UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO FIN DE GRADO

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN CLIENTE DE CORREO ELECTRÓNICO SEGURO PARA ANDROID

Autor: Adrián Cáceres Domínguez
Tutor: Agustín Orfila Díaz-Pabón

Junio 2012

Índice

Índice de tablas

Índice de ilustraciones

1. Introducción

1.1. Motivación

En la actualidad el uso de teléfonos inteligentes (*smartphones* en inglés) está ampliamente extendido [1], y con ellos el uso de aplicaciones de correo electrónico en dichos dispositivos.

El correo electrónico se utiliza para enviar todo tipo de información, entre la cual se incluyen datos confidenciales. Aun así el correo electrónico fue diseñado sin tener en cuenta las medidas de seguridad referentes a la confidencialidad, integridad y no repudio. Por lo que los mensajes enviados y/o recibidos son transmitidos con su contenido en claro, pudiendo un usuario malintencionado interceptar o manipular estos datos. Debido a esto, es conveniente la incorporación de mecanismos para garantizar la confidencialidad, integridad y no repudio de los correos electrónicos.

En base a esto, hemos investigado las opciones existentes en el mercado para los dos grandes sistemas operativos móviles (iOS y Android), comprobando que para el sistema operativo iOS ya existen aplicaciones que cubren esta necesidad (la aplicación de correo electrónico de Apple [2]) y que para Android no existe una aplicación que cumpla los principios básicos necesarios.

Teniendo en cuenta lo anterior, nos hemos centrado en el desarrollo de una aplicación de correo electrónico para el sistema operativo Android, que permite enviar y recibir correos de forma segura.

1.2. Objetivos

El objetivo de este trabajo de fin de grado es incorporar servicios de seguridad al envío, recepción y almacenamiento de los correos electrónicos en una aplicación para el sistema operativo Android. Concretamente estos servicios son: confidencialidad, integridad, autenticación y no repudio. Aunque existen aplicaciones que ya hacen uso

de algunas de los servicios nombrados, no hemos encontrado ninguna que los cumpla todos y que sea de software libre y compatible con todos los servidores de correos.

Los objetivos a cumplir son:

1) Enviar correos firmados y/o cifrados.
2) Recibir correos firmados y poder validar la firma.
3) Recibir correos cifrados y poder descifrarlos.
4) Recibir correos firmados y cifrados, y poder descifrarlos y verificar su firma.
5) Tener un gestor de contactos mediante el cual se gestionen los certificados de clave pública de los mismos.
6) Guardar los datos de la aplicación de forma segura.

Para lograr estos objetivos se estudiarán los requisitos que debe cubrir la aplicación, así como analizar que otras opciones existen en el mercado en el momento del desarrollo del Trabajo de Fin de Grado. Y diseñara e implementará un sistema con los requisitos de la fase anterior.

1.3. Fundamentos básicos

1.3.1. Correo electrónico

El correo electrónico es un método de intercambio de mensajes digitales, el cual se encuentra definido por varios RFC, con los que se detallan diferentes partes: el formato de los mensajes [3], el protocolo de envío de correos como SMTP [4], y los protocolos de recepción POP3 [5] e IMAP [6].

En el formato de los correos electrónicos básico se especifica como son los campos requeridos para poder enviar correos de solo texto, es decir sin imágenes, ni ficheros... En él también se especifica que campos son requeridos y que formato tienen que tener.

Los únicos campos obligatorios son la fecha de origen y origen del mensaje. Además se especifica que solo está permitido utilizar caracteres de tipo US-ASCII [7] que estén entre el rango de 1 y 127 de valor decimal.

Para poder enviar contenido diferente a texto por correo se creó el estándar MIME, que detallaremos en el punto 1.3.5.

1.3.2. SMTP

El protocolo SMTP (*Simple Mail Transfer Protocol*) se basa en un modelo cliente-servidor y es el encargado de enviar los mensajes de correo electrónico del usuario al servidor de correos. Más concretamente se encarga de enviar el correo a nuestro servidor SMTP y este servidor dejara el mensaje en el servidor SMTP del destinatario, para que pueda obtenerlo mediante POP3 o IMAP.

1.3.3. POP3

El protocolo POP3 (*Post Office Protocol* versión 3) también está basado en la arquitectura de cliente-servidor, pero en este caso se utiliza para obtener los correos electrónicos almacenados en el servidor. En el caso de POP3 solo puede acceder a una cuenta de correo desde una aplicación a la vez. El uso normal del protocolo POP3 es conectarse al servidor descargarse los mensajes y eliminarlos del servidor, aunque hay muchos clientes de correo electrónico que dan la opción de mantenerlos en el servidor una vez descargados.

1.3.4. IMAP

Al igual que el protocolo POP3, IMAP (*Internet Message Access Protocol*) es un protocolo utilizado para la descargas de los correos desde el servidor. Pero a diferencia de POP3 permite múltiples accesos desde diferentes clientes de correo, gestión de carpetas en el servidor, mantener el estado de los mensajes en el servidor (ej. el estado de leído o no del mensaje). Por todo esto y algunas diferencias más el protocolo IMAP permite gestionar de mejor manera los mensajes en el servidor, aunque tiene la desventaja que es más complejo que el protocolo POP3.

1.3.5. MIME

Como el estándar de correo electrónico solo permite enviar texto plano, se desarrolló el estándar MIME [8-13] (*Multi Purpose Internet Mail*) en el cual se especifica cómo definir diferentes partes de un mensaje y de qué tipo es cada parte. Para ellos se añade una nueva cabecera al mensaje indicando que es de tipo MIME (*MIME-Version*: 1.0).

A continuación detallaremos algunas de las características de estándar MIME. En cada parte del mensaje se define mínimo dos cabeceras:

- ***Content-Type*:** Específica el tipo de contenido de esa parte, a continuación se muestran algunos de los tipos.

Tipo MIME	Tipo de archivo	Extensión asociada
image/gif	Imágenes GIF	gif
image/png	Imágenes PNG	png
image/jpeg	Imágenes JPEG	jpg, jpeg, jpe
text/plain	Archivos de texto sin formato	txt, g, h, c
text/html	Archivos HTML	htm, html

Tabla 1 - Tipos básicos MIME

También existen otros tipos MIME que permiten el uso de varias partes, es decir poder enviar varios elementos en un mismo correo.

Tipo MIME	Significado
multipart/mixed	Contenido con múltiples partes
multipart/alternative	Contenido con partes alternativas
multipart/related	Contenido con partes relacionadas
multipart/parallel	Orden de las partes irrelevante

Tabla 2 - Tipos multipart MIME

Un ejemplo de un mensaje MIME sería:

Content-Type: multipart/alternative; boundary="_----------=_MCPart_293616404"
MIME-Version: 1.0

This is a multi-part message in MIME format

--_----------=_MCPart_293616404
Content-Type: text/plain; charset="utf-8"; format="fixed"
Content-Transfer-Encoding: quoted-printable

Ejemplo de mensaje con tipos MIME, con codificaci=F3n Quoted
--_----------=_MCPart_293616404

```
Content-Type: text/html; charset="utf-8"
Content-Transfer-Encoding: quoted-printable

<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv=3D"Content-Type" content=3D"text/html; charset=3D=
UTF-8">
    <title>Ejemplo de mensaje</title>
    </head>
  <body>
    Ejemplo de mensaje con tipos MIME, con codificaci=F3n Quoted
  </body>
</html>
--_----------=_MCPart_293616404--
```

En este ejemplo podemos observar que utilizamos la especificación de MIME porque tenemos la cabecera *MIME-Version*: 1.0, además que el contenido del mensaje son partes alternativas. Y cada una de esas partes define su tipo, texto plano y HTML. Para separar las partes se utiliza un *boundary*, que es una cadena de caracteres aleatorios y que no se repitan en el contenido del mensaje.

- ***Content-Transfer-Encoding*:** Define el tipo de codificación utilizada para el contenido de esa parte del mensaje. Se necesita indicar el tipo de codificación para poder codificar los caracteres que no son ASCII, los tipos de codificación básicos son:
  - ***7-bit*:** Es la codificación base de MIME, con la que se codifica todo en ASCII 7 bits.
  - ***Quoted printable*:** Esta codificación se aplica cuando hay caracteres que no pertenecen al juego de caracteres US-ASCII. Con esta codificación convertimos los caracteres no incluidos en US-ASCII haciendo que sean representables en US-ASCII. Se diseñó de tal forma que fuera fácilmente legible por los humanos.
  - **Base 64:** Se utiliza para codificar datos binarios convirtiéndolos en texto, más concretamente son el conjunto de los símbolos alfanuméricos más los símbolos '+' y '/'. Aparte se utiliza el símbolo '='

como carácter de relleno. Para codificar en base 64 se agrupan 3 bytes de los datos de origen y se subdivide en 4 grupos de 6 bits cada uno, estos 4 grupos son representados cada uno con uno de los símbolos anteriores. Si no se puede agrupar 3 bytes, por falta de bytes, se rellena con ceros y los bytes faltantes se representan con el símbolo de relleno.

Para decodificar se realiza el procedimiento inverso, es decir, se omiten los caracteres de relleno, y por cada 4 caracteres de base 64 se obtiene 3 bytes, que son los datos resultantes. Con esta codificación conseguimos enviar cualquier tipo de dato en un mensaje de tipo MIME.

1.3.6. Estándares PKCS

En criptografía PCKS [14] (*Public-Key Cryptography Standards*) se refiere a un grupo de estándares de criptografía de clave pública, los cuales han sido desarrollados por los laboratorios RSA Security. Dicha colección de estándares están numerados del 1 al 15. Y concretamente para nuestro trabajo nos es necesario describir tres de ellos para familiarizar al lector acerca de ellos.

- **PKCS#5 v2:** Define una función de derivación de clave para obtener un hash de la clave, pero añadiéndole mayor seguridad, esto se consigue añadiendo unos datos extras (conocido como *salt)* para evitar que se puedan utilizar tablas pre-computadas (*Rainbow tables*) de resultados de funciones resumen. Para llegar al resultado final de la derivación hay que realizar la función resumen muchas veces, haciendo que el coste computacional sea alto, para evitar que averigüen la contraseña haciendo uso de fuerza bruta.

- **PKCS#7:** Este estándar hace referencia a la sintaxis de los mensajes firmados y/o cifrados en una arquitectura PKI (esta arquitectura será descrita en la sección 2.1.1.1.). Este estándar fue la base para crear S/MIME.

- **PKCS#12:** Es un estándar que define el formato de un fichero, el cual es usado comúnmente para el almacenamiento de claves privadas junto a su certificado de clave pública, protegidos mediante una clave.

1.4. Estructura del documento

En el índice de la memoria mostramos la estructura de la misma. A continuación exponemos con mayor detalle las secciones de las que está compuesta la memoria.

Capítulo 1: Introducción

En este capítulo se describe globalmente lo que se pretende realizar. Inicialmente exponemos el motivo por el cual se quiere desarrollar la aplicación y después los objetivos que se pretenden alcanzar. Así como aspectos básicos para correcta comprensión de esta documentación.

Capítulo 2: Estado del arte

El contenido de este capítulo describe el estado de la tecnología en el momento del desarrollo del trabajo, y qué otras opciones existen en el mercado.

Capítulo 3: Gestión del proyecto

En este apartado se describen los aspectos relacionados con la gestión del proyecto, entre los que se encuentran los diagramas de la planificación e hitos seguidos para el desarrollo del proyecto.

Capítulo 4: Análisis del sistema

En este capítulo se realiza el estudio en profundidad del problema a resolver y se define la solución propuesta. Para ellos se exponen los requisitos de usuario y después los requisitos de software del trabajo.

Capítulo 5: Diseño del sistema

A partir de los requisitos de software obtenidos en la fase anterior se propone la arquitectura del sistema, y se especifican los componentes. También se detalla el modelo relacional de los datos.

Capítulo 6: Implementación y pruebas

En este capítulo se indican qué consideraciones se han tenido en cuenta a la hora de implementar el sistema, así como el entorno de desarrollo y las convenciones de codificación utilizadas. También se incluye el plan de pruebas realizado para comprobar la funcionalidad del sistema.

Capítulo 7: Presupuesto

Se indica el cálculo del coste que ha supuesto la realización del sistema, desglosado en sus diferentes fases y tipos de coste.

Capítulo 8: Conclusiones

Para finalizar la memoria del trabajo se comentan las conclusiones que se han llegado y los posibles trabajos futuros.

Capítulo 9: Bibliografía

Se muestra el listado de referencias bibliográficas ordenadas por orden de aparición en el documento.

Anexo 1: Acrónimos y abreviaturas

Este anexo incluye la lista de los acrónimos y abreviaturas utilizados en todo el documento ordenados alfabéticamente.

Anexo 2: Manual de usuario

Se indican los pasos básicos para configurar la aplicación y poder enviar y recibir mensajes confidenciales y autenticados con el prototipo desarrollado.

2. Estado de la cuestión

En este apartado vamos a comentar los estándares relacionados con el correo electrónico seguro. También trataremos las aplicaciones existentes en el mercado y por qué no nos sirven como solución a nuestro problema.

2.1. Correo electrónico seguro

Con respecto al correo seguro existen dos grandes opciones, el correo seguro mediante PGP o mediante S/MIME, a continuación comentaremos cada uno de los dos métodos.

2.1.1. S/MIME

S/MIME [15] (*Secure / Multipurpose Internet Mail Extensions*) es un estándar que permite el envío de correos firmado y/o cifrados mediante el uso de criptografía de clave pública encapsulado en MIME. Este estándar se basa en PKCS#7 el cual detalla la infraestructura de clave pública (PKI).

2.1.1.1. PKI

La arquitectura PKI define una estructura de certificado X.509 y una lista de certificados revocados (CRL, para conocer que certificados ya no son válidos). También define una estructura jerárquica de autoridades de certificación que son las encargadas de autenticar, mediante firma, los certificados de los usuarios. Es decir, un usuario se genera un par de clave (privada y pública), y con su clave pública crea una petición de certificado, la cual reenvía a una autoridad certificadora para que se lo firme. La autoridad certificadora después de haber verificado que los datos del certificado son válidos firma el certificado y lo pone a disposición del usuario.

Para verificar un certificado se comprueba que la firma del certificado es válida y que confiemos en la autoridad certificadora que ha firmado el certificado. Para ello se tiene una lista de autoridades certificadoras en las que confiamos, si el certificado está

firmado por una autoridad certificadora en la que no confiamos el certificado no es válido. A continuación se muestra la estructura general de la infraestructura de clave pública jerárquica.

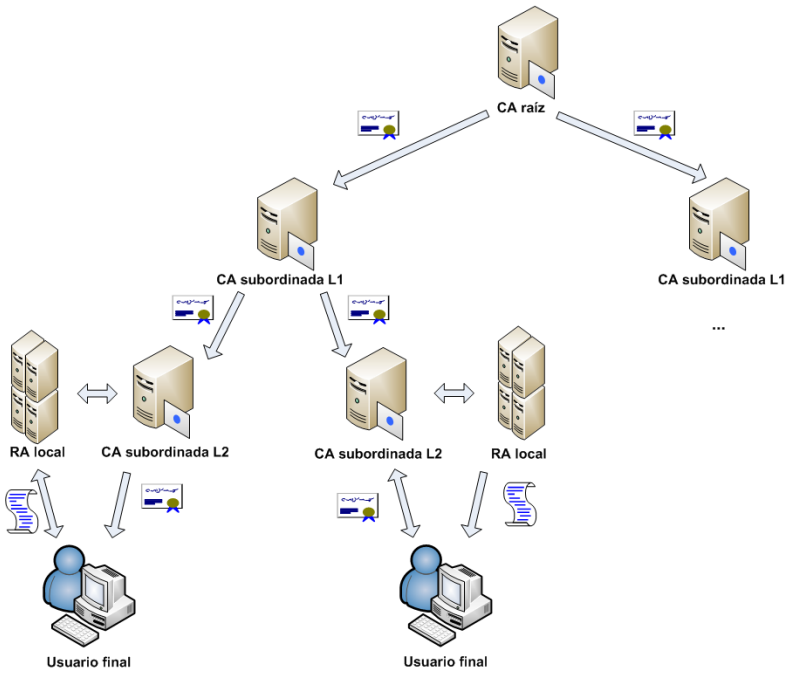

Ilustración 1 - Infraestructura de clave pública jerárquica

En la imagen anterior se puede apreciar el modelo PKI jerárquico, donde está la autoridad de certificación raíz (*Certificate Authority*), y sus subordinadas, hasta llegar al usuario final. También se encuentran otras entidades, llamadas autoridad de registro (*Registration Authority*), que son las encargadas de comprobar la veracidad de los datos de un certificado cuando se solicita la firma del mismo.

2.1.1.2. Estructura de los mensajes S/MIME

Para implementar el firmado y cifrado al estándar MIME se han añadido varios tipos nuevos:

Tipo MIME	*Smime-type*	Significado
multipart/signed		Identifica un mensaje firmado en dos partes, el mensaje y la firma
application/pkcs7-mime	*SignedData*	Entidad S/MIME firmada
	EnvelopedData	Entidad S/MIME cifrada
application/pkcs7-signature		Subparte correspondiente a la firma de un mensaje multiparte firmado.

Tabla 3 - Tipos S/MIME

- **Correos firmados (*SignedData*):** Los mensajes firmados se identifican por el tipo MIME *application/pkcs7-mime* y por el parámetro *smime-type* con el valor *singed-data.* Para crear un correo firmado hay que seguir los siguientes pasos:
  - Calcular el resumen del contenido del correo.
  - Ciframos el resumen calculado con la clave privada del firmante.
  - Creamos un bloque *SignerInfo*, que estará compuesto por el certificado de clave pública del firmante, un identificador del algoritmo utilizado para cifrar el resumen, y la firma del mensaje.
  - Se codifica el bloque anterior en base64 y se envía.
  - El destinatario decodifica el bloque *SignerInfo* y procede a verificar la firma. Para ello calcula el resumen del mensaje, después descifra la firma con la clave pública del origen y compara los dos resúmenes. Si los resúmenes coinciden la firma es válida.

Si se utiliza este tipo de mensaje el destinatario solo podrá obtener el contenido del mensaje si acepta S/MIME. La siguiente forma de firmado permite que los gestores de correos que no acepten S/MIME puedan leer el contenido del mensaje.

- **Correos firmados en claro (*Clear-signed data*):** Para conseguir lo comentado anteriormente, lo que se hace es firmar el contenido del mensaje y añadir la firma con el certificado de clave pública como un tipo MIME *application/pkcs7-signature.* Y el contenido del mensaje se añade dentro del tipo *multipart/signed.* Por lo que sigue conservando la estructura de tipo MIME. Se muestra a continuación un ejemplo:

Content-Type: multipart/signed; protocol="application/pkcs7-signature"; micalg=sha-1;
boundary="----=_Part_3_1097951768.1337740157375"

------=_Part_3_1097951768.1337740157375
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit

Ejemplo de mensaje firmado
------=_Part_3_1097951768.1337740157375
Content-Type: application/pkcs7-signature; name=smime.p7s; smime-type=signed-data
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="smime.p7s"
Content-Description: S/MIME Cryptographic Signature

MIAGCSqGSIb3DQEHAqCAMIACAQExCzAJBgUrDgMCGgUAMIAGCSqGSIb3DQEHAQAAoIAwggOl
7KGEw/Iy0dxBuIK3UgGkJEjD9FVGDDygBASvMUDw+kr0mQSdAAAAAAAA
------=_Part_3_1097951768.1337740157375--

Se ha quitado contenido de la parte *application/pkcs7-signature* para que no ocupe demasiado.

- **Correos cifrados (*Enveloped data*):** Los mensajes cifrados aparecen con el tipo MIME *application/pkcs7-mime* y con el parámetro *smime-type* con el valor *enveloped-data.* Para crear un correo cifrado se realizan los siguientes pasos:
    - Se genera una clave pseudoaleatoria para un algoritmo de cifrado simétrico.
    - Se cifra el contenido del correo con la clave de sesión.
    - Ciframos la clave de sesión con la clave pública del destinatario, para cada uno de los destinatarios.
    - Por cada destinatario se crea un bloque *RecipientInfo* que contiene un identificador del certificado de clave pública del destinatario, un identificador del algoritmo con el que se cifró la clave se sesión y la clave sesión cifrada.
    - Con los diferentes *RecipientInfo* y el mensaje cifrado se crea el *EnvelopedData*, se codifica en base64 y se envía.
    - Para descifrarlo el destinatario decodifica el *EnvelopedData*, busca el *RecipientInfo* asociado a su clave pública, descifra la clave se sesión con su clave privada y descifra el mensaje con la clave sesión.

- **Correos firmados y cifrados (*Signed and enveloped data*):** En S/MIME se permite la posibilidad de anidar firmado y cifrado. Por lo que es posible cifrar un mensaje firmado o firmar un mensaje cifrado. Aunque es más común utilizar primero el firmado y después el cifrado, puesto que cuando se firma se entiende que se está firmando.

Como se puede observar con S/MIME podemos asegurar su integridad y su confidencialidad, pero solo del contenido del mensaje, es decir, las cabeceras como asunto, destino, origen... no son cifradas. Esto es debido a que se necesita dar compatibilidad con los gestores de correo que no soportan S/MIME, y también porque no se puede firmar ni cifrar todas las cabeceras puesto que entre origen y destinatario algunas de las cabeceras son modificadas, como la fecha de recepción de cada uno de los servidores por los que pasa el correo hasta su destino.

2.1.2. PGP

El estándar PGP [16] (*Pretty Good Privacy*) también se ayuda de MIME para crear partes firmadas y cifradas, pero la gran diferencia con respecto a S/MIME es que se basa en una red de confianza, en vez de una infraestructura de clave pública.

2.1.2.1. Red de confianza

La red de confianza de PGP es una red descentralizada donde cada usuario tiene su lista de certificados públicos en los que confía. Es decir, no existen entidades intermedias para verificar un certificado, sino que el usuario decide si confía en el certificado, y en qué grado confía. En PGP existen cuatro niveles de confianza: desconocido, no fiable, ligeramente fiable y absolutamente fiable. Dependiendo del nivel que se le asigne indicara el grado de fiabilidad que se le asigna a la clave pública para firmar otros certificados.

En conclusión PGP utiliza la red de confianza para asignarle un valor de confianza a los certificados de clave pública.

2.2. Gestores de correo seguro en Android

En el momento del análisis del estado de la cuestión se encontraron diferentes gestores de correo para Android. Las principales aplicaciones tenidas en cuenta son: Djigzo, Moxier Mail, TouchDown, K-9 Mail junto con APG y por último X509Tools. Las funcionalidades de cada una de estas aplicaciones son comentadas a continuación:

- Djigzo [17]: En esta aplicación gratuita se puede enviar correos electrónicos firmados y/o cifrados, pero no existe la funcionalidad de recibir correos. Por lo que no se le puede tratar como un gestor de correos. Permite autenticar un correo recibido con otro gestor de correos.

- Moxier Mail [18]: Es una aplicación de pago que permite el envío y recepción de correos firmados y/o cifrados, pero con el inconveniente de que solo acepta cuentas de tipo Microsoft Exchange ActiveSync.

- TouchDown [19]: Realiza las mismas funcionalidades que la aplicación anterior, teniendo las mismas deficiencias con respecto al tipo de cuentas aceptadas.

- K-9 Mail [20] y APG [21]: En este caso se necesitan dos aplicaciones, una es K-9 Mail que es un gestor de correos de uso libre y código abierto, y APG que es un gestor de certificados de tipo PGP. Al instalar las dos aplicaciones podemos enviar y recibir correos electrónicos seguros, con el inconveniente de que como usamos certificados tipo PGP tenemos que ser nosotros quienes gestionamos si aceptamos la autenticidad de los certificados.

- X509Tools [22]: Otra aplicación con la cual se pueden enviar correos firmados y/o cifrados, en este caso es con la arquitectura X509 para los certificados. Tampoco puede recibir correos pero si puede descifrar los correos abriendo el archivo smime.p7m con esta aplicación. Durante el desarrollo del proyecto se

publicó una nueva aplicación basada en X509Tools, la cual añade la funcionalidad de poder recibir correos seguros (R2Mail2 [23]).

Una vez habiendo analizado las herramientas existentes se puede concluir que no existe ninguna aplicación que cumpla con los objetivos marcados para este Trabajo Fin de Grado. Por lo que se decide analizar, diseñar e implementar un gestor de correo electrónico seguro para el sistema operativo Android.

3. Gestión del proyecto

3.1. Gestión del proyecto software

En este capítulo se resumen las decisiones tomadas con respecto a la planificación del proyecto y los hitos necesarios para desarrollar el sistema.

Primero se ha determinado el orden en el que se va dividir el desarrollo del proyecto. Para este caso se ha elegido un diseño en cascada en el cual se va desarrollando por fases. Los cambios producidos en fases anteriores repercuten en las fases posteriores. En la ilustración 2 se muestra el diagrama del ciclo de vida del proyecto.

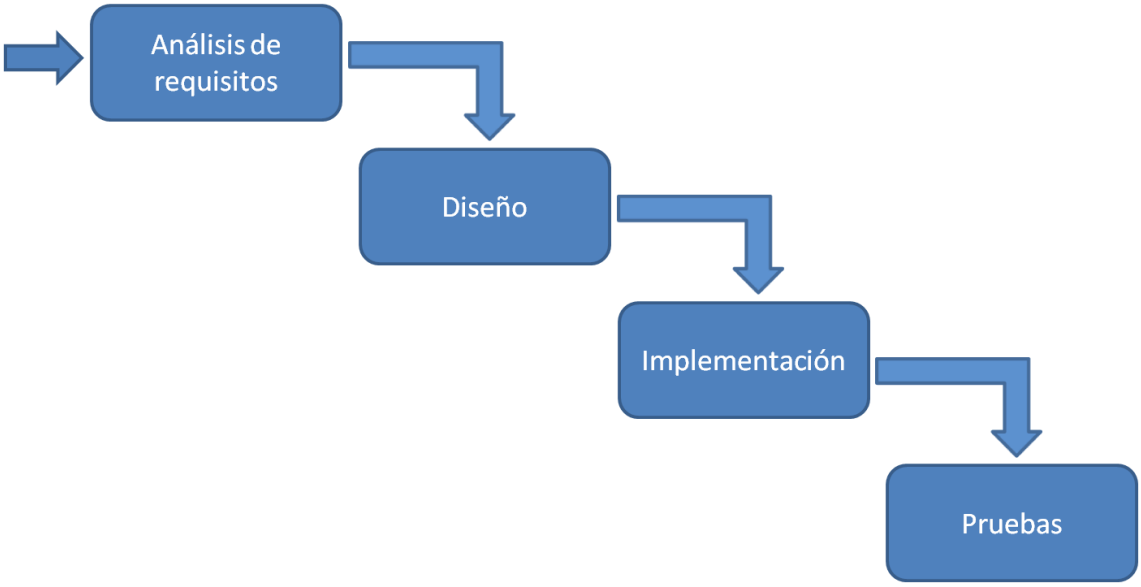

Ilustración 2 - Ciclo de vida

Durante la fase implementación se irá desarrollando la aplicación y a la vez realizando pruebas de lo desarrollado. Y al finalizar dicha fase se realizan pruebas de toda la aplicación desarrollada.

3.2. Organización del trabajo

La organización del trabajo se ha realizado subdividiendo en tareas el trabajo a realizar. Se ha utilizado un diagrama de Gantt para diseñar el reparto de las tareas en el tiempo. A continuación mostramos la jerarquía seguida en el proyecto.

- Inicio
  - Definición de objetivos
- Estado de la cuestión
  - Estudio de las soluciones existentes
  - Estudio de las tecnologías a utilizar
- Gestión del proyecto
  - Metodología a emplear
  - Organización del trabajo
- Análisis
  - Descripción del modelo
  - Requisitos de usuario
  - Requisitos de software
- Diseño
  - Descripción de la arquitectura
  - Descripción de los componentes
- Implementación
  - Generación del código
- Pruebas
- Documentación
  - Realización de la memoria
  - Generación de manuales

Como podemos observar se han definidos 8 fases para la realización del proyecto, incluyendo en ellas las fases de la gestión del proyecto software. A continuación se mostraran los diagramas de Gantt con la programación seguida, donde se indica la duración en días de cada una de las tareas. En dicha planificación la media diaria de horas empleadas es de cuatro.

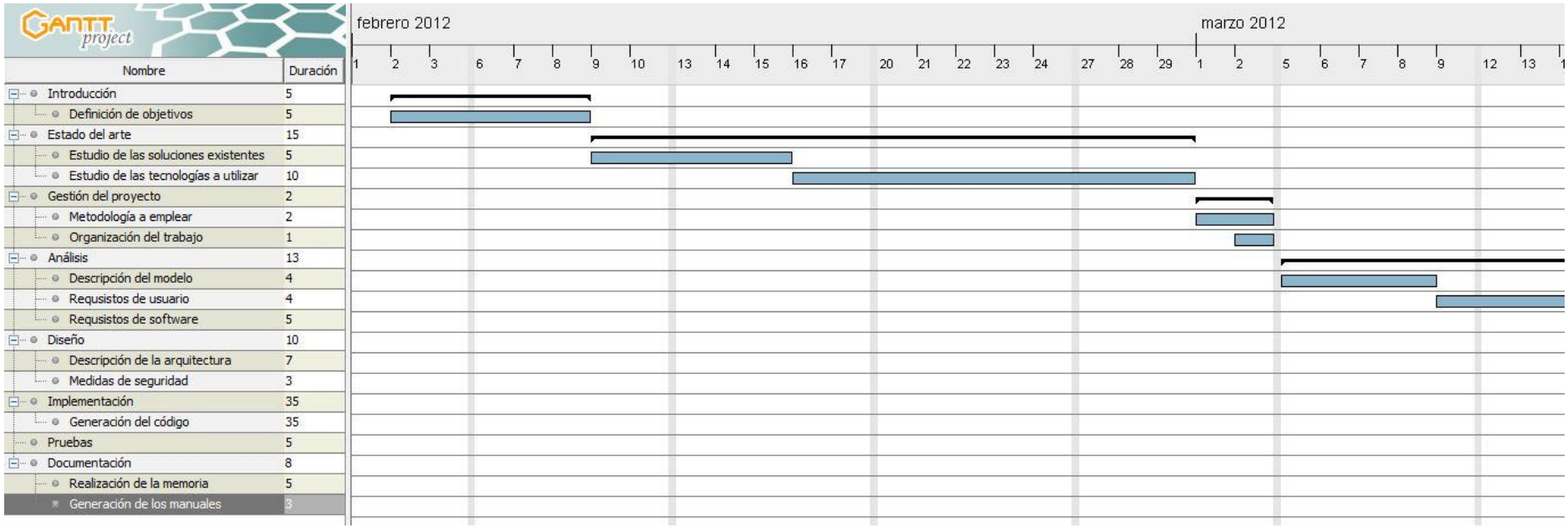

Ilustración 3 - Diagrama de Gantt 1

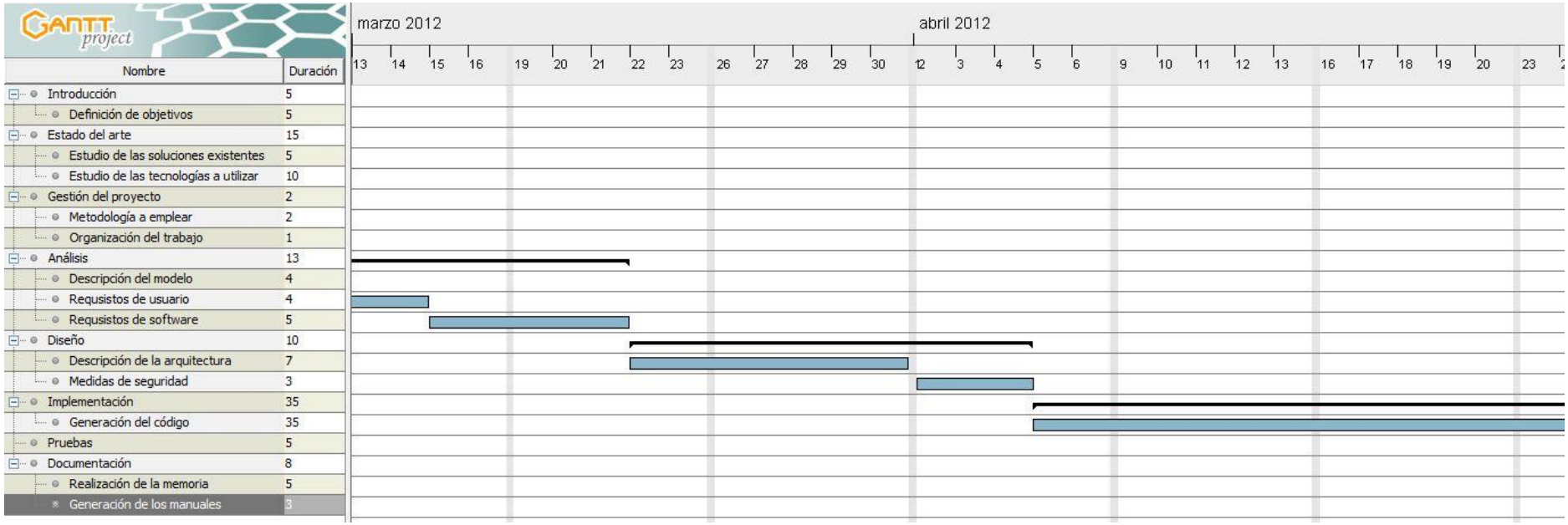

Ilustración 4 - Diagrama de Gantt 2

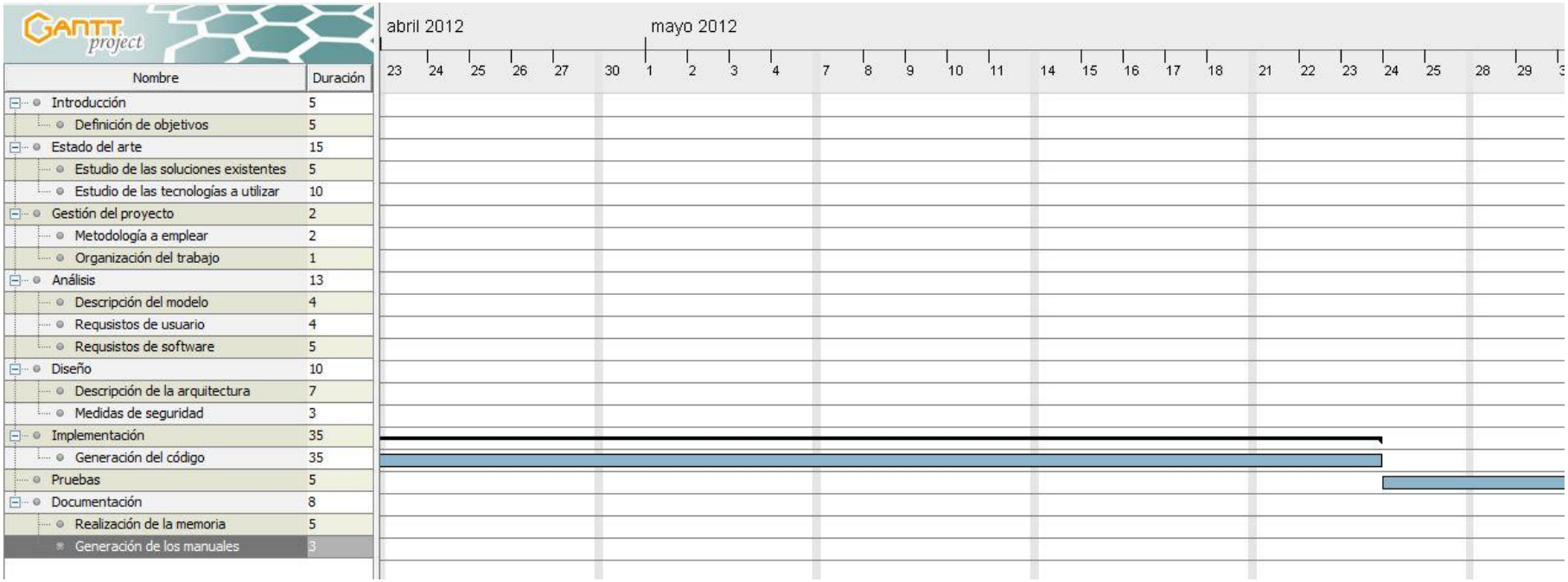

Ilustración 5 - Diagrama de Gantt 3

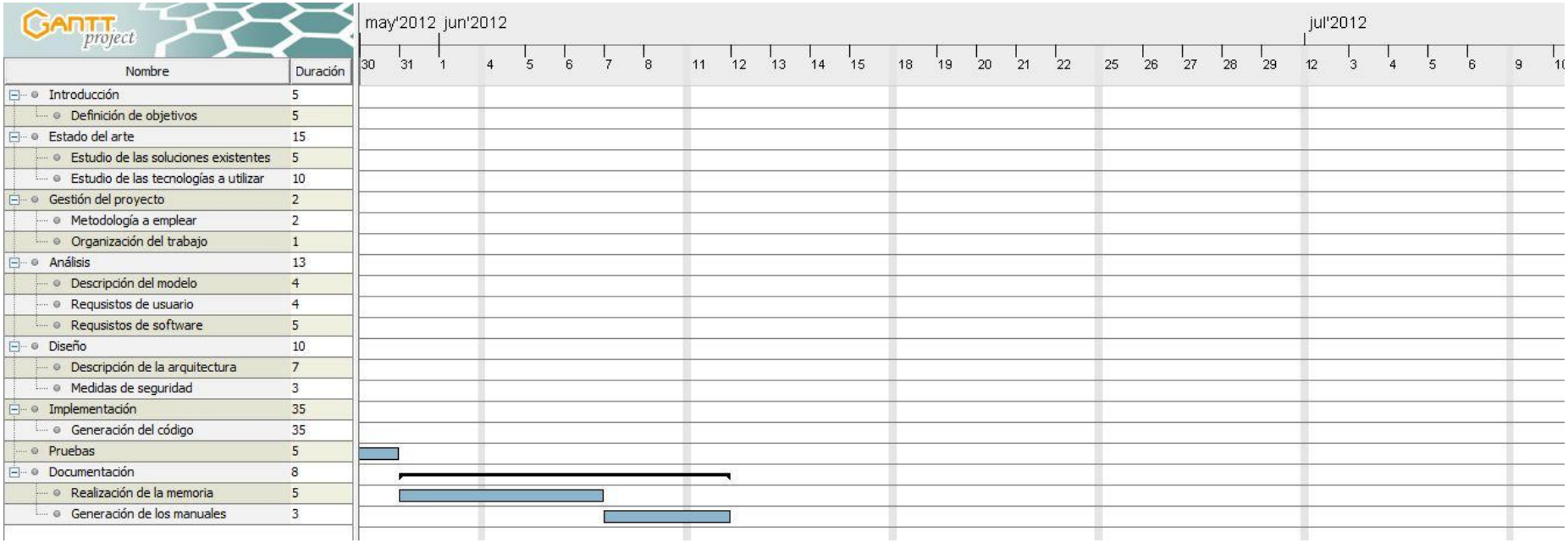

Ilustración 6 - Diagrama de Gantt 4

4. Análisis del sistema

4.1. Planteamiento del problema

El problema a resolver en este trabajo es el diseño e implementación de un cliente de correo electrónico seguro en el sistema operativo Android. Para ello se ha estudiado en profundidad el sistema, la tecnología actual y las alternativas existentes.

Aunque los clientes de correo en sistemas operativos de escritorio cuentan con las funcionalidades necesarias para poder enviar los correos de forma segura, ya sea firmados y/o cifrados, en el sistema operativo Android no es tan fácil de implementar debido en parte, a que hasta la última versión del sistema operativo (Android 4.0) no existían las funcionalidades necesarias para poder gestionar los certificados privados de forma adecuada. La falta de una API propia del sistema operativo Android para el envío y recepción de correos es otro de los inconvenientes existentes.

4.1.1. Propósito y funcionalidad

El sistema a implementar consiste en desarrollar desde cero un gestor de correo básico, que incorpora las funcionalidades de seguridad comentadas (firma y cifrado). Con este fin se va utilizar el estándar S/MIME. Las funcionalidades de gestor de correos son:

- Escribir correos.
- Enviar correos.
- Recibir correos.
- Leer correos enviados y recibidos.
- Gestionar los contactos.

Y por otro lado, en lo referente al correo seguro:

- Gestionar certificados de clave pública de los contactos.
- Gestionar la clave privada del usuario.
- Enviar correos firmados y/o cifrados.

- Validar la firma de los correos recibidos.
- Descifrar los correos recibidos.

Se ha decidido desarrollar íntegramente el gestor de correos porque las soluciones de código abierto existentes eran demasiado complejas y sin la API necesaria para poder modificarla.

4.1.2. Consideraciones de entorno

Para utilizar el sistema los usuarios requerirán tener como mínimo instalado en su dispositivo móvil la versión 4.0 de Android, también conocida como *Ice Cream Sandwich*. Se ha requerido esta versión para poder hacer uso de las nuevas implementaciones introducidas con respecto al almacenamiento seguro de claves privadas.

Otro de los requisitos es que el dispositivo disponga de conexión a internet.

También se ha tenido en cuenta el marco legal, en concreto la ley de protección de datos española [24], pero en nuestro caso no se aplica puesto que los datos almacenados están siempre bajo supervisión del usuario.

"Artículo 2. Ámbito de aplicación.

El régimen de protección de los datos de carácter personal que se establece en la presente Ley Orgánica NO será de aplicación: A los ficheros mantenidos por personas físicas en el ejercicio de actividades exclusivamente personales o domésticas."

4.1.3. Relación con otros sistemas

El sistema se relacionará con el servidor de envío seleccionado por el usuario (servidor SMTP) y el servidor de recepción de mensajes (servidor IMAP).

4.2. Especificación de requisitos

En el trascurso del análisis y diseño del sistema se seguirán las normas definidas por el estándar PSS-05-0 de la ESA [25]. Para realizar el análisis del sistema se va a proceder a

elaborar un listado de requisitos. Estos requisitos primero recogerán las características deseadas de forma general (requisitos de usuario) y después se especificarán dando un mayor grado de detalle (requisitos de software). Para terminar se realizara una matriz de trazabilidad donde se relacionan los requisitos de usuario con los requisitos de software, y así saber de dónde proceden los requisitos de software.

4.2.1. Requisitos de usuario

Para poder nombrar y referirnos a los requisitos de usuario usaremos la siguiente notación:

UR-[C | R]nnn

Donde:

UR: Significa requisito de usuario.

C: Indica que es requisito de capacidad.

R: Indica que es requisito de restricción.

nnn: Es el número identificativo del requisito dentro de su tipo de requisito.

En lo referente a los atributos de cada requisito, indicamos el significado de cada uno de los atributos.

- Fuente: Indica el origen del requisito.
- Necesidad: Puede tener tres valores: esencial (que el requisito es imprescindible), deseable (que el requisito conviene incluirlo), opcional (que no es obligatorio incluirlo).
- Prioridad: Orden de implementación del requisito.
- Estabilidad: Informa de la probabilidad de cambio del requisito durante el desarrollo del sistema. Si es alta significa que es muy difícil que vaya a cambiar, y si es baja muy probable que cambie el requisito.

4.2.1.1. Capacidades generales

El propósito del sistema es que el usuario pueda enviar mensajes firmados y/o cifrados. Poder recibirlos y validar su firma, si la tuviera, y descifrarlos cuando estén cifrados. Para ello el usuario tiene que poder gestionar su clave privada y los certificados de clave pública de sus contactos. Como el sistema lo hemos desarrollado íntegramente, también tiene que permitir el envío y recepción de correos sin firmar ni cifrar. Y por último se permitirá enviar y recibir documentos adjuntos.

4.2.1.2. Restricciones generales

Las restricciones existentes en el desarrollo del sistema solo obedecen a la necesidad de utilizar como mínimo la versión 4 del sistema operativo Android.

4.2.1.3. Requisitos de capacidad

UR-C001	
Descripción	Tiene que permitir el envío de correos electrónicos sin firmar ni cifrar o firmados y/o cifrados.
Fuente	Cliente
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 4 - UR-C001

UR-C002	
Descripción	Tiene que permitir gestionarse contactos.
Fuente	Cliente
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 5 - UR-C002

UR-C003	
Descripción	Se tiene que garantizar la privacidad de los datos privados del usuario.
Fuente	Cliente
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 6 - UR-C003

UR-C004	
Descripción	Se tiene que permitir gestionar los certificados públicos de los contactos.
Fuente	Cliente
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 7 - UR-C004

UR-C005	
Descripción	Tiene que permitir la gestión de la clave privada del usuario.
Fuente	Cliente
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 8 - UR-C005

UR-C006	
Descripción	Se tiene que permitir enviar adjuntos.
Fuente	Cliente
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 9 - UR-C006

UR-C007	
Descripción	Se tiene que permitir recibir adjuntos.
Fuente	Cliente
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 10 - UR-C007

UR-C008	
Descripción	Se tiene que permitir recibir correos electrónicos.
Fuente	Cliente
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 11 - UR-C008

UR-C009	
Descripción	Se deberá descifrar los mensajes que se reciban cifrados.
Fuente	Cliente
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 12 - UR-C009

UR-C010	
Descripción	Se permitirá validar la firma de los correos recibidos.
Fuente	Cliente
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 13 - UR-C010

UR-C011	
Descripción	Ha de poderse borrar la cuenta del usuario de forma segura.
Fuente	Cliente
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 14 - UR-C011

4.2.1.4. Requisitos de restricción

UR-R001	
Descripción	La aplicación tiene que ejecutarse como mínimo en Android 4.0
Fuente	Cliente
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 15 - UR-R001

UR-R002	
Descripción	Se tiene que utilizar el estándar S/MIME para el cifrado y firmado de los mensajes.
Fuente	Cliente
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 16 - UR-R002

UR-R003	
Descripción	La clave de usuario tiene que tener mínimo 8 caracteres y estar compuesta por números y letras.
Fuente	Cliente
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 17 - UR-C003

4.2.2. Requisitos de software

Antes de comenzar a integrar los requisitos vamos a especificar cuál es el método que hemos seguido para establecer su identificador:

SR-[F | I | O | R | S]nnn

Donde:

SR: Significa requisito de software.

F: Requisito de funcionalidad.

I: Requisito de interfaz.

O: Requisito de operación.

R: Requisito de recurso

S: Requisito de seguridad

nnn: Número identificativo del requisito dentro de su tipo de requisito.

En lo referente a los atributos de cada requisito, indicamos el significado de cada uno de los atributos.

- Fuente: Indica el requisito de origen que se está especificando.
- Necesidad: Puede tener tres valores: esencial (que el requisito es imprescindible), deseable (que el requisito conviene incluirlo), opcional (que no es obligatorio incluirlo).
- Prioridad: Orden de implementación del requisito.
- Estabilidad: Informa de la probabilidad de cambio del requisito durante el desarrollo del sistema. Si es alta significa que es muy difícil que vaya a cambiar, y si es baja muy probable que cambie el requisito.

4.2.2.1. Requisitos funcionales

SR-F001	
Descripción	Los mensajes cifrados se cifran con la clave pública del usuario destinatario, si no existe certificado público se avisa al usuario y no se envía.
Fuente	UR-C001
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 18 - SR-F001

SR-F002	
Descripción	Los mensajes firmados se firman con la clave privada del usuario, si no existe se envía sin firmar.
Fuente	UR-C001
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 19 - SR-F002

SR-F003	
Descripción	El usuario tiene que poder crear contactos.
Fuente	UR-C002
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 20 - SR-F003

SR-F004	
Descripción	El usuario tiene que poder modificar contactos.
Fuente	UR-C002
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 21 - SR-F004

SR-F005	
Descripción	El usuario tiene que poder borrar contactos.
Fuente	UR-C002
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 22 - SR-F005

SR-F006	
Descripción	El usuario podrá guardar un certificado de clave pública para cada contacto.
Fuente	UR-C004
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 23 - SR-F006

SR-F007	
Descripción	El usuario podrá borrar el certificado de clave pública de los contactos.
Fuente	UR-C004
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 24 - SR-F007

SR-F008	
Descripción	El usuario podrá modificar el certificado de clave pública de los contactos.
Fuente	UR-C004
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 25 - SR-F008

SR-F009	
Descripción	El usuario podrá guardar un certificado de clave privada.
Fuente	UR-C005
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 26 - SR-F009

SR-F010	
Descripción	El usuario podrá borrar el certificado de clave privada.
Fuente	UR-C005
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 27 - SR-F010

SR-F011	
Descripción	El usuario podrá modificar el certificado de clave privada.
Fuente	UR-C005
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 28 - SR-F011

SR-F012	
Descripción	El usuario podrá seleccionar ficheros adjuntos de su tarjeta de memoria externa.
Fuente	UR-C006
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 29 - SR-F012

SR-F013	
Descripción	El usuario podrá descargar los documentos adjuntos en la tarjeta externa.
Fuente	UR-C007
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 30 - SR-F013

SR-F014	
Descripción	El usuario podrá recibir correos electrónicos y almacenarlos en el dispositivo móvil.
Fuente	UR-C008
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 31 - SR-F014

SR-F015	
Descripción	Se podrá borrar la cuenta del usuario de forma segura.
Fuente	UR-C011
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 32 - SR-F015

4.2.2.2. Requisitos de interfaz

SR-I001	
Descripción	Para seleccionar un fichero adjunto se mostrara el gestor de archivos por defecto.
Fuente	UR-C006
Necesidad	Deseable
Prioridad	Alta
Estabilidad	Alta

Tabla 33 - SR-I001

SR-I002	
Descripción	Se visualizaran los correos recibidos en forma de lista en la carpeta de recibidos.
Fuente	UR-C008
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 34 - SR-I002

SR-I003	
Descripción	Se visualizaran los correos enviados en forma de lista en la carpeta de enviados.
Fuente	UR-C001
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 35 - SR-I003

SR-I004	
Descripción	Se podrá visualizar el contenido completo del correo electrónico.
Fuente	UR-C008 y UR-C001
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 36 - SR-I004

4.2.2.3. Requisitos operacionales

SR-O001	
Descripción	Se descifrara los mensajes recibidos que estén cifrados con la clave privada del usuario.
Fuente	UR-C009 y UR-R002
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 37 - SR-O001

SR-O002	
Descripción	Se validara los mensajes recibidos que estén firmados con la clave pública del contacto origen.
Fuente	UR-C010 y UR-R002
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 38 - SR-O002

SR-O003	
Descripción	Para borrar la cuenta del usuario de forma segura se cifrará con AES 128 en modo CTR todo el contenido de la base de datos con una clave generada aleatoriamente y después se liberará el contenido de la memoria.
Fuente	UR-C011
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 39 - SR-O003

4.2.2.4. Requisitos de recurso

SR-R001	
Descripción	Se necesita tener conexión a internet para poder enviar y recibir correos y descargar adjuntos.
Fuente	UR-C001, UR-C006, UR-C007 y UR-C008
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 40 - SR-R001

SR-R002	
Descripción	La aplicación tiene que ejecutarse como mínimo en Android 4.0
Fuente	UR-R001
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 41 - SR-R002

SR-R003	
Descripción	Se necesita tener un certificado de clave pública para poder firmar los mensajes.
Fuente	UR-C001
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 42 - SR-R003

SR-R004	
Descripción	Se necesita tener un certificado de clave pública para poder firmar descifrar los mensajes cifrados.
Fuente	UR-C009
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 43 - SR-R004

SR-R005	
Descripción	Se necesita tener el certificado de clave pública del contacto destino si queremos cifrar el mensaje.
Fuente	UR-C001
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 44 - SR-R005

4.2.2.5. Requisitos de seguridad

SR-S001	
Descripción	Los datos privados almacenados en la base de datos se cifraran con AES 128 en modo CTR.
Fuente	UR-C003
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 45 - SR-S001

SR-S002	
Descripción	Para cifrar los datos se solicitara al usuario una contraseña, la cual se derivara y se usara para cifrar los datos privados en la base de datos.
Fuente	UR-C003
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 46 - SR-S002

SR-S003	
Descripción	El certificado de clave privada del usuario se guardara en el almacén de claves proveído por Android.
Fuente	UR-C003
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 47 - SR-S003

SR-S004	
Descripción	La clave de usuario tiene que tener mínimo 8 caracteres y estar compuesta por números y letras.
Fuente	UR-R003
Necesidad	Esencial
Prioridad	Alta
Estabilidad	Alta

Tabla 48 - SR-S004

4.2.3. Matriz de trazabilidad

	UR-C001	UR-C002	UR-C003	UR-C004	UR-C005	UR-C006	UR-C007	UR-C008	UR-C009	UR-C010	UR-C011	UR-R001	UR-R002	UR-R003
SR-F001	X													
SR-F002	X													
SR-F003		X												
SR-F004		X												
SR-F005		X												
SR-F006				X										
SR-F007				X										
SR-F008				X										
SR-F009					X									
SR-F010					X									
SR-F011					X									
SR-F012						X								
SR-F013							X							
SR-F014								X						
SR-F015											X			
SR-I001						X								
SR-I002								X						
SR-I003	X													
SR-I004	X							X						
SR-O001									X				X	
SR-O002										X			X	
SR-O003											X			
SR-R001	X					X	X	X						

Tabla 49 - Matriz de trazabilidad 1

	UR-C001	UR-C002	UR-C003	UR-C004	UR-C005	UR-C006	UR-C007	UR-C008	UR-C009	UR-C010	UR-C011	UR-R001	UR-R002	UR-R003
SR-O002										X			X	
SR-O003											X			
SR-R001	X					X	X	X						
SR-R002												X		
SR-R003	X													
SR-R004									X					
SR-R005	X													
SR-S001			X											
SR-S002			X											
SR-S003			X											
SR-S004														X

Tabla 50 - Matriz de trazabilidad 2

Con la matriz de trazabilidad podemos comprobar que todos los requisitos de software provienen de uno o varios requisitos de usuario, y que todos los requisitos de usuario tienen su correspondencia con algún requisito de software.

4.3. Casos de uso

En la siguiente figura se detallan los casos de uso de la aplicación desarrollada. Como contamos con una gran funcionalidad, hemos decidido incorporar dos actores que representan al mismo usuario, los correspondientes a 'usuario autenticado', para mejorar la visualización. Y también existe otro actor llamado 'usuario no autenticado'.

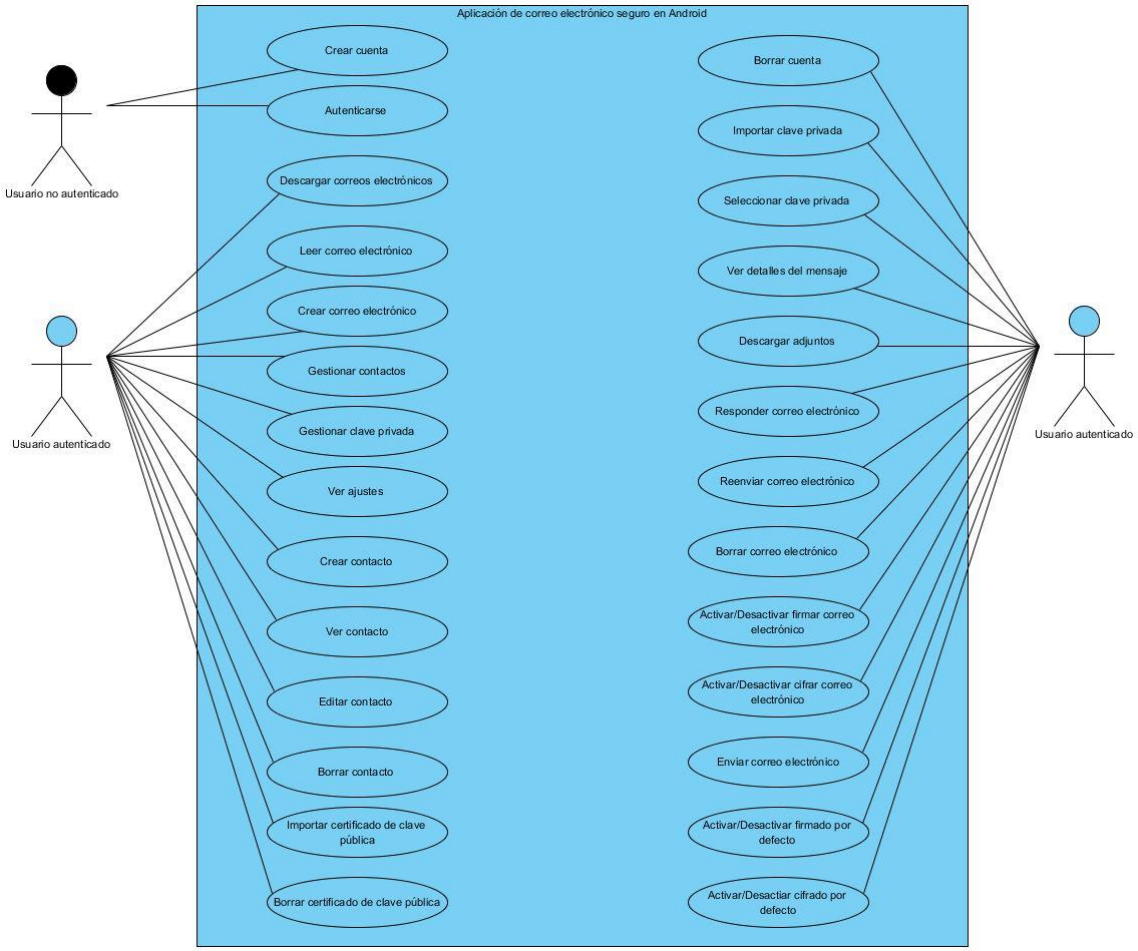

Ilustración 7 - Casos de uso

A continuación presentamos un esquema resumen con los casos de uso de nuestro sistema donde describimos la funcionalidad de cada uno de los casos de uso:

Caso de Uso: Crear cuenta.

Actores: Usuario no autenticado.

Objetivo: Crear una cuenta para utilizar la aplicación.

Precondiciones: Que no exista una cuenta de usuario.

Postcondiciones: Poder autenticarse.

Caso de Uso: Autenticarse.

Actores: Usuario no autenticado.

Objetivo: Autenticarse en la aplicación.

Precondiciones: Tener una creada una cuenta en la aplicación.

Postcondiciones: Poder utilizar la aplicación.

Caso de Uso: Descargar correos electrónicos.

Actores: Usuario autenticado.

Objetivo: Descargar los nuevos correos electrónicos recibidos.

Precondiciones: Estar autenticado en la aplicación.

Postcondiciones: Poder leer los correos electrónicos.

Caso de Uso: Leer correo electrónico.

Actores: Usuario autenticado.

Objetivo: Leer el contenido de un correo electrónico.

Precondiciones: Estar autenticado y haber descargado algún correo.

Postcondiciones: Poder leer los detalles del correo.

Caso de Uso: Crear correo electrónico.

Actores: Usuario autenticado.

Objetivo: Crear un correo electrónico para poder enviarlo.

Precondiciones: Estar autenticado en la aplicación.

Postcondiciones: Poder adjuntar un archivo, enviar el correo al destinatario.

Caso de Uso: Gestionar contactos.

Actores: Usuario autenticado.

Objetivo: Poder gestionar los contactos existentes en la aplicación.

Precondiciones: Estar autenticado en la aplicación.

Postcondiciones: Poder crear, editar o borrar un contacto.

Caso de Uso: Gestionar clave privada.

Actores: Usuario autenticado.

Objetivo: Poder gestionar la clave privada del usuario.

Precondiciones: Estar autenticado en la aplicación.

Postcondiciones: Poder firmar y descifrar correos electrónicos.

Caso de Uso: Ver Ajustes.

Actores: Usuario autenticado.

Objetivo: Acceder a los ajustes de la aplicación.

Precondiciones: Estar autenticado en la aplicación.

Postcondiciones: Poder borrar la cuenta, seleccionar el valor por defecto de cifrar y firmar.

Caso de Uso: Crear contacto.

Actores: Usuario autenticado.

Objetivo: Crear un nuevo contacto.

Precondiciones: Estar en la vista de gestionar contactos.

Postcondiciones: Importar certificado de clave pública al contacto.

Caso de Uso: Ver contacto.

Actores: Usuario autenticado.

Objetivo: Visualizar los datos del contacto.

Precondiciones: Estar en la vista de gestionar contactos.

Postcondiciones: Podrá editar o borrar el contacto.

Caso de Uso: Editar contacto.

Actores: Usuario autenticado.

Objetivo: Editar los valores del contacto.

Precondiciones: Estar visualizando un contacto.

Postcondiciones: Podrá visualizar los datos editados.

Caso de Uso: Borrar contacto.

Actores: Usuario autenticado.

Objetivo: Eliminar los datos del contacto.

Precondiciones: Estar visualizando un contacto.

Postcondiciones: Se borrara los datos del contacto.

Caso de Uso: Importar certificado de clave pública.

Actores: Usuario autenticado.

Objetivo: Obtener certificado de clave pública de la tarjeta externa y guardarlo en el contacto.

Precondiciones: Estar visualizando un contacto.

Postcondiciones: Cifrar correos electrónicos dirigidos a ese contacto.

Caso de Uso: Borrar certificado de clave pública.

Actores: Usuario autenticado.

Objetivo: Eliminar el certificado de clave pública del contacto.

Precondiciones: Estar visualizando un contacto.

Postcondiciones: No se podrá cifrar correos electrónicos dirigidos a ese contacto.

Caso de Uso: Borrar cuenta.

Actores: Usuario autenticado.

Objetivo: Eliminar todos los datos de la aplicación.

Precondiciones: Estar en la vista de ajustes.

Postcondiciones: -

Caso de Uso: Importar clave privada.

Actores: Usuario autenticado.

Objetivo: Importar la clave privada del usuario en el *KeyChain* de Android.

Precondiciones: Estar en la vista de gestionar clave privada.

Postcondiciones: Poder seleccionar la clave para usar en la aplicación de correo electrónico seguro.

Caso de Uso: Seleccionar clave privada.

Actores: Usuario autenticado.

Objetivo: Seleccionar la clave privada para utilizarla en la aplicación.

Precondiciones: Estar en la vista de ajustes.

Postcondiciones: Poder enviar correos electrónicos firmados, y descifrar correos recibidos.

Caso de Uso: Ver detalles del mensaje

Actores: Usuario autenticado.

Objetivo: Ver los detalles del mensaje, como documentos adjuntos, estado de la firma (en su caso)...

Precondiciones: Estar en la vista de leer correo electrónico.

Postcondiciones: Poder descargar documentos adjuntos.

Caso de Uso: Descargar adjuntos.

Actores: Usuario autenticado.

Objetivo: Descargar los ficheros adjuntos del correo electrónico.

Precondiciones: Estar en la vista detalles del mensaje.

Postcondiciones: Poder visualizar el contenido del fichero.

Caso de Uso: Responder correo electrónico.

Actores: Usuario autenticado.

Objetivo: Responder a un correo electrónico.

Precondiciones: Estar en la vista de leer correo electrónico.

Postcondiciones: Enviar correo electrónico.

Caso de Uso: Reenviar correo electrónico.

Actores: Usuario autenticado.

Objetivo: Reenviar un correo electrónico.

Precondiciones: Estar en la vista de leer correo electrónico.

Postcondiciones: Enviar correo electrónico a los destinatarios seleccionados.

Caso de Uso: Borrar correo electrónico.

Actores: Usuario autenticado.

Objetivo: Eliminar un correo electrónico.

Precondiciones: Estar en la vista de leer correo electrónico.

Postcondiciones: Borrado seguro del correo.

Caso de Uso: Enviar correo electrónico.

Actores: Usuario autenticado.

Objetivo: Eliminar un correo electrónico.

Precondiciones: Estar en la vista de crear correo electrónico.

Postcondiciones: -.

Caso de Uso: Activar/Desactivar firmar por defecto.

Actores: Usuario autenticado.

Objetivo: Activar/Desactivar la opción de firmar por defecto.

Precondiciones: Estar en la vista de ajustes.

Postcondiciones: Enviar correos firmados de forma predeterminada.

Caso de Uso: Activar/Desactivar cifrar por defecto.

Actores: Usuario autenticado.

Objetivo: Activar/Desactivar la opción de cifrar por defecto.

Precondiciones: Estar en la vista de ajustes.

Postcondiciones: Enviar correos cifrados de forma predeterminada.

5. Diseño del sistema

Después de haber realizado un análisis exhaustivo del sistema y haber definido las características que debe reunir el sistema, a continuación se expone el diseño del sistema.

5.1. Arquitectura

Para que el sistema desarrollado funcione se necesita un dispositivo con el sistema operativo Android, un servidor SMTP (para el envío de correos), un servidor IMAP (para la recepción de los correos) y conexión a internet. En la siguiente imagen se puede visualizar dicha estructura.

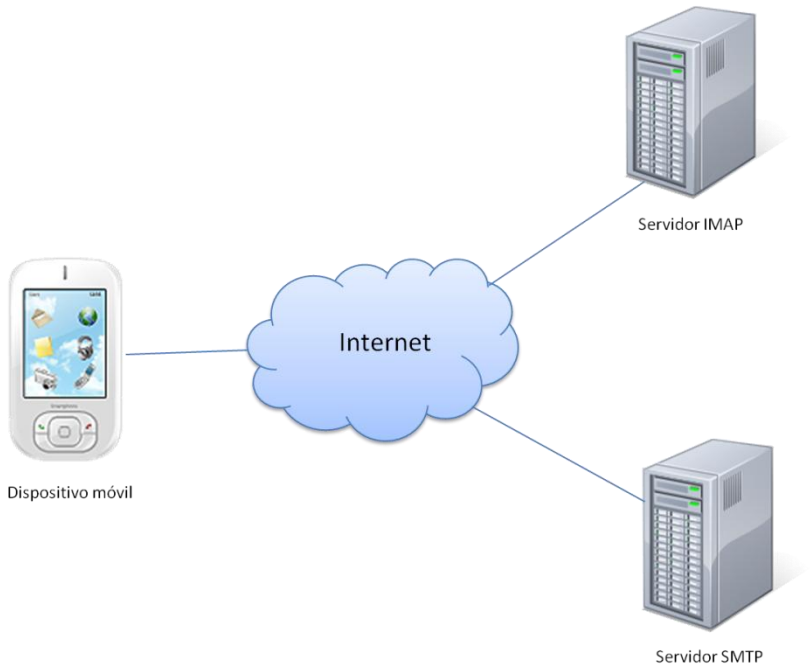

Ilustración 8 - Arquitectura del sistema

Las conexiones existentes entre el dispositivo móvil y los servidores pueden ser seguras o no, dependiendo de si el servidor acepta conexiones con SSL. Aunque la comunicación entre el dispositivo móvil y el servidor SMTP sea segura, la comunicación existente entre el servidor SMTP del usuario origen y el servidor SMTP del destinatario no es una conexión segura. Esta es una de las razones por las que es importante firmar y cifrar los correos electrónicos enviados, otra razón es que el proveedor de correo no

tiene por qué tener acceso al contenido de los mensajes enviados o recibidos por los usuarios.

Con respecto a la arquitectura utilizada para el desarrollo de la aplicación se ha utilizado la arquitectura MVC (Modelo Vista Controlador).

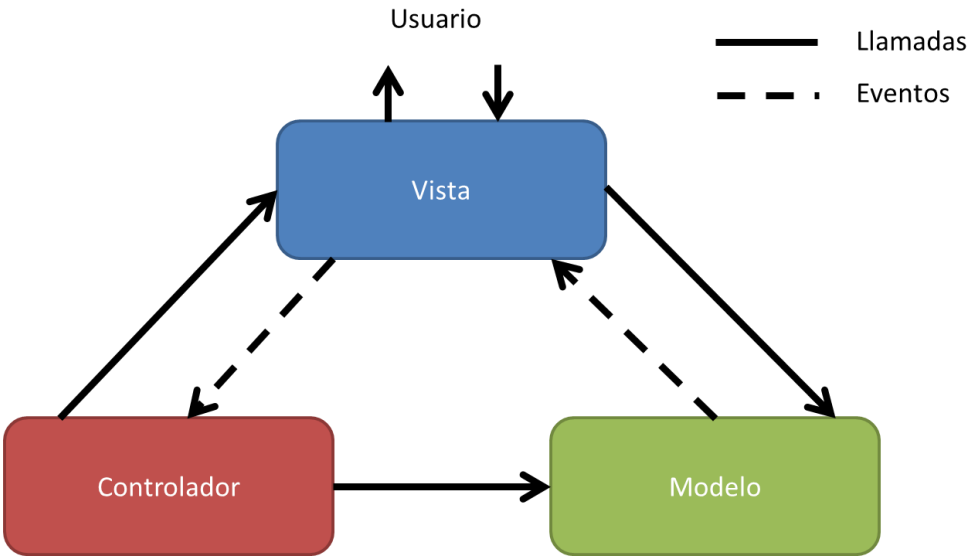

Ilustración 9 – MVC

La arquitectura MVC, como bien indica su nombre, se divide en:

- **Modelo:** es la representación de la información con la cual el sistema opera.
- **Vista:** aquí se representa el modelo de una forma adecuada para interactuar con el usuario, usualmente es la interfaz de usuario.
- **Controlador:** este responde a eventos, normalmente acciones del usuario, e invoca peticiones al modelo y la vista.

A continuación se va a proceder a describir cada uno de las partes del modelo MVC en nuestra aplicación. Se ha decidido describir cada una de las partes por separado para hacer más sencilla su comprensión. Se explicará en la medida de lo posible donde se produce la interacción entre las distintas capas de la arquitectura MVC.

5.1.1. Modelo

Para la descripción del modelo hemos utilizado el modelo entidad/relación para detallar la estructura de la base de datos puesto que nuestra base de datos es

relacional y con este modelo se visualiza fácilmente como está estructurada y cuáles son sus relaciones.

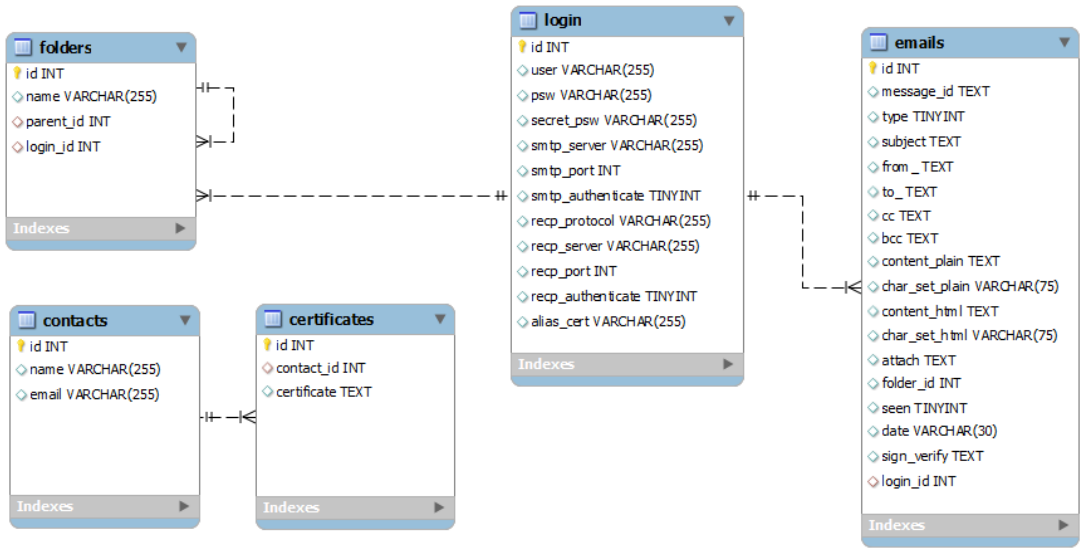

Ilustración 10 - Modelo relacional

5.1.2. Vista

Para mostrar el diseño de la vista, se va a utilizar un diagrama de clases donde se muestran las diferentes clases involucradas en dicha parte. Y después describiremos que hace cada una de las clases.

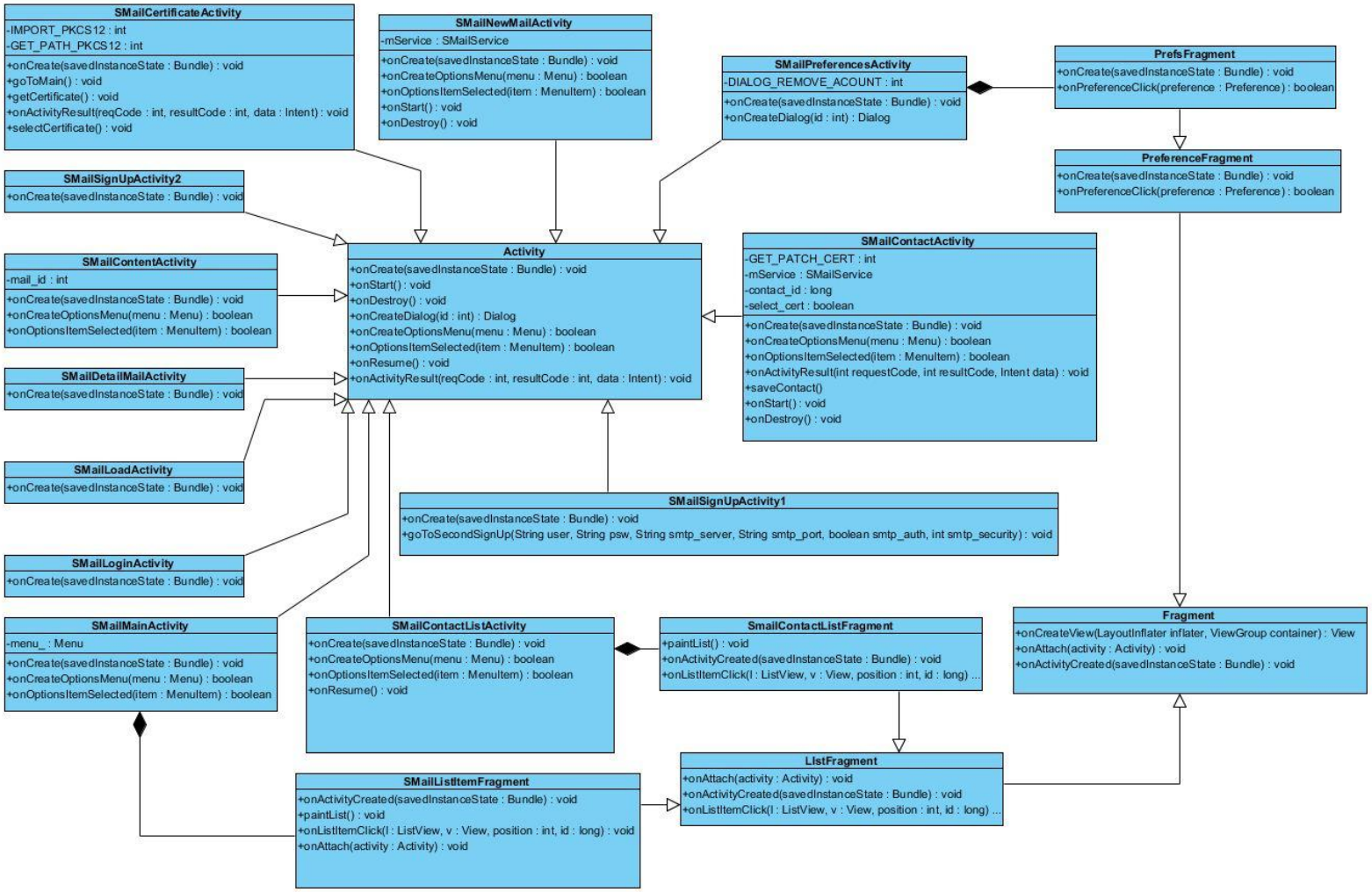

Ilustración 11 - Diagrama de clases

Las clases mostradas en el diagrama de clases son las correspondientes a la vista de la aplicación. En el caso de Android todas las interfaces de la aplicación heredan de las clases *Activity* o *Fragment*. Estás clases están compuestas por muchos más métodos pero solo hemos incluido los que usamos en nuestro trabajo. Otras clases propias de Android son *ListFragment* y *PreferenceFragment*, que heredan de la clase *Fragment*.

Ahora detallaremos cada una de las clases diseñadas para el gestor de correo seguro, el orden en el que se explicarán es el orden de navegación mientras se utiliza la aplicación:

- ***SMailLoadActivity*:** Esta es la primera interfaz que se encuentra el usuario y en ella aparece una barra de carga. Internamente la clase se encarga de comprobar si existe alguna cuenta creada o no. Si no existe ninguna cuenta redirige al usuario a la pantalla de registro (*SMailSignUpActivity1*). En caso contrario iría a la pantalla de registro (*SMailLoginActivity*).

- ***SMailSignUpActivity1*:** Es la clase encargada de recopilar los datos referentes a usuario, contraseña y datos del servidor SMTP y pasárselo a la segunda vista de registro (*SMailSignUpActivity2*).

- ***SMailSignUpActivity2*:** Recibe los datos de la primera pantalla de registro, y recopila los datos referentes al servidor de IMAP y la contraseña que proveerá de seguridad a la aplicación. Una vez recopilado todos los datos comprueba si los datos son correctos, mediante el uso de la clase *SMailMail* (se describirá en el apartado 5.1.3 Controlador). Si son correctos continua a la clase *SMailCertificateActivity*, en caso contrario vuelve a solicitar los datos al usuario.

- ***SMailCertificateActivity*:** Es la clase dedicada a la gestión de la clave privada del usuario. Desde aquí se puede importar claves privadas al *KeyChain* de Android y/o seleccionar una clave del *KeyChain* para usarla en la aplicación para el firmado y descifrado de correos electrónicos.

- ***SMailLoginActivity*:** Se encarga de pedir al usuario la contraseña de acceso a la aplicación, y una vez autenticado lo redirige a la vista principal (*SMailMainActivity*).

- ***SMailMainActivity* y *SMailListItemFrament*:** La clase *SMailMailActivity* contiene a la clase *SMailListItemFragment* que es la encargada de mostrar los correos de una carpeta en forma de lista. Cuando se selecciona uno de los correos dirige al usuario a la clase *SMailContentActivity*, para visualizar su contenido. Desde está vista y siendo controlado por *SMailMainActivity* se puede acceder a las preferencias de la aplicación (*SMailPreferenceActivity*), al gestor de contactos (*SMailContactListActivity*), al gestor de clave privada (*SMailCertificateActivity*),ir a la ventana de creación de nuevos correos (*SMailNewMailActivity*) y sincronizar con el servidor para descargar los nuevos mensajes recibidos (esto se realiza mediante el uso de la clase *SMailMail*).

- ***SMailContentActivity*:** Muestra el contenido del correo electrónico seleccionado y permite acceder a la pantalla de detalles del mensaje (*SMailDetailMailActivity*), reenviar o responder el correo electrónico.

- ***SMailDetailMailActivity*:** Se encarga de mostrar los detalles del correo electrónico y permite descargar los documentos adjuntos.

- ***SmailNewMailActivity*:** Desde esta clase se pueden crear nuevos correos, adjuntar documentos y enviar el correo electrónico a sus destinatarios. El correo puede estar sin firmar ni cifrar, o firmado y/o cifrado. El envío de los correos se realiza con la clase *SMailMail*.

- ***SMailPreferenceActivity* y *PrefsFragment*:** Se encargan de mostrar los ajustes de la aplicación, que son activar/desactivar firmar o cifrar por defecto los mensajes enviados, y también permite eliminar la cuenta del usuario, borrando todo el contenido de la aplicación.

- ***SMailContactListActivity* y *SMailContactListFragment*:** Estas clases forman el gestor de contactos, donde se visualizan en forma de listado los contactos existentes. La clase *SMailContactListActivity* contiene a la clase *SMailContactListFragment* que es la encargada de mostrar el listado, y *SMailContactListActivity* es la encargada de mostrar el menú que permite la creación de nuevos contactos.

- ***SMailContactActivity*:** En esta clase se puede crear un contactos nuevo o editar un contacto existente, permitiendo importar un certificado de clave pública para el contacto, con el fin de poder enviarle correos electrónicos cifrados.

5.1.3. Controlador

Una vez descrita la capa de vista, se procede a comentar la capa controlador, donde se encuentran las clases encargadas de realizar las operaciones de mayor computo. Como en el caso anterior procederemos a mostrar el diagrama de clases y después comentaremos las principales tareas que realiza.

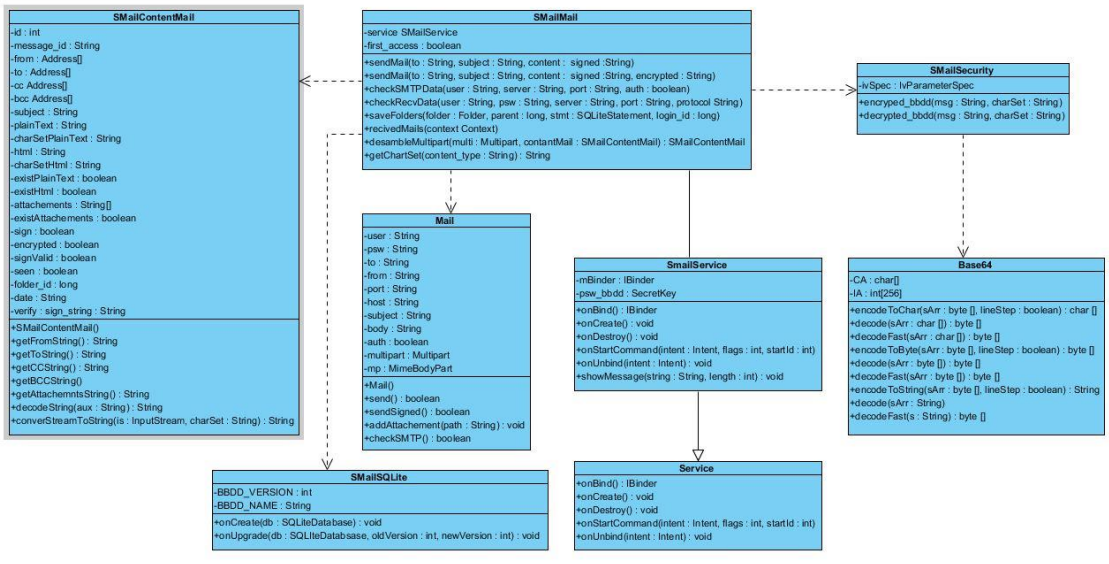

Ilustración 12 - Diagrama de clases, capa controlador

Para una mayor simplicidad del diagrama de clases se han omitido los métodos *get* y *set* de los atributos de las clases. En este diagrama de clases también se encuentra una clase de la librería de Android, la clase Servicio. Como se comentó anteriormente, ahora se procede a detallar las funcionalidades de cada clase.

- ***SMailContentMail*:** Esta se utiliza para almacenar los datos de los correos electrónicos recibidos, para después guardarlos en la base de datos.

- ***SMailMail*:** Es la clase principal, se encarga de recibir y enviar correos, así como firmarlos, cifrarlos, verificar la firma y descifrar los mensajes. También comprueba que los datos insertados por el usuario a la hora de registrarse sean correctos. Para poder realizar estas funcionalidades se ayuda de las demás clases que aparecen relacionadas con ella en el diagrama de clases. También hace uso de las librerías de *JavaMail* y *BouncyCastle* (estas librerías serán descritas en el apartado 6.2.4 Descripción de la implementación y APIs utilizadas).

- ***Mail*:** Es la encargada de enviar los correos electrónicos mediante el protocolo SMTP.

- ***SMailSQLite*:** Para poder utilizar SQLite en Android es necesario implementar esta clase, que es la que permite la interacción con la base de datos.

- ***SMailSecurity*:** Esta clase está compuesta por dos funciones que son las que cifran y descifran los datos utilizando la clave de cifrado almacenada durante el uso de la aplicación el servicio *SMailService*.

- ***Base64*:** Clase de ayuda encargada de la codificación y decodificación en *Base64*.

- ***SMailService*:** Servicio de la aplicación donde almacenamos la clave de cifrado/descifrado. En versiones futuras será la clase encargada de sincronizar los correos automáticamente.

5.1.4. Diagramas de secuencia

Para mostrar las relaciones existentes entre las clases diseñadas y los casos de uso se va a proceder a realizar los diagramas de secuencia de cada caso de uso. Como simplificación se han realizado los diagramas de secuencia asumiendo que el usuario ya se encontraba en la vista necesaria para hacer la acción, haciendo referencia a las precondiciones nombradas en los casos de uso.

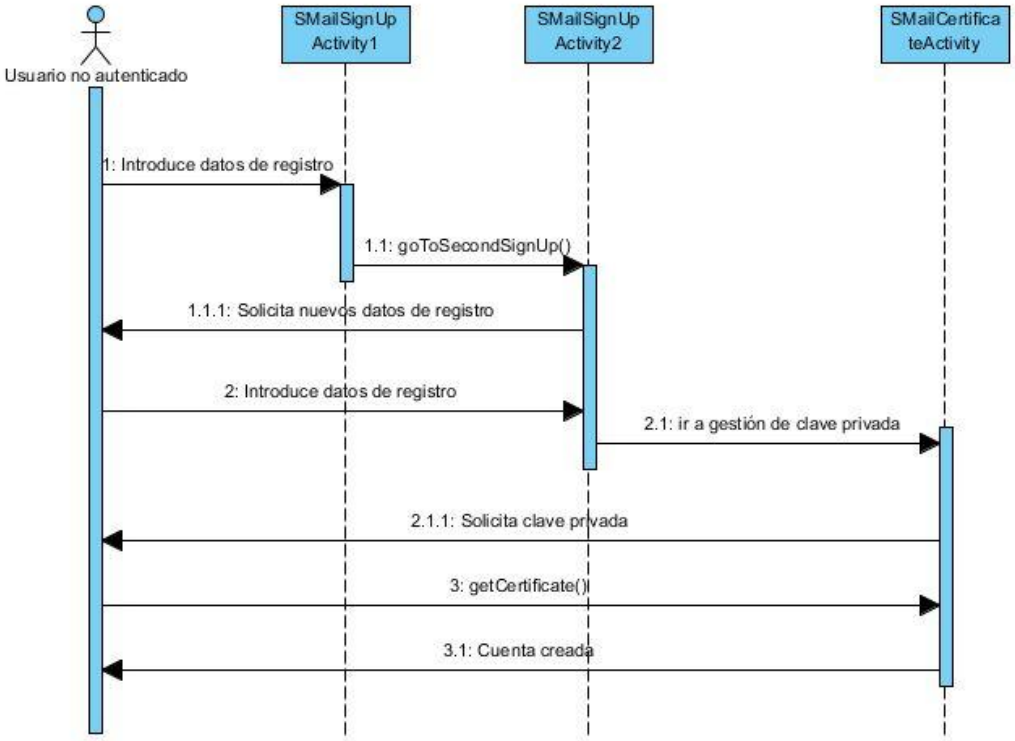

Ilustración 13 - Diagrama de secuencia, crear cuenta

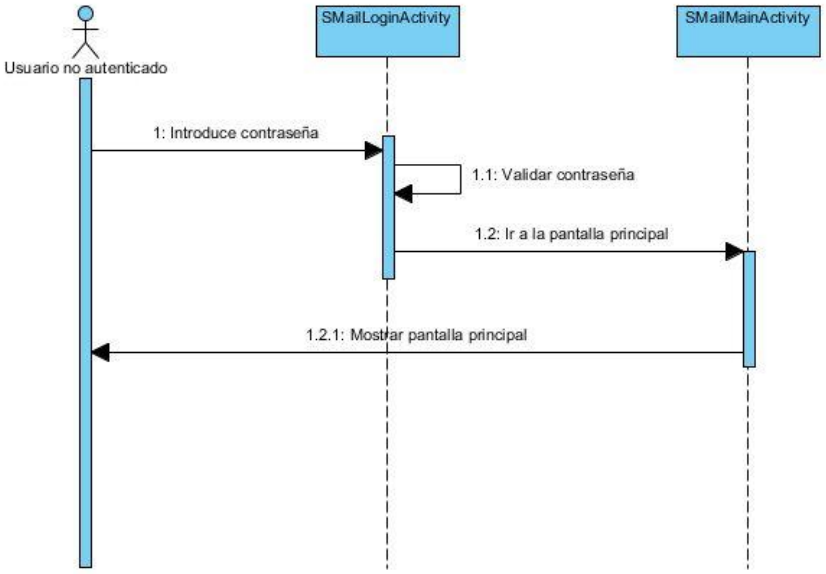

Ilustración 14 - Diagrama de secuencia, autenticarse

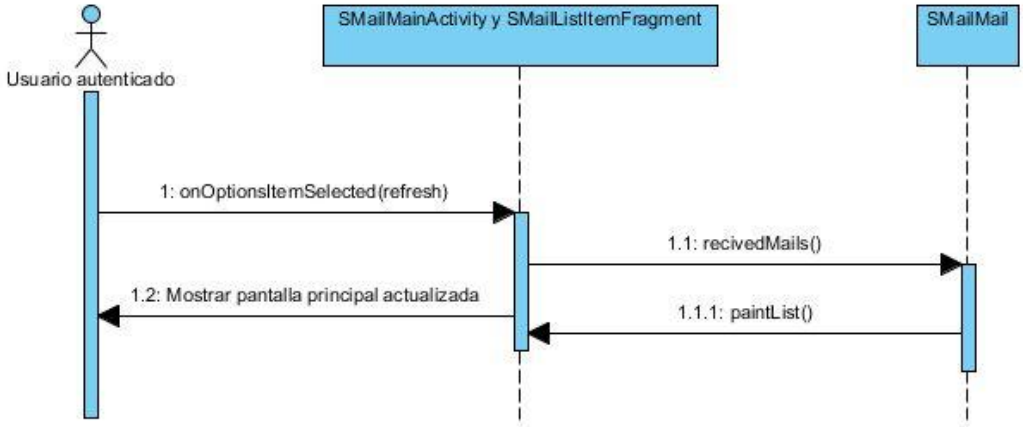

Ilustración 15 - Diagrama de secuencia, descargar correos electrónicos

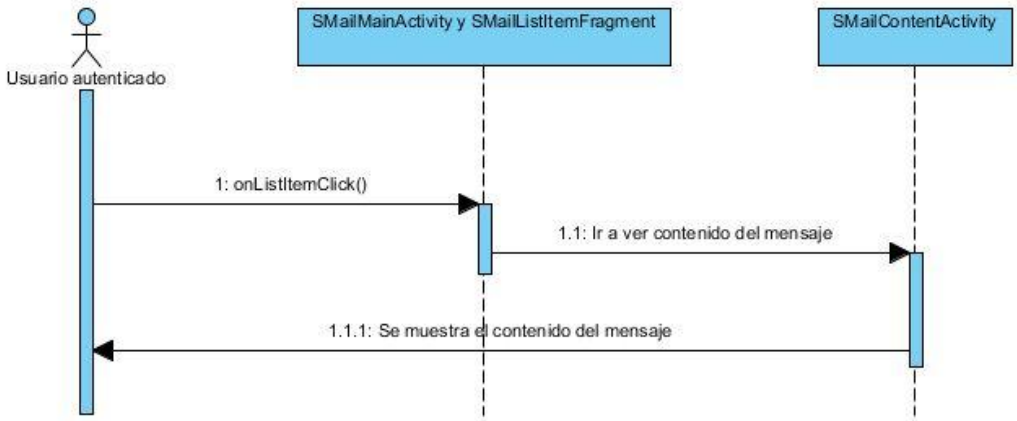

Ilustración 16 - Diagrama de secuencia, leer correo electrónico

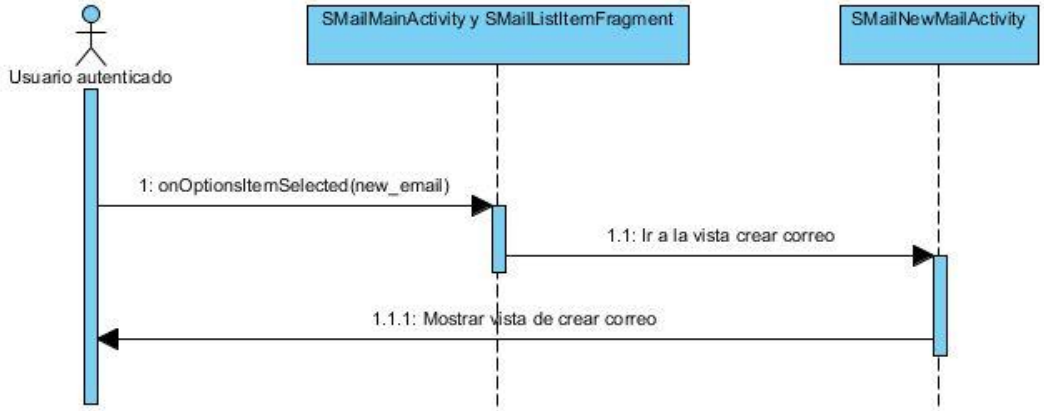

Ilustración 17 - Diagrama de secuencia, crear correo electrónico

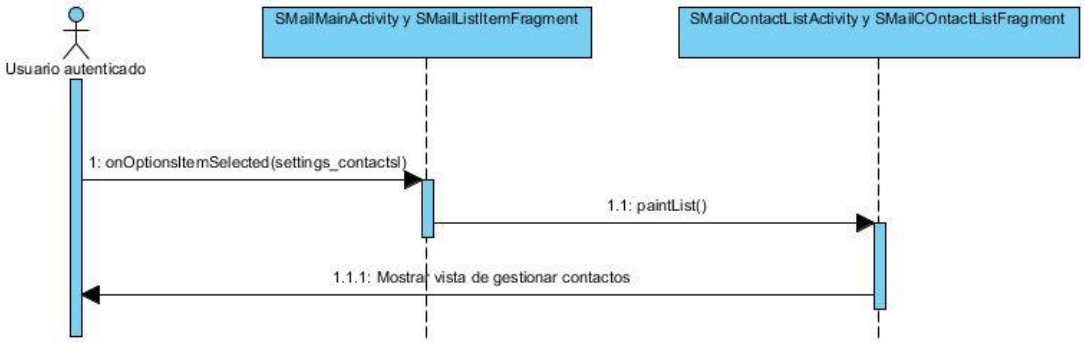

Ilustración 18 - Diagrama de secuencia, gestionar contactos

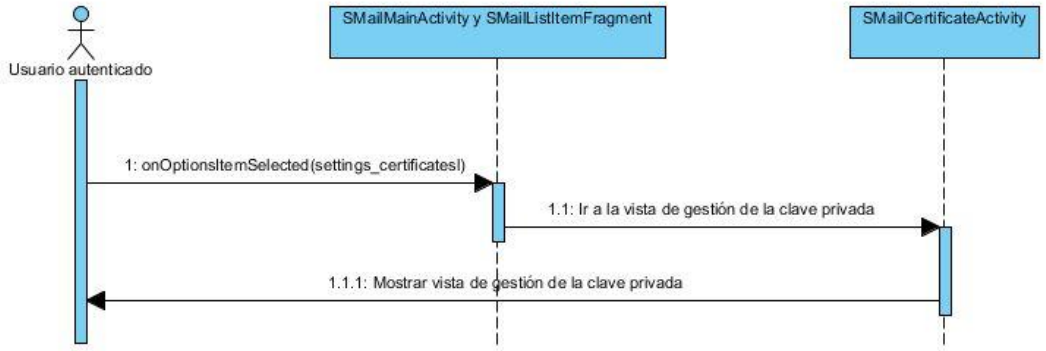

Ilustración 19 - Diagrama de secuencia, gestionar clave privada

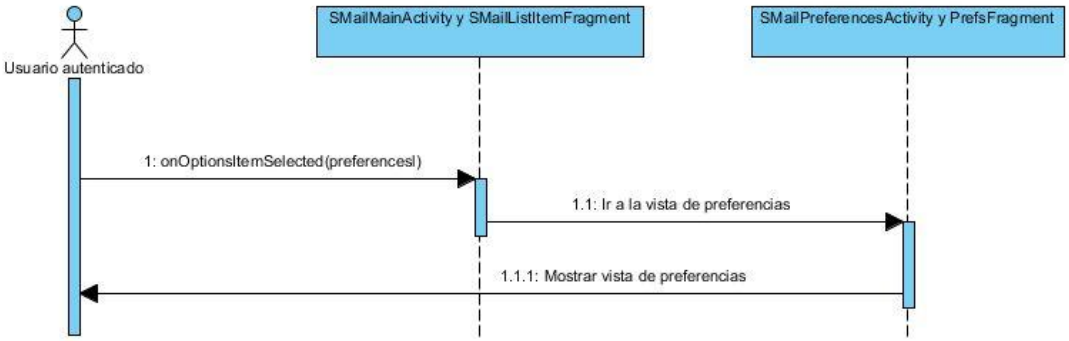

Ilustración 20 - Diagrama de secuencia, ver ajustes

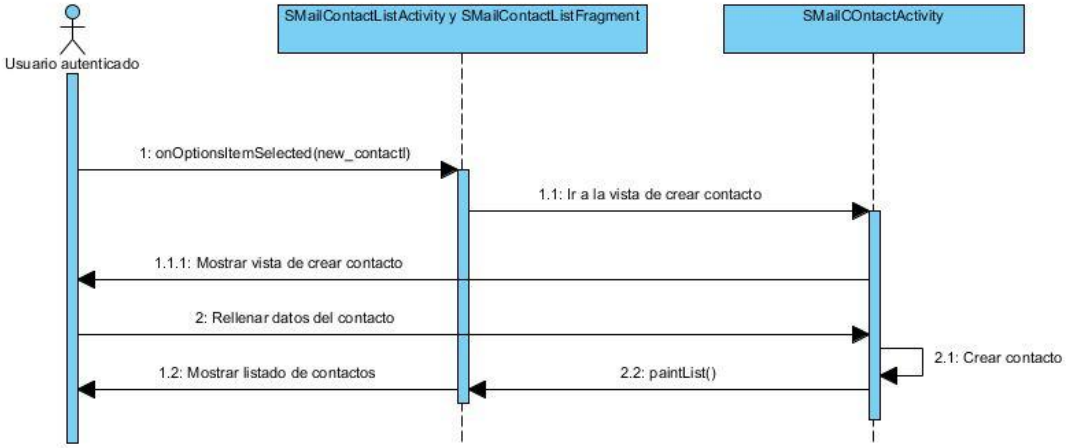

Ilustración 21 - Diagrama de secuencia, crear contacto

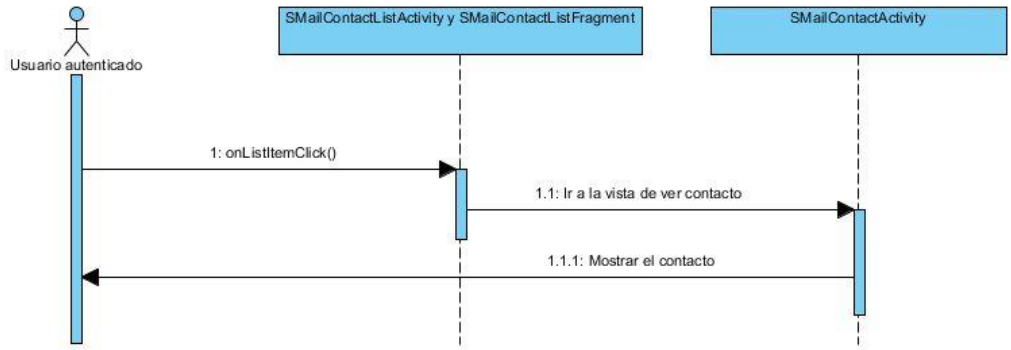

Ilustración 22 - Diagrama de secuencia, ver contacto

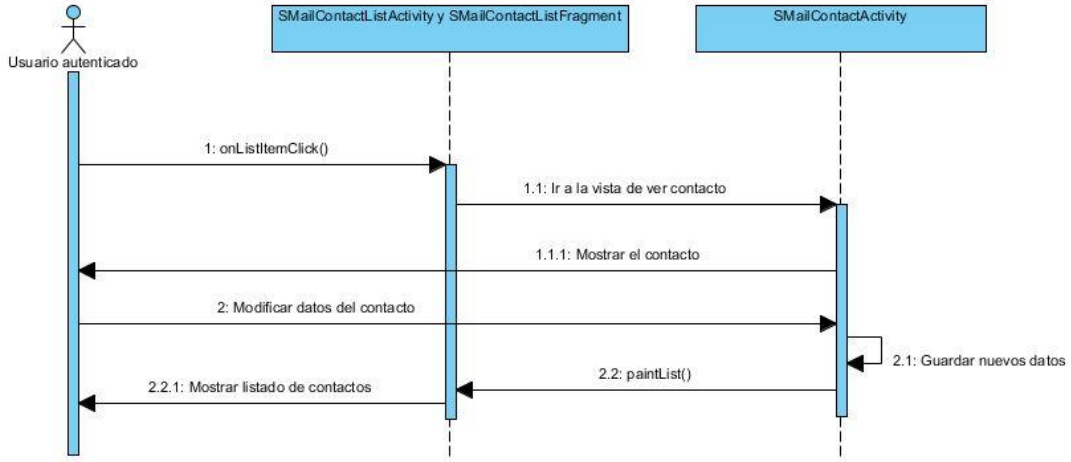

Ilustración 23 - Diagrama de secuencia, editar contacto

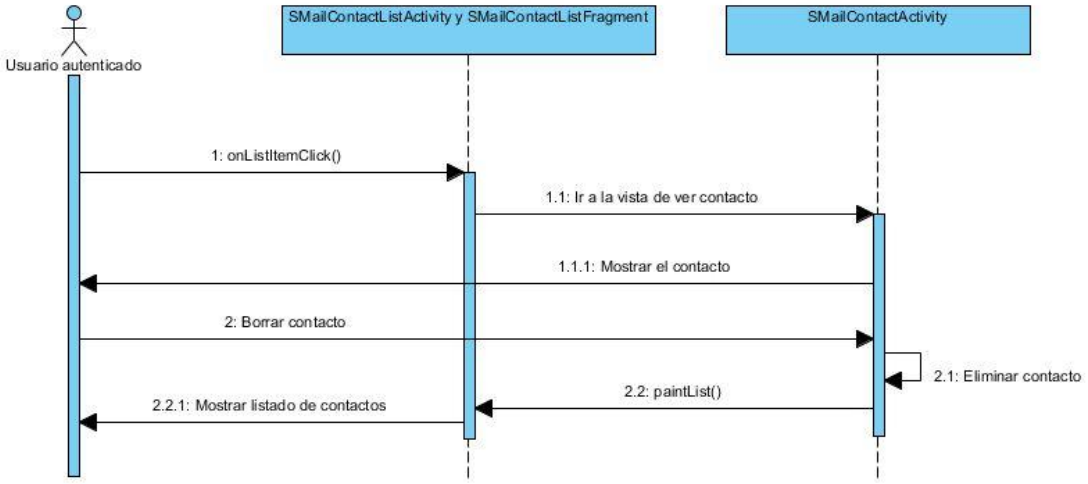

Ilustración 24 - Diagrama de secuencia, borrar contacto

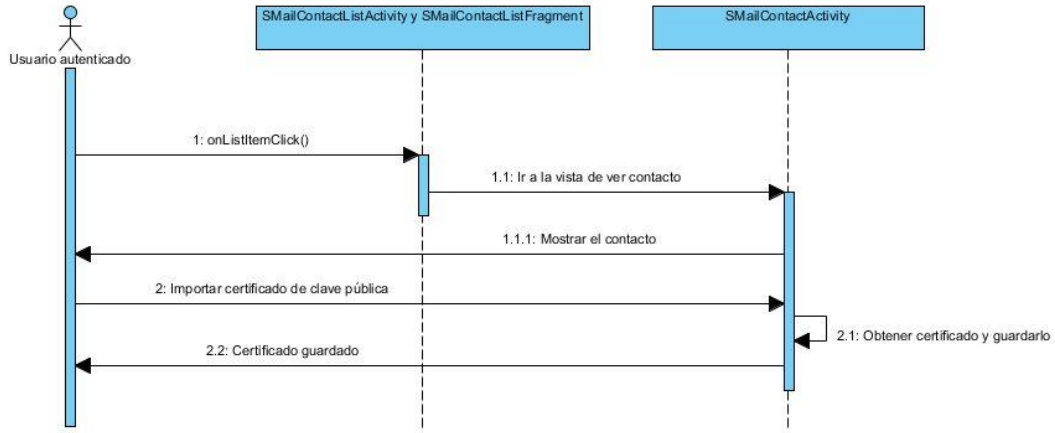

Ilustración 25 - Diagrama de secuencia, importar certificado de clave pública

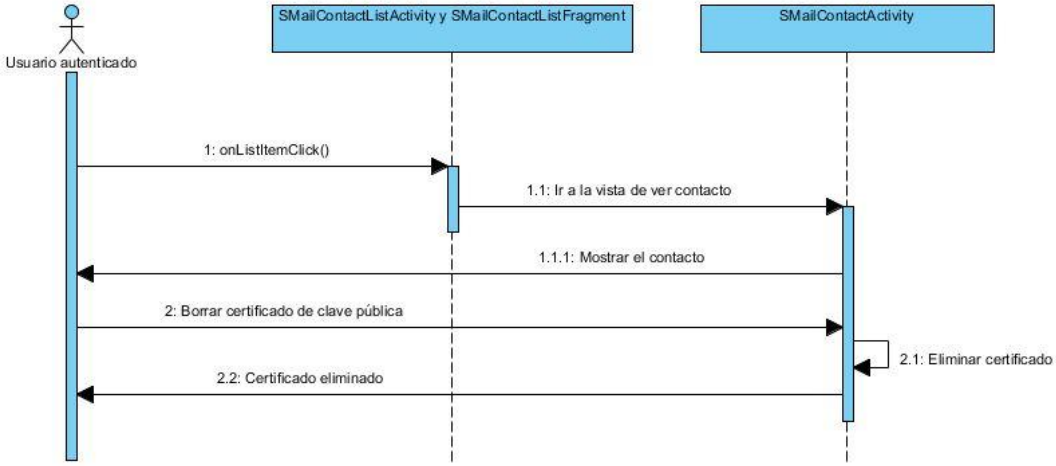

Ilustración 26 - Diagrama de secuencia, borrar certificado de clave pública

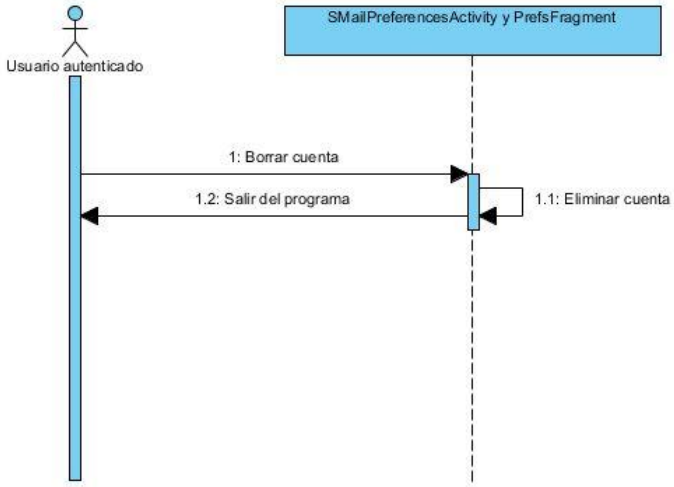

Ilustración 27 - Diagrama de secuencia, borrar cuenta

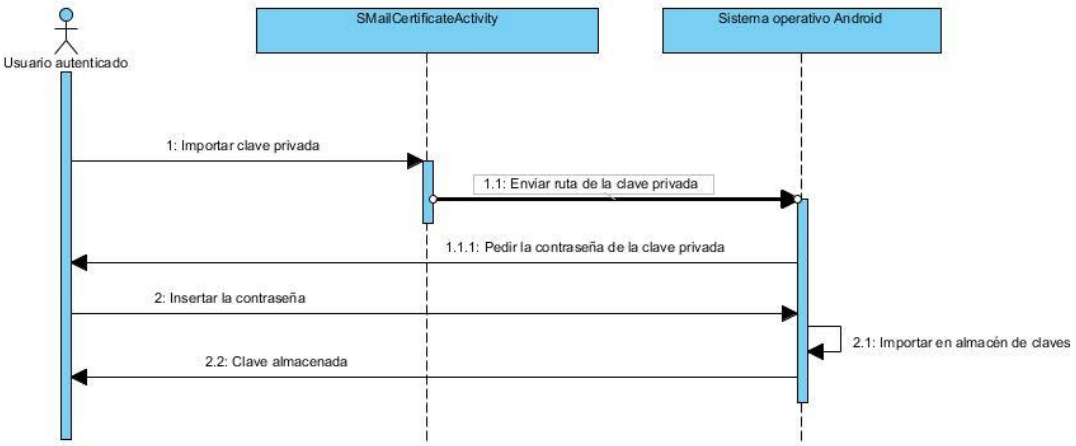

Ilustración 28 - Diagrama de secuencia, importar clave privada

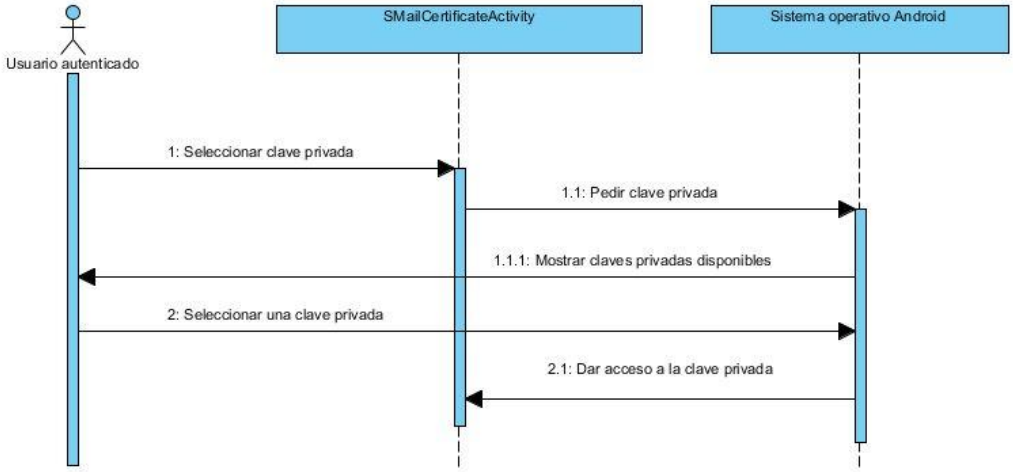

Ilustración 29 - Diagrama de secuencia, seleccionar clave privada

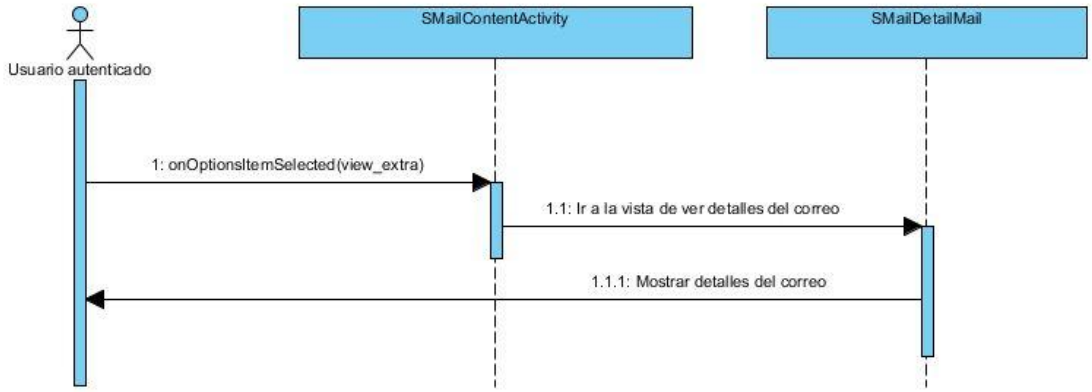

Ilustración 30 - Diagrama de secuencia, ver detalles del mensaje

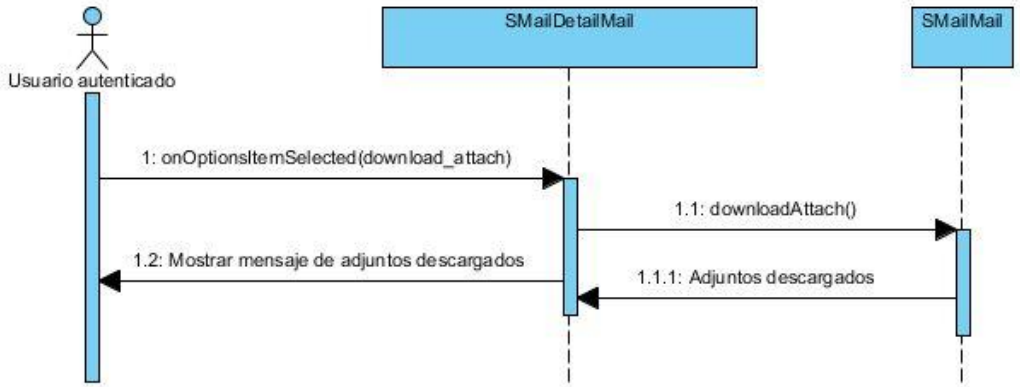

Ilustración 31 - Diagrama de secuencia, descargar adjuntos

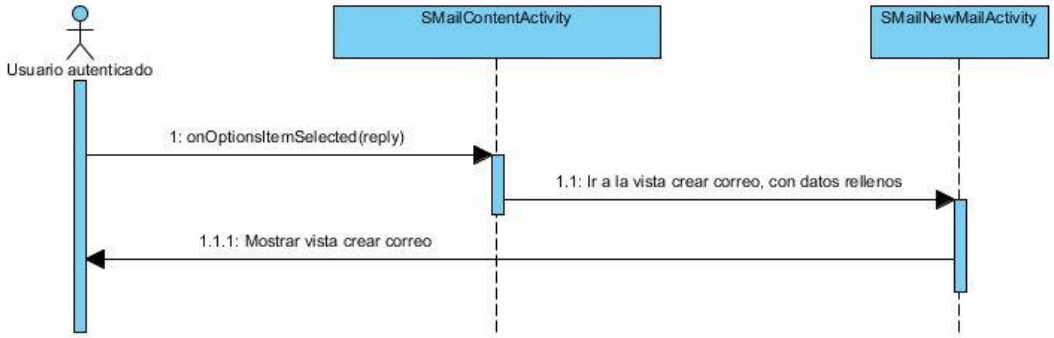

Ilustración 32 - Diagrama de secuencia, responder correo electrónico

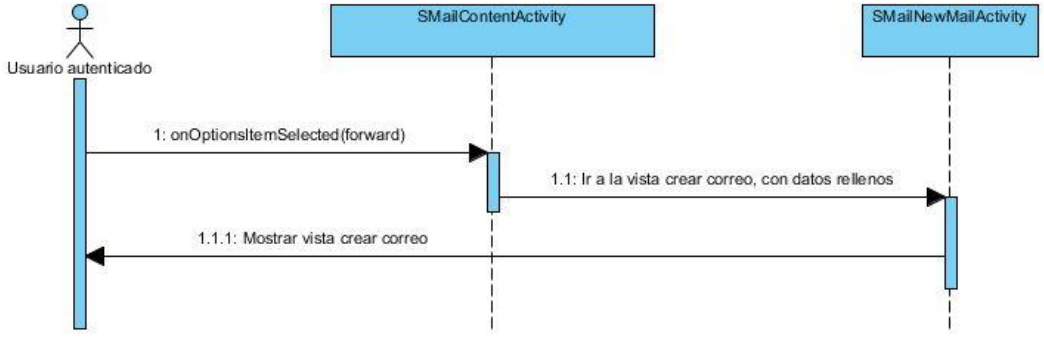

Ilustración 33 - Diagrama de secuencia, reenviar correo electrónico

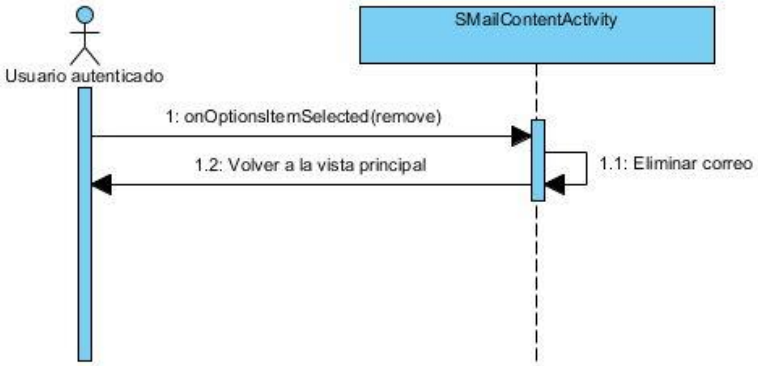

Ilustración 34 - Diagrama de secuencia, borrar correo electrónico

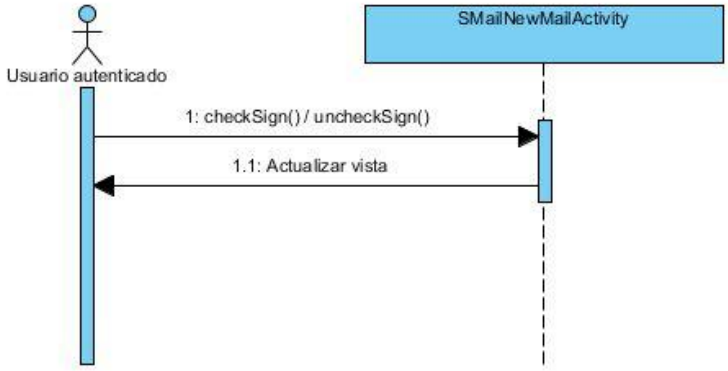

Ilustración 35 - Diagrama de secuencia, activar/desactivar firmar correo electrónico

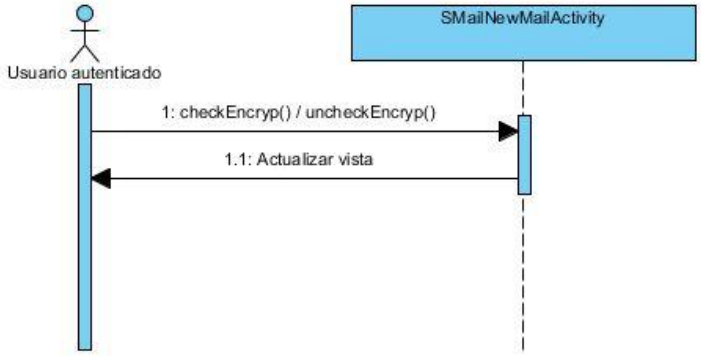

Ilustración 36 - Diagrama de secuencia, activar/desactivar cifrar correo electrónico

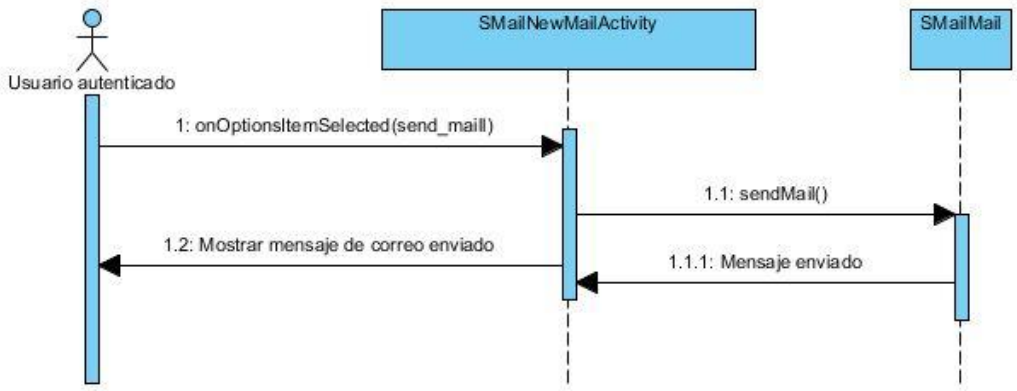

Ilustración 37 - Diagrama de secuencia, enviar correo electrónico

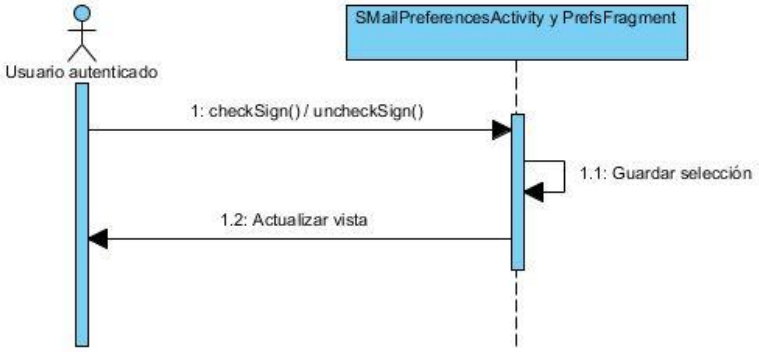

Ilustración 38 - Diagrama de secuencia, activar/desactivar firmado por defecto

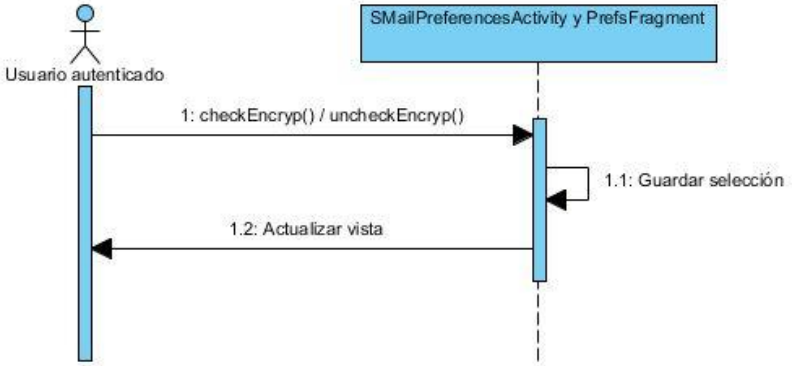

Ilustración 39 - Diagrama de secuencia, activar/desactivar cifrado por defecto

5.2. Medidas de seguridad

Como en toda aplicación informática se debe tener en cuenta en todo momento la seguridad de los datos sensibles, sobre todo como es en este caso, cuando se trata de aplicaciones que van a ser utilizadas en dispositivos móviles. Ya que estos dispositivos pueden llegar a ser extraviados y/o robados más fácilmente que otros dispositivos.

En nuestro sistema contamos con varios tipos de datos sensibles, las contraseñas del usuario, el contenido de los correos enviados y recibidos y la clave privada para el descifrado y firmado de mensajes.

Por lo comentado anteriormente, se procede a explicar las medidas de seguridad utilizadas en esta aplicación.

5.2.1. Almacenamiento de la clave privada y su certificado de clave pública

Para almacenar de forma segura la clave privada se almacenará en el KeyChain de Android, esto es un almacén de claves proveído por Android, en el cual se pueden almacenar claves privadas, y su correspondiente certificado de clave pública, de manera segura. Para ello se le solicita al usuario que indique la dirección del fichero en el dispositivo, y se le comunica el destino del fichero al almacén de claves. El propio sistema operativo es el que se encarga de pedirle al usuario la clave con la que descifrar la clave privada (en formato PKCS#12) e importar la clave privada, junto al certificado de clave pública y su cadena de certificación, en el KeyChain.

Cuando se utiliza por primera vez el almacén de claves y no se tiene definida una clave de bloqueo en el dispositivo móvil se obliga al usuario a poner una contraseña, mejorando así la seguridad del dispositivo.

Una vez que el usuario ha importado alguna clave privada en el almacén de claves, se necesita que el usuario de permiso a la aplicación para utilizar la clave privada. Para ello, se le solicita al usuario que seleccione una clave privada del almacén de claves. Una vez que el usuario ha dado acceso a nuestra aplicación, la aplicación ya puede utilizar esta clave privada.

Con este método delegamos el control del almacenamiento de la clave privada al sistema operativo. Este método tiene el inconveniente de que puede ser objeto de *phishing*, por lo que una aplicación malintencionada podría hacerse pasar por nuestra aplicación y solicitar el acceso de la clave privada al usuario, y posteriormente utilizarla con fines maliciosos.

5.2.2. Contraseñas y correos

Para el almacenamiento seguro de los datos en la base datos se utilizará una clave de usuario, está clave se solicitará al usuario en el registro de la aplicación. Con esta clave se derivará dos claves diferentes, una se utilizará para cifrar y descifrar los datos (correos electrónicos y contraseña del servidor de correos). Y la otra se almacenará en la base de datos para realizar la confirmación de la clave en futuros accesos del usuario.

Para la derivación de las claves utilizaremos *Password Based Encryption* y más concretamente el algoritmo definido por PKCS5 v2. Para derivar cada una de las claves utilizaremos un Salt diferente y un mismo número de iteraciones, para que las dos sean igual de robustas frente ataques de fuerza bruta. Y como tamaño de clave derivada utilizaremos 128 bits.

Al utilizar *Password Based Encryption* conseguimos que aunque el usuario no utilice un alfabeto suficientemente grande para generar la clave, el intentar obtener su derivación mediante fuerza bruta sea inviable, esto lo conseguimos asignando un número de iteraciones que hagan que generar la clave tenga un alto coste computacional. En nuestro caso utilizaremos el número de iteraciones que hagan que derivar la clave tarde un segundo en un dispositivo con un procesador doble núcleo a 1 GHz. Teniendo en cuenta que al usuario se le obliga a introducir una contraseña de mínimo 8 caracteres y debe contener mínimo letras y números, conseguimos que el número de posibilidades sea de entorno a los dos billones, y teniendo en cuenta que para probar cada una de las posibilidades se tardaría en ejecutar en torno a un segundo hace que las posibilidades de utilizar ataques por fuerza bruta sean inviables.

Para almacenar los datos sensibles de forma segura se utilizará el algoritmo AES 128 en modo CTR y *padding* PCKS1 utilizando para ello la clave derivada para el cifrado/descifrado de datos.

5.3. Descripción de los paquetes y clases

Para el desarrollo del trabajo se han creado tres paquetes, aparte de las librerías importadas, los cuales se detallan a continuación.

- ***smail*:** contiene las clases que ejecutan los controladores, es decir, donde se procesan los datos, se crean, envían, reciben los correos.
- ***activity*:** contiene las todas las clases relativas a las pantallas de visualización en el dispositivo Android.
- ***utils*:** contiene algunas clases de ayuda.

6. Implementación y pruebas

6.1. Entorno de desarrollo

El desarrollo del trabajo se ha llevado a cabo en un ordenador con las siguientes características.

- Procesador Intel Core 2 Duo 3GHz.
- 8 GB de RAM.
- 10 GB de disco duro libre.

Y el software necesario para su desarrollo ha sido:

- Sistema operativo Windows 7.
- Java SDK versión 1.6.
- Eclipse Classic 3.7.1
- Android Development Tools Plugin for Eclipse
- Android SDK Manager
- Android 4.0.3
- API JavaMail 1.4.1
- API SpongyCastle
- GanttProject
- MySQL Workbench 5.2
- Microsoft Office
- Visual Paradigm

6.2. Implementación

6.2.1. Estándares de diseño

Para la implementación del trabajo se ha usado los siguientes estándares de diseño:

- Programación orientada a objetos: es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas informáticos.

- Modelo relacional: es un modelo de datos basado en la lógica de predicados y la teoría de conjuntos. Se basa fundamentalmente en las relaciones existentes entre los campos de diferentes tablas, aunque también existen dentro de una misma tabla.

6.2.2. Convenciones de nombrado

En la implementación de los paquetes y clases del sistema se ha aplicado las siguientes normas de nombrado:

- Los paquetes son nombrados con todas las letras en minúscula, y tienen que empezar con es.acaceres.smail.
- Las clases son nombradas con la primera letra en mayúscula y cuando corresponden a una pantalla del sistema terminan en Activity.

6.2.3. Herramientas de desarrollo software

Para el desarrollo del sistema se han utilizado las siguientes herramientas:

- **Eclipse Classic:** Para la generación del código fuente.
- **MySQL WorkBench:** Para la creación del diagrama relacional.

6.2.4. Descripción de la implementación y APIs utilizadas

A la hora de implementar el sistema se ha necesitado varias APIs diferentes a las que trae el sistema operativo Android, en nuestro caso hemos necesitado usar la API de *JavaMail* [26] y la de *BouncyCastle* [27]. La primera nos ha hecho falta para poder enviar y recibir correos electrónicos desde el dispositivo Android y la segunda el cifrado y firmado de los correos, es decir, para la implementación de S/MIME.

A continuación describiremos las tecnologías previamente mencionadas y el papel que han tenido en el desarrollo de la aplicación. Después continuaremos explicando las medidas de seguridad tomadas a la hora de desarrollar la aplicación. Continuaremos

con la descripción de la arquitectura empleada. También se detallaran los paquetes y clases utilizados y por último se muestra el modelo relacional utilizado para almacenar los datos de la aplicación.

6.2.4.1. JavaMail

El API de *JavaMail* es un paquete opcional de java para poder leer, crear, enviar y recibir correos electrónicos. En este paquete vienen incluidas las clases necesarias para interactuar con los correos. A la hora de utilizar esta API en nuestro trabajo hemos tenido el inconveniente de que no existe para Android, ya que, aunque Android está basado en Java no tiene todas las funcionalidades del mismo. Así se ha tenido que buscar si existía algo parecido para este sistema operativo. Finalmente se terminó encontrando la misma librería portada a Android [28]. En concreto la versión portada, es la 1.4.1 de la librería de JavaMail.

6.2.4.2. BouncyCastle

El paquete *BouncyCastle* está compuesto por varias librerías criptográficas entre las que se encuentra S/MIME, por lo se evitó tener que implementar el cifrado, descifrado, firmado y validación de los correos electrónicos. En este caso Android tiene en su API la librería BouncyCastle pero no la última versión, por lo que falta el paquete de S/MIME. Se decidió buscar la librería de *BouncyCastle* portada a Android y se encontró, pero en este caso la llamarón *SpongyCastle* [29] para que no existiera problemas de espacio de nombre al utilizarla en Android.

Una vez importada la librería necesaria para utilizar S/MIME se ha podido utilizar para cifrar, descifrar, firmar y validar la firma de los correos.

6.2.4.2.1. Firmar un mensaje

Para poder firmar un correo electrónico es preciso obtener la clave privada del firmante. Dicha clave (si existe) está guardada en el almacén de claves del sistema operativo. Una vez solicitada y obtenida la clave privada se aplica una función resumen al contenido del correo electrónico y se cifra con la clave privada, obteniendo así la

firma del correo. En el correo se añade como adjunto el certificado de clave pública del usuario para que los receptores del mensaje puedan validar la firma.

6.2.4.2.2. Verificar la firma de un mensaje

La validación del correo se realiza haciendo dos comprobaciones, por un lado se verifica la firma y por otro lado se verifica que el certificado sea válido. Para comprobar que la firma sea válida se realiza la misma función resumen al contenido del correo y se compara con la firma que tiene el correo, esta firma hay que descifrarla con la clave pública del contacto origen, y si son iguales es que la firma es válida y estamos seguros que nadie ha modificado el correo, asegurando así la integridad del mensaje. Aunque la firma se verifique, se tiene que comprobar que el certificado sea válido, para ello comprobamos que la cadena de certificación sea correcta (que este firmado por una autoridad certificadora en la que confiemos), también comprobamos que la fecha actual se encuentre dentro del periodo de utilización del certificado y que la dirección de correo electrónico del certificado sea la misma que la dirección origen del correo electrónico. Si alguno de los casos anteriores da no es verificado no se puede considerar la firma del correo electrónico como válida.

6.2.4.2.3. Cifrar un mensaje

El método de cifrado requiere la clave pública del destinatario, estas claves las almacenamos en *base64* en la base de datos, así que obtenemos la clave pública del destinatario, generamos una clave pseudoaleatoria con la que ciframos el contenido del mensaje con un algoritmo de cifrado simétrico, y la clave pseudoaleatoria es cifrada con la clave pública de cada uno de los destinarios. Una vez cifrado el mensaje se adjunta como un fichero llamado smime.p7m en donde se encuentra el contenido cifrado. Los gestores de correo que no soporten S/MIME solo podrán ver que existe un archivo adjunto pero no podrán descifrarlo, y los que si soportan S/MIME descifrarán el contenido y se lo mostrarán al usuario.

6.2.4.2.4. Descifrar un mensaje

Para descifrar un correo electrónico necesitamos la clave privada del usuario, por lo que solicitamos la clave al almacén de claves del sistema operativo, y con ella desciframos la clave de sesión con la que se cifro el mensaje. Y con esta clave de sesión y utilizando el mismo algoritmo de cifrado simétrico desciframos el contenido del mensaje.

6.2.4.2.5. Firmar y cifrar un mensaje

En el caso de querer enviar un mensaje firmado y cifrado, primero se tiene que proceder a la firma del contenido del mensaje (de la misma forma en la que se explicó anteriormente) y después se cifra el contenido del mensaje con la firma incluida (se cifra realizando la misma metodología anterior). Descifrar y verificar la firma de un mensaje

Cuando recibimos un mensaje firmado y cifrado, primero procedemos a descifrar el mensaje como se explicó anteriormente y una vez descifrado, verificamos la firma del mensaje recibido.

6.3. Pruebas realizadas

Para asegurar que el producto final cumple con los requisitos de usuario se realizaron las siguientes pruebas de aceptación.

Cada prueba quedará definida a través de la siguiente información:

- Identificador de la prueba: formado por PA- y tres dígitos.
- ID del requisito de usuario
- Resumen del requisito
- Especificaciones de entrada
- Especificaciones de salida
- Criterio de aceptación

Las pruebas se realizaron en el emulador proveído por Android y en un Samsung Galaxy Nexus. En los dos entornos se ejecuta Android 4.0.3.

Para que la prueba se considere aceptada se tendrá que cumplir el criterio de aceptación, ya que esto implica que el sistema se comporta como se esperaba.

PA-001	
Requisito de usuario	UR-C001
Resumen requisito	Tiene que poder enviar correos electrónicos sin firmar ni cifrar o firmados y/o cifrados.
Especificaciones de entrada	El usuario envía el correo sin firmar ni cifrar.
Especificaciones de salida	El mensaje es enviado al destinatario.
Criterio de aceptación	El mensaje llega al destinatario.

Tabla 51 - PA-001

PA-002	
Requisito de usuario	UR-C001
Resumen requisito	Tiene que poder enviar correos electrónicos sin firmar ni cifrar o firmados y/o cifrados.
Especificaciones de entrada	El usuario selecciona firmar y envía el correo.
Especificaciones de salida	El mensaje firmado es enviado al destinatario.
Criterio de aceptación	El mensaje llega al destinatario con una firma valida.

Tabla 52 - PA-002

PA-003	
Requisito de usuario	UR-C001
Resumen requisito	Tiene que poder enviar correos electrónicos sin firmar ni cifrar o firmados y/o cifrados.
Especificaciones de entrada	El usuario selecciona cifrar y envía el correo.
Especificaciones de salida	El mensaje cifrado es enviado al destinatario.
Criterio de aceptación	El mensaje llega al destinatario cifrado, y se puede descifrar.

Tabla 53 - PA-003

PA-004	
Requisito de usuario	UR-C001
Resumen requisito	Tiene que poder enviar correos electrónicos sin firmar ni cifrar o firmados y/o cifrados.
Especificaciones de entrada	El usuario selecciona firmar y cifrar y envía el correo.
Especificaciones de salida	El mensaje firmado y cifrado es enviado al destinatario.
Criterio de aceptación	El mensaje llega al destinatario cifrado, se puede descifrar y verificar la firma.

Tabla 54 - PA-004

PA-005	
Requisito de usuario	UR-C002
Resumen requisito	Tiene que poder gestionarse los contactos.
Especificaciones de entrada	El usuario crea un contacto.
Especificaciones de salida	Se crea el contacto en la base de datos.
Criterio de aceptación	El nuevo contacto aparece en la base de datos.

Tabla 55 - PA-005

PA-006	
Requisito de usuario	UR-C002
Resumen requisito	Tiene que poder gestionarse los contactos.
Especificaciones de entrada	El usuario modifica un contacto.
Especificaciones de salida	Se modifica el contacto en la base de datos.
Criterio de aceptación	La información modificada aparece modificada en la base de datos.

Tabla 56 - PA-006

PA-007	
Requisito de usuario	UR-C002
Resumen requisito	Tiene que poder gestionarse los contactos.
Especificaciones de entrada	El usuario elimina un contacto.
Especificaciones de salida	Se elimina el contacto en la base de datos.
Criterio de aceptación	Los datos del contacto desaparecen de la base de datos.

Tabla 57 - PA-007

PA-008	
Requisito de usuario	UR-C003
Resumen requisito	Se tiene que garantizar la privacidad de los datos privados del usuario.
Especificaciones de entrada	El usuario se da de alta en la aplicación.
Especificaciones de salida	Se crean los datos del usuario.
Criterio de aceptación	La información privada se almacena cifrada y se puede descifrar con la clave introducida por el usuario.

Tabla 58 - PA-008

PA-009	
Requisito de usuario	UR-C003
Resumen requisito	Se tiene que garantizar la privacidad de los datos privados del usuario.
Especificaciones de entrada	El usuario ingresa un certificado de clave privada.
Especificaciones de salida	Se transmite la dirección del certificado al almacén de claves.
Criterio de aceptación	Se comprueba que no se puede acceder a la clave privada sin el permiso explícito del usuario.

Tabla 59 - PA-009

PA-010	
Requisito de usuario	UR-C004
Resumen requisito	Se tiene que poder gestionar los certificados públicos de los contactos.
Especificaciones de entrada	El usuario introduce un certificado público de un contacto.
Especificaciones de salida	Se almacena en la base de datos.
Criterio de aceptación	El nuevo certificado público aparece en la base de datos.

Tabla 60 - PA-010

PA-011	
Requisito de usuario	UR-C004
Resumen requisito	Se tiene que poder gestionar los certificados públicos de los contactos.
Especificaciones de entrada	El usuario modifica un certificado público de un contacto.
Especificaciones de salida	Se borra el antiguo y se almacena el nuevo en la base de datos.
Criterio de aceptación	El antiguo certificado desaparece de la base de datos y el nuevo certificado público aparece en la base de datos.

Tabla 61 - PA-011

PA-012	
Requisito de usuario	UR-C004
Resumen requisito	Se tiene que poder gestionar los certificados públicos de los contactos.
Especificaciones de entrada	El usuario borra un certificado público de un contacto.
Especificaciones de salida	Se borra el de la base de datos.
Criterio de aceptación	El certificado desaparece de la base de datos.

Tabla 62 - PA-012

PA-013	
Requisito de usuario	UR-C005
Resumen requisito	Tiene que poder gestionarse la clave privada del usuario.
Especificaciones de entrada	El usuario ingresa un certificado de clave privada.
Especificaciones de salida	Se transmite la dirección del certificado al almacén de claves.
Criterio de aceptación	El certificado aparece en el almacén de claves y en la base de datos está el alias del certificado.

Tabla 63 - PA-013

PA-014	
Requisito de usuario	UR-C005
Resumen requisito	Tiene que poder gestionarse la clave privada del usuario.
Especificaciones de entrada	El usuario borra un certificado de clave privada.
Especificaciones de salida	Se borra el alias de la base de datos.
Criterio de aceptación	En la base de datos desaparece el alias del certificado.

Tabla 64 - PA-014

PA-015	
Requisito de usuario	UR-C006
Resumen requisito	Se tiene que poder enviar adjuntos.
Especificaciones de entrada	El usuario selecciona un fichero adjunto y envía el correo.
Especificaciones de salida	Se envía el correo con el fichero adjunto.
Criterio de aceptación	Llega el correo con el adjunto al destinatario.

Tabla 65 - PA-015

PA-016	
Requisito de usuario	UR-C006
Resumen requisito	Se tiene que poder enviar adjuntos.
Especificaciones de entrada	El usuario selecciona un fichero adjunto y envía el correo firmado.
Especificaciones de salida	Se envía el correo firmado con el fichero adjunto.
Criterio de aceptación	Llega el correo con el adjunto al destinatario y se verifica la firma.

Tabla 66 - PA-016

PA-017	
Requisito de usuario	UR-C006
Resumen requisito	Se tiene que poder enviar adjuntos.
Especificaciones de entrada	El usuario selecciona un fichero adjunto y envía el correo cifrado.
Especificaciones de salida	Se envía el correo cifrado con el fichero adjunto.
Criterio de aceptación	Llega el correo cifrado al destinatario, se descifra y se puede obtener el adjunto.

Tabla 67 - PA-017

PA-018	
Requisito de usuario	UR-C006
Resumen requisito	Se tiene que poder enviar adjuntos.
Especificaciones de entrada	El usuario selecciona un fichero adjunto y envía el correo firmado y cifrado.
Especificaciones de salida	Se envía el correo firmado y cifrado con el fichero adjunto.
Criterio de aceptación	Llega el correo cifrado y firmado al destinatario, se descifra, se verifica la firma y se puede obtener el adjunto.

Tabla 68 - PA-018

PA-019	
Requisito de usuario	UR-C007
Resumen requisito	Se tiene que poder recibir adjuntos.
Especificaciones de entrada	Se recibe un correo con un adjunto y se selecciona descargar.
Especificaciones de salida	Se descarga el adjunto en la tarjeta externa.
Criterio de aceptación	Se crea un nuevo fichero en la tarjeta externa con el nombre del fichero adjunto.

Tabla 69 - PA-019

PA-020	
Requisito de usuario	UR-C007
Resumen requisito	Se tiene que poder recibir adjuntos.
Especificaciones de entrada	Se recibe un correo firmado con un adjunto y se selecciona descargar.
Especificaciones de salida	Se descarga el adjunto en la tarjeta externa.
Criterio de aceptación	Se crea un nuevo fichero en la tarjeta externa con el nombre del fichero adjunto.

Tabla 70 - PA-020

PA-021	
Requisito de usuario	UR-C007
Resumen requisito	Se tiene que poder recibir adjuntos.
Especificaciones de entrada	Se recibe un correo cifrado con un adjunto y se selecciona descargar.
Especificaciones de salida	Se descifra el correo y se descarga el adjunto en la tarjeta externa.
Criterio de aceptación	Se crea un nuevo fichero en la tarjeta externa con el nombre del fichero adjunto.

Tabla 71 - PA-021

PA-022	
Requisito de usuario	UR-C007
Resumen requisito	Se tiene que poder recibir adjuntos.
Especificaciones de entrada	Se recibe un correo firmado y cifrado con un adjunto y se selecciona descargar.
Especificaciones de salida	Se descifra el correo y se descarga el adjunto en la tarjeta externa.
Criterio de aceptación	Se crea un nuevo fichero en la tarjeta externa con el nombre del fichero adjunto.

Tabla 72 - PA-022

PA-023	
Requisito de usuario	UR-C008
Resumen requisito	Se tiene que poder recibir correos electrónicos.
Especificaciones de entrada	Se recibe un correo.
Especificaciones de salida	Se guarda en la base de datos.
Criterio de aceptación	Aparece en el listado de correos recibidos y se puede visualizar en la vista detallada.

Tabla 73 - PA-023

PA-024	
Requisito de usuario	UR-C009
Resumen requisito	Se deberá poder descifrar los mensajes que se reciban cifrados.
Especificaciones de entrada	Se recibe un correo cifrado.
Especificaciones de salida	Se descifra y se guarda en la base de datos.
Criterio de aceptación	Aparece en el listado de correos recibidos y se puede visualizar en la vista detallada.

Tabla 74 - PA-024

PA-025	
Requisito de usuario	UR-C010
Resumen requisito	Se podrá validar la firma de los correos recibidos.
Especificaciones de entrada	Se recibe un correo firmado.
Especificaciones de salida	Se verifica la firma y se guarda en la base de datos.
Criterio de aceptación	Aparece en el listado de correos recibidos y se puede visualizar en la vista detallada, con la información de la verificación de la firma.

Tabla 75 - PA-025

PA-026	
Requisito de usuario	UR-C009 y UR-C010
Resumen requisito	Se podrá descifrar y validar la firma de los correos recibidos.
Especificaciones de entrada	Se recibe un correo firmado y cifrado.
Especificaciones de salida	Se descifra el correo y se verifica la firma y se guarda en la base de datos.
Criterio de aceptación	Aparece en el listado de correos recibidos y se puede visualizar en la vista detallada, con la información de la verificación de la firma.

Tabla 76 - PA-026

PA-027	
Requisito de usuario	UR-C011
Resumen requisito	Ha de poderse borrar la cuenta del usuario de forma segura.
Especificaciones de entrada	Se selecciona borrar la cuenta del usuario.
Especificaciones de salida	Se genera una contraseña aleatoria, se cifra toda la base de datos y se borra toda la base de datos.
Criterio de aceptación	Aparece la base de datos sin datos.

Tabla 77 - PA-027

7. Presupuesto

Como en todo trabajo se va a realizar el presupuesto del sistema creado, para ello se dividirá el presupuesto en tres partes, costes fijos, costes variables y costes indirectos.

7.1. Recursos materiales

Para realizar la parte de recursos materiales tendremos en cuenta el hardware y software utilizado. Para ello hemos tenido en cuenta que el material tiene un periodo de amortización por lo que solo incluimos en el presupuesto la parte proporcional al periodo de desarrollo, dicho periodo ha sido de cuatro meses y medio.

Elemento	Cantidad	Coste	Tiempo de uso	Periodo de amortización	Subtotal
Ordenador	1	700 €	4,5 meses	3 años	87,5 €
Microsoft Windows 7	1	0 € (Licencia de estudiante)	4,5 meses	-	0 €
Microsoft Office 2010	1	99 €	4,5 meses	-	99 €
Samsung Galaxy Nexus	1	500 €	4,5 meses	2 años	93,75 €
Visual Paradigm	1	0 € (Licencia de prueba)	4,5 meses	-	0 €
Total					280,25 €

Tabla 78 - Recursos materiales

7.2. Recursos humanos

En el apartado de recursos humanos se tendrán en cuenta todas las horas empleadas para cada una de las fases de desarrollo. Como se ha comentado anteriormente se ha trabajado unas 4 horas de media por día, por lo que obtenemos la siguiente tabla de con la distribución de las horas en cada fase de proyecto.

Fase	Precio/hora	Horas	Subtotal
Análisis del entorno	25 €	80	2000 €
Gestión del proyecto	25 €	8	200 €
Análisis del sistema	25 €	52	1300 €
Diseño del sistema	25 €	40	1000 €
Implementación	25 €	140	3500 €
Pruebas	25 €	20	500 €
Documentación	25 €	32	800 €
Total		372	9300 €

Tabla 79 - Recursos humanos

7.3. Costes indirectos

En lo referente a los costes indirectos es complicado establecer una lista de los costes indirectos aplicables al trabajo. Se intentará estimar mediante el gasto derivado de la cuota de internet y de la luz.

Concepto	Precio/mes	Tiempo de uso	Subtotal
Conexión a internet	48,20 €	4,5 meses	216,9 €
Luz	60 €	4,5 meses	270 €
Total			486,9

Tabla 80 - Costes indirectos

7.4. Presupuesto total

Para obtener el presupuesto total se realiza la suma de las dos partes descritas anteriormente, más el I.V.A. que se le aplica al proyecto. Obteniendo:

Recurso	Coste
Recursos materiales	280,25 €
Recursos humanos	9300 €
Costes indirectos	486,9 €
Subtotal	10067,15 €
I.V.A. (18%)	1812,09 €
Total	11879,24 €

Tabla 81 - Presupuesto total

8. Conclusiones

8.1. Conclusiones generales

Un este Trabajo Fin de Grado se ha llevado a cabo el análisis, diseño, e implementación de un prototipo funcional de un gestor de correos seguro para Android. Para ello se han implementado las funcionalidades básicas de un gestor de correo, como son enviar y recibir correos. Y además se ha implementado poder enviar y recibir correos firmados y/o cifrados. Todo ello se ha realizado para el sistema operativo Android, en su última versión (4.0).

Después de analizar la tecnología actual se decidió utilizar el estándar S/MIME para la implementación del correo seguro. También se ha tenido en consideración los aspectos de seguridad referentes al almacenamiento seguro de los datos, por lo que se decidió cifrar los datos privados del usuario con una clave, que se le solicita al usuario.

Por otro lado para poder almacenar de forma segura los certificados de clave privada se utilizó el almacén de claves proveído por Android, el cual sólo permite el acceso a aplicaciones que el usuario dé permiso explícito.

En lo referente al aprendizaje personal, puedo comentar que este trabajo de fin de grado me ha supuesto un gran reto de desarrollo, puesto que ha sido la primera vez que me enfrentaba a un proyecto de esta magnitud, y teniendo que desarrollar todas las fases (análisis, diseño...). También me ha servido para el aprendizaje del funcionamiento de los correos electrónicos y su variante segura.

Para terminar decir que queremos liberar el código de la aplicación con licencia libre, aunque aún tenemos que decidir que licencia elegir, para que quien quiera, pueda seguir desarrollando el gestor de correos. También se publicara la aplicación en Google Play para se pueda hacer uso de ella.

8.2. Trabajos futuros

En relación a los posibles trabajos futuros, podemos destacar los siguientes:

- Soporte para el protocolo POP3, en el proceso de análisis del sistema se decidió acotar el trabajo a que solo utilizara IMAP para la recepción de correos, por lo que sería interesante permitir el uso de POP3 puesto que es un protocolo ampliamente utilizado, dando así compatibilidad a muchos servicios de correo electrónico.

- Mejorar la vista de visualización del correo, debido a que en el desarrollo del proyecto nos hemos centrado en conseguir la funcionalidad y seguridad requerida se ha dejado un poco de lado la interfaz de la aplicación, quedando la parte de visualización del contenido del correo electrónico bastante básico. Por lo que sería una de las partes a desarrollar en trabajo futuros.

- Para poder hacer uso de la aplicación dentro de una organización. Se podría crear un sistema mediante el cual el dispositivo móvil se conectara a un servidor de la empresa y obtuviese los certificados de clave pública de los destinatarios. Por lo que mejoraría la usabilidad del sistema, evitando que el usuario tenga que descargarse manualmente con antelación los certificados públicos de los destinatarios.

- En la versión actual del prototipo desarrollado sólo se permite el uso de un certificado de clave pública para cada contacto, así que una mejora futura sería permitir tener varios certificados de distintas entidades para un mismo contacto.

- Otra mejora en referencia a la gestión de contactos, sería permitir la sincronización de los contactos con los contactos existentes en el dispositivo móvil.

- Para mejorar la usabilidad de la aplicación, se podría permitir la creación de certificados desde el dispositivo, y una vez creado poder enviarlo a una autoridad certificadora y que nos lo devolviera firmado.

9. Bibliografía

[1] Gartner Says Sales of Mobile Devices Grew 5.6 Percent in Third Quarter of 2011; Smartphone Sales Increased 42 Percent [en línea]
<http://www.gartner.com/it/page.jsp?id=1848514>[Consulta: Febrero, 2012].

[2] Mail; un buzón con muchas luces. [en línea]
<http://www.apple.com/es/iphone/built-in-apps/mail.html>[Consulta: Febrero, 2012].

[3] Internet Message Format [en línea]
<http://tools.ietf.org/html/rfc5322>[Consulta: Febrero, 2012].

[4] Simple Mail Transfer Protocol [en línea]
<http://tools.ietf.org/html/rfc5321>[Consulta: Febrero, 2012].

[5] Post Office Protocol-version 3 [en línea]
<http://tools.ietf.org/html/rfc1939>[Consulta: Febrero, 2012].

[6] Internet message access protocol - version 4rev1 [en línea].
<http://tools.ietf.org/html/rfc3501> [Consulta: Febrero, 2012].

[7] ASCII Format for Network Interchange [en línea]
<http://tools.ietf.org/html/rfc20>[Consulta: Febrero, 2012].

[8] Multipurpose Internet Mail Extensions(MIME) Part One: Format of Internet Message Bodies [en línea]
<http://tools.ietf.org/html/rfc2045 >[Consulta: Febrero, 2012].

[9] Multipurpose Internet Mail Extensions(MIME) Part Two: Media Types [en línea]
<http://tools.ietf.org/html/rfc2046>[Consulta: Febrero, 2012].

[10] MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text [en línea]
<http://tools.ietf.org/html/rfc2047>[Consulta: Febrero, 2012].

[11] Media Type Specifications and Registration Procedures [en línea]
<http://tools.ietf.org/html/rfc4288>[Consulta: Febrero, 2012].

[12] Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures [en línea]
<http://tools.ietf.org/html/rfc4289>[Consulta: Febrero, 2012].

[13] Multipurpose Internet Mail Extensions(MIME) Part Five: Conformance Criteria and Examples [en línea]
<http://tools.ietf.org/html/rfc2049>[Consulta: Febrero, 2012].

[14] Public-Key Cryptography Standards (PKCS) [en línea]
<http://www.rsa.com/rsalabs/node.asp?id=2124> [Consulta: Febrero, 2012].

[15] Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2; Message Specification [en línea]
<http://tools.ietf.org/html/rfc5751> [Consulta: Febrero, 2012].

[16] MIME Security with Pretty Good Privacy (PGP) [en línea]
<http://tools.ietf.org/html/rfc2015> [Consulta: Febrero, 2012].

[17] DJIGZO Email Encryption [en línea]
<https://play.google.com/store/apps/details?id=com.djigzo.android.application&hl=es> [Consulta: Febrero, 2012].

[18] Moxier Mail (Exchange)-Key [en línea]
<https://play.google.com/store/apps/details?id=com.emtrace.hermes.key&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5lbXRyYWNlLmhlcm1lcy5rZXkiXQ..>
[Consulta: Febrero, 2012].

[19] TouchDown for Smartphones [en línea]
<https://play.google.com/store/apps/details?id=com.nitrodesk.droid20.nitroid&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5uaXRyb2Rlc2suZHJvaWQyMC5uaXRyb2lkIl0.>[Consulta: Febrero, 2012].

[20]K-9 Mail [en línea]
<https://play.google.com/store/apps/details?id=com.fsck.k9&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5mc2NrLms5Il0.> [Consulta: Febrero, 2012].

[21] APG [en línea]
<https://play.google.com/store/apps/details?id=org.thialfihar.android.apg&feature=search_result#?t=W251bGwsMSwxLDEsIm9yZy50aGlhbGZpaGFyLmFuZHJvaWQuYXBnIl0.> [Consulta: Febrero, 2012].

[22] X509Tools [en línea]
<https://play.google.com/store/apps/details?id=at.rundquadrat.android.x509tools&feature=search_result#?t=W251bGwsMSwxLDEsImF0LnJ1bmRxdWFkcmF0LmFuZHJvaWQueDUwOXRvb2xzIl0.> [Consulta: Febrero, 2012].

[23] R2Mail2 [en línea]
<https://play.google.com/store/apps/details?id=at.rundquadrat.android.r2mail2&feature=search_result#?t=W251bGwsMSwyLDEsImF0LnJ1bmRxdWFkcmF0LmFuZHJvaWQucjJtYWlsMiJd>[Consulta: Febrero, 2012].

[24] Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal. [en línea]
<http://www.boe.es/buscar/doc.php?id=BOE-A-1999-23750> [Consulta: Febrero, 2012].

[25] PSS download page [en línea]
<http://www.esa.int/TEC/Software_engineering_and_standardisation/TECBUCUXBQE_2.html> [Consulta: Febrero, 2012].

[26] JavaMail API Release 1.4.1 [en línea]
<http://www.oracle.com/technetwork/java/javamail-1-4-1-141959.html> [Consulta: Febrero, 2012].

[27] The Legion of the Bouncy Castle [en línea]
<http://www.bouncycastle.org/>[Consulta: Febrero, 2012].

[28] Javamail-android [en línea]
<http://code.google.com/p/javamail-android/>[Consulta: Febrero, 2012].

[28] Spongy Castle [en línea]
<http://rtyley.github.com/spongycastle/>[Consulta: Febrero, 2012].

10. Anexo 1: Acrónimos y abreviaturas

AES: *"Advanced Encryption Standard".*

API: "*Application Programming Interface".*

GB: Giga Byte.

GHz: Giga Hercio.

CRL: "*Certificate Revocation List*"

CTR: "*Counter"*, modo de cifrado.

IMAP: *"Internet Message Access Protocol".*

I.V.A.: impuesto sobre el valor añadido.

MIME: "*Multipurpose Internet Mail Extensions".*

MVC: Modelo Vista Controlador .

PGP: "*Pretty Good Privacy".*

PKCS: *"Public-Key Cryptography Standards".*

PKI: "*Public Key Infrastructure* ".

POP3: "*Post Office Protocol version 3"*.

RAM: "*Random Access Memory"*.

RFC: "Request for Comments".

SMTP: *"Simple Mail Transfer Protocol".*

SSL: *"Secure Sockets Layer".*

S/MIME: *"Secure / Multipurpose Internet Mail Extensions"*

11. Anexo 2: Manual de usuario

En este manual se explicarán las funcionalidades básicas del gestor de correo implementado.

En primer lugar el usuario tiene que instalar la aplicación en el dispositivo, para ello bastará obtener el instalador de la aplicación (próximamente estará disponible en el Google Play) y proceder con la instalación como con cualquier otra aplicación de Android.

11.1. Creación de una cuenta

Una vez el usuario tiene instalado la aplicación, tiene que configurar una cuenta de usuario, para ello se le requerirá que indique:

- **Usuario de correo.**
- **Contraseña del correo.**
- **Dirección del servidor SMTP.**
- **Puerto del servidor SMTP.**
- **Seleccionar si hace falta autenticación.**
- **Seleccionar si necesita usar SSL o no para SMTP.**
- **Dirección del servidor IMAP**
- **Puerto del servidor IMAP**
- **Seleccionar si necesita usar SSL o no para IMAP**
- **Contraseña con la que cifrara la aplicación.**

Estos datos le serán requerido al usuario en las siguientes dos pantallas.

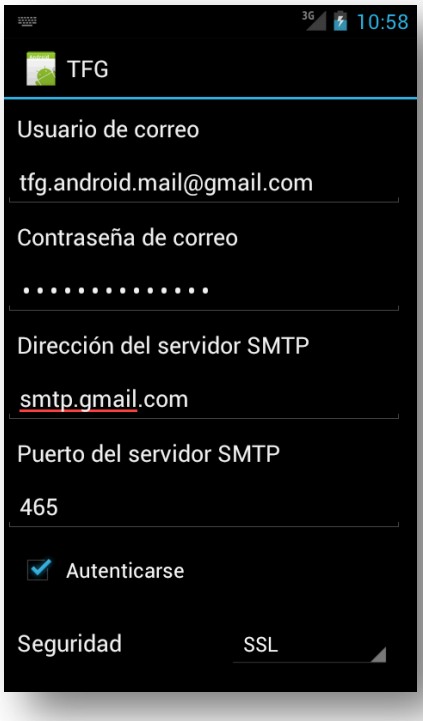

Ilustración 40 - Registro de la aplicación 1

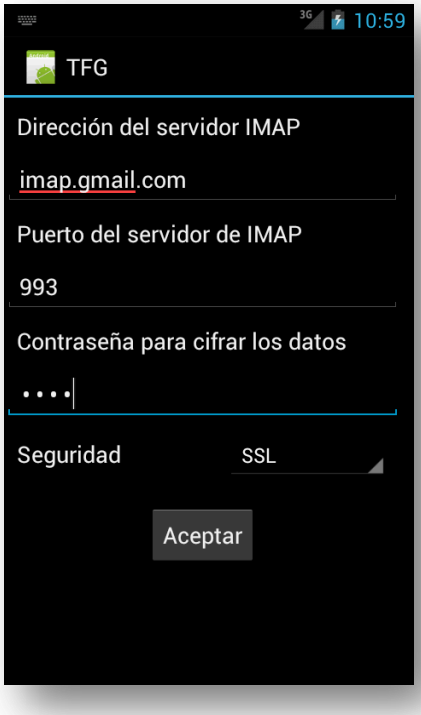

Ilustración 41 - Registro de la aplicación 2

Una vez que se verifique que los datos son correctos se le ofrecerá al usuario la opción de importarse una clave privada para utilizarla en la aplicación. Para ello tiene dos formas, importar el fichero PCKS#12 desde un directorio de la tarjeta de memoria externa, y después de importarse en el almacén de claves, se solicitara al usuario que dé permiso a la aplicación para utilizar la clave. Y la otra opción, solo si ya dispone de alguna clave privada en el almacén de claves, sería seleccionar la clave que quiera utilizar y darle así permisos de uso a la aplicación.

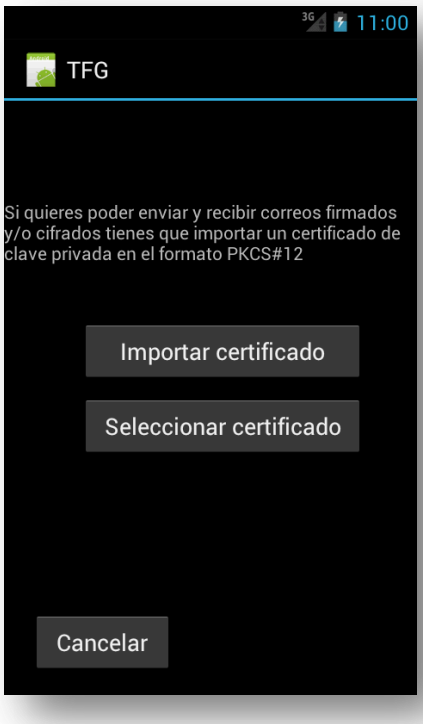

Ilustración 42 - Importar/Seleccionar clave privada

11.2. Pantalla principal

Después de esta última pantalla entraríamos a la aplicación y se nos mostrarían los últimos mensajes recibidos. Esta es la pantalla principal de nuestra aplicación, desde ella se puede visualizar correos, crear correos, acceder a las opciones de la aplicación… Lo primero que vamos a mostrar es como crear y enviar un correo, para ello seleccionamos el botón que aparece en la esquina inferior izquierda en la siguiente pantalla.

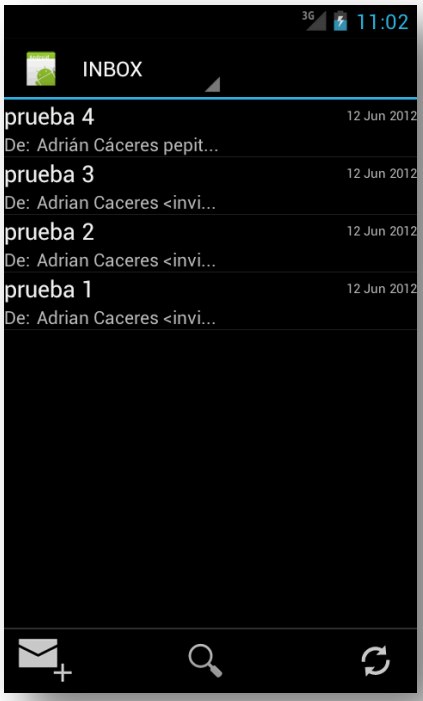

Ilustración 43 - Pantalla principal

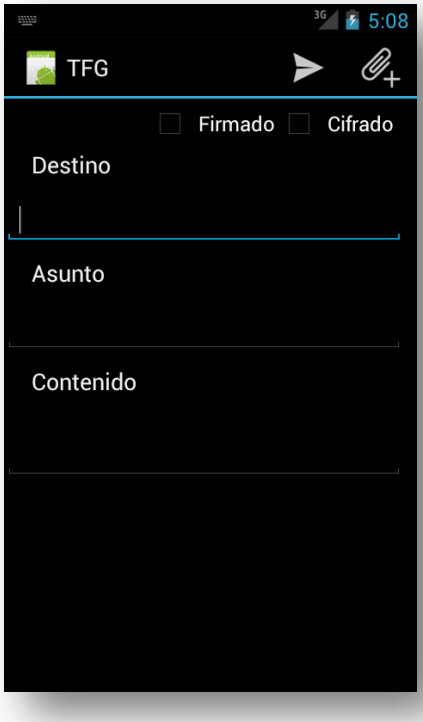

Ilustración 44 - Pantalla de creación de correos

11.3. Creación y envío de un correo

En la pantalla de creación de correos podemos ver que se pueden rellenar los campos necesarios para el envío de un correo electrónico, y también se puede marcar si queremos firmar y/o cifrar el correo (siempre que tengamos una clave privada para poder firmar, y el certificado de clave pública del destinatario para poder cifrar). Para adjuntar un documento seleccionamos el botón que aparece en la esquina superior derecha y nos mostrará el gestor de ficheros que tengamos instalado y ahí seleccionaríamos el documento que queramos adjuntar. Una vez hayamos rellenado todos los datos del correo, seleccionamos el botón situado a la izquierda de adjuntar y se procederá al envío del correo electrónico.

11.4. Visualización de un correo

Para poder ver el contenido de uno de los correos, tenemos estar en la vista principal y pulsar sobre el elemento de la lista que queramos visualizar, y nos aparecerá la siguiente pantalla.

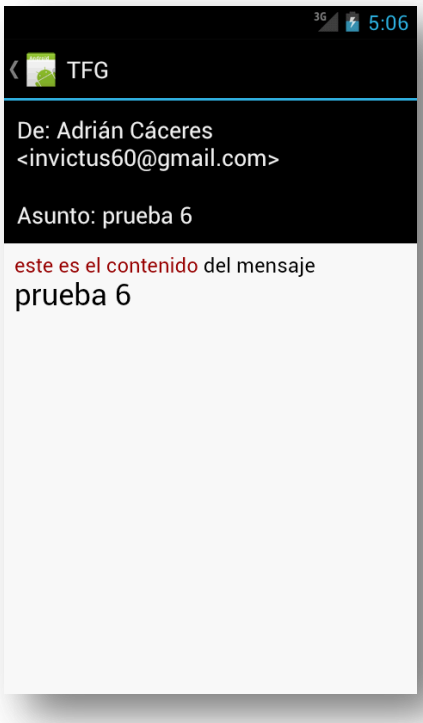

Ilustración 45 - Pantalla de visualización del contenido de un correo

En esta vista se visualizan los datos más relevantes de un correo, para ver los detalles del correo tendremos que pulsar en el botón menú del dispositivo móvil y seleccionar ver detalles.

11.5. Gestión de contactos y sus certificados

Para acceder al gestor de contactos tenemos que volver a la pantalla principal y apretar el botón menú, a continuación seleccionaremos gestión de contactos. En esta nueva pantalla se nos mostrara los contactos existentes. En dicha pantalla se puede ir a la pantalla de creación de contactos, o editar un contacto. Para ir a la creación de contactos seleccionaremos el icono que aparece en la esquina superior derecha de la siguiente imagen.

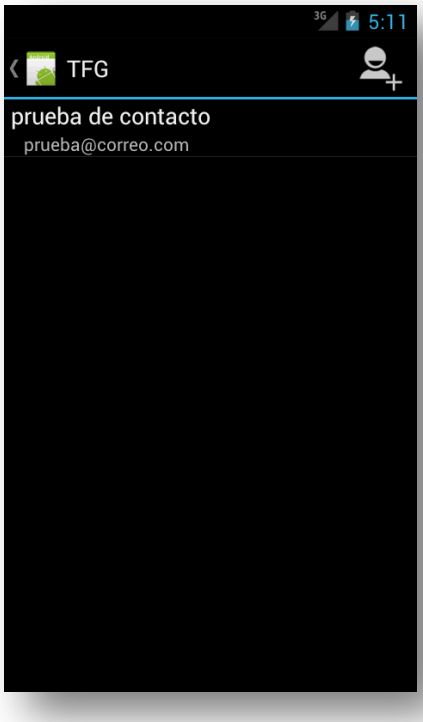

Ilustración 46 -- Gestor de contactos

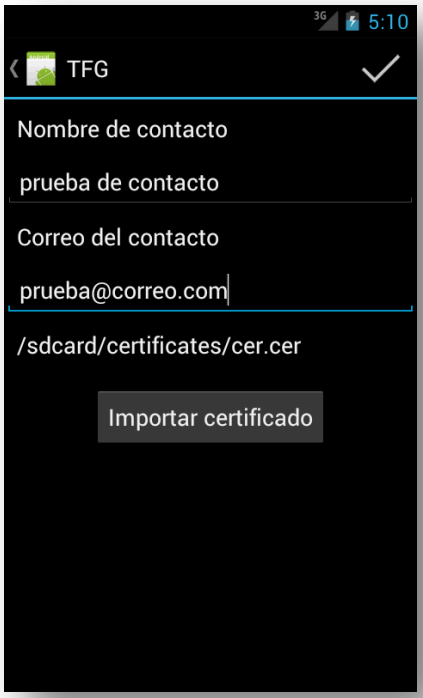

Ilustración 47 - Pantalla de creación y edición de contactos

Los datos que se pueden introducir en contactos son el nombre, el correo y el certificado de clave pública. Para incorporar el certificado de clave pública seleccionamos el botón de importar certificado y se nos abrirá nuestro gestor de ficheros, seleccionamos el certificado y se importa. Una vez terminado de editar el contacto hay que pulsar el botón de guardar (situado en la esquina superior derecha).