

Xavier Quintana i Vidal

SECURE SYSTEM FOR PERSONAL FINANCES ON THE CLOUD

TREBALL FINAL DE MÀSTER

dirigit pel Dr. Jordi Castellà-Roca

Màster Interuniversitari en Seguretat de les Tecnologies de la Informació i Comunicació
- *MÍSTIC*

Universitat Oberta de Catalunya, 2014-2015

Vull agraïr, i espero també dedicar, aquest projecte a una persona molt especial a la meva vida, que si no l'ha patit com jo, és perquè ho ha fet encara més. T'estimo, Roser.

Tampoc seria just no dedicar unes paraules d'agraïment al meu tutor, que segurament dec haver sigut dels alumnes més complicats amb qui ha hagut de treballar. Moltes gràcies per tot el suport tècnic i moral, Jordi.

Per acabar agraïr al meu company de feina, Esteve, les nombroses xerrades que hem tingut entorn a la gestió de treballs d'aquests tipus, i el recolzament que m'ha donat en moments més crítics. Gràcies amic!

0. ÍNDEX

0. ÍNDEX.....	4
1. INTRODUCCIÓ I OBJECTIUS.....	6
1.1. INTRODUCCIÓ	6
1.2. MOTIVACIONS	7
1.3. OBJECTIUS.....	7
1.4. ORGANITZACIÓ DE LA MEMÒRIA.....	8
2. METODOLOGIA	9
3. PLANIFICACIÓ	13
4. MARC DE TREBALL I CONCEPTES PREVIS	16
4.1. INTRODUCCIÓ AL BLUETOOTH.....	16
4.1.1. QUÈ ÉS EL BLUETOOTH?.....	16
4.1.2. BREU HISTÒRIA DEL BLUETOOTH.....	17
4.1.3. SEGURETAT EN LES COMUNICACIONS.....	18
4.2. LA FACTURA ELECTRÒNICA	18
4.3. EL PATRÓ MVC	18
4.3.1. QUÈ ÉS EL PATRÓ MVC?.....	19
4.3.2. AVANTATGES.....	19
4.3.3. INCONVENIENTS	20
5. DISSENY I ARQUITECTURA.....	21
5.1. REQUISITS.....	21
5.1.1. REQUISITS FUNCIONALS.....	21
5.1.2. REQUISITS SEGURETAT.....	21
5.2. ESTUDIS I DECISIONS.....	22
5.3. ARQUITECTURA	25
5.4. DISSENY DEL SISTEMA.....	26
5.4.1. APLICACIÓ LOCAL.....	26
5.4.2. APLICACIÓ ANDROID	26
5.4.3. APLICACIÓ WEB	27
6. DESENVOLUPAMENT	28
6.1. SERVIDOR DE FACTURES	28

6.2. APLICACIÓ ANDROID	29
6.3. SERVEI AL NÚVOL.....	31
7. JOC DE PROVES	34
7.1 ENVIAMENT D'UNA FACTURA	34
7.2 APLICACIÓ ANDROID	35
7.3. SERVEI AL NÚVOL.....	39
8. CONCLUSIONS	44
8.1. TREBALL FUTUR	44
9. BIBLIOGRAFIA	46

TAULA DE FIGURES

FIGURA 1 - DIAGRAMA DE GANTT DEL PROJECTE	15
FIGURA 2 - ESQUEMA DE COMUNICACIÓ RFCOMM.....	28
FIGURA 3 - TEXTOS QUE UTILITZA L'APLICACIÓ ANDROID	31
FIGURA 4 - PROCÉS D'ENVIAMENT DE FACTURA FINALITZAT CORRECTAMENT	34
FIGURA 5 - PROCÉS D'ENVIAMENT DE FACTURA FINALITZAT AMB ERRORS.....	35
FIGURA 6 - PANTALLA D'INICI DE L'APLICACIÓ	36
FIGURA 7 - PANTALLA DE CONFIRMACIÓ PER ACTIVAR EL SERVEI BLUETOOTH	36
FIGURA 8 - PANTALLA ON ELS MENÚS ES VEUEN ACTUALITZATS PER HAVER ACTIVAT EL SERVEI BLUETOOTH.....	37
FIGURA 9 - PANTALLA ON ES MOSTREN ELS DISPOSITIUS ACCESSIBLES EN LA FASE DE DESCOBRIMENT DE DISPOSITIUS.....	37
FIGURA 10 - PANTALLA DE VISUALITZACIÓ DE RESULTATS UN COP ACABAT EL PROCÉS DE DESCOBRIMENT DE DISPOSITIUS	38
FIGURA 11 - DIÀLEG DE CONFIRMACIÓ DE CONNEXIÓ.....	38
FIGURA 12 - CONNEXIÓ EXECUTADA CORRECTAMENT I FACTURA ENVIADA AL TELÈFON	39
FIGURA 13 - TRES CAPTURES QUE MOSTREN ELS PASSOS PER A L'AUTENTICACIÓ NECESSÀRIA PER ACCEDIR AL PORTAL	40
FIGURA 14 - TRES CAPTURES QUE MOSTREN EL PROCÉS QUE PERMET CARREGAR UNA FACTURA A L'APLICACIÓ WEB.....	41
FIGURA 15 - PANTALLA DE RESULTAT DESPRÉS DE PENJAR UNA FACTURA CORRECTAMENT (VEURE MISSATGE).....	41
FIGURA 16 - CORREU D'INFORMACIÓ CONFORME S'HA CREAT UNA FACTURA.	42
FIGURA 17 - CAPTURA ON ES POT LLEGIR EL CONTINGUT DEL MISSATGE QUE S'ENVIA AL PROPIETARI DE LA FACTURA.....	42
FIGURA 18 - DUES CAPTURES PER MOSTRAR QUE LA PRIMERA CONTÉ EL LLISTAT INICIAL I LA SEGONA EL LLISTAT ACTUALITZAT AMB LA NOVA FACTURA CARREGADA	43

1. INTRODUCCIÓ I OBJECTIUS

1.1. INTRODUCCIÓ

Les tecnologies de la informació i de les comunicacions han suposat un gran canvi a la nostra societat. Cada cop podem accedir a més informació i serveis amb independència del moment i lloc des d'on ho fem. La venda d'entrades a esdeveniments esportius o espectacles d'entreteniment, la compra a supermercats online, etc., en són alguns exemples. En aquests casos obtenim un tiquet o rebut en paper quan fem la compra de forma tradicional, és a dir, presencialment. El tiquet o factura que obtindrem serà en format digital. En direm una e-factura o e-tiquet per indicar que està en format electrònic (digital). Malgrat el format les e-factures (e-tiquets) han de tenir les mateixes propietats dels que estan en paper. Els usuaris han d'estar segurs que són vàlids i que els podran lliurar per demostrar una compra. Potser volem fer un canvi en una compra que hem fet, o si s'espatlla el que hem comprat necessitarem reclamar la garantia. En aquests casos ens podem demanar el rebut de compra. Per proporcionar aquestes propietats a les e-factures el Ministeri d'Indústria, Energia i Turisme del Govern d'Espanya va fer el Reial Decret 1619/2012, de 30 de novembre. Aquest decret aprova el reglament mitjançant el qual es regulen les obligacions de facturació. Estableix les normes que han de complir obligatòriament les factures, tant en paper com electròniques. A més a més també va desenvolupar una aplicació i una llibreria per tal de facilitar la implantació de les factures electròniques. Aquesta iniciativa va rebre el nom de Facturae (<http://www.facturae.gob.es>).

Justament amb la e-factura podem destacar la implantació dels telèfons intel·ligents i la possibilitat de pagar amb aquests. Si fem el pagament amb un telèfon intel·ligent, perquè no podem rebre el tiquet o factura en format electrònic? Aprofitant les propietats del facturae podrem estalviar els costos del paper i oferir funcionalitats avançades als usuaris. El format electrònic facilita la gestió de les dades de manera que els usuaris tenen una forma més fàcil de controlar les seves despeses. Ara bé, necessiten d'una infraestructura per gestionar les e-factures. Han de poder validar els rebuts, enviar-los, i per exemple que no es quedin al mòbil vell quan se'l canvien.

1.2. MOTIVACIONS

La principal motivació que ha fet possible aquest projecte és degut principalment a l'adopció que ha tingut els nous telèfons mòbils amb accés a Internet des de qualsevol punt en la societat actual.

Per altra banda, no seria possible aquest projecte sense que s'impulsés evolucionar la facturació en paper per l'electrònica, tal com està fent el Ministeri d'Indústria, Energia i Turisme del Govern d'Espanya, adaptant la legislació i donant accés als recursos necessaris per tal de poder adoptar tal canvi.

1.3. OBJECTIUS

L'objectiu principal d'aquest projecte és desenvolupar un sistema que permeti carregar les factures electròniques al núvol mitjançant un telèfon intel·ligent. Al núvol l'usuari podrà gestionar-les de forma més còmoda. Tot el procés de càrrega de les factures s'ha de fer de forma segura.

La realització d'aquest projecte es pot desglossar en un conjunt d'etapes. En primer lloc, les prèvies a qualsevol implementació, que es refereixen a un estudi previ compost per:

- Estudi conceptual del format de factura electrònica, el *Facturae* i de la seva implementació (metadades i programari existent per gestionar les factures electròniques)
- Estudi del protocol de comunicacions Bluetooth i de les llibreries d'on s'ha fet la implementació amb Java
- Estudi de l'estructura d'una aplicació Android: estructura de fitxers i carpetes, fitxers de configuració, cicle de vida i els seus estats, etc.
- Estudi de l'estructura d'una aplicació desenvolupada en el Google App Engine: estructura de fitxers i carpetes, fitxers de configuració, etc.

Les etapes a dur a terme un cop completades les inicials seran les d'implementació, que són:

- Desenvolupar una aplicació local que permeti enviar una factura mitjançant Bluetooth al telèfon mòbil (Android)
- Desenvolupar una aplicació Android que permeti carregar la factura al núvol mitjançant el protocol HTTPS
- Desenvolupar una aplicació web que rebi, proporcioni i gestioni les factures electròniques
- Realitzar proves per tal de validar el desenvolupament creat

Aquesta implementació, sobretot la part web, s'ha d'efectuar tenint en ment que el codi ha de quedar com menys acoblat millor, pensant que es poden ampliar les funcionalitats i el sistema ha de ser àgil als canvis.

1.4. ORGANITZACIÓ DE LA MEMÒRIA

A la secció 1 es descriuen els objectius, motivacions i en general tota la informació que pot interessar al lector per conèixer la raó d'aquest projecte.

La secció 2 conté la metodologia pel desenvolupament del projecte.

A la secció 3, s'hi troba la planificació per fases i dates que s'ha seguit per portar el projecte al dia.

A la secció 4, l'usuari trobarà el marc de treball, amb unes breus explicacions sobre els temes més significatius d'aquest projecte.

A partir de la secció 5, 6 i 7 es descriu la implementació del projecte, començant pel disseny i arquitectura, el desenvolupament i finalment els jocs de proves.

Finalment, a la secció 8 analitzarem els resultats obtinguts i el treball futur a realitzar.

2. METODOLOGIA

Per a la realització d'aquest projecte, m'he basat en la metodologia del desenvolupament en cascada. Aquest tipus de desenvolupament de software es caracteritza principalment per dividir les etapes del procés de desenvolupament de software, mantenint un ordre rigorós de tal manera que no es comenci a tractar una etapa sense haver finalitzat l'anterior.

Les etapes de desenvolupament que caracteritzen aquest tipus de desenvolupament de software, en la versió final són les següents (ordenades per ordre d'execució, essent la número 1 la primera a completar i la 7 la darrera):

1. *Anàlisi de requeriments.* Aquesta etapa és la que defineix què ha de complir el sistema, a nivell de funcionalitat. S'especifica un document que sol estar format per una descripció dels requeriments funcionals i no funcionals de l'aplicació.
2. *Disseny del sistema.* Un cop s'ha establert què ha de complir el projecte, es procedeix al disseny del sistema. El primer disseny a realitzar és d'alt nivell, especificant l'estructura de la solució a implementar. En aquest disseny, es realitza una descomposició de tasques generals en tasques més concretes, i a la relació que mantenen aquestes per tal de satisfer l'objectiu determinat. Aquesta descomposició de la feina en tasques facilita el treball en equip, essent possible establir tasques concretes independents a un grup de persones. Perquè tingui èxit, és important senyalar com ha de relacionar-se cada tasca concreta amb les altres, és a dir, establir les relacions o dependències de cada una de les tasques.

Un cop establert el primer disseny, es procedeix a realitzar un segon disseny de més detall. En aquest, s'especifica els algorismes que s'utilitzaran, així com la organització del codi, de cares a facilitar l'etapa de la implementació.

3. *Disseny del programa.* Aquesta etapa és la que realitza els algorismes necessaris pel compliment dels requeriments d'usuari. A més a més, també s'analitza les eines que s'utilitzaran en l'etapa d'implementació.
4. *Codificació.* Etapa en la que es comença a construir la solució, es comença a codificar els algorismes escollits, respectant l'estructura dissenyada en les etapes anteriors. També s'introdueix el concepte d'assaig-error, per testejar la solució de les diferents tasques abans de posar-les en comú.
5. *Proves.* A mesura que s'ha anat desenvolupament correctament una tasca, s'ha anat mesclant amb les altres tasques que finalment solucionen els requeriments determinats. Aquesta etapa doncs, serveix per comprovar que s'ha cohesionat correctament les diferents tasques, que totes les funcionalitats estan implementades i a més a més que tots els requeriments es troben completats. És la etapa final abans de presentar la solució al client.
6. *Implantació.* Amb la solució implementada, es procedeix a instal·lar la solució al sistema del client. Aquesta etapa pot fer referència a una fase d'aprenentatge per part del client, per tant, pot incloure la creació de manuals d'usuari per tal que l'adaptació sigui més eficaç.
7. *Manteniment.* Finalment, la última etapa del cicle de vida d'un projecte desenvolupat en model cascada és l'actualització o manteniment del sistema. En aquesta etapa es poden resoldre des de petits problemes o suggerències que hagi detectat el client com a usuari final, a canvis importants que poden requerir una re-definició del projecte. És la part més feixuga, on es calcula que un 75% dels recursos es destinen a aquesta etapa.

Un cop vistes les diferents etapes, s'entén a què es refereix el nom de “cascada”: en una cascada, és més fàcil seguir el corrent que anar contra ella. És la metàfora utilitzada per la definició d'aquest model, es vol donar a entendre que és més complicat passar d'una etapa inferior a una superior, respectant l'ordre especificat. Si bé no estableix que sigui impossible, sí que dóna la idea que requereix un gran esforç de dedicació (i moltes vegades també econòmic).

El fet que sigui un model àmpliament utilitzat no l'exempta de fortes crítiques pels desenvolupadors de software. Els defensors del *Agile software development*¹ critiquen que aquesta idea és impossible a la pràctica degut principalment al següent argument:

- Creuen impossible, per a un projecte no trivial, finalitzar una fase del cicle de desenvolupament del software a la perfecció abans d'avançar cap a la següent fase i aprendre d'elles, fent-les servir de referència.

Argumenten que el client pot no saber exactament els requeriments sense abans haver comprovat i revisat una primera solució. Poden realitzar canvis contínuament dels requeriments. Els dissenyadors i programadors poden preveure una petita part d'aquests canvis. D'altra banda, si el client canvia els requeriments després de que el disseny es trobi finalitzat, aquest s'ha de tornar a desenvolupar per adaptar-se als nous requeriments. Això comporta invalidar una quantitat d'hores ja fetes, o sigui que s'incrementa el cost del projecte, a nivell temporal i econòmic (especialment si s'ha destinat una gran quantitat de recursos en l'anàlisi del disseny per tal d'evitar aquests problemes).

¹ Agile software development és un grup de mètodes de desenvolupament de software que es basen en el desenvolupament incremental i interactiu, on els requeriments i les solucions evolucionen degut a la col·laboració entre la pròpia organització, l'equip multi-funcional que ha anat solucionant diferents tasques del projecte. Promou la planificació adaptativa, el desenvolupament i servei evolutiu, un període cíclic de reunions i una resposta ràpida i flexible als canvis. És un marc conceptual que promou interaccions previstes durant el cicle de desenvolupament.

D'altra banda, existeixen una sèrie d'avantatges que són els que ha permès que aquest model de desenvolupament sigui altament utilitzat:

- Els errors o detalls que es tenen en compte en les fases d'especificació dels requeriments o l'anàlisi del disseny estalvien un cost monetari i temps en solucionar-ho que si aquest error es troba un cop la solució ja ha estat implementada. Amb aquesta idea en ment, la responsabilitat del líder del projecte és que es dediqui tan temps com sigui necessari en les primeres fases del projecte, les que tracten de la confecció dels requeriments i de l'anàlisi de disseny, per tal d'estalviar temps i esforç posteriors.
- Permet gestionar l'equip de persones que formen part del projecte d'una manera eficient. Al ser un desenvolupament que requereix de molta documentació abans d'iniciar la implementació, en el cas que l'equip de desenvolupadors es vegi modificat, és ràpid fer entendre a les noves incorporacions les característiques del projecte.
- Al mantenir una estructura ordenada de la fase del projecte, s'ha batejat com un sistema "disciplinat". Això és gràcies a que el propi model ofereix una informació sobre l'estat del projecte depenent de la fase en la que es trobi, essent més senzill d'explicar al client el progrés que s'ha realitzat del seu projecte.
- Per acabar, el model és apte per a projectes de llarga durada, projectes anomenats "estables". Aquests projectes es caracteritzen per no modificar els requeriments inicials. Això permet que els dissenyadors siguin capaços de preveure totalment problemes en diferents àrees del sistema i produir un disseny correcte abans de que es comenci a processar l'etapa d'implementació. Per a això, els encarregats de l'etapa d'implementació han de seguir, d'una manera pautada, el disseny a la perfecció, assegurant que la integració del sistema evoluciona sense inconvenients.

3. PLANIFICACIÓ

Per desenvolupar aquest projecte s'establirà una temporització sobre objectius parcials que s'han d'aconseguir per tal de realitzar la implementació correcta del projecte global, procurant que aquesta es trobi marcada en funció de les etapes que la metodologia de programació en cascada així ho requeria.

En primer lloc s'explica la definició de cadascuna de les etapes i subetapes que s'hauran de seguir per a desenvolupar el projecte. Finalment s'adjunta el diagrama de temporització, de manera que es posa data als objectius parcials, de manera que en data d'entrega el projecte, la implementació compleixi els objectius globals.

- 1a etapa. Etapa d'introducció, es tracta de familiaritzar-se amb l'entorn: des del funcionament i l'estructura d'una aplicació Android fins a conèixer l'estructura i els recursos oferts per a la gestió de les factures electròniques amb format *facturae*. També forma part d'aquesta etapa l'estudi sobre els diferents protocols de comunicació implementats pel servei Bluetooth. No oblidar configurar tots els serveis necessaris: disposar d'un compte a Google, registrar una aplicació a la interfície del Google App Engine, instal·lar el programari de desenvolupador, actualitzar les llibreries pertinents, etc.
- 2a etapa. Anàlisi dels requisits, bé siguin funcionals com de seguretat. Dissenyar l'arquitectura del sistema necessària per complir l'objectiu principal.
- 3a etapa. Implementacions prèvies a la implementació final del projecte.
 - Implementació d'una aplicació de proves per fer un *Hello-World* en Android
 - Implementació d'scripts de prova per tal de comunicar el telèfon i l'ordinador per Bluetooth.
 - Implementació d'scripts en Java per tal de validar factures prèviament creades, per a testejar el component de validació que s'ha de desenvolupar en el telèfon.
- 4a etapa. Implementació del projecte.

- Implementació de l'aplicació que envia una factura des de l'ordinador al telèfon Android mitjançant Bluetooth
- Implementació de l'aplicació Android per rebre la factura de l'ordinador i carregar-la al núvol
- Implementació de l'aplicació web per tal de rebre les factures a carregar i presentar a l'usuari un portal per poder veure les que ja ha carregat

Aquest serà el *core* del projecte.

- 5a etapa. Millora del resultat. Identificar les millores que donarien més valor al sistema implementat i que en un principi no s'havien tingut en compte.
- 6a etapa. Validació de les millores. Testejar que realment les millores implementades serveixen pel que es van afegir.
- 7a etapa i última. Revisió formal. Demostració de la funcionalitat total del projecte.

Fins aquí l'explicació de les etapes. A continuació i per acabar de detallar la temporització del projecte, adjunto un diagrama de Gantt que s'ha planificat amb el temps calculat a dedicar a cada etapa:

4. MARC DE TREBALL I CONCEPTES PREVIS

En aquest apartat s'introdueixen tots els conceptes per situar el lector.

Per tant, s'introdueixen des dels conceptes com ara què és el servei Bluetooth, passant per les possibilitats que ens dóna la factura electrònica.

4.1. INTRODUCCIÓ AL BLUETOOTH

En aquest primer apartat s'introdueix a l'usuari tots els conceptes que involucren el Bluetooth: des de la història i els seus inicis fins a l'ús actual que es present en el marc tecnològic i com es veu cobert l'apartat de la seguretat.

Podria explicar molta més informació que crec que al lector no li serviria, entrant en detalls tècnics dels diferents protocols que implementen el Bluetooth així com les aplicacions que gestionen tots els serveis del port que utilitza el Bluetooth.

4.1.1 QUÈ ÉS EL BLUETOOTH?

És la norma que defineix un Standard global de comunicació sense fils, que possibilita la transmissió de veu i dades entre diferents equips mitjançant un enllaç per radiofreqüència. Els principals objectius que es pretén aconseguir amb aquesta norma són:

- Facilitar les comunicacions entre equips mòbils i fixes
- Eliminar cables i connectors entre aquests
- Oferir la possibilitat de crear petites xarxes sense fils i facilitar la sincronització de dades entre els nostres equips personals

La tecnologia Bluetooth comprèn maquinari, programari i requeriments d'interoperabilitat, de manera que per al seu desenvolupament ha estat necessària la participació dels principals fabricants dels sectors de les telecomunicacions i la

informàtica, com ara: Ericsson, Nokia, Toshiba, IBM, Intel i altres. Posteriorment s'han anat incorporant moltes més companyies, i es preveu que properament els facin també empreses de sectors tan variats com: automatització industrial, maquinària, oci i entreteniment, fabricants de joguines, electrodomèstics, etc., de manera que en poc temps se'ns presentarà un panorama de total connectivitat dels nostres aparells tant a casa com a la feina.

4.1.2. BREU HISTÒRIA DEL BLUETOOTH

En el 1994 Ericsson va iniciar un estudi per investigar la viabilitat d'una interfície via ràdio, de baix cost i baix consum, per a la interconnexió entre telèfons mòbils i altres accessoris amb la intenció d'eliminar cables entre aparells. L'estudi partia d'un llarg projecte que investigava sobre uns multi-comunicadors connectats a una xarxa cel·lular, fins que es va arribar a un enllaç de ràdio de curt abast, anomenat MC link. D'acord aquest projecte avançava es va anar veient clar que aquest tipus d'enllaç podia ser utilitzat àmpliament en un gran nombre d'aplicacions, ja que tenia com a principal virtut el que es basava en un xip de ràdio relativament econòmic.

Al començament de 1997, segons avançava el projecte MC link, Ericsson va ser despertant l'interès d'altres fabricants d'equips portàtils. De seguida es va veure clarament que perquè el sistema tingués èxit, un gran nombre d'equips haurien d'estar equipats amb aquesta tecnologia. Això va ser el que va originar a principis de 1998, la creació d'un grup d'interès especial (SIG), format per 5 promotors que van ser: Ericsson, Nokia, IBM, Toshiba i Intel. La idea era aconseguir un conjunt adequat d'àrees de negoci, dos líders del mercat de les telecomunicacions, dos líders del mercat dels PCS portàtils i un líder de la fabricació de xips. El propòsit principal del consorci va ser i és, el d'establir un estàndard per a la interfície aèria juntament amb el seu programari de control, amb la finalitat d'assegurar la interoperabilitat dels equips entre els diversos fabricants.

4.1.3. SEGURETAT EN LES COMUNICACIONS

Per assegurar la protecció de la informació s'ha definit un nivell bàsic d'enciptació, que s'ha inclòs en el disseny del clip de ràdio per proveir de seguretat en equips que no tinguin capacitat de processament, les principals mesures de seguretat són:

- Una rutina de pregunta-resposta, per autenticació.
- Un corrent xifrada de dades, per enciptació
- Generació d'una clau de sessió (que pot ser canviada durant la connexió)

Tres entitats són utilitzades en els algorismes de seguretat: la direcció de la unitat Bluetooth, que és una entitat pública; una clau d'usuari privada, com una entitat secreta; i un nombre aleatori, que és diferent per cada nova transacció.

La direcció Bluetooth es pot obtenir a través d'un procediment de consulta. La clau privada es deriva durant la inicialització i no és revelada posteriorment. El nombre aleatori es genera en un procés pseudo-aleatori en cada unitat Bluetooth.

4.2. LA FACTURA ELECTRÒNICA

Una factura electrònica és, sobretot, una factura. És a dir, té els mateixos efectes legals que una factura en paper. Recordem que una factura és un justificant del lliurament de béns o la prestació de serveis.

Una factura electrònica és una factura que s'expedeix i rep en format electrònic.

És important destacar que l'expedició d'una factura electrònica està condicionada al consentiment del seu destinatari. La factura electrònica, per tant, és una alternativa legal a la factura tradicional en paper.

4.3. EL PATRÓ MVC

En aquest apartat s'explica els conceptes més generals del patró de disseny que he hagut de respectar per implementar el projecte.

4.3.1. QUÈ ÉS EL PATRÓ MVC?

El patró de disseny *Model Vista Controlador*, d'ara en davant MVC, és un patró de disseny de software utilitzat en l'actualitat per implementar aplicacions web. El patró identifica tres àrees que identifiquen l'aplicació web:

- *Model*: agrupa tot el que es refereix a la definició de la informació amb les que es treballarà a l'aplicació. És l'encarregat de definir els objectes, amb els seus atributs i mètodes que treballen els atributs. També s'utilitza per implementar els mètodes que obtenen resultats de la base de dades. En resum, és qui gestiona la **lògica** de l'aplicació.
- *Vista*: s'encarrega de mostrar d'una manera comprensible per a l'usuari final la informació referent al model. En els orígens del patró, s'utilitzava per aplicacions d'escriptori, sense estar destinades a la web: aquí es definia la **interfície gràfica** de l'aplicació.
- *Controlador*: és qui gestiona la informació que s'ha de mostrar i la manera amb la que s'ha de mostrar. Obté els models que vol, n'obté la informació que es requereixi, opera amb ella segons convingui i la passa a la vista de tal manera que es mostri el que l'usuari necessiti.

A continuació, en els següents apartats, exposo els avantatges i els inconvenients d'utilitzar aquest patró.

4.3.2. AVANTATGES

Els resultats que s'obtenen si s'implementa correctament aquest patró són els següents:

- No hi ha codi referent a la lògica de l'aplicació a la interfície d'usuari. És a dir, l'apartat de la vista únicament mostra informació. Si la manera d'obtenir aquesta, des del controlador, es modifica, no es veurà afectada.
- Permet testejar els diferents mòduls per unitats. Això facilita la feina del desenvolupador.
- Reutilització del codi: amb l'estructura i organització de fitxers i carpetes que comporta manté el codi de tal manera que facilita modificar-lo i adaptar-lo segons convingui.

4.3.3. INCONVENIENTS

Com tots els patrons de disseny, el MVC té una part que no resulta agradable pel desenvolupador: requereix més treball aconseguir la funcionalitat seguint el patró que sense seguir-lo. D'altra banda, convé indicar que tota la feina ben feta respectant el patró, és feina que estalviem de cares a un futur.

5. DISSENY I ARQUITECTURA

En aquesta secció es descriuen els requisits funcionals i tècnics. També es justifiquen les decisions de disseny preses. Com a resultat es presenta el disseny del sistema.

5.1. REQUISITS

Els requisits del projecte es divideixen entre els requisits funcionals i els de seguretat.

5.1.1. REQUISITS FUNCIONALS

Els requisits funcionals fan referència a la usabilitat de tota l'aplicació que gestiona les factures electròniques: això inclou sobretot l'aplicació del servidor web, però també la del telèfon i la local. L'usuari final ha d'aprovar el funcionament d'aquestes aplicacions, ja que és ell qui n'ha de fer ús i, per tant, el sistema, gestionant tots els recursos possibles sigui en l'entorn que sigui, li ha de posar com més fàcil possible fer qualsevol operació.

Aquesta aprovació per part de l'usuari se sol aconseguir si s'assoleix:

- Una interfície d'usuari clara i concisa. Avui dia les grans aplicacions solen tenir dissenys minimalistes, on l'usuari no se sent "atabalat". Aquest és un procés que ha evolucionat molt en el desenvolupament de software, sobretot de cares a usuaris finals (a nivell empresarial encara se sol donar més pes a la productivitat que a la aparença).
- Un temps de resposta ràpid. Per exemple, minimitzar els temps de càlcul de l'aplicació, ja sigui del telèfon com del servidor, i que siguin totalment transparents a l'usuari. Si fem que un menú desapareix per donar pas a un altre, procurar que aquesta animació sigui fluida si és que depèn d'un càlcul per fer la transició: no mostrar una animació que es queda clavada esperant el resultat d'un càlcul per continuar.

5.1.2. REQUISITS SEGURETAT

Tenint en compte les característiques que defineixen aquest tipus d'aplicació, s'ha d'oferir, per damunt de tot, el següent:

1. Confidencialitat de les factures. L'obtenció i emmagatzematge de les factures s'ha de fer preservant la confidencialitat d'aquestes, ja que contenen informació personal (dades del emissor i del receptor).
2. Les factures també han de mantenir-se íntegres, el seu contingut no s'ha poder modificar. La utilització del facturae permet assolir aquest requisit.
3. Les factures s'han de poder enviar, han de ser transferibles, bé sigui des de l'ordinador al telèfon mòbil, des del telèfon mòbil al núvol i també, si escau, entre telèfons mòbils
4. L'autoria de la factura s'ha de poder validar. Un altre cop el Facturae permet assolir aquest objectiu.

5.2. ESTUDIS I DECISIONS

Quan vam començar a planificar el projecte es varen seleccionar les tecnologies següents:

- *Google App Engine (GAE)* com a servei d'allotjament web. Aquest sistema el proporciona, gratuïtament fins a unes quotes d'ús, l'empresa Google. Com es comentarà més endavant, el llenguatge per excel·lència que ha de governar tot el projecte és el *Java*, així que de les variants d'implementació que ofereix el *GAE* es va escollir el *GAEJ - Google App Engine for Java*. Cal comentar que existeix en altres llenguatges àmpliament coneguts, com són el *Python* i recentment també el *PhP*.

Els principals avantatges per utilitzar aquesta tecnologia són els següents:

- Integritat de dades: el sistema proporciona un potent servei d'emmagatzematge de dades distribuït entre servidors virtuals, incloent un motor de cerca i de transaccions. El magatzem de dades utilitza un control de concurrència optimista, en el que destaca que, per sobre de tot, es garanteix la integritat de les dades.
- Integració del registre d'usuari i l'accés per usuaris concrets. D'aquesta manera ens evita haver de desenvolupar i implementar un sistema de comptes d'usuari.

- Serveis d'administració incorporats de gran utilitat, entre els quals destaca la possibilitat d'enviar correus utilitzant les eines de Google i el fet de disposar d'una memòria caché que es gestiona automàticament, oferint el que més importa de cares a l'usuari final: velocitat en les operacions.
- Java. El fet que es desenvolupi l'aplicació local, la del telèfon i la del núvol en el mateix llenguatge ajuda a reciclar components i fins i tot llibreries, quasi-sempre adaptades entre Android i Java pur, fet que facilita les tasques de desenvolupament i manteniment del software.
- Gratuït. Fins que no se sobrepassen uns límits que solen ser difícil de superar per aplicacions orientades a un públic no massa extens, tots aquests serveis són gratuïts. En qualsevol cas, si l'aplicació esdevé un èxit i el sistema comença a notar lentitud per mantenir les quotes, sempre es pot contractar més potència i ample de banda pagant en funció del trànsit.
- *Java*, com a llenguatge d'implementació global. L'aplicació local que envia la factura per Bluetooth al telèfon, el codi de l'aplicació Android que ha de rebre la factura i enviar-la al núvol i al servidor Web estaran implementats en aquest llenguatge. El primer avantatge és que utilitzar el mateix llenguatge en tot el desenvolupament agilitza molt la feina i fins i tot a reciclar llibreries que funcionen ja sigui per Android com per web. Si bé no totes les llibreries són fàcilment adaptables, les més conegudes (com les del servei Bluetooth) sí que tenen adaptacions fetes. El segon avantatge és que és un llenguatge àmpliament utilitzat, i això vol dir que ha de ser fàcil trobar llibreries per qualsevol funcionalitat que ens pugués passar pel cap d'implementar (per exemple, tractar fitxers XML, utilitzar el servei Bluetooth d'un telèfon i d'un ordinador, entre d'altres).

Per la banda del servidor web, en la plataforma de *GAEJ*, el Java també és el llenguatge en el que s'implementen els *servlets* encarregats de comunicar el telèfon amb el servidor, per tant aquest punt també es va tenir en compte.

- HTML5 i Javascript/jQuery. La interfície web s'havia d'utilitzar en telèfons i en ordinadors, i el llenguatge web que uneix tots els dispositius és l'HTML5. Gràcies a aquest llenguatge, es pot accedir a elements com ara el sistema de fitxers del dispositiu programant-lo únicament una vegada, però adaptable a telèfons mòbils (de qualsevol sistema operatiu) i també a ordinadors.
- *Eclipse*, com a entorn de desenvolupament (IDE). Un entorn potent que ja havia utilitzat en altres projectes de programació durant la universitat. Ofereix una integració total amb aplicacions desenvolupades ja sigui per Android com per Google App Engine.
- Google Android SDK per a OSX. Google posa a disposició dels desenvolupadors totes les llibreries i software necessaris, diferenciant, si escau, cada recurs per versió d'Android disponible al mercat.
- *Git*, com a sistema de control de versions. Aquest sistema manté una estructura de fitxers i directoris internament que permet controlar en tot moment la versió actual del projecte a nivell local i de servidor. Una eina indispensable si més d'un col·laborador vol portar un control sobre l'estat del projecte, així com canvis i possibilitat de restaurar versions anteriors a l'actual en cas que així es desitgi. A més, cada programador pot tenir les seves versions a nivell local que parteixen del mateix projecte, i en funció de les necessitats, poden acoblar-les a la versió remota per tal de que els altres programadors puguin actualitzar la versió del seu projecte.

Durant el desenvolupament, però s'han pres les dacions següents:

- Enviar la factura al mòbil i d'aquest enviar-la al núvol
- Enviar informació del compte per poder pujar una factura (similar a *one-time-password*). Aquest cas requeria una programació addicional, com ara generar els *tokens* de la operació, validar-los, és a dir, mantenir-los en general.

La opció que vam acotar, doncs, va ser la primera, ja que era més senzilla de desenvolupar i la gestió del compte es podia fer directament, tal com s'ha comentat, utilitzant la gestió d'accés que porta implícit el *Google App Engine*.

L'altra gran decisió d'aquesta aplicació era si faríem un desenvolupament en paral·lel de les funcionalitats en el telèfon i en el núvol. Què ens aporta un desenvolupament d'aquest tipus? Bàsicament, que cada operació estarà optimitzada per la plataforma en la que s'executa. Per exemple, les opcions de visibilitat d'un telèfon i la interfície són diferents que les d'un ordinador de sobretaula.

Fins aquí sembla que seria el més adient, però també suposa desenvolupar per diverses plataformes i també requereix un manteniment per cada operació, vam decidir que el telèfon seria només un visor del contingut que es mostra en el servidor web: d'aquesta manera, oferíem unes opcions a l'usuari menys intuïtives, però aconseguíem un estalvi de temps important per cada funcionalitat que implementéssim.

Vam decidir, doncs, que per anar bé tota la interfície gràfica l'hauria de fer el núvol i el mòbil només hauria de ser el visor, actuar com un navegador web.

5.3. ARQUITECTURA

L'arquitectura del projecte la podem descriure identificant tres grans elements, que són l'aplicació local, l'aplicació Android i l'aplicació Web.

S'ha pensat en aquesta arquitectura per tal de que el flux lògic del sistema (enviar factura des de l'ordinador al mòbil, carregar-la des del mòbil al núvol i descarregar-la al telèfon de nou des del núvol) es pugui assolir satisfactòriament.

L'aplicació local i la d'Android es comuniquen via Bluetooth, de manera que l'aplicació local actua com a servidor d'informació i és l'aplicació Android que la consumeix.

L'aplicació Android però també es connecta a l'aplicació web, per carregar-hi factures que es troben emmagatzemades en el dispositiu. D'altra banda, però, l'aplicació web també es comunica amb la d'Android, ja que permet descarregar les factures al telèfon. Per tant, en funció de l'operació, les aplicacions d'Android i del servidor poden actuar de servidors o consumidors d'informació.

El pas d'informació queda de la següent manera:

LOCAL ==> (Bluetooth) ==> ANDROID <== (https) ==> SERVIDOR WEB

5.4 DISSENY DEL SISTEMA

El disseny del sistema es desglossa en les parts següents:

- Aplicació local
- Aplicació Android
- Aplicació web del Google App Engine

5.4.1. APLICACIÓ LOCAL

L'aplicació local està pensada per funcionar sobre la plataforma UNIX, perquè l'aplicació està orientada a un públic avançat i perquè s'ha volgut tenir en ment els dispositius *empotrats* com ara la *raspberry pi*, que funcionen sobre aquesta plataforma, que cada vegada s'estan utilitzant més a nivell personal i a nivell empresarial, doncs de manera molt ràpida donen solucions a problemes molt concrets (en aquest cas, a la gestió de la facturació electrònica).

El funcionament de l'aplicació local és el següent:

1. La seva funció és esperar una connexió Bluetooth per tal d'enviar la factura que l'usuari n'ha especificat la ruta.
2. Un cop rep aquesta connexió, provar d'enviar la factura, i tan si ho aconsegueix com si no ho fa, acabar l'execució, informant del resultat a l'usuari.

Ha d'utilitzar un identificador per tal de que la connexió sigui entre l'aplicació desenvolupada i la pròpia aplicació local, ja que sinó, podria enviar el fitxer a qualsevol dispositiu que efectués una connexió Bluetooth amb l'ordinador.

5.4.2. APLICACIÓ ANDROID

Aquesta aplicació ha de permetre gestionar l'estat del servei Bluetooth del dispositiu (sempre que el terminal disposi d'aquest).

1. Buscar connexions amb l'ordinador sempre que aquest permeti les connexions d'entrada.
2. Un cop connectat amb l'ordinador que té la factura ha de rebre la factura i l'emmagatzema a la memòria interna del telèfon.
3. A partir d'aquí, s'habilita la funció per carregar la factura al núvol. S'ha de gestionar en funció de l'estat de la connexió amb el telèfon, és a dir, no habilitar el botó corresponent si no hi havia cap factura a carregar.

4. Un cop habilitat el botó, s'ha de deixar escollir a l'usuari la factura a carregar, i en el moment que l'ha seleccionat, enviar-la per HTTPS al servidor mitjançant una petició POST al *servlet* de recepció de factures.

5.4.3. APLICACIÓ WEB

L'aplicació web ha de servir com a receptor de factures.

A partir d'aquí, tota la funcionalitat s'executa en el propi servidor per a totes les operacions: validació, classificació, persistència, consulta,...

Per realitzar la connexió amb el telèfon s'ha de realitzar una connexió segura. Per això s'ha decidit d'utilitzar el protocol HTTPS, ja que un dels requisits funcionals és la integritat i anonimat de la informació.

6. DESENVOLUPAMENT

El desenvolupament s'ha estructurat en tres mòduls:

- Servidor;
- Aplicació Android; i
- Servei al núvol.

6.1. SERVIDOR DE FACTURES

El servidor utilitza el servei Bluetooth en Java i les llibreries *bluecove-2.1.1*

S'ha desenvolupat un "servidor" que té preestablert un identificador de servei Bluetooth amb el que es comunicarà amb l'aplicació.

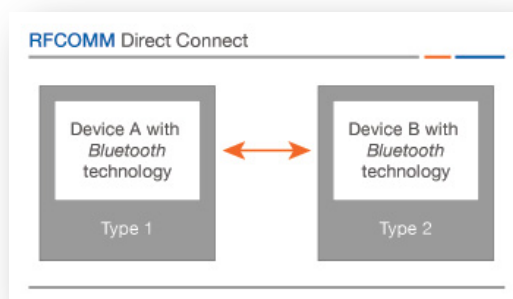


Figura 3 - Esquema de comunicació RFCOMM

A la Figura 3 el dispositiu de tipus 1 seria l'ordinador amb l'aplicació local que conté la ruta de la factura a carregar; el dispositiu de tipus 2 seria el terminal Android que executa l'aplicació per rebre la factura.

Un cop rep la connexió, el servidor envia el fitxer i l'aplicació el guarda en una ubicació de la memòria interna del dispositiu, essent un procés totalment transparent a l'usuari. Si durant qualsevol moment hi ha algun error en fer qualsevol operació per part del servidor Bluetooth, l'aplicació local s'atura i el telèfon informa que la connexió ha estat cancel·lada.

Com a resultat del desenvolupament s'ha desenvolupat un mòdul amb els elements següents:

- Llibreria *bluecove-2.1.1.jar* per poder utilitzar el Bluetooth de l'ordinador. Aquesta versió és la última disponible al moment de desenvolupar el projecte i la que assegura una màxima compatibilitat.
- Scripts per compilar i executar el programari.
- El programa espera una connexió Bluetooth per tal d'enviar la factura que l'usuari ha especificat la ruta (sigui correcta, incorrecta o bé incompleta). Un cop rep aquesta connexió, prova d'enviar la factura, i tan si ho aconsegueix com si no ho fa, acaba l'execució (informant del resultat)

Per executar aquesta aplicació, s'ha d'especificar la ruta de la factura a carregar. Per fer-ho, és tan fàcil com executar el programa que es facilita al client *executar*, de la següent manera: *./executar /Users/osxevi/Desktop/fac1.xml*. D'aquesta manera, estem indicant a l'aplicació local que es vol transferir la factura del meu escriptori que s'anomena *fac1.xml*.

6.2. APLICACIÓ ANDROID

El desenvolupament de l'aplicació Android contempla diversos paquets:

El primer paquet correspon a la funcionalitat de l'aplicació. Aquest, es troba, dins la carpeta del projecte, penjant de la branca */src/com.example.myfirstapp*:

- Classe **MainActivity.java**: aquesta classe és la que governa tota la funcionalitat de l'aplicació i recull tots els inputs de l'usuari i els externs (per exemple, el fet de desactivar el Bluetooth mentre l'aplicació es troba minimitzada).

A la classe, hi ha definits els atributs necessaris per construir l'aplicació, com ara els propis de la interfície (botons, llistes), els més funcionals, els que s'encarreguen de detectar els canvis d'estat i d'iniciar una connexió Bluetooth, fins els que s'encarreguen d'emparellar els *listeners* a diferents accions del telèfon (per exemple: quan s'ha acabat d'iniciar el servei Bluetooth, es mostra per pantalla; quan s'ha rebut la factura, també es mostra per pantalla).

Tota aquesta funcionalitat s'ha adaptat de tal manera que fos com més simple possible, i tot i així queda una presentació complexa.

- Classe **BluetoothDeviceArrayAdapter.java**: aquesta classe s'encarrega de definir una estructura de tal manera que l'aplicació pugui gestionar els dispositius que descobreix en la fase de descobriment de dispositius. A més, permet mapejar la informació per poder mostrar-la en una llista on s'hi mostra el nom i l'adreça mac del dispositiu en qüestió. És una classe de suport de l'anterior.
- Classe **BluetoothService.java**: aquesta classe s'encarrega de gestionar, a nivell de programació de *Threads*, les possibles tipologies de connexions Bluetooth diferents que es poden fer: com a client i com a servidor. A més, també inclou els mètodes necessaris per a establir la connexió inicial amb el dispositiu destí, independentment de si l'aplicació actuarà com a client (que és el nostre cas) o com a servidor (es va deixar implementat de cares a una funcionalitat d'enviar factures entre dispositius, fet que converteix un telèfon en client i l'altre en servidor).

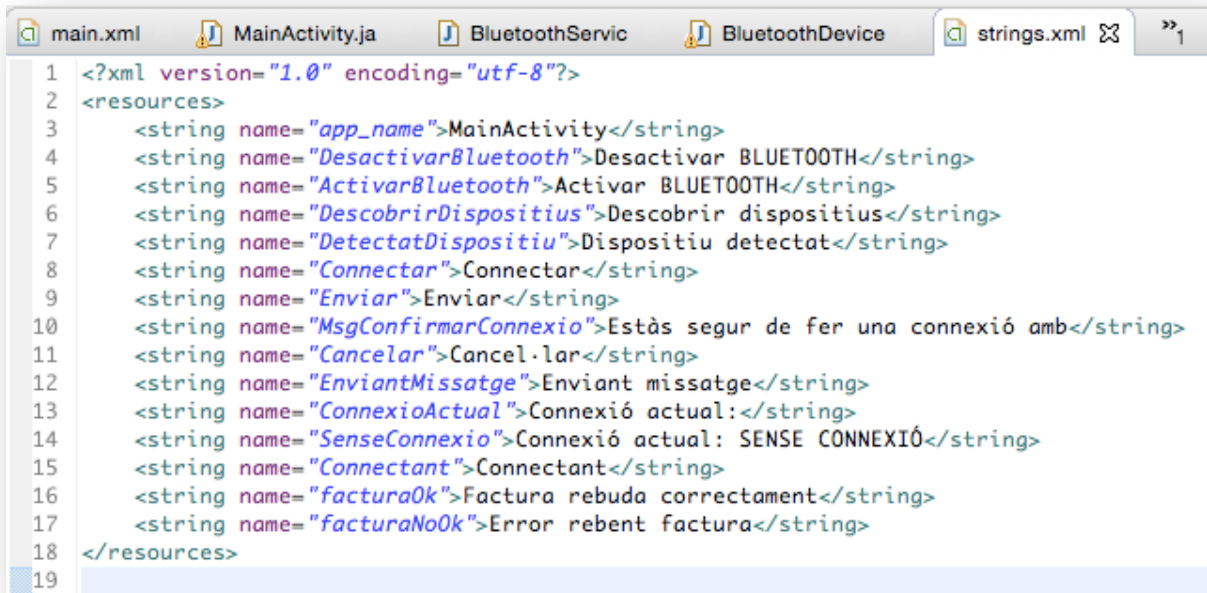
A continuació, el següent paquet és el del *layout*, que és qui defineix la interfície d'usuari de l'aplicació. En ella, s'hi vincula el feedback de l'usuari amb accions concretes a la classe *MainActivity* explicada en l'apartat anterior.

El layout es troba ubicada a **/res/layout** i consta de la següent classe:

- **main.xml**: de moment només hi ha un layout en tota l'aplicació, que s'adapta a mesura que s'activen o desactiven certs elements del telèfon. Però el més convenient seria tenir més d'una vista, així que tindríem més d'un fitxer en aquesta carpeta. Per ara, conté tots els elements que pot necessitar l'usuari, però els amaga o bé els modifica en funció de les operacions que efectui.

Els elements més importants a destacar són els botons que activen o desactiven el Bluetooth, el botó que inicia la cerca de dispositius, el llistat d'elements (que cada un d'ells és seleccionable) que vindria a ser els dispositius disponibles per fer-hi connexió i els missatges d'informació que es mostren quan es realitza una connexió a un dels elements d'aquest llistat.

El següent paquet és important per la opció multi-idioma, doncs permet definir claus associats a valors per tal de mostrar els missatges que es desitgin. D'aquesta manera, en lloc d'haver de programar a mà cada missatge que es mostri en funció de l'idioma actual, simplement es diu que es mostra una clau, i el fitxer de claus i missatges es manté en aquest: `\res\values\strings.xml`. El contingut el veiem a continuació:



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <string name="app_name">MainActivity</string>
4   <string name="DesactivarBluetooth">Desactivar BLUETOOTH</string>
5   <string name="ActivarBluetooth">Activar BLUETOOTH</string>
6   <string name="DescobrirDispositius">Descobrir dispositius</string>
7   <string name="DetectatDispositiu">Dispositiu detectat</string>
8   <string name="Connectar">Connectar</string>
9   <string name="Enviar">Enviar</string>
10  <string name="MsgConfirmarConnexio">Estàs segur de fer una connexió amb</string>
11  <string name="Cancelar">Cancel·lar</string>
12  <string name="EnviantMissatge">Enviant missatge</string>
13  <string name="ConnexioActual">Connexió actual:</string>
14  <string name="SenseConnexio">Connexió actual: SENSE CONNEXIÓ</string>
15  <string name="Connectant">Connectant</string>
16  <string name="facturaOk">Factura rebuda correctament</string>
17  <string name="facturaNoOk">Error rebent factura</string>
18 </resources>
19
```

Figura 4 - Textos que utilitza l'aplicació Android

6.3. SERVEI AL NÚVOL

El servei al núvol conté una estructura de classes per tal de gestionar les factures i varies operacions.

L'aplicació resideix en el següent lloc web, accessible per http i https (si s'intenta fer-ho per http es redirigeix a https):

<https://tfm-facturae-xavierquintana.appspot.com>

Per fer l'aplicació web s'han desenvolupat les següents classes, i que s'han agrupat per mòduls. Per començar, el mòdul del codi font, ubicat a `/src/com.tfm.facturapp`:

- **DBUtils.java**: conté els mètodes necessaris per gestionar les factures contra la base de dades. Per exemple: donar d'alta factures i emmagatzemar-les

- **DescarregarFactura.java:** és un servlet que permet descarregar una factura concreta. Es pot fer des de la pantalla de visualització de la factura:
- **Factura.java:** és la classe principal. Conté informació bàsica de la factura, que és el seu creador (usuari des del qual s'ha carregat, ha de tenir compte de Google), el document i l'estat. Per defecte, l'estat és "nova". Els possibles estats són:
 - Nova: per les factures donades d'alta
 - Acceptada: per les factures que estan validades
 - Corregir: per les factures que estan mal formades sintàcticament
- **Notificar.java:** aquest servlet envia un correu al propietari de la factura informant que la factura s'ha creat correctament, especificant la data i el nom del fitxer.
- **PenjarFactura.java:** aquest servlet dóna d'alta la factura, és a dir, recull el fitxer que l'usuari ha seleccionat i l'assigna a una nova factura donant-la d'alta a la base de dades. Un cop ha acabat de penjar-la, informa amb un missatge de text del resultat de l'operació a l'usuari.
- **PMF.java:** classe de gestió obligada per a la persistència de dades. En ella es defineix l'estratègia de persistència (en aquest cas, m'he decantat pel JDO). La classe invoca la factory per obtenir una única instància de la classe.
- **Validacio.java:** aquesta classe guarda tots els intents de validar una idea que es faci del sistema. D'aquesta manera es manté una informació que a vegades pot resultar útil, ja que s'hi inclou un camp d'observacions on s'avalua el resultat de la validació.
- **ValidarFactura.java:** servlet per realitzar la validació. S'hi hauria de parametritzar les característiques de la validació (sintàctica i/o de confiança, OCSP, etc).

Aquesta part es referia al codi java de l'aplicació, que seria on es defineixen les operacions. A continuació explico la interfície, i per acabar altres recursos interessants de comentar.

La interfície, és a dir, les pàgines que mostren informació i interactuen amb l'usuari, es troba dins el directori **/war**. Hi trobem el següent:

- **imatges:** carpeta on hi ha unes imatges que serveixen per mapejar l'estat de la factura

- **javascript:** dins aquesta carpeta hi trobem les llibreries de jQuery que utilitzen a validar dades al cantó del client. De moment no s'utilitza.
- **stylesheets:** fulles d'estil on es personalitza l'aparença dels elements html.
- **WEB-INF**
 - **appengine-web.xml:** fitxer de configuració que habilita o inhabilita serveis de l'app engine. No s'ha canviat la configuració per defecte.
 - **logging.properties:** fitxer per gestionar la sortida de logs de les operacions que s'executin en l'aplicació.
 - **web.xml:** fitxer de configuració on es mapeja la ruta d'un servlet amb el fitxer de codi font. S'utilitza per indicar que la ruta /penjarfactura fa referència al servlet PenjarFactura.java, per exemple.

En aquest fitxer s'hi afegeix informació important, sobretot les següents:

- És necessari iniciar sessió a un compte de Google per accedir a l'aplicació. Això s'assoleix amb la següent regla:
 - L'accés a la web ha de ser únicament per **https**. Utilitzem un certificat proporcionat de Google, tot i que l'app engine em deixaria utilitzar-ne un de propi. Això s'aconsegueix amb la següent regla:
- **index.jsp:** aquesta és la pantalla de benvinguda. En ella, s'hi presenta un llistat amb les factures que l'usuari ha carregat al núvol. La interfície permet a l'usuari refrescar el llistat, per exemple per si està donant d'alta factures a través de varis dispositius, i un botó per dirigir-se a la pàgina que li permetrà carregar una factura des del seu telèfon.
 - **afegirfactura.jsp:** aquesta pàgina serveix per penjar un document (factura) al servidor. Quan s'ha penjat el document, es carrega la pàgina de visualització de la factura.
 - **veurefactura.jsp:** aquesta pàgina serveix per visualitzar les metadades d'una factura, habilitant un botó per descarregar el fitxer (invoca el servlet de descàrrega de factures)

7. JOC DE PROVES

Per demostrar el funcionament complet del sistema efectuaré els passos que s'han de seguir perquè la factura que resideix en el meu ordinador acabi carregada en el servidor.

7.1 ENVIAMENT D'UNA FACTURA

Primer de tot es comença enviant la factura des de l'ordinador al telèfon. Es poden veure el cas que es carrega una factura correctament i un cas en el que no hi ha factura.



```
servidor-bluetooth — bash — 115x14
Xaviers-MacBook-Air:servidor-bluetooth osxevi$ ./executar /Users/osxevi/Desktop/Facturae_20141223/Borradores/1.xml
Establint el dispositiu com a visible...
BlueCove version 2.1.1-SNAPSHOT on mac
Iniciant el procés de descobriment...
Esperant una connexió entrant...
El Client s'ha connectat...
BlueCove stack shutdown completed
DONE.
Xaviers-MacBook-Air:servidor-bluetooth osxevi$
```

Figura 5 - Procés d'enviament de factura finalitzat correctament

En la captura, s'informa a l'usuari dels diferents processos que executa l'aplicació: s'activa el dispositiu perquè sigui visible, informa del procés i finalment notifica a l'usuari que està esperant una connexió. Quan, des de l'aplicació Android, s'efectua la connexió amb l'ordinador, automàticament informa que el Client s'hi ha connectat i se li envia el fitxer. Un cop fet tot això, finalitza l'execució.



```
servidor-bluetooth — bash — 115x14
Xaviers-MacBook-Air:servidor-bluetooth osxevi$ ./executar /Users/osxevi/Desktop/facturaInexistent.xml
Establint el dispositiu com a visible...
BlueCove version 2.1.1-SNAPSHOT on mac
Iniciant el procés de descobriment...
Esperant una connexió entrant...
El Client s'ha connectat...
Hi ha hagut un error: java.io.FileNotFoundException: /Users/osxevi/Desktop/facturaInexistent.xml (No such file or d
irectory)
BlueCove stack shutdown completed
DONE.
Xaviers-MacBook-Air:servidor-bluetooth osxevi$
```

Figura 6 - Procés d'enviament de factura finalitzat amb errors

En aquesta altra captura es mostra el resultat d'intentar enviar una factura inexistente. En aquest cas, els passos que s'informa són els mateixos que els explicats anteriorment, però en aquest cas es notifica a l'usuari amb l'error que ha impedit l'execució correcta del programa (en aquest cas, perquè no existeix el fitxer a enviar). Com s'ha comentat, el programa finalitza la seva execució normalment, tant pels casos correctes com pels incorrectes també.

7.2 APLICACIÓ ANDROID

L'aplicació Android ofereix un botó per activar el Bluetooth. Si l'usuari prem sobre el botó d'Activació, el telèfon sol·licitarà confirmació a l'usuari per activar el servei Bluetooth:

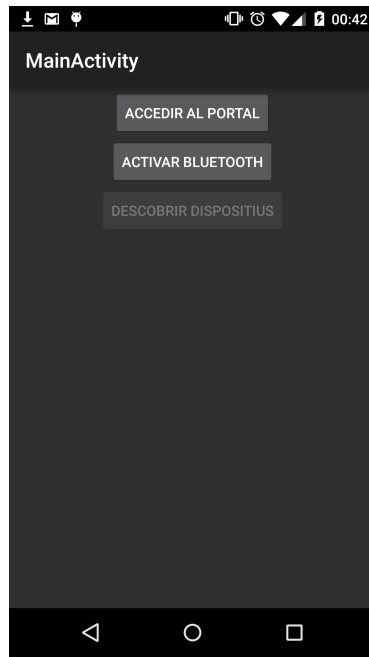


Figura 7 - Pantalla d'inici de l'aplicació

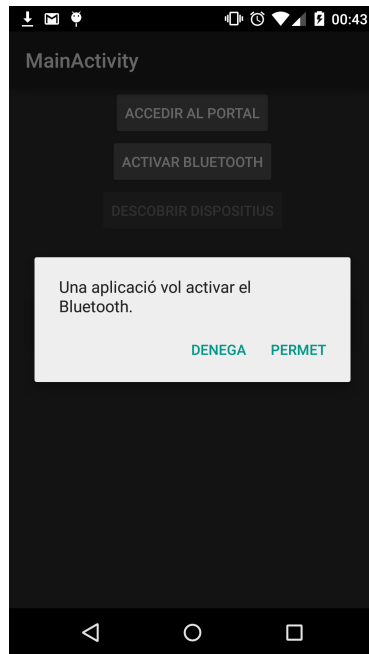


Figura 8 - Pantalla de confirmació per activar el servei Bluetooth

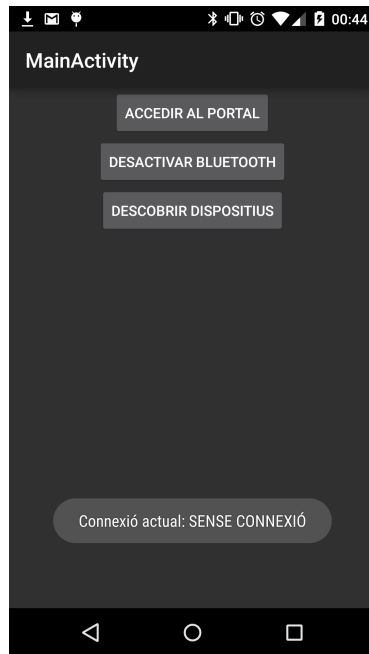


Figura 9 - Pantalla on els menús es veuen actualitzats per haver activat el servei Bluetooth

En aquest punt, ja podem iniciar la cerca de dispositius. Per a fer-ho, cal prémer el botó "DESCOBRIR DISPOSITIUS". Això detectarà el nostre ordinador, és a dir, el servidor Bluetooth.

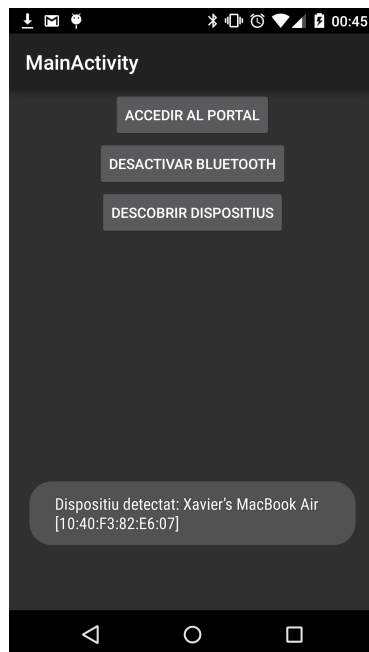


Figura 10 - Pantalla on es mostren els dispositius accessibles en la fase de descobriment de dispositius

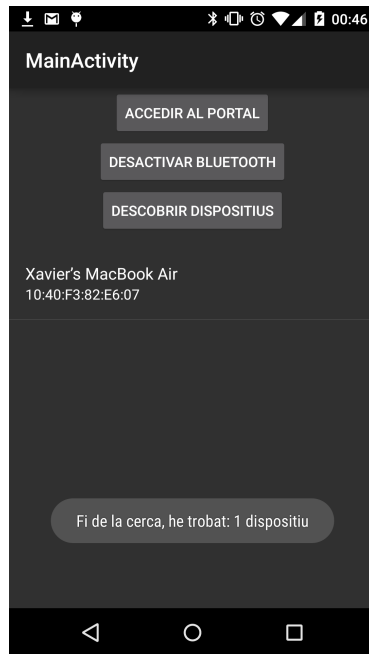


Figura 11 - Pantalla de visualització de resultats un cop acabat el procés de descobriment de dispositius

En aquest punt, l'usuari coneix el dispositiu amb el seu nom "amigable" i l'adreça MAC per identificar-lo de manera única i poder fer-hi la connexió. Si l'usuari prem sobre el dispositiu desitjat, se li presentarà un diàleg per confirmar si desitja efectuar la connexió amb el dispositiu indicat, i si accepta, l'aplicació local automàticament ho detectarà i enviarà la factura. Vegem-ho.

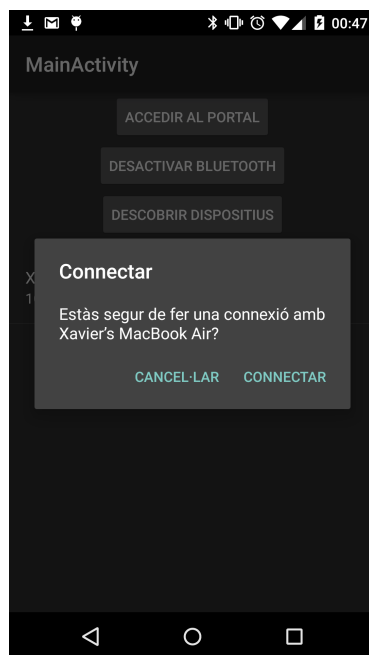


Figura 12 - Diàleg de confirmació de connexió

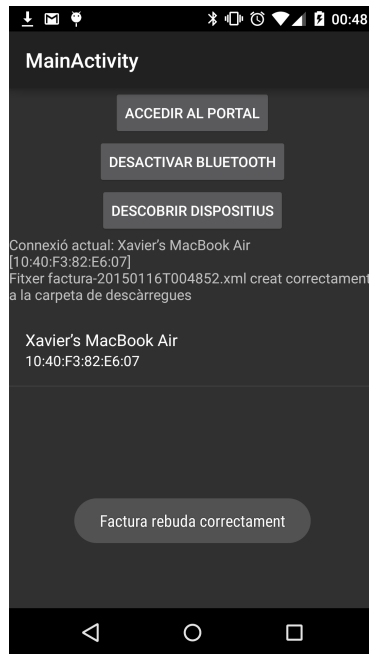


Figura 13 - Connexió executada correctament i factura enviada al telèfon

En aquest punt, la factura resideix en el telèfon, i s'emmagatzema amb el nom format per "facturaAnyMesDiaTHoraMinutSegon.xml". D'aquesta manera cada factura que hi haurà al telèfon serà única i no se sobreescriran dades.

7.3. SERVEI AL NÚVOL

Per tal de que l'usuari tingui un control de les factures que ha carregat al núvol, se li ofereix accedir al portal des de la pròpia aplicació. Per fer-ho només ha de clicar sobre el botó "ACCEDIR PORTAL". Si l'usuari no té una sessió iniciada amb algun servei de Google, se li demanarà autenticar-se. Quan ho ha fet, se'l redirigeix a l'aplicació web:

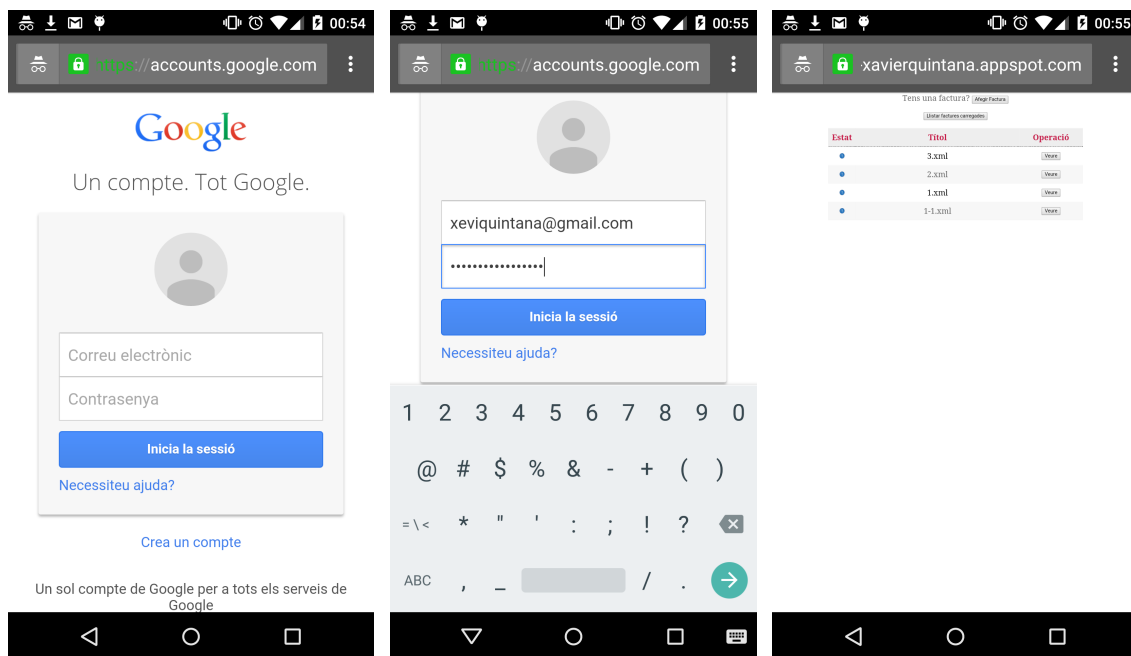


Figura 14 - Tres captures que mostren els passos per a l'autenticació necessària per accedir al portal

En aquest punt, l'usuari visualitza les seves factures. Les pot descarregar des de la pròpia pàgina de visualització de la factura tal com s'ha mostrat anteriorment.

Per seguir la demostració, carregarem la factura que ens hem enviat al telèfon: la factura *factura-20141226T004852.xml*.

Per fer-ho, el primer pas que hem de fer és prémer el botó "Afegir Factura" i a continuació prémer el botó "Tria un fitxer". Tot seguit busquem la factura a la carpeta de descàrregues del nostre dispositiu fins que trobem la factura desitjada, i seleccionem el botó de Guardar. A continuació ja es carregarà la factura des del nostre telèfon cap al servidor.

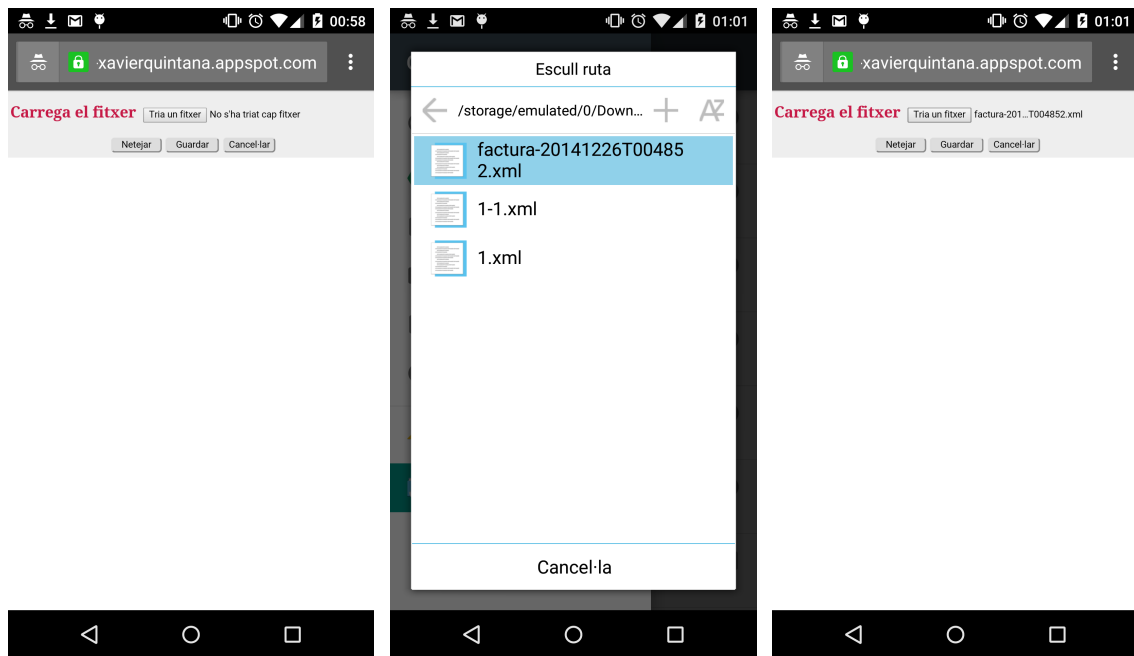


Figura 15 - Tres captures que mostren el procés que permet carregar una factura a l'aplicació web

A la pantalla següent podem veure com ja tenim la factura carregada al núvol. Podem descarregar la factura o bé tornar a la pàgina d'inici.

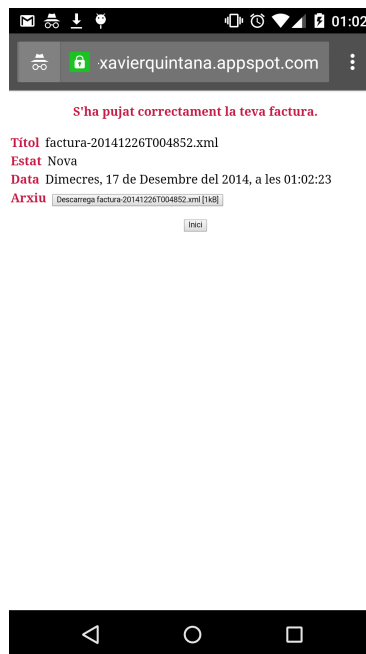


Figura 16 - Pantalla de resultat després de penjar una factura correctament (veure missatge)

Quan es carrega una factura s'envia un correu al seu propietari perquè tingui constància que la seva factura s'ha creat correctament en el servidor.

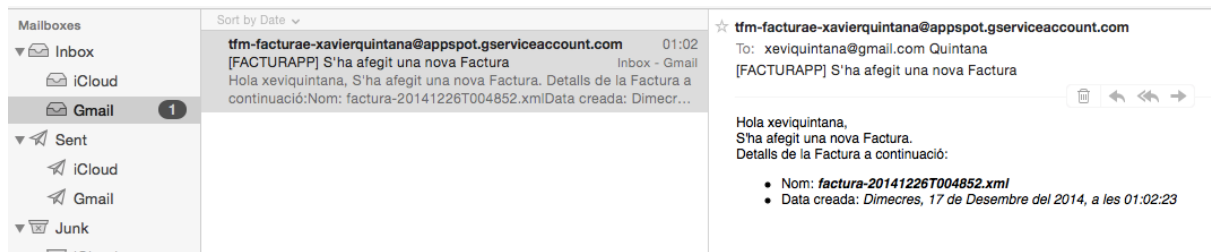


Figura 17 - Correu d'informació conforme s'ha creat una factura.

L'estructura del correu és molt bàsica, però mostra a l'usuari la informació bàsica i necessària per ésser informat de la nova factura carregada:



Figura 18 - Captura on es pot llegir el contingut del missatge que s'envia al propietari de la factura

El llistat de factures carregades per l'usuari es veu actualitzat al moment. Si ens dirigim a la pantalla d'inici a través del botó corresponent, ho podem comprovar: a l'esquerra hi ha el llistat inicial, i a la dreta l'actualitzat.

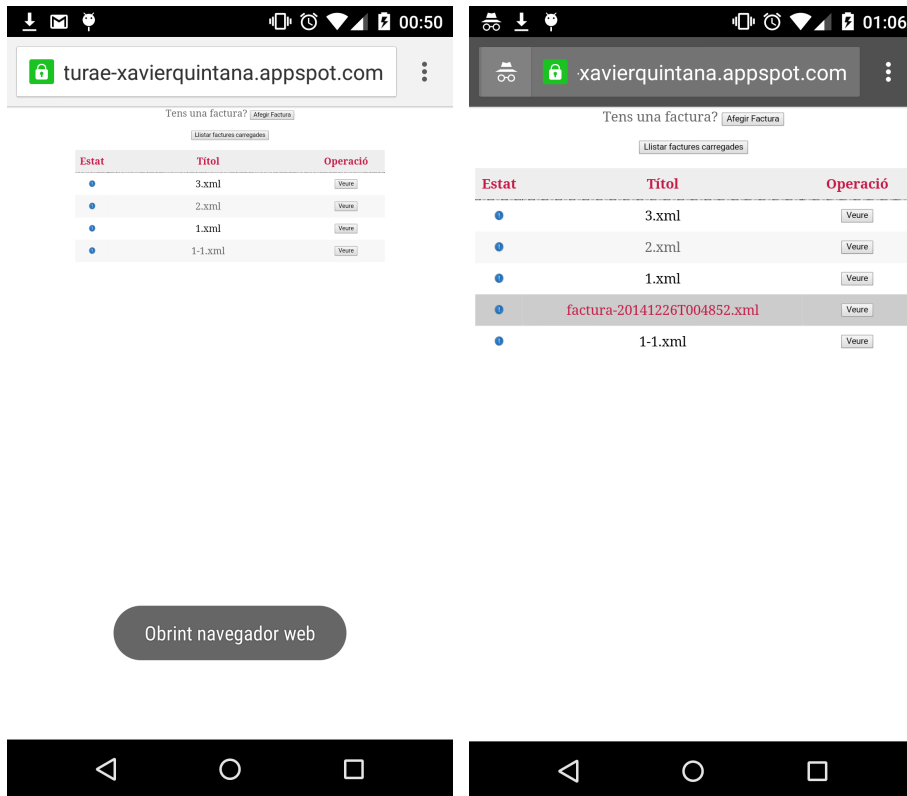


Figura 19 - Dues captures per mostrar que la primera conté el llistat inicial i la segona el llistat actualitzat amb la nova factura carregada

8. CONCLUSIONS

En aquest TFM s'ha desenvolupat un sistema que permet carregar les factures electròniques al núvol mitjançant un telèfon intel·ligent. Al núvol l'usuari havia de poder gestionar-les de forma més còmoda. Tot el procés de càrrega de les factures s'havia de fer de forma segura.

Tal com s'ha demostrat en el joc de proves, l'aplicació permet enviar una factura des d'un ordinador al dispositiu mòbil a través d'una comunicació per Bluetooth; permet també carregar aquesta factura de forma segura, que es troba emmagatzemada al dispositiu, al núvol; i permet fer la gestió que resulta més còmode per l'usuari de les factures carregades, com ara visualitzar les metadades o bé descarregar-les. En resum, tal i com es va fixar als objectius del TFM, la funcionalitat ha quedat totalment implementada.

A més, s'ha incorporat una funcionalitat que inicialment no es contemplava, que dona un valor afegit a l'aplicació, i és el fet de notificar per correu electrònic a l'usuari quan la factura s'ha carregat al núvol correctament.

Hi ha un treball futur a fer que es comenta en el següent capítol.

8.1. TREBALL FUTUR

El projecte es podria completar amb les funcionalitats següents:

- classificar les factures: si s'implementa aquesta funcionalitat, es podrà desenvolupar un mòdul per fer estadístiques contra les factures
- enviar factures a un altre dispositiu
- validar la signatura de les factures en el núvol
- permetre operacions amb factures mentre el dispositiu mòbil es trobi sense connexió, és a dir, permetre treballar offline (per exemple, ser capaços de gestionar una càrrega quan la connexió a Internet sigui disponible)

Cada operació d'aquestes requereix el seu mòdul de programació en el servidor web, i la seva interfície amb el telèfon Android. Si s'aconsegueix mantenir aquest tipus de desenvolupament, l'aplicació seguirà sempre oberta a noves funcionalitats que puguin sorgir al llarg del temps.

Una última operació interessant a fer seria adaptar-ho no només a comunicacions per Bluetooth sinó al nou BLE (*Bluetooth Low Energy*) i al NFC (*Near Field Communication*), uns altres sistemes de transmissió de dades que cada vegada s'està posant més de moda pel baix consum que requereixen en els terminals mòbils.

9. BIBLIOGRAFIA

Per construir la bibliografia, m'he basat amb la metodologia especificada segons la APA – *American Psychological Association*, concretament la 5a edició, adaptada al català. Aquesta metodologia s'ha utilitzat per citar els següents tipus de fonts:

- Llibres en paper
- Capítols o parts d'un llibre
- Pàgines web

Bibliografia pel que fa a llibres en paper:

- D.Sanderson (2012). *Programming Google App Engine* (2a ed.). Google Press.

Bibliografia pel que fa a capítols o parts d'un llibre:

- Pitt, Chris (2012). Introduction to MVC: What Is MVC?. Dins Pitt Chris (ed.), *Pro PHP MVC: Everything you need to know about using MVC with PHP in a single reference* (1a ed., p. 1-9). New York: Apress.

Finalment, bibliografia pel que fa a pàgines web utilitzades:

- GOBIERNO DE ESPAÑA. (2014). *Factura electrónica*. Recuperat entre 1 Octubre 2014 i 31 Desembre 2014, des de <http://www.facturae.gob.es>
- BlueCove. (2008). *BlueCove*. Recuperat entre 30 Octubre 2014 i 14 Desembre 2014 des de <http://www.bluecove.org>
- CERES. (2014). *Estado - FNMT*. Recuperat 29 Octubre de 2014, des de <http://www.cert.fnmt.es>
- Wikipedia. (2013). *Waterfall model*. Recuperat 9 de Gener 2015, des de http://en.wikipedia.org/wiki/Waterfall_model
- Google. (2014). *Java Runtime Environment - Java*. Recuperat entre 30 Octubre 2014 i 31 de Desembre de 2014 des de <https://cloud.google.com/appengine/docs/java/>

- Google. (2014). *Getting Started / Android Developers*. Recuperat entre 30 Octubre 2014 i 31 Desembre 2014 des de <https://developer.android.com/training/index.html>
- Bluetooth. (2014). *Bluetooth Development Portal*. Recuperat entre 15 Octubre 2014 i 31 Desembre 2014 des de <https://developer.bluetooth.org>

