

# *Locate 'N' track*

Localiza y rastrea tus  
dispositivos desde  
cualquier parte

Proyecto de Sistemas  
Informáticos

Facultad de informática

Curso 2013/2014



UNIVERSIDAD COMPLUTENSE  
MADRID

Alejandro Blasco de Miguel





Autorizo a la Universidad Complutense a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, los contenidos audiovisuales incluso si incluyen imágenes de los autores, la documentación y/o el prototipo desarrollado.

Madrid, a 12 de septiembre de 2014.

Alejandro Blasco de Miguel



## Prólogo

Imagine que es un padre de familia. Podría interesarle conocer la ubicación del teléfono móvil de su hijo cuando éste se encuentra fuera de casa.

O también usted podría ser una de esas personas que han podido extraviar el móvil o les ha sido sustraído y está interesado en conocer su paradero.

A pesar de que existen numerosas soluciones en Internet que resuelven el problema, todas ellas provienen de grandes compañías interesadas en manejar datos sensibles del usuario. Muchas personas, conocedoras de los últimos escándalos relativos al espionaje por parte de organismos gubernamentales como la NSA que se nutre de la información de los usuarios de Google, Apple o Microsoft entre otras, temen usar estas herramientas pues creen que atentan contra su intimidad.

La gente no desea ser controlada, no quiere que una compañía o un gobierno pueda almacenar en grandes bases de datos sus datos, sus gustos, su ubicación... Sin duda es un gran problema.

*Locate 'N' Track* viene a dar solución a la demanda de una herramienta que permita conocer la ubicación de un dispositivo móvil sin pasar por los servidores de empresas. La ubicación irá directamente al servidor, y usted podrá visualizar en un navegador web la situación del teléfono móvil o tablet.







# Índice

Prólogo.....	II
Índice de tablas.....	V
Índice de ilustraciones.....	VI
Resumen.....	VII
Abstract.....	VIII
Palabras clave.....	IX
Keywords.....	IX
<b>1. Introducción.....</b>	<b>1</b>
<b>2. Posicionamiento.....</b>	<b>2</b>
2.1. Oportunidad de negocio.....	2
2.2. Android Device Manger.....	2
2.3. Find My iPhone.....	3
<b>3. Historias de usuario.....</b>	<b>4</b>
3.1. Introducción.....	4
3.2. Usabilidad.....	9
3.3. Fiabilidad.....	9
3.4. Rendimiento y compatibilidad.....	10
3.5. Herramientas y Tecnologías utilizadas.....	10
3.6. Restricciones de diseño.....	11
<b>4. Especificaciones de requisitos.....</b>	<b>12</b>
4.1. Requisitos de la web.....	12
4.2. Requisitos de la Aplicación móvil.....	17
<b>5. Arquitectura del sistema.....</b>	<b>21</b>
5.1. Introducción.....	21
5.2. Objetivos del diseño.....	21

5.3. Vista Componentes.....	22
5.3.1. Componentes más significativos de la arquitectura. Dominio de clases. ....	25
5.4. Vista de interacción.....	31
<b>6. Manual de usuario.....</b>	<b>33</b>
6.1. Instalación.....	33
6.1.1. Android App.....	33
6.1.2. iOS App.....	35
6.1.3. Servidor.....	36
6.2. Envío de localización.....	38
6.2.1. Android App.....	38
6.2.2. iOS App .....	40
6.3. Localizar los dispositivos.....	43
<b>7. Resultado.....</b>	<b>46</b>
7.1. Metodología de trabajo.....	46
7.2. Líneas de trabajo futuro.....	47
7.2.1. Aplicaciones como demonios .....	47
7.2.2. Acciones sobre los dispositivos.....	47
7.2.3. Log-in.....	47
7.2.4. Rastreo desde otra app .....	48
7.2.5. Llévame hasta mi dispositivo.....	48
<b>8. Conclusiones.....</b>	<b>49</b>
<b>Bibliografía.....</b>	<b>52</b>

## Índice de tablas

Tabla 1. Historia de usuario 1.....	4
Tabla 2. Historia de usuario 2.....	5
Tabla 3. Historia de usuario 3.....	5
Tabla 4. Historia de usuario 4.....	6
Tabla 5. Historia de usuario 5.....	6
Tabla 6. Historia de usuario 6.....	7
Tabla 7. Historia de Usuario 7 .....	7
Tabla 8. Historia de usuario 8.....	8
Tabla 9. Historia de usuario 9.....	8
Tabla 10. Caso de uso: Mostrar dispositivos .....	13
Tabla 11. Caso de uso: Mostrar localización .....	15
Tabla 12. Caso de uso: Configuración .....	18
Tabla 13. Caso de uso: Enviar localización .....	19

## Índice de ilustraciones

Ilustración 1. Diagrama general: Web .....	12
Ilustración 2. Diagrama de actividad: Mostrar dispositivos .....	14
Ilustración 3. Diagrama de actividad: Mostrar localización .....	16
Ilustración 4. Diagrama general: Aplicación móvil .....	17
Ilustración 5. Diagrama de actividad: Configuración .....	18
Ilustración 6. Diagrama de actividad: enviar localización .....	20
Ilustración 7. Arquitectura de componentes .....	24
Ilustración 8. Dominio de clases Servidor .....	26
Ilustración 9. Dominio de clases Android aplicación .....	28
Ilustración 10. Dominio de clases iOS aplicación .....	29
Ilustración 11. Modelo relacional base de datos .....	30
Ilustración 12. Diagrama de flujo .....	32
Ilustración 13. Aspecto aplicación .....	34
Ilustración 14. Aspecto aplicación .....	35
Ilustración 15. Instalación en servidor .....	36
Ilustración 16. Muestra de localización del dispositivo .....	37
Ilustración 17. Visión de la posición del dispositivo .....	38
Ilustración 18. Simulación de ubicación .....	39
Ilustración 19. Aplicación .....	40
Ilustración 20. Mensaje informativo .....	41
Ilustración 21. Posición del dispositivo .....	42
Ilustración 22. Localización dispositivo desde navegador .....	43
Ilustración 23. Localización .....	44
Ilustración 24. Localización dispositivo .....	44
Ilustración 25. Traza del recorrido del dispositivo .....	45

## Resumen

*Locate 'N' Track* es un sistema informático que se centra en la localización y rastreo libre de cualquier dispositivo móvil con sistema operativo Android o iOS. Libre en el sentido de que la información viaja anónimamente y no es vigilada ni soportada por empresas.

De manera alternativa, el usuario puede arrancar el servidor en su ordenador personal para que la información de ubicación de sus dispositivos quede registrada en un archivo 'log' en su máquina. En tal caso habría que cambiar la dirección url y el puerto en el que escucha el servidor, tanto en la configuración del servidor como en la de las apps.

Todo esto hace que *Locate 'N' Track* sea una buena herramienta para personas que busquen una forma sencilla y limpia de ubicar dispositivos con tan solo disponer de un navegador web y conexión a Internet.

## Abstract

*Locate 'N' Track* is a software system focused on locating and Free tracking of any Android or iOS mobile devices. By free we mean in the important sense of freedom because the information travels anonymously and is not monitored or supported by companies.

Alternatively, the user can start the server on your home computer to allow the devices' location is recorded in a log file on your machine. In such a case, the url address and the port should be changed on which the server listens, both for the server and for the apps.

All of this makes *Locate 'N' Track* a good tool for people who are looking for a easy and clean way to locating devices even by just having a web browser and Internet connection.

## Palabras clave

- Localización
- Rastreo
- Libre
- Multiplataforma

## Keywords

- Locating
- Tracking
- Free
- Cross-platform





# 1. Introducción

El objetivo principal de este proyecto es el de brindar al usuario la posibilidad de conocer la ubicación de sus dispositivos móviles de una manera sencilla y libre, sin tener que recurrir a aplicaciones típicas que almacenan datos personales.

De esta manera, se propone un sistema bastante interesante para personas preocupadas por su anonimato que desean tener control de la ubicación de sus terminales móviles, o aquellas que deseen implementar en su hogar un sistema capaz de recoger la ubicación de móviles de forma que la información quede registrada en un ordenador privado.

A lo largo del documento se expone el trabajo realizado y distintos aspectos de éste, de tal forma que cualquier persona interesada conozca las distintas partes del proyecto desarrollado y pueda conocer de la manera más profunda posible la forma en que se han elaborado las distintas partes.

También se muestra un pequeño manual de uso y los conceptos básicos para la utilización del sistema, una autoevaluación del trabajo, qué se ha aprendido, y qué se cambiaría. Además, se ofrece una visión de lo que pudiera llegar a ser un producto comercial a partir de los logros obtenidos.

## 2. Posicionamiento

### 2.1. Oportunidad de negocio

Existen actualmente aplicaciones similares a *Locate 'N' Track* que son conocidas comúnmente como *Android Device Manager* o *Find My iPhone*. Lo que el proyecto aporta de nuevo al usuario, como se ha comentado ya anteriormente, es el concepto de libertad: tener la tranquilidad de que sus datos no serán almacenados en ninguna base de datos para comerciar con ellos o con otras finalidades mucho menos éticas.

Además, la posibilidad de tener tu propio servidor para recoger localizaciones con un par de pasos de configuración también resulta muy interesante.

La competencia sobre este tipo de aplicaciones es extensa, ya que existen aplicaciones que ya vienen de serie en los sistemas operativos de móviles y tabletas, de uso extendido y también de pago con un gran equipo de trabajo detrás. Pero ninguna ofrece la sencillez de uso, de configuración y de sensación de honestidad que *Locate 'N' Track* ya que el resto son complejos y están orientados a capturar rutinas de los usuarios o movidos por fines económicos.

### 2.2. Android Device Manger

*Android Device Manager* es una aplicación proporcionada de manera gratuita en la que podemos interactuar remotamente con nuestro dispositivo en caso de robo o pérdida. Entre las opciones

disponibles encontramos borrado remoto, reproducción de un sonido de alerta en nuestro dispositivo (para encontrarlo en caso de que lo hayamos perdido por casa) o localizarlo mediante *Google Maps*.

Para acceder al servicio, basta con entrar en la web, iniciar sesión con tu cuenta de Google.

*<https://www.google.com/android/devicemanager>*

### 2.3. Find My iPhone

*Find My iPhone* es una aplicación integrada en los dispositivos de Apple con iCloud (iPhone, iPod, iPad y Mac) para localizar en un mapa la dirección de tu dispositivo. El funcionamiento es exactamente el mismo que el que ofrece su compañía rival Google.

Para acceder al servicio, basta con entrar en la web e iniciar sesión con el ID de Apple.

*<https://www.apple.com/icloud/fid-my-iphone.html>*

## 3. Historias de usuario

### 3.1. Introducción

*Locate 'N' Track* se centra en la localización de dispositivos Android y iOS desde un navegador web.

Partiendo de esa idea, se debe proporcionar un sistema capaz de mantener una comunicación ininterrumpida entre los terminales móviles y el servidor. Y que sea posible visualizar en un mapa la ubicación de estos.

Todo lo anterior lo expresamos a continuación a partir de historias de usuario que reflejan algunos de los objetivos iniciales que nos planteamos.

Historia de Usuario	
<b>Número: 1</b>	<b>Usuario:</b> Cliente(Android)
<b>Nombre historia:</b> Envío de datos al servidor	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Alejandro Blasco	
<b>Descripción:</b> El cliente Android envía datos de su localización al servidor. Datos su dirección y puerto, su identificación y su localización en sintaxis JSON	
<b>Observaciones:</b>	

TABLA 1. HISTORIA DE USUARIO 1

Historia de Usuario	
<b>Número: 2</b>	<b>Usuario:</b> Cliente(iOS)
<b>Nombre historia:</b> Envío de datos al servidor	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b> 3.5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Alejandro Blasco	
<b>Descripción:</b> El cliente iOS envía datos de su localización al servidor. Datos su dirección y puerto, su identificación y su localización en sintaxis JSON	
<b>Observaciones:</b>	

TABLA 2. HISTORIA DE USUARIO 2

Historia de Usuario	
<b>Número: 3</b>	<b>Usuario:</b> Cliente(web)
<b>Nombre historia:</b> Mostrar dispositivos	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Alejandro Blasco	
<b>Descripción:</b> Desde la página web, el usuario debe ser capaz de ver los dispositivos activos que están enviando su localización en una tabla.	
<b>Observaciones:</b>	

TABLA 3. HISTORIA DE USUARIO 3

Historia de Usuario	
<b>Número: 4</b>	<b>Usuario:</b> Cliente(web)
<b>Nombre historia:</b> Mostrar localización	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Baja
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Alejandro Blasco	
<b>Descripción:</b> El usuario que selecciona un dispositivo de la tabla, debe poder ver la localización del mismo en el mapa	
<b>Observaciones:</b>	

TABLA 4. HISTORIA DE USUARIO 4

Historia de Usuario	
<b>Número: 5</b>	<b>Usuario:</b> Cliente(Android)
<b>Nombre historia:</b> Mostrar ubicación actual	
<b>Prioridad en negocio:</b> Baja	<b>Riesgo en desarrollo:</b> Baja
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Alejandro Blasco	
<b>Descripción:</b> El cliente Android debe ser capaz de mostrar en un mapa la ubicación actual	
<b>Observaciones:</b>	

TABLA 5. HISTORIA DE USUARIO 5

Historia de Usuario	
<b>Número: 6</b>	<b>Usuario:</b> Cliente(iOS)
<b>Nombre historia:</b> Mostrar ubicación actual	
<b>Prioridad en negocio:</b> Baja	<b>Riesgo en desarrollo:</b> Baja
<b>Puntos estimados:</b> 4	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Alejandro Blasco	
<b>Descripción:</b> El cliente iOS debe ser capa de mostrar en un mapa la ubicación actual	
<b>Observaciones:</b>	

TABLA 6. HISTORIA DE USUARIO 6

Historia de Usuario	
<b>Número: 7</b>	<b>Usuario:</b> Servidor
<b>Nombre historia:</b> Log	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Baja
<b>Puntos estimados:</b> 3.5	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Alejandro Blasco	
<b>Descripción:</b> El servidor debe añadir en un archivo de Log cada nueva ubicación que reciba de cada dispositivo	
<b>Observaciones:</b> El archivo contendrá además de las coordenadas, una marca de tiempo	

TABLA 7. HISTORIA DE USUARIO 7

Historia de Usuario	
<b>Número: 8</b>	<b>Usuario:</b> Cliente(Android)
<b>Nombre historia:</b> Singleton	
<b>Prioridad en negocio:</b> Baja	<b>Riesgo en desarrollo:</b> Baja
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Alejandro Blasco	
<b>Descripción:</b> No se permitirá crear más de una instancia para el cliente de Android	
<b>Observaciones:</b>	

TABLA 8. HISTORIA DE USUARIO 8

Historia de Usuario	
<b>Número: 9</b>	<b>Usuario:</b> Cliente(iOS)
<b>Nombre historia:</b> Singleton	
<b>Prioridad en negocio:</b> Baja	<b>Riesgo en desarrollo:</b> Baja
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Alejandro Blasco	
<b>Descripción:</b> No se permitirá crear más de una instancia para el cliente de iOS	
<b>Observaciones:</b>	

TABLA 9. HISTORIA DE USUARIO 9



## 3.2. Usabilidad

El usuario no necesita tener conocimiento alguno de cómo funciona *Locate 'N' Track*. Una vez instalada y ejecutada la app, ella se encarga automáticamente de enviar los datos, basta que el usuario ingrese en la web y pinche sobre su dispositivo para empezar a visualizar la localización.

Solo en el caso de que se quiera instalar la parte del servidor, el usuario debería hacer un cambio en la url y puerto en el fichero del servidor para que apuntara a su ordenador (por supuesto hacer el pertinente *port forwarding* en el router) y en las app's.

## 3.3. Fiabilidad

La fiabilidad del sistema se basa, por un lado, en la disponibilidad que tenga el dispositivo móvil de cobertura GPS y de acceso a Internet para aportar los datos de localización al servidor. Y si no puede ser así por cualquier motivo, el usuario puede acceder a la última localización conocida.

En caso de tener acceso al servidor, podrá tener a disposición un archivo log con una traza de todas las coordenadas y su respectiva marca de tiempo.

Es importante señalar que el sistema es totalmente accesible desde cualquiera de los navegadores web Chrome, Firefox, Opera, Safari e Internet Explorer en sus versiones más recientes.

En cuanto a las aplicaciones móviles, para iOS ha sido probada satisfactoriamente para terminales con la versión 7.1 o posterior; y en Android funciona satisfactoriamente para cualquier terminal que cuente con una versión entre 2.2 (Froyo) y 4.2 (Jelly Bean).

Concrétamente se realizaron pruebas sobre un terminal móvil real LG Optimus L4.

### 3.4. Rendimiento y compatibilidad

El rendimiento de la aplicación vendrá determinado, por un lado, por la conexión a internet de la que disponga el usuario tanto en el lado del servidor para visualizar el mapa en el navegador, como en el lado del cliente ya que los terminales necesitan enviar su localización haciendo uso de una red de datos.

Por otro lado, este rendimiento apenas se verá afectado por la capacidad de terminales que envíen su localización. La naturaleza asíncrona de *Javascript*, es una ventaja a la hora de lanzar procesos paralelos. Si queremos lanzar varias peticiones paralelas a una web, por defecto no tenemos que hacer nada. Esta misma propiedad, es la que ha hecho que los servidores creados con *Node*, gestionen bien la concurrencia y soporten cantidades ingentes de peticiones concurrentes.

### 3.5. Herramientas y Tecnologías utilizadas.

Para el desarrollo de la aplicación del cliente en Android, se hizo uso del lenguaje de programación Java y la herramienta Android SDK (Kit de Desarrollo de Software para Android) instalado como un plugin para el IDE Eclipse

Para el desarrollo de la aplicación del cliente en iOS, se hizo uso del lenguaje de programación Objective-C y la herramienta XCode (Entorno de desarrollo para iOS)

De cara a la realización del servidor web, se empleó Node.js, que es el entorno de programación en la capa del servidor basado en Javascript.

La parte web se ha apoyado en las tecnologías HTML5 para la página web, CSS3 para los estilos, JavaScript y jQuery para interactuar con HTML y el API de Google Maps v2 para poder mostrar los mapas.

### 3.6. Restricciones de diseño

Para la implementación de la aplicación en Android se usó el entorno de desarrollo Eclipse, versión Juno, junto con el SDK de Android.

Se ha usado el SDK de Android 4.2 Jelly Bean para desarrollar la aplicación Android, pero se ha asegurado que la aplicación funcione con una mínima versión de 2.2 Froyo, que es la mínima versión que tiene cualquier dispositivo en la actualidad.

Para la implementación de la aplicación en iOS, el entorno de desarrollo fue Xcode 5 que debe correr forzosamente sobre un sistema operativo Mac OS X. En este caso se usó Mac OS X 10.9.4 Mavericks.

## 4. Especificaciones de requisitos

### 4.1. Requisitos de la web

El siguiente apartado trata de explicar las funcionalidades con las que cuenta *Locate 'N' Track* en la parte Web que permite visualizar la ubicación de los dispositivos. Para tal fin se muestran los casos de uso y diagramas de actividades que aparecen a continuación.

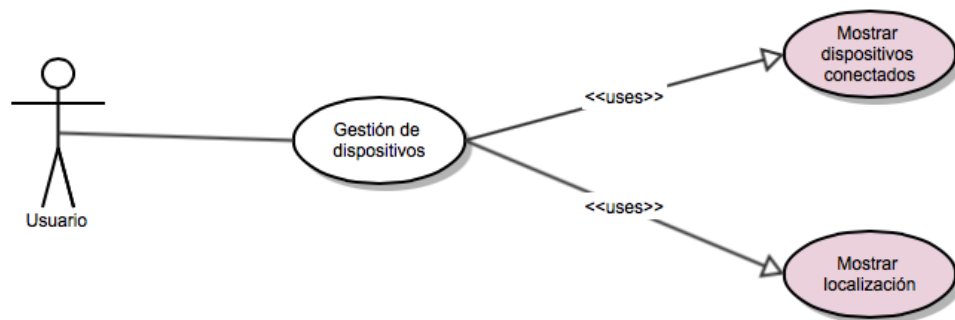


ILUSTRACIÓN 1. DIAGRAMA GENERAL: WEB

Caso de uso: Mostrar dispositivos		
Descripción	Permite mostrar el nombre de todos los dispositivos conectados al sistema	
Restricciones	Pre- Condiciones	El servidor ha arrancado correctamente
	Post- Condiciones	El usuario dispone de una lista de todos los dispositivos y puede acceder a su localización clicando sobre ellos.
Secuencia de eventos	<ol style="list-style-type: none"> <li>1. El usuario ingresa en la web a través de la dirección y el puerto en el que está escuchando el servidor.</li> <li>2. Se muestra una lista de dispositivos que ya estuvieran enviando su localización.</li> <li>3. La lista de dispositivos se modifica dinámicamente               <ol style="list-style-type: none"> <li>3.1. Un nuevo dispositivo comienza a enviar su localización                   <ol style="list-style-type: none"> <li>3.1.1. Se añade el nuevo dispositivo a la lista.</li> <li>3.1.2. Regresa a 2</li> </ol> </li> <li>3.2. Un dispositivo que estaba en la lista se desconecta o deja de enviar su localización.                   <ol style="list-style-type: none"> <li>3.2.1. Se mantiene en la lista y se puede acceder a la última localización conocida.</li> <li>3.2.2. Regresa a 2</li> </ol> </li> </ol> </li> </ol>	

TABLA 10. CASO DE USO: MOSTRAR DISPOSITIVOS

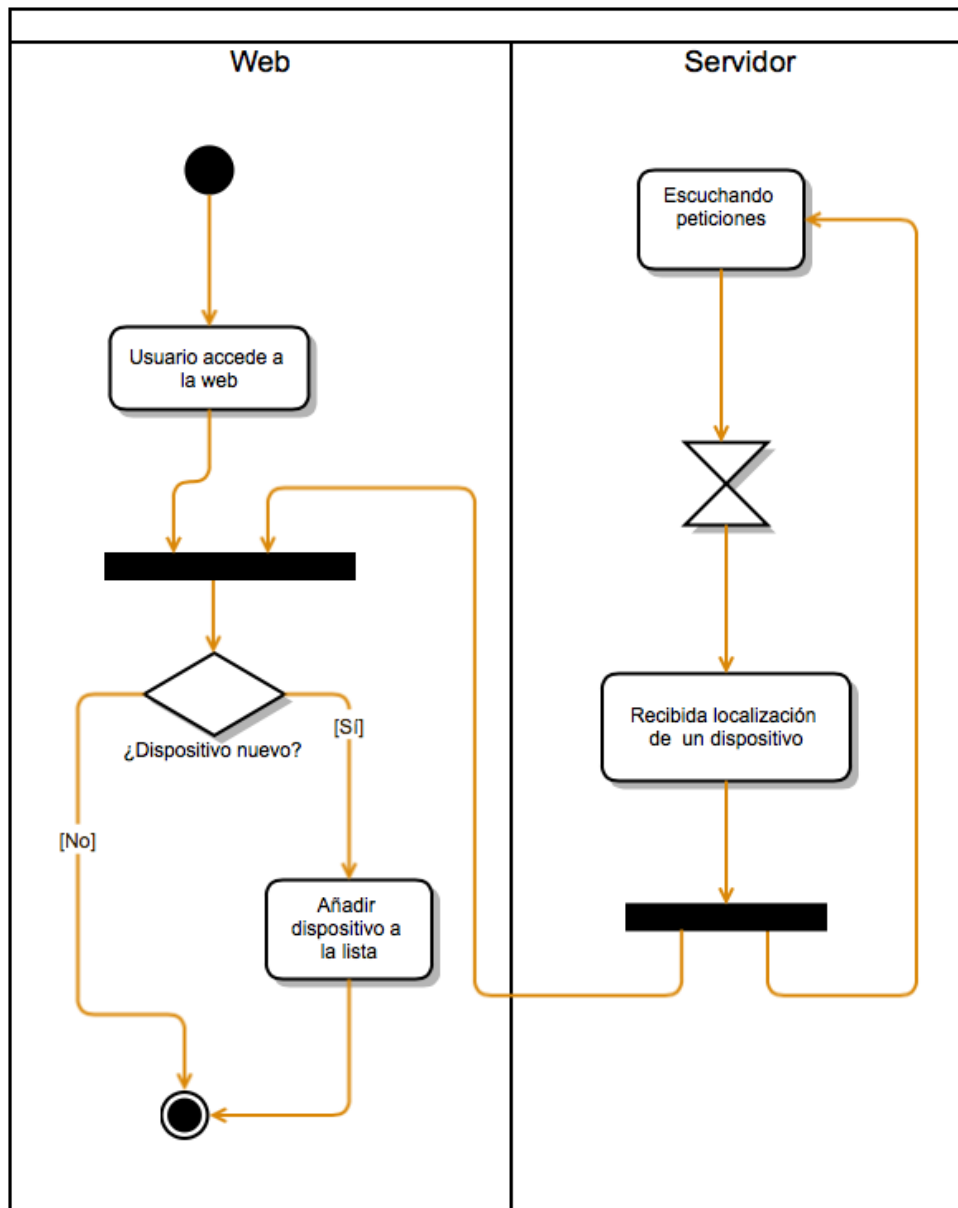


ILUSTRACIÓN 2. DIAGRAMA DE ACTIVIDAD: MOSTRAR DISPOSITIVOS

Caso de uso: Mostrar localización		
Descripción	Permite mostrar en un mapa la localización actual de un dispositivo conectado.	
Restricciones	Pre- Condiciones	El dispositivo está enviando su localización al servidor.
	Post- Condiciones	El usuario puede ver en un mapa la localización de un determinado dispositivo y ver cómo va cambiando su posición.
Secuencia de eventos	<ol style="list-style-type: none"> <li>1. El usuario pincha sobre un dispositivo concreto en la lista de dispositivos conectados</li> <li>2. Se muestra sobre un mapa un icono representando el dispositivo móvil</li> <li>3. El icono se actualice a nuevas posiciones del mapa a medida que envía su posición al servidor</li> </ol>	

TABLA 11. CASO DE USO: MOSTRAR LOCALIZACIÓN

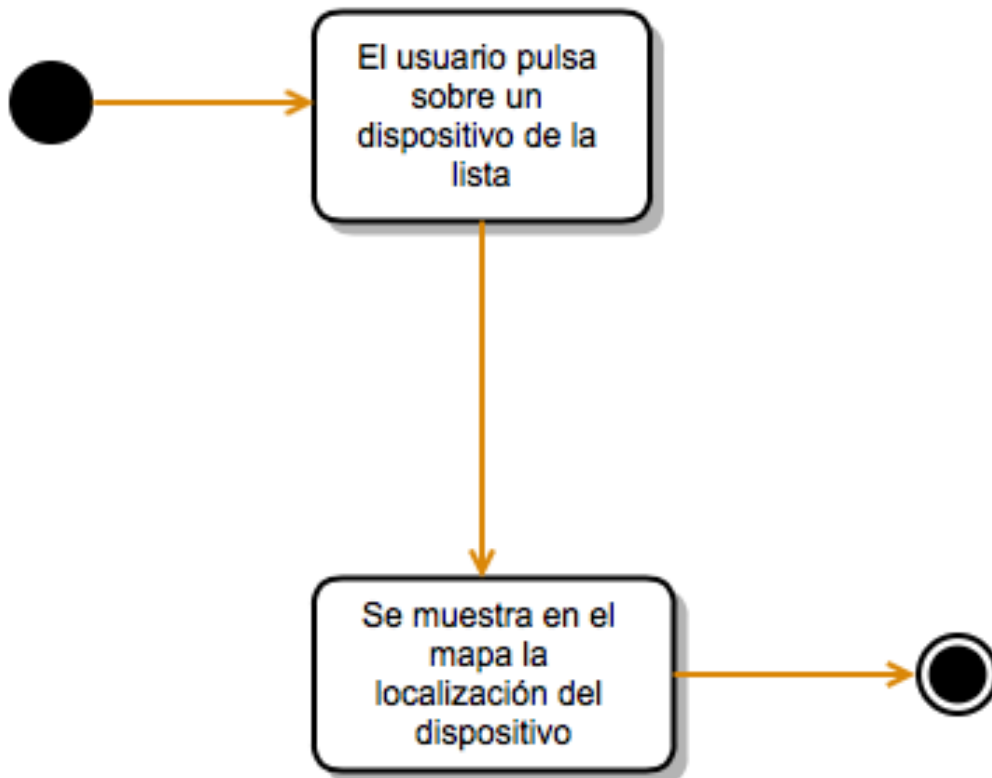


ILUSTRACIÓN 3. DIAGRAMA DE ACTIVIDAD: MOSTRAR LOCALIZACIÓN



## 4.2. Requisitos de la Aplicación móvil

El siguiente apartado trata de explicar las funcionalidades con las que cuenta *Locate 'N' Track* desde el punto de vista de las aplicaciones móviles que se instalarán en los dispositivos móviles. Se apoya en los casos de uso y diagramas de actividades que se muestran a continuación.

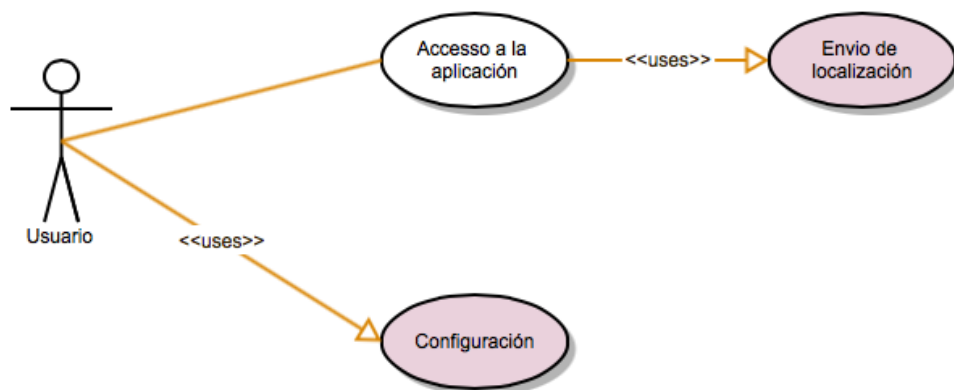


ILUSTRACIÓN 4. DIAGRAMA GENERAL: APLICACIÓN MÓVIL

Caso de uso: Configuración (Android / iOS)		
Descripción	Permite cambiar parámetros sobre la dirección y el puerto en que se encuentra ubicado el servidor. También el modo en que se envía la localización al servidor: cada vez que cambia de posición, o periódicamente cada 6 segundos aunque no haya cambiado de posición.	
Restricciones	Pre- Condiciones	
	Post- Condiciones	La aplicación sabe dónde enviar la información de localización del dispositivo y el modo en que ésta se envía.
Secuencia de eventos	1. El usuario tiene acceso a un archivo de configuración que puede modificar. En el caso de Android se trata de un fichero XML. Para iOS es un fichero .plist	

TABLA 12. CASO DE USO: CONFIGURACIÓN

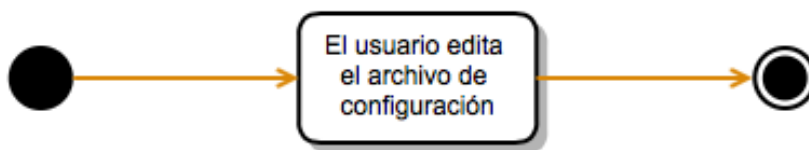


ILUSTRACIÓN 5. DIAGRAMA DE ACTIVIDAD: CONFIGURACIÓN

Caso de uso: Enviar localización (Android / iOS)		
Descripción	Permite cambiar parámetros sobre la dirección y el puerto en que se encuentra ubicado el servidor. También el modo en que se envía la localización al servidor: cada vez que cambia de posición, o periódicamente cada 6 segundos aunque no haya cambiado de posición.	
Restricciones	Pre- Condiciones	
	Post- Condiciones	La aplicación sabe dónde enviar la información de localización del dispositivo y el modo en que ésta se envía.
Secuencia de eventos	<ol style="list-style-type: none"> <li>1. El usuario arranca la aplicación.</li> <li>2. La aplicación carga los valores de los parámetros que se encuentran en el fichero de configuración.</li> <li>3. El modo de envío configurado es inmediato: <ol style="list-style-type: none"> <li>3.1. Obtener la localización GPS cuando la posición del dispositivo cambia.</li> <li>3.2. No se ha podido obtener correctamente la localización <ol style="list-style-type: none"> <li>3.2.1. Regresa a 3.1</li> </ol> </li> <li>3.3. Sí se obtiene la localización <ol style="list-style-type: none"> <li>3.3.1. Envía localización al servidor.</li> <li>3.3.2. Regresa a 3.1</li> </ol> </li> </ol> </li> <li>4. El modo de envío configurado es cada 6 segundos: <ol style="list-style-type: none"> <li>4.1. Obtener la localización GPS</li> <li>4.2. Se ha podido obtener correctamente la localización <ol style="list-style-type: none"> <li>4.2.1. Enviar localización al servidor</li> <li>4.2.2. Esperar 6 segundos</li> <li>4.2.3. Regresar a 4.1</li> </ol> </li> <li>4.3. No se ha podido obtener la localización <ol style="list-style-type: none"> <li>4.3.1. Esperar 6 segundos</li> <li>4.3.2. Regresar a 4.1</li> </ol> </li> </ol> </li> </ol>	

TABLA 13. CASO DE USO: ENVIAR LOCALIZACIÓN

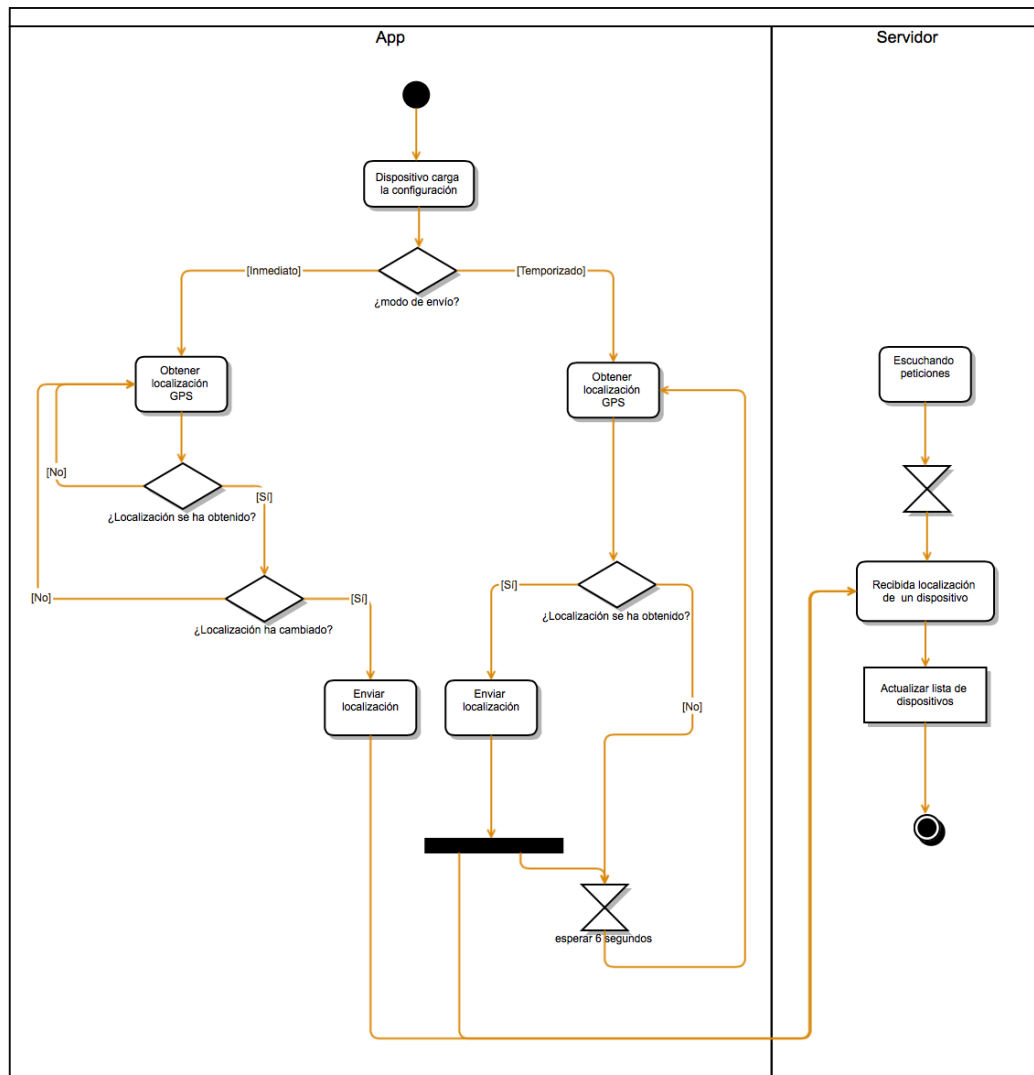


ILUSTRACIÓN 6. DIAGRAMA DE ACTIVIDAD: ENVIAR LOCALIZACIÓN

## 5. Arquitectura del sistema

### 5.1. Introducción

A continuación se presenta un resumen de la arquitectura software de Locate 'N' Track, analizando diferentes aspectos del mismo y dejando ver las decisiones arquitectónicas que se han tomado. Decisiones tomadas para ofrecer un mayor potencial y utilidad al sistema, de tal forma que quede lo suficientemente bien estructurado para continuar con su desarrollo tanto en el presente como en el futuro.

### 5.2. Objetivos del diseño

El sistema consiste en captar la localización de dispositivos móviles, posiblemente un número muy alto. Por tanto, el sistema debe ser capaz de recoger mucha información que envían los dispositivos y ésta debe estar actualizada en todo en momento. Esto implica que la forma de obtener estos datos debe ser lo más rápida posible. Por otro lado, debe ofrecer distintas funcionalidades al usuario de forma que afecte lo menos posible al rendimiento del sistema.

Por tanto, es importante dividir las distintas partes de las que va a estar compuesto nuestro sistema y estudiar su independencia unas de otras, para paralelizarlas y lograr un sistema rápido y fiable. Además, es importante crear un sistema modulado para que los fallos en una de esas partes no afecte a las otras, o al menos minimizarlos.

El sistema consta de varios componentes con distintas funcionalidades. Cada componente gestiona la funcionalidad

ofrecida por uno o más casos de uso y manteniéndola de forma independiente del resto de componentes, aunque evidentemente algunas funcionalidades de uno dependan de otro.

La arquitectura de Locate 'N' Track está diseñada para satisfacer los casos de uso que surgen a partir de las historias de usuario. Este diseño está pensado para ser de fácil comprensión y modulable, de manera que resulte sencillo modificarlo y ampliarlo con otras funcionalidades.

### 5.3. Vista Componentes

Para empezar, se va a detallar las características de Locate 'N' Track a través de un diagrama de componentes con la intención de permitir una visión global del sistema.

El sistema se centra en tres módulos básicos bien diferenciados que ya hemos comentado anteriormente. Cada uno de estos módulos debe ofrecer ciertas funcionalidades. Para ello, consideramos que dentro de cada módulo también debe cumplirse la facilidad para ampliar y quitar funcionalidad. Es por esto que se crean distintos componentes para poder facilitar esta tarea. Posteriormente se mostrarán la especificación de alguno de ellos:

A continuación, se listan los módulos principales que intervienen en la ejecución del sistema, razonando su división y las tecnologías empleadas en cada uno de ellos:

- En la parte de Front end se encontraría el módulo Cliente (Maps): este módulo permite dibujar sobre un mapa la disposición actual de los dispositivos que estén enviando su localización. También permite seleccionar qué dispositivo se quiere visualizar.

- En la parte de Back end, estarían los módulos de Web Server y Database Server: estos módulos funcionan de modo que el servidor recibe la localización y éste almacena dicha información en la BBDD y en un fichero, y se comunica con el Cliente (Maps) para actualizar la posición.
- En la parte de aplicación móvil tenemos dos versiones, una Android y otra iOS, que hacen uso de diferentes librerías y frameworks, y que envían automáticamente su posición al Server.

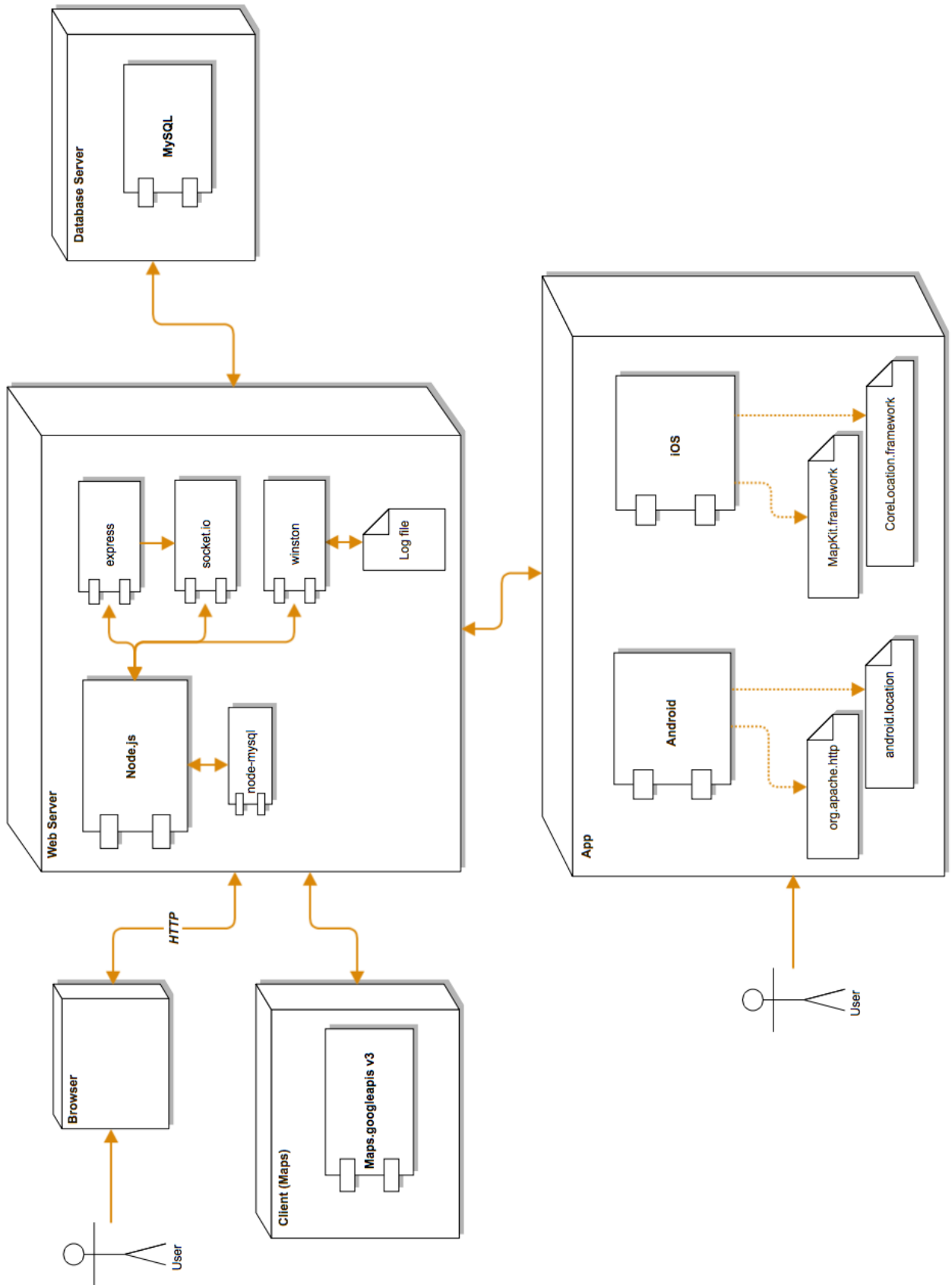


ILUSTRACIÓN 7. ARQUITECTURA DE COMPONENTES



### 5.3.1. Componentes más significativos de la arquitectura. Dominio de clases.

En este apartado se ha tratado de mostrar más en detalle los principales componentes del sistema. Hace referencia al modelo de datos de dominio, de cómo están estructuradas las clases que utiliza el sistema, con sus métodos y atributos más destacables, relaciones entre ellas y dependencias.

#### *Módulo de servidor*

En este módulo se observa la relación entre las clases que especifican el sistema cuya principal tarea es tomar las localizaciones de los dispositivos y mostrarlas en el navegador. En el diagrama aparecen las clases y objetos que toman parte del módulo.

Básicamente el funcionamiento sería el siguiente:

El servidor está escuchando por un puerto. El router funcionaría como middleware y enruta las peticiones que se le hacen al petición con determinadas funciones. Si se le está haciendo un post al servidor con una localización, entonces el router hace uso del socket que envía dicha localización al Client(Map) para que actualice el mapa. Winston manda esas localizaciones a un fichero externo para poderse leer con posterioridad.

Las tecnologías utilizadas son:

- Express, es un framework para crear aplicaciones web de manera sencilla
- Socket.io, que permite la comunicación basada en eventos, bidireccional y en tiempo real.

- Winston, usado para poder mantener un fichero de log que contenga todas las peticiones que se han hecho al servidor.
- Node-mysql, para la parte de conexión con la base de datos.

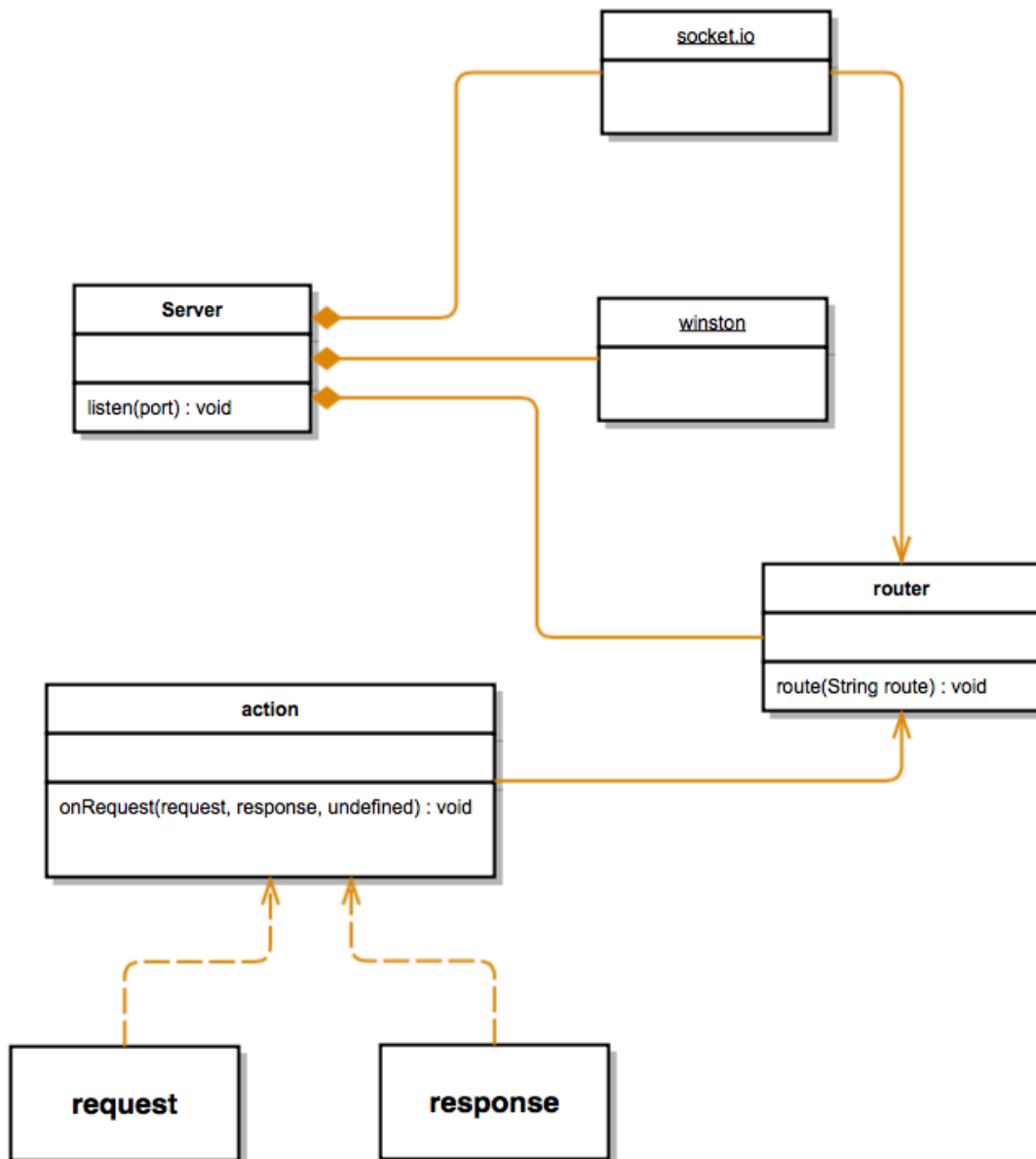


ILUSTRACIÓN 8. DOMINIO DE CLASES SERVIDOR

### *Módulo de aplicaciones*

En esta parte podemos observar las clases en las que se basa todo lo relativo a las aplicaciones móviles

En el caso de Android, al arrancar la aplicación la primera clase que se ejecuta es MainActivity, la cual crea un objeto LocationWatcherSingleton haciendo uso del patrón singleton, y muestra en un mapa la localización del dispositivo, entre otras cosas.

En LocationWatcherSingleton se carga la configuración de envío, y las principales funciones son las siguientes:

- startListening(): método que es llamado desde la instancia de LocationWatcherSingleton en la clase principal para empezar a captar y enviar la localización.
- initializeLocationManager(): inicializa el proveedor de localización del dispositivo, en nuestro caso GPS, con una determinada precisión.
- initializeTimer(): programa un temporizador para que cada 6 segundos pueda ejecutar la tarea de enviar la localización.
- postData(Location): realiza una conexión http con el servidor, en la que el contenido del POST es la localización en formato JSON de la localización.
- onLocationChanged(Location): permite enviar la localización solo cuando la localización ha cambiado.

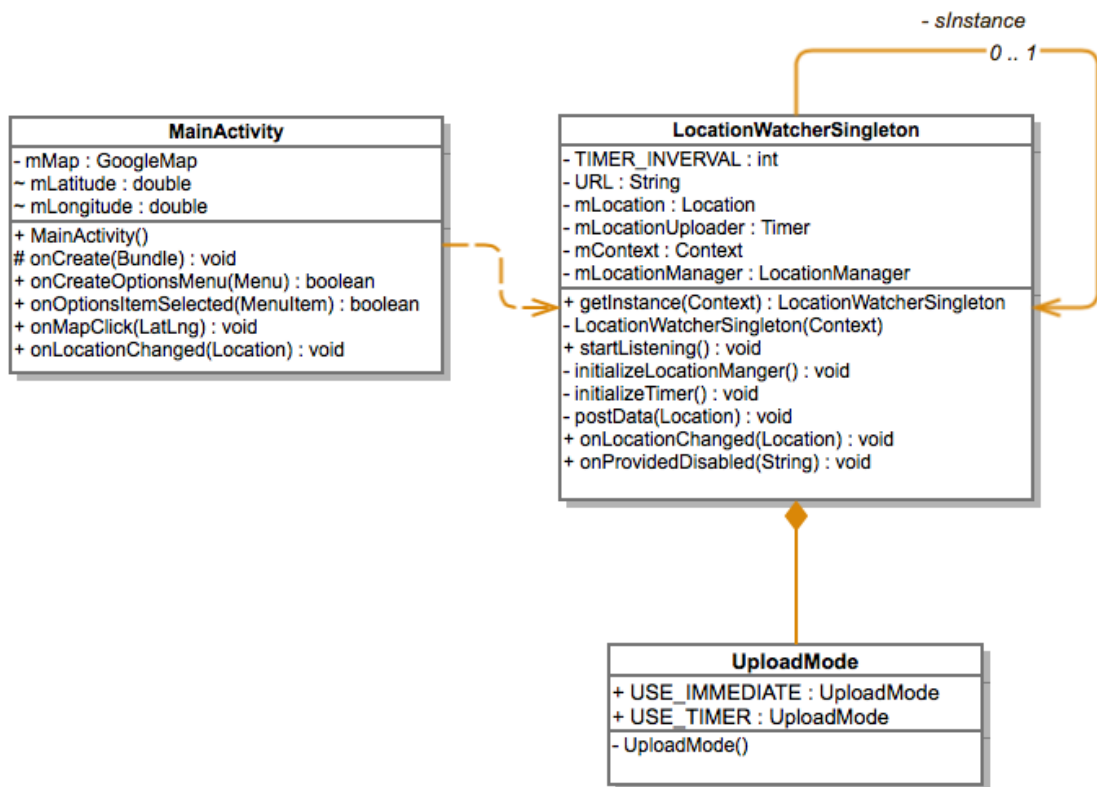


ILUSTRACIÓN 9. DOMINIO DE CLASES ANDROID APLICACIÓN

En el caso de iOS, al arrancar la aplicación, ABMAppDelegate (que por ser muy simple no se muestra en el diagrama) crea una instancia de LocationWatcherSingleton haciendo uso del patrón singleton, y muestra en un mapa la localización del dispositivo, entre otras cosas.

En LocationWatcherSingletonEn se carga la configuración de envío, y las principales funciones son las siguientes:

- startListening(): método que es llamado desde la instancia de LocationWatcherSingleton en ABMAppDelegate para empezar a captar y enviar la localización.
- initializeLocationManager(): inicializa el proveedor de localización del dispositivo, en nuestro caso GPS, con una determinada precisión.
- initializeTimer(): programa un temporizador para que cada 6 segundos pueda ejecutar la tarea de enviar la localización.

- sendLocation(Location): realiza una conexión http con el servidor, en la que el contenido del POST es la localización en formato JSON de la localización.
- onLocationChanged(Location): permite enviar la localización solo cuando la localización ha cambiado.

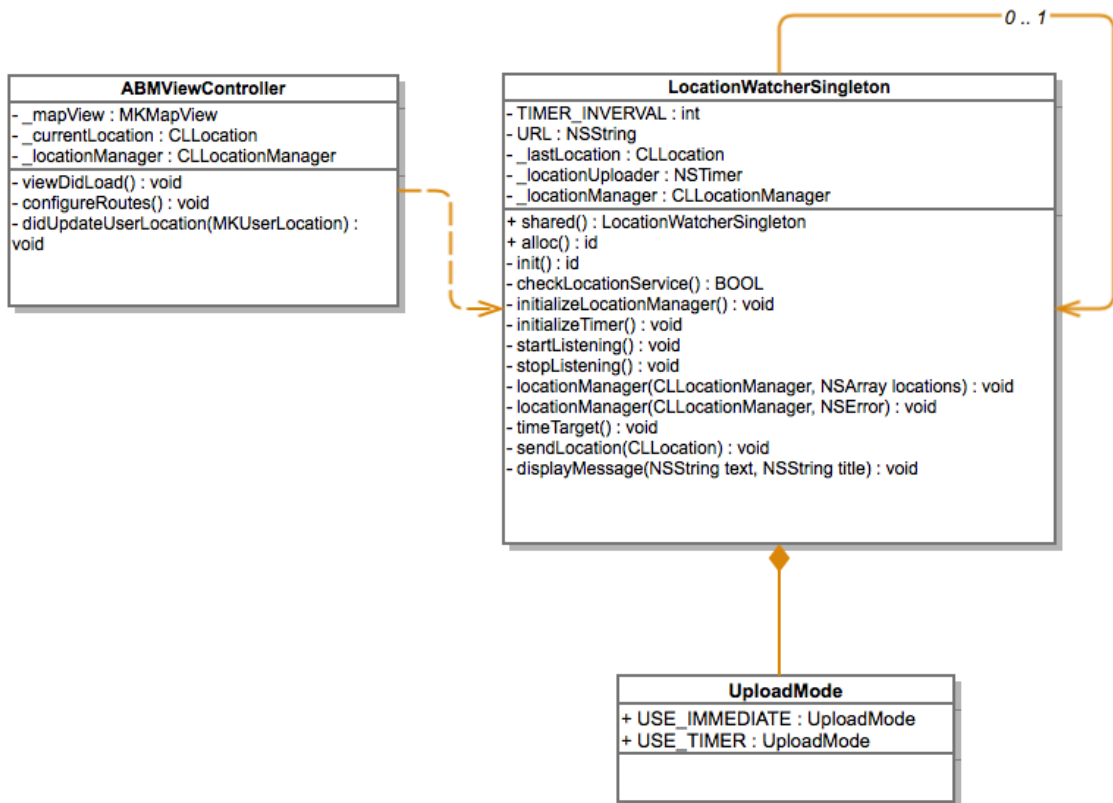


ILUSTRACIÓN 10. DOMINIO DE CLASES IOS APLICACIÓN

### *Módulo de Base de datos*

Una vez visto el dominio de clases, el sistema debe poder respaldada la localización de los dispositivos para poder recuperarla posteriormente. De esta manera se le podrá ofrecer al usuario la posición de sus dispositivos en el tiempo en versiones posteriores. También será la base para la implementación de una fase de login en el futuro.

Para ello, definimos las clases persistentes, es decir, aquellas que deben mantener su valor en el espacio y en el tiempo. Refinamos las clases para adaptarlo a nuestro diseño y realizamos el diagrama de clases para facilitar el trabajo. Una vez que tenemos el modelo de persistencia, proponemos un modelo relacional de base de datos que lo sostenga. A partir del modelo de información y del diseño de la persistencia.

A continuación se muestra el modelo relacional de la base de datos sobre el que se respalda el sistema persistente y modelo de información.

- La tabla users mantiene la información referente a la determinación unívoca del usuario que entra en el sistema.
- La tabla devices relaciona al usuario con la localización de sus dispositivos. Contiene el nombre del dispositivo y el modelo.
- La tabla locations contiene todas las localizaciones y la marca de tiempo en que han sido insertadas, de los dispositivos.

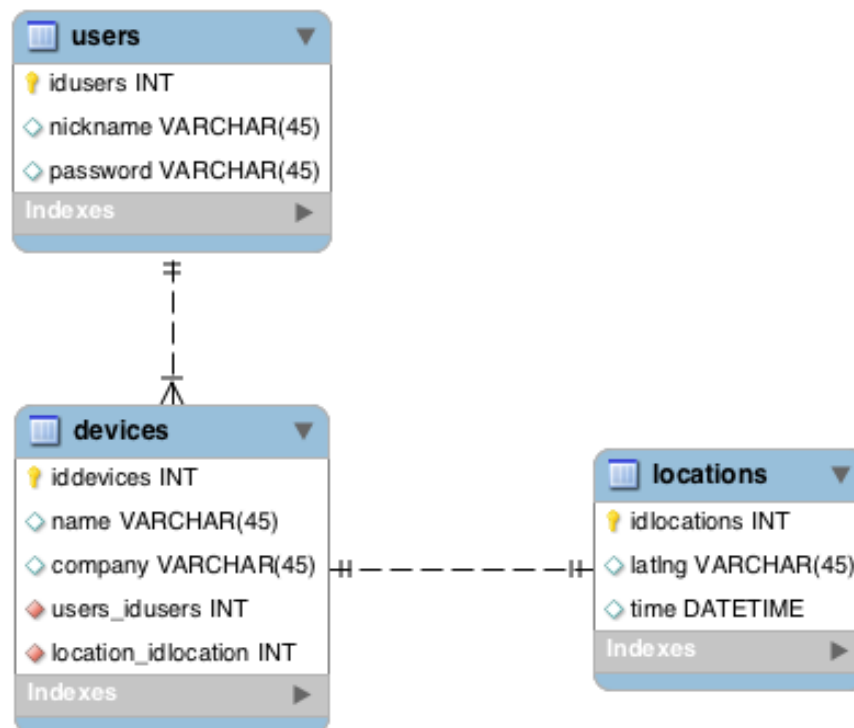


ILUSTRACIÓN 11. MODELO RELACIONAL BASE DE DATOS

## 5.4. Vista de interacción

Se ha visto cómo se comporta el sistema y cómo es la comunicación entre los módulos. Sin embargo, en este apartado se pretende mostrar la interacción entre los distintos módulos a partir de un caso de uso, o más concretamente, un escenario de caso de uso. Uno de los casos de uso, por ejemplo, puede ser el de enviar una localización:

Se le da al navegador el fichero `index.html`, se cargan las imágenes que correspondan y el archivo `Client.js` que permite manipular el mapa. Luego el dispositivo envía la localización al servidor haciendo `POST`, y se actualiza la posición del dispositivo en el mapa.

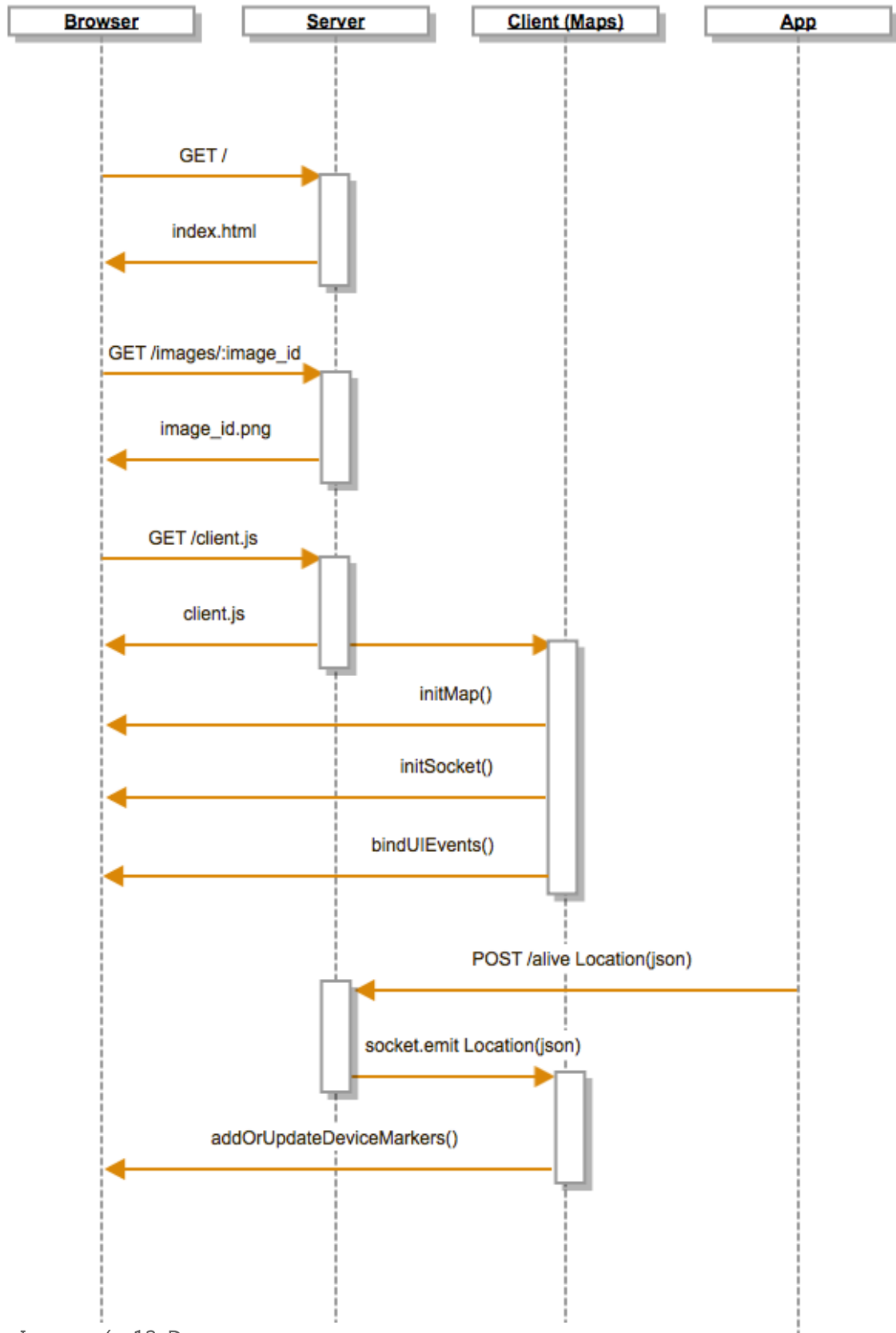


ILUSTRACIÓN 12. DIAGRAMA DE FLUJO



## 6. Manual de usuario

Este manual corresponde con el caso de que el usuario desee correr el servidor en una máquina personal.

Si no fuera así, el usuario no debe efectuar ningún cambio sobre los archivos de configuración de las app's tanto de Android como de iOS. Y por supuesto tampoco en el fichero del servidor.

### 6.1. Instalación

#### 6.1.1. Android App

Para instalar la aplicación del cliente Android, primero debemos modificar los elementos '*server*', '*port*' y '*mode*' del archivo de configuración que se encuentra en `/res/raw/config.xml`

En '*mode*' tenemos dos alternativas: '*timer*' o '*immediate*'.

Por motivos académicos, viene configurado para usar el emulador, que se encuentra en la misma máquina que el servidor, por lo tanto usaremos en '*server*' la dirección `http://10.0.2.2` que es el alias que le da el emulador a la dirección loopback en nuestra máquina, es decir `127.0.0.1`

Luego tenemos que recompilar el proyecto para obtener el APK y después tenemos que ir a la carpeta donde se encuentra: `/android/Android_Tracking/bin`

Después podemos instalarlo en nuestro dispositivo (siempre que hayamos permitido previamente la instalación de aplicaciones desde fuentes desconocidas). Por motivos académicos, vamos a instalarlo en el emulador arrancado "emulator-5554"

```
adb -s emulator-5554 install Android_Tracking.apk
```

Una vez instalado podremos ver en las aplicaciones de nuestro dispositivo la aplicación *Locate 'N' Track*

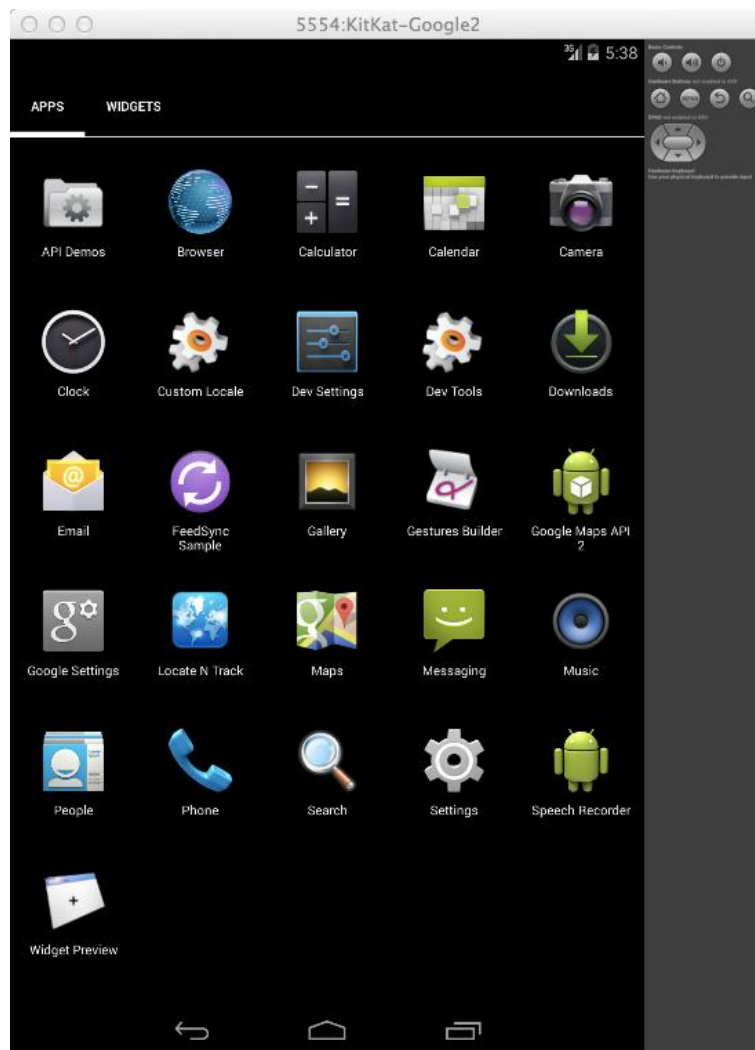


ILUSTRACIÓN 13. ASPECTO APLICACIÓN

## 6.1.2. iOS App

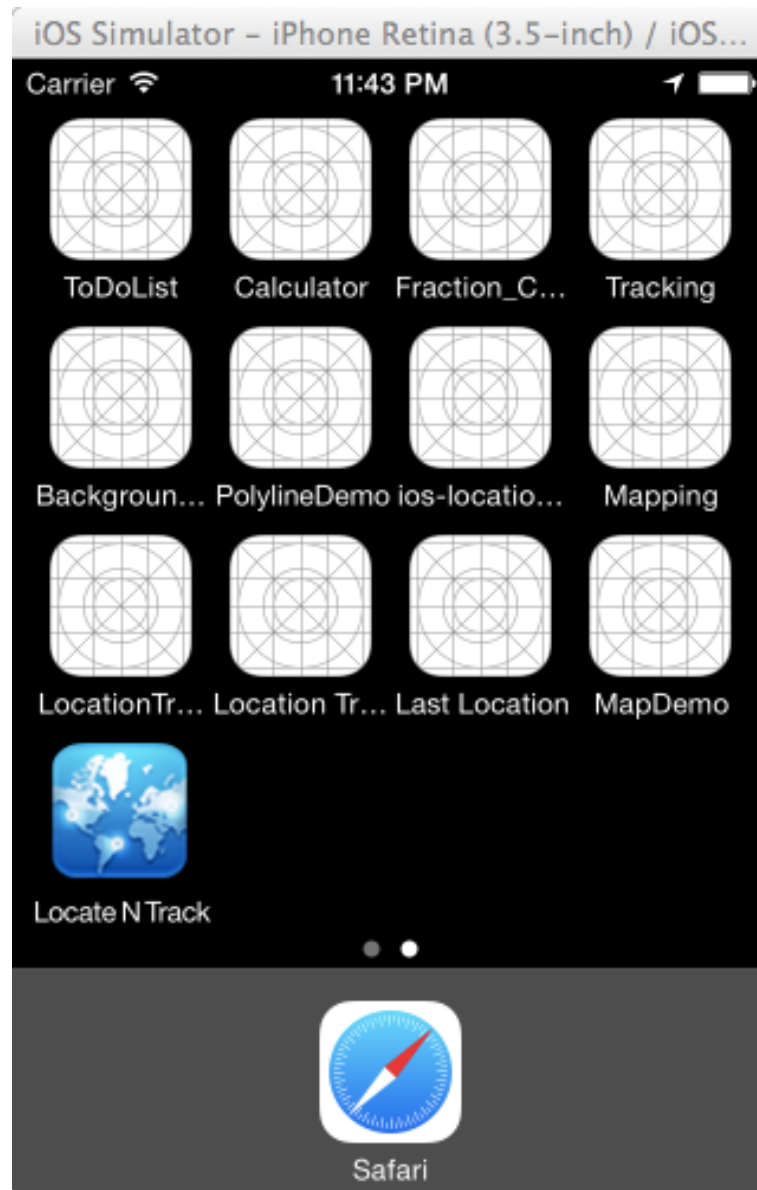


ILUSTRACIÓN 14. ASPECTO APLICACIÓN

Para instalar la aplicación del cliente iOS, primero debemos modificar los elementos *'server'*, *'port'* y *'mode'* del archivo de configuración *config.plist*

En *'mode'* tenemos dos alternativas: *'timer'* o *'immediate'*.

Por motivos académicos, viene configurado para usar el emulador que se encuentra en la misma máquina que el servidor, por lo tanto usaremos en *'server'* la dirección 127.0.0.1

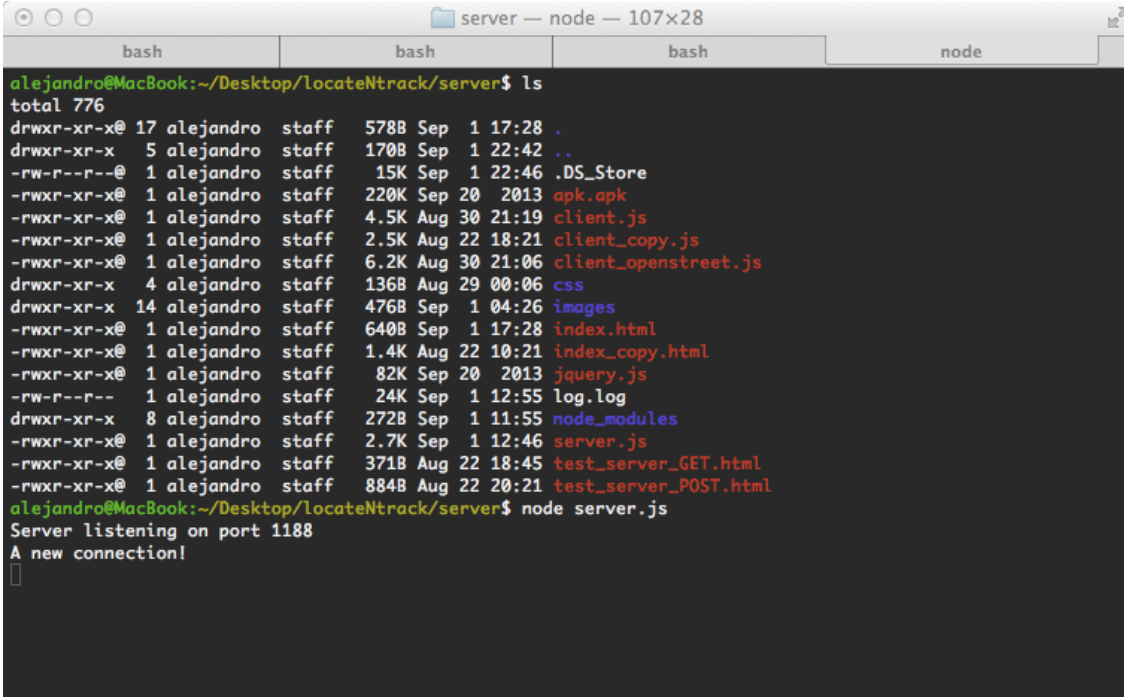
Instalar una app no certificada en iOS supone mucho esfuerzo. En este manual vamos a instalar la aplicación en un emulador desde XCode.

Tras abrir el proyecto y ejecutarlo en el emulador, podemos ver la home del dispositivo que cuenta con la aplicación *Locate 'N' Track*:

### 6.1.3. Servidor

Para comenzar la instalación del servidor de *Locate 'N' Track* en un ordenador personal, primero debe tener instalado Node.js. Después acceda a la carpeta `/server` y arranque el servidor escribiendo “`node server.js`”

El servidor mostrará que ha arrancado en el puerto indicado



```

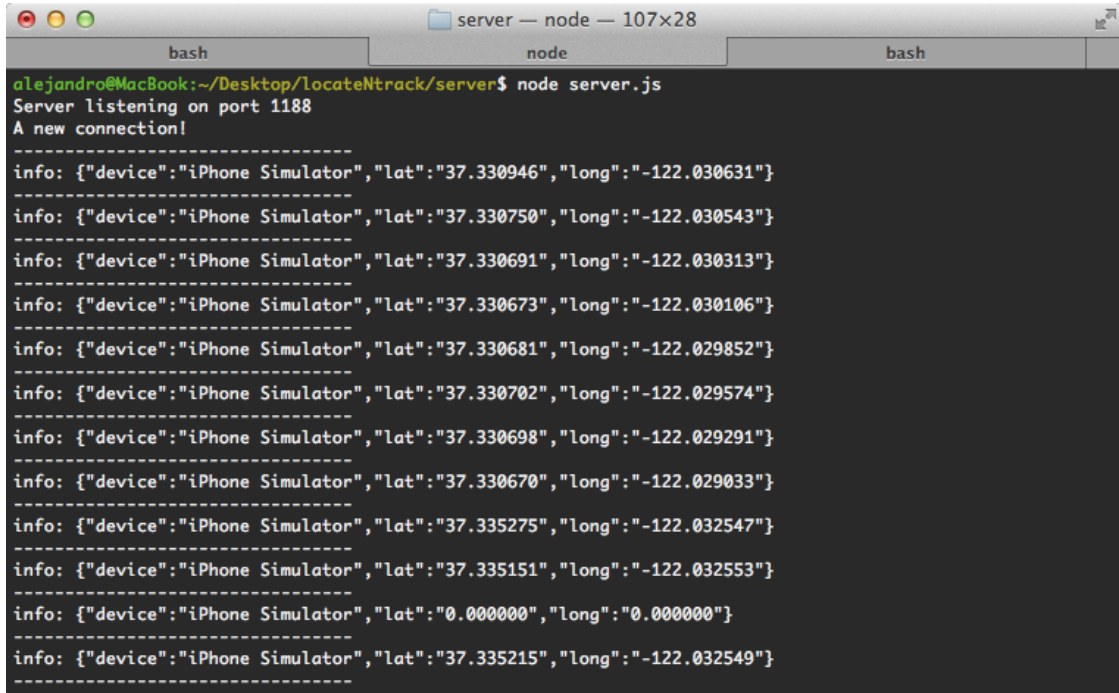
server — node — 107x28
bash bash bash node
alejandro@MacBook:~/Desktop/LocateNtrack/server$ ls
total 776
drwxr-xr-x@ 17 alejandro  staff   578B Sep  1 17:28 .
drwxr-xr-x   5 alejandro  staff  170B Sep  1 22:42 ..
-rw-r--r--@ 1 alejandro  staff   15K Sep  1 22:46 .DS_Store
-rwxr-xr-x@ 1 alejandro  staff  220K Sep 20 2013 apk.apk
-rwxr-xr-x@ 1 alejandro  staff   4.5K Aug 30 21:19 client.js
-rwxr-xr-x@ 1 alejandro  staff   2.5K Aug 22 18:21 client_copy.js
-rwxr-xr-x@ 1 alejandro  staff   6.2K Aug 30 21:06 client_openstreet.js
drwxr-xr-x  4 alejandro  staff  136B Aug 29 00:06 css
drwxr-xr-x 14 alejandro  staff  476B Sep  1 04:26 images
-rwxr-xr-x@ 1 alejandro  staff  640B Sep  1 17:28 index.html
-rwxr-xr-x@ 1 alejandro  staff   1.4K Aug 22 10:21 index_copy.html
-rwxr-xr-x@ 1 alejandro  staff   82K Sep 20 2013 jquery.js
-rw-r--r--@ 1 alejandro  staff   24K Sep  1 12:55 log.log
drwxr-xr-x  8 alejandro  staff  272B Sep  1 11:55 node_modules
-rwxr-xr-x@ 1 alejandro  staff   2.7K Sep  1 12:46 server.js
-rwxr-xr-x@ 1 alejandro  staff   371B Aug 22 18:45 test_server_GET.html
-rwxr-xr-x@ 1 alejandro  staff   884B Aug 22 20:21 test_server_POST.html
alejandro@MacBook:~/Desktop/LocateNtrack/server$ node server.js
Server listening on port 1188
A new connection!

```

ILUSTRACIÓN 15. INSTALACIÓN EN SERVIDOR

\*Por defecto, el servidor viene configurado para escuchar sobre el Puerto 1188. Para cambiar este parámetro, acceda al fichero `server.js` y modifique el valor de la variable `port`.

En el momento en que haya algún dispositivo conectado enviando la localización, empezarán a aparecer por consola la información:



```
server — node — 107x28
bash node bash
alejandro@MacBook:~/Desktop/locateNtrack/server$ node server.js
Server listening on port 1188
A new connection!
-----
info: {"device":"iPhone Simulator","lat":"37.330946","long":"-122.030631"}
-----
info: {"device":"iPhone Simulator","lat":"37.330750","long":"-122.030543"}
-----
info: {"device":"iPhone Simulator","lat":"37.330691","long":"-122.030313"}
-----
info: {"device":"iPhone Simulator","lat":"37.330673","long":"-122.030106"}
-----
info: {"device":"iPhone Simulator","lat":"37.330681","long":"-122.029852"}
-----
info: {"device":"iPhone Simulator","lat":"37.330702","long":"-122.029574"}
-----
info: {"device":"iPhone Simulator","lat":"37.330698","long":"-122.029291"}
-----
info: {"device":"iPhone Simulator","lat":"37.330670","long":"-122.029033"}
-----
info: {"device":"iPhone Simulator","lat":"37.335275","long":"-122.032547"}
-----
info: {"device":"iPhone Simulator","lat":"37.335151","long":"-122.032553"}
-----
info: {"device":"iPhone Simulator","lat":"0.000000","long":"0.000000"}
-----
info: {"device":"iPhone Simulator","lat":"37.335215","long":"-122.032549"}
-----
```

ILUSTRACIÓN 16. MUESTRA DE LOCALIZACIÓN DEL DISPOSITIVO

## 6.2. Envío de localización

### 6.2.1. Android App

Abrir la aplicación haciendo clic en el icono *Locate 'N' Track*

Se abre la aplicación y automáticamente comenzará el envío de la localización al servidor.

Si presionamos sobre *Current location* en el menú podremos tener una visión de la posición.

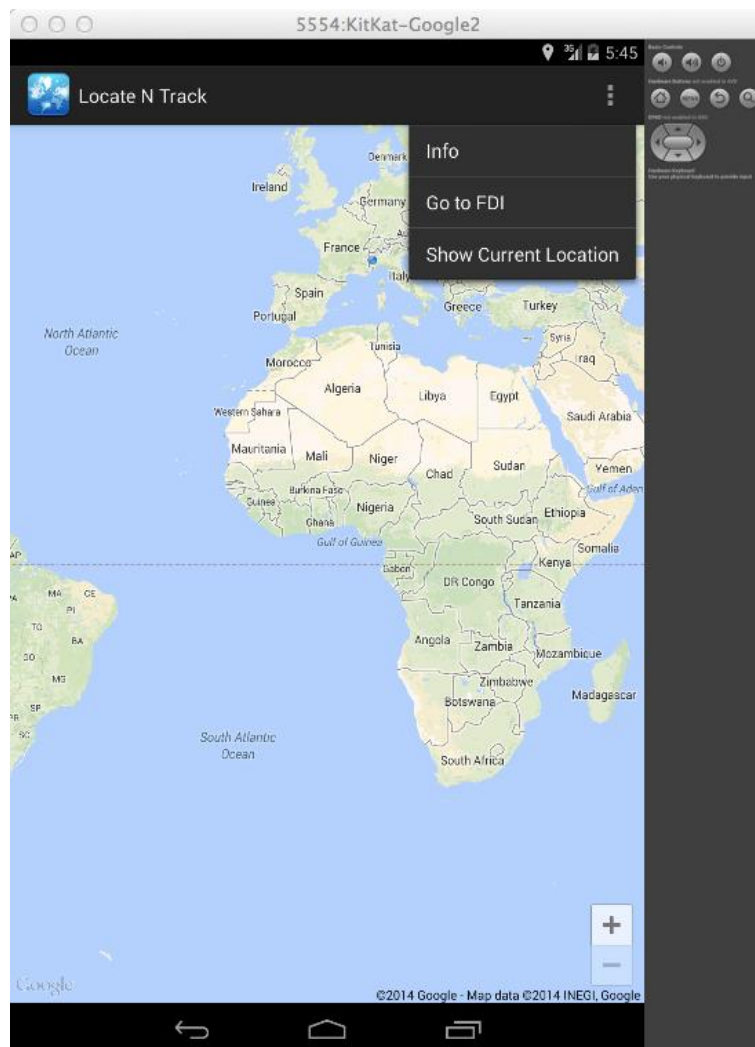


ILUSTRACIÓN 17. VISIÓN DE LA POSICIÓN DEL DISPOSITIVO

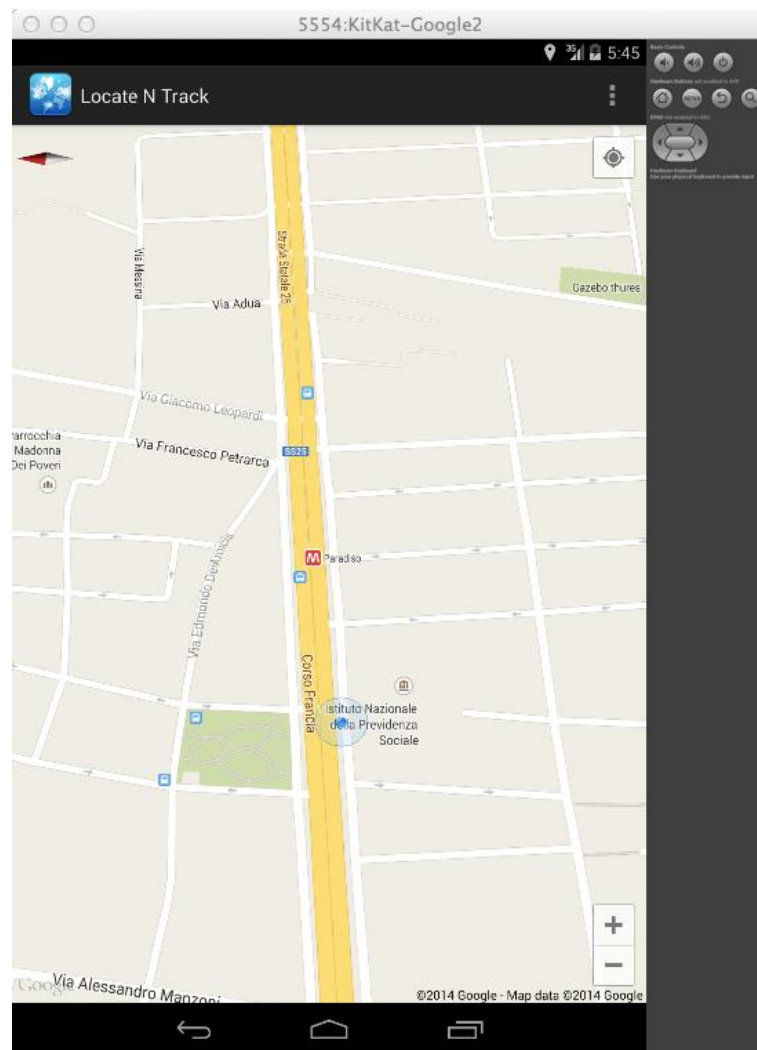


ILUSTRACIÓN 18. SIMULACIÓN DE UBICACIÓN

*\*se puede simular una localización falsa, o una trayectoria falsa por la que pasa el dispositivo cargando un fichero de rutas GPX o KML desde el IDE*

## 6.2.2. iOS App

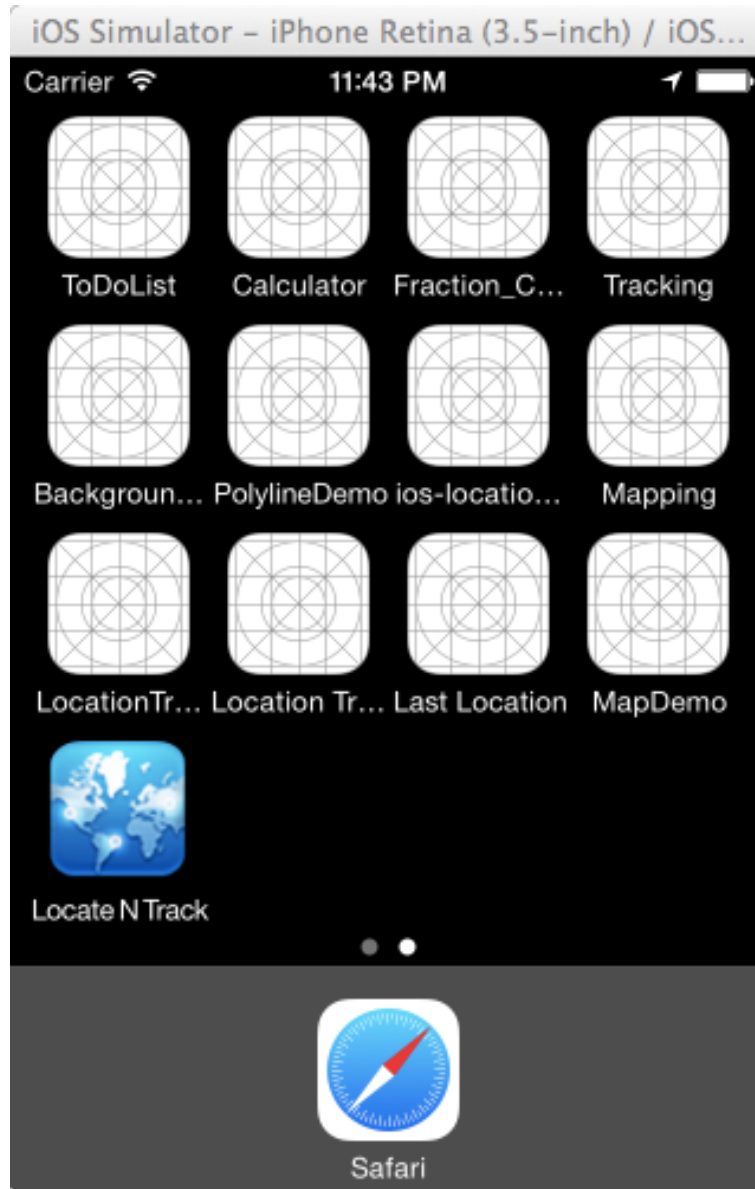


ILUSTRACIÓN 19. APLICACIÓN

Abrir la aplicación haciendo clic en el icono *Locate 'N' Track*

Se abre la aplicación y automáticamente comenzará el envío de la localización al servidor.



Se muestra un mensaje informando de que el servicio de localización se encuentra activado.

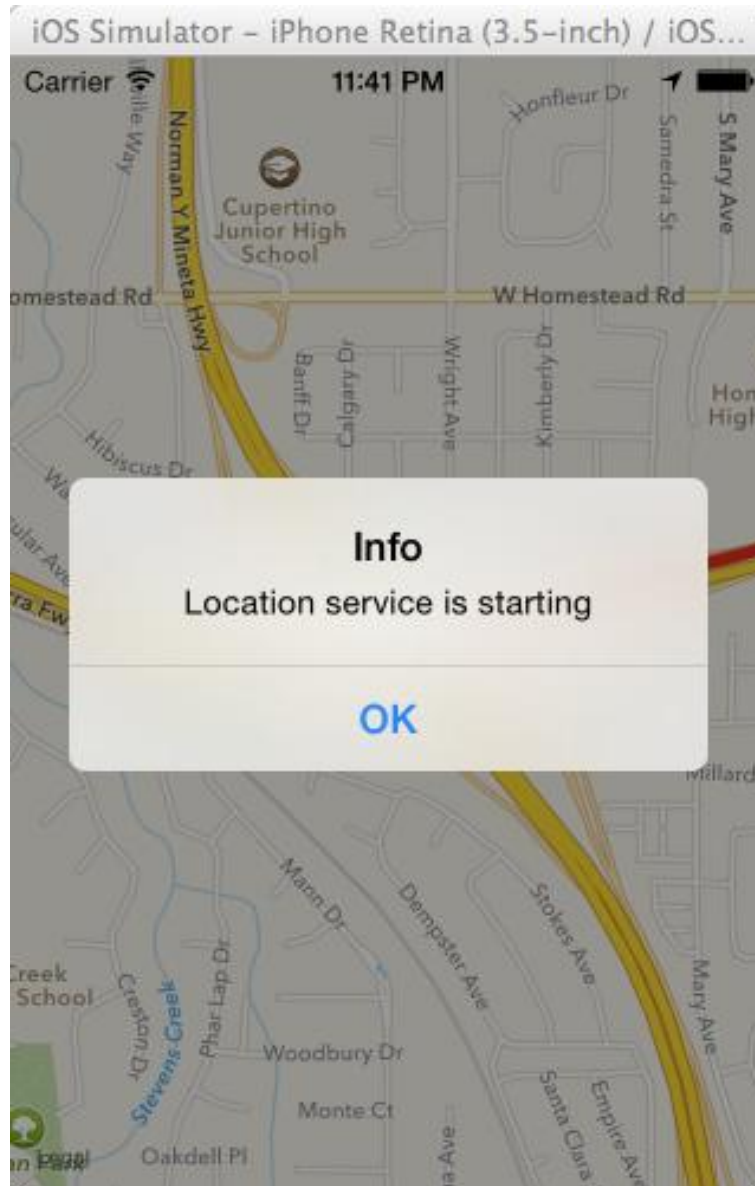


ILUSTRACIÓN 20. MENSAJE INFORMATIVO

Tras ello, se muestra el mapa con la posición del dispositivo.

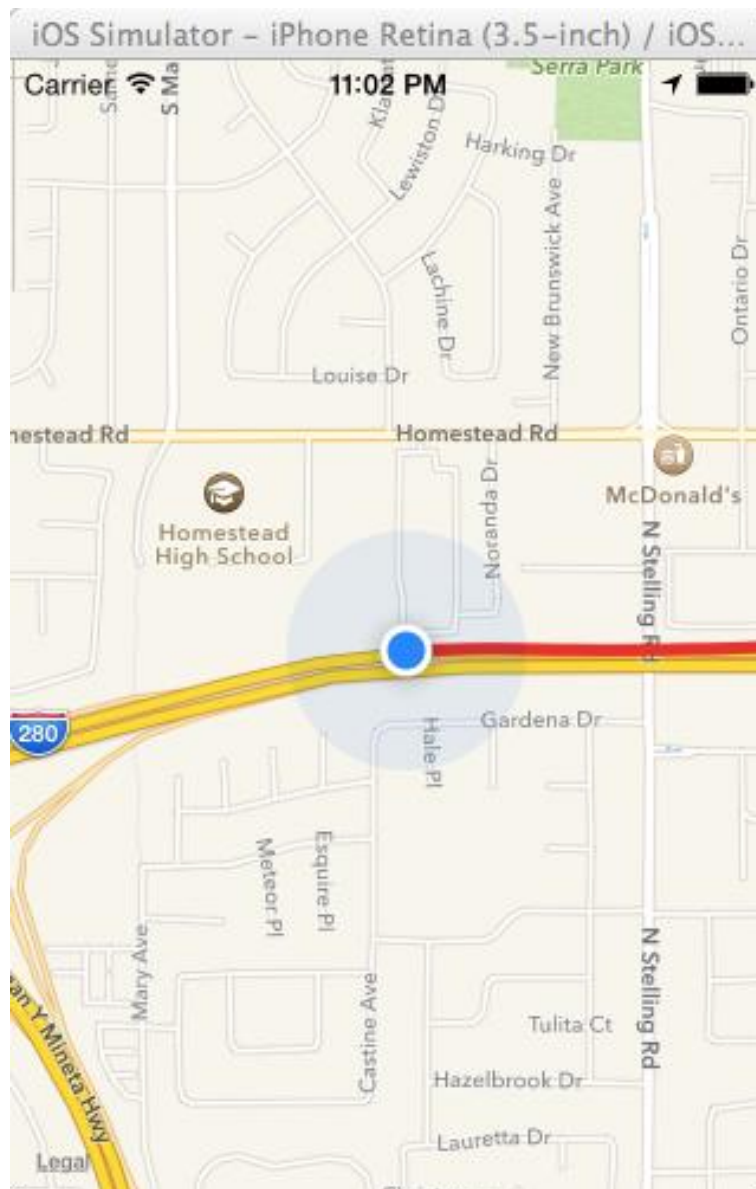


ILUSTRACIÓN 21. POSICIÓN DEL DISPOSITIVO

*\*Se puede simular una localización falsa, o una trayectoria falsa por la que pasa el dispositivo desde el menú Debug -> Simulate Location y luego añadiendo un fichero GPX o rutas prediseñadas.*

### 6.3. Localizar los dispositivos

Accediendo desde un navegador web ingresando la dirección del servidor y el puerto. En nuestro caso como lo estamos ejecutando desde local, desde el puerto por defecto 1188 escribiremos:

*<http://localhost:1188>*

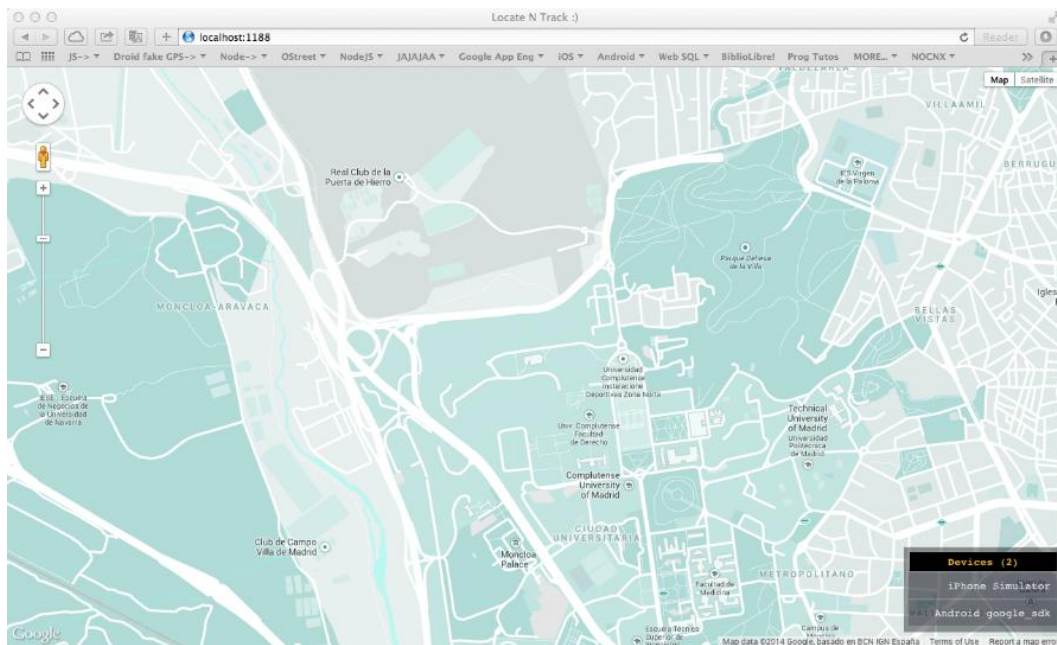


ILUSTRACIÓN 22. LOCALIZACIÓN DISPOSITIVO DESDE NAVEGADOR

Se muestra en la parte inferior derecha los dispositivos activos que se encuentran emitiendo su localización hacia nuestro servidor, en este caso un simulador de iPhone y un simulador de Android.

Si seleccionamos cualquiera de ellos, el mapa se actualiza mostrando la posición actual.

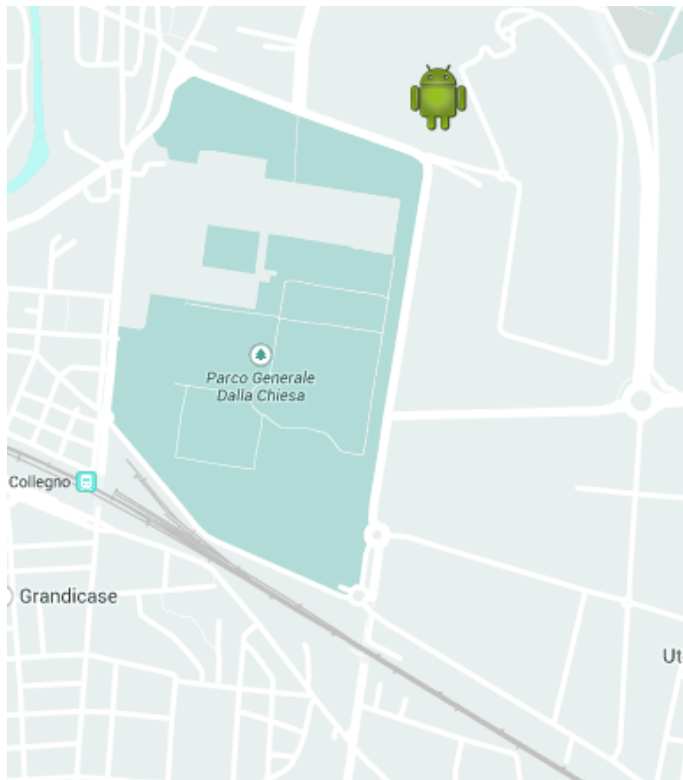


ILUSTRACIÓN 23. LOCALIZACIÓN  
DISPOSITIVO



ILUSTRACIÓN 24. LOCALIZACIÓN DISPOSITIVO

Si el usuario desea un obtener una traza para los dispositivos de todos los sitios por los que ha ido pasando y su respectiva marca de tiempo, puede acceder al archivo situado en /server/log.log.

```

{"level":"info","message":{"device":"iPhone Simulator","lat":"37.332529","long":
-122.055717},"timestamp":"2014-09-01T23:08:59.118Z"}
{"level":"info","message":{"device":"Android google_sdk","lat":"45.0791100000001","long":
7.55913000000001},"timestamp":"2014-09-01T23:09:04.103Z"}
{"level":"info","message":{"device":"iPhone Simulator","lat":"37.332529","long":
-122.055717},"timestamp":"2014-09-01T23:09:05.118Z"}
{"level":"info","message":{"device":"Android google_sdk","lat":"45.0791100000001","long":
7.55913000000001},"timestamp":"2014-09-01T23:09:10.102Z"}
{"level":"info","message":{"device":"iPhone Simulator","lat":"37.332529","long":
-122.055717},"timestamp":"2014-09-01T23:09:11.117Z"}
{"level":"info","message":{"device":"Android google_sdk","lat":"45.0791100000001","long":
7.55913000000001},"timestamp":"2014-09-01T23:09:16.105Z"}
{"level":"info","message":{"device":"iPhone Simulator","lat":"37.332529","long":
-122.055717},"timestamp":"2014-09-01T23:09:17.118Z"}
{"level":"info","message":{"device":"Android google_sdk","lat":"45.0791100000001","long":
7.55913000000001},"timestamp":"2014-09-01T23:09:22.103Z"}
{"level":"info","message":{"device":"iPhone Simulator","lat":"37.332529","long":
-122.055717},"timestamp":"2014-09-01T23:09:23.115Z"}
{"level":"info","message":{"device":"Android google_sdk","lat":"45.0791100000001","long":
7.55913000000001},"timestamp":"2014-09-01T23:09:28.102Z"}
{"level":"info","message":{"device":"iPhone Simulator","lat":"37.332529","long":
-122.055717},"timestamp":"2014-09-01T23:09:29.120Z"}
{"level":"info","message":{"device":"Android google_sdk","lat":"45.0791100000001","long":

```

ILUSTRACIÓN 25. TRAZA DEL RECORRIDO DEL DISPOSITIVO

## 7. Resultado

Este apartado trata de mostrar una crítica al método de trabajo empleado, la organización para llevar a cabo las tareas y las conclusiones alcanzadas entre otras cosas.

### 7.1. Metodología de trabajo

En primer lugar, subrayar que al no contar con un director que pudiera guiarme no ha sido posible realizar reuniones ni para definir los requisitos ni para adecuar el proyecto a ningún método de desarrollo del software.

De modo que, decidí llevar a cabo un método de desarrollo iterativo e incremental partiendo de unos objetivos iniciales que consideré oportunos y realicé el análisis, diseño e implementación de estos requisitos

Una vez determinado esto, el objetivo era documentarse acerca de las herramientas a utilizar, y a realizar tareas de autoaprendizaje. Para el seguimiento del desarrollo inicialmente se empezó a utilizar la herramienta de Bitbucket pero rápidamente cayó en desuso por la ausencia de colaboración.

En cuanto a las herramientas utilizadas, al ser un sistema programado en multitud de lenguajes, se ha utilizado el IDE de Eclipse con diversos plugins y XCode.

## 7.2. Líneas de trabajo futuro

Este proyecto admite una cantidad grande de posibles extensiones. Muchas de las cuales se plantearon al comienzo pero no han podido llevarse a cabo.

En los siguientes apartados se muestran las propuestas.

### 7.2.1. Aplicaciones como demonios

Esta ampliación permitiría que las app's se iniciaran en segundo plano nada más iniciarse el dispositivo sin necesidad de ejecutarlas manualmente

### 7.2.2. Acciones sobre los dispositivos

Esta funcionalidad interesante permitiría enviar mensajes al dispositivo, o borrar datos del mismo desde la página web.

### 7.2.3. Log-in

Debido a la pretensión de sencillez, el logueo no está implementada ni en la web ni en los clientes de los dispositivos. Conllevaría mantener una base de datos de los clientes y una fase de autenticación por parte de los dispositivos al momento de instalar la aplicación.

#### 7.2.4. Rastreo desde otra app

No es tan interesante pues todos los terminales móviles del mercado cuentan con un navegador web incorporado desde el que poder acceder al mapa que ofrece el servidor. Pero sí sería una ventaja, si permitiera mostrar la ubicación actual del usuario y la de otro dispositivo del cual tuviéramos sus credenciales de acceso.

#### 7.2.5. Llévame hasta mi dispositivo

Se trataría de mostrar señalizar la posición actual en la que se encuentra el usuario y proponer una ruta hasta el dispositivo. Tanto para la web como para la aplicación móvil.



## 8. Conclusiones

Desde el punto de vista de negocio, el sistema muestra un ejemplo de lo que podría ser un rastreador de dispositivos móviles para distintos sistemas operativos de fácil uso y libre de la intromisión de grandes compañías o gobiernos, que permite acceder a la localización instantánea de los mismos.

Respecto a la metodología de trabajo, comentar que parte de los objetivos no conseguidos se deben a la falta de rigor en la metodología de desarrollo. Se han incumplido innumerables hitos intermedios, se han acumulado retrasos y no se ha evaluado los riesgos derivados de éstos. Hacer en unos días el trabajo de semanas se ha revelado como una mala práctica que puede que haya pasado factura al final del proyecto. Consecuencias que se lamentan pero que se tendrán muy presente en el futuro, como la optimización de la planificación y las métricas, el desarrollo incremental y algunas buenas prácticas como el desarrollo guiado por pruebas.

Detalle a señalar y que afectó al trabajo fue la falta de atención por parte del director del proyecto desde el principio del curso. Su falta de interés me llevó a prolongar excesivamente fases decisivas del proyecto, con la consiguiente falta de tiempo para cumplir con los objetivos originales. Tratándose de un proyecto ambicioso y muy completo, en el que hay que aprender una enorme cantidad de tecnologías, sumado a la ausencia de compañero, resultó un hándicap para el desarrollo.

Con respecto a las tecnologías empleadas, decir que debido a la cuasi-ausencia de un director, se han investigado y probado numerosas. Muchas de las cuales, se han visto rechazadas después de haber supuesto un trabajo importante tras haberlas estudiado.

Android estaba claro desde el principio pero la inclusión de iOS no se decidió hasta el final. Un sistema que rastrea dispositivos móviles,

debe al menos monitorizar terminales con los sistemas operativos móviles más ampliamente utilizados en el mundo a día de hoy.

En cuanto al desarrollo del servidor, finalmente se decidió usar Node.js una plataforma basada en el motor de JavaScript V8 de Google. Está pensada para facilitar el desarrollo de aplicaciones basadas en red, rápidas y fiables. Node.js utiliza un modelo I/O (entrada/salida) orientado a eventos y basado en el no-bloqueo que lo hace ligero y eficiente, ideal para la aplicación que hace uso de datos intensivos y que se ejecuta a través de dispositivos distribuidos.

Nunca antes se había trabajado con ninguna de estas tecnologías, por lo que desarrollar sobre ellas al comienzo resultó difícil. Por otro lado, existe mucha documentación y ejemplos en la web acerca de su uso por lo que el aprendizaje no se hizo excesivamente severo.

A pesar de estos inconvenientes, todos me han servido de mucha utilidad para ampliar mis conocimientos informáticos y ahondar en numerosos conceptos de la ingeniería del software.

En lo personal, se extraen un gran número de aspectos en los que se ha fracasado, como la optimización de la planificación y las métricas, el desarrollo incremental y algunas buenas prácticas como el desarrollo guiado por pruebas que si bien no pudieron llevarse a cabo, sin duda hubieran resultado beneficiosas.

He aprendido que es muy importante seguir una buena metodología de trabajo para lograr un buen desarrollo, lo que hace que al final todo sean facilidades, la cual debió haber sido más estricta y con el empleo de alguna herramienta de seguimiento de errores, dada la importancia que esta tiene para sacar conclusiones al final del proyecto. Algo de ayuda y orientación hubiera sido muy bien recibida.

Desde el punto de vista de la ingeniería del software, destacar lo importante que es modular un proyecto, de tal forma que sea sencillo y trivial acoplar y desacoplar módulos. También el análisis de

requisitos y elaboración de casos de uso y escenarios, a partir de los cuales es más sencilla diseñar e implementar los módulos que intervienen en cada uno de ellos.

Ha sido un ejemplo de qué se debe hacer pero sobre todo, qué no se debe hacer y estoy seguro de que al igual que he visto mejorada la capacidad de desarrollo de sistemas informáticos, las vicisitudes vividas durante este año harán de mi un mejor ingeniero informático.

## Bibliografía

- [1] Java (2013) *Java Platform Standard Edition 7 Documentation* [en línea].  
Available: <http://docs.oracle.com/javase/7/>
- [2] Google (noviembre 2013) [en línea].  
Available: <http://developer.android.com/develop/index.html>
- [3] Google (noviembre 2013) *API de Google Maps para Android: biblioteca externa* [en línea].  
Available:  
<https://developers.google.com/maps/documentation/android/>
- [4] Google (febrero 2014) *Versión 3 del API de JavaScript de Google Maps* [en línea].  
Available:  
<https://developers.google.com/maps/documentation/javascript/>
- [5] Apple (julio 2014) *iOS 7 Design resources. Learn how to transition your app's user interface* [en línea].  
Available: <https://developer.apple.com/library/ios/navigation/>
- [6] Apple (julio 2014) *Programming with Objective-C* [en línea].  
Available:  
[https://developer.apple.com/library/ios/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html#//apple\\_ref/doc/uid/TP40011210-CH1-SW1](https://developer.apple.com/library/ios/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html#//apple_ref/doc/uid/TP40011210-CH1-SW1)
- [7] M. Gallagher (julio 2014) *Matt Gallagher programming cocoa with love* [en línea].  
Available: <http://www.cocoawithlove.com>

- [8] W3C (noviembre 2013) *Community whose mission is to lead the Web to its full potential*. [en línea].  
Available: <http://www.w3.org/>
  
- [9] HTML5 Rocks (febrero 2014) *Creating an experience for a small screen requires different thinking* [en línea].  
Available: <http://www.html5rocks.com/en/>
  
- [10] jQuery (junio 2014) *jQuery API* [en línea].  
Available: <http://api.jquery.com>
  
- [11] Node.js (julio 2014) *Node.js v0.10.31 Manual & Documentation* [en línea].  
Available: <http://nodejs.org/documentation/api/>
  
- [12] Express (julio 2014) *Web application framework for node* [en línea].  
Available: <http://expressjs.com/3x/api.html>
  
- [13] Socket.io (julio 2014) *The fastest and most reliable real-time engine* [en línea].  
Available: <http://socket.io/docs/>
  
- [14] Winston (agosto 2014) *Multi-transport async logging library for node.js* [en línea].  
Available: <https://github.com/flatiron/winston>
  
- [15] StackOverflow (2013-2014) *Question and answer site for professional and enthusiast programmers* [en línea].  
Available: <http://stackoverflow.com>
  
- [16] S. G. Kochan. *Programming in Objective-C*. 6a ed. Ed. Addison-Wesley. 2014.

- [17] C. Keuer, A. Hillegass & J. Conway. *iOS Programming. The big nerd ranch guide*. 4a ed. Ed. Big nerd ranch. 2014.



## *Locate 'N' track*

Localiza y rastrea tus dispositivos  
móviles desde cualquier parte

Proyecto fin de carrera realizado por Alejandro  
Blasco de Miguel para Ingeniería Informática  
en la asignatura de Sistemas Informáticos