

City Angels

Victor Montes Garcia, Universitat Autònoma de Barcelona

Resumen— City Angels se desarrolla con el objetivo de conectar en caso de emergencia a las personas con necesidades especiales (que pueden ir desde niños pequeños que requieran supervisión adulta, a personas mayores con problemas de movilidad y/o lucidez) y las personas que desean ayudarlas o bien se dedican a su supervisión. Este sistema engloba una red social para el registro y gestión de usuarios, un panel de control y una aplicación móvil, cuyo objetivo es la notificación de alertas lanzadas por personas con necesidades y la búsqueda de personas dispuestas a ayudar que estén próximas a la alerta, proporcionando a ambas partes un sistema que cumpla con sus necesidades en cada caso (desde proporcionar contactos de emergencia, llamadas a servicios de emergencia o dar la información necesaria para llegar al lugar donde se encuentra la persona que necesita ayuda).

Palabras clave — accesibilidad, emergencias, notificación, búsqueda de voluntarios, alerta, contactos, familia, voluntarios, asistencia, geolocalización, red social, web service, Android, panel de control

Abstract— City Angels has been developed with the goal of connecting, in an emergency case, the people with special needs (which can be kids, disabled individuals, elder people...) and the people who want to help them or have responsibility on their supervision. This system encompasses a social network, a control panel and a mobile application, that allows the notification of alerts from the people with special needs and the search of the closest people who can help, bringing a system that enables both parts to achieve their needs (providing emergency contacts, emergency calls or giving information to reach the person who needs help).

Index Terms — accessibility, emergencies, notification, volunteer search, alert, contacts, family, volunteers, assistance, geolocation, social network, web service, Android, control panel



1 INTRODUCCIÓN

ACTUALMENTE la tecnología ha proporcionado a las personas que necesiten una atención especial y/o ayuda (ya sea médica o de otros tipos) la posibilidad de solicitarla cuando quieran independientemente de su lugar y circunstancias. Paralelamente la tecnología ha permitido que las personas que desean ofrecer su ayuda y tiempo a los demás tengan muchas más facilidades para posibilitarlo, ya sea en forma de portales web, aplicaciones, redes sociales....

Con el objetivo de aunar a estos dos tipos de usuarios, se ha desarrollado el sistema City Angels. El sistema permite a las personas que necesitan asistencia crear una alerta a los voluntarios cercanos. Ambos usuarios deberán ser inscritos en una red social que permitirá la gestión de estos usuarios. Las alertas serán creadas por los asistidos y

recibidas por los voluntarios mediante una aplicación Android, que precisara de una base de datos y un web service para que pueda gestionar las alertas y los usuarios involucrados. Además, se dispondrá de un panel de control, mostrando las alertas activas y los usuarios involucrados.

El objetivo de este artículo es exponer en que ha consistido el trabajo realizado. Se presentará un breve estudio del estado del arte, los objetivos fijados para el proyecto, así como la metodología, planificación y herramientas usadas para conseguirlos. Finalmente se presentarán los resultados obtenidos y las conclusiones.

Es importante destacar que este proyecto se ha desarrollado como trabajo de fin de grado [1], pero la idea original surgió en la empresa Mass Factory [2] y posteriormente continuará su desarrollo en dicha empresa, por lo que no se podrán proporcionar algunos datos (diagramas, funciones...) debido al carácter privado de este proyecto.

- E-mail de contacto: victormontes_garcia@hotmail.com
- Mención realizada: Ingeniería del Software.
- Trabajo tutorizado por: Jordi Roig de Zárate y Marc Vallribera Ros (Departamento de Microelectrónica y Sistemas Electrónicos)
- Curso 2015/16

2 ESTADO DEL ARTE

Actualmente, la posibilidad de crear una alerta desde un teléfono móvil es algo muy común. La inmensa mayoría de las aplicaciones permiten generar una alerta a los contactos predefinidos y a los servicios de emergencia, así como proporcionar la localización. Entre ellas podríamos poner como ejemplo las aplicaciones S.O.S. Emergencias [3], Alpify [4] o Dr Security [5], pero estas aplicaciones son unidireccionales: solo envían las alertas a los sistemas de emergencias o contactos. Además, en algunos casos como el de Dr Security requieren suscripción mensual. Referente a la gestión de voluntarios tenemos aplicaciones como el sistema de gestión de la Cruz Roja [6] o portales web como Instant Karma [7] o GiveGab [8], pero se basan en acciones sociales que ofrecen diversas ONG. No se basan en dar servicios puntuales a personas, que es lo buscamos con el sistema City Angels.

Con el sistema City Angels se puede ofrecer ayuda específica a personas y ofrecer un servicio para que los voluntarios involucrados puedan ofrecer su ayuda de la manera más eficiente posible. Además, el sistema permite tener un tercer rol de usuario que posibilita la creación de alertas como asistido o recibirlas como voluntario. El objetivo de esta aplicación es que, además de poder notificar a emergencias y contactos, se pueda notificar a los voluntarios disponibles cercanos a la alerta, que estén inscritos en el sistema, de manera que pueden ofrecer su ayuda en situaciones en que la ayuda de los servicios de emergencia sea algo desmesurado, o bien como primera opción antes de llamar a emergencias.

3 OBJETIVOS

Como se ha indicado en secciones anteriores de este documento, el objetivo principal del sistema es ofrecer a los usuarios asistidos que necesiten ayuda (rol asistido) la posibilidad de solicitar ayuda a los usuarios voluntarios (rol voluntario) y/o a los servicios de emergencias, ofreciendo la posibilidad de comunicarnos con los contactos definidos y notificar a estos la creación, el estado y la resolución de la alerta. También debe ofrecer a los usuarios voluntarios la información necesaria sobre la alerta creada (desde la información sobre el asistido, pasando por sus contactos y los detalles de la alerta) y también la información para llegar a la localización de la alerta. Además, se dispondrá de un tercer usuario que podrá realizar las acciones de los dos roles anteriores (usuario asistido-voluntario). Como se ha comentado en la introducción, nuestro sistema está formado por 5 elementos: la red social, la base de datos, el web service, la aplicación Android y el panel de control. A continuación se enumeran los objetivos principales de cada uno de los elementos, los cuales, como se verá en el apartado de resultados, han sido cumplidos.

3.1 Objetivos de la red social

1. La red social debe permitir registrar a los usuarios de la aplicación.
2. La red social debe permitir separar a los usuarios en perfiles (asistido, voluntario, asistido - voluntario).
3. La red social debe permitir crear nuestras propias

preguntas de perfil de usuario, dependiendo del tipo de usuario.

4. La red social debe permitir la creación de grupos de usuarios de emergencias (contactos).

3.2 Objetivos del web service

1. El web service debe proveer la información almacenada en la base de datos a la aplicación del dispositivo móvil.
2. El web service debe actualizar la localización de los voluntarios y asistidos en la base de datos, con la información proporcionada por la aplicación del dispositivo móvil.
3. El web service debe crear, actualizar y cerrar las alertas en la base de datos con la información proporcionada por las aplicaciones.
4. El web service debe buscar a los voluntarios cercanos y notificarles las alertas.
5. El web service debe registrar en la base de datos los eventos importantes ocurridos durante el desarrollo de la alerta.
6. El web service debe notificar la alerta a los contactos de emergencia del asistido que estén registrados en el sistema.

3.3 Objetivos de la aplicación Android (Rol Asistido)

1. La aplicación debe permitir a un asistido poder dar una alerta al sistema para solicitar ayuda a un voluntario.
2. La aplicación debe enviar la localización exacta del asistido al dar una alerta.
3. La aplicación debe permitir avisar a los servicios de emergencias.
4. La aplicación debe dar la opción de llamar a un número de contacto preestablecido en caso de alarma.
5. La aplicación debe mostrar la foto y el nombre del voluntario que asistirá la alerta.
6. La aplicación debe mostrar el tiempo estimado de la llegada del voluntario una vez éste haya confirmado que acudirá a la alerta y su actualización en el tiempo.
7. La aplicación debe permitir especificar la naturaleza de la alerta.
8. La aplicación debe estar adaptada a personas con discapacidad visual.
9. La aplicación debe llamar de forma automática a emergencias o a la persona de contacto en caso de que esta se quede sin voluntarios que la asistan.
10. La aplicación debe permitir hacer sonar una alarma acústica.
11. La aplicación debe permitir que se tomen fotos de las cámaras delantera y trasera de forma automática en el momento de crear la alerta, por si estas pueden servir de ayuda.

3.4 Objetivos de la aplicación Android (Rol Voluntario)

1. La aplicación debe permitir la recepción de notificaciones de alerta.

2. En caso de alerta y notificación de la misma, la aplicación debe mostrar la localización, el tipo de alerta (si ha sido especificada), y un tiempo aproximado de llegada.
3. Cuando la aplicación informe de la alerta, el usuario debe poder elegir aceptarla o no.
4. Si una vez confirmada la asistencia se notifica la naturaleza de la alerta, ésta se actualiza con la nueva información.
5. En caso de alerta, la aplicación debe mostrar la información del asistido, junto con su localización en un mapa.
6. La aplicación mostrará los familiares o personas de contacto del asistido, dando la opción de llamarles directamente.
7. En caso de alerta, la aplicación debe mostrar el tiempo previsto de llegada, así como varias rutas posibles para llegar al asistido tanto a pie como en transporte público o vehículo privado.
8. La aplicación debe mostrar el nombre y la foto del asistido que necesite ayuda.
9. La aplicación debe permitir cerrar las alertas una vez se hayan resuelto o renunciar si el voluntario lo desea.
10. La aplicación debe permitir solicitar la ayuda de un voluntario auxiliar en caso de que se necesite ayuda con la alerta.
11. La aplicación debe permitir llamar al asistido, voluntario auxiliar y a los contactos del asistido
12. La aplicación debe permitir hacer sonar la alarma acústica de forma remota.

La aplicación cambiará de rol en función del tipo de usuario que acceda a la aplicación. En el caso de que el usuario sea asistido-voluntario, la aplicación tendrá un rol de asistido por defecto, pero con la posibilidad de recibir notificaciones de alertas de otros asistidos, en caso de aceptar esta alerta su aplicación pasará a tener un rol de voluntario.

3.5 Objetivos del panel de control

1. El panel de control debe permitir visualizar el registro de todo lo sucedido en el transcurso de una alerta.
2. El panel de control debe permitir localizar a los asistidos y voluntarios en un mapa.
3. El panel de control debe permitir crear nuevas alertas manualmente, así como asignarlas a voluntarios de forma manual.

En cuanto a la base de datos, esta cumplirá la función de almacenar los datos referentes a las alertas y usuarios, así como contener un registro de lo acontecido en las alertas.

4 METODOLOGÍA

Aunque en los planteamientos iniciales se pensó metodologías de desarrollo en cascada [9] y desarrollo incremental [10], debido al proceso de entrevistas con el tutor

finalmente se ha escogido la metodología SCRUM [11] para la realización de este proyecto. Aunque en un principio está orientada a equipos de desarrollo, esta metodología ha sido la más adecuada, ya que las reuniones de control con los tutores sirven como reuniones con el cliente.

Al estar este proyecto englobado en la asignatura “Trabajo de fin de grado” el alumno es el único miembro del equipo de desarrollo, por lo que ha asumido el rol de *equipo* y *SCRUM Master*, mientras que los tutores han asumido el rol de clientes o *Product Owner*.

Como se ha indicado anteriormente, las reuniones con los tutores han servido como reuniones con el cliente en las que se presentaba el grado de desarrollo conseguido de los objetivos establecido en la reunión anterior (*product backlog*), y en las que se genera feedback sobre el trabajo realizado (sugiriendo diversos aspectos a mejorar, indicando posibles errores, sugerencias en el método de desarrollo...) también llamado *Sprint review* y estableciendo los objetivos a desarrollar para la siguiente reunión (*Sprint backlog*), definiendo así los objetivos del *Sprint*, o iteración en el desarrollo del sistema. Ver Fig.1.



Fig. 1. En esta imagen se puede ver un esquema de lo expuesto en el párrafo anterior. Los *Sprints* han representado cada uno de los hitos establecidos en la planificación o, dicho de otra forma, el periodo de desarrollo de cada una de las partes que componen el sistema City Angels.

La gran ventaja de esta metodología es que no solo ha ofrecido un gran *feedback* sobre el trabajo desarrollado, también ha permitido mitigar futuros riesgos por decisiones erróneas u objetivos poco definidos, mejorar el planteamiento inicial de la aplicación añadiendo nuevas funcionalidades e incluso han servido como test de usabilidad, mejorando la comprensión de la aplicación.

5 PLANIFICACIÓN Y EVOLUCIÓN

El desarrollo de este proyecto empezó a mediados de septiembre de 2015 y ha finalizado a finales de enero del 2016. En la Fig.2 se muestra el diagrama de Gantt en la que se muestra la planificación temporal del proyecto organizada en *Sprints*.

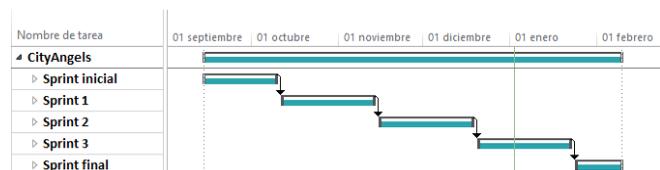


Fig. 2. Diagrama de Gantt en la que se muestra la planificación temporal del proyecto organizada en *Sprints*.

Como se ha mencionado en el apartado anterior, el desarrollo de este proyecto se ha dividido en Sprints, concretamente en cinco, cuyos objetivos pasan a ser desarrollados a continuación.

El objetivo del Sprint inicial era la definición del proyecto. Mediante reuniones con los tutores se definieron los requisitos del sistema con la ayuda de la realización del estudio del estado del arte y se definieron las pantallas de la aplicación. También se decidieron las herramientas que se utilizarían para el desarrollo del proyecto, así como la preparación de las infraestructuras necesarias, como por ejemplo el servidor de aplicaciones y la red social.

En el Sprint 1 se procedió a diseñar la base de datos que usaría el sistema, para a continuación desarrollar el web service, el cual implementa las funcionalidades de nuestro sistema.

En el Sprint 2 se procedió a implementar la aplicación Android. Se implementó las pantallas de la aplicación previamente definidas en el Sprint inicial, pero el principal objetivo fue la implementación de las funciones específicas del rol asistido.

En el Sprint 3 se terminó de implementar la aplicación Android, finalizando de implementar las funcionalidades específicas del rol voluntario. Durante este Sprint también se implementó el panel de control del sistema y se realizaron las pruebas unitarias del sistema.

El Sprint final se centró en la redacción de la documentación de proyecto, así como la preparación de la presentación del proyecto.

Respecto al cumplimiento de la planificación inicial, esta se ha cumplido finalmente con éxito, solo habiendo pequeñas variaciones en los días destinados en algunas funciones debidas que estas eran o bien más complicadas de lo esperado o bien más sencillas.

6 HERRAMIENTAS UTILIZADAS

Un proyecto de estas características ha precisado de diversas herramientas, aunque algunos lectores ya estén familiarizados con algunas de ellas, a continuación se proceden a comentarlas:

6.1 Lenguaje de programación: Java

Java [12], al ser un lenguaje de programación orientado a objetos, permite que el código desarrollado sea ejecutado en cualquier plataforma sin necesidad de compilar de nuevo. Entre sus muchos usos destaca que es el lenguaje que se usa para desarrollar aplicaciones Android, además también dispone de diversas tecnologías para desarrollar servidores de aplicaciones, así como diferentes herramientas para facilitar el trabajo a los desarrolladores. Como se ha podido intuir en lo dicho anteriormente, este lenguaje ha sido clave en este proyecto, ya que se ha utilizado tanto

para desarrollar el web service del sistema como la aplicación Android.

6.2 Lenguaje de programación: MySQL

MySQL [13] es uno de lenguajes de bases de datos más usados debido a su facilidad de uso, simplicidad y rendimiento. Pese a que no dispone de funciones tan avanzadas como otros lenguajes, ha cumplido las funcionalidades requeridas para el proyecto. En este proyecto se ha usado para crear una base de datos relacional, diseño de disparadores y operaciones con la información de la red social.

Para visualizar las tablas y realizar consultas se ha usado el cliente phpMyAdmin [14], que nos permite realizar las acciones anteriormente comentadas, además de gestionar y editar tablas, así como definir disparadores en la base de datos.

6.3 Lenguajes de programación: HTML, PHP y JavaScript

HTML [15] es un lenguaje basado en hipertexto, que permite definir la arquitectura de una página web. En este proyecto se ha usado junto los lenguajes de JavaScript [16] y PHP [17]. JavaScript es un lenguaje interpretado que se suele usar para implementar diferentes funcionalidades en una página web. PHP es un lenguaje ejecutado en servidores web que permite implementar diferentes funcionalidades que otros lenguajes orientados web son más difíciles de realizar.

En este proyecto se han usado estos lenguajes para el desarrollo del panel de control del sistema.

6.4 JSON

JSON [18], al ser un formato de intercambio de datos, el cual puede ser interpretado por la mayoría de lenguajes de programación, se ha usado en este proyecto para enviar y recibir la información del web service, lo cual facilita su interpretación y hace que el web service pueda ser consultado desde diferentes plataformas.

6.5 Entorno de desarrollo: IntelliJ IDEA

IntelliJ IDEA [19] es un entorno de desarrollo integrado (IDE) orientado a la programación en lenguaje JAVA que da muchas facilidades para usar muchas tecnologías diferentes orientadas al desarrollo en JAVA, así como el uso de elementos externos como servidores de aplicaciones en entornos locales o persistencia de datos en bases de datos.

En el caso de este proyecto este IDE ha sido usado para la programación del web service.

6.6 Entorno de desarrollo: Android Studio

Android Studio [20] es un entorno de desarrollo integrado (IDE) orientado a la programación en lenguaje JAVA y especialmente a entornos Android. Incorporando facilidades a los desarrolladores como por ejemplo la gestión del SDK o la incorporación de emuladores, dando facilidades para el desarrollo.

En el caso de este proyecto este IDE ha sido usado para la programación de la aplicación Android.

6.7 Servidor de aplicaciones: WildFly

WildFly [21] es un servidor de aplicaciones basado en las tecnologías Java Enterprise Edition (Java EE), que nos permitirá hospedar y usar el web service. Incorpora las últimas versiones de diversas tecnologías Java, incluidas algunas que se comentaran a continuación como por ejemplo Hibernate o RESTeasy.

6.8 Framework: Hibernate

Hibernate [22] es una tecnología Java orientada al “mapeo” de entidades de bases de datos. Esto permite que las tablas y filas de una base de datos sean usadas como objetos, facilitando la gestión de la base de datos y pudiendo realizar operaciones más complejas que las que se pueden realizar con lenguaje SQL y facilitando la manipulación de datos por parte del código Java.

6.9 Framework: RESTeasy

RESTeasy [23] es una tecnología Java que permite definir funciones Java definidas en un servidor Java EE como web services REST [24], de modo que se facilita la construcción de un web service. Además, permite convertir las entidades JSON que recibe en el cuerpo de las peticiones a clases Java, facilitando las operaciones, también permite enviar directamente una clase Java que pasará a formato JSON automáticamente si se le indica.

6.10 Framework: Retrofit

Retrofit [25] es una tecnología Java, aunque más orientada a ser usada en entornos Android, que permite realizar peticiones REST a un servidor de manera muy sencilla y simple. Al igual que RESTeasy permite convertir las entidades JSON que recibe (y en este caso también envía) en las peticiones a clases Java, facilitando las operaciones.

6.10 Mapbox

Mapbox [26] es un conjunto de APIs y web services que nos permite implementar y diseñar un mapa, ya sea en una aplicación móvil o en una página web. En este proyecto se ha usado tanto en la aplicación Android como en el panel de control no solo para mostrar la localización de usuarios sino para mostrar, también, información diversa como por ejemplo rutas de llegada a los usuarios, las paradas en caso de usar transporte público o en el caso del panel de control mostrar el área de búsqueda y posibles voluntarios candidatos a aceptar la alerta.

6.10 Open Trip Planner

Open Trip Planner [27] es un web service que nos permite calcular la ruta a seguir entre dos puntos geográficos, en este proyecto se ha usado para calcular las rutas de llegada a los asistidos así como saber el itinerario de transporte público y las direcciones asociadas a unas coordenadas.

6.11 Oxwall Social Network

Oxwall [28] es una red social de código abierto implementada en PHP y SQL. Las propiedades de esta aplicación web son la de ofrecer una red social funcional (desde

registro de usuarios, perfiles, mensajes, grupos, chats...) y configurable por parte del administrador. Como es de suponer, se usó esta aplicación para implementar la parte de gestión de usuarios y las funcionalidades de red social.

6.12 Google Cloud Message

Google Cloud Message [29] es un servicio de Google que nos permite realizar notificaciones PUSH [30] que recibirán las aplicaciones móviles, se usa por parte del web service para informar a los usuarios de cambios en las alertas o para notificarles que se ha creado una.

6.13 JustInMind

JustInMind [31] es un programa de diseño orientado a diseñar el aspecto de las pantallas de una aplicación móvil o una página web. También permite definir el comportamiento y la navegación de las pantallas definidas. Se usó para la definición inicial de las pantallas y el comportamiento de la aplicación Android.

6.13 GitHub

GitHub [32] es una plataforma para el desarrollo colaborativo y para el control de versiones de proyectos de software. En este proyecto se ha usado para el control de cambios en los diferentes elementos que componen el sistema (web service, aplicación Android y panel de control).

7 RESULTADOS

A continuación se expondrán los resultados del sistema City Angels. Debido a la diversidad de este proyecto, se procederá a explicar los resultados de cada una de las partes que lo componen:

7.1 Red Social

Como se ha comentado en el apartado de herramientas, para la implementación de la red social se ha usado la aplicación Oxwall, que ya implementa todas las funcionalidades típicas de una red social, entre las que destaca el registro de usuarios. A continuación se muestra su pantalla principal en la Fig.3



Fig. 3. Pantalla principal de la red social del sistema City Angels.

En este caso era necesario la gestión de diversos perfiles de usuario (asistido, voluntario y asistido-voluntario).

Gracias a las opciones de configuración de Oxwall no solo se ha conseguido definir los perfiles de usuario, sino que también se ha conseguido implementar diferentes preguntas de registro en función del perfil. Esto permite, por ejemplo, que en el caso de los usuarios asistidos se pregunte por la persona de contacto o en el caso de los voluntarios, su disponibilidad horaria para asistir a alertas, experiencia médica o si dispone de vehículo para llegar a ellas (en el caso de los usuarios asistido-voluntario tendrán las preguntas de perfil de ambos). También se ha conseguido que en caso de que un usuario asistido o asistido-voluntario se registre en la red social, se cree automáticamente un grupo de contactos de emergencia, donde tendrán que registrarse los usuarios que deseen ser contactos de emergencia del usuario asistido / asistido-voluntario.

7.2 Web service

En el caso del web service no solo se ha conseguido cumplir los objetivos marcados, sino que se ha podido desarrollar funcionalidades nuevas. El objetivo principal era la gestión de alertas, este objetivo se ha conseguido con creces, no solo permite crear alertas dadas por los usuarios asistidos sino que además permite asignar voluntarios a la alerta (máximo 2 voluntarios), o eliminarlos de la alerta en caso de que el voluntario haya indicado que no asistirá a la alerta, actualizarla (nueva localización, tipo de alerta), y resolver estas alertas por parte de los actores implicados. También se ha gestionado todo el tema de las fotos tomadas en el momento de crear la alerta, subiéndolas al servidor, mostrándolas solo a los voluntarios asignados y eliminándolas después de que se haya resuelto la alerta.

También se ha conseguido implementar las funciones de búsqueda de voluntarios cuando se crea una alerta (actualmente se busca en 2 Km a la redonda del lugar donde se ha declarado la alerta), la búsqueda de voluntarios auxiliares por parte de los voluntarios que han aceptado una alerta (por proximidad, por experiencia médica o por vehículo), así como las notificaciones de alerta y cambio de estado (que recibirán en sus aplicaciones Android) a todos los implicados, desde voluntarios en el radio de búsqueda a contactos del asistido (que recibirán una alerta específica).

Para el correcto funcionamiento del sistema, el web service mantiene actualizada la localización de los usuarios con la información recibida de la aplicación Android. También registra todo lo sucedido en la alerta (si ha sido creada, si ha sido aceptada por un voluntario, si ha sido resuelta...).

Por último, se ha conseguido implementar una función de cálculo de rutas que proporciona a los voluntarios la ruta más óptima para llegar al asistido, esta ruta puede ser a pie, en vehículo propio o en transporte público, indicando si son precisas las paradas de transporte público, así como la línea de transporte que se debe usar. De cara a los usuarios asistidos se les indicará el tiempo aproximado de llegada de los voluntarios a su localización.

El web service está diseñado para que en caso de recepción y envío de información solo envíe respuestas de servidor o datos en formato JSON, esto hace que el web

service no tenga restricciones de plataformas, puede ser usado tanto por aplicaciones móviles como por páginas web.

7.3 Base de datos

Aunque la base de datos necesaria para el funcionamiento de la red social ya se crea automáticamente al instalar la aplicación Oxwall, se han diseñado todas las tablas y relaciones pertenecientes a la gestión de alertas y datos de usuarios necesarios para la gestión de alertas. En este diseño se ha tenido en cuenta que no haya información redundante y se almacene solo la información necesaria para el correcto funcionamiento del sistema.

7.4 Aplicación Android (Rol Asistido)

El principal objetivo de este rol de la aplicación se basaba en la simplicidad de uso, ya que no solo está enfocada a personas con necesidades especiales (niños, ancianos, discapacitados...) si no que estas personas, en el momento de crear una alerta pueden encontrarse en una situación de estrés. Además, esta parte también está enfocada a personas con discapacidad visual, por lo que el número de elementos mostrados se mantiene en los mínimos indispensables y en el orden adecuado para su acceso en forma secuencial, así como adaptando el comportamiento de la aplicación si es necesario.

Para conseguir todo esto se ha diseñado la aplicación para que solo tenga 2 pantallas: la pantalla principal (Fig.4.A) será la destinada a crear la alerta, que solo tendrá 2 botones, uno para crear una alerta en nuestro sistema (avisar voluntarios) y otro para llamar a los servicios de emergencias. La segunda es una pantalla auxiliar (Fig.4.B) de contactos donde se dispone de una lista de los contactos de emergencia facilitados en la red social, así como la posibilidad de hacer sonar la alarma acústica y llamar a los servicios de emergencias.

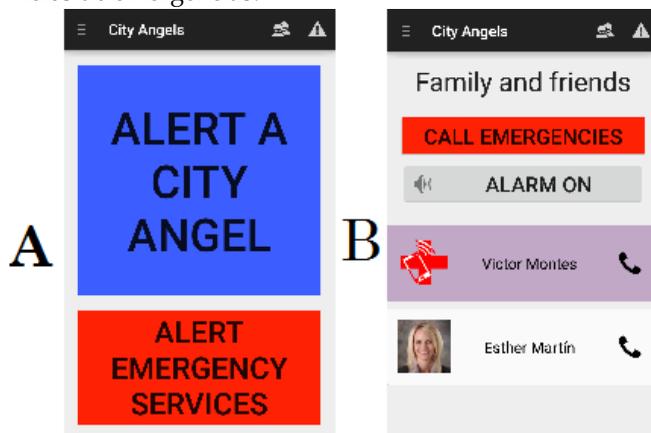


Fig. 4. A) Pantalla principal. B) Pantalla de contactos (se remarca la persona de contacto facilitada en el registro).

En caso de que el usuario cree una alerta en nuestro sistema, mientras se buscan usuarios voluntarios, se muestra la pantalla de contactos de modo que siempre hay la opción de comunicarse todo momento.

Si un voluntario o voluntarios han aceptado la alerta, la pantalla de crear alerta (pantalla principal -Fig.4.A-) es

sustituida por la pantalla de alerta aceptada (Fig. 5.A), de ese modo siempre se dispone de solo 2 pantallas principales.

Como se puede ver en la Fig.5.A al asistido se le mostrará la foto y el nombre del voluntario, con el objetivo de facilitar la identificación, además podrá ver el tiempo estimado de llegada a su localización, el cual se irá actualizando en función de la localización y el tipo de desplazamiento del voluntario (a pie, corriendo o en vehículo). También tendrá la posibilidad de indicar la naturaleza de la alerta y activar la alarma acústica, además continúa teniendo la posibilidad de acceder a la pantalla de contactos para poder llamar a sus contactos o a los servicios de emergencia. Todas estas funcionalidades se mantienen también cuando hay 2 voluntarios que hayan aceptado la alerta, como se muestra en la Fig.5.B.

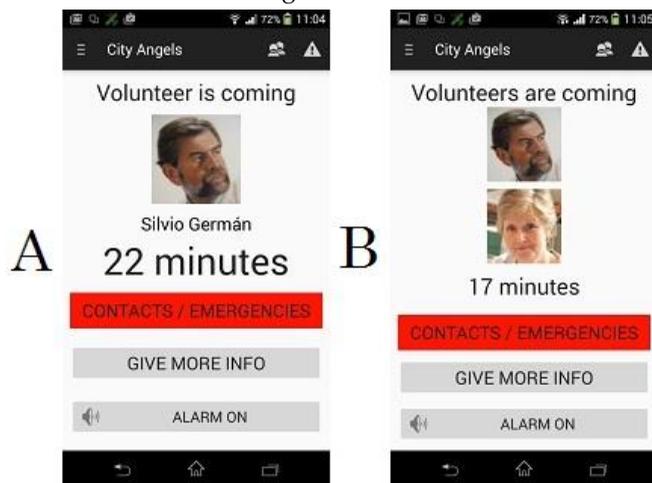


Fig. 5. A) Pantalla de alerta aceptada por un solo voluntario. B) Pantalla de alerta aceptada por dos voluntarios.

Todas las opciones antes descritas son totalmente funcionales, además de algunas no mencionadas como la actualización de la localización del asistido (tanto a voluntarios como a contactos), como la de tomar fotos de las cámaras delantera y trasera en el momento de crear la alerta y llamar a emergencias automáticamente o a la persona de contacto si no se encuentran voluntarios, o volver a crear la alerta si los voluntarios implicados renuncian, por lo cual se han cumplido todos los objetivos principales previstos.

En el apéndice 1 se expone el caso de uso del desarrollo de una alerta.

7.5 Aplicación Android (Rol Voluntario)

El principal objetivo de este rol de la aplicación es la de facilitar al voluntario los datos necesarios para gestionar la alerta, relacionar los actores involucrados en ella y facilitar la información necesaria para llegar hasta al asistido.

El voluntario deberá aceptar la alerta antes de asignarse a ella, por lo cual se ha desarrollado una pantalla de notificación de alerta (Fig.6.A) que permite al voluntario saber la persona que necesita ayuda, el tipo de emergencia si ha sido indicado, la localización de la alerta y su tiempo de llegada al lugar del asistido (en coche, transporte público y a pie).

Una vez aceptada la alerta al voluntario se le mostrará la foto y nombre del asistido para facilitar su identificación, podrá llamarle a él y a sus contactos, además de poder llamar si es necesario a los servicios de emergencia. Por otra parte podrá activar remotamente la alarma acústica del asistido para facilitar su localización (algunas de estas opciones se pueden ver en la pantalla principal de la alerta - Fig.6.B-).

También permite buscar un voluntario auxiliar si no hay ninguno asignado, buscando por cercanía, por conocimientos médicos y si dispone de vehículo. En caso que ya haya un voluntario auxiliar se podrá consultar su perfil que da la opción a llamarle si es preciso.

También dispondrá de la posibilidad de ver la localización del asistido en un mapa, así como calcular la ruta más óptima para desplazarse a su localización. Además, podrá resolver la alerta o darse de baja de su asistencia si así lo desea (si todos los voluntarios implicados renuncian a la alerta esta será creada de nuevo).

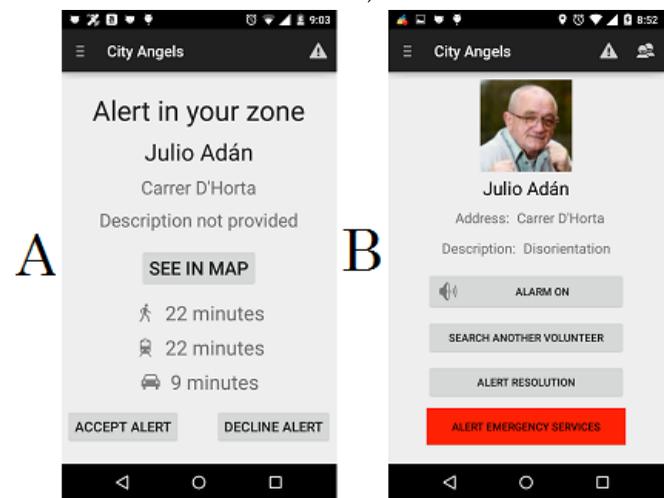


Fig. 6. A) Pantalla de notificación de alerta. B) Pantalla principal en el momento que el voluntario ha aceptado la alerta.

Al igual que en la parte del asistido todas las funcionalidades cumplen con los objetivos principales. Como se ha mencionado en el apartado anterior, en el apéndice 1 se describe un caso de uso que muestra el desarrollo de una alerta.

7.6 Aplicación Android (Alerta a contactos)

Con el objetivo de notificar a los contactos de un asistido que este se encuentra en una alerta se ha desarrollado la funcionalidad de informar a los contactos (de momento solo a los contactos que estén en un rol asistido-voluntario o voluntario -Fig.7.A-), así como la posibilidad de mostrar la localización del asistido (Fig.7.B) y ver los voluntarios que están involucrados en la alerta, así como darles la opción de llamar directamente a los involucrados en la alerta.

Hay que destacar que los usuarios tienen la opción de configurar la aplicación para que solo reciban notificaciones de alerta de sus contactos.

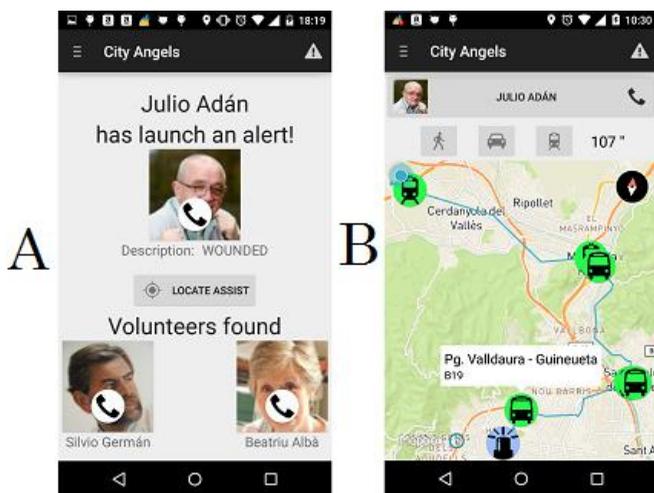


Fig. 7. A) Pantalla de notificación de alerta a un contacto. B) Pantalla de ruta hacia el asistido / contacto (misma pantalla que la aplicación de voluntario).

7.7 Panel de control

Aunque durante el desarrollo del proyecto no se podía asegurar el desarrollo de esta parte, no solo se ha implementado con sus funcionalidades previstas inicialmente, sino que además se han complementado con más funcionalidades.

En el panel de control, y tal como se puede ver en la Fig.8, se puede ver en un mapa no solo las alertas y los actores involucrados, también podemos ver, en caso de que una alerta no tenga todavía todos los voluntarios asignados posibles, el área de búsqueda y los voluntarios candidatos a aceptar la alerta.

Además del mapa se muestra una lista de las alertas activas donde tendremos opción de centrar en el mapa la alerta y los voluntarios implicados, así como la posibilidad de resolver la alerta por el operador del panel de control.

También se ha implementado la posibilidad de visualizar un histórico de los sucesos de la alerta, ver los perfiles de usuario, y finalmente la posibilidad de crear una alerta desde el panel de control.

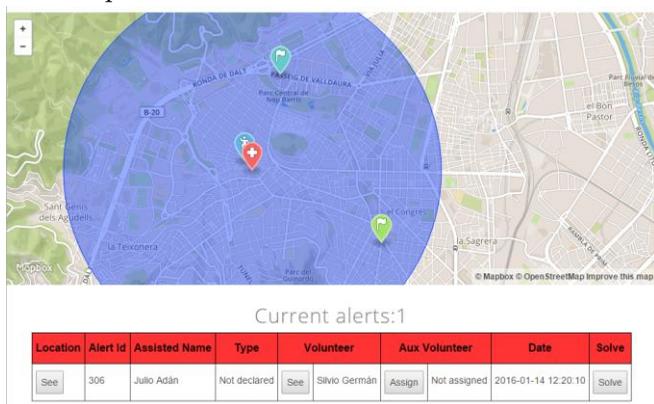


Fig. 8. Imagen del panel de control con el mapa y la lista de alertas activas.

8 CONCLUSIONES

Como se ha podido ver a lo largo de este documento, este proyecto ha cumplido todos los objetivos marcados,

además de añadir nuevas funcionalidades que originariamente no estaban previstas. Durante el desarrollo se ha cumplido la planificación sin necesidad de redefinir las tareas originalmente planificadas.

El sistema City Angels permitirá que las personas con necesidades especiales puedan solicitar ayuda siempre que la necesiten, tanto a la plataforma de voluntarios como a los servicios de emergencia. Además, permitirá notificar a sus familiares y amigos de la situación de emergencia en la que se encuentran. También ofrecerá a las personas que desean ayudar una plataforma en la que puedan hacerlo con facilidad, ofreciéndoles un sistema sencillo que les proporcionará la información que necesiten para cumplir sus objetivos.

Aunque este sistema puede funcionar como un sistema independiente, también puede ser usado por diversas organizaciones y empresas para facilitar la coordinación entre las personas a las que dan sus servicios de ayuda y asistencia, y sus voluntarios / empleados.

Como mejoras y futuros desarrollos se propone la implementación de funcionalidades secundarias que se han detectado durante el desarrollo del presente proyecto, tales como, en el caso del web service, implementar una funcionalidad que permita crear alertas que no estén enfocadas a un individuo sino a una zona, que permitirá dar servicio a múltiples personas (por ejemplo, un accidente grave que involucre a varias personas o un incendio, sin limitación de 2 voluntarios).

En el caso de la aplicación Android en un rol asistido, el uso de wearables para detectar ciertos tipos de alerta, como por ejemplo enfermedades cardíacas, además de facilitar el lanzamiento de alertas mediante widgets [33]. También cabe la posibilidad de implementar la funcionalidad de mostrar hospitales y comisarias en el mapa, así como ofrecer manuales y protocolos de actuación en la aplicación Android del rol voluntario.

También es posible mejorar en aspectos como enfocar la red social Oxwall no solo como una plataforma de registro sino como una plataforma totalmente orientada a la función del sistema (ofrecer incentivación a los usuarios voluntarios, ranking de usuarios que más alertas solucionan, registros de alertas solventadas, ofrecer cursos y manuales de actuación) y ofrecer un panel de control a los usuarios para poder saber el estado de sus contactos, sin posibilidad de crear o asignar alertas.

AGRADECIMIENTOS

Quisiera expresar mi más sincero agradecimiento a los tutores de este proyecto, el doctor Jordi Roig y Marc Vallribera. Gracias a su ayuda y consejos se ha podido cumplir con este proyecto y dar un resultado mucho mejor del que se esperaba inicialmente. También agradecer al personal de la empresa Mass Factory: Daniel del Arco, Daniel González y Jordi López que han ayudado a solventar dudas en el aspecto técnico. También una mención especial para el doctor Miquel Domenech (Departament de Psicologia so-

cial. UAB), que ayudó con el diseño de las diferentes pantallas, así como a definir el comportamiento de la aplicación Android.

Finalmente, me gustaría agradecer a mi familia y amigos su apoyo durante la realización de este proyecto.

REFERENCIAS

- [1] Universitat Autònoma de Barcelona, "Treball de fi de grau: Grau d'enginyeria informàtica", 2014. [Online]. Disponible en: <http://www.uab.cat/web/estudiar/llistat-de-graus/pla-d-estudis/treball-de-fi-de-grau/enginyeria-informatica-1345468416862.html?param1=1263367146646>. [Accedido: 15/01/2016].
- [2] Mass Factory. , "MASS FACTORY URBAN ACCESS MOBILITY, S.L.",2015. [Online].Disponible en: <http://www.uab.cat/servelet/BlobServer?blobtable=Document&blobcol=urlidocument&blobheader=application/pdf&blobkey=id&blobwhere=1345682723142&blobnocache=true>. [Accedido el 15/01/16]
- [3] S.O.S. Emergencias. , "Prestaciones de la aplicación",2015. [Online].Disponible en: <http://www.emergenciasos.com>. [Accedido el 15/01/16]
- [4] Alpify. , "Prestaciones de la aplicación",2015. [Online].Disponible en: <http://www.alpify.com/es/>. [Accedido el 15/01/16]
- [5] Dr security. , "Prestaciones de la aplicación",2015. [Online].Disponible en: <http://drsecurityapp.com/es/>. [Accedido el 15/01/16]
- [6] Red Cross Mobile Apps. , "Aplicaciones disponibles con sus descripciones",2015. [Online].Disponible en: <http://www.redcross.org/prepare/mobile-apps>. [Accedido el 15/01/16]
- [7] Instant Karma. , "Descripción de la plataforma",2015. [Online].Disponible en: <http://instant-karma.co/es/pages/about>. [Accedido el 15/01/16]
- [8] GiveGab. , "Descripción de la plataforma",2015. [Online].Disponible en: https://www.givegab.com/about/product_tour. [Accedido el 15/01/16]
- [9] Wikipedia, "Desarrollo en cascada", 2015. [Online].Disponible en: https://es.wikipedia.org/wiki/Desarrollo_en_cascada. [Accedido el 15/01/16]
- [10] ProyectosAgiles.org, "Desarrollo iterativo e incremental", 2015. [Online].Disponible en: <http://proyectosagiles.org/desarrollo-iterativo-incremental/>. [Accedido el 15/01/16]
- [11] Manuel Trigas Gallego, "Gestión de proyectos informáticos. Metodología Scrum", 2010. [PDF].Disponible en: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>. [Accedido el 15/01/16]
- [12] Java. , "¿Qué es la tecnología Java y para qué la necesito?",2015. [Online].Disponible en: https://www.java.com/es/download/faq/whatis_java.xml. [Accedido el 15/01/16].
- [13] Wikipedia. , "Definición de MySQL",2015. [Online].Disponible en: <https://es.wikipedia.org/wiki/MySQL>. [Accedido el 15/01/16].
- [14] phpMyAdmin. , "Definición de la plataforma",2015. [Online].Disponible en: <https://www.phpmyadmin.net/>. [Accedido el 15/01/16].
- [15] W3Schools, "HTML Tutorial", 2015. [Online]. Disponible en: <http://www.w3schools.com/html/>. [Accedido el 15/01/16].
- [16] W3Schools, "JavaScript Tutorial", 2015. [Online]. Disponible en: <http://www.w3schools.com/js/default.asp>. [Accedido el 15/01/16].
- [17] The PHP Group, "PHP Manual", 2015. [Online]. Disponible en: <https://secure.php.net/manual/en/index.php>. [Accedido el 15/01/16].
- [18] JSON. , "Introducción a JSON",2015. [Online].Disponible en: <http://www.json.org/json-es.html>. [Accedido el 15/01/16].
- [19] IntelliJ. , "Funcionalidades del programa",2015. [Online].Disponible en: <https://www.jetbrains.com/idea/>. [Accedido el 15/01/16].
- [20] Android Studio. , "Funcionalidades del programa",2015. [Online].Disponible en: <http://developer.android.com/intl/es/tools/studio/studio-features.html>. [Accedido el 15/01/16].
- [21] WildFly. , "Descripción del servidor y de sus funcionalidades",2015. [Online].Disponible en: <http://www.oxwall.org/>. [Accedido el 15/01/16]
- [22] Hibernate. , "Descripción del framework",2015. [Online].Disponible en: <http://hibernate.org/orm>. [Accedido el 15/01/16]
- [23] RESTeasy. , "Descripción del framework",2015. [Online].Disponible en: <http://resteasy.jboss.org>. [Accedido el 15/01/16]
- [24] Enrique Amodeo, "Servicios web ¿Que es REST?", 2010. [Online].Disponible en: <https://eamodeorubio.wordpress.com/2010/07/26/servicios-web-2-%C2%BFques-rest/>. [Accedido el 15/01/16]
- [25] Retrofit. , "A type-safe HTTP client for Android and Java",2015. [Online].Disponible en: <http://square.github.io/retrofit/>. [Accedido el 15/01/16].
- [26] Mapbox. , "Descripción y funcionalidades del sistema",2015. [Online].Disponible en: <https://www.mapbox.com/>. [Accedido el 15/01/16].
- [27] Open Trip Planner. , "Descripción y documentación del sistema",2015. [Online].Disponible en: <http://docs.opentripplanner.org/en/latest/>. [Accedido el 15/01/16]
- [28] Oxwall. , "Descripción del sistema",2015. [Online].Disponible en: <http://www.oxwall.org/>. [Accedido el 15/01/16]
- [29] Google. , "Cloud Messaging",2015. [Online].Disponible en: <https://developers.google.com/cloud-messaging/>. [Accedido el 15/01/16].
- [30] Margaret Rouse. , "Push notification definition",2014. [Online].Disponible en: <https://developers.google.com/cloud-messaging/>. [Accedido el 15/01/16].
- [31] JustInMind. , "Funcionalidades del programa",2015. [Online].Disponible en: <http://www.justinmind.com/overview>. [Accedido el 15/01/16].
- [32] GitHub. , "Definición y propiedades de la plataforma",2015. [Online].Disponible en: <https://github.com/features>. [Accedido el 15/01/16].
- [33] Wikipedia, "Definición de Widjet", 2015. [Online].Disponible en: https://www.givegab.com/about/product_tour. [Accedido el 15/01/16]

APÉNDICE

A.1 Caso de uso

Para simular un escenario, utilizaremos un usuario asistido llamado Julio Adán y un voluntario llamado Silvio Germán para ejemplificar la creación, y posterior aceptación de una alerta. Para mostrarlo mejor crearemos la alerta en la calle Horta de Barcelona, y ubicaremos en la calle Maragall al voluntario.

Como se ha indicado en el apartado de resultados y más concretamente en la subsección de aplicación Android (rol asistido) del artículo City Angels, el asistido podrá crear una alerta a través de la pantalla principal (Fig.1.A), pulsando el botón “Alert a City Angel”. También puede llamar a emergencias pulsando el botón “Alert emergency services”. Una vez creada la alerta se abrirá la pantalla de contactos mientras el sistema busca voluntarios (Fig.1.B), si desea llamar a los contactos solo tiene que pulsarlos en lista.

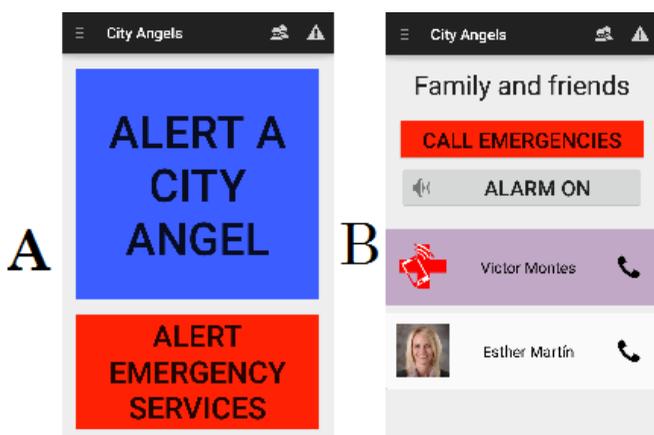


Fig. 1. A) Pantalla principal. B) Pantalla de contactos (se remarca la persona de contacto facilitada en el registro).

En la Fig.2 se muestra el mapa del panel de control para facilitar su comprensión en el entorno geográfico donde se desarrolla la alerta. En el centro de la imagen y remarcado en rojo la localización de la alerta, a su lado y en azul claro la localización del asistido, en gris la localización de los voluntarios que están en la zona de alerta (marcada en azul oscuro).

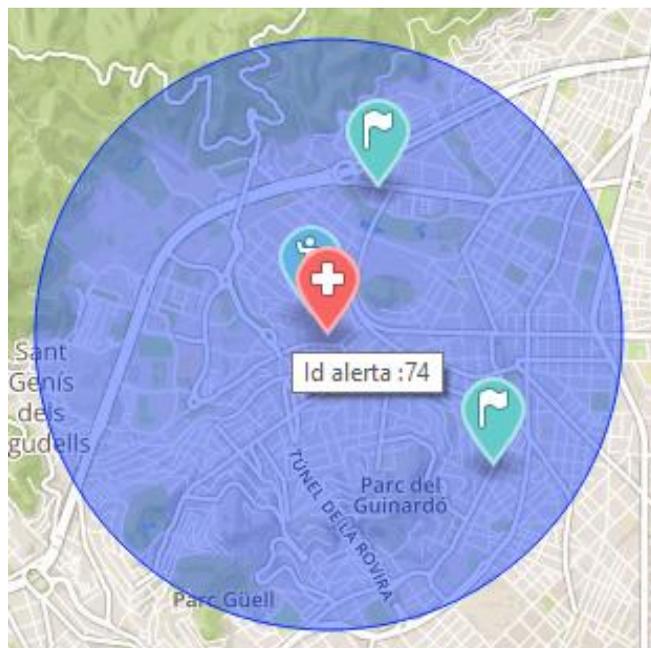


Fig. 2. Mapa de la situación de la alerta sin usuarios asignados

Como se comentó en el artículo City Angels, en la subsección de aplicación Android (rol voluntario) de la sección resultados, el voluntario recibirá una notificación indicándole una alerta en su zona próxima (Fig.3.A), en la que constará el nombre, la dirección, la naturaleza de la alerta (si ha sido indicada por el asistido) y el tiempo de llegada estimado a la zona de alerta a pie, coche y transporte público, también podrá consultarla en el mapa al pulsar el botón “See in map” (Fig.6.B). Para declinar la alerta tendrá que pulsar el botón “Decline alert” y en este caso, para aceptarla, el botón “Accept alert”.

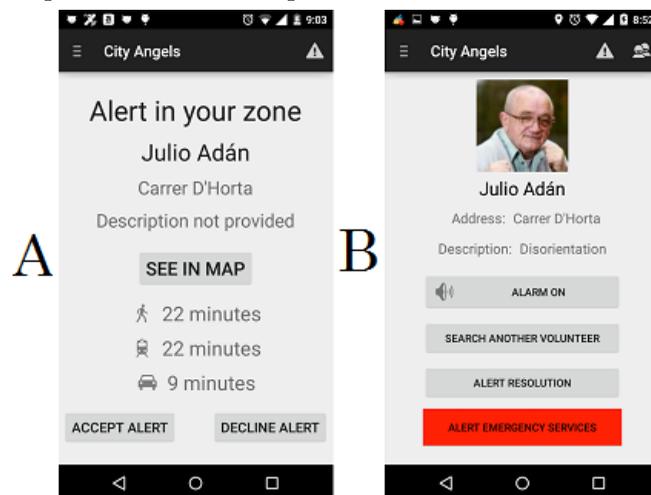


Fig. 3. A) Pantalla de notificación de alerta. B) Pantalla principal en el momento que el voluntario ha aceptado la alerta.

Para mostrar la aceptación de la alerta en el entorno geográfico, en la Fig.4 se muestra como el voluntario ha aceptado la alerta (pasando a color verde su indicador, señalando que está involucrado en la alerta).

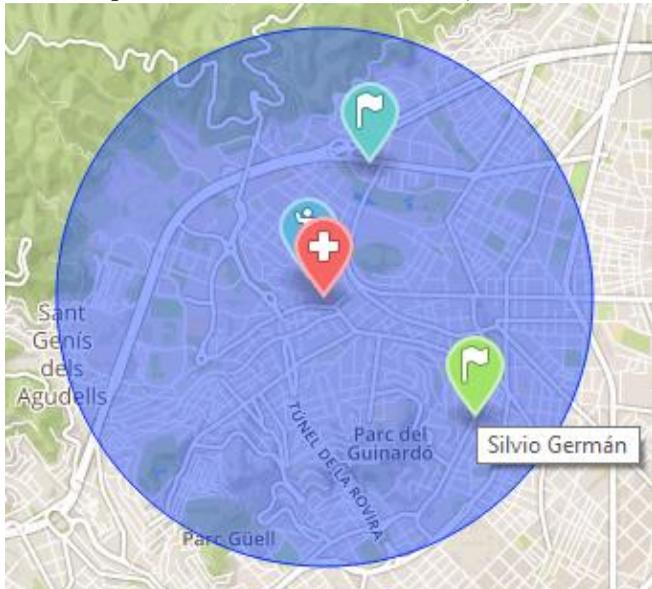


Fig. 4. Mapa de la situación de la alerta con un usuario voluntario asignado.

En el caso del usuario asistido, su pantalla principal pasará a ser la pantalla de alerta aceptada. En este caso y como se puede ver en la Fig.5.A, usamos el anteriormente citado voluntario Silvio Germán (también se expone como se muestra la pantalla principal con 2 voluntarios Fig. 5.B).

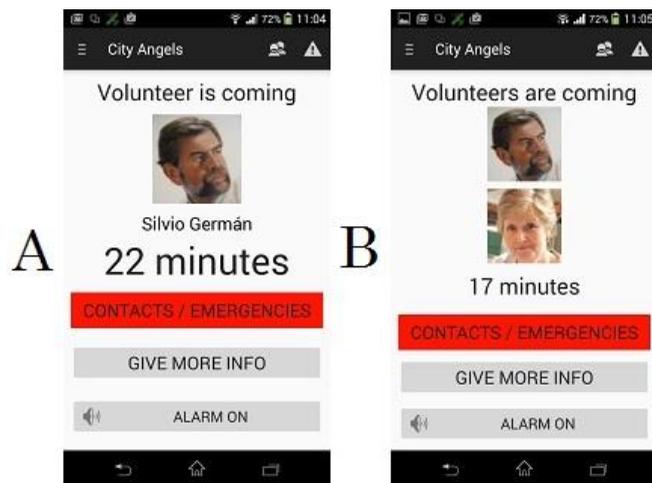


Fig. 5. A) Pantalla de alerta aceptada por un solo voluntario. B) Pantalla de alerta aceptada por dos voluntarios.

La pantalla de alerta aceptada muestra un botón con la foto del voluntario (botón que permite acceder al perfil del voluntario donde se verá la localización del voluntario y se dispondrá la opción de llamarle, en caso de que este activa la accesibilidad para discapacitados visuales, hará directamente la llamada), el nombre del voluntario y el tiempo estimado de llegada a su localización (tiempo que se irá actualizando en función de la localización y el tipo de desplazamiento del voluntario). El usuario también dispone

de la opción de actualizar la alerta, activar la alarma acústica o llamar a los servicios de emergencias y contactos pulsando los botones correspondientes.

En el caso del voluntario se muestra la pantalla principal (Fig.3.B) donde se visualiza el nombre del asistido, así como un botón-foto con la imagen del asistido con el objetivo de identificarlo (como se puede ver corresponde con el asistido Julio Adán del caso de uso), al pulsar este botón visualizará el perfil del asistido, que además permite acceder a las funciones de llamarlo, de localizarlo en el mapa, así como ver las fotos tomadas en el momento de la alerta.

En la pantalla de mapa (Fig.7.B) se puede ver en el mapa la localización del asistido, la nuestra y la del voluntario auxiliar si existe, también se pueden generar rutas para llegar al usuario en coche, transporte público (indicando paradas y líneas de transporte) y a pie, indicando además de la ruta el tiempo estimado de llegada.

Continuando con la descripción de la pantalla principal (Fig.3.B), se presentará la dirección de la alerta, que se actualiza en función de la localización del asistido, la opción de activar remotamente la alarma acústica (que sonará en el móvil del asistido ayudando a su localización) y la opción de llamar a los servicios de emergencias.

También permite buscar un voluntario auxiliar si no hay ninguno asignado (si es necesario), buscando por cercanía, por conocimientos médicos y si dispone de vehículo propio (Fig.6.A). En caso que un voluntario auxiliar acepte se podrá ver su perfil (Fig.6.B), que indica nombre, foto y conocimientos médicos, además de dar la opción a llamarle si es preciso (no se utilizará en este caso de uso).

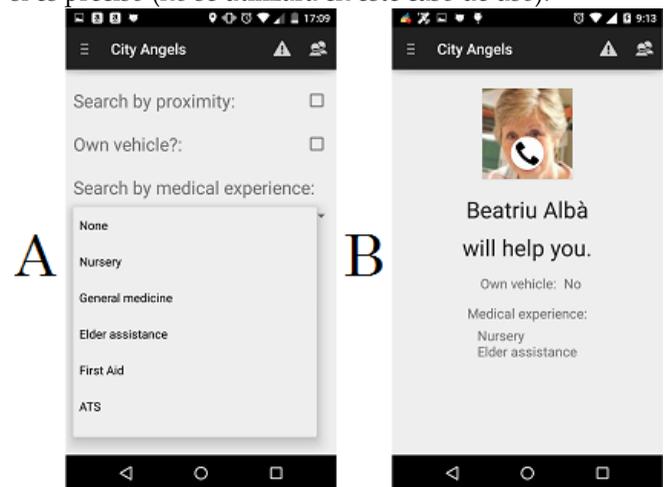


Fig. 6. A) Pantalla búsqueda de voluntarios auxiliares. B) Pantalla de perfil del voluntario auxiliar.

También se puede acceder a los contactos del asistido (Fig.1.B) para llamarlos si es preciso.

También se dispone de la opción de resolver la alerta, el cual no solo permite indicar como se ha resuelto, sino también permite que el voluntario se desligue de la alerta si hace falta.

Todo el desarrollo de la alerta, desde su creación, acep-

tación por el voluntario y posterior resolución es comunicada a los contactos del asistido mediante la notificación de la alerta de contacto (Fig.7.A), donde además de llamar a los involucrados en la alerta, se podrá ver la naturaleza de esta si el asistido la ha proporcionado y ver la localización de los actores involucrados en el mapa al pulsar el botón "Locate Assist" (Fig.7.B).

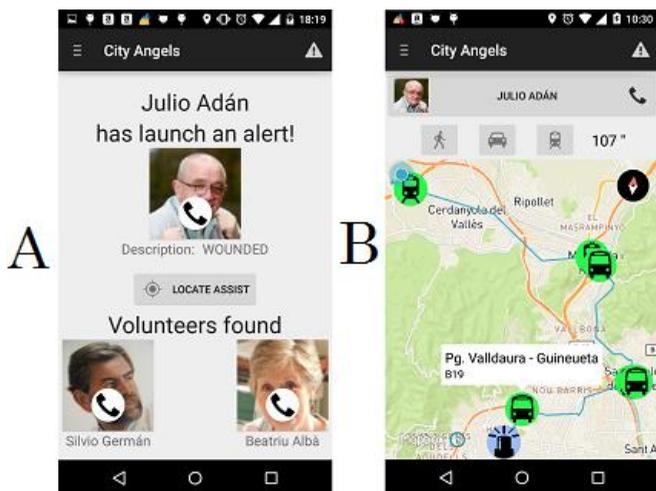


Fig.7. A) Pantalla de notificación de alerta a un contacto (se muestra con dos voluntarios para ejemplificar una situación con todos los actores). B) Pantalla de ruta hacia el asistido / contacto (misma pantalla que la aplicación de voluntario).

Una vez aceptada la alerta, el voluntario deberá desplazarse a la zona de la alerta para dar ayuda al asistido. En la Fig.8 mostramos como el voluntario ya ha llegado al asistido.



Fig. 8. Mapa de la situación de la alerta con el voluntario llegando a la zona de asistido / alerta.

Una vez que el voluntario ha ayudado al asistido, el voluntario resuelve la alerta. A continuación, en la Fig.9.A se muestra la pantalla de resolución y en la Fig.9.B se muestra el resumen de la resolución, resumen que será visto por todos los implicados en la alerta. También hay que indicar que el asistido puede cancelar la alerta siempre que lo desee, acción que se notificará al voluntario y a los contactos.

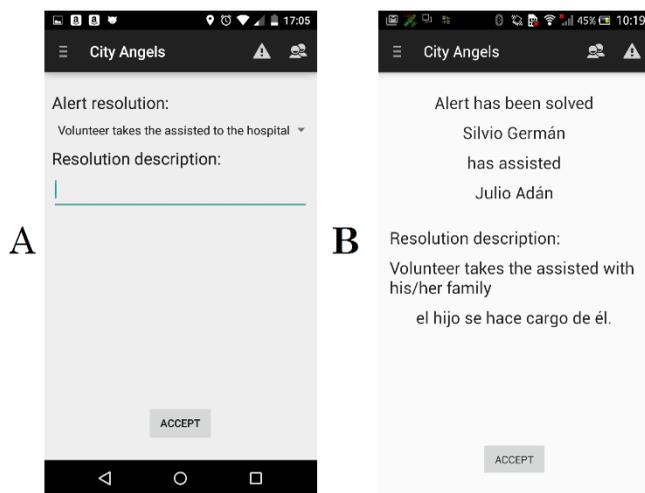


Fig.9. A) Pantalla de resolución de alerta. B) Pantalla de resumen de la resolución de alerta.