecha la imagen completa de la vista de detalle indicando donde comenzaría a actuar el Scroll.



Botón de Link

El botón de Link nos permite redirigirnos a la página oficial del ayuntamiento para poder ver más datos que la incidencia. Este botón lanza un nuevo controlador que carga en una WebView el contenido de la URL que se adjunta en el feed sin que tengamos que salir de la aplicación y abrir el navegador del móvil.

Al igual que ocurría en la vista de detalle la navigation bar es personalizada y mostrará como título que nos encontramos en la sección de “Página oficial”, también incorporará el botón “Back” personalizado que nos devolverá a la vista de detalle.



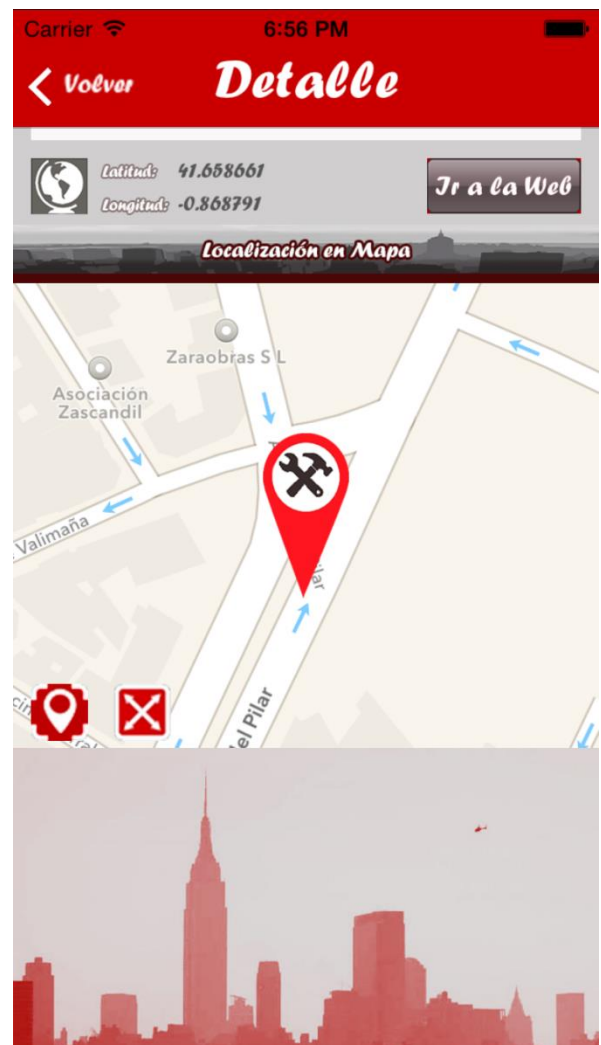
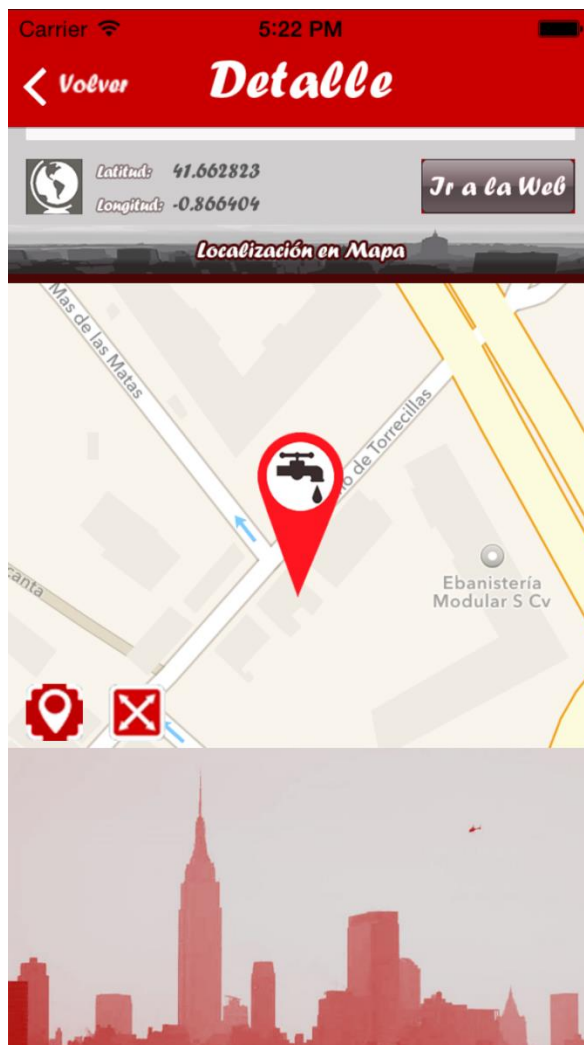
Mapa→Pines del mapa personalizado

Como se puede observar el pin del mapa es personalizado en función del tipo de incidencia que se muestra en detalle.

De esta manera podremos tener los siguientes pines:



A continuación vemos como se verían en el mapa:



Mapa→Anotaciones personalizadas y dinámicas

Cuando presionamos en el pin aparecerá una anotación de detalle, en la que podremos ver el logo de la app, el tipo de incidencia, el lugar donde se produce y la fecha en la que se registra sobre el pin. Para hacer que esta desaparezca el usuario únicamente deberá pinchar en otra parte del mapa o desplazarse por él cambiando la región de este que se está visualizando.

Al igual que ocurría con los campos de la vista detalle el campo del lugar es dinámico y se ajusta al contenido (ver en Mejoras de estilo).



Mapa→Funcionalidades del mapa: Botones

El mapa presenta dos botones en la parte inferior izquierda:

- Botón  Centrar
- Botón Pantalla Completa  

Funcionalidad: Botón Centrar

El botón centrar nos permite centrar el mapa en la posición del pin, de forma que si nos hemos desplazado por el mapa perdiendo la vista la ubicación de la incidencia, el usuario sólo deberá presionar el botón para centrar el mapa en el pin.

Funcionalidad: Botón Pantalla Completa

El botón de pantalla completa nos permite ampliar la vista del mapa para que este ocupe toda la vista de detalle. Esto se realiza mediante una animación que desplaza las etiquetas superiores hacia la derecha escondiéndolas y aumenta el tamaño del mapa.



El botón correspondiente al mostrar en pantalla completa cambia su icono y nos permite regresar a la pantalla de detalle cómo se mostraba en su forma original, igualmente con una animación.

Aquí es donde realmente nos es útil la vista de anotación, puesto que al pasar a pantalla completa todos los campos con la información quedan ocultos. Gracias a la anotación de detalle podemos ver datos de la incidencia dentro del modo pantalla completa.

En este modo funciona igual el botón de centrar, la anotación se muestra igualmente pulsando sobre el pin y se oculta de la misma manera.

PARTE 4: ESTILO GENERAL DE LA APLICACIÓN

Para dar un aspecto más compacto y profesional a la app se han realizado las siguientes mejoras:

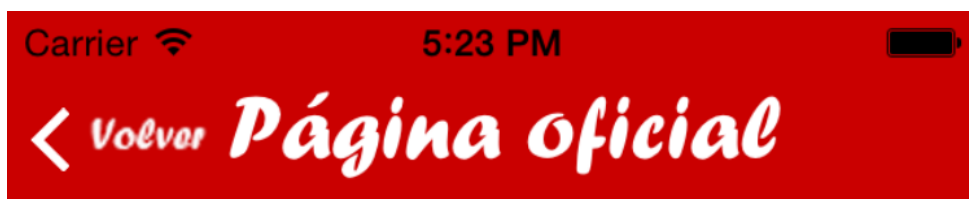
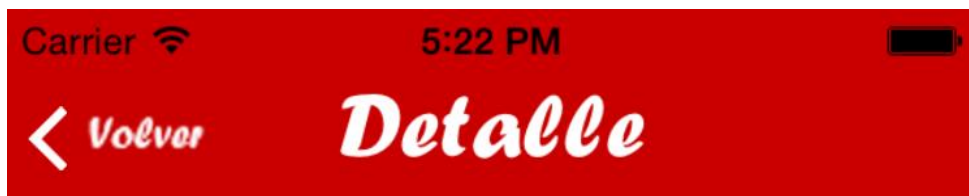
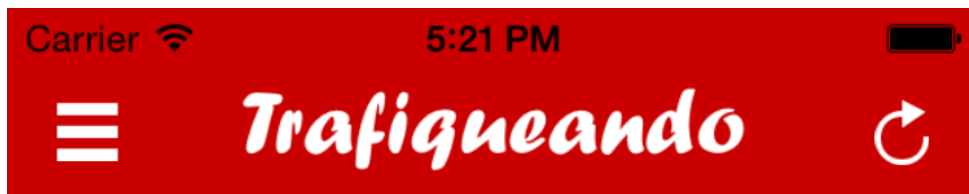
Navigation Bar Personalizada

La Navigation Bar se ha personalizado, como hemos comentado en algún que otro apartado, utilizando el tipo de letra externo y manteniendo los colores rojo y blanco.

Asimismo los botones que corresponden con el filtro, el refresco de incidencias en la tabla y el botón de volver utilizados son personalizados en lugar de usar los que xcode proporciona por defecto.

El título de la Navigation Bar varía en función de la sección en la que se encuentra el usuario, como podemos observar:

- En la sección del listado→Título: Trafiqueando
- En la sección de Detalle→Título: Detalle
- En la sección de Link→Título: Página Oficial



Uso de una fuente externa→Forte.ttf

Para darle un estilo más personal y un acabado mejorado se han utilizado una fuente de letra externa importada al proyecto.

abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ
1234567890.:; ' " (!?) +.*/=

Creación de etiquetas dinámicas

Vista Detalle

Dentro de la vista detalle, tanto la etiqueta que muestra el lugar como la que muestra la descripción calculan dinámicamente el espacio que va a ocupar su contenido y redimensionan su frame para mostrar todo el contenido adecuadamente sin modificar el tamaño de la fuente.

Debido a esto toda la vista detalle se recoloca dinámicamente para mostrar sus vistas en función del contenido de cada incidencia.



Vista Anotación de detalle en el mapa

Igualmente que ocurría en la vista de Detalle en la anotación que aparece al pulsar el pin en el mapa, el campo que presenta el lugar recalcula su tamaño de forma dinámica y con ello toda la tarjeta.



Estilo al campo descripción

Para una mejor visualización del contenido del campo descripción de la vista Detalle se ha procedido:

- Justificación y padding del texto.
- Búsqueda de los subtítulos y cambio de color en rojo para distinguirlos.

Utilización de un Catálogo de imagen

Para una mejor organización del proyecto se han utilizado un catálogo de imágenes como ya comentamos en la parte de la "Launcher image" y el icono de la aplicación. El catálogo llamado `images.xcassets` aglutina todas las imágenes y vistas utilizadas en la aplicación, ordenadas en carpetas según su temática o lugar de utilización.