

TeenControl

MEMORIA TÉCNICA



David Galindo Jiménez

IT Informática de Sistemas 2003

Consultores: Marc Domingo Prieto

Antonio Rodríguez Gutiérrez

Entrega: 06/2015



Esta obra está sujeta a una licencia de [Reconocimiento – NoComercial – SinObraDerivada \(by-nc-nd\)](https://creativecommons.org/licenses/by-nc-nd/4.0/):
No se permite un uso comercial de la obra original ni la generación de obras derivadas.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>TeenControl</i>
Nombre del autor:	<i>David Galindo</i>
Nombre del consultor:	<i>Marc Domingo Prieto</i>
Fecha de entrega:	<i>06/2015</i>
Área del Trabajo Final:	<i>Desarrollo aplicaciones dispositivos móviles (Android)</i>
Titulación:	<i>IT Informática de Sistemas 2003</i>

Resumen del Trabajo:

TeenControl es un proyecto tecnológico dentro del marco de trabajo de final de carrera que pretende ofrecer una aplicación para la plataforma Android.

El proyecto se lleva a cabo con la finalidad de ofrecer al usuario una solución mediante la cual se le facilite una serie de herramientas de control sobre uno o varios sujetos.

Este control viene motivado por varias necesidades, entre las que se incluyen la necesidad de los padres o tutores de ejercer una vigilancia positiva sobre sus hijos para garantizar su seguridad y comprobar una correcta conducta por parte de los mismos. Dentro de este enfoque positivo también se puede cubrir la necesidad de controlar a personas mayores que ya no gozan de todas sus facultades mentales o físicas.

Los usuarios podrán conocer la posición exacta de cada uno de sus sujetos localizados en un mapa cartográfico. Además podrán visualizar información relacionada relevante como puede ser la dirección exacta en la que se encuentra el sujeto, o la fecha y hora del posicionamiento observado.

Se podrán consultar las rutas realizadas por cada uno de los sujetos en días concretos.

También se ofrece al sujeto localizado la herramienta de activar una alerta de pánico la cual se hace llegar al usuario que controla a dicho sujeto.

Abstract (in English, 250 words or less):

TeenControl is a technology final year project that aims to provide an application for the Android platform.

The project is carried out in order to offer users a solution whereby it supplies control tools over one or more targets.

This control is motivated by various needs, among which include the need for parents or guardians to exercise a positive supervision over their children to ensure their safety and verify proper behavior from them. Around this positive approach you can also cover the need to control older people who no longer enjoy all his mental or physical faculties.

Users can know the exact position of each of their objectives under surveillance on a cartographic map and see relevant related information such as the exact address, date, and positioning time.

It will be possible to consult the routes taken by each of the targets on specific days or offer to supervised person the possibility of activate a panic alarm which will arrive to the electronic mail of his manager.

Palabras clave (entre 4 y 8):

Seguimiento, control, vigilancia, geolocalización, GPS, cartografía

Índice de contenidos

1. INTRODUCCIÓN.....	1
1.1. Contexto y justificación del Trabajo.....	1
1.2. Objetivos del Trabajo.....	1
1.3. Enfoque y método de desarrollo elegido.....	2
1.4. Planificación del Trabajo	2
1.4.1. Recursos	2
1.4.2. Coste recursos	3
1.4.3. Tareas	4
1.4.4. Planificación de las tareas.....	5
1.5. Breve sumario de productos obtenidos.....	7
2. ETAPA DE ANÁLISIS: Usuarios y contextos de uso	7
2.1. Métodos de indagación utilizados	7
2.1.1. Justificación y planteamiento de los métodos elegidos	7
2.1.2. Desarrollo, resultados y conclusiones de los métodos elegidos.....	8
2.1.2.1. Dinámicas de grupo.....	8
2.1.2.2. Análisis competitivo	10
2.2. Perfiles de usuarios identificados.....	12
2.2.1. Perfil y descripción	12
2.2.2. Factores demográficos.....	13
2.2.3. Contextos de uso.....	15
2.2.4. Relación de tareas agrupadas por áreas y objetivos	15
2.3. Resumen de necesidades detectadas en el análisis	16
2.3.1. Necesidades a nivel de implementación	16
2.3.2. Necesidades a nivel de diseño.....	17
3. ETAPA DE DISEÑO: Diseño conceptual	17
3.1. Personajes.....	17
3.2. Escenarios de uso	18
3.3. Flujos de interacción.....	20
4. PROTOTIPADO: Sketches y prototipos horizontales de alta fidelidad.....	21
4.1. Autenticación	22
4.1.1. Autenticación: Sketches	22
4.1.2. Autenticación: Prototipos horizontales	22
4.2. Área de Localización	24
4.2.1. Área de Localización: Sketches	25
4.2.2. Área de Localización: Prototipos horizontales	25
4.3. Área de Gestión.....	26
4.3.1. Área de Gestión: Sketches.....	27
4.3.2. Área de Gestión: Prototipos horizontales.....	27
5. ETAPA DE EVALUACIÓN DE PROTOTIPOS.....	29
5.1. Entorno para la evaluación.....	29
5.2. Cuestionario a candidatos para ser testadores.....	29
5.3. Tareas de validación	30
5.4. Cuestionario relacionado con las tareas	31
6. ETAPA DE DISEÑO: Diseño Técnico	31
6.1. Definición de los casos de uso	31
6.2. Diseño de la arquitectura	35

6.2.1.	Diseño de la BBDD.....	35
6.2.2.	Diseño de entidades y clases.....	35
6.2.3.	Arquitectura del sistema.....	37
7.	IMPLEMENTACIÓN.....	37
7.1.	Entorno de desarrollo elegido.....	38
7.2.	Solución de Hosting.....	38
7.3.	Modificaciones respecto a la fase de prototipado.....	39
7.4.	Diseño de los Webservices.....	40
7.5.	Solución cartográfica.....	41
7.5.1.	Descripción de la tecnología.....	41
7.5.2.	Comentarios relevantes sobre código relacionado.....	42
7.6.	Localización de los sujetos.....	43
7.6.1.	Obtención de las localizaciones.....	43
7.6.2.	Control desactivación por parte del sujeto de la localización.....	44
7.7.	Tratamiento información devuelta por los Webservices.....	45
7.8.	Botón de pánico.....	46
7.9.	Localizaciones del sujeto en el mapa.....	47
7.9.1.	Localización Global.....	48
7.9.2.	Localización específica.....	48
7.9.3.	Ruta diaria de sujeto.....	49
7.10.	Implementación de control del nivel de batería del dispositivo localizado.....	49
7.10.1.	Nivel de batería.....	50
7.10.2.	Indicador gráfico de la batería.....	50
7.10.3.	Notificación de nivel crítico de batería.....	51
7.11.	Otras implementaciones.....	52
7.11.1.	Registro.....	52
7.11.2.	Recordar contraseña.....	52
7.11.3.	Alta y baja de sujetos.....	53
7.11.3.1.	Alta de sujetos.....	53
7.11.3.2.	Baja de sujetos.....	53
7.11.4.	Sincronización entre Activities de sujetos.....	53
7.11.5.	Implementación de controles.....	54
8.	CONCLUSIONES.....	55
9.	GLOSARIO.....	56
10.	BIBLIOGRAFÍA.....	58

Listado de figuras

Figura 1: Planificación proyecto - 1	5
Figura 2: Planificación proyecto - 2	Error! No s'ha definit l'adreça d'interès.
Figura 3: Pantalla inicial Pet Locator	Error! No s'ha definit l'adreça d'interès.
Figura 4: Marcador GPS Tracker 365	Error! No s'ha definit l'adreça d'interès.
Figura 5: Botones tipos de mapa Buscar mi Iphone	Error! No s'ha definit l'adreça d'interès.
Figura 6: Control de batería Alert.Us – Localización	Error! No s'ha definit l'adreça d'interès.
Figura 7: Intervalo de actualización Find my kids – Footprints	Error! No s'ha definit l'adreça d'interès.
Figura 8: Distancia mínima recorrida Find my kids – Footprints	Error! No s'ha definit l'adreça d'interès.
Figura 9: Ruta Buscar mi Iphone	Error! No s'ha definit l'adreça d'interès.
Figura 10: Estadística penetración Smartphones países desarrollados	Error! No s'ha definit l'adreça d'interès.
Figura 11: Penetración Smarthphones España por edades	Error! No s'ha definit l'adreça d'interès.
Figura 12: Previsión mayores de 65 años 2000-2020	Error! No s'ha definit l'adreça d'interès.
Figura 13: Porcentaje condenados juveniles en España	Error! No s'ha definit l'adreça d'interès.
Figura 14: Media inicio consumo alcohol y cannabis	Error! No s'ha definit l'adreça d'interès.
Figura 15: Personaje Gestor Laura Pérez Delgado	Error! No s'ha definit l'adreça d'interès.
Figura 16: Persona adolescente localizada Sara Martínez Delgado	Error! No s'ha definit l'adreça d'interès.
Figura 17: Flujos interacción TeenControl	Error! No s'ha definit l'adreça d'interès.
Figura 18: Sketch autenticación pantalla inicial	22
Figura 19: Sketch autenticación registro	22
Figura 20: Sketch autenticación recordar contraseña	22
Figura 21: Sketch autenticación ayuda autenticación	Error! No s'ha definit l'adreça d'interès.
Figura 22: Sketch menú principal TeenControl	Error! No s'ha definit l'adreça d'interès.
Figura 23: Prototipo horizontal pantalla inicial	23
Figura 24: Prototipo horizontal registro 1	23
Figura 25: Prototipo horizontal registro 2	23
Figura 26: Prototipo horizontal recordar contraseña 1	23
Figura 27: Prototipo horizontal recordar contraseña 2	23
Figura 28: Prototipo horizontal ayuda autenticación	24
Figura 29: Prototipo horizontal menú principal TeenControl	24
Figura 30: Sketch área de localización	25
Figura 31: Sketch alta sujeto a rastrear	25
Figura 32: Sketch baja sujeto rastreado	25
Figura 33: Sketch activar localización	25
Figura 34: Prototipo horizontal área de localización	25
Figura 35: Prototipo horizontal alta y baja sujeto a rastrear 1	26
Figura 36: Prototipo horizontal alta y baja sujeto a rastrear 2	26
Figura 37: Prototipo horizontal alta y baja sujeto a rastrear 3	26
Figura 38: Prototipo horizontal activar localización 1	26
Figura 39: Prototipo horizontal activar localización 2	26
Figura 40: Prototipo horizontal activar localización 3	26
Figura 41: Prototipo horizontal activar localización 4	26
Figura 42: Sketch área de gestión	27
Figura 43: Sketch área de gestión localizar	27
Figura 44: Sketch área de gestión ruta	27
Figura 45: Sketch fijar hora de retorno	27
Figura 46: Prototipo horizontal área de gestión 1	27
Figura 47: Prototipo horizontal área de gestión 2	27
Figura 48: Prototipo horizontal localizar 1	28
Figura 49: Prototipo horizontal localizar 2	28
Figura 50: Prototipo horizontal ver ruta	28

Figura 51: Prototipo horizontal hora de retorno 1	28
Figura 52: Prototipo horizontal hora de retorno 2	28
Figura 53: Prototipo horizontal hora de retorno 3	28
Figura 54: Prototipo horizontal hora de retorno 4	28
Figura 55: Justinmind Prototyper simulación prototipos 1	29
Figura 56: Justinmind Prototyper simulación prototipos 2	29
Figura 57: Justinmind Prototyper simulación prototipos 3	29
Figura 58: Justinmind Prototyper simulación prototipos 4	29
Figura 59: Diadrama de BBDD	35
Figura 60: Diagrama de entidades y clases	36
Figura 61: Patrón MVC	37
Figura 62: Visión tablas BBDD en PhpMyadmin	39
Figura 63: Complemento IP exclusiva del Hosting	39
Figura 64: ID y número del proyecto en la consola de Google	42
Figura 65: API Google Maps Android API v2 activada	42
Figura 66: Permisos API Google Maps Android API v2 activada	42
Figura 67: Funcionalidad botón alerta de pánico de TeenControl	46
Figura 68: Ejemplo correo alerta de pánico 1	47
Figura 69: Ejemplo correo alerta de pánico 2	47
Figura 70: Funcionalidad localización global de TeenControl	48
Figura 71: Funcionalidad localización específica de TeenControl	48
Figura 72: Funcionalidad ruta de un sujeto de TeenControl	49
Figura 73: Representación gráfica del nivel de batería del dispositivo del sujeto	51
Figura 74: Ejemplo correo nivel crítico de batería	51
Figura 75: Ejemplo correo recovery del nivel crítico de batería	52
Figura 76: Ejemplo correo recordatorio de contraseña	53
Figura 77: Control registro cuenta gestor ya dada de alta	55
Figura 78: Control no acceso a Internet o al Servidor	55
Figura 79: Control ningún proveedor de localización activado	55
Figura 80: Control proveedor de localización favorito no activado	55
Figura 81: Control ningún sujeto dado de alta	55
Figura 82: Control no posiciones en BBDD de los sujetos	55
Figura 83: Libro Android 4. Desarrollo de aplicaciones	58

1. INTRODUCCIÓN

1.1. Contexto y justificación del Trabajo

En primer lugar me gustaría explicar cuáles han sido las motivaciones que me han hecho elegir el área de TFC-Desarrollo de aplicaciones para dispositivos móviles y no otra.

La opción fácil hubiese sido elegir un área de TFC relacionada con la administración de sistemas, como podría ser el área TFC-Administración de redes y sistemas operativos, ya que es la rama a la que me dedico profesionalmente. Pero lo cierto es que quería profundizar en un área en la que no tuviera conocimientos tan amplios. Se da la circunstancia que ocasionalmente, como hobby, había realizado algún desarrollo y si a esto le sumamos que hace tiempo que me llamaba la atención el realizar una aplicación en Android, para publicarla en el Play Store, hace que me haya inclinado definitivamente por la elección del área de TFC que nos ocupa.

La principal finalidad, del presente proyecto, es la elaboración de una aplicación para la plataforma Android que permita la localización, por parte de un colectivo, de uno o varios sujetos.

El target, hacia el que está enfocado el diseño de la interface de la herramienta, es el control de adolescentes por parte de sus padres, aunque también podría ser útil para controlar la actividad de personas mayores con trastornos mentales o minusvalías físicas. Con lo que podemos decir que la aplicación cubre las siguientes necesidades:

- Protección y control parental en el caso de la localización de adolescentes
- Garantizar la integridad física de personas que no están cualificadas como autónomas a nivel de toma de decisiones, como podría ser el caso de personas mayores con demencia senil o alzhéimer.

1.2. Objetivos del Trabajo

A parte del desarrollo de la aplicación con sus funcionalidades concretas, con la realización de este proyecto se pretende cumplir con los siguientes objetivos:

- Poner en práctica gran parte de los conocimientos adquiridos a lo largo de los estudios de Ingeniería Técnica en Informática de Sistemas
- Conocer de primera mano, el proceso completo de desarrollo de un software, desde la planificación inicial hasta el despliegue definitivo del producto
- Conseguir un producto final con la suficiente calidad como para poder ser incluido en el Play Store de Android y que se cumplan las expectativas de los usuarios
- Conseguir una integración correcta y funcional de las diferentes tecnologías que compondrán la solución a diferentes niveles (BBDD, Servicios Web, desarrollo Java de la aplicación,...)
- Adquirir nuevos conocimientos sobre temáticas y aplicaciones orientadas a la realización de desarrollos, así como la correcta utilización de las mismas

1.3. Enfoque y método de desarrollo elegido

Como se pretende poner en prácticas gran parte de los conocimientos adquiridos durante la carrera se opta por desarrollar una aplicación completamente nueva para cubrir las necesidades detalladas.

Dado que mi área de especialización es sistemas y no desarrollo, se ha enfocado el proyecto al uso de una metodología en la que obtengamos información empírica de una manera rápida, la cual nos permita probar parte de la solución en etapas tempranas, y que dichas pruebas sirvan para retroalimentar la solución de manera continua. Con este enfoque evito el darme cuenta que cuando llego a la fase de implementación no soy capaz de llevar a cabo la solución que he propuesto en la fase de análisis y diseño (Típico problema de los métodos basados en el ciclo de vida clásico o en cascada).

Es por este motivo que he decidido trabajar con un método de desarrollo iterativo e incremental. Por lo que se van desarrollando cada una de las funcionalidades como proyectos concretos, lo que me permite tener resultados parciales utilizables. Lógicamente al principio se hace más hincapié en detectar los requisitos y el análisis y diseño necesario para llevar a cabo los mismos, y en las últimas fases se presta más atención a la implementación y las pruebas, de manera que se acabe de depurar toda la solución y que se interrelacionen correctamente cada una de la implementaciones desarrolladas por separado.

1.4. Planificación del Trabajo

En este apartado se abordarán todos los aspectos que se han de tener en cuenta a la hora de planificar el proyecto teniendo en cuenta de los recursos que se ha de disponer.

1.4.1. Recursos

A continuación detallaré los recursos, tanto a nivel de hardware como de software, necesarios para la consecución del proyecto. También se detallaran los servicios contratados vinculados a la operatividad de la solución.

a) Hardware:

Para llevar a cabo el desarrollo se utilizara un equipo de gama alta cuyas prestaciones permitan un trabajo fluido a la hora de realizar las pruebas en el dispositivo virtual Android:

Dell Inspiron 17R SE:

Procesador: Intel Core i7-3630QM (4x2,40 GHz / 6 MB caché).

Pantalla / Resolución: 17,3" mate / 1920x1080.

Memoria RAM / HDD: 8 GB RAM DDR3 / 1 TB (5400 RPM).

Tarjeta gráfica: nVidia GT 650m de 2 GB GDDR5 (dedicada).

Dimensiones / Peso: 417x276x38 mm / 3,28 kg.

Otras especificaciones: Windows 8, batería 6 celdas, 1xHDMI, 4xUSB3.0, Wifi b/g/n, grabador DVD-DL, Webcam 1,0 Mpx, lector tarjetas SD, Bluetooth 4.0, WiDi.

Para el entorno de pruebas real, se utilizará el siguiente Smartphone:

Motorola Moto G 2014:

Sistema operativo: Android 4.4.4 (KitKat)

Procesador: Qualcomm® Snapdragon™ 400 processor with 1.2 GHz quad-core CPU, Adreno 305 @450 MHz GPU

Memoria RAM: 1GB

Almacenamiento: Single SIM - 8GB & 16GB versions

Pantalla: 5.0-inch 720p HD 720x1280 IPS, 294 ppi, Corning Gorilla Glass

Bateria: 2070 mAh

Otros: US GSM Model GSM/GPRS/EDGE (850, 900, 1800, 1900 MHz) UMTS/HSPA+ (850, 1700 (AWS), 1900 MHz)

Global GSM Model GSM/GPRS/EDGE (850, 900, 1800, 1900 MHz) UMTS/HSPA+ (850, 900, 1900, 2100 MHz)

Wifi 802.11b/g/n

b) Software:

Para la implementación de la solución se utilizará la siguiente relación de software gratuito:

- Kit de desarrollo de java (JDK)
- Entorno de desarrollo integrado (IDE) Eclipse
- Kit de desarrollo de Android (SDK)
- Herramientas de desarrollo de Android (ADT) + Herramientas de las plataformas (Platform tools)
- WAMP Server EasyPHP DevServer 14.1 VC1

c) Servicios de software contratados:

De cara a alojar la BBDD y los Webservices está prevista la contratación de:

- Un plan de hospedaje en www.hospedajeydominios.com denominado HyD (En apartado de implementación se dan más detalles)
- Complemento al hosting de una IP Exclusiva

1.4.2. Coste recursos

El coste asociado al proyecto en materia de recursos son 40 euros, 20 euros de plan de hospedaje más 20 euros del complemento de IP exclusiva, ya que ya se disponía de los 2 dispositivos que componen el hardware utilizado y el software ya mencionado es gratuito.

Si bien es cierto que se podrían obtener los mismos servicios instalando y configurando en local las herramientas que dan el soporte Web a la solución (Apache, PHP y Mysql), he valorado los siguientes aspectos a la hora de tomar la decisión de contratar dicho plan de hospedaje:

- Alta disponibilidad del entorno
- Escalabilidad del proyecto
- Panel de control muy completo para la gestión global de la solución de hosting
- Satisfacción con la empresa proveedora del servicio en proyectos anteriores

1.4.3. Tareas

A continuación enumeramos cada una de las tareas necesarias para poder finalizar el proyecto, agrupadas por ámbitos:

- a) **Introducción:** contexto y justificación del trabajo, objetivos del proyecto, enfoque y método de desarrollo elegido, planificación y sumario de productos obtenidos
- b) **Instalación y configuración del entorno de trabajo:** referente al software enumerado en el apartado b) del punto 1.4.
- c) **Análisis, diseño conceptual, prototipado + evaluación y diseño técnico de la aplicación:** Definición de usuarios y contextos de uso, diseño conceptual de escenarios de uso y flujos de interacción, prototipado y evaluación de los mismos. Definición de los casos de uso, diagrama UML de la BBDD y de las entidades y clases y diagrama explicativo de la arquitectura del sistema.
- d) **Implementación requisitos funcionales:** implementación de las funcionalidades desprendidas de las etapas de análisis y diseño conceptual y técnico.
- e) **Implementación requisitos no funcionales:** tareas de implementación de aquellas circunstancias que han de ser controladas mediante el desarrollo y que no son la finalidad del usuario
- f) **Tareas de BBDD:** Creación de las tablas que compondrán la BBDD, junto con sus claves primarias y foráneas más el desarrollo de los Webservices que interactuaran con la BBDD para llevar a cabo las transacciones de consulta, inserción, actualización y borrado.
- g) **Tareas de integración entre desarrollos parciales:** Contratación plan de hospedaje e IP exclusiva, importación de la estructura de BBDD en la BBDD del hospedaje e integración de los desarrollos parciales.
- h) **Tareas de validación:** Tareas de testeo y validación de la implementación.
- i) **Tareas de documentación finales:** revisión y finalización de la memoria técnica, junto con la inclusión de los apartados de conclusiones, glosario, bibliografía y anexos. Creación del manual de instrucciones de usuario y elaboración del video presentación.

1.4.4. Planificación de las tareas

A continuación plasmaré en un diagrama de Gant cada una de las tareas que se han detallado que son necesarias para conseguir cada uno de los objetivos del proyecto. Este diagrama ya contempla la planificación real del proyecto, detallando requisitos funcionales y no funcionales y la correcta temporización, ya que en fase de implementación se realizó un estudio de las desviaciones positivas y negativas y se rehizo la planificación temporal de tareas.

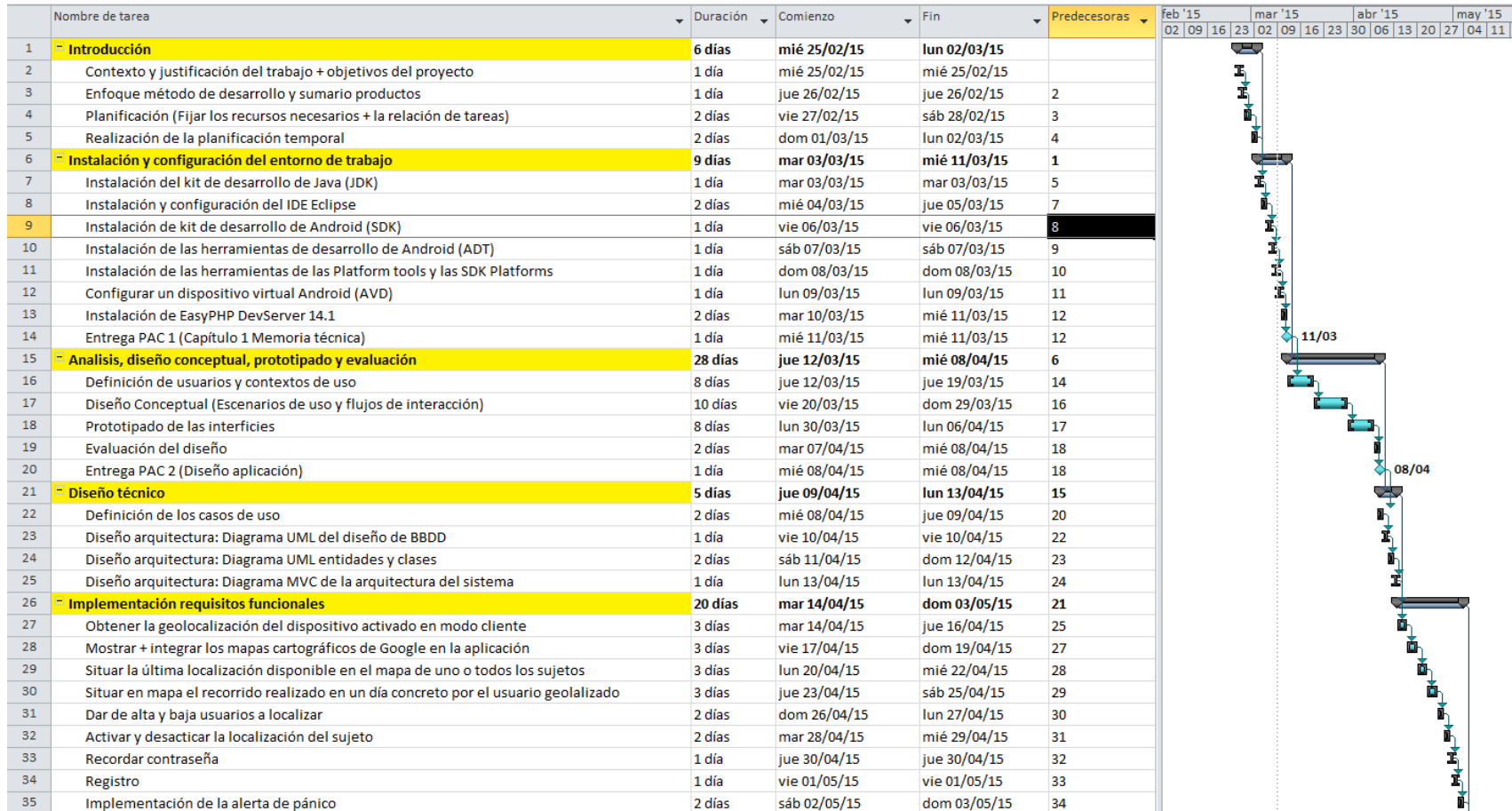


Figura 1

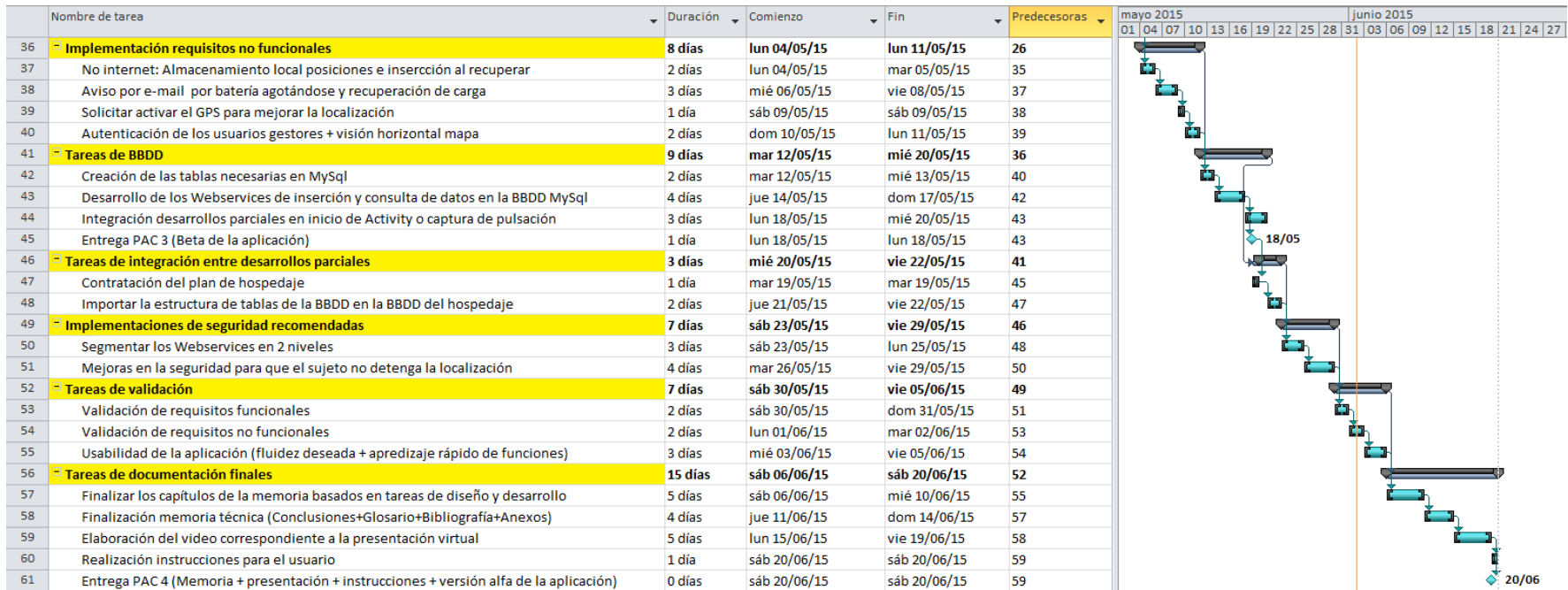


Figura 2

1.5. Breve resumen de productos obtenidos

El conjunto de entregables finales que compone la solución son los siguientes:

- Memoria técnica del proyecto: Documento que contiene toda la información relativa al proyecto. Información que va desde la justificación de la elección de la solución, objetivos del mismo, metodología usada y planificación, hasta el detalle de cada una de las fases de estudio e implementación de las soluciones particulares para objetivos concretos. La memoria se completa con las conclusiones, glosario, bibliografía y anexos.
- Presentación virtual: Perspectiva general del proyecto, ilustrada mediante un video, en el que se explican los aspectos generales de la solución y se muestran las funcionalidades y bondades de la aplicación.
- Código fuente: Se entrega el proyecto completo en el formato de carpetas de la herramienta de desarrollo integrado Eclipse, más el conjunto de ficheros PHP que componen los Webservices más una exportación de la BBDD Mysql.
- Paquete de aplicación Android (APK): Se entrega la aplicación ya compilada en formato APK para poder ser instalada y utilizada.

2. ETAPA DE ANÁLISIS: Usuarios y contextos de uso

En esta etapa lo que se pretende conseguir es definir quiénes serán los usuarios de la aplicación, las funcionalidades de la aplicación y los contextos en los que será usada la misma. También se analizan los aspectos de diseño a tener en cuenta tomando como referencia tanto la usabilidad, como la mejor manera de implementar cada una de las soluciones de cada una de las funcionalidades.

2.1. Métodos de indagación utilizados

2.1.1. Justificación y planteamiento de los métodos elegidos

Como la finalidad del proyecto no es implementar una aplicación que integra un conjunto de acciones o tareas que llevan a cabo normalmente un grupo de personas de una empresa o colectivo, sino que se trata de una herramienta que ofrece unas funcionalidades concretas bastante claras desde un principio, he optado por descartar los métodos de observación e investigación contextual, seguimiento, método del diario, entrevistas en profundidad y encuestas.

En primer lugar he elegido utilizar el método de dinámicas de grupo. El colectivo con el que he aplicado éste método ha sido con mis compañeros del departamento de Tecnología (7 personas), realizando 2 coloquios de aproximadamente 45 minutos de duración cada uno.

Por otro lado se ha utilizado el método del análisis competitivo para investigar productos existentes en el mercado con el fin de detectar la competencia, identificar puntos de mejora y tomar ideas sobre el diseño de las interfaces.

2.1.2. Desarrollo, resultados y conclusiones de los métodos elegidos

2.1.2.1. Dinámicas de grupo

1er Coloquio: La finalidad de este coloquio era detectar los requisitos funcionales iniciales, fijar cual sería el contexto de uso más habitual y detectar el mayor número de contextos de uso desfavorables, los cuales nos delimitarán nuestros requisitos no funcionales.

Guion del coloquio:

- Exposición por parte del moderador de los aspectos que se tratarán
- Tiempo dedicado a detectar los requisitos funcionales
- Tiempo dedicado a decidir cuál sería el contexto de uso habitual
- Tiempo dedicado a encontrar contextos de uso que pudiesen afectar al correcto funcionamiento tanto operativo como de accesibilidad y de usabilidad de la aplicación

Se llegaron a las siguientes conclusiones:

- 1) El contexto de uso habitual de la aplicación, será aquel condicionado por las tareas de configuración de la aplicación (que se corresponderán con requisitos funcionales):
 - Registrarse
 - Recordar contraseña
 - Dar de alta y baja los sujetos a localizar
 - Activar y desactivar localización de sujetos

A la aplicación se le dará un uso ocasional en el que el usuario se interrelacionará con las funcionalidades interactivas (También requisitos funcionales). Estas funcionalidades son:

- Ubicar a un sujeto o al conjunto de ellos en el mapa
- Ver el recorrido del sujeto localizado en un día concreto
- Activar la alerta de pánico por parte del sujeto localizado

También existirá una interacción no directa con la aplicación que será la de la lectura de los correos electrónicos de las siguientes casuísticas:

- Recordatorio de contraseña
- Carga de batería por debajo del 10% y recuperación de este umbral
- Lectura del correo con la alerta de pánico

El contexto de uso habitual será el del uso de la aplicación en lugares frecuentados normalmente como pueden ser el trabajo, el domicilio y el domicilio de amigos y familiares.

2) Los contextos de uso desfavorables son los siguientes:

- a) Coste económico para el usuario: El dispositivo localizado o el dispositivo del gestor pueden NO estar conectados a internet a través de una red Wifi gratuita y estar conectados a internet a través de la red móvil sin una tarifa plana
- b) Nivel de batería bajo (Entorno tecnológico): En periodos de tiempo grande el dispositivo localizado se puede encontrar con niveles de batería bajo

- c) No conexión a internet (Entorno tecnológico): Es posible que tanto los dispositivos localizados como los dispositivos de los gestores se encuentren en zonas donde la cobertura de red móvil sea pobre o inexistente y no existan redes Wifi a las que conectarse
- d) Tamaño de la pantalla del dispositivo (Entorno tecnológico): los dispositivos móviles de los usuarios gestores pueden disponer de pantallas de pequeño tamaño
- e) Ausencia de GPS (Entorno tecnológico): Existirán dispositivos móviles que no posean tecnología GPS
- f) Intento de boicot del sujeto (Factor humano): El sujeto podría intentar detener la localización

2º Coloquio: El objetivo de este segundo coloquio era la de fijar los perfiles de usuarios necesarios para la aplicación

Guion del coloquio:

- Explicación del planteamiento inicial del responsable del proyecto, en el que se hace constar que el planteamiento inicial es de 2 perfiles de usuarios “Gestores” y “Anónimos”. En el que el perfil de “Gestores” sería el que engloba a aquellos usuarios que se autentican en la aplicación y a partir de ese momento pueden dar de alta sujetos a localizar y utilizar todas las herramientas de gestión disponibles. Mientras que el perfil de “Anónimos”, sería el perfil que se utilizaría en el dispositivo del sujeto a localizar mientras este dispositivo está enviando a la BBDD las diferentes ubicaciones. De manera que desde este dispositivo no se pudiesen acceder a las herramientas de gestión.
- Tiempo dedicado a discernir si el planteamiento inicial de perfiles de usuarios es el adecuado.

Se llegó a la siguiente conclusión:

- 1) El perfil de usuarios “Anónimos” no era necesario. Y no solamente eso, sino que el perfil de usuario que se tenía que utilizar en el dispositivo a localizar era también el perfil de “Gestores”. Los motivos de esta decisión fueron los siguientes:
 - a) Para que el dispositivo envíe ubicaciones a la BBDD ha de saber quién es el usuario “Gestor” al que ha de asociar las ubicaciones del sujeto que porta el dispositivo localizado.
 - b) Como la herramienta permitiría a cada “Gestor” localizar a más de un sujeto simultáneamente (de ahí que pueda dar de alta a más de un sujeto), no solamente ha de autenticarse en la aplicación para que las ubicaciones se vinculen a dicho “Gestor”, sino que además, ha de ser el “Gestor” el que seleccione de sus sujetos dados de alta a quien va a entregar el dispositivo a localizar y una vez seleccionado inicie la localización del dispositivo.
 - c) Se ha de asegurar que el sujeto localizado no pueda interactuar con la aplicación y evitar que esta deje de localizar. Por este motivo se ha de autenticar el usuario “Gestor” y que la aplicación, una vez se inicie la localización, sea cerrada para evitar el detener la localización.

Estos motivos están orientados a un enfoque en el que la manipulación de los dispositivos se realiza directamente por los “Gestores” en el caso de padres, tutores e hijos que controlan a sus padres mayores. Pero si bien es cierto que no se creará un acceso a TeenControl con un usuario anónimo, el perfil del usuario anónimo sí que lo usaremos más adelante en la definición

de los escenarios de uso. Además el usuario localizado interactuará con TeenControl al pulsar el botón del pánico.

2.1.2.2. Análisis competitivo

En primer lugar indicar que he constatado que mi aplicación es una oportunidad de mercado ya que en la plataforma Android no existen muchas aplicaciones que ofrezcan funcionalidades de localización basadas en el control y seguimiento de personas. Como consecuencia de esta escasa existencia la mayoría de soluciones existentes analizadas pertenecen a la plataforma IOS de Apple.

Relación de aplicaciones analizadas y su origen:

- OPS Localización v2.2 (Google Play)
- GpsAsintec v1.0.44 (Google Play)
- Alert.Us – Localización v1.3.1 (Google Play)
- Find my kids – Footprints v1.6 (AppStore)
- GPS Tracker 365 v5.5 (AppStore)
- Team Locator v1.1.3 (AppStore)
- Buscar mi Iphone v4.0 (AppStore)
- Buscar a mis amigos v4.0.1 (AppStore)
- Pet Locator v1.1.5 (AppStore)
- MyGPSTracker v1.0.1 (AppStore)

A continuación destacaré de cada una de las aplicaciones analizadas aquellos aspectos de diseño o puntos de mejora que tendré en cuenta de cara a la fase de diseño y prototipado. El análisis realizado fue mucho más profundo de lo que se expondrá, pero se ha intentado sintetizar las conclusiones para hacer mucho más amigable el estudio, por lo que únicamente mostraré la solución de diseño que más me gustó de todos los productos analizados, por lo que es posible que no aparezcan referencias de todas las aplicaciones enumeradas.

Todas las imágenes utilizadas de las aplicaciones, se han tomado prestadas al amparo del criterio jurisprudencial llamado “Fair use”, el cual indica que se puede hacer un uso limitado de material protegido sin necesidad de tener permiso del autor bajo unas condiciones concretas. En nuestro caso particular la condición es la finalidad académica (Fuentes: http://es.wikipedia.org/wiki/Fair_use, <http://www.ajilc.cat/pdf/annals/Comunicaci%C3%B3%20Sr.%20Sancho%20Gargallo%2026-04-2011.pdf>)

a) Pantalla inicial

- Pet Locator: Tomaremos esta aplicación como referencia de pantalla inicial de autenticación, con el campo para el usuario y la contraseña. Botón de iniciar sesión, el link de registro, el link de recuerdo de contraseña y un link de ayuda:



Figura 3

b) Tipos de marcadores

- GPS Tracker 365: Esta solución será la referencia en el diseño del marcador de posicionamiento. Se fija la posición del sujeto mostrando en un bocadillo superior la fecha, la hora, la dirección y la velocidad del sujeto en ese momento:

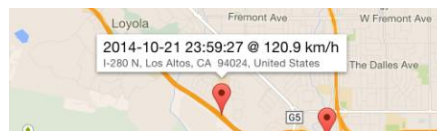


Figura 4

c) Formato de los mapas

- Buscar mi Iphone: Utilización de botones con los tipos de mapa normal, satélite o híbrido:



Figura 5

d) Control de batería remoto

- Alert.Us – Localización: Referencia en el diseño del indicador de la carga de la batería del dispositivo remoto localizado.

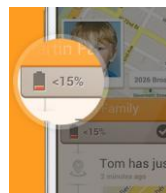


Figura 6

e) Modificación intervalo de actualización de la localización

- Find my kids – Footprints: Herramienta para modificar el intervalo de actualización de la posición del sujeto localizado

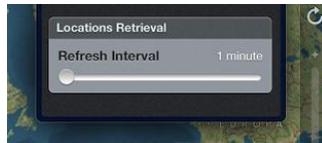


Figura 7

- f) Modificación de la distancia mínima recorrida para notificar
 - Find my kids – Footprints

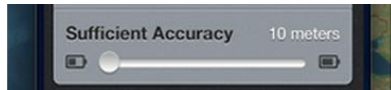


Figura 8

- g) Representación de la ruta
 - Buscar mi Iphone: Ejemplo de ruta en el que se marcan también las ubicaciones de los lugares donde se ha parado:

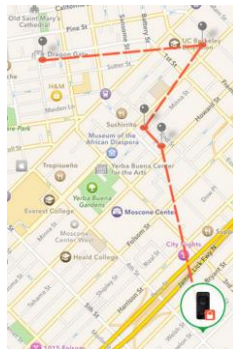


Figura 9

2.2. Perfiles de usuarios identificados

La finalidad principal de la aplicación es la de localizar a sujetos. Todo y que la misma aplicación correrá en dispositivos distintos, por un lado en el dispositivo del sujeto a localizar y por otro lado en el dispositivo del sujeto localizador, el usuario para utilizar la aplicación, como ya se ha comentado en la conclusión de la segunda dinámica de grupo, será único. Ya que la persona que lleve el dispositivo localizado tendrá corriendo la aplicación en modo “Localización”, pero con el entorno de gestión cerrado, y únicamente el gestor mediante autenticación podrá entrar a la aplicación para detener el posicionamiento del sujeto.

Aunque sí hemos de mencionar que la persona física que porte el dispositivo localizado, tendrá una pequeña interrelación con la aplicación cuando active la alerta de pánico mediante el botón diseñado para tal fin, visible en el servicio en primer plano que se podrá ver en el área de notificación.

2.2.1. Perfil y descripción

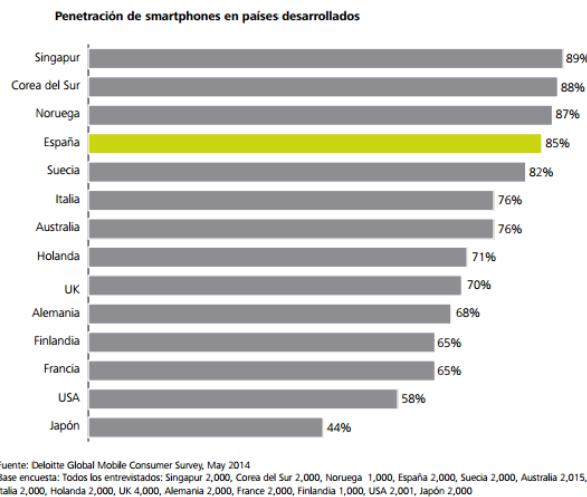
Al único tipo de perfil de usuario existente en la aplicación se le da la denominación de “Gestores”.

Dicho perfil tendrá libre acceso a las funcionalidades de la solución. La cual se dividirán en 2 áreas: El área de “Localización” y el área de “Gestión”. Los objetivos y tareas de cada una de las dos áreas se detallan en el apartado 2.2.4.

2.2.2. Factores demográficos

Las personas que podrían ejercer este rol de “Gestores” podrían tener diferentes niveles de experiencia en el uso de la tecnología móvil, en función de a cuál de los 3 tipos de targets para los que se ha desarrollado esta aplicación pertenecen.

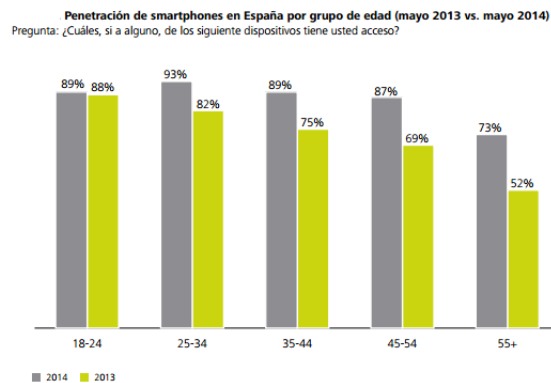
En el caso de padres, tutores o hijos de personas mayores el nivel de experiencia con dispositivos móviles podría oscilar entre bajo/medio o alto. Si atendemos al estudio de penetración de los Smartphones en los países desarrollados realizado en mayo de 2014 por Deloitte Global Mobile, en general el nivel de experiencia no será un problema para la mayoría de la población:



Fuente: Deloitte Global Mobile

Figura 10

Dentro del mismo estudio, podemos observar, que en la franja de edad entre 25 y los 54 años (franja de edad que hemos de analizar para nuestro proyecto, ya que sería aquella comprendida entre la edad que los adultos suelen ser padres hoy en día y el umbral de edad en la que los hijos están al cuidado de sus padres mayores), tampoco sería un problema el nivel de experiencia con los teléfonos inteligentes:



Fuente: Deloitte Global Mobile Consumer Survey, Spain edition, May 2013, May 2014
Base encuesta (2013/2014): Todos los entrevistados (2.000/2.000)

Fuente: Deloitte Global Mobile

Figura 11

Otro dato que podemos evidenciar es que los gestores, que utilicen la aplicación para controlar la integridad de sus personas mayores, será un target que irá en aumento, ya que como podemos observar en la tabla que muestro a continuación la tendencia, por lo menos en la Unión Europea, es que el porcentaje de personas mayores aumente considerablemente:

Tabla 1. Población de 65 y más años. Previsiones nacionales Unión Europea 2000-2020

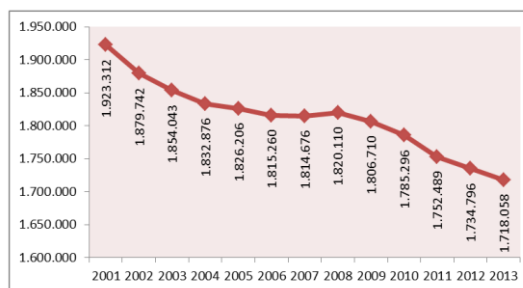
Países	2000		2020	
	Absoluto (*)	%	Absoluto (*)	%
Total UE-15	61.388,8	16,3	79.129,1	20,6
Alemania	13.654,1	16,7	17.387,5	21,7
Austria	1.254,9	15,5	1.660,4	20,0
Bélgica	1.709,8	16,7	2.051,3	19,8
Dinamarca	790,4	14,8	1.081,4	18,8
España	6.589,0	16,7	7.803,0	19,8
Finlandia	765,5	14,8	1.176,4	22,2
Francia	9.413,4	16,0	12.010,3	19,6
Grecia (f)	1.819,2	17,3	2.297,0	21,3
Irlanda	423,8	11,2	655,2	14,8
Italia	10.370,5	18,0	13.714,0	23,6
Luxemburgo	62,5	14,7	86,8	17,8
Países Bajos	2.152,4	13,6	3.219,7	18,4
Portugal (f)	1.535,2	15,3	1.897,8	18,0
Reino Unido	9.311,0	15,6	12.038,0	19,0
Suecia	1.536,9	17,3	2.050,4	22,2

Fuente: Eurostat: Statistiques sociales européennes. Luxemburgo, 2001; 174 P; Tablas 1-5, p. 130; Tablas 1-8, p. 132.

Figura 12

Por otro lado los adolescentes es un colectivo en el que el grado de conflictividad puede ser elevado. Si observamos la siguiente gráfica, veremos que todo y que el porcentaje ha bajado en los últimos años, el número de condenados juveniles en España en la franja de edad que va de los 14 a los 17 años es muy elevada:

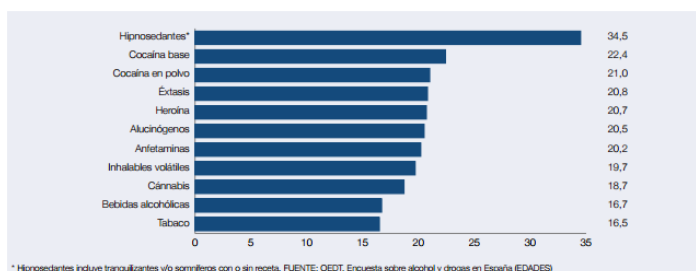
Evolución población española de 14 a 17 años



Fuente: Instituto Nacional de Estadística (INE)

Figura 13

Otro factor que puede condicionar la conflictividad de los adolescentes, y que por ende hace necesario ejercer un control sobre ellos, es el inicio del consumo de bebidas alcohólicas y el cannabis, que como se puede observar en la siguiente tabla se inicia entre los 16 y los 18 años:



Fuente: Plan nacional sobre drogas. Estadísticas 2013

Figura 14

2.2.3. Contextos de uso

Los contextos de uso (tanto habitual como desfavorables) en los que los usuarios del perfil “Gestores” utilizarán la aplicación, son los que se obtuvieron tras el análisis en la primera dinámica de grupo. Se pueden consultar dicha relación en las conclusiones 1 y 2 del apartado 2.1.2.1.

2.2.4. Relación de tareas agrupadas por áreas y objetivos

Para la relación de tareas necesarias para llevar a cabo cada uno de los objetivos se parte de la base que la aplicación está cerrada y se muestran los procesos completos. Algunas de las tareas especificadas se consideraran opcionales, y su desarrollo dependerá de si el tiempo destinado al desarrollo permite poder implementarlas con garantías.

1) Área de Gestión

- a) Consultar ubicación actual de todos los sujetos localizados
 - Autenticarse en el sistema
 - Acceder al área de Gestión
 - Pulsar sobre el botón localización global
 - En el mapa se muestra la ubicación del sujeto seleccionado

- b) Consultar ubicación actual usuario localizado
 - Autenticarse en el sistema
 - Acceder al área de Gestión
 - Seleccionar el sujeto que se quiere localizar
 - Pulsar sobre la opción “Localizar”
 - En el mapa se muestra la ubicación del sujeto seleccionado

- c) Consultar recorrido en un día concreto
 - Autenticarse en el sistema
 - Acceder al área de Gestión
 - Seleccionar el sujeto que se quiere consultar el trayecto
 - Seleccionar la fecha deseada
 - Pulsar sobre la opción “Ver recorrido”
 - En el mapa se muestra el recorrido del sujeto en la fecha seleccionada

2) Área de Localización

- a) Dar de alta sujeto a localizar
 - Autenticarse en el sistema
 - Acceder al área de Localización
 - Acceder al apartado “Alta y baja de objetivos”
 - Rellenar los datos pertinentes y pulsar en el botón “Confirmar alta”

- b) Dar de baja sujeto a localizar
 - Autenticarse en el sistema
 - Acceder al área de Localización
 - Acceder al apartado “Alta y baja de objetivos”
 - Seleccionar sujeto a eliminar del desplegable
 - Eliminar sujeto pulsando en el botón “Confirmar baja”

- c) Activar la localización de un sujeto
 - Autenticarse en el sistema
 - Acceder al área de Localización
 - Acceder al apartado “Activar localización”
 - Seleccionar en el desplegable el sujeto sobre el que se quiere activar la localización
 - Activar la localización con el botón “Activar localización”
- d) Desactivar la localización de un sujeto
 - Autenticarse en el sistema
 - Acceder al área de Localización
 - Acceder al apartado “Activar localización”
 - Desactivar la localización con el botón “Desactivar localización”
- e) Activar alerta de pánico
 - Acceder al área de notificación de Android
 - Pulsar sobre el botón “Panic” del servicio en primer plano de la localización

2.3. Resumen de necesidades detectadas en el análisis

Fruto de la etapa de análisis, podemos decir que ya tenemos claro los perfiles de usuario que nuestra aplicación necesitará (un único perfil en este caso), una visión global de las necesidades a nivel de diseño de las interfaces con las que tendrán que interactuar los usuarios de dicho perfil y un listado de contextos desfavorables que habrá que tener en cuenta.

Si bien es cierto, como ya hemos comentado anteriormente, que no todos los elementos interesantes detectados podrán ser implementados por falta de tiempo, sí que existirán unos mínimos que se tendrán que tener en cuenta, a parte de la implementación de los requisitos funcionales, y serán los siguientes:

2.3.1. Necesidades a nivel de implementación

- Se tendrá que tener un control de la conexión a internet, tanto para saber si la conexión a internet tiene un coste asociado y poder informar al usuario, como para detectar la ausencia de la conexión a internet y saber que se ha de llevar a cabo una gestión con las ubicaciones detectadas hasta que vuelva a haber internet
- Se implementará un control de la batería del dispositivo localizado
- Se implementará el evitar detener la localización salvo que el usuario gestor se haya autenticado en TeenControl
- Se implementará la posibilidad de modificar el intervalo de notificación de la posición
- Se implementará la posibilidad de modificar la distancia mínima recorrida para notificación la posición
- Se implementará la detección de la ubicación por redes móviles o Wifi en aquellos dispositivos que no tengan GPS (Así como la posibilidad de seleccionar cuál de las 2 posibilidades de localización se quiere activar)

2.3.2. Necesidades a nivel de diseño

- Los botones de navegación entre menús se fijarán abajo a la izquierda (decisión tomada en base a la lectura de las buenas prácticas para el diseño en dispositivos móviles), ya que es la zona de más fácil interacción cuando se coge el teléfono con la mano derecha. Ya que el porcentaje de zurdos a nivel mundial se comprende tan sólo entre el 8 y el 13% (Fuente: <http://es.wikipedia.org/wiki/Zurdera>)
- Pantalla inicial, que tendrá que contener un campo para el usuario y contraseña para poder autenticarse. Desde esta página también se podrá acceder al registro de un nuevo usuario, y desde esta página se podrá pedir el recordatorio de la contraseña
- De todos los marcadores analizados para fijar la posición del dispositivo localizado, se intentará implementar el de la aplicación GPS Tracker 365, ya que es el que más información aporta (hora, fecha, dirección, velocidad del dispositivo en ese momento, ...)
- En los mapas se incluirá la posibilidad de elegir el tipo de mapa normal, satélite o híbrido.
- En la pantalla "Localizar" se podrá ver el nivel de batería del dispositivo localizado mediante una imagen que representará el porcentaje aproximado de carga
- Se intentará llevar a cabo un diseño de interfaces que tenga en cuenta las pantallas pequeñas de algunos dispositivos, permitiendo el modo panorámico en las pantallas de visualización del mapa

3. ETAPA DE DISEÑO: Diseño conceptual

En esta etapa, gracias a la información obtenida en la etapa de análisis, se describirán los escenarios de uso de la aplicación y los flujos de interacción, los cuales nos servirán para ver la estructura general de la aplicación.

3.1. Personajes

Para describir los escenarios de uso, me ayudaré de la técnica de los personajes, que consiste en crear personajes a partir de la información obtenida en la fase de análisis. Dicha información es estudiada y analizada para modelar cada uno de los personajes.

El personaje que he modelado para representar a perfil de usuarios "Gestores" es el siguiente:



Laura Pérez Delgado

45 años | Secretaria

Secretaria de una entidad financiera. Tiene una hija de 16 años llamada Sara. Tiene 3 dioptrías en el ojo izquierdo y 2,5 dioptrías en el ojo derecho.

Figura 15

Tiene la necesidad de poder controlar a su hija que es muy rebelde y con las que ya ha tenido algún problema serio de conducta.

Tiene un móvil Motorola Moto G 2014. Hace un uso moderado del móvil como entretenimiento y mantenerse en contacto con amigos y familiares.

La fotografía utilizada está libre derechos de autor obtenida de <http://www.imagenessincopyright.com/>

Todo y que no existe un perfil de usuario para la persona que lleva el dispositivo localizado, será importante definir escenarios de uso en los que importa el comportamiento de dicha persona. Por este motivo he creado el siguiente personaje.



Sara Martínez Delgado

16 años | Estudiante

Estudiante de ESO. Realiza muchas actividades extraescolares, entre las que se encuentra el senderismo.

Figura 16

Ha tenido problemas de indisciplina y acostumbra a mentir a su madre que es con quien vive.


Tiene un móvil Samsung Galaxy mini con 3.5 pulgadas de pantalla. Hace un uso intensivo del teléfono utilizando varias aplicaciones vinculadas a las redes sociales.

La fotografía utilizada está libre derechos de autor obtenida de <http://pixabay.com/>

3.2. Escenarios de uso

Ejemplos de escenarios de uso en un contexto de uso habitual:

	<p>Laura, al levantarse por la mañana, coge el móvil de su hija Sara. Se autentica en TeenControl, accede al Área de Localización, selecciona a su hija Sara y activa la localización.</p> <p>Al llegar a casa después del trabajo, concretamente a las 14:45, antes de ponerse a hacer la comida quiere comprobar que su hija Sara no se ha saltado las clases. Coge su móvil, se autentica en TeenControl, accede al área de gestión, selecciona a su hija Sara, pulsa en la herramienta “Localizar”. Acto seguido observa en el mapa que se acaba de abrir que su hija está en el colegio.</p>
	<p>Son las 18:30. Sara está dando una vuelta con la bici por una ruta cercana a casa. Cae en un accidente muy aparatoso que hace que se de en la cabeza y empiece a marearse. Antes de perder la consciencia activa la alerta de pánico pulsando sobre el botón “Panic” del servicio en primer plano de TeenControl que se encuentra en el área de notificación Android.</p>
	<p>Laura tiene el día libre y está viendo un documental tranquilamente en casa. Como Sara hoy tenía una excursión en el colegio para ir a ver el museo de la Ciencia de Barcelona, le entre la curiosidad de ver el recorrido que ha hecho hasta el momento su hija. Se autentica en TeenControl, accede al Área de Gestión, selecciona a su hija Sara, pulsa en la herramienta “Ver recorrido” y selecciona la fecha y el recorrido se muestra en el mapa.</p>

	<p>Son las 10:15. Laura está trabajando. Recuerda que la noche anterior a Sara se le olvidó poner a cargar su móvil. Se autentica en TeenControl, accede al Área de Gestión, selecciona a su hija Sara y pulsa en la herramienta “Localizar”. Se abre el mapa y se ubica a Sara en el colegio. Aprovecha para comprobar, en el indicador que aparece en el mapa, que aún le queda la mitad de la batería.</p>
---	---

Ejemplos de escenarios de uso en un contexto de uso no habitual y desfavorable:

 	<p>Es sábado. Hoy es un día en el que Sara ha quedado con unos amigos para hacer senderismo. Sara lleva unos 30 minutos en una zona cerca de Monserrat en la que no hay cobertura de red móvil. En ese momento su madre le da por consultar en TeenControl la ruta. Al hacerlo la ruta finaliza media hora antes de la hora actual. 10 minutos después Sara ya ha entrado en una zona de cobertura y ha enviado las ubicaciones acumuladas que no había podido enviar. A los 15 minutos de la primera consulta Laura vuelve a consultar la ruta y ya aparece el recorrido completo.</p>
 	<p>Sara acaba de llegar a Casa. Laura lleva todo el día sin el móvil porque el día anterior dejó reparando la pantalla. Le pregunta a su hija donde ha estado la tarde. Como no se fía le pide su móvil, para consultarlo. Se autentica en TeenControl accede al Área de gestión, selecciona a Sara y se dispone a pulsar en la herramienta “Ver recorrido”. De entrada, como no lleva sus gafas de lectura, le cuesta diferenciar el texto de cada uno de los botones. Se pone sus gafas y pulsa en el botón “Ver recorrido” y selecciona la fecha. De primeras le sorprende que no ve la ruta tan claro como le ocurre en su móvil de mayor pantalla. Por intuición gira el móvil de Sara (cuya pantalla como comentamos era de 3.5 pulgadas) y el mapa se muestra en formato panorámico lo que facilita la visión de la ruta.</p>
 	<p>Laura está con su hija Sara en casa de sus amigos. Sara quiere irse con María hija de los amigos a tomar algo. Laura quiere activar la localización pero Sara se ha dejado el móvil olvidado en casa. Acuerdan con los amigos instalar la aplicación en el móvil de María. Laura instala TeenControl, accede al Área de Localización, selecciona a su hija y le da a “Activar localización”, le salta un mensaje conforme la localización está desactivada y que active de los 3 modos de ubicación existentes los que disponga el móvil y se deseen activar. El móvil no dispone de GPS por lo que se activa la ubicación por Wifi y redes móviles.</p>
	<p>Laura está visitando a unos clientes de la entidad bancaria. Cuando sale desea consultar la ubicación de Sara para ver si está en el colegio. Se autentica en TeenControl, accede al Área de Gestión, selecciona a su hija Sara y pulsa en la herramienta “Localizar”. TeenControl le avisa que esta consulta puede tener coste ya que no está conectado a una red Wifi.</p>

3.3. Flujos de interacción

En el siguiente diagrama se muestra el flujo de interacción de la aplicación. Flujo que nos sirve para hacernos una idea de la estructura de la aplicación.

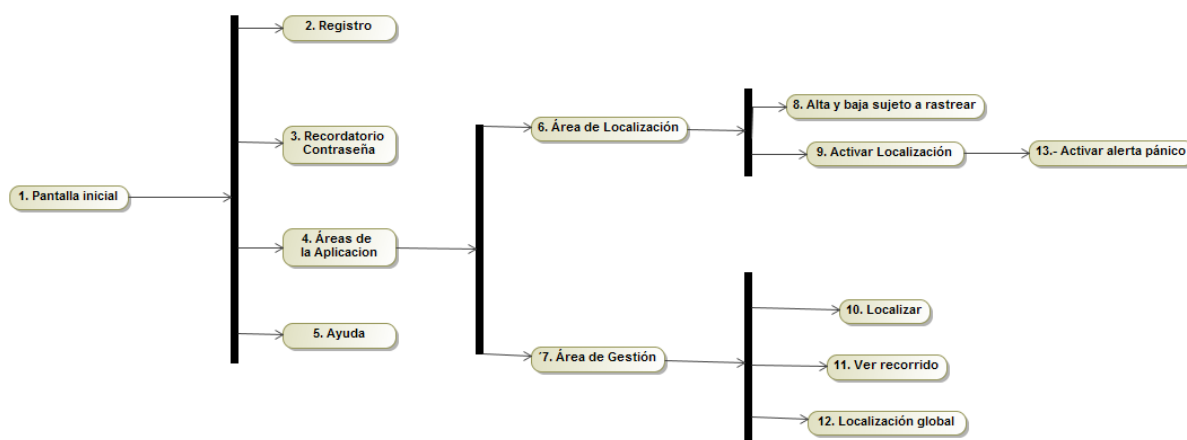


Figura 17

A continuación, con la numeración de cada uno de los nodos, se elabora una tabla en la que se puede observar una descripción de la información contenida en cada nodo, las opciones de acciones en cada nodo y la repercusión que tiene la elección de cada una de las acciones:

Nodo	Descripción interna	Acciones disponibles	Reacciones
1	Pantallas accesibles desde "Pantalla inicial"	a) Registro b) Recordatorio de contraseña c) Autenticación d) Ayuda pantalla inicial	a) Acceso al nodo 2 b) Acceso al nodo 3 c) Acceso al nodo 4 d) Acceso al nodo 5
2	Opciones a realizar	a) Registro b) Volver atrás	a) Registro nuevo Gestor b) Acceso al nodo 1
3	Opciones a realizar	a) Recordar contraseña b) Volver atrás	a) Envío mail con contraseña b) Acceso al nodo 1
4	Pantallas accesibles desde pantalla "Áreas de la Aplicación"	a) Área de Localización b) Área de Gestión c) Logout	a) Acceso al nodo 6 b) Acceso al nodo 7 c) Acceso al nodo 1
5	Información sobre las opciones de la pantalla inicial	Volver atrás	Acceso al nodo 1
6	Opciones a realizar	a) Alta y baja sujeto b) Activar localización c) Volver atrás	a) Acceso al nodo 8 b) Acceso al nodo 9 c) Acceso al nodo 4
7	Opciones a realizar	a) Selección usuario b) Localizar c) Ver Recorrido d) Localización global e) Volver atrás	a) Selección del sujeto con el que operar b) Acceso al nodo 10 c) Acceso al nodo 11 d) Acceso al nodo 12 e) Acceso al nodo 4

8	Opciones a realizar	a) Alta sujeto b) Selección sujeto c) Baja sujeto d) Volver atrás	a) Se da de alta un sujeto a rastrear b) Selección sujeto a eliminar c) Se da de baja a sujeto seleccionado d) Acceso al nodo 6
9	Opciones a realizar	a) Selección usuario b) Activar localización c) Desactivar localización c) Volver atrás	a) Se selecciona el usuario a rastrear b) Se activa el rastreo del usuario y se habilita el acceso al nodo 13 c) Se desactiva el rastreo del usuario d) Acceso al nodo 6
10	Se muestra la ubicación del usuario seleccionado en el mapa	a) Mapa normal b) Mapa satélite c) Mapa híbrido d) Volver atrás	a) Visualización mapa de carretera b) Visualización del mapa con imágenes de satélite c) Visualización mixta de mapa normal y satélite d) Acceso al nodo 7
11	Opciones a realizar	a) Selección fecha b) Mapa normal c) Mapa satélite d) Mapa híbrido e) Volver atrás	a) Se muestra el recorrido del sujeto en el mapa en la fecha seleccionada b) Visualización mapa de carretera c) Visualización del mapa con imágenes de satélite d) Visualización mixta de mapa normal y satélite e) Acceso al nodo 7
12	Se muestra la ubicación de todos los sujetos del gestor en el mapa	a) Mapa normal b) Mapa satélite c) Mapa híbrido d) Volver atrás	a) Visualización mapa de carretera b) Visualización del mapa con imágenes de satélite c) Visualización mixta de mapa normal y satélite d) Acceso al nodo 7
13	Opciones a realizar	a) Activar alerta de pánico	a) Se envía un correo informando al gestor de la posición del sujeto con la fecha y hora de la alerta

4. PROTOTIPADO: Sketches y prototipos horizontales de alta fidelidad

A partir de los flujos de interacción, de la etapa de diseño conceptual, se elaboran una serie de prototipos que nos serán útiles de cara a validar dicho diseño en la etapa evaluación. Previo al diseño de los prototipos definitivos, se realizan unos esbozos llamados sketches.

Los sketches son esbozos a mano alzada que servirán como guía para el diseño de los prototipos horizontales de alta fidelidad y que también ayudarán a detectar errores en el proceso de realización de los mismos.

Los prototipos horizontales de alta fidelidad son aquellos diseños definitivos realizados a partir de los borradores (sketches), los cuales incorporan el máximo nivel de detalle de la mayor parte de las funcionalidades de la aplicación.

Para implementar los prototipos horizontales se ha utilizado la herramienta Justinmind Prototyper v6.4.1

He organizado cada una de las pantallas de la aplicación en 3 grupos en función del ámbito funcional al que corresponden o con el que interactúan: Autenticación, Área de Localización y Área de Gestión.

En cada una de estos 3 grupos observaremos la evolución desde los sketches hacia los prototipos definitivos.

Para poder observar el flujo natural en el diseño podréis observar que existen diferencias entre los sketches y los prototipos horizontales de alta finalidad definitivos. Precisamente he mantenido las diferencias para que quede plasmada la detección de errores durante el proceso. En el momento que exista alguna diferencia, indicaré la relación de las mismas antes de que se muestren los prototipos horizontales e indicaré los motivos que las han producido. Además en los diseños de los prototipos horizontales también añadiré los diálogos de información o confirmación principales.

* Nota aclaratoria: En fase de implementación se introdujeron cambios respecto al prototipado los cuales serán detallados en el apartado de implementación.

4.1. Autenticación

La pantalla representativa de este grupo es la pantalla inicial, desde la que accederemos a realizar el registro, realizar la autenticación, recordar la contraseña o consultar la ayuda vinculada a estas acciones.

Una vez nos autenticamos accederemos al menú principal de TeenControl, desde el que accederemos a los grupos de pantallas que darán nombre a los siguientes dos apartados.

4.1.1. Autenticación: Sketches

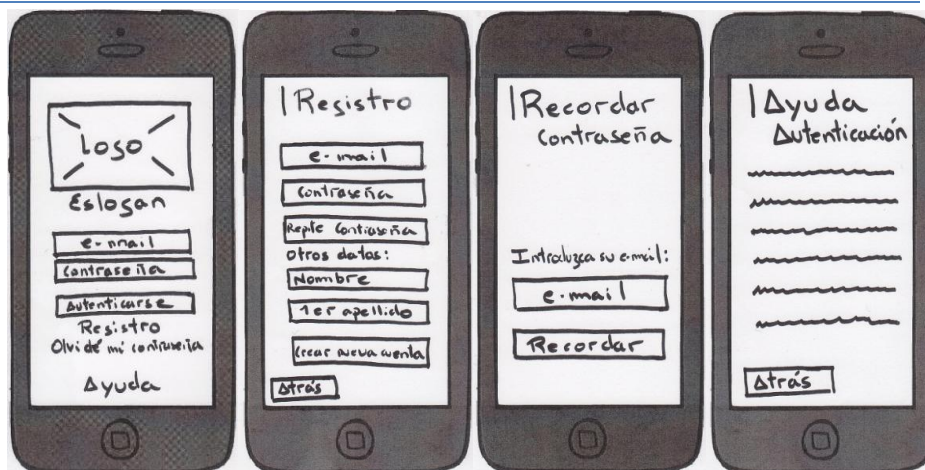


Figura 18

Figura 19

Figura 20

Figura 21



Figura 22

4.1.2. Autenticación: Prototipos horizontales

En todos los prototipos horizontales cuando el botón navegacional "Atrás" accedía a la pantalla inicial, al menú principal o al menú de las 2 áreas de la aplicación, se ha substituido el

descriptivo “Atrás” por un descriptivo que al usuario le permita conocer en cada momento hacia que pantalla se dirigirá cuando pulse dicho botón. Únicamente se ha mantenido el botón atrás cuando el botón accede a la pantalla anterior en la que no se cambia la funcionalidad o la gestión de configuración en la que nos encontrábamos.

En el caso de la pantalla para recordar la contraseña, se ha subido la ubicación de todos los elementos, para evitar que al salir el teclado se tuvieran que desplazar los mismos.

a) Pantalla inicial



Figura 23

b) Registro



Figura 24



Figura 25

c) Recordar contraseña



Figura 26



Figura 27

d) Ayuda autenticación



Figura 28

e) Menú principal



Figura 29

4.2. Área de Localización

En esta área se ubican todas las pantallas relacionadas con las acciones necesarias para poder disponer de las localizaciones de los sujetos rastreados: Alta y baja de sujetos a rastrear y activación/desactivación de la localización de cada uno de los sujetos que cada gestor haya dado de alta.

4.2.1. Área de Localización: Sketches

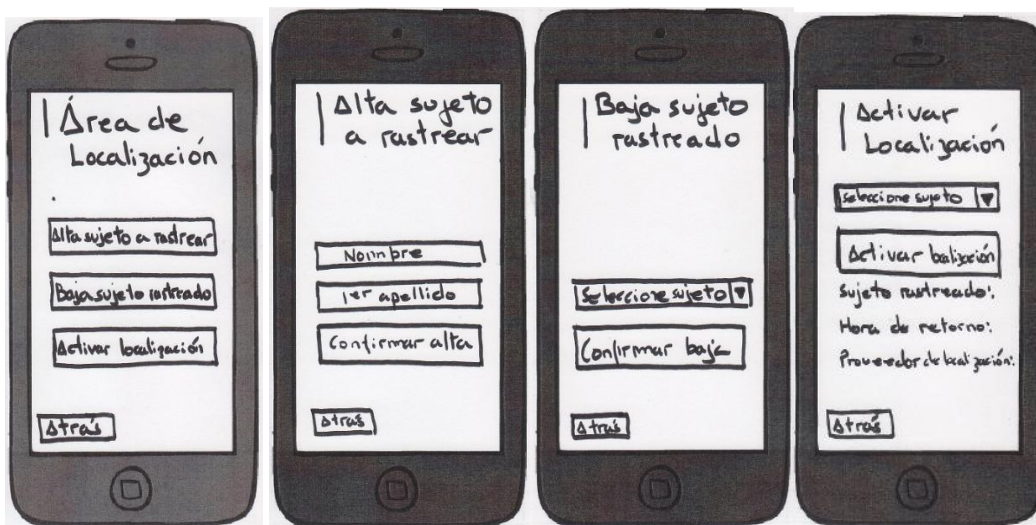


Figura 30

Figura 31

Figura 32

Figura 33

4.2.2. Área de Localización: Prototipos horizontales

Al interactuar con la posibilidad de simulación que ofrece Justinmind Prototyper, observé que no tenía mucho sentido separar las opciones de alta y baja de sujeto en 2 pantallas. De manera que quedaron agrupadas reduciendo la complejidad de la estructura y facilitando la usabilidad en caso de tener que utilizar la función de alta y baja de sujetos de manera consecutiva (Por ejemplo al producirse un error al realizar el alta)

a) Área de localización



Figura 34

b) Alta y baja de sujetos a rastrear

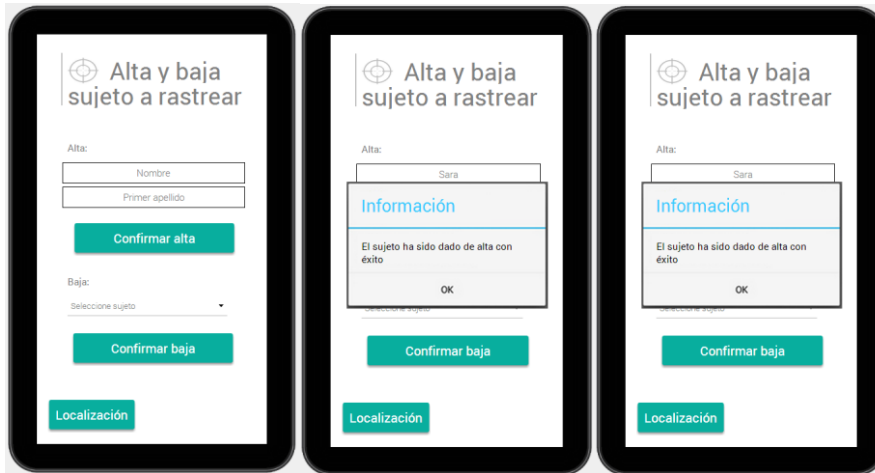


Figura 35

Figura 36

Figura 37

c) Activar localización

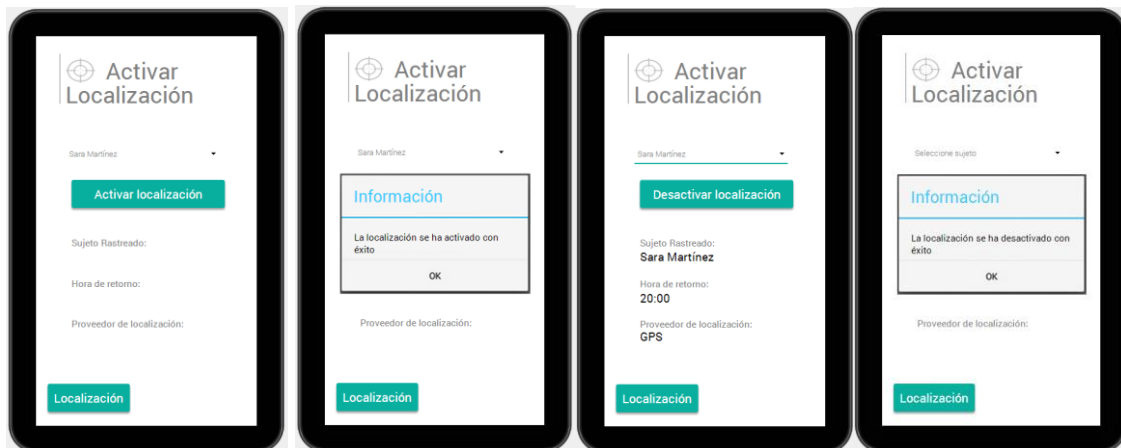


Figura 38

Figura 39

Figura 40

Figura 41

4.3. Área de Gestión

En esta área se ubican todas las pantallas relacionadas con las acciones que pueden llevar a cabo los gestores relacionadas con el control de los sujetos rastreados: Localizar al sujeto, ver el recorrido de una fecha concreta, fijar lugares prohibidos, definir un perímetro de seguridad y fijar una hora de retorno. Eso sí, antes se ha de seleccionar el sujeto con el que se ha de trabajar.

4.3.1. Área de Gestión: Sketches

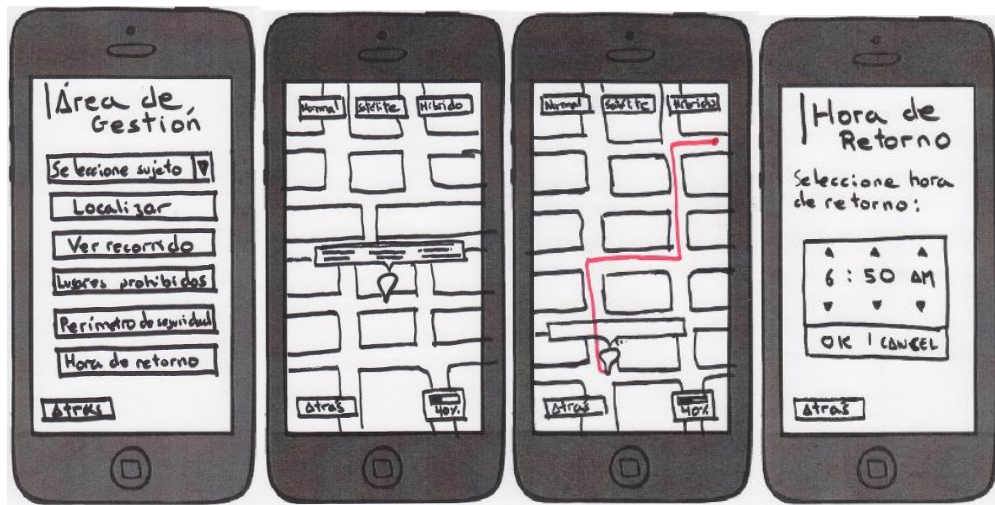


Figura 42

Figura 43

Figura 44

Figura 45

4.3.2. Área de Gestión: Prototipos horizontales

En las pantallas de configuración de parámetros (Lugares prohibidos, perímetro de seguridad y hora de retorno), he añadido un identificador del sujeto seleccionado actualmente y un desplegable para cambiar el sujeto seleccionado. De esta manera se mejora la usabilidad, ya que se evita al gestor tener que volver a la pantalla del área de gestión para cambiar el sujeto.

Al interactuar con la posibilidad de simulación que ofrece Justinmind Prototyper, observé que no tenía mucho sentido separar las opciones de alta y baja de lugares prohibidos en 2 pantallas. De manera que quedaron agrupadas reduciendo la complejidad de la estructura y facilitando la usabilidad en caso de tener que utilizar la función de alta y baja de lugares de manera consecutiva (Por ejemplo al producirse un error al realizar el alta)

La pantalla de hora de retorno se ha modificado substancialmente respecto al borrador, con la intención de mejorar la información mostrada y adaptar el diseño al funcionamiento de los objetos Android. En primer lugar he añadido un campo para que se muestre la hora fijada del sujeto seleccionado. Por otro lado el componente para fijar la hora se muestra a partir de pulsar un botón.

a) Área de gestión



Figura 46

Figura 47

b) Localizar al sujeto



Figura 48

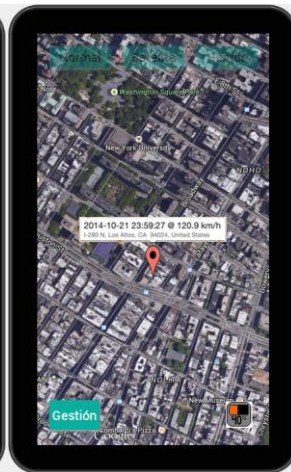


Figura 49

c) Ver el recorrido de una fecha concreta



Figura 50

d) Hora de retorno

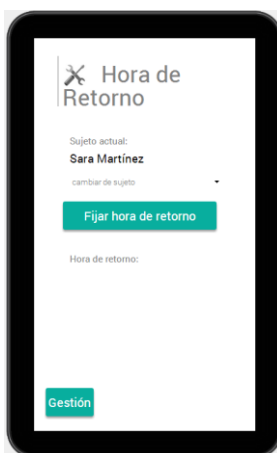


Figura 51

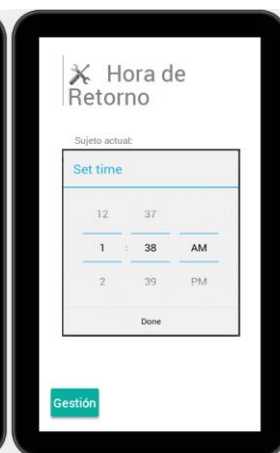


Figura 52



Figura 53

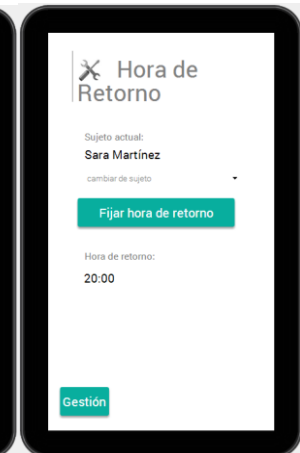


Figura 54

5. ETAPA DE EVALUACIÓN DE PROTOTIPOS

La finalidad de esta etapa es la validación de los prototipos por parte de terceras personas mediante la ejecución de unas tareas concretas y la posterior realización de unos cuestionarios en base a dichas tareas. Previamente a la ejecución de las tareas se realiza una criba de candidatos mediante un cuestionario para asegurarnos que cumplen con el perfil deseado.

5.1. Entorno para la evaluación

Para llevar a cabo la ejecución de las tareas se contará con la posibilidad de simulación que ofrece la herramienta Justinmind Prototyper. Esta permite simular los comportamientos de los botones y desplegables, mostrar u ocultar mensajes de notificación, definir acciones al pulsar un elemento de la pantalla y un largo etcétera. De manera que se puede conseguir una interacción y un flujo entre pantallas bastante aproximado al que será el resultado final.

La simulación se llevará a cabo en un terminal Android. Ya que otra de las posibilidades que ofrece Justinmind Prototyper es vincular los proyectos a una cuenta de correo y contraseña, y una vez descargada la app Justinmind desde Google Play, previa autenticación, el proyecto se descarga y se nos muestra la simulación en el dispositivo. A continuación os muestro unas capturas de ejemplo del prototipo ejecutándose en un terminal Android:

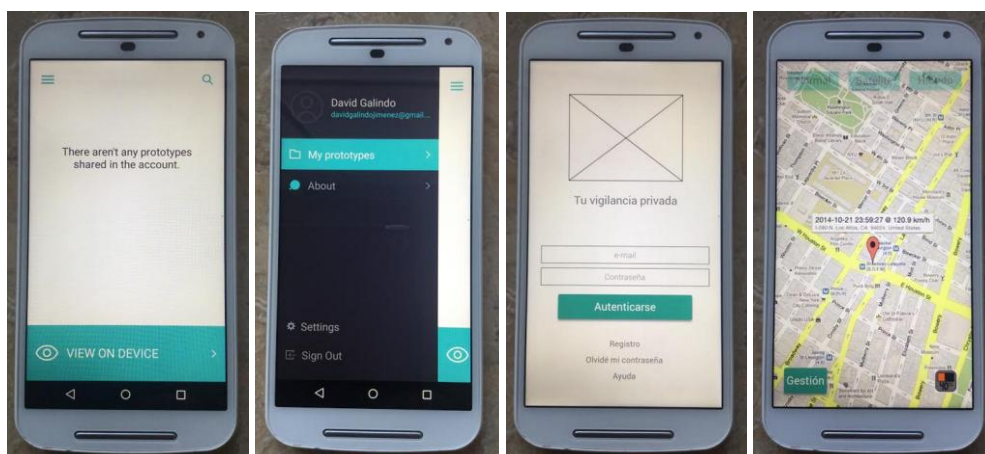


Figura 55

Figura 56

Figura 57

Figura 58

En esta etapa precisamente lo que se pretende es un punto de vista profano, en cuanto a lo que a tecnología se refiere, precisamente para detectar errores de implementación o de diseño que una visión tecnológica pueden pasar por alto.

5.2. Cuestionario a candidatos para ser testadores

El siguiente cuestionario está enfocado a encontrar candidatos a testadores que cumplan con el perfil deseado. El contexto del perfil buscado queda definido en los argumentos que explican la finalidad de cada una de las preguntas.

Las preguntas 1 y 2 están enfocadas a descartar a todas aquellas personas que tengan un punto de vista tecnológico, ya que precisamente lo que se busca son testadores que no tengan conocimientos relacionados con la implementación de la aplicación, ya sea a nivel de programación o a nivel de diseño de interfaces:

- **Pregunta 1:** ¿Trabajas o tienes conocimientos sobre el desarrollo de software?
- **Pregunta 2:** ¿Trabajas o tienes conocimientos sobre diseño gráfico?

Las preguntas 3, 4, y 5 pretenden seleccionar a personas con unos mínimos conocimientos sobre teléfonos inteligentes. Aunque en el apartado “2.2.2 Factores demográficos” ya vimos que el nivel de experiencia raramente será un problema ya que el porcentaje de penetración de los Smartphones en España en la franja de edad en la que buscaremos candidatos a testadores es muy alto.

- **Pregunta 3:** ¿Tienes Smartphone?
- **Pregunta 4:** ¿Cuánto tiempo al día sueles usar el Smartphone?
- **Pregunta 5:** ¿Sueles descargar bastantes aplicaciones del market en tu tiempo de ocio?

La pregunta 6 busca descartar personas que hayan tenido contacto con aplicaciones similares para evitar que estén condicionados por sus experiencias anteriores. Aunque estas personas también podrían ser útiles para llevar a cabo una validación paralela para detectar errores de implementación que ellos ya han visto cómo se han solucionado en otras aplicaciones o dar su opinión sobre mejores soluciones de diseño que hayan experimentado.

- **Pregunta 6:** ¿Has utilizado aplicaciones para el control de personas?

Las preguntas 7, 8 y 9 tienen como objetivo seleccionar a aquellas personas que pueden ser un target para la aplicación, es decir que tienen hijos en la adolescencia o personas mayores a su cargo, ya que esto implicará una mayor implicación por su parte, ya que se pueden ver reflejados en el uso del producto que están validando:

- **Pregunta 7:** ¿Tienes hijos en la adolescencia?
- **Pregunta 8:** Si la respuesta a la pregunta 5 fue “Sí” ¿Ha tenido problemas de indisciplina o de problemas de conducta con ellos?
- **Pregunta 9:** ¿En tu familia existen personas mayores que empiezan a tener problemas para poder valerse por ellos mismos?

5.3. Tareas de validación

A la hora de llevar a cabo las tareas, por parte de los testadores, la única información que se les da es una descripción general de cuales son los objetivos de la aplicación y cuales son las tareas de gestión que permite la aplicación para poder cumplir con sus objetivos. Precisamente la idea es no dar información de contextualización, en cada una de las tareas a ejecutar, para no condicionarlos y ver si realmente queda claro lo que se les está pidiendo y si son capaces de ejecutar las tareas con éxito y con una fluidez aceptable. Todo el proceso de ejecución de las tareas será filmado para poderlo analizar minuciosamente y detectar comportamientos anómalos por parte de los testadores consecuencia de una mala implementación o diseño deficiente:

- **Tarea 1:** Registrarse en la aplicación
- **Tarea 2:** Acceder al recordatorio de la contraseña
- **Tarea 3:** Autenticarse en la aplicación
- **Tarea 4:** Dar de alta sujeto para la localización

- **Tarea 5:** Activar la localización del sujeto dado de alta en la tarea 4.
- **Tarea 6:** Localizar al sujeto dado de alta en la tarea 4.
- **Tarea 7:** Ver el recorrido del sujeto dado de alta en la tarea 4.
- **Tarea 8:** Mandar una alerta de pánico por el sujeto dado de alta en la tarea 4.
- **Tarea 9:** Desactivar la localización del sujeto dado de alta en la tarea 4.
- **Tarea 10:** Dar de baja al sujeto dado de alta en la tarea 4.
- **Tarea 11:** Hacer logout de la aplicación.

5.4. Cuestionario relacionado con las tareas

A parte de analizar en los videos la manera cómo interactúan los usuarios con la aplicación nos interesa conocer su punto de vista sobre ciertos aspectos que se evalúan con las siguientes cuestiones:

- **Pregunta 1:** ¿Has echado en falta alguna funcionalidad interesante? Si la respuesta es afirmativa ¿Cuál?
- **Pregunta 2:** ¿Se podría decir que has asimilado de una manera rápida la estructura de la aplicación para saber hacia que pantalla te has de dirigir para llevar a cabo cualquier acción?
- **Pregunta 3:** ¿Crees que faltarían más pantallas de ayuda en alguna de las áreas de la aplicación? Si la respuesta es afirmativa ¿En qué área, funcionalidad o actividad de gestión concreta?
- **Pregunta 4:** ¿Te parece acertada la estructuración en dos áreas?
- **Pregunta 5:** ¿Te ha parecido más complejo de lo esperado la ejecución de alguna funcionalidad o actividad de gestión? Si la respuesta es afirmativa ¿Qué funcionalidad o actividad de gestión?
- **Pregunta 6:** ¿Hay algún aspecto a nivel de diseño que mejorarías? Si la respuesta es afirmativa ¿Qué aspecto?
- **Pregunta 7:** A nivel de usabilidad ¿Hay algún tipo de interacción para llevar a cabo alguna funcionalidad o actividad de gestión que cambiarías? Si la respuesta es afirmativa ¿Qué interacción y de que otra manera lo implementarías?
- **Pregunta 8:** A nivel que general, y teniendo en cuenta que se está evaluando un prototipo del 1 al 10 ¿Qué grado de madurez creed que tiene la aplicación?

6. ETAPA DE DISEÑO: Diseño Técnico

6.1. Definición de los casos de uso

Para poder especificar los procesos de comunicación y el comportamiento del sistema a través de la interacción con los usuarios utilizaremos el recurso denominado casos de uso. De manera que obtendremos 10 casos de uso a nivel de usuario que definen el total de requisitos funcionales de TeenControl.

Identificador	TC01
Nombre	Registrarse
Actor principal	Gestor
Ámbito	Organización (TeenControl)
Nivel de objetivo	Usuario
Escenario principal de éxito	1.- El usuario rellena todos los campos necesarios para el registro. 2.- El usuario pulsa el botón "Registrarse", se realiza el registro correctamente y el nuevo gestor accede directamente al mapa principal de TeenControl
Escenarios alternativos	2a.- TeenControl informa que el e-mail facilitado ya se encuentra dado de alta 2a1a.- El usuario rellena todos los campos de nuevo, introduciendo esta vez un email no dado de alta y pasa al punto 2 2a1b.- El usuario no desea darse de alta con otra cuenta de correo y finaliza el caso de uso. 2b.- TeenControl informa que no se han rellenado todos los datos 2b1a.- El usuario rellena los campos que faltan y se pasa al punto 2 2b1b.- El usuario no desea rellenar todos los datos y finaliza el caso de uso 2c.- TeenControl informa que las contraseñas no coinciden 2c1a.- El usuario iguala los campos de la contraseña y se pasa al punto 2 2c1b.- El usuario no desea rellenar todos los datos y finaliza el caso de uso 2d.- TeenControl informa que la contraseña no tiene un mínimo de 8 caracteres 2d1a.- El usuario alarga la contraseña hasta los 8 caracteres y se pasa al punto 2 2d1b.- El usuario no desea rellenar todos los datos y finaliza el caso de uso 2e.- TeenControl informa que no hay conexión a internet o el Servidor no está disponible 2e1a.- El usuario espera a que el Servidor esté disponible o soluciona su problema de acceso a internet y se pasa de nuevo al punto 2 2e1b.- El usuario no desea esperar o mirar cual es el problema y finaliza el caso de uso

Identificador	TC02
Nombre	Recordar contraseña
Actor principal	Gestor
Ámbito	Organización (TeenControl)
Nivel de objetivo	Usuario
Escenario principal de éxito	1.- El gestor introduce su usuario (que es un e-mail) en el campo correspondiente, 2.- El gestor pulsa sobre la etiqueta "Recordar contraseña" y se le envía un correo electrónica que contiene su contraseña.
Escenarios alternativos	2a.- El gestor no introduce nada en el campo "e-mail" 2a1a.- El usuario introduce un correo electrónico y se pasa al punto 2 2a1b.- El usuario no realiza ninguna acción y finaliza el caso de uso. 2b.- El e-mail facilitado no se encuentra dado de alta 2b1a.- El usuario introduce su e-mail correcto y se pasa al punto 2 2b1b.- El usuario no realiza ninguna acción y finaliza el caso de uso 2c.- TeenControl informa que no hay conexión a internet o el Servidor no está disponible 2c1a.- El usuario espera a que el Servidor esté disponible o soluciona su problema de acceso a internet y pasa de nuevo al punto 2 del escenario principal 2c1b.- El usuario no desea esperar o mirar cual es el problema y finaliza el caso de uso

Identificador	TC03
Nombre	Alta sujeto
Actor principal	Gestor
Ámbito	Organización (TeenControl)
Nivel de objetivo	Usuario
Precondiciones	El gestor se ha de haber registrado e identificado en el sistema y accedido a la pantalla de alta y baja de sujetos
Escenario principal de éxito	1.- El gestor rellena los campos del nuevo sujeto 2.- El gestor pulsa en el botón "Alta" y se da de alta correctamente al sujeto
Escenarios alternativos	2a.- TeenControl informa que no se han rellenado todos los datos 2a1a.- El gestor rellena los campos que faltan y pasa al punto 2 2a1b.- El gestor no desea rellenar todos los datos y finaliza el caso de uso 2b.- TeenControl informa que no hay conexión a internet o el Servidor no está disponible 2b1a.- El gestor espera a que el Servidor esté disponible o soluciona su problema de acceso a internet y pasa de nuevo al punto 2 2b1b.- El gestor no desea esperar o mirar cual es el problema y finaliza el caso de uso

Identificador	TC04
Nombre	Baja sujeto
Actor principal	Gestor
Ambito	Organización (TeenControl)
Nivel de objetivo	Usuario
Precondiciones	El gestor se ha de haber registrado e identificado en el sistema y accedido a la pantalla de alta y baja de sujetos
Escenario principal de éxito	1.- El gestor selecciona el sujeto a dar de baja 2.- El gestor pulsa en el botón "Baja" y se da de baja satisfactoriamente al sujeto
Escenarios alternativos	2a.- TeenControl informa que no se ha seleccionado ningún sujeto 2a1a.- El gestor selecciona un sujeto y se pasa al punto 2 2b.- TeenControl informa que no hay conexión a internet o el Servidor no está disponible 2b1a.- El gestor espera a que el Servidor esté disponible o soluciona su problema de acceso a internet y se pasa de nuevo al punto 2 2b1b.- El gestor no desea esperar o mirar cual es el problema y finaliza el caso de uso

Identificador	TC05
Nombre	Activar localización
Actor principal	Gestor
Ambito	Organización (TeenControl)
Nivel de objetivo	Usuario
Precondiciones	El gestor se ha de haber registrado e identificado en el sistema y haber accedido a la pantalla "Activar localización"
Escenario principal de éxito	1.- El gestor selecciona un sujeto 2.- Selecciona un intervalo de notificación o deja el por defecto 3.- Selecciona la distancia mínima recorrida para notificar el posicionamiento o dejan el por defecto 4.- Seleccionar el proveedor de localización preferido o dejan el por defecto 5.- El gestor desliza el interruptor de localización a la posición de "Si" y se activa la localización * Comentar que si existe un proveedor de localización activado que no es el deseado, se activa con este la localización
Escenarios alternativos	5a.- No se tiene ningún proveedor de localización activado 5a1a.- El gestor pulsa en el botón "Abrir" para ver la configuración de ubicación y activa un proveedor de localización. Se vuelve al paso 5 5a1b.- El gestor no realiza ninguna acción y finaliza el caso de uso 5b.- No se ha seleccionado ningún sujeto 5b1a.- El gestor selecciona un usuario y vuelve al paso 5 5b1b.- No existen sujetos dados de alta. El gestor pulsa en la opción del menú "Alta y baja sujetos" y da de <u>alta un sujeto</u> . Se vuelve a la pantalla inicial del caso de uso. Se selecciona el sujeto y se vuelve al paso 5. 5b1c.- El gestor no realiza ninguna acción y finaliza el caso de uso
Inclusiones	TC03(Alta sujeto)

Identificador	TC06
Nombre	Detener localización
Actor principal	Gestor
Ambito	Organización (TeenControl)
Nivel de objetivo	Usuario
Precondiciones	El gestor se ha de haber registrado e identificado en el sistema y haber accedido a la pantalla "Activar localización"
Escenario principal de éxito	1.- El gestor desliza el interruptor de localización a la posición de "No" y desactiva la localización

Identificador	TC07
Nombre	Alerta de pánico
Actor principal	Sujeto localizado (Usuario anónimo desde el punto de vista de TeenControl)
Ámbito	Organización (TeenControl)
Nivel de objetivo	Usuario
Escenario principal de éxito	1.- El usuario pulsa en el botón "PANIC" de la notificación del servicio de localización y la alerta se envía por correo electrónico al gestor
Escenarios alternativos	1a.- Si no existe conexión a internet la alerta de pánico no se envía

identificador	TC08
Nombre	Localización global
Actor principal	Gestor
Ámbito	Organización (TeenControl)
Nivel de objetivo	Usuario
Precondiciones	El gestor se ha de haber registrado e identificado en el sistema
Escenario principal de éxito	1.- El gestor accede al menú de acciones de TeenControl y pulsa sobre la acción "Localización global" y todos los sujetos localizados del gestor se sitúan en el mapa *Indicar que al autenticarse TeenControl realiza una localización global automáticamente
Escenarios alternativos	1a.- El gestor no tiene sujetos dados de alta 1a1a.- El gestor pulsa en la opción del menú "Alta y baja sujetos" y da de <u>alta un sujeto</u> y regresa al paso 1 1a1b.- El gestor no realiza ninguna acción y finaliza el caso de uso 1b.- No se dispone de localizaciones de los sujetos 1b1a.- El gestor en el dispositivo de uno o varios de sus sujetos se autentica, pulsa en la opción del menú "Activar localización" y <u>activa la localización</u> de dicho/s sujeto/s. Se vuelve al paso 1 1b1b.- El gestor no realiza ninguna acción y finaliza el caso de uso 1c.- TeenControl informa que no hay conexión a internet o el Servidor no disponible 1c1a.- El usuario espera a que el Servidor esté disponible o soluciona su problema de acceso a internet y pasa de nuevo al punto 1 del escenario principal 1c1b.- El usuario no desea esperar o mirar cual es el problema y finaliza el caso de uso
Inclusiones	TC03(Alta sujeto), TC05 (Activar localización)

Identificador	TC09
Nombre	Localizar sujeto
Actor principal	Gestor
Ámbito	Organización (TeenControl)
Nivel de objetivo	Usuario
Precondiciones	El gestor se ha de haber registrado e identificado en el sistema
Escenario principal de éxito	1.- El gestor selecciona un sujeto 2.- El gestor accede al menú de acciones de TeenControl, pulsa sobre el icono de "Localizar sujeto" y se produce la localización del sujeto en el mapa
Escenarios alternativos	2a.- No se ha seleccionado ningún sujeto 2a1a.- El gestor selecciona un usuario y vuelve al paso 2 2a1b.- No existen sujetos dados de alta. El gestor pulsa en la opción del menú "Alta y baja sujetos" y da de <u>alta un sujeto</u> . Se vuelve a la pantalla inicial del caso de uso. Se selecciona el sujeto y se vuelve al paso 2 2a1c.- El gestor no realiza ninguna acción y finaliza el caso de uso 2b.- No se dispone de localizaciones de los sujetos 2b1a.- El gestor en el dispositivo de uno o varios de sus sujetos se autentica, pulsa en la opción del menú "Activar localización" y <u>activa la localización</u> de dicho/s sujeto/s. Se envía un posicionamiento y se vuelve al paso 2 2b1b.- El gestor no realiza ninguna acción y finaliza el caso de uso 2c.- TeenControl informa que no hay conexión a internet o el Servidor no disponible 2c1a.- El usuario espera a que el Servidor esté disponible o soluciona su problema de acceso a internet y pasa de nuevo al punto 2 del escenario principal 2c1b.- El usuario no desea esperar o mirar cual es el problema y finaliza el caso de uso
Inclusiones	TC03(Alta sujeto), TC05 (Activar localización)

Identificador	TC10
Nombre	Ruta de un sujeto
Actor principal	Gestor
Ámbito	Organización (TeenControl)
Nivel de objetivo	Usuario
Precondiciones	El gestor se ha de haber registrado e identificado en el sistema
Escenario principal de éxito	<ol style="list-style-type: none"> 1.- El gestor selecciona un sujeto 2.- El gestor accede al menú de acciones de TeenControl y pulsa sobre el icono "Ruta" 3.- El gestor selecciona la fecha sobre la que quiere visualizar la ruta del sujeto y la ruta histórica realizada por el sujeto se muestra
Escenarios alternativos	<ol style="list-style-type: none"> 1a.- El gestor no tiene sujetos dados de alta y finaliza el caso de uso 1b.- No se dispone de localizaciones del sujeto y finaliza el caso de uso 1c.- TeenControl informa que no hay conexión a internet o el Servidor no disponible <ol style="list-style-type: none"> 1c1a.- El usuario espera a que el Servidor esté disponible o soluciona su problema de acceso a internet y pasa de nuevo al punto 1 del escenario principal 1c1b.- El usuario no desea esperar o mirar cual es el problema y finaliza el caso de uso

6.2. Diseño de la arquitectura

6.2.1. Diseño de la BBDD

A continuación se muestra el diagrama de la BBDD relacional. En este podremos ver las tablas que servirán para almacenar los registros pertenecientes a los gestores, los sujetos a localizar y las localizaciones de los mismos.

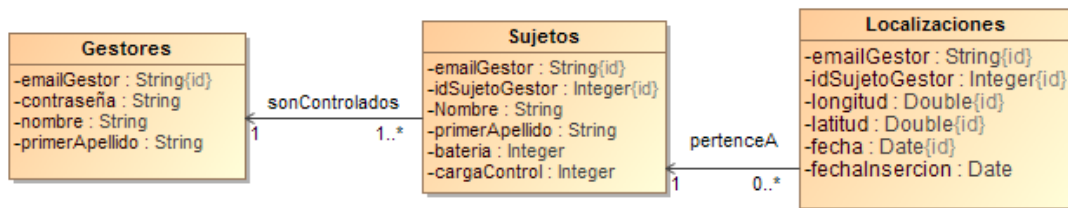


Figura 59

A parte de las claves primarias que hemos visto definidas en el esquema anterior, también se han definido las siguientes claves foráneas para garantizar la integridad de los datos:

```
ALTER TABLE `sujetos` DROP FOREIGN KEY `emailGestor_FK`; ALTER TABLE `sujetos` ADD CONSTRAINT `emailGestor_Fk` FOREIGN KEY (`emailGestor`) REFERENCES `projectc_teencontrol`.`gestores` (`emailGestor`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE `localizaciones` DROP FOREIGN KEY `sujetos_FK`; ALTER TABLE `localizaciones` ADD CONSTRAINT `sujetos_Fk` FOREIGN KEY (`emailGestor`, `idSujetoGestor`) REFERENCES `projectc_teencontrol`.`sujetos` (`emailGestor`, `idSujetoGestor`) ON DELETE CASCADE ON UPDATE CASCADE;
```

6.2.2. Diseño de entidades y clases

A continuación mostramos el diagrama UML correspondiente a las clases java que componen la aplicación TeenControl y las dependencias de una sobre otras en cuanto al uso de operaciones. Podemos ver las operaciones, tanto públicas como privadas y las clases en las que se realizan llamadas a tareas asíncronas.

He ocultado los atributos de las clases ya que no son relevantes (su finalidad es la de ser recursos de las operaciones para conseguir su finalidad)

De todas maneras si se desea consultar algún aspecto específico del diseño de cualquiera de los atributos de las clases o de sus operaciones, he generado una documentación global que se puede consultar en: <http://46.29.49.71/Documentacion/TeenControl.html>

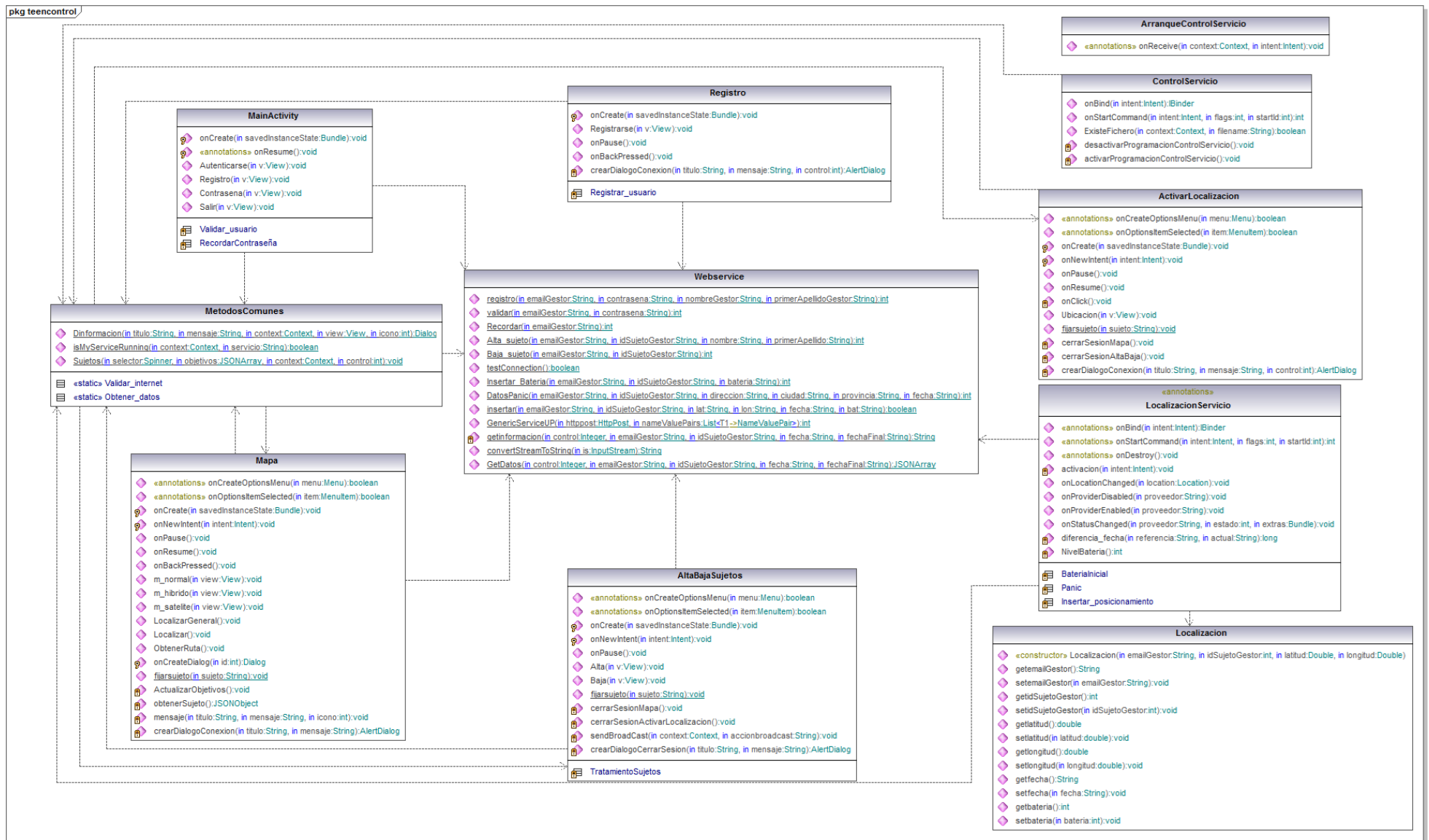


Figura 60

6.2.3. Arquitectura del sistema

La arquitectura del sistema se basa en el patrón de arquitectura de software denominado modelo-vista-controlador (MVC).

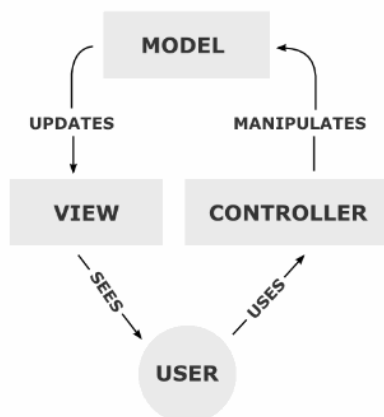


Figura 61

He de indicar que no he seguido un patrón MVC puro, ya que en mi arquitectura la Vista no tiene acceso directo al Modelo, y es el Controlador quien se encarga de servir los datos del Modelo a la Vista. Es una arquitectura similar al MVC usado por Apple en su framework Cocoa denominado Modelo-Interface-Control.

Del mismo modo comentar que en esta primera versión de TeenControl no ha dado tiempo de separar completamente el código del modelo del código controlador y estos se encuentran de manera mixta en los ficheros correspondientes a los Webservices que dan Servicio a TeenControl. Ha excepción de las credenciales de acceso que sí que se han separado en un fichero independiente.

- Modelo: Estaría formado por la BBDD MySQL, el fichero PHP con las credenciales de acceso a la misma y el conjunto de operaciones de consulta, actualización, altas y bajas de los diferentes Webservices.
- Vista: Está definido como el conjunto de layouts programados en la aplicación Android que se encargan de mostrar la información tratada por los Activities y los FragmentActivities.
- Controlador: Estaría formado por cada una de las clases Java implementadas (entre las funciones de las cuales estarían las de capturar las peticiones de los usuarios que interactúan con la vista) + el código PHP de los Webservices que se encarga de hacer de intermediario entre las peticiones de los usuarios y el Modelo. Una vez ejecutadas las operaciones de tratamiento de datos correspondientes al Modelo, el WebService sirve la información a las clases Java, que a su vez la sirve a la Vista para ser mostrada al usuario.

7. IMPLEMENTACIÓN

En esta etapa se explicarán cuáles son las decisiones que se han tomado a nivel tecnológico para dar respuesta a los requisitos funcionales y no funcionales que se fijaron en etapas anteriores. Se detallarán que herramientas de desarrollo se han utilizado, que lenguajes de programación se han elegido, cual es el sistema gestor de base de datos implementado, proveedores externos contratados para dar servicio al proyecto, las APIs necesarias, como se

gestiona el flujo de información, así como implementaciones concretas que han servido para dar cobertura a necesidades del proyecto.

7.1. Entorno de desarrollo elegido

Para decidir cuáles son las herramientas a utilizar para componer el entorno de desarrollo, se han tenido en cuenta 2 variables:

- Utilizar software libre
- Utilizar software sobre el que se tiene alguna experiencia anterior de uso

Por este motivo se han seleccionado las siguientes herramientas para desarrollar TeenControl:

- IDE: Eclipse Luna
- Lenguaje programación aplicación Android: Java
- Lenguaje programación Webservices: PHP
- SGBD: Mysql

Por último se utilizan herramientas vinculadas inherentemente al desarrollo de aplicaciones Android con Java como son:

- Kit de desarrollo de Android (SDK)
- Herramientas de desarrollo de Android (ADT) + Herramientas de las plataformas (Platform tools)
- Kit de desarrollo de java (JDK)

El proceso de instalación y configuración que se siguió para configuración del entorno de trabajo, es el proceso que está perfectamente documentado en la siguiente url: <http://www.sgoliver.net/blog/entorno-de-desarrollo-android/>

7.2. Solución de Hosting

Para realizar la primera parte de la implementación se configuró un Servidor local mediante la herramienta WAMP Server EasyPHP DevServer 14.1 VC1.

Esta herramienta ofrece todos los servicios necesarios para comenzar el desarrollo que ha de funcionar bajo el paraguas de una arquitectura Modelo Vista Controlador (MVC). Ya que dicha herramienta nos provee de un Servidor Web (Apache), que albergará los Webservices programados en PHP (Módulo que también incorpora la herramienta) que junto con las clases Java formarán la vista Controlador. Y por otro lado incorpora el SGBD Mysql que será el núcleo de la capa Modelo. WAMP también incorpora una herramienta escrita en PHP para la administración de la BBDD llamada PHPMyAdmin.

Todo y que dentro del ámbito académico en el que se realiza el proyecto podría ser suficiente la instalación y configuración del Servidor local comentado, se ha valorado darle un enfoque profesional a la solución y se han contratado los servicios profesionales a la empresa Hospedajes y Dominios S.L. Se ha contratado 2 servicios:

- Plan de hospedaje HyD con las siguientes características principales:

Almacenamiento	500 MB
Transferencia Mensual	20 GB

La relación de cambios y la argumentación de los mismos son los siguientes:

- Desaparece la pantalla para el recordatorio de contraseña: Esta funcionalidad se implementa en la pantalla inicial, de manera que se simplifica la usabilidad y la interacción de los usuarios.
- Desaparecen las pantallas que contienen botones únicamente con funcionalidades navegacionales: Desaparece la pantalla de "Áreas de TeenControl", "Área de Localización" y "Área de gestión". Al autenticarse se accede directamente al mapa donde se mostrarán a los sujetos y sus rutas. Para acceder a las diferentes pantallas operativas se dota a TeenControl de una "Action bar" que contiene los enlaces necesarios en función de la pantalla donde te encuentras. También simplifica la usabilidad y la interacción de los usuarios.
- Desaparecen los botones de vuelta atrás: Condicionado por la inclusión de la "Action bar" desaparecen los botones navegacionales comentados. Igualmente esta funcionalidad sigue activa ya que el SO Android incorpora este botón.
- Agrupación de alta y baja de sujetos: Se simplifica el uso agrupando las funciones de alta y baja de sujetos en una misma pantalla.
- Aparece funcionalidad "Localización global": En el momento de diseño de los prototipos no existía esta funcionalidad que aumenta las posibilidades de TeenControl y que en fase de implementación ya está recogida
- Desaparece indicador de batería en la funcionalidad "Ruta de un día concreto": No se incluye esta implementación ya que no aporta nada conocer el estado de la batería en un día concreto. Se mantiene conocer el estado de la batería en el momento de acceder a la última localización conocida de un sujeto.
- Desaparece funcionalidad "Definir hora de retorno": En su momento se decide cambiar esta funcionalidad por la funcionalidad de "Botón de pánico", ya que me pareció mucho más interesante y acorde al contexto de la aplicación TeenControl.

7.4. Diseño de los Webservices

Para servir los datos solicitados por la aplicación TeenControl, el envío de correos asociados a ciertas funcionalidades y para llevar a cabo ciertos mecanismos de control se han desarrollado un entramado de Webservices desarrollados en PHP.

La relación de los diferentes Webservices y su información relacionada se enumera a continuación:

Fichero	Objetivos	Parámetros recibidos	Salida
testConnection.php	<ul style="list-style-type: none"> • Validar conexión con el servidor 	<ul style="list-style-type: none"> • Null 	<ul style="list-style-type: none"> • "1"
registro.php	<ul style="list-style-type: none"> • Dar de alta al gestor en BBDD mediante Transact SQL 	<ul style="list-style-type: none"> • Email gestor • Contraseña • Nombre • Primer apellido 	<ul style="list-style-type: none"> • "1" Alta correcta • "0" Error en Alta
login.php	<ul style="list-style-type: none"> • Autenticar gestor 	<ul style="list-style-type: none"> • Email gestor • Contraseña 	<ul style="list-style-type: none"> • "1" Autenticación OK • "0" Autenticación KO
recordarContraseña.php	<ul style="list-style-type: none"> • Si el correo está dado de alta en TeenControl se envía correo recordando contraseña 	<ul style="list-style-type: none"> • Email gestor 	<ul style="list-style-type: none"> • "1" Correo enviado • "2" Correo no enviado

altasujeto.php	<ul style="list-style-type: none"> Dar de alta sujeto del gestor en BBDD mediante Transact SQL 	<ul style="list-style-type: none"> Email Id sujeto gestor Nombre Primer apellido 	<ul style="list-style-type: none"> "1" Alta sujeto OK "0" Alta sujeto KO
bajasujeto.php	<ul style="list-style-type: none"> Dar de baja sujeto del gestor en BBDD mediante Transact SQL 	<ul style="list-style-type: none"> Email gestor Id sujeto gestor 	<ul style="list-style-type: none"> "1" Baja sujeto OK "0" Alta sujeto KO
panic.php	<ul style="list-style-type: none"> Enviar e-mail con alerta de pánico con posicionamiento y fecha del sujeto del gestor mediante función mail 	<ul style="list-style-type: none"> Email gestor Id sujeto gestor Dirección Ciudad Provincia Fecha 	<ul style="list-style-type: none"> "1" Alerta enviada "0" Alerta no enviada
readsujetos.php	<ul style="list-style-type: none"> Devolver sujetos dados de alta del gestor 	<ul style="list-style-type: none"> Email gestor 	<ul style="list-style-type: none"> Array codificado en JSON de sujetos del gestor
inBateria.php	<ul style="list-style-type: none"> Inserción inicial nivel de batería Alerta por e-mail si nivel de batería <=10% Alerta por e-mail si recovery nivel de batería >10% 	<ul style="list-style-type: none"> Email gestor Id sujeto gestor Nivel de batería 	<ul style="list-style-type: none"> "1" Nivel de batería actualizado "0" Nivel de batería no actualizado
insert.php	<ul style="list-style-type: none"> Inserta posición sujeto Alerta por e-mail si nivel de batería <=10% Alerta por e-mail si recovery nivel de batería >10% 	<ul style="list-style-type: none"> Email gestor Id sujeto gestor Latitud Longitud Fecha Nivel de batería (si tiempo transcurrido último envío mayor > 15 min) 	<ul style="list-style-type: none"> "1" Posicionamiento insertado "0" Posicionamiento no insertado
read.php	<ul style="list-style-type: none"> Devolver última posición del sujeto del gestor 	<ul style="list-style-type: none"> Email gestor Id sujeto gestor 	<ul style="list-style-type: none"> Array codificado en JSON con 1 única posición del sujeto
readruta.php	<ul style="list-style-type: none"> Devolver el conjunto de posiciones de un sujeto en 1 día concreto 	<ul style="list-style-type: none"> Email gestor Id sujeto gestor 	<ul style="list-style-type: none"> Array codificado en JSON con todas las posiciones

7.5. Solución cartográfica

7.5.1. Descripción de la tecnología

La principal finalidad de la aplicación es ubicar a los dispositivos de los sujetos localizados. Esta localización se hace a dos niveles:

- Indicándonos la dirección concreta del sujeto
- Ubicando al sujeto visualmente en el mapa

Para poder disponer de la cartografía se ha utilizado la API Google Maps Android API v2.

Para poder hacer uso de esta API hemos de acceder con nuestra cuenta Gmail a la consola de desarrolladores de Google:

<https://code.google.com/apis/console>

Una vez estamos en la consola, nos crearemos un proyecto, al que se asignará una ID de proyecto y un número:

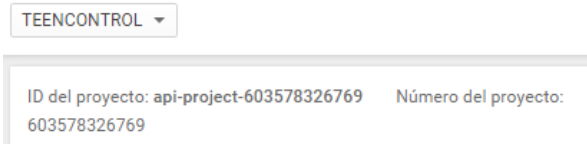


Figura 64

El siguiente paso es activar la API comentada en el apartado de “APIs y autenticación”:



Figura 65

Después estableceremos el acceso a la API. Debemos indicar las huellas digitales SHA1, que están asociadas a la clave con la que firmamos la aplicación, y el nombre del paquete.

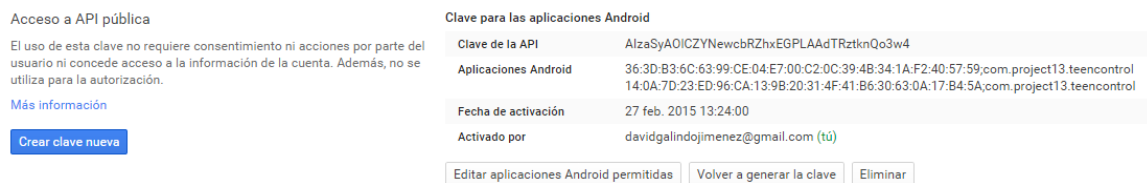


Figura 66

Como se puede apreciar en la imagen, existe una clave de la API (vinculada a las huellas digitales y nombres de paquete que hemos indicado), que tendremos que incluir en el fichero AndroidManifest.xml:

En cuanto a los permisos necesarios deberemos indicar en AndroidManifest.xml que se necesita permiso para acceder a internet, conocer el estado de la red y acceder al almacenamiento externo del dispositivo para generar la caché de mapas. Adicionalmente se ha dar permisos para poder hacer uso de los Servicios web de Google:

Por último, como API Google Maps Android API v2 utiliza OpenGL ES versión 2, deberemos indicar dicho uso también en AndroidManifest.xml.

7.5.2. Comentarios relevantes sobre código relacionado

Paso a enumera algunos fragmentos de código concretos que sirven para implementar aspectos visuales relevantes en la aplicación:

1. Carga del mapa dentro del fragmento

```
SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.map);
mapa =mapFragment.getMap();
```

2. Habilitar la brújula y los controles de zoom

```
mapa.getUiSettings().setZoomControlsEnabled(true);
mapa.getUiSettings().setCompassEnabled(true);
```

Indicar que para cambiar las posiciones de los controles de zoom se ha utilizado, el siguiente código obtenido de la URL <http://stackoverflow.com/questions/14071230/android-maps-library-v2-zoom-controls-custom-position>

3. Código para cambiar el tipo de mapa con cada uno de los botones:

```
public void m_normal(View view) {
    mapa.setMapType(GoogleMap.MAP_TYPE_NORMAL);
}

public void m_hibrido(View view) {
    mapa.setMapType(GoogleMap.MAP_TYPE_HYBRID);
}

public void m_satelite(View view) {
    mapa.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
}
```

7.6. Localización de los sujetos

7.6.1. Obtención de las localizaciones

Una vez ya tenemos definido como generamos los mapas en la aplicación, vamos a pasar a detallar de qué manera obtenemos la posición de cada sujeto para posteriormente ubicarlo en el mapa e indicar en qué dirección se encuentra.

Para esta finalidad nos serviremos de los proveedores de localización disponibles en los teléfonos móviles:

- GPS (No disponible en todos los móviles)
- Redes móviles y Wifi (proveedor disponible en todos los móviles)

El Core de la localización en TeenControl se centra en una clase llamada "LocalizaciónServicio" que extiende Service e implementa LocationListener.

Para disponer de la más alta disponibilidad del servicio, y evitar que la gestión del sistema operativo Android pudiera matar el servicio ante determinadas situaciones de gestión de recursos, se ha decidido ejecutar el servicio en modo Foreground.

La actividad que se encarga de iniciar el servicio, se encarga de pasar el gestor y el sujeto a localizar, de manera que el servicio inserte las localizaciones asociadas de manera adecuada.

De por sí la tecnología basada en GPS, es una tecnología que consume una cantidad importante de batería. Para poder abordar esta situación se le permite al gestor, en la interfaz de la Activity, decidir qué tiempo mínimo ha de pasar antes de notificar la posición (1, 5 o 10 minutos) que distancia mínima se ha tenido que recorrer (5, 15 o 30 metros) y el proveedor de localización deseado (GPS o redes móviles y Wifi). De esta manera se puede conseguir el balance equilibrado deseado entre consumo y precisión.

Si ninguno de los 2 proveedores de localización está activado se informa de esta situación. Si el proveedor de localización deseado por el gestor no se encuentra activado

también se informa de esta situación. Se comprueba si los proveedores están activados mediante un objeto LocationManager.

Para corregir las situaciones acabadas de comentar he incorporado un botón que permite acceder directamente a la activación de los proveedores de localización.

La aplicación TeenControl sirve tanto para la gestión como para el cliente localizado. Es por este motivo que una vez el gestor inicia la localización en un dispositivo se supone que cerrará la aplicación TeenControl, para que el servicio se quede activado en modo foreground enviando las localizaciones y el gestor controle al dispositivo cliente desde su propio dispositivo. Aunque TeenControl, una vez activas la localización, te da la opción mediante un AlertDialog de permanecer en la aplicación por si quieres realizar alguna otra gestión.

Una vez la Activity ha iniciado el servicio, en el mismo para decidir que proveedor se seleccionará se utiliza el objeto Criteria, al cual se le fijan algunos criterios en función de la selección del gestor. El proveedor que cumpla más criterios será seleccionado.

Como la clase implementa LocationListener, se han de implementar obligatoriamente los siguientes 4 métodos, que se utilizan de la manera detallada:

1. Los siguientes 3 métodos se utilizan para volver a seleccionar el proveedor de localizaciones, ya que se ha activado o desactivado algún proveedor o se ha producido un cambio en el estado del mismo.

```
public void onProviderDisabled(String proveedor)
public void onProviderEnabled(String proveedor)
public void onStatusChanged(String proveedor, int estado, Bundle
extras)
```

2. El método "public void onLocationChanged(Location location)" se utiliza para notificar los cambios de posición. Se obtiene la longitud, y la latitud del objeto "Location", y junto con el nivel de batería, la fecha, el gestor y el sujeto localizado se crea un objeto de la clase "Localizacion" que se inserta en BBDD mediante la tarea asíncrona "Insertar_posicionamiento" que llama al método "insertar" de la clase "Webservice" que a su vez se encargará de hacer la llamada a la URL específica con los parámetros necesarios (a través del método intermedio y genérico de la clase "Webservice" llamado "GenericServiceUP").

En caso de que no exista conexión a internet o el Servidor no esté disponible el objeto del tipo "Localizacion" se guarda en un Array, el cual será vaciado e insertado en BBDD cuando vuelva a haber conexión a internet.

La sentencia SQL que se encarga de insertar todos los datos excepto el nivel de batería es la siguiente:

```
Insert into localizaciones(emailGestor,idSujetoGestor,latitud,longitud,fecha)
values
('".$emailGestor."', '".$idSujetoGestor."', '".$latitud."', '".$longitud."', '".$
fecha."');
```

La inserción del nivel de carga de la batería se explica en el apartado 6.7.

7.6.2. Control desactivación por parte del sujeto de la localización

Para evitar que el sujeto consciente o inconscientemente desactive la localización se han implementado 2 controles a 2 niveles distintos:

- Evitar desactivar la localización por cierre de servicios o procesos:

Al iniciar el servicio de localización, de manera secuencial, se inicia un servicio llamado "ControlServicio" el cual comprueba si la primera línea del fichero llamado "controlServicio.txt", ubicado en la memoria interna, contiene el valor 1 (localización activada). Si esto es así este servicio se autoprograma para ejecutarse cada 1 minuto y comprobar que el servicio "LocalizaciónServicio" se está ejecutando. Si no se está ejecutando lo vuelve a iniciar cogiendo los parámetros de configuración (gestor, sujeto, tiempo de actualización, distancia mínima recorrida y proveedor de localización preferido) del fichero "controlServicio.txt".

- Evitar desactivar la localización por apagado del dispositivo:

Al encender el dispositivo, después de haberse apagado, cuando el sistema operativo Android envía la acción "BOOT_COMPLETED" se ejecuta la clase "ArranqueControlServicio" que extiende "BroadcastReceiver" que desencadena el arranque del Servicio llamado "ControlServicio" comentado en el punto anterior, el cual a partir de este momento ya lleva a cabo las acciones para que "LocalizaciónServicio" se ejecute. Este control implica dar permisos a TeenControl para que reciba la acción de arranque completado.

7.7. Tratamiento información devuelta por los Webservices

En el detalle que se dará en los siguientes apartados, sobre la implementación de las diferentes funcionalidades de TeenControl, se hará referencia en numerosas ocasiones a la tarea asíncrona "Obtener_Datos". Esta es una tarea genérica y en todas las llamadas a la misma se le pasa un primer parámetro de control, para saber quién ha iniciado la petición. Este mismo control se le pasa al método de la clase "Webservice" que es llamado, para que luego se gestione la llamada a la URL correspondiente en función del origen de la petición. En el caso del botón de pánico, la localización global y la localización específica el control=1, en la generación de rutas el control=2 y en la llamada para obtener el total de sujetos dados de alta por el gestor el control=3.

En los Webservices llamados, para obtener dicha información, esta se devuelve codificada en el formato ligero para el intercambio de datos denominado JSON, recogiendo el resultado de la sentencias SQL e introduciéndolos en un Array multidimensional, para posteriormente codificarlos en formato JSON y devolverlos:

```
$json = array();

if (mysqli_num_rows($query_exec)) {
    while($row=mysqli_fetch_assoc($query_exec)){
        $json['datos'][] = $row;
    }
}
.
.
echo json_encode($json);
```

Posteriormente en el método correspondiente de la clase Webservices se recogen cada uno de los objetos formados por un Array y se van introduciendo en un objeto del tipo JSONArray que será utilizado en el posterior tratamiento de los datos:

```
try {
    json = new JSONObject(data);
    jsonArray = json.optJSONArray("datos");
```

```

} catch (JSONException e) {
    e.printStackTrace();
}

```

7.8. Botón de pánico

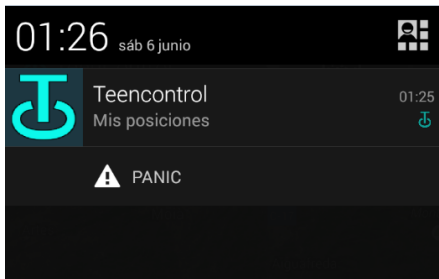


Figura 67

Una vez se activa la localización en el dispositivo localizado, tal y como hemos comentado se inicia el servicio “LocalizacionServicio” en modo Foreground. Esto implica que se nos muestre una notificación en el área de notificación de Android, que informa que el servicio se encuentra en primer plano.

He aprovechado esta notificación, para incorporar un botón que nos prestará la función de botón de pánico para que los sujetos que están

siendo localizados lo puedan utilizar si creyesen en algún momento en se encuentran en una situación de emergencia.

El código que hace que aparezca dicho botón en la notificación es el siguiente:

```

PendingIntent pendingIntent = PendingIntent.getActivity(this, 0,
notificationIntent, 0);

Intent panicIntent = new Intent(this, LocalizacionServicio.class);
panicIntent.setAction("ACTION.PANIC_ACTION");

PendingIntent ppanicIntent = PendingIntent.getService(this, 0, panicIntent,
0);

Notification notification = new NotificationCompat.Builder(this)
.addAction(android.R.drawable.stat_sys_warning, "PANIC", ppanicIntent).build();

```

Comentar que, por el comportamiento normal de los botones de las notificaciones del área de notificación, si se recibe una notificación prioritaria del SO Android u otra aplicación que se sitúe sobre nuestra notificación, el botón de pánico se oculta. Se ha de descartar la notificación superior y el botón vuelve a aparecer.

Al pulsar en el botón se genera un nuevo Intent de inicio de servicio con la acción ("ACTION.PANIC_ACTION"), ante esta situación lo que sucede es que se realiza una llamada a la tarea asíncrona “Obtener_datos”, que a su vez llama al método “GetDatos” de la clase “Webservice” que a su vez se encarga (mediante el uso de otros métodos de la clase) de hacer la llamada a la URL específica con los parámetros necesarios. El resultado final es que se devuelve un JSONArray con el último dato de posicionamiento del sujeto que está siendo localizado (En esta ocasión el JSONArray sólo contiene 1 objeto).

De este objeto se obtiene la longitud, la latitud y la fecha del último posicionamiento insertado en BBDD del sujeto. A partir de la longitud y la latitud mediante un objeto Geocoder, se obtiene la dirección, la ciudad y la provincia.

Si por algún motivo no existiese posicionamiento en BBDD se recoge la última longitud y la latitud conocida del proveedor de posicionamiento activo. Y se aplica de nuevo el procedimiento de codificar la longitud y la latitud comentada en el párrafo anterior. Si finalmente el proveedor de localización activado no tiene una ubicación histórica, se envían los parámetros “dirección”, “ciudad” y “provincia” nulos y se gestiona en el Webservice PHP esta situación para construir un mail en este sentido.

Una vez que tenemos todos los datos mediante la tarea asíncrona “Panic” se llama al método “DatosPanic” de la clase “Webservice” que se encargará de llamar a una URL específica con los parámetros necesarios (a través del método intermedio y genérico de la clase “Webservice” llamado “GenericServiceUP”). Una vez la petición llega al Webservice se realiza el envío de un correo electrónico mediante el siguiente código correspondiente del webservice panic.php

Ejemplos de correos de alerta de pánico con y sin posicionamiento conocido:

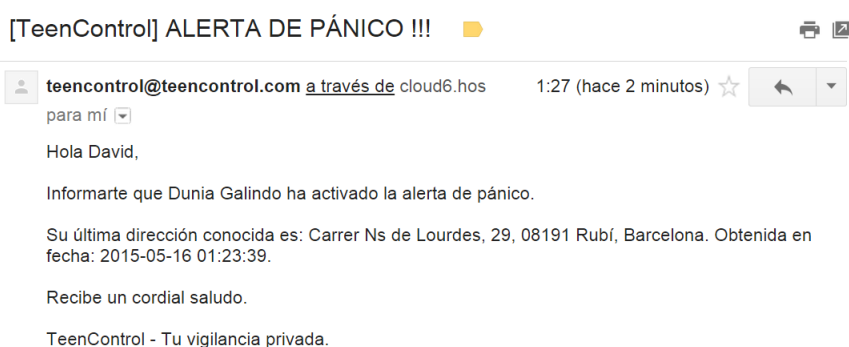


Figura 68



Figura 69

7.9. Localizaciones del sujeto en el mapa

Una vez que ya tenemos implementada la cartografía y la obtención de las localizaciones de cada sujeto, estamos en disposición de poder situar al sujeto en el mapa con su última localización conocida y poder visionar rutas históricas recorridas por los sujetos localizados, para lo que necesitaremos implementar el código apropiado para representar esto visualmente.

7.9.1. Localización Global

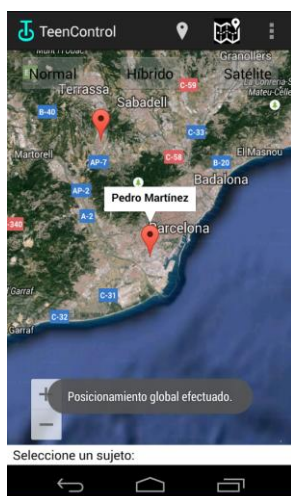


Figura 70

Cuando el gestor se autentica, la primera acción que realiza TeenControl es mostrar el mapa y realizar una localización global de todos los sujetos que tiene localizado el gestor.

Esta localización se realiza desde una panorámica bastante alejada por si sus sujetos estuviesen muy dispersos. Se añade un marcador a cada uno de ellos y se les sitúa una etiqueta sobre el marcador en el que únicamente se visualiza el nombre y el primer apellido del sujeto localizado.

En el método “onCreate” mediante la llamada al método “ActualizarObjetivos” se ejecuta la tarea asíncrona “Obtener_datos” que a su vez llama al método “GetDatos” de la clase “Webservice” que se encarga (mediante el uso de otros métodos de la clase) de hacer la llamada a la URL específica con los parámetros necesarios. El resultado final es que se devuelve un JSONArray con cada uno de los sujetos localizados.

Una vez disponemos de los sujetos localizados del Gestor mediante un For se itera sobre cada uno de los sujetos obteniendo la última posición de cada uno de ellos (no me reiteraré en la forma de obtener la última posición de cada sujeto ya que es de la misma manera que indicábamos en el apartado 6.4 “Botón de pánico”). Para cada una de las últimas localizaciones de cada sujeto se crea un marcador y la etiqueta comentada al comienzo del apartado, las cuales serán añadidas al mapa.

El método “showInfoWindow()” hace que sólo se muestre la etiqueta del último marcador generado. Para ver la etiqueta del resto tendremos que pulsar sobre cada uno de los marcadores.

Por último comentar que el gestor de manera activa puede volver a ejecutar la acción de “Localización global” ya que es una de las opciones disponibles en el menú de la “Action bar” de la clase Mapa que extiende FragmentActivity.

7.9.2. Localización específica



Figura 71

Cuando ya tenemos realizada la localización global efectuada, una de las opciones que puede llevar a cabo el gestor es una localización más específica de cada uno de sus sujetos.

Esta localización consistirá en una visualización mucho más cerca del sujeto, en la que se mostrará una etiqueta que contendrá el nombre del sujeto, la fecha y hora de la localización y la dirección completa. También se mostrará en el mapa el nivel de batería del dispositivo del sujeto, pero este aspecto se tratará en un apartado posterior.

Para poder realizar la localización específica se debe seleccionar previamente el sujeto. Esta selección se puede hacer pulsando en la opción “Seleccionar sujeto” de la “Action bar” o directamente pulsando en el Spinner de la parte inferior del mapa.

El método para mostrar la localización específica en el mapa es el mismo que en la localización global, pero sin iterar en un For sobre el total de sujetos, ya que ya hemos especificado el sujeto sobre el que queremos realizar la localización.

7.9.3. Ruta diaria de sujeto

Otra de las opciones principales que puede llevar a cabo es visualizar la ruta recorrida por un sujeto en un día concreto. Como en el caso de la localización específica es necesario haber seleccionado previamente al sujeto deseado.

Existen 3 diferencias respecto a la implementación de la localización específica:

- Se realiza la creación de 2 marcadores. Uno al inicio de la ruta y otro al final. Por este motivo la dirección de la etiqueta de cada marcador es diferente correspondiendo a cada posición concreta. Al igual que en la “Localización global” se ha de pulsar sobre cada marcador para ver la etiqueta no mostrada (En este caso la etiqueta del marcador origen).
- Se utiliza un objeto del tipo PolylineOptions para ir generando en un For una línea entre cada uno de las diferentes posiciones registradas, para finalmente añadir al mapa dichas líneas que unen la posición inicial de la ruta con la posición final
- Se utiliza un objeto del tipo DatePickerDialog que tiene un OnDateSetListener() para devolver la el día mes y año una vez se ha fijado. Yo recojo estos 3 parámetros para generar 2 fechas una inicial (con tiempo 00:00:00) y otra final (con tiempo 23:59:59). Entonces se llama a la tarea asíncrona “Obtener_Datos” de la misma manera como lo hacíamos con el botón de pánico o la localización específica, pero esta vez pasando los parámetros de fecha inicial y fecha final.

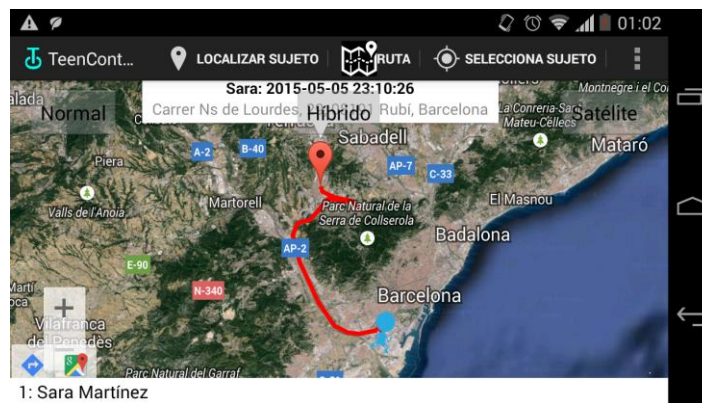


Figura 72

7.10. Implementación de control del nivel de batería del dispositivo localizado

Relacionado con el nivel de la batería se ejecutan 3 acciones:

1. Conocer el nivel de batería del dispositivo del sujeto localizado insertándolo en BBDD
2. Informar gráficamente en el mapa del nivel de batería del sujeto cuando el gestor hace una localización precisa.

3. Informar al gestor cuando el nivel de batería sea inferior al 10%.

7.10.1. Nivel de batería

Se realiza un primer proceso inicial que consisten en registrar el nivel de batería en el momento de arrancar el servicio. El nivel de batería se puede obtener con el método indicado en:

<http://developer.android.com/training/monitoring-device-state/battery-monitoring.html>

Que fundamentalmente consiste en las siguientes líneas de código:

```
IntentFilter ifilter = new IntentFilter(Intent.ACTION_BATTERY_CHANGED);  
batteryStatus = this.registerReceiver(null, ifilter);
```

Una vez tengo el estado de la batería asigno el porcentaje de batería a una variable de tipo integer que se utilizará para actualizar dicho valor en BBDD.

```
int level = batteryStatus.getIntExtra(BatteryManager.EXTRA_LEVEL, -1);
```

Concretamente se actualizará el valor en la tabla sujetos que corresponde al sujeto localizado, mediante la llamada a una tarea asíncrona llamada "BateriaInicial" que a su vez llamará al método "Insertar_Bateria" de la clase "Webservice" que se encargará de hacer la petición a la URL específica con los parámetros necesarios (a través del método intermedio y genérico de la clase "Webservice" llamado "GenericServiceUP").

A partir de este de este momento, cada vez que se obtiene una nueva localización, se obtiene el nivel de batería del mismo modo. Pero existe un método llamado "private long diferencia_fecha(String referencia, String actual)" que comprueba si han pasado 15 minutos, respecto a una fecha de referencia, si los han pasado se hace un Update del nivel de batería del sujeto. La fecha de referencia se actualiza con la fecha de inserción del último nivel de batería.

Si no han pasado los 15 minutos, la tarea asíncrona "Insertar_posicionamiento" que llama al método "insertar" de la clase "Webservice" pasará el parámetro de la batería con un 0 a la URL específica (a través del método intermedio y genérico de la clase "Webservice" llamado "GenericServiceUP"), y el Webservice PHP ya sabrá que no tiene que actualizar el valor de la misma.

La sentencia SQL que se encarga de actualizar el nivel de batería tanto en el momento inicial, como en las posteriores actualizaciones cuando la diferencia de tiempo es superior a 15 minutos es:

```
update sujetos set bateria = ".$bateria."  
where emailGestor=".$emailGestor."  
and idSujetoGestor = ".$idSujetoGestor."";
```

7.10.2. Indicador gráfico de la batería

El indicador gráfico de la batería aparece cada vez que el gestor realiza una localización precisa.

Esto se consigue insertando una imagen en el Layout de la clase Mapa que extiende android.support.v4.app.FragmentActivity. Inicialmente esta imagen es una imagen sin contenido que no se puede apreciar. Esta imagen está acompañada de una etiqueta que muestra el porcentaje de carga, la cual también está oculta inicialmente.

Cuando el gestor ha seleccionado al sujeto y ha pulsado sobre la acción “Localizar sujeto” se ejecuta en la clase Mapa una tarea asíncrona llamada “Obtener_datos” que llama al método “GetDatos” de la clase “Webservice”, que a su vez se encarga (mediante el uso de otros métodos de la clase) de hacer la llamada a la URL específica con los parámetros necesarios. El resultado final es que se devuelve un JSONArray con el último dato de posicionamiento y el nivel de batería del sujeto seleccionado. (En esta acción el JSONArray está formado por un solo objeto). De este único objeto se obtiene el valor de la batería del sujeto, y en función del valor de la batería se cambia la imagen del Layout por la correspondiente en función del nivel de carga y se hace visible la etiqueta que muestra el porcentaje de carga.

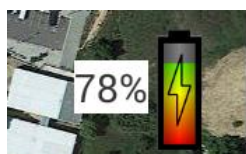


Figura 73

7.10.3. Notificación de nivel crítico de batería

Tanto en el proceso inicial de actualizar el valor de la carga de la batería, como en los procesos posteriores cuando haya pasado un tiempo mínimo de 15 minutos, existe un control por si la batería del dispositivo del sujeto localizado se encontrase igual o por debajo del 10%.

Lo que se hace en el Webservice inBateria.php (Webservice para la actualización de batería inicial) y en el Webservice insert.php (Que además de insertar las localizaciones, sirve para actualizar el nivel de batería si ha pasado un mínimo de 15 minutos), es comprobar si la batería se encuentra igual o por debajo del 10% de nivel de carga. Y si se encuentra en estas circunstancias se envía un mail al gestor del sujeto para que este al corriente de que podría quedarse sin localizaciones por apagarse el dispositivo.

Además se ha implementado un control, para que cada vez que se chequee el nivel de batería si este está igual o por debajo del 10% no vuelva a enviar otro correo electrónico. Lo que se hace es utilizar una variable de control llamada controlCarga, de manera que si el nivel de carga está igual o por debajo del 10% y esta variable está fijada a 1 quiere decir que ya se ha notificado el nivel crítico de batería y no se vuelve a notificar.

Del mismo modo cuando el dispositivo vuelve a tener el nivel de carga por encima del 10% se notifica al gestor que el dispositivo del sujeto localizado ya no se encuentra en un nivel de carga crítico. En el momento de esta notificación se vuelve a poner la variable controlCarga=0, para que si vuelve a disminuir el nivel se vuelva a notificar.

Ejemplos de correos de nivel de carga crítico y nivel de carga que ya no se encuentra en estado crítico:



Figura 74



Figura 75

7.11. Otras implementaciones

Todo y que ya hemos descrito las principales funcionalidades que forman el Core de TeenControl, existen otras funcionalidades secundarias pero imprescindibles para el funcionamiento de TeenControl.

7.11.1. Registro

Para poder ejecutar TeenControl primero es necesario registrarse.

Cuando accedemos a la pantalla de registro se han de rellenar todos los campos y pulsar sobre el botón “Nueva cuenta”.

En esta implementación simplemente controlamos que todos los campos estén rellenos, que la contraseña y la confirmación de contraseña coincidan y que la contraseña tenga un mínimo de 8 caracteres.

Una vez tenemos todos los campos correctamente rellenos se ejecuta la tarea asíncrona “Registrar_usuario” que ejecuta el método “registro” de la clase Webservice y que devuelve el resultado del registro.

7.11.2. Recordar contraseña

Desde la pantalla inicial se puede solicitar el que se recuerde la contraseña de acceso a TeenControl.

Una vez pulsas sobre la etiqueta “Recordar contraseña”, se comprueba que el usuario (el cual ha de ser una cuenta de correo electrónico) se ha introducido en el campo E-mail. Si se ha introducido se ejecuta la tarea asíncrona “RecordarContraseña” que llama al método “Recordar” de la clase Webservice.

Si el correo electrónico no se encuentra en la BBDD se informa de esta situación. Por el contrario si la cuenta de correo existe se envía el correo electrónico correspondiente.

Ejemplo de correo de recordatorio de contraseña:

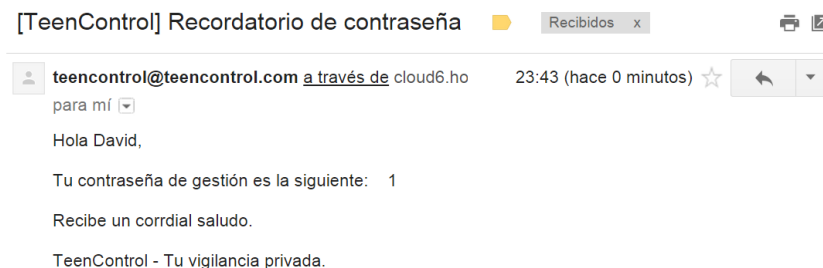


Figura 76

7.11.3. Alta y baja de sujetos

7.11.3.1. Alta de sujetos

Una vez el gestor se autentica en TeenControl lo primero que debería hacer es dar de alta sus sujetos a localizar, para posteriormente activar la localización sobre ellos.

El sistema de TeenControl permite que el nombre y usuario del sujeto a localizar ya exista (Ya que puede darse la circunstancia que se tengan a 2 sujetos a localizar con el mismo nombre y primer apellido). Por lo que lo que identifica al sujeto es un identificador numérico.

Como los identificadores de los sujetos puede ser que no sean consecutivos (por bajas que se hayan producido). Lo primero que hace TeenControl al dar de alta a un sujeto en el método "public void Alta (View v)" es buscar el primer "id" disponible.

Una vez todos los campos se han rellenado y se ha pulsado sobre el botón "Alta", se ejecuta la tarea asíncrona TratamientoSujetos (pasando como parámetro de control un "1" para indicar que se trata de un alta), la cual llama al método "Alta_sujeto" de la clase "Webserice" que devuelve el resultado de la operación.

7.11.3.2. Baja de sujetos

El gestor, por equivocación al dar de alta al sujeto o porque ya no se pretende localizar más a un sujeto, tiene la opción de darlos de baja.

Para llevar a cabo esta acción, se ha de seleccionar del desplegable el sujeto que se desea dar de baja y pulsar sobre el botón "Baja".

Una vez pulsado el botón se ejecuta la tarea asíncrona TratamientoSujetos (pasando como parámetro de control un "2" para indicar que se trata de una baja), la cual llama al método "Baja_sujeto" de la clase "Webserice" que devuelve el resultado de la operación.

Como en la BBDD están implementadas las claves foráneas, al eliminar el sujeto se eliminan todas sus posiciones, ya que así se lo he indicado al crear la clave foránea:

7.11.4. Sincronización entre Activities de sujetos

TeenControl tiene implementado un control que permite que el FragmentActivity correspondiente a la clase "Mapa" y la Activity correspondiente a la clase "ActivarLocalización"

mantengan actualizado el total de sujetos dados de alta por un gestor, mientras cada una de ellas permanezca abierta.

Esto se consigue mediante la implementación de BroadcastReceiver en cada una de las 2 clases comentadas anteriormente, que escuchan los broadcast enviados a la espera de que alguno de ellos incluya la acción "ACTUALIZACION_SUJETOS_MAPA". Por este motivo en la clase "AltaBajaSujetos" cada vez que se da de alta o baja un sujeto se envía un broadcast con el nombre de la acción identificadora ("ACTUALIZACION_SUJETOS_MAPA" en este caso) y un Intent que contiene como parámetro el total de sujetos del gestor

7.11.5. Implementación de controles

Cada una de las operaciones, llevadas a cabo en TeenControl, tienen implementadas uno o varios controles sobre casuísticas concretas, para evitar excepciones de ejecución e informar de dichas situaciones a los usuarios de la aplicación.

Los controles más reseñables son:

- Servidor no disponible o no conexión a internet
- Credenciales facilitadas en la autenticación no válidas
- Indicar cuando existen campos vacíos obligatorios en las operaciones
- Al intentar dar de alta a un gestor si la cuenta de correo, que servirá como identificador, se encuentra ya dada de alta en TeenControl se informa de esta situación
- Si al dar de alta un gestor la confirmación de contraseña no coincide con la contraseña se informa de esta situación
- Si la contraseña de gestor no tiene un mínimo de 8 caracteres se informa de esta situación
- Si se ejecuta la funcionalidad de recordar contraseña sin introducir la cuenta de correo se informa de esta situación
- Si se introduce una cuenta de correo de gestor para que se recuerde la contraseña y esta no existe dada de alta en TeenControl se informa de esta situación
- La no selección de un sujeto cuando se intenta ejecutar una funcionalidad que requiere de esta acción previa
- No existencia de sujetos dados de alta cuando se intenta ejecutar alguna de las funcionalidades vinculadas a sujetos
- La no existencia de datos de posicionamiento cuando se intenta ejecutar alguna funcionalidad que precisa de ellos
- No tener ningún proveedor de localización activado, cuando se intenta activar la localización sobre un sujeto
- Información conforme se ha seleccionado como favorito un proveedor de localización que no se encuentra activado

Capturas de ejemplo de algunos de estos controles:

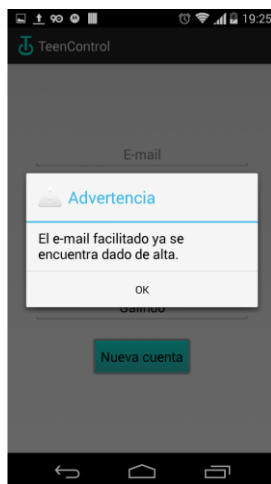


Figura 77

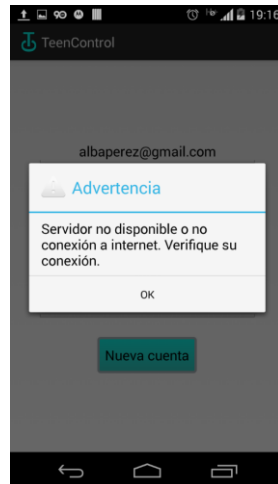


Figura 78

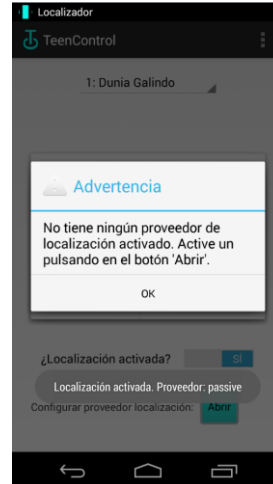


Figura 79

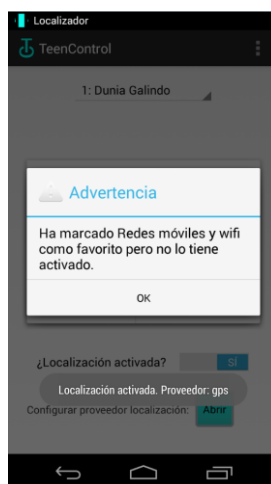


Figura 80

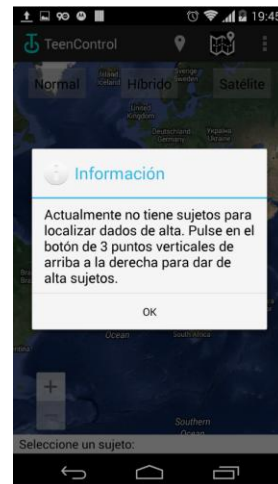


Figura 81

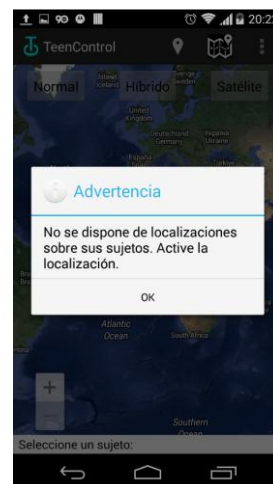


Figura 82

8. CONCLUSIONES

Como análisis final, me gustaría concluir diciendo que, en mi modesta opinión, globalmente se han cumplido los objetivos inicialmente fijados. El detalle de porqué llego a esta conclusión es el siguiente:

- En primer lugar he conseguido crear una aplicación con unas funcionalidades claramente definidas y totalmente operativas.
- He puesto en práctica varios conocimientos adquiridos durante mi formación en la Ingeniería Técnica de Sistemas. Por poner algún ejemplo:
 - He utilizado conocimientos adquiridos en Competencia Comunicativa para profesionales de las TIC en la elaboración de la memoria del presente trabajo de final de carrera

- Se han aplicado procedimientos explicados en Ingeniería del Software para detectar requisitos funcionales y no funcionales y elaboración de los casos de uso
- He aplicado enseñanzas que se impartieron en BBDD I en la manipulación de los datos de mi BBDD.
- Se han aplicado conceptos explicados en Estructura de la Información para el almacenamiento de información en el código implementado en Java
- He realizado el ciclo completo en la elaboración de un programa. Experimentando de primera mano el método de desarrollo iterativo e incremental. Viendo cómo se podía trabajar en diferentes fases, evolucionando en cada iteración los entregables de las mismas.
- Gracias a los procesos de evaluación de prototipos y testeo de la aplicación, tanto a nivel personal, como por colaboradores, creo que se ha conseguido un producto con el mínimo exigible de calidad para poder ser distribuido a través de la plataforma de aplicaciones Play Store de Android.
- Se ha conseguido una correcta integración de las diferentes plataformas que componen la solución global (BBDD en MySQL, Webservices en PHP, desarrollo Java de la aplicación, fuentes XML de las interfaces,...), consiguiendo una correcta fluidez de la aplicación, aparentemente libre de errores.
- He adquirido nuevos conocimientos sobre herramientas vinculadas al desarrollo de un producto de software
 - Creación de prototipos horizontales y creación de simulaciones con Justinmind Prototyper v6.4.1
 - Creación de diagramas UML y documentación de la aplicación con Altova Umodel
 - Creación de videos con la integración del móvil en el PC con Mobizen de Rsupport

9. GLOSARIO

A continuación se enumera el significado del conjunto de acrónimos o palabras técnicas aparecidos durante la memoria técnica para una mayor comprensión de la misma.

ADT	Android Developments Tools. Complemento del software Eclipse que contiene herramientas para el desarrollo de programas Android.
API	Application Programming Interface. Interfaz de programación de aplicaciones que contiene un conjunto de subrutinas, funciones y procedimientos que es utilizado por otro software como una capa de abstracción.
BBDD	Base de datos. Bancos de información que contienen datos relativos a diversas temáticas y categorizados de distinta manera, pero que comparten entre sí algún tipo de vínculo o relación que busca ordenarlos y clasificarlos en conjunto.
APK	Application Package File. Formato de archivo que se usa para empaquetar aplicaciones para el sistema operativo para móviles Android.
DVD-DL	Disco Versátil Digital de Doble Capa que puede almacenar hasta 8.3GB de información.
ESO	Hace referencia a las siglas de Educación Secundaria Obligatoria en España.

GPS	Sistema de posicionamiento global que permite determinar en todo el mundo la posición de un objeto.
HDD	Hard Disk Drive. Unidad de disco rígido que se utiliza como dispositivo de almacenamiento de datos.
IDE	Integrated Development Environment. Entorno de desarrollo integrado. Es una aplicación que proporciona servicios integrales para facilitar el desarrollo de software.
INE	Instituto Nacional de estadística: Organismo autónomo de España encargado de la coordinación general de los servicios estadísticos de la Administración General del Estado y la vigilancia, control y supervisión de los procedimientos técnicos de los mismos.
JAVA	Java es un lenguaje de programación de propósito general, concurrente orientado a objetos.
JDK	Java Development Kit es un software que provee herramientas de desarrollo para la creación de programas en Java.
JSON	JavaScript Object Notation, es un formato ligero para el intercambio de datos.
Mpx	Un pixel es la menor unidad homogénea en color que forma parte de una imagen digital. Mpx Significa megapixel, es decir 1 millón de píxeles.
MVC	Modelo Vista Controlador. Es un patrón de diseño en la ingeniería del software.
PC	Personal Computer. Ordenador personal
POP	Post Office Protocol. Protocolo diseñado para la recepción de correo electrónico.
PHP	es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico
RAM	Random-Access Memory. Memoria de acceso aleatorio utilizada como memoria de trabajo de computadoras para el sistema operativo, los programas y la mayor parte del software.
RPM	Revolución por minuto. Número de rotaciones completadas cada minuto por un cuerpo que gira alrededor de un eje.
SDK	Software Development Kit. Conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto.
SGBD	Sistema Gestor de Base de Datos. es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos.
Sketch	Sinónimo de esbozo
SQL	Structured Query Language. Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.
SO	Sistema operativo
TIC	Tecnologías de la información y la comunicación
UML	Unified Modeling Language. s el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad
URL	Uniform Resource Locator. Es un identificador de recursos uniforme (URI) cuyos recursos referidos pueden cambiar, esto es, la dirección puede apuntar a recursos variables en el tiempo.
Webservice	Es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones
WIDI	Protocolo utilizado para la transmisión de imagen y sonido mediante Wifi.
WIFI	Es un mecanismo de conexión de dispositivos electrónicos de forma inalámbrica.
XML	eXtensible Markup Language. Es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.

10. BIBLIOGRAFÍA

A parte de la documentación facilitada para la realización de la asignatura TFC y del resto de documentación de otras asignaturas de la carrera, se han utilizado las siguientes referencias externas:

I. Libros de texto

Android 4. Desarrollo de aplicaciones (Anaya Multimedia/Wrox), como libro de consulta general a lo largo de la realización del proyecto.



Figura 83

II. Referencias en la red

- Posicionamiento y mapas

<http://www.androidcurso.com/index.php/curso-android-basico/tutoriales-android-basico/41-unidad-7-seguridad-y-posicionamiento/284-localizacion>

<http://www.maestrosdelweb.com/editorial/curso-android-geolocalizacion-utilizacion-mapas-google/>

<http://www.androidcurso.com/index.php/tutoriales-android/41-unidad-7-seguridad-y-posicionamiento/223-google-maps-api-v2>

<http://www.sgoliver.net/blog/?p=1932>

<http://www.sgoliver.net/blog/mapas-en-android-google-maps-android-api-v2-i/>

<http://www.sgoliver.net/blog/mapas-en-android-google-maps-android-api-v2-ii/>

<http://www.sgoliver.net/blog/mapas-en-android-google-maps-android-api-v2-iii/#>

<http://stackoverflow.com/questions/14071230/android-maps-library-v2-zoom-controls-custom-position>

- Datepicker

http://www.tutorialspoint.com/android/android_datepicker_control.htm

- Ficheros en memoria interna

<http://www.sgoliver.net/blog/ficheros-en-android-i-memoria-interna/>

- Interrelación entre Webservices y la BBDD

<http://picarcodigo.blogspot.com.es/2014/05/webservice-conexiones-base-de-datos.html>

<http://picarcodigo.blogspot.com.es/2014/06/webservice-obtener-datos-mysql-desde.html>

<http://notasweb.com/articulo/php/obtener-un-campo-de-una-base-de-datos-mysql-en-php.html>

<http://www.codeproject.com/Articles/267023/Send-and-receive-json-between-android-and-php>

<http://json.parser.online.fr/>

- Permisos sobre BBDD

<https://blog.infranetworking.com/crear-base-de-datos-mysql-dar-privilegios/>

- Animaciones

<http://cursoandroidstudio.blogspot.com.es/2014/12/animacion-transicion-entre-activities.html>

<http://developer.android.com/guide/topics/resources/animation-resource.html>

- Tareas Asynk Task

<http://jarroba.com/asynctask-en-android/>

<http://www.androidcurso.com/index.php/tutoriales-android-basico/36-unidad-5-entradas-en-android-teclado-pantalla-tactil-y-sensores/365-ejecutar-una-tarea-en-un-nuevo-hilo-con-asynctask>

<http://picarcodigo.blogspot.com.es/2014/10/tareas-en-2-plano-android-parte-i-clase.html>

- Estado de la batería

<http://developer.android.com/training/monitoring-device-state/battery-monitoring.html>

- Tratamiento imágenes

<http://stackoverflow.com/questions/2294569/show-hide-image-in-android>

<http://stackoverflow.com/questions/8937180/how-to-show-image-in-android-using-java-code>

<http://publicdomainvectors.org/es/vectoriales-gratuitas/Ilustraci%C3%B3n-vectorial-de-conjunto-de-iconos-de-estado-nivel-pila-diferente/17434.html>

- Array

<http://jagonzalez.org/arraylist-de-objetos-en-java-con-ejemplo-practico/>

- Orientación Activities

<http://stackoverflow.com/questions/27601050/how-to-disable-rotation-in-all-activities-which-belong-to-my-android-application>

- Comportamiento Activities en función de la orientación

<http://stackoverflow.com/questions/17945130/prevent-activity-restarting-when-orientation-changes>

- Servicios

<http://androidgps.blogspot.com.es/2008/09/simple-android-tracklogging-service.html>

<http://stackoverflow.com/questions/14896574/how-to-call-a-method-in-activity-from-a-service>

<http://javatechig.com/android/android-service-example>

<http://developer.android.com/reference/android/app/Service.html>

<http://www.truiton.com/2014/10/android-foreground-service-example/>

- Broadcastreceiver

<http://www.nosinmiubuntu.com/enviar-y-recibir-acciones-con/>

<http://www.vogella.com/tutorials/AndroidBroadcastReceiver/article.html>

- Spinner

<http://stackoverflow.com/questions/9863378/how-to-hide-one-item-in-an-android-spinner>