

**Universidad Internacional de La Rioja
Máster Universitario en Ingeniería de Software y
Sistemas Informáticos**

Aplicación para combatir la soledad entre personas de más de 60 años

Trabajo Fin de Máster

Presentado por: García Sánchez, Andrés

Director/a: Sánchez Rubio, Manuel

Ciudad: Sevilla
Fecha: 15/09/2017

Resumen

Las personas mayores de 60 años hacen un uso cada vez mayor y más amplio de las funciones de los móviles de nueva generación o *smartphones*, lo que abre la posibilidad de cubrir necesidades específicas de esta población a través de aplicaciones para dispositivos móviles y, en concreto, fomentar el envejecimiento activo y la prevención del aislamiento social. Con esa finalidad desarrollamos una aplicación distribuida que permite poner en contacto a personas de esa franja de edad entre sí con el objetivo de que encuentren compañía y relaciones amistosas.

En su desarrollo seguimos un proceso de software iterativo e incremental, con un enfoque de Análisis y Diseño Orientado a Objetos y la incorporación de buenas prácticas de seguridad en el ciclo de vida de desarrollo de software (S-SDLC). De esta forma, exploramos el uso de estas tecnologías con fines sociales.

Palabras Clave: Aplicación móvil, servicio web, envejecimiento activo, citas online

Abstract

People over the age of 60 make increasing use of smartphone functions, which opens up the possibility of meeting specific needs of this population through applications for mobile devices and, in particular, promoting active ageing and prevention of social isolation. For this purpose we developed a distributed application that allows people in that age group to be contacted with each other in order to find company and friendly relationships.

In its development, we followed an iterative and incremental software process, with a focus on Object Oriented Analysis and Design and the incorporation of good security practices in the Software Development Life Cycle (S-SDLC). In this way, we explore the use of these technologies for social purposes.

Keywords: Mobile application, web service, active ageing, online dating

Índice de contenido

1.	Introducción	8
1.1.	Justificación	9
1.1.1.	Aclaraciones previas sobre el término “mayores”	9
1.1.2.	Soledad en los mayores.....	10
1.1.3.	Envejecimiento activo	11
1.1.4.	Mayores y uso de las TIC.....	13
1.2.	Enfoque del concepto de la aplicación	18
2.	Contexto tecnológico.....	21
2.1.	Aplicaciones o servicios de contactos	21
2.2.	Destinadas a personas mayores	23
2.3.	Tecnologías	24
2.3.1.	Lado servidor web.....	25
2.3.2.	Lado cliente – app móvil.....	28
3.	Objetivos y metodología.....	30
3.1.	Enfoque metodológico. Modelo de proceso de software	30
3.2.	Protección ante riesgos de seguridad	31
3.3.	Objetivos	33
4.	Desarrollo de la aplicación	35
4.1.	Especificaciones	35
4.1.1.	Requisitos	36
4.1.2.	Modelo del sistema de información y arquitectura.....	39
4.2.	Análisis y diseño del sistema.....	41
4.2.1.	Análisis del sistema.....	42
4.2.1.1.	Descripción del sistema	42
4.2.1.2.	Casos de uso	48
4.2.1.3.	Clases de análisis	57

4.2.2.	Diseño del sistema.....	60
4.2.2.1.	Modelo de clases de diseño.....	60
4.2.2.2.	Modelos de interacción	64
4.2.2.3.	Componentes del sistema.....	68
4.2.2.4.	Modelo lógico y físico de datos	69
4.2.2.5.	Interfaz de usuario	73
4.3.	Implantación del sistema.....	76
4.3.1.	Cliente – aplicación para Android.....	78
4.3.2.	Servidor – aplicación en el servidor.....	79
4.3.2.1.	Capa de aplicación.....	79
4.3.2.2.	Capa de persistencia	80
4.3.3.	Pruebas unitarias y funcionales.....	81
5.	Despliegue y validación.....	82
5.1.	Despliegue en entorno de producción	82
5.2.	Test de penetración	84
5.3.	Pruebas de usuario	84
6.	Conclusiones.....	86
7.	Referencias bibliográficas	88
8.	Anexos	92
8.1.	Artículo.....	92
8.1.1.	Introducción	92
8.1.2.	Contexto tecnológico.....	93
8.1.3.	Objetivos y metodología.....	95
8.1.4.	Desarrollo de la aplicación	96
8.1.5.	Despliegue y validación.....	99
8.1.6.	Conclusiones	100

Índice de ilustraciones

Figura 1:	Los determinantes del envejecimiento activo. (OMS, Edwards, 2001, p. 17)	12
Figura 2:	Evolución del porcentaje de mayores que había utilizado Internet en los últimos 3 meses (INE, 2016)	14
Figura 3:	Comparación de la realización de actividades en Internet en personas de 55 a 64 años de edad y población general en 2015 (Fundación Telefónica, 2016, p. 12)	15
Figura 4:	Factores que afectan a la percepción del balance entre costes y beneficios de las diferentes TIC en personas mayores (Fundación Vodafone España, 2012, p. 7)	16
Figura 5:	Uso / disponibilidad de diferentes tecnologías en mayores (Fundación Vodafone España, 2012, p. 11).....	17
Figura 6:	Arquitectura GPS en Android (XDA, 2012)	29
Figura 7:	Modelo de espiral común (Pressman, 2010, p. 39).....	30
Figura 8:	Riesgos de seguridad en aplicaciones (OWASP, 2013, p. 5).....	32
Figura 9:	Modelo cliente – servidor.....	39
Figura 10:	Arquitectura del sistema.....	40
Figura 11:	Arquitectura del sistema (detalle capas).....	40
Figura 12:	Proceso de análisis y diseño de sistemas (Kendall y Kendall, 2011, p. 18)....	41
Figura 13:	DFD de contexto	43
Figura 14:	DFD Nivel 0 – Procesos principales	43
Figura 15:	DFD Nivel 1 – Proceso <i>Localizar usuarios próximos</i>	44
Figura 16:	DFD Nivel 1 – Proceso <i>Compartir perfiles</i>	45
Figura 17:	DFD Nivel 1 – Proceso Compartir propuestas de actividades	46
Figura 18:	DFD Nivel 1 – Proceso <i>Acordar citas</i>	47
Figura 19:	DFD Nivel 1 – Proceso <i>Autenticar y autorizar</i>	48
Figura 20:	DFD Nivel 1 – Proceso <i>Registro usuarios</i>	48
Figura 21:	Diagrama de caso de uso <i>Alta en el sistema y elaboración del perfil</i>	49
Figura 22:	Diagrama de caso de uso <i>Conocer perfiles y actividades</i>	51

<i>Figura 23:</i>	Diagrama de caso de uso <i>Propuestas de actividades y acuerdos de encuentro</i>	
	52	
<i>Figura 24:</i>	Diagrama de caso de abuso <i>Usuario con falsa identidad o con el propósito de engañar a otros usuarios</i>	56
<i>Figura 25:</i>	Diagrama de abuso <i>Atacante</i>	56
<i>Figura 26:</i>	Diagrama de Clases de análisis	59
<i>Figura 27:</i>	Diagrama de Clases de Diseño – Clases básicas	61
<i>Figura 28:</i>	Diagrama de Clases de Diseño – Servidor.....	62
<i>Figura 29:</i>	Diagrama de Clases de Diseño – Cliente	63
<i>Figura 30:</i>	Diagrama de Secuencias de Sistema – Activar la app	65
<i>Figura 31:</i>	Diagrama de Secuencia de Sistema – Conocer usuarios próximos	66
<i>Figura 32:</i>	Diagrama de Secuencia de Sistema – Acordar cita.....	67
<i>Figura 33:</i>	Diagrama de componentes	68
<i>Figura 34:</i>	Diagrama Entidad Relación – Vista general	69
<i>Figura 35:</i>	Diagrama Entidad Relación – Base de datos	70
<i>Figura 36:</i>	Esquema de la GUI – <i>Activar la app</i>	73
<i>Figura 37:</i>	Esquema de la GUI – <i>Perfil, Conocer usuarios próximos y Actividades</i>	74
<i>Figura 38:</i>	Esquema de la GUI – <i>Acordar citas</i>	75

Índice de tablas

Tabla 1	Porcentaje de personas de 55 a 74 años que habían utilizado Internet en los últimos 3 meses	14
Tabla 2	Requerimientos funcionales.....	36
Tabla 3	Requerimientos no funcionales.....	38
Tabla 4	Escenario de CU Registro de usuario.....	49
Tabla 5	Escenario de CU Rellenar perfil.....	50
Tabla 6	Escenario de CU Conocer usuarios próximos.....	51
Tabla 7	Escenario de CU Proponer actividad	53
Tabla 8	Escenario de CU Apuntarse a actividad.....	53
Tabla 9	Escenario de CU Acordar cita.....	54
Tabla 10	Tabla “usuarios”	70
Tabla 11	Tabla “telefono_verificacion”	71
Tabla 12	Tabla “actividades”	72
Tabla 13	Tabla “citas”	72

1. Introducción

En el presente documento describimos la labor de preparación y desarrollo correspondiente al Trabajo de Fin de Máster, para el que se propuso la realización de una aplicación web destinada a paliar la soledad en personas de más de 60 años de edad. En concreto, planteamos el diseño y desarrollo de una aplicación de contactos orientada a que personas de esa franja de edad compartan el tiempo libre.

Si bien las aplicaciones de redes sociales o “de citas” pensadas para establecer contactos o encuentros abundan en el mercado, buscábamos dar un enfoque distintivo, dentro de nuestras modestas posibilidades. La mejora más evidente con respecto a servicios de este tipo orientados a un público general es que se persigue la orientación específica a las necesidades de las personas de la franja de edad por encima de los 60 años, que no se sentirían identificadas con muchas de estas aplicaciones, ya que suelen ser utilizadas mayoritariamente por un público más joven y que usualmente están enfocadas a la búsqueda de contactos románticos más que a mera compañía. Buscábamos con este enfoque además beneficios de carácter social o cubrir necesidades relacionadas con el bienestar psicológico y la calidad de vida de este colectivo.

Por otro lado, una finalidad esencial de este trabajo era experimentar y poner en práctica los conocimientos adquiridos en el transcurso del máster, así como otros nuevos, particularmente relativos a la construcción y despliegue de aplicaciones web, apps o aplicaciones cliente para móviles, tecnología de geoposicionamiento y servicios web.

Para el desarrollo del trabajo seguimos un modelo de proceso de software iterativo e incremental, con un enfoque orientado a objetos. Tanto para definir el dominio del problema (análisis) como el de la solución (diseño) utilizamos métodos de modelado y nos apoyamos en la realización de diferentes diagramas, algunos según el estándar UML (Lenguaje Unificado de Modelado). Por otra parte, tuvimos en cuenta aspectos relativos a seguridad (S-SDLC, Ciclo de Vida de Desarrollo de Software Seguro) y riesgos de seguridad en aplicaciones web.

Por último, comentar que la interfaz de usuario se diseñó siguiendo criterios de usabilidad y accesibilidad.

1.1. Justificación

La justificación para desarrollar la aplicación se asienta en tres aspectos:

- Mayor vulnerabilidad a la soledad en personas de más de 60 años y sus consecuencias.
- Aplicar el concepto de envejecimiento activo. Al menos en cuanto a propiciar más oportunidades de interacción social y de compartición del tiempo libre.
- Incremento en la utilización de TIC (Tecnologías de la Información y Comunicación) entre personas mayores y especialmente de aplicaciones para dispositivos móviles.

A continuación profundizamos en ellos, para analizar el contexto en el que se circunscriben el enfoque de la aplicación, sus especificaciones funcionales y el dominio del problema al que trata de dar respuesta.

1.1.1. Aclaraciones previas sobre el término “mayores”

Tenemos que aclarar que el término “mayores” lo usamos en esta memoria para referirnos a un grupo amplio, diverso y heterogéneo como es el de las personas de más de 60 años. Es el criterio utilizado por la OMS (2001) y también por las Naciones Unidas en sus previsiones demográficas para referirse a las edades “más avanzadas”. La edad cronológica no es el único o mejor de los indicadores aparejados al proceso de envejecimiento, ya que existe una amplia diversidad en las condiciones de salud de personas pertenecientes a esta franja de edad, si bien comparten necesidades psicosociales que son relevantes para el propósito de la aplicación que planteamos. Es el segmento de edad que nos resulta relevante además porque, como veremos, es en el que hay una clara tendencia de crecimiento en el manejo de dispositivos móviles para usos diversos.

También es interesante comentar que hay numerosas personas de esa franja de edad que, por sus condiciones físicas y psicológicas, no se sienten “mayores”, trayendo a colación la denominada “cuarta edad”, es decir mayores de 80 años. La mayor esperanza de vida y las mejores condiciones de salud en las que las personas afrontan el proceso de envejecimiento, al menos en los países con mayores niveles de desarrollo, hacen necesaria esta nueva categoría.

1.1.2. Soledad en los mayores

Las personas de más de 60 años son las más vulnerables a la soledad y a padecer aislamiento social, lo que puede mermar notablemente su bienestar e influir de manera negativa en su estado de salud psicológica y física (Edwards, 2001).

El sentimiento de soledad experimentado en esta etapa de la vida puede conllevar dependencias de índole social, funcional, cognitivo y problemas de salud que pueden interferir en el nivel de autonomía y el desenvolvimiento de la vida cotidiana de la persona (Rodríguez, 2009).

En términos objetivos la soledad se puede definir como la ausencia de compañía. Es en el plano subjetivo o de vivencia de la soledad donde podemos indagar en el estado psicológico consecuente con esa situación. La soledad en sí misma no conlleva en toda circunstancia un sentimiento negativo o es vivida de forma desagradable, ya que puede resultar una experiencia buscada y positiva para la persona. Sin embargo, la soledad no deseada, sobrevenida, puede desencadenar un sentimiento negativo, de malestar emocional, que puede generar graves consecuencias psicológicas en situaciones sostenidas en el tiempo. Este sentimiento puede tener origen en el hecho de carecer de compañía para llevar a cabo actividades deseadas o para lograr intimidad emocional.

Entre las causas de la soledad en personas mayores Laforest (1991 citado en Rodríguez, 2009) define tres crisis asociadas al envejecimiento: (1) la crisis de identidad, (2) la de autonomía, consecuencia del deterioro del organismo y (3) la de pertenencia, resultante de la disminución de la vida social. Asimismo se pasan por situaciones vitales con un alto coste emocional como el “síndrome del nido vacío” (los hijos se van del hogar familiar para iniciar una vida independiente), la muerte del cónyuge o la pérdida de otros seres queridos y amigos, la salida del mercado laboral, la falta de actividades placenteras, etc.

Los recursos que las personas mayores utilizan para paliar la soledad son diversos, así como el afrontamiento de la misma. Según Rodríguez (2009) son importantes los recursos personales (actividades domésticas, televisión, radio, prácticas religiosas, contacto telefónico, centros destinados a mayores, actividades de ocio, turismo, culturales) así como los apoyos familiares y sociales (amistades, voluntariado social).

Insistiendo en los recursos sociales, contar con relaciones amistosas con las que realizar actividades de ocio y tiempo libre, comunicarse, tener apoyo o acercamiento emocional es uno de los posibles soportes para prevenir o afrontar las situaciones de soledad.

1.1.3. Envejecimiento activo

En nuestra sociedad se observan prejuicios que asocian el envejecimiento a no poder tener una vida productiva, dejar de ser útil, no tener experiencias ni necesidades sexuales, estar anticuado y ser dependiente. En suma, se vincula a una etapa vital en la que se deterioran las esferas física, mental y social de la persona hasta su total declive. Sin embargo, el hecho es que cada vez más personas avanzan en edad manteniendo un nivel de autonomía, capacidades y satisfacciones vitales, tal y como se refleja en diversos estudios longitudinales (Sarabia, 2009), sin perder de vista que la población de mayores es un grupo heterogéneo y muy diverso.

La Organización Mundial de la Salud (OMS) adoptó a finales del pasado siglo el concepto de *envejecimiento activo*, más completo que el de *envejecimiento saludable*, ya que extiende la idea más allá de estar activo desde un punto de vista físico, incluyendo también la de mantener o potenciar la implicación en cuestiones sociales, económicas, espirituales, culturales y cívicas. Una definición de envejecimiento activo la proporciona Edwards (2001): “es el proceso por el cual se optimizan las oportunidades de bienestar físico, social y mental durante toda la vida con el objetivo de ampliar la esperanza de vida saludable, la productividad y la calidad de vida en la vejez” (p.15)

Uno de los potenciales beneficios de trasladar políticas que fomenten el envejecimiento activo es según el mismo autor que las personas mayores sigan siendo independientes y disfruten de una calidad de vida positiva (Edwards, 2001).

En la misma línea, compartimos la visión de Kalache y Keller (1999):

“Para que los individuos contribuyan a la sociedad, la buena salud es un facilitador clave. Sin embargo, la buena salud a menudo refleja el grado de apoyo que uno recibe de la sociedad. Por lo tanto, es crucial garantizar que las personas mayores tengan todas las oportunidades de mantenerse activos. Así como la salud sostiene la actividad, una vida activa constituye la mejor oportunidad de estar saludable.”

El envejecimiento activo se encuentra determinado por diferentes factores (entorno, personales, sanidad y servicios sociales, conductuales y sociales). Entre ellos los factores del entorno social y, en particular, el apoyo social, resulta especialmente importante en la vejez ya que, como hemos comentado anteriormente, es la etapa vital en la que se es más vulnerable a la soledad y al aislamiento social.



Figura 1: Los determinantes del envejecimiento activo. (OMS, Edwards, 2001, p. 17)

Siguiendo a Edwards (2001) un objetivo primordial para materializar el concepto ha de ser el de mantener la autonomía, la propia capacidad para controlar, afrontar y tomar decisiones sobre la vida cotidiana. La salud, el factor clave de la autonomía, posibilita una experiencia de envejecimiento activo.

El concepto de envejecimiento activo está del mismo modo ligado a la prevención, tanto de la enfermedad o patología física, como de la salud mental y el bienestar psicológico, muy relacionados estos últimos con la interacción social.

Otro concepto que podríamos relacionar es el de “envejecimiento exitoso”, más subjetivo, ya que cabe preguntarse qué se entiende por “éxito” en este contexto. Según Sarabia (2009, p. 172) “ocurre cuando las personas sienten satisfacción por poder adaptarse a las situaciones cambiantes de su vida”. Entre otros aspectos que son englobados en ese concepto se encuentra la calidad de vida en los mayores. La calidad de vida “es la interacción entre la satisfacción de vida y las condiciones objetivas de vida. Estas últimas engloban la salud, el cuidado familiar, las actividades laborales o recreativas, las prestaciones estatales y la satisfacción de las necesidades de relación social” (Sarabia, 2009, p. 173). De nuevo, las actividades recreativas y la interacción social satisfactoria vuelven a tener un lugar relevante

en asegurar esa calidad de vida o bienestar psicosocial, sin despreciar los otros aspectos relativos a salud, estabilidad económica, recursos comunitarios, etc.

1.1.4. Mayores y uso de las TIC

Se presume la dificultad y la reticencia en el acceso y la utilización de las nuevas tecnologías (móvil y aplicaciones para dispositivos portátiles, acceso a Internet y utilización de servicios en red, etc.) por parte de las personas mayores y la existencia, como la denomina Norris (2001), de una “brecha digital social”, es decir, de una polarización en la adopción y utilización o no de las TIC entre diferentes grupos sociales, sin embargo algunos estudios muestran que la realidad es diferente y más alentadora, al menos en el entorno español.

Los mayores desean aprovecharse de los beneficios que se derivan de estas tecnologías y pueden vencer las posibles dificultades para adoptar estos avances en su vida cotidiana (Pino, Soto y Rodríguez, 2015). Las ventajas que pueden reportarles son similares a las que tienen para otros grupos de población, con los especiales beneficios que para las personas mayores ofrecen en cuanto a mantener el contacto y comunicarse con otras personas independientemente de la distancia que los separe, contar con nuevas fuentes de información y recursos de aprendizaje, incrementar el contacto con familiares y amigos, proporcionar sensación de seguridad, realizar determinadas tareas con mayor facilidad y fomentar las relaciones inter-generacionales. El uso de estas tecnologías redundan en la mejora de su autonomía personal y social (Pino et al., 2015) y de su nivel de autoestima (Shaw y Gant, 2002 citado en Rodríguez, 2009).

Por otro lado, la mencionada brecha digital está reduciéndose de manera drástica en España (Fundación Vodafone España, 2012; Fundación Telefónica, 2016; INI, 2016), incluso en lo tocante a la utilización de una mayor diversidad de aplicaciones y servicios, lo que ha debido suponer para los nuevos usuarios vencer la dificultad adicional que implica cambiar de hábitos o formas de proceder que estaban arraigadas.

Los datos recogidos en la encuesta sobre equipamiento y uso de tecnologías de información y comunicación en los hogares del INE (2016) reflejan un elevado crecimiento en el uso de las TIC en las personas mayores en España, tal y como puede observarse en la tabla 1. Como indicativo, si comparamos los datos del año 2006 con los del 2016, el crecimiento en la utilización de Internet en el caso de las personas en la franja de edad de 55 a 64 años fue

de aproximadamente un 358% y de un 680% en el caso de los que se situaban entre los 65 y 74 años.

Tabla 1

Porcentaje de personas de 55 a 74 años que habían utilizado Internet en los últimos 3 meses

	2016	2015	2014	2013	2012	2011	2010	2009	2008	2007	2006
55 a 64	64,8	61,4	55,4	46,5	43,8	37,9	34,6	29,3	24,6	21,5	18,1
65 a 74	34,7	31,3	26,2	21,9	19,0	16,2	13,8	11,2	8,6	6,6	5,1

INE (2016)

La tendencia está claramente encaminada a un crecimiento en estas franjas de edad, tal y como se muestra en el gráfico de la siguiente figura:

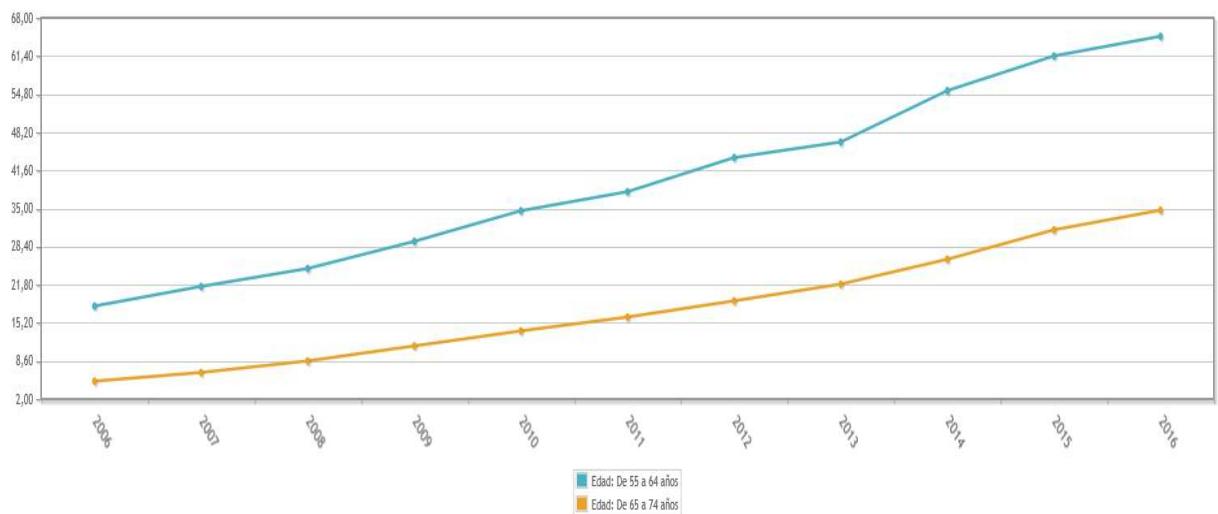


Figura 2: Evolución del porcentaje de mayores que había utilizado Internet en los últimos 3 meses (INE, 2016)

En el mismo sentido apunta el estudio llevado a cabo por la Fundación Telefónica (2016), que recoge cómo la mencionada brecha digital ha ido reduciéndose a un buen ritmo año tras año en la franja de edad que nos ocupa, siendo el incremento del doble en términos porcentuales en el acceso a Internet y a diferentes usos productivos de esta red con respecto al resto de la población, con tasas que se situaron alrededor de los 21 puntos porcentuales anuales de crecimiento. El mencionado estudio también señala cómo estaba próxima a desaparecer en el 2015 la brecha en la franja de 55 a 64 años con respecto a ciudadanos de otras generaciones en cuanto a la utilización de diversas aplicaciones o

servicios disponibles a través del uso de las TIC, por ejemplo efectuar compras, realizar cursos de formación, operaciones bancarias o trámites con la Administración, como puede observarse en la siguiente figura:

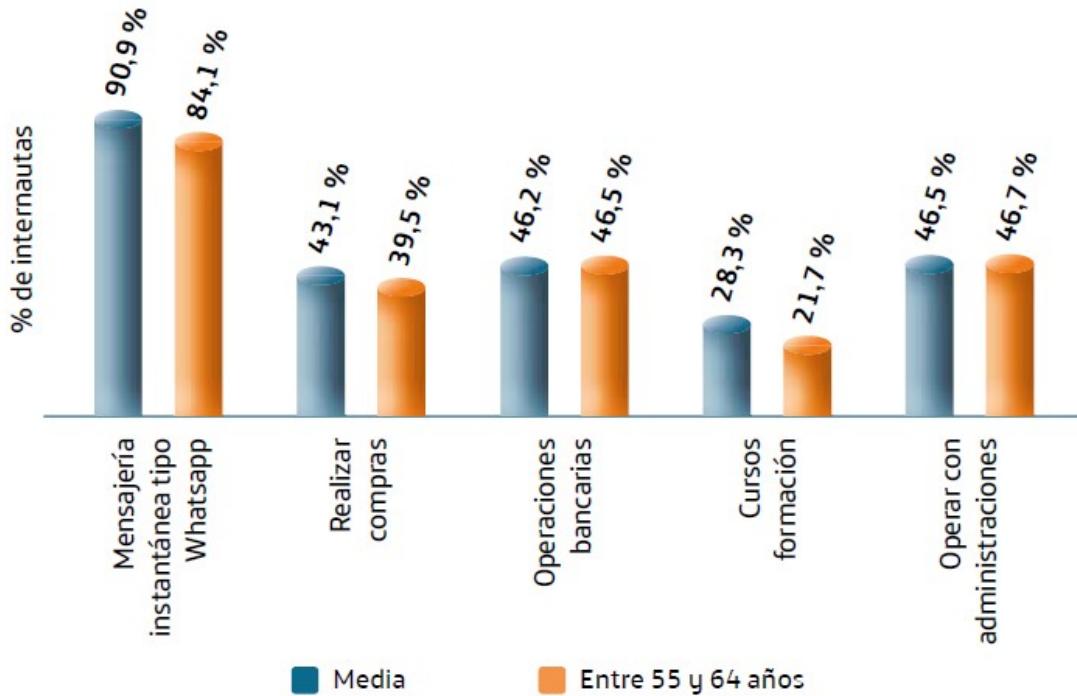


Figura 3: Comparación de la realización de actividades en Internet en personas de 55 a 64 años de edad y población general en 2015 (Fundación Telefónica, 2016, p. 12)

Según el estudio realizado por la Fundación Vodafone España (2012) existen una serie de factores que influyen en la percepción entre los costes y beneficios, no sólo de tipo económico, sino de recursos materiales e inmateriales que se deben aportar frente a las ventajas asociadas, en la adopción de la utilización de TIC en personas mayores. Estos factores son esencialmente el nivel formativo, la edad, la situación económica, el nivel de actividad (requerido por el uso de la herramienta) y la situación personal y familiar.



Figura 4: Factores que afectan a la percepción del balance entre costes y beneficios de las diferentes TIC en personas mayores (Fundación Vodafone España, 2012, p. 7)

En el mencionado estudio de la Fundación Vodafone España (2012) se diferencian tres grupos en base al balance costes / beneficios y de sus utilidades: (1) las de uso universal, como la utilización del móvil para efectuar llamadas telefónicas o realizar fotos, (2) las que tienen un grado medio de penetración, como el envío de mensajes, búsqueda de información en Internet, prensa digital, mensajería, correo electrónico, y (3) finalmente las que tienen una utilización testimonial, las redes sociales y el acceso a Internet a través del móvil estarían en este último grupo.

En este avance del uso de las TIC el papel de los móviles clásicos y de nueva generación (*smartphones*) es destacado como puede observarse en la figura siguiente. Es una tecnología adoptada por la práctica totalidad de la población, si bien el grado de utilización de las diferentes aplicaciones y servicios disponibles es todavía diverso, pudiendo diferenciarse grupos de usuarios dentro de los mayores, desde los que utilizan funcionalidades básicas hasta los que hacen un uso más extensivo de los servicios disponibles en sus terminales.

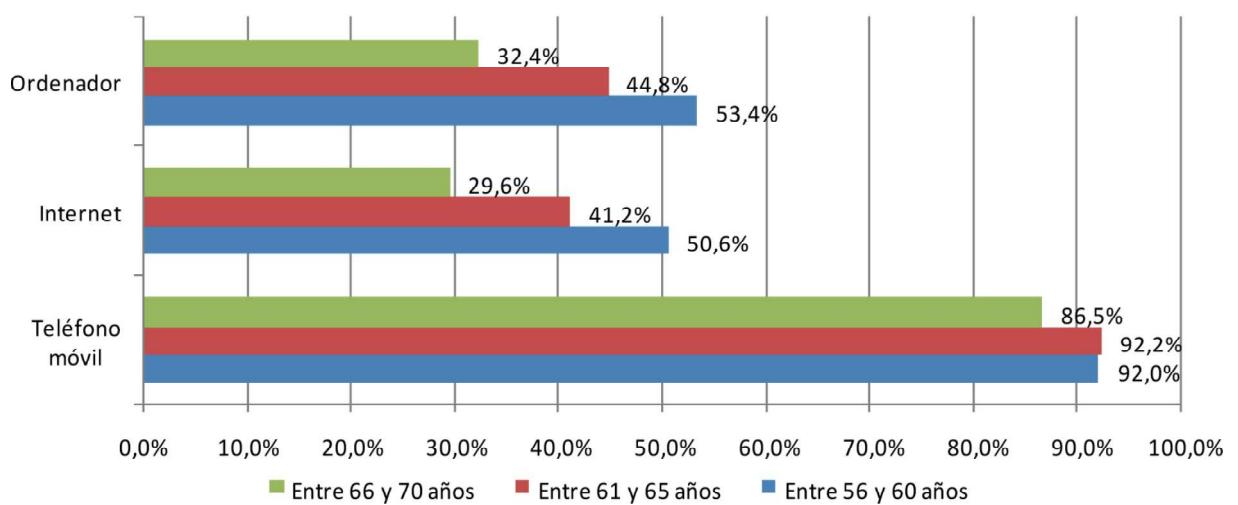


Figura 5: Uso / disponibilidad de diferentes tecnologías en mayores (Fundación Vodafone España, 2012, p. 11)

En conclusión, buena parte de los mayores desean ser partícipes de las ventajas asociadas a la utilización de las TIC. Los datos evidencian que la adquisición por parte de las personas mayores de estas nuevas tecnologías para su uso cotidiano, incluyendo la utilización de diferentes aplicaciones y servicios, se está igualando con respecto a grupos de personas de otras generaciones. En lo que respecta a España la llamada “brecha digital” o diferencia en el uso de las TIC de las personas mayores con respecto a las pertenecientes a otros grupos de edad, se reduce de manera significativa año tras año y previsiblemente desaparecerá de forma completa en los próximos años, antes en la franja de edad comprendida entre los 55 y 64 años y posteriormente en edades más avanzadas.

La proliferación de la aceptación y utilización de móviles con características avanzadas (*smartphones*) entre las personas de mayor edad también es una clara tendencia. Si bien la utilización de los servicios que brindan es todavía menor en las personas mayores, son, dentro del rango de edad que estamos considerando, los más jóvenes los que tienen dispositivos más modernos y utilizan más servicios de entre los disponibles.

Por último, es necesario considerar que las personas pueden verse limitadas en sus capacidades físicas y cognitivas como consecuencia del envejecimiento. Por tanto, si queremos que participen de los nuevos desarrollos y servicios, éstos han de ser diseñados para que puedan ser utilizados por el mayor número de personas considerando criterios de

usabilidad y accesibilidad, eliminando barreras, aparte de la utilización de las ayudas o adaptaciones especiales que posibiliten la adaptación de las personas en función de sus habilidades físicas o sensoriales.

1.2. Enfoque del concepto de la aplicación

El objetivo principal que tratamos de conseguir es el de sacar ventaja a la utilización de las TIC y para ello diseñar y desarrollar una aplicación para dispositivos móviles destinada a paliar la soledad de personas de más de 60 años. Conforme a lo que hemos comentado en el apartado de Justificación, consideramos que existían una necesidad y una oportunidad para desarrollar la aplicación.

- Necesidad: Es en la franja de edad comprendida a partir de los 60 años cuando más útil puede resultar una aplicación de este tipo. Son las personas de este grupo de edad en las que el círculo social se suele ver resentido y existe un mayor riesgo de soledad y aislamiento social, como hemos analizado en el apartado anterior.
- Oportunidad: Incremento de la utilización de TIC, en particular dispositivos y aplicaciones móviles en la franja de edad, con tendencia a continuar creciendo tanto en cantidad como en tipo de usos.

A partir de cuestionarnos cómo combinar esta necesidad y oportunidad barajamos diferentes ideas acerca de la aplicación. Imaginándola surgieron diversas ideas que finalmente quedaron en descartes: por ejemplo desarrollar la aplicación para poner en contacto a mayores con asociaciones y organizaciones para realizar actividades de voluntariado o de implicación en actividades de carácter social; facilitar establecer contacto con personas de cualquier franja de edad; establecer contacto con familiares; contactar de forma virtual (modelo de red social); recoger ofertas de ocio, culturales o de tiempo libre dirigidas a un público en general y destacando las que fueran para personas mayores; actividades de turismo para personas mayores, etc. Cualquiera de estas ideas podían haber dado como fruto una aplicación con un determinado enfoque y funcionalidades, pero dada la finalidad principal de su realización como trabajo académico, así como por centrarnos en una funcionalidad básica y esencial, decidimos que la funcionalidad principal de la aplicación sería la de poner en contacto entre sí a personas de la misma franja de edad (con más probabilidad de tener intereses, objetivos y circunstancias comunes) y facilitar su encuentro

en persona, es decir fomentar las interacciones sociales reales. Si bien se conocen las ventajas del uso de los contactos virtuales para paliar la soledad en este grupo de edad, como se puso de manifiesto en el estudio de Clark (2002 citado en Rodríguez, 2009), buscábamos una mayor calidad en los contactos y una mayor posibilidad de generar lazos amistosos. De forma análoga consideramos que la proximidad geográfica era una variable que había que tener en cuenta para propiciar dichos encuentros.

Nos centramos en conseguir reforzar uno de los recursos para afrontar la soledad en esta franja de edad: los recursos sociales y las interacciones amistosas, facilitando que los usuarios puedan establecer nuevas relaciones sociales con personas de su grupo de edad, próximas geográficamente. Entroncando con el concepto de envejecimiento activo, se trataría de propiciar el encuentro para la realización de actividades y compartir aficiones o el tiempo libre.

Los potenciales usuarios serían personas con una edad avanzada, pero que son autónomas y quieren realizar actividades que les resulten accesibles; un segmento de la población que, como hemos analizado anteriormente, está en crecimiento.

La solución planteada es una aplicación destinada al sistema operativo Android que se ejecutaría en dispositivos móviles (*smartphones*) y exclusivamente orientada a usuarios de 60 años o más de edad.

La aplicación precisaría que el dispositivo estuviera dotado de pantalla táctil, cámara, GPS y conectividad a una red de datos (vía WiFi o conexión de datos móviles), que son características habituales en la mayor parte de *smartphones* en la actualidad.

La aplicación basaría su funcionamiento en un modelo cliente-servidor. Se conectaría a un servicio web residente en el servidor que contaría a su vez con una base de datos centralizada que contendría los datos de registro de los diferentes usuarios, así como su posición geográfica, información relativa a intereses, propuestas de actividades y citas.

En cuanto a la operativa con el sistema, una vez registrado el usuario, la aplicación cliente envía la ubicación geográfica al servidor. El usuario dispondría también de opciones para proporcionar información acerca de sus intereses, aficiones, preferencias de tiempo libre, proponer actividades, etc.

Con la información recogida, el sistema propondría a cada usuario la posibilidad de establecer contacto con otros usuarios próximos a su ubicación, priorizados u ordenados en

función de la proximidad y la coincidencia en intereses, haciendo llegar las propuestas de actividades, pudiendo entonces decidir si quiere sumarse a la actividad y establecer contacto y encuentro. Del mismo modo, le llegarían cuántas personas se han sumado a las propuestas de actividades realizadas por él o ella.

Una vez que se ha establecido una conexión (dos usuarios aceptan compartir una actividad), el que había sugerido la actividad propone fecha, hora y sitio para conocerse en persona.

A parte de las cuestiones técnicas relativas al funcionamiento interno del sistema de información, un aspecto que consideramos que sería relevante cuidar es la interfaz de usuario, en especial por las características de la población a la que iría destinada la aplicación que, como parte del proceso de envejecimiento, podrían haber sufrido deterioro en sus capacidades físicas, motoras o sensoriales, o cognitivas.

Siguiendo criterios de usabilidad, el diseño de la aplicación tendría como objetivo conseguir una interfaz de usuario en la que destacara la sencillez (minimalismo) y facilidad de uso. Hemos de considerar que el tamaño de la pantalla en los móviles es reducido y que se interactúa con las aplicaciones a través de determinados movimientos en la pantalla táctil, por ello sería conveniente ajustar la precisión y rapidez necesarias para interactuar con la aplicación, así como el esfuerzo cognitivo para operar, procurando en suma que la aplicación cliente resulte accesible, por ejemplo mediante la utilización de iconos y tipografía de tamaño grande, con apoyo de señales de tipo auditivo, reduciendo al mínimo el número de pasos necesarios hacer uso de las funcionalidades, incrementando el tamaño de los botones o de la superficie de pantalla que el usuario tenga que pulsar para interactuar.

2. Contexto tecnológico

En este apartado repasamos la situación presente y analizamos los servicios y aplicaciones destinadas a conocer a otras personas a través de las redes sociales y citas *online*, que son el contexto en el que se circunscribe el sistema que desarrollamos. Por otra parte, describimos las tecnologías que utilizamos tanto en el lado del servidor como del cliente y las enmarcamos en el entorno actual.

2.1. Aplicaciones o servicios de contactos

La importancia de las relaciones sociales tiene un reflejo en las aplicaciones online denominadas “redes sociales”, en las que los usuarios pueden compartir con otros su propia identidad, o al menos la imagen de uno mismo que se quiere proyectar hacia los demás, y establecer relaciones con otros. Estos sitios y aplicaciones se han integrado en la vida cotidiana de muchas personas y mediatizan parte de sus interacciones sociales, en algún caso de forma sustancial. Del mismo modo, han experimentado un elevado crecimiento los sitios web expresamente destinados a conocer a otras personas con la finalidad de concertar citas de carácter amistoso, amoroso o sexual.

Según Boyd y Ellison (2008) los *“social network sites”* (SNS) o redes sociales, son servicios basados en la web que permiten a las personas: (1) construir un perfil público o semipúblico dentro de un formato definido, (2) elaborar una lista de otros usuarios con quienes comparten una determinada conexión y (3) poder recorrer su lista de conexiones y las realizadas por otros conforme a los límites que establezca el sistema. La naturaleza y la denominación de estas conexiones pueden variar de un sitio a otro.

No es este el espacio para hacer un recorrido histórico riguroso de este fenómeno, pero podemos citar varios hitos que enmarcan su contexto, como la aparición en los años noventa de las primeras “comunidades de usuarios” en Internet. Estas comunidades, frente a las páginas en las que el contenido era decidido, recopilado o elaborado y publicado por los responsables de la web o bien el administrador, eran sitios web en los que los usuarios podían generar y compartir contenido, abriendo a los usuarios la posibilidad de intercambiar información sobre intereses y aficiones. Otro avance fue la creación de los *chats*, el primero el de la compañía AOL en 1997 y, finalmente, la llegada a Internet de las redes sociales tal y

como las conocemos ahora, Linkedin en el 2002, MySpace y Facebook el año siguiente, Twitter en 2006 y Google+ en 2011, por citar alguna de las más relevantes.

Esta evolución ha ido allanando el camino y eliminando prejuicios en cuanto a la utilización de la Red para establecer contactos de tipo amistoso o bien sentimental, pasando de estar socialmente estigmatizado a estar mejor visto y experimentar un gran crecimiento en los últimos años. Debemos aquí establecer una diferenciación entre las así denominadas redes sociales y las aplicaciones de citas o contactos (“*online dating*”), aunque puedan tener similitudes. La diferenciación con respecto a otros medios de conocer a otras personas es que precisamente los SNS permiten articular y hacer visible la red social del usuario, pero el objetivo de la mayor parte de los usuarios no es necesariamente establecer contacto o conocer a otros que les resultan desconocidos, sino que se comunican principalmente con personas que ya forman parte de su red social extensa (Boyd y Ellison, 2008). En contraste, los sitios destinados a establecer contactos tienen como objetivo primordial que sus usuarios puedan conocer y tratar relación con personas desconocidas con las que compartir encuentros románticos, sexuales, compañía, aficiones o actividades de ocio.

Según nuestras indagaciones, en el panorama actual las aplicaciones “*online dating*” o “*dating sites*” son muy numerosas. En un mercado con tal grado de madurez, algunas buscan llegar a determinados nichos de mercado, ya sea estando destinadas sólo a personas con una orientación sexual concreta o con determinadas preferencias sexuales, o predilecciones acerca de características personales, como la edad, raza, nacionalidad o incluso religión de la posible pareja, o bien diferenciándose por la naturaleza de la relación que se busca, ya sea de tipo amistoso, romántico, sólo para contactos sexuales esporádicos, o para otros usos específicos, como compartir tiempo libre o actividades. En algunos países, como Estados Unidos, operan tal cantidad de estos servicios que existen incluso sitios que ofrecen búsquedas y análisis para ayudar a los potenciales usuarios a elegir (por ejemplo <https://www.datingsitesreviews.com/>).

Como ejemplos de servicios de *online dating* para la búsqueda de pareja o bien relaciones de tipo sexual podemos citar las siguientes aplicaciones: Tinder (<https://www.gotinder.com/>), eDarling (<http://pareja.edarling.es/>), Zoosk (<https://www.zoosk.com/>), Meetic (<https://www.meetic.es/>), Badoo (<https://badoo.com/es/>), Feeld (<https://feeld.co/>), LocalSin (<https://www.localsin.com/>), Hapnn (<https://www.happn.com/es/>), Grindr (<https://www.grindr.com/>), Wapa (<http://wapa-app.com/>), Down (<https://www.downapp.com/>), Craigslist (<https://www.craigslist.org>). Para compartir aficiones citamos a modo de ejemplo a

Meetup (<https://www.meetup.com>) y Mitmi (<http://www.mitmiapp.com/index-es.html>). Estas son tan sólo una muestra en la que hemos intentado recoger las más relevantes o de mayor difusión, o que pueden ser representativas de las destinadas a un determinado segmento, ya que en el mercado podemos encontrar muchas más.

La práctica totalidad de estos sitios web tienen su correlato en aplicaciones para móviles, o bien de forma directa han sido aplicaciones desarrolladas para este dispositivo, como es el caso de Tinder, la aplicación líder en el mercado. La modalidad de la aplicación móvil abre además nuevas posibilidades en la utilización y en la forma de concebir el servicio. Una muestra de ello es el uso de la geolocalización de los dispositivos para procurar el encuentro con usuarios cercanos geográficamente, esta tecnología es la base de Happn, por ejemplo, que la utiliza para dar a conocer a usuarios que se “han cruzado”, es decir, que han estado cerca físicamente en algún momento.

2.2. Destinadas a personas mayores

Si bien alguna de las aplicaciones citadas en el anterior apartado pueden ser utilizadas, por supuesto, por personas mayores, y permiten en algún caso establecer filtros de edad, si así lo desea el usuario, en cuanto a las personas con las que le gustaría contactar, queremos recoger aquí aquellas aplicaciones que de forma específica van destinadas a este público objetivo. Además, la mayor parte de los servicios existentes están copados por un público más joven, por lo que los mayores pueden percibir que “no es su sitio”, lo que supone una barrera de entrada.

El mercado objetivo de las personas por encima de los 50 años que pueden demandar este tipo de servicios, que es al que se dirige las aplicaciones de *dating* para mayores (o *seniors* como gustan de calificarlos), es extenso, sobre todo en ciertos países, por lo que varias compañías intentan hacerse un hueco o ampliar su cuota penetrando en este segmento de la población. En concreto hemos encontrado cuatro servicios de *dating* destinados a mayores que son accesibles a través de sitio web y de aplicaciones para móvil (Android e iOS): Stitch, Senior People Mingle, SeniorPeopleMeet y SeniorMatch.

Stitch (<http://www.stitch.net>) opera en algunas ciudades de Australia y Estados Unidos. Es un servicio de contactos destinado a personas de más de 50 años (su lema es “*Friendship, Activities, Travel and Romance for mature adults*”). Su negocio busca la diferenciación y está

volcado netamente a este colectivo, por lo que consideramos que es la propuesta más centrada en las necesidades y requerimientos de las personas de estas edades.

Mingle es otra de las aplicaciones que dispone de versiones específicas para establecer contactos románticos o amistosos. Pertenece a un desarrollador que, siguiendo una estrategia de segmentación para hacerse con potenciales clientes, ha lanzado aplicaciones diferenciadas, al menos en apariencia, como Mingle Citas, Single Parents, Asian Mingle. En concreto, la destinada a mayores de 50 años es Senior People Mingle.

Otra aplicación que opera sobre todo en Estados Unidos y Canadá es SeniorPeopleMeet (<https://www.seniorpeoplemeet.com/>), afiliada a Match.com (Meetic) y otras aplicaciones de citas como BlackPeopleMeet, LoveAndSeek y BabyBoomerPeopleMeet.

Por último, también operando en Estados Unidos y Canadá, encontramos a SeniorMatch (<https://www.seniormatch.com/>), en la que no admiten a nadie por debajo de 30 años, aunque su foco está en los que superan los 50. El mismo sitio enlaza con aplicaciones también dirigidas a segmentos concretos (Millionaire Dating, Older Women Dating, BBW Dating, Bisexual Dating, Interracial Dating, etc.)

Como podemos ver, son compañías que buscan su espacio en el mercado del *online dating*, o bien ampliarlo, abriendo el abanico a través de una estrategia basada en la segmentación o bien en la diferenciación, ofreciendo sitios específicos para mayores. Las aplicaciones que conseguimos encontrar están destinadas al mercado internacional y, en concreto, a Estados Unidos, Canadá y Australia.

2.3. Tecnologías

Conocer específicamente qué tecnologías utilizan las aplicaciones que existen en el mercado no es sencillo, ya que no hemos encontrado ninguna que lo haga público, si bien podemos observar que son aplicaciones de tipo distribuido, con alguna arquitectura que sigue el modelo cliente-servidor, en el que el usuario accede a través de su navegador o aplicación móvil (cliente) y se conecta con un servidor web o de aplicaciones que centraliza la gestión de los usuarios y el acceso a la base de datos.

Alguna de las aplicaciones hace uso de servicios de geolocalización para ubicar a los usuarios y poder ofrecerles contactos próximos, siendo incluso la base de los servicios de alguna de ellas.

En los siguientes sub-apartados nos referimos a la base tecnológica de la aplicación que hemos desarrollado. Tanto en el lado del servidor web (servicio web de tipo REST, formato de datos JSON, Gestor de Base de Datos SQL), como en el del cliente (aplicación para dispositivos móviles y georreferenciación).

2.3.1. Lado servidor web

Las tecnologías que empleamos en el lado del servidor fueron la puesta en marcha de un servidor web Apache que alojó la implementación de un servicio web tipo REST (basado en HTTP y JSON) conectado a un Gestor de Base de Datos MySQL.

Un componente esencial del sistema es el *servidor web*, el programa que atiende las peticiones de la aplicación cliente mediante el protocolo HTTP y ejecuta los scripts que constituyen el servicio web. Las dos opciones principales en cuanto a nivel de implantación son Apache y IIS (Internet Information Services). Según la encuesta de Netcraft (2017) la mayor cuota de mercado la ocupan estos dos, siendo el primero Apache, seguido de cerca de IIS; en la tercera posición, con un estimable crecimiento en los últimos años, se sitúa NGINX y, a más distancia, la última posición es para GWS (Google Web Sever).

Centrándonos en Apache y IIS, hemos de comentar que Apache es software libre, de código abierto, desarrollado por la Apache Software Foundation, mientras que IIS es propietario (Microsoft), ambos son modulares (IIS desde la versión 6), Apache es utilizado en servidores raíz de Internet y IIS es más utilizado en entornos corporativos (UNIR, 2016d). Aunque a priori tanto Apache como IIS hubieran sido elecciones igualmente válidas para desplegar el servidor web, nos decantamos por Apache. La elección vino determinada por varias razones. Contábamos con experiencia administrando ambos tipos de servidores, pero era mayor con Apache y, como pretendíamos utilizar y bastionar un servidor propio para hospedar el servicio, previmos que íbamos a realizarlo en un plazo menor al estar más familiarizado con este. Por otra parte, utilizamos el lenguaje de programación PHP para codificar los scripts del servicio web, por lo que la opción de Apache parecía la más conveniente, ya que puede extenderse con un módulo que le dota de un intérprete y su desempeño es más eficiente con este lenguaje.

Según Mifsuf (2012), Apache es robusto y tiene un ciclo de desarrollo muy rápido ya que cuenta con un nutrido grupo de colaboradores que intervienen en el mismo. Este autor también abunda en sus cualidades: (1) estabilidad, con una robustez contrastada que impide

caídas o cambios inesperados, (2) flexible y eficiente, capaz de trabajar con la mayor parte de las extensiones web actuales (por ejemplo PHP, SSL, CGI, SSL, proxy), (3) es extensible a través de módulos que amplían su funcionalidad, y (4) es multiplataforma, contando con versiones para los principales sistemas operativos (GNU/Linux, Windows, MacOS). Además, permite la configuración de *virtual hosts*, lo que permite operar diferentes sitios web en el mismo servidor de forma inadvertida para los usuarios. Desde la perspectiva de la seguridad, su naturaleza modular permitió instalar sólo aquellos módulos que eran necesarios para su funcionamiento, reduciendo así la superficie expuesta a potenciales ataques. Permite igualmente configurar el servidor para impedir acciones de *fingerprinting* que expongan información acerca del sistema (por ejemplo versión de Apache instalada, información del sistema operativo) así como la exposición de información de archivos o directorios (*directory listing*) que pueda ser utilizada por los atacantes.

En cuanto al servicio web, optamos por una arquitectura de tipo REST (*REpresentational State Transfer*). Esta arquitectura tiene como principal característica que no conserva datos entre peticiones, ya que utiliza el protocolo de comunicación HTTP, que no conserva estado, haciendo uso de sus métodos (GET, POST, PUT, DELETE, etc.) Una ventaja de esta arquitectura es que, al no precisar memoria para almacenar información de estado, es más eficiente en el uso de recursos para las peticiones y puede atender un número más amplio de ellas. Como contrapunto tiene la desventaja de que, precisamente por carecer de estado, cada solicitud de cliente a servidor debe contener toda la información necesaria para ser entendida y procesada, y no se puede tomar ventaja de ningún contexto almacenado en el servidor. Por tanto, se hace más complicado el manejo de sesiones (que el cliente pueda seguir interactuando con la aplicación web tras haberse autenticado una vez, sin tener que hacerlo en cada interacción), y el estado de la sesión se mantiene por completo en el cliente (Fielding, 2000; UNIR, 2016c).

Decidimos utilizar REST porque iba a resultar más fácil y conveniente su implementación, frente a otro tipo de servicios web, como los basados en SOAP. REST facilita escribir y documentar el servicio ya que, como hemos comentado, utiliza HTTP y sus métodos para realizar las operaciones CRUD básicas, si bien en nuestro caso utilizamos primordialmente POST y creamos nuestros propios métodos en la API para las diferentes acciones o peticiones solicitadas a la aplicación web; hemos de precisar que tales métodos o acciones no iban destinados a ser consumidos de manera pública, sino de forma exclusiva por la aplicación móvil. Otros aspectos a favor de REST es que hace un uso eficiente del ancho de

banda, ya que es mucho menos verboso que SOAP y las peticiones pueden almacenarse en caché para un mejor rendimiento y escalabilidad (Fielding, 2000).

REST soporta diversidad de formatos de datos, pero JSON (*JavaScript Object Notation*) es el predominante, alrededor de un 57% de las aplicaciones basadas en web lo utilizan o tienen componentes de JavaScript (Hazlewood, 2016). Para portar datos entre la aplicación cliente y la aplicación en el servidor web utilizamos este formato. JSON es un formato de intercambio de datos ligero introducido en 2011. Está basado en texto (Unicode) y se derivó del lenguaje de programación ECMAScript, si bien es independiente del lenguaje de programación. JSON define un pequeño conjunto de reglas para la representación de datos estructurados que permiten su portabilidad (ECMA International, 2013). Nuestra elección vino determinada por su simplicidad y flexibilidad, aparte de ser un formato ampliamente recomendado en servicios web basados en REST.

Otro de los componentes fundamentales es el Sistema Gestor de Base de Datos (SGBD), que da soporte a la capa de permanencia en el servidor. Era necesario contar con un motor de base de datos robusto, eficiente, seguro y que se integrara en el servidor web Apache. Entre las opciones más extendidas en bases de datos relacionales se encuentran MySQL / MariaDB, Oracle y MS SQL Server; el primero es de código abierto y los restantes se distribuyen bajo licencia comercial. MySQL parece tener mayor proyección en las bases de datos asociadas a sitios web, mientras que Oracle y SQL Server se orientan a clientes empresariales de gran tamaño (Lee, 2013). MySQL y Oracle son multiplataforma, mientras que MS SQL Server es exclusivo para sistemas operativos de Microsoft. Nos decantamos por MySQL, tanto por la posibilidad de uso libre, como por la amplia información disponible y su habitual integración en sistemas con sistema operativo GNU/Linux, servidor Apache e intérprete de PHP (las denominadas instalaciones tipo LAMP). Además MySQL aportó unas mejoras relevantes a partir de su versión liberada en 2010 (GA release 5.5), que incluyen replicación semi-sincrónica, el particionamiento personalizado, soporte mejorado para SMP (multiprocesamiento simétrico, que permite que unidades de procesamiento comparten el acceso a la memoria de forma simétrica) y las actualizaciones del subsistema InnoDB E/S, su motor de almacenamiento de datos.

2.3.2. Lado cliente – app móvil

En el lado cliente del sistema desarrollamos una aplicación destinada a dispositivos móviles de nueva generación con sistema operativo Android.

Los principales sistemas operativos para móviles en la actualidad son Android, iOS y Windows Phone. El propio de los dispositivos pertenecientes a Apple es iOS, Windows Phone es el desarrollo de Microsoft y Android está basado en Linux y es desarrollado por la *Open Handset Alliance*, una alianza comercial entre diversas empresas liderada por Google.

Android es el sistema operativo preeminente en el mercado español, cuya cuota se sitúa alrededor del 94% (Cinco Días, 2016). Esta es la plataforma que elegimos por tanto para nuestra aplicación. En concreto, la aplicación debía ser compatible desde la versión 4.2, ya que el porcentaje de usuarios que cuentan con Android a partir de dicha versión es al menos del 85,9% (Xataka, 2016), con lo que garantizábamos que sería ejecutable en la mayor parte de los móviles actuales en el entorno español.

La otra tecnología primordial para la aplicación móvil desarrollada es la georreferenciación o geolocalización. Un buen número de móviles de nueva generación integran en su hardware dispositivos que permiten la recepción de señales de satélite para utilizar el GPS (Sistema de Posicionamiento Global), este es uno de los mecanismos utilizados para fijar la posición del dispositivo, si bien existen otros dos métodos para hacerlo: (1) triangulación a partir de la intensidad de la señal recibida por las antenas de telefonía móvil próximas y (2) estimando la posición a partir de las señales WiFi cercanas captadas por el dispositivo, que previamente han sido registradas junto a su posición GPS (por ejemplo Google identificó sobre el terreno multitud de señales y las mapeó) o bien pertenecen a la compañía operadora y tienen identificada su situación (Brownworth, 2011).

En la figura siguiente recogemos los componentes de la arquitectura Android de geolocalización. Como puede observarse, el núcleo es el *GL Engine* que, según la configuración y la plataforma, adquiere, entre otros datos, información de ubicación de antenas, datos de asistencia de GPS, información de los satélites GPS, datos de servidores SUPL / NTP, información de redes, etc. Sobre este motor se sitúa la capa de abstracción de los Servicios de Localización, *frameworks* de clases que comentamos más adelante.

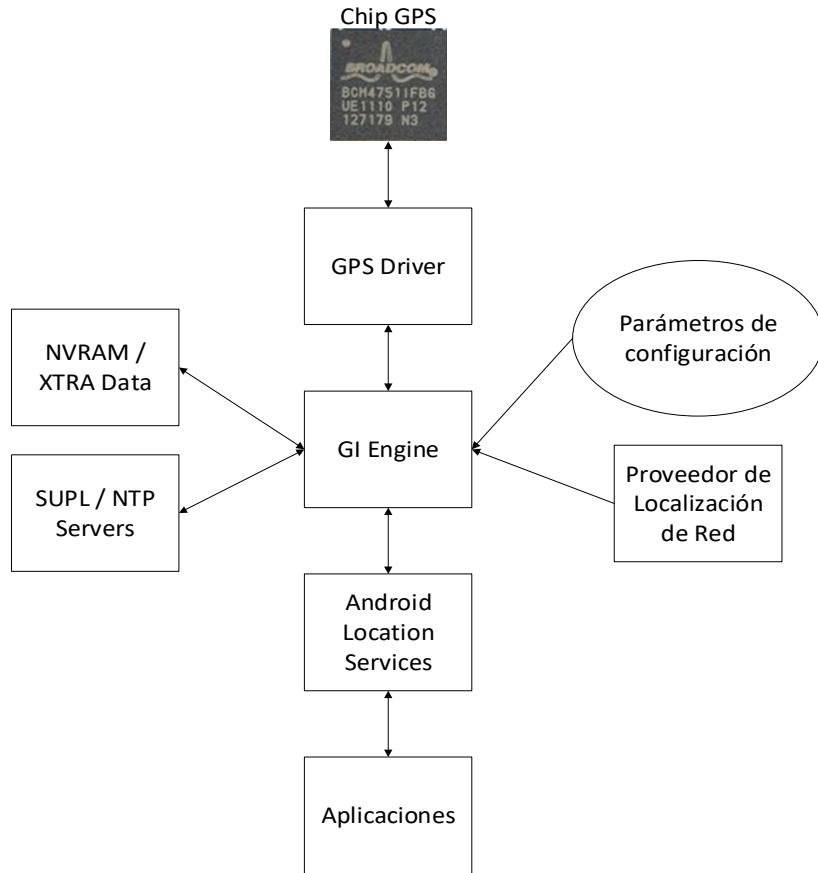


Figura 6: Arquitectura GPS en Android (XDA, 2012)

En Android existen dos formas de obtener acceso a los servicios de localización (Google, s.f.b). Una de ellas es mediante el *framework* de Android de APIs de localización (`android.location`) y la clase `LocationManager`, que provee acceso a los servicios de localización del sistema, lo que permite a las aplicaciones obtener con cierta frecuencia la localización geográfica del dispositivo, o bien lanzar una determinada acción (*Intent*) cuando el dispositivo se halla próximo a una determinada ubicación. La segunda forma es usar las APIs de localización de Google Play Services (Google Location Services API), esta es la recomendada actualmente por Google (s.f.b), puesto que constituye un *framework* de alto nivel más potente que el anterior, automatiza tareas como la elección de proveedores de ubicación y la administración de energía, siendo más eficiente en el uso de la batería; además, añade nuevas funciones y requiere menos codificación para su implementación.

3. Objetivos y metodología

3.1. Enfoque metodológico. Modelo de proceso de software

Según Sommerville (2005) un proceso de software es “un conjunto de actividades que conducen a la creación de un producto software” (p. 60). El mismo autor enumera las que considera las actividades fundamentales en los procesos de software, en concreto estas cuatro: (1) Especificación, en la que se definen la funcionalidad y restricciones; (2) Diseño e implementación, donde se produce el software que cumpla con las especificaciones; (3) Validación, que consiste esencialmente en contrastar con el cliente que el software realiza lo esperado, y (4) Evolución, en la que se modifica el software para atender las necesidades cambiantes futuras.

En nuestro caso, de estas actividades se llevaron a cabo las tres primeras, ya que el alcance del trabajo es un primer desarrollo de la aplicación, el objetivo básico era alcanzar un prototipo o versión funcionalmente completa de la misma. Sería en realizaciones futuras cuando se podría evolucionar el sistema para dotarlo de funcionalidades adicionales o bien modificar aquellas que se entienda conveniente.

En cuanto al modelo de partida, seguimos el modelo de proceso de software en espiral, propuesto por Barry Boehm en 1988 (citado por Sommerville, 2005 y por Pressman, 2010). La figura siguiente refleja una representación simplificada del mismo:

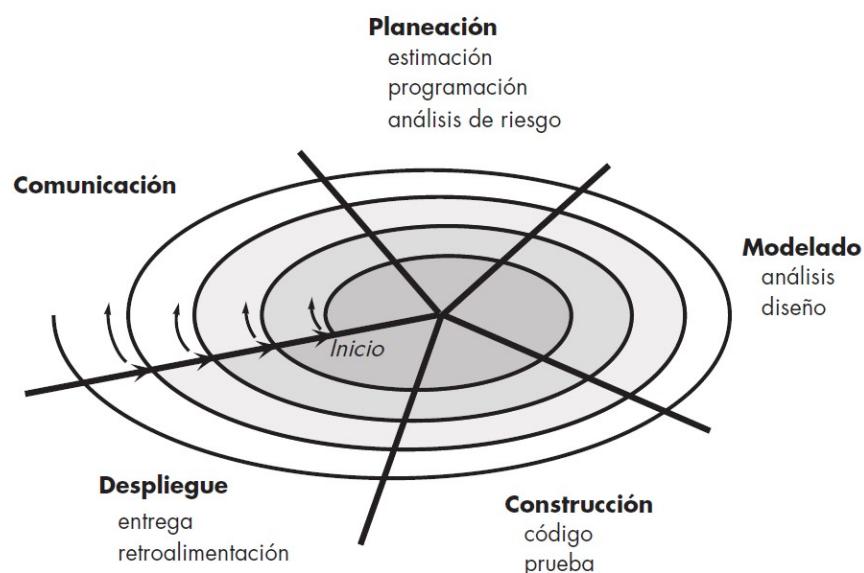


Figura 7: Modelo de espiral común (Pressman, 2010, p. 39)

El mencionado modelo se caracteriza por seguir un patrón iterativo e incremental, por tanto nos permite desarrollar el sistema a través de iteraciones que le añaden incrementos o funcionalidades hasta lograr una versión del sistema completo. Además, no descartamos futuras iteraciones del proceso que nos conduzcan a desarrollos mejorados del producto software, refinando y ampliando sus funcionalidades.

El enfoque metodológico seguido queda resumido en las siguientes fases:

1. Especificaciones del sistema. Planteamiento inicial, determinación de los requisitos funcionales y no funcionales.
2. Desarrollo del sistema. Análisis y diseño del sistema. Descripción del sistema y modelado para describir, por un lado, el dominio del problema y los requerimientos y, por otro lado, el dominio de la solución conceptual.
3. Implementación. En esta fase llevamos a cabo la programación y codificación que implementan el diseño.
4. Despliegue. Puesta en producción del sistema de información y validación del mismo en cuanto al cumplimiento de los requerimientos.

3.2. Protección ante riesgos de seguridad

Una de las prioridades en el desarrollo es el seguimiento de un modelo de S-SDLC (ciclo de vida del desarrollo de software seguro) acorde con el modelo en espiral, iterativo e incremental. Para ello tuvimos en cuenta los “Seven Touchpoints” de McGraw (2005, citado en UNIR, 2016a). Según el diccionario Oxford de inglés, el término “touchpoint” tendría, entre otras, como acepción la de “un tiempo, condición o circunstancia que es vulnerable o inestable como para precipitar un resultado altamente desfavorable, posiblemente devastador”. La propuesta de McGraw recoge de manera priorizada prácticas de seguridad que se pueden aplicar al SDLC, en cada uno de los artefactos (intangibles generados en el proceso de software). En concreto, ordenadas según su importancia para el autor, serían las siguientes:

1. Revisión del código, que se aplicaría a la codificación.
2. Análisis de riesgos. Aplicada a los requisitos, la arquitectura y el diseño y las pruebas.
3. Pruebas de penetración. En la fase de pruebas y producción.
4. Pruebas basadas en riesgo. Durante el plan de pruebas.

5. Casos de abuso. Aplicable a la fase de definición de los requisitos.
6. Requisitos de seguridad. También aplicable a los requisitos.
7. Operaciones de seguridad. Fase de producción.

Por otro lado, dado el tipo de arquitectura prevista, teníamos que considerar los riesgos propios de aplicaciones on-line. Los potenciales atacantes pueden explotar vulnerabilidades o debilidades en la seguridad a través de diversas “rutas” que comprometerían activos o afectarían a la confidencialidad, integridad o disponibilidad de la información y los servicios. Rutas de ataque que combinan un determinado agente de amenaza, vector de ataque y vulnerabilidad en la seguridad, que conllevan un impacto técnico y en el negocio (OWASP, 2013), como puede observarse en la siguiente figura.

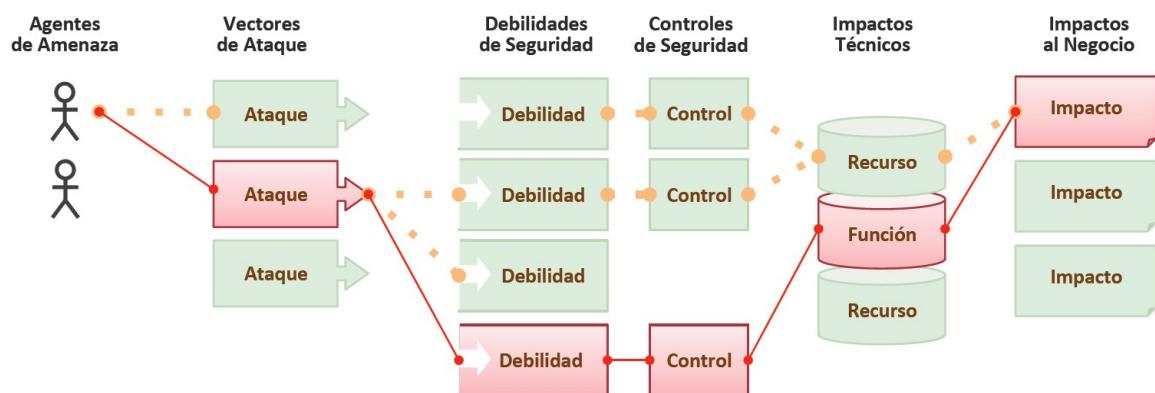


Figura 8: Riesgos de seguridad en aplicaciones (OWASP, 2013, p. 5)

Para identificar dichos riesgos seguimos los documentos publicados por la fundación OWASP (The Open Web Application Security Project), en particular los “Top 10”, que reúnen, en un listado que se esfuerzan por mantener al día, los diez riesgos de seguridad más comunes en este tipo de aplicaciones. Según el que publicaron en 2013 (OWASP, 2013) son los siguientes:

- A1 - Inyección.
- A2 - Pérdida de autenticación y gestión de sesiones.
- A3 - Secuencia de Comandos en Sitios Cruzados (XSS).
- A4 - Referencia directa insegura a objetos.
- A5 - Configuración de seguridad incorrecta.
- A6 - Exposición de datos sensibles.

-
- A7 - Ausencia de control de acceso a las funciones.
 - A8 - Falsificación de peticiones en sitios cruzados (CSRF)
 - A9 - Uso de componentes con vulnerabilidades conocidas.
 - A10 - Redirecciones y reenvíos no validados.

Enunciamos, como complemento, los cambios recogidos en la versión “Release Candidate 1” del Top 10 de 2017 (OWASP, 2017a), cuya publicación en versión definitiva se prevé para el mes de agosto. Con respecto al publicado en 2013 cambian los riesgos A4, A7 y A10, siendo sustituidos por los siguientes (A7 y A10 son nuevas categorías):

- A4 - Rotura en el control de acceso.
- A7 - Protección insuficiente contra ataques (automatizados o manuales).
- A10 - APIs desprotegidas.

Además hay que aclarar que, si bien estos son los más habituales, existen más riesgos de seguridad que pueden afectar a una aplicación web. En este sentido, seguimos como guía el Estándar de Verificación de Seguridad en Aplicaciones (ASVS por sus siglas en inglés), que constituye “un esfuerzo comunitario por establecer un marco de referencia para los requisitos de seguridad, controles funcionales y no funcionales necesarios al diseñar, desarrollar y testear aplicaciones web modernas” (OWASP, 2017b, p.7).

Como herramienta de apoyo utilizamos OWASP Zed Ataque Proxy (ZAP), con la que ya tuvimos oportunidad de trabajar como parte de las actividades prácticas del máster, y que sigue el ASVS para la detección de vulnerabilidades en aplicaciones web.

3.3. Objetivos

Los objetivos que nos propusimos alcanzar son los siguientes:

- Realizar el análisis y diseño del sistema, empleando métodos y técnicas de modelado para describirlo.
- Conseguir un prototipo o aplicación operativa conforme a los requerimientos funcionales y no funcionales que se establezcan.
- Desplegar el sistema para comprobar su funcionamiento en un entorno real de producción.

-
- Procurar que la implementación y el despliegue de la aplicación siga criterios de desarrollo seguro (S-SDLC), así como proteger frente a las amenazas propias de aplicaciones on-line.

4. Desarrollo de la aplicación

El enfoque que adoptamos para el desarrollo de la aplicación es el de “orientación a objetos”, ya que entendemos que el dominio (el problema planteado y su solución) puede ser susceptible de ser representado por objetos software (instancias de clases representativas) que mantienen relaciones de colaboración entre sí, con características o elementos estructurales y operaciones o comportamientos definitorios. Además, queremos aprovechar las ventajas que este enfoque puede aportarnos en cuanto a: (1) facilitar el análisis del dominio a un alto nivel de abstracción, (2) mejorar la comunicación entre personas que puedan intervenir en el desarrollo, (3) tolerancia a cambios en el sistema, (4) facilitar la reutilización, (5) favorecer la comprobación de su adecuación a los requisitos y (6) beneficiar las posibilidades de ampliar la funcionalidad (UNIR, 2016b).

Para definir las funcionalidades concretas del sistema y tener una visión de conjunto del mismo así como de los diferentes subsistemas que lo componen y su funcionamiento, modelamos paulatinamente el sistema para describirlo, como labor previa a la programación. Para ello, combinamos métodos de análisis (dominio del problema) y diseño (dominio de la solución) orientado a objetos (OOA/D, del inglés *Object-Oriented Analysis and Design*). Este enfoque se centra en la definición de clases y el modo en que colaboran entre sí para alcanzar los requerimientos de la aplicación (Larman, 2004, Pressman, 2010).

La fase de implementación se llevó a cabo mediante la utilización de entornos de desarrollo para facilitar la programación. En el lado cliente se desarrolló una aplicación nativa para Android, que se codificó en Java. En cuanto al servicio web, el lenguaje de programación fue PHP y se utilizaron herramientas de administración de bases de datos. Como práctica de seguridad se realizó una revisión de código mediante una herramienta de tipo SAST (*Static Analysis Security Tools*).

4.1. Especificaciones

En el apartado dedicado a la *justificación*, así como en el de *enfoque del concepto de la aplicación*, realizamos un pormenorizado planteamiento del problema y determinamos las especificaciones generales del sistema. Para no repetirlos aquí, de forma abreviada recordamos que el sistema implementado tiene que facilitar el contacto entre usuarios (personas mayores de 60 años) interesados en realizar alguna actividad en común. Del

mismo modo, ha de permitir fijar encuentros entre los usuarios, así como modificarlos en cuanto a fecha, horario o lugar y notificar dichos cambios. Además, planteamos que un componente esencial del sistema sería una aplicación destinada a dispositivos móviles (smartphones) con sistema operativo Android.

En primera instancia, para iniciar la descripción del sistema, recogemos a continuación los requisitos funcionales y no funcionales y el modelo del sistema de información y su arquitectura.

4.1.1. Requisitos

A continuación, presentamos el catálogo de requisitos, definidos a partir de la especificación del problema. En primer lugar se recoge en la siguiente tabla los requisitos funcionales, relativos a los servicios o funcionalidades que la aplicación ha de aportar a los usuarios finales y, a continuación, los requerimientos no funcionales, es decir, las condiciones tecnológicas o de otra índole con las que tendrá que cumplir el desarrollo de la solución.

Tabla 2

Requerimientos funcionales

ID	DESCRIPCIÓN	PRIORIDAD
RF-01	<p>Altas y bajas de usuarios del sistema.</p> <p>Debe facilitar el alta y la baja del sistema a los usuarios.</p> <p>Sólo permitirá que se puedan registrar usuarios con una edad mínima de 60 años.</p>	Alta
RF-02	<p>Perfiles de usuario.</p> <p>Los usuarios han de poder completar perfiles en función de sus intereses u objetivos para compartir el tiempo libre.</p>	Media
RF-03	<p>Subir foto de perfil.</p> <p>Ha de facilitar que el usuario añada una foto al perfil.</p> <p>Como medida de seguridad se propone que la foto sólo pueda proceder de la cámara del terminal.</p>	Baja

RF-04	Detectar usuarios próximos geográficamente.	Alta
	El sistema ha de conseguir ubicar geográficamente a los usuarios y determinar la distancia física entre ellos.	
RF-05	Proponer actividades.	Alta
	Los usuarios han de poder realizar propuestas de actividades.	
RF-06	Posibilidades de contacto con otros usuarios.	Alta
	El sistema ha de informar acerca de usuarios próximos a su ubicación, priorizando proximidad física y similitud en el perfil de intereses.	
	El usuario ha de poder conocer los perfiles y las actividades propuestas por esos usuarios.	
RF-07	Avisos propuestas de actividades.	Media
	El sistema ha de notificar al usuario las nuevas propuestas de actividades de usuarios próximos a su ubicación.	
RF-08	Unirse (mostrar interés) por propuesta de actividad	Alta
	Ha de permitir al usuario que pueda unirse a las actividades propuestas por otros usuarios (mostrar que están interesados en la actividad).	
RF-09	Aviso de contacto.	Alta
	El sistema ha de notificar al usuario cuando otro usuario se quiera unir a alguna de las actividades que haya propuesto.	
RF-10	Acordar encuentro.	Alta
	Cuando algún usuario desee unirse a alguna de las propuestas de actividades de otro usuario, éste ha de poder sugerirle fecha, hora y lugar para encontrarse.	
	El sistema ha de avisar al usuario, si está de acuerdo ha de poder confirmar la cita, si no es así, podrá sugerir otra fecha, horario o lugar que deberá ser confirmada por el otro usuario.	

Este ciclo se repetirá hasta que se llegue a un acuerdo o bien alguno desestime participar en la propuesta de actividad.

RF-11 Gestión de la cita	Alta
Los usuarios han de poder notificar o solicitar cambios o cancelación de las citas acordadas.	

Tabla 3*Requerimientos no funcionales*

ID	DESCRIPCIÓN	PRIORIDAD
RNF-01	Cliente móvil La aplicación cliente, que contendrá la interfaz gráfica de usuario, se programará para el sistema operativo Android. Ha de ejecutarse en terminales móviles que dispongan de GPS, cámara, pantalla táctil y conectividad a Internet.	Alta
RNF-02	Servidor web Los servicios web y la base de datos que suministrarán los datos a la aplicación cliente estarán alojados en un servidor accesible a través de Internet.	Alta
RNF-03	Seguridad Se seguirán pautas de S-SDLC en el desarrollo del código, así como medidas de seguridad específicas para mitigar amenazas de aplicaciones web.	Media
RNF-04	Usabilidad y accesibilidad La interfaz de usuario ha de cumplir con criterios de accesibilidad y usabilidad.	Baja

4.1.2. Modelo del sistema de información y arquitectura

La arquitectura de software define de qué forma se identifican, localizan e interactúan los diferentes elementos que componen el sistema de información. En el caso del sistema desarrollado, se trata una arquitectura de “aplicación distribuida”, esto es, “está formada por una colección de ordenadores autónomos enlazados por una red de ordenadores y soportados por un software que posibilita que la colección actúe como un servicio integrado” (Barceló, Iñigo y Llorente, 2008, p. 50). En concreto, se adecúa a un modelo cliente – servidor. En este tipo de modelo estos dos componentes actúan de la siguiente forma: los clientes hacen peticiones de servicio, habitualmente iniciando la comunicación con el servidor, y éste, a su vez, provee servicios, permaneciendo a la espera de las correspondientes peticiones, cuando reciben una la resuelven y responden al cliente. Este proceso se ilustra en la figura a continuación.

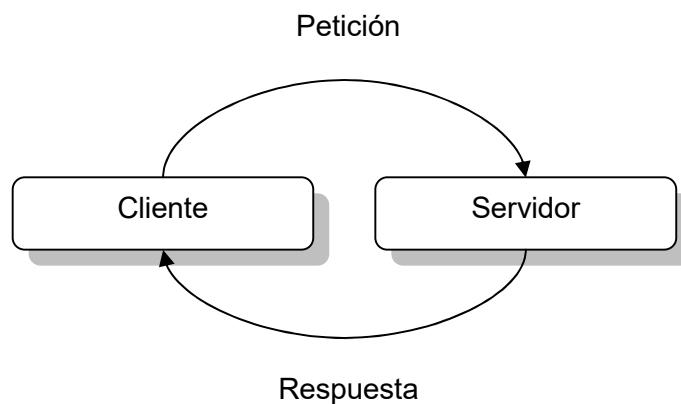


Figura 9: Modelo cliente – servidor

En particular, el sistema propuesto consiste en una aplicación cliente pensada para ser ejecutada en dispositivos móviles que se conecta a través de Internet con un servidor en el que residen un servicio web que es invocado a través del protocolo HTTP y un gestor de base de datos.

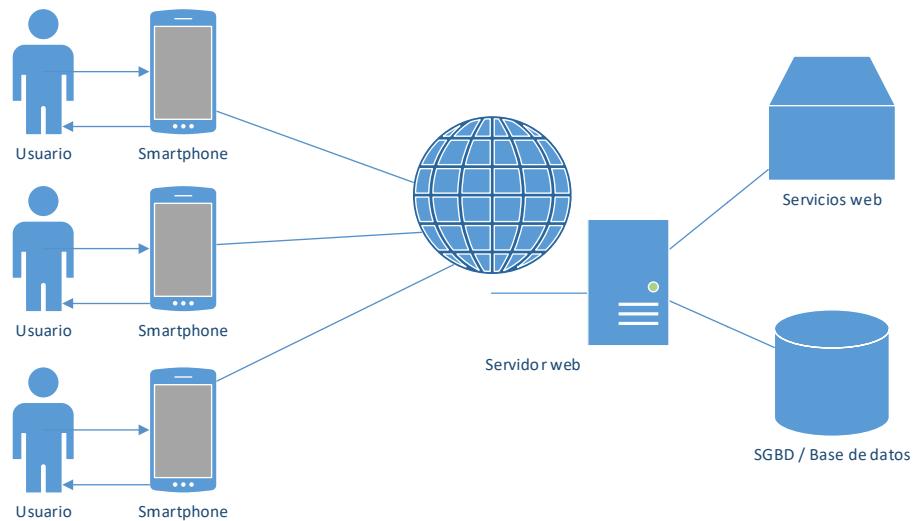


Figura 10: Arquitectura del sistema

En la figura siguiente se representa en más detalle las diferentes capas del sistema. Las líneas discontinuas representan los “límites de confianza”, es decir, muestran las zonas donde los niveles de confianza en cuanto a las potenciales amenazas de seguridad cambian; también se representan los elementos esenciales del sistema y sus diferentes capas: el cliente (aplicación Android), el servidor web y la capa de permanencia (el sistema gestor de base de datos o SGBD y la base de datos SQL).

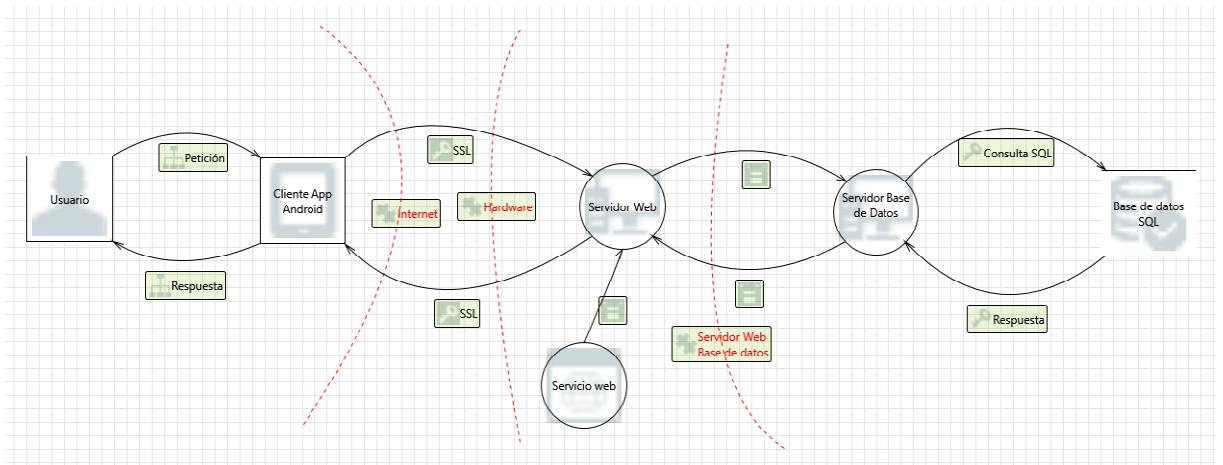


Figura 11: Arquitectura del sistema (detalle capas)

4.2. Análisis y diseño del sistema

En este apartado recogemos la labor realizada para describir el sistema de información propuesto, de forma previa a su implementación. Para ello, empleamos tanto métodos de análisis, en los que profundizamos en el dominio del problema y los requerimientos, como de diseño, en los que tratamos el dominio de la solución, de forma análoga al proceso que representa la figura siguiente:

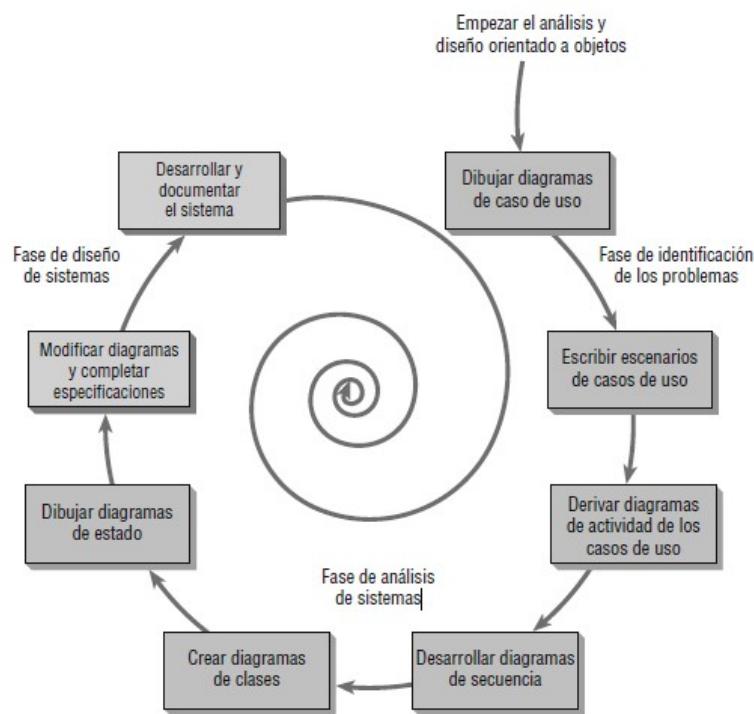


Figura 12: Proceso de análisis y diseño de sistemas (Kendall y Kendall, 2011, p. 18)

El análisis y diseño de sistemas “busca comprender qué necesitan los humanos para analizar la entrada o el flujo de datos de manera sistemática, procesar o transformar los datos, almacenarlos y producir información en el contexto de una organización específica” (Kendall y Kendall, 2011, p. 6). En línea con la idea de Larman (2004), tenemos como objetivos tanto identificar y resolver los problemas correctos (análisis), como describir el sistema para abordar su desarrollo de forma adecuada (diseño). Además, cabe recordar los beneficios añadidos del análisis y diseño en cuanto a documentar, comunicar y hacer más comprensible la estructura y funcionamiento del sistema, así como posibilitar futuros cambios y evoluciones.

El enfoque OOA/D (de las siglas en inglés correspondientes a Análisis y Diseño Orientados a Objetos) que adoptamos nos aporta una ventaja esencial en esta labor, ya que tanto en una fase como en la otra podemos utilizar la misma notación para representar los objetos y sus relaciones. A través de los diferentes diagramas y modelos pretendemos comprender lo mejor posible el sistema a desarrollar y, para conseguirlo, desplegamos los casos de uso para modelar las clases de análisis y de diseño y su comportamiento.

4.2.1. Análisis del sistema

A partir de las especificaciones, para efectuar el análisis del sistema utilizamos diferentes procedimientos, en concreto nos apoyamos en Diagramas de Flujo de Datos (DFD) para describir el sistema, diagramas y escenarios de casos de uso para modelar los requerimientos y, finalmente, el modelado de las clases de análisis que podemos trazar hasta las correspondientes clases de diseño.

4.2.1.1. Descripción del sistema

Como primer paso para la descripción del sistema de información, utilizamos Diagramas de Flujo de Datos (DFD). Estos diagramas sirven a los propósitos del análisis del sistema, ya que “describen con la mayor generalidad posible las entradas, los procesos y las salidas del sistema (...) Podemos usar una serie de diagramas de flujo de datos en capas para representar y analizar con detalle los procedimientos del sistema más amplio” (Kendall y Kendall, 2011, p. 193).

Este tipo de diagramas nos hace más comprensible la complejidad del sistema descomponiéndolo en partes más pequeñas y en sus interacciones. Además, aporta otras ventajas, según Kendall y Kendall (2011), que nos resultan igualmente interesantes, tales como la libertad conceptual, la independencia con respecto a la plataforma tecnológica que se vaya a implementar, entender mejor la interrelación de los sistemas y subsistemas, facilitar la comunicación acerca del sistema y analizar si se han definido los datos y procesos necesarios; además los DFD de datos lógicos aportan sistemas más estables, flexibilidad y facilidad de mantenimiento y posibilitan eliminar redundancias; en cuanto a los DFD de datos físicos permiten distinguir qué procesos son operados por humanos (manuales) y cuáles se automatizan.

Comenzamos con un DFD de contexto en el que, a bajo nivel de detalle, se representa el conjunto del sistema:

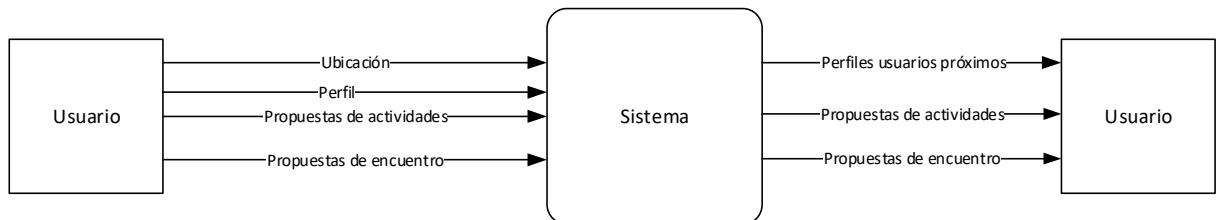


Figura 13: DFD de contexto

A continuación un diagrama con mayor nivel de detalle, en el que se recogen los procesos principales:

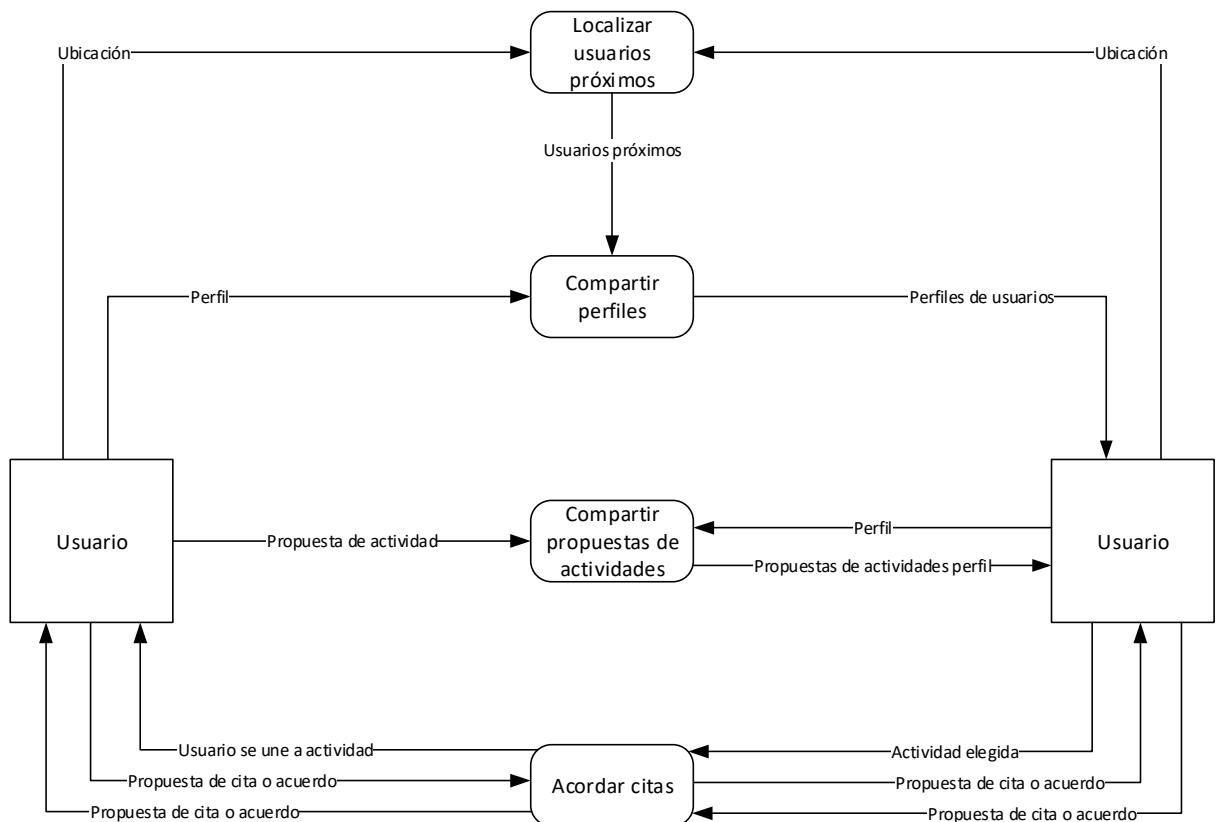


Figura 14: DFD Nivel 0 – Procesos principales

Proseguimos ampliando el nivel de detalle, representamos a continuación el proceso “Localizar usuarios próximos”:

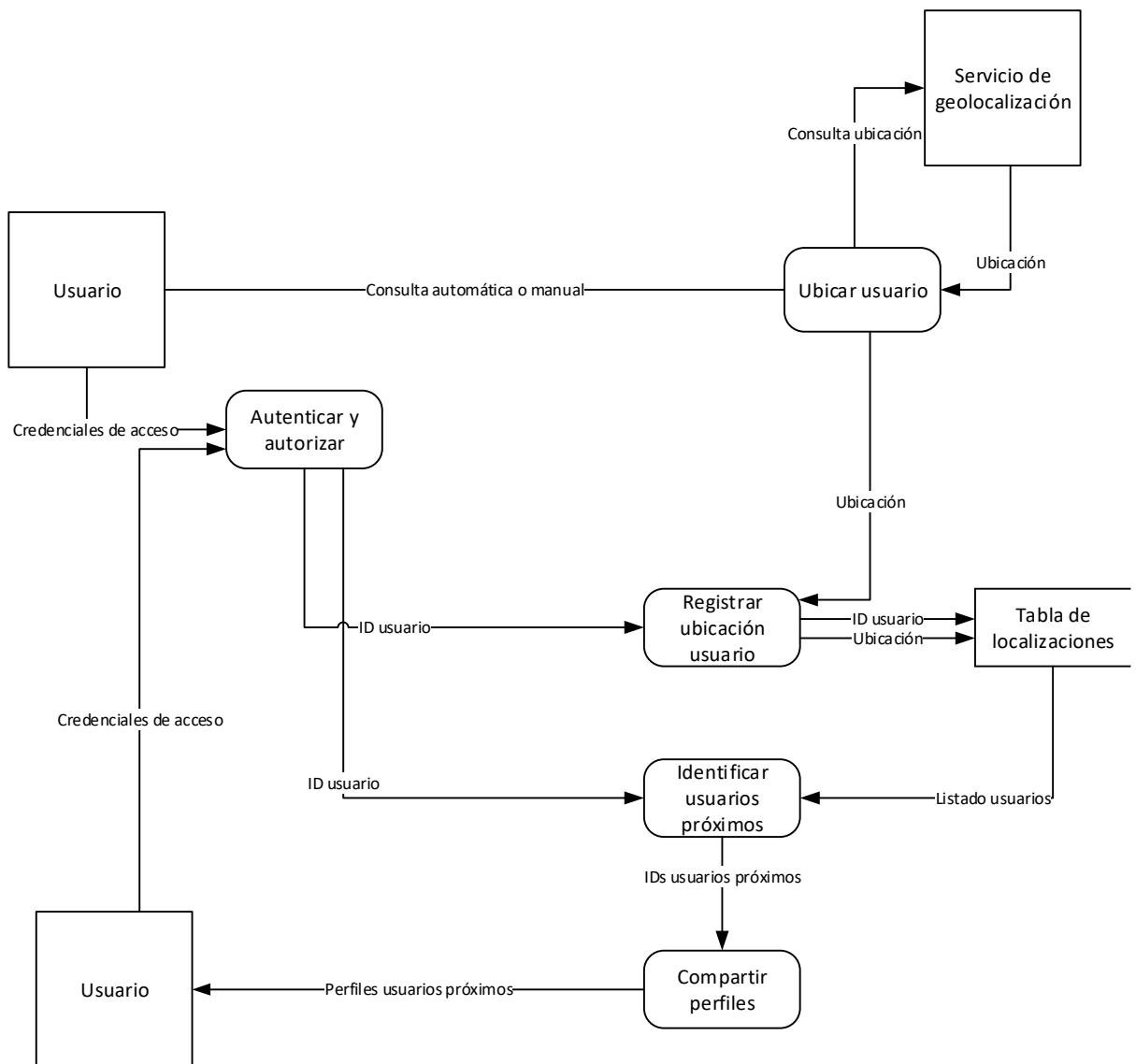


Figura 15: DFD Nivel 1 – Proceso Localizar usuarios próximos

Con similar nivel de detalle nos adentramos en el proceso “Compartir perfiles”:

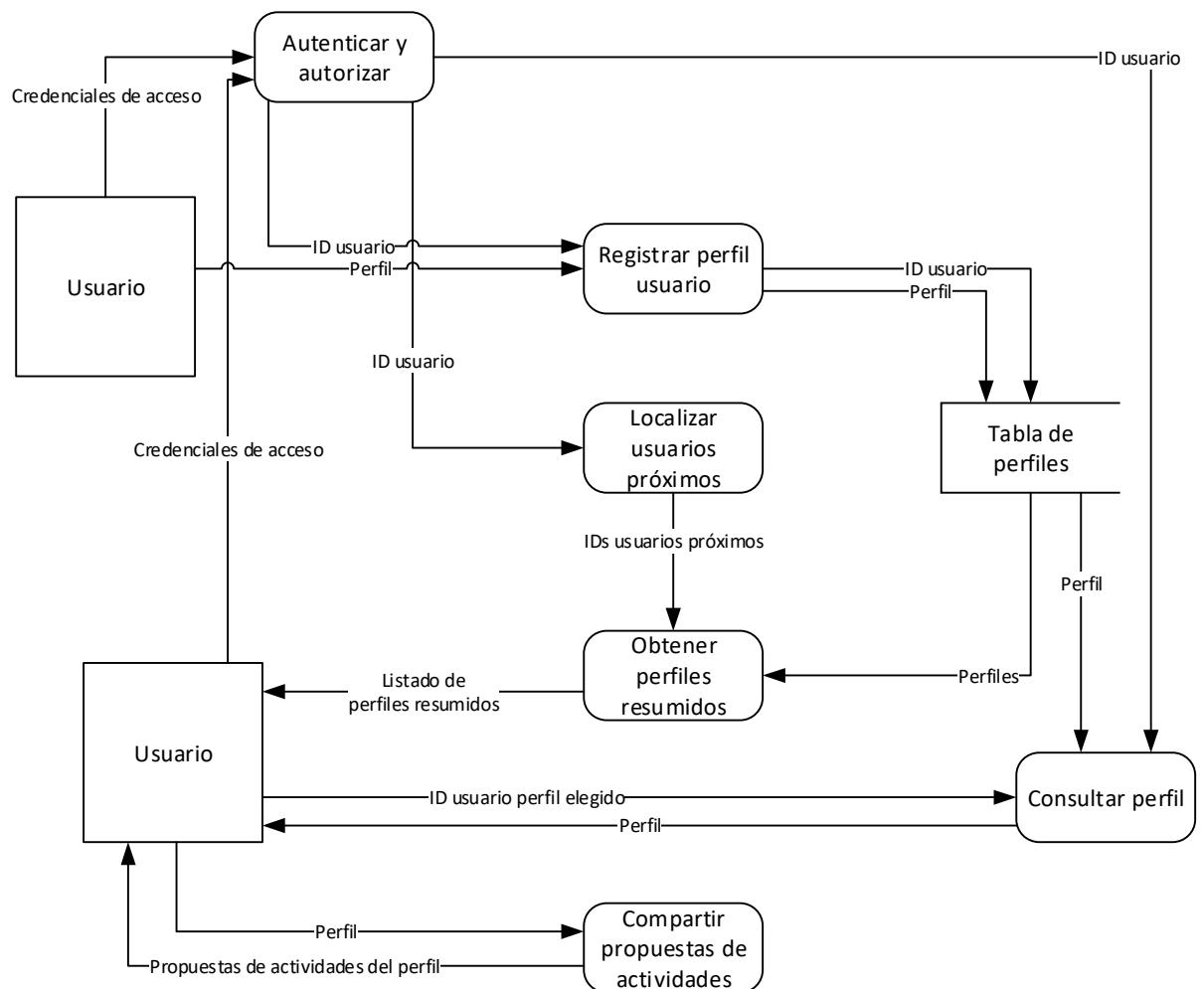


Figura 16: DFD Nivel 1 – Proceso Compartir perfiles

Seguimos con el proceso “Compartir propuestas de actividades”:

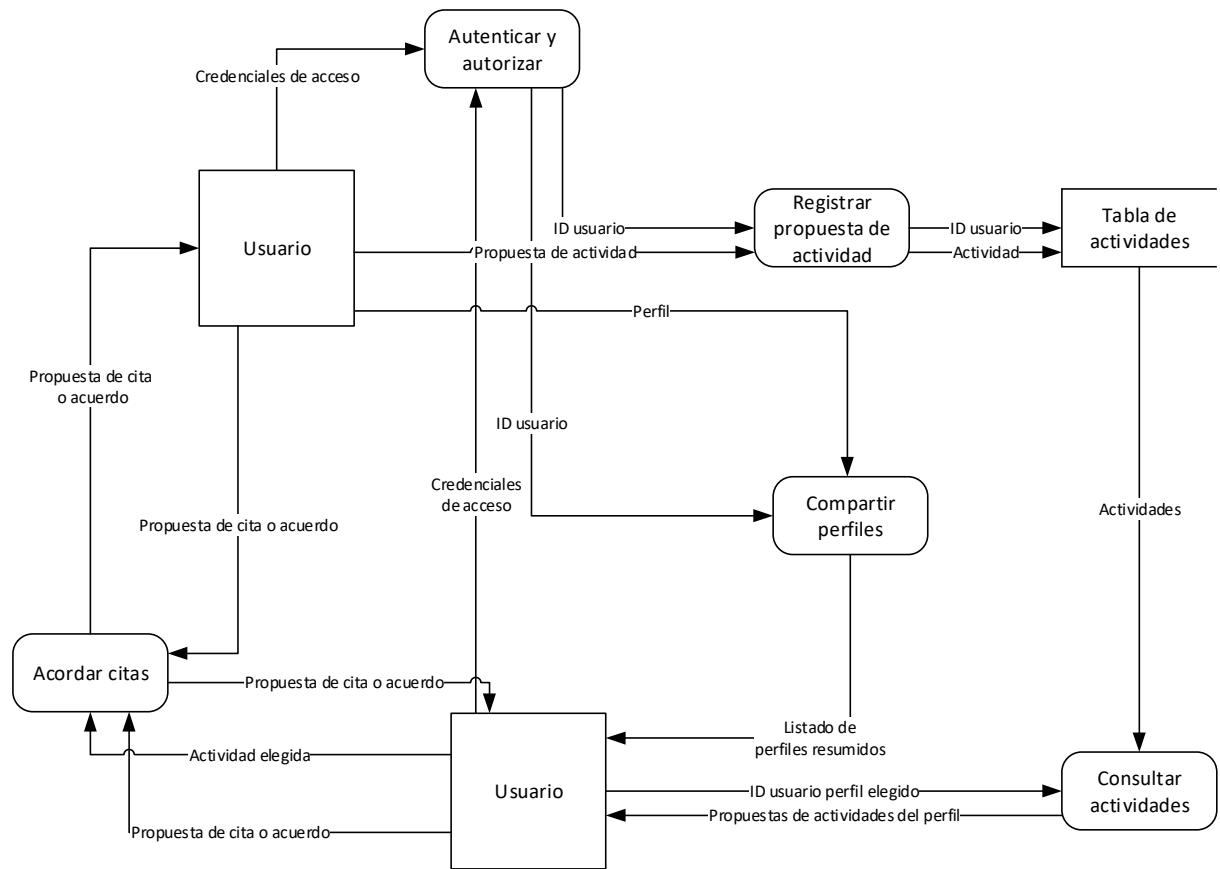


Figura 17: DFD Nivel 1 – Proceso Compartir propuestas de actividades

Continuamos con otro de los procesos principales, “Acordar citas”:

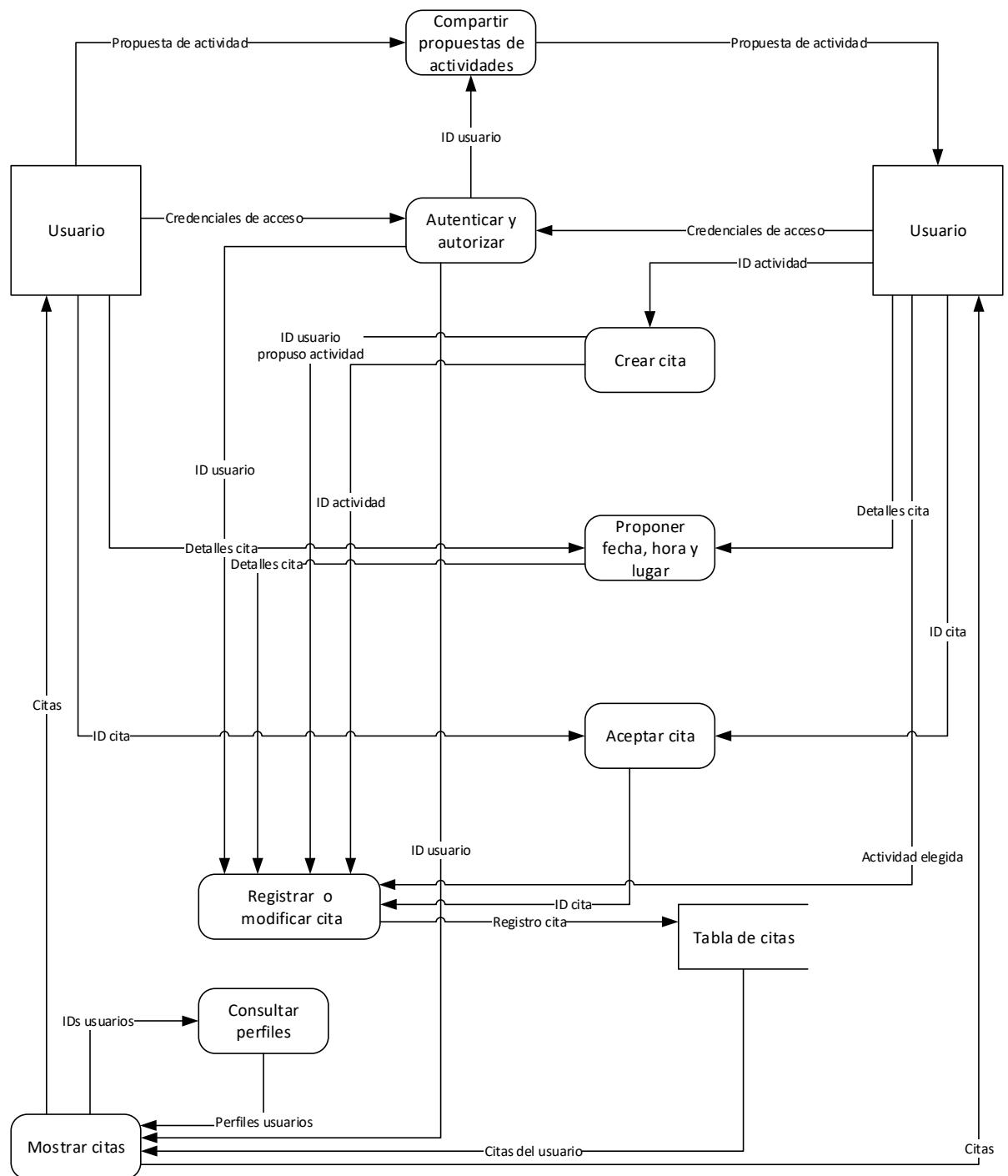


Figura 18: DFD Nivel 1 – Proceso Acordar citas

Por último, aunque no figura entre los procesos principales del sistema en el DFD de Nivel 0, representamos a continuación los DFD correspondientes a los procesos de control de acceso (autenticación y autorización) y de registro de usuarios:



Figura 19: DFD Nivel 1 – Proceso Autenticar y autorizar

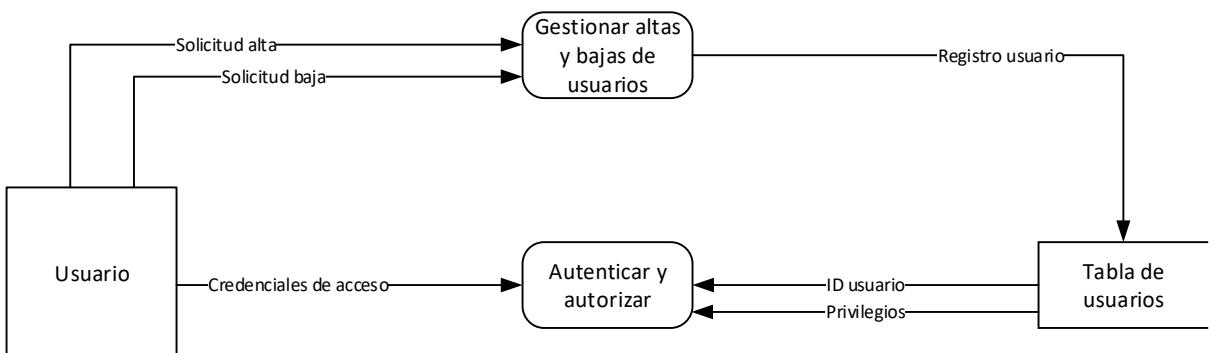
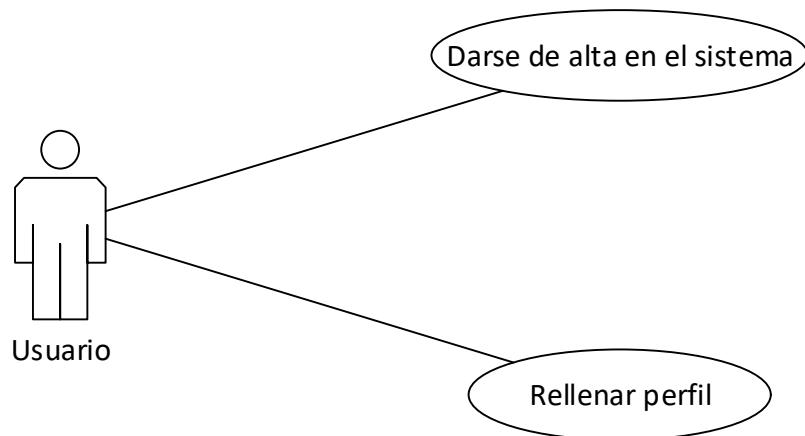


Figura 20: DFD Nivel 1 – Proceso Registro usuarios

4.2.1.2. Casos de uso

A continuación, como recurso para continuar con el análisis del sistema, empleamos diagramas de casos de uso y sus correspondientes escenarios. Recordamos que un caso de uso “describe la manera en la que un usuario interactúa con el sistema, definiendo los pasos requeridos para lograr una meta específica” (Pressman, 2010, p. 730).

En favor de la claridad, hemos desglosado en tres grupos los casos de uso, el primero se refiere a los primeros pasos del usuario con el sistema, alta y elaboración del perfil; el segundo alude a los procesos destinados a conocer a otros usuarios próximos geográficamente, así como las propuestas de actividades que esos usuarios han formulado. El tercer grupo se refiere a las acciones que permiten a los usuarios tanto proponer actividades como unirse a las propuestas por otros usuarios, así como acordar el encuentro para la realización de la actividad.

*Figura 21: Diagrama de caso de uso Alta en el sistema y elaboración del perfil***Tabla 4***Escenario de CU Registro de usuario*

Identificador	CU – 01
Nombre	<i>Registro de usuario</i>
Descripción	Permitir que una persona pueda registrarse en el sistema para operar con la app.
Precondición	La persona ha instalado la app en su móvil, pero no está activada (no tiene cuenta en el sistema o, si ya tiene una, no se ha confirmado que se ha instalado en un móvil del mismo usuario) y cumple con los requisitos para ello (condiciones de uso, edad, ciudad).
Postcondición	La persona ha creado una nueva cuenta o ha validado la existente en el sistema y la app pasa a estar activada.
Actores e intereses	<i>Usuario</i> . Quiere activar y operar con la app y utilizar su cuenta de usuario.
Secuencia	Acción (pasos)
Normal	<ol style="list-style-type: none"> El usuario instala la app en su móvil. Se le solicitan los permisos para que funcione correctamente. Se le pregunta si es mayor de 60 años, si no lo fuera no continúa, se le informa de que la aplicación es exclusiva para usuarios de esa franja de edad. Se le solicita el número de teléfono para crear nueva cuenta o bien registrarse con datos de cuenta si ya tuviera una creada. El sistema comprueba si el número introducido tiene el formato de un número de teléfono, en caso de que haya alguna incorrección iría al paso 1 del flujo alternativo. El sistema envía al usuario un SMS de validación. Si el sistema recibe confirmación de que el número de teléfono es correcto, procede al registro del nuevo usuario y la app pasa a estar operativa y el usuario logueado. La aplicación muestra la pantalla de bienvenida y, a continuación, la de creación del perfil.

Secuencia alternativa	Acción (pasos)
	<ol style="list-style-type: none"> Se le indica al usuario que no es un número de teléfono válido, vuelve al paso 4 del flujo de la secuencia normal.
Secuencia de error	Acción (pasos)
	<ol style="list-style-type: none"> Si no recibe el SMS transcurrido un margen de tiempo solicitar el envío de otro mensaje de validación.
Importancia	Alta
Urgencia	Alta
Observaciones	El usuario queda identificado por su número de teléfono. Hasta que no se haya “activado” (justo cuando el número de teléfono del móvil del usuario haya sido validado), la aplicación aparece como “no activada” y no se puede utilizar más allá de la pantalla de bienvenida y la destinada a introducir el número de teléfono.

Tabla 5*Escenario de CU Rellenar perfil*

Identificador	CU – 02
Nombre	<i>Cumplimentación del perfil</i>
Descripción	Facilita que el usuario pueda crear o modificar el perfil que se mostrará.
Precondición	El usuario tiene cuenta, la app está activa y el usuario logueado.
Postcondición	El usuario consigue elaborar o modificar su perfil. Se obtiene la ubicación del usuario.
Actores e intereses	<i>Usuario</i> . Quiere compartir un perfil para darse a conocer a otros usuarios.
Secuencia Normal	Acción (pasos)
	<ol style="list-style-type: none"> El usuario accede a la pantalla del perfil. Si cuenta ya con un perfil, puede modificarlo (paso 1 de la secuencia alternativa), si no iniciaría la creación de uno. Incorpora la información al perfil: nombre, intereses, adicionalmente foto. Tras informar al usuario, se incorpora su ubicación al perfil.
Secuencia alternativa	Acción (pasos)
	<ol style="list-style-type: none"> El usuario puede seleccionar el apartado del perfil que desea modificar. Se muestra el apartado con los datos actuales, el usuario puede entonces modificarlos.

Secuencia de error	Acción (pasos)
	<ol style="list-style-type: none"> 1. Si se detecta algún error o valor no válido se indica al usuario que vuelva a introducir el dato. 2. Si no se ha obtenido su ubicación, facilitar que el usuario pueda pedirla de nuevo de forma manual. 3. Si no funciona el acceso al servicio de geolocalización solicitar localidad de residencia al usuario (se mostrará en el perfil como "no verificada").
Importancia	Alta – cumplimentación inicial Media – modificación
Urgencia	Alta – cumplimentación Baja - modificación
Observaciones	Al perfil se le añade la ubicación, si no se hubiera podido ubicar, constará en el perfil como "ubicación no verificada".

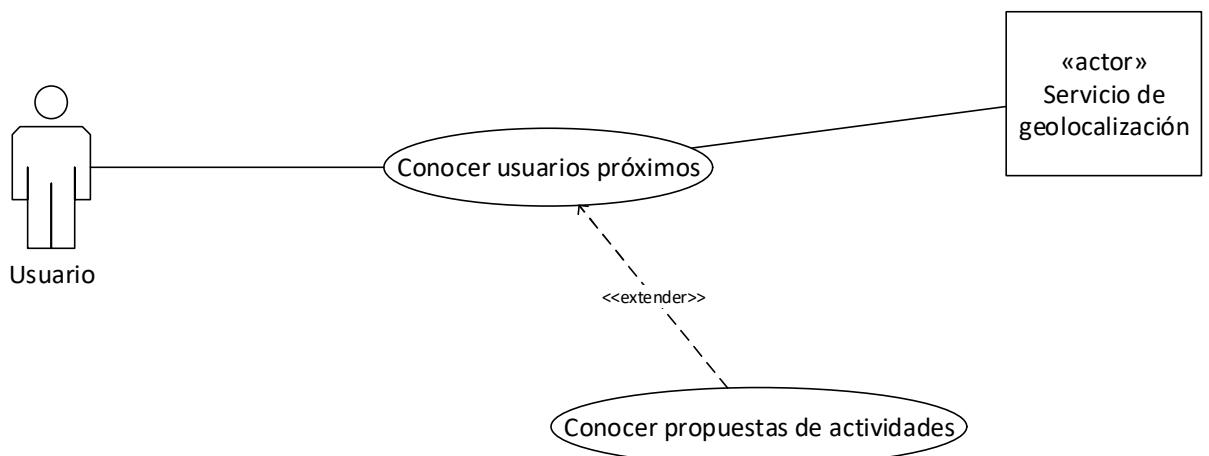


Figura 22: Diagrama de caso de uso *Conocer perfiles y actividades*

Tabla 6

Escenario de CU *Conocer usuarios próximos*

Identificador	CU – 03
Nombre	<i>Perfiles próximos</i>
Descripción	Facilita que el usuario pueda conocer los perfiles de usuarios próximos geográficamente.
Precondición	Tiene que haber cumplido con las postcondiciones de CU-01 y CU-02 (usuario con cuenta, app activa, logueado y con perfil propio cumplimentado) y haberse obtenido su ubicación a través del servicio de geolocalización.
Postcondición	El usuario consigue acceder a los perfiles de usuarios próximos.
Actores e intereses	<i>Usuario</i> : Ha compartido su perfil para darse a conocer a otros usuarios y, a su vez, quiere conocer a otros usuarios próximos.

Secuencia Normal	Acción (pasos)
	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de contacto con otros usuarios. 2. Se muestra un listado con información resumida de usuarios próximos. 3. Puede acceder al perfil completo del usuario que seleccione. 4. En el perfil seleccionado puede acceder a las actividades propuestas por el usuario.
Secuencia alternativa	Acción (pasos)
	<ol style="list-style-type: none"> 1. En la pantalla contacto con otros usuarios puede acceder a una muestra de actividades propuestas por usuarios próximos. 2. Puede pulsar sobre la actividad que le interese para acceder a la misma y, desde ella, al perfil de usuario del que partió la propuesta.
Secuencia de error	Acción (pasos)
	<ol style="list-style-type: none"> 1. Avisar si no se encuentran usuarios próximos si quiere acceder a otros que se encuentren a mayor distancia.

Importancia Alta

Urgencia Alta

Observaciones

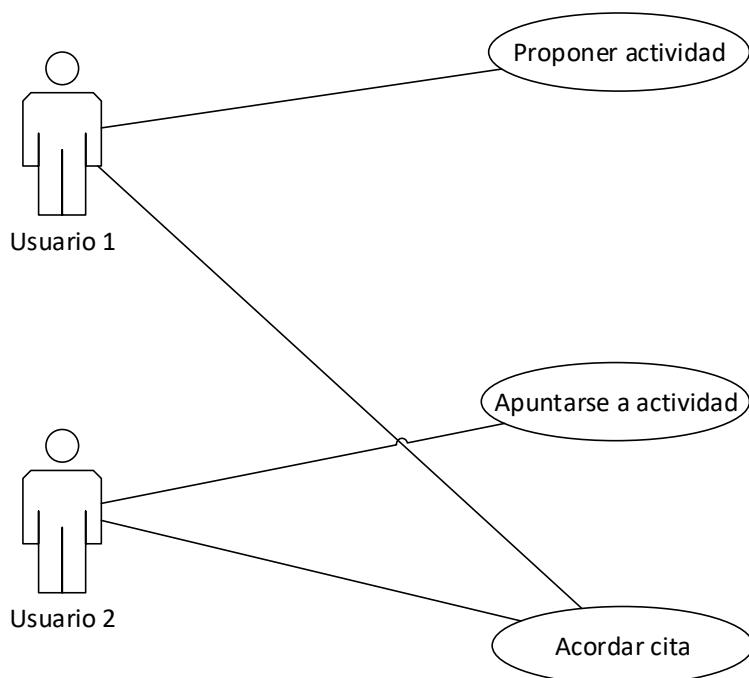


Figura 23: Diagrama de caso de uso *Propuestas de actividades y acuerdos de encuentro*

Tabla 7*Escenario de CU Proponer actividad*

Identificador	CU – 04
Nombre	<i>Propuesta de actividad</i>
Descripción	Permite que el usuario pueda sugerir actividades en común.
Precondición	Tiene que haber cumplido con las postcondiciones de CU-01 y CU-02 (usuario con cuenta, app activa, logueado, con perfil propio cumplimentado y ubicado geográficamente).
Postcondición	El usuario comparte la propuesta de actividad desde su perfil.
Actores e intereses	<i>Usuario:</i> Ha compartido su perfil para darse a conocer a otros usuarios y propone actividad/es para conocer a otros usuarios próximos.
Secuencia	Acción (pasos)
Normal	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de actividades, elige proponer actividad. 2. Selecciona una de las actividades sugeridas o bien puede cumplimentar un campo para proponer una inédita. 3. También puede modificar o eliminar las actividades creadas con anterioridad (secuencia alternativa).
Secuencia alternativa	Acción (pasos) <ol style="list-style-type: none"> 1. El usuario puede acceder a una pantalla para eliminar o modificar actividades propuestas con anterioridad.
Secuencia de error	Acción (pasos) <ol style="list-style-type: none"> 1. Si se detecta algún error o valor no válido se indica al usuario para que vuelva a introducir el dato.
Importancia	Alta
Urgencia	Alta
Observaciones	Las actividades son el punto de conexión con otros usuarios. Dos usuarios “conectan” cuando uno se une a la actividad propuesta por el otro.

Tabla 8*Escenario de CU Apuntarse a actividad*

Identificador	CU – 05
Nombre	<i>Apuntarse a actividad</i>
Descripción	Permite que el usuario pueda “unirse”, mostrar su interés por quedar para realizar una actividad propuesta por otro usuario.

Precondición	Tiene que haber cumplido con las postcondiciones de CU-01, CU-02 y CU-03 (usuario con cuenta, app activa, logueado, con perfil propio cumplimentado y ubicado geográficamente, y haber accedido al perfil del usuario que propone la actividad).
Postcondición	El usuario muestra su interés al usuario del que partió la propuesta por realizarla con él/ella.
Actores e intereses	<i>Usuario:</i> Se une a la/s actividad/es propuestas por otros usuarios próximos para poder conocerlos en persona.
Secuencia Normal	<p>Acción (pasos)</p> <ol style="list-style-type: none"> 1. Desde la pantalla de perfiles de usuarios próximos, selecciona perfil y después la actividad sugerida. 2. Se muestra la pantalla de la actividad. 3. Pulsa botón (“me interesa”) para “unirse” a la actividad. 4. Recibe mensaje de confirmación. 5. En la pantalla de inicio, al usuario que realizó la propuesta de la actividad, se le notifica que existe un usuario interesado en unirse a la misma. La secuencia continuaría en el caso de uso Programación de cita.
Secuencia alternativa	<p>Acción (pasos)</p> <ol style="list-style-type: none"> 1. En la pantalla contacto con otros usuarios puede acceder a una muestra de actividades propuestas por usuarios próximos. 2. Puede pulsar sobre la actividad que le interese para acceder a la misma y, desde ella, al perfil de usuario del que partió la propuesta. Continúa en el paso 2 de la secuencia normal.
Secuencia de error	<p>Acción (pasos)</p> <ol style="list-style-type: none"> 1. Si la actividad es eliminada figurará como “eliminada por el autor”. 2. No puede apuntarse 2 veces a la misma actividad, se avisaría que ya mostró su interés por la actividad.
Importancia	Alta
Urgencia	Alta
Observaciones	Si el usuario accede a la pantalla de actividades a las que se ha apuntado puede ver el estado (por ejemplo, pendiente de respuesta, cita pendiente de acuerdo, cita acordada).

Tabla 9*Escenario de CU Acordar cita*

Identificador	CU – 06
Nombre	<i>Programación de cita</i>
Descripción	Permite que dos usuarios puedan programar una cita para conocerse en persona y realizar la actividad propuesta por uno de ellos.
Precondición	Tiene que haber cumplido con las postcondiciones de CU-01, CU-02 y

Postcondición	CU-04 o CU-05 (usuario con cuenta, app activa, logueado, con perfil propio cumplimentado y ubicado geográficamente, y haber propuesto la actividad o bien haberse “unido” a hacerla).
Actores e intereses	Ambos usuarios consiguen programar un encuentro o bien alguno de ellos desiste y no se materializa. <i>Usuario:</i> Desea ponerse en contacto personal con otros usuarios para realizar actividades propuestas por él/ella o bien que le hayan sido sugeridas por otros usuarios.
Secuencia Normal	Acción (pasos) <ol style="list-style-type: none"> 1. En la pantalla de inicio, al usuario que realizó la propuesta de la actividad, se le notifica que existe un usuario interesado en unirse a la misma y se le pide que proponga fecha, hora y lugar de encuentro si no lo hubiera hecho ya. 2. El usuario accede a la pantalla de citas y puede programarla. 3. El usuario que se unió a la actividad recibe aviso de cita propuesta y puede aceptarla o bien, si lo desea, proponer otra fecha, hora y/o lugar, lo que nos lleva al paso 1 de la secuencia alternativa. 4. El usuario recibe notificación de que ha sido aceptada la cita y queda programada. 5. Una vez aceptada, si cualquiera de los usuarios desea cambiar la cita programada antes de que se cumpla, iría a la pantalla de citas, seleccionaría cambiar o cancelar y seguiríamos en el paso 1 de la secuencia alternativa.
Secuencia alternativa	Acción (pasos) <ol style="list-style-type: none"> 1. El usuario desea cambiar la cita programada o realizar una contra-propuesta. Pasa a pantalla para proponer otra fecha, hora y/o lugar. 2. El otro usuario recibe notificación de la contra-propuesta de cita, puede aceptarla, pasaríamos al paso 4 de la secuencia normal, o bien proponer otra, iríamos al paso 1 de la esta secuencia alternativa; o bien puede rechazar finalmente realizar la actividad (figuraría entonces en la actividad como “no acuerdo para cita”).
Secuencia de error	Acción (pasos) <ol style="list-style-type: none"> 1. Comprobar y avisar si se proponen fechas fuera de rango. 2. Si finalmente no se llega a acuerdo se puede desestimar realizar la actividad.
Importancia	Alta
Urgencia	Alta
Observaciones	Como parte de la descripción de la actividad los usuarios pueden hacer constar su disponibilidad para realizarla (o su preferencia en cuanto a fechas, horarios o lugares). Los usuarios podrían recibir notificación de recordatorios de citas próximas a su cumplimiento.

Siguiendo pautas de seguridad, a continuación modelamos posibles *casos de abuso*.

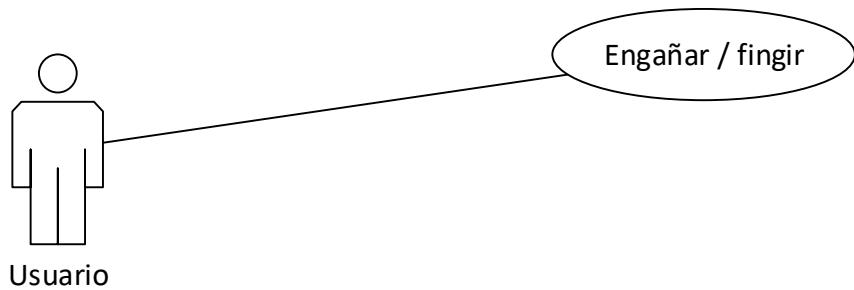


Figura 24: Diagrama de caso de abuso Usuario con falsa identidad o con el propósito de engañar a otros usuarios

El caso de abuso representado en la figura estaría relacionado con una situación en la que un usuario dado de alta en el sistema pretende abusar del mismo fingiendo ser otra persona o haciendo uso de una identidad falsa o inventada, o bien con unos propósitos diferentes a la finalidad y términos de uso de la aplicación.

Hemos de prevenir esta situación, por ejemplo mediante la solicitud de alguna prueba de identidad (como puede ser la foto realizada por el propio dispositivo), registro del número de teléfono real que pertenezca al usuario, obligación de aceptar las condiciones y términos de uso, en la que se concreten los usos legítimos y la limitación de edad, proporcionando mecanismos para que los usuarios puedan informar de actuaciones irregulares o potencialmente dañinas o fraudulentas.

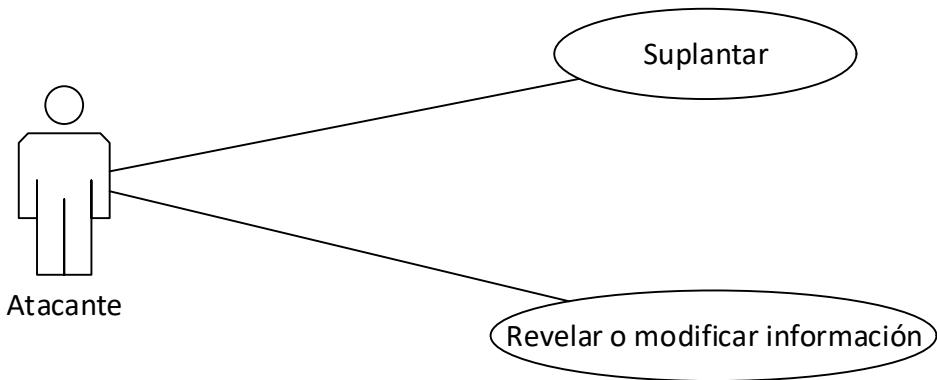


Figura 25: Diagrama de caso de abuso Atacante

Este caso de abuso estaría relacionado con atacantes que pudieran comprometer alguno de los activos, bien suplantando a un usuario legítimamente autorizado, bien por otro tipo de procedimientos que les sirviera para revelar o modificar la información (comprometer su

confidencialidad o su integridad). Para ello, hay que establecer controles de acceso (autenticación, autorización y seguimiento), fijar salvaguardas y controles de seguridad para prevenir los potenciales ataques (inyección de código, suplantación de identidad, desvelar variables de sesión, credenciales de acceso, etc.) y seguir pautas de desarrollo seguro (SSDLC) tal y como comentamos el apartado *Protección ante riesgos de seguridad*.

4.2.1.3. Clases de análisis

De manera consecuente con el enfoque adoptado de AOO (análisis orientado a objetos), el modelado que realizamos está basado en clases, esto es, “representa los objetos que manipulará el sistema, las operaciones (...) que se aplicarán a los objetos para efectuar la manipulación, las relaciones (...) entre los objetos y las colaboraciones que tienen lugar entre las clases definidas” (Pressman, 2010, p. 142). De tal modo que, como colofón del análisis del sistema, a continuación recogemos las *clases de análisis* identificadas. A partir de ellas procederemos a definir las clases de diseño, siendo trazables las unas desde las otras.

Del análisis de requisitos, trajimos los siguientes nombres en lenguaje natural: usuario, app, móvil, credenciales de acceso, condiciones de uso, edad, ciudad, cuenta, permisos, número de teléfono, SMS de validación, registro, perfil, ubicación, nombre, intereses, foto, servicio de geolocalización, localidad, actividad, propuesta de actividad, conexión, cita, fecha, hora, lugar de encuentro, aviso, propuesta / contra-propuesta, acuerdo, disponibilidad, preferencia.

Seleccionamos como posibles clases:

- Usuario. Este término lo hemos derivado en tres: Usuario, que sería toda persona que interactúe con el sistema, Cliente, que, en este caso, serían los que hemos aludido en los requisitos como usuarios, es decir, las personas que utilizarán los servicios provistos por el sistema y, por último, Administrador, que sería otro tipo de usuario, los responsables de gestionar o realizar operaciones de administración del sistema (incluimos la clase aunque no es objetivo del desarrollo inicial implementar el *backend*, o panel de administración, pero puede resultar útil documentarlo para futuras evoluciones).
- Ubicación.
- Perfil.

- Actividad.
- Cita.

Añadimos:

- Agenda. El conjunto de citas de un usuario, ya sean propuestas, pendientes de acuerdo o acordadas. Por tanto se refiere a la agenda de citas. Cada cita pertenecerá a dos agendas (ya que implica a dos usuarios) y cada agenda puede tener ninguna o varias citas.

Las clases se relacionarían de la siguiente forma: la clase Cliente se asocia a las clases Ubicación (cada cliente tiene una única ubicación), la clase Perfil (cada cliente tiene un solo perfil) y Agenda (cada usuario tiene una agenda propia), esta última se relaciona con Cita, ya que cada agenda recogerá las citas del usuario. Hay que aclarar que cada cita pertenecerá a dos agendas, puesto que constará en la agenda respectiva de cada uno de los dos usuarios implicados en la cita. La clase Cita, además, se relaciona con Actividad, ya que cada cita es para realizar una determinada actividad. A su vez, la clase Actividad se relaciona con la clase Perfil, ya que cada actividad se asociaría al perfil del usuario que la hubiera propuesto.

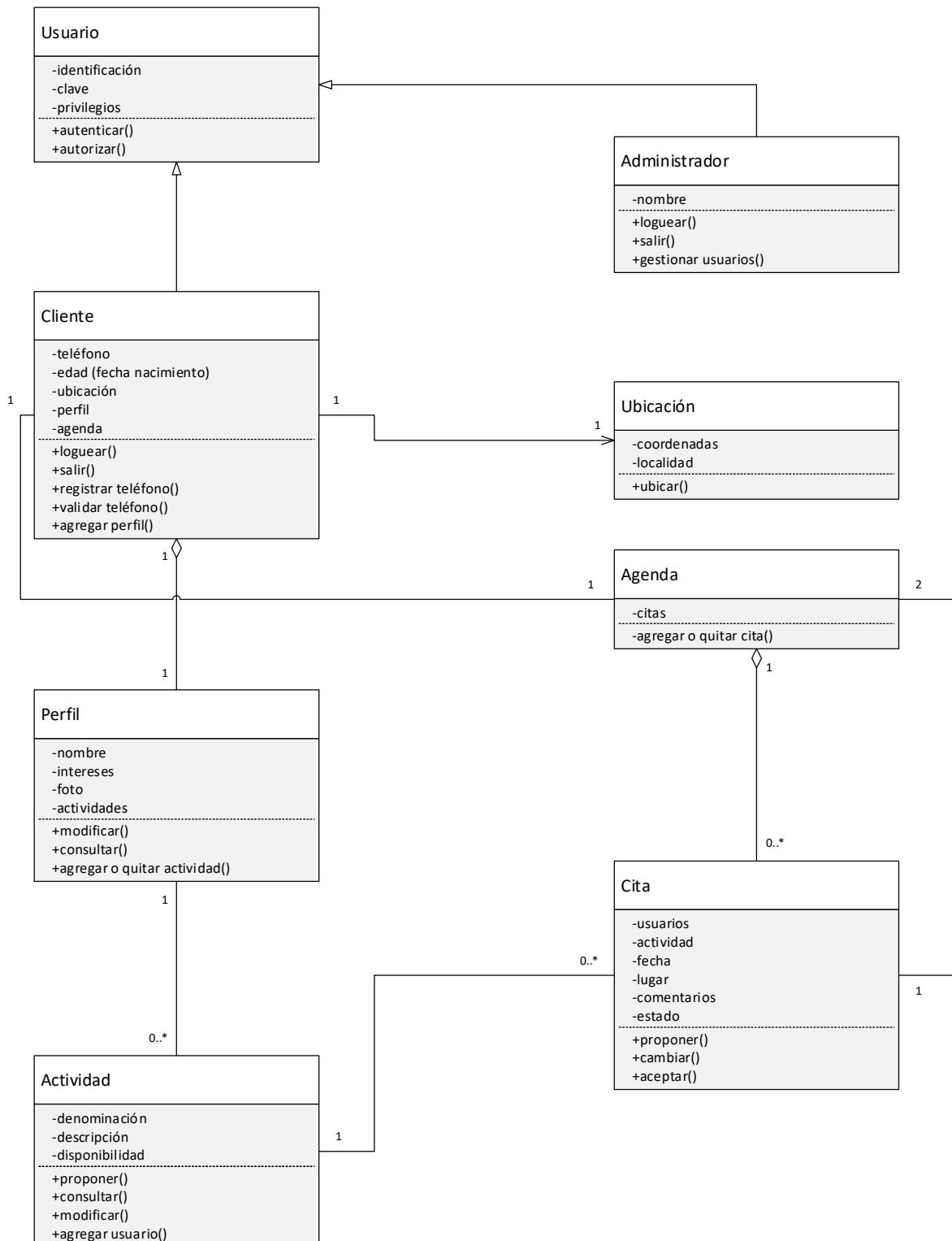


Figura 26: Diagrama de Clases de análisis

4.2.2. Diseño del sistema

El diseño “pone énfasis en una solución conceptual que satisface los requisitos, en vez de ponerlo en la implementación (...) Finalmente los diseños pueden ser implementados.” (Larman, 2004, p. 6). Como el enfoque que adoptamos es el de *diseño orientado a objetos* (OO), en esta fase nuestro objetivo se centra en definir los objetos y determinar cómo colaboran entre sí para cumplir los requerimientos del sistema.

Por otro lado, procuraremos aportar descripciones de los diferentes módulos que componen el sistema y sus dependencias.

En este apartado presentamos tanto una visión estática como dinámica del sistema, es decir, representamos tanto los componentes del sistema software como su comportamiento o interacción en el tiempo (secuencias). Todo ello con la finalidad de describir el sistema y como apoyo a la implementación.

4.2.2.1. Modelo de clases de diseño

A partir de las clases de análisis generamos las clases de diseño que son la base del sistema, tal y como se muestra en la siguiente figura:

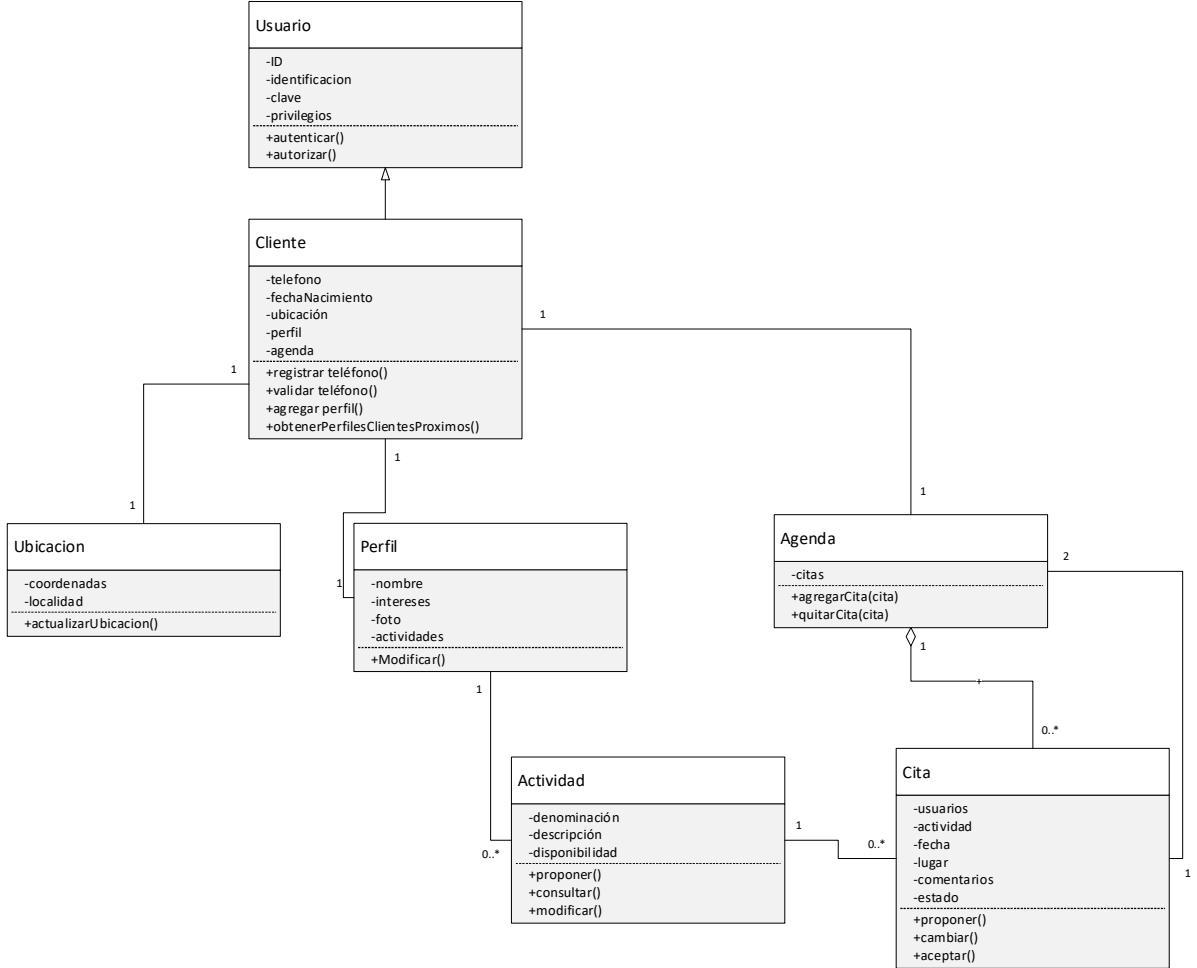


Figura 27: Diagrama de Clases de Diseño – Clases básicas

El diagrama de clases de la figura anterior podría ser útil para una arquitectura monolítica, con alto nivel de acoplamiento. Si tenemos en cuenta la arquitectura del sistema, y su división en capas, cliente (app) y servidor (servicio web, SGBD), nos parece oportuno especificar y detallar las clases de diseño resultantes para una mejor comprensión del sistema, de tal forma que las recogemos en diferentes diagramas, agrupadas en paquetes.

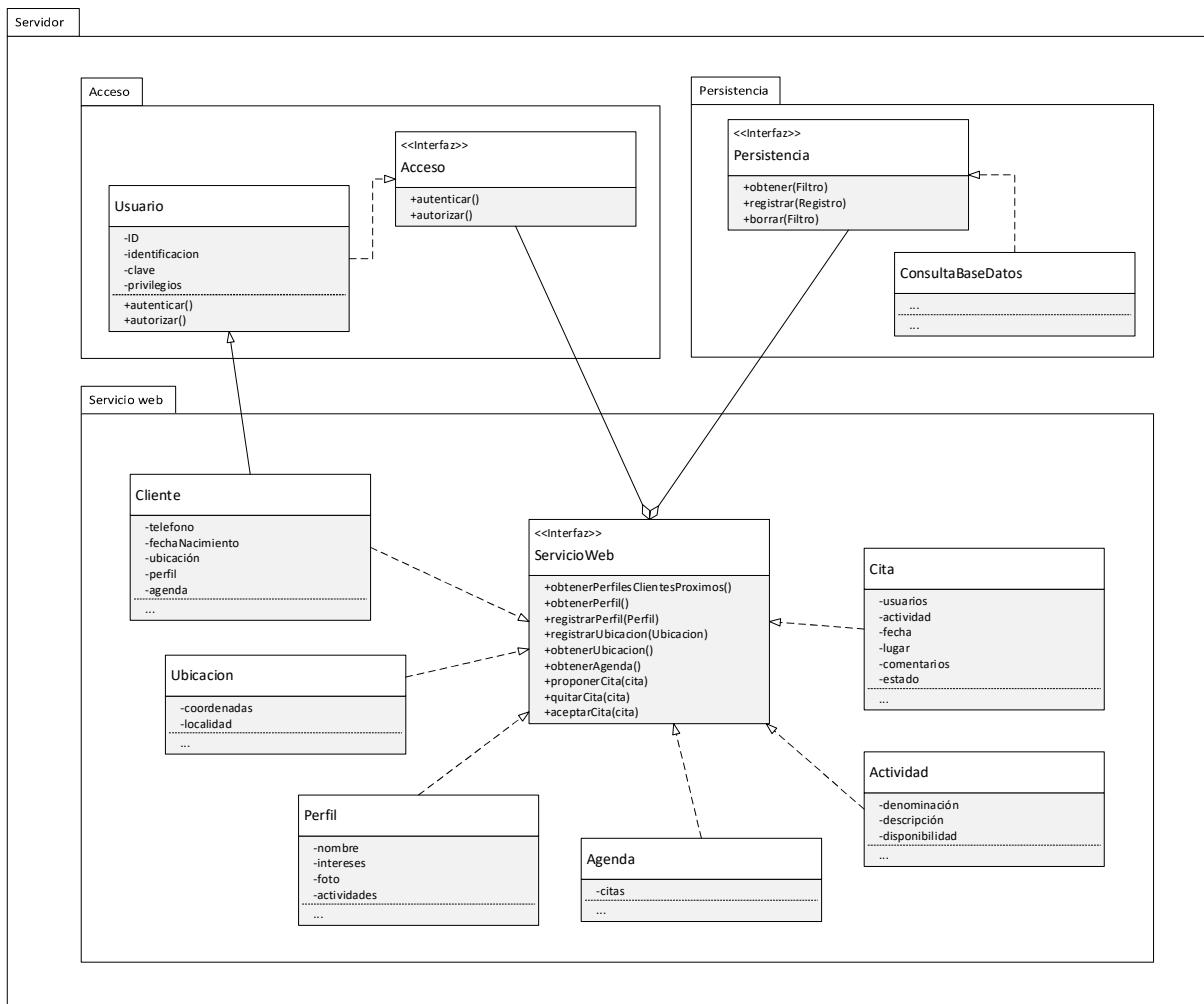


Figura 28: Diagrama de Clases de Diseño – Servidor

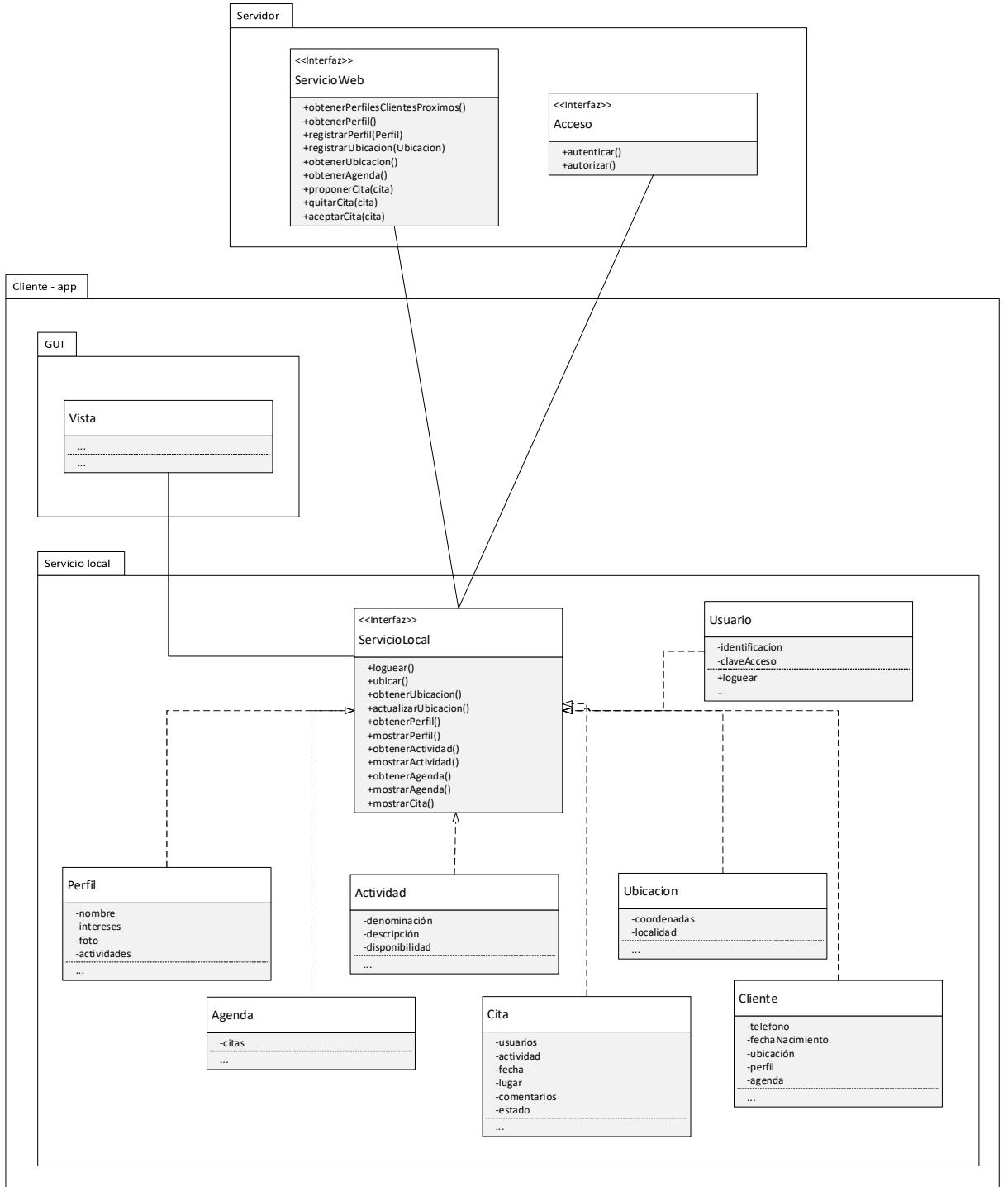


Figura 29: Diagrama de Clases de Diseño – Cliente

Hay que aclarar que aunque se utilicen las mismas denominaciones, y contengan similares atributos, las entidades pertenecientes al subsistema del servidor son diferentes a las del cliente. Las primeras tendrán mayores responsabilidades en cuanto a las operaciones en el servidor, que responde sobre todo a los procesos y cálculos de la “lógica de negocio” (por

ejemplo, determinar los perfiles que se comparten, estructurar el acceso a los datos en base a los modelos y privilegios del usuario, permitir la creación, modificación o eliminación de los registros en la capa de persistencia) mientras que las segundas estarán principalmente dedicadas a la adquisición de la información ya procesada y devuelta desde el servidor, a la captura y envío de datos al servidor y al servicio de la presentación de información estructurada al usuario a través de la GUI.

4.2.2.2. Modelos de interacción

En este apartado extendemos tres casos de uso a través de Diagramas de Secuencias de Sistema (DSS), en particular el CU-01 (Registro de usuario), en el que hemos tratado de perfilar el proceso de activación de la app, el CU-03 (Conocer usuarios próximos) y el CU-06 (Acordar cita).

El primero de ellos, relativo a la activación de la app, puede darse en dos situaciones: es un nuevo usuario no registrado en el sistema, o bien es un usuario ya existente, pero que está instalando la app en un nuevo terminal, o bien reinstalándola. Recordemos que cada usuario será identificado por su número de teléfono móvil y la app ha de estar instalada en un terminal que opere ese número.

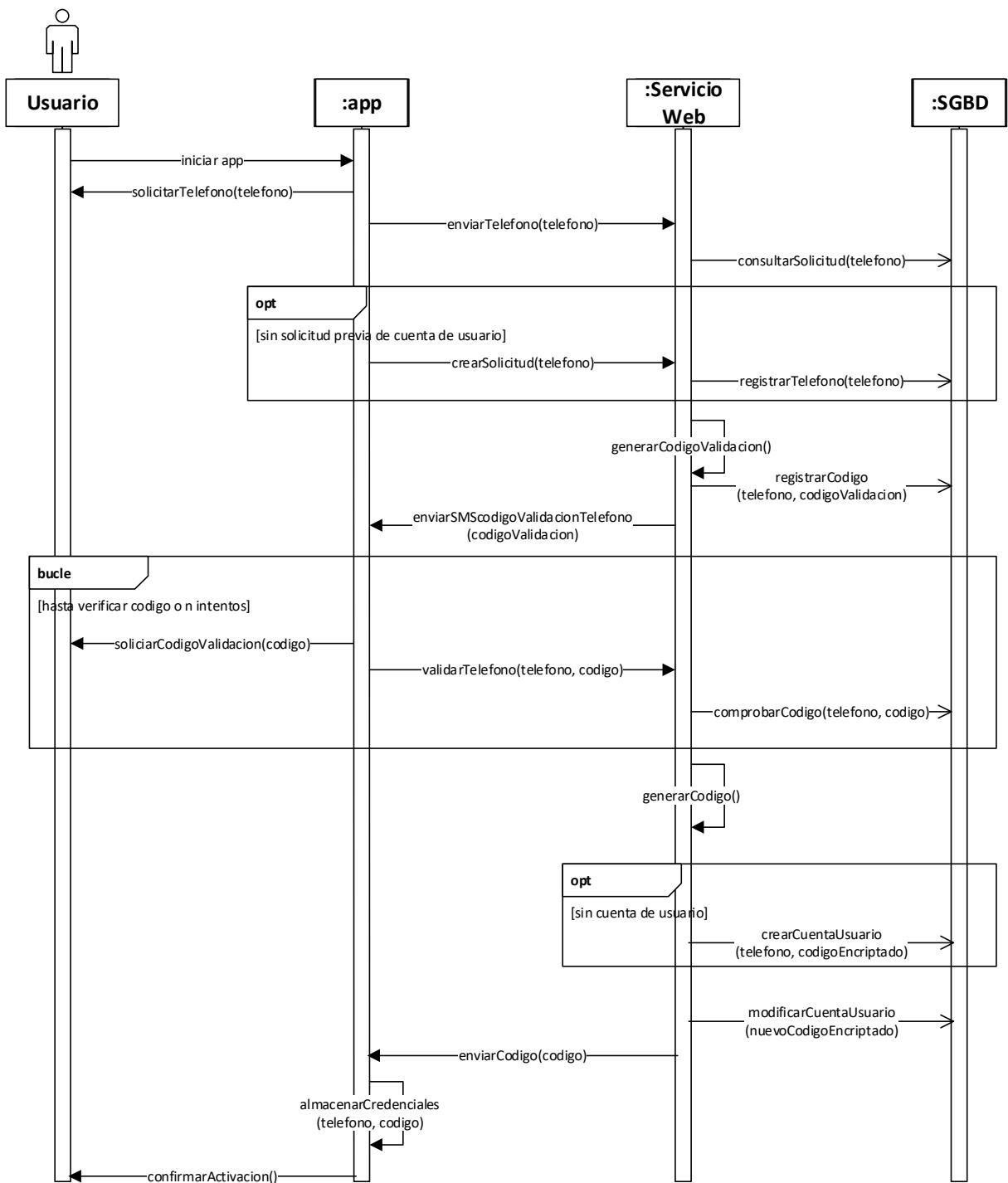


Figura 30: Diagrama de Secuencias de Sistema – Activar la app

Tenemos que aclarar que, como puede observarse en los DSS que presentamos a continuación, debido a las características propias del protocolo de comunicación con el servicio web, sin estado, es necesario proceder a la autenticación y autorización del usuario cuando inicia alguna de las acciones, ya que no se mantiene abierta ninguna sesión en el

servidor. Para ello se utiliza un *código* que se registra en el dispositivo del usuario y que se contrasta con la versión modificada por algún algoritmo de reducción criptográfica que se conserva en la base de datos.

En la figura siguiente procedemos a desplegar el diagrama que corresponde al caso de uso relativo a la obtención de perfiles de usuarios en un determinado margen de distancia, ordenados en virtud de tal proximidad geográfica.

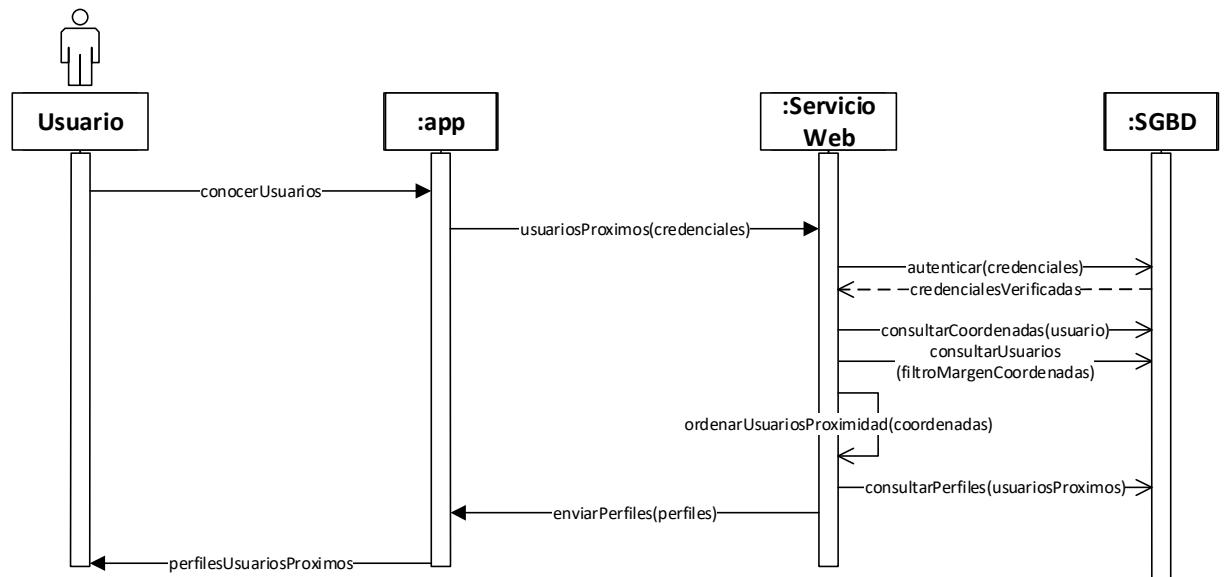


Figura 31: Diagrama de Secuencia de Sistema – Conocer usuarios próximos

En la próxima figura representamos el DSS relativo al CU-06, destinado a procurar la concertación de citas entre usuarios:

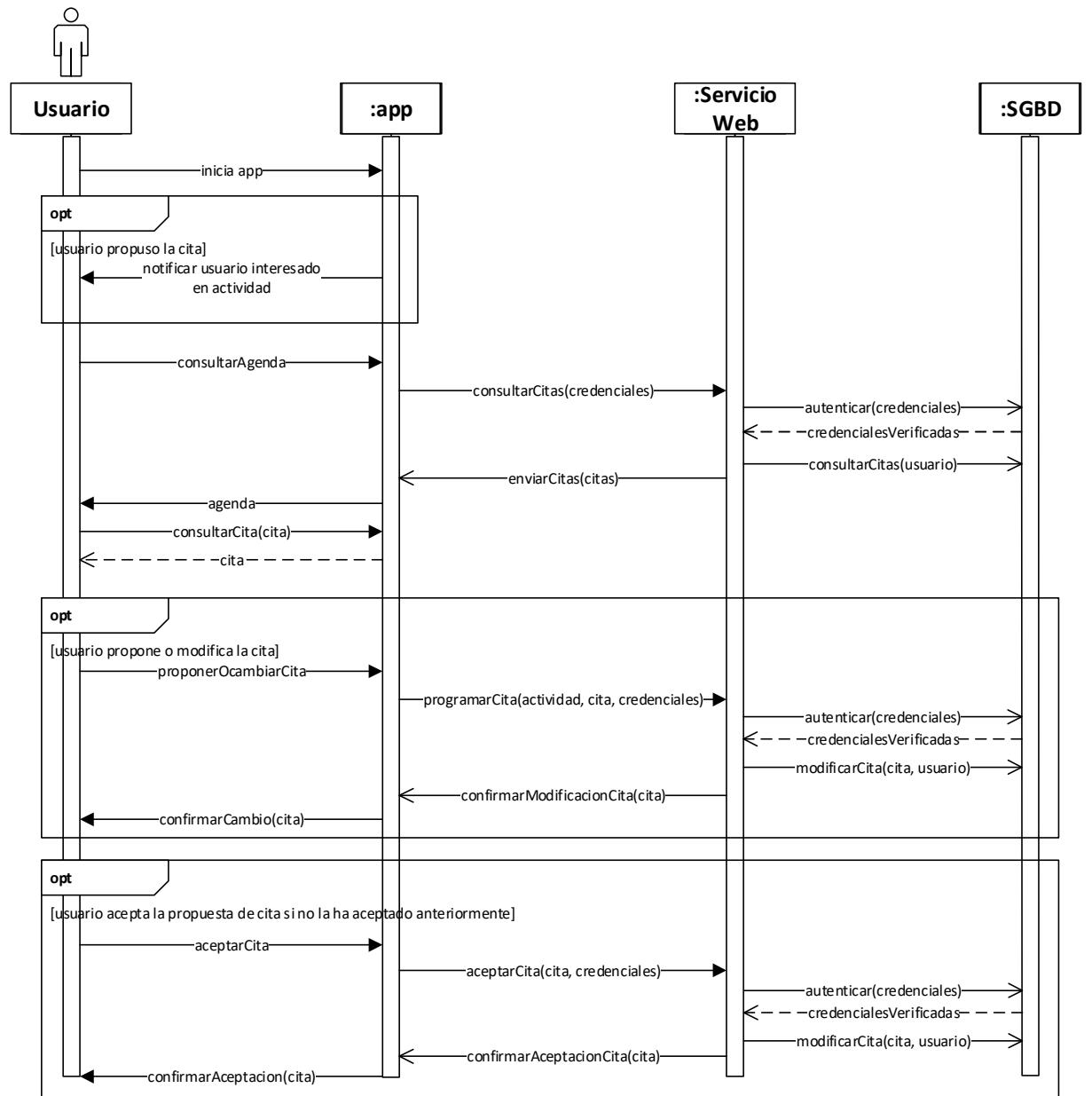


Figura 32: Diagrama de Secuencia de Sistema – Acordar cita

4.2.2.3. Componentes del sistema

La figura siguiente, que recoge un Diagrama de Componentes, resume los diferentes módulos que componen el sistema así como sus dependencias.

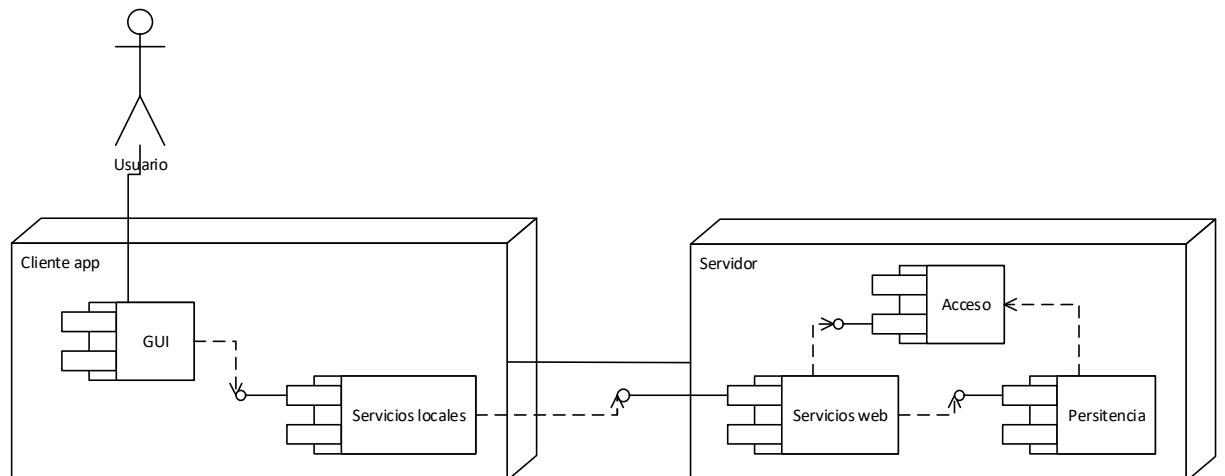


Figura 33: Diagrama de componentes

Los nodos muestran una relación cliente-servidor (la app y el servidor web) cuya comunicación se establece a través de Internet, por medio del protocolo HTTP (*Hypertext Transfer Protocol*).

Como puede observarse, desacoplamos los componentes utilizando para ello las correspondientes interfaces, esto supondría una ventaja en iteraciones posteriores o evoluciones del sistema, ya que permite que los módulos puedan ser modificados o sustituidos por otros que las realicen.

El cliente, la app, es el que mantiene la presentación y la lógica de control de la interacción con el usuario, mientras que la “lógica de negocio” reside en el servidor.

La parte visible y con la que puede relacionarse el usuario es la GUI (siglas en inglés de interfaz gráfica de usuario), lo que ésta muestra depende de la información que le es provista por los servicios locales, éstos además de la gestión de las vistas o *actividades* que mostrarán la GUI, ejecutan otros procesos, tales como obtener la ubicación mediante GPS, envío de credenciales para autenticación y autorización del usuario, filtrar o clasificar datos en alguno de los procesos. Para su realización, la mayor parte de los servicios locales consumen servicios web provistos desde el servidor.

El componente denominado *Acceso*, del que dependen los servicios web, se encarga de autenticar y autorizar al usuario para operar con los servicios provistos, así como para el acceso al módulo de persistencia.

El módulo correspondiente a la persistencia se encarga de la interacción con la base de datos, y establece cómo se crean, muestran o modifican los registros de datos. Los servicios web hacen uso del módulo de persistencia para cumplir con sus procesos.

4.2.2.4. Modelo lógico y físico de datos

En este apartado recogemos el modelado realizado para la base de datos del sistema. Para ello nos apoyamos en el Diagrama Entidad Relación (DER), siguiendo, por un lado, el modelo conceptual propuesto por Peter Chen en 1976 y, por otro lado, la notación UML.

La primera figura recoge un esquema básico concerniente al esquema conceptual de la base de datos.

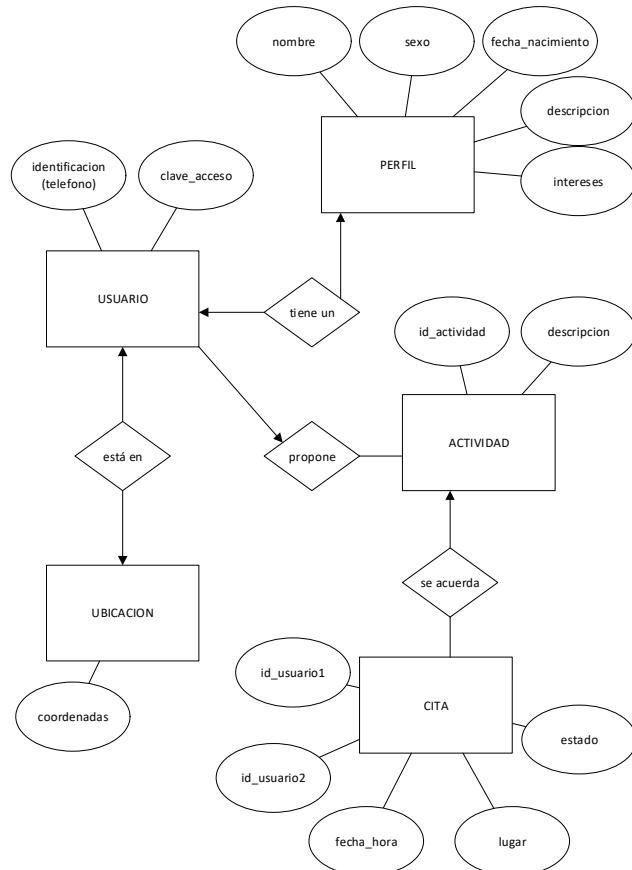


Figura 34: Diagrama Entidad Relación – Vista general

Adelantando el detalle de la estructura de tablas de la base de datos, en la figura siguiente presentamos un Diagrama de Entidad Relación con otro tipo de notación:

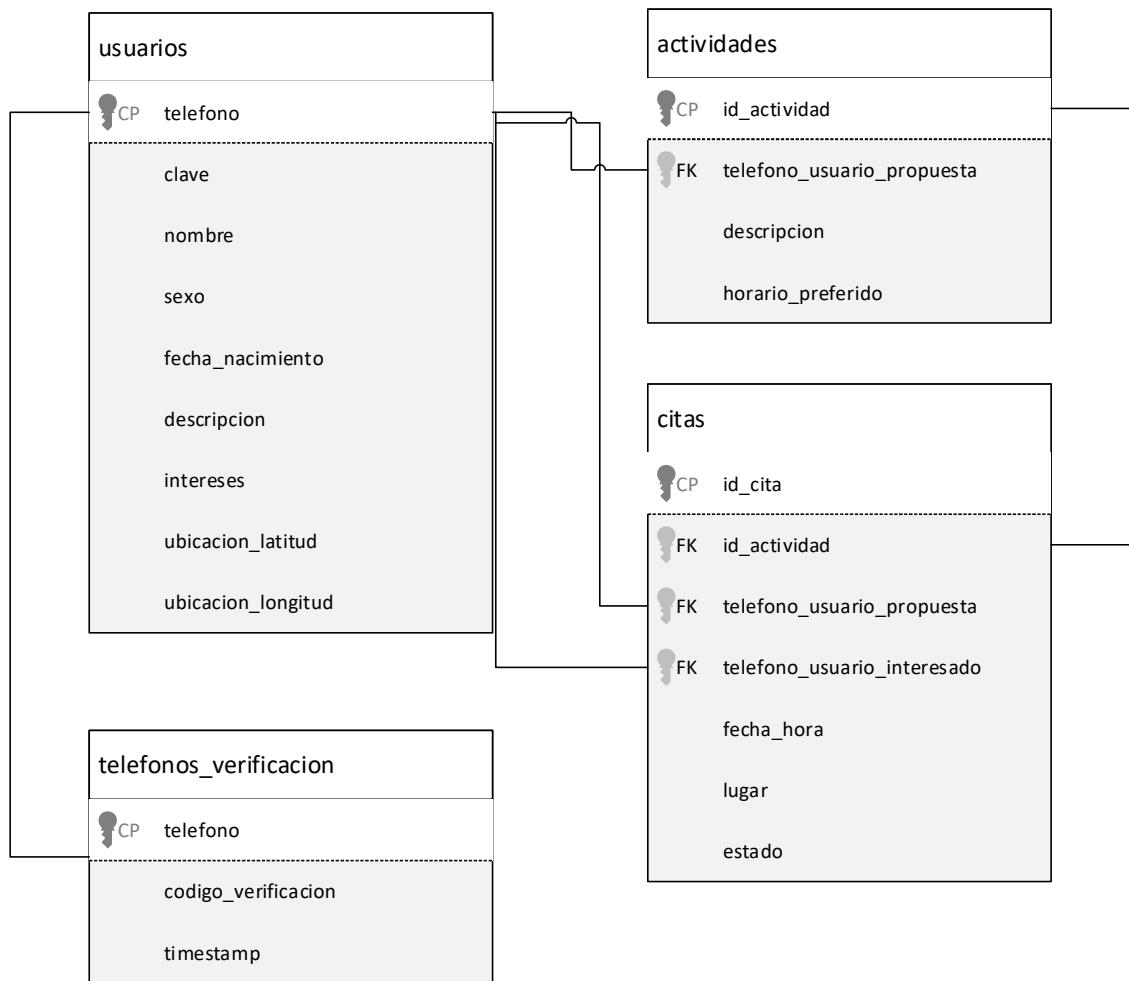


Figura 35: Diagrama Entidad Relación – Base de datos

Siguiendo este diseño, la estructura de la base de datos consta de cuatro tablas (usuarios, actividades, citas y telefonos_verificacion) que paso a detallar a continuación. Primero aclarar que en las tablas las claves principales o primarias se referencian con las siglas PK (*Primary Key*), y las claves externas con FK (*Foreign Key*).

Tabla 10

Tabla “usuarios”

Campo	Tipo	Índice	Relación
Teléfono	Alfanumérico (12 caracteres)	PK	
Clave	Alfanumérico (32 caracteres)		

Nombre	Alfanumérico (32 caracteres)
Sexo	Alfanumérico (6 caracteres)
fecha_nacimiento	Fecha
descripción	Texto (255 caracteres)
Intereses	Texto (500 caracteres)
ubicacion_latitud	Coma flotante doble precisión
ubicacion_longitud	Coma flotante doble precisión
Id	Alfanumérico (32 caracteres)

La tabla “usuarios” es la más importante de la base de datos. Contiene los datos relativos a cada uno de los usuarios registrados en el sistema. Recordemos que cada usuario queda identificado por su número de teléfono, que es la clave primaria de la tabla, contenido en un alfanumérico de 12 caracteres, de tal forma que pueda recoger también prefijos internacionales (los nueve últimos caracteres de la cadena recogen el número de teléfono y los 3 restantes se reservan al prefijo). El campo “clave” almacena la clave de acceso asociada al usuario y que sirve de credencial para ser autenticado en el sistema, para que no esté expuesta se utilizará algún tipo de algoritmo de encriptación (md5 por ejemplo). Esta tabla contiene también los datos del perfil del usuario, que son los que se comparten con otros usuarios, y su ubicación en coordenadas GPS. Por último, añadimos el campo “id” para poder referenciar a un usuario sin tener que hacerlo a través de su número de teléfono, de esta forma cuando desde la aplicación cliente se tuviera que identificar a un usuario evitamos utilizar el número de teléfono, que no ha de ser accesible por ninguna entidad externa a la aplicación del servidor, este campo sirve como parte de las credenciales del usuario para su autenticación.

Tabla 11*Tabla “telefono_verificacion”*

Campo	Tipo	Índice	Relación
Teléfono	Alfanumérico (12 caracteres)	PK	
Código	Alfanumérico (32 caracteres)		
Timestamp	Fecha (datetime)		

La tabla “telefono_verificacion” sirve para almacenar provisionalmente el código de validación que se utiliza para activar la app del usuario, una vez utilizado, o si ha pasado un largo periodo de tiempo sin que se utilice (se toma fecha y hora del registro en “timestamp”),

el registro es eliminado y, en caso de activación, se genera el de usuario en la tabla “usuarios”.

Tabla 12*Tabla “actividades”*

Campo	Tipo	Índice	Relación
id_actividad	Entero (32 dígitos)	PK	
telefono_usuario_propuesta	Alfanumérico (12 caracteres)	FK	telefono (usuarios)
descripcion	Texto (400 caracteres)		
horario_preferido	Intervalo (time)		

La tabla “actividades” recoge las actividades propuestas por cada usuario. Un usuario puede proponer múltiples actividades y, cada una de ellas, queda asociada al usuario que tuvo la iniciativa mediante la clave externa “telefono_usuario_propuesta”. El campo “horario_preferido” registra la franja horaria por la que tenga preferencia el usuario que sugiere la actividad.

Tabla 13*Tabla “citas”*

Campo	Tipo	Índice	Relación
id_cita	Entero (32 dígitos)	PK	
id_actividad	Entero (32 dígitos)	FK	id_actividad (actividades)
telefono_usuario_propuesta	Alfanumérico (12 caracteres)	FK	telefono (usuarios)
telefono_usuario_interesado	Alfanumérico (12 caracteres)	FK	telefono (usuarios)
fecha_hora	Fecha (datetime)		
Lugar	Alfanumérico (255)		
Estado	Entero (1 dígito)		

Finalmente, la tabla “citas” registra las citas asociadas a las diferentes actividades. Cada actividad puede tener múltiples citas, pero cada cita sólo estará asociada a una actividad determinada, de tal forma que la clave externa “id_actividad” sirve para establecer esta relación. Cada cita intenta poner de acuerdo a dos usuarios, por eso en esta tabla se vincula a la identificación del usuario del que partió la propuesta (“telefono_usuario_propuesta”) y del que está interesado en unirse a la actividad (“telefono_usuario_interesado”). El campo “estado” sirve para conocer si ambos usuarios han alcanzado algún acuerdo con respecto a

la cita, podrá tener cuatro valores posibles: ambos están de acuerdo, no se alcanzó acuerdo, propuesta de cita de primer usuario (pendiente aceptación del segundo), propuesta de cita de segundo usuario (pendiente aceptación del primero).

4.2.2.5. Interfaz de usuario

En las siguientes figuras mostramos la GUI (Interfaz Gráfica de Usuario), conforme al flujo de diferentes acciones: activar la app, operar para cumplimentar o modificar el perfil, conocer usuarios próximos, definir actividades, acordar citas.

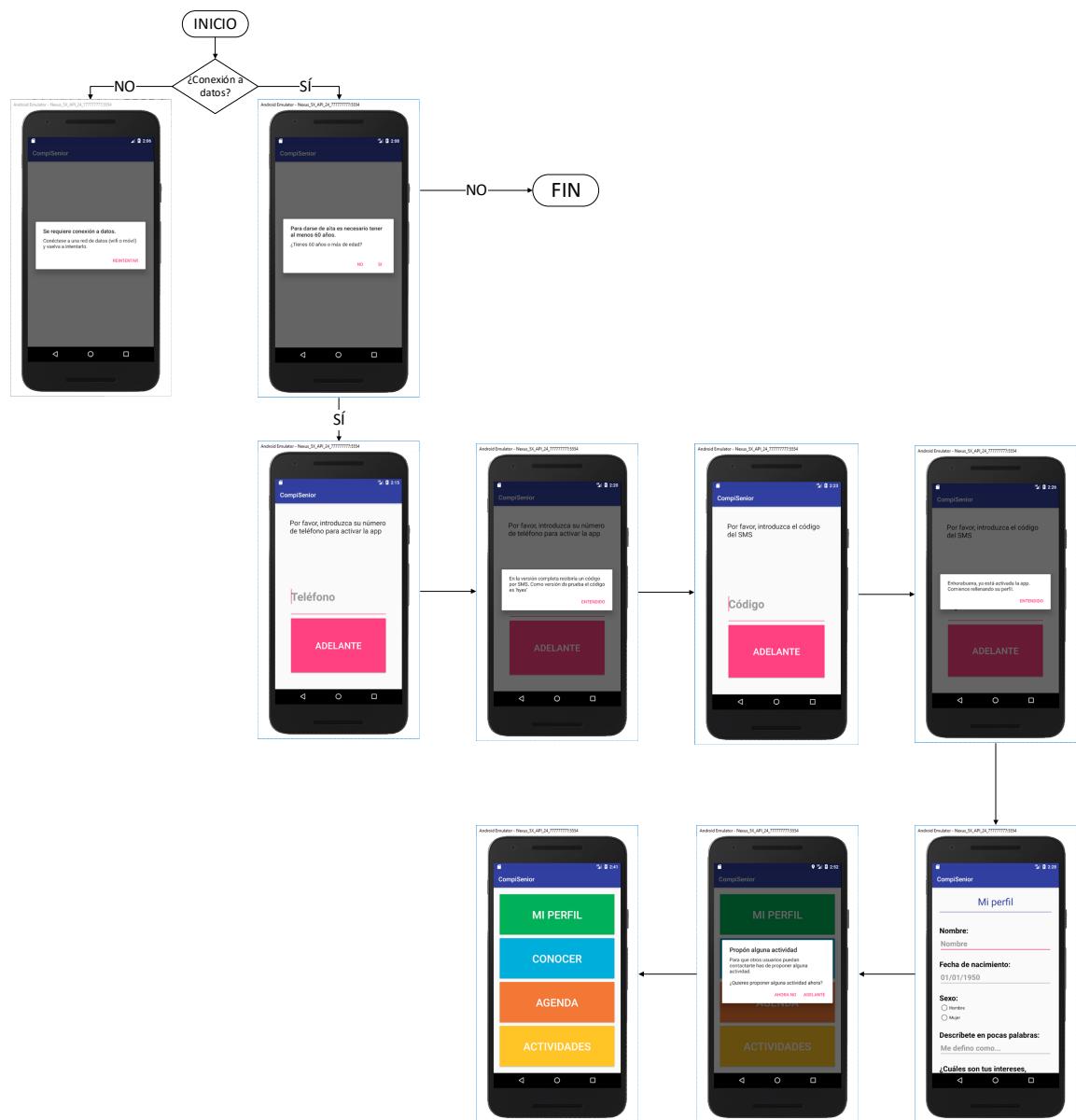


Figura 36: Esquema de la GUI – Activar la app

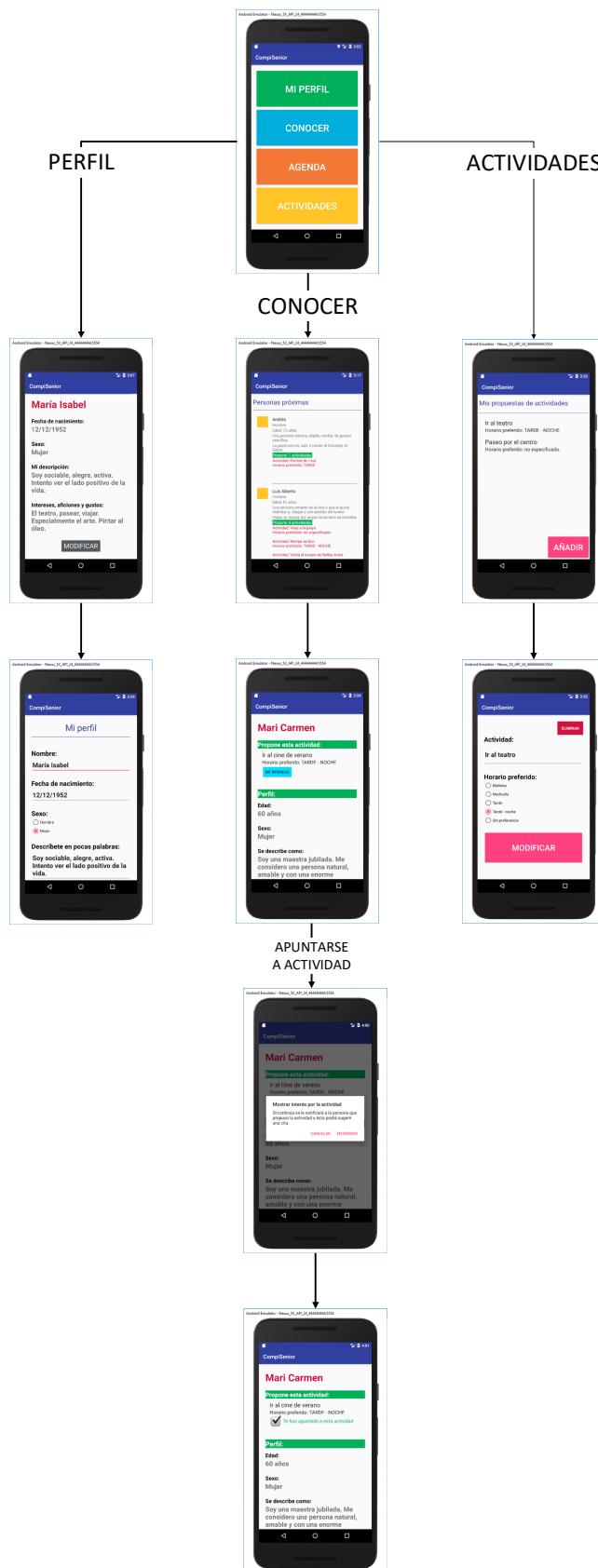


Figura 37: Esquema de la GUI – Perfil, Conocer usuarios próximos y Actividades

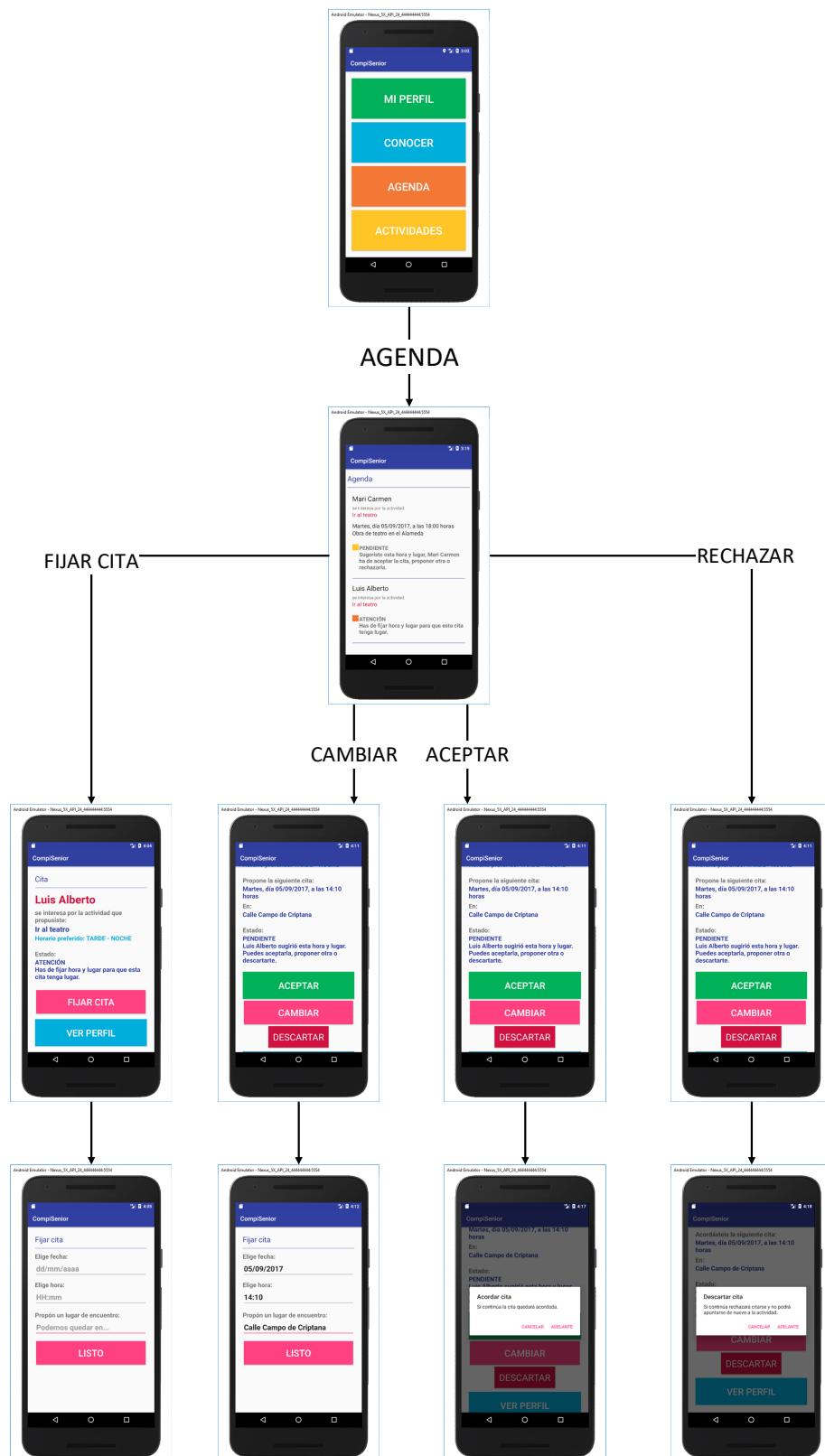


Figura 38: Esquema de la GUI – Acordar citas

4.3. Implantación del sistema

La programación de la aplicación en el lado cliente se emprendió en paralelo con la del servicio web y su conexión con la base de datos. Aunque no era imprescindible hacerlo de esta forma, lo preferimos para ir efectuando pruebas funcionales a lo largo de la codificación y asegurarnos así de que la interacción cliente – servidor funcionaba conforme a lo esperado según avanzábamos. No obstante, en general, por la dependencia que la aplicación cliente tiene de las respuestas del servidor, acometíamos en primer lugar la implementación en el lado servidor y, posteriormente, implementábamos en el lado cliente, realizando las modificaciones oportunas en las diferentes validaciones en ambos lados según resultara necesario. De forma resumida, procedimos de tal forma que (1) primero preparamos el SGBD y generamos la base de datos, (2) a continuación extendimos la API y clases que la realizaban en el lado del servidor y (3) posteriormente en la aplicación cliente implementamos en primer lugar las clases de diseño y seguidamente las operaciones y servicios y la interacción con las actividades y con la interfaz de usuario.

Con respecto a los requisitos funcionales y el cumplimiento de los casos de uso, la aplicación consiguió la práctica completitud, salvo en dos funcionalidades: el envío de SMS desde el sistema y dotar de foto a los perfiles.

En este apartado nos limitados a enunciar los Casos de Uso implementados de forma resumida, ya que una descripción más pormenorizada, que incluye sus escenarios, se encuentra en el apartado correspondiente de Análisis del Sistema dedicado a los mismos. En concreto fueron los siguientes:

- Registro de usuario (CU-01). Que permite al usuario generar y validar una nueva cuenta, así como activar la app para empezar a operar con ella.
- Rellenar el perfil y compartirlo con otros usuarios (CU-02).
- Conocer usuarios próximos (CU-03), tanto su perfil como las actividades propuestas por estos.
- Proponer y compartir actividades (CU-04) y mostrar interés o “apuntarse” a actividades propuestas por otros usuarios, con la finalidad de fijar encuentros para realizarlas (CU-05).
- Facilitar que los usuarios puedan proponer fecha, hora y lugar, para fijar, modificar y acordar citas con otros usuarios para llevar a cabo las actividades propuestas por el usuario o por otros, así como desmarcarse de las mismas si así lo decide alguna de las partes (CU-06).

En cuanto a las funcionalidades no implementadas, la primera de ellas corresponde al caso de uso CU-01, en el que estaba previsto el envío de un SMS con el código de validación destinado a completar el proceso de registro del usuario. Recordemos que la clave primaria para identificar a cada usuario es su número de teléfono, nos parecía interesante este canal de comunicación para asegurarnos de que el proceso de alta del usuario se estaba asociando a un teléfono que estaba en sus manos, por un lado este era un procedimiento que considerábamos más sencillo para el usuario que la recepción de un correo electrónico, que es otro de los canales habitualmente utilizados para la validación de cuentas y, por otro lado, podía servir como medida para paliar el uso de varias cuentas por parte de un mismo usuario, así como de protección ante potenciales fraudes.

La principal razón por la que no se implementó es que dependía de servicios externos ofrecidos por las operadoras de telefonía móvil, es decir, “la implementación de aplicaciones de mensajería móvil es difícil y costosa debido a las barreras propias de la interacción con servicios de telecomunicaciones cerrados” (Yuan, 2012, p.1), en algunos países ciertas compañías de telefonía inalámbrica permiten el envío de SMS sin coste a través de gateways (pasarelas), por ejemplo AT&T, una de las mayores compañías que opera en Estados Unidos, permite enviar el mensaje por correo electrónico a direcciones del tipo numeroTelefono@txt.att.net, lo que abre la posibilidad de implementar el envío de SMS gratuito en el servidor (Yuan, 2012), por ejemplo con la librería PEAR y sus clases para el manejo de mensajería. No es el caso de las compañías que operan en España, al menos de las que hemos podido indagar (Movistar, Vodafone y Orange), si bien alguna sí ofrecía este servicio en el pasado, en la actualidad no está habilitado. Por tanto, para el mercado español, implementar el envío de SMS pasaría por adquirir o contratar un servicio de envío masivo de este tipo de mensajes, cargando con el coste correspondiente.

A efectos prácticos, en el estado actual de desarrollo de la aplicación, la no implantación de este requisito no supone ninguna traba para probar su funcionalidad, ya que la aplicación sólo se ha dirigido en estos momentos a un grupo de probadores (“beta testers”), para los cuales se facilita un código de validación que sustituye al que se haría llegar a través del SMS. No obstante, no hemos eliminado ese paso del escenario del caso de uso porque nos parece interesante incorporarlo en una futura versión del sistema, por las ventajas asociadas a su uso que hemos comentado.

La segunda de las funcionalidades no implementada, pero recogida en los requisitos funcionales, es la de subir una foto al perfil, que se calificó con una prioridad baja y

decidimos posponer a una futura iteración. Pretendemos que la foto del usuario sea tomada desde la cámara del propio móvil, como medida para evitar la falsificación de perfiles.

4.3.1. Cliente – aplicación para Android

La aplicación para el dispositivo móvil suponía una cierta dificultad de partida, ya que era nuestra primera aplicación de cierta complejidad para Android. Partíamos de conocimientos del lenguaje de programación Java, que utilizamos en la implementación, pero tuvimos que ir aprendiendo las singularidades de desarrollar para esta plataforma.

Hasta hace no mucho Eclipse era el entorno de elección a la hora de desarrollar aplicaciones para Android, configurándolo con el SDK (*kit* de desarrollo) de Android. A partir de su lanzamiento el año pasado, el entorno oficial de desarrollo, sobre el que trabajamos, es *Android Studio* (Google, s.f.a). Incluye, entre otras características interesantes para el desarrollador, dos componentes que nos resultaban especialmente convenientes para nuestra implantación: un framework para la realización de pruebas unitarias y la posibilidad de emular un amplio abanico de terminales, configurando las características del terminal virtual sobre el que podemos ejecutar la app, conocido como *Android Virtual Device* (AVD). Este dispositivo emulado o virtual es sobre el que realizamos las pruebas de tipo funcional durante la implementación.

El lenguaje de programación utilizado para codificar la aplicación móvil fue Java, utilizando las librerías adecuadas, así como los correspondientes archivos XML para configurar diversos aspectos, algunos tan relevantes como la apariencia y componentes de la GUI.

El desarrollo se realizó con compatibilidad para la versión de plataforma Android 4.1 (*Jelly Bean*), nivel de API 17. Según las estadísticas proporcionadas a través de Android Studio, alcanzaría al 95,2% de los dispositivos (fecha consulta: 28 de junio de 2017).

En Android las aplicaciones se estructuran en *actividades* (subclases de la clase *Activity*), servicios (clase *Service*), proveedores de contenido y receptores de mensajes (Google, s.f.c). Para nuestros propósitos, los servicios (subclases de *Service*) nos podían haber ayudado a implementar el diseño trazado, en el que queríamos desacoplar todo lo posible la interfaz de usuario (UI) de la interfaz del “servicio local”, la API prevista para realizar las operaciones internas y la interacción con el servidor. En la práctica no conseguimos desacoplar o evitar dependencia al nivel que nos hubiera gustado, según recogimos en el

diseño, entre la UI y la interfaz de servicio local. Por ejemplo, algunas tareas tenían que ser lanzadas considerando un determinado *contexto* de la actividad, o bien tenían que ejecutarse recomendablemente en segundo plano a partir de la actividad, por lo que parcialmente alguno de los métodos de la interfaz del servicio local dependen para cumplir su cometido de una o varias actividades que se ejecutan en la UI. Un ejemplo claro de ello fue el método previsto para establecer la ubicación del dispositivo, que depende en buena medida para su resolución de una determinada actividad (en concreto, utilizamos la “página principal” de la aplicación tras la actividad principal), si bien nos hubiera satisfecho más implementar el diseño previsto, en el que el método contenido en el servicio local conseguía resolver la funcionalidad por completo. En futuras iteraciones podría ser un aspecto mejorable.

Consideramos interesante comentar que empleamos un patrón de diseño *singleton* para procurar un solo punto de acceso global para clases que sólo requerían de una instancia (tales como las instancias de ServicioLocal, Usuario, UsuariosProximos).

Siguiendo una de las más importantes prácticas de seguridad del S-SDLC, efectuamos una revisión del código fuente, o prueba estática, mediante la herramienta HP Fortify SCA (<https://saas.hpe.com/es-es/software/sca>).

4.3.2. Servidor – aplicación en el servidor

En la fase de implementación instalamos y configuramos un servidor de desarrollo con Apache, PHP y MariaDB. En este apartado pasamos a comentar el lenguaje de programación e IDE utilizado para la implementación de la aplicación y de la herramienta de administración de la base de datos.

4.3.2.1. Capa de aplicación

El lenguaje de programación elegido para implementar la aplicación en el lado del servidor fue PHP (acrónimo recursivo de *PHP: Hypertext Preprocessor*). Este es un lenguaje de código abierto cuyo uso está muy extendido, sobre todo por las ventajas que ofrece para el desarrollo de webs, especialmente por el hecho de poder ser utilizado de forma embebida en código HTML, lo que le aporta flexibilidad a la hora de realizar webs con contenido dinámico generado en el servidor.

Las razones para su elección fueron varias. Por un lado, contábamos con experiencia previa con el lenguaje, lo que nos iba a facilitar la labor de codificación. Por otro lado, la integración con el servidor web elegido, Apache 2, ya que PHP puede instalarse como un módulo de éste, incorporando de forma directa el intérprete en su propia estructura, lo que redunda en un mejor desempeño en cuanto a velocidad de ejecución (The PHP Group, s.f.) Además, queríamos aprovechar la formación recibida en el máster acerca de este lenguaje y su puesta en práctica.

Existen multitud de IDEs que facilitan la labor de codificación en PHP, tales como Netbeans, Sublime, PHPStorm, Brackets, Adobe Dreamweaver, etc. En nuestro caso, optamos por Eclipse, ya que estábamos familiarizados en la realización de otros proyectos, y nos aportó todas las herramientas necesarias, incluyendo soporte para el framework para la realización de pruebas unitarias PHPUnit, a través del plugin Makegood.

Para la programación de los diferentes scripts en PHP pusimos en valor los conocimientos adquiridos en el máster. Recordar que el servicio implementado utiliza como protocolo de comunicación HTTP y el formato JSON para el intercambio de los datos entre cliente y servidor. Siguiendo el diseño trazado elaboramos una API, en la que establecimos las diferentes acciones que debían realizarse desde el servidor, requiriendo en cada caso determinados parámetros a través del método POST.

Realizamos una prueba estática de código usando RIPS (<https://sourceforge.net/projects/rips-scanner/files/>), en su versión Open Source (RIPS 0.55), esta es una versión limitada frente a la más potente de tipo comercial.

4.3.2.2. Capa de persistencia

Para la creación y administración de la base de datos MySQL empleamos la herramienta *PhpMyAdmin*, desarrollada en PHP y distribuida bajo licencia GPL, que permite crear y eliminar bases de datos, exportarlas en diferentes formatos y administrar usuarios y privilegios de acceso. Del mismo modo, facilita crear, eliminar y modificar las tablas de la base de datos, borrar, editar y añadir campos, ejecutar sentencias SQL y administrar los índices en campos.

4.3.3. Pruebas unitarias y funcionales

Los test unitarios fueron realizados utilizando dos *frameworks* en correspondencia con los lenguajes de programación utilizados, JUnit 4 (<http://junit.org/junit4/>), que viene integrado en Android Studio, se utilizó para el desarrollo de la app, escrita en Java. Por otro lado, empleamos PHPUnit 3 (<https://phpunit.de/>), incorporado en el servidor de desarrollo (Xampp) y en el IDE de desarrollo empleado (Eclipse Neon).

En cuanto a las pruebas funcionales, si bien no procedimos con un plan estructurado de pruebas funcionales, se tomaron como referencia los requisitos para comprobar el correcto funcionamiento del sistema, así como el correcto funcionamiento de la aplicación conforme se evolucionó en su implementación tanto en el lado cliente como en el servidor.

5. Despliegue y validación

En este apartado nos referimos a las operaciones de despliegue y puesta en funcionamiento de la aplicación en un entorno real de producción, así como las prácticas de seguridad realizadas. La validación se realizó en base a pruebas funcionales, en contraste con los requisitos funcionales y casos de uso, y las pruebas con la participación de usuarios.

5.1. Despliegue en entorno de producción

Si tenemos en cuenta las capas de la arquitectura cliente-servidor que vertebría el sistema y la modalidad en la que la aplicación cliente se distribuiría (las apps suelen lanzarse a través de una tienda de aplicaciones móviles), en la parte del cliente quedaría fuera de nuestro alcance la configuración directa del dispositivo y su mantenimiento, ya que sería el usuario el responsable de la misma. Por tanto, nuestra directa responsabilidad se vuelca en que la aplicación móvil haga un uso adecuado de los permisos de acceso a servicios del dispositivo en el que se ejecute y, sobre todo, en asegurar el funcionamiento fiable, seguro y eficiente en el lado del servidor. Por eso, a continuación nos centramos en la descripción del sistema operativo y hardware del entorno de producción en el lado del servidor, así como las operaciones para el despliegue y puesta a disposición del sistema.

En principio, teníamos que tomar decisiones acerca del equipamiento físico y medio de alojamiento (*hosting*) del servidor web y de la base de datos. Las posibilidades de alojamiento que barajamos fueron o bien en la nube, por ejemplo en una plataforma como Azure, AWS o la Google Cloud Platform, o bien utilizando equipamiento propio, esto es, configurando y bastionando un servidor y dispositivos de comunicaciones en una red propia.

Entre las principales ventajas de optar por la nube están (1) la escalabilidad, ya que permite el crecimiento conforme las necesidades del servicio se amplían, primordialmente por la afluencia o tráfico de usuarios, (2) proporciona mayores garantías de disponibilidad de los servicios, puesto que son compañías de gran tamaño que cuentan en sus CPD (Centros de Proceso de Datos) con abundantes recursos de mantenimiento, equipos y personal y, unida a la anterior ventaja, (3) mayor protección frente a amenazas y ciberataques.

Por otra parte, disponíamos de un ordenador de tipo servidor, de la línea de servidores Proliant del fabricante Hewlett Packard Enterprise, con tarjeta de red de alta velocidad, memoria RAM ECC (*Error Correcting Code*) y cuatro bahías para discos duros. La opción de

configurarlo para disponer del alojamiento para el servidor web tenía una serie de desventajas frente a las plataformas en la nube, principalmente la menor capacidad y escalabilidad, mayor exposición a un ataque de denegación de servicio, así como a amenazas y a daños, por lo que las garantías de un funcionamiento continuado son mucho menores. A pesar de todas estas desventajas, teníamos unos alicientes que eran importantes en el desarrollo de un trabajo académico como este, entre los que podemos citar un menor coste inicial que la solución en la nube, puesto que disponíamos del equipamiento (en otro caso habría que valorar el coste de adquisición y amortización) y de conexión de banda ancha a través de fibra óptica, mayor control sobre el entorno de producción, suficiente para ejecutar el sistema como prototipo y, por encima de todo, la oportunidad de experimentar con los conocimientos adquiridos en el máster sobre administración de servidores y, en particular, acerca de su bastionado y puesta en marcha.

Tomada la decisión de que la instalación del servidor web sería en equipos propios, el siguiente paso fue decidir el sistema operativo del servidor. El servidor web Apache es multiplataforma por lo que el abanico de opciones estaba abierto, pero, como explicamos anteriormente, nos resultó más oportuna una configuración de tipo “LAMP” (Linux, Apache, MySQL y PHP). De las múltiples distribuciones GNU/Linux disponibles optamos por una basada en Debian, Ubuntu Server, que instalamos sin GUI para aligerar de este componente al servidor y aprovechar así algo más su capacidad de procesamiento. Configuramos la comunicación SSH y FTP para administrar el servidor desde otro terminal de la red local, ya que el ordenador destinado a servidor no iba a contar con pantalla ni dispositivos de entrada.

Siguiendo recomendaciones de seguridad en operaciones, así como específicas para la configuración segura de Apache (Mifsuf, 2012; Gómez, Candela y Sepúlveda, 2013), integramos el servidor en la red local aislándolo del resto de equipos (salvo puertos destinados a su administración a través de SSH y FTP), abrimos puertos para permitir la conexión de clientes externos, activamos y configuramos firewalls, en el enrutador y en el propio servidor, instalamos y activamos los módulos estrictamente necesarios para limitar la superficie expuesta y configuramos adecuadamente el servidor web, impidiendo mostrar información sensible o sobre el sistema y el acceso a directorios. Por último, configuramos un dominio para dirigir las peticiones de la aplicación cliente a nuestra IP pública, así como el correspondiente *virtual host* en el servidor web para dar acceso a la API del servicio web. Del mismo modo, seguimos recomendaciones para la instalación, configuración, gestión de cuentas, optimización y monitorización del Sistema Gestor de la Base de Datos,

en este caso MySQL, en especial del manual de Hueso (2014). Para realizar copias de seguridad y recuperación de la base de datos, en el futuro se puede implantar alguna configuración tipo RAID, por el momento optamos por automatizar un backup en un segundo disco duro mediante un script lanzado por el *daemon* del sistema operativo.

5.2. Test de penetración

Como paso previo a la puesta en producción, realizamos una prueba de caja negra semiautomática, un *test de penetración* mediante el uso de la herramienta de OWASP ZAP Scan, un proxy interceptador que permite la manipulación de las peticiones y respuestas del servidor. Aunque en la práctica fue más de “caja gris”, ya que aprovechamos el conocimiento que teníamos del sistema en la auditoría técnica para ajustar los *adds-on* necesarios y el *crawling* (recopilación de información sobre la aplicación web), ya que, como es obvio, conocíamos la estructura de la aplicación, el tipo de servidor, lenguaje y demás, así como los directorios y archivos de los scripts y configuración y podíamos utilizar las credenciales de algún usuario para autenticarnos en el sistema.

De esta manera cumplíamos con otras de las buenas prácticas del modelo de S-SDLC, con la finalidad de comprobar el comportamiento del sistema frente a diversos ataques (inyección de código, manipulación de variables de sesión) y si las medidas de seguridad implementadas eran suficientes, así como detectar errores de configuración que podían suponer algún tipo de vulnerabilidad.

5.3. Pruebas de usuario

Para someter a evaluación la GUI (Interfaz de Usuario) de la aplicación y su funcionamiento realizamos, por un lado, un análisis heurístico, sin usuarios, siguiendo los principios de usabilidad propuestos por Mordecki (2012, pp. 100 - 107) y, por otro lado, contamos con la colaboración de cinco usuarios a los que observamos mientras utilizaban la aplicación. Se les pidió que interactuaran con ella desde su inicio y que realizaran las siguientes acciones: (1) que procedieran a su activación, (2) cumplimentación del perfil, (3) contacto con un determinado usuario próximo y (4) gestionar una cita; después podían utilizarla de manera libre. Posteriormente, nos aportaron sus impresiones.

En general, concluyeron que la interfaz era clara, intuitiva y fácil de usar. Se echó a faltar la funcionalidad de recibir notificaciones sin que se tuviera que acceder a la sección de agenda para saber de forma inmediata si se había producido algún contacto con otro usuario, de tal forma que no fuera necesario estar pendiente o realizar ninguna acción periódica por parte del usuario para saber si ha contactado o tiene que fijar una cita con otro. Algunos usuarios comentaron que se les había hecho algo tediosa la cumplimentación del perfil, en los apartados de descripción de uno mismo e intereses, aunque opinaron que veían necesario incluir dicha información en los perfiles. También comentaron que no resultaba intuitivo que se tuviera que registrar alguna actividad para que otros usuarios contactaran, aunque entendían que era una forma más cercana de tomar contacto, ya que se establecía de antemano la actividad común de interés.

6. Conclusiones

A lo largo del trabajo realizado hemos conseguido los objetivos que nos marcamos inicialmente, así como el propósito esencial de poner en práctica los conocimientos adquiridos a lo largo del máster. Exploramos y tomamos decisiones en cuanto a la arquitectura del sistema y las tecnologías que utilizaríamos, de forma análoga a cómo lo haríamos en un entorno de producción de software real.

El marco general de nuestro trabajo fue un proceso de software iterativo e incremental. Logramos realizar el análisis y diseño del sistema, con una orientación a objetos, empleando métodos y técnicas de modelado para describirlo. Implementamos y desplegamos un prototipo o aplicación operativa conforme a los requerimientos funcionales y no funcionales que se establecieron, comprobando su funcionamiento en un entorno real de producción. Además, seguimos buenas prácticas de desarrollo seguro (S-SDLC), teniendo en cuenta las amenazas específicas de aplicaciones on-line.

En resumen, se consiguieron los propósitos de los apartados técnico y académico. El tercer aspecto que consideramos relevante analizar es el de la utilidad práctica, que fue la idea esencial de la que partió el proyecto, es decir, la capacidad de la aplicación para disminuir las situaciones de soledad no deseada o fomentar las relaciones sociales en los mayores. En este punto, al tratarse de una aplicación basada en la creación de una red social o de contactos, para que resulte realmente útil y consiga su propósito ha de alcanzar una determinada masa crítica de usuarios, al menos en un alcance limitado a determinadas ciudades, podemos especular que la aplicación serviría a este propósito si lo consiguiera. En la misma línea, consideramos que la aplicación contiene las funciones esenciales, pero sería interesante dotarla de unas funcionalidades añadidas que le proporcionen más atractivo a los ojos de los usuarios y mayor seguridad en cuanto a evitar un uso fraudulento (tal y como ya tratamos en el *caso de abuso* correspondiente). De tal forma que a continuación planteamos mejoras futuras o nuevas líneas de trabajo:

- Posibilidad de subir foto para incluirla en los perfiles públicos de los usuarios.
- Permitir filtros en la búsqueda de usuarios (por ejemplo edad, actividades, intereses).
- Dotar de análisis semántico de la información introducida en el campo “intereses”, de tal forma que se puedan sugerir usuarios afines en los gustos o aficiones.
- Facilitar que los usuarios valoren los encuentros, lo que puede favorecer a los usuarios más activos y, al mismo tiempo, evitar determinados abusos, plantones, etc.

- Dotar a la aplicación de un *chat* para que los usuarios que establezcan contacto puedan comunicarse de forma más libre antes de citarse.
- Posibilitar que los usuarios pongan fechas límite a las actividades
- Permitir organizar grupos para actividades.

Mejoras expresamente dirigidas a evitar uso fraudulento o inadecuado de la aplicación:

- Tomar la foto del perfil del usuario directamente con la cámara del dispositivo.
- Envío del código de activación por SMS, para intentar comprobar que la instalación se realiza en un dispositivo con ese número de teléfono.
- Proporcionar un “botón denunciar” para que los usuarios puedan informar de malas prácticas o usos abusivos que detecten.
- Realizar seguimiento y análisis de las operaciones y transacciones de los usuarios en el sistema (*accountability*), por ejemplo para detectar tendencias en accesos no autorizados.

Por último, habría que dotar a la aplicación de un entorno *backend* o de administración del sistema, que podría ser soportada en una aplicación web. En el estado actual esta labor hay que realizarla operando directamente sobre la base de datos.

7. Referencias bibliográficas

- Barceló, J. M., Íñigo, J. y Llorente, S. (2008) *Protocolos y aplicaciones Internet*. Barcelona: UOC.
- Boyd, D. M. y Ellison, N. B. (2008) Social Network Sites: Definition, History, and Scholarship. *Journal of Computer-Mediated Communication*, 13, 210–230.
- Brownworth (2011) *How Location Services Work on Mobile Devices*. Recuperado el 22 de julio de 2017 de <https://anders.com/cms/389/iPad/GPS/Location.Based.Services>
- Cinco Días (2016) *Android alcanza su mayor cuota de mercado en España*. Recuperado el 5 de Junio de 2017 de https://cincodias.elpais.com/cincodias/2016/06/15/tecnologia/1465987235_750846.html
- ECMA International (2013) *Standard ECMA-404. The JSON Data Interchange Format*. Recuperado de <https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>
- Fielding, R. T. (2000) *Architectural Styles and the Design of Network-based Software Architectures* (Disertación doctoral no publicada). University of California Irvine.
- Fundación Telefónica (2016) *La Sociedad de la Información en España 2015*. Barcelona: Ariel.
- Fundación Vodafone España (2012) TIC y mayores, conectados al futuro. Recuperado de http://www.fundacionvodafone.es/sites/default/files/tic_y_mayores_conectados_al_futuro.pdf
- Gómez, C. E., Candela, C. A. y Sepúlveda L. E. (2013) Seguridad en la configuración del Servidor Web Apache. *Revista INGE CUC*, 9, 31-38.
- Google (s.f.a) *Conoce Android Studio*. Recuperado el 26 de junio de 2017 de <https://developer.android.com/studio/intro/index.html?hl=es-419>
- Google (s.f.b) *Making Your App Location-Aware*. Recuperado el 23 de julio de 2017 de <https://developer.android.com/training/location/index.html>
- Google (s.f.c) *Aspectos fundamentales de la aplicación*. Recuperado el 04 de agosto de 2017 de <https://developer.android.com/guide/components/fundamentals.html?hl=es-419>

Hazlewood, L. (2016) *The Fundamentals of REST API design*. Recuperado el 20 de julio de 2017 de <https://stormpath.com/blog/fundamentals-rest-api-design>

Hueso, L. (2014) *Administración de sistemas gestores de bases de datos*. Madrid: RA-MA Editorial.

Instituto Nacional de Estadística / INE (2016) Encuesta sobre equipamiento y uso de tecnologías de información y comunicación en los hogares. Recuperado de http://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736176741&menu=resultados&secc=1254736194579&idp=1254735976608

Kalache, A. y Keller, I. (1999) The WHO perspective on active ageing. *Promotion & Education*, 6(4), 20-23.

Kendall, K. y Kendall, J. E. (2011) *Análisis y diseño de sistemas*. México: Pearson Educación.

Larman, C. (2004) *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Segunda edición. Madrid: Pearson Educación.

Lee, J. (2013) *Oracle vs. MySQL vs. MS SQL Server: A comparison of popular RDBMS*. Recuperado el 22 de julio de 2017 de <https://blog.udemy.com/oracle-vs-mysql-vs-sql-server/>

Mifsuf, E. (2012). *Apache*. Ministerio de Educación de España.

Mordecki, D. (2012). *Miro y entiendo*. Recuperado de <http://www.mordecki.com/html/descargamye.php>

Netcraft (2017) *February 2017 Web Server Survey*. Recuperado el 17 de junio de 2017 de <https://news.netcraft.com/archives/2017/02/27/february-2017-web-server-survey.html>

Norris, P. (2001). *Digital divide: Civic engagement, information poverty, and the Internet worldwide*. Cambridge University Press.

Organización Mundial de la Salud (OMS); Edwards, P. (2001) Salud y envejecimiento. Un documento para el debate. IMSERSO *Perfiles y tendencias*, 4-5. Recuperado de <http://envejecimiento.csic.es/documentos/documentos/boletinsobreenvjec04y05.pdf>

OWASP (2013) *OWASP Top 10 – 2013. Los diez riesgos más críticos en Aplicaciones Web*. Recuperado de https://www.owasp.org/images/5/5f/OWASP_Top_10_-2013_Final_-Espa%C3%B1ol.pdf

OWASP (2017a) *OWASP Top 10 – 2017 rc1. The Ten Most Critical Web Application Security Risks.* Recuperado de <https://github.com/OWASP/Top10/raw/master/2017/OWASP%20Top%2010%20-%202017%20RC1-English.pdf>

OWASP (2017b) *Estándar de Verificación de Seguridad en Aplicaciones 3.0.1.* Recuperado de https://www.owasp.org/images/a/aa/Est%C3%A1ndar_de_Verificaci%C3%B3n_de_Seguridad_en_Aplicaciones_3.0.1.pdf

Pino, M. R., Soto, J. G. y Rodríguez, B. (2015) Las personas mayores y las TIC. Un compromiso para reducir la brecha digital. *Pedagogía Social, Revista Interuniversitaria*, 26, 337-359.

Pressman, R. S. (2010) *Ingeniería del software. Un enfoque práctico.* México: McGraw-Hill.

Rodríguez, M. (2009) La soledad en el anciano. *Gerokomos*, 20(4), 159-166.

Sarabia, C. M. (2009) Envejecimiento exitoso y calidad de vida. Su papel en las teorías del envejecimiento. *Gerokomos*, 20(4), 172-174.

Sommerville, I. (2005) *Ingeniería del software.* Madrid: Pearson Educación.

The PHP Group (s.f.) *PHP: Apache – Manual.* Recuperado el 20 de julio de 2017 de <http://php.net/manual/es/book.apache.php>.

Universidad Internacional de la Rioja / UNIR (2016a) *Seguridad en el Software. Tema 2: Principios de diseño seguridad del software.* Material no publicado.

Universidad Internacional de la Rioja / UNIR (2016b) *Metodologías, Desarrollo y Calidad en la Ingeniería de Software. Tema 3: Desarrollo de software orientado a objetos.* Material no publicado.

Universidad Internacional de la Rioja / UNIR (2016c) *Computación en el Servidor Web. Tema 6: Introducción a los servicios web.* Material no publicado.

Universidad Internacional de la Rioja / UNIR (2016d) *Administración de servidores web. Tema 1: Introducción a servidores web.* Material no publicado.

Xataka (2016) *Estadísticas de distribución de Android en Febrero 2016*. Recuperado el 5 de Junio de 2017 de <https://www.xataka.com.mx/celulares-y-smartphones/estadisticas-de-distribucion-de-android-en-febrero-2016-marshmallow-apenas-en-el-1-2>

XDA (2012) *Understanding Android GPS Architecture*. Recuperado el 22 de julio de 2017 de <https://forum.xda-developers.com/showthread.php?t=2063295>

Yuan, M. J. (2012) Construya mensajería de texto móvil en sus aplicaciones web. IBM DeveloperWorks. Recuperado el 15 de agosto de 2017 de <https://www.ibm.com/developerworks/ssa/library/mo-sms-web/mo-sms-web-pdf.pdf>

8. Anexos

8.1. Artículo

8.1.1. Introducción

Las personas mayores de 60 años hacen un uso cada vez mayor y más amplio de las funciones de los móviles de nueva generación o *smartphones*, lo que abre la posibilidad de cubrir necesidades específicas de esta población a través de aplicaciones para dispositivos móviles y, en concreto, fomentar el envejecimiento activo y la prevención del aislamiento social. Con esa finalidad desarrollamos una aplicación distribuida que permite poner en contacto a personas de esa franja de edad entre sí con el objetivo de que encuentren compañía y relaciones amistosas. En su desarrollo seguimos un proceso de software iterativo e incremental, con un enfoque de Análisis y Diseño Orientado a Objetos y la incorporación de buenas prácticas de seguridad en el ciclo de vida de desarrollo de software (S-SDLC).

Para justificar el enfoque de la aplicación profundizamos en tres aspectos: (1) la vulnerabilidad a la soledad en personas de más de 60 años y sus consecuencias, (2) el concepto de *envejecimiento activo* y (3) la utilización de TIC (Tecnologías de la Información y Comunicación) entre personas mayores y especialmente de aplicaciones para dispositivos móviles.

A partir de los 60 años somos más vulnerables a la soledad y a padecer aislamiento social, lo que puede mermar notablemente el bienestar e influir de manera negativa en el estado de salud psicológica y física (Edwards, 2001), pudiendo conllevar dependencias de índole social, funcional, cognitivo y problemas de salud que alteran el nivel de autonomía y el desenvolvimiento de la vida cotidiana de la persona (Rodríguez, 2009).

El “envejecimiento activo” se contrapone a la vinculación de la vejez con una etapa vital de irremediable deterioro. Edwards (2001) lo define como “el proceso por el cual se optimizan las oportunidades de bienestar físico, social y mental durante toda la vida con el objetivo de ampliar la esperanza de vida saludable, la productividad y la calidad de vida en la vejez” (p.15). De hecho, cada vez más personas avanzan en edad manteniendo un nivel de autonomía, capacidades y satisfacciones vitales, tal y como se refleja en diversos estudios longitudinales (Sarabia, 2009)

Por otra parte, los mayores desean aprovechar los beneficios que se derivan de las TIC y pueden vencer las posibles dificultades para adoptar estos avances (Pino, Soto y Rodríguez, 2015). Los datos evidencian que la adquisición por parte de las personas mayores de estas nuevas tecnologías para su uso cotidiano, incluyendo la utilización de diferentes aplicaciones y servicios, se está igualando con respecto a grupos de personas de otras generaciones. En lo que respecta a España la diferencia en el uso de las TIC de las personas mayores con respecto a las pertenecientes a otros grupos de edad se reduce de manera significativa año tras año y previsiblemente desaparecerá de forma completa en los próximos tiempos. Así lo hemos podido constatar en los resultados de los estudios realizados por Fundación Vodafone España (2012), Fundación Telefónica, (2016) y la encuesta sobre equipamiento y uso de tecnologías de información y comunicación en los hogares del INE (2016). Dentro de esta tendencia, la proliferación de la aceptación y utilización de móviles con características avanzadas (*smartphones*) entre las personas de mayor edad también muestra su avance.

El enfoque de la aplicación desarrollada se centró en reforzar los recursos sociales y las interacciones amistosas, facilitando que los usuarios puedan establecer nuevas relaciones sociales con personas de su grupo de edad próximas geográficamente. Entroncando con el concepto de envejecimiento activo, se trataría de propiciar el encuentro para la realización de actividades y compartir aficiones o el tiempo libre.

8.1.2. Contexto tecnológico

La evolución de las redes sociales ha ido allanando el camino a la utilización de la Red para establecer contactos de tipo amistoso o sentimental, pasando de estar socialmente estigmatizado a estar mejor visto y experimentar un gran crecimiento en los últimos años, proliferando las aplicaciones de citas o contactos (“*online dating*”). Las redes sociales permiten articular y hacer visible la red social del usuario, pero el objetivo de la mayor parte de los usuarios no es necesariamente establecer contacto o conocer a otros que les resultan desconocidos, sino que se comunican principalmente con personas que ya forman parte de su red social extensa (Boyd y Ellison, 2008). En contraste, los sitios destinados a establecer contactos tienen como objetivo primordial que sus usuarios puedan conocer y tratar relación con personas desconocidas con las que compartir encuentros románticos, sexuales, compañía, aficiones o actividades de ocio.

Según nuestras indagaciones, en el panorama actual las aplicaciones “*online dating*” o “*dating sites*” son muy numerosas. En un mercado con tal grado de madurez, algunas buscan llegar a determinados nichos de mercado, ya sea estando destinadas sólo a personas con una orientación sexual concreta o con determinadas preferencias sexuales, o predilecciones acerca de características personales, como la edad, raza, nacionalidad o incluso religión de la posible pareja, o bien diferenciándose por la naturaleza de la relación que se busca, ya sea de tipo amistoso, romántico, sexual, o para otros usos específicos, como compartir tiempo libre o actividades.

La práctica totalidad de los *dating sites* más conocidos tienen su correlato en aplicaciones para móviles, o bien de forma directa han sido aplicaciones desarrolladas para este dispositivo. La modalidad de la aplicación móvil abre además nuevas posibilidades en la utilización y en la forma de concebir el servicio.

Si bien muchas de las aplicaciones pueden ser utilizadas por personas mayores, la mayor parte de los servicios existentes están copados por un público más joven. No obstante, el mercado objetivo de las personas por encima de los 50 años que pueden demandar este tipo de servicios es extenso, sobre todo en ciertos países, por lo que varias compañías intentan hacerse un hueco o ampliar su cuota siguiendo una estrategia basada en la segmentación o bien en la diferenciación, ofreciendo sitios específicos para mayores. Las aplicaciones que conseguimos encontrar están destinadas al mercado internacional y, en concreto, a Estados Unidos, Canadá y Australia.

Conocer específicamente qué tecnologías utilizan las aplicaciones que existen en el mercado no es sencillo, ya que no hemos encontrado ninguna que lo haga público, si bien pudimos observar que son aplicaciones de tipo distribuido, con alguna arquitectura que sigue el modelo cliente-servidor, en el que el usuario accede a través de su navegador o aplicación móvil (cliente) y se conecta con un servidor web o de aplicaciones que centraliza la gestión de los usuarios y el acceso a la base de datos. Esta fue, a su vez, la base tecnológica que utilizamos para la aplicación desarrollada.

Elegimos Apache porque previmos que íbamos a configurar y bastionar el servidor para hospedar el servicio en un plazo menor al estar más familiarizado con este servidor web. Por otra parte, el lenguaje de programación para codificar los scripts del servicio web fue PHP, por lo que la opción de Apache parecía la más conveniente, ya que puede extenderse con un módulo que le dota de un intérprete y su desempeño es más eficiente con este lenguaje.

El servicio web implementado tiene una arquitectura REST (*REpresentational State Transfer*) y es invocado a través de una API para las diferentes acciones o peticiones solicitadas a la aplicación web. REST hace un uso eficiente del ancho de banda, ya que es mucho menos verboso que SOAP y las peticiones pueden almacenarse en caché para un mejor rendimiento y escalabilidad (Fielding, 2000), en contraposición, al no tener estado, el manejo de sesiones es más difícil.

Optamos por JSON (*JavaScript Object Notation*) para portar datos entre la aplicación cliente y la aplicación en el servidor web por su simplicidad y flexibilidad, aparte de ser un formato ampliamente utilizado y recomendado en servicios web basados en REST.

En cuanto al Sistema Gestor de Base de Datos (SGBD), que da soporte a la capa de permanencia en el servidor, nos decantamos por MySQL / MariaDB, ya que es un motor de base de datos de uso libre, robusto, eficiente, seguro y que se integra en el servidor web Apache.

En el lado cliente del sistema desarrollamos una aplicación destinada al sistema operativo Android, que cuenta en España con una cuota de mercado alrededor del 94% en los dispositivos móviles (Cinco Días, 2016). Hicimos uso de la tecnología de geolocalización a través de la capa de abstracción de los Servicios de Localización y de los correspondientes *frameworks* de clases disponibles en la arquitectura de Android.

8.1.3. Objetivos y metodología

Los objetivos que nos propusimos alcanzar son los siguientes:

- Realizar el análisis y diseño del sistema, empleando métodos y técnicas de modelado para describirlo.
- Conseguir un prototipo o aplicación operativa conforme a los requerimientos funcionales y no funcionales que se establezcan.
- Desplegar el sistema para comprobar su funcionamiento en un entorno real de producción.
- Procurar que la implementación y el despliegue de la aplicación siga criterios de desarrollo seguro (S-SDLC), así como proteger frente a las amenazas propias de aplicaciones on-line.

Seguimos el modelo de proceso de software en espiral, propuesto por Barry Boehm en 1988, que se caracteriza por seguir un patrón iterativo e incremental y, por tanto, nos permite desarrollar el sistema a través de iteraciones que le añaden incrementos o funcionalidades hasta lograr una versión del sistema completo (Sommerville, 2005; Pressman, 2010). El enfoque metodológico seguido queda resumido en las siguientes fases:

1. Especificaciones del sistema. Planteamiento inicial, determinación de los requisitos funcionales y no funcionales.
2. Desarrollo del sistema. Análisis y diseño del sistema. Descripción del sistema y modelado para describir, por un lado, el dominio del problema y los requerimientos y, por otro lado, el dominio de la solución conceptual.
3. Implementación. En esta fase llevamos a cabo la programación y codificación que implementan el diseño.
4. Despliegue. Puesta en producción del sistema de información y validación del mismo en cuanto al cumplimiento de los requerimientos.

Una de las prioridades en el desarrollo fue el seguimiento de un modelo de S-SDLC (ciclo de vida del desarrollo de software seguro) acorde con el modelo en espiral, iterativo e incremental. Para ello tuvimos en cuenta los “Seven Touchpoints” de McGraw (2005, citado en UNIR, 2016a) que recoge de manera priorizada prácticas de seguridad que se pueden aplicar al SDLC, en cada uno de los artefactos.

Por otro lado, dado el tipo de arquitectura prevista, teníamos que considerar los riesgos propios de aplicaciones on-line. Para identificar dichos riesgos seguimos los documentos publicados por la fundación OWASP (2013, 2017), así como la guía del Estándar de Verificación de Seguridad en Aplicaciones (ASVS por sus siglas en inglés).

8.1.4. Desarrollo de la aplicación

Para definir las funcionalidades concretas del sistema y tener una visión de conjunto del mismo así como de los diferentes subsistemas que lo componen y su funcionamiento, modelamos paulatinamente el sistema para describirlo, como labor previa a la programación. Para ello, combinamos métodos de análisis (dominio del problema) y diseño (dominio de la solución) orientado a objetos (OOA/D, *Object-Oriented Analysis and Design*). Este enfoque se centra en la definición de clases y el modo en que colaboran entre sí para alcanzar los requerimientos de la aplicación (Larman, 2004, Pressman, 2010).

El paso inicial fue elaborar el catálogo de requisitos, definidos a partir de la especificación del problema, tanto los requisitos funcionales, relativos a los servicios o funcionalidades que la aplicación ha de aportar a los usuarios finales, como los no funcionales, es decir, las condiciones tecnológicas o de otra índole que tenía que cumplir el desarrollo de la solución.

A continuación, se elaboró el modelo del sistema de información y su arquitectura, que recogieron los elementos esenciales del sistema y sus diferentes capas, así como los “límites de confianza”, esto es, las zonas donde los niveles de confianza en cuanto a las potenciales amenazas de seguridad cambiaban.

A partir de las especificaciones, para efectuar el análisis del sistema utilizamos diferentes procedimientos, en concreto nos apoyamos en diagramas de flujo de datos para describir el sistema, diagramas y escenarios de casos de uso para modelar los requerimientos y, finalmente, el modelado de las clases de análisis que podemos trazar hasta las correspondientes clases de diseño.

En cuanto al diseño, presentamos tanto una visión estática como dinámica del sistema, es decir, representamos tanto los componentes del sistema software como su comportamiento o interacción en el tiempo (secuencias). Todo ello con la finalidad de describir el sistema y como apoyo a la implementación. Utilizamos diagramas de clases de diseño, diagramas de secuencias de sistema y diagrama de componentes. El modelo lógico y físico de datos se modeló mediante el diagrama entidad relación, según el modelo que Peter Chen propuso en 1976 y notación UML.

Para el diseño de la interfaz de usuario (GUI) tuvimos en consideración que las personas pueden verse limitadas en sus capacidades físicas y cognitivas como consecuencia del envejecimiento. Siguiendo criterios de usabilidad y accesibilidad, pretendimos que la GUI se caracterizara por la sencillez (minimalismo) y la facilidad de uso. Hemos de considerar que el tamaño de la pantalla en los móviles es reducido y que se interactúa con las aplicaciones a través de determinados movimientos en la pantalla táctil, por ello era conveniente ajustar la precisión y rapidez necesarias para interactuar con la aplicación, así como el esfuerzo cognitivo para operar, por ejemplo mediante la utilización de iconos y tipografía de tamaño grande, reduciendo al mínimo el número de pasos necesarios para hacer uso de las funcionalidades, incrementando el tamaño de los botones o de la superficie de pantalla que el usuario tenga que pulsar para interactuar.

La programación de la aplicación en el lado cliente se emprendió en paralelo con la del servicio web y su conexión con la base de datos para ir efectuando pruebas funcionales a lo largo de la codificación y asegurarnos así de que la interacción cliente – servidor funcionaba conforme a lo esperado según avanzábamos. No obstante, por la dependencia que la aplicación cliente tiene de las respuestas del servidor, acometíamos en primer lugar la implementación en el lado servidor y, posteriormente, implementábamos en el lado cliente, realizando las modificaciones oportunas en las diferentes validaciones en ambos lados según resultara necesario. De forma resumida, procedimos de tal forma que (1) primero preparamos el SGBD y generamos la base de datos, (2) a continuación extendimos la API y clases que la realizaban en el lado del servidor y (3) en la aplicación cliente implementamos en primer lugar las clases de diseño y seguidamente las operaciones y servicios y la interacción con las actividades y con la interfaz de usuario.

Con respecto a los requisitos funcionales y el cumplimiento de los casos de uso, la aplicación consiguió la práctica completitud:

- Registro de usuario. Que permite al usuario generar y validar una nueva cuenta, así como activar la app para empezar a operar con ella.
- Rellenar el perfil y compartirlo con otros usuarios.
- Conocer usuarios próximos, tanto su perfil como las actividades propuestas por estos.
- Proponer y compartir actividades y mostrar interés o “apuntarse” a actividades propuestas por otros usuarios, con la finalidad de fijar encuentros para realizarlas.
- Facilitar que los usuarios puedan proponer fecha, hora y lugar, para fijar, modificar y acordar citas con otros usuarios para llevar a cabo las actividades propuestas por el usuario o por otros, así como desmarcarse de las mismas si así lo decide alguna de las partes.

Siguiendo una de las más importantes prácticas de seguridad del S-SDLC, efectuamos una revisión del código fuente, o prueba estática, mediante la herramienta HP Fortify SCA, para la aplicación cliente (Java), y con RIPS en el lado servidor (PHP).

Los test unitarios fueron realizados utilizando dos *frameworks* en correspondencia con los lenguajes de programación utilizados, JUnit 4, que viene integrado en Android Studio, se utilizó para el desarrollo de la app, escrita en Java. Por otro lado, empleamos PHPUnit 3 en el IDE de desarrollo empleado (Eclipse Neon).

En cuanto a las pruebas funcionales, se tomaron como referencia los requisitos para comprobar el correcto funcionamiento del sistema.

8.1.5. Despliegue y validación

Si tenemos en cuenta las capas de la arquitectura cliente-servidor que vertebría el sistema y la modalidad en la que la aplicación cliente se distribuiría, en la parte del cliente quedaría fuera de nuestro alcance la configuración directa del dispositivo y su mantenimiento, ya que sería el usuario el responsable de la misma. Por tanto, nuestra directa responsabilidad se vuelca en que la aplicación móvil haga un uso adecuado de los permisos de acceso a servicios del dispositivo en el que se ejecute y, sobre todo, en asegurar el funcionamiento fiable, seguro y eficiente en el lado del servidor.

Tomamos la decisión de que la instalación del servidor web fuera en equipos propios. Esta opción tiene una serie de desventajas frente a las plataformas en la nube, principalmente la menor capacidad y escalabilidad, mayor exposición a un ataque de denegación de servicio, así como a amenazas y a daños, por lo que las garantías de un funcionamiento continuado son mucho menores, pero en contrapartida teníamos unos clientes que eran importantes en el desarrollo de un trabajo académico como este, entre los que podemos citar un menor coste inicial que la solución en la nube, puesto que disponíamos del equipamiento y de conexión de banda ancha a través de fibra óptica, mayor control sobre el entorno de producción, suficiente para ejecutar el sistema como prototipo y, por encima de todo, la oportunidad de experimentar con los conocimientos adquiridos en el máster sobre administración de servidores y, en particular, acerca de su bastionado y puesta en marcha.

Como paso previo a la puesta en producción, cumplimos con otras de las buenas prácticas del modelo de S-SDLC, efectuamos un *test de penetración* con la finalidad de comprobar el comportamiento del sistema frente a diversos ataques (inyección de código, manipulación de variables de sesión) y si las medidas de seguridad implementadas eran suficientes, así como detectar errores de configuración que podían suponer algún tipo de vulnerabilidad.

Para someter a evaluación la GUI de la aplicación y su funcionamiento realizamos, por un lado, un análisis heurístico, sin usuarios, siguiendo los principios de usabilidad propuestos por Mordecki (2012) y, por otro lado, contamos con la colaboración de cinco usuarios a los que observamos mientras utilizaban la aplicación. Se les pidió que interactuaran con ella

desde su inicio y que realizaran unas determinadas acciones y después podían utilizarla de manera libre. Posteriormente, nos aportaron sus impresiones.

8.1.6. Conclusiones

El marco general de nuestro trabajo fue un proceso de software iterativo e incremental. Logramos realizar el análisis y diseño del sistema, con una orientación a objetos, empleando métodos y técnicas de modelado para describirlo. Implementamos y desplegamos un prototipo o aplicación operativa conforme a los requerimientos funcionales y no funcionales que se establecieron, comprobando su funcionamiento en un entorno real de producción. Además, seguimos buenas prácticas de desarrollo seguro (S-SDLC), teniendo en cuenta las amenazas específicas de aplicaciones on-line.

En resumen, se consiguieron los propósitos del apartado técnico y ejercitar los conocimientos adquiridos a lo largo del máster. El tercer aspecto que consideramos relevante analizar es el de la utilidad práctica, es decir, la capacidad de la aplicación para disminuir las situaciones de soledad no deseada o fomentar las relaciones sociales en los mayores. En este punto, al tratarse de una aplicación basada en la creación de una red social o de contactos, para que resulte realmente útil y consiga su propósito ha de alcanzar una determinada masa crítica de usuarios, al menos en un alcance limitado a determinadas ciudades, podemos especular que la aplicación serviría a este propósito si lo consiguiera. Consideramos que la aplicación contiene las funciones esenciales, no obstante sería conveniente dotarla de unas funcionalidades añadidas que le proporcionen más atractivo a los ojos de los usuarios y mayor seguridad en cuanto a evitar un uso fraudulento.