Verificación y gestión de facturas electrónicas en un teléfono Android

Nombre Estudiante: María José Moreno de Frutos

Programa: Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones (MISTIC)

Nombre Consultor: Jordi Castellà Roca y Alexandre Viejo Galicia

Centro: Universidad Rovira i Virgili

Fecha entrega: Junio 2014

FICHA DEL TRABAJO FINAL

Título del trabajo:	Verificación y gestión de facturas electrónicas en un teléfono móvil Android
Nombre del autor:	María José Moreno de Frutos
Nombre del consultor:	Jordi Castellà Roca y Alexandre Viejo Galicia
Fecha de entrega (mm/aaaa):	06/2014
Área del Trabajo Final:	Seguridad en aplicaciones web
Titulación:	Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones (MISTIC)

Resumen del Trabajo (máximo 250 palabras):

Las facturas en papel están siendo sustituidas por facturas electrónicas, por motivos económicos y de eficiencia. Aunque existen distintos formatos de factura electrónica admitidos legalmente todos ellos deben garantizar la autenticidad de su origen y la integridad de su contenido, lo que se consigue mediante la firma electrónica.

En España, el formato XML facturae [3] es el formato obligatorio para las facturas cuyo destinatario es un organismo de la Administración General del Estado (AGE). Este formato también se emplea frecuentemente entre empresas privadas y además ha sido adoptado por algunas Comunidades Autónomas.

El trabajo consiste en el desarrollo de una aplicación para los teléfonos Android que ayude a los usuarios a gestionar sus facturas electrónicas en formato facturae. Estas facturas son documentos XML firmados electrónicamente según formato XMLDSig ENVELOPED con extensiones XADES-EPES, por lo que es necesario validar que se ajustan al schema XML definido y que la firma electrónica es correcta, garantizando su autenticidad e integridad.

La aplicación desarrollada valida y almacena en una base de datos la información de las facturas electrónicas cargadas en el teléfono móvil mediante Bluetooth. Las facturas son clasificadas según categorías definidas por el usuario y se pueden realizar consultas del consumo total para cada una de dichas categorías y por suministrador. También se ofrece al usuario la posibilidad de realizar consultas por emisor, fecha y categoría.

Asimismo la aplicación permite reenviar por correo electrónico las facturas electrónicas almacenadas y realizar copias de seguridad de la información en un ordenador.

Abstract (in English, 250 words or less):

Paper invoices are being replaced by electronic invoices, for economic and efficiency issues. Although there are different formats of electronic invoice legally admitted, they all must ensure the authenticity of the origin and integrity of their content, which is achieved by an electronic signature.

In Spain, after the publication of Order PRE/2971/2007 [3], facturae XML format became the mandatory format for invoices whose recipient is an agency of the State General Administration (AGE). This format is also often used between private companies and has also been adopted by some Autonomous Communities (Catalonia, Basque Country, Valencian Community, La Rioja, etc.).

The work consists in developing an application for Android phones, which helps users manage the facturae format of their electronic invoices. These invoices are electronically signed documents as XML format extensions XMLDSig enveloped XADES-EPES (this needs to be revised), so it is necessary to check they adjust to the XML schema defined and that the electronic signature is correct, thus guaranteeing its authenticity and integrity.

A developed application validates and stores in a database information of electronic invoices loaded into the mobile phone via Bluetooth. Invoices are classified according to user-defined categories and you can query the total consumption for each of these categories and by supplier (this needs to be revised). It also offers the user the possibility to make queries by issuer, date and category.

The application can also send emails with the stored electronic invoices and back up information on a computer.

Palabras clave (entre 4 y 8):

Factura electrónica, firma electrónica, facturae, Android

Agradecimientos

Desearía agradecer el apoyo y ayuda que me ha proporcionado mi Tutor durante el desarrollo del trabajo, corrigiendo mis errores, dándome ideas y aportando soluciones cuando lo he necesitado.

También desearía agradecer a mi marido su ánimo en los momentos difíciles y el que siempre esté a mi lado, respetando mis prioridades, aunque no siempre las comparta.

Índice

Lista de figuras

Lista de tablas

1. Introducción

1.1. Contexto y justificación del Trabajo

Con el fin de ganar en eficiencia y ahorrar costes las empresas están sustituyendo las facturas en papel por facturas electrónicas. Según el artículo 1 de la Ley 56/2007 [1] una factura electrónica es "un documento electrónico que cumple con los requisitos legal y reglamentariamente exigibles a las facturas y que, además, garantiza la autenticidad de su origen y la integridad de su contenido, lo que impide el repudio de la factura por su emisor". En España, la firma electrónica es el mecanismo más generalizado para garantizar la autenticidad de origen y la integridad del contenido de la factura electrónica ([2]).

Las facturas electrónicas se pueden emitir en distintos formatos (doc, PDF, EDIFACT, etc.), siempre que se respeten los aspectos señalados en el párrafo anterior. No obstante, tras la publicación de la Orden PRE/2971/2007 [3] el formato XML facturae se convirtió en el formato obligatorio para las facturas cuyo destinatario fuera uno de los organismos de la Administración General del Estado (AGE). Este formato también se emplea frecuentemente entre empresas privadas y además ha sido adoptado por algunas Comunidades Autónomas (Cataluña, País Vasco, Comunidad Valenciana, La Rioja, etc.). Estas facturas electrónicas son documentos XML firmados electrónicamente según formato XMLDSig ENVELOPED con extensiones XADES-EPES.

La facturación electrónica, que inicialmente era utilizada por la Administración y por las grandes empresas, se va imponiendo y empieza a sustituir a la facturación en papel en todos los ámbitos. La mayoría de las empresas de cierto tamaño ofrecen a sus clientes la posibilidad de emisión de facturas electrónicas.

No obstante, las facturas electrónicas no se pueden validar sin un ordenador o dispositivo que cuente con la aplicación adecuada. En este sentido los Smart-phone ofrecen muchas posibilidades.

En el mundo actual el teléfono móvil nos proporciona continuamente nuevas funcionalidades: fotografía, música, Internet, GPS. Una funcionalidad reciente es la posibilidad de realizar pagos, como si se tratara de una tarjeta. Sólo como ejemplo, el BBVA proporciona una App para los teléfonos móviles Android e iOS denominada BBVA wallet [5] que funciona como una tarjeta virtual prepago.

Figura 1: BBVA wallet [5]

Como contrapartida a los pagos realzados con el teléfono recibiremos los tiques o facturas electrónicas, que deberán cumplir los requisitos

legales. Y dado que el pago se ha realizado con el móvil, deberíamos poder recibir la factura o tique también en el teléfono móvil.

1.2. Objetivos del Trabajo

El objetivo de este trabajo de fin de máster es el desarrollo de una aplicación para teléfonos con el S.O. Android que permita la gestión de facturas electrónicas en el formato facturae. Dichas facturas se cargarán en el teléfono mediante Bluetooth.

La aplicación deberá validar la factura electrónica, operación que consiste por un lado en confirmar que se ajusta al schema XML definido y por otro en verificar que la firma electrónica es correcta y el documento no ha sido modificado con posterioridad a su emisión. De esta forma se valida su autenticidad e integridad.

Los datos de las facturas se almacenarán en una base de datos en el teléfono y se permitirá al usuario definir categorías para clasificarlas. Se desarrollará un módulo que permita la gestión de estas categorías.

La aplicación ofrecerá la opción de mostrar datos sobre el consumo total para cada una de las categorías definidas o para cada uno de los suministradores. Asimismo, el usuario podrá realizar búsquedas de facturas por, emisor, fecha y categoría.

Se facilitará la posibilidad del envío de una factura por correo electrónico al destino que se indique y de hacer copias de seguridad de la información en un ordenador, permitiendo así cargarla en otro terminal si se estropea el actual o se desee cambiarlo.

Se pretende desarrollar una aplicación amigable y de fácil uso.

1.3. Enfoque y método seguido

Se ha decidido desarrollar un nuevo producto, aunque para ello se emplearán librerías existentes, especialmente para la validación de las facturas electrónicas, la comunicación Bluetooth o la presentación de gráficos.

1.4. Planificación del Trabajo

El desarrollo del trabajo se divide en las siguientes tareas:

T0: instalar el entorno de desarrollo, definición de los objetivos y fijar una planificación.

T1: crear el juego de pruebas que permita asegurar el correcto funcionamiento de la aplicación. Esta tarea consiste en descargar el programa Facturae [4] y crear facturas para hacer las pruebas. Se creará

un juego suficientemente grande y variado para poder probar toda la funcionalidad de la aplicación.

T2: desarrollar la aplicación que envía una factura a un teléfono móvil. Esta aplicación se ejecuta en un ordenador.

T3: desarrollar una aplicación Android que sea capaz de recibir una factura enviada por la aplicación desarrollada en la tarea T2.

T4: desarrollo del módulo para validar la factura recibida (validación del schema y la firma electrónica). Este módulo se incorpora al desarrollado realizado en la tarea T3.

T5: desarrollar el módulo de persistencia en la App. Es decir, se implementará el módulo que permite gestionar las categorías para clasificar las facturas y guardarlas en la base de datos.

T6: desarrollar el módulo de gestión y de control de los gastos.

T7: desarrollar el módulo de consultas (buscar facturas).

T8: desarrollar las funciones de copia de seguridad y envío de las facturas por correo electrónico.

T9: documentación (tarea incremental a lo largo de todo el desarrollo).

T10: preparar la presentación.

T11: Gestión del proyecto (tarea a realizar a lo largo de todo el trabajo)

En la *Figura 2* se muestra el diagrama de Gantt con la planificación en el tiempo de las tareas descritas anteriormente.

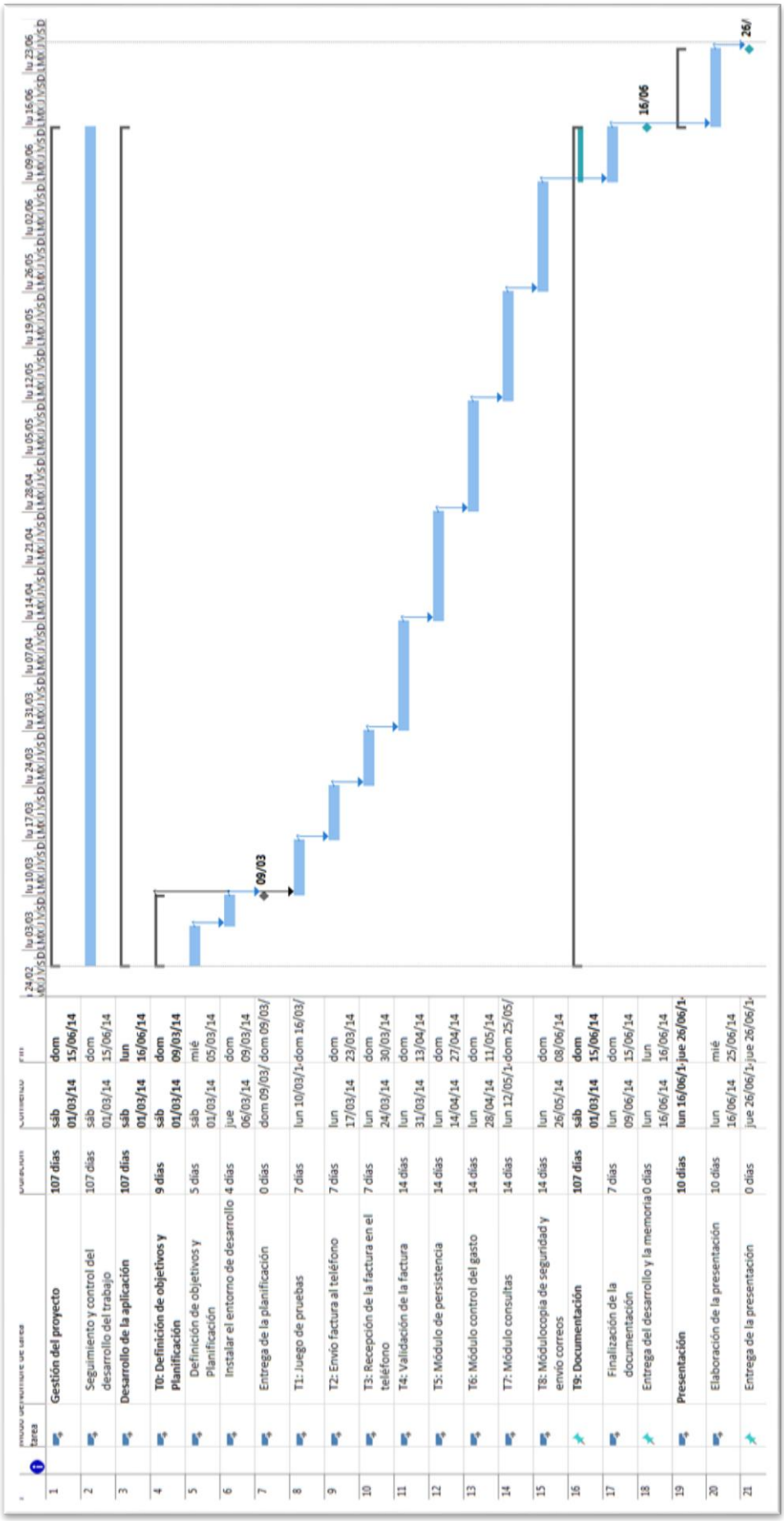

Figura 2: Diagrama de Gannt

1.5. Breve sumario de productos obtenidos

El trabajo ha consistido en el desarrollo de los siguientes productos:

- Aplicación Java para el envío de facturas electrónicas al teléfono móvil mediante Bluetooth.
- Aplicación para el teléfono Android que recibe, valida y gestiona las facturas, permitiendo un control del gasto.
- Aplicación Java para realizar copias de seguridad en un ordenador mediante Bluetooth.

1.6. Breve descripción de los otros capítulos de la memoria

La memoria se estructura en los siguientes apartados:

- En el apartado 1 se hace una introducción al trabajo señalando los objetivos, la motivación para la realización de mismo y una planificación de las tareas que lo comprenden.
- En el apartado 2 se detallan las tecnologías empleadas en el desarrollo.
- El apartado 3 se dedica a mostrar aspectos de la arquitectura y el diseño de la solución.
- En el apartado 4 se señalan los aspectos más relevantes de la fase de desarrollo de la aplicación.
- En el apartado 5 se muestra el proceso seguido para conseguir el juego de pruebas y la ejecución del mismo.
- Finalmente el apartado 6 se dedica a las conclusiones, resultados obtenidos y opinión personal.

2. Tecnologías utilizadas

En este apartado se describe las tecnologías empleado para el desarrollo de la aplicación.

2.1. Android

Se ha decidido realizar el desarrollo para esta plataforma dado que es el sistema operativo (S.O.) más extendido entre los teléfono móviles inteligentes. Durante 2013, un 78,6% de los smartphone comercializados en el planeta utilizaba este S.O. [20].

Android es una plataforma de software de código abierto creada para teléfonos móviles. Se trata de un proyecto de Google realizado en colaboración con la Open Handset Alliance. Incluye un sistema operativo basado en Linux, una interfaz de usuario, aplicaciones, bibliotecas de código, compatibilidad multimedia, entre otros elementos ([6]).

Android se comercializa bajo dos licencias de código abierto. El núcleo Linux se comercializa bajo la licencia General Public License (GPL) y la plataforma Android, sin el núcleo, tiene una licencia Apache Software License (ASL).

Los principales componentes de Android son:

- Un núcleo Linux.
- Bibliotecas de código: navegador WebKit, base de datos SQLite, funciones SSL del proyecto Apache, etc.
- Diferentes administradores de servicios: actividades y vistas, telefonía, ventanas, recursos, servicios basados en ubicación, etc.
- El entorno de ejecución de Android proporciona lo siguiente:
    - Paquetes Java para obtener un entorno de programación Java (no es un entorno J2ME)
    - La máquina virtual Dalvik para proporcionar un entorno de alojamiento para las aplicaciones Android.

2.2. Bluetooth

Para la comunicación entre el servidor de facturas y el teléfono móvil se ha decidido emplear Bluetooth pues hoy en día está disponible en todos los teléfonos móviles. Otras tecnologías como NFC aún no están tan extendidas.

Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la

banda ISM de los 2,4 GHz [16]. Se trata de un protocolo para comunicaciones locales sin cable.

La distancia máxima que admite el sistema Bluetooth a la hora de intercambiar información se clasifica en tres clases [17]:

- Clase 1: Alcance máximo 100 metros
- Clase 2: Alcance máximo 25 metros
- Clase 3: Alcance máximo 1 metro

Otra forma de clasificar los dispositivos Bluetooth depende del ancho de banda del que dispongan para intercambiar información:

- Versión 1.2: Ancho de banda máximo de 1Mbit/s
- Versión 2.0 EDR: Ancho de banda máximo de 3Mbit/s
- Versión 3.0 HS: Ancho de banda máximo 24Mbit/s

Para la implementación de la comunicación Bluetooth en el servidor emisor de facturas se emplea la librería Bluecove por tratarse de una librería de código abierto, ampliamente utilizada, que se distribuye bajo licencia Apache License, versión 2.0.

2.2.1. BlueCove

BlueCove es una librería Java para Bluetooth (implementación de API JSR-82) que actualmente funciona con Mac OS X, WIDCOMM, BlueSoleil y la pila Bluetooth de Microsoft (Microsoft Bluetooth stack) que se encuentra en Windows XP SP2, Windows Vista o Windows 7 y WIDCOMM y la pila Bluetooth de Microsoft para Windows Mobile [18].

2.3. Componentes de firma proporcionados por MINETUR

Los componentes de firma son una iniciativa del Ministerio de Industria, Energía y Turismo (MINETUR) para potenciar el uso de la firma digital en la sociedad de la información. Consiste en una serie de librerías desarrolladas en Java que proporcionan funcionalidad de firma digital [19]. Incluye las siguientes funcionalidades:

- Acceso a almacenes de certificados
- Cifrado y descifrado de datos
- Consultas OCSP
- Petición de sellos de tiempo
- Validación de sellos de tiempo
- Firma XAdES
- Validaciones de firmas XAdES

Los componentes se desglosan en los siguientes:

- MITyCLibAPI.- Sirve de base para el resto de componentes de firma del Ministerio de Industria, Turismo y Comercio (MITyC).
- MITyCLibCerts.- Permite el acceso a almacenes de certificados.
- MITyCLibCrypt.- Proporciona la funcionalidad necesaria para cifrar y descifrar datos de diversas maneras.
- MITyCLibOCSP.- Permite la consulta del estado de los certificados mediante el protocolo OCSP (Online Certificate Status Protocol).
- MITyCLibTSA.- Contiene la lógica necesaria para obtener y manejar sellos de tiempo de autoridades de sellado a través de la implementación del apéndice A.1 de la especificación RFC 2560 para tratar las peticiones y obtener dichos sellos de tiempo vía http teniendo en cuenta las normas descritas en el punto 3.4 de la especificación RFC 3161.
- MITyCLibTrust.- Proporciona implementaciones concretas de validadores de confianza, de acuerdo con las interfaces definidas en MITyCLibAPI.
- MITyCLibPolicy.- Implementa una serie de políticas XAdES. Actualmente se proporcionan las políticas de FacturaE 3.0, de FacturaE 3.1, política de no transformadas y la política propia del MITyC.
- MITyCLibXAdES.- Proporciona la funcionalidad necesaria para el manejo de ficheros con metadatos tipo XML, uso de certificados, y para la construcción de firmas electrónicas XAdES, así como para su validación. Todas las firmas generadas son de acuerdo a las especificaciones definidas por el ETSI (European Telecommunications Standards Institute).

2.4. XML

Las facturas electrónicas con formato facturae son documentos XML. Para la lectura e interpretación de las facturas se han empleado las librerías XML incorporadas en Android (paquetes *javax.xml.parsers*, *javax.xml.xpath* y *org.w3c.dom*). Pero para poder validar que se ajustan al schema XML definido para la versión 3.2 de facturae se ha empleado la librería Xerces de Apache, en su versión adaptada para Android [22]. Se ha empleado esta librería por ser de código abierto, ampliamente utilizadas y estar adaptada para Android.

2.5. SQLite

SQLite [31] es un gestor de base de datos relacional, compatible ACID, de dominio público. Entre sus características se puede señalar que es de pequeño tamaño, de código abierto, no requiere servidor, prácticamente no necesita configuración y es transaccional.

SQLite es una elección ampliamente extendida como motor de base de datos de teléfonos móviles, PDAs, reproductores de MP3 y otros aparatos electrónicos debido a su pequeño tamaño, a su eficiente uso de la memoria y espacio en disco, a que no requiere mantenimiento por parte de un administrador de base de datos y a que es altamente confiable.

Android incorpora todas las herramientas necesarias para crear y gestionar bases de datos SQLite.

3. Arquitectura y diseño

En este apartado se describen los aspectos relevantes de la arquitectura y el diseño de la aplicación.

3.1. Requisitos

La aplicación cumplirá los requisitos señalados en la *Tabla 1*.

Tabla 1: Requisitos de la aplicación

Código	Descripción
RQ-01	La comunicación entre el servidor de facturas y el teléfono móvil se realizará empleando Bluetooth.
RQ-02	El servidor y el teléfono deberán intercambiar el mismo pin para establecer la conexión.
RQ-03	El servidor de facturas realizará la búsqueda de dispositivos y seleccionará al que desea enviar la factura.
RQ-04	La aplicación instalada en el teléfono móvil tendrá una opción para recibir facturas que le permitirá recibir la factura enviada por el servidor.
RQ-05	La factura recibida se validará analizando dos aspectos: schema XML y firma electrónica.
RQ-06	Sólo se aceptarán facturas válidas.
RQ-07	Una vez recibida la factura en el teléfono móvil y validada, se ofrecerá al usuario la posibilidad de almacenarla o no.
RQ-08	El usuario podrá definir categorías para clasificar las facturas.
RQ-09	Se ofrecerá una utilizad de control del gasto que permita consultar el gasto total efectuado por categoría o por proveedor.
RQ-10	Se podrán realizar búsquedas entre las facturas por proveedor (nombre y NIF), fecha y categoría.
RQ-11	La aplicación tendrá una utilizada para enviar las facturas por correo electrónico.
RQ-12	Se podrán realizar copias de seguridad de la información gestionada por la aplicación.
RQ-13	La aplicación estará preparada para traducir el interfaz de usuario a cualquier idioma de forma fácil.
RQ-14	La aplicación permitirá al usuario definir un límite de gasto mensual para cada categoría.
RQ-15	Cuando se supere el límite de gasto fijado para una categoría, se mostrará un aviso.

3.2. Casos de uso

En la *Figura 3* se muestran los casos de uso de la aplicación, que se describirán en los siguientes apartados. En la *Figura 4* se muestran los casos de uso de la aplicación que enviará las facturas al teléfono.

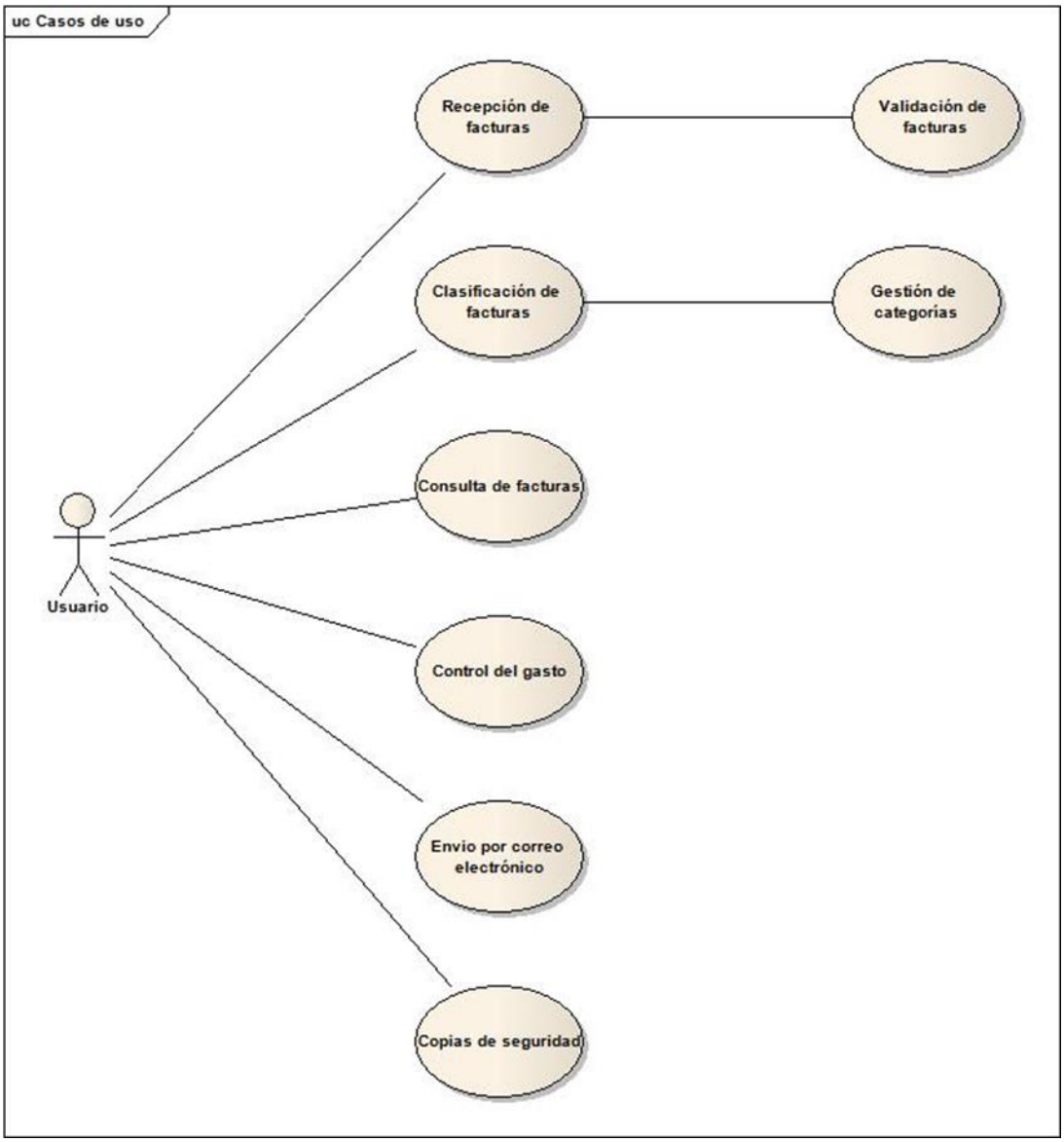

Figura 3: Casos de uso – teléfono móvil

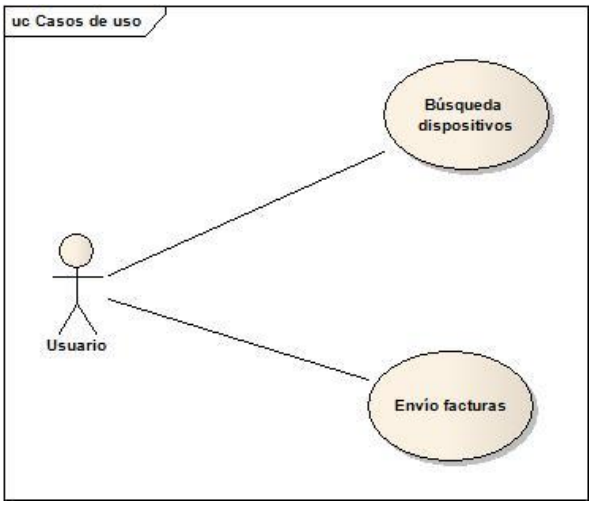

Figura 4: Casos de uso – servidor de facturas

3.3. Envío de facturas

Este módulo permite el envío de facturas a un teléfono móvil mediante Bluetooth. En la comunicación Bluetooth uno de los dispositivos que intervienen realiza la función de servidor, permaneciendo a la espera de que un cliente solicite una conexión y otro de los dispositivos actúa como cliente, buscando el servicio deseado en los dispositivos disponibles y estableciendo la conexión. Una vez establecida la conexión, la transferencia de información se puede realizar en los dos sentidos.

En esta aplicación se ha decidido que el servidor de facturas implemente la funcionalidad del cliente. Esto le permite seleccionar el dispositivo con el que desea establecer la comunicación. Las facturas electrónicas pueden tener datos protegidos por la Ley Orgánica de Protección de Datos (LOPD) y se ha pensado que es conveniente que sea en esta aplicación la que dirija el establecimiento de la conexión, confirmando el destinatario.

El procedimiento a seguir será:

- Buscar los dispositivos accesibles.
- Seleccionar el dispositivo destinatario.
- Seleccionar el fichero que se desea enviar.
- Realizar el envío.

Si la aplicación del teléfono móvil no se encuentra preparada para recibir la factura, el emisor recibirá un mensaje indicándolo.

Por motivos de seguridad, antes de poder establecerse la conexión, el emisor y el destinario deberán intercambiar un pin.

En el diagrama de secuencia de la *Figura 6* se muestra el protocolo seguido para el envío de las facturas.

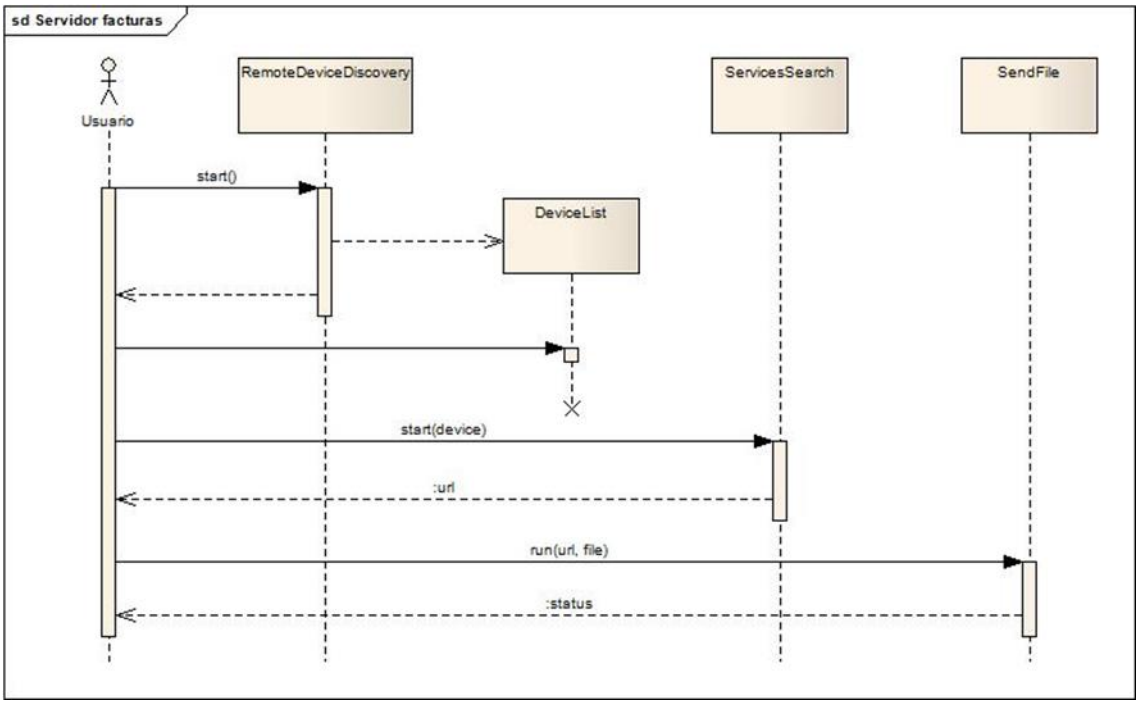

Figura 6: Diagrama de secuencia del envío de facturas

3.4. Recepción de las facturas

Este módulo permite la recepción de facturas en el teléfono móvil con S.O. Android utilizando Bluetooth. Como se ha indicado en el apartado anterior, en la comunicación Bluetooth uno de los dispositivos hace de servidor, permaneciendo a la espera de que un cliente abra una conexión, y en este caso es el teléfono el que realiza esta función. Cuando el usuario desea recibir una factura activa el servicio mediante una opción de menú. El servicio se lanza en un hilo independiente que el usuario puede cancelar en cualquier momento, si así lo desea. Para evitar dejar hilos no controlados en ejecución, cuando el usuario abandona la pantalla (actividad) que realiza el seguimiento de la recepción se finaliza la ejecución del hilo.

Si el servidor de facturas (cliente Bluetooth) consigue abrir la conexión con el teléfono, le enviará la factura y si la recepción es correcta éste le devuelve un mensaje indicándolo.

La factura recibida se almacena temporalmente en la memoria interna del teléfono, en la carpeta correspondiente al paquete de la aplicación:

```
/data/data/edu.mistic.tfm.einvoice/files
```

Tras la validación de la factura el usuario decidirá si conservarla o eliminarla.

Para que la aplicación pueda utilizar Bluetooth es necesario que en fichero *AndroidManifest.xml* se incluyan los siguientes permisos:

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

3.5. Validación de las facturas

La validación de las facturas consta de dos procesos: validación del schema XML y validación de la firma electrónica.

La validación del schema XML consiste en confirmar que la factura recibida se ajusta al schema definido para las facturas con el formato facturae v.3.2, es decir se confirma que la factura tiene la estructura esperada. El fichero *xsd* que define el schema de facturae v.3.2 se incorpora a la aplicación en la carpeta res/raw. Como este fichero importa otro schema (el correspondiente a la firma electrónica) es necesario para que funcione la validación que la aplicación tenga acceso a Internet. Para ello se deberá incluir en el fichero *AndroidManifest.xml* el siguiente permiso:

```
<uses-permission android:name="android.permission.INTERNET" />
```

Si en el proceso de validación se detecta que la factura recibida no se ajusta al schema se informará al usuario y se eliminará, no continuando con el resto del proceso de validación.

Las facturas electrónicas con formato facturae incluyen firmas electrónicas que se deben comprobar para asegurar su validez. El formato de firma que se utiliza es XMLDSig ENVELOPED con extensiones XADES-EPES. Se trata de una firma XML avanzada que incluye datos sobre la política a la que debe ajustarse. Una política define las características específicas que deben cumplir unas firmas determinadas. La política de la facturae, para las versiones 3.1 y 3.2, se puede consultar en [25]. Las firmas XADES-EPES incluyen un nodo *SignaturePolicyIdentifier*, que en el caso de la facturae contiene la siguiente información:

```
<etsi:SignaturePolicyIdentifier>
<etsi:SignaturePolicyId>
        <etsi:SigPolicyId>
                <etsi:Identifier>http://www.facturae.es/politica_de_firma_formato_facturae/politi
                ca_de_firma_formato_facturae_v3_1.pdf</etsi:Identifier>
                <etsi:Description>Política de Firma FacturaE v3.1</etsi:Description>
        </etsi:SigPolicyId>
        <etsi:SigPolicyHash>
                <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
```

```
                <ds:DigestValue>Ohixl6upD6av8N7pEvDABhEL6hM=</ds:DigestValue>
            </etsi:SigPolicyHash>
        </etsi:SignaturePolicyId>
        </etsi:SignaturePolicyIdentifier>
```

Por tratarse de firmas avanzadas basadas en política expresa (XADES-EPES), además de las validaciones que se deben realizar a cualquier firma electrónica (que la firma es correcta, que la información firmada no ha sido modificada después de la firma, que el certificado empelado es correcto y está dentro de su periodo de validez, etc.), es necesario comprobar que la firma se ajusta a la política (que incluye el elemento *SignaturePolicyIdentifier* con los datos señalados, que es un afirma XADES enveloped correctamente formada, que el certificado está emitido por una de las Autoridades de Certificación de confianza según la Agencia Tributaria [26], etc.).

En resumen, el proceso de validación de la firma electrónica consta de las siguientes comprobaciones:

- La firma es correcta, es decir, no se han modificado los datos firmados con posterioridad a la firma.
- La firma se ajusta a la política de facturae, es decir:
    - Se trata de una firma XADES-EPES enveloped, y se ajusta a una de las siguientes versiones del schema XML de las firmas XADES: 1.2.2 [27] o 1.3.2 [28].
    - En el nodo *SignaturePolicyIdentifier* se especifica la política de facturae, como se ha señalado anteriormente.
    - El nodo *KeyInfo* contiene, al menos, el certificado empleado en la firma codificado en base64.
    - Si existe el nodo *SignerRoles*, este se debe ajustar a lo especificado en la política.
    - El certificado ha sido emitido por una Autoridad de Certificación de Confianza [26].
    - Si tiene sello de tiempo, éste debe haber sido emitido por una Autoridad de Sellado de Tiempo (TSA) de confianza.
- El certificado empleado para la firma se encuentra en su periodo de validez (caducidad).
- El certificado no ha sido revocado.

Durante el periodo de validez de un certificado este se puede revocar y dejaría de ser válido. Para confirmar si un certificado ha sido revocado es necesario consultar a la Autoridad de Certificación que lo emitió. Existen dos formas de consultar el estado de revocación de un certificado: las Listas de Revocación de Certificados (CRL) y empleando el Protocolo On-

line de estado de revocación de certificados (OCSP). Las CRL son documentos que contienen información sobre los certificados revocados, fecha de revocación y causa, identificándolos por su número de serie. Las Autoridades de Certificación (AC) disponen de servidores que mediante el protocolo OCSP responden del estado de revocación de un certificado determinado. En la aplicación se empleará este segundo método y para ello es necesario identificar los servidores que ofrecen el servicio OCSP para cada AC. La aplicación almacena en el fichero *OCSPServersInfo.xml* esta información. Para que sirva de ejemplo, a continuación se muestra la información almacenada en relación con los certificados del DNI electrónico:

```
<tns:proveedor nombre="DNI electronico" descripcion="DNI electronico">
        <!-- 002 -->
        <tns:ca nameHash="5454af8b4ad0b423813e6a27fd3bd7e9f5490dcc"
                pkHash="3aa689ec15e8246471e0257ec9b1623107e906a2" />
        <!-- 001 -->
        <tns:ca nameHash="d5355fd02a340cbe53962337fca8ea2a0000423b"
                pkHash="1a89a8c5ee8f765d557189f33b35bdaa0500956f" />
        <!-- 003 -->
        <tns:ca nameHash="3d998a6095c925bf8c5932ccf08424d98c7f057a"
                pkHash="039543407488d46156854e9bba7c4054f04fa2ee" />
        <tns:servidorOCSP URI="http://ocsp.dnie.es"
                        descripcion="Servidor OCSP DNIe" />
</tns:proveedor>
```

Cuando se validan los certificados lo correcto es comprobar todos los certificados de la cadena de certificación, es decir, el certificado final, el certificado de su AC emisora, el certificado de la AC emisora de ésta y así hasta llegar a la AC raíz. No obstante, en la aplicación sólo se ha abordado la validación del certificado final (el empleado en la firma), dejando la validación completa como una posible mejora.

Como se ha señalado, sólo se admiten los certificados emitidos por una AC reconocida por el MINETUR (Organismo que tiene la competencia). La aplicación almacena en su carpeta *assets/trust* (Figura 7) los certificados de las AC válidas, así como los certificados válidos para las firmas de las respuestas OCSP y los sellos de tiempo. Esta información no está totalmente actualizada, lo que se plantea como una posible mejora de la aplicación.

Una vez finalizada la validación, si el fichero recibido es correcto se muestra al usuario su contenido, formateado, y se le da la opción de aceptarlo e incorporarlo a su gestor de facturas o eliminarlo.

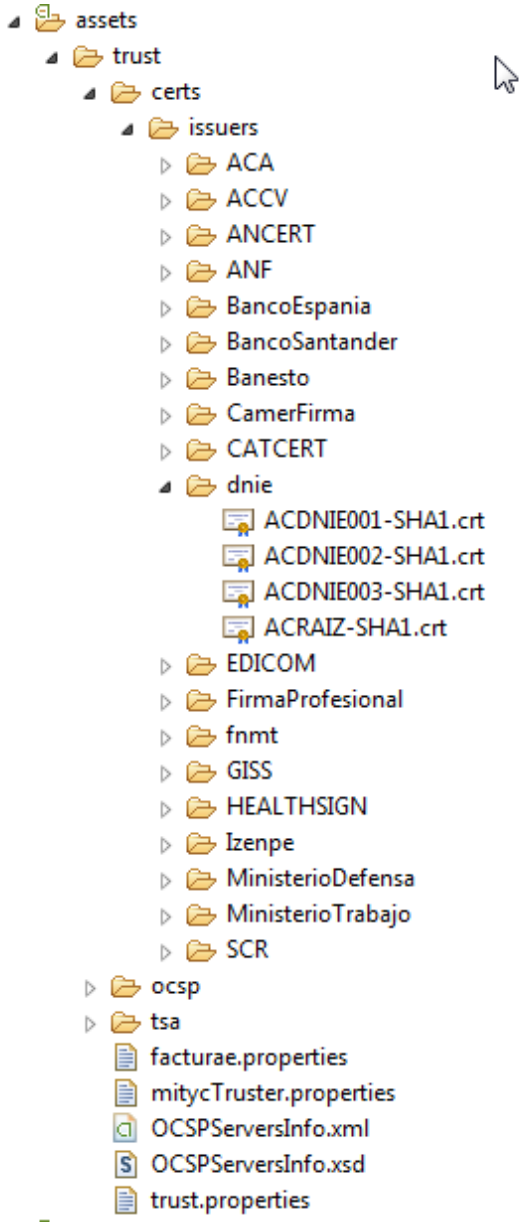

Figura 7: Autoridades de certificación válidas

3.6. Almacenamiento de la información

Una vez recibida la factura, ésta se almacena temporalmente en el sistema de ficheros del teléfono, en la memoria interna. Tras validarla se extraen los campos que se han considerado relevantes y se ofrece al usuario la posibilidad de almacenarla definitivamente o eliminarla. Para el almacenamiento definitivo de las facturas se ha decidido emplear una base de datos SQLite, que es el gestor de base de datos incorporado de serie en Android.

La base de datos se podría almacenar en la memoria interna del teléfono o en la tarjeta SD. Por motivos de seguridad y eficiencia se ha decidido almacenarla en la memoria interna. El único inconveniente es que estamos limitados por el espacio disponible. No obstante, la base de

datos con la primera factura almacenada ocupa 40 KB, aproximadamente, y por cada factura que añadimos sólo se requieren entre 10 y 16 KB. Los teléfonos de última generación cuentan con más de 8 GB de memoria [32]. Pero incluso los un poco más antiguos disponen de, al menos, 1 GB, lo que les permitiría almacenar más de 60.000 facturas.

En la *Figura 8* se muestra el modelo de datos de la aplicación.

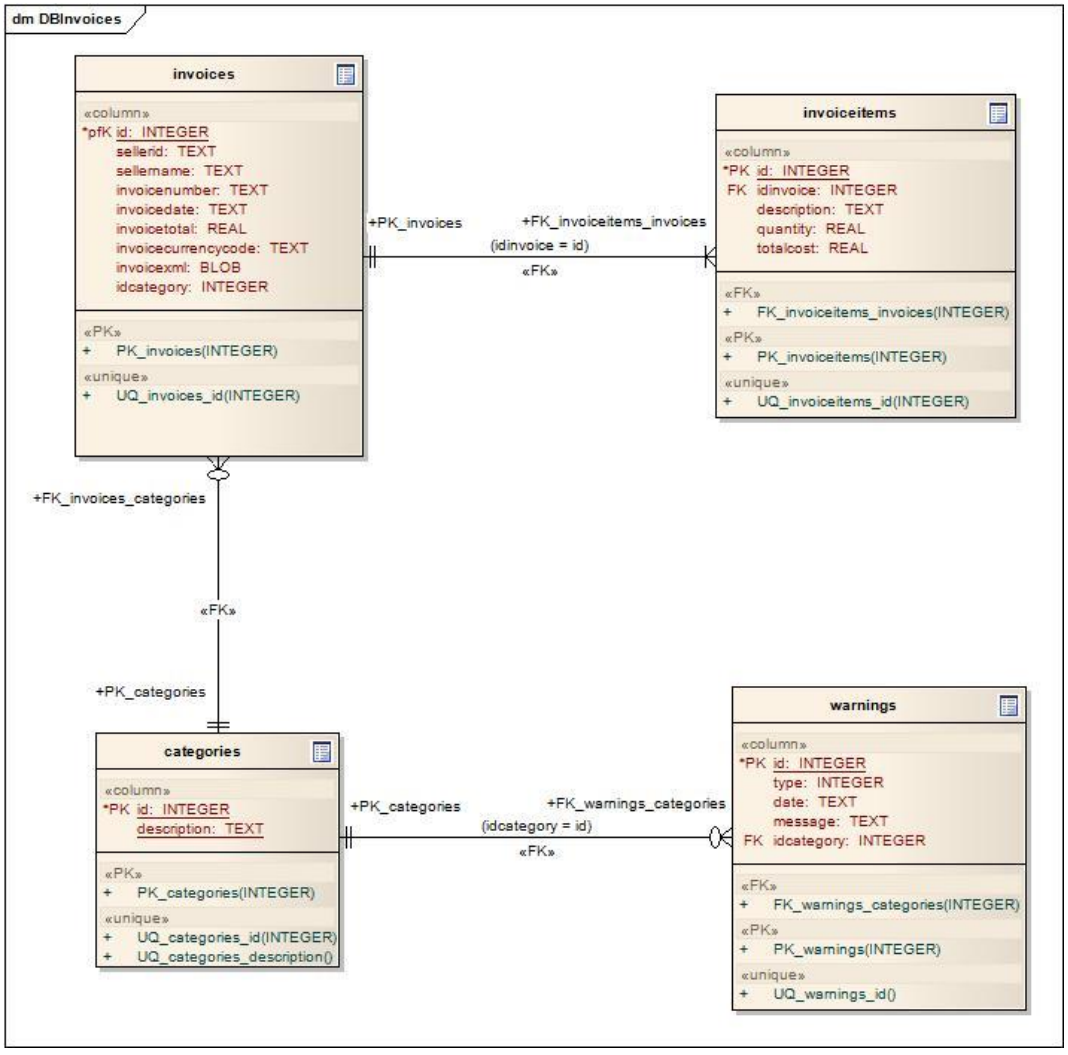

Figura 8: Modelo de datos

SQLite permite utilizar claves foráneas y borrado en cascada, pero sólo a partir de la versión 3.6.19, siendo necesaria la activación expresa. En el desarrollo de la aplicación se han empleado estas funcionalidades por lo que no funcionará con versiones anteriores de SQLite. Desde la versión 2.2 de Android ya se emplea una versión superior de SQLite [33], por lo que nos ha parecido una restricción aceptable.

3.7. Clasificación, consulta y borrado de facturas

Para ayudar al usuario en la gestión de sus facturas, la aplicación almacena la información de las facturas en una base de datos. La aplicación permite ver una lista de facturas ordenadas alfabéticamente por el nombre del vendedor y por fecha. Además se ofrece la posibilidad de buscar facturas indicando la siguiente información (toda o parte): NIF del vendedor, nombre del vendedor, fecha de la factura y categoría. Se pueden buscar las facturas de un año o las facturas de un mes concreto de un año. También se dispone de la opción de borrar facturas.

Las facturas se pueden clasificar según unas categorías definidas por el usuario. La aplicación dispone de una opción para gestionar las categorías, permitiendo altas, modificaciones y bajas. No se puede dar de baja una categoría que tenga asociadas facturas. Tampoco se pueden duplicar categorías, considerándose la misma categoría si la diferencia está solo en el cambio de mayúsculas y minúsculas.

3.8. Envío de facturas por correo electrónico

Para el envío de las facturas por correo electrónico se emplean los clientes de correo disponibles en el teléfono, dejando que el usuario elija, si tiene más de uno.

Las facturas están almacenadas en un campo BLOB de la base de datos SQLite. Esta información debe enviarse sin ningún tipo de manipulación para evitar que la firma electrónica deje de ser válida. Por este motivo no puede ser incluida en el cuerpo del mensaje y debe ser enviada como un anexo. El envío como anexo sin escribir la información en un fichero no es posible y si se almacena en la memoria interna, el fichero no es accesible para la aplicación de correo. Como se ha mencionado, esta ha sido la opción elegida, por motivos de seguridad. La solución escogida pasa por utilizar la clase de Android *FileProvider* [34] que es una subclase especial de *ContentProvider*, que facilita la compartición segura de ficheros entre aplicaciones. Por medio de ella se concede permiso temporal al paquete *com.android.email* para acceder a los ficheros de una carpeta de la caché de la aplicación (/data/data/edu.mistic.tfm.einvoice/cache/invoices), donde se dejan temporalmente las facturas a enviar.

Para la utilización de la clase *FileProvider* es necesario incluir la siguiente información en el fichero AndroidManifest.xml:

```
<provider
        android:name="android.support.v4.content.FileProvider"
        android:authorities="edu.mistic.tfm.einvoice.provider"
        android:exported="false"
        android:grantUriPermissions="true">
        <meta-data
```

```
            android:name="android.support.FILE_PROVIDER_PATHS"
            android:resource="@xml/file_path" />
    </provider>
```

Además es necesario crear el fichero *file_path.xml*, en la carpeta del proyecto /res/xml, con la siguiente información:

```
<paths xmlns:android="http://schemas.android.com/apk/res/android">
    <cache-path name="my_invoices" path="invoices/" />
</paths>
```

3.9. Copias de seguridad

La aplicación dispone de distintas opciones para realizar una copia de seguridad de los datos almacenados:

- Copia de seguridad en la nube.
- Copia de seguridad en la memoria externa (tarjeta SD).
- Copia de seguridad en un ordenador mediante Bluetooth.

3.9.1. Copia de seguridad en la nube

La aplicación emplea el Servicio de Backup de Android [35] que permite copiar los datos de una aplicación a un almacén en la nube. El proceso es transparente para el usuario. Cada vez que se realiza una acción que modifica la base de datos (altas de facturas, altas de categorías, clasificación de facturas, etc.), la aplicación solicita al Servicio una operación de copia de seguridad. Esta copia no se realiza inmediatamente. El Servicio de Backup la realiza cuando lo considera oportuno.

Además de la copia automática que se realiza tras cada actualización de la base de datos, el usuario puede solicitar actualizar la información almacenada en la nube mediante una opción disponible en el menú de “Copias de seguridad”. En este mismo menú se encuentra una opción para restaurar la última copia almacenada en la nube.

Cuando se desinstala la aplicación, se borra la base de datos. Gracias al Servicio de Backup de Android, cuando se vuelve a instalar la aplicación, se restaura automáticamente la última copia de la base de datos almacenada en la nube.

Para poder utilizar el servicio es necesario implementar un agente de copia de seguridad que es invocado por el gestor de backup, para proporcionarle los datos que se desean copiar. El agente también es invocado para restaurar el backup cuando la aplicación es reinstalada.

Para implementar el agente es necesario realizar previamente los siguientes pasos:

- Declarar el agente en el fichero *manifest*. En este caso el agente se llama *EinvoiceBackupAgent*:

```
<application
        ...
        android:backupAgent=".EinvoiceBackupAgent">
```

- Registrar la aplicación para poder utilizar el Servicio de Backup de Android empleando el transporte proporcionado por Google [36]. Al realizar el proceso de registro se obtiene una clave como se muestra en la *Figura 9*.

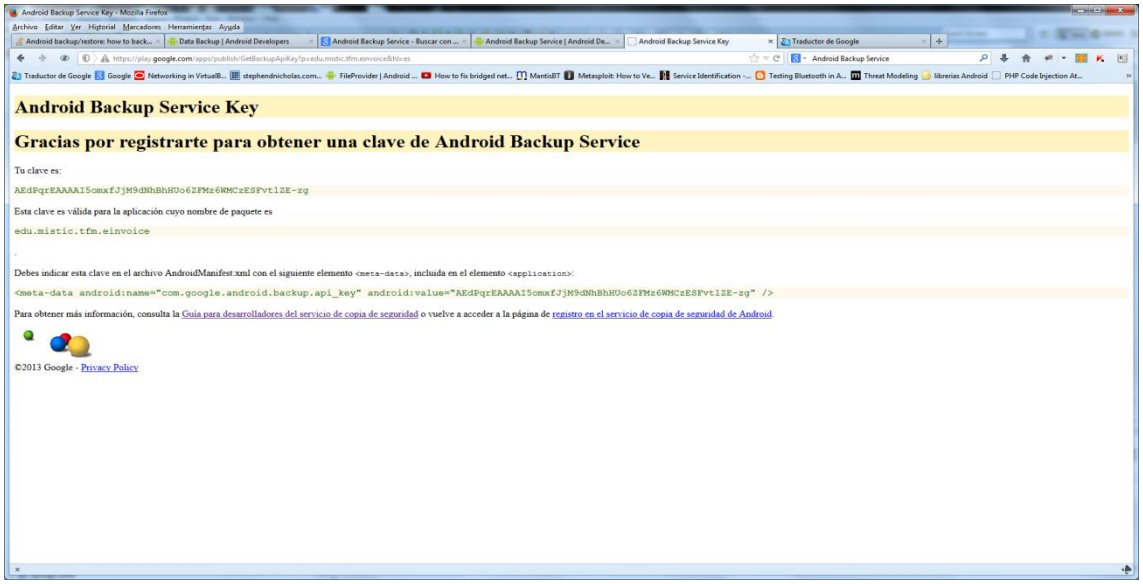

Figura 9: Registro para emplear el Servicio de Backup de Android

La clave obtenida también debe ser incluida en el fichero *manifest*:

```
<meta-data android:name="com.google.android.backup.api_key"
        android:value="AEdPqrEAAAAI5omxfJjM9dNhBhHUo6ZFMz6WMCzESFvtlZE-zg"/>
```

Para que funcione la copia de seguridad en la nube es necesario activar en el teléfono la opción *Copias de seguridad y restauración* y asociar una cuenta de copia de seguridad, como se muestra en la *Figura 10*.

Si se desea forzar la sincronización de los datos con el almacén de la nube, sin esperar a que lo realice el Servicio de Backup cuando lo considere adecuado, se puede utilizar la opción de *Sincronizar ahora* en la cuenta asociada a la copia de seguridad (ver *Figura 11*).

Aunque en la documentación proporcionada por Android se indica que el Servicio de Backup está disponible desde la versión 2.2.x, las pruebas

realizadas con la versión 4.2.2 han sido satisfactorias, pero no ha sido posible hacerlo funcionar en un teléfono con Android versión 2.3.6.

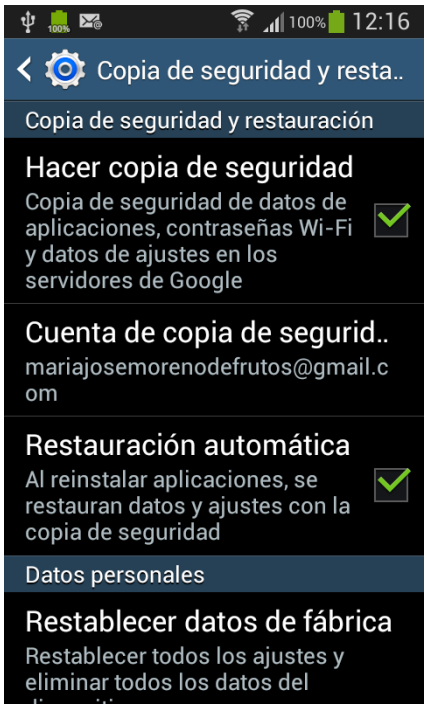

Figura 10: Copia de seguridad y restauración

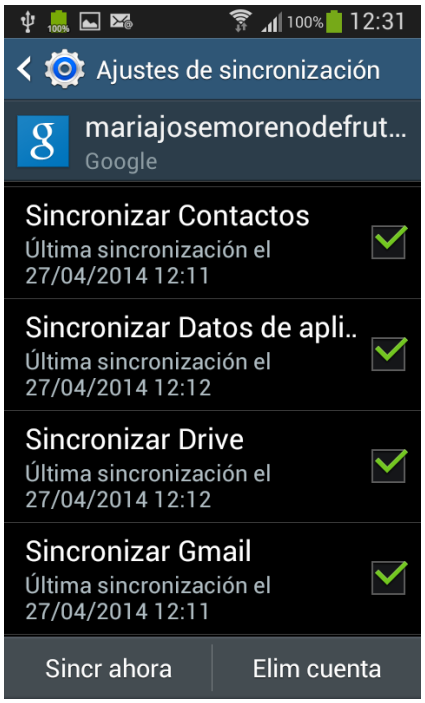

Figura 11: Sincronizar cuenta

3.9.2. Copia de seguridad en la memoria externa

La aplicación permite realizar una copia de seguridad de la base de datos SQLite con los datos de la aplicación en la tarjeta SD, si el teléfono dispone de una.

Antes de realizar la copia se cifran los datos empleando cifrado simétrico AES 256 modo CBC y Cifrado Basado en Contraseña (PBE - Password Based Encryption). Para ello se utiliza *Java Cryptography Architecture* (JCA) [37] y como proveedor criptográfico *SponglyCastle*, cuyas librerías ya estaban incluidas en el proyecto porque las utilizan los Componentes de Firma del MINETUR.

Cuando en el teléfono se selecciona la opción de realizar o restaurar copia de seguridad en la memoria externa, se solicita al usuario la contraseña o frase de paso que se empleará para cifrar/descifrar la información (ver *Figura 12*). Si en el proceso de restauración no se indica la misma contraseña que se especificó durante el proceso de copia, la restauración no se lleva a cabo y se indica el motivo al usuario.

Figura 12: Solicitud contraseña de cifrado

Cuando el usuario selecciona la opción *Realizar copia en memoria externa* el sistema crea en la tarjeta una carpeta con el nombre *invoices_copy* (ver *Figura 13*), en la que almacena la base de datos SQLite cifrada, con el nombre *DBInvoices_copy.db*.

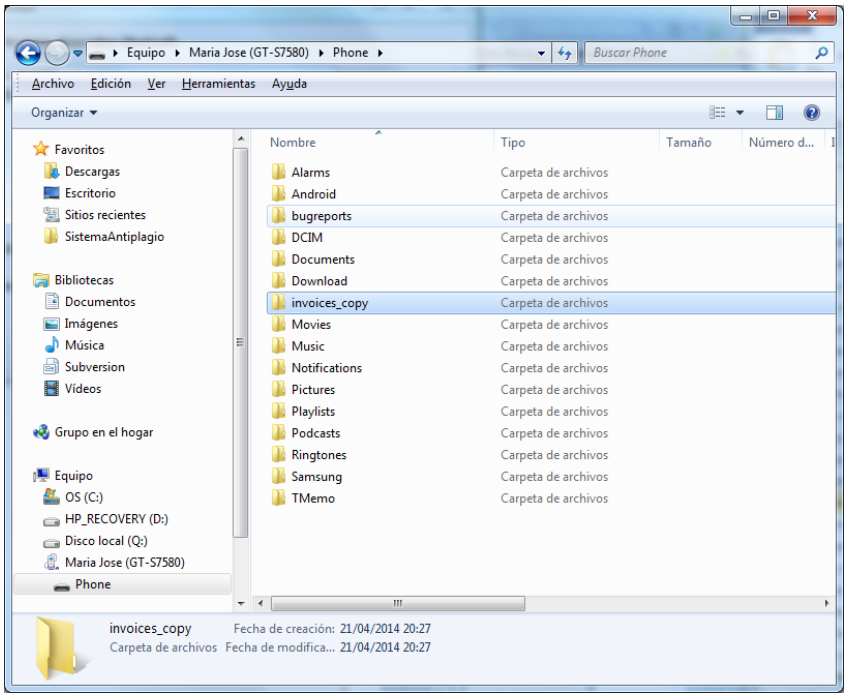

Figura 13: Tarjeta SD de un teléfono inteligente

Mediante la opción *Restaurar datos de la memoria externa* se copia la base de datos almacenada en la carpeta *invoices_copy* de la memoria externa a la carpeta /data/data/edu.mistic.tfm.einvoice/databases de la memoria interna, que es la ubicación de la base de datos. Durante el proceso de restauración se descifran los datos.

Conectando dos teléfonos a un ordenador es posible copiar la carpeta *invoices_copy* con su contenido de la tarjeta SD de uno al otro, permitiendo la trasferencia de la base de datos entre dos dispositivos.

3.9.3. *Copia de seguridad en un ordenador mediante Bluetooth*

Utilizando las mismas librerías que en el envío/recepción de facturas, se ha desarrollado una utilidad para realizar una copia de seguridad de la base de datos en un ordenado, utilizando Bluetooth.

Antes de realizar el envío se cifran los datos empleando cifrado simétrico AES 256 modo CBC y Cifrado Basado en Contraseña (PBE - Password Based Encryption). Para ello se utiliza *Java Cryptography Architecture* (JCA) [37] y como proveedor criptográfico *SponglyCastle*, cuyas librerías ya estaban incluidas en el proyecto porque las utilizan los Componentes de Firma del MINETUR.

Para el envío y restauración de la base de datos a un ordenador se ha creado una pequeña aplicación en Java que implementa el cliente de la comunicación Bluetooth. En la aplicación del teléfono se ha implementado el servidor de la comunicación.

Cuando en el teléfono se selecciona la opción de realizar o restaurar copia de seguridad empleando Bluetooth se le solicita al usuario la contraseña o frase de paso que se empleará para cifrar/descifrar la información. Si en el proceso de restauración no se indica la misma contraseña que se especificó durante el proceso de copia, la restauración no se lleva a cabo y se indica el motivo al usuario.

En el proceso de realización de la copia de seguridad (desde la aplicación Java) se selecciona el dispositivo origen de los datos y la carpeta de destino para almacenar el fichero con la base de datos cifrada, según se muestra en la *Figura 14*. El nombre del fichero de la base de datos cifrada es *BDInvoices_copy.db*.

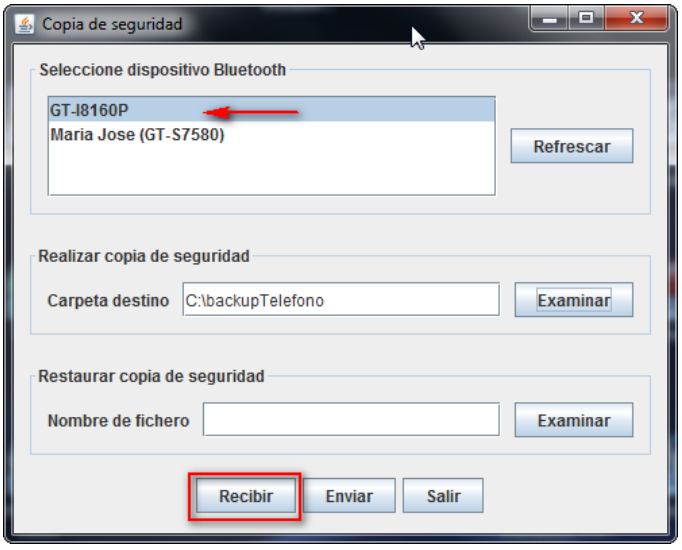

Figura 14: Aplicación Java para la copia de seguridad empleando Bluetooth

En el proceso de restauración de la copia de seguridad, desde la aplicación Java se selecciona el dispositivo destino de los datos y el fichero con la copia de la base de datos cifrada, según se muestra en la *Figura 15*.

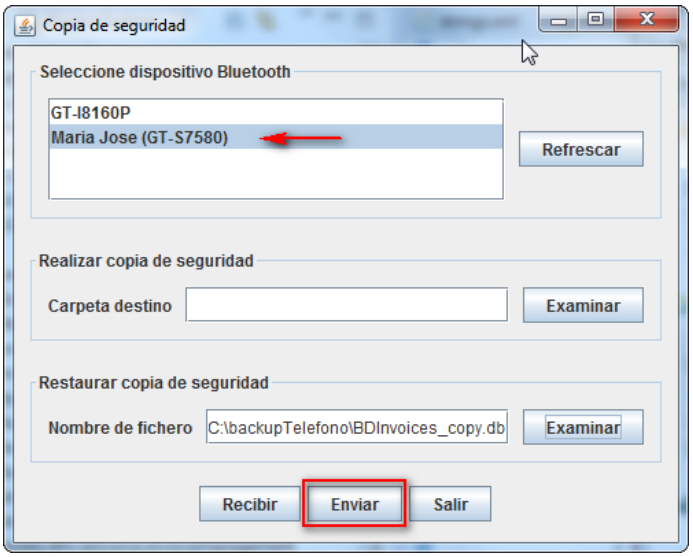

Figura 15: Aplicación Java para la restauración de la copia de seguridad empleando Bluetooth

Utilizando un ordenador como intermediario, esta utilidad también permite transferir la base de datos de un dispositivo a otro.

3.10. Gestión y Control del Gasto

La aplicación proporciona al usuario la posibilidad de hacer un seguimiento de sus gastos a lo largo del tiempo y de establecer límites mensuales para cada categoría definida.

Al dar de alta una categoría es posible establecer un límite de gasto mensual. Cuando el usuario clasifica una factura, si el importe mensual acumulado supera el límite fijado, la aplicación mostrará un aviso como el de la *Figura 16*.

Estos avisos se almacenan en una tabla de la base de datos y es posible consultarlos mediante una opción del menú inicial (ver *Figura 17*).

Figura 16: Aviso límite gasto superado

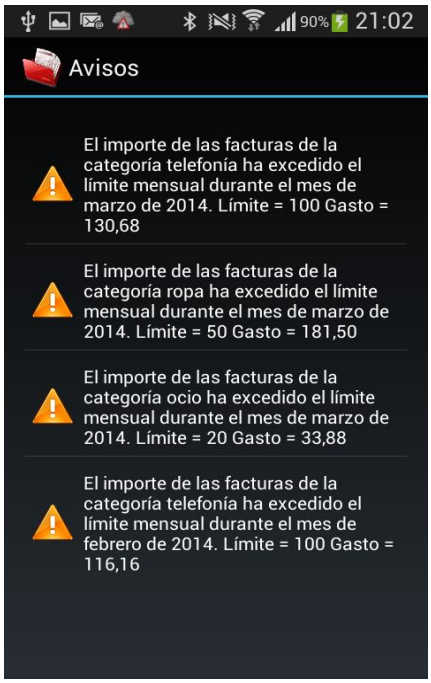

Figura 17: Listado de avisos

Desde la pantalla en la que se visualizan los avisos recibidos es posible borrarlos. También es posible revisar todos los gastos realizado hasta el momento y generar de nuevo los avisos. Con esta opción, además de los avisos por superación de límites aparecerán avisos indicando los periodos en los que no se han superado los límites, como se muestra en la *Figura 18*.

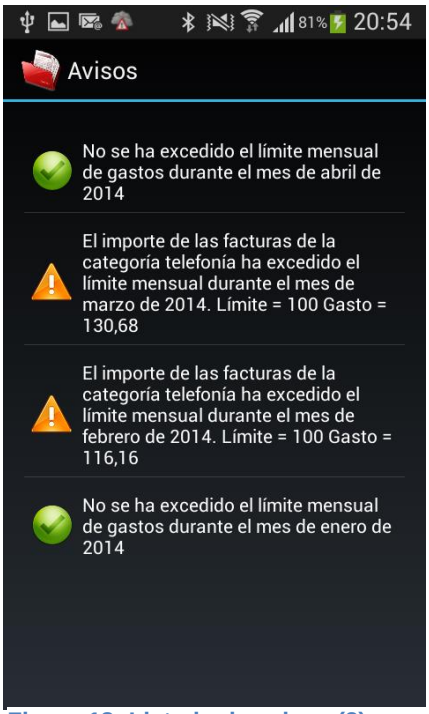

Figura 18: Listado de avisos (2)

Mediante distintas gráficas, la aplicación permite al usuario ver los gastos realizados a través del tiempo, clasificados por categorías o por proveedor.

Gastos por categorías

Mediante esta opción la aplicación muestra un diagrama de tarta con los gastos realizados en las distintas categorías. Se puede especificar un año y un mes. Si no se indican, el diagrama refleja todos los gastos realizados (ver *Figura 20* y *Figura 19*). Pulsando encima de cualquiera de los segmentos es posible ver el valor correspondiente. Los colores se seleccionan aleatoriamente, y si no nos gustan bastará con volver a pulsar el botón *Ver*, para que cambien.

Figura 20: Diagrama de tarta por categorías (2)

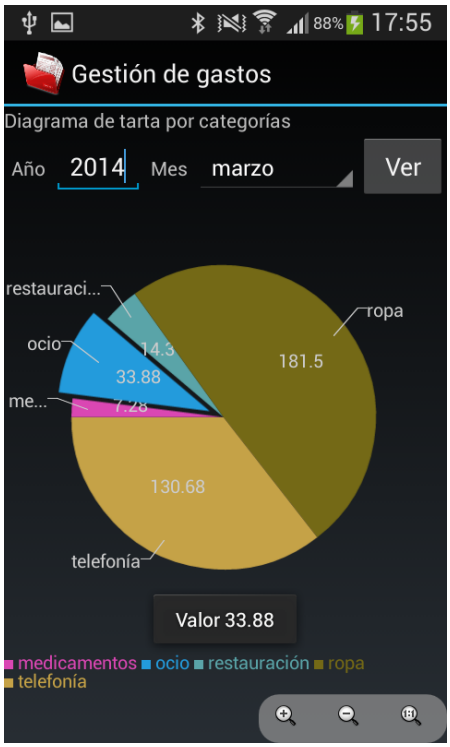

Figura 19: Diagrama de tarta por categorías

Gastos por proveedores

Mediante esta opción la aplicación muestra un diagrama de tarta con los importes pagados a los distintos proveedores. Se puede especificar un año y un mes. Si no se indican, el diagrama refleja todos los gastos realizados (ver *Figura 21*). El funcionamiento es similar al diagrama anterior. Si no se ven los literales de los proveedores, es posible desplazar la gráfica como se muestra en la *Figura 22*.

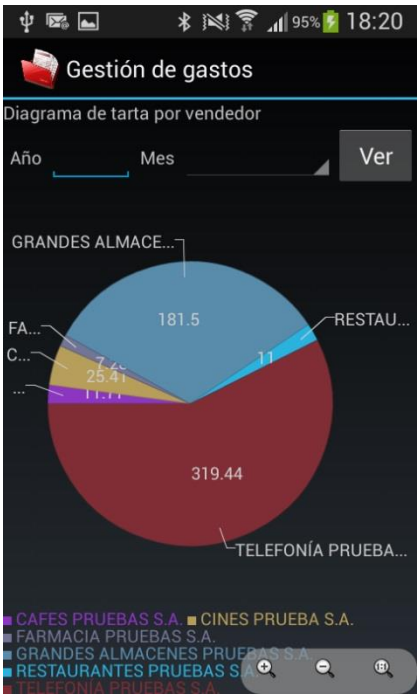

Figura 21: Diagrama de tarta por proveedores

Figura 22: Diagrama de tarta por proveedores (2)

Gastos anuales

Esta opción muestra un diagrama de barras con los gastos del año que se le indique. Si se le especifica una categoría, el diagrama muestra los gastos realizados cada uno de los meses en artículos de esa categoría (ver *Figura 23*). Si no se indica una categoría se muestra el total de los gastos de cada mes (ver *Figura 24*).

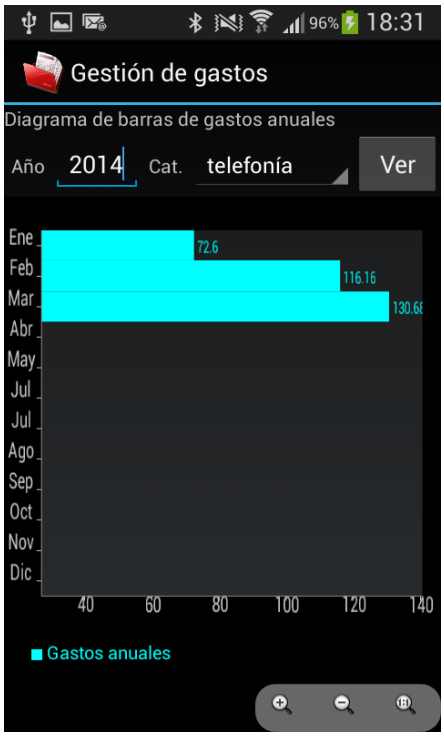

Figura 23: Diagrama de barras

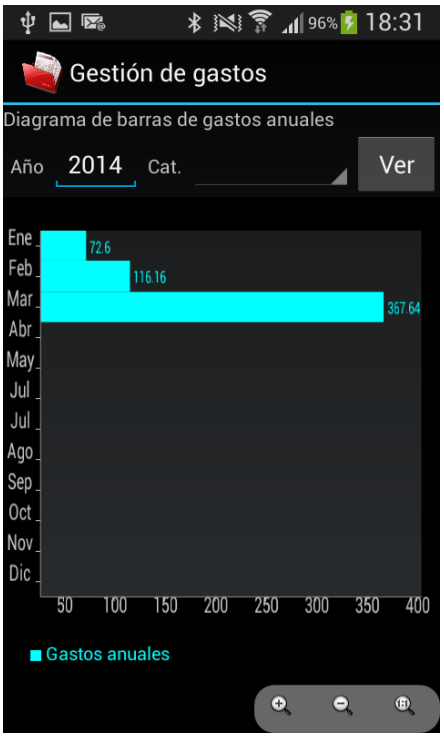

Figura 24: Diagrama de barras (2)

Situación de gastos

Esta opción muestra un diagrama de líneas con dos series. Para cada año y categoría la aplicación muestra una serie con el límite mensual y otra con los gastos realizados cada mes en esa categoría. Tras seleccionar el año, la aplicación actualiza el desplegable con las categorías, marcándolas con un color, con el siguiente significado:

- gris: no se han realizado gasto de esta categoría.
- rojo: se ha superado el límite mensual durante algún mes.
- amarillo: durante algún mes se ha estado próximo al límite (entre 80-100% del límite) pero no se ha superado.
- verde: en el resto de los casos.

Figura 25: Situación de gastos

Tras seleccionar una categoría y al pulsar el botón *Ver* aparece la siguiente gráfica con dos series:

- En azul el límite mensual
- En el color del desplegable los gastos mensuales de la categoría seleccionada

En la *Figura 27*, *Figura 26* y *Figura 28* se muestra un ejemplo de cada color.

Figura 28: Diagramas de líneas con gastos anuales de una categoría (3)

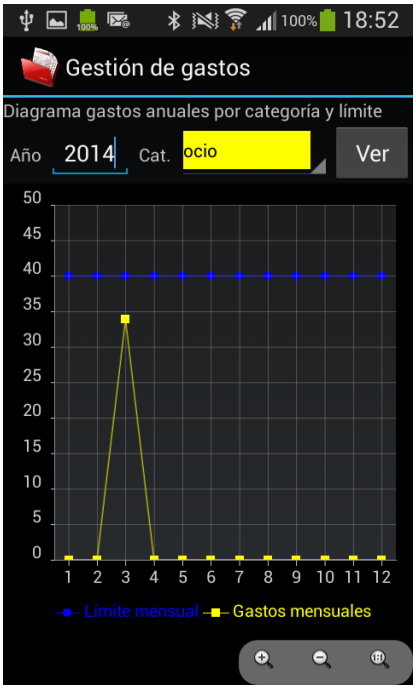

Figura 27: Diagramas de líneas con gastos anuales de una categoría (2)

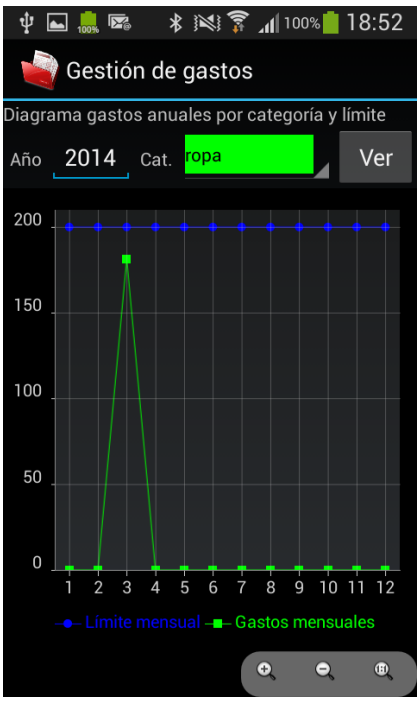

Figura 26: Diagramas de líneas con gastos anuales de una categoría (1)

4. Desarrollo de la solución

En este apartado se describen los aspectos más relevantes de la fase del desarrollo de la aplicación.

4.1. Instalación del entorno de desarrollo

El primer paso en el desarrollo de cualquier aplicación es la instalación del entorno de desarrollo. El kit para el desarrollo de aplicaciones (SDK) de Android proporciona las bibliotecas y las herramientas de desarrollo necesarias para crear, probar y depurar aplicaciones para Android. Aunque es posible descargar el SDK e integrarlo en un entorno de desarrollo integrado (IDE) preinstalado, se ha elegido la opción de instalar lo que denominan *ADT Bundle* (paquete de herramientas para desarrolladores de Android) que incluye lo siguiente:

- Eclipse + el plugin ADT para Eclipse
- Herramientas del SDK de Android (herramientas de depuración y pruebas)
- Herramientas de la plataforma SDK (herramientas dependientes de la plataforma para el desarrollo y la depuración)
- La plataforma SDK de Android. Hay una plataforma SDK disponible para cada versión de Android que incluye el fichero android.jar con las librerías.
- Imágenes del sistema Android para el emulador. Cada versión de la plataforma ofrece varias imágenes de sistemas para le emulador que ofrece un entorno de pruebas para las aplicaciones.

Como alternativa al emulador incorporado entre las herramientas del SDK de Android, se ha creado una máquina virtual Android con VirtualBox [7]. La imagen iso de la máquina (*android-x86-4.3-20130725.iso*) se ha descargado de [8]. Para poder utilizar la máquina virtual en la depuración de la aplicación es necesario configurar el adaptador de red en la opción "Adaptador puente" y seguir los pasos especificados en [9].

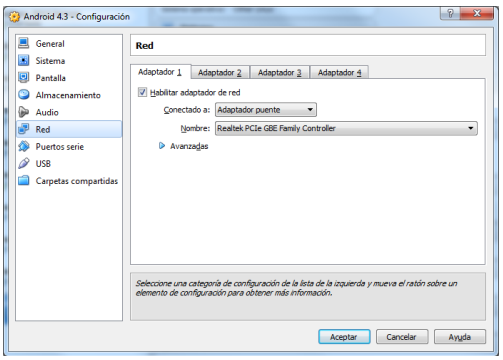

Figura 29: Adaptador de red de la máquina virtual Android

Además, para poder utilizar en el desarrollo, depuración y pruebas un teléfono móvil con S.O. Android (Samsung Galaxy Ace 2), se ha descargado del servidor de Samsung el driver para USB. Una vez instalado, para confirmar que el dispositivo está accesible se ejecuta el siguiente comando [10]:

```
adb devices
```

El desarrollo se realizará en un PC de sobremesa con sistema operativo (S.O.) Windows 7, al que se ha instalado un adaptador Bluetooth para poder realizar las pruebas del envío de facturas electrónicas.

Finalmente, para el desarrollo del servidor que envía las facturas al teléfono se ha instalado el JDK 1.7.0_51 [11] y Eclipse [12]. Aunque se podía haber utilizado el Eclipse instalado junto con el SDK de Android, se ha empleado uno distinto para facilitar la depuración.

4.2. Envío de facturas

Este módulo se ha desarrollado en Java y se han utilizado la librería *Bluecove*, para la comunicación Bluetooth, el paquete *Swing* de Java, para el diseño del interfaz de usuario y la librería log4j para la generación de los logs. En la *Figura 30* se muestran las clases principales de la aplicación.

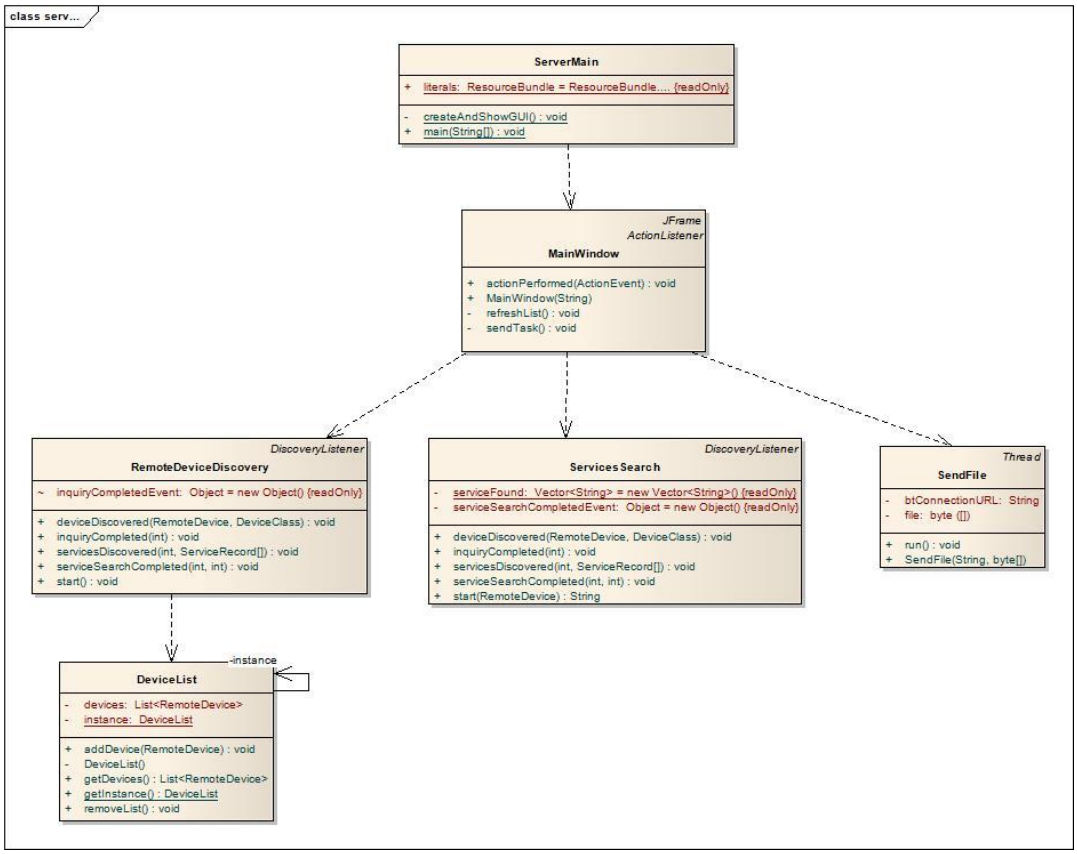

Figura 30: Diagrama de clases de la aplicación para el envío de facturas

La clase *ServerMain* tiene el método *Main* y se encarga de crear la ventana principal de la aplicación.

La clase *MainWindow* extiende la clase *JFrame* de Swing y es la ventana principal de la aplicación, desde la que el usuario podrá realiza todas las operaciones.

La clase *RemoteDeviceDiscovery* implementa la interfaz *DiscoveryListener* de la librería BlueCove y es la encargada de buscar dispositivos Bluetooth disponibles, devolviendo un objeto de tipo *DeviceList* con los datos de los dispositivos descubiertos.

La clase *ServiceSerch* también implementa la interfaz *DiscoveryListener* de la librería BlueCove. Esta clase recibe la información correspondiente a un dispositivo e intenta localizar un servicio específico, identificado por su UUID (Universally unique identifier), en dicho dispositivo. Si lo encuentra, devolverá su URL. Si el servicio no está disponible mostrará un mensaje indicándolo.

Finalmente la clase *SendFile* se encarga del envío del fichero seleccionado por el usuario al dispositivo deseado. Esta clase extiende la clase *Thread* y en su método *run* crea la conexión con el servidor y realiza el envío del fichero.

Los literales de la interfaz de usuario y del fichero de log no están escritos directamente en el código, sino que se extraen de un fichero de propiedades, *textGui.properties*, mediante la clase *ResourceBundle*. De esta forma la interfaz de usuario podrá ser fácilmente traducida a cualquier idioma. En principio se incorporan los ficheros en castellano y en inglés.

La aplicación genera un fichero de log para lo que se ha empleado la librería Apache Log4j [21]. Esta funcionalidad es configurable mediante el fichero *log4j.properties*.

4.3. Recepción de facturas

Para el desarrollo de este módulo se han empleado las librerías que incorpora Adroid para la gestión de las comunicaciones Bluetooth: android.bluetooth. A continuación se analizan las clases que intervienen en el proceso.

La clase *AndroidBluetoothReceiveActivity* extiende la clase *ListActivity*, por lo que se trata de una actividad de Android. En ella se muestran los mensajes relacionados con el proceso de recepción de la factura. En su método *onCreate* se instancia un objeto de la clase *BluetoothServerAsyncTask* que será el encargado de gestionar la recepción de la factura.

La pantalla dispone de un botón que permite al usuario cancelar la operación cuando lo desee, pero además en su método *onStop* también se cancela el proceso.

La clase *BluetoothServerAsyncTask* extiende la clase *AsyncTask*. Esta clase incorporada por Android permite realizar en background operaciones y publicar los resultados en el hilo del interfaz de usuario. Se trata de una clase *helper* (ayudante) en relación con las clases *Thread* y *Handler*. La clase *BluetoothServerAsyncTask*, en su método *doInBackground*, implementa el proceso de recepción y almacenamiento de la factura en la memoria interna del teléfono. Por otro lado, en su método *onPostExecute*, que se ejecuta cuando finaliza el método anterior, lanza una nueva actividad que se encargará de validar la factura recibida.

4.4. Validación de las facturas

La validación de las facturas consta de dos procesos que se ejecutan secuencialmente. En primer lugar se comprueba si la factura se ajusta al schema XML definido para el formato facturae versión 3.2. De ser así, se validará la firma electrónica. Para ello se han creado dos clases que extienden la clase *AsyncTask*: *ValidateSchemaXmlTask* y *ValidateSignTask*. La primera instancia un objeto de la clase *XmlManager para* la validación del schema y la segunda un objeto del a clase *SignValidation* para validar la firma.

Para la validación del schema XML se ha empleado la librería Xerces de Apache [23], que de forma muy sencilla permite comprobar si un fichero XML se ajusta a un schema determinado.

Se ha implementado la clase *XmlManager* para la gestión de los ficheros XML (las facturas). Su método *validateSchema* recibe el nombre del fichero XML a validar y devuelve un valor booleano con el resultado de la operación. El método *formatInvoice* devuelve una lista de objetos de tipo *Invoice* con los datos de las facturas almacenadas en el fichero XML. Es necesario tener en cuenta que en un documento XML se pueden recibir varias facturas del mismo vendedor.

Para la validación de la firma electrónica se han empleado los componentes proporcionados por el MINETUR. No obstante estos componentes utilizan la librería BouncyCastle y esta no funciona en Android. Es necesario sustituirla por la librería SponglyCastle y hacer los ajustes necesarios. La adaptación de los componentes para Android fue realizada por Moisés Fernández Blanco en su TFM [24], quien nos ha hecho el favor de proporcionárnoslos, evitándonos repetir el trabajo.

Por otro lado, los componentes tienen dependencias de otras librerías que tampoco se pueden incorporar directamente en los proyectos Android, pues incluyen paquetes *javax.**. Es necesario reenpaquetarlas, lo que se ha realizado siguiendo el procedimiento especificado en [29] y [30]. Se ha creado un nuevo artefacto (xml-security-for-android-1.0.jar) que incluye

las librerías xml-apis-1.1.03, xmlsec-1.4.7 y xercerImpl-2.9.1, modificando los paquetes *javax.** por *android.javax.**.

Además, en ciertas ocasiones ha sido necesario crear nuevas clases con una funcionalidad idéntica a las clases incluidas en los componentes pero realizando ajustes para que funcionen en Android, en especial cuando es necesario leer algún fichero de configuración, pues la forma de hacerlo en Java y en Android es diferente.

En la *Figura 31* se muestra el diagrama de las clases principales que intervienen en el proceso de validación de la firma electrónica de las facturas. En la clase *SignValidation* se instancias dos objetos: uno de la clase *Factura31ManagerAndroid* y otro de la clase *MisticPropsTruster*. Estos objetos se pasan a la clase *ValidarFirmaXML*, al invocar a su método *validar*, que es el encargado de realizar el proceso. Todas estas clases son adaptaciones de las existentes en los componentes del MINETUR pero la clase *ValidarFirmaXML* es la única de las mencionadas que no ha tenido que ser modificada.

Mediante los métodos de la clase *MisticPropsTruster* se comprueba si el certificado ha sido emitido por una de la AC de confianza. Para ello consulta el fichero de propiedades *assets/trust/miticTruster.properties* en el que se especifica la localización de los certificados de las AC de confianza. A continuación se muestran las primeras líneas de fichero para ver su estructura:

```
# Indica los certificados de prestadores admitidos por el Ministerio,
separados por comas.
signcerts.issuers.dnie=trust/certs/issuers/dnie/ACRAIZ-
SHA1.crt,trust/certs/issuers/dnie/ACDNIE001-
SHA1.crt,trust/certs/issuers/dnie/ACDNIE002-
SHA1.crt,trust/certs/issuers/dnie/ACDNIE003-SHA1.crt
signcerts.issuers.fnmt=trust/certs/issuers/fnmt/FNMT_CA_Cert.cer,trust/c
erts/issuers/fnmt/ACAdministracionPublica.cer,trust/certs/issuers/fnmt/F
NMT_CA_RAIZ.cer,trust/certs/issuers/fnmt/FNMT_CA_RAIZ_SHA1.cer,trust/cer
ts/issuers/fnmt/ACRAIZAPE.cer,trust/certs/issuers/fnmt/FnmtFirmaMovil.ce
r
signcerts.issuers.ACA=trust/certs/issuers/ACA/ACA_corporativos_csrs.crt,
trust/certs/issuers/ACA/ACA_raiz_csrs.crt,trust/certs/issuers/ACA/ACA_tr
usted_csrs.crt
signcerts.issuers.ACCV=trust/certs/issuers/ACCV/accv-
ca1.cer,trust/certs/issuers/ACCV/accv-
ca2.cer,trust/certs/issuers/ACCV/accv-
ca3.cer,trust/certs/issuers/ACCV/ACCVCA110.cer,trust/certs/issuers/ACCV/
ACCVCA120.cer,trust/certs/issuers/ACCV/ACCVRAIZ1.cer,trust/certs/issuers
/ACCV/ca.cer,trust/certs/issuers/ACCV/rootca.cer
```

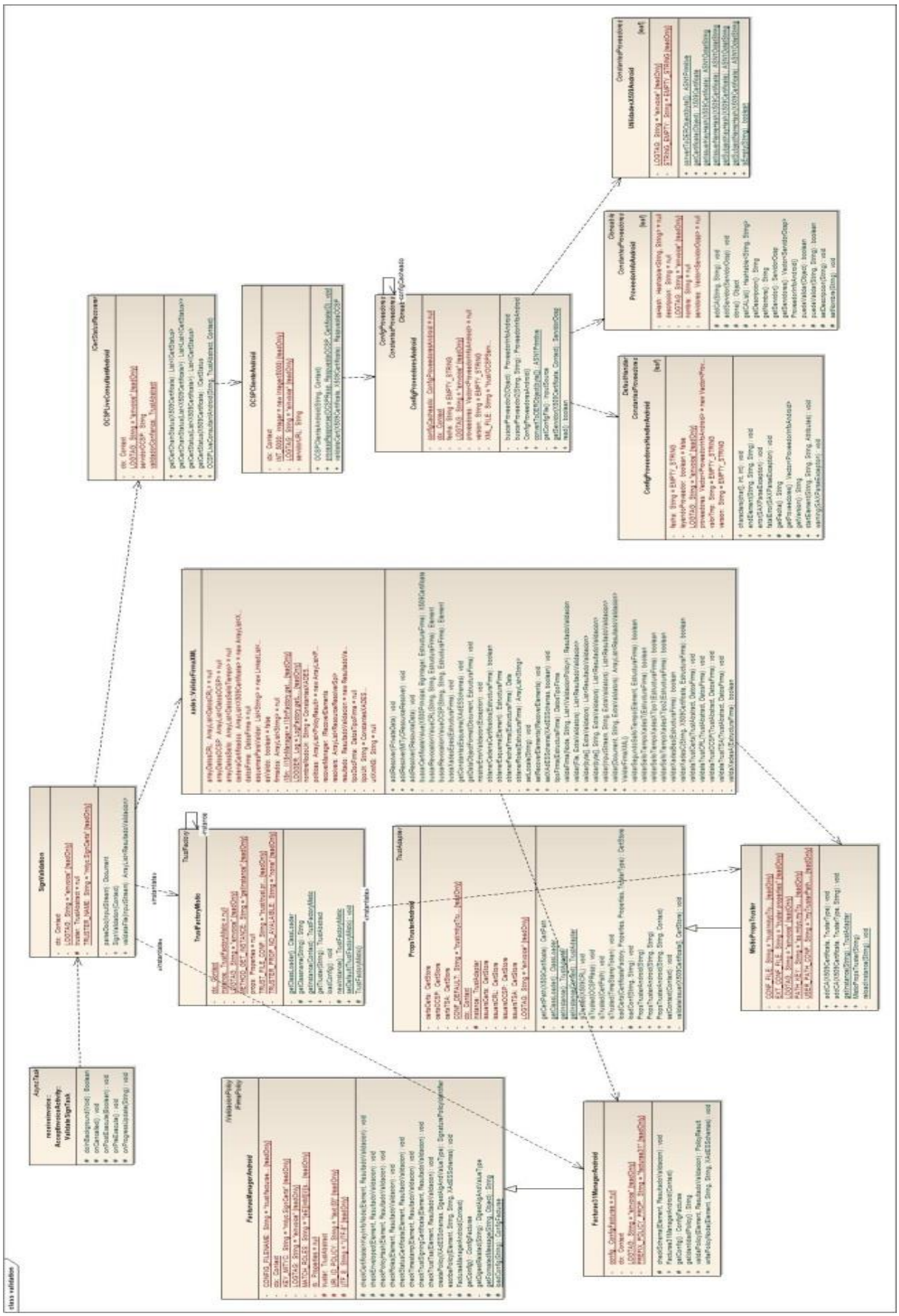

Figura 31: Diagrama de clases de la validación de la firma electrónica

El método *validaPolicy* de la clase *Factura31ManagerAndroid* se encarga de confirmar que la firma se ajusta a la política de la facturae: confirmación del schema XML de la firma XADES, confirmación de los nodos *SignaturePolicyIdentifier* y *KeyInfo*, etc. Para ello utiliza los datos de la política almacenados en el fichero de propiedades *assets/trust/facturae.properties*.

Una vez finalizadas las validaciones realizadas por la clase *ValidarFirmaXML*, si el proceso ha sido correcto, se comprueba el estado de revocación del certificado. Este proceso podría haber sido realizado por la clase *ValidarFirmaXML,* si se la pasa un objeto de la clase *OCSPLiveConsultantAndroid*. Pero se ha decidido realizar esta comprobación de manera independiente para darle un tratamiento diferente. Dado que la comprobación de la revocación depende de comunicaciones que podrían no estar disponibles, el hecho de que no se haya podido comprobar la revocación no supone el rechazo de la factura. Se avisará al usuario del hecho y él decidirá si la acepta o la rechaza.

El método *getCertStatus* de la clase *OCSPLiveConsultantAndroid* se encarga de comprobar si el certificado de firma ha sido revocado, empleando el servidor de OCSP señalado en el fichero *OCSPServersInfo.xml* para cada una de las AC. Para ello invoca diversos métodos de las clases *OCSPClienteAndroid*, *ConfigProveedoresAndroid*, *ConfigProveedoresHandlerAndroid*, *ProveedorInfoAndroid* y *UtilidadesX509Android*.

4.5. Clasificación, consulta y borrado de facturas

Como se ha indicado, para la persistencia de los datos se ha empleado la base de datos SQLite incorporada en el S.O. Android. Éste dispone de todas las herramientas necesarias para crear y gestionar este tipo de bases de datos.

Como medida de seguridad, todos los accesos a la base de datos (altas, bajas modificaciones o consultas) se realizan mediante “prepared statements”. Sólo por mostrar un ejemplo de cómo se puede hacer en Android, en el caso de las consultas:

```
String selection = "id = ?";
String[] selectionArgs = {String.valueOf(invoiceId)};
result = db.delete("invoices", selection, selectionArgs);
```

Se han creado clases para manejar la información almacenada en la base de datos, que se han incluido en el paquete *edu.mistic.tfm.einvoice.bean*: *Category*, *Invoice* y *InvoiceItem*.

Las clases encargadas de los accesos a la base de datos se han incluido en el paquete *edu.mistic.tfm.einvoice.dao*. La clase *InvoiceSQLiteHelper* extiende a la clase abstracta de Android *SQLiteOpenHelper* y es en la que

se define la estructura de la base de datos y en la que se configura la utilización de las claves foráneas:

```
db.execSQL("PRAGMA foreign_keys=ON;")
```

En este paquete también se encuentra las interfaces donde se declaran los métodos a implementar por las clases que realizan el acceso a la base de datos: *InvoiceDaoInterface*, *InvoiceItemDaoInterface* y *CategoryDaoInterface*, y las clases correspondientes: *InvoiceDao*, *InvoiceItemDao* y *CategoryDao*.

Al dar de alta la tabla de las categorías se ha creado una restricción para que la descripción sea única y además se ha añadido el parámetro "Collate Nocase" para que se consideren iguales las entradas OCIO y ocio (si ya se ha dado de alta una no se puede dar de alta la segunda). No obstante esto no es suficiente para considerar iguales alimentación y “alimentacion” (los acentos). Se plantea como una posible mejora de la aplicación el manejo correcto de los acentos.

```
CREATE TABLE categories
(id INTEGER PRIMARY KEY AUTOINCREMENT,
 description TEXT COLLATE NOCASE,
monthlylimit INTEGER,
CONSTRAINT description_UNIQUE UNIQUE (description))
```

En el paquete *edu.mistic.tfm.einvoice.invoicemanagement* se encuentran las clases que implementan la interfaz del usuario en lo relativo a la gestión de las facturas. Estas clases extienden la clase de Android *ActionBarActivity*. En las distintas pantallas se emplean menús de opciones y menús contextuales. En lugar de estos últimos se podía haber empleado lo que Android denomina “contextual action mode” pero sólo está disponible a partir de la versión 3.0 de Android. Por este motivo se ha decidido no emplearlo para poder ejecutar la aplicación en dispositivos con versiones anteriores.

4.6. Envío de facturas por correo electrónico

La clase que implementa el envío de las facturas por correo electrónico es *SendInvoiceMailActivity*, que extiende la clase *ActionBarActivity*, como todas las clases del interfaz de usuario.

Cuando el usuario ha seleccionado la factura que desea enviar por correo, se le muestra una pantalla para que indique la dirección a la que quiere realizar el envío, el asunto del mensaje y el cuerpo del mismo. El asunto aparece inicialmente cumplimentado con el número de la factura y el cuerpo con el número de la factura y el vendedor. A continuación el sistema le muestra los clientes de correo disponibles para que seleccione el que desee utilizar, como se ve en la *Figura 32*.

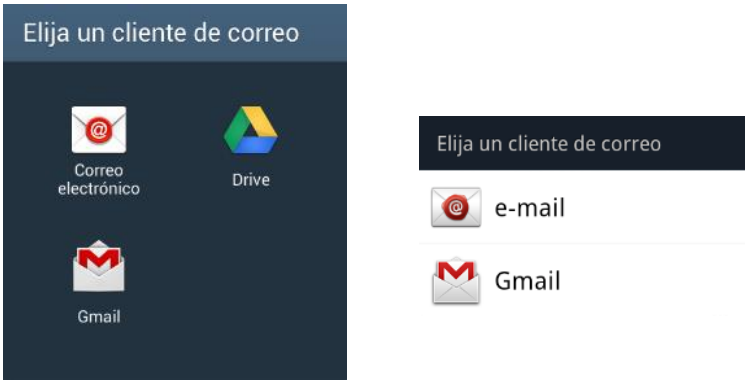

Figura 32: Selección de clientes de correo

4.7. Copias de seguridad

La aplicación permite realizar copias de seguridad de la información de tres formas distintas:

- Copias en la nube.
- Copias en la memoria externa.
- Copias en un ordenador mediante Bluetooth.

4.7.1. Copias de seguridad en la nube

Para poder realizar copias de seguridad en la nube empleando el Servicio de Backup de Android es necesario implementar un agente de backup (en adelante agente). En este caso el agente está implementado con la clase *EinvoiceBackupAgent* que extiende *BackupAgentHelper*. Esta clase es proporcionada por Android para minimizar el código a desarrollar.

En la clase *EinvoiceBackupAgent* se deben emplear objetos ayudantes (*helper*) que realizan copias de seguridad de ciertos tipos de información. Android proporciona dos tipos de ayudantes: uno para realizar copias de las preferencias y otro para realizar copias de la memoria interna. En este caso se ha empleado el ayudante que realiza la copia de los ficheros de la memoria interna: *FileBackupHelper*. Pero como sólo se desea hacer copia de seguridad de la base de datos, que se encuentra en una carpeta específica, se ha realizado un pequeño cambio en el método *getFilesDir*.

El proceso de lectura/escritura que realiza esta clase no es "*threadsafe*". Para asegurar que el agente no lee ni escribe la base de datos a la vez que las actividades se emplean bloques sincronizados. Para ello en la actividad *MainActivity* se ha declarado el objeto *sDataLock* que luego será empleado cada vez que se desea acceder a la base de datos, por las distintas actividades (operaciones de actualización) o por el agente.

```
synchronized (MainActivity.sDataLock) {
      <operación sobre la base de datos>
}
```

La clase *EinvoiceBackupAgent* ha sido desarrollada basándonos en las clases de ejemplo proporcionadas por Android.

4.7.2. Copia de seguridad en la memoria externa

Para la realización de copias de seguridad en la memoria externa se han desarrollado cinco clases:

- *PasswordActivity* extiende *ActionBarActivity* y se encarga de solicitar la contraseña que se empleará para cifrar/descifrar la base de datos.

- *BackupInExternalMemory* implementa los métodos para realizar la copia y la restauración de la base de datos en la memoria externa, invocando a los métodos de cifrado y descifrado de la información.

- *BackupExternalMemoryActivity extiende ActionBarListActivity* y se encarga de mostrar al usuario el estado del proceso de copia de seguridad, indicándole los posibles errores que se produzcan o la finalización correcta de la operación.

- *RestoreExternalMemoryActivity extiende ActionBarListActivity* y se encarga de mostrar al usuario el estado del proceso de restauración de la copia de seguridad, indicándole los posibles errores que se produzcan o la finalización correcta de la operación.

- *PasswordBasedEncryption* implementa los métodos para cifrar y descifrar la información empleando como contraseña la proporcionada por el usuario. Para ello emplea las clases del paquete JCA de Java y el proveedor criptográfico Spongycastle. En concreto se ha empleado el algoritmo PBEWITHSHA256AND256BITAES-CBC-BC proporcionado por este proveedor.

4.7.3. Copias de seguridad en un ordenador empleando Bluetooth

Para realizar las copias de seguridad mediante Bluetooth se han empleado las mismas librerías que para la transferencia de facturas desde un ordenador al teléfono.

Como en toda comunicación Bluetooth es necesario que uno de los dispositivos realice la función de servidor y el otro de cliente. En este caso el teléfono vuelve a hacer de servidor y el ordenador de cliente. Como siempre, es necesario que los dos dispositivos intercambien un pin.

Ahora el teléfono puede levantar tres servicios diferentes: recepción de factura, envío de copia de seguridad y restauración de copia de seguridad. Para distinguirlos, cada uno se identifica con un ID diferente:

- "0000110100001000800000805F9B34FA": Envío al ordenador de la copia de seguridad.

- "00001101000010008000008O5F9B34FC": Restauración de la copia de seguridad.

- "00001101000010008000008O5F9B34FB": Recepción de facturas.

La aplicación cliente (ejecutada en el ordenador) debe buscar el servicio adecuado en cada caso.

Se ha desarrollado una pequeña aplicación Java *EInvoiceBackup* que se debe ejecutar en el ordenador que almacenará las copias se seguridad. Esta aplicación es muy similar a la desarrollada para el envío de facturas desde un ordenador al teléfono, por lo que tiene las mismas clases (ver *Envío de facturas*). Sólo se ha añadido una nueva clase *ReceiveFile* que implementa la recepción de un fichero enviado desde el teléfono (la copia de la base de datos). La nueva aplicación permite enviar y recibir ficheros hacia y desde el teléfono.

Para la realización de copias de seguridad empleando Bluetooth se emplean cuatro clases:

- *PasswordActivity* extiende *ActionBarActivity* y se encarga de solicitar la contraseña que se empleará para cifrar/descifrar la base de datos. Es la misma clase utilizada para las copias en memoria externa.

- *AndroidBluetoothSendBackupActivity* se encarga de implementar el envío del fichero al ordenador empleando Bluetooth. Para ello levanta el servicio correspondiente y cuando recibe la conexión del cliente cifra los datos con la contraseña recibida y manda los datos cifrados.

- *AndroidBluetoothReceiveBackupActivity* implementa la restauración de una copia de seguridad, empleando Bluetooth. Para ello levanta el servicio correspondiente y cuando detecta la conexión de un cliente, recibe los datos, los descifra con la contraseña que ha recibido de la clase *PasswordActivity* y restaura los datos descifrados en el lugar adecuado, es decir, en la carpeta /data/data/edu.mistic.tfm.einvoice/databases.

- *PasswordBasedEncryption* implementa los métodos para cifrar y descifrar la información empleando como contraseña la proporcionada por el usuario. Para ello emplea las clases del paquete JCA de Java y el proveedor criptográfico Spongycastle. En concreto se ha empleado el algoritmo PBEWITHSHA256AND256BITAES-CBC-BC proporcionado por este proveedor. Es la misma clase utilizada en las copias en memoria externa.

4.8. Gestión y Control del Gasto

Para el desarrollo de los gráfico incorporados en esta sección se ha empleado la librería *AChartEngine* [38]. En su página web ofrece una aplicación de demostración que ha sido de gran utilidad para el desarrollo. La incorporación de la librería se ha realizado mediante la inclusión de la siguiente sección en el fichero pom.xml:

```
<dependency>
        <groupId>org.achartengine</groupId>
        <artifactId>achartengine</artifactId>
        <version>1.1.0</version>
</dependency>
```

Las clases que implementan los distintos gráficos se encuentran en el paquete *edu.mistic.tfm.einvoice.expensemanagement* y son las siguientes:

- PieChartExpenditureByCategories implementa el gráfico de tarta con los gastos de las distintas categorías.
- PieChartExpenditureBySeller implementa el diagrama de tarta con las compras realizadas a los distintos proveedores.
- BarChartByCategoryByYear implementa el diagrama de barras con los gastos mensuales a lo largo de un año, totales o de una categoría.
- XYChartByCategoryByYear implementa el diagrama de líneas con los gastos mensuales a lo largo de un año, de una categoría, junto con su límite mensual.

Además, la clase *invoiceDao* incluye las consultas a la base de datos para obtener los datos necesarios para los gráficos.

En la gestión de los avisos interviene las siguientes clases:

- *Warning* para la gestión de los objetos recuperados de la base de datos.
- *WarningDao* y *WarningManagement* que implementan las operaciones de base de datos relacionadas con los avisos.
- *WarningControl* que realiza las operaciones necesarias para determinar cuándo es necesario generar un aviso.
- *ListWarningsActivity* extiende *ActionBarListActivity* y se encarga de mostrar al usuario la lista de avisos generados por la aplicación, permitiéndole borrarlos todos. Desde el menú de esta pantalla es posible revisar todos los gastos realizado hasta el momento y generar

de nuevo los avisos. Con esta opción, además de los avisos por superación de límites aparecerán avisos indicando los periodos en los que no se han superado los límites.

5. Juego de pruebas

En este apartado se describen los casos de prueba que se llevarán a cabo para confirmar el correcto funcionamiento de la aplicación, así como el resultado de su ejecución.

5.1. Creación de las facturas electrónicas

Para la ejecución de los casos de prueba se han creado un conjunto de facturas electrónicas de diversos proveedores ficticios, firmadas correctamente con distintos certificados electrónicos, así como facturas incorrectas en su estructura o en su firma (firmante no de confianza o error en la firma).

Para la creación de las facturas electrónica se ha descargado la aplicación Facturae [13] y se han empleado certificados electrónicos emitidos por las siguientes autoridades de certificación reconocidas: la FNMT y la Dirección General de la Policía (DNIe).

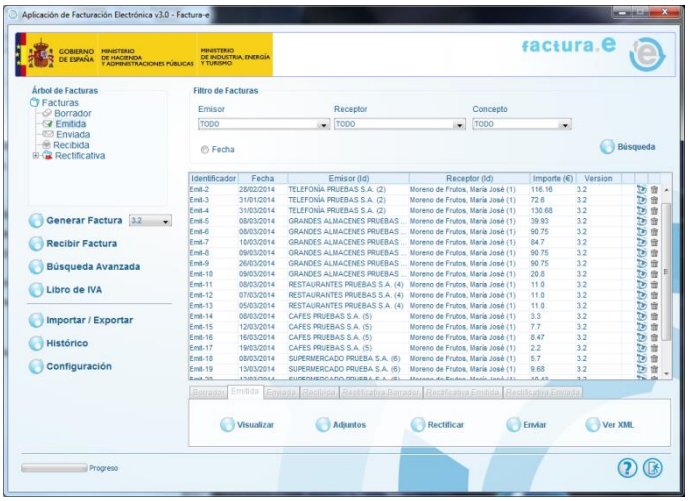

Figura 33: Aplicación Facturae

La aplicación Facturae permite la generación de facturas electrónicas según tres perfiles: versión 3.0, 3.1 o 3.2. Se ha empleado este último para generar el juego de pruebas, pues es el formato que validará la aplicación.

Todos los emisores de las facturas son ficticios y he inventado sus CIF, aunque respetando el formato de los mismos:

- A28000016: TELEFONÍA PRUEBAS S.A.
- A28000024: GRANDES ALMACENES PRUEBAS S.A.
- A28000032: RESTAURANTES PRUEBAS S.A.
- A28000040: CAFES PRUEBAS S.A.

- A28000058: SUPERMERCADO PRUEBA S.A.
- A28000065: FARMACIA PRUEBAS S.A.
- A28000073: CINES PRUEBAS S.A.
- A28000081: TEATRO PRUEBAS S.A.

Para poder probar las utilidades de clasificación y control de consumo se han creado facturas por diversos conceptos que nos permitirá clasificarlas según las siguientes categorías:

- Alimentación
- Ocio
- Ropa
- Restauración
- Telefonía
- Artículos cosméticos.
- Medicamentos

Las facturas se han firmado electrónicamente con distinto certificado en función del suministrador:

- Con el certificado de la FNMT: A28000016, A28000024, A28000073 y A28000081.
- Con el DNIe: A28000032, A28000040, A28000058 y A28000065

Para validar que las facturas emitidas son correctas se ha utilizado la aplicación web proporcionada por el MINETUR y el MINHAP [14]. En la *Figura 34* se muestra un ejemplo de las validaciones.

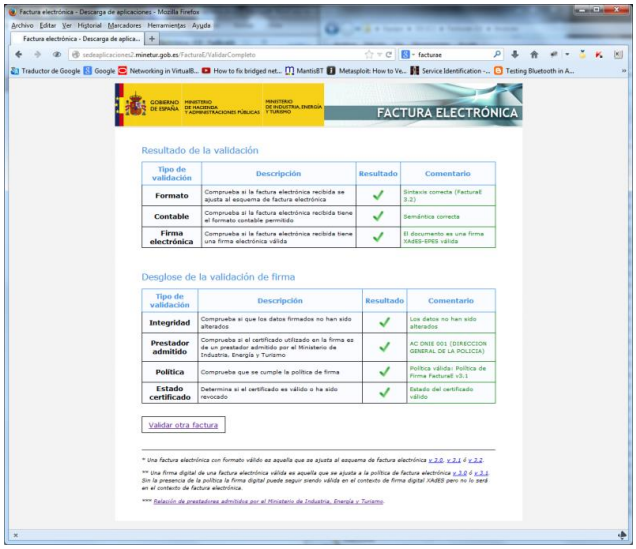

Figura 34: Aplicación web MINETUR y MINHAP para validar facturas electrónicas

En la *Tabla 2* se muestran las facturas correctas generadas.

Tabla 2: Casos de prueba correctos

Nombre	Descripción	Certificado	Resultado
Emit-2_A28000016_telefonía.xsig	Factura telefonía importe total 116,16 €	FNMT	OK
Emit-3_A28000016_telefonía.xsig	Factura telefonía importe total 72,6 €	FNMT	OK
Emit-4_A28000016_telefonía.xsig	Factura telefonía importe total 130,68 €	FNMT	OK
Emit-5_A28000024_bellez.xsig	Factura art. belleza importe total 39,93 €	FNMT	OK
Emit-6_A28000024_ropa.xsig	Factura ropa importe total 90,75 €	FNMT	OK
Emit-7_A28000024_ropa.xsig	Factura ropa importe total 84,7 €	FNMT	OK
Emit-8_A28000024_ropa.xsig	Factura ropa importe total 90,75 €	FNMT	OK
Emit-9_A28000024_ropa.xsig	Factura ropa importe total 90,75 €	FNMT	OK
Emit-10_A28000024_ocio.xsig	Factura ocio importe total 20,8 €	FNMT	OK
Emit-11_A28000032_restauración.xsig	Factura restauración importe total 11,0 €	DNIe	OK
Emit-12_A28000032_restauración.xsig	Factura restauración importe total 11,0 €	DNIe	OK
Emit-13_A28000032_restauración.xsig	Factura restauración importe total 11,0 €	DNIe	OK
Emit-14_A28000040_restauración.xsig	Factura restauración importe total 3,3	DNIe	OK
Emit-15_A28000040_ocio.xsig	Factura ocio importe total 7,7 €	DNIe	OK
Emit-16_A28000040_ocio.xsig	Factura ocio importe total 8,47 €	DNIe	OK
Emit-17_A28000040_restauración.xsig	Factura restauración importe total 2,2 €	DNIe	OK
Emit-18_A28000058_alimentación.xsig	Factura alimentación importe total 5,7 €	DNIe	OK
Emit-19_A28000058_belleza.xsig	Factura art. belleza importe total 9,68 €	DNIe	OK
Emit-20_A28000058_alimentación.xsig	Factura alimentación importe total 10,43 €	DNIe	OK
Emit-21_A28000058_alimentación.xsig	Factura alimentación importe total 8,62 €	DNIe	OK
Emit-22_A28000058_alimentación.xsig	Factura alimentación importe total 7,53 €	DNIe	OK
Emit-23_A28000058_alimentación.xsig	Factura alimentación importe total 9,42 €	DNIe	OK
Emit-24_A28000058_alimentación.xsig	Factura alimentación importe total 4,89 €	DNIe	OK
Emit-25_A28000065_medicamentos.xsig	Factura medicamentos importe total 7,28 €	DNIe	OK
Emit-26_A28000065_belleza.xsig	Factura art. belleza importe total 24,2 €	DNIe	OK
Emit-27_A28000065_medicamentos.xsig	Factura medicamentos importe total 7,28 €	DNIe	OK
Emit-28_A28000073_ocio.xsig	Factura ocio importe total 8,47 €	FNMT	OK
Emit-29_A28000073_ocio.xsig	Factura ocio importe total 16,94 €	FNMT	OK
Emit-30_A28000081_ocio.xsig	Factura ocio importe total 60,5 €	FNMT	OK
Emit-31_A28000081_ocio.xsig	Factura ocio importe total 90,75 €	FNMT	OK

Para confirmar que la aplicación funciona correctamente es imprescindible probar que es capaz de detectar errores en las facturas recibidas. La aplicación no debe almacenar facturas incorrectas.

La aplicación Facturae no permite utilizar certificados caducados ni emitir facturas con estructura incorrecta. Para poder generar los ejemplos erróneos se han empleado los componentes de firma del MINETUR, que permite generar firmas XADES-EPES enveloped con la política de Facturae.

En la *Tabla 3* se muestran las facturas erróneas generadas.

Tabla 3: Casos de prueba erróneos

Nombre	Descripción	Certificado	Resultado
XAdES-EPES-Facturae-caducado.xsig	Factura correctamente firmada con certificado caducado	FNMT caducado	ERROR: Certificado caducado
XAdES-BES-Factura.xsig	Factura firmada sin especificar política con certificado correcto.	FNMT	ERROR: Política incorrecta
XAdES-EPES-Facturae-modificada.xsig	Factura firmada correctamente pero modificada con posterioridad.	FNMT	ERROR: Firma alterada
XAdES-EPES-Facturae-modificada2.xsig	Factura firmada correctamente pero modificada con posterioridad.	DNIe	ERROR: Firma alterada
XAdES-EPES-Facturae-NoConfianza.xsig	Factura correctamente firmada con certificado emitido por una AC de pruebas.	prueba	ERROR: El certificado no es de un prestados admitido
XAdES-EPES-Facturae-SinNumFactura.xsig	Factura incorrecta (sin número de factura -InvoiceNumber-) firmada correctamente	FNMT	ERROR: Estructura de la factura incorrecta.
XAdES-EPES-Facturae-SinImpTotal.xsig	Factura incorrecta (sin importe total -TotalInvoicesAmount-) firmada correctamente	FNMT	ERROR: Estructura de la factura incorrecta.
XAdES-EPES-Facturae-SinImpTotal-NoConfianza.xsig	Factura incorrecta (sin importe total -TotalInvoicesAmount-) firmada con certificado de pruebas	prueba	ERROR: Estructura de la factura incorrecta.
XAdES-EPES-Facturae-SinImpTotal-caducado.xsig	Factura incorrecta (sin importe total -TotalInvoicesAmount-) firmada con certificado caducado	FNMT caducado	ERROR: Estructura de la factura incorrecta.

5.2. Casos de prueba

Los casos de prueba se han clasificado en las siguientes categorías:

- Pruebas relacionadas con la recepción de facturas.
- Pruebas relacionadas con la validación del schema XML de las facturas.
- Pruebas relacionadas con la validación de las firmas electrónicas de las facturas.
- Pruebas de la clasificación, consulta y borrado de las facturas.
- Pruebas del módulo de control y gestión del gasto.

- Pruebas del envío de facturas electrónica por correo electrónico.
- Pruebas de las copias de seguridad de las facturas.

Para cada uno de los apartados se describirán los objetivos perseguidos con las pruebas, los casos de prueba a realizar, el resultado esperado y el resultado obtenido de la ejecución.

5.2.1. Pruebas de la recepción de facturas

El objetivo de estas pruebas es confirmar el correcto funcionamiento de los módulos de envío y recepción de facturas mediante Bluetooth. Si las dos aplicaciones (la del servidor y la del teléfono) están preparadas, la factura se debe recibir en el teléfono. En caso contrario se deben mostrar los mensajes de error correspondientes.

En la *Tabla 4* se muestran las pruebas definidas y el resultado obtenido tras su ejecución.

Tabla 4: Pruebas de la recepción de facturas

Código	Descripción	Resultado esperado	Resultado
CP-1-01	Búsqueda de dispositivos en el servidor con el teléfono móvil apagado	No encuentra dispositivos.	OK
CP-1-02	Búsqueda de dispositivos en el servidor con el teléfono móvil encendido	Encuentra el dispositivo y lo muestra en la pantalla.	OK
CP-1-03	Intento de envío de fichero sin seleccionar dispositivo	La aplicación muestra un mensaje de error.	OK
CP-1-04	Intento de envío de fichero sin seleccionar fichero	La aplicación muestra un mensaje de error.	OK
CP-1-05	El pin introducido en el servidor y en el teléfono no son iguales	La aplicación muestra un mensaje de error.	OK
CP-1-06	Intento de envío de fichero sin que el teléfono haya levantado el servicio	La aplicación muestra un mensaje de error.	OK
CP-1-07	Intento de envío de fichero con el servicio levantado en el teléfono	La factura se recibe en el teléfono	OK

5.2.2. Pruebas de la validación del schema XML de las facturas

El objetivo de estas pruebas es confirmar que la aplicación valida correctamente el schema XML de la facturas recibidas, confirmando que se ajustan al schema definido para el formato facturae versión 3.2.

En la *Tabla 5* se muestran las pruebas definidas y el resultado obtenido tras su ejecución.

Tabla 5: Pruebas de la validación del schema XML de las facturas

Código	Descripción	Resultado esperado	Resultado
CP-2-01	Se envía una factura correcta: (Emit-11_A28000032_restauración.xsig)	La factura es aceptada por la aplicación y se consulta al usuario	OK

		sobre si desea conservarla o no.	
CP-2-02	Se envía una factura correcta: (Emit-21_A28000058_alimentación.xsig	La factura es aceptada por la aplicación y se consulta al usuario sobre si desea conservarla o no.	OK
CP-2-03	Se envía una factura incorrecta (sin número de factura -InvoiceNumber-) firmada correctamente: XAdES-EPES-Facturae-SinNumFactura.xsig	La factura es rechazada por la aplicación y se avisa al usuario	OK
CP-2-04	Se envía una factura incorrecta (sin importe total -TotalInvoicesAmount-) firmada correctamente: XAdES-EPES-Facturae-SinImpTotal.xsig	La factura es rechazada por la aplicación y se avisa al usuario	OK

5.2.3. Pruebas de la validación de la firma electrónica

En este apartado se describen las pruebas destinadas a confirmar que la aplicación valida correctamente la firma electrónica de las facturas.

En la *Tabla 6* se muestran las pruebas definidas y el resultado obtenido tras su ejecución.

Tabla 6: Pruebas de la validación de la firma electrónica

Código	Descripción	Resultado esperado	Resultado
CP-3-01	Se envía una factura correcta (Emit-2_A28000016_telefonía.xsig)	La factura es aceptada por la aplicación y se consulta al usuario sobre si desea conservarla o no.	OK
CP-3-02	Se envía una factura correcta (Emit-28_A28000073_ocio.xsig	La factura es aceptada por la aplicación y se consulta al usuario sobre si desea conservarla o no.	OK
CP-3-03	Se envía una factura correctamente firmada con certificado caducado (XAdES-EPES-Facturae-caducado.xsig)	La factura es rechazada por la aplicación y se avisa al usuario.	OK
CP-3-04	Se envía una factura firmada sin especificar política con certificado correcto (XAdES-BES-Factura.xsig)	La factura es rechazada por la aplicación y se avisa al usuario.	OK
CP-3-05	Se envía una factura firmada correctamente pero modificada con posterioridad (XAdES-EPES-Facturae-modificada.xsig)	La factura es rechazada por la aplicación y se avisa al usuario.	OK
CP-3-06	Se envía una factura firmada correctamente pero modificada con posterioridad (XAdES-EPES-Facturae-modificada2.xsig)	La factura es rechazada por la aplicación y se avisa al usuario.	OK
CP-3-07	Se envía una factura incorrecta (sin importe total -TotalInvoicesAmount-) firmada con certificado de pruebas (XAdES-EPES-Facturae-SinImpTotal-NoConfianza.xsig)	La factura es rechazada por la aplicación y se avisa al usuario.	OK
CP-3-08	Se envía una factura incorrecta (sin importe total -TotalInvoicesAmount-) firmada con certificado caducado (XAdES-EPES-Facturae-SinImpTotal-caducado.xsig)	La factura es rechazada por la aplicación y se avisa al usuario.	OK
CP-3-09	Se envía una factura firmada correctamente pero con un certificado que no es de confianza (XAdES-EPES-Facturae-NoConfianza.xsig)	La factura es rechazada por la aplicación y se avisa al usuario.	OK

5.2.4. Pruebas de la clasificación, consulta y borrado de facturas

En este apartado se describen las pruebas destinadas a comprobar el correcto funcionamiento de la aplicación en la gestión de las facturas: clasificación, consultas y facturas.

Para confirmar que la aplicación gestiona adecuadamente las facturas y sus categorías se han ejecutado las pruebas definido para ello, obteniéndose los resultados de la *Tabla 7.* Con estas pruebas nos aseguramos de que la aplicación realiza adecuadamente las siguientes operaciones: altas y borrados de categorías, visualización de facturas, consulta de facturas (por nombre o NIF del vendedor, por fecha y por categoría), clasificación de facturas (asignar una categoría) y borrado de facturas.

Tabla 7: Pruebas de la clasificación, consulta y borrado de facturas

Código	Descripción	Resultado esperado	Resultado
CP-4-01	Se envían tres facturas al teléfono: Emit-6_A28000024_ropa.xsig Emit-16_A28000040_ocio.xsig, Emit-21_A28000058_alimentación.xsig	Tras seleccionar la opción de Gestionar facturas aparecen en el listado y podemos consultar su detalle.	OK
CP-4-02	Se dan de alta las siguientes categorías: Alimentación, Ocio, Ropa, Restauración, Telefonía, Artículos cosméticos, y Medicamentos	Tras seleccionar la opción de Gestionar categorías aparece un listado de las nuevas categorías.	OK
CP-4-03	Se borra la categoría telefonía	La categoría borrada desaparece del listado de categorías.	OK
CP-4-04	Se asigna una categoría a cada una de las facturas cargadas: ropa, ocio y alimentación.	Al mostrar el detalle de la factura se ve la categoría asignada.	OK
CP-4-05	Se intenta borrar la categoría alimentación	El sistema muestra un mensaje diciendo que no se puede borrar porque tiene facturas asociadas.	OK
CP-4-06	Se consultan las facturas clasificadas de la categoría alimentación	La aplicación muestra la factura Emit-21.	OK
CP-4-07	Se consultan las facturas correspondientes a los Grandes Almacenes	La aplicación muestra la factura Emit-6.	OK
CP-4-08	Se consultan las facturas correspondientes a los Grandes Almacenes de la categoría alimentación	La aplicación muestra un mensaje diciendo que no existen resultados.	OK
CP-4-09	Se consultan las facturas del año 2014	La aplicación muestra las tres facturas.	OK
CP-4-10	Se consultan las facturas de marzo del año 2014	La aplicación muestra las tres facturas.	OK
CP-4-11	Se consultan las facturas de febrero del año 2014	La aplicación muestra un mensaje diciendo que no existen resultados.	OK
CP-4-12	Se consultan las facturas correspondientes al vendedor con NIF A28000040	La aplicación muestra la factura Emit-16.	OK
CP-4-13	Se consultan las facturas correspondientes al vendedor con NIF que empiece con a28, empleando el comodín %	La aplicación muestra las tres facturas.	OK
CP-4-14	Se borra la factura del supermercado (Emit-21)	Desaparece la factura del listado.	OK

5.2.5. Pruebas del envío de facturas por correo electrónico

En este apartado se definen las pruebas destinadas a confirmar el correcto comportamiento de la aplicación en el proceso de envío de facturas por correo electrónico.

En la *Tabla 8* se muestran las pruebas definidas y el resultado obtenido tras su ejecución.

Tabla 8: Pruebas del envío de facturas por correo electrónico

Código	Descripción	Resultado esperado	Resultado
CP-5-01	Se envía la siguiente factura por correo electrónico: Emit-6_A28000024_ropa.xsig	Se recibe correctamente el mensaje en el buzón seleccionado.	OK
CP-5-02	Se envía la siguiente factura por correo electrónico: Emit-16_A28000040_ocio.xsig,	Se recibe correctamente el mensaje en el buzón seleccionado.	OK
CP-5-03	Se envía la siguiente factura por correo electrónico: Emit-21_A28000058_alimentación.xsig	Se recibe correctamente el mensaje en el buzón seleccionado.	OK

5.2.6. Pruebas de las copias de seguridad

En este apartado se definen las pruebas destinadas a comprobar el correcto funcionamiento de las distintas opciones implementadas para realizar una copia de seguridad de los datos almacenados. Se deben comprobar los procesos de copia y de restauración y, cuando sea posible, comprobar la restauración en un dispositivo distinto al que ha hecho la copia.

En la *Tabla 9* se muestran las pruebas definidas y el resultado obtenido tras su ejecución.

Tabla 9: Pruebas de las copias de seguridad

Código	Descripción	Resultado esperado	Resultado
CP-6-01	Se envían tres facturas al teléfono: Emit-6_A28000024_ropa.xsig Emit-16_A28000040_ocio.xsig, Emit-21_A28000058_alimentación.xsig Tras un periodo de tiempo se desinstala la aplicación y se vuelve a instalar	Las tres facturas están disponibles en la aplicación	OK
CP-6-02	Tras la reinstalación anterior se borran las tres facturas y se ejecuta la opción “Restaurar datos de la nube”	Las tres facturas están disponibles en la aplicación	OK
CP-6-03	Se añaden las categorías ropa, ocio y alimentación y se clasifican las tres facturas anteriores. Transcurrido un periodo de tiempo se desinstala la aplicación y se vuelve a instalar.	Las tres facturas y las tres categorías están disponibles en la aplicación	OK
CP-6-04	Tras la reinstalación anterior se borran las tres facturas y las tres categorías y se ejecuta la opción “Restaurar	Las tres facturas y las tres categorías están disponibles en la	OK

	datos de la nube"	aplicación	
CP-6-05	Se añaden a las tres facturas anteriores dos nuevas: Emit-12_A28000032_restauración.xsig Emit-27_A28000065_medicamentos.xsig Y se añaden las categorías medicamentos y restauración, clasificando las facturas en la categoría correspondiente. Se ejecuta la opción "Realizar copia en memoria externa" A continuación se ejecuta la opción "Restaurar datos de la nube".	Volvemos a tener sólo tres facturas y tres categorías	OK
CP-6-06	Partiendo de la situación anterior se ejecuta la opción "Restaurar datos de la memoria externa"	Volvemos a tener 5 facturas y 5 categorías	OK
CP-6-07	Copiamos la carpeta invoices_copy de la tarjeta SD de este teléfono a otro teléfono que tenga instalada la aplicación y ejecutamos la opción "Restaurar datos de la memoria externa".	En el segundo teléfono tenemos las 5 facturas y las 5 categorías.	OK
CP-6-08	Se ejecuta la opción "Restaurar datos de la memoria externa", pero se introduce una contraseña distinta a la que se empleó en el proceso de backup.	El sistema muestra el siguiente mensaje: "Error en el proceso de recuperación de datos. Revise la contraseña"	OK
CP-6-09	Teniendo las 5 facturas y las 5 categorías en el teléfono, ejecutamos la opción "Realizar copia mediante Bluetooth" y empleando la aplicación Java hacemos la copia en una carpeta de un ordenador. Borramos las 5 facturas del teléfono. Ejecutamos la opción "Restaurar copia mediante Bluetooth", y empleando la aplicación Java restauramos la copia desde el ordenador.	Volvemos a tener 5 facturas y 5 categorías	OK
CP-6-10	Se ejecuta la opción "Restaurar copia mediante Bluetooth", pero empelamos una contraseña distinta a la empleada en el proceso de backup.	El sistema muestra el siguiente mensaje: "Error en el proceso de recuperación de datos. Revise la contraseña"	OK
CP-6-11	En un segundo teléfono con la aplicación instalada pero sin facturas se ejecuta la opción "Restaurar copia mediante Bluetooth" y utilizando la aplicación java se le restaura la base de datos desde el ordenador.	En el segundo teléfono tenemos las 5 facturas y las 5 categorías.	OK

5.2.7. Pruebas de la gestión y control del gasto

En este apartado se muestras las pruebas definidas para comprobar el correcto funcionamiento de la generación de los gráficos de gastos y de la generación de avisos. Por un lado es necesario comprobar que al superar el límite fijado para los gastos mensuales de una categoría la aplicación genera un aviso. Además se debe comprobar que los gráficos generados se ajustan a la información almacenada.

En la *Tabla 10* se muestran las pruebas definidas y el resultado obtenido tras su ejecución.

Tabla 10: Pruebas de la gestión y control de gasto

Código	Descripción	Resultado esperado	Resultado
CP-7-01	Se carga en el teléfono la factura Emit-2_A28000016_telefonía.xsig que tiene un importe de 116,16 €. Se fija el límite mensual para la categoría "telefonía" en 100 €. Se clasifica la factura de la categoría "telefonía"	Aparece un aviso indicando que se ha superado el límite mensual.	OK
CP-7-02	Se cargan en el teléfono las factura Emit-6_A28000024_ropa.xsig y Emit-9_A28000024_ropa.xsig que tiene un importe de 90,75 € cada una. Se fija el límite mensual para la categoría "ropa" en 100 €. Se clasifican las dos facturas de la categoría "ropa"	Al clasificar la primera factura no aparece aviso, pero al clasificar la segunda aparece un aviso indicando que se ha superado el límite mensual.	OK
CP-7-03	Tas las dos operaciones anteriores se consultan los avisos recibidos.	Aparecen los dos avisos señalados anteriormente.	OK
CP-7-04	Desde la pantalla en la que se relacionan los avisos se selecciona la opción "Borrar avisos"	Desaparecen todos los avisos de la relación.	OK
CP-7-05	Desde la pantalla en la que se relacionan los avisos se selecciona la opción "Actualizar avisos""	Vuelven a aparecer los dos avisos anteriores.	OK
CP-7-06	Se cargan en el teléfono las factura siguientes: Emit-2_A28000016_telefonía.xsig Emit-3_A28000016_telefonía.xsig Emit-4_A28000016_telefonía.xsig Emit-6_A28000024_ropa.xsig Emit-9_A28000024_ropa.xsig Emit-12_A28000032_restauración.xsig Emit-14_A28000040_restauración.xsig Emit-16_A28000040_ocio.xsig Emit-27_A28000065_medicamentos.xsig Emit-28_A28000073_ocio.xsig Emit-29_A28000073_ocio.xsig Se clasifican en la categoría indicada en su nombre.	Si no especificamos ni año ni mes, en el **gráfico de tarta** por **categorías** aparecen los siguientes importes: telefonía = 319,44 ropa = 181,5 restauración = 14,30 ocio = 33,88 medicamentos= 7,28	OK
CP-7-07	Con las facturas de la prueba anterior.	Si se especifica marzo de 2014, en el **gráfico de tarta** por **categorías** aparecen los siguientes importes: telefonía = 130,68 ropa = 181,5 restauración = 14,30 ocio = 33,88 medicamentos= 7,28	OK
CP-7-08	Con las facturas de la prueba anterior.	Si no especificamos ni año ni mes, en el **gráfico de tarta** por **proveedores** aparecen los siguientes importes: FARMACIA = 7,28 CINES = 25,41 CAFES = 11,77 GRD.ALAMCENES = 181,5 TELEFONÍA = 319,44 RESTAURANTE = 11,00	OK
CP-7-09	Con las facturas de la prueba anterior.	Si se especifica marzo de 2014, en el **gráfico de tarta** por **proveedores** aparecen los siguientes importes: FARMACIA = 7,28 CINES = 25,41 CAFES = 11,77 GRD.ALAMCENES = 181,5	OK

		TELEFONÍA = 130,68 RESTAURANTE = 11,00	
CP-7-10	Con las facturas de la prueba anterior.	Si se indica año 2014 y no se especifica categoría, en el **gráfico de barras** aparecerán los siguientes importes mensuales: enero = 72,60 febrero = 116,16 marzo = 367,64	OK
CP-7-11	Con las facturas de la prueba anterior.	Si se indica año 2014 y se especifica categoría "telefonía", en el **gráfico de barras** aparecerán los siguientes importes mensuales: enero = 72,60 febrero = 116,16 marzo = 130,68	OK
CP-7-12	Con las facturas de la prueba anterior, se fijan los siguientes límites mensuales: telefonía = 100 ropa = 100 restauración = 100 ocio = 40 medicamentos= 20 Además se añaden un par de categorías sin facturas: alimentación y productos cosméticos	En el desplegable de categorías del **gráfico de líneas** con los gastos anuales del año 2014, las categorías aparecerán en los siguientes colores: telefonía = rojo ropa = rojo restauración = verde ocio = amarillo medicamentos= verde alimentación y productos cosméticos = gris	OK
CP-7-13	Partiendo de la situación de la prueba anterior, en el gráfico de líneas seleccionamos el año 2014 y la categoría "telefonía".	En el **gráfico de líneas** se muestra en azul una serie con el valor constante 100 y otra serie en rojo con los valores: enero = 72,60 febrero = 116,16 marzo = 130,68	OK
CP-7-14	Partiendo de la situación de la prueba anterior, en el gráfico de líneas seleccionamos el año 2014 y la categoría "ocio".	En el **gráfico de líneas** se muestra en azul una serie con el valor constante 40 y otra serie en amarillo con el valor: marzo = 33,88	OK
CP-7-15	Partiendo de la situación de la prueba anterior, en el gráfico de líneas seleccionamos el año 2014 y la categoría "medicamentos".	En el **gráfico de líneas** se muestra en azul una serie con el valor constante 20 y otra serie en verde con el valor: marzo = 7,28	OK

6. Conclusiones

Se ha desarrollado una aplicación para teléfonos con S.O. Android que permite gestionar facturas electrónicas en formato facturae. Dichas facturas se envían al teléfono mediante Bluetooth.

Cuando la aplicación recibe una factura en formato facturae, su primera tarea es validar su autenticidad e integridad. Comprueba que se ajusta el schema XML definido, que está firmada por un emisor de confianza, que la firma electrónica es correcta y que no ha sido modificada con posterioridad a su emisión.

Las facturas correctas son almacenadas en una base de datos, dando la posibilidad al usuario de clasificarlas según unas categorías definidas por él mismo. Al definir las categorías, el usuario puede fijar un límite de gasto mensual. Dicho límite se puede modificar en cualquier momento. Si al clasificar una factura se supera el límite mensual establecido, la aplicación muestra un aviso indicando el límite y el gasto mensual acumulado para esa categoría.

El usuario dispone de una utilidad de búsqueda que le permite localizar una factura por emisor, fecha y categoría.

La aplicación proporciona al usuario la posibilidad de hacer un seguimiento de sus gastos a lo largo del tiempo mediante varios gráficas (diagramas de tarta, de barras y de líneas). Además la aplicación permite al usuario ver los gastos realizados a través del tiempo, clasificados por categorías o emisores.

Mediante la aplicación es posible enviar por correo electrónico las facturas almacenadas al destinatario que se indique.

Para asegurar la información y facilitar, en su caso, el cambio de dispositivo, la aplicación proporciona distintas formas de realizar copia de seguridad de los datos almacenados: copias en la nube, copias cifradas en la memoria externa y copias cifradas en un ordenador mediante Bluetooth.

La aplicación es fácil de usar, permite tener organizadas las facturas electrónicas recibidas y realizar un seguimiento y control de los gastos efectuados.

Se han alcanzado los objetivos planteados en el plazo fijado, aunque en cuanto a la planificación, en algunas tareas se ha invertido más tiempo del planificado y en otras menos.

6.1. Trabajos futuros

La aplicación podría ser complementada y mejorada con las siguientes líneas de trabajo:

- Para la comunicación entre el servidor de facturas y el teléfono móvil se podría emplear NFC, en lugar de Bluetooth.

- Implementar la recepción, validación y almacenamiento de otros formatos de factura electrónica.

- Mejorar la validación del estado de revocación de los certificados empleados en las firmas electrónicas, incluyendo los certificados de toda la cadena de certificación.

- Actualizar las Autoridades de Certificación de confianza.

- Mejorar la gestión de los acentos en las descripciones de las categorías. Aunque la aplicación no distingue entre mayúsculas y minúsculas (“OCIO” y “ocio” son la misma categoría), sí distingue entre vocales acentuadas y no acentuadas por lo que se consideran dos categorías distintas “alimentación” y “alimentacion”.

- Analizar las posibilidades que existen para establecer comunicación SSL/TLS en las copias de seguridad en la nube.

- Añadir nuevos controles de gastos semanales y diarios.

- Desarrollar nuevos gráficos con valores de los últimos 12 meses en lugar de por años naturales.

- Incluir nuevos avisos cuando se supere en un porcentaje la media de gastos mensuales.

6.2. Opinión personal

Al iniciar este TFM tenía en mente dos objetivos: profundizar en conocimientos adquiridos durante mis estudios y en mi trabajo, como son la factura y la firma electrónica, e iniciarme en el S.O. Android, tan ampliamente utilizado y para el que no había tenido la oportunidad de desarrollar hasta el momento.

Puedo concluir que he alcanzado mis objetivos personales en el desarrollo del TFM. Me ha supuesto un gran esfuerzo, ya que he tenido que adquirir los conocimientos necesarios a la vez que avanzaba el desarrollo, pero ha merecido la pena pues he disfrutado durante el proceso de investigación y aprendizaje.

7. Glosario

Autoridad de Certificación (AC o CA de las siglas en inglés): Entidad de confianza responsable de emitir y revocar certificados electrónicos utilizados en la firma electrónica, empleando criptografía de clave pública. La CA debe confirmar la veracidad de los datos del titular del certificado antes de su emisión.

Autoridad de Sellado de Tiempo (TSA de las siglas en inglés): Entidad de confianza encargada de emitir sellos de tiempo que permiten confirmar que un documento electrónico fue generado con anterioridad a una fecha determinada. Un sello de tiempo es un documento electrónico firmado por la TSA.

Certificado electrónico: Documento firmado electrónicamente por una Autoridad de Certificación que contiene ciertos datos identificativos del titular del certificado y su clave pública (criptografía asimétrica).

Certificado electrónico revocado: Certificado no válido a pesar de encontrarse dentro de su periodo de vigencia.

Factura electrónica: Según la definición recogida en el servidor Web de la Agencia Tributaria [40] es un "Documento tributario generado por medios informáticos en formato electrónico, que reemplaza al documento físico en papel, pero que conserva el mismo valor legal con unas condiciones de seguridad no observadas en la factura en papel. En términos informáticos, consiste en un fichero con el contenido exigido por ley a cualquier factura, que se puede transmitir de emisor a receptor por medios telemáticos (de un ordenador a otro) y que posee unas características que aseguren la autenticidad e integridad.

Facturae: Formato de factura electrónica definido por la Administración Española. Es un formato estructurado (documento XML) que debe ajustarse a un schema XML definido y a las especificaciones indicadas en su Política de Firma. La última versión publicada es la 3.2.1

Firma electrónica: Según Ley 59/2003, de 19 de diciembre, de firma electrónica, "la firma electrónica es el conjunto de datos en forma electrónica, consignados junto a otros o asociados con ellos, que pueden ser utilizados como medio de identificación del firmante" [39]. Una forma de conseguir una firma electrónica es empleando criptografía asimétrica o de clave pública. Para ello se cifran los datos a firmar con la clave privada del firmante. Para hacer más ligera la operación de firma, en lugar de cifrar todos los datos, se calcula un resumen (hash) de los datos a firmar y es éste el que se firma.

Lista de certificados revocados (CRL de las siglas en inglés): Documento firmado electrónicamente por una Autoridad de Certificación que recoge los certificados revocados antes de finalizar su vigencia.

Protocolo en línea del estado de revocación de certificados (Siglas en ingles OCSP): Protocolo que define el mecanismo para consultar a una CA el estado de revocación de un certificado. Ante una solicitud, la CA genera un documento firmado electrónicamente indicando si el certificado se encuentra revocado.

8. Siglas y Acrónimos

AC: Autoridad de Certificación

ADT: Android developer tools (Herramientas para desarrolladores de Android).

AGE: Administración General del Estado.

API: Application Programming Interface (Interfaz para la programación de aplicaciones)

ASL: Apache Software License (Licencia de software de Apache).

CRL: Certificate Revocation List (Listas de Revocación de Certificados)

EPES: Explicit Policy Electronic Signatures (Firmas electrónicas basadas en política explícita)

FNMT: Fábrica Nacional de Moneda y Timbre.

GPL: General Public License (Licencia pública general).

IDE: Integrated development environment (Entorno de desarrollo integrado).

JCA: Java Cryptography Architecture (Arquitectura Criptográfica de Java)

JVM: Java Virtual Machine (Máquina virtual de Java).

LOPD: Ley Orgánica de Protección de Datos.

MINETUR: Ministerio de Industria, Energía y Turismo.

MINHAP: Ministerio de Hacienda y Administraciones Públicas.

MITyC: Ministerio de Industria, Turismo y Comercio (Actualmente MINETUR)

OCSP: Online Certificate Status Protocol (Protocolo en línea del estado de revocación de certificados).

PBE: Password Based Encryption (Cifrado basado en contraseña)

SDK: software development kit (kit para el desarrollo de aplicaciones).

S.O.: Sistema Operativo.

TFM: Trabajo de Fin de Máster.

TSA: Time Stamp Authority (Autoridad de Sellado de Tiempo)

URL: Uniform resource locator (Localizador uniforma de recurso)

UUID: Universally unique identifier (identificador único universal)

XADES: XML Advanced Electronic Signatures (Firma electronica avanzada XML).

9. Bibliografía

[1] Ley 56/2007, de 28 de diciembre, de Medidas de Impulso de la Sociedad de la Información (https://www.boe.es/buscar/doc.php?id=BOE-A-2007-22440)

[2] La factura electrónica. Manuales Plan Avanza (http://www.facturae.es/es-ES/Aspectos/Manuales%20Plan%20Avanza/1Manual_%20Facturae_Ed2010.pdf)

[3] ORDEN PRE/2971/2007, de 5 de octubre, sobre la expedición de facturas por medios electrónicos cuando el destinatario de las mismas sea la Administración General del Estado u organismos públicos vinculados o dependientes de aquélla y sobre la presentación ante la Administración General del Estado o sus organismos públicos vinculados o dependientes de facturas expedidas entre particulares (https://www.boe.es/diario_boe/txt.php?id=BOE-A-2007-18009)

[4] MINHAP-MINEUR. “Factura electrónica” http://www.facturae.gob.es/formato/Paginas/descarga-aplicaciones.aspx

[5] BBVA https://www.bbva.es/particulares/subhome/wallet/pagomovil.jsp

[6] W. Frank Ableson, Robi Sen y Chris King. Android. Guía para desarrolladores.

[7] Oracle. VirtualBox. https://www.virtualbox.org/wiki/Downloads

[8] https://code.google.com/p/android-x86/downloads/list

[9] Bobby Chan. http://www.bobbychanblog.com/2011/07/faster-android-emulator-alternative-using-virtualbox/

[10] http://developer.android.com/tools/device.html

[11] Oracle. Java SE downloads. http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html?ssSourceSiteId=otnes

[12] Eclipse. https://www.eclipse.org/downloads/

[13] MINHAP y MINETUR. Factura electrónica. http://www.facturae.es/es-ES/Descargas/DescargaAplicaciones/Paginas/descarga-aplicaciones.aspx

[14] MINHAP-MINEUR. http://sedeaplicaciones2.minetur.gob.es/FacturaE/

[15] Oracle. http://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html

[16] Wikipedia. http://es.wikipedia.org/wiki/Bluetooth

[17] Paolo Caffelli. http://etecnologia.com/gadgets/funcionamiento-bluetooth

[18] Bluecove. http://bluecove.org/

[19] MINETUR. http://oficinavirtual.mityc.es/componentes/

[20] Faro de Vigo. http://www.farodevigo.es/vida-y-estilo/tecnologia/2014/02/14/android-consolida-dominio-mercado-smartphones/967461.html

[21] Apache Software Foundation. http://logging.apache.org/log4j/1.2/

[22] http://code.google.com/p/xerces-for-android/

[23] Apache Software Foundation. http://xerces.apache.org/

[24] Moisés Fernández Blanco. Verifying electronic invoices in smartphones. http://openaccess.uoc.edu/webapps/o2/handle/10609/22163

[25] MINETUR. http://www.facturae.es/politica_de_firma_formato_facturae/politica_de_firma_formato_facturae_v3_1.pdf

[26] MINETUR. Prestadores de servicios de certificación de firma electrónica. https://sedeaplicaciones2.minetur.gob.es/prestadores/

[27] ETSI TS 101 903 V1.2.2 (2004-04). http://uri.etsi.org/01903/v1.2.2/

[28] ETSI TS 101 903 V1.3.2 (2006-03). http://uri.etsi.org/01903/v1.3.2/

[29] dalvik. Code and documentation from Android's VM team. http://code.google.com/p/dalvik/wiki/JavaxPackages

[30] Sokol Kosta. Including additional javax.* packages in Android 2.3. https://sites.google.com/site/sokolkosta/internal-blog/includingadditionaljavaxpackagesinandroid23

[31] SQLite. https://sqlite.org/

[32] http://www.phonearena.com/news/Comparison-shows-how-much-internal-storage-you-actually-get-with-popular-smartphones_id51856

[33] http://stackoverflow.com/questions/2421189/version-of-sqlite-used-in-android

[34] Android. http://developer.android.com/reference/android/support/v4/content/FileProvider.html

[35] Android. http://developer.android.com/guide/topics/data/backup.html

[36] Android. https://developer.android.com/google/backup/signup.html?csw=1

[37] Oracle. Java.

http://docs.oracle.com/javase/6/docs/technotes/guides/security/crypto/CryptoSpec.html

[38] AChartEngine. http://www.achartengine.org/index.html

[39] Ley 59/2003, de 19 de diciembre, de firma electrónica. http://www.boe.es/buscar/doc.php?id=BOE-A-2003-23399

[40] Agencia Tributaria. Gobierno de España. www.agenciatributaria.es

Otras páginas de interés

- Android. http://developer.android.com/guide/index.html
- Salvador Gómez Oliver. http://www.sgoliver.net/blog/?page_id=3011
- http://homepages.ius.edu/RWISMAN/C490/html/Android-Bluetooth.htm
- Artisan Tommi Laukkanen.

  http://www.substanceofcode.com/2008/06/20/sending-files-to-mobile-phone-using-bluetooth-and-obex/
- http://stackoverflow.com/
- Sebastian Engel.
- http://svn.verinice.org/svnroot/TRUNK/sernet.verinice.encryption/src/sernet/verinice/encryption/impl/PasswordBasedEncryption.java
- David 'Digit' Turner
- https://android.googlesource.com/platform/development/+/gingerbread/samples/

10. Anexos

10.1. Manual de Usuario de la aplicación Servidora de Facturas

La aplicación permite enviar facturas a un dispositivo móvil.

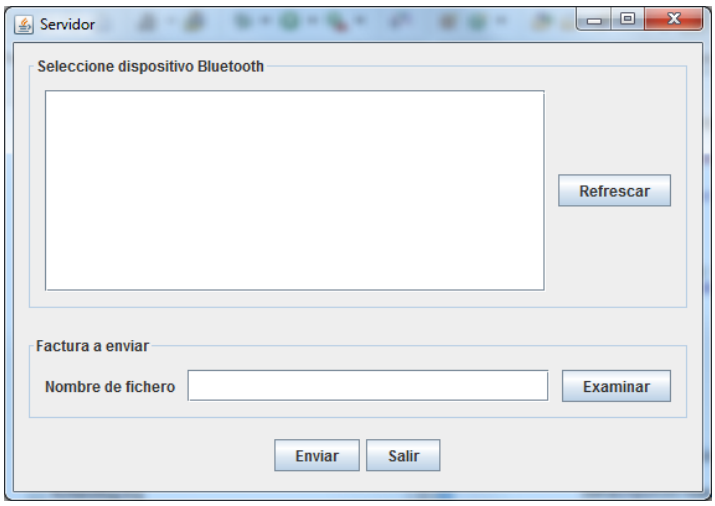

Figura 35: Aplicación servidora de facturas (1)

Mediante el botón Refrescar se buscan los dispositivos accesibles y el botón Examinar permite seleccionar el fichero al enviar.

Una vez seleccionado el dispositivo y el fichero, el botón Enviar enviará la factura.

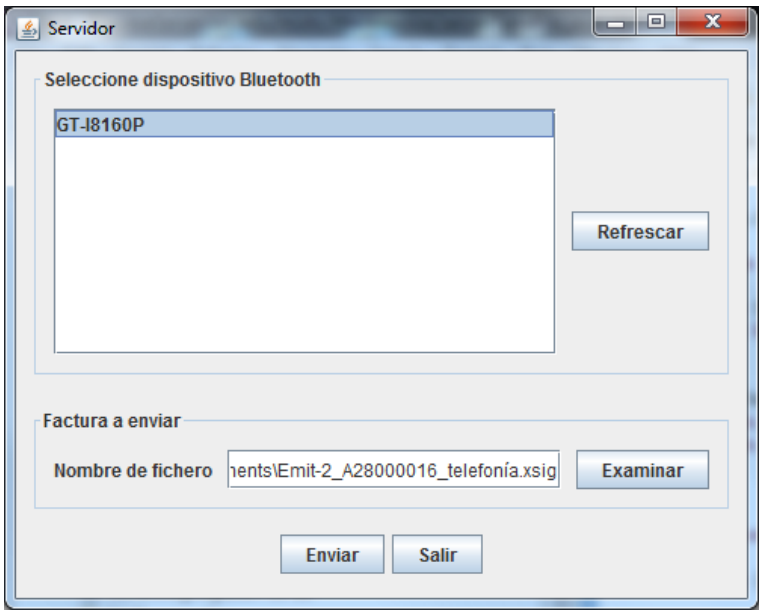

Figura 36: Aplicación servidora de facturas (2)

Si no se selecciona un dispositivo o un fichero antes de pulsar el botón Enviar, la aplicación mostrará el error correspondiente:

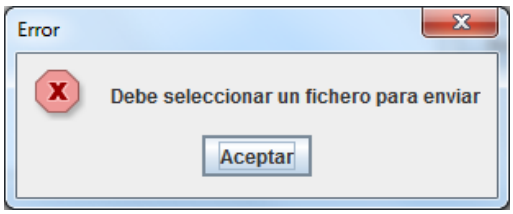

Figura 38: Aplicación servidora de facturas – Error 1

Figura 37: Aplicación servidora de facturas – Error 2

Si el teléfono móvil no está preparado para recibir facturas, es decir, no tiene levantado el servicio, la aplicación mostrará el siguiente mensaje de error:

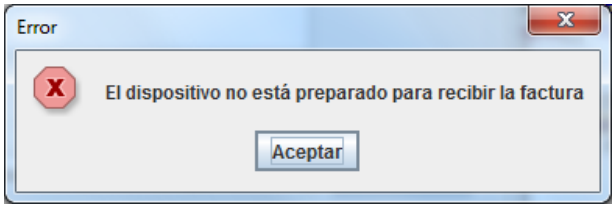

Figura 39: Aplicación servidora de facturas – Error 3

Si la factura es reciba por el teléfono móvil se mostrará el siguiente mensaje:

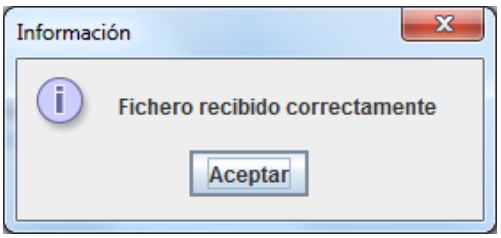

Figura 40: Aplicación servidores de facturas – Info 1

10.2. Manual de Usuario de la aplicación Java para realizar copias de seguridad mediante Bluetooth en un ordenador

La aplicación permite realizar una copia de seguridad cifrada de la base de datos en un ordenador empleando Bluetooth así como su restauración.

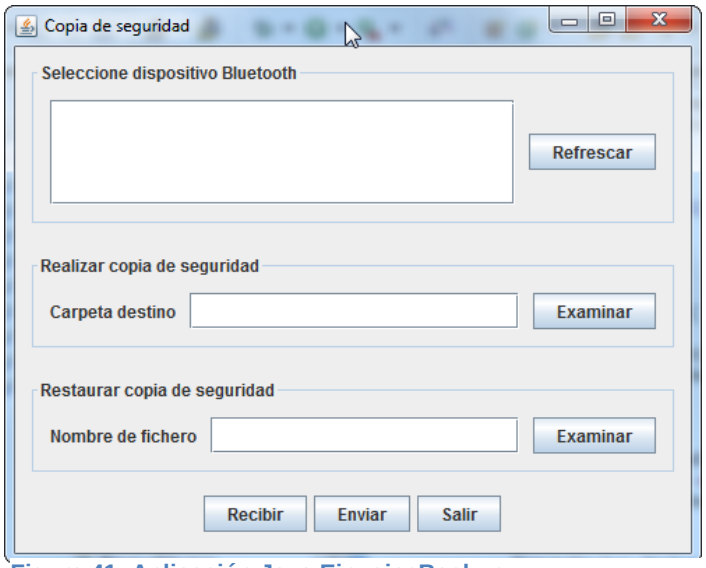

Figura 41: Aplicación Java EinvoiceBackup

Mediante el botón *Refrescar* se buscan los dispositivos accesibles. Como en toda comunicación Bluetooth el teléfono y el ordenador deberán intercambiar un pin la primera vez que establezcan comunicación.

Para realizar una copia de seguridad, en el teléfono se selecciona la opción correspondiente, se introduce la contraseña para cifrar la información y se levanta el servicio, quedando a la espera de que un cliente Bluetooth se ponga en contacto.

Figura 42: Contraseña de cifrado

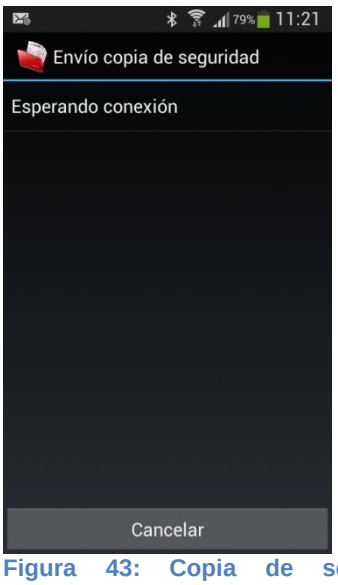

Figura 43: Copia de seguridad empleando Bluetooth

Desde la aplicación Java se selecciona el dispositivo, la carpeta destino de la copia y se pulsa el botón *Recibir*.

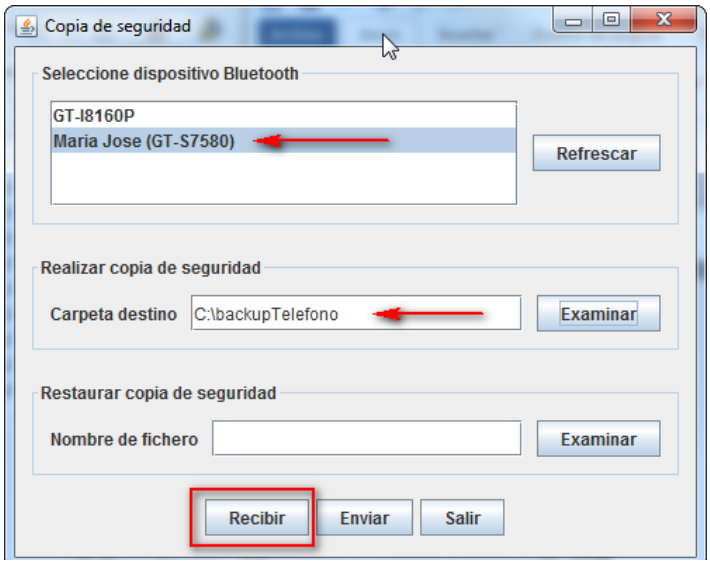

Figura 44: Aplicación Java. Realizar copia de seguridad

Si la transferencia termina correctamente aparece el siguiente mensaje en el ordenador.

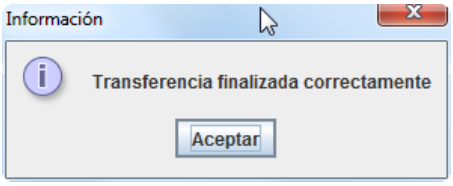

Figura 45: Aplicación Java. Mensaje backup OK

Y en el teléfono también se informa que el proceso ha terminado correctamente.

Figura 46: Teléfono. Mensajes backup OK

Para restaurar una copia de seguridad, en el teléfono se selecciona la opción correspondiente, se introduce la contraseña para descifrar la información y se levanta el servicio, quedando a la espera de que un cliente Bluetooth se ponga en contacto. La contraseña debe coincidir con la que se empleó durante el proceso de copia.

Desde la aplicación Java se selecciona el dispositivo y el fichero que almacena la base de datos cifrada que se desea restaurar y a continuación se pulsa el botón *Enviar*.

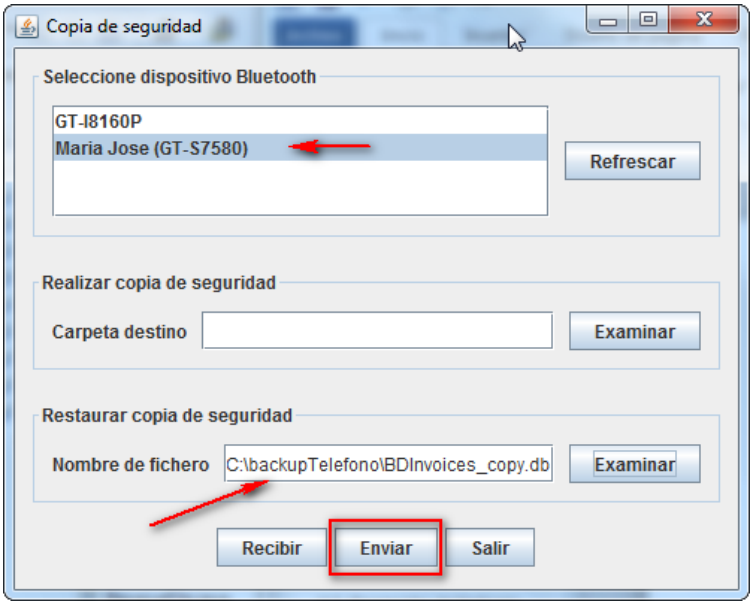

Figura 47: Aplicación Java. Restaurar copia de seguridad

Si la transferencia termina correctamente aparece el siguiente mensaje en el ordenador.

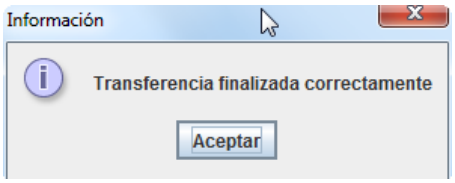

Figura 48: Aplicación Java. Mensaje restauración OK

Y en el teléfono también se informa que el proceso ha terminado correctamente.

Figura 49: Teléfono. Mensaje restauración OK

Si al realizar la copia o la restauración no se selecciona un dispositivo, la aplicación mostrará el siguiente error.

Figura 50: Aplicación Java. Error1

Si al realizar una copia o una restauración no se selecciona la carpeta destino o el fichero a restaurar, respectivamente, aparecerá uno de los siguientes mensajes.

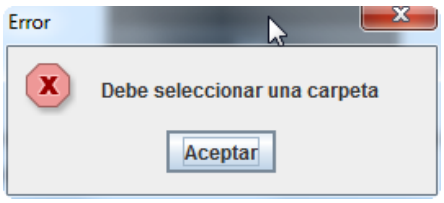

Figura 51: Aplicación Java. Error3

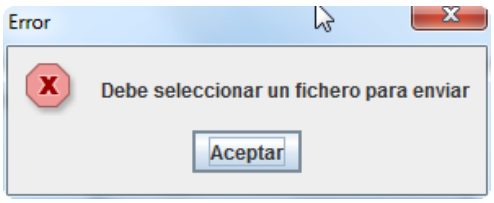

Figura 52: Aplicación Java. Error2

Si se intenta hacer el envío o la recepción del fichero de copia de seguridad y el teléfono no ha levantado el servicio correspondiente, aparece uno de los siguientes mensajes.

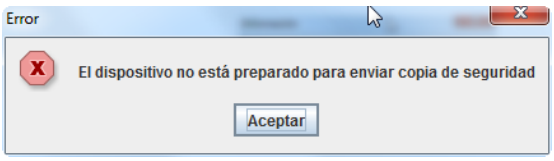

Figura 54: Aplicación Java. Error5

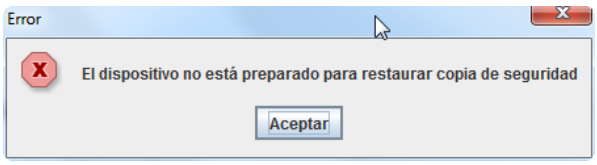

Figura 53: Aplicación Java. Error4

10.3. Manual de Usuario de la aplicación del teléfono móvil

Al iniciar la aplicación aparece un menú, como se muestra en la *Figura 56*.

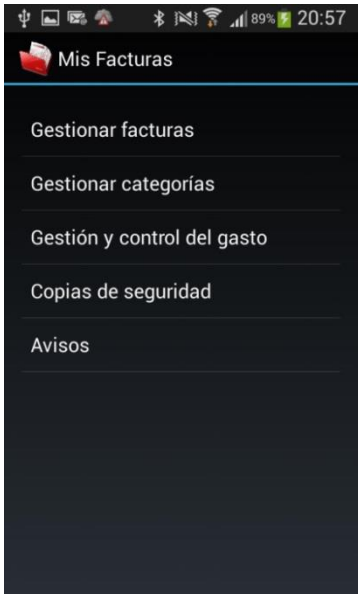

Figura 56: Aplicación – Menú inicial (2)

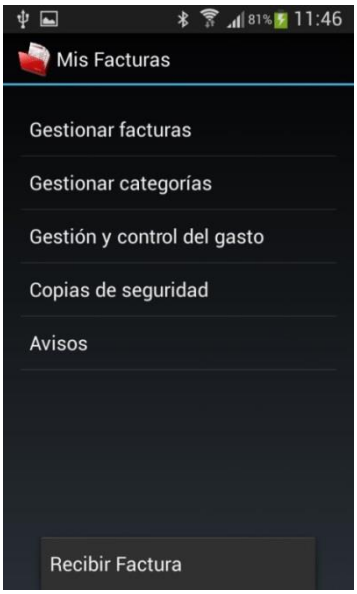

Figura 55: Aplicación – Menú inicial (1)

Esta pantalla dispone de un menú con la opción *Recibir Factura*, que se explicará más adelante, pues también está presente en la opción *Gestionar Facturas*.

1. Opción Gestionar facturas

Desde esta opción se puede visualizar facturas, clasificarlas (asignar categoría), buscarlas, borrarlas, enviarlas por correo electrónico y recibir nuevas facturas.

Tras pulsar la opción aparece un listado con las facturas almacenadas en el teléfono, ordenadas por proveedor, indicando proveedor, fecha e importe (ver *Figura 58*).

Esta pantalla dispone de un menú con las siguientes opciones, como se muestra en la *Figura 57*:

- Recibir factura
- Buscar factura
- Ordenar por vendedor
- Ordenar por fecha

Las dos últimas opciones permiten cambiar el orden del listado: por proveedor o por fecha de factura.

Figura 58: Aplicación – Listado facturas (2)

Figura 57: Aplicación – Listado facturas (1)

La opción ***Recibir factura*** levanta el servicio Bluetooth y se queda a la espera de recibir una factura (ver *Figura 59*). Si no se ha realizado con anterioridad, será necesario intercambiar un pin entre los dos dispositivos que van a participar en la comunicación Bluetooth (ver *Figura 60*). La espera se puede cancelar en cualquier momento pulsando el botón *Cancelar*. Si se abandona la pantalla también se finalizará el servicio.

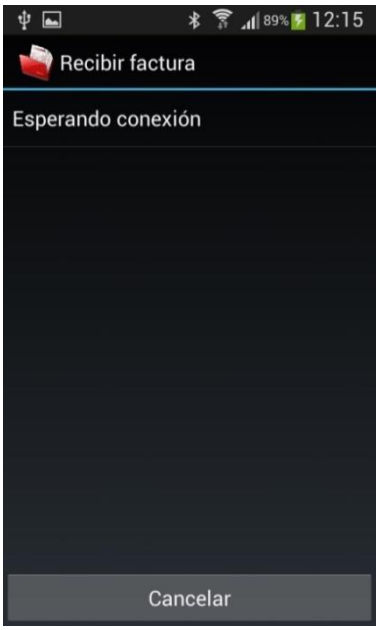

Figura 59: Aplicación – Aceptar factura (1)

Figura 60: Aplicación – Vinculación Bluetooth

Tras recibir la factura, la aplicación analiza si es correcta (schema XML y firma electrónica). Si es correcta, se muestran los datos de la factura y se da la posibilidad de aceptarla, con lo que se añadirá a la base de datos, o borrarla (ver *Figura 61*).

Figura 62: Aplicación – Aceptar factura (3)

Figura 61: Aplicación – Aceptar factura (2)

Si al analizar la factura se detecta algún problema la aplicación lo muestra al usuario. Si el error está relacionado con el schema XML no se da ninguna opción al usuario. La factura recibida es borrada. En el resto de los casos (problemas con la firma electrónica) se le da la opción al usuario de borrar o no la factura, aunque en ningún caso se incorpora a la base de datos.

Figura 63: Aplicación – Detección de facturas erróneas

La opción de **Buscar facturas** muestra la pantalla de la Figura 65.

Figura 65: Aplicación – Buscar factura (2)

Figura 64: Aplicación – Buscar factura (1)

En el NIF del vendedor se pueden utilizar el comodín %. Si no se emplea, se busca un NIF idéntico al introducido (no distingue entre mayúsculas y minúsculas). En el campo vendedor también se puede utilizar el comodín, pero en cualquier caso se realiza una búsqueda introduciendo el comodín al principio y al final. El formato de las fechas de las facturas es AAAA-MM-DD, por lo que para buscar las facturas de un año sólo se debe poner el año y para buscar las facturas de un mes se pone el año y el mes. En el campo de categorías se muestra un desplegable con las categorías disponibles.

Al pulsar el botón *Buscar* la aplicación muestra un listado con las facturas que cumplen el criterio de búsqueda especificado.

Si en el listado de las facturas pulsamos encima de cualquiera de ellas vemos toda su información, como se muestra en *Figura 66*. Los signos ¿? se muestran en el campo categoría cuando aún no se ha clasificado al factura. Desde la pantalla de detalle de una factura podemos acceder a un menú con las siguientes opciones (ver *Figura 67*):

- Borrar factura
- Recibir factura
- Clasificar factura
- Enviar factura por correo

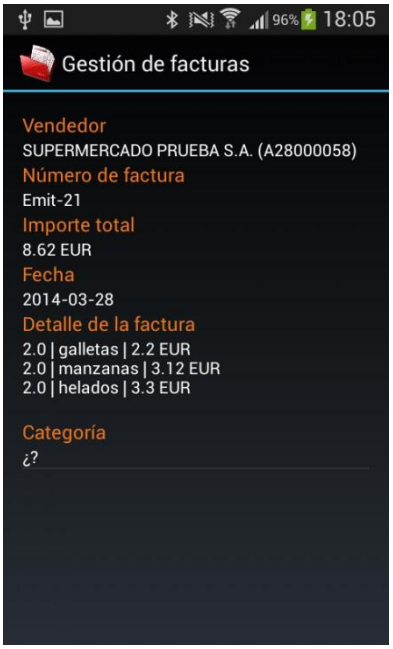

Figura 66: Aplicación – Detalle factura (2)

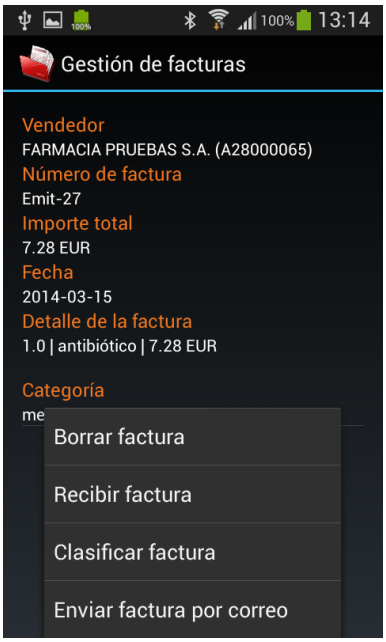

Figura 67: Aplicación – Detalle factura (1)

La opción ***Borrar factura*** borra la factura que estamos visualizando.

La opción ***Recibir factura*** ya se ha explicado anteriormente

La opción ***Clasificar Factura*** permite asignar una categoría a la factura. Al pulsar esta opción aparece la pantalla que se muestra en *Figura 68*.

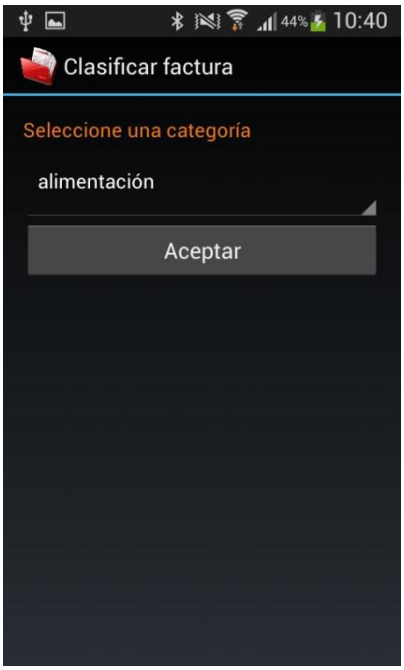

Figura 68: Aplicación – Clasificar factura

Si al pulsar el botón *Aceptar*, tras seleccionar la categoría, con el importe de la factura se supera el límite mensual para esa categoría, la aplicación lo indicará mostrando un aviso como el de la *Figura 69*.

Figura 69: Aplicación – Aviso límite mensual superado

La opción ***Enviar factura por correo*** permite enviar la factura seleccionada por correo electrónico. Tras seleccionar la opción aparece la pantalla de la *Figura 71* para que se cumplimente el destinatario, el asunto y el cuerpo del mensaje. Al pulsar el botón *Enviar* la aplicación muestra los clientes de correo disponibles para que se seleccione el que se desee emplear (ver *Figura 70*).

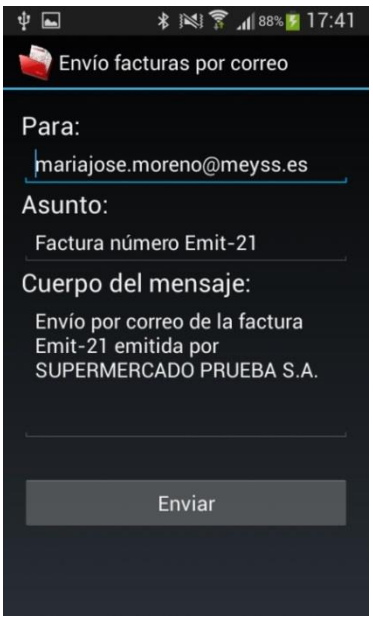

Figura 71: Aplicación – Envío de factura por correo (2)

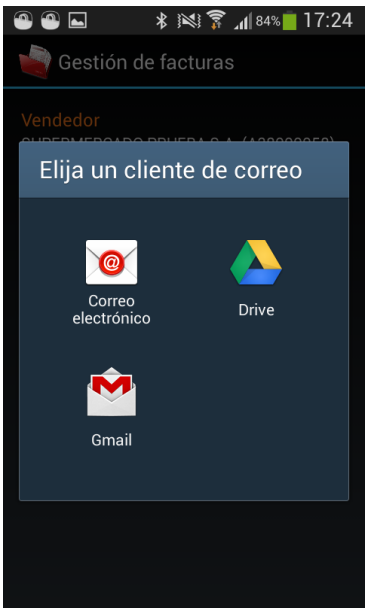

Figura 70: Aplicación – Envío de factura por correo (1)

Una vez seleccionado el cliente, la información del envío (destinatario, asunto, cuerpo y anexos) son enviados a dicho cliente desde donde se finaliza el proceso.

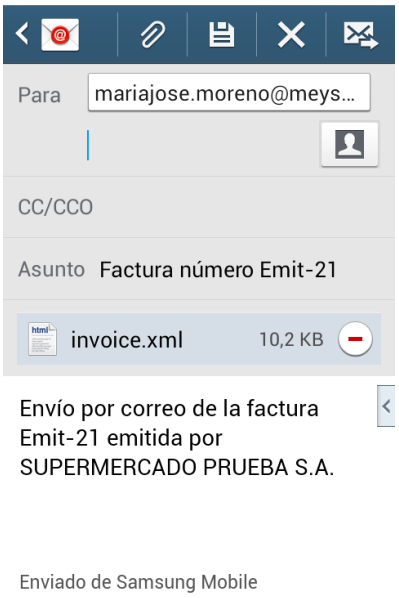

Figura 72: Aplicación – Envío de factura por correo (3)

2. Opción Gestionar categorías

Esta opción permite añadir, borrar y modificar categorías así como buscar las facturas clasificadas en una categoría determinada. Tras seleccionar la opción la aplicación muestra un listado con las categorías disponibles como en la *Figura 74*. Esta pantalla dispone de un menú con la opción Añadir, que permite dar de altas nuevas categorías (ver *Figura 73*).

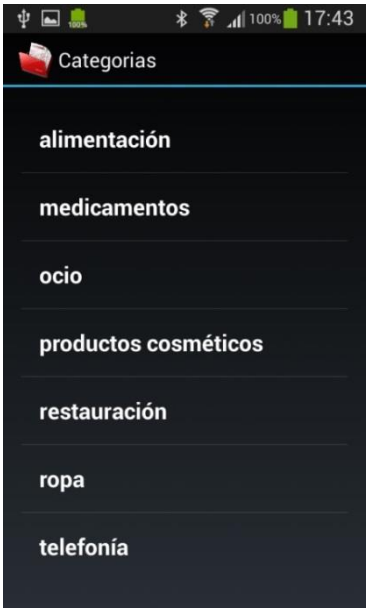

Figura 74: Aplicación – Lista categorías (2)

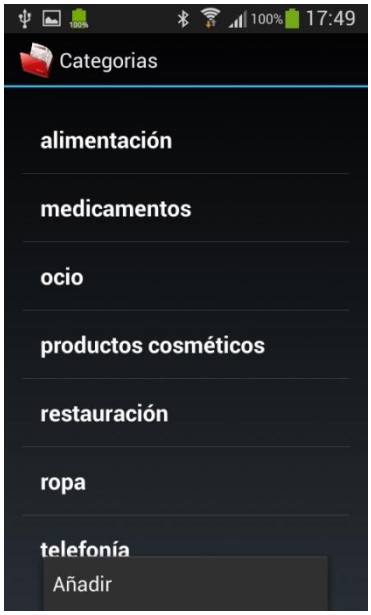

Figura 73: Aplicación – Lista categorías (1)

Figura 76: Aplicación – Alta categoría

Figura 75: Aplicación – Gestión categorías

Al pulsar la opción ***Añadir*** aparece la pantalla de la *Figura 76* en la que se debe indicar la descripción de la categoría y el importe del límite mensual. La aplicación no permite duplicar una categoría.

Pulsando encima de una categoría se muestra un menú contextual con las opciones de la *Figura 75*.

La opción ***Borrar*** elimina la categoría, siempre que no tenga facturas asignadas.

La opción ***Modificar*** permite modificar la descripción y el límite mensual.

La opción ***Mostrar facturas*** presenta un listado con las facturas que se han clasificado dentro de la categoría seleccionada.

3. Opción Gestión y control del gasto

Esta opción permite ver distintos gráficos con los gastos de las facturas almacenadas, mostrando una visión general de los mismos, por categorías, en un periodo de tiempo. Las opciones disponibles se muestran en la *Figura 77*.

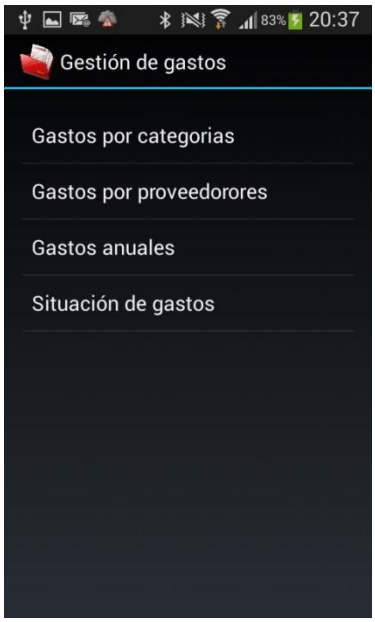

Figura 77: Aplicación – Gestión y control del gastos

La opción ***Gastos por categorías*** muestra un diagrama de tarta con los gastos realizados en las distintas categorías. Se puede especificar un año y un mes. Si no se indican, el diagrama refleja todos los gastos realizados (ver *Figura 78* y *Figura 79*). Pulsando encima de cualquiera de los segmentos es posible ver el valor correspondiente. Los colores se seleccionan aleatoriamente, y si no nos gustan bastará con volver a pulsar el botón *Ver*, para que cambien.

Figura 79: Aplicación – Gastos por categorías (2)

Figura 78: Aplicación - Gastos por categorías (1)

La opción ***Gastos por proveedores*** muestra un diagrama de tarta con los importes pagados a los distintos proveedores. Se puede especificar un año y un mes. Si no se indican, el diagrama refleja todos los gastos realizados (ver *Figura 80*). El funcionamiento es similar al diagrama anterior. Si no se ven los literales de los proveedores, es posible desplazar la gráfica como se muestra en la *Figura 81*.

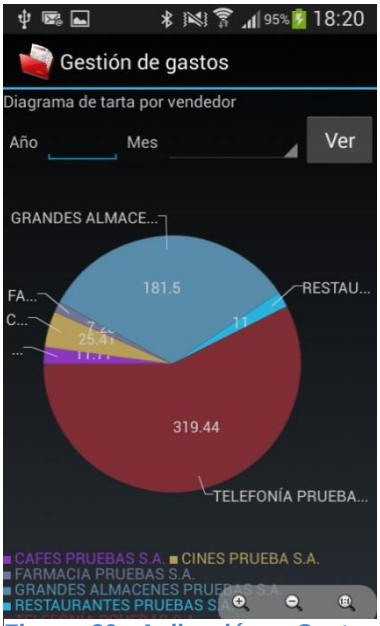

Figura 80: Aplicación - Gastos por proveedores (2)

Figura 81: Aplicación - Gastos por proveedores (2)

La opción ***Gastos anuales*** muestra un diagrama de barras con los gastos del año que se le indique. Si se le especifica una categoría, el diagrama muestra los gastos realizados cada uno de los meses en artículos de esa categoría. Si no se indica una categoría se muestra el total de los gastos de cada mes (ver *Figura 82* y *Figura 83*).

Figura 83: Aplicación – Gastos anuales (2)

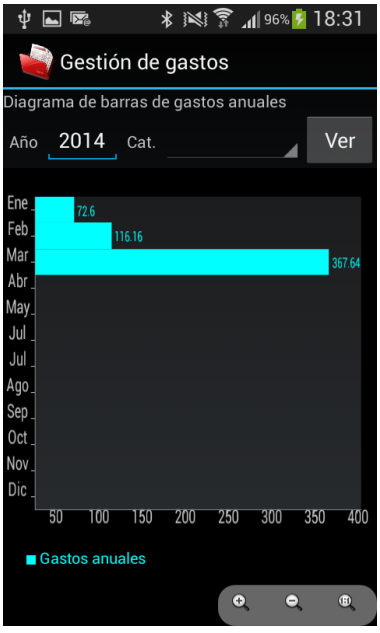

Figura 82: Aplicación – Gastos anuales (1)

La opción ***Situación de gastos*** muestra un diagrama de líneas con dos series. Para cada año y categoría la aplicación muestra una serie con el límite mensual y otra con los gastos realizados cada mes en esa categoría. Tras seleccionar el año, la aplicación actualiza el desplegable con las categorías, marcándolas con un color, con el siguiente significado:

- gris: no se han realizado gasto de esta categoría.
- rojo: se ha superado el límite mensual durante algún mes.
- amarillo: durante algún mes se ha estado próximo al límite (entre 80-100% del límite) pero no se ha superado.
- verde: en el resto de los casos.

Figura 84: Aplicación – Situación de gastos (2)

Figura 85: Aplicación – Situación de gastos (1)

Tras seleccionar una categoría y al pulsar el botón *Ver* aparece la siguiente gráfica con dos series:

- En azul el límite mensual
- En el color del desplegable los gastos mensuales de la categoría seleccionada

En la *Figura 85*, *Figura 86* y *Figura 87* se muestra un ejemplo de cada color.

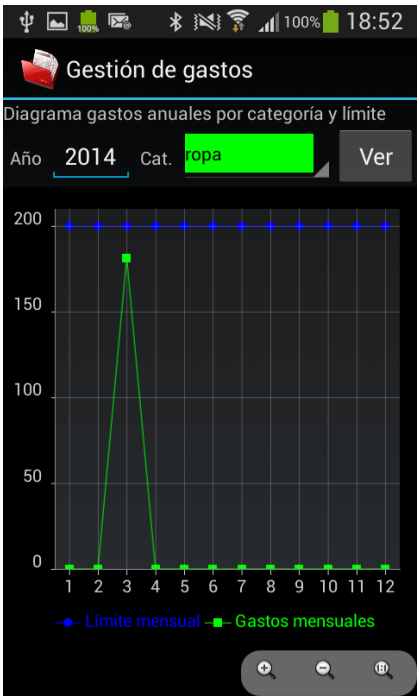

Figura 86: Aplicación – Situación de gastos (3)

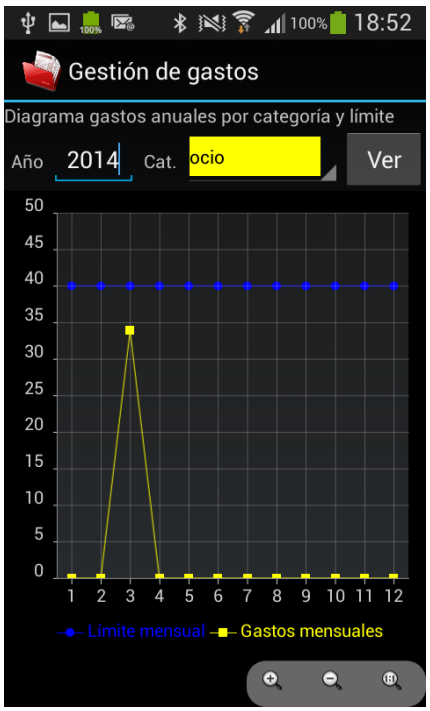

Figura 87: Aplicación – Situación de gastos (4)

4. Opción Copias de seguridad

Esta opción permite realizar copias de seguridad en distintas ubicaciones, como se muestra en la *Figura 88*.

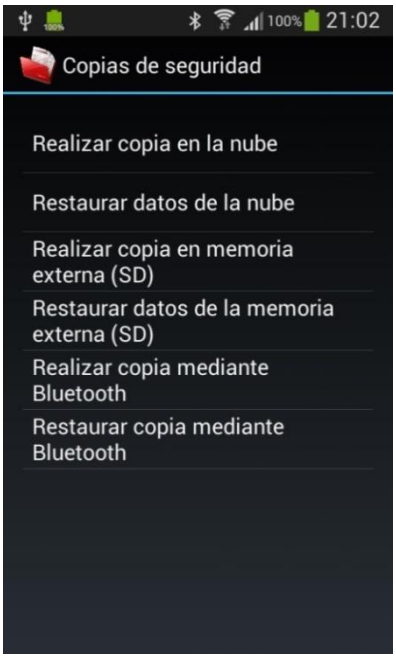

Figura 88: Aplicación – Copias de seguridad

La opción ***Realizar copia en la nube*** permite copiar los datos de la aplicación en un almacén en la nube empleando el Servicio de Backup de Android. Esta copia no se realiza inmediatamente. El Servicio de Backup la realiza cuando lo considera oportuno. Además, de forma transparente para el usuarios, la aplicación solicita al Servicio de Backup un acopia en la nube cada vez que se actualiza la información de la base de datos.

La opción ***Restaurar datos de la nube*** recupera la última copia de la base de datos almacenada en la nube. Este proceso es inmediato.

Al desinstalar la aplicación se borra la base de datos, no obstante al volverla a instalar se recupera la última copia de seguridad almacenada en la nube.

La opción ***Realizar copia en memoria externa (SD)*** permite hacer una copia cifrada en la memoria externa (tarjeta SD). Cuando se selecciona esta opción se solicita la contraseña que se empleará para cifrar la información (ver *Figura 89*). Si el proceso finaliza correctamente se mostrará la pantalla de la *Figura 90*.

Figura 89: Aplicación – Copia en memoria externa (2)

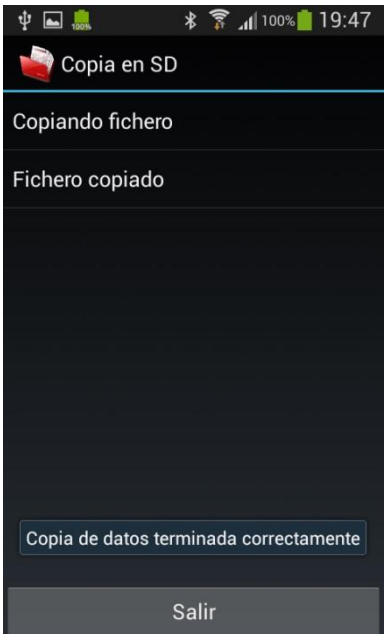

Figura 90: Aplicación – Copia en memoria externa (1)

La opción ***Restaurar datos de la memoria externa (SD)*** recuperar una copia almacenada con anterioridad en la memoria externa (tarjeta SD). Cuando se selecciona esta opción se solicita al usuario la contraseña que se empleará para descifrar la información. Si en el proceso de restauración no se indica la misma contraseña que se especificó durante el proceso de copia, la restauración no se lleva a cabo y se indica el motivo al usuario (ver *Figura 91*). Si el proceso finaliza correctamente se mostrará la pantalla de la *Figura 92*.

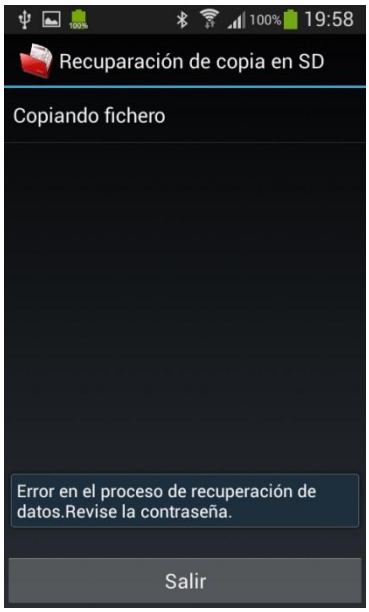

Figura 91: Aplicación – Restauración de datos de memoria externa (1)

Figura 92: Aplicación – Restauración de datos de memoria externa (2)

La opción ***Realizar copia mediante Bluetooth*** permite realizar una copia cifrada de la base de datos en un ordenador, transfiriendo los datos mediante Bluetooth. Para ello es necesario el programa explicado en el apartado anterior de este anexo.

Tras seleccionar esta opción se solicita la contraseña que se empleará para cifrar la información (ver *Figura 94*) y se levanta el servicio, quedando a la espera de que un cliente Bluetooth se ponga en contacto. Si no se ha realizado con anterioridad, será necesario intercambiar un pin entre los dos dispositivos que van a participar en la comunicación Bluetooth. Si el proceso finaliza correctamente se mostrará la pantalla de la *Figura 93*.

Figura 94: Aplicación – Copia mediante Bluetooth (1)

Figura 93: Aplicación – Copia mediante Bluetooth (2)

La opción ***Restaurar copia mediante Bluetooth*** permite recuperar los datos almacenados anteriormente en un ordenador, transfiriendo los datos mediante Bluetooth. Para ello es necesario el programa explicado en el apartado anterior de este anexo.

Tras seleccionar esta opción se solicita la contraseña que se empleará para descifrar la información (ver *Figura 94*) y se levanta el servicio, quedando a la espera de que un cliente Bluetooth se ponga en contacto. Si no se ha realizado con anterioridad, será necesario intercambiar un pin entre los dos dispositivos que van a participar en la comunicación Bluetooth. Si el proceso finaliza correctamente se mostrará la pantalla de la *Figura 95*.

Figura 95: Aplicación – Restauración mediante Bluetooth

5. Opción *Avisos*

Desde esta opción podemos consultar y borrar los avisos que se generan al superar el límite mensual de gasto fijado para las categorías. Al seleccionar esta opción aparecerá un listado con los avisos generados hasta el momento, como se muestra en la *Figura 97*. Esta pantalla dispone de un menú con dos opciones (ver *Figura 96*):

- Borrar avisos
- Actualizar avisos

Figura 97: Aplicación - Listado de avisos (2)

Figura 96: Aplicación - Listado de avisos (1)

La opción ***Borrar avisos*** elimina de la base de datos todos los avisos generados hasta el momento.

La opción ***Actualizar avisos*** revisa todos los gastos realizado hasta el momento y generar de nuevo los avisos. Con esta opción, además de los avisos por superación de límites aparecerán avisos indicando los periodos en los que no se han superado los límites, como se muestra en la *Figura 98*.

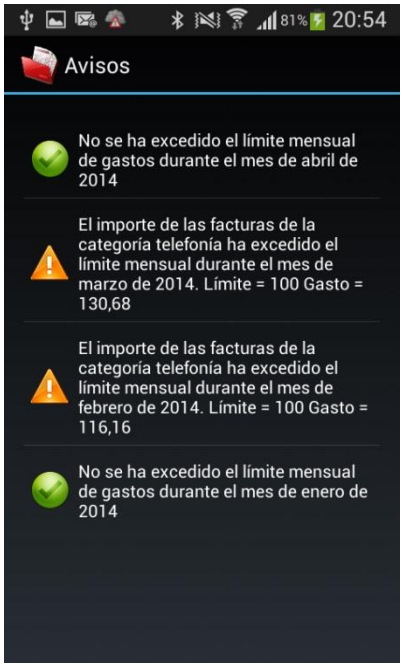

Figura 98: Aplicación - Listado de avisos (3)