Disseny i desenvolupament d'un sistema per a guardar factures electròniques de manera privada i segura al núvol

Nom Estudiant: José Luis Herranz Martín

Programa: Màster Universitari en Seguretat de les Tecnologies de la Informació i de les Comunicacions (MISTIC)

Nom Consultor: Jordi Castellà Roca

Centre: Universitat Oberta de Catalunya

Data Lliurament:
11/06/2019

FITXA DEL TREBALL FINAL

Títol del treball:	Disseny i desenvolupament d'un sistema per a guardar factures electròniques de manera privada i segura al núvol
Nom de l'autor:	*José Luis Herranz Martín*
Nom del consultor:	*Jordi Castella Roca*
Data de lliurament (mm/aaaa):	*06/2019*
Àrea del Treball Final:	*Seguretat en aplicacions web*
Titulació:	Màster Universitari en Seguretat de les Tecnologies de la Informació i de les Comunicacions (MISTIC)

Resum del Treball (màxim 250 paraules):

Les empreses estan canviant les factures en paper per factures electròniques per tal de reduir costos i millorar l'eficiència. A Espanya després de la publicació de l'ordre PRE/2971/2007 on es defineix el format de factura electrònica "Facturae", les administracions publiques obliguen als seus proveïdors a presentar les factures en aquest format. Les factures electròniques tenen un format XML i estan signades electrònicament segons el format XMLDSig Enveloped amb extensions XADES-EPES.

Aquest treball té com objectiu cobrir una necessitat com és la de validar i guardar, de manera segura i en un únic punt accessible només per l'usuari, totes les factures electròniques de les despeses que ha realitzat. A més es vol que l'usuari pugui guardar una còpia de les factures en un servei al núvol en que es garanteixi la privacitat de les dades de l'usuari de manera que pel proveïdor del servei no sigui possible saber quines són les despeses dels usuaris.

El sistema desenvolupat tindrà dos parts: una aplicació servidor que cobrirà la necessitat de guardar de manera segura les factures al núvol garantint a l'usuari la integritat, privacitat i disponibilitat de les mateixes, i una aplicació per a dispositius mòbils que permetrà gestionar, validar i guardar al núvol les factures de l'usuari.

Abstract (in English, 250 words or less):

Companies are changing paper invoices for electronic invoices in order to reduce costs and improve efficiency. In Spain after the publication of the order PRE / 2971/2007 where the electronic billing format "Facturae" is defined, public administrations oblige their suppliers to present invoices in this format. Electronic invoices have an XML format and are electronically signed in XMLDSig Enveloped format with XADES-EPES extensions.

This work aims to cover a need such as to validate and save, in a secure and in a single point accessible only to the user, all electronic invoices for the expenses that the user has made. In addition, the user could save a copy of the invoices in a cloud service that guarantees the privacy of the user's data so that it is not possible for the service provider to know what the costs of the users are.

The system developed will have two parts: a server application that will cover the need to safely store the invoices in the cloud guaranteeing the user the integrity, privacy and availability of its invoices, and an application for mobile devices that will allow to manage, validate and save the invoices of the user in the cloud.

Paraules clau (entre 4 i 8):

Factures electròniques, emmagatzemament remot, segur, privat,

Agraïments

Vull agrair la ajuda i el suport del meu tutor del TFM. Per segona vegada ens trobem en una situació com aquesta després de 12 anys. Gràcies per les seves idees, arguments i reptes.

Gràcies a totes les persones a les que he explicat de què anava el treball i quins eren els meus dubtes, i que sense saber-ho m'han respost a les meves preguntes.

I sobre tot gràcies a la meva dona i a la meva filla per tot el suport, recolzament, i ànims per poder portar a terme aquest treball. Sense elles no hagués estat possible.

Índex

Llista de figures

Llista de taules

1. Introducció

Les empreses estan canviant les factures en paper per factures electròniques per tal de reduir costos. La Llei 56/2007 [1] de 28 de desembre, de “Medidas de Impulso de la Sociedad de la Información” defineix en l’article 1 una factura electrònica com un “document electrònic que compleix amb els requisits legals i reglamentàriament exigibles a les factures i que, a més, garanteix l'autenticitat del seu origen i la integritat del seu contingut, el que impedeix el repudi de la factura pel seu emissor”. Les factures electròniques estan signades electrònicament amb la qual cosa es garanteix la autenticitat, integritat i no repudi de la mateixa.

Hi ha diversos formats de factura electrònica. A Espanya després de la publicació de l’ordre PRE/2971/2007 on es defineix el format de factura electrònica “Facturae”, les administracions publiques obliguen als seus proveïdors a presentar les factures en aquest format. Les factures electròniques tenen un format XML i estan signades electrònicament segons el format XMLDSig Enveloped amb extensions XADES-EPES.

Les empreses de més de 100 treballadors o que facturin més de 6 milions d’euros a l’any en general estan obligades per llei a emetre les factures en format electrònic. Això implicarà que en un temps no molt llunyà totes les empreses emetran factures electròniques.

Aquestes factures electròniques han de poder validar-se, és a dir, determinar que s’han emès pel proveïdor correcte (autenticitat), i que aquesta factura no ha estat manipulada (integritat). Aquesta validació s’ha de fer amb un dispositiu, sigui un ordinador, una tauleta o bé un mòbil. Donada la implantació dels mòbils aquests poden fer aquesta validació degut a la seva capacitat de càlcul.

Amb aquest treball és vol cobrir una necessitat com és la de validar i guardar, de manera segura i en un únic punt accessible només per l’usuari, totes les factures electròniques de les despeses que ha realitzat.

Actualment aquestes dades es guarden en paper, si la compra es fa en un establiment, o bé en cadascun dels comerços online o web d’empreses subministradores de serveis, de manera que per a poder tenir una visió global de les seves despeses l’usuari ha d’entrar amb usuari i contrasenya en cadascuna d’elles, descarregar-se les factures, homogeneïtzar formats i posteriorment fer l’anàlisi que necessiti.

En aquest treball es vol crear una eina (aplicació mòbil i servei al núvol) que permeti gestionar les factures electròniques en un format homogeni, i tenir la possibilitat de pujar-les a un servei al núvol de manera que sempre les tinguem disponibles, tot evitant que el proveïdor tingui la possibilitat de saber què i en què ens gastem els diners, ja que les dades es pujaran degudament protegides (xifrades).

1.1 Objectius del Treball

L'objectiu del treball és dissenyar i implementar un sistema que permeti validar i guardar de manera segura les factures electròniques.

Les necessitats a cobrir són:

- La càrrega de les factures en un format homogeni.
- La validació de les factures, de manera que es verifiqui:
  - la integritat, és a dir, que no hagin estat manipulades, i
  - l'autenticitat, que hagin estat signades pel proveïdor que les ha emès.
- La conservació de manera xifrada de les factures, evitant que el proveïdor de serveis d'emmagatzematge pugui veure el contingut de les factures, assegurant la privacitat de l'usuari.
- La possibilitat de recuperació en qualsevol moment de les factures emmagatzemades
- La possibilitat de veure les despeses que l'usuari ha realitzat agrupades per proveïdor.

1.2 Enfocament i mètode seguit

L'objectiu a cobrir és que l'usuari pugi disposar d'un sistema per a validar, guardar i gestionar les factures electròniques i a més poder disposar d'una còpia de les mateixes en un servei al núvol. El focus d'aquest treball és que les còpies de les factures al núvol no puguin ésser manipulades ni analitzades pel proveïdor del servei, mantenint la integritat de les mateixes i garantint la privacitat de les despeses de l'usuari. Un objectiu a banda és que el proveïdor no pugui comerciar amb aquestes dades.

El mètode a seguir és crear una aplicació de servidor que cobreix l'objectiu de guardar les factures electròniques de manera segura, i una aplicació per a mòbil per a fer les operacions de validació, gestió i enviament de les factures al servidor al núvol.

S'aniran creant i provant cadascun dels requeriments de l'aplicació. Per això és generaran una sèrie de factures signades digitalment. Algunes

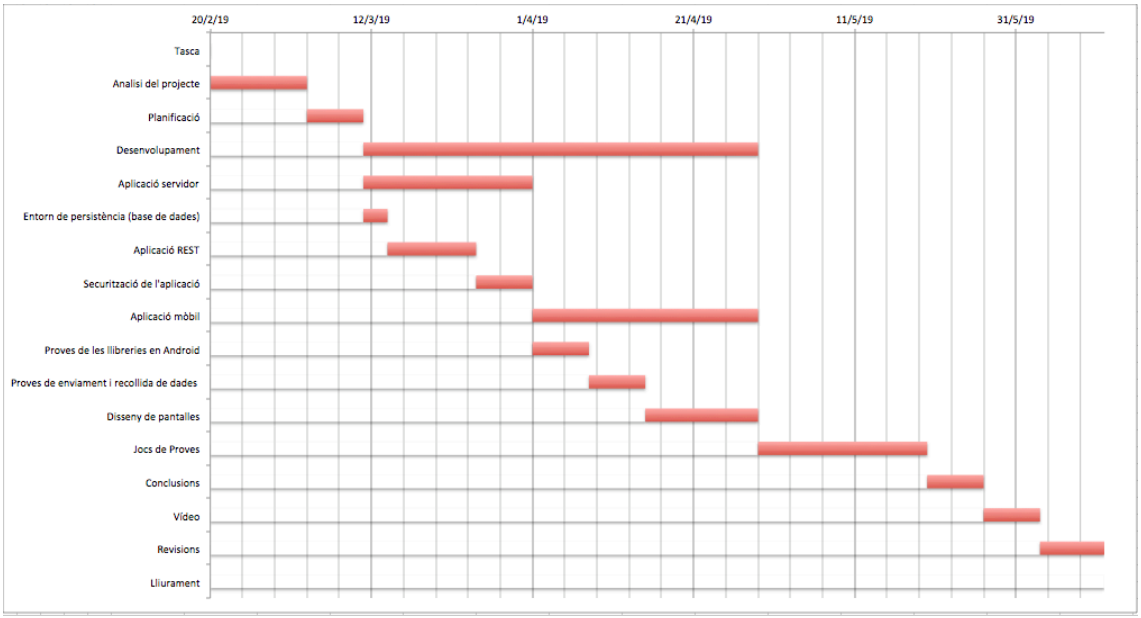
d'aquestes factures seran errònies o manipulades per certificar que la validació de les mateixes funciona.

Les factures es pujaran al servidor de manera que aquest no les pugui visualitzar. Per això es necessitarà xifrar-les de manera que només l'usuari pugui desxifrar-les.

Es provarà que aquestes es poden guardar en el servidor i descarregar-les i desxifrar-les correctament.

1.3 Planificació del Treball

En el següent diagrama de Gantt es mostra la planificació del desenvolupament del projecte a grans trets.



Il·lustració 1 Planificació del desenvolupament del projecte

La primera part del projecte és la planificació del mateix. L'aplicació de servidor es dediquen unes tres setmanes. S'ha de desenvolupar tant l'aplicació que rebrà les dades de l'aplicació mòbil, la persistència de les dades, i la seguretat. El sistema serà multiusuari.

La següent fase és fer l'aplicació mòbil, que ha de poder validar les factures. Es valoraran les llibreries disponibles i es faran les proves necessàries. Per aquesta fase es necessiten unes factures d'exemple per tal de verificar el correcte funcionament.

Es dissenyaran les pantalles necessàries per a que l'usuari pugui portar a terme les tasques de validació, càrrega i descàrrega de factures.

Es generarà un joc de proves per tal de validar cadascuna de les característiques implementades.

1.4 Breu sumari de productes obtinguts

S'obtindrà un sistema format per dos components:

Una aplicació mòbil per al sistema operatiu Android que permetrà validar les factures de l'usuari, gestionar-les i guardar de manera segura i encriptada en format Facturae en un servidor, amb una estructura de clau pública/clau privada.

Una aplicació en el servidor que s'encarregarà de guardar unes dades bàsiques de les factures i els fitxers de les factures encriptades de manera que només l'usuari pugui desencriptar-les amb la seva clau privada.

1.5 Breu descripció dels altres capítols de la memòria

En el capítol 2 d'aquesta memòria s'expliquen les tecnologies utilitzades i la justificació de la seva elecció. En el capítol 3 es detallen els requisits a cobrir, i es detallen els casos d'ús que cobrirà el sistema desenvolupat. En el capítol 4 es detalla la implementació del sistema. En el capítol 5 es mostra el funcionament del sistema cobrint cadascun dels requeriments detallats al capítol 3. En el capítol 6 s'expliquen les conclusions i les línies de treball futures. En el capítol 7 es dona un glossari dels termes utilitzats en aquesta memòria. En el capítol 8 es detalla la bibliografia consultada. En el capítol 9 s'indica informació que pot ser d'interès.

2. Tecnologies emprades

Les tecnologies que s'han utilitzat per a desenvolupar el projecte són les següents:

2.1 Servidor:

En la part de servidor s'ha utilitzat una aplicació web sense interfície d'usuari, i un sistema de persistència. L'aplicació web ha disposar de la opció de connexions segures, l'opció de poder accedir, consultar i operar amb una base de dades.

2.1.1 Spring Boot

S'ha de desenvolupar un sistema que permeti guardar de manera segura les factures i a més que garanteixi la privacitat de l'usuari. Per a fer-ho es dissenyarà i implementarà una aplicació que es posarà en funcionament en un servidor i al qual es connectaran els usuaris de manera segura i encriptada. Les connexions es faran a través de internet per la qual cosa s'utilitzarà el protocol TCP/IP per a que els usuaris accedeixin al servidor.

L'aplicació de servidor s'ha decidit desenvolupar-la en la plataforma Spring [4]. La plataforma Spring és un entorn de treball que marca una sèrie de conceptes, pràctiques i criteris per a crear aplicacions d'una manera ràpida basades en el principi de convenció sobre configuració [5] que busca minimitzar les decisions que el desenvolupador ha de prendre, sense perdre la flexibilitat.

El projecte Spring Boot [3] permet crear d'una manera ràpida aplicacions amb la plataforma Spring [4]. Aquestes són aplicacions que poden córrer de manera autònoma, ja que contenen un servidor web integrat, evitant així haver de dependre d'un determinat servidor d'aplicacions. Simplifiquen molt la posada en marxa d'aplicacions. El llenguatge de programació que es fa servir per al desenvolupament és Java, per la qual cosa es necessita una màquina virtual Java.

Aquesta plataforma permet la connexió a diversos sistemes de base de dades. També permet la creació d'una infraestructura de seguretat de manera que puguin accedir a diversos recursos usuaris que s'hagin autenticat en l'aplicació.

2.1.2 MySQL

Les dades encriptades que ha de gestionar el servidor es poden emmagatzemar de diverses formes: es poden guardar en memòria, en

estructures de dades que estaran vives mentre estigui el servidor funcionant, o bé en sistemes de persistència que ens assegurin que les dades es mantindran si l'aplicació es tanca o es reinicia.

Es per això que s'ha d'utilitzar un sistema de gestió de base de dades que permeti guardar indefinidament aquestes dades.

D'entre els diversos sistemes de base de dades existents al mercat com poden ser PostgreSQL[6], MySQL[7], SQL Server[8], Oracle[9] s'ha decidit utilitzar MySQL. Les raons:

- És un sistema de base de dades molt popular amb molta implantació i suportat per la companyia Oracle,
- L'entorn Spring en el que es desenvoluparà l'aplicació de servidor disposa de suport per a aquest sistema de base de dades.
- Permet fer consultes en format SQL estàndard i està ben integrat amb el llenguatge de programació Java
- És un sistema multiplataforma per la qual cosa ho podem instal·lar en gairebé qualsevol servidor
- Personalment tinc experiència en aquest sistema i l'he utilitzat en d'altres projectes per la qual cosa ho considero una avantatge.

2.2 Per a la part mòbil

Es descriuen a continuació les tecnologies que es faran servir en el desenvolupament de l'aplicació mòbil del sistema.

2.2.1 Android

Android és actualment al 2019 el sistema operatiu mòbil amb més implantació arreu del mon segons les dades que proveeix StatCounter [10]. Aquesta és una de les raons per haver triat aquest sistema operatiu per a desenvolupar l'aplicació mòbil.

Android és una plataforma de programari de codi obert creada per a telèfons mòbils. Es tracta d'un projecte de Google realitzat en col·laboració amb l'Open Handset Alliance [11]. Inclou un sistema operatiu basat en Linux, una interfície d'usuari, aplicacions, biblioteques de codi i compatibilitat multimèdia, entre d'altres elements.

Android es comercialitza sota dues llicències de codi obert. El nucli Linux es comercialitza sota la llicència General Public License (GPL) [12] i la plataforma Android, sense el nucli, té una llicència Apache Software License (ASL) [13].

2.2.2 SQLite

És el sistema de base de dades propi d'Android. És la base de dades on es desaran les dades de factures, en el mòbil.

SQLite és una base de dades relacional, igual que ho és MySQL que s'utilitza en la part servidor.

El fitxer que conté la base de dades està dintre del sistema de fitxers de l'aplicació amb accés de lectura i escriptura només per l'aplicació.

2.2.3 Factura-e

És un estàndard promogut pel Ministerio de Industria, Energía y Turismo per tal que les empreses proveïdores de les administracions públiques presentin les factures electròniques d'una manera homogènia amb un format estructurat.

Les factures a tractar estan en format Factura-e, que són fitxers XML signats amb una signatura digital.

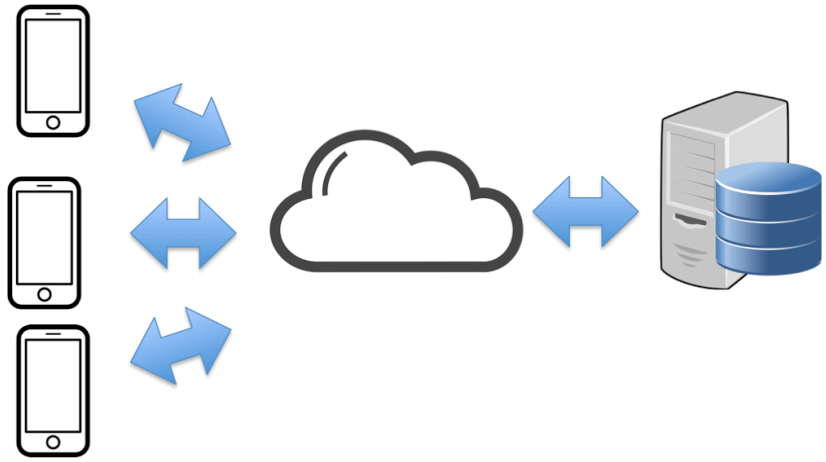
Per aquesta raó es necessiten llibreries que puguin gestionar el format Facturae, els fitxers en format XML i les signatures d'aquests fitxers.

Per a gestionar el format Facturae generarem una llibreria per tal de poder manegar el document XML com si fos un objecte. La llibreria Java que farem servir s'anomena JIBX [21].

Per a verificar les signatures Android no disposa de llibreries natives que puguin facilitar aquesta tasca. Si que estan disponibles en la versió d'escriptori. Per tal d'aconseguir la verificació de la signatura farem servir la llibreria Apache Santuario [14] que ens permet fer-la servir en un dispositiu Android.

3. Arquitectura i disseny

El sistema a desenvolupar consisteix en una aplicació mòbil on l'usuari gestionarà les factures rebudes i una aplicació de servidor on l'usuari podrà guardar de manera segura les factures de les seves despeses.



Il·lustració 2 Esquema general del sistema

3.1 Requeriments del projecte

Els requeriments a implementar en el projecte són els següents:

3.1.1 Servidor

Per a la part del servidor es defineixen els següents requisits a implementar:

Codi	*Descripció*
ReqSer-01	El servidor ha de poder informar del seu estat, si està o no actiu
ReqSer-02	El servidor ha de permetre que un usuari es registri mitjançant un usuari i contrasenya
ReqSer-03	El servidor ha de enviar per correu electrònic a l'usuari un codi per tal de verificar la seva identitat. La resposta al codi habilitarà a l'usuari en el sistema
ReqSer-04	El servidor ha de poder acceptar una petició d'habilitació d'usuari vàlida
ReqSer-05	El servidor ha de poder lliurar un certificat signat per una autoritat certificadora a petició de l'usuari

ReqSer-06	El servidor ha de poder guardar una sèrie de claus simètriques xifrades que l'usuari li enviï
ReqSer-07	El servidor ha de poder guardar una sèrie de dades de factura juntament amb la factura en format XML de tal manera que aquestes dades no puguin ser desencriptades pel servidor.
ReqSer-08	El servidor ha de poder enviar a petició de l'usuari el llistat de les seves factures xifrades
ReqSer-09	El servidor ha de poder enviar una factura concreta a petició de l'usuari

Taula 1 Requeriments del servidor

3.1.2 Aplicació mòbil

Per a la part de l'aplicació mòbil es defineixen els següents requisits a implementar:

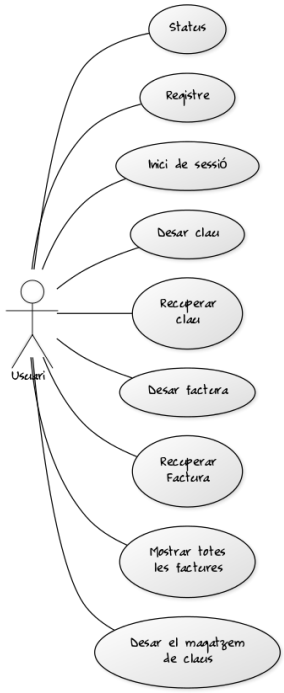
Codi	*Descripció*
ReqMob-01	L'aplicació ha de poder consultar l'estat del servidor
ReqMob-02	L'aplicació ha de permetre que un usuari es registri mitjançant un formulari amb usuari i contrasenya
ReqMob-03	L'aplicació ha de poder crear un parell de claus pública i privada per a xifrar i desxifrar asimètricament
ReqMob-04	L'aplicació ha de poder sol·licitar un CSR (Certificate Signing Request) en el moment del primer inici de sessió en el sistema
ReqMob-05	L'aplicació mòbil ha de poder rebre i desar de manera segura el certificat signat per una autoritat certificadora que enviï el servidor
ReqMob-06	L'aplicació ha de poder xifrar i desxifrar amb clau simètrica
ReqMob-07	L'aplicació mòbil ha de poder enviar al servidor una sèrie de claus simètriques xifrades per a que aquest les guardi de manera segura
ReqMob-08	L'aplicació mòbil ha de poder enviar al servidor dades de factura xifrades per a que aquest les guardi de manera segura
ReqMob-09	L'aplicació ha de poder desxifrar una sèrie de dades de factura juntament amb la factura en format XML que rebi del servidor.

ReqMob-10	L'aplicació ha de poder rebre el llistat de les seves factures xifrades
ReqMob-11	L'aplicació ha de poder desar factures en el mòbil
ReqMob-12	L'aplicació ha de poder esborrar factures que hi hagi al mòbil
ReqMob-13	L'aplicació ha de poder fer estadístiques amb les dades de les factures que hi ha al mòbil
ReqMob-14	L'aplicació ha de poder descarregar-se una factura del servidor i carregar-la al mòbil.
ReqMob-15	L'aplicació ha de permetre a l'usuari tancar la sessió

Taula 2 Requeriments de l'aplicació mòbil

3.2 Casos d'ús

En la Il·lustració 3 es mostren els casos d'us de l'usuari de l'aplicació mòbil.



Il·lustració 3 Casos d'ús de l'usuari

L'usuari ha de poder saber en tot moment si el servidor està o no actiu segons s'ha establert als requeriments.

Ha de tenir la possibilitat de registrar-se al servei per tal de poder fer servir el sistema.
Ha de poder iniciar sessió en el sistema. Ha de poder desar les claus de xifratge per tal de tenir una còpia de seguretat d'aquestes.

L'usuari ha de poder guardar una factura al servidor i recuperar-la. Ha de poder recuperar totes les factures.

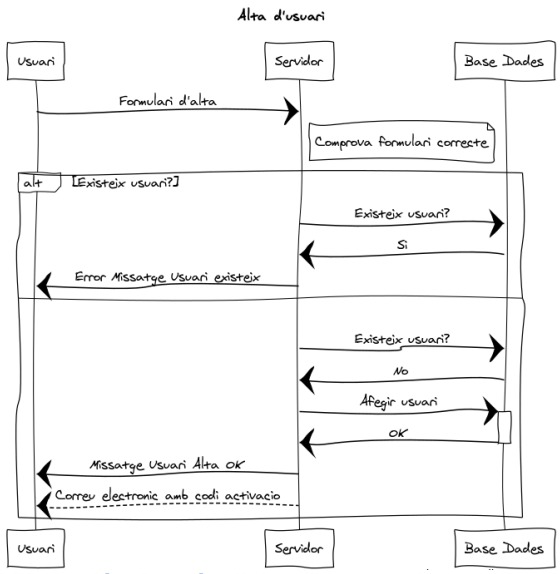
L'usuari ha de tenir la possibilitat de guardar el magatzem de claus, on tindrà les claus i contrasenyes.

Els casos d'ús del servidor són els següents:

- Ha de indicar si està o no actiu.
- Ha de disposar d'una funció que permeti registrar a l'usuari, i enviar-li un correu electrònic per tal de verificar l'autenticitat de l'usuari.
- Ha de permetre l'inici de sessió de l'usuari.
- Ha de permetre guardar les claus enviades per l'usuari i que aquest les pugui recuperar.
- Ha de permetre que l'usuari pugi dades de factura i que aquest les pugui recuperar.
- Ha de permetre que l'usuari pugi al servidor el magatzem de claus on l'usuari guarda les claus i contrasenyes.

3.2.1 Alta d'usuari en el sistema

La Il·lustració 4 descriu el procés d'alta d'un usuari en el sistema:



Il·lustració 4 Procés d'alta d'un usuari en el sistema

Per a poder utilitzar l'aplicació l'usuari s'ha de registrar en el servei. L'usuari ha de facilitar un correu electrònic que servirà com a usuari de l'aplicació i una contrasenya. La contrasenya ha de tenir les següents condicions:

- com a mínim 12 caràcters,
- al menys una lletra minúscula,
- al menys una lletra majúscula,
- al menys un número,
- al menys un caràcter especial.

Aquestes dades s'envien de manera segura mitjançant una connexió *https* al servidor que verifica que l'usuari no està donat d'alta en el sistema. Si el correu electrònic ja està donat d'alta al sistema, s'envia un missatge d'error a l'aplicació mòbil indicant aquesta situació. Si el correu electrònic no està utilitzat, el servidor guarda l'usuari i la contrasenya encriptada i es torna a l'aplicació un missatge indicant que l'operació ha estat satisfactòria.

En el moment en que l'usuari queda correctament registrat al sistema el servidor envia un missatge al compte de correu de l'usuari amb un codi d'activació. Aquest codi té una vida útil de 60 minuts.

S'ha decidit aquest temps per a que l'usuari pugui accedir a un client de correu i seleccionar l'enllaç que se li proporciona, en el cas que no ho tingui disponible en el dispositiu on té instal·lada l'aplicació. És un temps prou llarg per a que l'usuari disposi de temps per fer aquesta operació. Aquest temps de validesa del correu de verificació és necessari, ja que es podria donar el cas de que un usuari malintencionat registri un correu electrònic vàlid, i mai no el verifiqui, de manera que el correu de l'usuari legítim quedaria bloquejat.

Un cop l'usuari ha seleccionat l'enllaç es valida el codi i s'habilita l'usuari que ja pot utilitzar l'aplicació.

Nota: La contrasenya que es demana és de 12 caràcters. La formula per a calcular la seva entropia, és a dir la seva fortalesa és la següent:

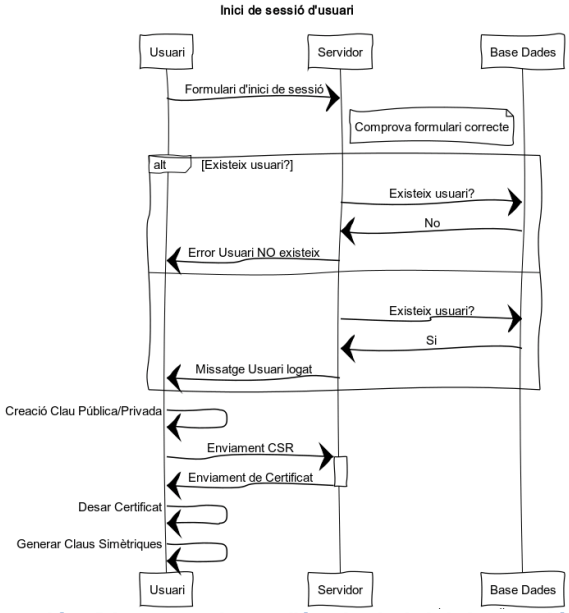
$H = (L^* \log N) / \log 2,$

On L és la longitud de la contrasenya, i N és el número de símbols diferents que es poden posar. En el cas mostrat els símbols poden ser majúscules (27, ABCDEFGHIJKLMNÑOPQRSTUVWXYZ), minúscules (27, abcdefghijklmnñopqrstuvwxyz), números (10) i símbols del teclat (22, abcdefghijklmnñopqrstuvwxyz, !@#$%&/()=?¿[]{},;.:-_)

Per tant, N = 86 i L = 12, i el número de bits d'entropia és de 77,11.

3.2.2 Inici de sessió

Per a poder utilitzar l'aplicació mòbil s'ha d'iniciar la sessió al servidor. En la Il·lustració 5 es descriu la funcionalitat.



Il·lustració 5 Diagrama de seqüència de inici de sessió

L'aplicació mòbil envia al servidor un formulari de inici de sessió amb l'usuari i contrasenya. La primera vegada que l'usuari inicia sessió l'aplicació crea un parell de claus pública i privada que es guarda al magatzem de claus de l'aplicació. Amb aquestes es genera un CSR (Certificate Signing Request), és a dir una sol·licitud de signatura de certificat i s'envia mitjançant una connexió https al servidor. Aquest rep el CSR i genera un certificat que envia de manera segura a l'aplicació mòbil i a la vegada es vincula a l'usuari que l'ha demanat. L'aplicació mòbil guarda el certificat rebut al magatzem de claus de l'aplicació.

L'aplicació genera una sèrie de claus simètriques per tal de poder encriptar un conjunt de camps que permetran a l'usuari identificar la factura i es guardaran al servidor de manera segura, encriptats amb la clau pública de l'usuari, de manera que es necessitarà la clau privada per tal de desencriptar-los. Aquestes claus simètriques es desen al magatzem de claus de l'aplicació. Es genera una clau diferent per a cada camp que es guarda al servidor.

Ara l'usuari ja pot utilitzar l'aplicació.

3.2.3 Validació de factures i còpia de seguretat al núvol

L'aplicació rep per NFC o Bluetooth les factures en format Facturae. Aquest procés està fora de l'abast del TFM. Les factures en aquest format es troben a la targeta SD del dispositiu.

Hi ha una opció que permet veure quins són els fitxers que hi ha al dispositiu. Els fitxers poden haver estat processats, és a dir, validada la seva signatura i incorporada la factura a la base de dades local, o bé pendents de processar. Es distingeixen ambdós casos de manera visual.

Per als fitxers que estan pendents de tractar es pot iniciar un procés en que es valida la signatura del fitxer i en cas que la signatura sigui vàlida es guarden les dades de la factura en una base de dades local, i per altra banda es fa una còpia d'aquestes en el servidor en el cas que aquest estigui actiu. Les dades que es guarden en local estan en clar, de manera que es poden fer consultes diverses, mentre que les dades que es guarden al servidor estan encriptades amb una sèrie de claus simètriques que s'han obtingut en el punt anterior.

3.2.4 Descàrrega de factura del núvol

Es poden visualitzar les dades de les factures que estan al servidor. Les dades estan encriptades per la qual cosa s'han de descarregar, desencriptar amb les claus simètriques utilitzades per a encriptar-les, i mostrar-les a l'usuari.

L'usuari té la possibilitat de descarregar-se-les en el cas de que la factura que està al núvol no estigui a la base de dades local.

Aquesta factura descarregada es pot incorporar a la base de dades local de factures.

3.2.5 Consultes de despeses

Les dades de les factures que es desen a la base de dades local són les mínimes imprescindibles per a poder fer consultes de despeses i proveïdors. Es desa el CIF i el nom del proveïdor, el número de factura, la data de la mateixa, i l'import total, els impostos i la base imposable. Aquestes dades ens permetran tenir estadístiques agrupades per aquests camps.

3.2.6 Còpia de seguretat del magatzem de dades

Es facilita una opció que permet a l'usuari fer una còpia de seguretat del magatzem de claus encriptada, i les dades de xifratge i la URL per a poder-les recuperar es guarden en un codi QR [20].

Aquesta opció ha de permetre que un usuari guardi de manera segura el magatzem de claus de l'aplicació en el servidor, de manera que ho pugui recuperar en cas de pèrdua de la contrasenya, o bé de la pèrdua del dispositiu.

El magatzem de claus de l'aplicació és un fitxer on en desen les claus pública i privada de l'usuari, així com les claus simètriques de xifratge dels camps identificadors de la factura que es desen al servidor.

3.3 Justificació d'una infraestructura de clau pública

Del que es tracta és de que les factures que l'usuari té en el seu dispositiu mòbil puguin estar guardades en un servidor extern de manera que l'usuari pugui recuperar-les en cas que perdi el dispositiu o perdi les dades que en ell s'hi troben.

A més es necessita que les dades estiguin encriptades de manera que l'operador del servidor no pugui desencriptar aquestes dades o li sigui especialment costós.

Per això el que es decideix és que totes les operacions de xifratge i desxifratge es facin en el mòbil amb la idea de que l'operador del servidor no tingui a l'abast les claus de xifratge i per tant no pugui saber què es el que està guardant.

S'ha de notar que si en algun moment l'operador del servidor tingués accés a les claus podria fer tant la desxifratge per a poder fer estadístiques de les dades de l'usuari. Com a contrapartida un usuari malintencionat de l'operador del servidor o el mateix operador disposaria de tota la informació de consum que l'usuari hagués posat en el servidor per la qual cosa podria treure profit econòmic d'aquestes dades i podria vulnerar la privacitat de l'usuari.

La solució que s'ha escollit per a satisfer les necessitats exposades és fer servir una infraestructura de clau pública de manera que les dades que es pugen al servidor s'encripten amb la clau pública de l'usuari de tal manera que només es poden desencriptar amb la clau privada de l'usuari que no surt del seu mòbil.

3.4 Justificació d'un xifratge simètric

El problema que ens trobem és que volem encriptar factures que tenen el format Facturae i són per tant fitxer XML d'una llargada considerable. Amb el sistema de xifratge RSA que volem utilitzar podem encriptar un fitxer sempre que la mida d'aquest sigui 11 bits menys que la llargada de

la clau. És a dir que si tenim una clau de 2048 bits, la llargada del fitxer ha d'ésser de com a molt 2037 bits, és a dir, 254 bytes. Això ens és totalment insuficient.

La solució que es proposa és fer un xifratge simètric del fitxer amb una clau aleatòria i posteriorment encriptar amb la clau pública la clau simètrica.

L'objectiu a assolir és poder enviar o guardar de manera segura grans fitxers.

Aquesta tècnica s'anomena sobre digital o en anglès, "digital envelope".

Aquesta tècnica és la que s'ha fet servir per a desar les dades de manera segura al servidor. Primer s'ha creat una estructura per a desar una sèrie de dades de factura que permetran a l'usuari identificar una factura que es desarà al servidor.

El que farà l'aplicació amb una factura que es vulgui processar és verificar que la signatura sigui vàlida. No és fan més validacions ja que hi ha hagut altres treballs que han fet aquest procés i no aportaria res de nou amb aquest treball. Si aquesta signatura és vàlida pressuposem que la factura ho és.

Un cop validada s'elimina la signatura del fitxer i es fa un "càsting" a una estructura Facturae. Amb aquesta podem extreure d'una manera senzilla les dades més rellevants de la factura: l'emissor, l'import, els impostos, el concepte, etc...

Aquestes dades s'encripten de manera simètrica amb les diverses claus aleatòries que es generen la primera vegada que l'usuari inicia sessió en l'aplicació. Les claus es desen en el magatzem de claus de l'aplicació. Aquest és un fitxer que es troba dintre dels fitxers de l'aplicació, concretament dintre del directori de l'aplicació i de manera que només el grup de l'usuari de i l'usuari de l'aplicació té drets de lectura i escriptura. Un usuari o bé una aplicació que no sigui aquesta no pot accedir al seu contingut, que a més està protegit per contrasenya. Per tant, disposem d'un magatzem segur on desar les claus de xifratge. Per seguretat també s'envien al servidor, encriptades amb la clau pública.

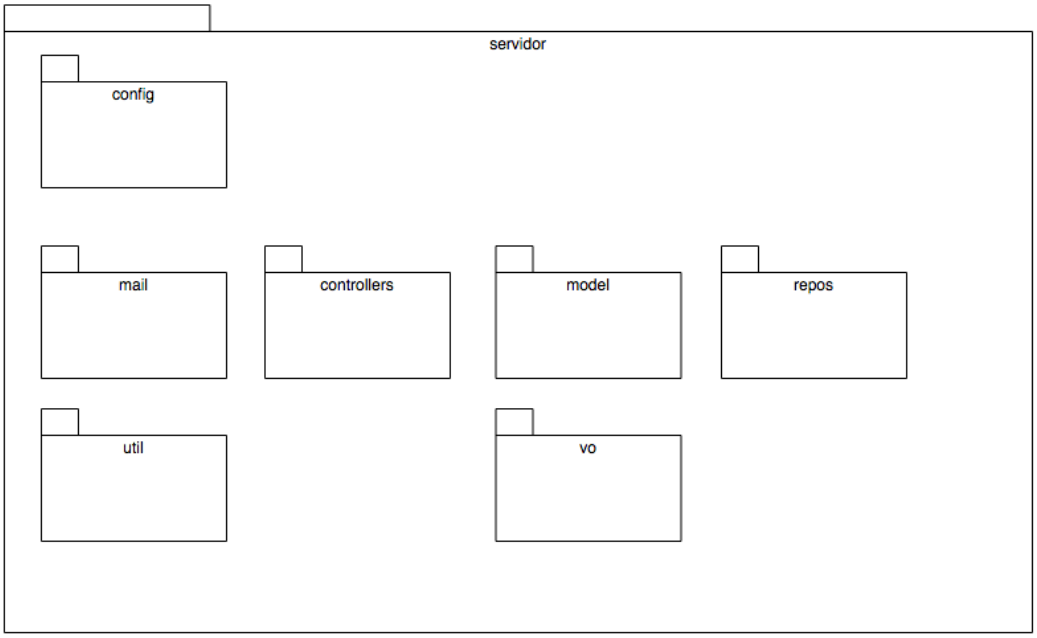
Amb això aconseguim poder descarregar-nos del servidor un registre amb les dades de factura que un cop desencriptades al dispositiu mòbil permeten a l'usuari identificar la factura i descarregar-se-la en cas que hagi perdut les dades al dispositiu.

4. Desenvolupament de la solució

En aquest capítol s'explica el desenvolupament de l'aplicació del servidor, explorant cadascun dels paquets en que s'organitza i les classes, amb les funcions més rellevants de cadascuna d'elles. Posteriorment s'explica el desenvolupament de l'aplicació mòbil, detallant algunes de les funcions més importants des del punt de vista de la seguretat.

4.1 Diagrama de components del servidor

En la Il·lustració 6 es mostra el diagrama de components del servidor:



Il·lustració 6 Diagrama de components del servidor

Descripció dels mòduls que componen el servidor:

Nom del Mòdul	Descripció
config	L'objectiu del mòdul és fer la configuració de l'aplicació de servidor. Inclou les constants que es faran servir en els altres mòduls, la classe per l'autenticació d'usuaris, el tipus de codificació de la contrasenya del usuaris, i la configuració de base de dades.

controllers	És el mòdul que conté les classes que proveeixen els punts d'entrada a l'aplicació. Hi ha punts d'entrada que estan securitzats i que només poden ser accedits en el cas en que l'usuari estigui autenticat i disposi dels permisos corresponents.
util	És el mòdul que conté classes que permeten generar un codi (token) per a la verificació per correu electrònic de l'usuari, i la generació d'un certificat a petició de l'usuari. També conté la classe que permet fer l'autenticació de l'usuari en el sistema
model	Conté les dades de les entitats que es persistiran, és a dir que es guardaran en una base de dades.
repos	Contenen les classes que implementen els interfases d'accés a dades. En Spring només cal *extendre* un determinat *interface* per implementar tots els mètodes d'accés a dades. No cal que el desenvolupador implementi aquests mètodes.
vo	Són classes d'utilitat per a passar dades entre classes dintre de l'aplicació.
mail	Classe per a gestionar l'enviament de correus electrònics a l'usuari

Taula 3 Mòduls que composen el servidor

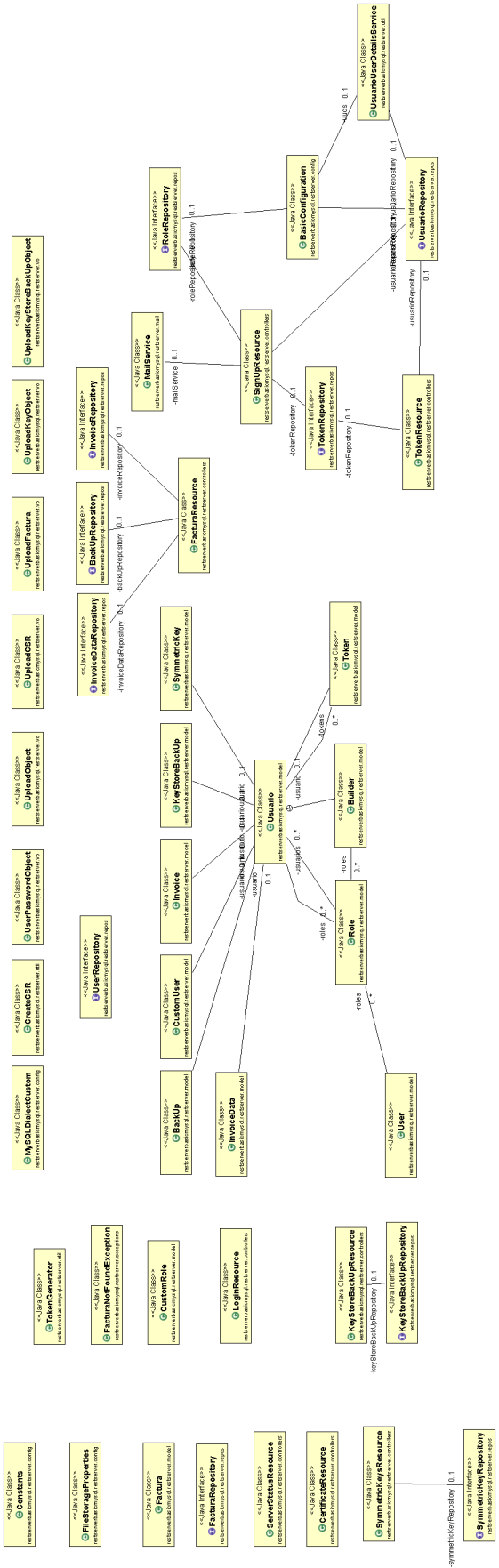
4.2 Diagrama de classes del servidor

En la Il·lustració 7 es mostren les classes que implementen l'aplicació del servidor i la relació entre elles.

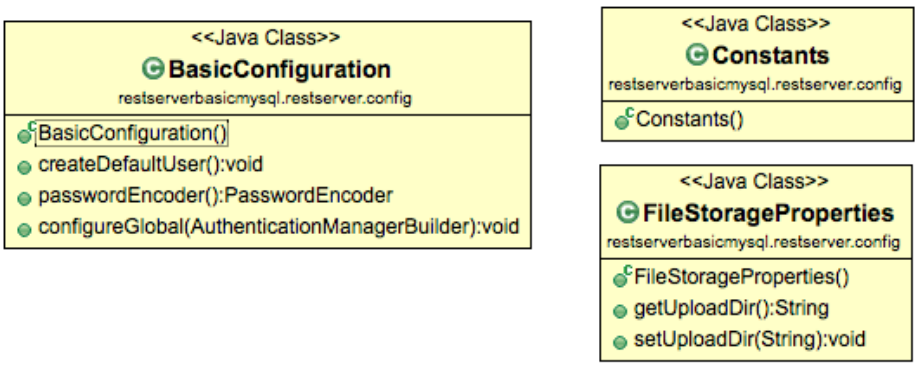
A continuació es fa una descripció de cadascuna d'elles.

La classe principal que arrenca el servidor és **RestServerApplication**. Com s'indicava en el punt 2.1.1 Spring és un sistema que funciona per configuració, i en aquesta classe es configura un objecte, en terminologia Spring un *bean*, que gestiona l'enviament de correus electrònics, ja que es necessita segons el requisit ReqSer-03. Spring s'encarrega de la creació, inicialització i gestió de cada objecte. És el concepte anomenat Inversió de Control [15].

L'aplicació s'encarrega de llegir un fitxer *application.properties* que està en l'arrel de l'aplicació del servidor i que conté una sèrie de paràmetres preestablerts i d'altres modificables que indiquen al *framework* Spring com ha de iniciar la seguretat, com ha de connectar-se a la base de dades i com ha de gestionar els fitxers que s'hagin de pujar al servidor, en el cas que hi hagués.



Il·lustració 7 Diagrama de classes del servidor



Il·lustració 8 Detall de les classes del mòdul config

En la Il·lustració 8 es mostra el detall de les classes que s'inclouen en el mòdul **config**. La classe Constants conté valors que es fan servir en diverses classes. *FileStorageProperties* permet configurar on es desaran els fitxers que es pugin al servidor. La classe *BasicConfiguration* estableix quin serà el encriptador de les contrasenyes que es facin servir en la taula d'usuaris. És important remarcar que com a norma de seguretat les contrasenyes que es guarden han d'estar codificades. En aquest cas es fa servir con a codificador la classe del *framework* org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder.

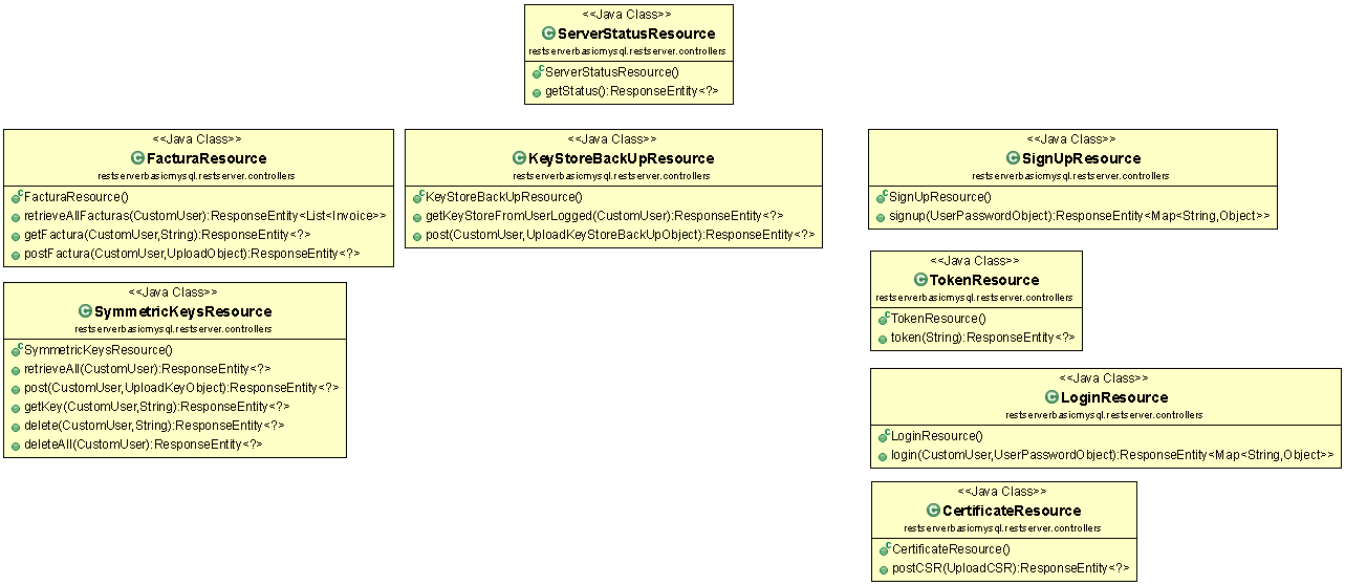
El mètode *configureGlobal* és el més significatiu d'aquesta classe ja que en ell es configuren els punt d'accés així com els rols que han de tenir els usuaris per a poder accedir.

Aquest punt és crucial, ja que el focus d'aquest treball està en la seguretat del servidor que és allà on es desen les factures encriptades de l'usuari. Els usuaris poden accedir a diferents punts d'entrada, que es tradueixen en URLs, en funció de si estan o no autenticats, i del rol que tenen. Per defecte un usuari quan es registra a l'aplicació se li assigna directament el rol d'usuari ("USER").

Els punts d'entrada que es defineixen són:

- /status, és el punt d'entrada que pot accedir qualsevol usuari estigui o no autenticat i que permet saber si el servidor està actiu o no,
- /signup, és el punt d'entrada que pot accedir qualsevol usuari estigui o no autenticat i que permet a l'usuari registrar-se en el sistema, segons s'explica al punt 3.2.1

- /token, és el punt d'entrada que pot accedir qualsevol usuari estigui o no autenticat i que permet a l'usuari habilitar l'usuari un cop seleccioni l'enllaç que se li envia a l'usuari per a verificar el seu correu electrònic, segons l'explicat al punt 3.2.1
- /login, és el punt d'entrada per a usuaris. S'ha de informar de l'usuari (correu electrònic) i la contrasenya.
- /keys, és el punt d'entrada per a desar les claus simètriques encriptades amb la clau pública de l'usuari, de manera que es guarden al servidor i únicament l'usuari amb la seva clau privada les pot desencriptar.
- /factures, és el punt d'entrada per a desar les dades de factures encriptades amb les claus simètriques. Permet desar les factures i recuperar-les.
- /ksb, és el punt d'entrada per a poder desar el fitxer d'emmagatzemament de claus del mòbil que conté tant la clau pública com la privada i que servirà per a poder recuperar les claus en el cas que l'usuari perdi la contrasenya o bé perdi el mòbil amb el magatzem de claus



Il·lustració 9 Diagrama de classes del mòdul controllers

En la Il·lustració 9 es mostra el diagrama de classes del mòdul **controllers**. Aquest conté les classes que donen servei als punts d'entrada que s'han descrit en els paràgrafs anteriors.

Cada punt d'entrada es pot accedir amb diferents mètodes HTTP. Per exemple, al punt d'entrada /factures es pot accedir amb un mètode GET per a poder recuperar una factura o diverses, i es pot accedir amb el mètode d'entrada POST per tal d'afegir una factura a les que guarda el servidor.

En aquestes classes es defineixen els mètodes que donen servei a aquestes operacions en els punts d'entrada.

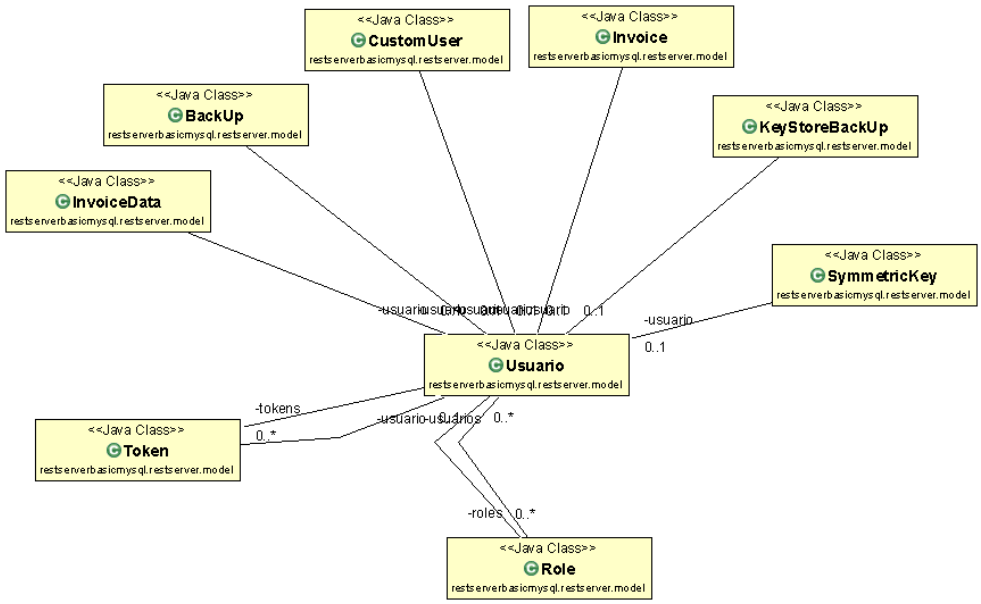
Per exemple, segons es pot veure en la Il·lustració 9 a la classe **FacturaResource** hi ha un mètode **retrieveAllFacturas**. Aquest mètode té com a anotació:

```
@RequestMapping(value="/facturas", method = RequestMethod.GET)
public ResponseEntity<List<Invoice>> retrieveAllFacturas(
@AuthenticationPrincipal CustomUser user) {
```

Aquesta anotació indica a Spring que configuri que aquest mètode serà el que s'ha d'executar quan l'aplicació rebi una petició GET en la URL https://nomdelservidor:port/factures, on nomdelservidor és la IP o el nom de domini del servidor i port és el número de port on escolta l'aplicació.

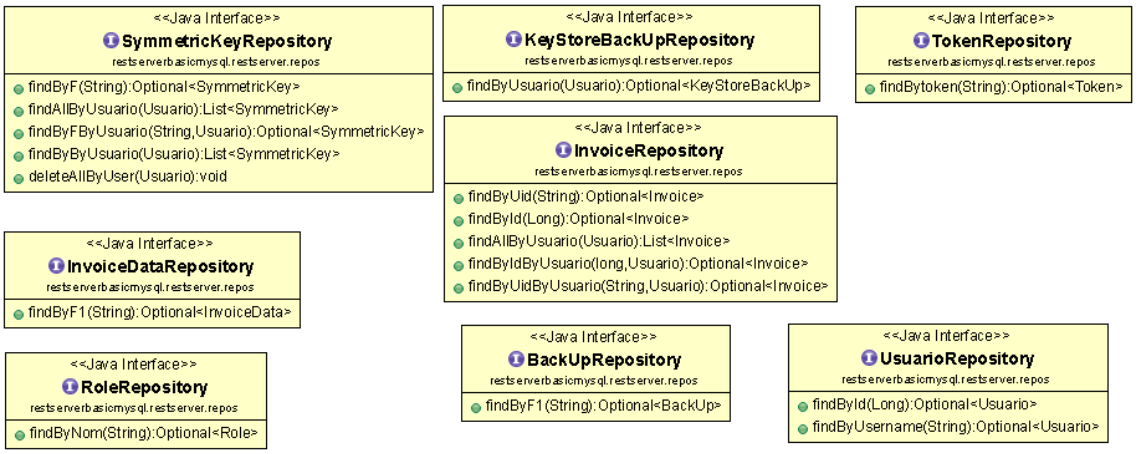
La resta de classes funciona de manera similar:

- ServerStatusResource, dona servei a les peticions al punt d'entrada /status, i només accepta peticions de tipus GET,
- SignUpResource, dona servei a les peticions al punt d'entrada /signup, i només accepta peticions de tipus POST, que vinguin amb un objecte de tipus JSON amb un camps determinats.
- TokenResource, dona servei a les peticions al punt d'entrada /token, i només accepta peticions de tipus GET,
- LoginResource, dona servei a les peticions al punt d'entrada /login, i només accepta peticions de tipus POST. L'usuari ha d'estar autenticat per a poder accedir. Si no ho està el servidor rebutja la connexió.
- SymmetricKeysResource, dona servei a les peticions al punt d'entrada /keys, i accepta peticions POST i GET, per afegir i recuperar respectivament les claus simètriques de l'usuari. L'usuari ha d'estar autenticat per a poder accedir. Si no ho està el servidor rebutja la connexió.
- FacturaResource, dona servei a les peticions al punt d'entrada /facturas, i accepta peticions POST i GET, per afegir i recuperar respectivament les factures de l'usuari. L'usuari ha d'estar autenticat per a poder accedir. Si no ho està el servidor rebutja la connexió.
- KeyStoreBackUpResource, dona servei a les peticions al punt d'entrada /ksb, i accepta peticions POST i GET, per afegir i recuperar respectivament el magatzem de claus de l'usuari. L'usuari ha d'estar autenticat per a poder accedir. Si no ho està el servidor rebutja la connexió.



Il·lustració 10 Diagrama de classes del mòdul model

En el mòdul **model** es troben les classes de totes les entitats que intervenen en el model de negoci i que es persistiran a la base de dades. La classe central és la classe *Usuario*, ja que per a que el sistema pugui ser multiusuari les entitats que es desen han d'estar relacionades amb el seu propietari, que serà l'usuari.



Il·lustració 11 Diagrama de classes del mòdul repos

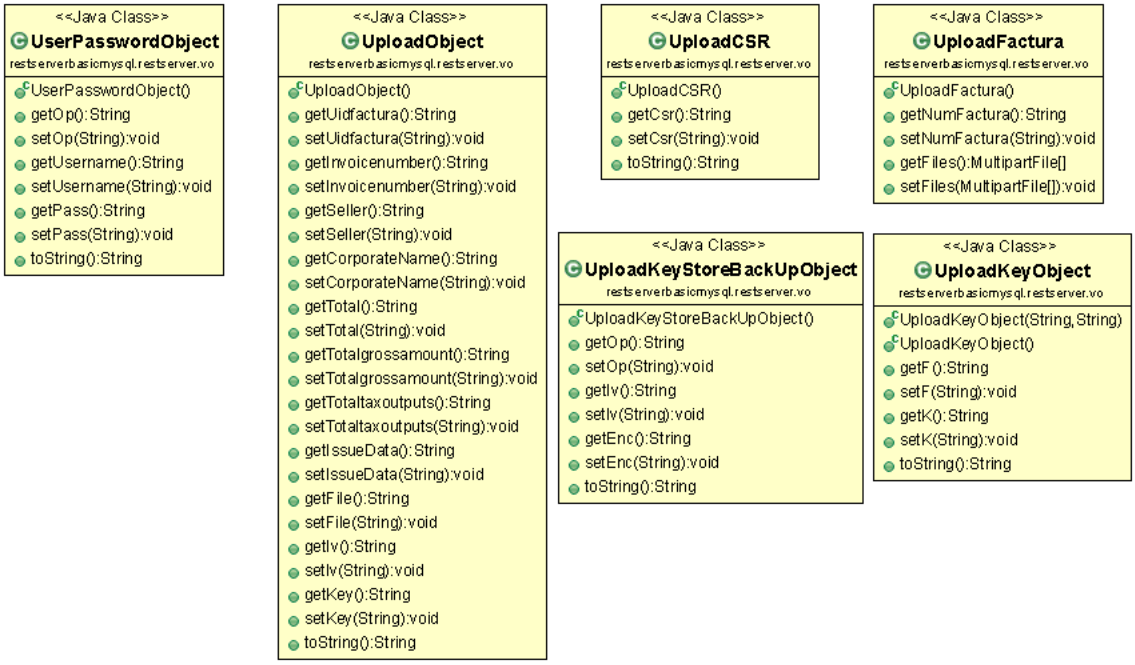
En aquest mòdul no hi ha classes sinó que hi ha *interfaces*. El motiu és que com s'ha explicat Spring funciona per configuració, i per tal de poder generar les connexions a les taules, s'han de crear els *interfaces* per a que Spring implementi les classes a partir d'aquests. S'han de definir els mètodes especials que no estiguin definits per l'*interface* JPARepository que és l'*interface* que han d'estendre.

Cada un dels *repositoris* que es defineixen a una de les classes del model.

Il·lustració 12 Diagrama de classes del mòdul util

El mòdul **util** l'integren tres classes que donen servei a d'altres classes.

- UsuarioUserDetailsService, permet localitzar a l'usuari en la taula, carregar els roles que té assignats i validar si està o no habilitat. Fa servir la infraestructura de seguretat de Spring implementant un dels interfaces (UserDetailsService) que proveeix el framework.
- CreateCSR, proveeix d'un mètode estàtic per a generar un certificat a partir d'una petició CSR que arriba per part de l'usuari en el moment de fer l'inici de sessió, segons s'explica en el punt 3.2.2



Il·lustració 13 Diagrama de classes del mòdul vo

L'últim del mòduls conté les classe que serveixen per a passar informació a l'aplicació. Tal i com s'ha explicat anteriorment en els punts d'entrada a l'aplicació es poden utilitzar dos mètodes: GET i POST. El mètode GET s'utilitza per demanar informació al servidor, deixant-lo en el mateix estat que es trobava. El mètode POST [16] es defineix com aquell que pot afegir informació a la que ja hi ha al servidor. Per això la crida al mètode POST ha de venir amb una estructura de dades. Aquesta estructura ha de ser prevista per el mètode que li dona servei. En la implementació s'ha fet que la informació vingui en format JSON (JavaScript Object Notation) [17], ja que és una estructura lleugera,

fàcilment implementable en diverses plataformes i que Spring entén de manera nativa. Simplement es tracta de definir la classe amb els camps que s'espera rebre, i posar-la com a paràmetre en la definició del mètode que dona servei. Per exemple, en el cas del mètode que permet pujar una factura encriptada al servidor, la signatura del mètode és:

```
@RequestMapping(value = "/facturas", method = RequestMethod.POST)
public ResponseEntity<?> postFactura(
@AuthenticationPrincipal CustomUser user,
@RequestBody UploadObject factura) {
```

Com s'observa hi ha dos paràmetres que rep el mètode:

- Customer user, amb l'anotació @AuthenticationPrincipal que indica que només accedirà a aquest mètode si l'usuari ha estat prèviament autenticat amb usuari i contrasenya en el moment de fer la petició POST,
- UploadObject factura, amb l'anotació @RequestBody, que indica que el mètode espera rebre en el BODY del POST un objecte amb l'estructura de camps definit a la classe UploadObject.

4.3 Creació d'una infraestructura de clau pública

L'objectiu és simular una estructura real de clau pública per tal de veure com funcionaria en un entorn real.

En un entorn de producció el servidor disposaria d'una parella de claus pública/privada amb un certificat emès per una entitat certificadora. Per exemple, com a ciutadans de l'estat espanyol podem disposar d'un certificat digital que ho emet la Fabrica Nacional de Moneda y Timbre (FNMT). Per a això el procés és generar un parell de claus, s'envia la clau pública en un CSR (Certificate Signing Request) a la FNMT, i aquesta genera un certificat. Un cop verificada la nostra identitat la primera vegada de manera presencial ens podem descarregar mitjançant un codi el certificat digital que ens permetrà fer una sèrie d'operacions criptogràfiques. Aquest certificat es desa en un fitxer de tipus PKCS12 que permet guardar de manera segura la clau privada i el certificat. Amb aquestes podem accedir a una sèrie de serveis com per exemple la presentació de la declaració d'Hisenda, etc...

En l'entorn de desenvolupament el que es fa és crear un parell de claus i amb aquestes un certificat autosignat. Aquest certificat el farem servir com a CA (Certification Authority o autoritat de certificació).

Un cop tenim aquest certificat, per a cada usuari de l'aplicació creem un parell de claus pública i privada i generem un CSR que ho signarà la CA, i generarà un certificat. Aquest inclourà la clau pública de l'usuari.

En principi es necessita un per al servidor, i un per a cada usuari.

En el servidor es necessiten els certificats del servidor i de la CA. La justificació és que cada usuari de l'aplicació mòbil crearà la seva parella de claus pública i privada i enviarà in CSR al servidor per a que aquest generi un certificat i ho enviï a l'usuari. El certificat estarà signat per la CA.

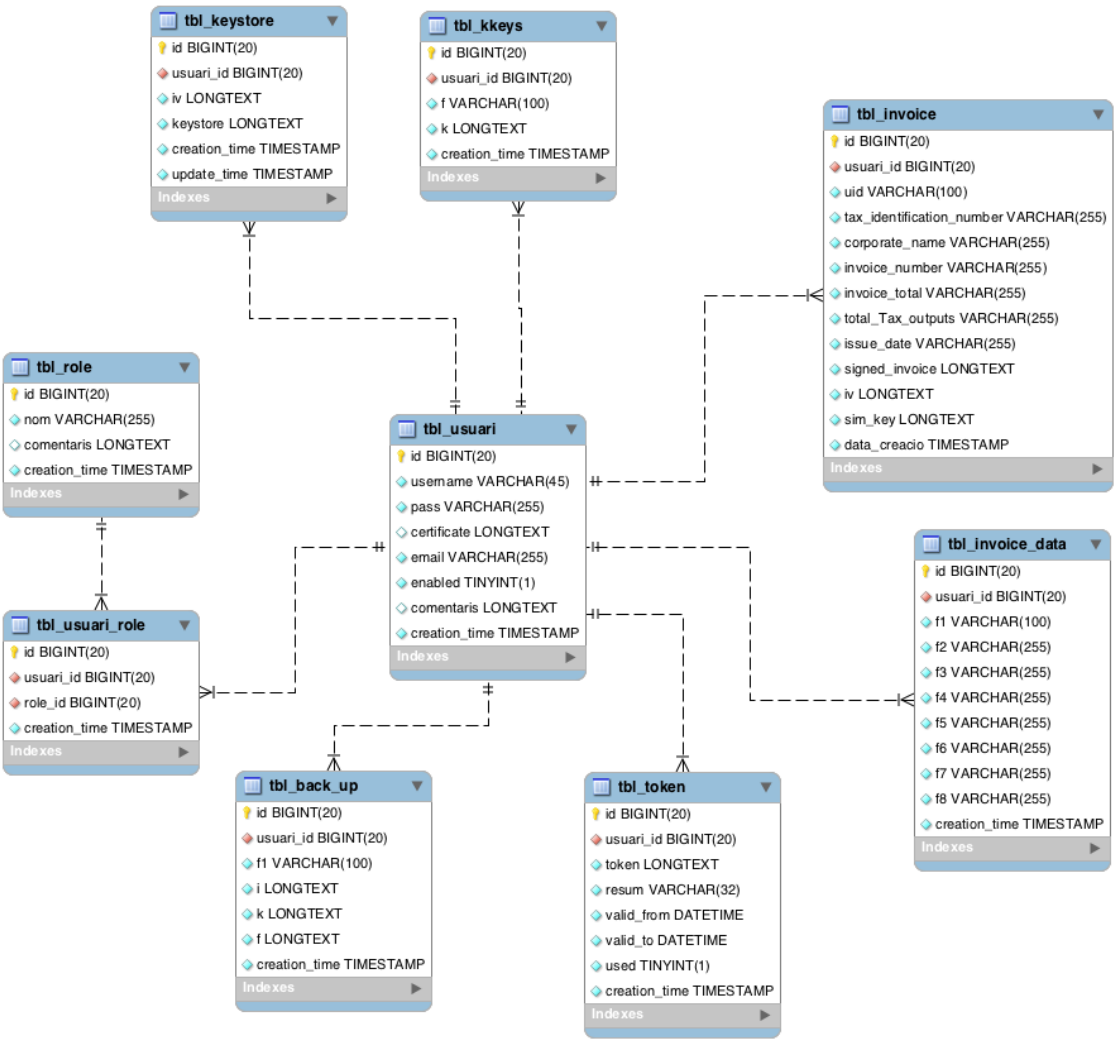
4.4 Creació de les taules en la base de dades del servidor

Mitjançant un fitxer de guió (script) es crea tant la base de dades en el servidor MySQL com les taules i les relacions entre elles que permetran disposar d'un entorn multiusuari, i permetran que es guardin de manera segura, encriptada, les dades bàsiques que permeten identificar una factura, així com les claus que permeten desencriptar-les degudament encriptades amb un sistema de clau pública-privada.

Les taules que aquí es descriuen corresponen a les classes del mòdul "model" que s'ha descrit anteriorment.

La taula tbl_usuari té una relació 1:N amb la resta de taules ja que cadascuna de les entitats que es volen persistir tenen una relació amb l'usuari que les crea i/o demana. Això permet crear un sistema multiusuari.

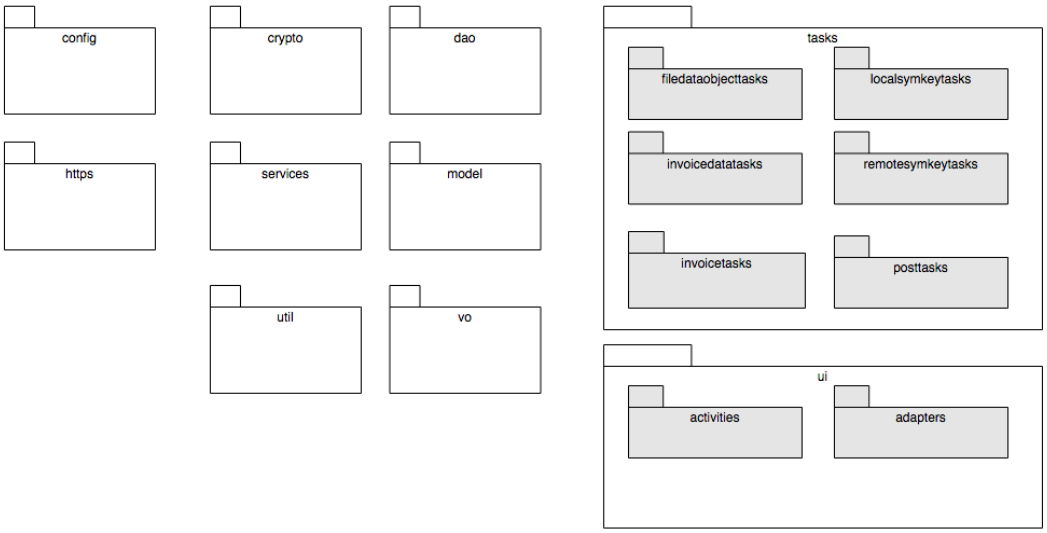
Les taules que guarden les dades de les factures només tenen camps de tipus cadena i no pas de tipus numèric o de data com es podria esperar. La raó és que la informació que se li envia al servidor està encriptada i el resultat del xifratge es codifica en Base64 [18] que és una cadena de caràcters alfanumèrics i que requereix d'un camp de tipus cadena per tal de poder guardar-se.



Il·lustració 14 Diagrama de taules de la base de dades del servidor

4.5 Diagrama de components de l'aplicació mòbil

Un cop descrita la part del sistema corresponent a l'aplicació de servidor que s'encarregarà d'emmagatzemar les factures de manera segura passem a descriure la part mòbil que s'encarregarà de la validació, xifratge i desxifratge de les mateixes. En la Il·lustració 15 podem observar el diagrama de paquets que componen l'aplicació mòbil.



Il·lustració 15 Diagrama de paquets de l'aplicació mòbil

Es passa ara a comentar els principals paquets que componen l'aplicació.

En el paquet *ui* (*user interface*) s'inclouen dos paquets que gestionen la interfície de l'usuari: *activities* i *adapters*. El primer paquet conté les diferents *activity* que gestionen les pantalles de l'usuari, i els *adapters*, contenen la lògica per a la presentació i gestió de les dades de pantalles amb format de llistes d'objectes, per exemple quan l'aplicació presenta una sèrie de fitxers o de factures.

El paquet *tasks* conté les classes que fan les operacions d'accés a dades tant remotes, al servidor, com locals a les bases de dades que són de tipus **asíncron**. Hi ha operacions que poden trigar més temps, i es tracten com a peticions que es demanen i quan es disposa de les dades, aleshores es mostren, mentre l'aplicació continua fent d'altres processos.

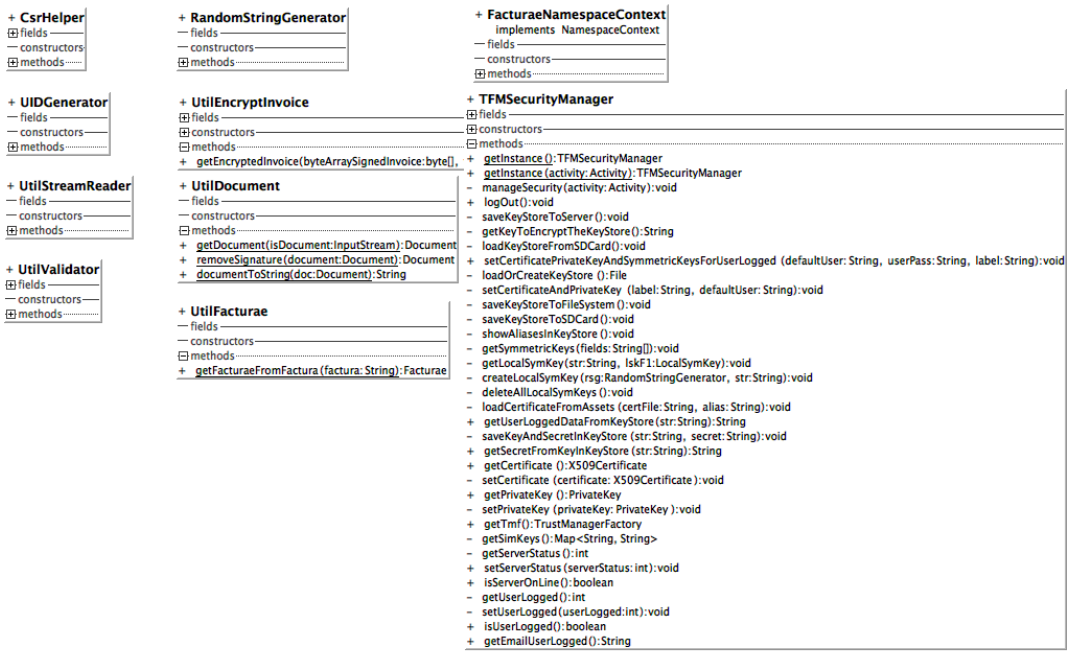
El paquet *services* conté classes que fan processos que són comuns a diverses *activities*, de manera que s'elimina codi repetitiu.

El paquet model conté les classes d'entitats que es persisteixen en les taules de la base de dades local

El paquet *dao* conté *interfaces* que permeten que Android estengui per tal de crear les classes d'accés a les taules de la base de dades local.

4.6 Gestió de la seguretat

L'aplicació en el moment d'arrencada comença obrint una *activity* anomenada *SplashActivity*, que a més de mostrar un codi BIDI com a pantalla de presentació fa una crida a la classe *TFMSecurityManager*, que és la classe principal del paquet *util, tal i com es mostra a la* Il·lustració 16.



Il·lustració 16 Diagrama de classes del paquet util

L'aplicació en el primer moment de posada en marxa ha de fer una sèrie de processos. En primer lloc ha de carregar els certificats que té inclosos en el directori "*assests*" de l'aplicació. Aquests certificats són els de la CA (Autoritat de Certificació) i el del servidor, creat i validat per l'autoritat de certificació. Aquests es faciliten en el moment de la instal·lació, ja que venen incorporats dintre del paquet d'instal·lació. Aquests permetran que puguem fer una comunicació SSL amb el servidor.

Els certificats es carreguen a la *KeyStore,* que és com s'anomena el magatzem de claus de l'aplicació. Aquesta permet generar una entitat de confiança (*TrustManagerFactory*) que farà possible les connexions segures amb el servidor.

Un cop es fa la càrrega dels certificats es fa un primer intent de connexió amb el servidor, per tal de verificar que està funcionant i que es poden fer consultes al mateix, tal i com es demana en el requisit ReqMob-01.

En la KeyStore es desen diverses dades que serviran per al correcte funcionament de l'aplicació:

- El certificat de la CA i el certificat del servidor, tal i com s'ha explicat anteriorment,
- Clau privada de l'usuari, protegida per contrasenya
- Clau públic de l'usuari, que es desarà en format de certificat emès i signat per una CA
- Claus simètriques, per a encriptar les dades de factures que es pugin al servidor,
- El nom de l'usuari que té iniciada la sessió,
- La contrasenya de l'usuari que té iniciada la sessió.

Aquesta KeyStore està emmagatzemada dintre de les carpetes de l'aplicació i es crea en el primer moment en que arrenca l'aplicació després de la seva instal·lació. Les següents vegades el fitxer es carrega, no cal crear-lo. A nivell de seguretat s'ha de indicar que els permisos amb que es crea aquest fitxer són de lectura i escriptura per part de l'aplicació, i que cap altre usuari i/o aplicació podrà accedir a aquest fitxer, que per més seguretat està protegit per una contrasenya.

En el cas que l'usuari no s'hagi registrat se li permet fer el registre mitjançant una pantalla de registre (Sign Up). En aquesta se li demana un usuari, que serà un correu electrònic vàlid, i una contrasenya amb les condicions explicades en el punt 3.2.1.

Es valida que es compleixen les condicions establertes i s'envia la petició al servidor.

El servidor valida que l'usuari sigui vàlid i no estigui utilitzat i el dona d'alta. Es marca com a inhabilitat fins que no es verifiqui la identitat de l'usuari.

Aquest pas es fa fóra de l'aplicació. El servidor envia un correu a l'usuari amb un codi (token). En seleccionar el codi, s'envia una petició d'habilitació a l'usuari. Si el codi és vàlid i no està caducat, el servidor habilita l'usuari.

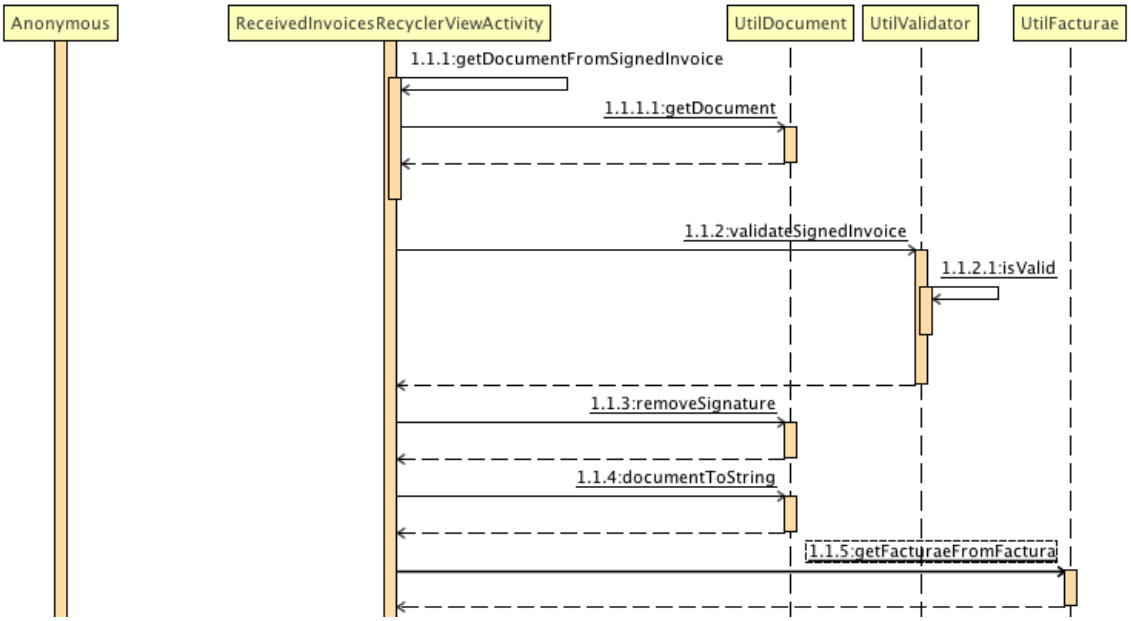
L'usuari pot iniciar sessió tal i com s'ha explicat en el punt 3.2.2. En el moment en que s'inicia la sessió es guarden les dades d'usuari i contrasenya de l'usuari logat en la KeyStore. Això permet que si es tanca l'aplicació quan l'usuari l'obri de nou l'aplicació comprova si hi ha un

usuari logat, i pot utilitzar l'usuari i contrasenya a més de les claus de l'usuari, tant la pública i privada com les claus simètriques.

4.7 Processat dels fitxers de factura rebuts

Els fitxers de factura es reben i es desen a la targeta de memòria del mòbil. Aquests es mostren en un llistat, marcats amb colors en funció de si estan o no processats. En el cas que estiguin processats, i les dades incorporades a la base de dades local estaran marcats amb un color verd.

En el cas que estiguin pendents de processar, a més d'estar marcats en un color gris, s'habilita un botó per executar el processat. En la Il·lustració 17 es mostren les fases inicials del processat.



Il·lustració 17 Diagrama de seqüència parcial del processat del fitxer signat

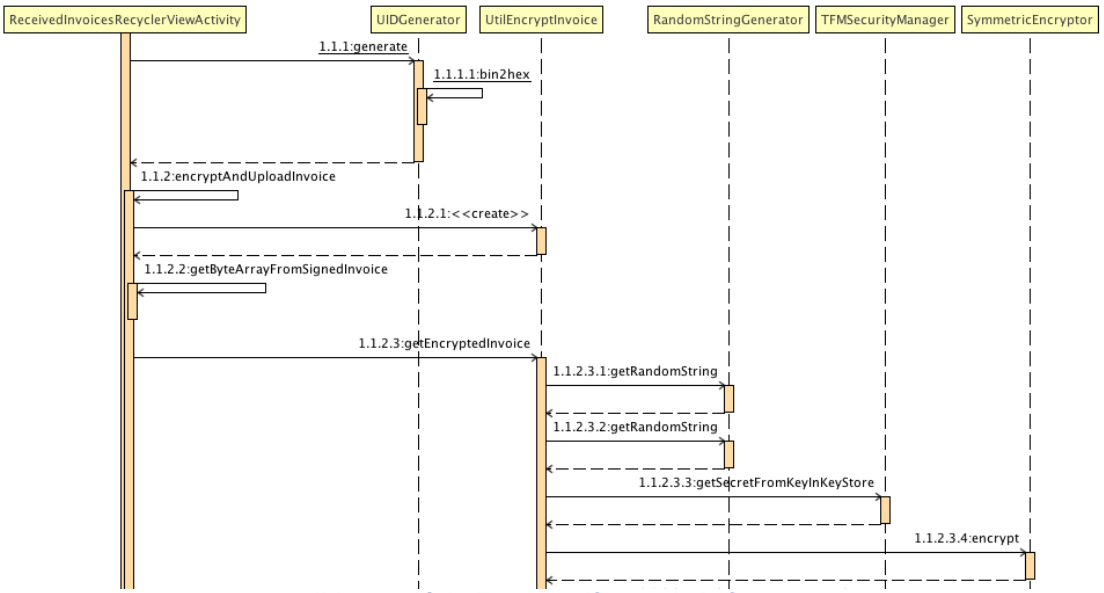
El primer pas és recuperar el fitxer, i extreure el document XML que conté. Aquest document es passa al mètode validateSignedInvoice per a que verifiqui la validesa. La resposta és booleana, és vàlid o no. Si és vàlid és que no s'ha manipulat després de la seva signatura.

Posteriorment s'elimina la signatura digital, quedant únicament el fitxer amb l'estructura Facturae. Aquesta estructura es carrega en un objecte Facturae per a poder gestionar-lo amb més comoditat.

Per a guardar la factura amb un codi únic es genera un resum amb una sèrie de camps significatius. L'objectiu és que serveixi com a identificador d'una factura tant en la base de dades local com a la base

de dades remota, i poder fer una operació de sincronització de les dades que hi han en ambdós bases de dades.

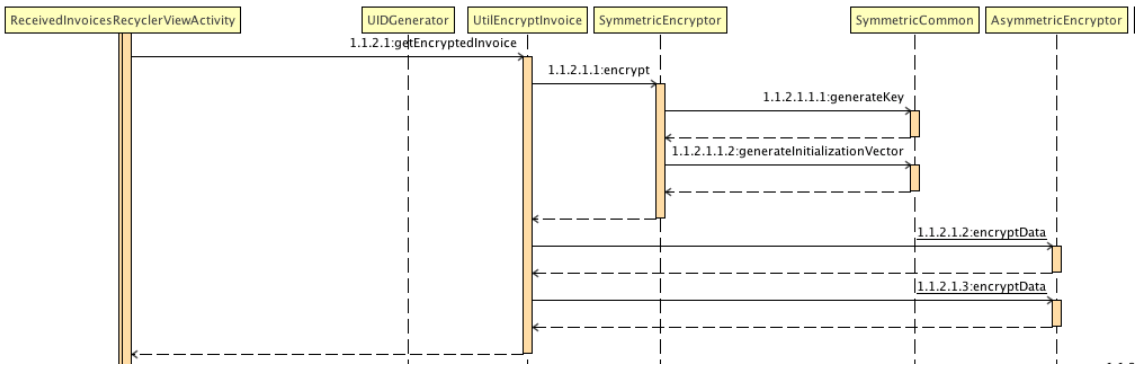
Posteriorment s'instancia la classe que ens permetrà fer el xifratge de la factura. I es generen dos codis de 256 bits (*getRandomString*), un és el IV (initialization vector) i l'altre és un codi que servirà per a xifrar el fitxer amb la factura signada i pujar-lo al servidor. Aquests passos es veuen a la Il·lustració 18.



Il·lustració 18 Fase de xifratge de la factura

A continuació es recuperen les claus simètriques del magatzem de claus, una per a cada camp que volem xifrar i pujar al servidor. En la Il·lustració 18 només es mostra el primer (*getSecretFromKeyInKeyStore*). Amb aquesta clau es xifra cada camp.

La següent fase és xifrar el fitxer, per això s'utilitzen el IV i la clau generades anteriorment. Un cop xifrat el fitxer, es xifren asimètricament les claus.

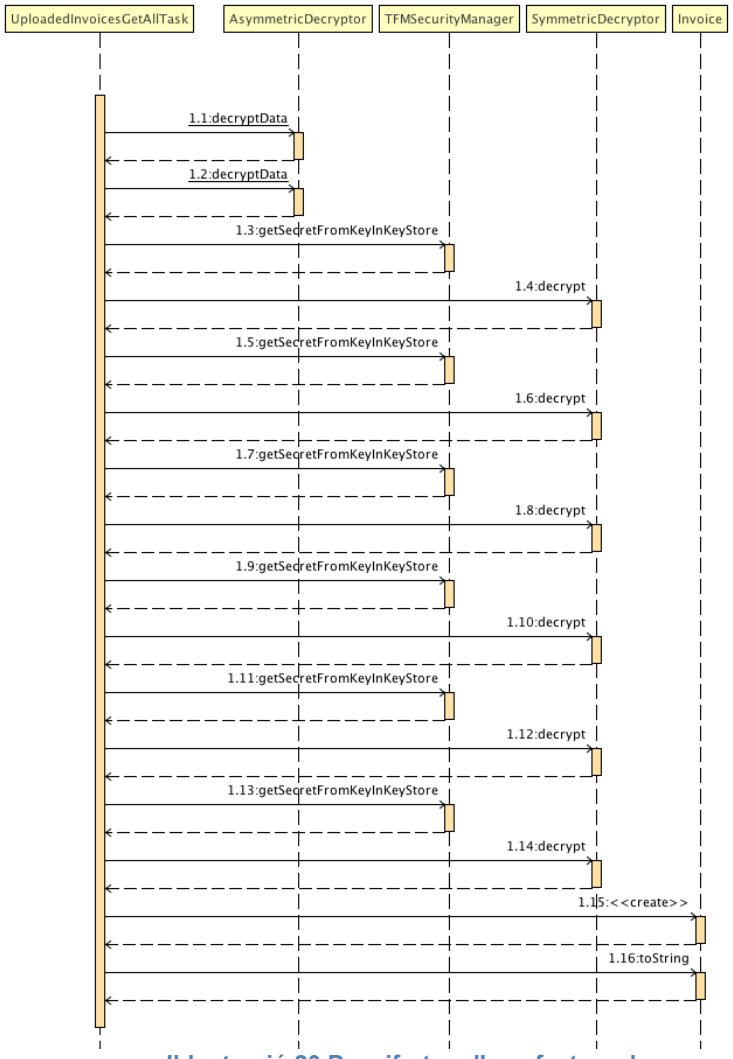


Il·lustració 19 Xifratge de la factura signada

Ara ja es pot pujar la factura al servidor. Per això es crea un objecte JSON que inclou tots els camps que s'han descrit. Aquest objecte es passa com a paràmetre al constructor de la classe *PostDataAuthenticatedToUrlTask* que s'encarrega de obrir una connexió https autenticada amb l'usuari i contrasenya i fer una petició POST al servidor per a que incorpori la factura a la base de dades.

4.8 Descàrrega de factures del servidor

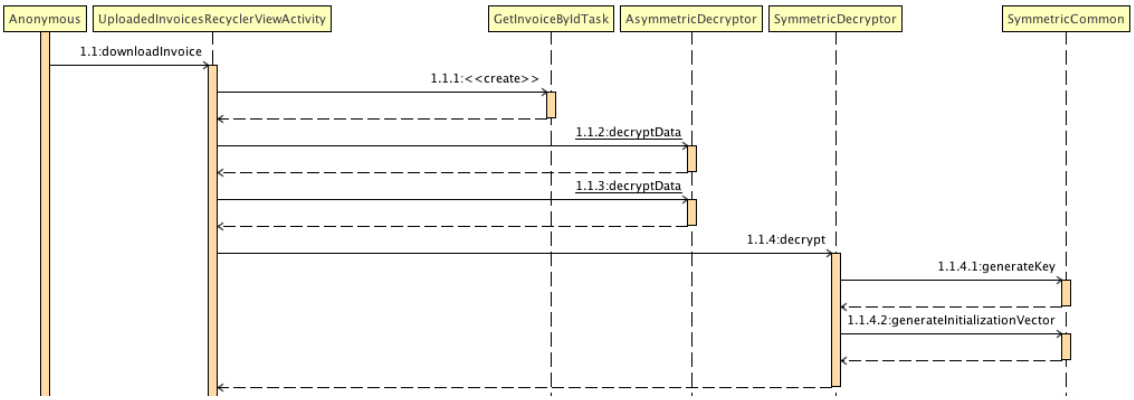
En descarregar-se del servidor la factura xifrada l'aplicació mòbil ha de fer el procés invers que s'ha descrit al punt 4.7 i que es mostra a la Il·lustració 20. Es desxifra cada camp amb la clau corresponent que s'extreu del magatzem de claus.



Il·lustració 20 Desxifratge d'una factura descarregada

S'han de distingir dos casos. El que s'acaba d'explicar és el que s'utilitza per a mostrar a l'usuari les dades bàsiques per a que pugui identificar una factura carregada al servidor.

Hi ha un altre cas, que es descriu en el requeriment ReqMob-08, en que a més s'ha de descarregar el fitxer XML. En aquest cas, s'han de descarregar els camps que contenen els valors del vector d'inicialització (IV) i la clau. Aquests estan xifrats amb la clau pública de l'usuari. Per tant, quan es descarreguen les dades, s'han de desxifrar amb la clau privada de l'usuari (classe *AsymmetricDecryptor,* mètode *decryptData*), i posteriorment aquestes claus desxifrades serviran per desxifrar el fitxer amb la factura signada (classe *SymmetricDecryptor,* mètode *decrypt*). Aquest procés es descriu a la Il·lustració 21.



Il·lustració 21 Desxifratge del fitxer XML descarregat

4.9 Còpia de seguretat manual del magatzem de dades en el servidor

L'usuari en el cas que perdi la contrasenya del servei o bé el dispositiu pot perdre tota la informació de que disposa al servidor, ja que per poder accedir al servei necessita autenticar-se i per a poder desxifrar la informació de les factures que hi ha al servidor necessita la seva clau privada ja que s'han encriptat amb la seva clau pública.

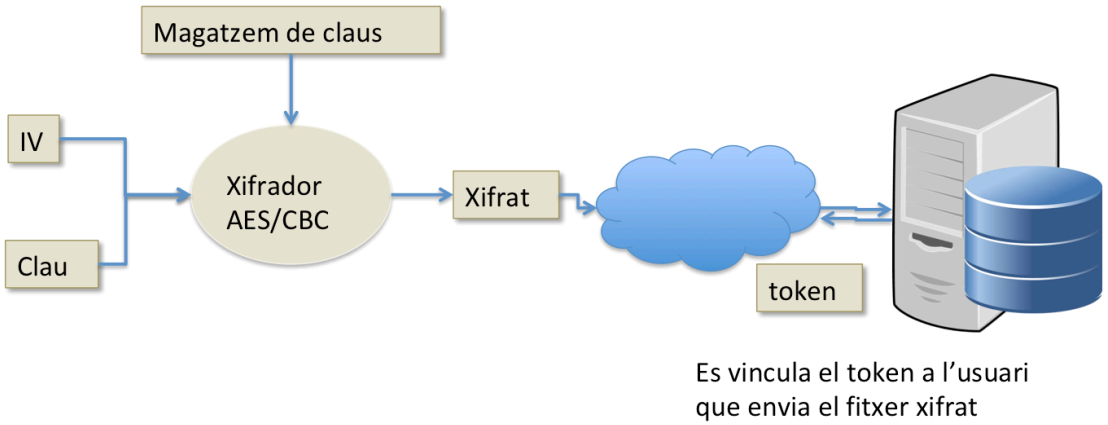
La solució passa per fer una còpia del fitxer de magatzem de claus. Aquesta còpia es fa de la següent manera:

- Es crea una clau que anomenarem vector d'inicialització (IV), de llargada 256 bits,
- Es crea una clau de 256 bits,
- Amb les dos claus s'encripta el fitxer de magatzem de claus amb l'algorisme AES256 en mode CBC [19]
- S'envia al servidor aquest fitxer encriptat.
- El servidor el guarda, i envia a l'usuari un token per a recuperar-lo.
- L'aplicació per tal de facilitar la tasca de guardar aquestes dades crea un codi QR [20] amb el codi IV, la clau i el token.

En el cas que l'usuari oblidi la contrasenya o perdi el mòbil pot llegir el codi QR amb un lector de codis i descarregar-se el fitxer del magatzem

de dades, descomprimir-lo amb les claus que estan en la imatge, i utilitzar-lo a l'aplicació.

És important indicar que la clau de xifratge no surt del dispositiu de l'usuari. Només surt el fitxer del magatzem xifrat. Això ens garanteix que el servidor no disposarà de la clau de desxifratge del magatzem de claus de l'usuari, mantenint la privacitat de l'usuari.



Il·lustració 22 Enviament del magatzem de claus xifrat

5. Joc de proves

Per fer les proves s'ha fet servir el següent entorn:

- El servidor s'ha executat en el IDE Eclipse Oxygen en un MacBook Pro del 2011 amb 16Gb de RAM DDR3 a 133MHz.
- El sistema de persistència del servidor ha estat una instancia de MySQL funcionant sobre la mateixa màquina.
- L'aplicació mòbil s'ha executat en l'emulador que proveeix el IDE de desenvolupament, ja que no es disposava de cap dispositiu Android.

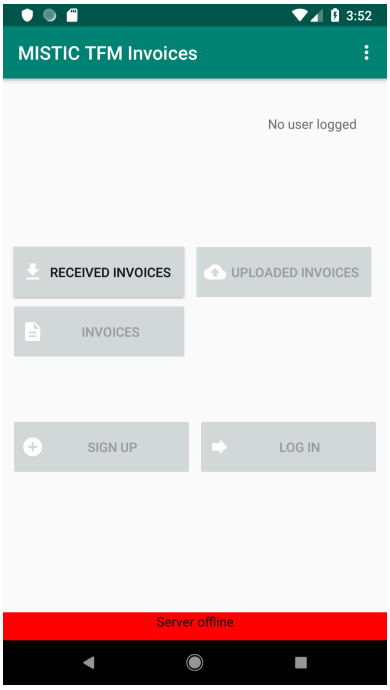
La metodologia ha estat provar cadascuna de les funcionalitats del servidor mitjançant scripts de test, i un cop verificat el seu funcionament bàsic, fer les proves amb l'aplicació mòbil i el joc de factures de prova.

Per a la part mòbil s'han provat totes les opcions desenvolupades obtenint els resultats que es mostren a continuació.

En els següents punts es descriuen i es mostren els resultats de les proves de cadascuna de les funcionalitats implementades.

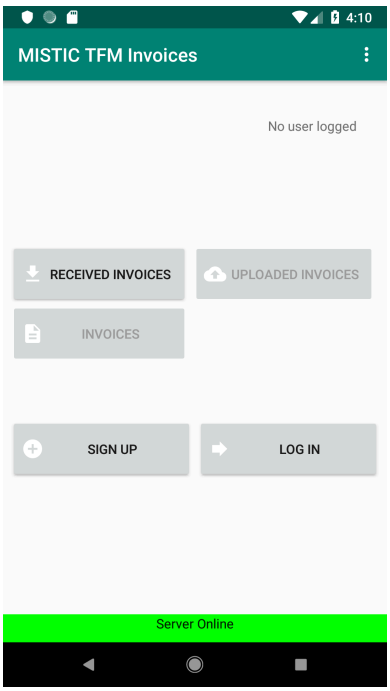
5.1 Inici de l'aplicació

En el primer moment d'inici de l'aplicació, en el cas que el servidor no estigui actiu es mostra la següent pantalla, en la que s'informa de la situació i en la que es pot veure els botons d'opcions deshabilitats. Aquest cas d'ús correspon al ReqMob-01, que indica que l'aplicació ha de poder consultar l'estat del servidor.



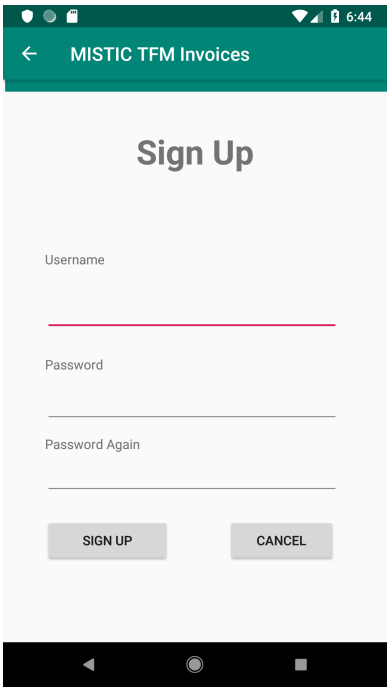
Il·lustració 24 Pantalla d'inici amb el servidor fóra de línia

En el moment en que l'aplicació detecta que hi ha connexió amb el servidor, aleshores les opcions de registre (sign up) i d'inici de sessió s'activen.



Il·lustració 25 Pantalla d'inici amb el servidor en línia

5.2 Operació de activació d'usuari

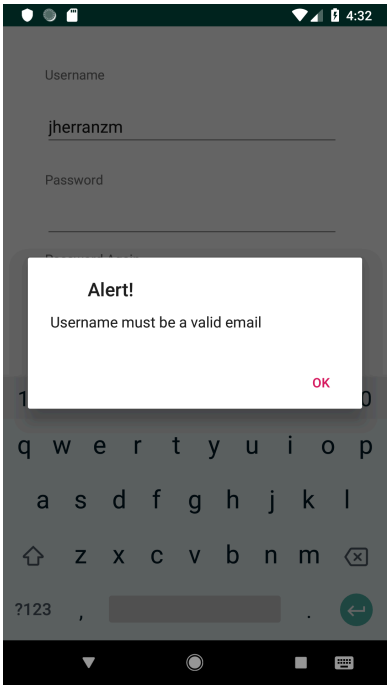


Il·lustració 26 Pantalla de registre d'usuari

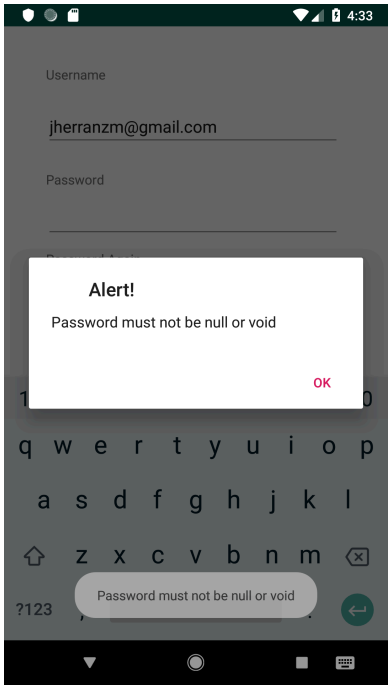
El cas que es mostra correspon al requisit ReqMob-02. En el cas que l'usuari no s'hagi registrat se li permet fer el registre mitjançant una pantalla de registre (Sign Up). En aquesta se li demana un usuari, que serà un correu electrònic vàlid, i una contrasenya amb les següents condicions requerides i explicades al punt 3.2.1.

Es valida que es compleixen les condicions establertes i s'envia la petició al servidor. Es correspon al requisit de servidor ReqSer-02.

En les imatges següents es mostren alguns dels errors que es poden donar en l'operació de registre d'un usuari.

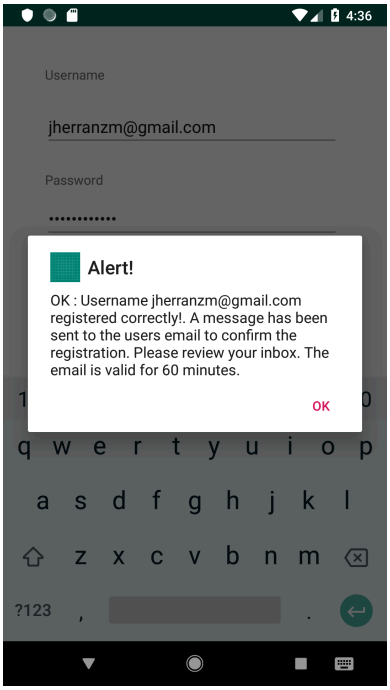


Il·lustració 27 Registre d'usuari: nom d'usuari no vàlid



Il·lustració 28 Registre d'usuari: contrasenya no vàlida

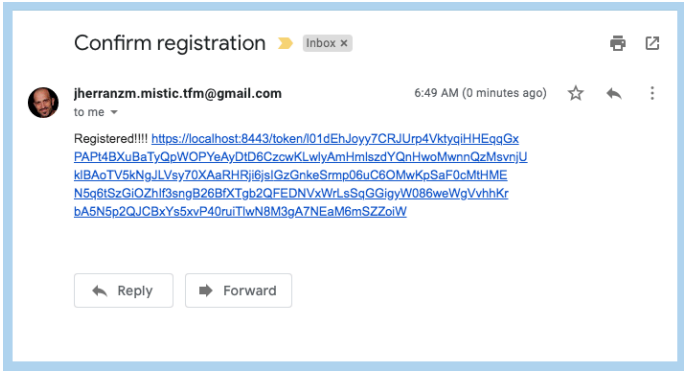
En el cas que l'usuari hagi ingressat un usuari vàlid i una contrasenya vàlida, aleshores es mostra un missatge com el següent:



Il·lustració 29 Registre d'usuari: usuari correctament registrat

En aquest moment l'usuari està creat a l'aplicació però no està habilitat, ja que encara no s'ha verificat que l'usuari sigui qui diu ser.

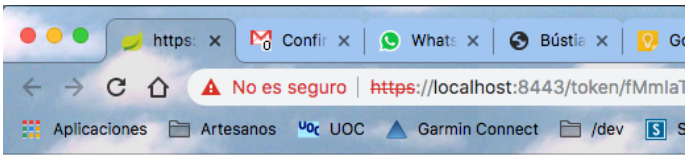
Per poder verificar l'usuari el servidor envia un correu electrònic (ReqSer-03) a l'adreça de correu proporcionada per l'usuari per tal que aquest en clicar-lo ho habiliti.



Il·lustració 30 Missatge per a verificar el correu de l'usuari

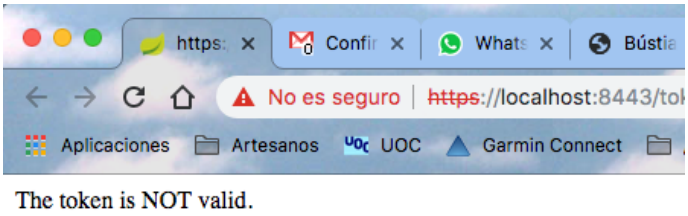
Un cop s'ha seleccionat l'enllaç ja està habilitat l'usuari, segons s'especifica al requeriment ReqSer-04. L'enllaç té una vida útil de 60

minuts. Al cap d'aquest temps caduca i s'ha de tornar a fer la operació de registre.



Il·lustració 31 Resposta a l'habilitació de l'usuari

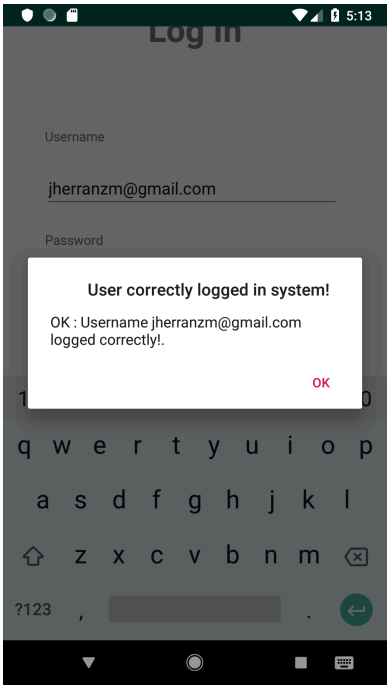
En el cas en que hagi passat més de 60 minuts o bé el "token" hagi estat utilitzat prèviament, aleshores s'informa a l'usuari.



Il·lustració 32 El correu ha caducat o bé ja ha estat utilitzat

5.3 Operació de inici de sessió d'usuari

Per a poder utilitzar l'aplicació l'usuari ha d'iniciar sessió. Accedeix a l'opció de "login" i introdueix l'usuari i contrasenya. El sistema valida prèviament que l'usuari sigui un correu electrònic formalment vàlid i que la contrasenya tingui la llargada mínima necessària. Aquestes dades s'envien al servidor per a completar l'operació d'inici de sessió.



Il·lustració 33 Usuari inicia correctament sessió

Al servidor es comprova que la combinació de usuari i contrasenya existeixin en el sistema, siguin vàlides i a més l'usuari estigui habilitat segons s'ha descrit en el punt 5.2.

En el moment en que l'usuari inicia la sessió, i l'aplicació mòbil rep el missatge de confirmació, si és la primera vegada, aleshores es genera un parell de claus pública i privada (requisit ReqMob-03). Aquestes claus es guarden al magatzem de claus de l'aplicació.

Amb aquestes claus es genera una petició signada de certificat, o CSR (Certificate Signing Request) (requisit ReqMob-04). Aquest s'envia al servidor per a que generi un certificat signat per la CA. Aquest certificat s'envia en el moment a l'aplicació (requisit ReqSer-05), i es guarda una còpia en una tabla del servidor. Un cop rebut per part de l'aplicació mòbil el certificat signat per la CA, s'incorpora al magatzem de dades (requisit ReqMob-05).

A partir d'aquest moment es generen les claus de xifratge simètriques per a desar les dades de factures en el servidor de manera segura.

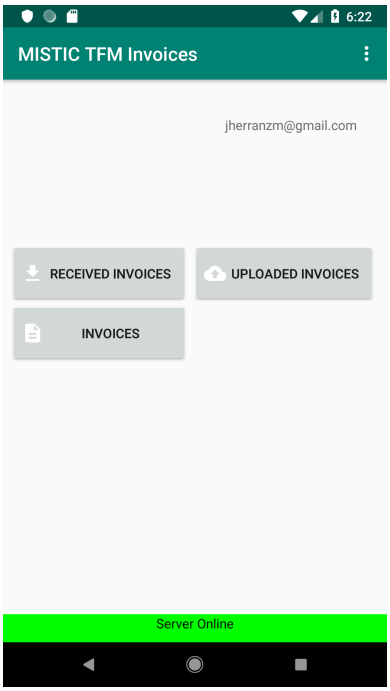
Es generen tantes claus com a camps es volen desar en el servidor per a que ajudin a l'usuari a identificar una factura prèviament guardada. En aquest cas s'ha decidit que els mínims camps necessaris per a identificar una factura són:

- Número de factura

- CIF del proveïdor
- Nom del proveïdor
- Data de la factura
- Import total de la factura
- Import dels impostos

Aquestes claus es guarden en el magatzem de claus de l'aplicació. Aquest magatzem només és accessible des de l'aplicació, pels permisos que li assigna aquesta en el moment de la creació, i a més està protegit per una contrasenya pròpia de l'aplicació.

Per seguretat, les claus s'encripten amb la clau pública de l'usuari i es desen en el servidor, de manera que només pugui ésser desxifrada per a la clau privada de l'usuari que com es pot comprovar no ha sortit de l'aplicació de l'usuari.



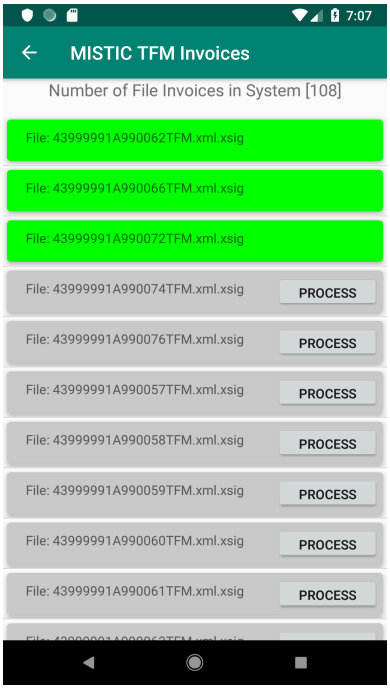
Il·lustració 34 Pantalla amb les opcions disponibles

5.4 Factures a ser tractades

L'aplicació mostra tres botons un cop l'usuari ha iniciat sessió. El botó de factures rebudes (Received Invoices) porta a una pantalla on es mostren els fitxers de factura que l'usuari ha rebut. Aquestes són les factures signades pels diferents proveïdors.

Hi ha un codi de colors per aquestes: si la factura ha estat prèviament tractada, i les dades estan desades a la base de dades local, aleshores

el ítem apareix en color verd. Si pel contrari el fitxer signat encara no ha estat tractat, aleshores apareix en color gris, i amb un botó habilitat per a fer el procés de validació, de guardar les dades en la base de dades local i si el servidor està disponible, fer la còpia encriptada al servidor.



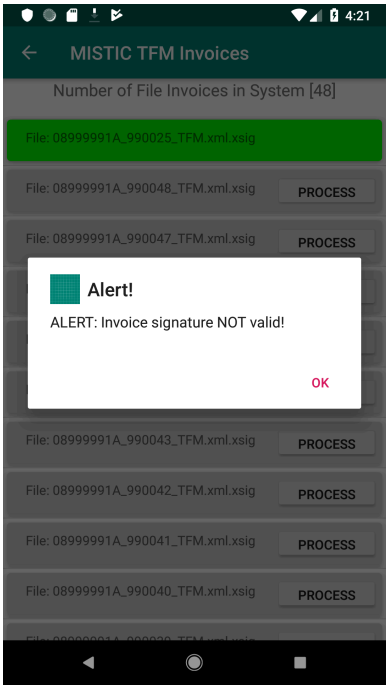
Il·lustració 35 Pantalla amb els fitxers de factura disponibles al sistema

En el cas de tenir ítems per processar l'usuari pot clicar sobre el botó "Process" i se li mostra un diàleg en que se li pregunta si vol processar el fitxer de factura seleccionat. En cas de fer clic sobre l'opció "Cancel" es torna a la pantalla del llistat de fitxers.

En cas de optar per "OK" el que es fa és:

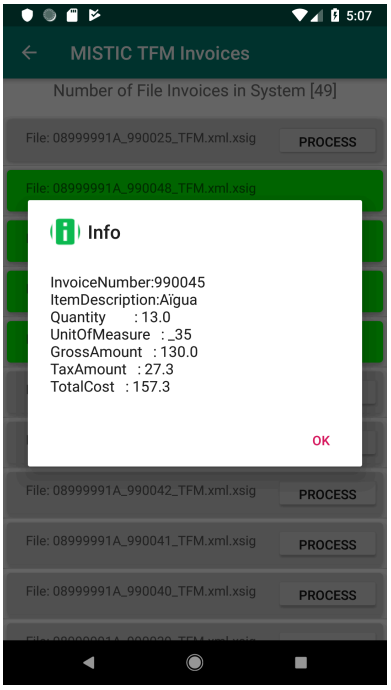
- Llegir el fitxer,
- Validar la signatura, que sigui correcta

Si la signatura dóna algun error aleshores el procés s'atura i es mostra un missatge d'alerta indicant la situació.

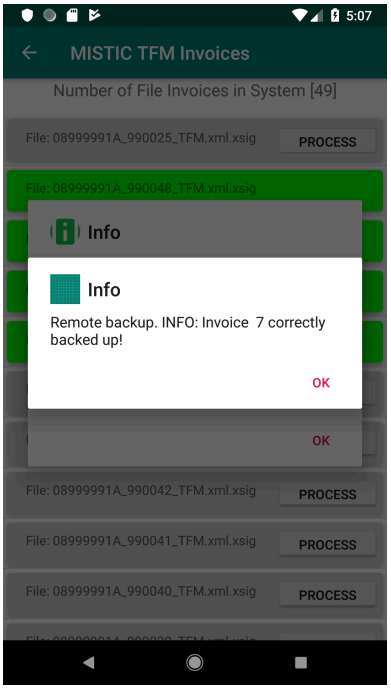


Il·lustració 36 El fitxer tractat té una signatura no vàlida

Si la signatura del fitxer és correcta, aleshores es mostra una pantalla d'informació de les dades bàsiques de la factura continguda en el fitxer tractat, i posteriorment es puja al servidor si hi ha connexió.



Il·lustració 37 Informació de la factura processada



Il·lustració 38 Informació de la còpia de seguretat de la factura al servidor

5.5 Factures pujades al servidor

En clicar sobre aquest botó en el cas que no s'hagi fet cap còpia de seguretat de factures, aleshores la pantalla que es mostra indica aquesta situació, com es pot veure a la Il·lustració 39

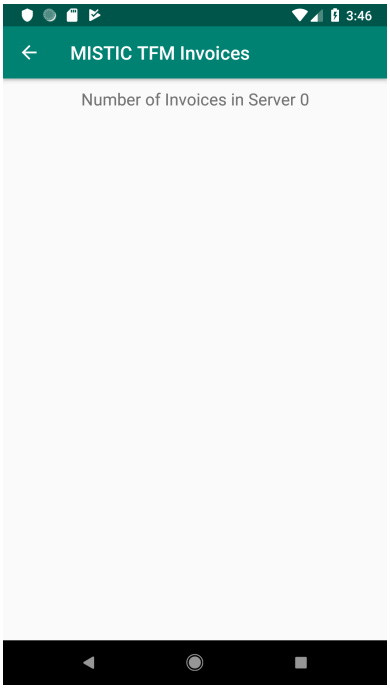
En el cas que l'usuari hagi fet servir l'opció de pujar factures al servidor, aleshores apareixeran a la pantalla del dispositiu en format de llista, com es pot apreciar a la Il·lustració 40

Les dades que es mostren es descarreguen encriptades del servidor. Un cop rebudes, s'han de desencriptar amb les claus simètriques de que disposa l'aplicació per l'usuari que té la sessió. Un cop desencriptades es mostren per a que l'usuari pugui identificar la factura que està en el servidor.

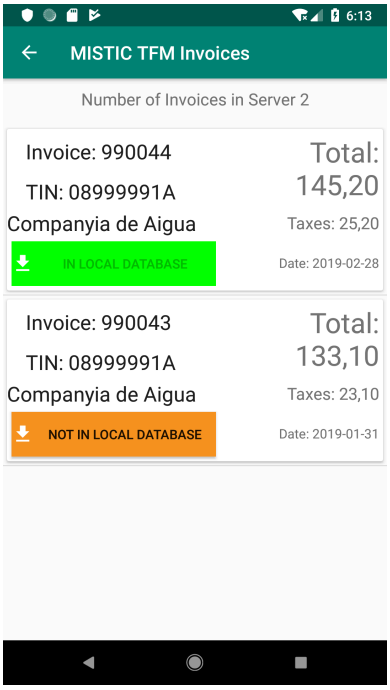
En el moment de desencriptar també es comprova si la factura està a la base de dades local. Tant si està com si no s'indica amb un literal i un codi de colors: verd, hi és; taronja, no hi és.

Es pot donar el cas que un usuari esborri dades de factura de la base de dades local, mentre que la còpia que hi ha al servidor es manté.

En el cas que l'usuari vulgui recuperar la còpia de la factura del servidor ho pot fer clicant el botó on s'indica que la factura no està a la base de dades local.



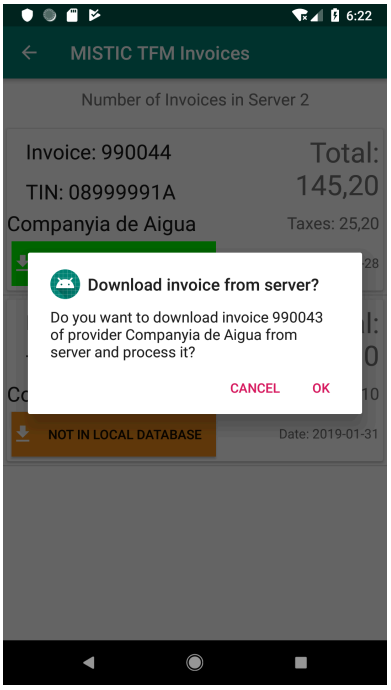
Il·lustració 39 No hi ha factures al servidor



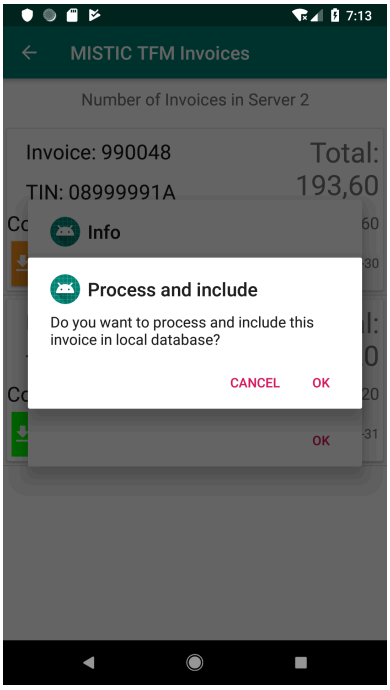
Il·lustració 40 Hi ha factures al servidor

Si l'usuari decideix descarregar-se la factura que no es troba a la base de dades local, aleshores el sistema li demana confirmació. Si l'usuari accepta aleshores el sistema li envia el fitxer de factura original xifrat amb la clau simètrica corresponent.

L'aplicació mòbil la desxifra, processa el fitxer i incorpora les dades a la base de dades local, de manera similar al procés fet amb els arxius locals.



Il·lustració 41 Descàrrega d'una factura del servidor



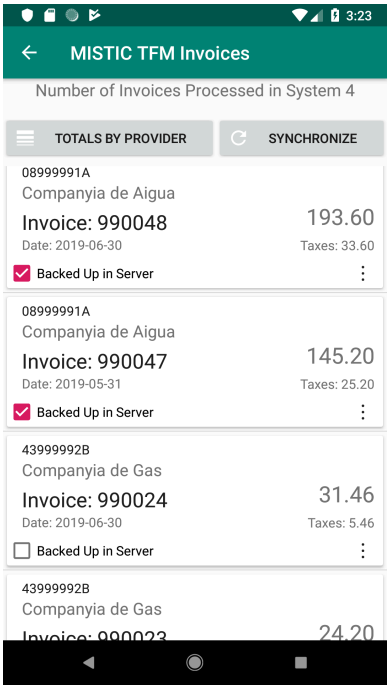
Il·lustració 42 Confirmació per a processar la factura descarregada

5.6 Factures processades

Les factures, tant si han estat processades a partir dels fitxers en local com si han estat descarregades de la còpia que hi ha al servidor, es desen en la base de dades local.

Aquesta base de dades s'ha dissenyat per a guardar dades bàsiques de la factura per tal de mostrar unes dades estadístiques.

Si l'usuari selecciona la opció factures (*Invoices*) de la pantalla principal de l'aplicació es mostra una llista de les factures prèviament validades i emmagatzemades al dispositiu, tal i com es mostra a la Il·lustració 43.



Il·lustració 43 Factures carregades al sistema

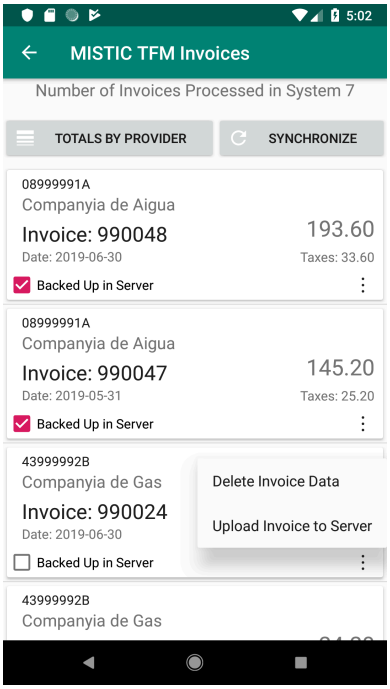
Des d'aquesta pantalla l'usuari pot desplaçar-se pel llistat de les factures que es mostren. En cada fitxa de factura es pot veure en la part de l'esquerra: el CIF del proveïdor, el nom del mateix, el número de factura i la data. A la dreta el número que apareix és l'import total, i a sota els impostos.

En la part inferior de cada fitxa de factura apareix una informació de si té o no còpia de seguretat al servidor, amb la casella de verificació (*Backed Up in Server*).

Les operacions que es poden fer amb aquestes factures estan definides al menú contextual que apareix en seleccionar els punts verticals que es mostren en la part inferior dreta, tal i com es pot observar a la Il·lustració 44 Menú contextual d'opcions amb la factura.. Les operacions implementades són: eliminar la factura seleccionada, i pujar la factura al servidor en cas que no hagi estat carregada.

En el cas de eliminar la factura de la base de dades local, es pot recuperar de la còpia que es té en el servidor, tal i com s'explica en el punt 5.5.

En el cas en que l'usuari hagi processat el fitxer de factura signada sense disposar de connexió amb el servidor no s'ha pogut fer la còpia. En el moment en que torna a disposar de connexió s'activa l'opció de fer la còpia de seguretat de la factura.

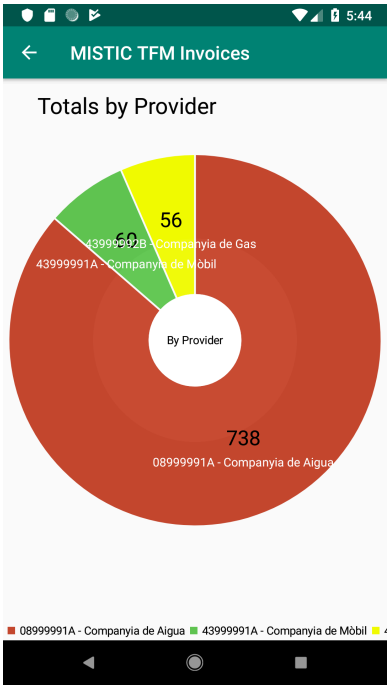


Il·lustració 44 Menú contextual d'opcions amb la factura.

L'usuari disposa en la mateixa pantalla de dues opcions: una per veure els totals per proveïdor, i una altra per a sincronitzar les dades que hi ha a la base de dades local amb la remota. En aquesta última es marquen o desmarquen les factures en funció de si tenen o no còpia en el servidor. Així l'usuari pot tenir controlat si té o no còpia de seguretat de les seves factures.

La pantalla de totals per proveïdor mostra el total de les factures agrupades per l'emissor de les mateixes. En la mateixa pantalla es pot escollir l'opció de presentar les dades en format de gràfic circular.

Il·lustració 45 Totals per proveïdor

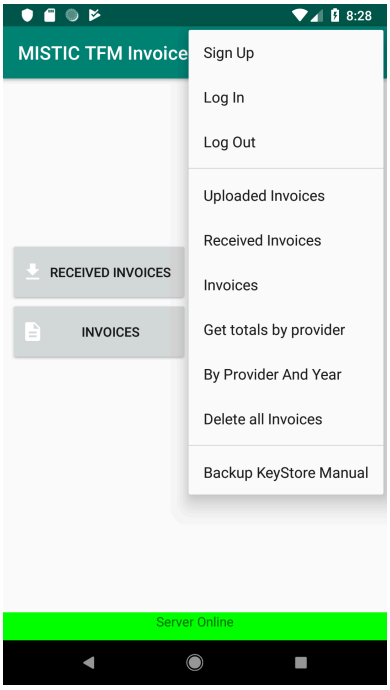


Il·lustració 46 Gràfic totals per proveïdor

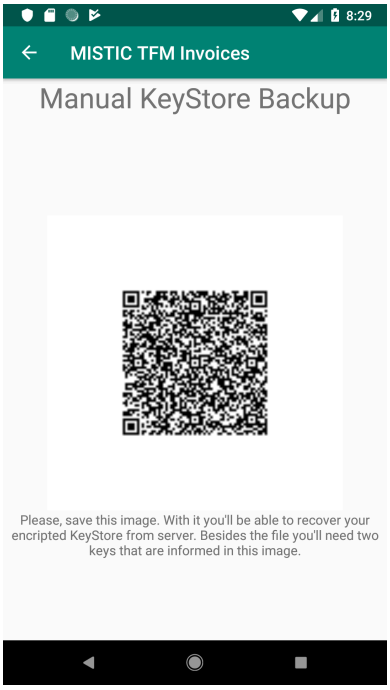
5.7 Còpia de seguretat del magatzem de claus

L'usuari pot fer manualment una còpia del magatzem de dades de l'aplicació. En el menú de la pantalla principal hi ha una opció que permet fer aquesta operació.

Un cop seleccionada l'opció s'encripta el fitxer i s'envia al servidor. Amb les claus de xifratge, junt amb el codi de resposta del servidor es genera un codi QR que permetrà recuperar el magatzem i desxifrar-lo.



Il·lustració 47 Opció de còpia de seguretat del magatzem de dades



Il·lustració 48 Codi QR amb les dades que permeten recuperar i desxifrar el magatzem de dades

6. Conclusions

En aquest treball s'ha desenvolupat un sistema que permet que un usuari validi i guardi les seves factures electròniques (Facturae) amb el seu telèfon mòbil. Aquesta aplicació permet garantir que les factures obtingudes són íntegres, autèntiques i que ningú (a banda de l'usuari) hi podrà accedir (confidencialitat). En tot el procés s'ha tingut cura de la seva usabilitat i sobretot la seguretat en tots els passos del procés. Les proves realitzades han permès validar que els usuaris disposen d'un sistema format per dos elements en el que es permet guardar les factures electròniques en el dispositiu mòbil i disposar d'una còpia xifrada al servidor, garantint que el servidor en cap moment disposa de les claus per a poder desxifrar les dades que l'usuari guarda.

Personalment ha estat tot un repte, ja que el fet de no haver treballat mai amb Android m'ha suposat en més d'una ocasió dedicar més temps del que hagués considerat a tractar de resoldre problemes que no tenien a veure amb la seguretat i si amb les característiques del sistema.

Al principi del projecte amb el meu tutor del treball hem tractar força sobre qui havia de fer què en el sistema. El focus d'aquest treball era fer que l'usuari pogués desar informació de manera segura en un entorn "hostil". Sobre aquesta premissa s'ha conclòs que no es podia deixar al servidor que fes els càlculs estadístics i d'agregació que en un principi s'havia pensat. La raó era que per fer-los o bé s'havien de pujar les dades en clar, o bé el servidor havia de desxifrar-les, amb la qual cosa ens veiem obligats a facilitar-li les claus pública i privada. Això trencava amb la premissa de la privacitat de l'usuari, ja que el servidor podia disposar de tota la informació amb aquestes claus.

Línies de treball futur.

El projecte té diverses línies de treball que es poden continuar.

- Una és el fet de fer una validació més exhaustiva dels fitxers de factures signades. En una extensió es tractaria de donar més informació de les raons per les que les factures no són vàlides. Ara mateix és una validació binària: és o no vàlida, sense especificar més.
- Una altra línia de treball és estendre la informació que es desa de cada factura en la base de dades local.
- D'aquí es podria permetre que l'usuari treies més informació de les dades que emmagatzema en el seu dispositiu mòbil.

7. Glossari

XML : de l'anglès eXtensible Markup Language (“llenguatge de marques extensible”), és un metallenguatge extensible, d'etiquetes, desenvolupat pel World Wide Web Consortium (W3C)

XADES: XML Advanced Electronic Signatures

XADES-EPES: XADES amb informació addicional sobre la política de signatura, quin certificat s’ha empleat i quina entitat certificadora ho ha emés.

TCP/IP : Transmission Control Protocol/Internet Protocol (Protocol de control de transmissió/Protocol de internet).

8. Bibliografia

[1] Ley 56/2007, de 28 de diciembre, de Medidas de Impulso de la Sociedad de la Información. https://www.boe.es/buscar/act.php?id=BOE-A-2007-22440
[2] Format “Facturae” https://www.boe.es/buscar/doc.php?id=BOE-A-2007-18009
[3] Spring Boot: https://spring.io/projects/spring-boot
[4] Plataforma Spring: https://spring.io/
[5] Convenció sobre configuració: https://ca.wikipedia.org/wiki/Convenci%C3%B3_sobre_configuraci%C3%B3
[6] PostgreSQL : https://www.postgresql.org/
[7] MySQL : https://www.mysql.com/
[8] SQL Server : https://www.microsoft.com/es-es/sql-server/sql-server-2019
[9] Oracle : https://docs.oracle.com/en/database/oracle/oracle-database/index.html
[10] Android market share: http://gs.statcounter.com/os-market-share/mobile/worldwide
[11] Open Handset Alliance: https://www.openhandsetalliance.com/
[12] GPL : https://opensource.org/licenses/GPL-2.0
[13] ASL : Apache Software License https://www.apache.org/licenses/LICENSE-2.0
[14] Apache Santuario : https://santuario.apache.org/
[15] Inversió de Control: https://en.wikipedia.org/wiki/Inversion_of_control
[16] Mètode POST: https://tools.ietf.org/html/rfc7231#section-4.3.3
[17] JSON: https://tools.ietf.org/html/rfc7159
[18] Base64: https://tools.ietf.org/html/rfc4648#section-5
[19] CBC mode of operation: https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Cipher_Block_Chaining_(CBC)
[20] Codi QR: https://ca.wikipedia.org/wiki/Codi_QR
[21] JIBX: http://jibx.sourceforge.net/

9. Annexos

Llistat d'apartats que són massa extensos per incloure dins la memòria i tenen un caràcter autocontingut (per exemple, manuals d'usuari, manuals d'instal·lació, etc.)

Depenent del tipus de treball, és possible que no calgui afegir cap annex.

9.1 Instal·lació de l'entorn de desenvolupament

Per a fer el desenvolupament del projecte ha calgut fer diverses instal·lacions. En primer lloc s'ha de disposar d'una màquina virtual Java. El desenvolupament en un sistema operatiu Mac i es té instal·lada una versió 1.8.0_45

```
MacBookPro-de-JoseLuis:~ jherranzm$ java -version
java version "1.8.0_45"
Java(TM) SE Runtime Environment (build 1.8.0_45-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)
```

9.1.1 Servidor

Per a la persistència de les dades en el servidor s'ha de disposar d'un sistema de base de dades. En el cas que ens ocupa ja es tenia disponible i funcionant una base de dades MySQL Community Server 5.5.23

```
MacBookPro-de-JoseLuis:~ jherranzm$ mysql -v
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10182
Server version: 5.5.23-log MySQL Community Server (GPL)
```

Per a fer les diverses consultes i creació de taules de la base de dades es fa servir el programari MySQL Workbench versió 6.3 per a Mac.

Per al desenvolupament de l'aplicació del servidor es fa servir com a IDE una instal·lació de Eclipse Java EE IDE for Web Developers versió Oxygen.3a Release (4.7.3a)

Per al desenvolupament ràpid de l'aplicació s'ha escollit la plataforma Spring Boot [3], ja que permet desenvolupar i posar en marxa una

aplicació web d'una manera àgil. Dintre de la configuració es pot fer que arrenqui un servidor d'aplicacions Tomcat que està integrat en la plataforma. D'aquesta manera es poden fer proves de la mateixa aplicació dintre del entorn de programació. De fet les proves que s'han fet han estat mitjançant el servidor d'aplicacions integrat.

9.1.2 Aplicació mòbil

Per a fer el desenvolupament de l'aplicació mòbil ha calgut instal·lar el IDE Android Studio en la seva versió 3.3.2

La part de persistència de dades de l'aplicació mòbil ve donada per la configuració de l'aplicació per la qual cosa no cal cap altra instal·lació.

9.2 Creació de la infraestructura de clau pública

```
#!/bin/sh

function showTitle {
	echo "*"
	echo "*"
	echo "*"
	echo $1
	echo "*"
}

BASEFILE=~/Dropbox/Jose_Luis/TFM_2019
JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0_45.jdk/Contents/Home

PASS=Th2S5p2rStr4ngP1ss
DEFAULT_PASS=changeit
LONG_CLAU_CA=8192
LONG_CLAU_IND=4096

CA_NAME=CA_TFM

SERVER_NAME=Server
USER_NAME=Usuari

PROJECT=AppTestValidationAndroid44
#LONG=2048
```

```
rm -rf $BASEFILE/certs
rm -rf $BASEFILE/PKI
rm -rf $BASEFILE/CA.key
rm -rf $BASEFILE/server.key
rm -rf $BASEFILE/usuari.key

showTitle "Generació de la parella de clau pública-privada de la CA!"
openssl genrsa \
      -aes256 \
      -f4 \
      -passout pass:$PASS \
      -out $BASEFILE/CA.key \
      $LONG_CLAU_CA
if [ ! -f $BASEFILE/CA.key ]; then
      echo "ERROR en la generació de la parella de clau pública-privada de la
CA!"
      exit
fi

showTitle "Generació de la parella de clau pública-privada del servidor!"
openssl genrsa \
      -aes256 \
      -f4 \
      -passout pass:$PASS \
      -out $BASEFILE/server.key \
      $LONG_CLAU_IND
if [ ! -f $BASEFILE/server.key ]; then
      echo "ERROR en la generació de la parella de clau pública-privada del
servidor!"
      exit
fi
```

```
showTitle "Generació de la parella de clau pública-privada de l'usuari!"
openssl genrsa \
	-aes256 \
	-f4 \
	-passout pass:$PASS \
	-out $BASEFILE/usuari.key \
	$LONG_CLAU_IND
if [ ! -f $BASEFILE/usuari.key ]; then
	echo "ERROR en la generació de la parella de clau pública-privada de l'usuari!"
	exit
fi

#(la contrasenya que tu has utilitzat és molt millor que la que poso d'exemple!!)

showTitle "Generació del certificat de la CA!"
openssl req \
	-new \
	-sha256 \
	-x509 \
	-key $BASEFILE/CA.key \
	-out $BASEFILE/CA.crt \
	-days 730 \
	-passin pass:$PASS \
	-config $BASEFILE/opensslPKISSL.cnf \
	-subj "/C=ES/ST=Catalunya/L=Barcelona/O=Universitat Oberta de Catalunya/OU=MISTIC/CN=CA TFM 2019"
if [ ! -f $BASEFILE/CA.crt ]; then
	echo "ERROR en la generació del certificat de la CA!"
	exit
fi
```

```
showTitle "Generació del CSR del Server!"
openssl req \
       -new \
       -sha256 \
       -key $BASEFILE/server.key \
       -out $BASEFILE/server.csr \
       -passin pass:$PASS \
       -subj     "/C=ES/ST=Catalunya/L=Barcelona/O=Universitat     Oberta     de
Catalunya/OU=MISTIC/CN=Server" \
       -passout pass:$PASS \
       -config $BASEFILE/opensslPKISSL.cnf
if [ ! -f $BASEFILE/server.csr ]; then
       echo "ERROR en la generació del CSR del Server!"
       exit
fi

showTitle "Generació del CSR de l'usuari!"
openssl req \
       -new \
       -sha256 \
       -config $BASEFILE/opensslPKISSL.cnf \
       -key $BASEFILE/usuari.key \
       -out $BASEFILE/usuari.csr \
       -passin pass:$PASS \
       -passout pass:$PASS \
       -subj     "/C=ES/ST=Catalunya/L=Barcelona/O=Universitat     Oberta     de
Catalunya/OU=MISTIC/CN=Usuari"
if [ ! -f $BASEFILE/usuari.csr ]; then
       echo "ERROR en la generació del CSR de l'usuari!"
       exit
fi
```

```
#Abans de seguir cal copiar o crear l'estructura definida al fitxer de
configuració adjunt.


showTitle "Creació de l'estructura de directoris"


mkdir PKI
cd PKI
mkdir certs
mkdir crl
mkdir newcerts
mkdir private
touch index.txt
echo "01" > serial
cp $BASEFILE/CA.key private
cp $BASEFILE/CA.crt certs



showTitle "Generació del P12 de la CA"
openssl pkcs12 \
      -export \
      -in $BASEFILE/PKI/certs/CA.crt \
      -inkey $BASEFILE/PKI/private/CA.key \
      -certfile $BASEFILE/PKI/certs/CA.crt \
      -out $BASEFILE/PKI/CAkeystore.p12 \
      -passin pass:$PASS \
      -passout pass:$PASS \
      -name "ca" \
      -caname $CA_NAME
```

```
showTitle "Generació del Certificat del Server..."

openssl ca \
       -batch \
       -config $BASEFILE/opensslPKISSL.cnf \
       -keyfile $BASEFILE/PKI/private/CA.key \
       -cert $BASEFILE/PKI/certs/CA.crt \
       -extensions v3_ssl \
       -passin pass:$PASS \
       -out $BASEFILE/PKI/certs/server.crt \
       -infiles $BASEFILE/server.csr
if [ ! -f $BASEFILE/PKI/certs/server.crt ]; then
       echo "ERROR en la generació del Certificat del Server!"
       exit
fi

showTitle "Generació del Certificat de l'usuari..."

openssl ca \
       -batch \
       -config $BASEFILE/opensslPKISSL.cnf \
       -keyfile $BASEFILE/PKI/private/CA.key \
       -cert $BASEFILE/PKI/certs/CA.crt \
       -extensions usr_cert \
       -passin pass:$PASS \
       -out $BASEFILE/PKI/certs/usuari.crt \
       -infiles $BASEFILE/usuari.csr
if [ ! -f $BASEFILE/PKI/certs/usuari.crt ]; then
       echo "ERROR en la generació del Certificat de l'usuari!"
       exit
fi
```

```
showTitle "Generació del Magatzem de claus del servidor..."

openssl pkcs12 \
       -export \
       -in $BASEFILE/PKI/certs/server.crt \
       -inkey $BASEFILE/server.key \
       -passin pass:$PASS \
       -CAfile $BASEFILE/PKI/certs/CA.crt \
       -name $SERVER_NAME \
       -caname $CA_NAME \
       -out $BASEFILE/PKI/private/server.p12 \
       -passout pass:$PASS \
       -aes256 \
       -chain
if [ ! -f $BASEFILE/PKI/private/server.p12 ]; then
       echo "ERROR en la generació del Magatzem de claus del servidor!"
       exit
fi

showTitle "Generació del Magatzem de claus de l'usuari..."

openssl pkcs12 \
       -export \
       -in $BASEFILE/PKI/certs/usuari.crt \
       -inkey $BASEFILE/usuari.key \
       -passin pass:$PASS \
       -CAfile $BASEFILE/PKI/certs/CA.crt \
       -name $USER_NAME \
       -caname $CA_NAME \
       -out $BASEFILE/PKI/private/usuari.p12 \
       -passout pass:$PASS \
       -aes256 \
       -chain
if [ ! -f $BASEFILE/PKI/private/usuari.p12 ]; then
       echo "ERROR en la generació del Magatzem de claus de l'usuari!"
       exit
fi

showTitle "Verificació dels certificats..."
openssl verify \
       -CAfile $BASEFILE/PKI/certs/CA.crt $BASEFILE/PKI/certs/server.crt
```

```
openssl verify \
      -CAfile $BASEFILE/PKI/certs/CA.crt $BASEFILE/PKI/certs/usuari.crt


showTitle "Còpia dels certificats i claus a res/raw..."
cp $BASEFILE/PKI/certs/CA.crt
/Users/jherranzm/AndroidStudioProjects/$PROJECT/app/src/main/res/raw/ca.crt
cp $BASEFILE/PKI/certs/server.crt
/Users/jherranzm/AndroidStudioProjects/$PROJECT/app/src/main/res/raw/server.cr
t
cp $BASEFILE/PKI/private/server.p12
/Users/jherranzm/AndroidStudioProjects/$PROJECT/app/src/main/res/raw/serverkey
.p12


showTitle "Còpia dels certificats i claus a assets..."
cp $BASEFILE/PKI/certs/CA.crt
/Users/jherranzm/AndroidStudioProjects/$PROJECT/app/src/main/assets/ca.crt
cp $BASEFILE/PKI/certs/server.crt
/Users/jherranzm/AndroidStudioProjects/$PROJECT/app/src/main/assets/server.crt
cp $BASEFILE/PKI/private/server.p12
/Users/jherranzm/AndroidStudioProjects/$PROJECT/app/src/main/assets/serverkey.
p12


showTitle "Còpia dels certificats i claus al Servidor..."
cp $BASEFILE/PKI/certs/server.crt
/Users/jherranzm/git/uoc.mistic.tfm.rest.server/rest-
server/src/main/resources/keystore/server.crt
cp $BASEFILE/PKI/private/server.p12
/Users/jherranzm/git/uoc.mistic.tfm.rest.server/rest-
server/src/main/resources/keystore/serverkey.p12


showTitle "Procés finalitzat!"
```

9.3 Creació de base de dades i taules al servidor

```
CREATE DATABASE tfm;
use tfm;
drop table if exists tfm.tbl_keystore;
drop table if exists tfm.tbl_back_up;
drop table if exists tfm.tbl_invoice;
drop table if exists tfm.tbl_invoice_data;
drop table if exists tfm.tbl_kkeys;
drop table if exists tfm.tbl_token;
drop table if exists tfm.tbl_usuari_role;
drop table if exists tfm.tbl_role;
drop table if exists tfm.tbl_usuari;
```

```
CREATE TABLE tfm.tbl_usuari (
  id bigint(20) NOT NULL AUTO_INCREMENT,

  username VARCHAR(45) NOT NULL,
  pass VARCHAR(255) NOT NULL,
  certificate LONGTEXT,
  email VARCHAR(255) NOT NULL,
  enabled tinyint(1) NOT NULL DEFAULT '0',
  comentaris LONGTEXT,
  creation_time TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,

  PRIMARY KEY (id),
  UNIQUE KEY username_UNIQUE (username),
  UNIQUE KEY email_UNIQUE (email)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;

CREATE TABLE tfm.tbl_role (
  id   bigint(20) NOT NULL AUTO_INCREMENT,
  nom VARCHAR(255) NOT NULL,
  comentaris LONGTEXT,
  creation_time TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (id),
  UNIQUE KEY nom_UNIQUE (nom)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;

CREATE TABLE tfm.tbl_usuari_role (
  id bigint(20) NOT NULL AUTO_INCREMENT,
  usuari_id  bigint(20) NOT NULL,
  role_id bigint(20) NOT NULL,
  creation_time TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (id),
  UNIQUE KEY usuari_role (usuari_id,role_id),
  KEY tfm_usuari_role_ibfk_2 (role_id),
  CONSTRAINT tfm_usuari_role_ibfk_1 FOREIGN KEY (usuari_id) REFERENCES
tfm.tbl_usuari (id) ON DELETE CASCADE,
  CONSTRAINT tfm_usuari_role_ibfk_2 FOREIGN KEY (role_id) REFERENCES
tfm.tbl_role (id) ON DELETE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;

CREATE TABLE tfm.tbl_token (
  id bigint(20) NOT NULL AUTO_INCREMENT,
```

```
  usuari_id bigint(20) NOT NULL,
  token longtext NOT NULL,
  resum VARCHAR(32) not null,
  valid_from DATETIME not null,
  valid_to DATETIME not null,
  used tinyint(1) NOT NULL DEFAULT '0',
  creation_time TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (id),
  unique u_resum (resum),
  CONSTRAINT tfm_token_fk_1 FOREIGN KEY (usuari_id) REFERENCES tfm.tbl_usuari
(id) ON DELETE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;


CREATE TABLE tfm.tbl_kkeys (
  id bigint(20) NOT NULL AUTO_INCREMENT,
  usuari_id bigint(20) NOT NULL,
  f VARCHAR(100) NOT NULL comment 'field',
  k longtext     NOT NULL comment 'Encrypted Symmetric Key',
  creation_time TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (id),
  UNIQUE u_f (f),
  CONSTRAINT tfm_kkeys_fk_1 FOREIGN KEY (usuari_id) REFERENCES tfm.tbl_usuari
(id) ON DELETE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;


CREATE TABLE tfm.tbl_back_up (
  id bigint(20) NOT NULL AUTO_INCREMENT,
  usuari_id bigint(20) NOT NULL,


  f1 VARCHAR(100) NOT NULL comment 'UID',
  i longtext NOT NULL comment 'Initialization Vector',
  k longtext NOT NULL comment 'Simmetric Key',
  f longtext NOT NULL comment 'Encrypted Invoice File',


  creation_time TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (id),
  UNIQUE u_f1 (f1),
  CONSTRAINT tfm_back_up_fk_1 FOREIGN KEY (usuari_id) REFERENCES
tfm.tbl_usuari (id) ON DELETE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;


CREATE TABLE tfm.tbl_invoice_data (
  id bigint(20) NOT NULL AUTO_INCREMENT,
```

```
  usuari_id bigint(20) NOT NULL,
  f1 VARCHAR(100) NOT NULL comment 'UID',
  f2 VARCHAR(255) NOT NULL comment 'Tax Identification Number',
  f3 VARCHAR(255) NOT NULL comment 'Corporate Name',
  f4 VARCHAR(255) NOT NULL comment 'Invoice Number',
  f5 VARCHAR(255) NOT NULL comment 'Invoice Total',
  f6 VARCHAR(255) NOT NULL comment 'Total Gross Amount',
  f7 VARCHAR(255) NOT NULL comment 'Total Tax Outputs',
  f8 VARCHAR(255) NOT NULL comment 'Issue Date',
  creation_time TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (id),
  UNIQUE u_f1 (f1),
  CONSTRAINT tfm_invoice_data_fk_1 FOREIGN KEY (usuari_id) REFERENCES
tfm.tbl_usuari (id) ON DELETE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;


CREATE TABLE tfm.tbl_invoice (
      id bigint(20) NOT NULL AUTO_INCREMENT,
      usuari_id bigint(20) NOT NULL,
      uid VARCHAR(100) NOT NULL,
      tax_identification_number VARCHAR(255) NOT NULL,
      corporate_name VARCHAR(255) NOT NULL,
      invoice_number VARCHAR(255) NOT NULL,
      invoice_total VARCHAR(255) NOT NULL,
      total_Tax_outputs VARCHAR(255) NOT NULL,
      issue_date VARCHAR(255) NOT NULL,
      signed_invoice LONGTEXT NOT NULL,
      iv LONGTEXT NOT NULL,
      sim_key LONGTEXT NOT NULL,
      data_creacio TIMESTAMP NOT NULL default CURRENT_TIMESTAMP,
      PRIMARY KEY (id),
      CONSTRAINT tfm_invoice_fk_1 FOREIGN KEY (usuari_id) REFERENCES
tfm.tbl_usuari (id) ON DELETE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;


DROP TABLE IF EXISTS tfm.tbl_keystore;
CREATE TABLE tfm.tbl_keystore (
  id bigint(20) NOT NULL AUTO_INCREMENT,
  usuari_id bigint(20) NOT NULL,
  iv LONGTEXT NOT NULL,
  keystore LONGTEXT NOT NULL,
```

```
  token LONGTEXT NOT NULL,
  creation_time TIMESTAMP DEFAULT '0000-00-00 00:00:00',
  update_time TIMESTAMP NOT NULL default CURRENT_TIMESTAMP on update
CURRENT_TIMESTAMP,
  PRIMARY KEY (id),
  unique u_id (id),
  CONSTRAINT tfm_keystore_fk_1 FOREIGN KEY (usuari_id) REFERENCES
tfm.tbl_usuari (id) ON DELETE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;


INSERT INTO tfm.tbl_role
(nom, comentaris)
VALUES ('ROLE_USER', ''),
('ROLE_ADMIN', ''),
('ROLE_DBA', ''),
('ROLE_CONSULTA', '');
```