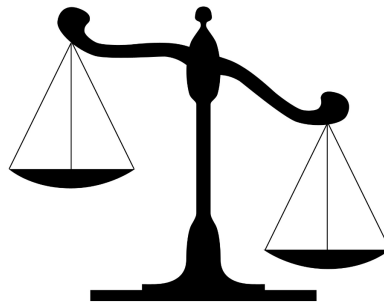


UNIVERSIDAD OBERTA DE CATALUNYA.
INGENIERIA TÉCNICA INFORMÁTICA DE SISTEMAS.



LAWYER INFO

*“SOFTWARE DE ATENCIÓN AL CLIENTE PARA DESPACHOS DE ABOGADOS EN DISPOSITIVOS MÓVILES
ANDROID”*

PROYECTO FINAL DE CARRERA.

ENERO- 2014

AUTOR: RAÚL SIERRA IGLESIAS.
CONSULTOR: ROBERTO RAMÍREZ VIQUE.

INDICE.:

- 1.- Introducción.
- 2.- Objetivos.
- 3.- Planificación.
 - 3.1 Análisis Previo.
 - 3.1.1. Estado del Arte.
 - 3.1.2. Arquitectura previa.
 - 3.2 Análisis.
 - 3.2.1. Escenario Inicial.
 - 3.2.2. Requisitos de Usuarios.
 - 3.2.3. Requisitos Funcionales.
 - 3.2.4. Requisitos no Funcionales.
 - 3.2.5. Casos de Uso.
 - 3.3 Diseño
 - 3.3.1. Arquitectura del Sistema.
 - 3.3.2. Diagrama estático de Diseño.
 - 3.3.3. Diagrama de bases de datos.
 - 3.4 Implementación.
4. Conclusiones y mejoras.

INTRODUCCIÓN.

El presente proyecto nace como propuesta para dar solución a la atención de clientes de despachos de abogados, los cuales, podrían integrar dentro de su servicio de atención al cliente este software como un valor añadido de atención personalizada.

OBJETIVOS.

El presente proyecto de fin de carrera, pretende que la comunicación entre los abogados y sus clientes sea más fluida, buscándose tanto la optimización del tiempo de los abogados como la satisfacción de los clientes por poder tener la posibilidad de estar al tanto en todo momento de en qué situación jurídica se encuentra un determinado expediente.

Como objetivos personales en la realización, se esperan alcanzar:

- Conocimiento del desarrollo para dispositivos móviles basados en la tecnología Android.
- Desarrollo e implementación del software objeto de estudio a través de dispositivos móviles android.
- Estudio y viabilidad de Web Services y Bases de Datos necesarios para el intercambio de información por parte del cliente y el servidor.

MOTIVACIÓN.

La utilidad de este software se centra en la característica de ubicuidad que tienen los dispositivos móviles para, en cualquier momento o lugar, dar solución a la necesidad que puedan tener los clientes de un despacho de abogados para conseguir información útil de alguno de los diferentes trámites de los que consta un expediente.

PLANIFICACIÓN.

El proyecto va a ser desarrollado tomando como base el ciclo de vida clásico de creación de software, es decir, el ciclo de vida en cascada el cual se compone de las siguientes etapas:

- Análisis previo a la planificación: Definición de objetivos y tareas concluyendo en la planificación del proyecto.
- Análisis: Estudio de las diferentes tecnologías que se van a utilizar en el proyecto de forma que el lector tenga una visión aproximada de cómo será el sistema final desarrollado. En este punto se especificarán cuáles son las condiciones generales que tiene el sistema y sus capacidades.
- Diseño: En esta fase se realizan las labores necesarias para definir una solución viable que cumpla con los objetivos del proyecto. Para ello debemos tener en cuenta el análisis realizado en la etapa anterior e incorporar las características y requisitos particulares para nuestra solución software.
- Implementación: En esta etapa del proyecto se habrán abordado las tareas para desarrollar la solución software y darán respuesta a las necesidades explicadas y definidas durante las dos etapas anteriores, es decir, en las etapas de análisis y de diseño de la solución software.

El diagrama de Gantt resultante del proyecto:

Estado del arte.

Introducción.

En esta parte del proyecto se lleva a cabo un estudio del contexto que rodea al desarrollo de aplicaciones para dispositivos móviles, centrándose en el análisis de los dispositivos móviles tipo smartphones.

La capacidad de modificar sus funcionalidades a través de la instalación de nuevas aplicaciones y la capacidad de conexión y uso continuo de redes de datos, ya sea a través de redes móviles o de redes Wifi, son las características principales que hacen que se diferencien del resto de terminales móviles.

Dentro de estos dispositivos smartphones podemos encontrar en el mercado diferentes plataformas entre las que destacan Android e IOS.

El desarrollo de esta proyecto dará como resultado un aplicación tipo cliente-servidor basada en tecnología Android que hará uso de las capacidades anteriormente comentadas de capacidad de conexión y uso continuo de redes de datos.

¿Por qué Android?

Android es la primera plataforma que combina una serie de características que la hacen realmente interesante de las cuáles podemos destacar:

- Plataforma abierta: Es una plataforma basada en Linux con código abierto la cual la hace interesante porque se puede modificar y usar sin tener que pagar royalties.
- Portabilidad: Las aplicaciones al estar desarrolladas en java nos asegura que podrán ejecutarse en cualquier CPU.
- Gran cantidad de servicios incorporados: Bases de datos con SQL, navegador, multimedia etc. Además de estos servicios existen una gran comunidad de desarrolladores escribiendo aplicaciones para agrandar estas funcionalidades. El android market es la tienda online administrada por Google aunque se puede conseguir software externo sin necesidad de acudir a este market.
- Nivel de seguridad en sus aplicaciones alto: Los programas en android están aislados los unos de los otros gracias a la seguridad que hereda de Linux del en el cual se basa.
- Optimización para una racionalización de los recursos: El uso de Dalvik (máquina virtual de Android) permite reducir considerablemente el tamaño del programa buscando información duplicada en las diversas clases y reutilizarla. El "recolector de basura" libera espacio en memoria de objetos que ya no se utilicen, esto se ha mejorado en Android con el fin de mantener el máximo posible de memoria libre. De igual forma, se utiliza el lenguaje de marcado XML para definir las interfaces y otros elementos, lo cual implica que a la hora de hacer uso de ellos deben ser linkados para compilarlos consiguiendo un rendimiento mejorado de las aplicaciones en la conversión a bytecode .

Versiones y niveles de API en Android.

A lo largo de la historia de android se han ido desarrollando diferentes versiones para los dispositivos por lo que es muy importante observar que hay métodos y clases que hay disponibles a partir de una versión.

A la hora de que se ha lanza una nueva versión siempre ésta ha sido compatible con la anterior, aunque evidentemente, la versión más antigua no soporte las funcionales de la nueva.

Por este motivo, hay que hacer un estudio de la versiones para poder determinar a partir de qué versión vienen incorporadas las funcionalidades necesarias para la aplicación objeto del proyecto.

A continuación describiré las plataformas lanzadas hasta la fecha explicando brevemente las nuevas novedades introducidas.

Las plataformas introducidas se identifican a través de la versión, nivel de API y nombre comercial.

- Android 1.0 nivel de API 1, fecha de lanzamiento Septiembre de 2008.

Primera versión de androide, nunca se comercializó por lo que el desarrollo para esta versión no tiene mucho sentido. Tuvo una versión mejorada para el dispositivo T- Mobile G1.

- Android 1.1 nivel de API 2, fecha de lanzamiento Febrero de 2009.

En esta versión se modificaron algunos errores de la versión anterior, existen usuarios de la misma aunque supone una parte ínfima de los usuarios de android.

- **Cupcake** . Android 1.5 nivel de API 3, fecha de lanzamiento Abril de 2009.



Primera versión con nombre comercial. En esta versión se introducen una serie de nuevas característica tanto para usuarios como para desarrolladores como son la posibilidad de teclado en pantalla con predicción de texto, widgets de escritorio y live folders, capacidad de grabación de audio y vídeo

- **Donut**. Android 1.6 nivel de API 4, fecha de lanzamiento Septiembre de 2009.



Entre las características destacadas incluye camcorder e interfaz para galería, búsqueda por voz, soporte para resoluciones WVGA, aparece un nuevo atributo XML llamado onclick que se puede especificar en la vista, Play store se mejora usando un motor de búsqueda de aplicaciones más sencillo, standards 802.1x y VPN's mejorando así la conectividad y las seguridad en las redes que utiliza, soporte multitáctil.

- **Éclair**. Android 2.0 nivel de API 5, fecha de lanzamiento Octubre 2009 .



Prácticamente no cuenta con usuarios debido a que los fabricantes pasaron directamente de la versión 1.6 a la 2.1. Como novedades se presentan una interfaz mejorada, mejoría en la sincronización de las cuentas de correo para Exchange y soporte para varias cuentas, optimización de la velocidad del hardware, teclado virtual mejorado y soporte para funciones clave para HTML5.

- Android 2.1 nivel de API 7, fecha de lanzamiento enero 2010.

Con el mismo nombre de su predecesora, destacan como nuevas funcionalidades el reconocimiento de voz como un campo texto el cual permite sustituir la entrada por teclado por la propia voz, fondos de pantalla animados, nuevos métodos incorporados con paquete Webkit para manipular bases de datos en la web.

- **Froyo.** Android 2.2 nivel de API 8, fecha de lanzamiento mayo de 2010.



De las características más destacadas están la mejora notable de la velocidad de ejecución de las aplicaciones que se elevan de 2 a 5 veces más rápido con respecto a su versión anterior 2.1.

Se mejora en las llamadas de voz, compartir contactos, soporte para Adobe flash 10.1 y la incorporación del motor de JavaScript v8 utilizado en Chrome.

Se puede instalar aplicaciones en un medio externo como una tarjeta de almacenamiento, las aplicaciones se actualizan automáticamente cuando hay alguna actualización y se facilita que las aplicaciones interaccionen con el reconocimiento de voz y que terceros proporcionen nuevos motores de reconocimiento.

Con respecto a la conectividad podemos utilizar el dispositivo para dar acceso a Internet a otros dispositivos, lo que se denomina tethering, soporte para bluetooth, para conectividad 802.11n y notificaciones push.

- **Gingerbread.** Android 2.3 nivel de API 9, fecha de lanzamiento Diciembre de 2010.



Incorpora nuevo diseño de interfaz mejorado donde se podrían destacar la capacidad de corta, copiar y pegar.

Nuevo recolector de basura de la máquina virtual dalvik mejorando así la espera en las aplicaciones ayudando a las aplicaciones de juegos a que sus animaciones tenga mejor capacidad de respuesta. Se mejora también en la gestión de la energía y en el control de sus aplicaciones.

Se cambia el sistema de ficheros a ext4 y por último cabría destacar soporte nativo para telefonía sobre Internet, soporte para NFC, sensores como giroscopio y barómetros y gestión de descargas para descargas largas.

- **Honeycomb.** Android 3.0 nivel de API 11, fecha de lanzamiento Febrero de 2011.



Como característica principal que destaca con respecto a las anteriores versiones de Android se puede comentar la optimización para pantallas grandes, esto es, para soporte de tabletas, incorporando la resolución por defecto de WXGA(1280*800).

Nuevas mejoras con respecto a los componentes como menú contextual y otras características para aprovechar las pantallas más grandes. Mejora en las notificaciones, sincronización de archivos por USB sin montar dispositivo de almacenamiento y mejora en la reproducción de animaciones 2D/3D con el render de OpenGL acelerado por hardware para sacar un mayor rendimiento de los gráficos de Android.

- Android 3.1 nivel de API 12, fecha de lanzamiento Mayo de 2011.

Se producen mejoras en la interfaz de usuario y de navegación, protocolo de transferencia de vídeo PTP/MTP y en tiempo real RTP.

- Android 3.2 nivel de API 13, fecha de lanzamiento Julio 2011.

Optimización para algunos tipos de tabletas, acceso directo para el sistema de almacenamiento SD card, API de pantalla extendida para manejar distintos tamaños de pantalla, zoom compatible de tamaño fijo.

- **Ice Cream Sandwich.** Android 4.0 nivel de API 14, fecha de lanzamiento Octubre de 2011.



Unificación de los 2 versiones anteriores 2.x y 3.x en una sola compatible con cualquier dispositivo.

Introducción de un interfaz de usuario totalmente mejorado sustituyendo el teclado físico por el virtual como mayor exponente de este cambio.

Mejoras en las redes sociales con nuevo API para su integración, cámara con detección de rostro, estabilización de la imagen, correo electrónico mejorado y calendario unificado.

Nuevo gestor de datos tráfico por internet donde podemos ver gráficamente y definir los límites de consumo con la operadora.

En Diciembre de 2011 aparece una actualización de mantenimiento, la cual no aumenta el nivel de API.

- Android 4.0.3 nivel de API 15, fecha de lanzamiento Diciembre de 2011

Alguna mejora en la API relacionadas con las redes sociales, calendario, revisor ortográfico. En Marzo del mismo año aparece otra actualización menor 4.0.4

- **Jelly Bean.** Android 4.1 nivel de API 16, fecha de lanzamiento Julio de 2012.



Se añaden mejoras en la interfaz de usuario que

repercuten en la fluidez de las aplicaciones, transiciones más ágiles y tacto mejoradas.

Se añaden notificaciones con un sistema de notificación expandible personalizada, se perfecciona el Google Search y se potencia la búsqueda por voz.

Se mejoran las comunicaciones agregando el cifrado y permitiendo actualizaciones parciales en las aplicaciones.

- Android 4.2 nivel de API 17, fecha de lanzamiento Noviembre de 2012.

Como curiosidad de esta versión de Android decir que su lanzamiento estaba previsto para Octubre en New York aunque se tuvo que retrasar debido al huracán Sandy lo que Google aprovechó para lanzarlo al mercado bajo un nuevo eslogan “A new flavour of Jelly Bean”, el cual estaba basado en un nuevo kernel de Linux 3.4.0 debutando en el evento donde también se presentaba al mercado el nuevo LG Nexus 4 y la tableta Samsung LG Nexus 10.

Las novedades incluidas en esta versión son posibilidad de fotos panorámicas, soporte para diferentes cuentas de usuarios en tabletas, todos los dispositivos utilizan el mismo interface de usuario y en general mejora en la seguridad de los dispositivos añadiendo cosas como SELinux (Security-Enhanced Linux) y always-on VPN los cuales permiten control de accesos y políticas de seguridad.

- Android 4.2.1 nivel de API 17, fecha de lanzamiento finales de Noviembre de 2012.

Corrige algunos errores en las aplicaciones usuarias en las que no se mostraba un evento en una fecha determinada y soporte para nuevos dispositivos de juegos (joysticks y gamepads) con conexión bluetooth.

- Android 4.2.2 nivel de API 17, fecha de lanzamiento Febrero de 2013.

Corrige fallos en la transmisión de audio través de bluetooth, mejora la galería de imágenes con una carga de las mismas más rápida algunos nuevos sonidos para cuando se está descargando la batería y nuevos accesos rápidos para wifi y bluetooth.

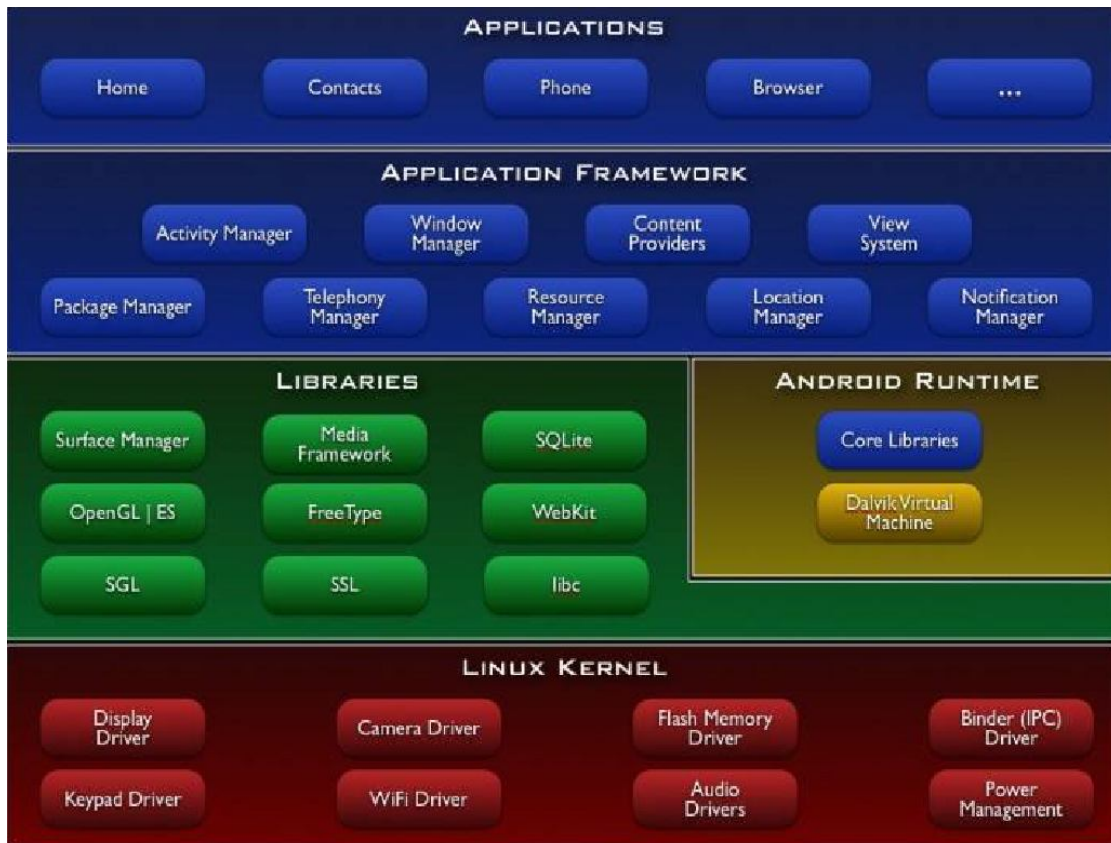
- Android 4.3 nivel de API 18, fecha de lanzamiento Julio de 2013.

Disminuye el consumo energético con conexiones bluetooth, soporte para nuevas librerías OpenGL ES 3.0 las cuales mejoran los gráficos para juegos, mejora en nuestra ubicación al permitir dejar siempre activo el Wifi para geolocalizarnos incluso estando el Wifi desconectado. Con respecto a las notificaciones cabría destacar que las aplicaciones pueden acceder y gestionar sin problemas todas nuestras notificaciones sin tener que recurrir a opciones de accesibilidad.

Una vez vistas las características de cada una de las versiones de la plataforma android se puede concluir que la aplicación objeto de estudio, denominada lawyer info, podrá ser usada a partir de la versión Android 2.2 o superior, llamada comercialmente Froyo, puesto que es la primera en introducir notificaciones push, una de las características principales de la aplicación en cuestión.

Arquitectura de Android.

La arquitectura de Android está compuesta por cuatro capas, como se puede observar en la ilustración, todas y cada una de las mismas están basadas en software libre.



- Núcleo Linux: Android depende de Linux para los servicios base del sistema como la seguridad, gestión de memoria, gestión de procesos, pila de red y controladores. El núcleo también actúa como capa de abstracción entre el hardware y el resto de la pila de software.
- Runtime de Android: Basado en el concepto de máquinas virtuales de Java. Debido a los problemas de espacio y procesamiento de los dispositivos móviles, Google optó por crear una máquina virtual nueva, Dalvik, optimizando así los recursos existentes. Dalvik ejecuta archivos en el formato Dalvik Executable (.dex), la máquina virtual está basada en registros y corre clases compiladas por el compilador Java que ha sido transformado al formato .dex por la herramienta incluida "dx".
- Bibliotecas nativas: Incluye un conjunto de librerías BSD de C/C++ usadas por varios componentes del sistema, algunas de estas son:

- System C Library que están para dispositivos embebidos basados en Linux.
 - Bibliotecas de medios, bibliotecas de gráficos 3D, SQLite para base de datos.
 - SGL motor de gráficos 2D.
 - Surface Manager, maneja el acceso al subsistema de representación gráfica 2D/3D.
 - Freetype, fuentes en bitmap y renderizado vectorial. Webkit que soporta un navegador web utilizado en Android y en la vista en webview, es el misma librería que utiliza Chrome y Safari.
 - SSL proporciona servicios de encriptación Secure Socket Layer.
- Marco de trabajo de aplicaciones: Acceso completo a las mismas APIs del Framework usados por las aplicaciones base. Las aplicaciones pueden publicar sus capacidades y otras pueden hacer uso de ellas. Los servicios más importantes que ofrece son:
- Views: conjunto de vistas, éstas son las partes visuales de los componentes.
 - Resource Manager: acceso a recursos que no son en código.
 - Notification Manager: muestra alertas personalizadas en la barra de estado.
 - Activity Manager: gestiona el ciclo de vida de las aplicaciones proporcionando sistema de navegación entre ellas.
 - Content Providers: mecanismo para acceder a datos de otras aplicaciones.
- Aplicaciones: Aplicaciones incluidas como gestor de correo, calendario, contactos y otros. Todas estas aplicaciones deben correr en la máquina Dalvik para garantizar la seguridad del sistema. Para desarrollar aplicaciones para Android en Java existe el Android SDK, en este proyecto se utilizará como se comentó en la introducción el plugin para Eclipse proporcionado Google ADT. Existen otras opciones que consisten en desarrollar en C/C++ utilizando el Android NDK (native development kit).

Web Services.

Según la W3C un posible definición podría ser un conjunto de aplicaciones o tecnologías con capacidad para interorperar entre ellas. Estas tecnología o aplicaciones se intercambian datos entre sí con el objetivo de ofrecer unos servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web. Se tratan de API que son publicadas e instaladas en un servidor y hay otras aplicaciones o servicios que pueden descubrirlas e invocar alguno de sus servicios.

Como norma general, el transporte de los datos se realiza a través del protocolo HTTP y la representación se puede hacer mediante varios formatos entre los cuales se encuentra XML y JSON.

JSON: (JavaScript Object Notation) es un formato ligero para intercambio de datos. JSON es un subconjunto de la notación literal de objetos de Javascript el cual no requiere el uso de XML.

La simplicidad de JSON ha dado lugar a la generalización de su uso como alternativa a XML.

Web Services basados en REST:

Es una técnica de arquitectura software para sistemas hipermedia distribuidos. Usa directamente el protocolo http obteniendo un serie de diseños como son:

- Protocolo cliente servidor sin estado, cada mensaje http contiene toda la información necesaria para comprender la petición, esto puede ser una desventaja puesto que cada solicitud es tratada de forma independiente por el servidor sin recordar solicitudes anteriores.
- Un conjunto de operaciones bien definidas, las más importantes son POST, GET, PUT y DELETE. Estas operaciones se asemejan a las operaciones CRUD (Create, Read, Update and Delete) que se requieren para la persistencia de datos.
- Sintaxis universal para identificar los recursos.

Deployd:

Deployd es una herramienta utilizada para la construcción simple de API's REST y javascript, las cuales nos van a proporcionar una base de datos para la realización de la parte servidora del proyecto. Está construido en base a CRUD usando funcionalidades llamados "eventos" para cada método REST. Está pensado para producir API's REST con poco esfuerzo usando Node.js como servidor y MongoDB como sistema de base de datos.

Análisis de la aplicación.

Esta sección recoge la fase del análisis del sistema una vez realizado el estudio de las tecnologías que van a utilizar. Se definirán de forma detallada las funcionalidades de las que va a constar el sistema dando una visión aproximada de el resultado final del mismo.

Escenario Inicial.

Lawyer info es una aplicación que permite gestionar desde un terminal móvil un despacho de abogados. Permite el acceso, tanto a abogados como a clientes, de cierta información relevante acerca de las gestiones que se llevan a cabo por parte de la abogados con sus relaciones con los clientes.

La aplicación se podría integrar como un valor añadido para los despachos, integrándose en lo que se denomina “protocolo de atención al cliente”, el cual se vería mejorado debido a que se obtendría mayor comunicación en las relaciones con los mismos.

Desde el punto de vista del abogado, la aplicación sería un complemento a su trabajo diario, puesto que con ella, tendría acceso directo a la información más relevante de los expedientes y cliente, pudiendo también realizar notificaciones a los mismos de los asuntos que requieran una mayor atención. Dependiendo del tipo de usuario que utilice la aplicación se pueden deducir dos tipos:

- Despacho de abogados unipersonal.
- Despacho de abogados Colectivos.

Despacho de abogados Unipersonal:

Pequeño despacho de abogados formado por un profesional.

Problemática general:

- Necesidad de movilidad permanente.
- Necesidad de atender de manera rápida a los clientes.
- Necesidad de acceder a la documentación del despachos de manera rápida desde cualquier lugar.

Casos de uso de la aplicación:

La aplicación permite a este despacho poder obtener información de sus clientes y de los procesos en los que están en trámite.

Permite ofrecer un servicio de atención a sus cliente, puesto que cada vez que lo contactan le puede dar a sus cliente información del estado de un expediente en concreto. Esta funcionalidad también estaría disponible para el cliente que solicitase el servicio de la consulta de expediente de la aplicación.

Permite realizar comunicaciones a los clientes a través de notificaciones.

Beneficios conseguidos por la aplicación:

- Ahorro en costes de desplazamientos.
- Ahorro en tiempo en la gestiones de la aplicación.

- Mejora de la imagen ante clientes.
- Reducción de horas dedicadas a clientes.

Despacho de Abogados Colectivos.

Las relaciones que pudieran tener entre los abogados pertenecientes al despacho exceden de las funcionalidades de este proyecto, por tanto la problemática, casos de uso de la aplicación y los beneficios son los mismos que si se encontraran en la situación de despacho unipersonal. Este tipo de relaciones entre abogados se propondrán como mejora futura para la aplicación.

Requisitos de Usuarios.

En el sistema se contemplan tres tipos de usuarios:

- Usuario Standard: Usuario final de la aplicación. Este tipo de usuario podrá acceder a los expedientes que tenga con el despacho en modo consulta. También podrá realizar notificaciones a su abogado.
- Usuario Abogado: Usuario final de la aplicación. Este tipo de usuario podrá acceder a la información de sus clientes y a la información de los expedientes de sus clientes.
- Usuario Administrador: Es el usuario encargado de gestionar la aplicación. Puede añadir o eliminar abogados y puede añadir o eliminar clientes. Es el usuario con más privilegios dentro de la aplicación.

Requisitos Funcionales.

El objetivo principal del sistema de información es la de permitir el acceso a la información del despacho por parte diferentes tipo de usuarios que, dependiendo de su tipología, se les facilitarán las herramientas necesarias para llevar a cabo sus tareas.

Por este motivo se ha agrupado los requisitos en 3 bloques fundamentales:

- Seguridad: Se refiere al mantenimiento de usuarios y control de acceso al sistema.
- Gestión de Clientes: Se refiere a la gestión de los clientes por parte del abogado. No hay que confundir con el usuario standard que es un usuario que interactúa con el sistema, este usuario "Cliente" es para gestión interna del abogado.
- Gestión de Expedientes: Se refiere tanto a la consulta como a la gestión de los expedientes.

A continuación se desarrollan cada uno de ellos:

Funcionalidades de seguridad:

- **Acceso a la aplicación:** Un usuario puede acceder a la aplicación usando su nombre de usuario y contraseña, dependiendo de sus credenciales accederá a la parte cliente, abogado o administrador.
- **Finalizar sesión:** Finaliza la sesión dando la posibilidad de la entrada a un nuevo usuario.
- **Cambiar clave:** permitirá al usuario validado modificar la clave de acceso al sistema.
- **Alta de usuarios:** permitirá a los administradores del sistema dar de alta a nuevos usuarios de la aplicación dándoles además el rol correspondiente (administrador, abogado, cliente).
- **Baja de usuarios:** permitirá a los administradores del sistema dar de baja a usuarios de aplicación. Como medida de seguridad en la gestión de usuarios, no será posible que un administrador se dé de baja a sí mismo.
- **Modificar usuarios:** Permitirá al administrador la modificación de los datos de usuario.

Funcionalidades de Gestión de Clientes:

Funcionalidades destinadas a la gestión de los clientes. Independientemente de que el cliente esté dado de alta o no como usuario de la aplicación, el rol abogado necesita gestionar a sus clientes.

- **Alta de Clientes:** Permitirá a los usuarios abogados introducir un nuevo cliente en el sistema para lo cual deberán rellenar la ficha correspondiente.
- **Baja de Clientes:** Se permitirá a los usuarios abogados eliminar clientes que ya no vayan a ser utilizados. Como restricción el cliente no puede tener expedientes asociados
- **Modificación de Clientes:** El abogado podrá hacer modificaciones de los datos de sus clientes.
- **Consultar Cliente:** Permitirá visualizar los datos del cliente objeto de consulta

Funcionalidades de Gestión de Expedientes:

Funcionalidades destinadas al tratamiento de los expedientes. Una vez creado un expediente nos encontramos con que este expediente puede contener diferentes procedimientos.

- **Alta de expedientes:** Permitirá a los usuarios abogados introducir un nuevo expediente en el sistema para lo cual deberán rellenar la ficha correspondiente.
- **Baja de expedientes:** Se permitirá a los usuarios abogados eliminar expedientes que ya no vayan a ser utilizados.
- **Modificación de Expedientes:** Se podrá modificar por parte de los usuarios abogados
- **Consultar Expedientes:** Permitirá visualizar los datos del expediente objeto de consulta

Procedimientos:

- **Alta de procedimientos:** Permitirá a los usuarios abogados introducir un nuevo procedimiento en el sistema para lo cual deberán rellenar la ficha correspondiente.
- **Baja de procedimientos:** Se permitirá a los usuarios abogados eliminar expedientes que ya no vayan a ser utilizados.
- **Modificación de procedimientos:** Se podrá modificar por parte de los usuarios abogados
- **Consultar procedimientos:** Permitirá visualizar los datos de los procedimientos asociados al expediente. Una vez seleccionado uno de ellos se podrá acceder a la información detallada del mismo.

Requisitos no funcionales:

El sistema deberá contar con una interfaz lo más visul y explicativa posible, no requiriendo el uso de instrucciones ni materiales de documentación.

Deberá disponer de los siguientes interfaces especializados

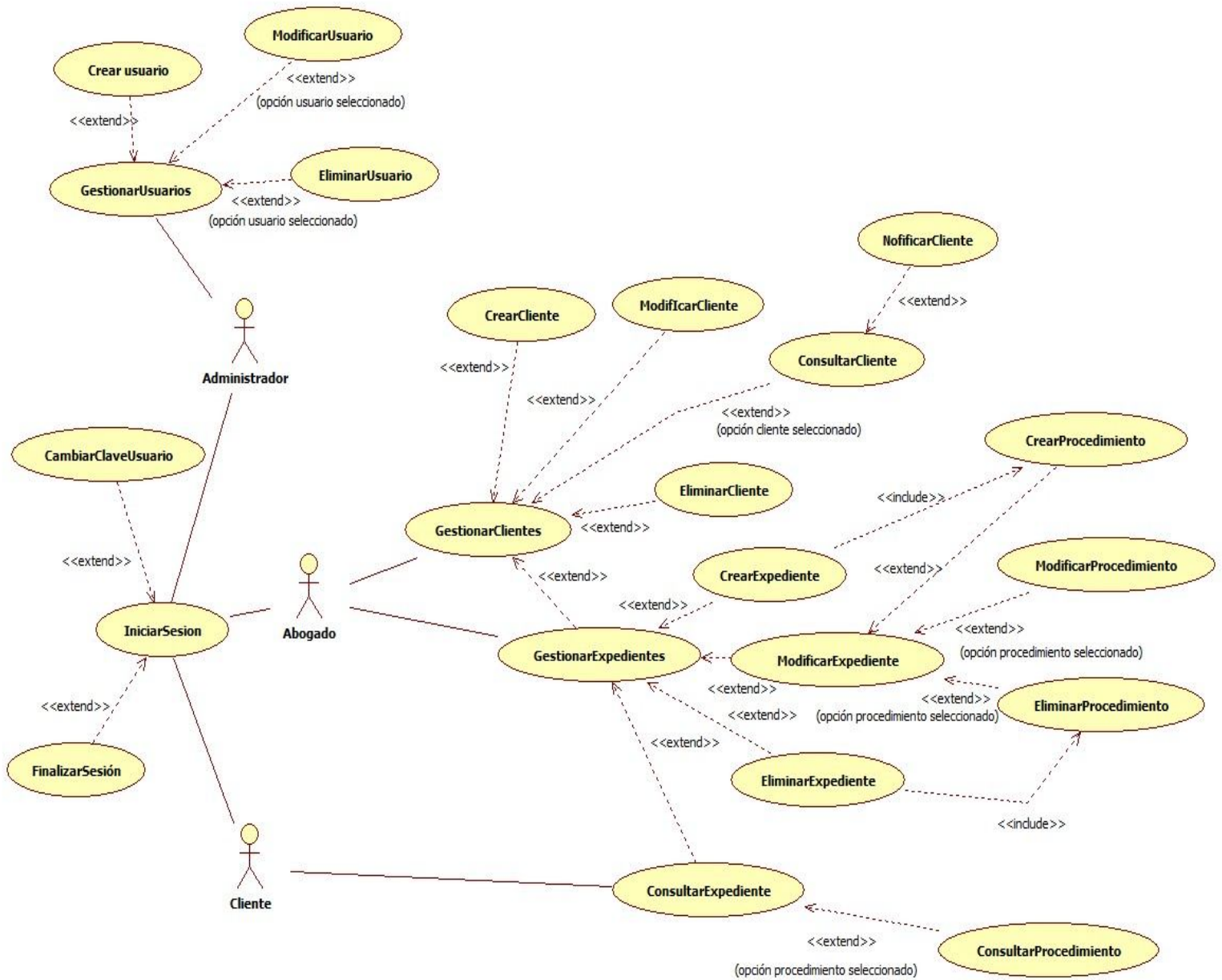
- **Pantalla inicial:** Es la pantalla que se verá al arrancar la aplicación. En la misma se presentarán las tareas de validación del usuario. Una vez validado el acceso correctamente se presentará el menú principal de cada usuario dependiendo del nivel de acceso a la aplicación.
- **Pantalla de configuración:** Una de las opciones comunes a todos los usuarios de la aplicación será la posibilidad de cambiar la contraseña de acceso.

- **Pantallas con los datos solicitados:** Dependiendo de las tareas solicitadas a la aplicación, ésta presentará los datos pedidos. En el caso de que sea una notificación, presentará una pantalla de datos diciéndole al usuario si se ha enviado correctamente o no.

Todas las pantallas dispondrán de un menú siempre visible en el cual se podrá acceder a las pantallas de la aplicación. Mostrará el título de la pantalla en la que se encuentre y los accesos a cada opción.

Casos de Uso:

Los casos de uso son las secuencias detalladas de los pasos o las actividades para llevar a cabo algún proceso en la aplicación. Los personajes que participarán en la secuencia se denominan actores. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento del sistema mediante la interacción con sus usuarios o con otros sistemas.



Descripción textual de los casos de Uso:

Se muestra una representación textual de los casos de uso con los siguientes atributos:

- **Identificador:** Muestra el caso de uso a explicar con un identificador alfanumérico tipo CUx, donde x es el identificador numérico.
- **Título:** Breve descripción del nombre del caso de uso a estudio.
- **Actores:** Actores involucrados en el caso de uso.
- **Resumen:** Pequeño resumen del caso de uso.
- **Precondición:** Condiciones previas que se tienen que dar para que se cumpla el caso de uso.
- **Postcondición:** Condiciones que se dan cuando se produce el caso de uso.
- **Flujo normal:** Flujo normal del caso de uso.
- **Flujo Alternativo:** Flujo alternativo del caso de uso.
- **Inclusiones:** inclusiones del caso de uso.
- **Exclusiones:** exclusiones del caso de uso.

Identificador	CU01
Título	IniciarSesión
Actores	Administrador, Abogado, Cliente
Resumen	Pantalla inicial de la aplicación donde los usuarios deben validarse para acceder.
Precondición	No haya ninguna sesión activa.
Postcondición	El usuario ha iniciado la sesión o bien ha cerrado la aplicación sin acceder.
Flujo Normal	1.-La aplicación solicita los datos del usuario y contraseña para darle acceso. 2.- El usuario introduce los datos. 3.- El sistema valida que los datos sean correctos. Si son correctos inicia la aplicación y muestra el menú principal de la aplicación. Si no son correctos, informa al usuario y solicita de nuevo los datos de acceso.
Flujo Alternativo	En el caso de que el usuario no introduzca los datos de acceso correctamente el sistema dará un error y volverá al paso 1.

Inclusiones	Ninguna
Extensiones	CU02(FinalizarSesión), CU03(CambiarClaveUsuario)

Identificador	CU2
Título	FinalizarSesión
Actores	Administrador, Abogado, Cliente
Resumen	Finaliza la sesión del usuario en curso.
Precondición	El actor a iniciado sesión anteriormente.
Postcondición	No hay ninguna sesión activa por parte del usuario.
Flujo Normal	1.- El usuario solicita el cierre de la sesión. 2.- El sistema solicita la confirmación del cierre de sesión. 3.- El actor confirma el cierre de sesión. 4.- El sistema finaliza la sesión actual volviendo al caso de uso CU01 (iniciar sesión).
Flujo Alternativo	En el caso de que el usuario cancele la confirmación la aplicación volverá al menú principal.
Inclusiones	Ninguna.
Extensiones	Extiende a CU01

Identificador	CU03
Título	CambiarClaveUsuario
Actores	Administrador, abogado y cliente
Resumen	Caso de uso en el que se da la opción al usuario de cambiar su clave de acceso a la aplicación
Precondición	El usuario está validado en la sesión.
Postcondición	El usuario modifica su clave de acceso al sistema o cancela el proceso.
Flujo Normal	La aplicación solicita los datos del usuario para darlo de alta. El usuario introduce los datos. El sistema valida que los datos no estén duplicados y los guarda en la base de datos.
Flujo Alternativo	Si la clave anterior no coincide se muestra un error y se vuelve al menú principal de la aplicación. Si no coinciden los dos valores de la clave nueva y la confirmación se muestra un error y se vuelve al menú principal de la aplicación.
Inclusiones	Ninguna.
Extensiones	Extiende a CU01

Identificador	CU04
Título	GestionarUsuarios
Actores	Administrador
Resumen	Se inicia cuando el administrador selecciona esta opción de su menú principal y le da acceso a la gestión de usuarios.
Precondición	El usuario está validado en la sesión.
Postcondición	El usuario accede a las opciones de gestión de usuarios.
Flujo Normal	1.- El administrador una vez entra en la gestión de usuario la aplicación le muestre los usuarios dados de alta hasta el momento y las opciones de gestión para los mismos. 2.- Una vez termina la gestión el administrador sale de la gestión y vuelve a su menú principal
Flujo Alternativo	El usuario puede iniciar los CU06 y CU07, seleccionando uno de los usuarios mostrados en pantalla. Una vez finalizados vuelve al punto 1 del flujo normal. El usuario puede iniciar el CU05(CrearUsuario), una vez termina este caso de uso vuelve al punto 1 del flujo normal.
Inclusiones	Ninguna.
Extensiones	CU05(CrearUsuario), CU06(ModificarUsuario), CU07(EliminarUsuario).

Identificador	CU05
Título	CrearUsuario
Actores	Administrador
Resumen	Da de alta a un usuario, dando el perfil de administrador, abogado o cliente.
Precondición	El usuario no está dado de alta en la aplicación.
Postcondición	El usuario está dado de alta en la aplicación con el rol seleccionado.
Flujo Normal	1.- La aplicación solicita los datos del usuario para darlo de alta. 2.- El actor introduce los datos.

Flujo Alternativo	3.- El sistema valida que los datos no estén duplicados y los guarda en la base de datos. El usuario está duplicado y la aplicación muestra un error retornando al paso 1. El actor cancela el alta antes de la confirmación y finaliza el caso de uso.
Inclusiones	Ninguna.
Extensiones	Extiende a CU04

Identificador	CU06
Título	ModificarUsuario
Actores	Administrador
Resumen	Modifica los datos del usuario seleccionado anteriormente.
Precondición	El usuario está dado de alta en la aplicación.
Postcondición	El actor ha modificado los datos del usuario correctamente.
Flujo Normal	1.- El administrador selecciona el usuario mostrado por el CU04 y le da a la opción modificar usuario. 2.- El administrador modifica los datos que desea modificar. 3.- El sistema le pregunta si desea modificar ese usuario. 4.- El administrador confirma la modificación El sistema modifica los datos en la base de datos.
Flujo Alternativo	El actor puede cancelar la modificación en el paso 3 del flujo normal y finaliza el caso de uso.
Inclusiones	Ninguna.
Extensiones	Extiende a CU04

Identificador	CU07
Título	EliminarUsuario
Actores	Administrador
Resumen	Da de baja a un usuario del sistema
Precondición	El usuario está dado de alta en la

Postcondición	aplicación. El usuario está dado de baja en la aplicación.
Flujo Normal	1.- El administrador selecciona el usuario mostrado por el CU04 y le da a la opción eliminar usuario. 2.- El sistema le pregunta si desea eliminar ese usuario seleccionado. 4.- El administrador confirma la eliminación. El sistema elimina los datos de ese usuario en la base datos.
Flujo Alternativo	El actor puede cancelar la eliminación en el paso 3 del flujo normal y finaliza el caso de uso.
Inclusiones	Ninguna.
Extensiones	Extiende a CU03

Identificador	CU08
Título	GestionarClientes
Actores	Abogado
Resumen	Se inicia cuando el abogado selecciona esta opción de su menú principal y le da acceso a la gestión de clientes.
Precondición	El abogado está validado en la sesión.
Postcondición	El abogado accede a las opciones de gestión de clientes.
Flujo Normal	1.- El abogado una vez entra en la gestión de clientes de la aplicación le muestra sus clientes dados de alta hasta el momento y las opciones de gestión para los mismos. 2.- Una vez termina la gestión el abogado sale del caso de uso y vuelve a su menú principal
Flujo Alternativo	El usuario puede iniciar los CU010 y CU011, seleccionando uno de los clientes mostrados en pantalla. Una vez finalizados vuelve al punto 1 del flujo normal. El usuario puede iniciar el CU09(CrearUsuario), una vez termina este caso de uso vuelve al punto 1 del flujo normal.
Inclusiones	Ninguna.
Extensiones	CU09(CrearCliente),

	CU010(ModificarCliente), CU011(EliminarCliente).
--	---

Identificador	CU09
Título	CrearCliente
Actores	Abogado
Resumen	Da de alta a un nuevo cliente para su gestión.
Precondición	El cliente no está dado de alta en la gestión de clientes de la aplicación.
Postcondición	El cliente está dado de alta en la gestión de clientes de la aplicación.
Flujo Normal	1.- La aplicación solicita los datos del cliente para darlo de alta. 2.- El actor introduce los datos. 3.- El sistema valida que los datos no estén duplicados y los guarda en la base de datos.
Flujo Alternativo	El cliente está duplicado y la aplicación muestra un error retornando al paso 1. El actor cancela el alta antes de la confirmación y finaliza el caso de uso.
Inclusiones	Ninguna
Extensiones	Extiende a CU08

Identificador	CU010
Título	ModificarCliente
Actores	Abogado
Resumen	Modifica los datos del cliente seleccionado anteriormente.
Precondición	El cliente está dado de alta en la aplicación.
Postcondición	El actor ha modificado los datos del cliente correctamente.
Flujo Normal	1.- El abogado selecciona el cliente mostrado por el CU08 y le da a la opción modificar cliente. 2.- El abogado modifica los datos que desea modificar. 3.- El sistema le pregunta si desea modificar ese cliente.

Flujo Alternativo	4.- El abogado confirma la modificación El sistema modifica los datos en la base de datos.
Inclusiones	El actor puede cancelar la modificación en el paso 3 del flujo normal y finaliza el caso de uso.
Extensiones	Ninguna. Extiende a CU08

Identificador	CU011
Título	ConsultarCliente
Actores	Abogado
Resumen	Consulta los datos del cliente seleccionado anteriormente.
Precondición	El cliente está dado de alta en la aplicación.
Postcondición	El actor consulta los datos del cliente correctamente.
Flujo Normal	1.- El abogado selecciona el cliente mostrado por el CU08 y el sistema le muestra los datos del cliente. 2.- El abogado consulta los datos del cliente y finaliza el caso de uso.
Flujo Alternativo	El actor puede cancelar la consulta de los datos del cliente y finaliza el caso de uso. El actor puede iniciar el CU012(NotificarCliente), una vez termina este caso de uso vuelve al punto 1 del flujo normal.
Inclusiones	Ninguna.
Extensiones	CU012 Extiende a CU08

Identificador	CU013
Título	EliminarCliente
Actores	Abogado
Resumen	Da de baja a un cliente del sistema
Precondición	El cliente está dado de alta en la aplicación.

Postcondición	El usuario a eliminar no puede tener expedientes asociados.
Flujo Normal	El usuario está dado de baja en la aplicación. 1.- El administrador selecciona el usuario mostrado por el CU04 y le da a la opción eliminar usuario. 2.- El sistema le pregunta si desea eliminar ese usuario seleccionado. 4.- El administrador confirma la eliminación. El sistema elimina los datos de ese usuario en la base datos.
Flujo Alternativo	El actor puede cancelar la eliminación en el paso 3 del flujo normal y finaliza el caso de uso. El sistema detecta que el cliente tiene asociados expedientes y genera un error avisando de la cancelación de la operación y finalizando el caso de uso.
Inclusiones	Ninguna.
Extensiones	Extiende a CU08

Identificador	CU014
Título	GestionarExpedientes
Actores	Abogado
Resumen	Se inicia cuando el abogado selecciona esta opción de su menú principal y le da acceso a la gestión de gestión de expedientes.
Precondición	El abogado está validado en la sesión.
Postcondición	El abogado accede a las opciones de gestión de expedientes.
Flujo Normal	1.- El abogado una vez entra en la gestión de expedientes de la aplicación le muestra sus expedientes dados de alta hasta el momento y las opciones de gestión para los mismos. 2.- Una vez termina la gestión el abogado sale del caso de uso y vuelve a su menú principal
Flujo Alternativo	El abogado puede iniciar los CU017, CU018 y CU019, seleccionando uno de los expedientes mostrados en pantalla. Una vez finalizados vuelve al punto 1 del flujo normal.

Inclusiones Extensiones	El usuario puede iniciar el CU015(CrearExpediente), una vez termina este caso de uso vuelve al punto 1 del flujo normal. Ninguna. CU015(CrearExpediente), CU017(ModificarExpediente), CU018(EliminarExpediente). CU019(ConsultarExpediente).
----------------------------	---

Identificador	CU015
Título	CrearExpediente
Actores	Abogado
Resumen	Da de alta a un nuevo expediente para su gestión.
Precondición	El expediente no está dado de alta en la gestión de expedientes de la aplicación.
Postcondición	El expediente está dado de alta en la gestión de expedientes de la aplicación.
Flujo Normal	1.- La aplicación solicita los datos del expediente para darlo de alta. 2.- El actor introduce los datos. 3.- El sistema valida que los datos no estén duplicados. 4.- El sistema inicia CU016 (CrearProcedimiento) 5.- El sistema confirma la creación del expediente con el procedimiento asociado.
Flujo Alternativo	El expediente está duplicado y la aplicación muestra un error retornando al paso 1. EL proceso de CrearProcedimiento falla y la aplicación muestra el error retornando al paso 1. El actor cancela el alta antes de la confirmación y finaliza el caso de uso.
Inclusiones	CU016(CrearProcedimiento)
Extensiones	Extiende a CU014

Identificador	CU016
Título	CrearProcedimiento
Actores	Abogado
Resumen	Da de alta a un nuevo procedimiento para su gestión.
Precondición	El procedimiento no está dado de alta en la gestión de procedimientos de la aplicación.

Postcondición	El procedimiento está dado de alta en la gestión de procedimientos de la aplicación.
Flujo Normal	1.- La aplicación solicita los datos del procedimiento para darlo de alta. 2.- El actor introduce los datos. 3.- El sistema valida que los datos no estén duplicados y los guarda en la base de datos.
Flujo Alternativo	El procedimiento está duplicado y la aplicación muestra un error retornando al paso 1. El actor cancela el alta antes de la confirmación y finaliza el caso de uso.
Inclusiones	Incluye a CU015
Extensiones	Extiende a CU017

Identificador	CU017
Título	ModificarExpediente
Actores	Abogado
Resumen	Modifica los datos del expediente seleccionado anteriormente.
Precondición	El expediente está dado de alta en la aplicación.
Postcondición	El actor ha modificado los datos del expediente correctamente.
Flujo Normal	1.- El abogado selecciona el expediente mostrado por el CU015 y le da a la opción modificar expediente. 2.- El abogado modifica los datos que desea modificar. 3.- El sistema le pregunta si desea modificar ese expediente. 4.- El abogado confirma la modificación El sistema modifica los datos en la base de datos. El usuario puede iniciar el CU018, seleccionando uno de los Procedimientos mostrados en pantalla. Una vez finalizados vuelve al punto 1 del flujo normal.

Flujo Alternativo	El usuario puede iniciar el CU016(CrearProcedimiento), una vez termina este caso de uso vuelve al punto 1 del flujo normal. El usuario puede iniciar el CU018(ModificarProcedimiento), una vez termina este caso de uso vuelve al punto 1 del flujo normal.
Inclusiones	El actor puede cancelar la modificación en el paso 3 del flujo normal y finaliza el caso de uso.
Extensiones	Ninguna. CU016(CrearProcedimiento) CU018(ModificarProcedimiento). CU020(EliminarProcedimiento) Extiende a CU014

Identificador	CU018
Título	ModificarProcedimiento
Actores	Abogado
Resumen	Modifica los datos del procedimiento seleccionado anteriormente en el CU017.
Precondición	El procedimiento está dado de alta en la aplicación.
Postcondición	El actor ha modificado los datos del procedimiento correctamente.
Flujo Normal	1.- El abogado selecciona el procedimiento mostrado por el CU017 y le da a la opción modificar procedimiento. 2.- El abogado modifica los datos que desea modificar. 3.- El sistema le pregunta si desea modificar ese procedimiento. 4.- El abogado confirma la modificación El sistema modifica los datos en la base datos.
Flujo Alternativo	El actor puede cancelar la modificación en el paso 3 del flujo normal y finaliza el caso de uso.
Inclusiones	Ninguna.
Extensiones	Extiende a CU017

Identificador	CU019
Título	EliminarExpediente

Actores	Abogado
Resumen	Da de baja un expediente del sistema
Precondición	El expediente está dado de alta en la aplicación.
Postcondición	El expediente a eliminar no puede tener procedimientos asociados. El expediente está dado de baja en la aplicación.
Flujo Normal	1.- El abogado selecciona el expediente mostrado por el CU015 y le da a la opción eliminar expediente. 2.- El sistema verifica si el expediente tiene procedimientos asociados. 3.- El sistema le pregunta si desea eliminar ese expediente seleccionado. 3.- El abogado confirma la eliminación. 4.- El sistema verifica que ese expediente no tenga procedimientos asociados. 5.- El sistema elimina los datos de ese expediente en la base datos.
Flujo Alternativo	El actor puede cancelar la eliminación en el paso 3 del flujo normal y finaliza el caso de uso. El sistema detecta que el expediente tiene asociados procedimientos activos y le da la opción al abogado de iniciar CU020. El abogado inicia CU020 y pasa al punto 2 del flujo normal.
Inclusiones	CU020
Extensiones	Extiende a CU015

Identificador	CU020
Título	EliminarProcedimiento
Actores	Abogado
Resumen	Da de baja un procedimiento del sistema
Precondición	El procedimiento está dado de alta en la aplicación.
Postcondición	El procedimiento está dado de baja en la aplicación.
Flujo Normal	1.- El abogado selecciona el procedimiento mostrado en los CU018 o CU019 y le da a la opción eliminar procedimiento. 2.- El sistema le pregunta si desea eliminar el procedimiento seleccionado. 3.- El abogado confirma la eliminación.

Flujo Alternativo	4.- El sistema elimina los datos de ese procedimiento en la base datos. El actor puede cancelar la eliminación en el paso 3 del flujo normal y finaliza el caso de uso.
Inclusiones	Incluye a CU019(EliminarExpediente)
Extensiones	Extiende a CU018 (ModificarExpediente)

Identificador	CU021
Título	ConsultarExpediente
Actores	Abogado, Usuario Standard
Resumen	Consulta los datos del expediente seleccionado anteriormente.
Precondición	El actor está dado de alta en la aplicación.
Postcondición	El actor consulta los datos del expediente correctamente.
Flujo Normal	1.- El actor selecciona el expediente a consultar 2.- El actor consulta los datos del expediente y finaliza el caso de uso.
Flujo Alternativo	El actor puede cancelar la consulta de los datos del expediente y finaliza el caso de uso. El actor puede iniciar CU022(ConsultaProcedimiento), una vez termina este caso de uso vuelve al punto 1 del flujo normal del caso de uso.
Inclusiones	Ninguna.
Extensiones	CU022 Extiende a CU017

Identificador	CU022
Título	ConsultarProcedimiento
Actores	Abogado, Usuario Standard
Resumen	Consulta los datos del procedimiento seleccionado anteriormente.
Precondición	El actor está dado de alta en la aplicación.
Postcondición	El actor consulta los datos del procedimiento correctamente.
Flujo Normal	1.- El actor selecciona el procedimiento a consultar 2.- El actor consulta los datos del procedimiento y finaliza el caso de uso.

Flujo Alternativo	El actor puede cancelar la consulta de los datos del procedimiento y finaliza el caso de uso.
Inclusiones	Ninguna.
Extensiones	Extiende a CU021

Diseño de la Aplicación.

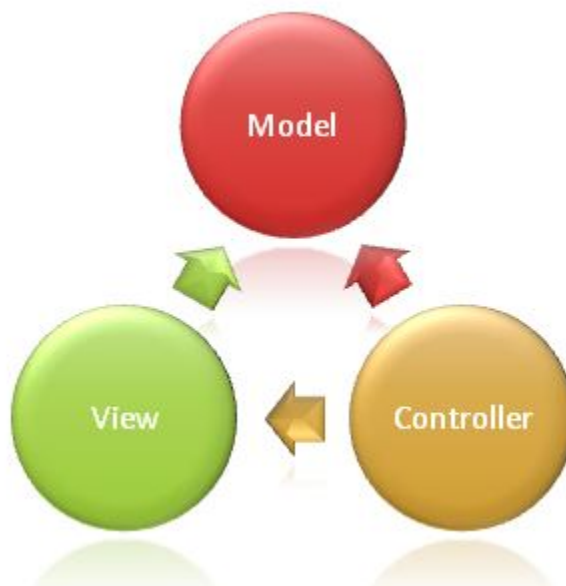
En esta parte del proyecto se tratará el diseño realizado para aplicación que se desea conseguir, cumpliendo con los requisitos y prerrequisitos realizados en la etapa de análisis comentados anteriormente.

Arquitectura del Sistema.

Para la arquitectura de este sistema se va a seguir el patrón de arquitectura de software llamado modelo vista controlador(MVC). Es una solución marcada en gran medida por el carácter distribuido de la aplicación.

Este tipo de modelo de arquitectura separa los datos y la lógica de negocio de la aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, por un lado, define componentes para la representación de la información y por otro para la interacción del usuario.

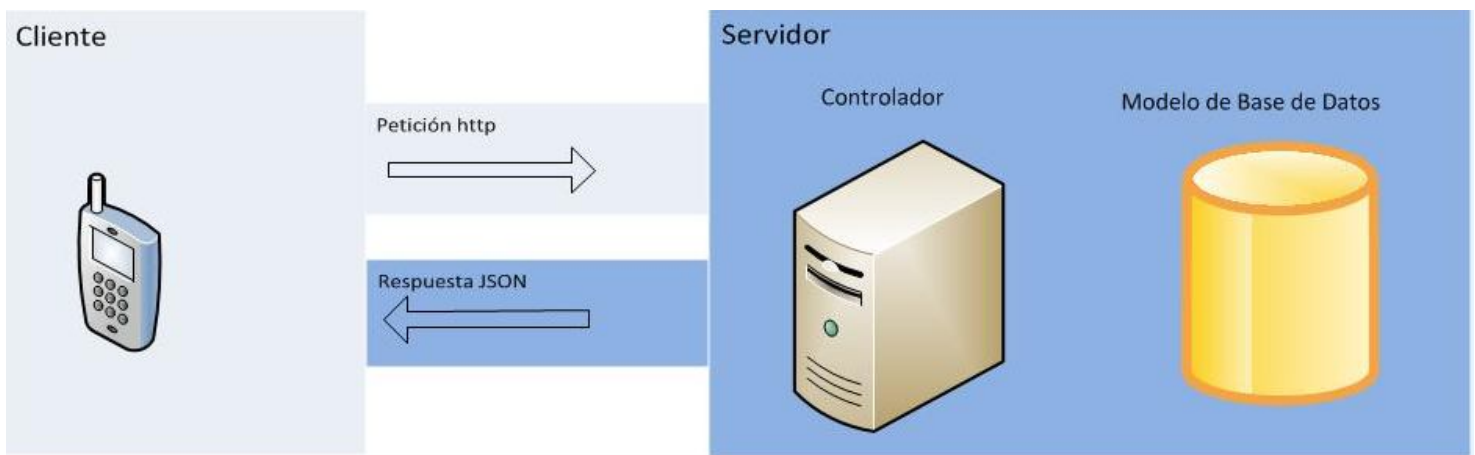
- **Modelo:** En esta parte se programará todo aquello lo relacionado con las bases de datos, es decir, todas las entradas y salidas de datos y se devuelven como se necesiten en el programa principal. Concretando en nuestro en esta parte accederemos a los datos alojados en un servidor de base de datos a través de una API. En nuestro caso hemos elegido deployd.
- **Vista:** En esta parte se programa la parte visual del software, es decir, la parte que usará el usuario con su interacción con el sistema.
- **Controlador:** Es la lógica del programa, es decir, la parte en la que se piden los datos al modelo y los muestra la vista.



Modelo Cliente- Servidor.

En este modelo se dividen dos partes diferenciadas que son:

- **Cliente:** En nuestro caso será la aplicación para el terminal móvil la cual realizará las funciones relacionadas con la manipulación y el despliegue de datos
- **Servidor:** Es el encargado de atender las peticiones solicitadas por los clientes.



Arquitectura del sistema

Los dos servicios fundamentales de la aplicación son los siguientes:

- Acceso a datos para la gestión de usuarios, clientes y expedientes: En esta parte se ha elegido como parte servidora la herramienta **deployd**.

Los motivos fundamentales por los que elegir deployd es que aporta una solución muy completa para obtener una base de datos en Internet funcionando en muy poco tiempo. Esto hace que puedas centrarte en la parte principal del proyecto que es el desarrollo móvil en android sin tener que preocuparte demasiado en la parte servidora. Deployd ofrece abundante documentación, ofrece no sólo una base de datos para que podamos cargar lo que queramos, ideal para el entorno de trabajo en el que nos estamos moviendo sino una API REST y una API javascript que permite que nos podamos desentender prácticamente de la parte servidora.

Los pasos a seguir para poder enviar peticiones al servidor y obtener los datos requeridos son:

- 1.- El usuario envía la petición al servicio web
- 2.- La aplicación crea una solicitud http y la envía a la dirección URL del servidor a través de una petición JSON.
- 3.- La petición llega al controlador que la gestiona y consulta la base de datos.
- 4.- El servidor crea una respuesta http que incluye la respuesta JSON del Servicio Web enviándolo al dispositivo móvil.
- 5.- El dispositivo recibe, procesa los datos obtenidos y los muestra por pantalla.

Diagrama Estático de Diseño

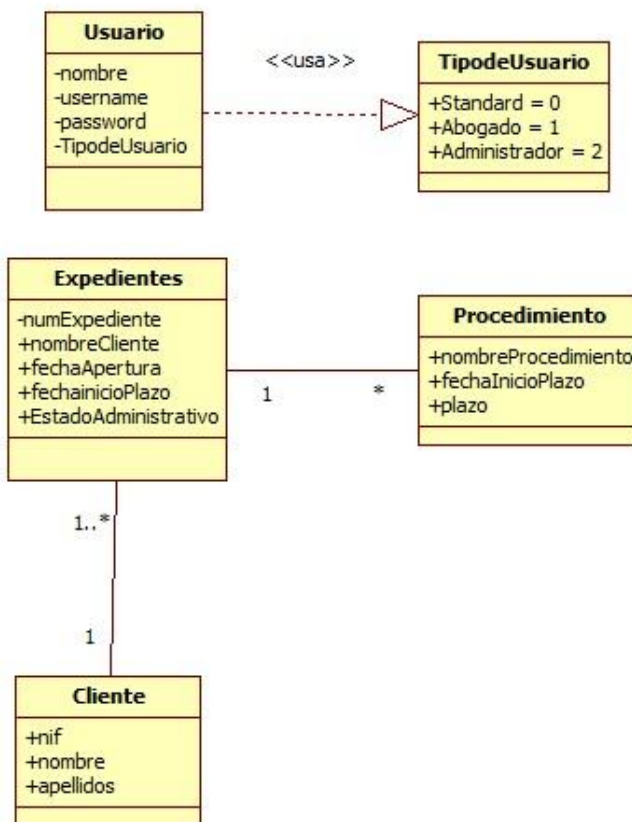
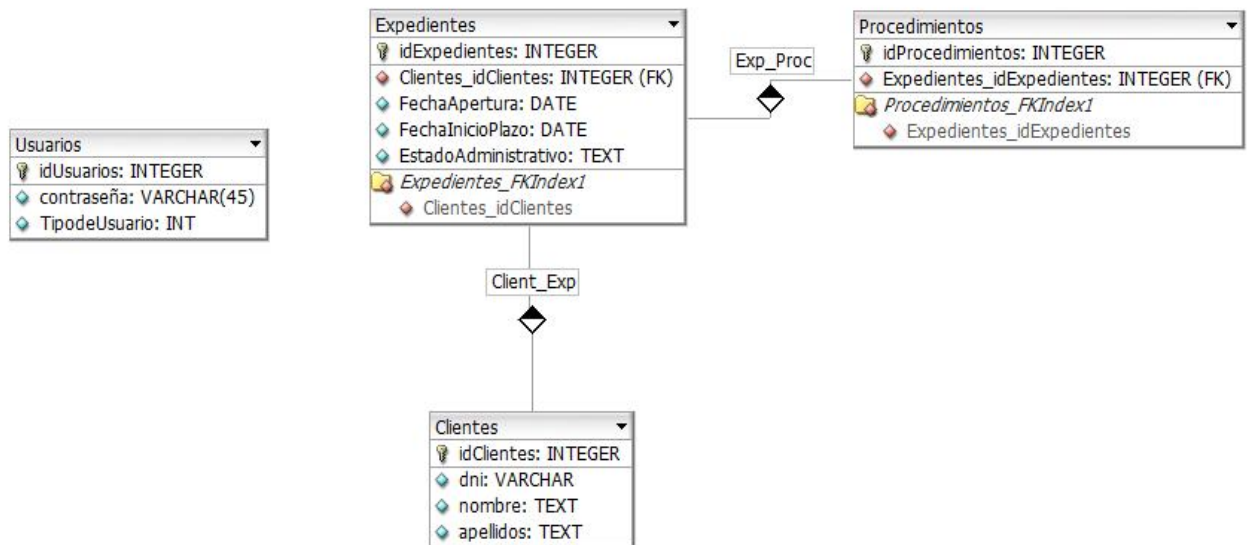


Diagrama de la base de datos.



Conclusiones y Mejoras.

Como conclusión al desarrollo de este proyecto se pueden decir que los objetivos propuestos al principio del mismo se han cumplido, entre los cuales se pueden destacar:

- Adquisición de conocimiento para el desarrollo de aplicaciones para dispositivos móviles basados en la tecnología android.
- Estudio y viabilidad de Web Services y Bases de datos necesarios para el intercambio de información entre la parte cliente y servidora.
- Adquisición de nuevos conocimientos del lenguaje de programación JAVA, formatos de intercambio de información como JSON y APIs de desarrollo rápido como Deployd.

Lawyer- Info está completamente desarrollada tomando como modelo de datos la red, por tanto se puede usar sólo con conexión a la misma, por lo que en un futuro se podría mejorar este aspecto, minimizando el uso de ésta. Por otro lado también se propone como mejora la gestión de ésta falta de conectividad.

Esta aplicación ha sido desarrollada pensando como cliente final un despacho de abogados con un solo profesional. La aplicación se podría mejorar por tanto, desde el punto de vista de despachos con varios letrados en la organización., como herramienta de comunicación interna entre los mismos De la misma manera, Lawyer- info podría mejorar los procesos de comunicación interna para organizaciones con un volumen de personal mucho mayor.

Bibliografía.

- Android Wikipedia.
<http://es.wikipedia.org/wiki/Android>.
- Google Developers. 2013.
<https://developers.google.com/>
- Deployd.
<http://www.deployd.com>
- Json.
<http://es.wikipedia.org/wiki/JSON>
<http://www.json.org/>
- Java.
<http://docs.oracle.com/javase/7/docs/api/>
- Memorias de proyectos realizados.
<http://www.uoc.com>