

## **MEMORIA**

TFC - Aplicación para móviles con iOS 6.0

**Andrés Cuervo Adame**

Ingeniería Técnica en Informática de Gestión

**Consultor:** Roberto Ramírez Vique

Junio 2013

## Resumen

Encuadrado en el trabajo fin de carrera correspondiente a la ingeniería técnica en informática de gestión, este proyecto consiste en el desarrollo de una aplicación iOS, es decir, una aplicación para dispositivos móviles funcionales bajo este sistema operativo, más concretamente aquellos dispositivos con versiones 6.0 o superiores de este sistema.

La aplicación tiene como objetivo la integración de los distintos perfiles en varias redes sociales pertenecientes a los contactos almacenados en el dispositivo móvil del usuario. Gracias a esto, podremos tener un conocimiento claro de las últimas interacciones que un contacto a realizado con determinadas redes sociales.

Como añadido, y si la red social lo permite, localizaremos las últimas interacciones de los contactos con dicha red, mostrando su posición en un mapa, a partir de la localización actual del usuario.

# Índice de contenidos

Resumen.....	2
Índice de contenidos.....	3
Índice de Ilustraciones.....	6
Introducción.....	7
Descripción del proyecto.....	7
Objetivos.....	8
Riesgos.....	8
Planificación.....	9
Ciclo de vida.....	9
Detalle de actividades.....	10
Fase Previa de Análisis y Planificación.....	10
Fase de Análisis funcional.....	10
Fase de Diseño.....	10
Fase de Codificación y pruebas.....	10
Fase de Integración.....	10
Temporalización del proyecto.....	11
Los hitos que se deben cumplir en cada fase son:.....	12
Análisis funcional.....	13
Requisitos iniciales.....	13
Requisitos funcionales.....	14
Autenticación en redes sociales.....	14
Lectura de datos del dispositivo.....	14
Datos de redes sociales.....	14
Visualización de contactos.....	14
Geolocalización.....	14
Actualización de información.....	15
Requisitos no funcionales.....	16
Interfaz.....	16
Dispositivos.....	16
Conectividad.....	16
Redes sociales.....	16
Control de errores.....	16
Productos obtenidos.....	17
Planificación del proyecto.....	17
Análisis de requerimientos.....	17
Productos desarrollados.....	17
Memoria.....	17
Presentación virtual.....	17
Diagrama de casos de uso.....	18

---

Se especifica los casos de uso de la aplicación a partir de los requerimientos funcionales descritos anteriormente. ....	18
Actores.....	19
Descripción textual de los casos de uso.....	19
Caso de uso 1: Cargar datos.....	19
Caso de uso 2: Cargar libreta de direcciones.....	20
Caso de uso 3: Almacenar datos.....	20
Caso de uso 4: Modificar datos.....	21
Caso de uso 5: Recuperar lista de contactos.....	21
Caso de uso 6: Seleccionar red social.....	22
Caso de uso 7: Autenticar red social.....	22
Caso de uso 8: Leer datos de red social.....	23
Caso de uso 9: Ver contactos.....	23
Caso de uso 10: Seleccionar un contacto.....	24
Caso de uso 11: Mostrar ubicaciones.....	24
Caso de uso 12. Mostrar detalle de contacto.....	25
Diseño.....	26
Arquitectura.....	26
Decisiones tecnológicas.....	28
Lenguaje de desarrollo.....	28
Sistema de comunicación con APIs externas.....	28
Acceso a datos.....	28
Geoposicionamiento.....	28
Libreta de direcciones.....	28
Diagrama de clases.....	29
Interfaz de Usuario.....	30
Pantalla de inicio.....	31
Pantalla de selección de carga de contactos.....	32
Pantalla de carga de datos.....	33
Pantalla de listado de contactos.....	34
Pantalla de autenticación Twitter.....	35
Pantalla de detalle de contacto.....	36
Pantalla de visualización en mapa.....	37
Pantalla de posicionamiento de contacto.....	38
Modelo de Datos.....	39
Conclusiones.....	40
Conocimiento de nuevas tecnologías.....	40
Iniciación en Objective C.....	40
Trabajo con API's externas.....	40
Expectativas iniciales y resultado final.....	40
Mejoras futuras.....	40
Código fuente.....	41
Bibliografía.....	42

---

## Índice de Ilustraciones

Ilustración 1 - Ciclo de vida iterativo.....	8
Ilustración 2 - Diagrama de Gantt de la planificación del proyecto.....	10
Ilustración 3- Diagrama de casos de uso.....	17
Ilustración 4: Cocoa Touch - Capas Aplicación iPhone.....	26
Ilustración 5 - Diagrama de clases.....	28
Ilustración 6- Esquema de navegación.....	29
Ilustración 7 - Pantalla de inicio.....	30
Ilustración 8- Pantalla de selección de carga de datos.....	31
Ilustración 9- Pantalla de carga de datos.....	32
Ilustración 10- Pantalla de listado de contacto.....	33
Ilustración 11- Pantalla de autenticación Twitter.....	34
Ilustración 12- Pantalla de detalle de contacto.....	35
Ilustración 13- Pantalla de visualización en mapa.....	36
Ilustración 14: Pantalla de posicionamiento de contacto.....	37
Ilustración 15- Modelo de datos.....	38

# Introducción

## *Descripción del proyecto*

En la actualidad, la gran invasión de información a la que estamos sometidos a través las redes sociales, hace que muchas veces perdamos el control sobre quién dice qué y dónde lo dice. Esta dispersión de contactos en estas redes sociales, hace que incluso no consigamos distinguirlos y en muchas ocasiones tenemos información por duplicado del mismo contacto.

Con esta aplicación, encuadrada dentro del trabajo fin de carrera de la ingeniería técnica en informática de gestión, se pretende poder obtener información de todos nuestros contactos a través del uso de los datos almacenados en las distintas redes sociales, evitando la duplicidad de estos contactos.

El aplicativo, al estar basado en tecnología móvil y gracias a que muchas redes sociales actuales disponen de servicios de localización, nos permitirá visualizar en un mapa, la posición de nuestros contactos en el momento que hayan realizado interacciones con alguna de las redes sociales de las que forma parte, enviando mensajes, subiendo fotos, etc.

La aplicación se realizara bajo Objective-C a través del SDK 6.0 de iOS, por lo que será compatible exclusivamente con los dispositivos que dispongan de este sistema operativo instalado. Esta enfocada a los dispositivos móviles encuadrados en la categoría de iPhone, si bien es ejecutable en iPad, no estará el uso optimizado para estos últimos.

## ***Objetivos***

Los objetivos que se pretenden conseguir mediante la realización de este trabajo son los siguientes:

- Poner en práctica conocimientos adquiridos a lo largo de la carrera sobre ingeniería del software y técnicas de desarrollo.
- Profundizar en el desarrollo de aplicaciones móviles para dispositivos Apple.
- Dar respuesta a una necesidad a través de un proyecto informático.
- Mejorar habilidades de planificación, estimación, desarrollo y presentación de proyectos.

## ***Riesgos***

El mayor riesgo que se presenta en el desarrollo de la aplicación, es cualquier cambio en las especificaciones de las redes sociales a las que se accede para obtener los datos relativos a los contactos.

Del mismo modo, al almacenarse los datos de los contactos en el dispositivo de forma síncrona, puede ocurrir que no se tengan todas las actualizaciones almacenadas, con lo que posibles cambios no sean visibles, existiendo una inconsistencia en el valor de los datos.

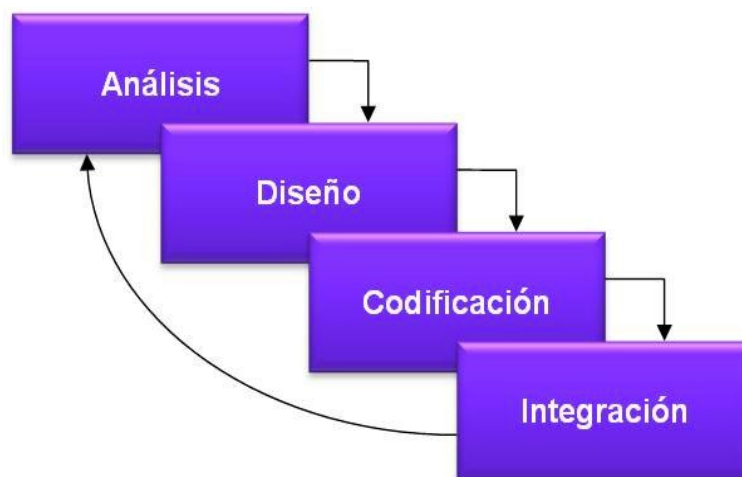
Como queremos mostrar en un mapa la posición desde la que un contacto interacciona con determinadas redes sociales, necesitamos que la red social permita el posicionamiento de sus interacciones, y que además, el contacto tenga activada la posibilidad de realizar esa publicación con la localización incluida.

El usuario de la aplicación debe dar permisos de forma explícita a la aplicación a través de su interfaz para la lectura de datos de alguna de las redes sociales en su nombre.

## Planificación

### *Ciclo de vida*

El proyecto ha sido realizado siguiendo el ciclo de vida iterativo, adaptándolo a las necesidades del trabajo fin de carrera. Durante la realización de las diferentes fases, se ha ido generando y entregando diferente documentación que ha sido condensada en esta memoria final del proyecto.



#### **Ilustración 1 - Ciclo de vida iterativo**

*En la práctica casi todos los proyectos no cubren todos los requisitos en su primera versión, por lo que una vez realizada una entrega se vuelve a abordar una versión posterior que incluye nuevas funcionalidades. Asumir que esto es así desde un principio es lo que plantea el **ciclo de vida iterativo**.*



## ***Detalle de actividades***

Las secciones siguientes nos indican como se desarrollará la primera iteración del proyecto, informando en cada etapa de las principales actividades realizadas, a fin de elaborar un primer producto que cumpla los requisitos iniciales, dentro de los plazos que se definen y marca el trabajo fin de carrera.

### **Fase Previa de Análisis y Planificación**

Se realiza en esta fase la visión global del sistema a crear, así como preparar la planificación temporal de las actividades que se llevarán a cabo, imponiendo hitos de entrega correspondientes a las necesidades del calendario propuesto.

### **Fase de Análisis funcional**

Se detallan las necesidades a cubrir por el software, desde el punto de vista funcional. No se entra en detalles de implementación o tecnología a utilizar. En esta fase se desarrolla y crea la documentación correspondiente a los casos de uso.

### **Fase de Diseño**

Se define aquí como debe ser implementado el sistema para alcanzar los objetivos planteados anteriormente, incluyendo aspectos concretos sobre las tecnologías a utilizar.

### **Fase de Codificación y pruebas**

En esta fase se realiza la codificación de la solución, añadiendo una batería de pruebas unitarias que ayuden a contrastar la calidad del código. De esta fase se obtiene en primera instancia un prototipo funcional que muestra la forma de navegación de la aplicación, sin funcionalidad aparente.

### **Fase de Integración**

Dentro de este proyecto esta fase podemos incluirla en la parte de entrega final, donde se genera la documentación que se entrega, y que consta de esta memoria, una presentación virtual, y el código de la aplicación finalizada.

## Temporalización del proyecto

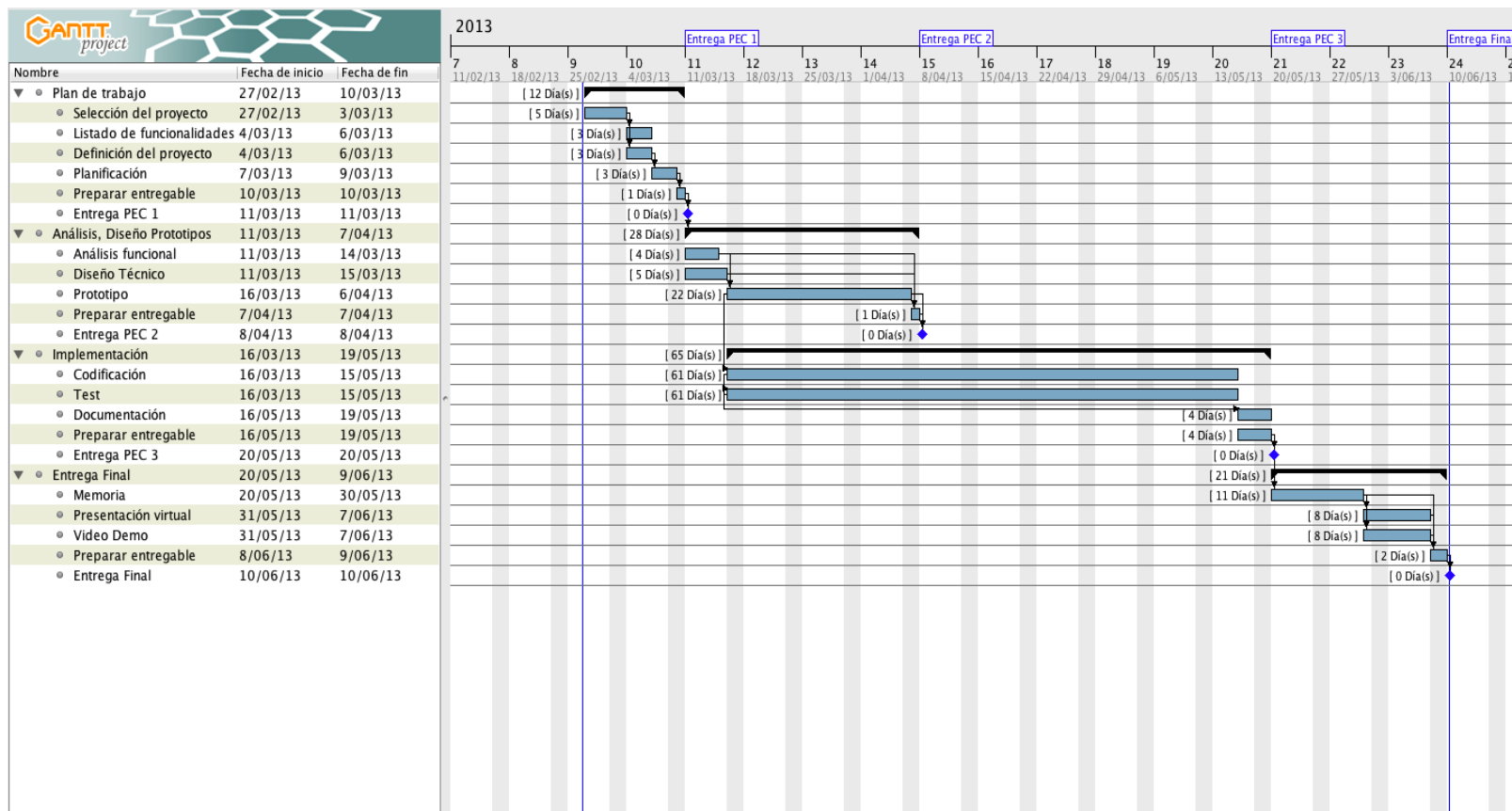


Ilustración 2 - Diagrama de Gantt de la planificación del proyecto

Los hitos que se deben cumplir en cada fase son:

<b>Fase</b>	<b>Periodo</b>
Plan de trabajo	27/02/2013 – 10/03/2013
Análisis y diseño de prototipos	11/03/2013 – 07/04/2013
Implementación	16/03/2013 – 19/05/2013
Entrega final	20/05/2013 – 03/06/2013

## **Análisis funcional**

Detallamos a continuación los requisitos necesarios para llevar a cabo la aplicación, así como la descomposición de los mismos en partes más simples del problema que planteamos resolver, convirtiendo una idea abstracta en factores tangibles simplificados que va a facilitar el diseño de la aplicación.

### ***Requisitos iniciales***

Como vimos en la descripción del proyecto, nuestra aplicación debe permitir al usuario de la misma, unificar los datos de los contactos almacenados en el dispositivo, recuperando la información de las redes sociales Facebook y Twitter. Ambas redes sociales disponen de una API propia gracias a la cual podemos obtener diferentes datos de los usuarios de la misma, así como la localización geográfica de las últimas interacciones realizadas sobre la misma.

Tenemos por tanto dos tipos de requisitos que tener en cuenta, aquellos que serán funcionales, que nos permitirán conocer que debe y hacer la aplicación y como. El otro tipo de requisitos serán los requisitos no funcionales, que nos indicarán de que manera y a través de que herramientas se van a alcanzar los requisitos iniciales.

## **Requisitos funcionales**

### **Autenticación en redes sociales**

El sistema debe permitir autenticar al usuario de la aplicación en las redes sociales indicadas. Esta autenticación se llevará a cabo mediante la API de cada red social, dejando registrada esta autenticación para no tener que hacer solicitudes cada vez que se inicie la aplicación.

### **Lectura de datos del dispositivo**

El sistema debe permitir leer la información de los contactos almacenados en el dispositivo. De estos contactos será de los que la aplicación debe obtener la información de las diferentes redes sociales.

### **Datos de redes sociales**

El sistema debe localizar y almacenar la información que se recupere desde las redes sociales en el dispositivo. Gracias a la información recuperada, se debe ampliar la información de cada contacto en la agenda, almacenándose para evitar búsquedas posteriores.

### **Visualización de contactos**

El sistema debe mostrar datos relativos a los contactos del dispositivo, añadiendo la información obtenida de las redes sociales. Se debe poder acceder a las últimas interacciones que el contacto haya realizado a través de las redes sociales modificándose la información ya preexistente en el dispositivo.

### **Geolocalización**

El sistema debe permitir visualizar en un mapa donde se han realizado las últimas interacciones de los contactos del dispositivo con las redes sociales. Podemos diferenciar las interacciones realizadas en cada red social, por ejemplo, si un contacto ha realizado una interacción a través de Twitter en una determinada localización y posteriormente a través de Facebook en otra localización, se mostrarán ambas.

### **Actualización de información**

El sistema debe permitir actualizar los datos de los contactos desde las distintas redes sociales de forma síncrona, a petición del usuario desde el dispositivo. De este modo, el usuario elegirá el mejor momento para hacerlo (disponibilidad de descarga de datos, conexión wi-fi, etc.)

## Requisitos no funcionales

### Interfaz

La interfaz debe ser amigable y sencilla de utilizar, con accesos directos desde la página principal a las diferentes funcionalidades de la aplicación. De este modo, podremos navegar directamente a los distintos elementos sin necesidad de pasar por menús no necesarios.

### Dispositivos

La aplicación está diseñada para dispositivos Apple que sean capaces de correr el sistema operativo iOS 6.0 o superiores, debiendo disponer de GPS integrado. La aplicación hace uso de la agenda de contactos del dispositivo, por lo que se deben tener contactos almacenados en la misma para que tenga utilidad real.

### Conectividad

Para recuperar la información desde las redes sociales, el usuario debe tener una conexión a internet, con la posibilidad de descargar datos desde la red. Si no existe esta conectividad, no se actualizará la información de los contactos.

### Redes sociales

Si no se tiene cuenta en alguna de las redes sociales que utiliza la aplicación (en este caso, Facebook y Twitter) y estas cuentas no están dadas de alta en el dispositivo móvil, no se podrá descargar información de estas redes, pues hace uso intensivo de los amigos y seguidores que el usuario disponga.

### Control de errores

El sistema controlará los posibles errores que puedan surgir durante su utilización, para no llevar al usuario a situaciones desagradables y que hagan el sistema robusto.

## **Productos obtenidos**

Durante el transcurso del trabajo fin de carrera de la Ingeniería Técnica de Informática de Gestión se han elaborado los siguientes documentos:

### **Planificación del proyecto**

Se indican los diferentes hitos a lograr para dar solución a los objetivos planteados en la toma de requisitos del sistema.

### **Análisis de requerimientos**

Este documento describe las funcionalidades que se necesitan desarrollar para cumplir con los objetivos solicitados en la planificación del proyecto.

### **Productos desarrollados**

Código de aplicación para dispositivo móvil que disponga de sistema operativo iOS 6 ó superior.

### **Memoria**

El documento actual, donde se resumen los trabajos realizados.

### **Presentación virtual**

Video donde se muestra de forma virtual una breve introducción a la solución planteada.



## Diagrama de casos de uso

Se especifica los casos de uso de la aplicación a partir de los requerimientos funcionales descritos anteriormente.

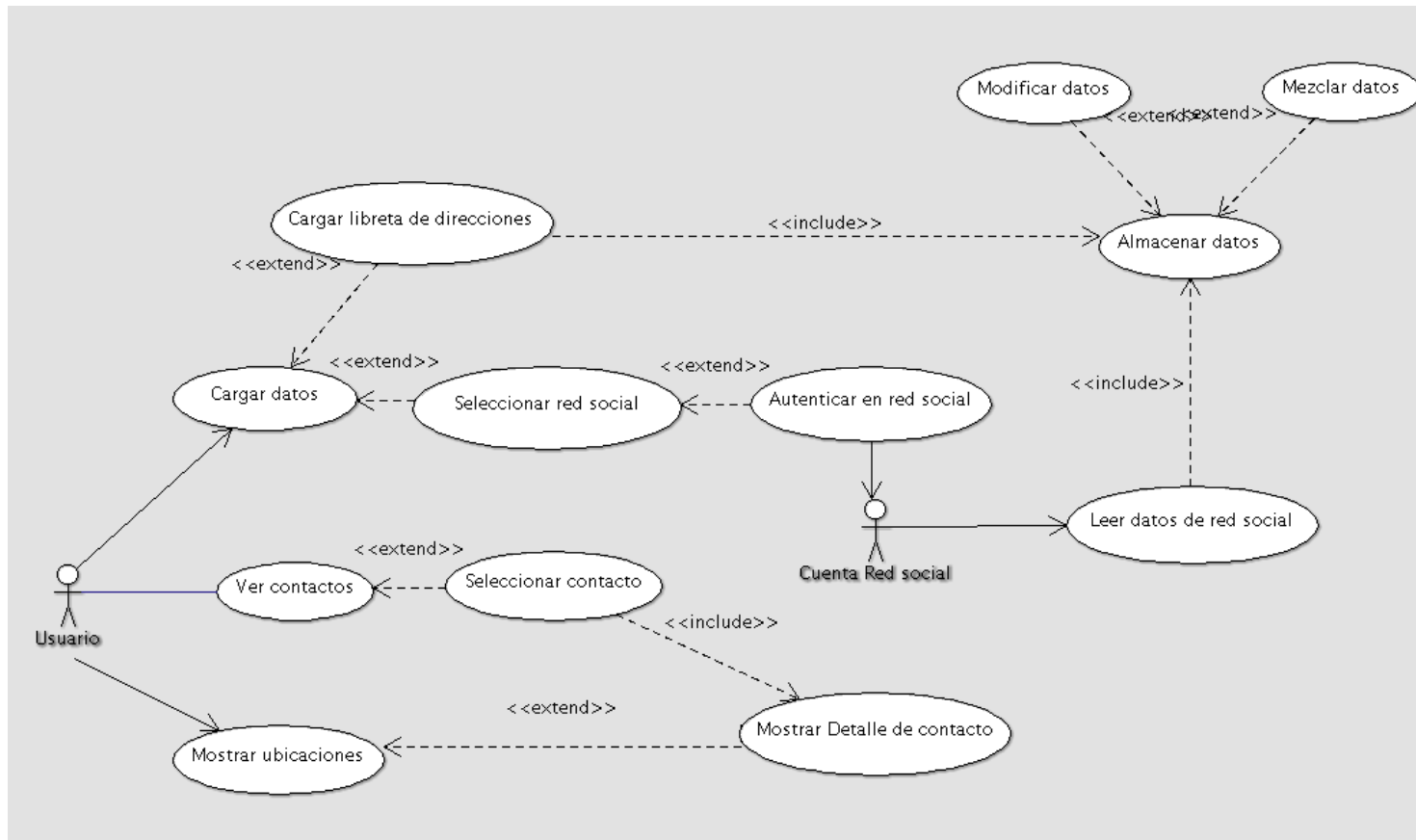


Ilustración 3- Diagrama de casos de uso

## Actores

Como podemos observar, el sistema distinguirá dos tipos de actores diferentes:

- **Usuario.** Representa todos los usuarios que a través de las interfaces pueden interactuar con la aplicación
- **Cuenta red social.** Representa la cuenta de una red social gracias a la cual podremos obtener información de las mismas.

## Descripción textual de los casos de uso

A continuación presentamos la descripción textual de los casos de uso de la aplicación.

### *Caso de uso 1: Cargar datos*

<b>Identificador</b>	UC001
<b>Nombre</b>	Cargar datos
<b>Resumen</b>	El actor selecciona la acción de cargar datos
<b>Requisitos asociados</b>	Ninguno
<b>Actor(es)</b>	Usuario Aplicación
<b>Descripción</b>	El actor selecciona la opción de cargar datos del menú principal, pasando al siguiente paso -
<b>Precondición</b>	Ninguna
<b>Postcondición</b>	Ninguna

*Caso de uso 2: Cargar libreta de direcciones*

<b>Identificador</b>	UC002
<b>Nombre</b>	Cargar libreta de direcciones
<b>Resumen</b>	El actor selecciona la opción cargar datos desde la libreta de direcciones del menú. Se realiza una lectura de los contactos almacenados en el dispositivo dentro de la libreta de direcciones.
<b>Requisitos asociados</b>	Ninguno
<b>Actor(es)</b>	Usuario Aplicación
<b>Descripción</b>	Se realiza una lectura de los contactos almacenados en el dispositivo dentro de la libreta de direcciones.
<b>Precondición</b>	Los datos de los contactos de la libreta de direcciones están disponibles para trabajar con ellos desde la aplicación.

*Caso de uso 3: Almacenar datos*

<b>Identificador</b>	UC003
<b>Nombre</b>	Almacenar datos
<b>Resumen</b>	Los datos de contacto son almacenados en el dispositivo.
<b>Requisitos asociados</b>	Modificar datos Mezclar datos
<b>Actor(es)</b>	Usuario Aplicación Usuario Red Social
<b>Descripción</b>	El actor actualiza los datos correspondientes a los contactos de su agenda con los valores obtenidos desde las redes sociales o libreta de contactos del dispositivo.
<b>Precondición</b>	El dispositivo debe tener conectividad hacia las redes sociales y datos en la agenda de contactos.
<b>Postcondición</b>	Se crea una tabla con los valores de los contactos almacenados en ella.

*Caso de uso 4: Modificar datos*

<b>Identificador</b>	UC004
<b>Nombre</b>	Modificar datos
<b>Resumen</b>	Los datos previamente almacenados en base de datos se modifican por los valores nuevos
<b>Requisitos asociados</b>	Almacenar datos
<b>Actor(es)</b>	Usuario Aplicación
<b>Descripción</b>	Se modifican los datos de un contacto de la base de datos si existía previamente en la misma. Los nuevos datos son los recibidos de la operación de carga.
<b>Precondición</b>	Datos previamente almacenados en la base de datos
<b>Postcondición</b>	Los valores del contacto almacenado en la base de datos están actualizados con los valores de la última lectura de datos de la red social o la libreta de contactos del dispositivo.

*Caso de uso 5: Recuperar lista de contactos*

<b>Identificador</b>	UC005
<b>Nombre</b>	Mezclar datos
<b>Resumen</b>	Los datos almacenados correspondientes a un contacto, son unidos en un único registro.
<b>Requisitos asociados</b>	Almacenar datos
<b>Actor(es)</b>	Usuario Aplicación
<b>Descripción</b>	Si los valores de un contacto determinado ya existen en la base de datos, se completa la información del mismo con la actualización obtenida
<b>Precondición</b>	El contacto leído ya existe en la base de datos y no dispone de toda la información almacenada
<b>Postcondición</b>	Un único contacto con información adicional, proveniente de diferente fuente.

*Caso de uso 6: Seleccionar red social*

<b>Identificador</b>	UC006
<b>Nombre</b>	Seleccionar red social
<b>Resumen</b>	El usuario selecciona desde que red social quiere descargar información.
<b>Requisitos asociados</b>	Cargar datos Autenticar en red social
<b>Actor(es)</b>	Usuario Aplicación
<b>Descripción</b>	El actor selecciona desde que red social quiere descargar información para actualizar la base de datos de la aplicación
<b>Precondición</b>	El actor ha seleccionado la opción de cargar datos del menú principal.
<b>Postcondición</b>	Conocemos desde que sistema queremos descargar los contactos.

*Caso de uso 7: Autenticar red social*

<b>Identificador</b>	UC007
<b>Nombre</b>	Autenticar red social
<b>Resumen</b>	El usuario selecciona con que cuenta se quiere autenticar en la red social, si fuera necesario.
<b>Requisitos asociados</b>	Seleccionar red social
<b>Actor(es)</b>	Usuario Aplicación Cuenta Red Social
<b>Descripción</b>	El actor recupera se autentica dentro de la red social que haya elegido el caso UC006 si fuera necesario.
<b>Precondición</b>	El usuario ha seleccionado en que red social quiere autenticarse
<b>Postcondición</b>	Usuario autenticado en red social seleccionada, si fuera posible. En caso contrario, obtendremos un error

*Caso de uso 8: Leer datos de red social*

<b>Identificador</b>	UC008
<b>Nombre</b>	Leer datos de red social
<b>Resumen</b>	Se recuperan los datos de los contactos de la red social seleccionada si estamos autenticados en ella
<b>Requisitos asociados</b>	Autenticar en red social Almacenar datos
<b>Actor(es)</b>	Usuario Aplicación Cuenta Red Social
<b>Descripción</b>	Se recupera la información de los contactos que el usuario tiene en la red social seleccionada. Estos datos se almacenarán en la base de datos del sistema
<b>Precondición</b>	El usuario debe estar autenticado en la red social seleccionada
<b>Postcondición</b>	Listado de contactos pertenecientes a la red social seleccionada, disponibles para ser almacenados en la base de datos del sistema

*Caso de uso 9: Ver contactos*

<b>Identificador</b>	UC009
<b>Nombre</b>	Ver contactos
<b>Resumen</b>	Se visualiza un listado con los contactos almacenados en la base de datos
<b>Requisitos asociados</b>	Ninguno
<b>Actor(es)</b>	Usuario Aplicación
<b>Descripción</b>	Se visualizara un listado con los contactos almacenados en la base de datos del sistema.
<b>Precondición</b>	La aplicación debe haber almacenado previamente datos de contactos en la base de datos del sistema
<b>Postcondición</b>	Ninguna

*Caso de uso 10: Seleccionar un contacto*

<b>Identificador</b>	UC010
<b>Nombre</b>	Seleccionar contacto
<b>Resumen</b>	Se selecciona un contacto desde la lista de contactos
<b>Requisitos asociados</b>	Ver contactos Mostrar detalle contacto
<b>Actor(es)</b>	Usuario Aplicación
<b>Descripción</b>	Se selecciona un contacto desde el listado previo con los contactos almacenados en la base de datos del sistema
<b>Precondición</b>	El contacto debe estar previamente almacenado en la base de datos del sistema
<b>Postcondición</b>	Tenemos un contacto seleccionado, del que veremos la información asociada

*Caso de uso 11: Mostrar ubicaciones*

<b>Identificador</b>	UC011
<b>Nombre</b>	Mostrar ubicaciones
<b>Resumen</b>	Se muestra un mapa con la localización de las últimas interacciones de los contactos con las redes sociales
<b>Requisitos asociados</b>	Mostrar detalle de contacto
<b>Actor(es)</b>	Usuario Aplicación
<b>Descripción</b>	Se muestra un mapa con la posición de las últimas interacciones de los contactos con las redes sociales, mostrando un marcador en esa posición.
<b>Precondición</b>	La interacción con la red social por parte de los contactos debe tener activa la opción de geoposicionamiento.
<b>Postcondición</b>	Ninguna

*Caso de uso 12. Mostrar detalle de contacto*

<b>Identificador</b>	UC012
<b>Nombre</b>	Mostrar detalle de contacto
<b>Resumen</b>	Se muestra el detalle del contacto seleccionado.
<b>Requisitos asociados</b>	Mostrar ubicaciones Seleccionar contacto
<b>Actor(es)</b>	Usuario Aplicación
<b>Descripción</b>	Muestra el detalle con la información existente en base de datos, del contacto seleccionado previamente
<b>Precondición</b>	EL contacto debe estar almacenado en la base de datos del sistema
<b>Postcondición</b>	Ninguna



## Diseño

### *Arquitectura*

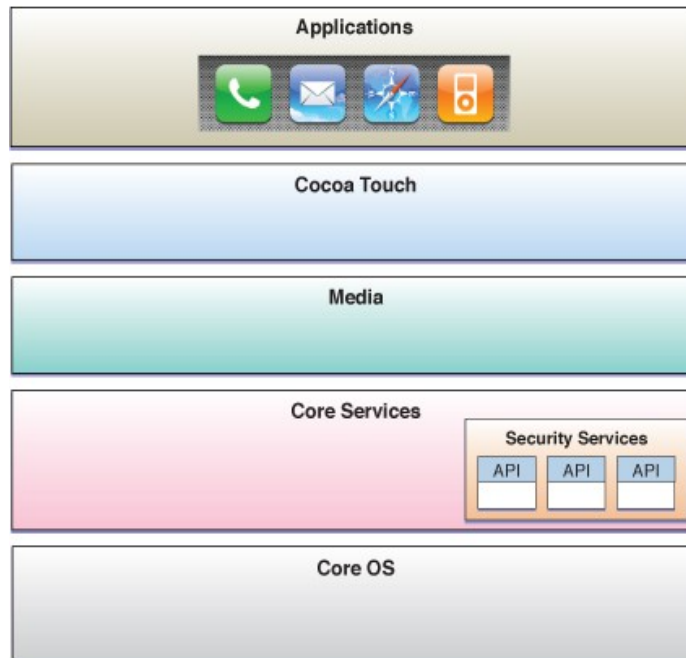
La solución hace un uso intensivo de las API de las redes sociales seleccionadas (en nuestro caso Facebook y Twitter), comunicándose con ellas a través de diferentes Framework, utilizando la tecnología json como vehículo de comunicación.

La información obtenida de las redes sociales, siempre a petición del usuario, se almacena en el dispositivo en una base de datos, para evitar continuas peticiones a través de internet a las API de las redes sociales (que normalmente tienen limitado el número de peticiones a realizar en un corto periodo de tiempo).

Para presentar la información dentro del mapa de ubicaciones, el dispositivo utiliza geoposicionamiento, utilizando para ello el hardware del Smartfone (tecnología GPS)

En iOS 6, la arquitectura de sistema es muy similar a la encontrada en MAC OS X, siendo su kernel una variación de este último. Por encima del kernel están las capas de servicios utilizadas para implementar aplicaciones en la plataforma. Esta estructura de capas aumenta las posibilidades a la hora de implementar código. Estas interfaces están basadas en C e incluyen tecnologías como Core Foundation, SQLite y acceso a hilos POSIX y sockets UNIX entre otros.

El punto de partida para cualquier proyecto es la capa Cocoa Touch y el Framework UIKit en particular. Cuando se plantea qué tecnologías adicionales utilizar, se recomienda empezar por los frameworks de nivel más alto e ir descendiendo por las capas conforme sea necesario. Los frameworks de más alto nivel hacen sencillo dar soporte a comportamientos del sistema estándar, con el mínimo esfuerzo por parte del desarrollador



*Ilustración 4: Cocoa Touch - Capas Aplicación iPhone*

## ***Decisiones tecnológicas***

Justificamos las decisiones tomadas para el desarrollo de la aplicación, respecto a que tecnologías utilizar para resolver los distintos aspectos de la misma:

### **Lenguaje de desarrollo**

Para poder desarrollar sobre los diferentes frameworks utilizados y siendo el lenguaje que mejor se adapta a ellos, utilizamos Objective C, utilizando como IDE el que provee Apple: XCode en su versión 4.6.2

### **Sistema de comunicación con APIs externas**

Tanto la API de Facebook, como la API de Twitter utilizan para devolver los resultados de las consultas que realizamos contra ellas, el lenguaje JSON. Las llamadas y respuestas de estas APIs, son asíncronas, por lo que utilizamos el framework AFNetworking, que facilita estas tareas de forma considerable. En algunos casos, es mucho más simple utilizar el framework Social, que Apple ofrece desde la versión 5.0 de su SDK

### **Acceso a datos**

La base de datos que utilizamos y almacenamos en el dispositivo es una SQLite, ya que además de su sencillez de uso, permite almacenar la información con un formato reducido, muy apropiado para sistemas móviles.

El acceso a los datos, se ha realizado mediante el framework MagicalRecord, que trabaja una capa por encima de CoreData y facilita la interacción con la base de datos.

### **Geoposicionamiento**

Para este fin, utilizaremos el framework estándar de Apple, que no es otro que MapKit, el cual nos permite posicionar diferentes puntos en un mapa mediante coordenadas (latitud y longitud) así como hacer un geoposicionamiento inverso, es decir, dadas unas coordenadas, nos ofrece el nombre del lugar seleccionado (haciéndolo más comprensible por el usuario)

### **Libreta de direcciones**

Para acceder a la libreta de direcciones del dispositivo, Apple nos lo pone sencillo con el framework Account, que es el que se ha utilizado.

## Diagrama de clases

En el siguiente diagrama se identifican las clases de las que constará la aplicación, así como sus interrelaciones. No se indican las clases necesarias para implementar la interfaz, ni la conectividad con las distintas redes sociales, que se harán con el framework correspondiente.

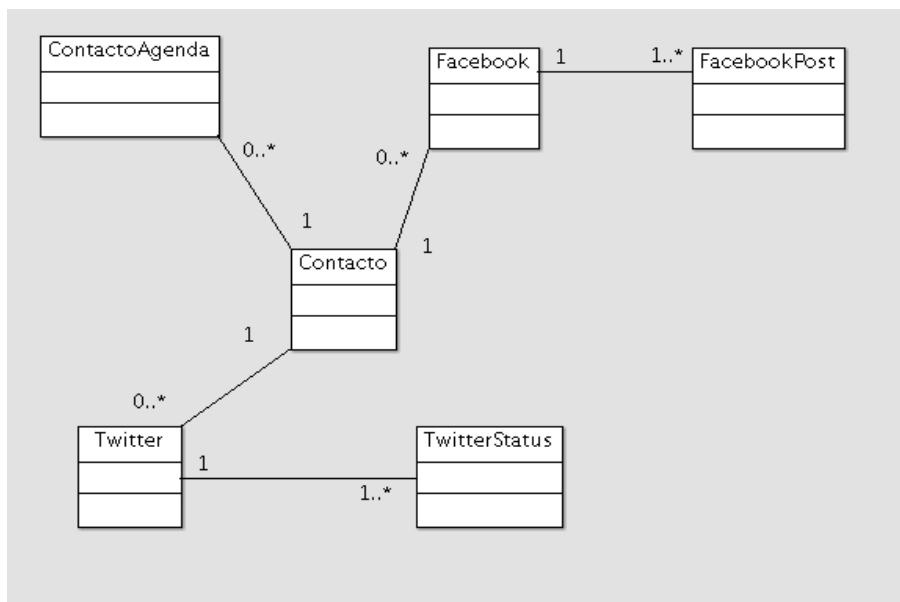


Ilustración 5 - Diagrama de clases

## Interfaz de Usuario

Aquí tenemos el esquema de navegación entre las diferentes pantallas. La construcción de esta interface, se realiza mediante el uso de XCode, a través de archivos xlib.

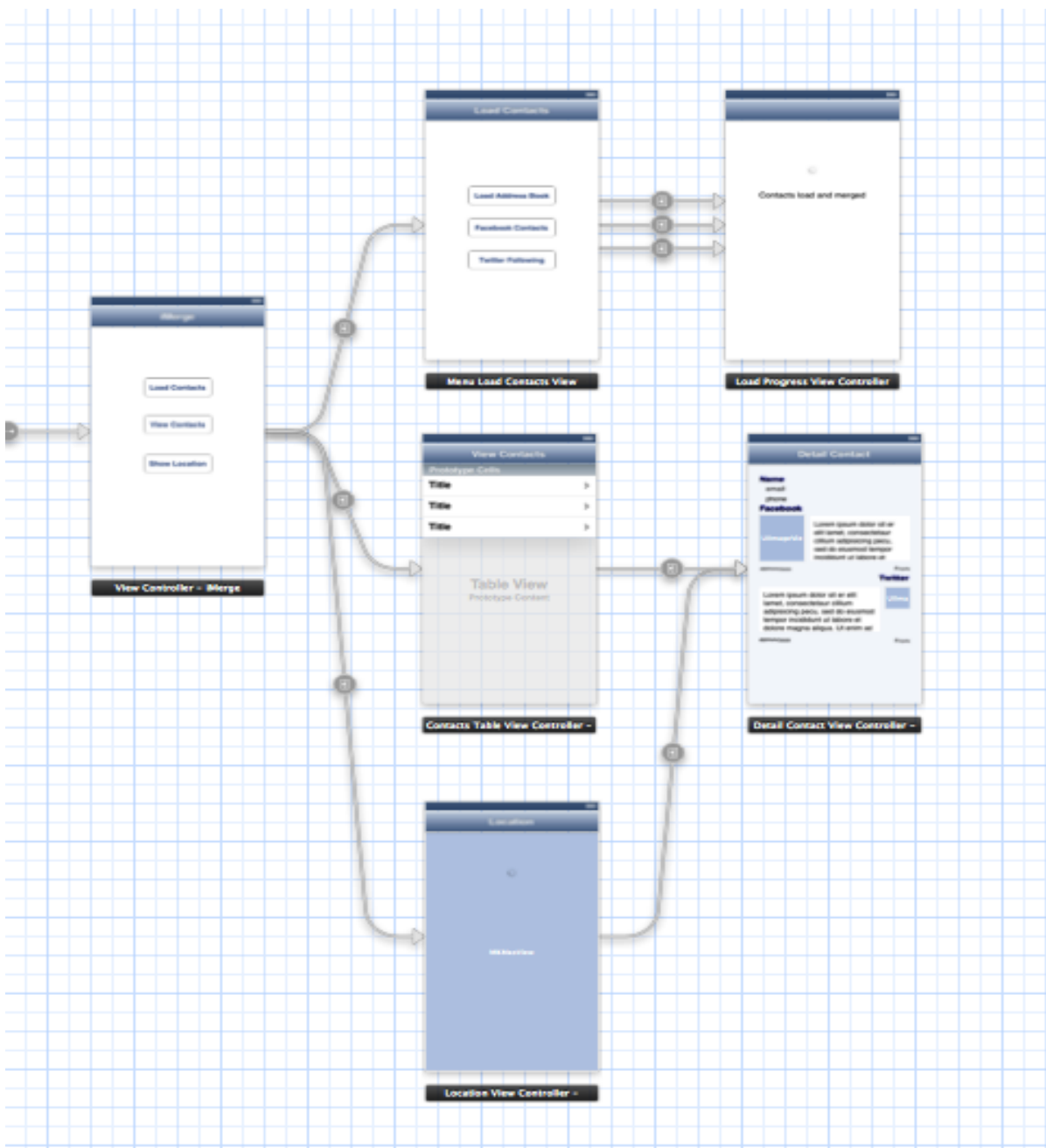
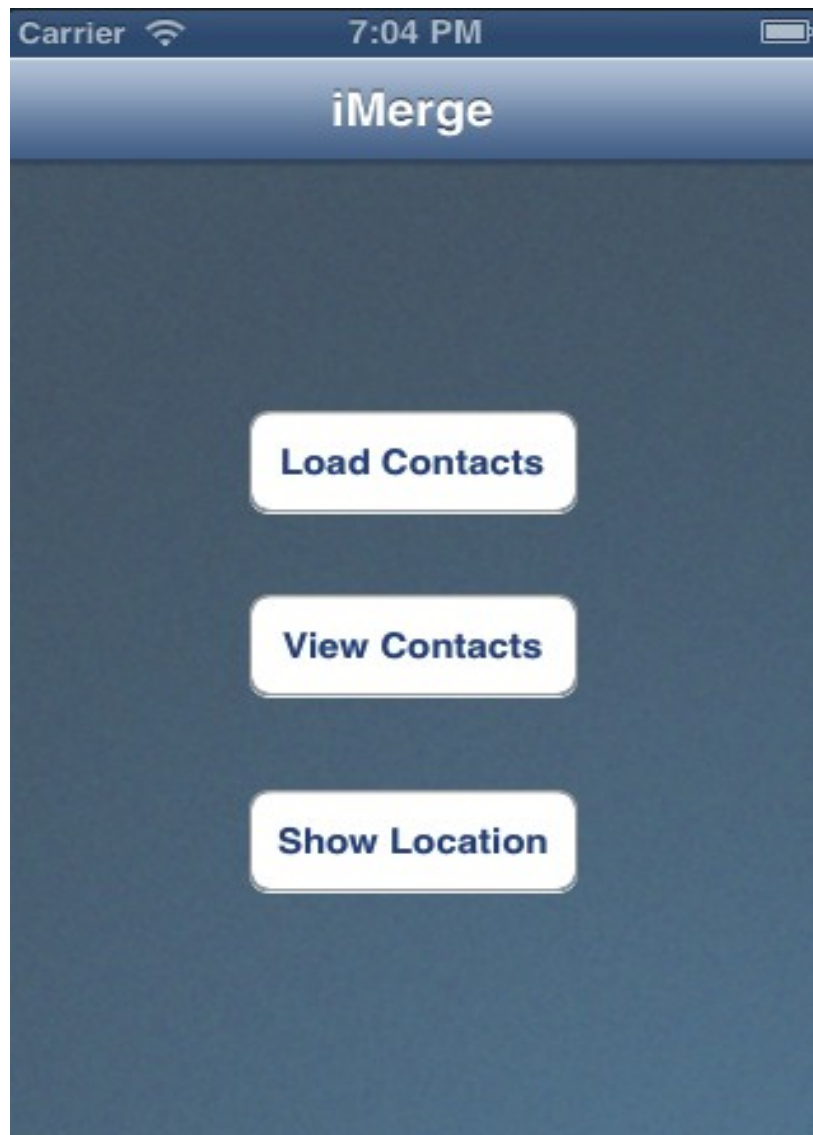


Ilustración 6- Esquema de navegación

A continuación se describen las diferentes pantallas del interfaz de usuario y sus funcionalidades

## **Pantalla de inicio**

Aquí se encuentra el menú principal desde el que tenemos acceso al resto de la aplicación. Esta compuesto por tres botones, uno para cargar los datos de los contactos desde las diferentes redes sociales, otro para mostrar un listado de los contactos que hayamos almacenado previamente y otro para ubicar en un mapa los contactos que tengamos en la agenda y hayan compartido sus interacciones con las redes sociales, utilizando geoposicionamiento.



**Ilustración 7 - Pantalla de inicio**

## Pantalla de selección de carga de contactos

Aquí tenemos un menú, gracias al cual podremos elegir desde que red social queremos descargarnos la información. En principio presenta tres botones, el primero para descargar la información desde la libreta de direcciones del dispositivo, otro para descargar datos de los amigos de Facebook y el último para que la descarga de datos se produzca desde los followers de Twitter

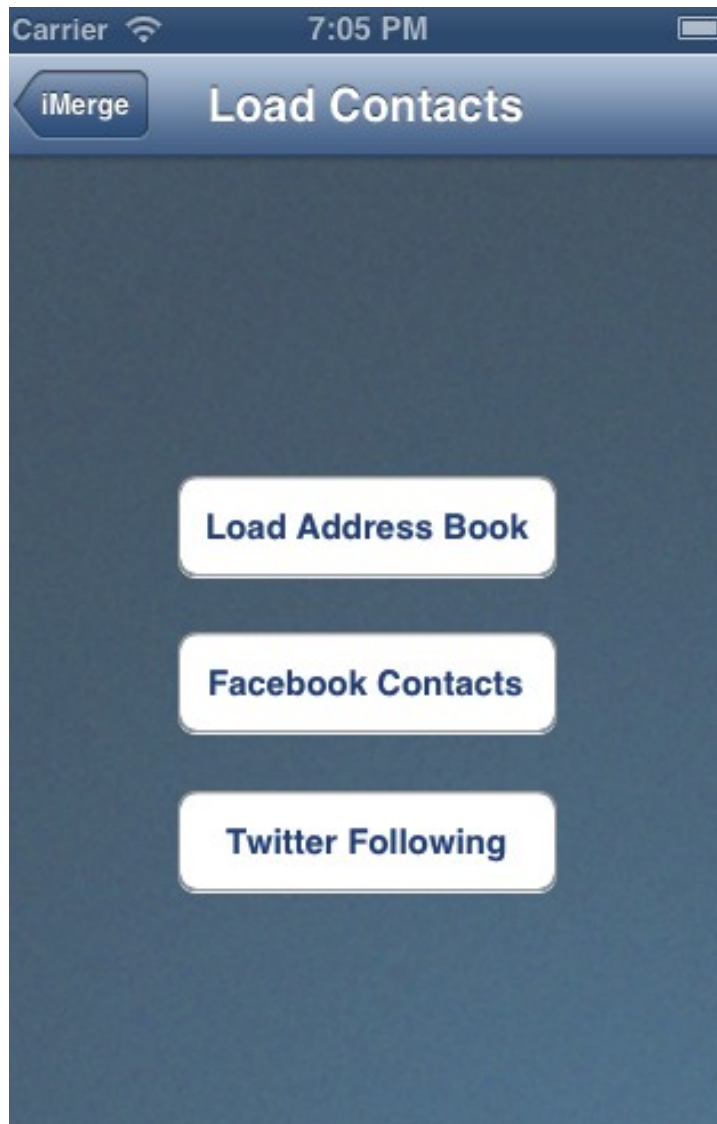
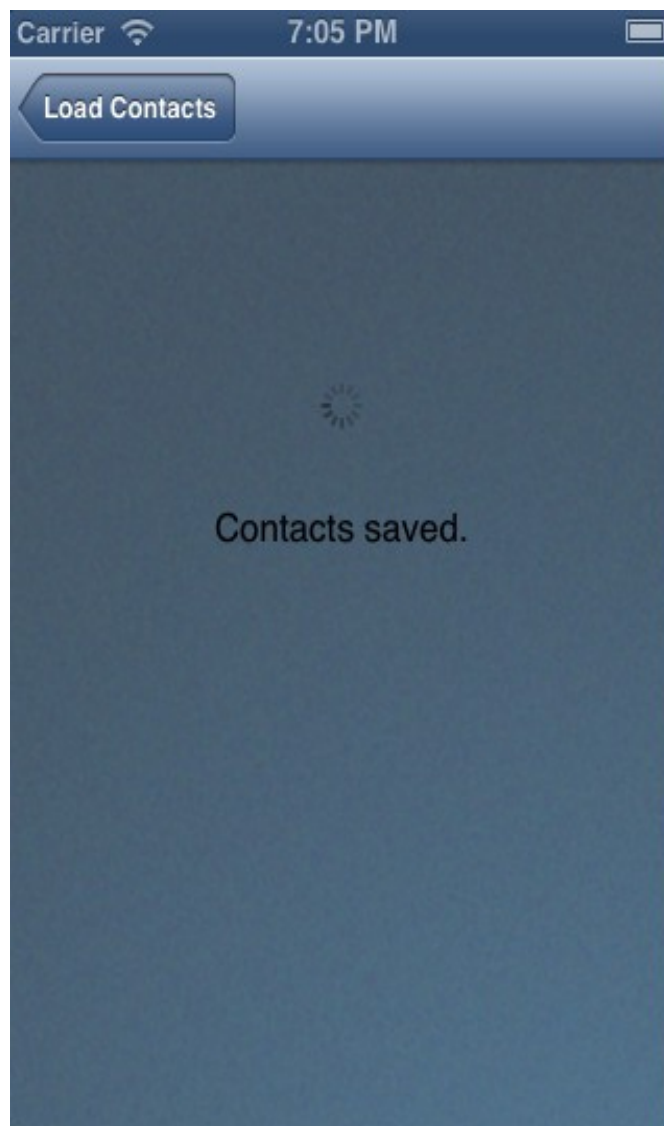


Ilustración 8- Pantalla de selección de carga de datos

### **Pantalla de carga de datos**

Vemos un indicador del proceso de carga de los datos desde la red social seleccionada en el menú anterior. Al finalizar la carga de esos datos y su almacenamiento (y mezclado si fuera necesario) en la base de datos, nos indicará el fin del proceso.



**Ilustración 9- Pantalla de carga de datos**



## Pantalla de listado de contactos

Nos muestra un listado con todos aquellos contactos que tenemos en la base de datos almacenados previamente. Se puede navegar hacia abajo en la tabla, arrastrando el dedo por la pantalla. Se acompaña cada contacto con la imagen que utiliza en la redes sociales.



Ilustración 10- Pantalla de listado de contacto

## Pantalla de autenticación Twitter

Cuando queremos acceder por primera vez a la red social Twitter, y disponemos de más de una cuenta dada de alta en el dispositivo para esta red, la aplicación nos preguntará con que usuario queremos conectarnos a la red social.



Ilustración 11- Pantalla de autenticación Twitter

## Pantalla de detalle de contacto

En esta pantalla obtenemos los detalles del contacto seleccionado en el listado general o a través del mapa de ubicación. Nos mostrara la información que dispongamos del contacto en la base de datos, así como el contenido de los últimos estados de las distintas redes sociales a las que pertenezca (y el sistema tenga constancia de ello)



Ilustración 12- Pantalla de detalle de contacto

### Pantalla de visualización en mapa

Aquí se visualizarán los lugares desde donde se han enviado interacciones con las redes sociales mediante una chincheta azul. El número de contactos/interacciones que se visualizan en el mapa, no corresponde al numero de contactos existente, puesto que se muestran aquellos mensajes que han utilizado geoposicionamiento. La información que pintamos aquí, no se almacena en el sistema, siendo recuperada cada vez que cargamos el mapa.



Ilustración 13- Pantalla de visualización en mapa

## Pantalla de posicionamiento de contacto

Tras pulsar en una chincheta azul del mapa, veremos a quien pertenece el mensaje geolocalizado, así como su ubicación de una forma legible. Si pulsamos sobre el icono azul, iremos al detalle del contacto, mostrándose la pantalla de detalle de contacto.



*Ilustración 14: Pantalla de posicionamiento de contacto*

## Modelo de Datos

Para el correcto desempeño de la funcionalidad de la aplicación es indispensable disponer de un sistema en el que almacenar la información. Por ese motivo, y en base al diagrama de clases diseñado anteriormente, se debe generar un modelo de datos que permita organizar la información para acceder a ella de una forma lo mas eficiente posible.

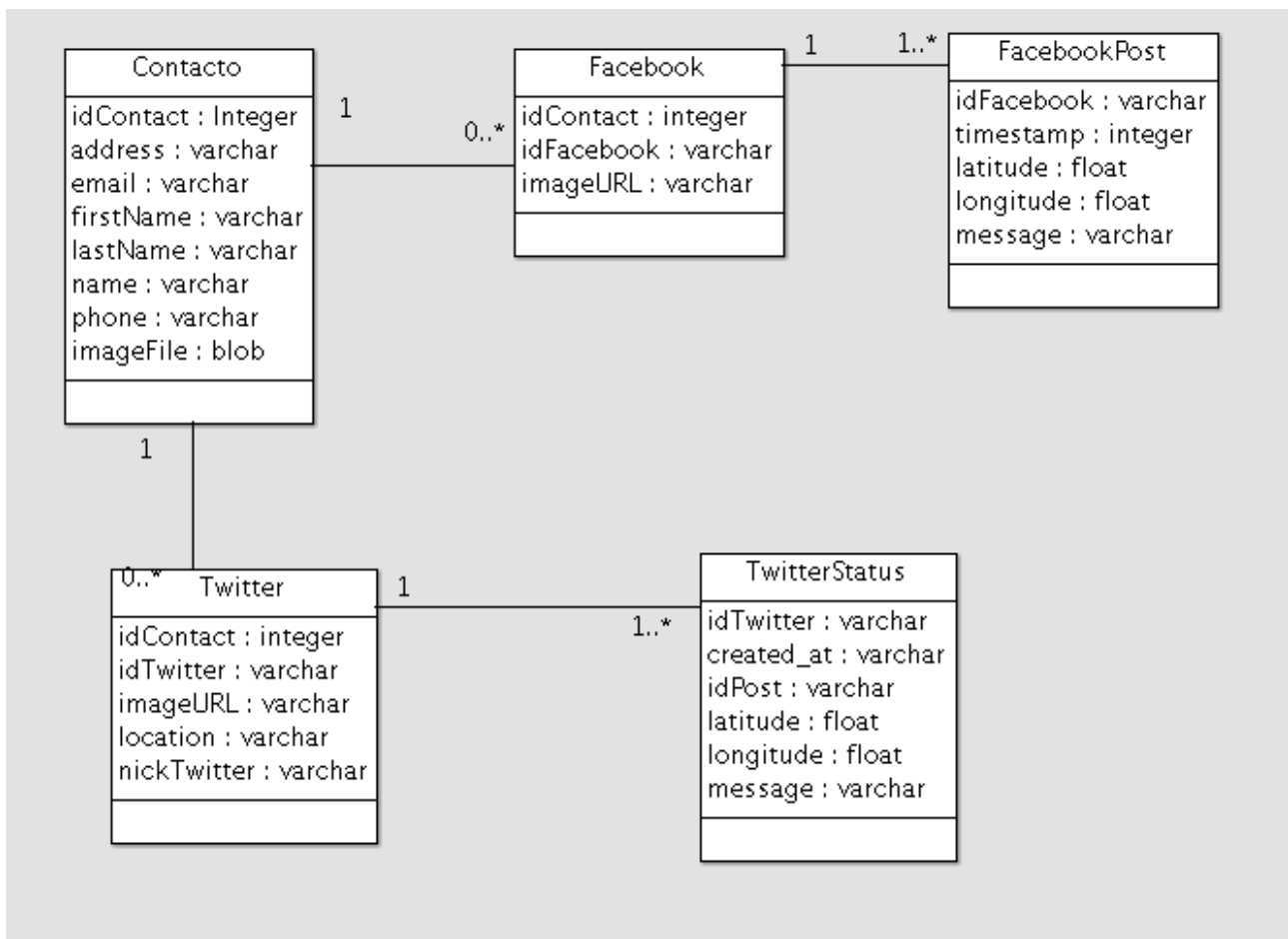


Ilustración 15- Modelo de datos

## Conclusiones

### Conocimiento de nuevas tecnologías

Este proyecto ha permitido conocer las grandes posibilidades que ofrece el desarrollo de aplicaciones móviles, siendo en la actualidad una de las competencias mas valoradas en el ámbito profesional.

### Iniciación en Objective C

Gracias a que el lenguaje de desarrollo que utilizan las aplicaciones para iPhone, están escritas en lenguaje Objective C, ha sido necesario realizar un estudio de este nuevo lenguaje, que tiene algunas características que lo hacen en algunos aspectos un tanto complicado, aunque su fundamento se encuentra en la programación orientada a objetos.

### Trabajo con API's externas

Al tener que descargar información desde las distintas redes sociales, se ha necesitado un estudio concienzudo de las API utilizadas, así como una comprensión de la forma de recibir esta información, lo que ha permitido conocer mas profundamente el funcionamiento de jSon

### Expectativas iniciales y resultado final

Si bien en un principio, la cantidad de necesidades que se pretendían cubrir con la aplicación eran muchas más de las que finalmente se han alcanzado, el resultado ha sido una aplicación robusta que permite un crecimiento posterior de una manera relativamente sencilla, gracias a la forma en la que se ha planteado y la versatilidad que tiene el lenguaje Objective C.

## Mejoras futuras

Para poder seguir avanzando en el desarrollo de la aplicación, podemos trabajar en algunos puntos que mejorarían el sistema en su conjunto.

- Ampliación del número de redes sociales desde el que poder traerse información
- Mejora en la forma de mezclar contactos. En la actualidad lo hace solamente si el nombre coincide en la red social, cosa muy poco habitual. Se debería hacer un filtrado mas fino para conseguir mas precisión.
- Añadir la posibilidad de mezclado manual de contactos, ya que algunos por sus características, o por la información existente de ellos en las redes sociales, pueden no ser mezclados.
- Posibilidad de envío de mensajes a los contactos a través de las redes sociales en las que los hemos localizado.

## Código fuente

Para facilitar el desarrollo de la aplicación, y la posibilidad de posibles colaboraciones futuras, se ha repositado el código fuente en la URL <https://github.com/acuervoa/Merge>



## Bibliografía

Para la realización del presente proyecto se han utilizado las fuentes de información que se relacionan a continuación:

### **iOS 6 Recipes. A Problem-Solution Approach**

Ed. Apress. (2012)

*Hans-Eric Grönlud, Colin Francis, Shawn Grimes*

### **iPhone Programming. The big nerd ranch guide**

Ed Big Nerd Ranch

*Joe Conway, Aaron Hillegass*

### **iOS 6 Programming (3rd Edition)**

Ed O'Reilly

*Matt Neuburg*

### **Beginning iOS Storyboarding using Xcode**

Ed. Apress (2012)

*Dr.Rory Lewis, Yulia McCarthy, Stephen Moraco*

### **Geolocation in iOS**

Ed O'Reilly

*Alasdair Allan*

### **iPhone and iPad Apps for Absolute Beginners**

Ed Apress

*Dr. Rory Lewis*

### **Stackoverflow**

<http://www.stackoverflow.com>

**Apple Developer**

<http://developer.apple.com>

**Facebook developers**

<https://developers.facebook.com>

**Twitter developers**

<https://dev.twitter.com>