



PLATAFORMA TELEMÀTICA DE SERVEIS ADMINISTRATIUS DE L'AJUNTAMENT DE RIPOLLET

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per:

Antonio BUIZA GRILLO

dirigit per:

Joan BORRELL VIADER (ETSE/UAB) i

Antonio QUESADA PUERTAS (Aj. Ripollet)

Bellaterra, 22 de juny de 2009



El sotasignat, **Joan Borrell Viader**

Professor de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en

Antonio Buiza Grillo

I per tal que consti firma la present.

Signat:

Bellaterra,de.....de 2009



El sotasignat, **Antonio Quesada Puertas**

Cap de recursos humans de l'Ajuntament de Ripollet,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en

Antonio Buiza Grillo

I per tal que consti firma la present.

Signat:

Ripollet,de.....de 2009

Índex

Capítol 1. Introducció.....	11
1.1 Presentació.....	11
1.2 Descripció del problema.....	11
1.3 Motivació.....	12
1.4 Objectius.....	14
1.5 Estructura de la memòria.....	16
Capítol 2. Anàlisi.....	17
2.1 Introducció.....	17
2.2 Estat de l'art.....	17
2.3 Divisió en mòduls.....	18
2.4 Mòdul 01 – Acreditació d'usuaris al sistema.....	19
2.4.1 Objectius específics.....	19
2.4.2 Requeriments.....	20
2.4.3 Punts forts / febles.....	21
2.4.4 Planificació.....	21
2.4.5 Casos d'ús.....	22
2.4.6 Programació temporal.....	22
2.5 Mòdul 02 – Registre d'entrada telemàtic.....	23
2.5.1 Objectius específics.....	24
2.5.2 Requeriments.....	25
2.5.3 Punts forts / febles.....	27
2.5.4 Planificació.....	27
2.5.5 Casos d'ús.....	28
2.5.6 Programació temporal.....	29
2.6 Mòdul 03 – Procediments informatius.....	30
2.6.1 Objectius específics.....	30
2.6.2 Requeriments.....	31
2.6.3 Punts forts / febles.....	32
2.6.4 Planificació.....	33
2.6.5 Casos d'ús.....	33
2.6.6 Programació temporal.....	34
2.7 Àmbit d'execució i recursos.....	35
2.7.1 Recursos humans.....	35
2.7.2 Recursos materials.....	36
Eines disponibles.....	36
Tria de les eines i justificació.....	38
2.8 Estudi de viabilitat.....	40
2.8.1 Viabilitat temporal.....	40
2.8.2 Viabilitat econòmica.....	40
2.8.3 Viabilitat legal.....	40
2.8.4 Viabilitat tècnica.....	41
2.8.5 Conclusions.....	42
Capítol 3. Disseny.....	43
3.1 Introducció i metodologia de disseny utilitzada.....	43
3.2 Disseny detallat dels mòduls analitzats.....	44
3.3 Mòdul 01 – Acreditació d'usuaris al sistema.....	45
3.3.1 Diagrames d'activitat i seqüència.....	47
3.3.2 Diagrama de classes.....	49
3.3.3 Entitat-relació.....	50

3.4 Mòdul 02 – Registre d'entrada telemàtic.....	50
3.4.1 Diagrames d'activitat i seqüència.....	52
3.4.2 Diagrama de classes.....	57
3.4.3 Entitat-relació.....	59
3.5 Mòdul 03 – Procediments informatius.....	60
3.5.1 Diagrames d'activitat i seqüència.....	61
3.5.2 Diagrama de classes.....	65
3.5.3 Entitat-relació.....	70
3.6 Comunicació entre els mòduls.....	76
Capítol 4. Desenvolupament del projecte.....	79
4.1 Introducció.....	79
4.2 Arquitectura i codificació.....	79
4.2.1 Definicions prèvies.....	80
4.2.2 Capes de l'aplicació.....	81
4.2.3 Model.....	82
4.2.4 Vista.....	83
4.2.5 Controlador.....	83
4.3 Generador de classes.....	85
4.4 Integració en portal web existent i comunicacions amb l'usuari.....	85
4.5 Integració amb les bases de dades.....	87
4.6 Publicació en servidor d'aplicacions existent.....	88
Capítol 5. Tests, validacions i manteniment.....	91
5.1 Conjunt de proves i resultats obtinguts.....	91
5.2 Punts de control.....	93
5.3 Informes d'activitat i productivitat.....	93
Capítol 6. Conclusió.....	95
6.1 Valoració de l'assoliment d'objectius.....	95
6.2 Vies de creixement i viabilitat.....	96
Capítol 7. Bibliografia i referències.....	99
7.1 Bibliografia.....	99
7.2 Enllaços.....	99
7.3 Programari.....	100
Capítol 8. Annexos.....	103
8.1 Conjunt de test realitzats.....	103
8.2 Pantalles de la implementació.....	110
8.3 Detalls tècnics d'implementació.....	120
8.3.1 Configuració IIS / .NET FrameWork.....	120
8.3.1.1 Habilitar canal SSL.....	120
8.3.1.2 Requerir certificats de client i configuració de la CTL.....	122
8.3.1.3 Registrar un component a la GAC.....	123
8.3.1.4 Referències a components en arxiu web.config.....	124
8.3.2 Validació de certificats amb CATCert/PSIS.....	125
8.3.2.1 Invocació senzilla del servei.....	125
8.3.2.2 Processament de la resposta.....	126
8.3.2.3 Invocació amb OptionalInputs.....	127
8.3.2.4 Processament de la resposta amb OptionalOutputs.....	129
8.3.3 Generació dinàmica de PDF amb webSupergoo.ABCPdf.....	130
8.3.3.1 Exemple bàsic "Hello world".....	131
8.3.3.2 Renderització HTML en PDF.....	131

8.3.3.3 Signatura electrònica amb certificat digital.....	132
8.3.4 Enviament d'arxius amb NeatUpload.....	133
8.3.4.1 Integració del component.....	133
8.3.4.2 Exemple d'enviament.....	134
8.3.4.3 Càlcul del HASH MD5.....	135

Capítol 1. Introducció

1.1 Presentació

El projecte que en aquesta memòria es documenta és una iniciativa de L'Ajuntament de Ripollet, recolzat pel departament de Serveis Informàtics, i que té com objectiu l'inici de gestions administratives entre el ciutadà i l'administració.

Es proposa crear una plataforma telemàtica de serveis integrada dins el portal web existent *ripollet.cat*. Ha de promoure la utilització dels canals electrònics segurs de comunicació aportant serveis d'acreditació, registre i consulta. Ha de maximitzar l'experiència d'usuari aportant els beneficis inherents d'un recurs d'aquest tipus com són els de l'alta disponibilitat horària (24x365), la immediatesa de gestió o l'estalvi dels desplaçaments. Recíprocament la implementació té la previsió de reduir les cues d'atenció al ciutadà i estalviar la transcripció de formularis manuscrits.

No és objectiu d'aquest projecte integrar serveis que requereixin col·laboració de tercers a la seva execució. Això comportaria una revisió i actualització dels models de procediments actuals i molt probablement una reorganització departamental. La integració d'aquests serveis complementaris als que es proposen s'hauria d'executar dins l'àmbit d'un nou projecte.

1.2 Descripció del problema

Ripollet és un municipi situat al Vallès Occidental, i forma part de l'àrea metropolitana de Barcelona. En aquests moments, al Padró Municipal, consten empadronats un total de 37.661 habitants.

El seu Ajuntament està organitzat en diferents departaments especialitzats segons la gestió que realitzen. Aquests desenvolupen el treball en oficines, algunes de les quals es troben en edificis territorialment separats. Cinc edificis municipals donen cabuda a la majoria dels serveis centrals de gestió. També existeixen Patronats que s'encarreguen de la gestió delegada de recursos específics del municipi.

Amb aquest perfil, s'entén que la gestió centralitzada dels serveis no és una tasca fàcil per aquest Ajuntament. Tant la informació, com els interessats i propietaris d'aquesta, han

de traslladar-se constantment d'una oficina a una altra amb els inconvenients que això en suposa.

Degut això i per centralitzar els serveis, a l'any 2003, es va crear l'Oficina d'Atenció al Ciutadà (OAC). La funció d'aquesta oficina és facilitar, mitjançant la centralització de serveis, la comunicació de l'Ajuntament amb els ciutadans. El seu objectiu, i segons el model de "finestreta única", és minimitzar les visites de terceres persones als departaments purament executors i centralitzar i maximitzar els serveis d'atenció, informació i assessorament.

Per posar en marxa aquest recurs, apart de la vessant humana, el departament de Serveis Informàtics va desenvolupar dues eines de software. Una d'elles és l'aplicatiu que permet la centralització de la informació de gestió: aporta el coneixement o la documentació necessària per l'execució dels serveis disponibles o sol·licitats pels interessats. Aquest aplicatiu ha evolucionat fins arribar al que és ara: el "Modelador d'expedients". La segona és el "Gestor d'expedients", eina de workflow col·laboracional que potencia el treball en equip. Permet la instanciació dels models, disposa d'eines que faciliten l'organització i la distribució del treball, recull la informació electrònica generada en el procés i en permet l'accés distribuït tan aviat aquesta es publica.

Amb aquestes dues eines és possible fer front a les necessitats informatives que es generen en aquesta oficina durant l'atenció al públic. Aportar la informació necessària per a iniciar una determinada gestió o informar de l'estat d'execució d'una gestió ja iniciada, es converteixen en tasques viables i sense esperes. La gestió es torna àgil i transparent.

En aquest punt, el principal problema amb el qual ens trobem és que aquesta informació segueix tancada dins les barreres de l'administració. No es poden iniciar gestions i, les iniciades, no són consultable remotament pels interessats. Detall que els obliga a desplaçar-se cada vegada que necessiten documentació o informació rellevant.

El desenvolupament d'un projecte com el que aquí es presenta garanteix una descarrega de treball d'atenció al públic i una major satisfacció a la part interessada.

1.3 Motivació

Durant els 8 anys que porto desenvolupant el meu treball com a funcionari dins el departament de Serveis Informàtics, he estat assignat en molts dels projectes de software de gestió i de sistemes de la casa. Entre d'altres, he estat responsable tècnic directe del desenvolupament complet de projectes com: la web corporativa, el modelador d'expedients o el seguiment d'expedients.

Els meus principals criteris d'organització d'objectius sempre han estat la simplicitat dels processos, l'ergonomia i l'accessibilitat. Així doncs, el meu treball s'ha desenvolupat amb la intenció de fer productes amb un impacte directe que faciliti el treball de l'usuari.

Un Ajuntament és una empresa pública. Si el comparem amb una empresa privada, podríem dir que el ciutadà és el client; i que els nostres serveis són els productes a

vendre. Continuant amb el sil·logisme, la llei de mercat ens hauria de forçar a oferir cada vegada millors serveis garantint que l'oferta d'aquests en sigués sostenible.

Aquest departament sempre ha col·laborat en la motivació del desenvolupament de nous projectes en relació amb les TIC i la societat de la Informació i el Coneixement. Hem intentat fer veure a la directiva els canvis tecnològics que s'esdevenien i els avantatges de la seva aplicació.

L'any 2006 es va presentar, a la inauguració de la Festa Major, el nou portal web de ripollet.cat. Es van executar molts canvis fortament lligats amb l'estil i estructura, amb l'objectiu de maximitzar l'accessibilitat i l'experiència de l'usuari, però sobre tot es va publicitar un calendari d'execució de nous serveis.

En referència aquest calendari, es va anunciar l'inici de la tramitació electrònica per a finals d'aquell mateix any. Malauradament el projecte era molt ambiciós. La seva execució havia de ser distribuïda en fases i amb una integració transversal. Tots els departaments havien de col·laborar, i per mancances de recursos i organització dels mateixos departaments no es va poder començar.

Aquest, diguem-ne, fracàs, em motiva a proposar un nou projecte estretament lligat amb aquell altre abandonat de fa anys, però amb una redacció més precisa i dirigida a uns objectius clarament tècnics i funcionals. Objectius que permetin un calendari d'execució realista, sense terceres col·laboracions que puguin distorsionar la vessant organitzativa de la casa. Un projecte individual que doni com a resultat un producte funcional i finit. Idoni per a presentar-ho com a projecte final de carrera.

Una altra principal motivació ha estat la d'aplicar la transició *Web 2.0* al portal corporatiu de l'Ajuntament.

L'any 2005, Tim O'Reilly escrivia¹:

«Web 2.0 is the network as platform, spanning all connected devices; Web 2.0 applications are those that make the most of the intrinsic advantages of that platform: delivering software as a continually-updated service that gets better the more people use it, consuming and remixing data from multiple sources, including individual users, while providing their own data and services in a form that allows remixing by others, creating network effects through an "architecture of participation," and going beyond the page metaphor of Web 1.0 to deliver rich user experiences.»

Aquesta i altres lectures sobre aquest concepte m'han fet incloure, sempre que he tingut oportunitat, la següent reflexió:

«Web 1.0: la informació és el web; Web 2.0: les persones són el web.»

Ara que ja es comença a parlar de *Web 3.0*, la transició d'un web 1.0 a un web 2.0 condiona la continuïtat d'aquest. Aquesta etapa significa deixar de banda el portal informatiu per convertir-lo en un portal de serveis on les persones no només el «consulten» sinó que també l'«utilitzen».

1 TIM O'REILLY. *Web 2.0: Compact Definition?* 2005.
<<http://radar.oreilly.com/archives/2005/10/web-20-compact-definition.html>>
(27 de maig de 2009).

Finalment un altre punt de motivació que recolza l'execució del projecte és el compliment de la Llei 11/2007 d'Accés electrònic dels ciutadans als serveis públics a partir de l'1 de gener de 2010. No cal més que fixar-se en l'enunciat d'aquesta llei per fer-se una idea del que s'exigeix en el seu text. Estructurada en 4 títols, 9 capítols i 16 disposicions, enuncia molts detalls d'aplicació i els justifica de forma adient.

La plataforma telemàtica de serveis administratius que es proposa en aquest projecte no implementa la totalitat de serveis que ofereix un Ajuntament, ja que això es tècnicament inviable, però és un gran pas en el llarg camí que té com a destí una administració electrònica completa i de qualitat.

1.4 Objectius

El mateix nom del projecte, *Plataforma Telemàtica de Serveis Administratius*, ens dona una idea dels objectius que es volen assolir. Entren en joc dues peces importants: una *Plataforma Telemàtica* i els *Serveis Administratius*.

Una *Plataforma Telemàtica* és un sistema de procés on es combinen les disciplines de telecomunicació i informàtica. Un sistema d'aquest estil, processa la informació de manera automàtica gràcies a la seva vessant informàtica, i la rep i la retorna processada, fent ús de canals electrònics de comunicació gràcies a la seva vessant de les telecomunicacions.

Els *Serveis Administratius* són el ventall de treballs que ofereix una administració, en aquest cas l'Ajuntament de Ripollet, als seus ciutadans o interessats. Tots i cadascun dels procediments o recursos que una administració ofereix de manera pública.

Entenent la definició d'aquestes dues peces fonamentals podem dir que l'objectiu principal d'aquest projecte és poder accedir als serveis que aquest Ajuntament ofereix mitjançant canals telemàtics de comunicació.

Què es vol aconseguir?

Es vol incrementar el grau de satisfacció del ciutadà amb la gestió administrativa de l'Ajuntament utilitzant i potenciant els recursos que ens ofereixen les tecnologies de la Informació. També, es pretén reduir les cues de treball en relació a l'atenció i informació al ciutadà. Aconseguir veure al ciutadà com a col·laborador i no com a client.

De l'objectiu principal es deriven un conjunt d'objectius tècnics que es numeren i s'analitzen en detall al capítol 2; i que es justifica el seu assoliment al capítol 6 d'aquesta memòria. A continuació exposaré els objectius inicials sense entrar en detalls específics.

Degut a que el catàleg de serveis d'una administració és molt extens i molts dels procediments no es poden automatitzar sense preveure canvis organitzatius, s'ha hagut de retallar en número, els serveis disponibles en aquesta plataforma que ara es projecta.

La selecció dels serveis candidats a ser integrats en la plataforma, s'ha realitzat tenint en compte certes característiques:

- Integrables telemàticament sense afegir càrregues de treball ni canvis organitzatius.
- Àmpliament reclamats per la ciutadania.
- El número de serveis escollits no ha de comprometre les dates d'execució del projecte.

Segons aquestes pautes, tots els serveis de consulta són integrables, sempre i quan aquesta consulta no comporti la producció de noves dades no emmagatzemades ja al sistema. Per exemple: el servei de consulta del Padró Municipal d'Habitants (PMH) sí és integrable, però el servei de consulta d'una Plus Vàlua no; doncs aquesta s'ha de calcular supervisat pel funcionari delegat.

Altre servei que ha d'estar disponible, ja que la seva naturalesa de gestió ho indica, és el Registre d'Entrada. Aquest és un tràmit universal, sent el canal oficial de petició formal a l'administració. La seva integració a la plataforma garanteix el resultat d'una aplicació funcional d'èxit.

De forma independent als serveis integrats, i degut a que les dades que es mouran per aquests serveis són dades protegides de caràcter personal, la plataforma ha de poder reconèixer el sol·licitant d'una forma segura i amb garanties de la seva identitat. Aquest, ha de poder iniciar sessió acreditant-se amb un binomi usuari/contrasenya o bé amb el seu certificat digital. Si l'acreditació es fa mitjançant certificat digital, el procés de validació d'aquest ha d'incloure la consulta a les llistes de revocació de l'agència certificadora que el va generar.

La informació generada per la plataforma s'ha de poder descarregar en un format que permeti una signatura electrònica, per exemple PDF, de tal manera que les dades en referència continuïn tenint validesa real dins el medi electrònic però fora de la plataforma.

El canal telemàtic d'accés a la plataforma es farà segons un aplicatiu web, integrat completament en el portal web *ripollet.cat*. Aquest, ha de satisfer els requisits d'estil i estructura. L'entrada i sortida d'informació es farà mitjançant canals webs segurs i es farà servir la infraestructura servidora d'aplicacions existents a l'Ajuntament.

Fent una petita anàlisi conceptual del problema i tenint tots aquests requeriments mencionats anteriorment, els objectius del projecte es resumeixen en:

- Acreditació usuaris amb binomi usuari/contrasenya o certificació digital.
- Registre d'entrada telemàtic.
- Serveis informatius (consulta de PMH, Registres, Expedients, Pagaments, ...)
- Integració amb portal existent *ripollet.cat*.
- Limitació d'implementació sobre la plataforma servidora d'aplicacions de l'Ajuntament.

Per acabar, recordar que al capítol 2 es fa una anàlisi d'objectius i requeriments amb detall.

1.5 Estructura de la memòria

Aquesta memòria es troba estructurada en quatre parts.

La primera part fa una introducció on es relacionen els detalls descriptius del problema, es comenten els objectius inicials i les motivacions que defensen l'execució del projecte.

La segona part, comprèn els capítols 2, 3 i 4. És un document més tècnic que presenta l'anàlisi del problema en profunditat, el disseny i els detalls de desenvolupament. El capítol 2 –anàlisi–, inclou l'estudi de l'estat de l'art i el de viabilitat. Al capítol 3 –disseny– es fa referència a la metodologia emprada i l'estratègia funcional que resol els problemes plantejats. El quart capítol –desenvolupament– explica l'arquitectura de l'aplicatiu generat i una abstracció semàntica en capes del funcionament lògic del mateix.

A la tercera part es comprova l'assoliment dels objectius al capítol 6 –conclusió– segons les proves realitzades al capítol 5 –Tests, validacions i manteniment–. Es comenten estratègies de control i diferents vies de creixement.

La quarta i última part del document centra la bibliografia i annexos als capítols 7 i 8 respectivament. En aquest últim capítol s'inclou una bateria d'exemples de referència de les tecnologies emprades en el desenvolupament del projecte.

Capítol 2. Anàlisi

2.1 Introducció

En aquest capítol es fa l'anàlisi del projecte. Busquem el *què* del projecte. La recerca de requeriments s'ha realitzat seguint un model evolutiu d'enginyeria del software: el de prototipatge. Aquesta metodologia permet prendre com a origen un conjunt inicial d'objectius globals i, aplicant refinaments de forma iterativa, aconseguir un conjunt de requeriments específics fàcilment definibles segons el context d'aplicació.

Segons el seu origen grec, la paraula *anàlisi* dona significat a l'acció de descompondre en parts o components. Així doncs la primera iteració d'aquesta tècnica ens justifica la partició del objectius inicials en tres mòduls amb uns objectius comuns i d'altres específics que es detallen a la secció 2.3 i següents.

Una vegada completada la recerca evolutiva, s'obté un llistat individual d'objectius i requeriments per cadascun dels tres mòduls analitzats. Els llistats es troben codificats per a permetre una referència clara en seccions i capítols següents. Als capítols 5 i 6 s'estudien les proves, els resultats i es justifiquen l'assoliment dels mateixos, respectivament.

La conclusió de tot el procés d'anàlisi dona com a resultat que el projecte és viable – justificat a la secció 2.8–, que la data prevista de començament és el dia 8 de gener i la data prevista d'acabament el dia 15 de maig del mateix any.

2.2 Estat de l'art

Entenent el projecte dins el marc dels serveis telemàtics a l'administració pública, existeix un projecte anomenat *e-Tram*, desenvolupat pel Consorci AOC (Administració Oberta de Catalunya). Aquesta plataforma guarda, aparentment, certes similituds amb la proposada com a projecte, ja que permet l'acreditació de l'usuari a partir de certificats digitals i el registre/petició dels processos administratius publicats.

El gran punt feble d'aquest producte existent, és la impossibilitat del tractament automàtic de la informació. La implementació d'*e-Tram* dona com a resultat un llistat de serveis, on l'únic tractament que es fa, és la recollida de la sol·licitud i la publicació dels resultats. En

cap dels processos es redueix la càrrega de gestió administrativa sobre els responsables executors. És tracta d'una finestra telemàtica de l'administració, però no es pot deixar desatesa, requereix la col·laboració explícita del personal executor per satisfer totes i cadascuna de les peticions rebudes. L'única millora donada la seva implementació és la possibilitat d'estalviar els desplaçaments dels interessats, res més; no es redueix la càrrega de treball ni desbloqueja la gestió de les limitacions horàries.

També existeixen d'altres plataformes propietàries de cadascuna de les empreses proveïdores de software. Algunes d'aquestes plataformes sí integren l'automatització dels processos, però la seva implementació es inviàble doncs crea un lligam entre l'administració i la empresa proveïdora en un paquet global de serveis i, per desgràcia, l'estudi de mercat actual no ens revela la confiança suficient com per portar a terme una contractació exclusiva. Hi ha empreses amb molta experiència en sectors específics però cap d'elles sembla guanyar a nivell global. Aquest motiu valida l'existència de diferents proveïdors de software a una mateixa corporació, detall que fa incompatibles les plataformes propietàries les unes amb les altres; i no només les existents, sinó que és molt dubtable que a una empresa l'interessi ampliar els serveis de la seva plataforma amb interfícies d'accés a dades on els responsables d'aquestes són empreses de la competència. En el cas extrem i suposant que aquesta hipotètica situació es portés a terme, queda ben clar que això invalidaria els contractes de manteniment existents amb cadascun dels proveïdors.

Analitzant-ho en detall i tenint en compte que el projecte proposat tracta d'una primera fase que donarà com a resultat un producte propi, modular i escalable es justifica clarament la seva viabilitat.

2.3 Divisió en mòduls

La lectura del punt 1.4 –objectius– d'aquesta memòria ens posiciona dins la complexitat del projecte. Agrupant de forma conceptual els requeriments o objectius inicials comentats, sembla pràctic fer una divisió modular del projecte amb l'objectiu de fer possible una recerca de requeriments específics i una anàlisi més realista, dirigit i detallat per a cada un dels mòduls plantejats.

Aquest particionament, també ens farà més senzilla la planificació de tasques i la programació temporal de l'execució prevista del mateix. Una programació més precisa propicia uns punts de control més aproximats i una previsió de la data de lliurament més ajustada.

Degut a aquest motiu, he considerat lògic agrupar els requeriments en tres mòduls específics:

- Acreditació d'usuaris al sistema
- Registre d'entrada telemàtic
- i Procediments informatius

De forma independent al particionament proposat el projecte té com objectius comuns els codificats segons la taula següent:

CODI	Descripció
OBJ_00_01	Tots els serveis seran publicats sobre una plataforma web integrada amb el portal existent ripollet.cat.
OBJ_00_02	La generació HTML validarà els estàndards del W3C segons l'especificació DTD HTML 4.01 Transitional així com ho fa la resta del portal.
OBJ_00_03	Tret de la pantalla de benvinguda, totes les comunicacions es faran sobre un canal segur HTTPS/SSL per garantir la privacitat d'accés.
OBJ_00_04	Es portarà un control de temps inactiu per tal d'anotar esdeveniments de desconnexió per inactivitat i evitar que l'usuari rebi excepcions degudes al tancament de la sessió per part del servidor web.

A continuació s'estudia l'anàlisi al detall de cadascun d'ells. Es fa un llistat d'objectius específics, un llistat de requeriments, un estudi de punts forts / febles, un llistat de tasques, diagrames de casos d'ús i una previsió de la programació temporal.

2.4 Mòdul 01 – Acreditació d'usuaris al sistema

Es tracta d'una aplicació de gestió d'usuaris independent i comuna als mòduls executors de processos. Aquest mòdul es troba constituït pels objectius/requeriments relacionats amb l'autenticació de l'usuari contra la plataforma. És l'encarregat de validar o negar l'accés de l'usuari a les dades relacionades generades pels serveis integrats.

Fent referència novament a la secció 1.4 –objectius–, la informació que es generi per aquests serveis no serà pública i per aquest motiu és obligatori el requeriment de la identitat de l'usuari per donar garanties d'accés segur i exclusiu a les seves dades.

2.4.1 Objectius específics

Derivats de la classificació d'objectius genèrics i segons les iteracions en la recerca d'objectius específics, és responsabilitat directa d'aquest mòdul l'identificar a l'usuari que sol·licita els serveis oferts per la plataforma.

Inicialment s'havia projectat que l'inici de sessió fos possible per dos canals: mitjançant un binomi usuari-contrassenya o bé amb l'ús de certificació digital. L'anàlisi en detall de l'inici de sessió fent servir usuari-contrassenya fa inviable la seva execució doncs no és un procés d'inici fiable ni que garanteixi la identitat de l'usuari. A part, condiona a desenvolupar els serveis de distribució i manteniment de claus que entra en conflicte amb un dels objectius inicials: el de no afegir càrregues de treball en cap àrea.

És degut aquest motiu que d'ara en endavant només es farà referència a l'inici de sessió mitjançant certificat digital.

La taula següent mostra el llistat d'objectius específics i el seu codi:

CODI	Descripció
OBJ_01_01	La identificació de l'usuari és farà mitjançant un certificat digital d'usuari emès per IDCat o bé DNle.
OBJ_01_02	Un cop validada l'acreditació de l'usuari és crearà l'inici de sessió de tal manera que la resta de mòduls puguin saber què usuari es troba connectat.
OBJ_01_03	L'usuari rebrà un informe amb els detalls del procés de validació tant si aquesta és correcta com si hi ha incidències.
OBJ_01_04	En cas d'acreditació amb èxit s'iniciaran les comunicacions a través de canals segurs (https-ssl) i se'l permetrà iniciar la gestió obtenint una pantalla llistat dels serveis disponibles
OBJ_01_05	L'usuari ha de poder tancar la sessió des de cadascuna de les pantalles de l'aplicatiu.
OBJ_01_06	El sistema apuntarà els detalls d'inici i tancament de la sessió.

2.4.2 Requeriments

Partint del llistat d'objectius del subapartat anterior, i fent una anàlisi en detall de tots d'ells, farem la següent classificació de requeriments funcionals:

Per satisfer l'objectiu OBJ_01_01 necessitarem una pantalla de benvinguda accessible mitjançant protocol no segur (http). Aquesta pantalla donarà informació de la plataforma així com les possibilitats d'accés que es troben habilitades.

Per l'objectiu OBJ_01_02 es requereix que el servidor mantingui un recurs comú a la resta d'aplicacions de manera que la resta de mòduls puguin consultar independentment quin usuari s'ha acreditat i les dades d'aquest.

Referent a l'objectiu OBJ_01_03 es necessitarà una pantalla informativa amb el resultat del procés d'inici de sessió.

L'objectiu OBJ_01_04 requereix els recursos necessaris client-servidor com per iniciar un canal de comunicacions segures. Serà condició indispensable que el servidor disposi d'un certificat de dispositiu segur reconegut i que el client tingui confiança en aquest. Els canals de comunicacions del servidor amb el client han de permetre la creació d'un canal segur sobre https-ssl.

Per garantir l'objectiu OBJ_01_05 es requerirà que la resta de pantalles generades en diferents mòduls incloguin una capçalera amb els detalls de l'usuari acreditat i l'accés a l'acció de tancament de la sessió.

Finalment l'objectiu OBJ_01_06 requereix que el servidor disposi d'un sistema que apunti els esdeveniments d'inici i tancament a una base de dades específica.

Com a requeriments no funcionals s'exigeix que l'usuari disposi d'un certificat d'usuari correctament instal·lat a la seva màquina i el coneixement suficient com per entendre com resoldre problemes de confiança de certificats.

2.4.3 Punts forts / febles

Paral·lelament a l'anàlisi anterior, més tècnica i estructurada, fem l'abstracció de punts forts i febles d'aquest mòdul de la següent manera:

Punts forts:

- Mòdul independent dels altres. Única funcionalitat d'acreditació dels usuaris vers la plataforma de serveis.
- Accés validat per certificat digital d'usuari.
- Interfície web, sinònim de d'alta accessibilitat.

Punts febles:

- Dependència de tercers en la validació de certificats al haver de consultar les llistes de revocació.
- Poca confiança dels usuaris en la certificació digital degut a la poca cultura digital.

2.4.4 Planificació

Per portar a terme la realització dels objectius analitzats, s'ha realitzat una distribució de tasques per aconseguir una programació estimada en temps.

Al llistat hi ha tasques purament de recerca i tasques on la seva execució correspon a l'etapa de disseny o bé desenvolupament. A la secció 2.7 es comenta part dels resultats de recerca i no és fins el capítol 3 i 4 que s'amplien els detalls d'execució.

Segons la distribució d'objectius, les tasques planificades serien:

- Aconseguir informació sobre els processos de validació de certificats segons les plataformes escollides (IDCat / DNle). En el cas que hi hagi despesa econòmica anotar en detalls de viabilitat.
- Investigar si ja existeix una plataforma acreditativa i es pot aplicar al nostre projecte.
- En cas que s'hagi de desenvolupar el mòdul, aconseguir el *know-how* sobre com demanar un certificat, recollir-lo al servidor d'aplicacions i comprovar la seva validesa contra les CAs associades.
- Avaluar necessitat de gestionar bloquejos d'entrada independent del que digui la CA.
- Avaluar necessitat de nivell d'accés als diferents mòduls de gestió (suggeriment: independitzar nivell de permisos dels mòduls, associar segons rols o nivells)
- Marcar punts i esdeveniments on fer log d'activitat.
- Valorar estratègies de mantenir l'acreditació a la sessió un cop acceptada.
- Prototipus de l'aplicació (GUI, sortida web)
- Codificació de la solució dissenyada.

2.4.5 Casos d'ús

Entenent la problemàtica del mòdul analitzat i després de varies iteracions analítiques presentem el diagrama de casos d'ús definitiu. Aquest té com actors: l'usuari, el servidor i el servei CATCert-PSIS:

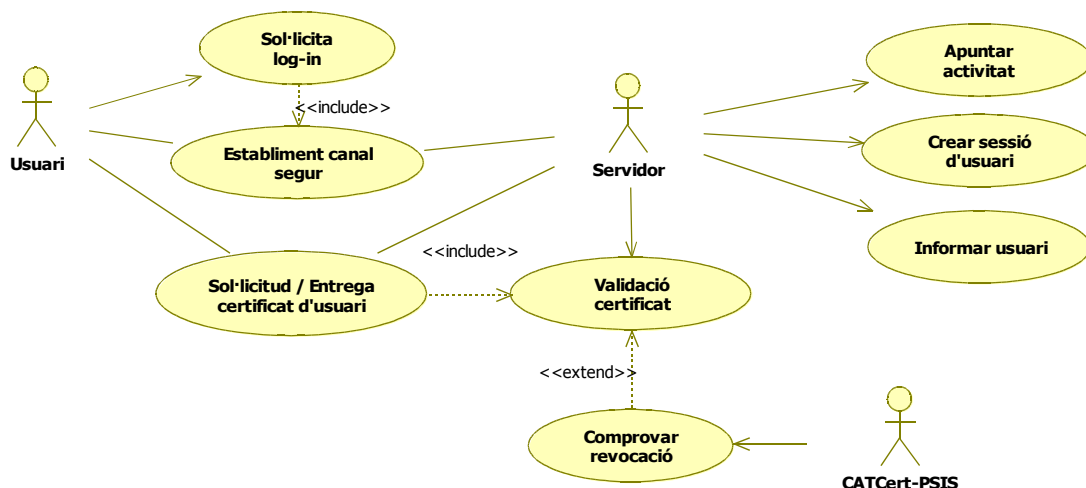


Fig. 2.1 Diagrama UML de casos d'ús del Mòdul 01

Segons el diagrama, l'usuari col·labora en només tres casos d'ús: en la sol·licitud de *log-in*, en l'establiment del canal segur i en l'entrega del certificat d'usuari. Aquestes associacions modelen els objectius OBJ_01_01 i OBJ_01_04.

D'altra banda el servidor, a més de col·laborar en l'establiment del canal segur i la sol·licitud del certificat d'usuari, també: apunta l'activitat de la transacció, crea la sessió d'usuari i informa aquest del procés. Aquestes associacions modelen els objectius OBJ_01_06, OBJ_01_02 i OBJ_01_3.

El modelat de l'objectiu OBJ_01_05 s'ha obviat en aquest diagrama doncs és un cas d'ús lligat als mòduls que gestionen els serveis de la plataforma i no pas a mòdul que ara analitzem: no es pot tancar una sessió d'usuari just en el procés que l'usuari demana crear-la; en tot cas es tancarà després.

En referència a l'actor CATCert-PSIS, només comentar que es tracta d'un servei extern que s'associa en el cas d'us de la comprovació del estat de revocació dels certificats. Els detalls d'aquesta associació s'expliquen a la secció 2.7 i més en profunditat al capítol 3.

2.4.6 Programació temporal

Inicialment el calendari d'execució d'aquest mòdul, proposat a l'informe previ d'aquest projecte, es programava segons el següent diagrama de Gantt:



Fig. 2.2 Diagrama de Gantt que representa el calendari d'execució del Mòdul 01

Llegenda:

- Tasques d'investigació
- Tasques d'avaluació i decisió
- Tasques de documentació
- Tasques de prototipus
- Tasques de codificació

Aquest calendari marcava una data de començament el 8 de gener i una data d'acabament el 10 de febrer.

L'execució de les tasques programades es van portar a terme amb una desviació temporal ocasionada per les tasques d'investigació. Al capítol 6 es comenten els detalls. No obstant la desviació en l'execució d'aquest mòdul es va compensar afegint recursos d'hores no programades.

2.5 Mòdul 02 – Registre d'entrada telemàtic

Aquest segon mòdul que ara s'analitza serà el servei de registre d'entrada telemàtic. Dins aquest projecte, és l'únic servei on el productor de la informació és l'interessat i el consumidor l'administració.

Fent un extracte de les lleis de procediment administratiu i centrant-nos en el que ens interessa portar a terme, podem dir que, tret d'aquells que s'inicien d'ofici, tots els procediments administratius que es realitzen a la corporació local s'inicien a partir d'un registre d'entrada. Un registre d'aquest tipus és una petició formal escrita per l'interessat, o el seu representant. Aquest document també inclou un extracte amb la petició explícita de les accions o serveis sol·licitats a la corporació i objecte del registre. És possible adjuntar amb el registre altres documents annexos relacionats amb el que s'exposa.

Aquesta acció es tradueix de forma funcional a l'actualitat, en aquest Ajuntament, a la recepció en paper de les sol·licituds dels interessats i la gestió administrativa des d'una única línia de registre. L'única interfície d'aquest procés és l'oficina de registre dins l'oficina d'atenció al ciutadà. És per tant que el servei queda bloquejat en horari i limitat geogràficament. Obliga els interessats a fer el desplaçament físic en hores d'oficina.

Motivat per aquesta problemàtica i tenint en compte que el registre d'entrada és un servei

al que tothom ha de poder tenir accés, veiem clarament que hauria de ser el primer servei en automatitzar-se per obrir la possibilitat d'externalitzar aquesta gestió.

Beneficis de la implementació:

- Eliminació de la franja horària de gestió de l'oficina.
- Descàrrega del treball de l'oficina, reduint les cues d'aquest servei.
- Descentralització del servei amb la possibilitat d'aplicació en d'altres oficines remotes.

2.5.1 Objectius específics

Després d'analitzar el procés de registre tal com s'explica a l'apartat anterior i segons diverses iteracions en la recerca d'objectius, sintetitzarem en aquest mòdul la funcionalitat requerida en aquest servei de registre d'entrada telemàtic.

L'objecte d'aquest projecte es troba centrat a l'interessat, és per això que tots i cadascun dels serveis que s'integraran tenen com objectiu principal ésser útils per l'usuari. És doncs aquest principal motiu que es deriven les següents conclusions:

- No cal recollir les dades del representant doncs és l'usuari acreditat el que registrarà en el seu nom.
- Si l'interessat no es troba en la base de dades d'interessats s'ha de permetre l'alta d'aquests. Les dades del certificat digital validen la creació de l'interessat.
- El formulari de recollida de dades serà semblant al formulari actual en paper. No és necessari recollir altres paràmetres que l'interessat pot desconèixer com per exemple: el departament gestor o el codi d'assumpte. En tot cas serà un delegat responsable de registre qui depurarà aquest paràmetres un cop rebuda la petició. Aquest comportament és el mateix que es fa en el procés de registre actual.

En referència als sistemes d'emmagatzemament de dades corporatius hem de dir que existeix un aplicatiu de registre d'entrada/sortida de documents que gestiona la base de dades dissenyada per aquesta finalitat. És objectiu d'aquest mòdul integrar la producció dels nous registres telemàtics amb aquesta base de dades existent.

Abans hem afegit el detall que la documentació annexa s'ha de poder adjuntar amb el registre. Analitzant aquest objectiu es proposa com a solució fer servir una tramesa d'arxiu entre el client i el servidor.

Els requeriments i les accions derivades per portar a terme la integració amb la base de dades existent i el sistema de càrrega de l'arxiu annex es comenten a la secció 2.5.2 i al capítol 3.

Tenint en compte aquestes consideracions codificarem els objectius específics d'aquest mòdul segons la següent taula descriptiva:

CODI	Descripció
OBJ_02_01	Les dades de registre s'introduiran en un formulari web semblant al formulari actual en paper.
OBJ_02_02	Abans de mostrar el formulari de registre es comprovarà si l'usuari existeix a a

	base de dades d'interessats. En cas que no estigui es mostrarà una pantalla informativa amb formulari d'alta d'interessat.
OBJ_02_03	Es farà validació de les dades introduïdes als formularis abans d'enviar-les al servidor. El camp extracte serà condicionant per acceptar o no el registre.
OBJ_02_04	Un cop les dades arribin al servidor es tornaran a comprovar i en cas d'èxit es guardaran a la base de dades segons un procediment transaccional.
OBJ_02_05	El procediment transaccional generarà una clau única de registre.
OBJ_02_06	El procediment transaccional apuntarà l'activitat i generarà un número únic d'operació.
OBJ_02_07	Es construirà una pantalla informativa amb la clau de registre i número d'operació.
OBJ_02_08	La pantalla de l'objectiu OBJ_02_07 informarà de la possibilitat d'adjuntar un arxiu al registre processat. S'han d'habilitar els controls necessaris un únic arxiu i enviar-ho.
OBJ_02_09	S'ha de visualitzar el progrés del procés de càrrega de l'arxiu i s'ha de permetre cancel·lar la tramesa.
OBJ_02_10	Un cop finalitzada la càrrega de l'arxiu, el servidor el copiarà al repositori d'arxius de registre afegint la clau de registre com a prefixe del seu nom.
OBJ_02_11	Es calcularà un codi criptogràfic, hash md5, un cop l'arxiu es trobi al repositori.
OBJ_02_12	S'informarà l'usuari de l'estat del procés de tramesa de l'arxiu i del codi calculat.
OBJ_02_13	S'anotarà l'activitat en tot moment i també es relacionarà el registre d'entrada amb l'arxiu enviat en una base de dades.
OBJ_02_14	La pantalla de l'objectiu OBJ_02_07 ha de permetre sol·licitar la visualització del registre efectuat.

2.5.2 Requeriments

A la llista d'objectius anterior es fa referència a la base de dades de registre i d'una operació transaccional. La base de dades existent a la corporació porta en funcionament molts anys i emmagatzema els registres d'entrada/sortida des de que aquest servei es va informatitzar.

Existeix un aplicatiu client d'aquesta base de dades, distribuït per tota la xarxa local de l'Ajuntament, que se n'encarrega de facilitar la gestió de registre.

Després de la seva anàlisi funcional, el procés de registre necessita el càlcul d'un comptador de document. Aquest càlcul ha de garantir que, un cop realitzat, no es produiran duplicats. Existeix una taula de comptadors que guarda l'últim número de document de registre calculat. El càlcul d'assignació de nou número és tan senzill com incrementar en 1 el número existent. Existeix un comptador específic per entrades i un per sortides. Així mateix, existeix també un comptador específic per cada any de registre.

Funcionalment requerirem que el procés de càlcul apliqui un bloqueig abans de llegir el número actual i no l'alliberi fins haver aplicat l'increment d'aquest. Per garantir que això succeeix encapsularem tot el procés de registre en una operació transaccional contra la base de dades.

Aquesta operació no només s'encarregarà de fer el càlcul del comptador sinó que just després d'aconseguir-ho intentarà el registre del document amb la nova clau. Si hi es produeix alguna excepció executant el procés, es deixarà la base de dades sense canvis i s'informarà.

Per garantir que el bloqueig d'exclusivitat es fa utilitzant el menys temps possible, hem de forçar que abans d'iniciar l'operació transaccional estiguin disponibles tots els paràmetres necessaris.

D'altra banda a l'objectiu OBJ_02_08 hem introduït la possibilitat d'adjuntar un arxiu un cop el procés de registre ha finalitzat.

En fases d'anàlisi anteriors i fins i tot en un prototipus de formulari de registre que es va implementar, es pensava en enviar un o varis arxius adjunts dins el mateix formulari de recollida de dades del registre. Després de tenir en compte la casuística del procés de registre, vam decidir que la manera correcta de fer-ho era el fer possible la tramesa dels arxius després de l'operació transaccional de registre, un cop ja tenim conformitat de l'assentament i número de document. També per facilitar la gestió vam limitar a un únic arxiu.

Aquest particionament del procés de registre en dues etapes: primer el registre de les dades i després la tramesa de l'arxiu adjunt; es justifica per dues raons. La primera: és possible que no sempre en sigui necessari adjuntar un arxiu, en aquest cas després de la primera etapa el procés finalitza. La segona: la tramesa de l'arxiu, per petit que aquest sigui, incrementa les dades enviades al servidor i, en conseqüència, les dades transferides. Aquest increment indueix a que la probabilitat d'error de les transmissions sigui més elevada, per tant, es poden donar casos en que el registre no sigui possible degut a problemes de comunicacions. Ja que en el cas d'enviar les dades de registre i arxiu conjuntes, l'operació de registre no s'efectuaria fins arribar-ne totes; un usuari bàsic pot fer-se a la idea que s'ha enregistrat la petició quan en realitat no ha estat possible. També i fent referència aquesta problemàtica es justifica la necessitat de l'objectiu OBJ_02_09.

En referència a la limitació a un únic arxiu la justificació és senzilla. La relació conceptual entre document de registre i arxius adjunts es d'un a varis. Molts procediments requereixen l'aportació de diferents documents adjunts a la petició formal de registre. No obstant en el domini de la informació electrònica és fàcil empaquetar varis documents dins un únic arxiu. Amb aquesta limitació de l'objectiu OBJ_02_08 es vol potenciar l'ús d'eines d'empaquetament i compactació d'arxius per part de l'usuari amb dos objectius finals: garantir un consum de disc equilibrat i simplificar els processos de registre i associació de documentació.

L'acceptació d'arxius produïts i propietaris de l'usuari a través d'un canal segur queda exempta d'alteracions. No obstant per la tranquil·litat de les dues parts, es calcularà un codi criptogràfic segons un algorisme de resum de missatges –*message digest*–, com és el MD5, àmpliament utilitzat. Aquest codi identificarà única i exclusivament l'arxiu enviat. S'informarà l'usuari d'aquest codi un cop rebut l'arxiu i s'anotarà a la base de dades d'arxius adjunts. També serà consultable junt amb la informació registrada. Aquest codi és garantia de que el fitxer no ha estat alterat durant la seva tramesa doncs l'usuari

pot comprovar la correctesa del codi fent el càlcul al seu arxiu local pel seu compte.

2.5.3 Punts forts / febles

Paral·lelament a l'anàlisi anterior fem una abstracció de punts forts i febles segons la següent classificació:

Punts forts:

- Descentralització de la gestió d'aquest tràmit universal.
- Desbloqueig territorial i temporal (oficines/horaris)
- Descàrrega de treball a l'oficina de registre i minimització de cues i retards d'alta al sistema.

Punts febles:

- Obligatorietat de certificació que acrediti el registre/sol·licitud
- Securitzar l'enviament d'arxius mitjançant sistemes que garanteixin la no-alteració dels mateixos.
- Departaments legals neguitosos a les noves tecnologies de la informació.
- Es possible que posterior al registre telemàtic es necessiti col·laboració per depurar les dades introduïdes per l'usuari.

2.5.4 Planificació

Per portar a terme la realització dels objectius analitzats, s'ha realitzat una distribució de tasques per aconseguir una programació estimada en temps.

Al llistat hi ha tasques purament de recerca i tasques on la seva execució correspon a l'etapa de disseny o bé desenvolupament. A la secció 2.7 es comenta part dels resultats de recerca i no és fins el capítol 3 i 4 que s'amplien els detalls d'execució.

Segons la distribució d'objectius, les tasques planificades serien:

- Avaluar necessitat de moderador. (El ciutadà farà una sol·licitud de registre i el moderador validarà definitivament el registre – No es recomana)
- Estudiar les opcions de les possibles solucions d'enviar dades (post-ssl vs webservice-xml)
- Analitzar sistema d'upload d'arxius de manera segura. Clau hash/MD5 al servidor i al client. Comprovacions.
- Inventari de dades estructurades. Documentar número de camps, tipus de dades a recollir, validacions prèvies. Facilitats a l'hora d'emplenar.
- Disseny del formulari de recollida de dades.
- Analitzar procés de notificació de registre (info en pantalla + email).
- Generar document de requeriments funcionals.
- Prototipus del mòdul

- Codificació de la solució

2.5.5 Casos d'ús

El diagrama de casos d'ús inicial, presentat junt amb l'anàlisi de l'informe previ era el següent:

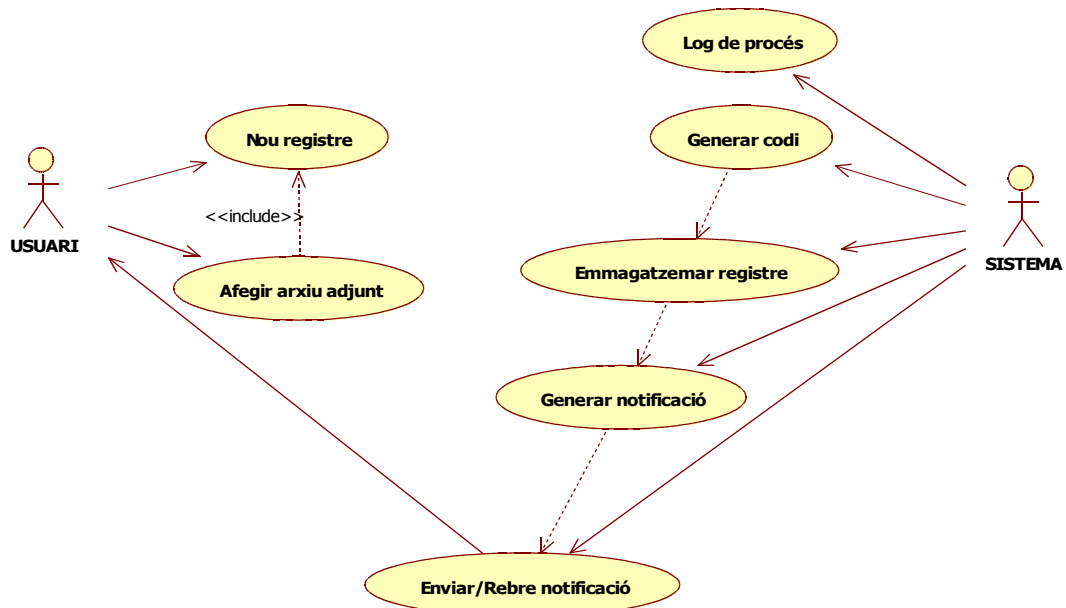


Fig. 2.3 Diagrama de casos d'ús inicial del Mòdul 02

Dels objectius comentats amb anterioritat fem l'abstracció als casos d'ús reflectits en el següent diagrama:

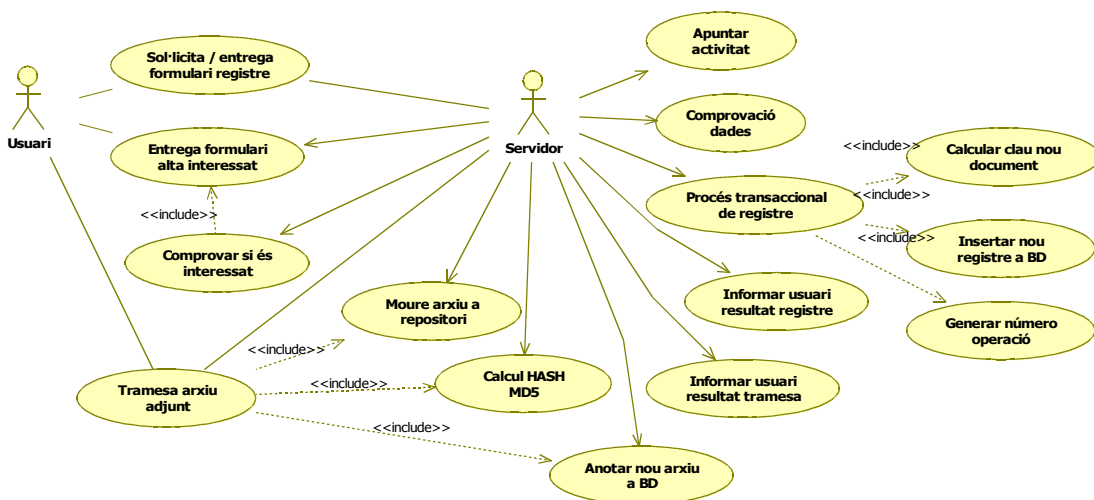


Fig. 2.4 Diagrama de casos d'ús definitiu del Mòdul 02

Comparant aquest diagrama amb el presentat a l'informe previ –figura 2.3– veiem clarament els canvis i refinaments resultat d'una anàlisi iterativa centrat en la recerca

d'objectius.

El diagrama definitiu modela com actors a l'usuari i el servidor. Aquests s'associen amb tot un conjunt de casos d'ús que es troben correspondència amb els objectius finals marcats per aquest mòdul.

No comentarem en aquest capítol el disseny funcional del mòdul doncs això es tractarà en profunditat a la secció 3.4 il·lustrat pels corresponents diagrames d'activitat i seqüència. No obstant notem que aquest diagrama associa correctament l'actor usuari en la petició i entrega de les dades del registre i, si fos cas necessari, les dades necessàries per processar l'alta d'interessat. També en la tramesa de l'arxiu adjunt. Referent a l'actor servidor, aquest s'associa a tot un conjunt de casos d'ús relacionats amb el procés transaccional de registre, el tractament de l'arxiu adjunt, així com en les generacions d'informació del procés.

2.5.6 Programació temporal

Segons el llistat de tasques configurat amb anterioritat i en dates d'anàlisi del sistema, es va proposar el següent calendari d'execució:

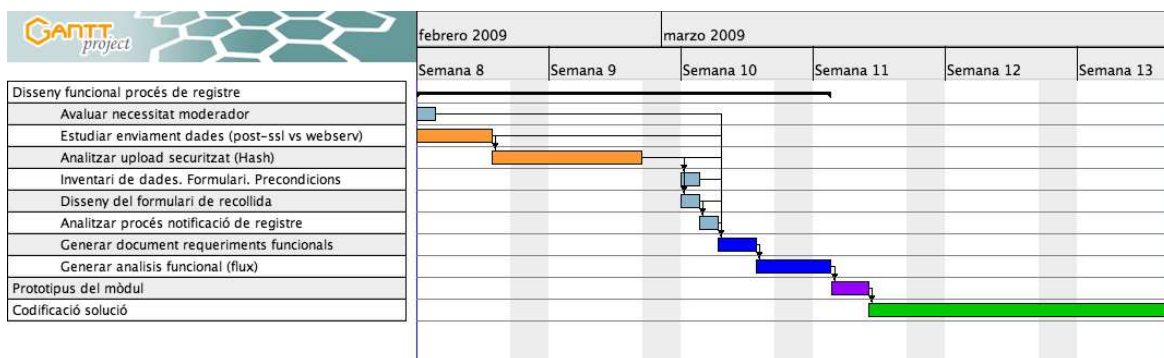


Fig. 2.5 Diagrama de Gantt que representa el calendari d'execució del Mòdul 02

Llegenda:

- Tasques d'investigació
- Tasques d'avaluació i decisió
- Tasques de documentació
- Tasques de prototipus
- Tasques de codificació

Aquest calendari marcava una data de començament el 16 de febrer i una data d'acabament el 27 de març.

En referència a l'execució d'aquest calendari i avançant el que es comentarà en detall a les conclusions del projecte, no hi va haver increment en la desviació temporal a conseqüència del desenvolupament d'aquest mòdul.

2.6 Mòdul 03 – Procediments informatius

En aquest tercer mòdul s'agrupen els serveis de consulta de dades a través de la plataforma telemàtica. A diferència del mòdul 02, on el productor de la informació era l'interessat i el consumidor l'ajuntament, ara parlarem de diferents serveis on el productor de la informació és l'ajuntament i el consumidor l'usuari.

Es volen desenvolupar i integrar els serveis de consulta majoritàriament demanats pels ciutadans i que no indueixin gestions internes ni sobrecàrrega en cap àrea de treball. Amb aquest objectiu i segons una estadística recollida en una visita a l'Oficina d'Atenció al Ciutadà (OAC) d'aquest Ajuntament, obtenim que els serveis de consulta més demanats es poden agrupar en tres categories:

- Justificants de registre i consulta d'expedients
- Justificants del Padró Municipal d'Habitants (PMH)
- Justificants de pagaments: plus vàlues, deutes, rebuts, ...

Com és evident la implementació i integració de tot els serveis de consulta que ofereix l'Ajuntament queda fora del domini temporal d'execució d'aquest projecte. Degut a aquest motiu que s'ha fet una tria dels serveis de més pes buscant un equilibri entre versatilitat i limitacions temporals.

La selecció de serveis és la següent:

- Consulta del registre d'entrada / sortida de documents.
- Justificat d'empadronament.
- Justificant de convivència.
- Consulta d'expedients.
- I consulta de pagaments.

En quant a les dades de consulta, aquestes es troben emmagatzemades al sistema gestor de bases de dades i pels serveis seleccionats s'haurà d'estudiar la documentació i arquitectura d'un total de 4 bases de dades: registre, expedients, padró i ingressos. L'anàlisi s'exposa a l'apartat 2.6.2 de requeriments.

2.6.1 Objectius específics

Com a objectiu comú i principal d'aquests serveis de similar funcionalitat, tenim el de fer arribar la informació a l'interessat de forma automàtica i per canals telemàtics. El ciutadà és el propietari i té el dret de consulta.

Un altre objectiu implícit és el de garantir les comunicacions segures entre el client i el servidor. La informació és propietària i privada. No pot viatjar per un canal no xifrat.

Finalment és molt interessant que l'usuari pugui disposar d'una còpia d'aquesta informació fent servir un fitxer d'intercanvi reconegut, com per exemple el PDF. Aquesta còpia hauria

de gaudir de reconeixement d'originalitat pel que s'inclourà una signatura digital de l'Ajuntament en la seva generació.

Tenint els detalls comentats anteriorment la llista d'objectius es configura de la següent manera:

CODI	Descripció
OBJ_03_01	Localitzar usuari a la base de dades destí de la consulta segons el servei sol·licitat. Si no es troba correspondència avisar.
OBJ_03_02	Filtrar llistat o registre específic del l'usuari acreditat utilitzat correspondència trobada segons OBJ_03_01.
OBJ_03_03	Formatejar la informació utilitzant una plantilla semblant al document que es genera en paper actualment.
OBJ_03_04	La plantilla amb la informació ha de poder-se veure en un format d'impressió.
OBJ_03_05	La plantilla amb la informació ha de poder-se descarregar en format PDF signat electrònicament per l'Ajuntament.
OBJ_03_06	Anotar activitat de l'usuari. Els documents generats portaran un codi d'operació de log.

Aquesta llista d'objectius és comuna i extensible a cada servei associat a aquest mòdul.

2.6.2 Requeriments

Tal i com s'ha introduït al començament d'aquesta secció i tenint en compte la selecció de serveis projectada per aquest mòdul, trobem, com a requeriment de model de dades, el documentar-se sobre l'arquitectura de quatre bases de dades corporatives.

Aquestes, tal i com hem enumerat abans, són:

- Registre d'entrada i sortida
- Expedients
- Padró Municipal d'Habitants (PMH)
- Ingressos

Aquest és el principal problema de l'anàlisi d'aquest mòdul doncs cada servei depèn de la consulta d'unes dades específiques i aquestes poden trobar-se distribuïdes en bases de dades complexes. La prova de que això és cert la tenim si ens fixem que de 5 serveis seleccionats, s'ha de fer estudi de 4 bases de dades diferents.

El cas de les dues primeres bases de dades, Registre i Expedients, el treball de mineria de dades és senzill doncs es coneix l'esquema d'aquestes ja que són de construcció i manteniment propi.

El cas del PMH i Ingressos és relativament complex ja que parlem de bases de dades grans. Si utilitzem el número de taules d'una base de dades com indicador de la seva complexitat, tenim que l'esquema de la BD del PMH està constituït per prop de 100 taules i que el de la BD d'Ingressos intervenen més de 300 taules.

No obstant i amb aquest números, el disposar de contracte de manteniment amb les empreses proveïdores d'aquestes dues aplicacions hauria de suposar un factor d'ajuda.

D'altra banda i com a requeriment funcional, s'ha proposat com objectiu OBJ_03_05, la generació dinàmica de contingut PDF signat electrònicament. La generació d'aquest tipus de documents és una característica comuna a l'àmbit de la documentació electrònica. Analitzant aquest objectiu i tenint en compte que la informació es genera i es transmet mitjançant una plataforma telemàtica i més en concret a través d'una interfície web, ens trobem el problema que seria molt interessant trobar-ne la solució equilibrada entre facilitat i semblança dels informes web amb els documents PDFs generats.

Les pantalles generades per a la visualització integrada amb el portal, en HTML i del PDF generat es poden consultar al capítol d'annexos a l'apartat 8.2.

Existeixen eines i controls, gratuïts i de pagament, dissenyats per aquest tipus de tasques. La tria d'aquest component clau per satisfer l'objectiu en qüestió es comenta i justifica a l'apartat 2.7.2 d'aquest capítol.

2.6.3 Punts forts / febles

Tal i com hem fet en les seccions dels mòduls anteriors fem una abstracció de punts forts i punts febles de la següent manera:

Punts forts:

- Afegeix funcionalitats bàsiques que tindran impacte en la descentralització de processos reduint el treball de gestió de la Oficina d'Atenció al Ciutadà (OAC).
- Desbloqueig territorial i temporal (oficines/horaris)
- Es disposa de contracte de manteniment amb les aplicacions de tercers. Punt que facilita, a priori, les tasques de mineria de dades contra base de dades amb esquemes complexos i desconeguts.
- Algunes aplicacions s'han desenvolupat internament. Punt que facilita, en aquest cas directament, les tasques de mineria de dades.
- Som propietaris de les bases de dades i de la plataforma d'emmagatzemament.
- La generació de documentació electrònica en format PDF signat electrònicament permet extendre la usabilitat i validesa legal de la informació generada fora de la plataforma.

Punts febles:

- Complexitat dels esquemes de les BD de tercers. Potser existeixen dades molt fragmentades.
- Problema de localització i equivalència de persones. Potser amb el DNI no serà suficient per trobar una correspondència única per tal de garantir la cessió de dades.
- Garantir la privacitat de l'accés.

2.6.4 Planificació

Inicialment es pensava en aquest mòdul com un catàleg dinàmic de serveis configurable per usuari i segons un gestor de continguts però a la fase d'anàlisi es decideix integrar, com ja hem comentat, els serveis majoritàriament demanats. És llavors que es deixa el catalogador de serveis com ampliació futura del projecte. No obstant aquestes tasques s'han tingut en compte durant aquesta fase i són reflectides en el llistat de tasques següents i en la seva programació temporal.

Tasques:

- Catalogar els serveis
- Analitzar requeriments del sistema catalogador de serveis. Valorar necessitat d'una implementació dinàmica amb gestor de continguts.
- Analitzar requeriments de dades –consulta– contra cadascuna de les bases de dades. Comunicar amb el proveïdors si fos necessari.
- Estudiar diferents metodologies d'implementació de les interfícies. (Script iteratiu vs webservice-xml)
- Disseny funcional de cada interfície
- Disseny del format de sortida de les dades.
- Prototipus del mòdul
- Codificació de la solució

2.6.5 Casos d'ús

Fent referència al diagrama de casos d'ús d'aquest mòdul dissenyat inicialment a l'informe previ:

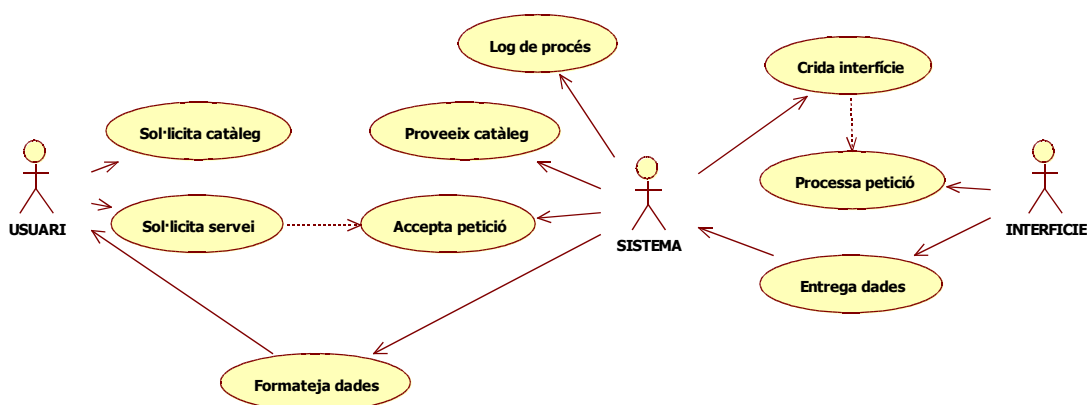


Fig. 2.6 Diagrama de casos d'ús inicial del Mòdul 03

i comparant-ho amb el diagrama de casos d'ús amb el que hem modelat el comportament que satisfà els objectius analitzats:

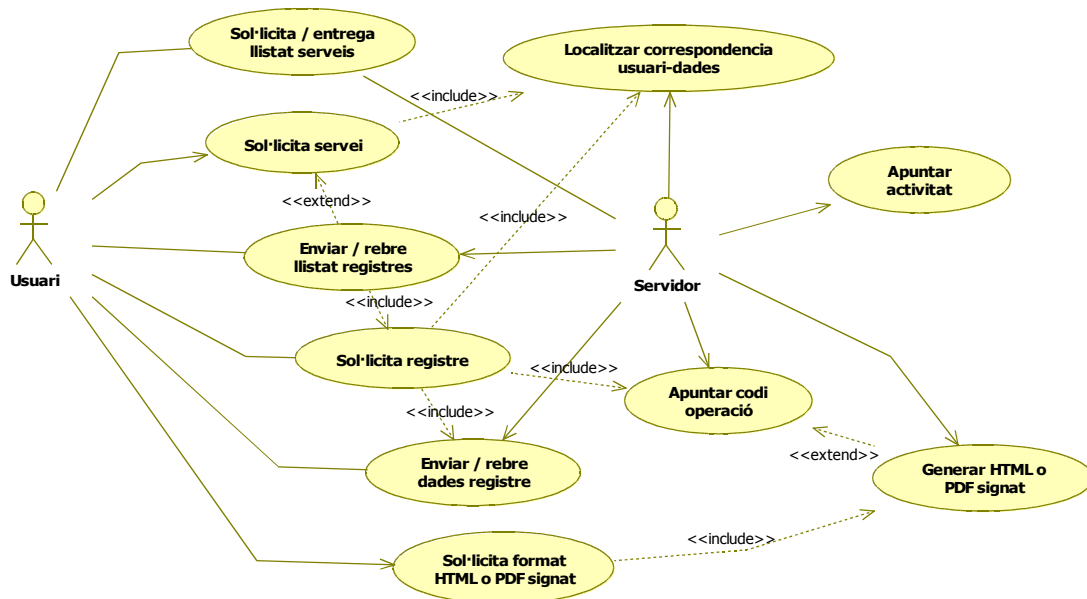


Fig. 2.7 Diagrama de casos d'ús definitiu del Mòdul 03

apreciem dos canvis importants. El primer és la desaparició de l'actor Interfície. Aquest canvi ha estat motivat després de diverses fases de refinament. L'anàlisi definitiva no justifica la creació d'una interfície específica per a cada base de dades, amb tota la logística de missatgeria Servidor-Interfície, doncs el paquet de dades a extreure d'aquesta no és gran i tampoc aquest projecte té com a objectius la interoperabilitat de la plataforma amb d'altres sistemes consumidors d'aquestes dades. D'altra banda hem justificat a la planificació de tasques l'abandonament del sistema catalogador per un llistat estàtic de serveis àmpliament demanats.

No obstant, observem com el nou diagrama modela els casos d'us per satisfer els objectius finals després de la recerca analítica.

2.6.6 Programació temporal

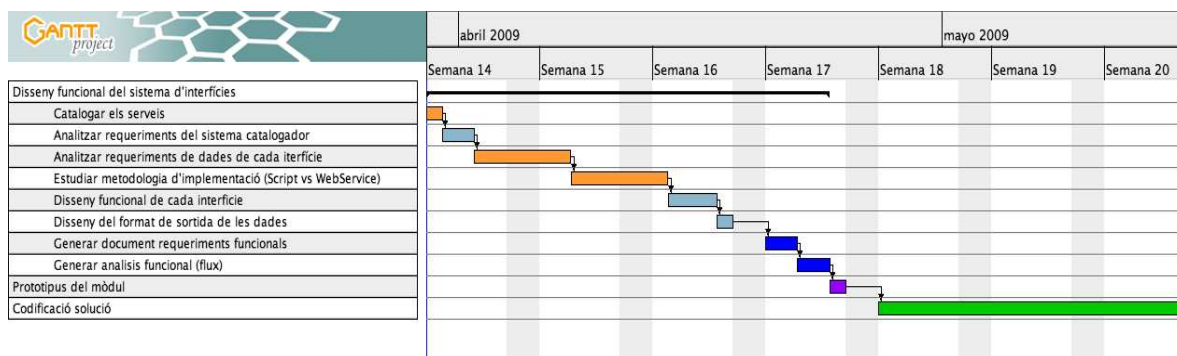


Fig. 2.8 Diagrama de Gantt que representa el calendari d'execució del Mòdul 03

Llegenda:

■ Tasques d'investigació

- Tasques d'avaluació i decisió
- Tasques de documentació
- Tasques de prototipus
- Tasques de codificació

Aquest calendari marcava una data de començament el 30 de març i una data d'acabament el 15 de maig.

2.7 Àmbit d'execució i recursos

En apartats anteriors hem comentat que aquest projecte és una iniciativa de l'Ajuntament de Ripollet amb l'assessorament del seu departament de Serveis Informàtics. Una estratègia valorada en quant a la redacció i assignació d'aquest projecte ha estat el desig optimista de començar a caminar amb destí fixe de l'administració electrònica, aprofitant la situació acadèmica actual d'un dels membres del departament. Recolzar en aquest la responsabilitat i confiança suficient i necessària com per realitzar l'execució del projecte implicava una mínima inversió de recursos i el resultat es favorable per ambdues parts. D'una banda l'Ajuntament veurà en execució la plataforma telemàtica de serveis i d'altra banda l'executor veurà el seu treball com una realització acadèmica reconeguda.

A continuació classificarem els recursos segons les subseccions següents:

2.7.1 Recursos humans

Hem introduït abans el departament de Serveis Informàtics de l'Ajuntament. Ampliarem ara la informació d'aquest i farem uns breus comentaris en quant a la seva projecció històrica i tècnica.

Aquest departament el componen tres persones de perfil tècnic i s'encarrega, a l'actualitat, de col·laborar i executar projectes relacionats amb les noves tecnologies de la Informació. Tant és així que donem suport a més de 300 treballadors distribuïts en més de 10 centres de treball.

Dins el catàleg de serveis que ofereix en destaquem, per posar exemples, l'assessorament tecnològic en compres i decisions de negoci, l'execució de projectes de caire tècnic com la integració de sistemes mitjans, desenvolupament de software de gestió organitzativa, desenvolupament i promoció web, manteniment de la infraestructura de xarxa i del centre de procés de dades, responsabilitat del sistema de còpies de seguretat i redundància de dades o bé el realitzar auditories de seguretat o de compliment de la LOPD.

Les càrregues de treball es distribueixen entre els tres membres de forma especialitzada. Un d'ells s'encarrega de la part de suport tècnic d'instal·lacions i manteniment de sistemes. El cap del departament porta la logística d'aquest i dona suport a projectes on l'àmbit d'execució és l'àrea de gestió econòmica. I el tercer membre executa projectes de

gestió en relació amb àrees organitzatives i desenvolupament web.

A l'any 2001 els motius de contractació d'un expert tècnic en execució de projectes web es varen justificar amb l'objectiu de la creació del primer portal web municipal i la meua contractació va ser resultat d'aquesta necessitat.

Durant el curs acadèmic 2007-2008 vaig completar la totalitat d'assignatures i crèdits que restaven al meu expedient i en veure les possibilitats d'execució del projecte final de carrera com a projecte d'empresa no vaig dubtar en proposar-ho com a negociació.

A mitjans del 2008 el resultat d'aquesta negociació amb el departament de Recursos Humans i amb un dels impulsors del projecte, el Regidor de Comunicació i Noves Tecnologies, va ser d'una conformitat absoluta d'ambdues parts en quant a l'execució del projecte.

Es va presentar una estimació d'hores i es va aprovar l'assignació de mitja jornada diària per l'execució d'aquest projecte. També es va ratificar el conveni de col·laboració entre l'Ajuntament i la UAB en la Junta de Govern Local el dia 24 de desembre de 2008, sota la supervisió del responsable de Recursos Humans.

Amb el consentiment d'ambdues parts interventores i amb l'assignació de 4 hores diàries a l'execució del projecte, es comptava amb els recursos humans suficients per a la seva execució.

Més endavant, a l'estudi de viabilitat, es tornarà a justificar.

2.7.2 Recursos materials

L'execució de projectes de caire similar al que en aquesta memòria es documenta, afavoreix el punt de partida en quant a recursos tècnics. El departament i el centre de procés de dades ja disposen dels recursos necessaris i suficients com per l'execució de tots i cadascun dels objectius.

Eines disponibles

En el moment de l'execució d'aquesta anàlisi, i degut a tractar-se d'eines tecnològiques en evolució constant, és difícil sinó impossible enumerar totes les possibilitats en quan a recursos materials disponibles.

Amb l'objectiu de minimitzar l'impacte econòmic en l'execució del projecte, hem centrat la recerca d'eines en aquelles que siguin de programari lliure o bé que estiguin degudament llicenciades a nom de l'Ajuntament.

Pel que fa a les necessitats d'aquest projecte, segons els objectius redactats, es necessitaran eines de diferent tipus. L'inventari de recursos haurà de cobrir els següents fronts:

- Plataforma hardware servidora.
- Servidor d'aplicacions web.

- Sistema gestor de bases de dades.
- Eina de codificació i desenvolupament web.
- Controls específics per a la realització de tasques comunes.

Quant a la **Plataforma hardware servidora**, l'Ajuntament disposa d'una màquina sobre dimensionada que se n'encarrega de l'execució del servidor d'aplicacions web. Aquesta màquina està basada en una plataforma Intel Xeon amb quatre nuclis de processament, quatre gigabytes de memòria RAM i dos dispositius RAID1 (un per sistema i un altre per dades). La màquina es troba en condicions òptimes de funcionament i manteniment i es troba instal·lada físicament en un dels racks del centre de procés de dades de l'Ajuntament.

També existia la possibilitat de destinar un equip de treball com a servidor de pre-producció o fins i tot el de crear una màquina virtual executant-se a la màquina de desenvolupament.

En quant al **servidor d'aplicacions web**, la plataforma hardware existent, executa el sistema operatiu de servidor família Microsoft Windows 2003 integrat amb Internet Information Server 6 i .NET Framework 2.0. En l'actualitat aquest servidor d'aplicacions s'encarrega de donar servei al portal web municipal i altres espais webs específics.

D'altra banda s'han tingut en compte altres servidors d'aplicacions i especialment l'Apache-Tomcat, de Apache Software Foundation, per la seva versatilitat i fiabilitat reconeguda en quant a contenidor de tecnologies Java-Servlets i Java Server Pages (JSPs).

En referència als **sistemes gestors de bases de dades**, l'Ajuntament disposa d'un servidor dedicat a aquesta missió i el SGBD existent és Microsoft SQL-Server 2000. A l'actualitat, totes les bases de dades de caràcter corporatiu es recolzen sobre aquesta plataforma. També hem valorat la disponibilitat i reconeixement de sistemes gestors de bases de dades com MySQL per la seva potència, disponibilitat multi plataforma i alt grau de satisfacció.

Quant a les **eines de desenvolupament** l'Ajuntament disposa d'una llicència de Microsoft Visual Studio Developer 2003 Enterprise Architect. Però s'ha valorat l'editor Crimson Editor, editor lliure amb comprovacions sintàctiques, pel desenvolupament de fitxers de recursos i codi, el Microsoft Visual Web Developer Express Edition, eina gratuïta, pel desenvolupament de solucions en ASP.NET i .NET Framework i l'Eclipse com a plataforma d'anàlisi i desenvolupament de solucions en Java.

En la **integració de controls** per a la realització de tasques comunes s'ha estudiat els controls iText i ABDPdf per a la generació dinàmica de contingut PDF; els controls FileUploadAjax i el NeatUpload per a la integració d'un component de tramesa d'arxius amb funcionalitats Ajax per implementar una barra de progrés.

En relació a la comprovació de l'estat de revocació dels certificats d'usuari, s'ha estudiat la complexitat de cridar els serveis de cadascuna de les Agències Certificadores o bé la integració del servei de validació de CATCert anomenat PSIS (Plataforma de Serveis d'Identificació i Signatura).

Tria de les eines i justificació

Inicialment s'inicia el desenvolupament del projecte en un entorn de pre-producció recolzat en una màquina virtual. Aquesta màquina virtual es crea com a model de la plataforma servidora existent. Els motius són la imposició de l'Ajuntament al no existir partida d'inversió a càrrec d'aquest projecte i l'objectiu d'evitar desviacions en les programacions temporals durant l'execució de les tasques de consultoria i investigació.

No només els motius anteriors condicionen el desenvolupament a la plataforma hardware i servidora d'aplicacions existents. També fa força un dels objectius principals comuns. Recordem que la plataforma telemàtica que es projecta ha d'estar integrada amb el portal web corporatiu existent ripollet.cat.

Aquest portal s'executa sobre l'Internet Information Server (IIS) existent i es desenvolupa en ASP-VBScript com a server-side-scripting. No obstant, aprofitant que IIS també integra les funcionalitats de la plataforma Microsoft .NET FrameWork, la fase de codificació del projecte es fa utilitzant llenguatge C# per la seva versatilitat, capacitat de desenvolupament orientat a objectes i les característiques d'integració amb molts components existents.

Un punt a favor del .NET FrameWork contra el clàssic ASP-VBScript és que la plataforma .NET compila els fonts generant objectes ensamblats. Aquest procés dona millors resultats de rendiment i de seguretat, doncs l'execució pot ser aïllada com a procés independent. També existeix la possibilitat de compilar la totalitat del projecte i fer el desplegament des de l'eina Microsoft Visual Web Developer 2008 Express Edition. Aquesta és l'eina de desenvolupament idònia per a la fase de codificació segons la plataforma escollida. L'editor és molt potent i fa comprovacions sintàctiques, semàntiques i fins i tot resolució de referències en temps d'edició. Ajuda a mantenir un codi ordenat en l'escriptura i en la predicció i depuració d'errors.

En referència a la integració de controls de tercers amb l'objectiu de la realització de tasques genèriques, hem detectat dues tasques: la tramesa d'arxius amb destí usuari a servidor i la generació de contingut PDF.

Relacionats amb la tramesa d'arxius s'ha estudiat les funcionalitats i capacitats d'integració dels controls .NET FileUploadAjax i NeatUpload; tots dos de lliure distribució. Encara i comptant amb la seva facilitat d'integració, es va descartar la utilització del primer d'aquests degut al tractar-se d'un disseny poc flexible, en quant a opcions de configuració, i no disposar de facilitats per integrar una barra de progrés. És doncs que l'utilitzat serà el NeatUpload per la seva reputació dins els controls d'aquesta família, la quantitat de documentació disponible i la flexibilitat en quant a la seva personalització.

D'altra banda i en relació amb la generació dinàmica de PDFs, es van analitzar els controls iText i el ABCPdf de WebSupergoo i la seva flexibilitat a l'hora d'integrar-los en el projecte. En primer lloc es va fugir del control ABCPdf al tractar-se d'un control de pagament i es van començar les proves analítiques amb iText. Aquesta llibreria és molt coneguda i utilitzada en el món Java però com aquest projecte s'implementa sobre .NET les proves es van centrar en el control iTextSharp, doncs és la implementació de la mateixa llibreria Java però codificada en llenguatge C# per a la plataforma .NET

FrameWork. Aquest control permet la generació de contingut PDF amb facilitat però es van detectar certes incidències. L'objectiu per minimitzar el temps d'implementació era que el mateix contingut formatejat en HTML vàlid, pogués renderitzar-se d'una manera acceptable pel control sobre un document PDF. El control iTextSharp és molt vàlid però la versió analitzada, encara que sí era capaç de renderitzar contingut HTML, no interpretava correctament els estils CSS i això obligava a una codificació dels continguts en dos formats. L'altre control analitzat, de l'empresa webSupergoo, accepta contingut HTML i estils CSS i el resultat és molt satisfactori. Com hem comentat abans aquest control no és lliure però, encara que el preu no és una despesa significativa doncs es realment econòmic, l'empresa ofereix un canal de suport a determinats projectes i segons negociacions amb ells i després de l'estudi del servei que donarà poden arribar a deixar-ho utilitzar de forma legal i gratuïta. El resultat de les negociacions amb webSupergoo van ser positius i ens van concedir una llicència gratuïta segons la naturalesa del projecte. És doncs que fem servir aquest control per a la generació dinàmica i signatura electrònica de documents PDF.

Per la comprovació de l'estat de revocació dels certificats d'usuari s'integrarà el client de la plataforma PSIS de CATCert. Un dels serveis d'aquesta plataforma és precisament aquesta comprovació. El punt fort de fer-ne l'ús és que la invocació contra aquest servei es pot realitzar enviant com a missatge el certificat d'usuari independentment de la CA que l'hagi emès. La no integració d'aquest client ens força a fer la comprovació específica de cada certificat contra la seva CA i això té dos punts negatius: el primer, que no tots els serveis de consulta de revocació són de lliure accés; per exemple la FNMT és de pagament; i segon, que els serveis de consulta són independents i llavors hauríem d'implementar les interfícies de consulta específiques de cadascuna de les CAs a reconèixer.

Degut aquests dos motius es projectava l'acceptació de CATCert-IDCat i Policia Nacional-DNIe com a CAs reconegudes per la plataforma doncs són els certificats d'usuari gratuïts i de major proximitat en quant a la seva concessió. No obstant la integració del client ens garanteix el reconeixement de totes les CAs classificades per l'Agència Catalana de Certificació (CATCert)² d'una manera centralitzada.

Un altre punt fort en la integració d'aquest és que el consorci de l'Administració Oberta de Catalunya (AOC) subvenciona l'accés d'administracions públiques locals per tant la invocació del servei, per aquest projecte, és de lliure accés.

En referència a les eines utilitzades en la confecció de l'anàlisi i recursos per aquesta memòria s'inclouen la suite ofimàtica OpenOffice 3 per la redacció de documents. El GanttProject per la programació temporal de tasques i estimació de temps i recursos. L'StarUML per a l'anàlisi i generació de diagrames UML. I el DBDesigner 4 per a la generació de diagrames d'entitat-relació de les bases de dades utilitzades.

Al capítol 7 hi ha un recull de referències d'aquestes eines.

2 En el moment de la redacció d'aquesta memòria eren 13 entitats classificades. CATCERT. *Llista d'entitats classificades*. 2009. <http://catcert.cat/web/cat/2_1_0_entitats_certificacio.jsp> (5 de juny de 2009).

2.8 Estudi de viabilitat

En aquesta secció es justificarà que el projecte redactat i analitzat anteriorment és viable i aquesta conclusió s'extreu de l'estudi de la seva viabilitat als següents marcs:

2.8.1 Viabilitat temporal

L'anàlisi funcional dels diferents objectius del projecte dóna com a resultat un particionament d'aquest en tres mòduls –comentats a l'apartat d'objectius– i el desglossament de tasques de cadascun d'ells –comentades a l'apartat de la planificació temporal– i la seva programació temporal, justifica que el projecte es viable dins el marc de dates planificades.

2.8.2 Viabilitat econòmica

El projecte proposat és, en gran part, un desenvolupament de software i, en una menor part, la integració d'aquest en un servidors d'aplicacions amb sortida web. L'Ajuntament de Ripollet ja compta amb els recursos hardware/software necessaris i suficients com per fer front tant a la càrrega de l'aplicatiu, com al seu desenvolupament. Existeix una infraestructura de servidors web al CPD que s'encarreguen de servir els projectes web propietaris. La posada en marxa d'aquest nou projecte és la integració de nous serveis sobre els projectes web ja existents. És evident que la càrrega suportada pel sistema augmentarà però en una fase inicial serà totalment inapreciable.

En referència als recursos humans, s'ha fet una programació temporal contant amb un únic recurs-programador: jo mateix. El departament de recursos humans és conscient de la càrrega de treball que tot això suposa i ha acceptat l'assignació a aquest projecte de mitja de la meva jornada laboral diària.

Així doncs, al no existir increment en la despesa ordinària en recursos materials ni en recursos humans, el projecte és viable econòmicament.

2.8.3 Viabilitat legal

Com hem comentat anteriorment, l'Ajuntament de Ripollet ja compta amb el software de desenvolupament i d'explotació degudament llicenciat, per tant no es viola cap llei de propietat intel·lectual.

D'altra banda val a dir que la plataforma, un cop desenvolupada i integrada, no produirà ni consumirà cap tipus de dades compromentent la legalitat del projecte. Sí en tindrà accés a bases de dades que emmagatzemen dades protegides de caràcter personal, però aquest accés es garanteix que serà privat i exclusiu a la persona propietària de les dades.

Recordem que l'accés a aquestes estarà protegit per un mòdul d'acreditació d'usuari (Mòdul 01). Aquest serà l'encarregat de validar a l'usuari i securitzar les comunicacions segons un canal xifrat mitjançant una parella de claus públiques/privades de client-servidor.

A més, cap de les interfícies que s'integraran (Mòdul 03) permetran el llistat o exportació d'altres dades protegides que no siguin propietàries a l'usuari; i tampoc es tindrà accés a cap base de dades que no hagi estat declarada i enregistrada com a fitxer protegit a l'Agència Catalana de la Protecció de Dades.

Tots aquest punts seran novament analitzats en profunditat pel departament de Secretaria General de la corporació abans de la publicació del projecte. Tot i així, des d'aquest departament, ja s'ha donat llum verda a la seva execució.

En aquest marc, val afegir l'existència de la Llei 11/2007 d'Accés Electrònic dels Ciutadans als Serveis Públics, publicada al BOE el 23 de juny del 2007, que obliga a les administracions públiques a implementar una finestra electrònica per a la gestió descentralitzada d'aquest i altres serveis a partir de principis de l'any 2010. El que vol dir que la no-execució d'aquest projecte conduiria a aquest Ajuntament a una situació il·legal.

Per tot això aquest projecte és viable dins el marc legal dins el seu àmbit territorial d'execució.

2.8.4 Viabilitat tècnica

En quant a hardware, ja hem comentat a la viabilitat econòmica, que no era necessari cap increment en recursos. A més val a dir que els membres del departament d'IT de la corporació, mantenen i donen suport del CPD a la seva totalitat: màquines servidores, serveis i la seva infraestructura de xarxa; i porten desenvolupant aquestes tasques des de fa més de la creació del departament. Per tant queda del tot justificat la viabilitat tècnica en quant a coneixements de sistemes.

D'altra banda, en el meu cas en concret, treballo a l'Ajuntament com a funcionari de carrera en aquest departament des de fa 8 anys i tinc l'experiència necessària com per executar amb èxit aquest projecte, doncs en tot aquest temps he estat responsable tècnic de desenvolupament software i integració d'eines i sistemes d'informació semblants i relacionats amb el que es proposa.

A la planificació s'ha tingut en compte diferents tasques d'investigació i auto formació que garanteixen la superació dels punts febles trobats a l'anàlisi funcional. Aquestes tasques hi figuren al principi de la programació de cadascun dels tres mòduls/objectius definits anteriorment.

En el cas que es produeixin retards a la planificació per problemes tècnics, remarcarem que el projecte es dirigeix des del departament d'EIC (Departament d'Enginyeria de la Informació i les Comunicacions) i que és aquest l'idoni segons la naturalesa del projecte. L'aportació de coneixement dels professionals d'aquesta àrea garanteixen indubtablement la superació dels problemes no previstos que hi puguin sorgir.

Així doncs queda justificada la viabilitat tècnica d'aquest.

2.8.5 Conclusions

Un cop estudiada la viabilitat del projecte dins el marc temporal, econòmic, legal i tècnic; i havent-se comprovant la seva viabilitat positiva en aquests quatre àmbits, direm que aquest projecte és viable en el domini d'execució descrit en els punts anteriors.

Capítol 3. Disseny

3.1 Introducció i metodologia de disseny utilitzada

E. S. Taylor definia al 1959³:

«Disseny és el procés d'aplicar distintes tècniques i principis amb el propòsit de definir un dispositiu, procés, o sistema, amb els suficients detalls com per a permetre la seva realització física.»

De forma més especialitzada al camp del disseny software, Coad i Yourdon el definien al 1991 com:

«la pràctica de prendre una especificació de comportament observable externament i afegir els detalls necessaris per a la implementació actual del sistema computacional, incloent detalls sobre la interacció dels usuaris, la gestió de tasques i la gestió de dades.»

Amb aquestes dues definicions com a referències principals de l'etapa de disseny, redactarem aquest capítol com una visió específica en la recerca de la solució que compleix amb els requisits analitzats al capítol 2.

L'anàlisi del projecte es basava en la recerca i documentació del *què* del projecte. El disseny és l'etapa de recerca i documentació del *com*.

Podem entendre la fase de disseny a dos nivells: el disseny del sistema i el disseny detallat. Aquest capítol utilitza la metodologia de disseny estructurat, orientat a objectes i components, i principalment es centra en el disseny detallat. No és fins a la secció 3.6 que es fa referència al disseny global del sistema com a detalls finals de disseny.

Es fa ús de notació i diagrames UML per il·lustrar el comportament funcional dissenyat amb l'objectiu de simplificar i facilitar la comprensió dels models. S'ha utilitzat els diagrames d'activitat per representar els flux de treball, les operacions dels components del sistema i el flux de control general. També es fa ús dels diagrames de seqüència per mostrar les interaccions entre objectes i actors del sistema ordenades segons una seqüència temporal. Aquest tipus de diagrama mostra els objectes que es troben a l'escenari i la seqüència de missatges intercanviats entre aquests per a la realització de la funcionalitat descrita per l'escenari. S'inclouen també el diagrama de classes i recursos

³ E. S. TAYLOR, *An Interim Report on Engineering Design*.
Massachusetts Institute of Technology, 1959.

generats per a la implementació i el d'entitat-relació de les bases de dades que hi intervenen.

Finalment a l'apartat 3.6 es comenta la comunicació entre els mòduls.

3.2 Disseny detallat dels mòduls analitzats

Respectant la partició en tres mòduls especialitzats realitzada en la fase d'anàlisi, continuarem ara la fase de disseny de manera estructurada.

Cadascun dels mòduls dissenyats correspon a una divisió horitzontal del sistema global. Aquest tipus de particionament promourà el disseny d'un sistema:

- més fàcil de provar
- més fàcil de mantenir
- amb menys efectes secundaris
- i més fàcil d'ampliar

Una contraprestació d'aquest, que es tracta en detall a la secció 3.6 d'aquest capítol, és que:

- augmenta la comunicació entre els mòduls podent complicar el control global del flux del sistema.

La següent figura il·lustra el particionament horitzontal:

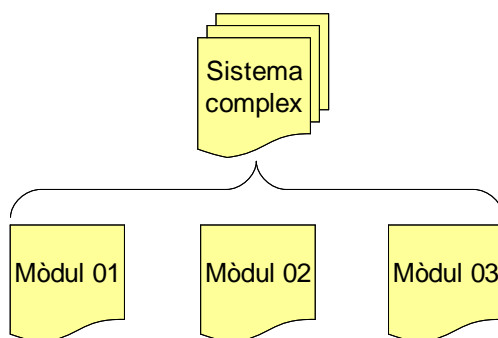


Fig. 3.1 Particionament horitzontal del sistema

A les seccions següents, s'aplica el refinament successiu com a tècnica de disseny *top-down* amb l'objectiu d'aconseguir l'arquitectura del sistema a partir dels requeriments especificats segons l'anterior capítol. Aquesta tècnica tracta de fer divisions verticals del sistema on a cada una d'aquestes, el nivell inferior guanya en especialització. Els nivells superiors s'encarreguen de la lògica de control i presa de decisions mentre que els nivells inferiors s'encarreguen de les tasques d'entrada, càlcul i sortida.

La figura següent il·lustra el particionament vertical resultat d'aplicar la tècnica de disseny *top-down*.

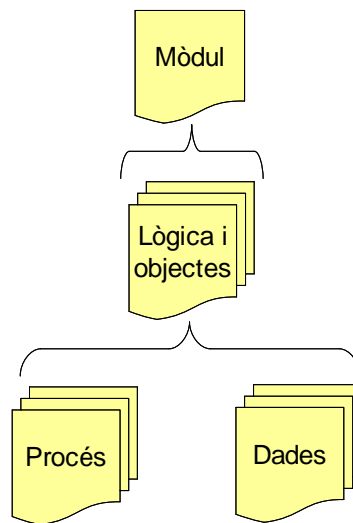


Fig. 3.2 Particionament vertical del sistema

Aquesta tècnica de disseny s'aplicarà de forma independent a cada un dels tres mòduls que componen el projecte. No obstant, i de forma comuna a tots tres, les particions verticals, resultants del disseny específic, permeten una abstracció en capes funcionals que es comenta al capítol 4 de la memòria on es parla del patró arquitectònic sobre el que s'ha executat el desenvolupament.

Com hem introduït abans, la metodologia de disseny utilitzada és l'orientació a objectes i això serà visible tant en la redacció dels següents apartats com en els diagrames que els il·lustren.

3.3 Mòdul 01 – Acreditació d'usuaris al sistema

Aquest mòdul, tal i com el seu nom indica, ha d'integrar la funcionalitat d'acreditar els usuaris al sistema.

El paràgraf següent explica el comportament funcional desitjat:

«L'usuari demana l'accés a la plataforma i el servidor rep aquesta petició. S'estableix un canal segur de comunicacions (https/ssl) i el servidor demana el certificat d'usuari. L'usuari entrega el certificat d'usuari i el servidor comprova la validació d'aquest. Si tot és correcte, informa a l'usuari del resultat de la comprovació, crea la sessió d'usuari i anota l'esdeveniment. A partir d'aquest moment, l'usuari ja pot iniciar la gestió i sol·licitar els serveis de la plataforma. Si el certificat no és vàlid, s'informa a l'usuari de la situació i no es crea la sessió d'aquest. Igualment s'anota l'esdeveniment.»

Veiem que es tracta d'un sistema de comunicació on interaccionen, en un principi, dos actors, tal i com s'analitzava al punt 2.4.5: l'usuari i el servidor. Fent una abstracció en emissor, receptor i missatges, i utilitzant un diagrama de seqüència per a modelar la comunicació entre aquests, el representàrem de la següent manera:

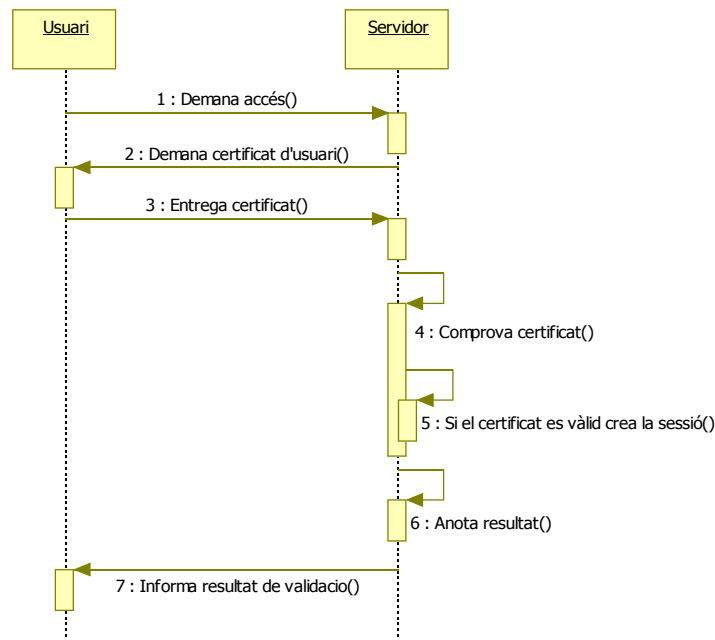


Fig. 3.3 Diagrama de seqüència que reflecteix la comunicació entre Usuari i Servidor

Estudiant la casuística d'aquesta comunicació, veiem amb claredat que en necessitarem els següents recursos:

En quant a pantalles d'interfície d'usuari:

- La de *benvinguda*: serà l'encarregada de mostrar una breu descripció de la plataforma i permetrà l'accés contra a aquesta. Aquesta pantalla la trobarem reflectida en els diagrames de les seccions següents amb el nom de *doLogin*. Consultar figura 8.2.1 de l'annex 8.2.
- La de *petició de certificat*: s'encarregarà de demanar a l'usuari el certificat digital que vol utilitzar per a iniciar la sessió. Al tractar-se d'una implementació web, aquesta pantalla és inherent als navegadors i no és necessari la seva construcció però fem menció ja que intervindrà en el procés. Consultar figura 8.2.2 de l'annex 8.2.
- La de *resultat* del procés de validació d'identitat. Se n'encarregarà de fer arribar a l'usuari el resultat del procés de validació del seu certificat. Comprovarà internament una bateria de tests per a permetre l'entrada a la plataforma. Es reflecteix als diagrames amb el nom de *doCertifiedLogin*. Consultar figura 8.2.4 de l'annex 8.2.

Baixant nivells dins la fase de disseny i arribant als nodes fulla *procés* i *dades* representats en la figura 3.2, ens és pràctic la definició d'objectes per distribuir les funcionalitats específiques de control i tractament de la informació. Modelarem llavors les següents classes:

- *L'usuari*: emmagatzemarà les dades necessàries de l'usuari acreditat.
- *Sessió*: inherent al servidor web. S'encarregarà de guardar les dades i estats de l'aplicatiu.
- *Logger*: l'apuntador d'esdeveniments. Té com atributs les dades de l'esdeveniment

produït.

En quant a la comunicació amb els altres mòduls i entre objectes i les bases de dades relacionades es tractarà en profunditat al punt 3.6.

3.3.1 Diagrames d'activitat i seqüència

A continuació en aquest apartat, utilitzarem els diagrames UML per a representar el comportament funcional dissenyat per a la implementació del mòdul que ara tractem:

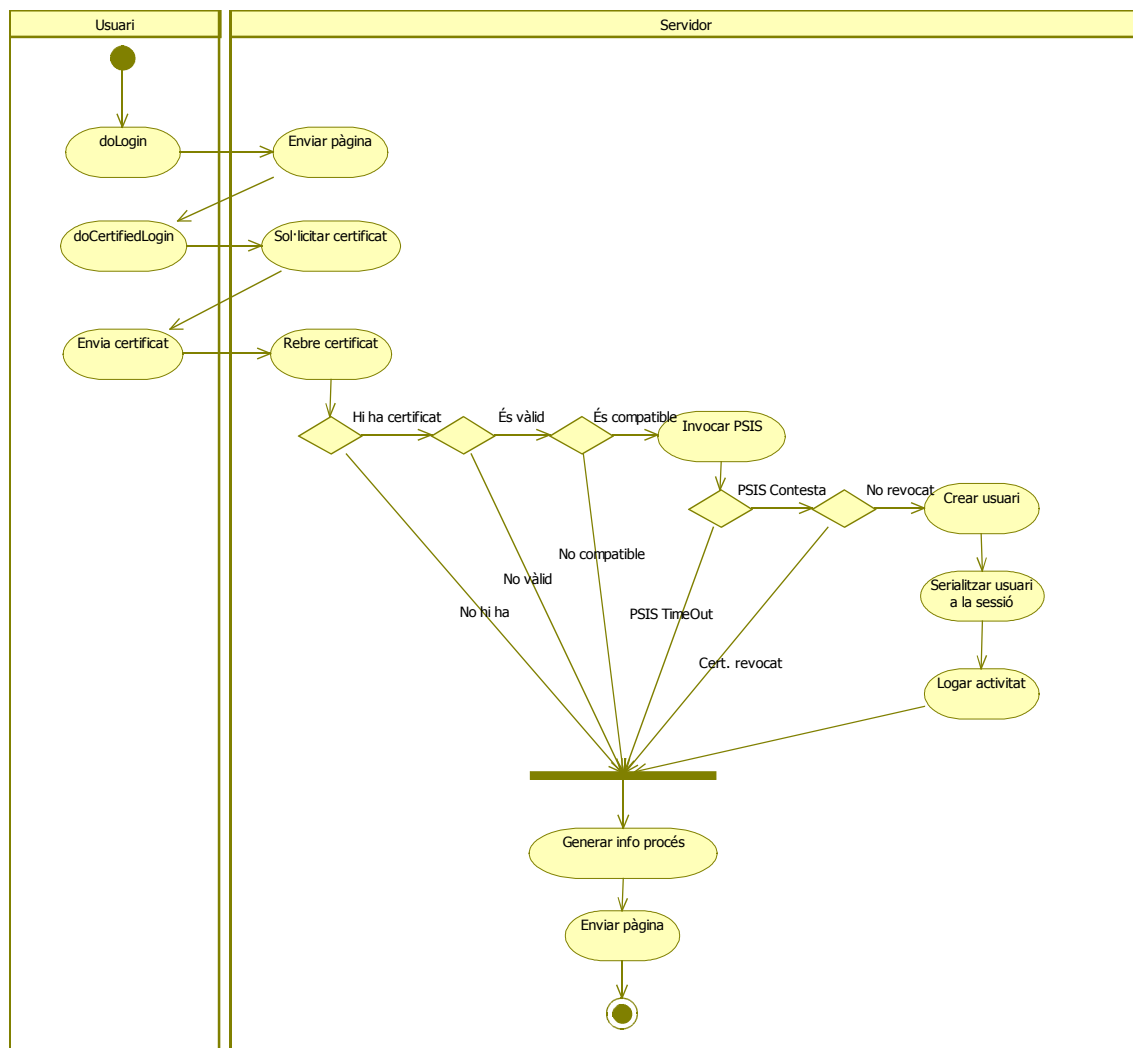


Fig. 3.3 Diagrama UML d'activitat del Mòdul 01

El diagrama d'activitat anterior modela la comunicació entre els dos actors. S'ha particionat el diagrama en dos blocs, l'usuari i el servidor, amb l'objectiu d'agrupar el comportament intern de cadascun d'ells. D'una banda veiem com queda representat la col·laboració de l'actor usuari, segons s'analitzava al diagrama de casos d'ús de la figura 2.1. D'altra banda observem que el diagrama defineix el flux de control intern per comprovar la validació del certificat d'usuari entregat. Aquest procés aplica una bateria de controls sobre aquest abans de crear l'usuari. Comprovem els punts següents i en aquest

ordre:

- El certificat ha arribat al servidor.
- Que sigui vàlid i no hagi caducat.
- Sigui compatible amb l'aplicació (emès per una de les CAs reconegudes).
- La invocació a PSIS satisfà que no es un certificat revocat.

Si el procés de validació acaba amb èxit es crea un objecte *usuari* que es serialitza a la sessió.

Per finalitzar el procés, es genera la pantalla de resultat informant del resultat de l'operació.

A continuació, el diagrama de seqüència organitza de forma seqüencial el flux de missatges entre els diferents objectes resultats del disseny d'aquest mòdul. A la part superior d'aquest es distribueixen els actors i objectes del sistema, de color groc i els recursos d'entrada-sortida, de color blau:

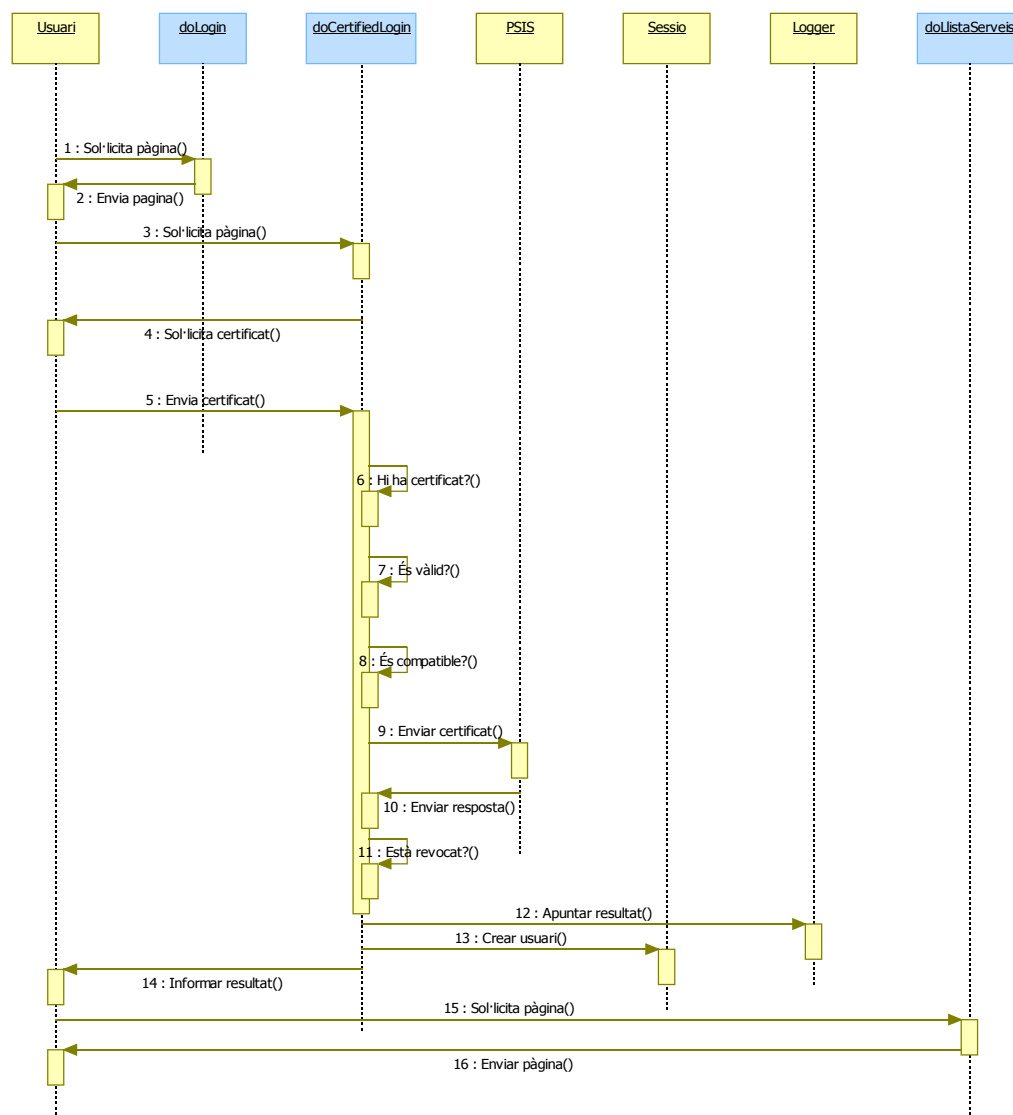


Fig. 3.4 Diagrama UML de seqüència del Mòdul 01

3.3.2 Diagrama de classes

Les classes dissenyades per a la realització dels processos interns d'aquest mòdul són les que apareixen en el següent diagrama:

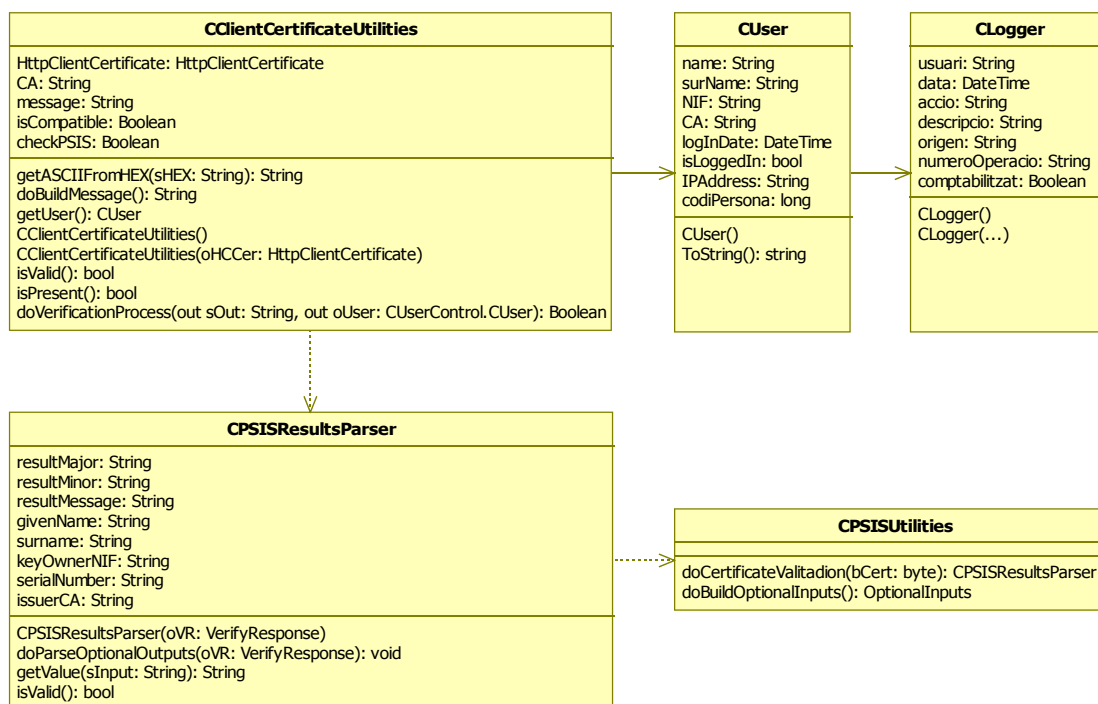


Fig. 3.5 Diagrama de classes del Mòdul 01

Descrivint breument la utilitat i funcionalitat de cada una de les classes dins del mòdul que ara tractem, tenim:

- **CUser**: recull les dades de l'usuari que inicia la sessió. S'instancia un objecte d'aquesta classe quan el procés de validació de certificat acaba amb èxit. Aquest objecte inicialitza els seus atributs amb les dades derivades del certificat digital presentat. Un cop instanciat, es serialitza i es guarda a la sessió web per que la resta de mòduls tinguin constància de l'usuari amb el que treballen.
- **CLogger**: aquesta classe és comuna a tota la plataforma doncs està dissenyada per guardar les dades d'un determinat esdeveniment quan aquest es produeix. En el mòdul que ara tractem, els esdeveniments possibles són els d'intent d'inici de sessió frustrats o no.
- **CClientCertificateUtilities**: recull el certificat d'usuari i inclou mètodes útils per a la gestió de les dades que aquest conté. Un objecte d'aquesta classe es que s'encarrega de fer el procés de validació del certificat en el moment de la invocació del mètode doVerificationProcess().
- **CPSISUtilities**: Abans hem introduït la col·laboració d'un agent extern en el procés de validació del certificat: l'actor PSIS. Aquesta classe fa d'interfície per invocar els serveis de validació del webservice de la plataforma PSIS. El mètode encarregat de l'execució d'aquest procés és doCertificateValidation que rep com a paràmetre un array de bytes amb el certificat d'usuari.
- **CPSISResultsParser**: Els resultats retornats pel webservice de la plataforma PSIS

no són directament consultables. Aquesta classe agrupa els mètodes necessaris que fan possible un accés a més alt nivell a aquests resultats. Entre d'altres, es calcula si el certificat es valid o no invocant al mètode `isValid()`, i es retorna, segons els atributs `givenName` i `surname`, el nom i cognoms de forma estructurada.

3.3.3 Entitat-relació

A diferència dels següents dos mòduls, aquest primer no requereix l'accés ni la construcció de cap base de dades específica. Degut a això, diagrama d'entitat-relació només reflexa una única taula: `tblEventLog`.

tblEventLog	
🔑	Usuari: VARCHAR(40)
🔑	Data: DATETIME
🔹	Accio: VARCHAR(15)
🔹	Descripcio: TEXT
🔹	Origen: VARCHAR(250)
🔹	NumeroOperacio: VARCHAR(11)

Fig. 3.6 Diagrama d'entitat-relació del Mòdul 01

Aquesta taula es troba ubicada dins de la base de dades del projecte, *PTSA*, i cohabita amb dues taules més: *tblComptadors* i *tblUploads*. És una base de dades molt senzilla creada amb l'objectiu únic del control d'esdeveniments i altres funcionalitats comunes que en detallarem més endavant i es detallarà al punt final d'aquest capítol.

tblEventLog emmagatzema registres que provenen d'objectes *CLogger*. En el cas particular d'aquest mòdul, s'envien insercions contra aquesta taula de registres com els següents:

Usuari	Data	Accio	Descripcio	Origen	Num...
46608461L	2009-05-20 ...	LOGIN	name=ANTONIO, surName=BUIZA GRILLO, NIF=46608461L, CA=EC...	/aspnet/PTSA/doCertifiedLogin.cs.aspx	NULL
Anònim	2009-05-20 ...	LOGIN_TRY	IP=192.168.61.5, Error en test: Validació certificat contra PSIS/CATCert	/aspnet/PTSA/doCertifiedLogin.cs.aspx	NULL

Fig. 3.7 Detall de registres inserits a *tblEventLog*

La figura 3.7 mostra els dos tipus d'esdeveniments enregistrats per aquest mòdul: l'esdeveniment *LOGIN*, que succeeix quan el certificat d'usuari passa el procés de validació i inclou les dades descriptives de la persona, i l'esdeveniment *LOGIN_TRY* que succeeix quan el procés de validació no es completa i llavors l'acompanyen les dades de l'error.

3.4 Mòdul 02 – Registre d'entrada telemàtic

El segon mòdul de la divisió horitzontal és el Registre d'entrada telemàtic. S'han de satisfer els objectius redactats a la fase d'anàlisi.

L'execució d'aquest mòdul comença un cop l'usuari ha iniciat sessió correctament a la plataforma, identificant-se amb el seu certificat digital, ha vist el llistat de serveis oferts per aquesta i ha triat el servei d'*Alta en registre d'entrada*.

La funcionalitat desitjada descrita en llenguatge natural seria la següent:

«L'usuari sol·licita el servei i el servidor comprova a la base de dades d'interessats si existeix la persona amb el DNI acreditat segons el certificat digital. Si no troba l'interessat, el servidor informa d'aquesta situació i envia una pantalla amb el formulari d'alta d'interessat. Els camps DNI i NOM d'aquest formulari resten bloquejats per evitar edicions. Quan l'usuari emplena aquest formulari i l'envia al servidor, aquest crea el nou interessat i torna a la pantalla de serveis. És llavors que l'usuari torna a sol·licitar el registre. Aquesta segona vegada el servidor ja el troba com interessat i llavors envia el formulari de registre. Aquest recull els camps a emplenar per al processament del registre. Aquest formulari serà semblant a l'actual en paper. L'usuari emplena el formulari amb les dades que vol registrar i l'envia al servidor. El servidor rep les dades i obre una operació transaccional. Dins aquesta operació es calcula la nova clau pel nou document de registre, es realitza una inserció a la base de dades i es genera un número d'operació. Si tot el procés acaba amb èxit s'informa a l'usuari amb la clau de registre inserit i el número d'operació. Des d'aquesta mateixa pantalla es convida a l'usuari a inserir un únic arxiu adjunt, visualitzar el registre o bé tornar al llistat de serveis. Si l'usuari introdueix un arxiu i l'envia es visualitzarà una barra de progrés amb el detall de càrrega de la tramesa de l'arxiu. Un cop aquest arriba al servidor, es copiarà al repositori d'arxius afegint la clau de registre com a prefixe del nom i es calcularà un codi criptogràfic HASH-MD5. El servidor també apuntarà a la base de dades el lligam entre el registre efectuat i l'arxiu enviat. Posteriorment el servidor informarà l'usuari de l'estat de la recepció i del codi criptogràfic calculat. Tota l'activitat durant el procés quedarà registrada.»

De la lectura de la descripció funcional, veiem que hi ha una comunicació entre l'usuari i el servidor i, en aquest cas, es distingeixen tres processos de gestió independents:

- l'alta de l'usuari com a interessat
- l'alta del document registre
- la tramesa de l'arxiu adjunt

També i de forma casual cada un d'aquests processos separables requereix l'accés a una base de dades determinada, respectivament:

- Base de dades: *Nucli*
- Base de dades: *Registre*
- Base de dades: *PTSA*

Aquest nou particionament funcional es reflecteix als diagrames de les seccions següents on també s'amplien els detalls.

També necessitarem les següents pantalles com a interfície d'usuari:

- *Alta d'interessat*: es tracta d'un formulari web que permet la recollida de les dades complementàries per a processar l'operació d'alta del nou interessat. El Nom i el DNI no seran camps editables per evitar que un usuari s'enregistri amb dades

falses. Consultar figura 8.2.6 de l'annex 8.2.

- *Alta document*: també és un formulari web que recollirà les dades necessàries per a processar l'operació d'alta del nou document de registre. Consultar figura 8.2.8 de l'annex 8.2.
- *Informativa*: informarà a l'usuari del resultat de l'operació i permetrà adjuntar un arxiu al document registrat. Consultar figura 8.2.9 de l'annex 8.2.
- *Tractament de l'arxiu adjunt*: tindrà els controls per fer la selecció de l'arxiu i mostrarà la barra de progrés de la tramesa d'aquest. Finalment informarà del resultat de la tramesa. Consultar figura 8.2.10 de l'annex 8.2.

En quant a classes, aquest mòdul també utilitza les introduïdes a la secció del mòdul anterior, doncs es tractaven de classes genèriques, pròpies de la plataforma. Però, a part d'aquestes, en necessitarem d'altres més específiques:

- *Document*: serà l'abstracció de les dades que construeixen un document de registre i en tindrà correspondència directa amb la base de dades relacionada.
- *Persona*: abstracció de dades personals. Un interessat en serà un objecte d'aquesta classe.
- *Domicili*: col·lecció de *Territoris* relacionats amb la *Persona* seleccionada.
- *Territori*: dades estructurades de dades territorials del municipi.
- *UploadFile*: recull de mètodes en relació als processos a implementar per a validar els objectius relacionats amb la càrrega i gestió de l'arxiu adjunt.

3.4.1 Diagrames d'activitat i seqüència

Si intentem representar tota la funcionalitat que ha d'implementar aquest mòdul en un únic diagrama d'activitat, com a resultat ens trobaríem un model complex i difícil d'entendre amb moltes activitats interrelacionades.

Aprofitant la detecció dels tres processos funcionals independents, hem dissenyat un primer diagrama que modela la lògica i el control de flux principal del mòdul. Aquest es complementa amb un segon on es representa la lògica i el flux detallat per cada un dels processos assenyalats. A tots dos, s'han utilitzat colors diferents per remarcar l'aïllament funcional i facilitar la lectura d'aquests.

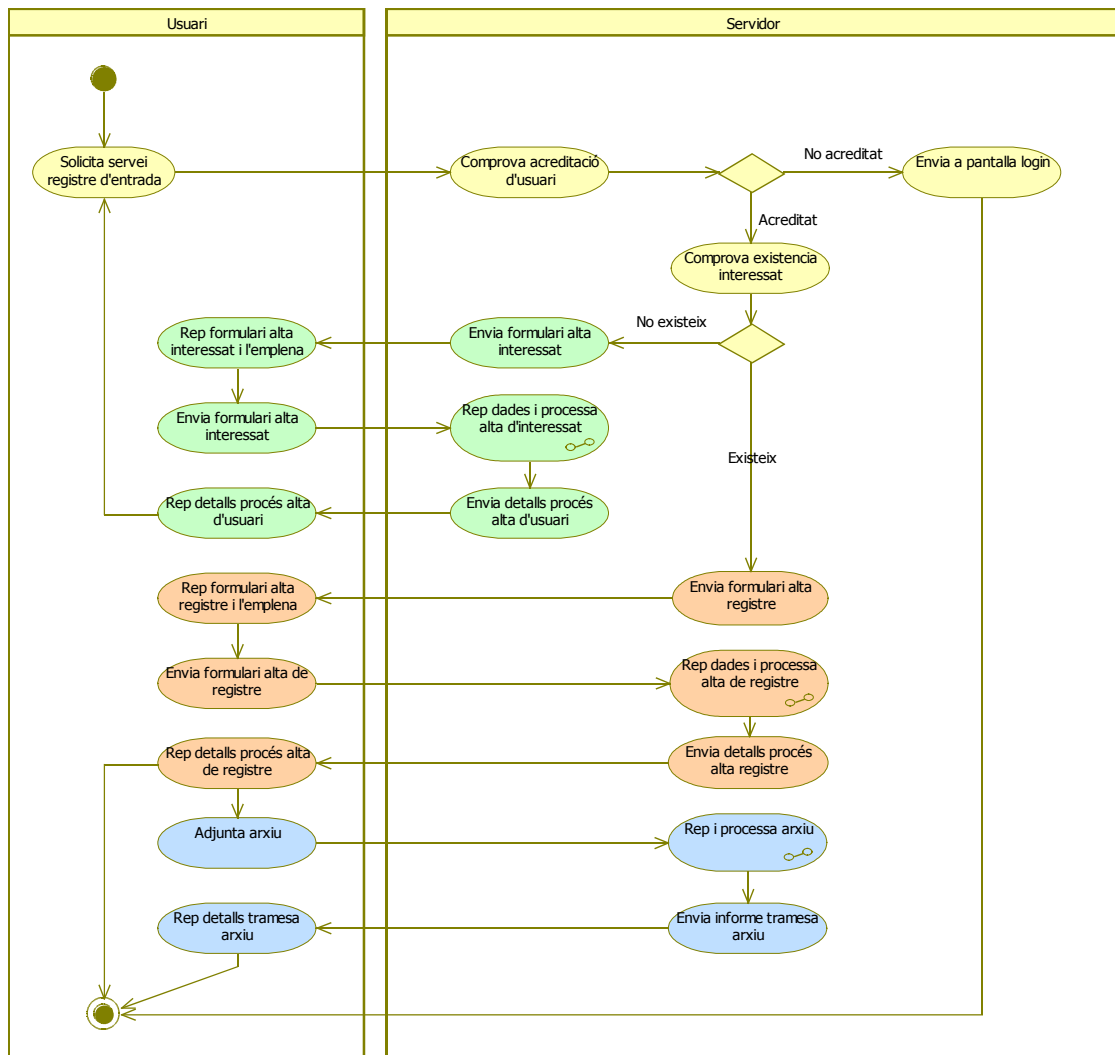


Fig. 3.8 Diagrama d'activitat del Mòdul 02

En la comprovació d'objectius veiem que el diagrama comprova correctament l'objectiu OBJ_02_02 segons l'activitat *Comprova existència interessat*. En el cas que no es trobi l'usuari a la base d'interessats el flux es deriva cap el subprocés de sol·licitud d'alta d'interessat –de color verd–. En el cas que existeixi, l'execució continua segons el subprocés de sol·licitud de les dades del nou document –de color taronja–. Finalment –en color blau– es representa el subprocés que modela l'objectiu OBJ_02_08 de la utilitat d'adjuntar arxIU al document registrat.

La figura següent recull el tres subdiagrames d'activitat que modelen el comportament en detall dels subprocessos assenyalats:

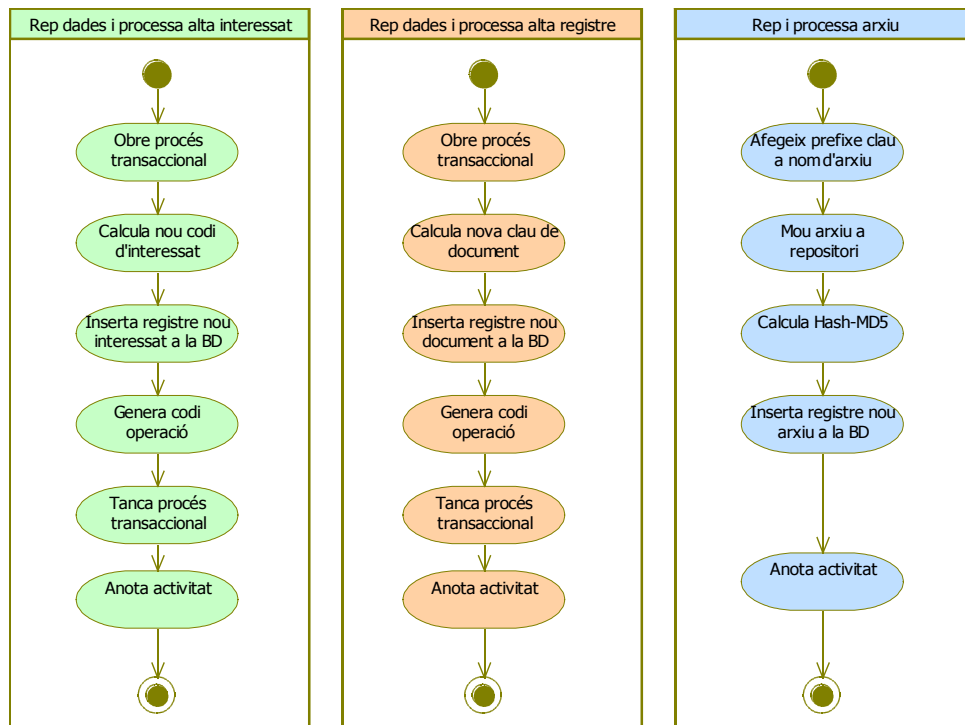


Fig. 3.9 Subdiagrama d'activitat del Mòdul 02

El primer subdiagrama –de color verd– fa referència al subprocés d'alta d'interessat. Les activitats i flux que modela comproven els requeriments derivats de l'objectiu principal OBJ_02_02.

Igual que abans, el segon –de color taronja– modela el subprocés de rebre les dades i processar l'alta d'un nou document de registre.

Tots dos processos fan servir una operació transaccional per calcular la clau primària, inserir el nou registre i generar el codi d'operació. Només que el primer insereix el nou interessat i el segon insereix el nou document. El comportament dissenyat valida els objectius OBJ_02_04, OBJ_02_05 i OBJ_02_06.

En referència a l'últim subdiagrama –de color blau–, veiem la especialització d'aquest segons el processament de l'arxiu adjunt. El model valida els objectius OBJ_02_10, OBJ_02_11 i OBJ_02_13.

A continuació segueixen els diagrames de seqüència. Seguint l'estratègia d'abans, no s'ha dissenyat un únic diagrama doncs la complexitat funcional del mòdul no seria llegible d'una manera senzilla.

Començarem introduint un subdiagrama on es representen els actors i la seqüència d'execució que resol el problema de saber si l'usuari s'ha acreditat o no. Aquest subprocés és comú a pràcticament tots els processos del projecte i es farà referència més endavant.

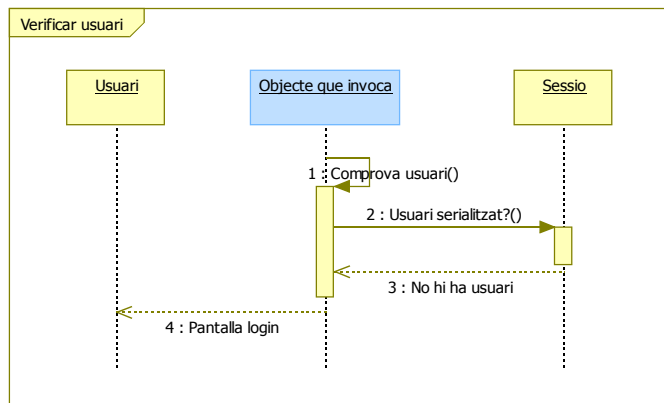


Fig. 3.10 Diagrama de seqüència de comprovació serialització de l'usuari

L'actor *Objecte que invoca* modela a qualsevol procés que inclou una invocació a aquest procés de *verificar usuari*. La funcionalitat dissenyada és senzilla: es comprova si existeix l'usuari serialitzat a la sessió web i si no el trobem, mostrem la pantalla inicial de *login*, convidant l'usuari a acreditar-se correctament. En cas que l'usuari sí el trobem serialitzat a la sessió, significa que ja es va acreditar correctament, és llavors quan retornem a l'execució de l'*objecte que invoca*.

El següent diagrama mostra les interaccions i missatges entre els actors i objectes modelant el comportament funcional quan un usuari no es troba com a interessat a la base de dades –objectiu OBJ_02_02–:

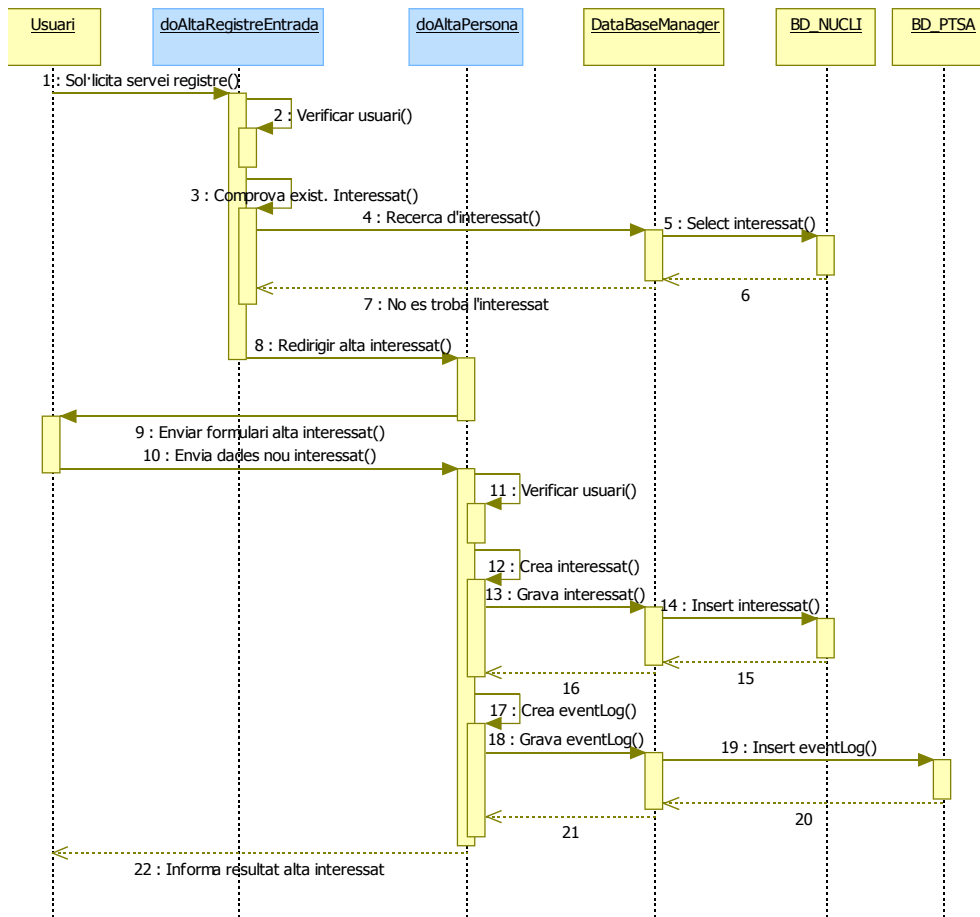


Fig. 3.11 Diagrama de seqüència quan no existeix l'interessat

Per primera vegada en aquest tipus de diagrames, es fa referència a l'objecte *DatabaseManager*. Aquest s'encarrega d'actuar com a interfície entre la capa d'objectes de l'aplicació i les bases de dades. S'explicarà amb més detall a a la secció 4.5.

Aquí el flux de procés canvia del *doAltaRegistreEntrada* a *doAltaPersona* al no trobar l'interessat a la base de dades.

Un altre detall a tenir en compte a la fase de desenvolupament és que les interaccions de la 13 a la 21 s'han d'executar de forma transaccional, tal i com s'ha comentat en diferents ocasions.

A continuació, el diagrama que modela les interaccions necessàries per processar l'alta d'un nou document de registre:

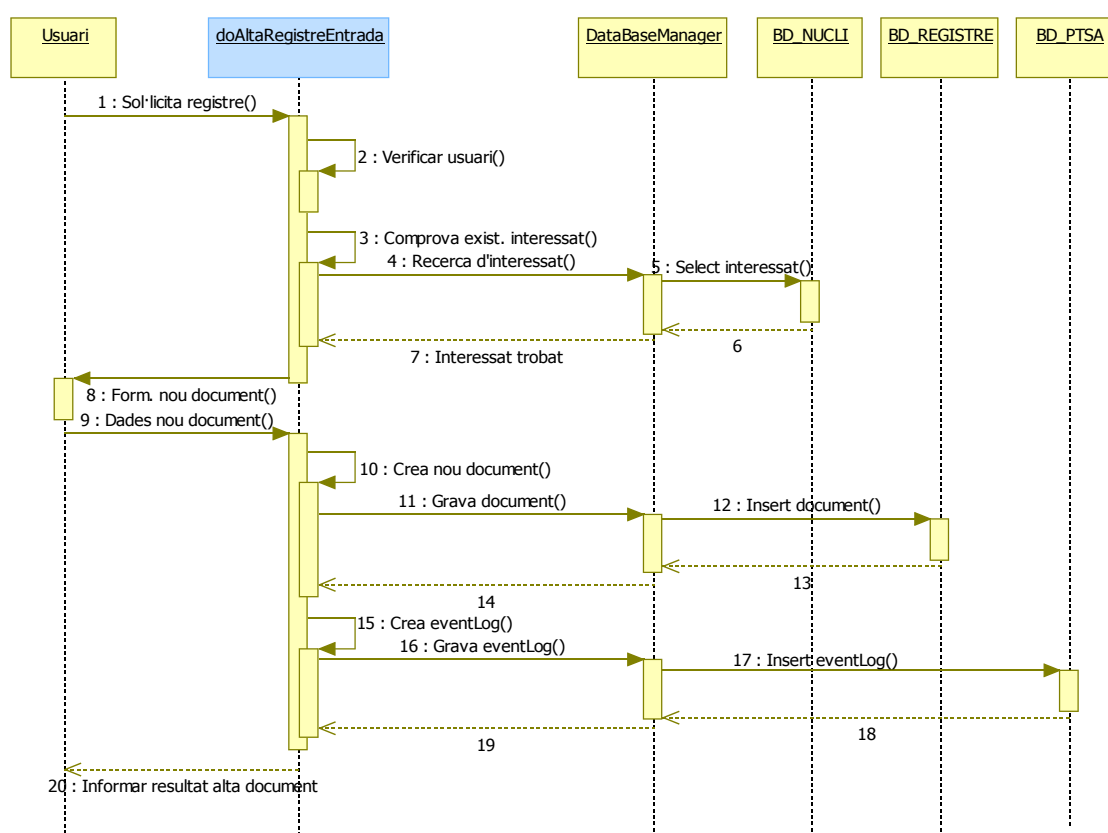


Fig. 3.12 Diagrama de seqüència del procés d'alta de nou document

El flux principal d'execució es resumeix en dos processos bàsics. El primer: comprovar que l'usuari sí és un interessat. El segon: crear un nou document i guardar-ho a la base de dades.

Novament remarcar en el desenvolupament que les interaccions de la 11 a la 19 s'han de processar dintre d'una operació transaccional.

A l'últim diagrama de seqüència es representa el comportament funcional dissenyat per a satisfer l'objectiu de permetre a l'usuari adjuntar un arxiu:

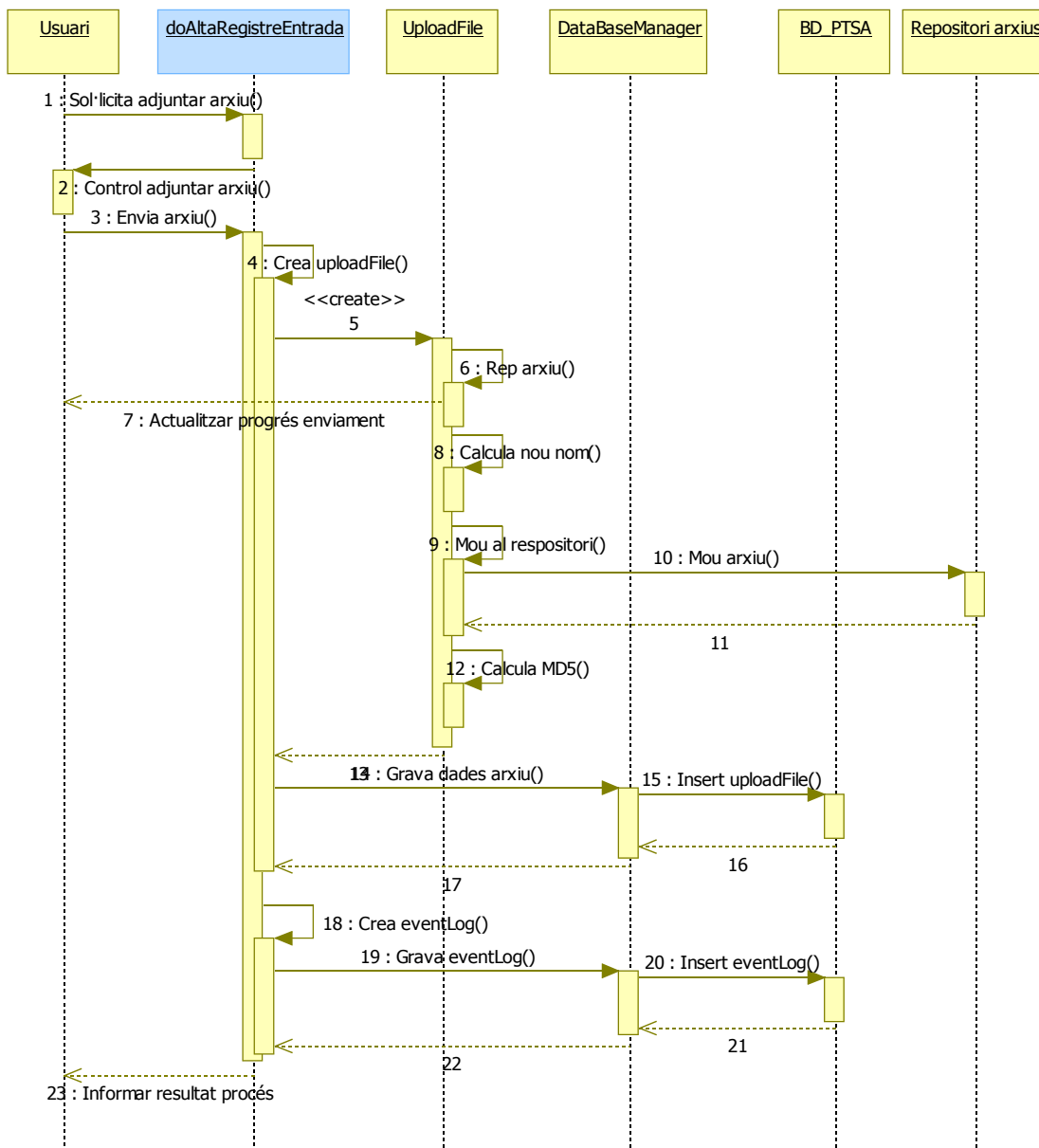


Fig. 3.13 Diagrama de seqüència del procés d'adjuntar un arxiu

El flux principal d'execució succeeix al procés *doAltaRegistreEntrada* i es construeix un objecte *uploadFile* que s'encarrega de la gestió de la tramesa i processament de l'arxiu rebut. Interactua amb aquest objecte el repositori d'arxius.

3.4.2 Diagrama de classes

A continuació el diagrama de classes dissenyat per implementar les funcionalitats requerides:

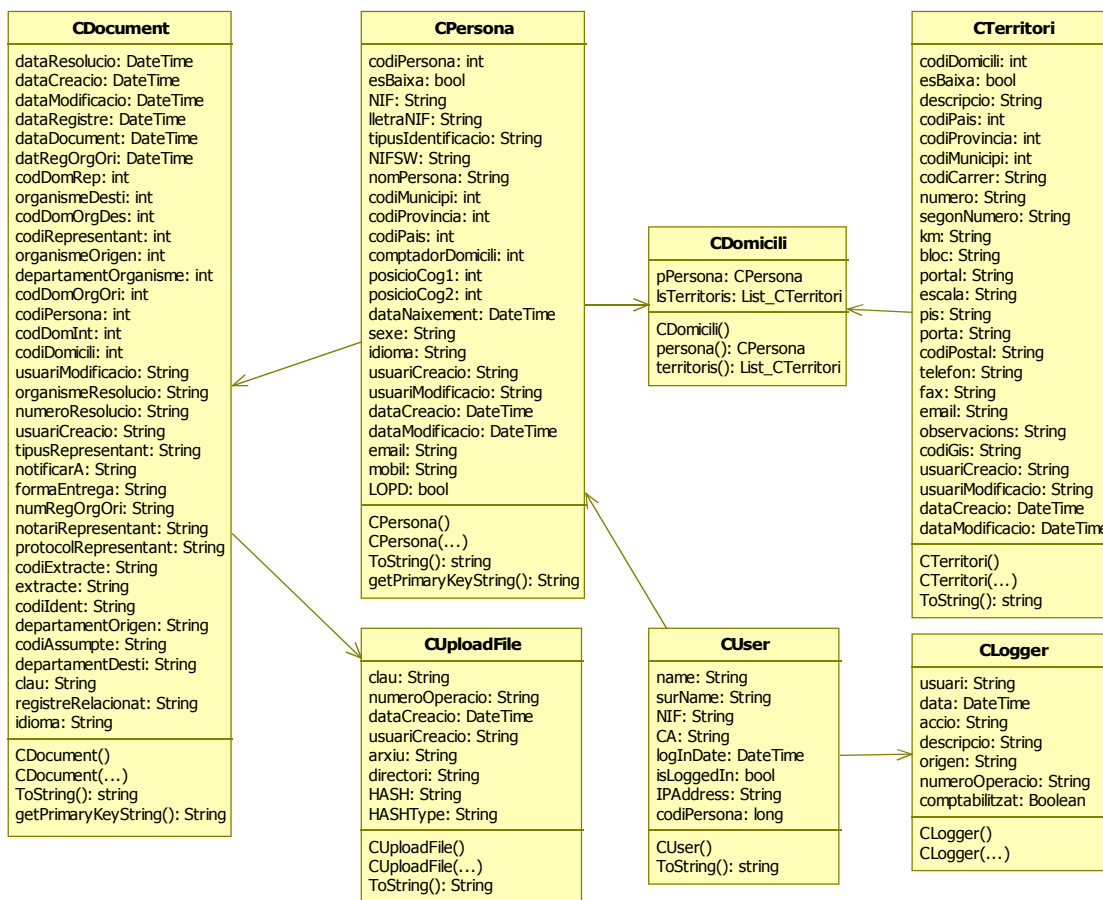


Fig 3.14 Diagrama de classes utilitzades en el Mòdul 02

Comprovem les relacions i dependència entre aquestes segons les fletxes existents.

Ampliant els detalls de cada classe:

- CDocument: a la interacció número 10 del diagrama 3.12 es crea un objecte d'aquest tipus. S'inicialitza amb les dades que l'usuari declara segons el formulari emplenat i posteriorment s'envia al DatabaseManager per tal que aquest faci l'emmagatzemament a la base de dades específica.
- CPersona: la interacció 7 del diagrama 3.12, representa el paràmetre de retorn del objecte DatabaseManager. Aquest retorna un objecte CPersona que representa l'interessat que s'ha trobat a la base de dades segons la correspondència del DNI.
- CDomicili: relaciona les classes CPersona i CTerritori. Una persona pot estar relacionada amb varis territoris i un territori pot estar relacionada amb vàries persones. És per aquest motiu que un objecte d'aquesta classe té com atribut una col·lecció de territoris i una persona. Es fa servir de forma implícita en la construcció del formulari d'alta de nou document en la interacció 8 del diagrama 3.12.
- CTerritori: abstracció de dades territorials. Comentat amb anterioritat.
- CUser i CLogger: classes amb funcionalitat general introduïdes anteriorment.
- CUploadFile: Abstracció de les dades relacionades amb el procés de càrrega de l'arxiu i relació d'aquest amb el document registrat. Les interaccions 5 i 14 del diagrama 3.13 fan, de forma respectiva, la creació de l'objecte i l'envien al DatabaseManager pel seu emmagatzemament.

3.4.3 Entitat-relació

Aquest mòdul requereix l'accés contra tres bases de dades: *RES DAR*: la base de dades de registre; *PTSA*: la del projecte; i *NUCLI*: la base de dades que emmagatzema dades compartides per diferents aplicacions corporatives.

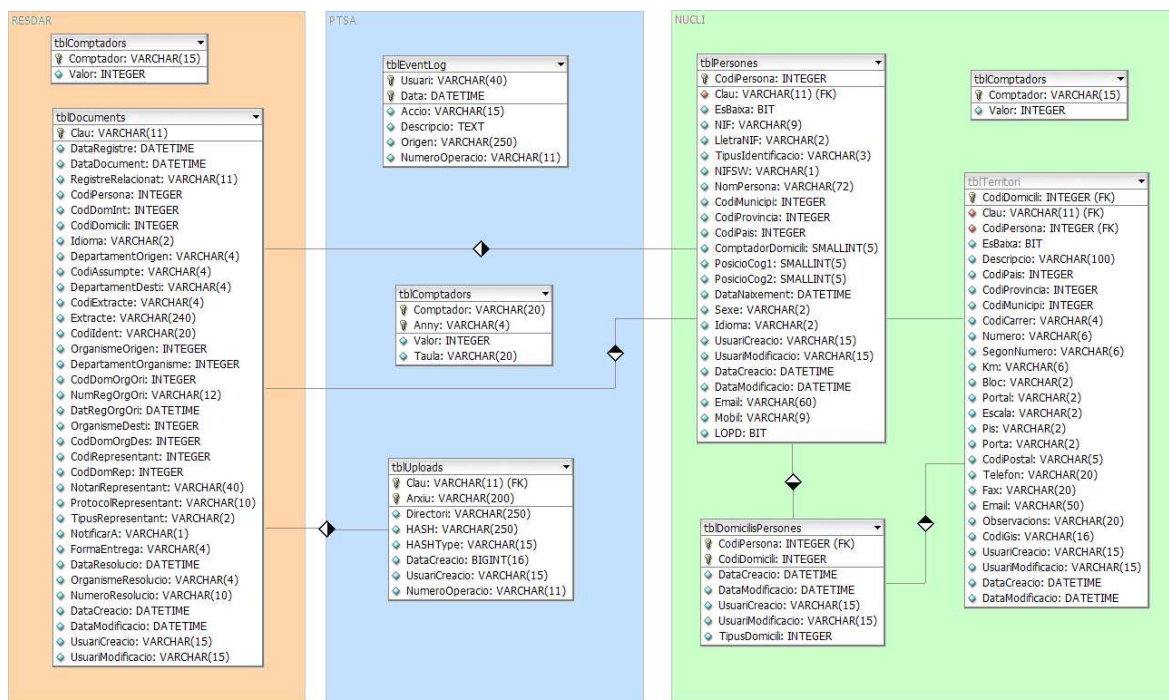


Fig. 3.15 Diagrama d'entitat-relació amb les tres bases de dades utilitzades al Mòdul 02

Els colors utilitzats de fons a les taules que figuren al diagrama serveix d'enllaç conceptual entre les bases de dades i els diagrames d'activitat de les figures 3.8 i 3.9. També agrupen les taules segons la base de dades a la que pertanyen.

Les taules representades al diagrama es troben implicades en el procés de registre de la manera següent:

- *RES DAR.tbiComptadors*: és la taula que emmagatzema els comptadors de l'aplicatiu de registre. Es treballa amb ella per generar el codi d'un nou registre.
- *RES DAR.tbiDocuments*: és la taula encarregada de guardar els documents de registre d'entrada i sortida. La clau primària és el camp clau i es calcula a fent servir una lletra (E: entrada; S: sortida) més 4 dígits de l'any en curs i un comptador de 6 dígits més.
- *PTSA.tbiEventLog*: guarda les dades dels esdeveniments i operacions produïts a la plataforma. Explicada anteriorment en la secció 3.3.3.
- *PTSA.tbiComptadors*: emmagatzema el comptador d'operacions realitzades per la plataforma.
- *PTSA.tbiUploads*: s'encarrega de guardar les dades de l'arxiu adjunt i la ubicació en el repositori d'arxius. Està relacionada amb *RES DAR.TBLDocuments*.
- *NUCLI.tbiPersones*: recull de dades personals. És on es comproven o s'insereixen els interessats. Es consulten pel camp *NIF* que és la dada a crear des del certificat digital.

- *NUCLI.tbIComptadors*: igual que en les altres bases de dades, emmagatzema els comptadors de persones o territoris, entre d'altres.
- *NUCLI.tbIDomicilisPersones*: taula que implementa la relació de varis a varis entre les taules *tbIPersones* i *tbITerritori*. Gràcies a aquesta sabem, per exemple els domicilis d'un interessat.
- *NUCLI.tbITerritori*: recull de dades territorials. D'aquesta taula es recupera el codi i descripció de l'adreça utilitzada en el procés de registre.

3.5 Mòdul 03 – Procediments informatius

Finalment aquest tercer mòdul el componen un conjunt de serveis de consulta contra diferents bases de dades corporatives. Com ja hem comentat en d'altres apartats, aquests serveis no alteren les dades. L'única base de dades que incrementarà el seu número de registres serà la destinada al control de la plataforma on s'anotará l'activitat de l'usuari amb l'objectiu de controlar la producció a efectes estadístics. Igual que a la resta de processos dissenyats, s'ha de satisfer el conjunt d'objectius redactats a la fase d'anàlisi.

Idènticament al mòdul 02, l'execució d'aquest comença un cop l'usuari ha iniciat sessió correctament a la plataforma, identificant-se amb el seu certificat digital, i vist el llistat de serveis oferts, tria qualsevol altre diferent de l'*Alta en registre d'entrada*.

La funcionalitat desitjada descrita en llenguatge natural seria la següent:

«L'usuari sol·licita el servei i el servidor intenta resoldre la correspondència DNI-Interessat, DNI-Habitant o bé DNI-Contribuent, en funció del servei sol·licitat. Si es troba la correspondència es continua executant el servei sol·licitat, si no, informa d'aquesta situació segons una pantalla específica. L'execució de tots els serveis és molt semblant: el servidor consulta contra la base de dades específica i recupera la informació demanada. Després la formateja en una plantilla HTML que acaba enviant, de forma integrada amb el portal ripollet.cat, a l'interessat. Aquesta plantilla integra els botons amb les funcionalitats que permeten veure la informació a pantalla completa, la visualització o descàrrega d'un arxiu PDF signat electrònicament o bé tornar al llistat de serveis. Quan la consulta retorna més d'un registre, el servidor mostra una taula de selecció de registre i, després que l'interessat en triï un, les dades d'aquest mateix.»

En aquest cas, la complexitat funcional es troba al voltant de la recerca de les dades contra les diferents bases de dades existents. El motiu és la fragmentació i descentralització de les dades segons aplicacions, bases de dades i proveïdors de serveis independents.

No obstant, aquest mòdul no queda exempt de pantalles d'interfície amb l'usuari:

- *Selecció de registre*: mostrarà una taula de registres vinculats amb l'usuari acreditat i permetrà l'accés a la informació amb detall de cada un d'ells.
- *Volant o justificant*: mostra les dades amb detall del registre seleccionat.
- *Versió HTML*: mostra les mateixes dades que la pantalla anterior però de forma independent als layers del portal ripollet.cat.
- *Versió PDF*: en realitat no és una pantalla però com s'ha de maquetar igualment la

tractarem com a tal. És la informació que es genera dinàmicament per generar l'arxiu PDF. Aquest inclou una signatura digital de l'Ajuntament de Ripollet.

Un detall funcional d'aquestes pantalles és que al mostrar informació protegida s'haurà d'apuntar un esdeveniment de consulta a la base de dades de la plataforma. Tant les pantalles de selecció de registre com les de volants o justificants, hauran de portar el codi d'operació imprès.

En quant a classes: la diversificació de bases de dades i taules a consultar guarda una relació de proporcionalitat directa amb el número de classes a definir. Evitem fer una introducció doncs aquestes es detallaran al punt 3.4.2.

3.5.1 Diagrames d'activitat i seqüència

La funcionalitat del el mòdul 03 queda recollida en el següent diagrama:

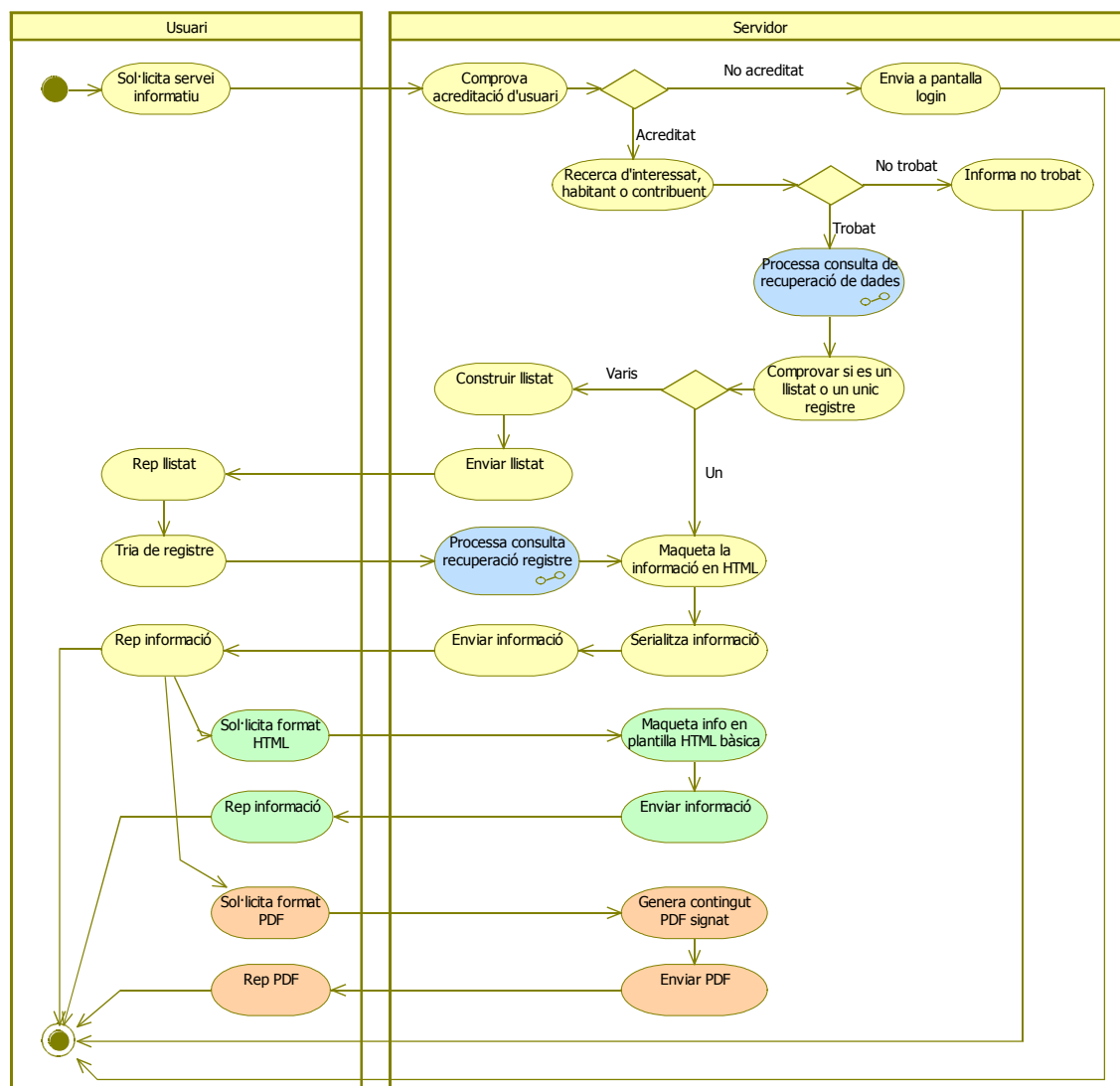


Fig. 3.16 Diagrama d'activitats del Mòdul 03

La lògica principal del procés comprova si l'usuari que demana el servei està acreditat o no i si es troba correspondència d'aquest a la base de dades del servei sol·licitat.

Les dues activitats colorejades en blau representen els processos de consulta contra les diferents bases de dades, segons el servei sol·licitat. D'aquesta manera tindrem la següent correspondència entre serveis i bases de dades:

- Consulta de registre d'entrada-sortida → Registre
- Justificant empadronament / convivència → Padró Mpal. Habitants
- Consulta d'expedients → Expedients
- Consulta de pagaments → Ingressos

Al punt 3.5.3 s'il·lustra la complexitat de les consultes de dades amb els diagrames E-R corresponents.

Una vegada la informació la tenim disponible, aquesta es maquetava en HTML i es guarda en un objecte que, posteriorment, s'integra segons la sortida escollida. La sortida de la informació pot derivar-se segons tres camins possibles:

- Integrada dins el portal *ripollet.cat*
- En format HTML, pantalla completa
- En format PDF signat electrònicament.

Tret dels casos en que l'usuari no està acreditat o bé que no es troba correspondència a la base de dades, aquests són els únics camins que porten a l'estat final del diagrama 3.16. Veiem que les activitats dels dos últims camins apareixen colorejades per facilitar-ne la distinció.

Els diagrames de seqüència següents mostren les interaccions de missatges entre actors i objectes que intervenen en aquest mòdul. L'esquema funcional, representat en aquests diagrames, s'ha distribuït en tres diagrames diferents per facilitar la comprensió.

El primer reflexa el comportament funcional bàsic de tots els serveis que s'agrupen en aquest mòdul. La informació produïda per l'execució de cada servei informatiu és diferent, però l'estratègia de recerca de les dades, la maquetació d'aquestes i la construcció de la sortida és comuna a tots ells.

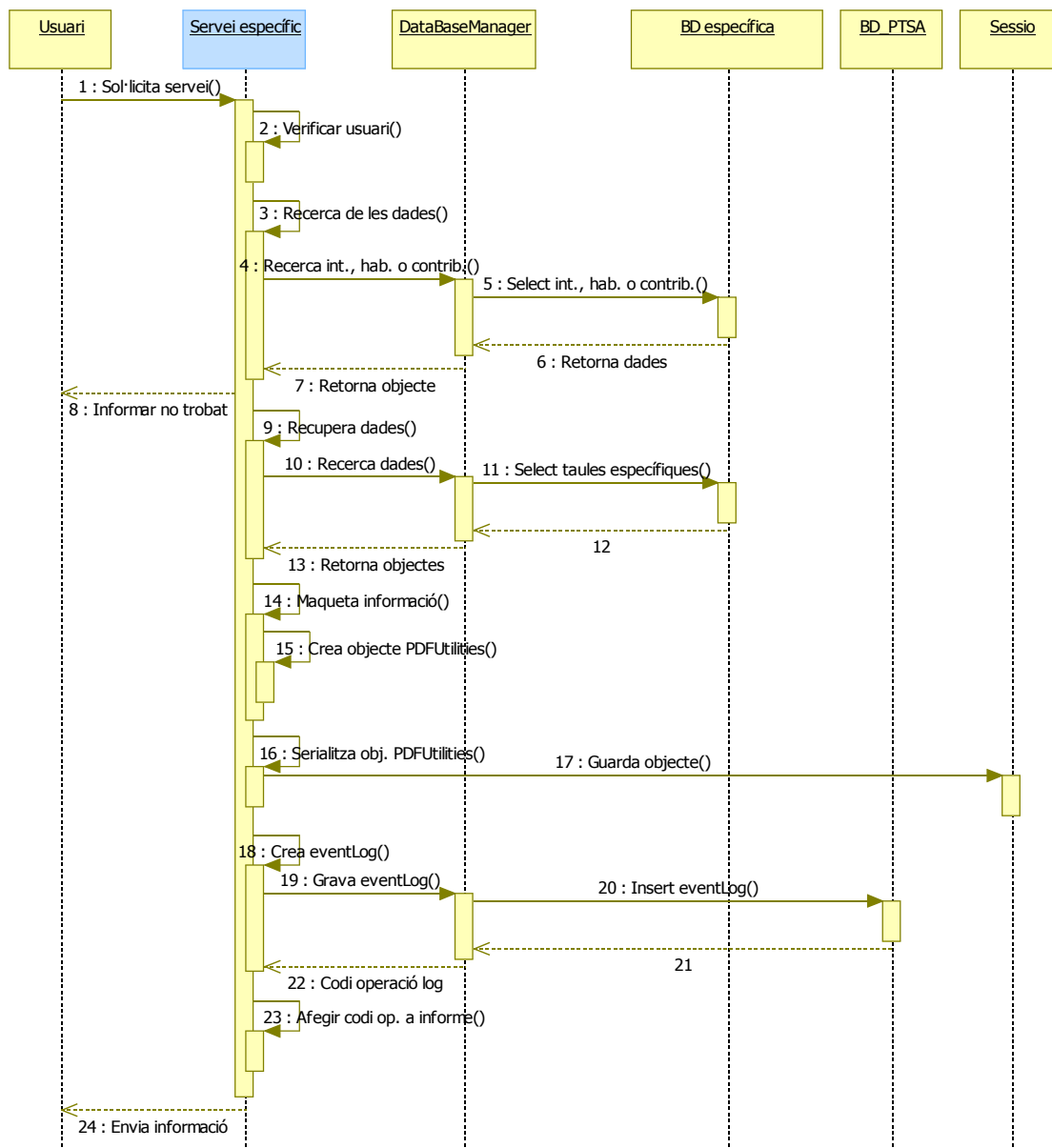


Fig. 3.17 Diagrama de seqüència principal del Mòdul 03

Per evitar la inserció d'un diagrama per cada un dels serveis, s'ha utilitzat noms generals per representar els actors. Així ens trobem amb l'actor *Servei específic* o bé *BD específic*. Abans ja hem parlat dels serveis als que ens referim i la seva correspondència amb la base de dades adequada.

Resumint la funcionalitat representada i de la mateixa manera que en d'altres diagrames ens referíem al subdiagrama de la figura 3.10, en aquest ho tornem a fer a la interacció 2, al verificar l'usuari acreditat.

La interacció 3 inicia un procés de recerca de dades que tracta de resoldre la correspondència entre l'usuari acreditat i l'interessat, l'habitant o bé el contribuent. Aquest problema el trobem a conseqüència de tenir com origen bases de dades independents. En cas que no trobem coincidències s'informarà d'aquesta situació a l'usuari i l'execució terminarà seguint la interacció 8.

En cas contrari, s'iniciarà un procés de recuperació de les dades necessàries per a construir el volant o justificant objecte del servei sol·licitat. Aquest procés serà més o menys complex segons la fragmentació de les dades. Als apartats següents es justifica la complexitat existent.

Quan es demana un servei de producció d'un justificant o volant nominal, a nom del mateix usuari, es crea directament. Però si la consulta demanada dóna com a resultat un conjunt de registres; per exemple: els registres d'entrada que l'usuari ha registrat; o bé els expedients amb els que hi figura com a interessat; o bé els pagaments que ha realitzat o en té pendent, es genera un llistat en format taula amb un selector de registre que en permet aïllar la producció del justificant adequat.

De la maquetació de les dades s'encarreguen els objectes *PDFUtilities* i *PageManager*. Es comenten més endavant però un detall important en quant a la funcionalitat és que l'objecte *PDFUtilities*, emmagatzemarà en un dels seus atributs el contingut HTML resultat del procés de maquetació. Posteriorment aquest objecte es serialitza i es guarda a la sessió web per evitar recalcular la informació en cas que es sol·liciti un altre tipus de sortida: HTML o PDF.

Finalment veiem com es pren nota de l'activitat, registrant l'*eventLog* a la base de dades del projecte: *PTSA*. Aquesta vegada es fa servir el codi d'operació o codi de log per incloure-ho als informes generats.

Una vegada el procés de producció del justificant ha acabat, l'usuari ho visualitza al navegador integrat amb la resta de *layers* del portal *ripollet.cat*, però des d'aquesta mateixa pantalla de visualització pot sol·licitar la generació en HTML o bé en PDF signat.

A continuació el diagrama de seqüència d'aquests dos processos similars:

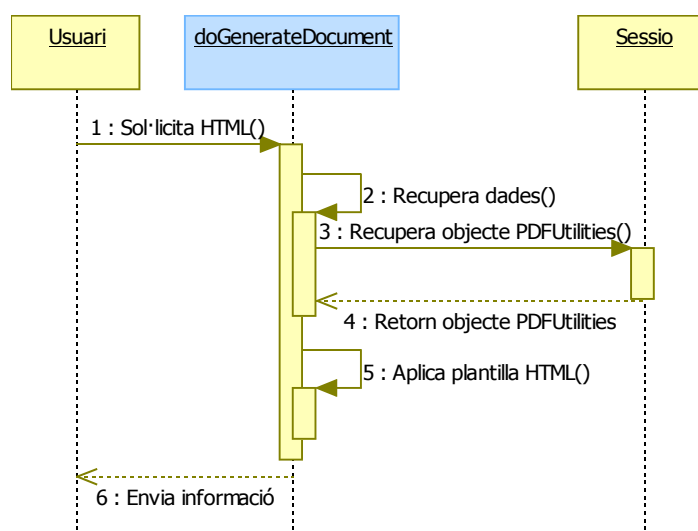


Fig. 3.18 Diagrama de seqüència del procés de generació d'informació en HTML

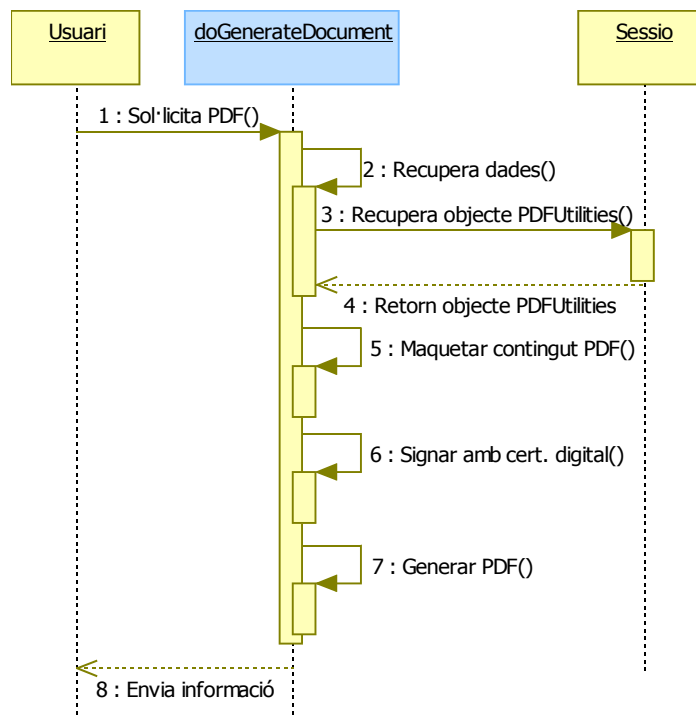


Fig. 3.19 Diagrama de seqüència del procés de generació informació en PDF

És interessant veure com tot dos processos consulten l'objecte *PDFUtils* serialitzat per recuperar la informació maquetada i estalviar recursos. En el cas de la generació HTML, el procés aplica una plantilla en blanc utilitzant com a fons el paper oficial de l'Ajuntament. Finalment en el cas de la generació PDF el document es produirà amb la col·laboració del component *ABCPdf* en quant a la renderització i incrustació del contingut HTML. Aquest últim procés conclou amb l'estampació d'una signatura digital de l'Ajuntament.

3.5.2 Diagrama de classes

El número de classes dissenyades que intervenen en el procés funcional d'aquest mòdul és proporcional al número de taules de les bases de dades que emmagatzemen la informació que utilitzarem per produir els informes.

D'una banda tenim classes que s'encarreguen del procés de generació i transformació de dades i d'altra, en tenim classes on els objectes instanciats corresponen a registres de taules o vistes de les bases de dades.

A continuació s'il·lustra les que s'encarreguen del procés:

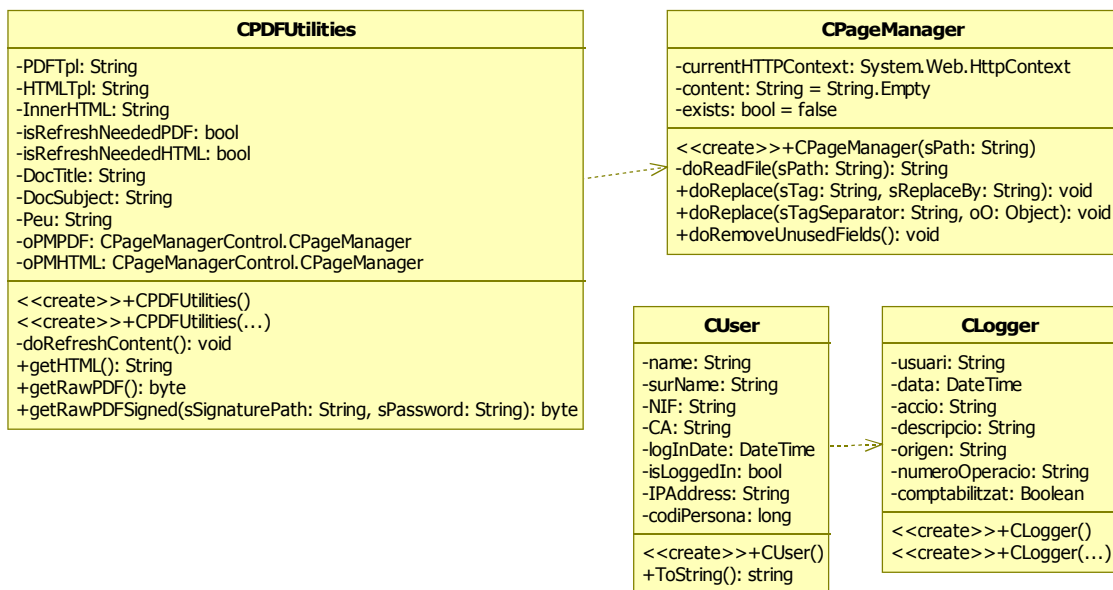


Fig. 3.20 Diagrama de classes implicades al procés de producció d'informes del Mòdul 03

Detallem cadascuna d'aquestes:

- *CPDFUtilities*: s'encarrega de la maquetació de les dades. Guarda les rutes a les plantilles utilitzades i la producció generada. Interacciona amb el control ABCPdf per a la producció de contingut PDF dinàmic. Els objectes d'aquesta classe són serialitzables per permetre el transport de les dades entre diferents línies de procés.
- *CPageManager*: realitza el procés de maquetació. S'explicarà a l'apartat 4.4, de desenvolupament.
- *CUser* i *CLogger*: classes amb funcionalitat general introduïdes anteriorment.

Per al tractament de la informació emmagatzemada a les bases de dades hem realitzat una abstracció directa de les vistes/taules disponibles amb l'objectiu de mantenir un equilibri entre homogeneïtat, rendiment i facilitar-ne el manteniment.

Tota la comunicació entre les bases de dades i la plataforma es recolza en la classe *CDatabaseManager* que es presentarà a l'apartat 4.5. Aquesta classe juga un paper molt important en el disseny d'aquest mòdul doncs s'encarrega de la instanciació de tots els objectes de les classes que ara presentem.

Continuant amb l'ordre establert per a la integració dels serveis d'aquest mòdul ara començaríem amb l'esquema de classes del servei *Consulta del registre d'entrada*, però aquest esquema ja s'ha introduït i comentat a l'apartat anterior 3.4.2.

Referent als serveis integrats en relació amb el *Padró Municipal d'Habitants* l'esquema és redueix a una única classe:

CCertificatEmpadronament
HABFNAC: DateTime
HABFALTA: DateTime
HOJPUER: String
BARDESS0: String
HOJPORT: String
HOJESCA: String
HOJPLAN: String
HOJKMT: String
HOJHMT: String
HOJBLOQ: String
HOJCNUMER: String
HOJNUMSUP: String
HOJCNUMSUP: String
HOJDIST: String
HOJSECCIO: String
HOJNUMER: String
VIACTIVIA: String
VIANVIA: String
HOJNHOP: String
HABIDEN: String
HABDCIDEN: String
MUNDESM: String
HABAPE2: String
HABPART2: String
PRODESPRO: String
HABNOMB: String
HABAPE1: String
HABPART1: String
HABNLOCAL: String
CCertificatEmpadronament()
CCertificatEmpadronament(...)
ToString(): string

Fig. 3.21 Classe CCertificatEmpadronament

- CCertificatEmpadronament: els seus atributs contenen els valors que provenen d'una vista de la base de dades del PMH. Els objectes d'aquesta classe permeten la construcció dels dos serveis de PMH que s'integren en aquest projecte. El primer d'aquests, el Justificant d'empadronament, només requereix la instanciació d'un objecte d'aquesta classe: el de l'empadronat amb HABIDEN igual al DNI de l'usuari acreditat. D'altra banda el Justificant de convivència, requereix la recerca del camp HABNLOCAL que indica el codi del full padronal de l'usuari acreditat i després, construir una col·lecció d'objectes que hi tinguin aquest mateix valor; és a dir, de persones que estiguin empadronades en aquest mateix full padronal.

Quant al servei *Consulta d'expedients*, l'esquema de classes és el següent:

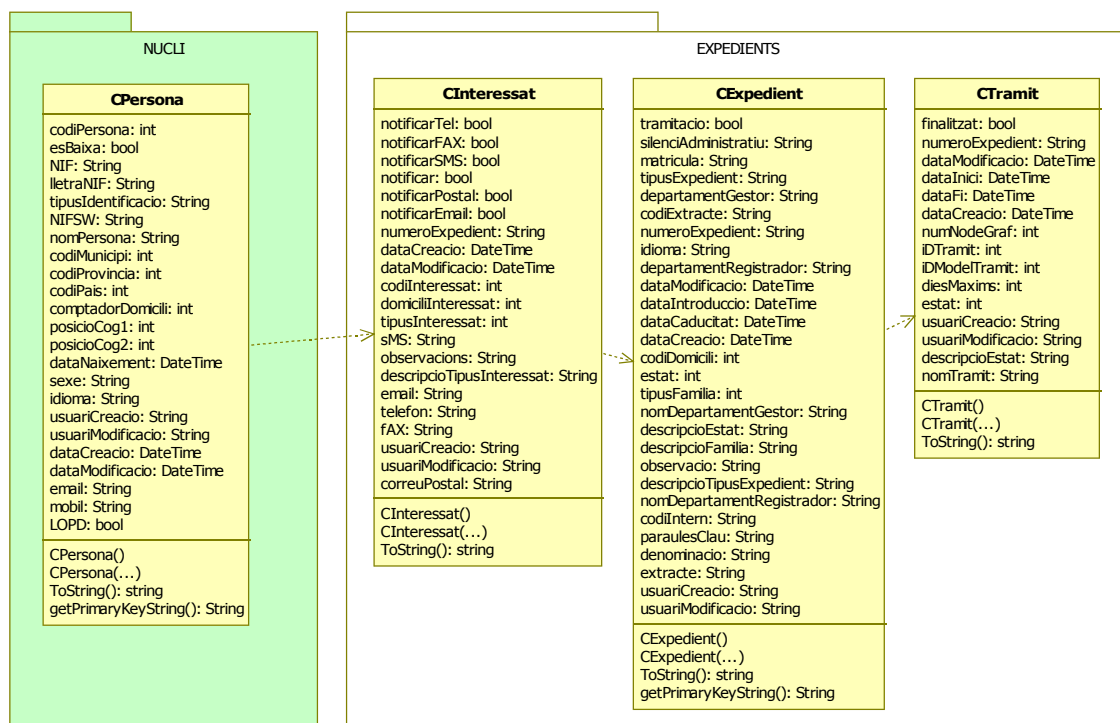


Fig. 3.22 Diagrama de classes implicades en la consulta d'expedients

La representació de la classe *CPersona* novament, en aquest diagrama, indica la compartició de recursos generals –persones, territoris, departaments, ...– entre els serveis de registre i expedients.

Les classes implicades en aquest procés són:

- *CPersona*: abstracció directa de la taula *tblPersones*. Introduïda anteriorment, recull les dades de la persona relacionada amb l'usuari acreditat.
- *CInteressat*: abstracció de la taula *tblInteressats*. Modela la relació entre una persona i un expedient.
- *CExpedient*: abstracció de la taula *tblExpedients*. Conjunt de dades referents a un expedient.
- *CTramit*: abstracció de la taula *tblTramits*. Conjunt de dades referents a un tràmit. La tramitació d'un expedient és una col·lecció de tràmits.

Finalment el servei *Consulta de pagaments* és el funcionalment més complex. L'esquema de classes dissenyat el componen 8 taules generades a partir del model de la base de dades. Recordem que la base de dades de l'aplicatiu d'*Ingressos* està formada per més de 300 taules. L'escassa documentació ha forçat realitzar un procés de disseny mitjançant enginyeria inversa. El resultat d'aquest procés ha seleccionat les classes del diagrama següent:

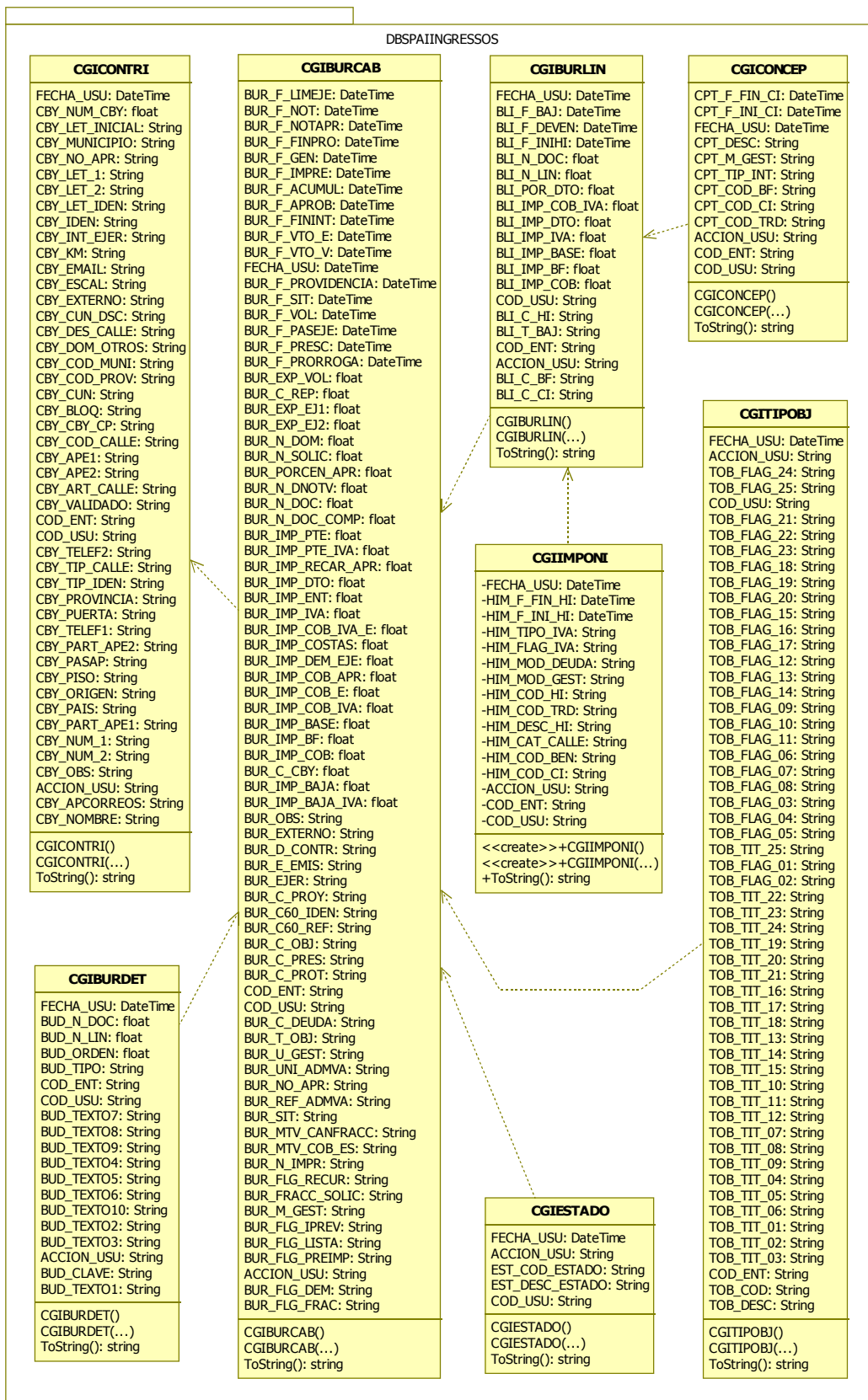


Fig. 3.23 Diagrama de classes implicades en la consulta de pagaments

La descripció d'aquestes classes:

- **CGI*CONTRI**: és l'abstracció de la taula **GICONTRI**. És la correspondència entre

l'usuari acreditat i el contribuent.

- *CGIBURCAB*: abstracció de la taula *GIBURCAB*. Recull de dades de documents d'ingressos.
- *CGIBURLIN*: abstracció de *GIBURLIN*. Dades de línies de conceptes dels documents d'ingressos
- *CGIESTADO*: abstracció de *GIESTADO*. Estat del document.
- *CGIBURDET*: abstracció de *GIBURDET*. Detalls del document.
- *CGICONCEP*: abstracció de *GICONCEP*. Conceptes d'ingrés de les línies del document.
- *CGITIPOBJ*: abstracció de *GITIPOBJ*. Tipus d'objecte tributari al que el document es refereix.
- *CGIIMPONI*: abstracció de *GIIMPONI*. Fet imposable de cada una de les línies de concepte d'ingrés.

Al tractar-se d'una abstracció literal de l'esquema de les bases de dades utilitzades, es pot consultar més informació a l'apartat següent. També es pot consultar una eina desenvolupada durant la fase de desenvolupament d'aquest projecte que permet la generació de codi automàtica de classes en C# a partir d'uns paràmetres relacionats amb la base de dades i taules. Aquesta eina es comenta a l'apartat 4.3.

3.5.3 Entitat-relació

Guardant molta similitud amb l'apartat anterior, ara es presenten els diagrames d'entitat-relació de les bases de dades estudiades en la fase de disseny d'aquest mòdul.

La recerca de la informació ha requerit la utilització de tècniques d'enginyeria inversa per aconseguir la localització de les dades necessàries dins els esquemes complexos de les bases de dades. La tècnica i procediment utilitzats es comenta a la secció 4.2.3 del capítol següent.

Nota: l'esquema de la base de dades del registre d'entrada, *RESDAR*, ja es va comentar a la secció 3.4.3, no tornarem a fer referència en aquest apartat.

Padró Municipal d'Habitants:

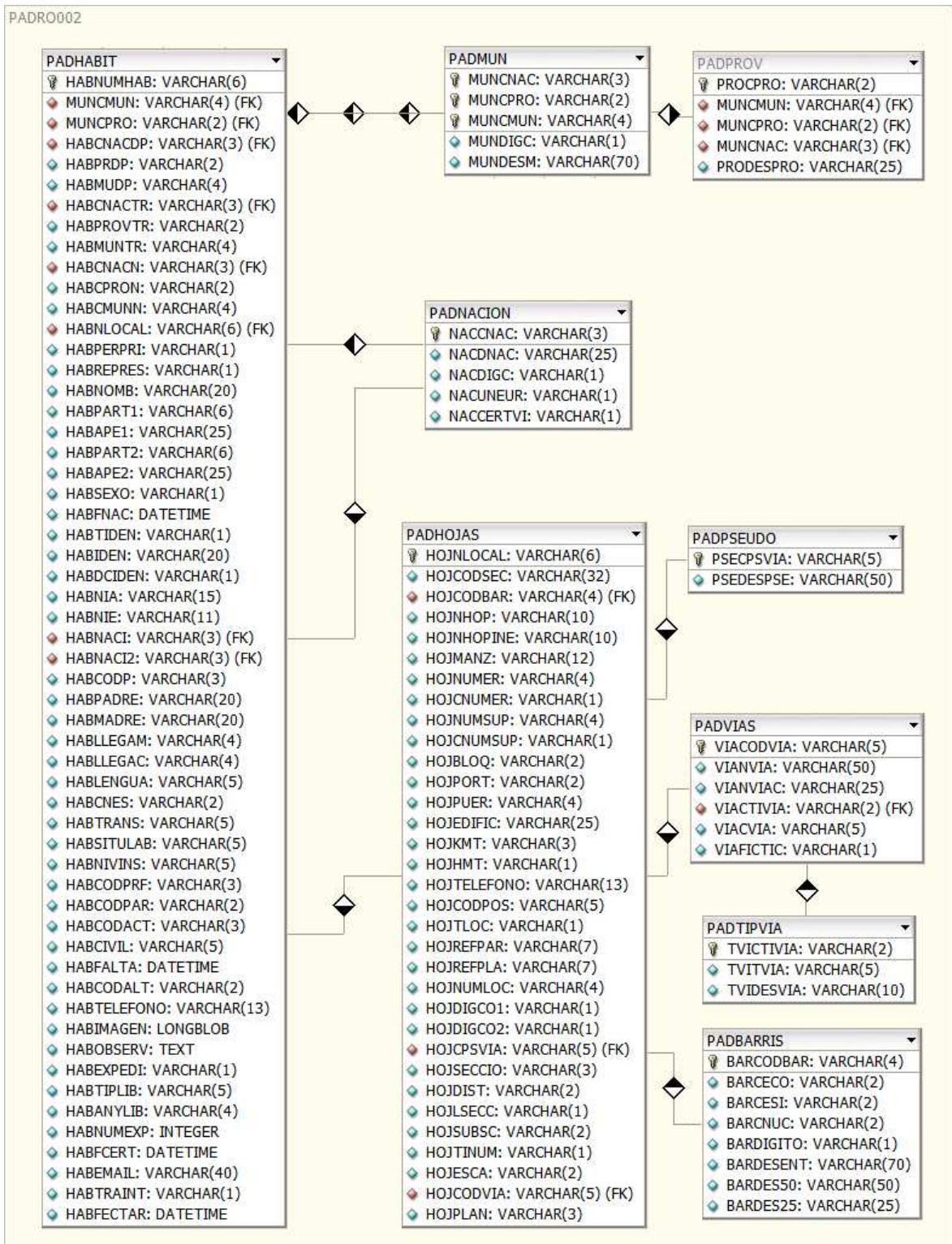


Fig. 3.24 Diagrama entitat-relació de les taules utilitzades del PMH

Segons el diagrama anterior veiem que la consulta de les dades necessàries per a produir l'informe justificant d'empadronament o convivència implica l'execució d'una sentència complexa.

Les taules que guarden les dades importants són *PADHABIT* i *PADHOJAS*, doncs emmagatzemen les dades de l'habitant empadronat i les del full padronal respectivament.

La resta, són taules de descriptors que afegixen per exemple la descripció del país natal, o el municipi d'origen, entre d'altres. Aquestes taules emmagatzemen pocs registres i aquesta és la raó per la que es va decidir implementar dues vistes per reduir la complexitat de la consulta d'aquestes dades. Les vistes, formen part dels recursos de la base de dades del projecte *PTSA* i són les que segueixen:

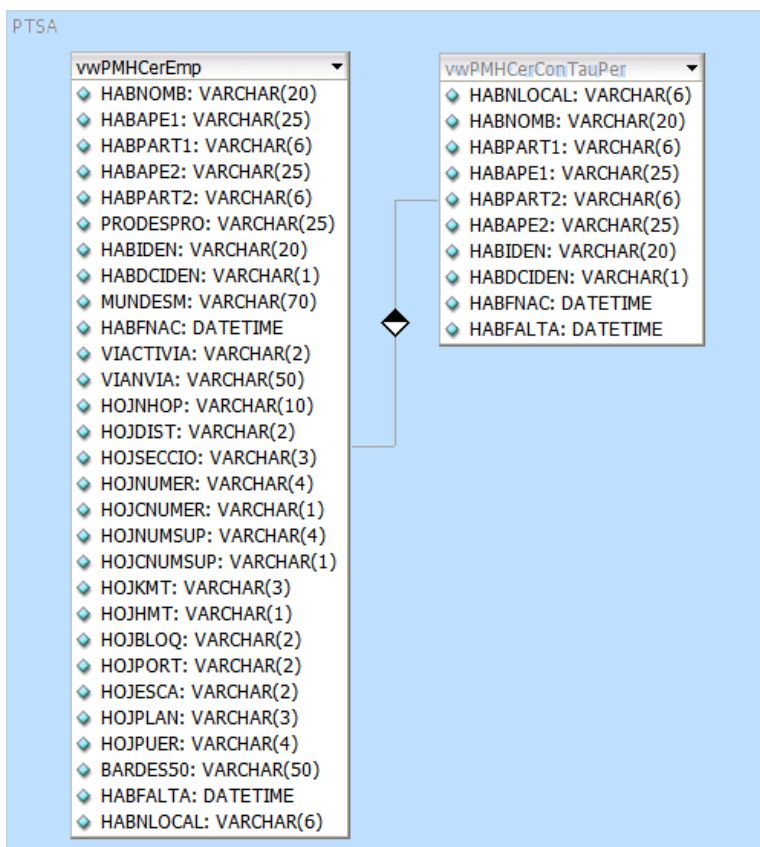


Fig. 3.25 Diagrama de entitat-relació amb les 2 vistes del PMH implementades en PTSA

Amb aquest canvi, la complexitat queda reduïda a la consulta de la vista *vwPMHCerEmp* quan volem produir el *justificant d'empadronament*. En el cas de la producció del certificat de convivència, s'ha de fer una segona consulta a la vista *vwPMHCerConTauPer*, doncs aquesta ens resumeix les dades necessàries per construir la taula de persones que conviuen amb l'usuari acreditat.

Insistim en que la creació de les vistes és possible doncs no hi ha caiguda de rendiment ni consum significant de recursos ja que les taules que intervenen en el procés emmagatzemen pocs registres.

Amb referència a la base de dades d'Expedients, s'ha aplicat també la tècnica d'abstreure en vistes l'agrupació de taules relacionades. El model inicial, atacant directament a la base de dades *Expedient* seria el representat pel següent diagrama:

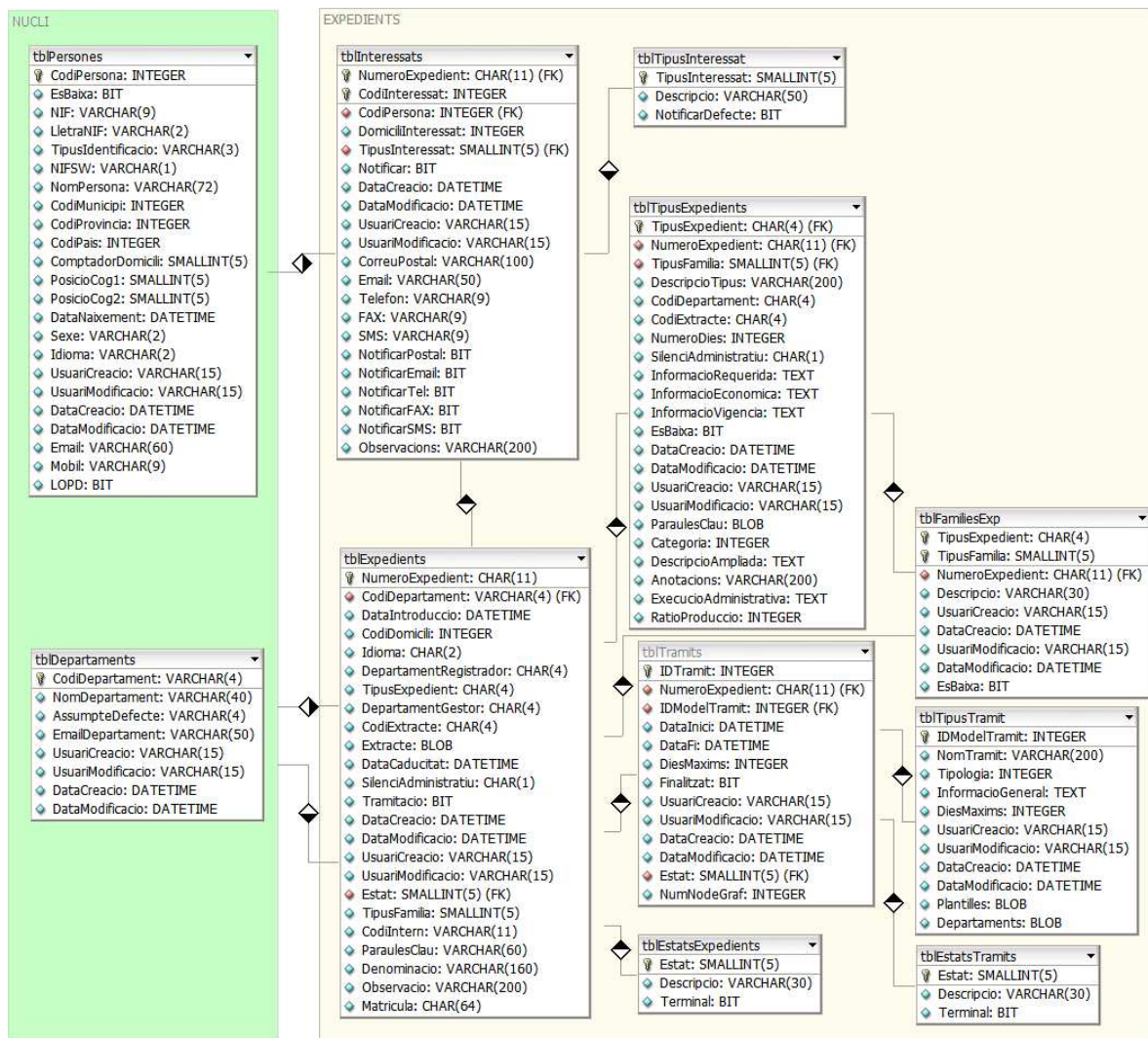


Fig. 3.26 Diagrama d'entitat-relació de les taules utilitzades d'Expedients

En aquest cas, les taules interessants són *tblInteressats*, *tblExpedients* i *tblTramits*. Aquestes emmagatzemen les dades relacionades amb els interessats, els expedients i els tràmits, respectivament. La resta de taules que intervenen, són taules de descriptors que afegixen els literals respectius a les tres taules principals. Aquest detall justifica el disseny de les vistes següents:

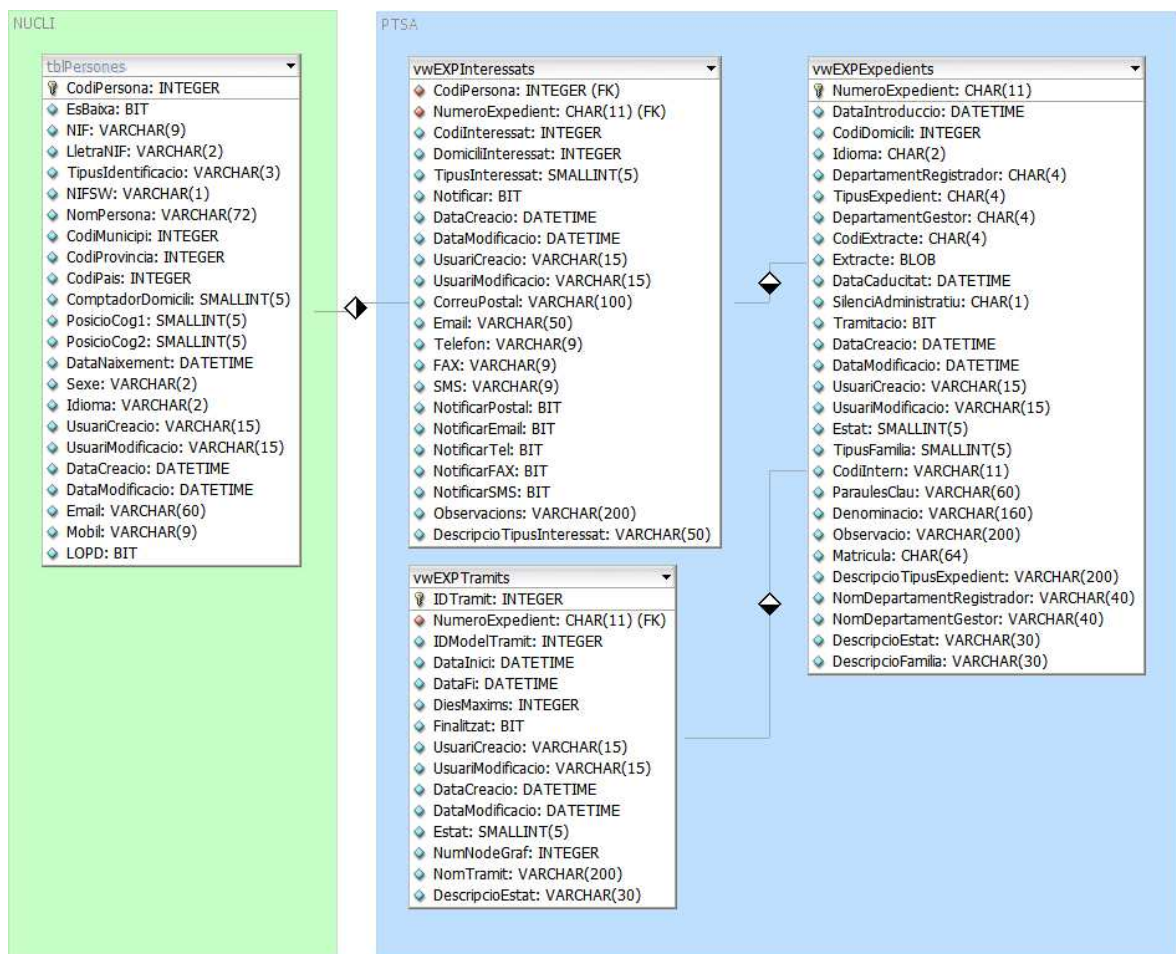


Fig. 3.27 Diagrama d'entitat-relació de les tres vistes d'Expedients implementades en PTSA

Igual que en el cas del Padró, les vistes s'han implementat com a recursos de la base de dades del projecte, *PTSA*. Comparant els diagrames 3.26 i 3.27 ens podem fer una idea de la simplificació de les consultes a l'hora de recuperar la informació. Aquesta reducció té un impacte directe en el disseny de les classes utilitzades com hem vist en la secció anterior.

Les relacions que apareixen en aquest diagrama ens revelen la correspondència entre les persones del *NUCLI* i els interessats d'*EXPEDIENTS*. El procés de localització dels expedients de l'usuari acreditat s'inicia a partir d'aquesta primera relació. Veiem doncs com una persona pot estar varies vegades com a interessat de varis expedients, que un expedient pot tenir varis interessats i varis tràmits.

Finalment a la base de dades d'Ingressos, *DBSPAIngressos*, amb més de 300 taules, no ha estat possible fer una abstracció en vistes doncs el número de registres emmagatzemats per les taules que intervenen en el procés és massa elevat i comportaria un consum de recursos important. Per exemple, les taules *GIBURCAB* i *GIBURLIN* emmagatzemem prop d'un milió de registres i *GIBURDET* prop de quatre milions.

Les taules consultades per aquest projecte es resumeixen en les representades en el següent diagrama:

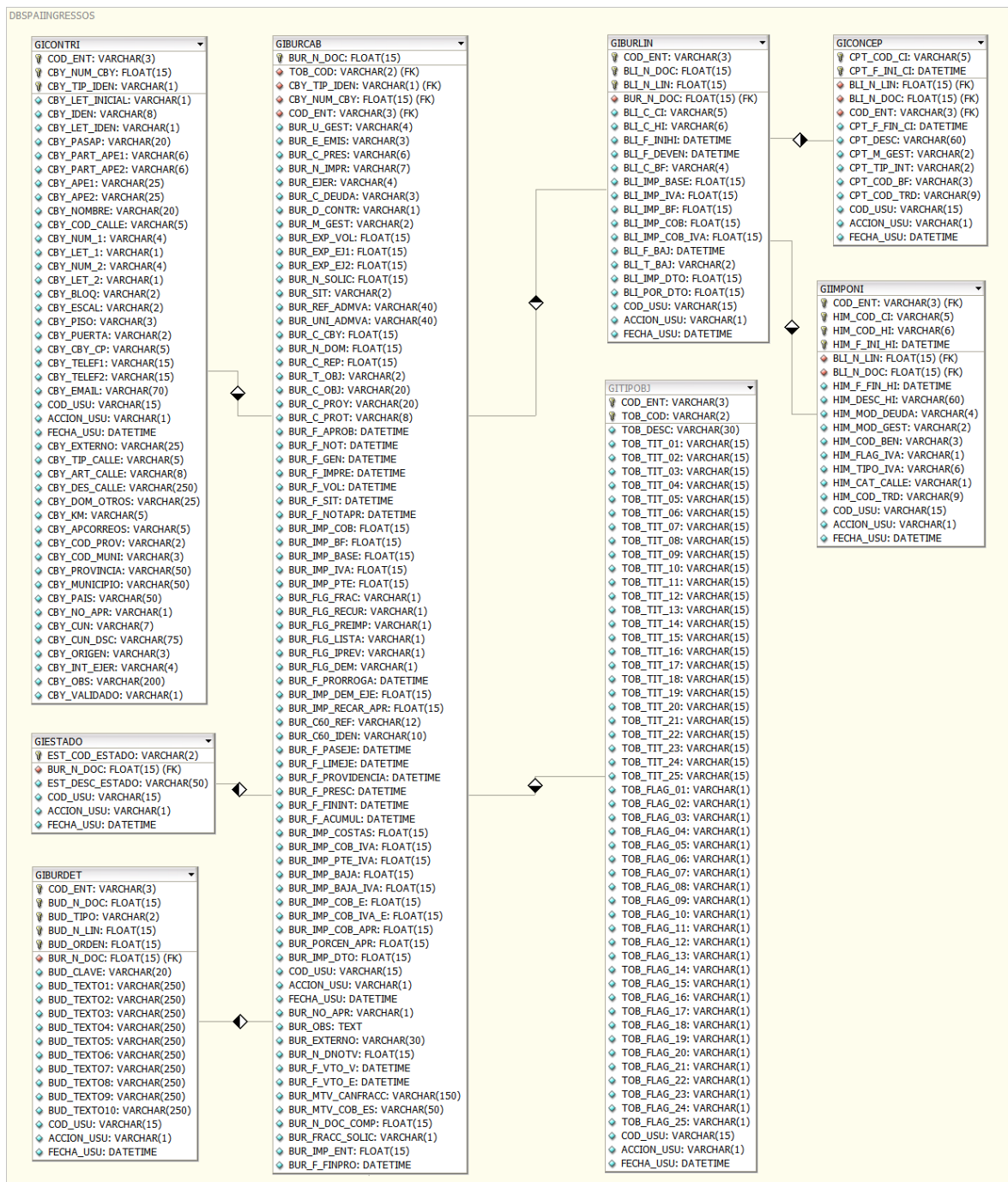


Fig. 3.28 Diagrama d'entitat-relació de les taules utilitzades de DBSPAllIngressos

En detall, cada una d'aquestes fan referència a les dades següents:

- **GICONTRI**: és la taula que emmagatzema les dades dels contribuents. Aquesta fa l'enllaç amb l'usuari acreditat doncs el camp CBY_IDEN guarda els dígits del DNI o document acreditatiu.
- **GIBURCAB**: és la taula encarregada de guardar les dades principals dels documents de recaudació. De forma independent guarda rebuts, autoliquidacions, liquidacions, ...
- **GIESTADO**: és la taula de descriptors de l'estat del document. Exemples: *cobrat en entitat bancària; cobrat en caixa; notificat i pendent de cobrament; ...*

- *GIBURDET*: detalls específics del document. Per exemple si el document és del tipus *BI* (Impost Bens Immobles), un dels detalls és la referència cadastral.
- *GIBURLIN*: línies del document. Conjunt de fets imposables extracte d'un mateix document.
- *GICONCEP*: taula de descriptors de conceptes d'ingrés.
- *GIIMPONI*: taula de descriptors de fets imposables.
- *GITIPOBJ*: taula de descriptors de tipus d'objecte tributari.

3.6 Comunicació entre els mòduls

La comunicació entre mòduls la componen els missatges que intercanvien els serveis d'un dels tres mòduls amb els altres o fins i tot amb ells mateixos. Aquest intercanvi de dades és necessari per a que la integració i el funcionament global del conjunt sigui l'esperat.

S'ha descartat una implementació basada en *webservices* ja que això hagués comportat una inversió en temps considerable a la fase de disseny i codificació en quant a definició de protocols, ports, etc... i no es preveu a curt termini la comunicació dels serveis de la plataforma amb d'altres consumidors remots.

El canal d'intercanvi de missatges que s'ha utilitzat ha estat l'emmagatzemament d'objectes serialitzats a la sessió web que el servidor d'aplicacions construeix per a tractar les peticions de l'usuari.

El procés de serialització a XML d'un objecte es basa en una codificació de l'objecte com a una cadena XML. Tant l'estructura com les dades es tradueixen a aquest format. Aquesta tècnica permet, segons aquesta transformació, convertir qualsevol objecte serialitzable en un tipus de dades primitiu com són les cadenes de caràcters. D'aquesta manera evitem els problemes de no acceptació d'estructures abstractes de dades com són els objectes.

La serialització d'objectes s'utilitza per a l'emmagatzemament d'objectes complexos en bases de dades, arxius, creació de còpies d'estat de sistemes, checkpoints, i en aquells processos on es requereix la persistència dels objectes.

El funcionament d'aquesta tècnica l'explicarem sobre un diagrama de seqüència:

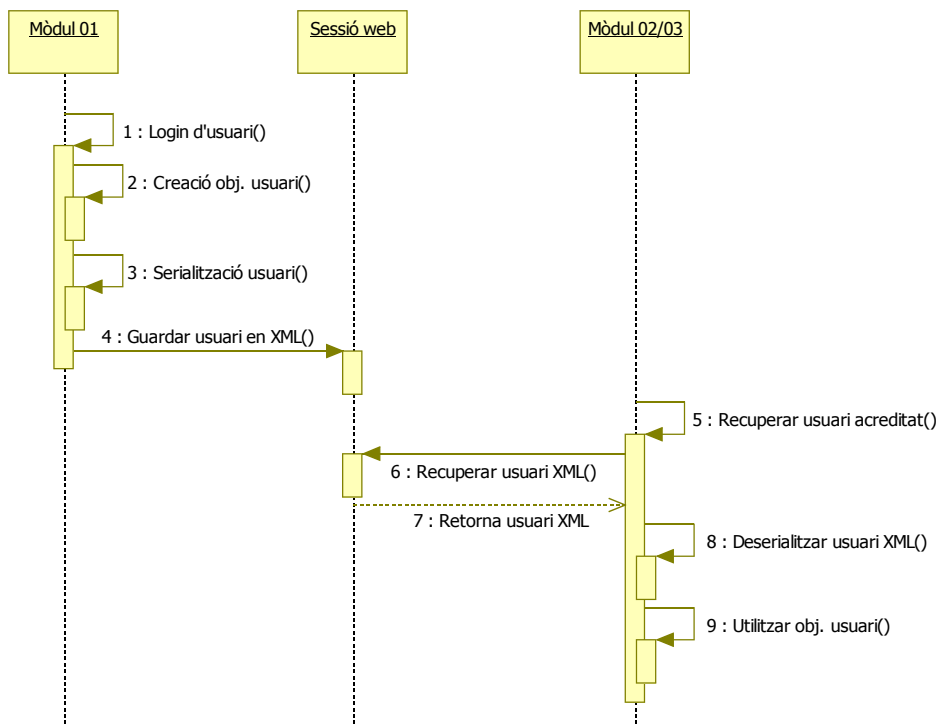


Fig. 3.29 Diagrama de seqüència de la serialització de l'objecte usuari

Comprovem segons l'anterior diagrama, com un objecte que es crea dins un mòdul, viatja fins a un altre gràcies al procés de serialització.

Aquest diagrama representa la implementació real de la comunicació que té lloc quan un usuari s'acredita a la plataforma, processat pel mòdul 01, i quan un dels altres mòduls comprova si l'usuari que demana els serveis es troba acreditat o no.

Fixem-nos com a la interacció 2 és el mòdul 01 qui crea l'objecte usuari. Després el serialitza en XML –interacció 3– i ho guarda a la sessió web –interacció 4–. Fins aquí el seu processament. Després un dels altres mòduls recupera l'objecte usuari segons la interacció 5. Demana la cadena XML a la sessió, aquesta la retorna i posteriorment es deserialitza a la interacció 8 obtenint com a resultat una còpia de l'objecte usuari original.

No és només en el procés d'acreditació on s'utilitza aquesta tècnica d'emmagatzemament d'objectes. També la utilitzem com a sistema de comunicació dels mòduls amb ells mateixos per evitar tornar a processar dades enviades per formularis si l'usuari torna enrere i accepta novament l'enviament, o bé quan es sol·licita recarregar la pàgina web des del seu navegador.

També la utilitzem al mòdul 03 quan l'usuari sol·licita un dels serveis informatius. La sol·licitud de l'usuari dispara un procés que té com a sortida una producció HTML integrada dins el portal ripollet.cat. Aquesta producció es guarda dins un objecte de la classe *CPDFUtilities* i es serialitza a la sessió. D'aquesta manera si l'usuari sol·licita la mateixa informació però en format HTML per imprimir, o en PDF signat, no es torna a processar la consulta de les dades ni la maquetació d'aquestes, ja que es recuperen de l'objecte serialitzat.

Capítol 4. Desenvolupament del projecte

4.1 Introducció

En aquest punt de la redacció del projecte tenim a la nostra disposició, d'una banda: un catàleg refinat d'objectius, resultat de la fase d'anàlisi i d'altra, el resultat de la fase de disseny: una descripció específica de cada bloc funcional junt amb el material objectiu que ens defineix com implementar la funcionalitat esperada.

Aquest capítol introdueix les tècniques i paradigmes de codificació aplicats a la fase de desenvolupament. Comença amb unes definicions prèvies dels recursos utilitzats. Segueix explicant l'abstracció en capes realitzada on s'introdueix el patró d'arquitectura utilitzat: el de *Model Vista Controlador*. A continuació d'aquest, es comenta cada una de les capes d'aquesta arquitectura i els seus detalls. El capítol finalitza amb els detalls d'integració i publicació en el servidor d'aplicacions existent.

4.2 Arquitectura i codificació

El patró arquitectònic utilitzat en el desenvolupament del projecte ha estat el *Model Vista Controlador (MVC)*. Aquest patró el va descriure Trygve Reenskaug⁴ a l'any 1979 quan treballava al projecte *Smalltalk* de *Xerox*.

Aquest pertany a la categoria de patrons per a sistemes iteratius i separa les dades, la interfície d'usuari i la lògica de control d'un sistema en tres components diferents.

És molt utilitzat en desenvolupaments on es vol independitzar la lògica de negoci de la interfície d'usuari. Un punt fort d'aquesta independització és la simplificació en la fase de desenvolupament de la interfície d'usuari ja que aquesta només ha d'integrar el codi per tractar les peticions de l'usuari. Això fa possible la implementació d'altres interfícies d'usuari coexistents i independents de la lògica de procés. Un altre punt fort és que els canvis que afecten a la lògica de negoci no alteren les interfícies d'usuari.

4 TRYGVE REENSKAUG. *Model-View-Controller (MVC)*. 1979.
<<http://heim.ifi.uio.no/~trygver/>>
(15 de juny de 2009).

En quant a la fase de codificació, recordem que el llenguatge de desenvolupament és el *C# –C Sharp–* sobre la plataforma *.NET FrameWork*, de Microsoft., executant-se de forma integrada dins del servidor d'aplicacions *Internet Information Server*. Aquest actuarà de contenidor dels *webforms* i objectes implementats per a satisfer les funcionalitats de les diferents capes de l'arquitectura utilitzada.

4.2.1 Definicions prèvies

Tot seguit introduïrem unes definicions prèvies lligades als detalls d'implementació en la plataforma *.NET FrameWork* integrada en *Internet Information Server (IIS)*.

Encara que aquesta plataforma permet la integració de projectes generats i desplegats segons diferents processos de desenvolupament, hem triat la tecnologia *ASP.NET* per dues raons principals:

- La integració amb els recursos del portal *ripollet.cat* és pràcticament directa doncs aquest està realitzat amb tecnologia *ASP*, predecessora de *ASP.NET*.
- Permet el desplegament directe de codi al servidor d'aplicacions evitant processos de compilació, enllaçat i registre.

Anomenem llavors ***webforms*** als arxius que contenen el codi desenvolupat per a satisfer els objectius del projecte. Són el principal mitjà de construcció d'aplicatius amb interfície web. Dins del codi d'aquests arxius, normalment ens trobarem etiquetes de format en *HTML* o bé *XHTML*.

El codi que genera el contingut dinàmic de les pàgines, pot anar dins el mateix arxiu – separat pels tags de control següents: `<% //codi %>`– o bé, a diferència de la tecnologia *ASP (Active Server Pages)*, en un altre arxiu referenciat seguint el model *code-behind*. Aquest codi correspon al *server-side-script* que s'executa al servidor. Tant en la primera com en la segona tècnica, en *ASP.NET*, el codi es precompila pel servidor d'aplicacions durant la primera execució, generant uns fitxers objectes que s'utilitzen en les properes execucions. Quan l'arxiu font s'actualitza, es torna a realitzar el procés de compilació de forma automàtica.

Els recursos de codi desenvolupat en aquest projecte no utilitzen el model *code-behind* amb l'objectiu de tenir una correspondència més aproximada entre els arxius de codi generats, *webforms*, i l'arquitectura que s'implementa, *MVC*. La capa vista d'aquesta la configuren aquests arxius de codi.

La plataforma *.NET* permet la integració de controls per a la realització de tasques específiques. Un ***control*** és un component dissenyat amb unes funcionalitats aïllades i específiques. Aquest encapsulament fa que en sigui fàcilment reutilitzable. En el desenvolupament d'aquest projecte s'ha utilitzat 2 tipus de controls. Les classes que configuren la capa controladora de l'arquitectura, s'han encapsulat com a controls per a permetre la seva integració amb els *webforms*. Per un altre costat s'ha integrat controls

de tercers per a resoldre tres qüestions particulars: el *NeatUpload*⁵ per a solventar la tramesa d'arxius entre l'usuari i la plataforma; l'*ABCpdf*⁶ per a la generació de contingut PDF dinàmic; i el *PSISNet*⁷ per a facilitar la invocació dels webservices de PSIS de CATCert.

4.2.2 Capes de l'aplicació

El patró arquitectònic sobre el que treballarem el desenvolupament, el *Model Vista Controlador*, és l'idoni en molts desenvolupaments amb interfície web ja que classifica l'aplicació en tres capes a les que es pot fer la següent abstracció directa:

- Bases de dades → Model
- Sortida web → Vista
- Entrada web → Controlador

Representant les capes del projecte segons l'arquitectura d'aquest patró podem observar el diagrama següent:

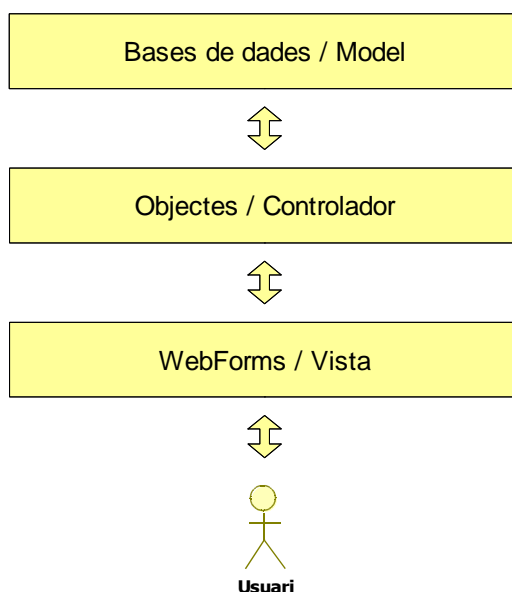


Fig. 4.1 Model de capes del patró arquitectònic MVC

A continuació es defineixen cada una d'aquestes capes i es comenten els recursos

5 BRETTLE DEVELOPMENT. *NeatUpload™ ASP.NET Component*. 2009.
<<http://www.brettle.com/neatupload>>
(15 de juny de 2009).

6 WEBSUPERGOO. *ABCpdf .NET™ 7.0*. 2009.
<<http://www.websupergoo.com/products.htm#pd>>
(15 de juny de 2009).

7 CATCERT. AGÈNCIA CATALANA DE CERTIFICACIÓ. *Plataforma de serveis d'identificació i signatura (PSIS)*. 2009.
<http://catcert.cat/web/cat/1_4_3_plataforma.jsp>
(15 de juny de 2009).

desenvolupats en relació amb la solució codificada.

4.2.3 Model

La capa *model* de l'arquitectura *MVC*, es troba constituïda per la representació específica de la informació i les regles que garanteixen la integritat i congruència de les dades.

Estretament lligada amb el model de negoci dels sistemes que s'integren en el nostre cas, aquesta capa la configuren les diferents bases de dades proveïdores de la informació que es processarà per entregar finalment a l'usuari.

Les bases de dades ja han estat introduïdes en anteriors apartats. Explicarem ara, en fase de desenvolupament, la càrrega que ha suposat entendre i implantar-les com a model de dades.

La base de dades de registre –*RES DAR*–, i la d'expedients –*EXPEDIENTS*–, són de desenvolupament intern. Aquest detall ha facilitat les tasques de generació del codi necessari pel tipus d'accés, la recerca de les dades i les consultes adequades.

D'altra banda, les bases de dades de PMH –*PADRO002*–, i Ingressos, –*DBSPAIngressos*–, han estat dissenyades i proveïdes per terceres empreses distribuïdores de software i serveis. A més, tal com hem introduït abans, aquestes bases de dades són de relativa complexitat, el seu esquema el componen més de 100 taules i més de 300 respectivament. La recerca de la informació quan l'origen es troba tan fragmentat, és complexa i el no disposar de documentació tècnica encara ho dificulta més.

En aquests dos casos hem utilitzat l'enginyeria inversa per aconseguir el coneixement suficient com per dissenyar els enllaços i consultes necessàries per recuperar les dades desitjades. Ho explicarem a continuació:

Aquestes dues bases de dades es troben en funcionament des de fa anys a l'Ajuntament i conformen un alt percentatge de les dades emmagatzemades pel sistema gestor. Entenem que la definició del seu model de dades no és objectiu d'aquest projecte, però sí que l'haurèm d'entendre per fer recerca de les dades que necessitem per satisfer els objectius marcats.

Actualment la informació que produirà la plataforma a través dels serveis que integrarà, es genera utilitzant les aplicacions client d'aquestes dues bases de dades. Són part de les eines que un funcionari utilitza diàriament per desenvolupar aquest tipus de tasques a petició de l'interessat. Parlem del programa del *Padró Municipal*. O bé del programa d'*Ingressos*.

El primer d'aquest s'utilitza per generar, entre d'altres, els justificants d'empadronament i convivència, mentre que el segon genera els justificants de pagament de qualsevol concepte tributari. En total, tres dels serveis a integrar en la plataforma.

Queda clar que aquests programes saben com i d'on aconseguir les dades.

El procés d'enginyeria inversa utilitzat per aconseguir aquesta informació ha estat l'aprofitament de tenir permisos d'administrador d'aquestes bases de dades i poder activar l'*analitzador d'sql*. Amb aquest programa ens ha estat possible veure quines sentències executen els programes contra aquestes bases de dades i aconseguir el procediment per a reproduir el mateix accés però aquesta vegada des de la plataforma del projecte.

4.2.4 Vista

La capa *vista* del patró MVC s'encarrega de la presentació de la informació a l'usuari. Un projecte pot tenir diferents vistes sense alterar ni la capa model ni la capa controladora.

En aquest projecte, presenta tota la informació a l'usuari a través d'una interfície web així com es plantejava a l'objectiu OBJ_00_01 i la configuren un conjunt de *webforms* que s'encarreguen del tractament de la informació generada a la capa controladora. Aquests *webforms* es localitzen al directori arrel del projecte i es diferencien dels altres donat que el seu nom d'arxiu comença amb el prefixe *do*.

També entenem que la generació de contingut en format PDF és una altra vista del projecte, ja que la producció de la informació es fa d'una manera diferent. En aquest cas particular el *webform* encarregat de la generació i enviament d'aquest tipus de vista és el *doGenerateDocument*. L'execució d'aquest implica la disponibilitat dels controls web *CABCPDFAnnotations* –package de classes necessàries per la generació de contingut PDF amb el control *ABCPdf*– i *CPDFUtilities* –classe que integra els mètodes de comunicació amb el control *ABCPdf*–.

En referència amb les vistes que generen contingut HTML, un *webform* del projecte pot produir informació directament o bé utilitzant els arxius de recursos del directori */template*, on es troben les plantilles HTML que s'envien, després del seu tractament, al client.

S'ha desenvolupat una classe per facilitar el tractament de les plantilles i la substitució dels camps dins d'aquestes pels valors produïts per la capa controladora. Aquesta classe és la *CPageManager* i s'explica en detall a l'apartat 4.4 en referència a la integració de la plataforma en el portal web existent.

4.2.5 Controlador

La capa controladora del patró MVC actua com interfície entre les dues altres. És on resideix la lògica de negoci de l'aplicatiu. Rep les peticions de part dels usuaris i recupera la informació del model, la processa i l'entrega a la capa de vista.

La lògica funcional d'aquest projecte ja la coneixem i es limita a:

- Validar el certificat digital de l'usuari i la construcció de la seva sessió a la plataforma.
- Entrada i sortida de les bases de dades implicades. De l'entrada a aquestes

s'encarrega el Mòdul 02 i de la sortida el Mòdul 03.

- Anotar l'activitat de l'usuari en la plataforma.

La capa controladora d'aquest projecte la configuren un número elevat d'arxius i recursos. Els comentem a continuació agrupant-los en tres blocs:

El primer bloc de recursos el componen el webform de tractament del certificat digital, *doCertifiedLogin*; els de tractament de les peticions d'alta d'interessat o de registre d'entrada del Mòdul 02, *doAltaPersona* i *doAltaRegistreEntrada*.

El segon bloc fa referència als controls web amb funcionalitats específiques per al tractament de la informació, són els següents:

- *CClientCertificateUtilities*: package que conté les classes útils pel tractament i gestió del certificat digital d'usuari.
- *CDatabaseManager*: classe que actua d'interfície amb les bases de dades e independitza els objectes del tractament amb aquestes. Explicada en detall a l'apartat 4.5.
- *CPSISUtilities*: package controlador del client del servei PSIS de CATCert. Utilitzat per a resoldre la comprovació de si el certificat digital està revocat o no.
- *CUtilities*: package de classes amb utilitats comunes. Algunes estàtiques.

El tercer i últim bloc inclou tots els controls webs que integren la definició de les classes dissenyades per a la gestió de continguts de les bases de dades implicades:

- *CABSISPMH*: package de classes relacionades amb la base de dades del Padró Mpal d'habitants.
- *CSSPAIngressos*: package de classes relacionades amb la base de dades d'Ingressos.
- *CNucli*: package de classes relacionades amb la base de dades *Nucli*.
- *CExpedients*: package de classes relacionades amb la base de dades *Expedients*.
- *CCounter*: classe que defineix un comptador. S'instancia un objecte durant el procés transaccional d'inserció de registres.
- *CDocument*: classe que modela un document de registre d'entrada.
- *CLogger*: classe que modela un objecte *log*. S'instancia un objecte durant el procés d'anotar un esdeveniment a la base de dades.
- *CUploadFile*: classe que modela el fitxer que s'adjunta al registre d'entrada.
- *CUser*: classe que modela les dades de l'usuari acreditat i es guarda a la sessió serialitzat en XML.

Aquest recursos s'integren amb la capa de vista proveint la informació adequada per a que aquesta la presenti a l'usuari en el format sol·licitat.

La major part del processament a la capa controladora és responsabilitat de la classe *CDatabaseManager* ja que s'encarrega de totes les operacions de lectura/escriptura contra les bases de dades o, dit d'una altra manera, de tota la missatgeria amb la capa *model*.

4.3 Generador de classes

Fent referència aquest últim bloc, la majoria de classes integrades en aquests controls web, s'han generat automàticament amb un generador de codi. El desenvolupament d'aquest generador, s'ha executat junt amb el del projecte i aquest és el motiu que fa que ara el comentem:

A la fase de disseny del projecte, ja ens fèiem a la idea que hauríem de desenvolupar un número elevat de classes. Els models d'entitat-relació de les bases de dades utilitzades són relativament complexos i es vol que cada taula requerida tingui correspondència amb una classe amb l'objectiu d'independitzar el processament dels objectes de les lectures/escriptures contra les bases de dades.

Arribat el punt de codificar una per una tantes classes com taules i tants atributs com columnes, vam fer recerca per veure si existia alguna utilitat de lliure distribució que generés el codi desitjat per cada una d'aquestes classes o el més acurat possible.

Finalment la recerca no va acabar amb èxit en el temps esperat i vaig pensar que invertint el temps de codificació d'un parell de les classes necessitades, en podria desenvolupar el generador de codi per mi mateix. Així ho vaig fer i el resultat va ser el webform *doClassGenerator* –vista del generador de codi– i el control web *CClassFromSQLServer* –controlador del generador de codi–.

Aquesta utilitat rep com a paràmetre el nom de la base de dades i la taula objectiu a modelar com a classe i retorna el codi *C#* que implementa la classe amb tots els atributs amb els tipus de dades correctes i un parell de mètodes per estendre la classe *CDatabaseManager* amb l'objectiu de l'acceptació dels nous objectes.

4.4 Integració en portal web existent i comunicacions amb l'usuari

Segons la redacció inicial d'objectius del projecte, en referència a la comunicació amb l'usuari, es requeria una plataforma amb sortida web i integrada amb el portal existent *ripollet.cat*. També, que tota la producció web sigués vàlida segons l'especificació *DTD HTML 4.01 Transitional* del *W3C*. Consultar els objectius *OBJ_00_01* i *OBJ_00_02*.

Un dels recursos existents del portal és la generació d'un fitxer estàtic amb el contingut HTML vàlid que maqueta tots els elements comuns a les pàgines d'aquest. Aquest integra les plantilles d'estil, les llibreries *JavaScript*, els blocs de navegació, els banners promocionals, etc... El nom d'aquest arxiu és *portada* i el trobem a l'arrel del directori del servidor virtual de *ripollet.cat*.

Aquest recurs es tradueix en un arxiu plantilla que disposa d'uns camps específics per a incloure el codi HTML produït per tercers. L'objectiu llavors en quant a producció de dades des de la plataforma es resumeix a produir codi HTML vàlid segons l'especificació imposada i integrar-ho en la plantilla existent.

Per fer això possible s'ha dissenyat una classe encarregada de fer la lectura dels arxius de recursos, plantilles, i disposa de mètodes per facilitar-ne la substitució de camps per valors. Aquesta classe, anomenada *CPageManager*, realitza el procés de maquetació. A partir d'un arxiu amb tags especials, els reemplaça pels valors adequats.

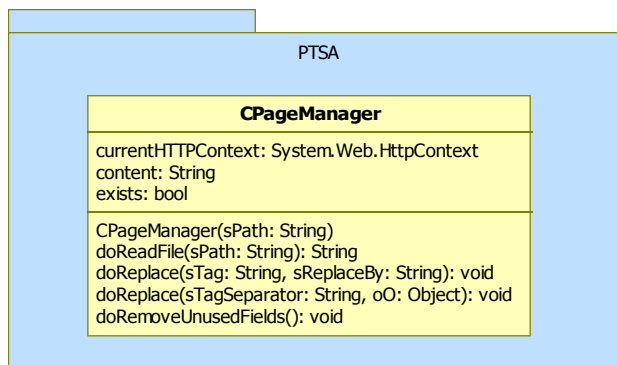


Fig. 4.2 Classe *CPageManager*

La utilització d'un objecte d'aquesta classe és molt senzill. El constructor d'aquest rep com a paràmetre la ruta absoluta a l'arxiu de recursos o plantilla i el llegeix. Si l'arxiu no es troba, l'atribut *exists* indicarà aquesta situació. Si l'arxiu existeix, el contingut d'aquest es trobarà disponible com un *String* a l'atribut *content*. En quant a mètodes *doReplace* realitza la substitució del camp passat com a paràmetre *sTag*, pel valor *sReplaceBy*. La sobrecàrrega d'aquest mètode realitza, de forma iterativa, la substitució dels camps que tinguin correspondència amb els atributs públics de l'objecte passat com a referència *oO*. Aquest mètode és molt pràctic per emplenar les plantilles de justificants amb els valors continguts als objectes.

Cada contingut estàtic disposa d'un arxiu de recursos dins el projecte. El repositori de plantilles es troba al directori */templates* del projecte i inclou per exemple: les pantalles informatives de *login*, *logout*, d'error, els formularis de recollida de dades, les maquetes dels justificants, etc...

La integració d'aquestes amb el projecte i amb *ripollet.cat* és possible gràcies aquesta classe i ho il·lustra la figura següent:

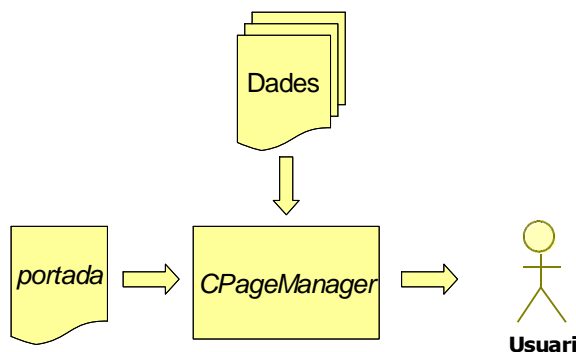


Fig. 4.3 Esquema de integració de la plataforma amb portal existent.

D'altra banda en quant a les comunicacions amb origen l'usuari, totes i cada una d'aquestes es fan sobre canals segurs de comunicació. Tret de la primera pàgina

doLogin, la resta es serveixen sobre HTTPS/SSL amb una encriptació de 128bit i utilitzant un certificat de dispositiu segur emès per CATCert per a *ripollet.cat*.

La recollida de dades en formularis es farà usant el mètode *POST* per garantir l'encriptació total d'aquestes. No s'utilitzarà el mètode *GET* ja que les dades s'envien com a paràmetre dins la nova petició URL que s'envia al servidor i aquest enviament es fa en *plain/text* sense encriptació.

4.5 Integració amb les bases de dades

En alguns dels diagrames del capítol anterior, ha estat impossible evitar nombrar la classe *CDatabaseManager* degut a la seva importància en quant a l'accés contra les bases de dades utilitzades.

La implementació d'aquesta classe està motivat per tres objectius principals:

- *Independència entre objectes i bases de dades*: no volem desvirtuar el concepte d'abstracció en objectes afegint mètodes de càrrega o actualització de dades dins la definició de les classes d'aquests.
- *Facilitar-ne el manteniment*: volem agrupar tots els mètodes que tenen accions contra les bases de dades en una mateixa classe.
- *Minimitzar consumició de recursos*: cada objecte d'aquesta classe només utilitza una connexió contra la base de dades, independentment del número d'operacions que en realitzi. Aquesta connexió s'estableix utilitzant un usuari de només lectura o bé de lectura/escriptura segons la necessitat del procés específic.

Un objecte *DatabaseManager* actua com interfície entre els objectes creats en els blocs de procés i les bases de dades. És la passarel·la que envia i recupera un objecte de la base de dades. A continuació la il·lustrarem amb el seu diagrama de classe:



Fig. 4.4 Diagrama de la classe *CDatabaseManager*

Fixant-nos a la llista de mètodes que integra, trobem un munt de mètodes que retornen objectes resultats d'una consulta contra les bases de dades. El nom d'aquests mètodes als que ens referim comencen amb el prefix *get*, així doncs trobem el mètode *getPersonaByPK*, que retorna un objecte de la classe *CPersona* segons el *CodiPersona* passat com a paràmetre, o bé el mètode *getLiniesDocument* que retorna una col·lecció d'objectes *CGIBURLIN* segons l'objecte *oDocument* passat com a referència.

Comentem també que el mètode *doInsert* es troba sobrecarregat i permet la inserció dels objectes de les classes *CLogger*, *CPersona*, *CDocument* i *CUploadFile*. Els objectes de la classe *CLogger* es construeixen i s'insereixen en qualsevol dels mòduls de la plataforma doncs s'encarreguen d'apuntar l'activitat de l'usuari, però els altres tres objectes només es construeixen i s'insereixen en els processos agrupats en el mòdul 02, el *registre d'entrada telemàtic*. Recordem que aquest mòdul és l'únic on l'usuari actua com a productor de les dades en tota la plataforma. Per aquest motiu no trobarem la sobrecàrrega d'aquest mètode per d'altres objectes que no siguin d'aquestes classes específiques.

Cada mètode *getX* integra dins el seu codi la consulta específica per a la recuperació de l'objecte desitjat. Aquest s'executa contra la base de dades utilitzant una connexió establerta amb un usuari de només lectura. En canvi cada mètode *doInsert*, integra una operació transaccional d'inserció contra la base de dades on s'estableix una connexió amb un usuari de lectura/escriptura. El tractament d'excepcions en aquests mètodes està dissenyat de tal manera que en cas de produir-se una, el procés de tractament evita incongruències a les bases de dades fent una operació *rollback*.

La utilització de la classe *CDatabaseManager* ens reporta beneficis directes a la fase d'implementació, doncs aïlla els processos específics de tota la càrrega de gestió i manipulació contra les bases de dades.

4.6 Publicació en servidor d'aplicacions existent

Els arxius i recursos que componen el projecte es divideixen en cinc grups:

- *els webforms*: contenen el *server-side-scripting* que processa la sortida web de la plataforma.
- *les classes*: abstracció de les dades i actors que intervenen al sistema.
- *les plantilles*: maquetes de documents que s'utilitzen en la producció de la informació sol·licitada.
- *els controls*: arxius i llibreries de tercers que aporten funcionalitats utilitzades al projecte.
- i *els certificats*: arxius amb els certificats que s'utilitzen per signar electrònicament el contingut PDF generat dinàmicament.

Tot aquest conjunt de recursos han d'integrar-se amb el servidor d'aplicacions amb l'objectiu d'aconseguir el funcionament global dissenyat. A continuació explicarem com publicar els continguts anteriors.

Ja que el projecte està integrat en el portal *ripollet.cat*, ha de tenir accés als recursos d'aquest. Es crearà llavors una carpeta amb el nom del projecte dins el mateix servidor virtual. La carpeta s'anomena *PTSA*.

El nou directori virtual *PTSA*, es configurarà dins l'*IIS* com una nova aplicació amb el nom *PTSA* amb permisos d'execució de "*Només seqüències de comandos*" i un nivell de protecció "*Alta (aïllada)*". Es fa d'aquesta manera per tal que el projecte s'executi com a procés independent i no es compromet l'estabilitat i rendiment d'aquest.

Dins aquest nou directori es copiaran els arxius dels *webforms* i de les *classes*. Es configurarà l'*IIS* per tal que tots aquests, tret del *webform* inicial *doLogin*, requereixin un canal segur de comunicacions. El *doCertifiedLogin* també es configurarà per a que requereixi certificats de client. S'aplica aquesta restricció per obligar a l'usuari a acreditar-se utilitzant un dels certificats digitals de la seva propietat.

Per a l'acceptació dels certificats d'usuari, s'ha de configurar la *Certificate Trusted List (CTL)* de *IIS* afegint les arrels de certificació de les agències certificadores acceptades pel projecte.

També i dins del mateix directori */PTSA* crearem les carpetes */certificates* i */templates*. A la primera s'ha de prohibir l'accés de l'usuari anònim, *IUSR_X / IWAM_X*, doncs és on guardarem els certificats amb els que signarem el contingut PDF generat. A la segona guardarem les plantilles HTML del projecte.

Finalment els controls utilitzats –*ABCPdf*, *NeatUpload* i *PSISNet*– s'instal·laran seguint les instruccions proporcionada per cada un d'ells. Normalment es copia els arxius *DLL* que implementen el control compilat a un directori del projecte i es realitza el procés de registre en la *global assembly cache (GAC)* del *.NET Framework* i es valida l'accés a aquests nous objectes en l'arxiu *web.config* del projecte.

Existeix una classe estàtica que conté varis paràmetres de configuració del projecte. Aquesta és la *CConfig* i amb ella configurem els noms dels directoris, els servidors d'accés a les bases de dades, els noms d'aquestes i altres paràmetres de la plataforma.

A l'apartat d'annexos d'aquesta memòria es poden consultar les guies pràctiques i els exemples de configuració i funcionament de tot el que s'ha introduït en aquest apartat.

Capítol 5. Tests, validacions i manteniment

5.1 Conjunt de proves i resultats obtinguts

Encara que en la fase de codificació es van realitzar constantment proves de *caixa blanca*, aquest apartat tractarà bàsicament de les proves de *caixa negra* realitzades i es documenten tant els resultats esperats com els aconseguits.

Les proves de *caixa blanca* examinen els detalls procedimentals, comproven la lògica interna dels mètodes, el flux de la informació i analitzen l'estat del sistema de forma puntual.

D'altra banda les proves de *caixa negra*, no tenen en compte els detalls de la lògica interna, examinen si les entrades s'accepten de forma adequada i si el resultat obtingut és l'esperat.

Kent Beck realitzava la següent observació allà a l'any 1999⁸:

«When you have a full test suite, a test suite is a collection of tests, and if you run it as often as possible, that it gives you the courage to move forward. Many people fear making a change to their code because they're afraid that they'll break it, but with a full test suite in place if you do break something you know you'll detect it and then fix it.»

Així doncs, seguint els consells d'en Kent, hem elaborat una suite de proves que s'encarrega de fer totes les comprovacions adients que demostren l'assoliment d'objectius inicials i específics de cada mòdul.

Els detalls de cada prova es troben redactats de forma estructurada segons una plantilla que recull la següent informació i que es mostra a continuació:

Prova: <Codi prova> – <Nom de la prova>	
Objectius i casos d'ús:	<Codi dels objectius i casos d'ús referenciats>
Actors que intervenen:	<Actors que intervenen>
Punt de partida:	<Punt de partida>

8 KENT BECK. *Extreme Programming Explained: Embrace Change*. 1999. <<http://books.google.es/books?id=G8EL4H4vf7UC>> (18 de juny de 2009).

Dades d'entrada:	<Dades d'entrada>
Descripció:	<Descripció de la prova>
Resultat esperat:	<Descripció del resultat esperat>
Resultat obtingut:	<Descripció del resultat obtingut>

Fig. 5.1 Model de plantilla que recull dades estructurades d'una prova

La suite de proves la componen 19 proves específiques de caixa negra que supervisen el funcionament global del projecte. Les proves es realitzen des del punt de vista de l'usuari i s'estudia tota la entrada-sortida que aquest observa. Es pot consultar aquest conjunt de proves a la secció 8.1 d'aquesta memòria.

Tal i com s'explica en els detalls de cada una d'aquestes, els resultats han estat els esperats i es compleixen els objectius i casos d'ús que en tenen vinculats a la capçalera.

Encara que aquestes proves s'han realitzat a l'acabament de la fase de desenvolupament, durant la codificació del projecte s'han executat proves per garantir l'assoliment d'objectius específics i globals. Per exemple: s'ha comprovat la correcta integració amb el portal *ripollet.cat* en cada un dels recursos afegits així com ho exigia l'objectiu OBJ_00_01. També s'ha validat el codi HTML produït segons l'especificació *DTD HTML 4.01 Transitional* del W3C segons s'exigia a l'objectiu OBJ_00_02.

Relacionades amb aquestes proves d'àmbit global s'ha comprovat la correcta visualització de la producció HTML en els següents navegadors:

- *Mozilla FireFox 2 i 3, de Mozilla Foundation*
- *Opera 9, d'Opera Software*
- *Safari 3 i 4, d'Apple Inc.*
- *i Internet Explorer 6, 7 i 8, de Microsoft Corp.*

Els ajusts que s'han realitzat per a homogeneïtzar la visualització en tots aquests, han estat aplicats sota la supervisió de la validació W3C.

També s'ha depurat el codi *JavaScript* de les llibreries i scripts insertats a les pàgines amb l'objectiu d'eliminar els *warnings* o errors i satisfer un correcte funcionament global en totes les plataformes comprovades.

Finalment s'ha comprovat l'accessibilitat multiplataforma des dels sistemes operatius següents:

- *GNU-Linux, distribució Fedora 11*
- *Mac OS X, d'Apple Inc.*
- *Windows XP Professional, de Microsoft Corp.*

Totes les proves realitzades han satisfet els objectius globals i específics i això marca la fi del procés de desenvolupament.

5.2 Punts de control

En quant a manteniment del sistema, la metodologia *mètrica* assenyalava quatre tipus específics:

- *Correctiu*: aquells canvis necessaris per a corregir errors del producte software.
- *Perfectiu*: on es porten accions per a millorar la qualitat interna dels sistemes en qualsevol dels seus aspectes: reestructuració del codi, definició més clara del sistema i optimització del rendiment i eficiència.
- *Evolutiu*: són les incorporacions, modificacions i eliminacions necessàries en un producte software per a satisfer l'expansió o canvi en les necessitats de l'usuari.
- *Adaptatiu*: són les modificacions que afecten als entorns en els que el sistema opera, per exemple, canvis de configuració del hardware, software de base, gestors de bases de dades, comunicacions, etc...

En relació amb aquesta introducció teòrica veiem que, dels quatre tipus de manteniment assenyalats, els dos primers es realitzen forçosament en la fase de desenvolupament i sempre abans de l'entrega del producte, mentre que els altres dos es realitzen com a manteniment real del producte ja entregat.

En aquest moment s'haurà de tenir cura llavors de dos possibles fronts que ens obligaran a realitzar aquest tipus de tasques. D'una banda els possibles canvis tècnics que afectin a la plataforma –manteniment adaptatiu– i, d'altra banda, la redacció de nous objectius en quant a afegir noves funcionalitats o serveis –manteniment evolutiu–.

És de sentit comú que els canvis introduïts pels dos fronts comentats en el paràgraf anterior implicaran tasques de manteniment. El que no és tant evident i sí molt important, són els canvis que afectin als productes software de tercers amb els que aquesta plataforma interactua.

Qualsevol canvi de versió dels aplicatius interns de gestió: *Registre, Expedients, Padró Municipal d'Habitants o Ingressos*; o bé de les plataformes de serveis utilitzades: *PSIS-CATCert*, implicarà una revisió del correcte funcionament i, en el cas necessari, aplicar el manteniment adaptatiu adequat. Aquesta revisió haurà de satisfer el joc de proves dissenyat a tal efecte.

D'altra banda la inclusió de noves funcionalitats suposarà una nova redacció d'objectius específica i acompanyada de la seva anàlisi, disseny i codificació. També s'haurà de dissenyar un nou conjunt de proves que mostrin l'assoliment dels nous objectius així com dels existents.

5.3 Informes d'activitat i productivitat

La plataforma de serveis desenvolupada respecta la llei 11/2007 d'Accés electrònic dels ciutadans als serveis públics i la 15/1999 i la llei orgànica 15/1999 de Protecció de Dades de Caràcter Personal.

Amb l'objectiu d'aquest compliment, tota activitat de l'usuari que es deriva de l'ús de la plataforma, queda enregistrada en una de les taules de la bases de dades del projecte.

La taula a la que ens referim és la *tblEventLog* i podem consultar el seu esquema a la figura 3.6 del capítol de disseny. Aquesta taula enregistra els següents tipus d'esdeveniments:

- *INSERT_DOCUMENT*: inserció d'un nou document al registre d'entrada corporatiu.
- *INSERT_PERSONA*: inserció d'una nova persona al nucli de persones.
- *LOGIN*: inici de sessió amb certificat digital processat amb èxit.
- *LOGIN_TRY*: intent d'inici de sessió frustrat.
- *LOGOUT*: tancament de sessió a petició explícita de l'usuari.
- *LOGOUT_TIMEOUT*: tancament de sessió per inactivitat.
- *SELECT_CERCON*: producció del justificant de convivència del *PMH*.
- *SELECT_CERPEM*: producció del justificant d'empadronament del *PMH*.
- *SELECT_DOC*: producció del llistat de documents del registre d'entrada/sortida o bé del justificant d'un d'aquests.
- *SELECT_DOCING*: producció del llistat de documents d'ingressos o bé del justificant d'un d'aquests.
- *SELECT_EXP*: producció del llistat d'expedients o bé del justificant d'un d'aquests.
- *UPLOAD_FILE*: enviament amb èxit d'un fitxer adjunt.

El nemònic –en cursiva– indica el valor que es guarda a la columna *acció* mentre que els detalls de l'esdeveniment s'emmagatzemen a la columna *descripció*.

Amb el tractament d'aquesta informació es poden realitzar informes d'activitat i productivitat amb l'objectiu de detectar possibles anomalies de funcionament o fins i tot accions malicioses amb intents de comprometre la seguretat del sistema o la integritat de les dades. També en podem extreure indicadors estadístics que ajudaran a prendre decisions sobre quins serveis requereixen promoció o d'aquells que més s'en produeixen.

Es recomana la producció i revisió periòdica d'aquests informes per garantir una correcta cobertura dels serveis oferts per la plataforma de serveis desenvolupada.

Capítol 6. Conclusió

6.1 Valoració de l'assoliment d'objectius

La *Plataforma Telemàtica de Serveis Administratius* que en aquest projecte es plantejava, tenia com a objectiu inicial el promoure la utilització dels canals de comunicació segurs entre el ciutadà i l'Administració. La integració amb el portal existent *ripollet.cat* el feia evolucionar d'un portal informatiu a un portal de serveis.

Els beneficis de la implementació havien de ser recíprocs ja que el ciutadà guanya en accessibilitat, disponibilitat i transparència dels serveis de gestió, mentre que la corporació veu com el ciutadà evoluciona de *client* a *col·laborador*, generant una reducció important en les cues d'atenció ciutadana de cada un dels serveis integrats.

L'anàlisi global del problema inicial va donar com a resultat la construcció de tres mòduls que s'identifiquen amb l'acreditació d'usuaris, el servei de registre d'entrada i els serveis informatius. Més tard, la recerca d'objectius durant la fase d'anàlisi ens va proporcionar un catàleg extens de reptes específics que donaven solució, en el seu conjunt, a cada un dels objectius inicials.

S'ha elaborat un conjunt de proves que justifiquen l'assoliment de tots i cadascun d'aquests reptes i el resultat del conjunt és una plataforma que:

- Accepta com acreditació d'usuari els certificats de les agències IDCat, DNle, FNMT i és molt senzill afegir-ne d'altres, gràcies a la integració dels serveis de la plataforma PSIS de CATCert.
- Integra el servei de registre d'entrada de forma completa, generant l'alta de l'interessat en el cas que en sigui necessari, evitant la figura del moderador i generant una entrada directa contra el mateix i únic registre d'entrada corporatiu. No és la implementació d'un registre paral·lel ni una simple comanda de registre. A efectes interns, l'usuari executa el procediment de registre tal i com es processaria des de l'oficina de registre. Aquest servei també inclou la possibilitat d'adjuntar un arxiu amb la documentació annexa en format electrònic. La tramesa es realitza de forma segura i optimitzada. Es calcula el codi HASH MD5 per reforçar la integritat i seguretat dels arxius emmagatzemats al repositori.
- Pel que fa als serveis informatius, s'ha integrat serveis que recuperen les dades de les bases d'emmagatzemament corporatives. La complexitat és mitjana o alta

segons el servei sol·licitat però el resultat és satisfactori en cada cas. La informació es maquetada integrada amb el portal ripollet.cat i viatja fins l'usuari a través de canals segurs. Inclou la possibilitat de generar una versió *printer-friendly* en HTML i la generació dinàmica de contingut PDF signat electrònicament per l'Ajuntament, el que permet l'exportació de la informació fora de la plataforma i amb el reconeixement d'autenticitat.

Personalment, crec que l'execució d'aquest projecte repercuteix beneficiosament. Les tasques d'investigació i documentació han estat molt productives ja que m'he mostrat a mi mateix la capacitat d'autoformació suficient com per tirar endavant el projecte sobre una nova plataforma de desenvolupament amb un nou llenguatge de modelatge i de codificació.

L'únic aspecte negatiu del tot el procés és que hi ha hagut una desviació sobre la programació temporal del primer mòdul, presentada el 16 de gener en l'informe previ. A la tasca d'investigació de la plataforma acreditativa, vaig iniciar una negociació amb CATCert sol·licitant ser integrador de la seva plataforma PSIS per poder gaudir dels seus beneficis. La latència d'aquesta gestió va afectar proporcionalment l'inici de la fase de codificació i proves d'aquest primer mòdul que va començar dues setmanes més tard de la data prevista. Segons l'informe previ, abans d'analitzar, dissenyar i desenvolupar res, es programava com la data d'acabament el dia 15 de maig.

Finalment, el dia 21 de maig, es fa la presentació formal demostrant la recuperació de la desviació comentada. La solució implementada satisfà tant els objectius inicials com els que van anant apareixent durant el decurs del projecte, i així ho van considerar en majoria absoluta els màxims promotors del projecte a l'Ajuntament decidint com a prioritària la tasca de posar-ho en producció. Tots i cada un d'ells, des de l'excel·lentíssim senyor Alcalde, el regidor de Noves Tecnologies i Comunicació, el gerent del Patronat Municipal de Comunicació, el cap de Recursos Humans –tutor del projecte–, el cap del Serveis Informàtics i el responsable de l'Oficina d'Atenció al Ciutadà, sentenciaren la utilitat de la plataforma.

6.2 Vies de creixement i viabilitat

La implementació de la plataforma descrita en aquest projecte es tracta d'una primera fase. Com a projecte final de carrera, seria massa optimista intentar desenvolupar una plataforma completa amb tots els serveis que una administració pública ofereix i que aquests estiguessin completament automatitzats. Entenem que un projecte d'aquesta naturalesa ha de créixer en funció de les necessitats, afegint serveis i funcionalitats en etapes, sempre i quan s'hagi establert un equilibri entre aquestes i els recursos disponibles.

Molts dels serveis que es poden afegir requereixen la col·laboració del personal tècnic i funcionari de l'Ajuntament. La integració d'aquests comportaria una reestructuració organitzativa per a atendre a les noves peticions i aquest és el motiu que justifica una implantació en fases o etapes.

Tenint això en compte, durant l'execució del projecte i en les típiques sessions de *brainstorming* que de vegades succeeixen al final de les reunions de seguiment, han anat sorgit idees que van ser anotades i que ara recollim en aquest apartat:

- *Nous serveis*: la implementació actual recull els serveis que més es demanen segons l'estadística de l'Oficina d'Atenció al Ciutadà. No obstant això, hi ha molts altres serveis de consulta encara per implementar.
- *Validació de justificants*: actualment la plataforma genera documents signats electrònicament amb un certificat digital. Aquest procés de signatura justifica la veracitat i la no alteració de les dades generades per l'Ajuntament dins el medi electrònic, però quan l'interessat fa la impressió del document, la signatura electrònica es perd i el document en paper perd la credibilitat. Per solucionar aquest problema es pot afegir al contingut del document, un peu de pàgina amb les dades d'accés a un servei de validació de documents. Aquestes dades serien la *URL* del servei, el codi del document generat i una clau *hash* d'aquest. Amb aquestes dades el servei de validació identificaria el document generat i el presentaria en pantalla sense la necessitat de l'acreditació d'usuari de manera que qualsevol persona que dubtés de l'autenticitat del document en pogués realitzar la validació.
- *Pagaments en línia*: un dels serveis informatius integrats ha estat la consulta de pagaments. Una millora relacionada amb aquest servei seria la detecció dels rebuts pendents de pagament i en període de cobrament voluntari, extreure les dades del quadern 60 de comunicació interbancària i fer l'enllaç amb la passarel·la de pagament adequada. Això facilitaria la gestió de l'usuari i l'Ajuntament tindria un nou indicador de previsió d'ingressos.
- *Carpeta del ciutadà*: hem comentat que la plataforma genera *justificants* o *volants*. No és possible la generació de *certificats* ja que, per definició administrativa, un *certificat* ha de tenir estampades les signatures de l'excel·lentíssim senyor Alcalde i del senyor Secretari de la corporació. Actualment la generació de certificats representa menys d'un 1% de les peticions i la seva generació evidentment no és immediata ni gratuïta. Per donar solució a aquest i altres serveis que impliquen el seguiment d'un procediment administratiu, s'ha pensat en crear una interfície documental anomenada *carpeta del ciutadà* semblant a una safata de correu a on l'usuari recuperaria els documents generats durant l'execució del procediment abans comentat. Per exemple per integrar els serveis de certificats hauríem d'implementar un primer servei de sol·licitud on es processaria l'alta de la petició, es proposaria la liquidació de les taxes i després el servei enllaçaria la petició amb la eina de *workflow* per a la generació del certificat. Un cop generat el document electrònic amb les signatures requerides, es notificaria a l'interessat de la seva disponibilitat a la seva *carpeta del ciutadà*.
- *Carpeta del funcionari*: de forma anàloga al servei anterior, aquest seria una abstracció del sistema de *workflow* aplicant els filtres adequats per a que cada funcionari o departament pogués consultar la càrrega de peticions realitzades des de la plataforma telemàtica. Seguint amb l'exemple anterior, en el cas de la

sol·licitud d'un certificat qualsevol, les persones que l'han de signar veurien a la seva carpeta el document pendent de la seva signatura i el sistema portaria el control de les signatures pendents abans de publicar el document en la carpeta del ciutadà corresponent.

- *Interoperabilitat*: tal i com ho recull la llei 11/2007 no només els ciutadans són els que haurien d'accedir als serveis públics de forma telemàtica. També altres administracions hauran de poder-ho realitzar. Aquest detall ens obliga a estar alerta en el cas que es defineixin protocols estàndards d'intercomunicació entre plataformes. Per a aconseguir-ho s'haurà de dissenyar i integrar les interfícies d'accés adequades que implementin la gestió d'aquests futurs protocols.

La redacció d'aquests nous serveis i la seva complexitat fa comprendre l'estructuració en fases d'aquest projecte. De moment en tenim una plataforma funcional que integra els serveis més demanats. Ara, està a les mans de l'Ajuntament, l'assignació de recursos amb l'objectiu de desenvolupar les fases posteriors amb les noves idees que en aquest apartat es proposen.

Capítol 7. Bibliografia i referències

7.1 Bibliografia

SIMON BENNETT, STEVE McROBB I RAY FARMER. *Análisis y diseño orientado a objetos de sistemas usando UML*. Traducción de Jorge Rodríguez Vega y Luis Rodríguez Martín. Tercera edición. McGraw-Hill, 2006. [ISBN: 978-84-481-5640-4]

JAY HILYARD & STEPHEN TEILHET. *C# 3.0 Cookbook*. Third edition. O'Reilly Media, Inc. 2008. [ISBN: 978-0-596-51610-9]

CATCERT, AGÈNCIA CATALANA DE CERTIFICACIÓ. *Guia bàsica per als integradors de PSIS*. Agència Catalana de Certificació, 2008. [Referència: D1220, Versió: 16, Data: 9/6/2008]

CRAIG LARMAN. *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Traducción de Begoña Moros Valle y Jesús García Molina. Segunda edición. Prentice Hall, Pearson Educación S.A. 2006. [ISBN: 978-84-205-3438-1]

PERDITA STEVENS & ROB POOLEY. *Utilización de UML en Ingeniería del Software con Objetos y Componentes*. Traducción de Marta Fernández Alarcón y Rubén González Crespo. Segunda edición. Addison Wesley, Pearson Educación S.A. 2006. [ISBN: 978-84-7829-086-4]

JUDITH BISHOP. *C# 3.0 Design Patterns*. O'Reilly Media, Inc. 2007. [ISBN: 978-0-596-52773-0]

JOSEP LLADÓS I F. XAVIER ROCA. *Problemes d'enginyeria del software I*. Servei de Publicacions de la UAB. 1995. [ISBN: 84-490-0465-9]

E. S. TAYLOR, *An Interim Report on Engineering Design*. 1959. Technical report. Massachusetts Institute of Technology. Cambridge, MA.

LLEI 11/2007, de 22 de juny, d'accés electrònic dels ciutadans als serveis públics. [BOE 150, de 23-6-2007]

LLEI ORGÀNICA 15/1999, de 13 de desembre, de Protecció de Dades de Caràcter Personal. [BOE 298, de 14-12-1999]

7.2 Enllaços

TIM O'REILLY. *Web 2.0: Compact Definition?* 2005. <<http://radar.oreilly.com/archives/2005/10/web-20-compact-definition.html>> (27 de maig de 2009).

CATCERT. *Lista d'entitats classificades*. 2009.
<http://catcert.cat/web/cat/2_1_0_entitats_certificacio.jsp>
(5 de juny de 2009).

TRYGVE REENSKAUG. *Model-View-Controller (MVC)*. 1979.
<<http://heim.ifi.uio.no/~trygver/>>
(15 de juny de 2009).

KENT BECK. *Extreme Programming Explained: Embrace Change*. 1999.
<<http://books.google.es/books?id=G8EL4H4vf7UC>>
(18 de juny de 2009).

R. HOUSLEY, W. POLK, W. FORD & D. SOLO. *Internet X.509 Public Key Infrastructure. Certificate and Certificate Revocation List (CRL) Profile*.
The Internet Society. 2002.
<<http://www.ietf.org/rfc/rfc3280.txt>>
(18 de juny de 2009).

FABRICA NACIONAL DE MONEDA Y TIMBRE. *Declaración de prácticas de certificación*.
Real Casa de la Moneda. 2004.
<http://www.cert.fnmt.es/content/pages_std/docs/DPC_13_octubre-2004.pdf>
(18 de juny de 2009).

JOHN LAMBERT. *Introducing CAPICOM*. Microsoft Developer Network.
Microsoft Corporation. 2001.
<<http://msdn.microsoft.com/en-us/library/ms995332.aspx>>
(18 de juny de 2009).

MANUEL CARDENAS THORLUND. *Usar certificados de cliente de la FNMT en una Aplicación Web*.
N-Idea, nuevas ideas. 2008.
<<http://nidea-soluciones.blogspot.com/2008/01/usar-certificados-de-cliente-en-una.html>>
(18 de juny de 2009).

MICROSOFT DEVELOPER NETWORK. *X509Certificate (Clase)*.
Microsoft Corporation. 2009.
<[http://msdn.microsoft.com/es-es/library/system.security.cryptography.x509certificates.x509certificate\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/system.security.cryptography.x509certificates.x509certificate(VS.80).aspx)>
(18 de juny de 2009).

MICROSOFT DEVELOPER NETWORK. *MD5CryptoServiceProvider (Clase)*.
Microsoft Corporation. 2009.
<<http://msdn.microsoft.com/es-es/library/system.security.cryptography.md5cryptoserviceprovider.aspx>>
(18 de juny de 2009).

7.3 Programari

MICROSOFT CORPORATION. *Visual Web Developer 2008 Express Edition*. 2008.
<<http://www.microsoft.com/express/download/default.aspx>>
(18 de juny de 2009).

BRETTLE DEVELOPMENT. *NeatUpload™ ASP.NET Component*. 2009.
<<http://www.brettle.com/neatupload>>
(15 de juny de 2009).

WEBSUPERGOO. *ABCpdf .NET™ 7.0*. 2009.

<<http://www.websupergoo.com/products.htm#pd>>
(15 de juny de 2009).

CATCERT. AGÈNCIA CATALANA DE CERTIFICACIO. *Plataforma de serveis d'identificació i signatura (PSIS)*. 2009.

<http://catcert.cat/web/cat/1_4_3_plataforma.jsp>
(15 de juny de 2009).

SUN MICROSYSTEMS INC. *OpenOffice – The free and open productivity suite*. 2009.
[GNU Lesser General Public License]

<<http://es.openoffice.org/>>
(18 de juny de 2009).

StarUML – The Open Source UML/MDA Platform. 2005.
[GNU Public License]

<<http://staruml.sourceforge.net/en/index.php>>
(18 de juny de 2009).

DMITRY BARASHEV & ALEXANDRE THOMAS. *GanttProject – Free tool for project scheduling and management*. 2009.

[GNU Public License]
<<http://www.ganttproject.biz/>>
(15 de juny de 2009).

FABULOUS FORCE. *DBDesigner 4*. 2003.

[GNU Public License]
<<http://fabforce.net/dbdesigner4/index.php>>
(15 de juny de 2009).

Capítol 8. Annexos

8.1 Conjunt de test realitzats

A continuació el conjunt de proves dissenyat per a la comprovació de l'assoliment del llistat d'objectius d'aquest projecte:

Prova: PR_01 – <i>Benvinguda a la plataforma</i>	
Objectius i casos d'ús:	OBJ_00_01, OBJ_00_02 i OBJ_00_03
Actors que intervenen:	L'usuari i el servidor
Punt de partida:	Punt de partida inicial.
Dades d'entrada:	La URL de la plataforma < http://ripollet.cat/aspnet/ptsa/dologin.cs.aspx >
Descripció:	L'usuari sol·licita la pantalla de benvinguda a la plataforma utilitzant la URL específica al seu programa de navegació. Es comprovarà la correcta visualització i càrrega d'aquesta pantalla.
Resultat esperat:	La pantalla de benvinguda carrega a través de http, es troba completament integrada en estructura i estil amb el portal <i>ripollet.cat</i> .
Resultat obtingut:	S'obtenen els resultats esperats. Es satisfan els objectius relacionats a la capçalera.

Prova: PR_02 – <i>Inici de sessió amb certificat d'IDCat</i>	
Objectius i casos d'ús:	OBJ_00_01, OBJ_00_02 i OBJ_00_03 OBJ_01_01, OBJ_01_02, OBJ_01_03, OBJ_01_04 i OBJ_01_05
Actors que intervenen:	L'usuari, el servidor i <i>PSIS-CATCert</i>
Punt de partida:	Pantalla de benvinguda.
Dades d'entrada:	Disposar d'un certificat digital d'IDCat.
Descripció:	Una vegada l'usuari fa clic al botó <i>inici de sessió</i> , se'l demanarà l'entrega d'un certificat digital d'usuari per acreditar-se. En aquest llistat apareixen els certificats configurats al navegador i compatibles amb l'aplicació. En aquest prova s'ha d'entregar un de <i>IDCat</i> .
Resultat esperat:	La petició del certificat d'usuari. La pantalla resultat de la validació d'aquest.
Resultat obtingut:	S'obtenen els resultats esperats. Es satisfan els objectius relacionats a la capçalera.

Prova: PR_03 – Inici de sessió amb certificat d'DNI-e	
Objectius i casos d'ús:	OBJ_00_01, OBJ_00_02 i OBJ_00_03 OBJ_01_01, OBJ_01_02, OBJ_01_03, OBJ_01_04 i OBJ_01_05
Actors que intervenen:	L'usuari, el servidor i <i>PSIS-CATCert</i>
Punt de partida:	Pantalla de benvinguda.
Dades d'entrada:	Disposar d'un DNI electrònic i un lector de tarja criptogràfica compatible.
Descripció:	Una vegada l'usuari fa clic al botó <i>inici de sessió</i> , se'l demanarà l'entrega d'un certificat digital d'usuari per acreditar-se. En aquest llistat apareixen els certificats configurats al navegador i compatibles amb l'aplicació. En aquest prova s'ha d'entregar el certificat d'autenticació del <i>DN/e</i> .
Resultat esperat:	La petició del certificat d'usuari. La pantalla resultat de la validació d'aquest.
Resultat obtingut:	S'obtenen els resultats esperats. Es satisfan els objectius relacionats a la capçalera.

Prova: PR_04 – Visualització del llistat de serveis	
Objectius i casos d'ús:	OBJ_00_01, OBJ_00_02, OBJ_00_03 i OBJ_00_04 OBJ_01_01, OBJ_02_02, OBJ_00_03, OBJ_01_04 i OBJ_01_05
Actors que intervenen:	L'usuari i el servidor
Punt de partida:	Pantalla informativa de resultat de procés acreditatiu.
Dades d'entrada:	L'usuari ha d'estar acreditat correctament
Descripció:	L'usuari fa clic al botó <i>iniciar gestió</i>
Resultat esperat:	Es carrega la pàgina que llista els serveis de la plataforma.
Resultat obtingut:	S'obtenen els resultats esperats. Es satisfan els objectius relacionats a la capçalera.

Prova: PR_05 – Realitzar un registre d'entrada	
Objectius i casos d'ús:	OBJ_00_01, OBJ_00_02, OBJ_00_03 i OBJ_00_04 OBJ_01_04 i OBJ_01_05 OBJ_02_01, OBJ_02_03, OBJ_02_04, OBJ_02_05, OBJ_02_06 i OBJ_02_07
Actors que intervenen:	L'usuari i el servidor
Punt de partida:	Pantalla llistat de serveis de la plataforma
Dades d'entrada:	L'usuari ha d'estar acreditat correctament i figura com interessat a la base de dades de registre
Descripció:	L'usuari selecciona el servei de registre d'entrada. L'usuari es troba com a interessat a la base de dades i no ha de fer-ne l'alta. Emplena el formulari de registre que se'l mostra i fa clic a <i>Enviar</i> .
Resultat esperat:	Després de la validació <i>JavaScript</i> si tot és correcte el servidor processa la inserció del nou document a la base de dades de registre segons una operació transaccional i informa del resultat de l'operació amb una pantalla informativa. Se'l permet adjuntar un arxiu.
Resultat obtingut:	S'obtenen els resultats esperats. Es satisfan els objectius relacionats a la capçalera.

Prova: PR_06 – Adjuntar arxiu al registre d'entrada	
Objectius i casos d'ús:	OBJ_00_01, OBJ_00_02, OBJ_00_03 i OBJ_00_04 OBJ_01_04 i OBJ_01_05 OBJ_02_07, OBJ_02_08, OBJ_02_09, OBJ_02_10, OBJ_02_11, OBJ_02_12, OBJ_02_13 i OBJ_02_14
Actors que intervenen:	L'usuari i el servidor
Punt de partida:	Pantalla informativa del resultat de registre
Dades d'entrada:	L'arxiu que es vol adjuntar
Descripció:	L'usuari fa clic al botó <i>Adjuntar arxiu</i> , llavors es visualitzen els controls que permeten seleccionar l'arxiu desitjat, el botó per començar la tramesa i la barra de progrés d'aquesta. L'usuari ha de seleccionar l'arxiu i clicar el botó <i>Enviar</i> .
Resultat esperat:	Mentre l'arxiu arriba al servidor l'usuari visualitza una barra de progrés amb possibilitat d'aturar la càrrega. Un cop que ha arribat s'informa del procés.
Resultat obtingut:	S'obtenen els resultats esperats. Es satisfan els objectius relacionats a la capçalera.

Prova: PR_07– Donar-se d'alta com interessat	
Objectius i casos d'ús:	OBJ_00_01, OBJ_00_02, OBJ_00_03 i OBJ_00_04 OBJ_01_04 i OBJ_01_05 OBJ_02_02
Actors que intervenen:	L'usuari i el servidor
Punt de partida:	Pantalla llistat de serveis de la plataforma
Dades d'entrada:	L'usuari ha d'estar acreditat correctament però no figura com interessat a la base de dades de registre
Descripció:	L'usuari intenta accedir al servei de registre d'entrada però no es troba com a interessat a la base de dades.
Resultat esperat:	La plataforma avisa d'aquesta situació i el convida a donar-se d'alta al sistema. Visualitza el formulari d'alta i, un cop s'ha emplenat i enviat, la plataforma informa del resultat de l'operació.
Resultat obtingut:	S'obtenen els resultats esperats. Es satisfan els objectius relacionats a la capçalera.

Prova: PR_08 – Consulta del registre d'entrada/sortida	
Objectius i casos d'ús:	OBJ_00_01, OBJ_00_02, OBJ_00_03 i OBJ_00_04 OBJ_01_04 i OBJ_01_05 OBJ_03_01, OBJ_03_02, OBJ_03_03 i OBJ_03_06
Actors que intervenen:	L'usuari i el servidor
Punt de partida:	Pantalla llistat de serveis de la plataforma o la pantalla amb el llistat de documents
Dades d'entrada:	Usuari acreditat i correspondència a la taula d'interessats
Descripció:	Comprovar llistat i detall de la consulta contra el registre d'entrada. L'usuari ha de ser capaç de veure el llistat dels seus registres d'entrada i de sortida i de veure'n els detalls.
Resultat esperat:	La primera pantalla serà una taula amb el llistat dels documents

	d'entrada i sortida vinculats amb l'usuari acreditat. Al fer clic sobre la clau de qualsevol document es consultarà i es mostrarà en detall.
Resultat obtingut:	S'obtenen els resultats esperats. Es satisfan els objectius relacionats a la capçalera.

Prova: PR_09 – Generar justificant d'empadronament	
Objectius i casos d'ús:	OBJ_00_01, OBJ_00_02, OBJ_00_03 i OBJ_00_04 OBJ_01_04 i OBJ_01_05 OBJ_03_01, OBJ_03_02, OBJ_03_03 i OBJ_03_06
Actors que intervenen:	L'usuari i el servidor
Punt de partida:	Pantalla llistat de serveis de la plataforma
Dades d'entrada:	Usuari acreditat i correspondència a la taula d'empadronats
Descripció:	L'usuari sol·licita el servei <i>justificat d'empadronament</i>
Resultat esperat:	Rep la informació del justificant sol·licitat.
Resultat obtingut:	S'obtenen els resultats esperats. Es satisfan els objectius relacionats a la capçalera.

Prova: PR_10 – Generar justificant de convivència	
Objectius i casos d'ús:	OBJ_00_01, OBJ_00_02, OBJ_00_03 i OBJ_00_04 OBJ_01_04 i OBJ_01_05 OBJ_03_01, OBJ_03_02, OBJ_03_03 i OBJ_03_06
Actors que intervenen:	L'usuari i el servidor
Punt de partida:	Pantalla llistat de serveis de la plataforma
Dades d'entrada:	Usuari acreditat i correspondència a la taula d'empadronats
Descripció:	L'usuari sol·licita el servei <i>justificat de convivència</i> .
Resultat esperat:	Rep la informació del justificant sol·licitat.
Resultat obtingut:	S'obtenen els resultats esperats. Es satisfan els objectius relacionats a la capçalera.

Prova: PR_11 – Consulta d'expedients	
Objectius i casos d'ús:	OBJ_00_01, OBJ_00_02, OBJ_00_03 i OBJ_00_04 OBJ_01_04 i OBJ_01_05 OBJ_03_01, OBJ_03_02, OBJ_03_03 i OBJ_03_06
Actors que intervenen:	L'usuari i el servidor
Punt de partida:	Pantalla llistat de serveis de la plataforma
Dades d'entrada:	Usuari acreditat i correspondència a la taula d'interessats
Descripció:	Comprovar llistat i detall de la consulta contra els expedients. L'usuari ha de ser capaç de veure el llistat dels expedients on figura com interessat i de veure'n els detalls.
Resultat esperat:	La primera pantalla serà una taula amb el llistat dels expedients relacionats amb l'usuari acreditat. Al fer clic sobre el número de qualsevol expedient es consultarà i es mostrarà en detall.
Resultat obtingut:	S'obtenen els resultats esperats. Es satisfan els objectius relacionats a la capçalera.

Prova: PR_12 – Consulta de pagaments	
Objectius i casos d'ús:	OBJ_00_01, OBJ_00_02, OBJ_00_03 i OBJ_00_04 OBJ_01_04 i OBJ_01_05 OBJ_03_01, OBJ_03_02, OBJ_03_03 i OBJ_03_06
Actors que intervenen:	L'usuari i el servidor
Punt de partida:	Pantalla llistat de serveis de la plataforma
Dades d'entrada:	Usuari acreditat i correspondència a la taula de contribuents
Descripció:	Comprovar llistat i detall de la consulta contra la base de dades de pagaments. L'usuari ha de ser capaç de veure el llistat dels seus documents d'ingressos i de veure'n els detalls.
Resultat esperat:	La primera pantalla serà una taula amb el llistat dels documents d'ingressos relacionats amb l'usuari acreditat. Al fer clic sobre el número de qualsevol document es consultarà i es mostrarà en detall.
Resultat obtingut:	S'obtenen els resultats esperats. Es satisfan els objectius relacionats a la capçalera.

Prova: PR_13 – Generació del document en HTML	
Objectius i casos d'ús:	OBJ_00_03 OBJ_03_03 i OBJ_03_06
Actors que intervenen:	L'usuari i el servidor
Punt de partida:	La visualització d'un justificant qualsevol o fins i tot un llistat de registres.
Dades d'entrada:	Usuari acreditat i correspondència amb interessat, empadronat o contribuent, segons el document en qüestió.
Descripció:	L'usuari sol·licita la versió HTML de qualsevol document o llistat generat per la plataforma. Fa clic al botó <i>HTML</i> .
Resultat esperat:	S'obre una finestra nova del navegador i aquesta visualitza per https única i exclusivament la informació del document triat.
Resultat obtingut:	S'obtenen els resultats esperats. Es satisfan els objectius relacionats a la capçalera.

Prova: PR_14 – Generació del document en PDF signat	
Objectius i casos d'ús:	OBJ_00_03 OBJ_03_05 i OBJ_03_06
Actors que intervenen:	L'usuari i el servidor
Punt de partida:	La visualització d'un justificant qualsevol o fins i tot un llistat de registres.
Dades d'entrada:	Usuari acreditat i correspondència amb interessat, empadronat o contribuent, segons el document en qüestió.
Descripció:	L'usuari sol·licita la versió PDF de qualsevol document o llistat generat per la plataforma. Fa clic al botó <i>PDF</i> .
Resultat esperat:	S'obre una finestra nova del navegador i aquesta envia per https un arxiu pdf generat dinàmicament amb única i exclusivament la informació del document triat. Aquest document inclou una signatura electrònica de l'Ajuntament de Ripollet.
Resultat obtingut:	S'obtenen els resultats esperats.

	Es satisfan els objectius relacionats a la capçalera.
--	---

Prova: PR_15 – Tancament de sessió a petició de l'usuari	
Objectius i casos d'ús:	OBJ_00_01, OBJ_00_02, OBJ_00_03 i OBJ_00_04 OBJ_01_04 i OBJ_01_05
Actors que intervenen:	L'usuari i el servidor.
Punt de partida:	Qualsevol pantalla de la plataforma un cop iniciada la sessió.
Dades d'entrada:	Usuari acreditat.
Descripció:	L'usuari sol·licita voluntàriament el tancament de sessió de la plataforma fent clic al botó <i>Desconnectar-se</i> .
Resultat esperat:	La sessió es tanca i s'avisava del resultat de l'operació. Es produeix un esdeveniment <i>LOGOUT</i> que s'anota a la base de dades.
Resultat obtingut:	S'obtenen els resultats esperats. Es satisfan els objectius relacionats a la capçalera.

Prova: PR_16– Tancament de sessió per inactivitat	
Objectius i casos d'ús:	OBJ_00_01, OBJ_00_02, OBJ_00_03 i OBJ_00_04 OBJ_01_04
Actors que intervenen:	L'usuari i el servidor.
Punt de partida:	Qualsevol pantalla de la plataforma un cop iniciada la sessió.
Dades d'entrada:	Usuari acreditat.
Descripció:	L'usuari no interacciona amb la plataforma durant el temps màxim configurat per a la sessió.
Resultat esperat:	La sessió es tanca automàticament segons un redireccionament <i>JavaScript</i> i s'avisava del resultat de l'operació. Es produeix un esdeveniment <i>LOGOUT_TIMEOUT</i> que s'anota a la base de dades.
Resultat obtingut:	S'obtenen els resultats esperats. Es satisfan els objectius relacionats a la capçalera.

Prova: PR_17 – Intent de inserció de duplicats	
Objectius i casos d'ús:	OBJ_00_01, OBJ_00_02, OBJ_00_03 i OBJ_00_04 OBJ_01_04 i OBJ_01_05
Actors que intervenen:	L'usuari i el servidor
Punt de partida:	Les pantalles que informen resultats de les operacions quan es fa <i>alta al registre d'entrada, alta d'interessat</i> o bé quan s'informa del procés de càrrega del fitxer adjunt.
Dades d'entrada:	Usuari acreditat i el nou objecte generat i inserit
Descripció:	L'usuari intenta comprometre la congruència de les dades intentant inserir duplicats tornant enrere al navegador o recarregant la pàgina del procés d'escriptura a la base de dades.
Resultat esperat:	S'espera la no inserció del duplicat i una pantalla informant de l'acció.
Resultat obtingut:	S'obtenen els resultats esperats. Es satisfan els objectius relacionats a la capçalera.

Prova: PR_18 – Intent d'accés a dades prohibides	
Objectius i casos d'ús:	OBJ_00_01, OBJ_00_02, OBJ_00_03 i OBJ_00_04 OBJ_01_04 i OBJ_01_05
Actors que intervenen:	L'usuari i el servidor
Punt de partida:	Les pantalles que informen amb una taula, tots els registres al que l'usuari en té accés. Són: el llistat de registre d'entrada/sortida, el llistat d'expedients i el llistat de documents d'ingressos.
Dades d'entrada:	Usuari acreditat i una URL amb un paràmetre <i>GET</i> amb un document prohibit per l'usuari.
Descripció:	L'usuari intenta accedir a dades prohibides, no vinculades amb ell, canviant el paràmetre <i>GET</i> de la URL que dona accés a un dels seus documents.
Resultat esperat:	Es mostra un pantalla amb un missatge de prohibició d'accés.
Resultat obtingut:	S'obtenen els resultats esperats. Es satisfan els objectius relacionats a la capçalera.

Prova: PR_19 – Revisió de les insercions a les bases de dades	
Objectius i casos d'ús:	OBJ_02_06, OBJ_02_13, OBJ_03_06
Actors que intervenen:	L'administrador de bases de dades i el servidor
Punt de partida:	Iniciar sessió amb usuari amb permisos de lectura contra la base de dades del projecte i les relacionades.
Dades d'entrada:	Acreditació amb usuari de la base de dades des d'un programa visualitzador d'aquesta.
Descripció:	L'administrador de bases de dades ha de comprovar el correcte registre contra les taules següents: <ul style="list-style-type: none"> • PTSA.tblEventLog • PTSA.tblUploadFiles • RESDAR.tblDocuments • NUCLI.tblPersones
Resultat esperat:	Cada un dels processos fa la inserció sense que es produeixen excepcions. Els registres són inserits correctament. No es compromet la integritat i congruència de les dades existents.
Resultat obtingut:	S'obtenen els resultats esperats. Es satisfan els objectius relacionats a la capçalera.

8.2 Pantalles de la implementació

A continuació tot un seguit de pantalles generades com a interfície d'usuari. La majoria d'elles integrades amb el portal web existent *ripollet.cat*.



Fig. 8.2.1 Pantalla inicial. DoLogin

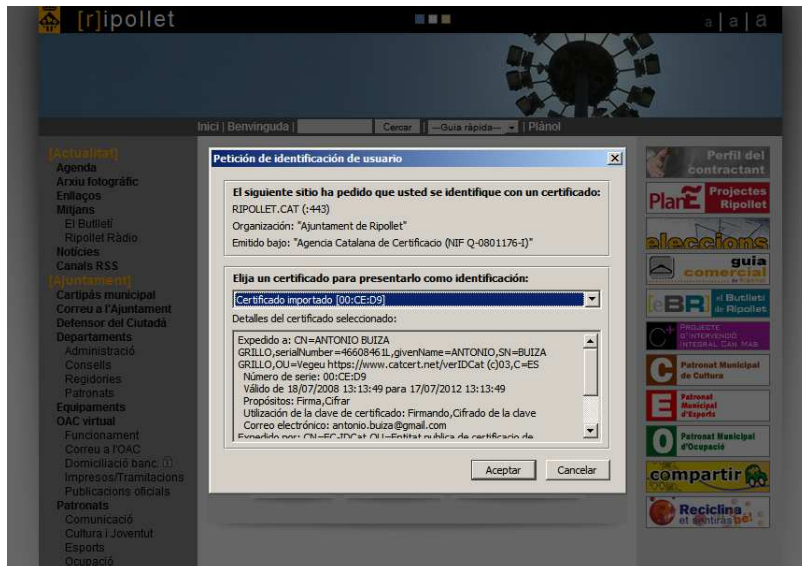


Fig. 8.2.2 Pantalla de selecció de certificat en Mozilla FireFox



Fig. 8.2.3 Bloqueig de la pantalla mentre es valida el certificat

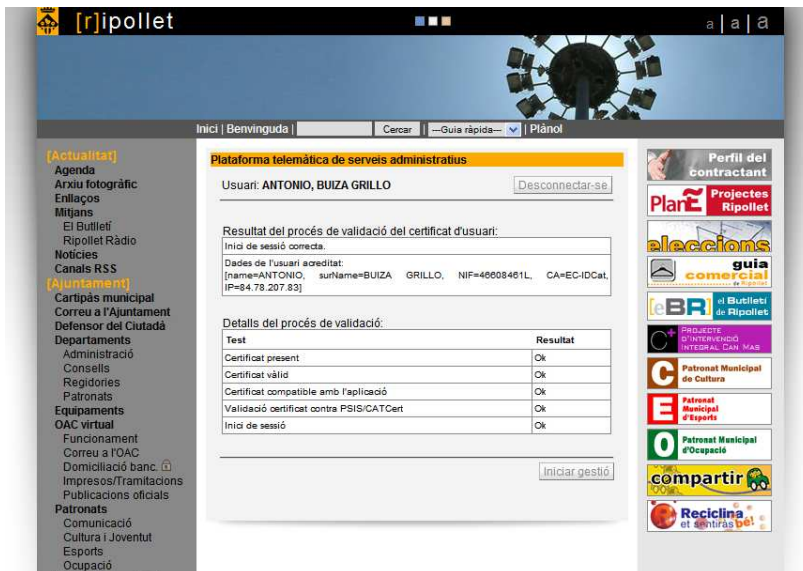


Fig. 8.2.4 Pantalla resultat de la validació. DoCertifiedLogin

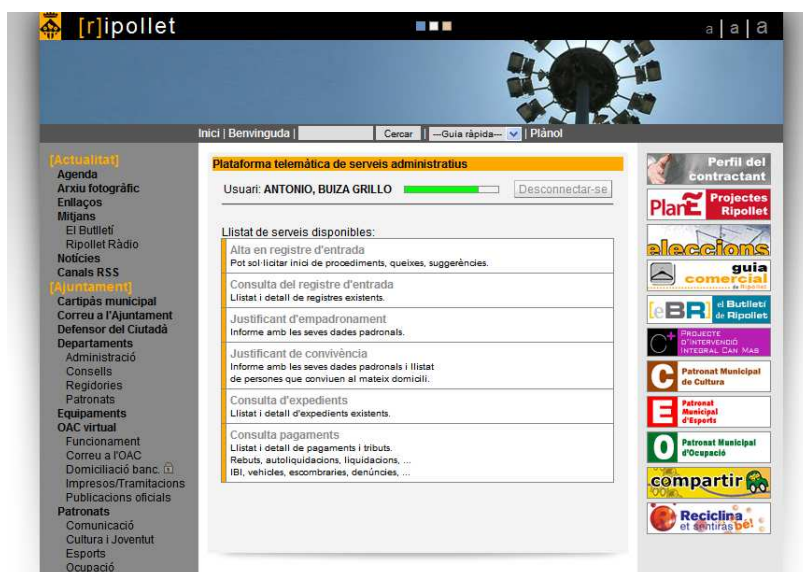


Fig. 8.2.5 Pantalla llistat de serveis. doLlistaServeis

Plataforma telemàtica de serveis administratius

Usuari: ANTONIO, BUIZA GRILLO [Desconnectar-se](#)

El servei sol·licitat requereix l'alta de les seves dades a la base de dades d'interessats doncs no s'ha trobat correspondència fent recerca pel seu NIF.

Si us plau, revisi i complemti les següents dades per processar l'alta telemàtica:

Ajuntament de Ripollet

Alta d'interessat

NIF: 46608461L Nom: ANTONIO Cognom 1: BUIZA Cognom 2: GRILLO

Data naixement (dd/mm/aaaa): Sexe: ---Selecció--- Idioma: ---Selecció---

Correu electrònic: Mòbil:

[Enviar](#) [Esborrar](#)

Fig. 8.2.6 Formulari d'alta d'interessat. doFormPersona

Plataforma telemàtica de serveis administratius

Usuari: ANTONIO, BUIZA GRILLO [Desconnectar-se](#)

Procés d'alta d'interessat completat.
Resultat de l'operació:

- Persona introduïda amb codi 58786.
- Número d'operació: O2009000878

[Continuar](#)

Fig. 8.2.7 Pantalla resultat del procés d'alta d'interessat. doAltaInteressat

Plataforma telemàtica de serveis administratius

Usuari: ANTONIO, BUIZA GRILLO [Desconnectar-se](#)

Ajuntament de Ripollet **Registre d'entrada telemàtic.**

Nom i cognoms: ANTONIO, BUIZA GRILLO NIF: 46608461L

Domicili: -- Selecció --

Telèfon: FAX: Mòbil:

Correu electrònic:

Exposició de fets i petició que es formula:

Registre o expedient relacionat:

Nota: després del procés de registre se l'informarà del número de registre/operació i se li proposarà afegir un anxiu amb la documentació annexa a la seva petició.

[Enviar](#) [Cancelar](#)

Fig. 8.2.8 Formulari d'alta de registre d'entrada. doFormRegistreEntrada

Plataforma telemàtica de serveis administratius

Usuari: ANTONIO, BUIZA GRILLO [Desconnectar-se](#)

Procés de registre d'entrada completat.
Resultat de l'operació:

- Registre processat amb èxit. Registre d'entrada: E2009003778
- Número d'operació: O2009000876

En cas necessari, pot adjuntar un anxiu al seu registre. Si en necessita adjuntar més d'un document, faci servir alguna utilitat de compressió/empaquetat.

[Adjuntar anxiu](#) [Veure registre](#) [Inici](#)

Fig. 8.2.9 Pantalla resultat del procés d'alta de registre. doAltaRegistreEntrada

Plataforma telemàtica de serveis administratius

Usuari: ANTONIO, BUIZA GRILLO [Desconnectar-se](#)

Procés de registre d'entrada completat.
Resultat de l'operació:

- Registre processat amb èxit. Registre d'entrada: E2009003778
- Número d'operació: O2009000876

En cas necessari, pot adjuntar un anxiu al seu registre. Si en necessita adjuntar més d'un document, faci servir alguna utilitat de compressió/empaquetat.

Arxiu: C:\Documents and Settings\Buiza\Escritori\PFC-1079 - MEM... [Examinar](#)

Progrés: 154,0/578,4 KB (27%) @ 17,3 KB/s - Resta 00:15 [Enviar](#)

[Veure registre](#) [Inici](#)

Fig.8.2.10 Integració del control NeatUpload per a la tramesa d'arxiu adjunt

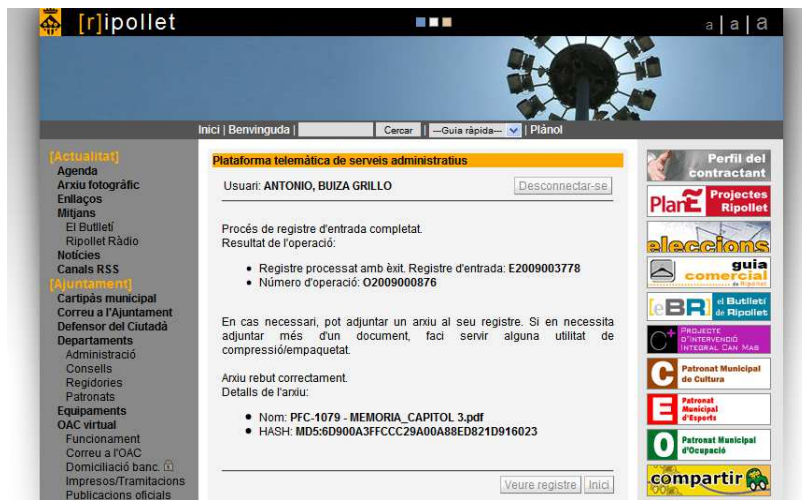


Fig. 8.2.11 Informació del procés de tramesa d'arxiu

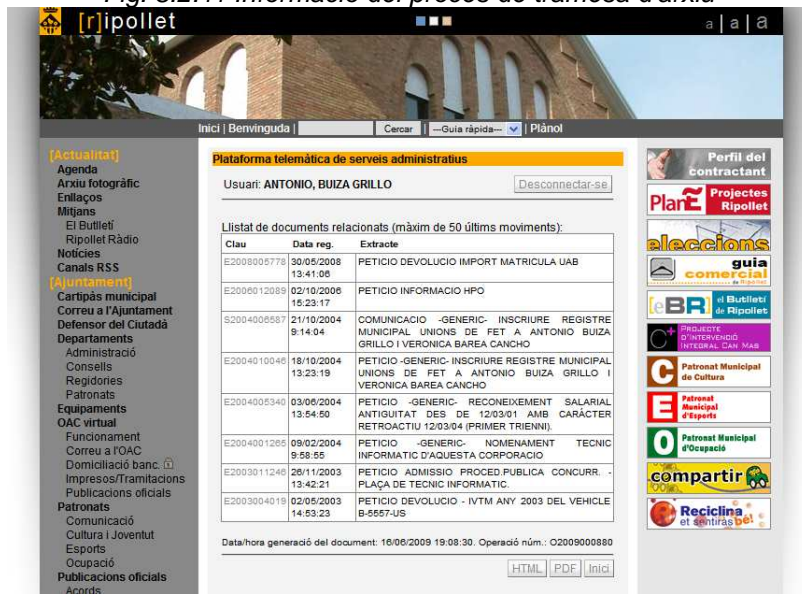


Fig. 8.2.12 Llistat de registres vinculats amb l'usuari. doConsultaDocumentsRegistre

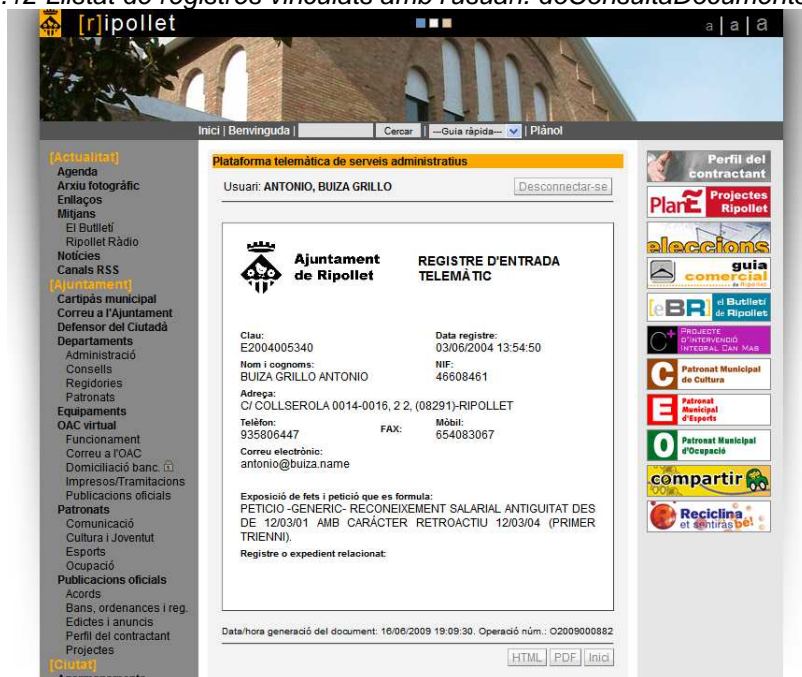


Fig. 8.2.13 Detall d'un document de registre. doConsultaDocumentsRegistre

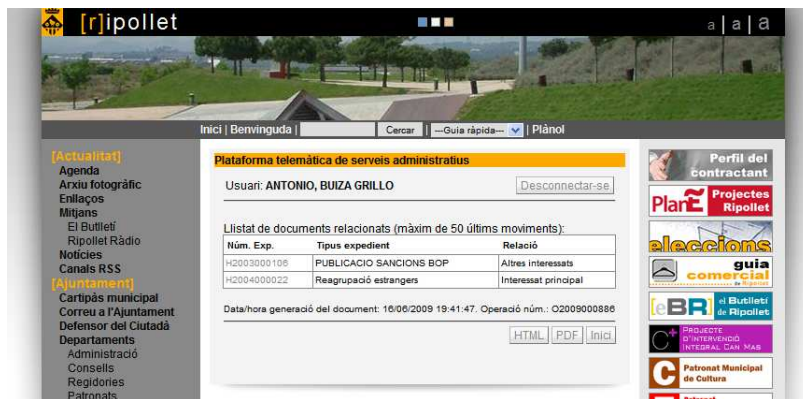


Fig. 8.2.14 Llistat d'expedients vinculats amb l'usuari. doConsultaExpedients

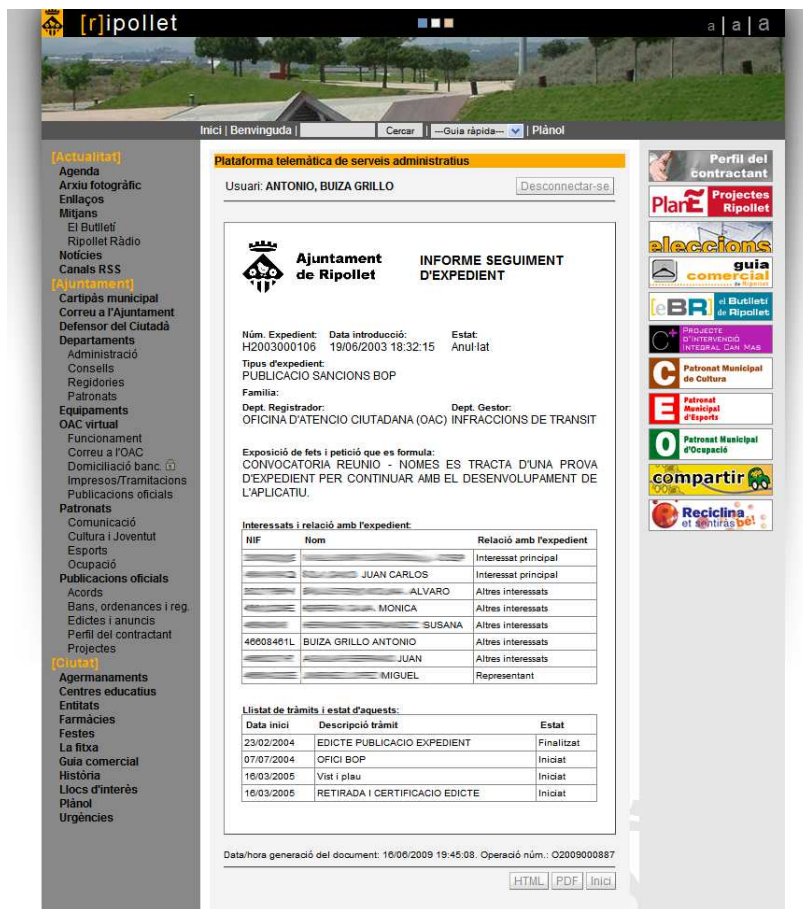


Fig. 8.2.15 Detall d'un dels expedients. doConsultaExpedients

Plataforma telemàtica de serveis administratius

Usuari: ANTONIO, BUIZA GRILLO [Desconnectar-se](#)

Ajuntament de Ripollet JUSTIFICANT TELEMÀTIC D'EMPADRONAMENT

VICESECRETARI EN FUNCIONS DE SECRETARI DE L'AJUNTAMENT DE RIPOLLET (BARCELONA)

FAIG CONSTAR: Que examinat el vigent Padró Municipal d'Habitants d'aquest Municipi, en el dia de la data hi figura inscrit(a):

ANTONIO BUIZA GRILLO

Amb document: el 46608461 L
 Natural de: SABADELL
 Nacut/a el: 26/08/1976
 Domicili: C/ [redacted]
 RIPOLLET (BARCELONA)

Districte: [redacted]
 Secció: [redacted]
 Full: [redacted]
 Data alta al padró: [redacted]

I, perquè consti, a petició seva, s'expedeix el present justificant.

RIPOLLET, 16/06/2009

Data/hora generació del document: 16/06/2009 19:17:49. Operació núm.: Q2009000884

[HTML](#) [PDF](#) [Inici](#)

Fig. 8.2.16 Pantalla justificant d'empadronament

Plataforma telemàtica de serveis administratius

Usuari: ANTONIO, BUIZA GRILLO [Desconnectar-se](#)

Ajuntament de Ripollet JUSTIFICANT TELEMÀTIC DE CONVIVÈNCIA

VICESECRETARI EN FUNCIONS DE SECRETARI DE L'AJUNTAMENT DE RIPOLLET (BARCELONA)

FAIG CONSTAR: Que examinat el vigent Padró Municipal d'Habitants d'aquest Municipi, resulta que:

ANTONIO BUIZA GRILLO

Amb el document: 46608461 L
 Natural de: SABADELL
 Nacut/a el: 26/08/1976
 Domicili: [redacted]
 RIPOLLET (BARCELONA)

Data alta al padró: [redacted]
 Informalitzat: [redacted]

Figura inscrit al mateix domicili amb les següents persones:

Nom	Cognoms	Document	D.Naix	D.Alta
VERONICA	[redacted]	[redacted]	[redacted]	[redacted]

I, perquè consti, a petició seva, s'expedeix el present justificant.

RIPOLLET, 16/06/2009

Data/hora generació del document: 16/06/2009 19:18:08. Operació núm.: Q2009000885

[HTML](#) [PDF](#) [Inici](#)

Fig. 8.2.17 Pantalla justificant de convivència

Plataforma telemàtica de serveis administratius

Usuari: ANTONIO, BUIZA GRILLO [Desconnectar-se](#)

Llistat de documents relacionats (màxim de 50 últims moviments):

T	Codi	Exer.	Concepte	Import
R	1554030	2009	I.B.I. Impost sobre bens immobles de nat. urbana	455,88 €
R	1636005	2008	TAXA RECOLLIDA D'ESCOBRARIES	42,0 €
R	1455557	2008	I.B.I. Impost sobre bens immobles de nat. urbana	424,16 €
R	1352871	2007	I.B.I. Impost sobre bens immobles de nat. urbana	401,19 €
R	1426838	2007	TAXA RECOLLIDA D'ESCOBRARIES	41,5 €
R	1129800	2006	TAXA RECOLLIDA D'ESCOBRARIES	40,15 €
R	1089052	2006	I.B.I. Impost sobre bens immobles de nat. urbana	378,39 €
R	1006941	2005	TAXA RECOLLIDA D'ESCOBRARIES	38,0 €
R	947511	2005	I.B.I. Impost sobre bens immobles de nat. urbana	352,97 €
R	780041	2004	TAXA RECOLLIDA D'ESCOBRARIES	37,5 €
R	689005	2004	I.B.I. Impost sobre bens immobles de nat. urbana	319,81 €
L	704475	2004	TAXA RECOLLIDA D'ESCOBRARIES	37,5 €
R	407352	2003	IMPOST VEHICLES TRACCIO MECANICA	54,18 €
R	337262	2002	IMPOST VEHICLES TRACCIO MECANICA	54,18 €
R	271590	2001	IMPOST VEHICLES TRACCIO MECANICA	54,18 €
R	229898	2000	IMPOST VEHICLES TRACCIO MECANICA	54,18 €
R	175210	1999	IMPOST VEHICLES TRACCIO MECANICA	13,55 €
A	18001	1999	IMPOST VEHICLES	27,09 €
R	110751	1998	IMPOST VEHICLES TRACCIO MECANICA	13,28 €
R	155220	1997	IMPOST VEHICLES TRACCIO MECANICA	12,86 €
L	152882	1996	IMPOST VEHICLES TRACCIO MECANICA	6,43 €

Legenda columna T: R->Rebut, A->Autoliquidació, L-> Liquidació.

Data/hora generació del document: 21/06/2009 18:28:58. Operació núm.: 02009000906

[HTML](#) [PDF](#) [Inici](#)

Fig. 8.2.18 Llistat de documents d'ingressos vinculats. doConsultaDocumentsIngressos

Ajuntament de Ripollet JUSTIFICANT DE PAGAMENT

L'Ajuntament de Ripollet

JUSTIFICA

Que de la documentació que consta en aquest Ajuntament resulta que el rebut indicat a continuació ha estat cobrat:

Contribuent:
46608461L
ANTONIO BUIZA GRILLO

Document:
Referència: 1455557
Exercici: 2008
Concepte: I.B.I. Impost sobre bens immobles de nat. urbana
d'ingrés: urbana
Objecte tributari: B

Ref. Cadastral:	
Domicili:	
VALOR CATASTRAL:	
VALOR SOL:	
VALOR CONSTRUCC:	

Estat document: Cobrat en Entitat Bancaria (V)

Total cobrat: 424,16 €
Total document: 424,16 €


Detalls del document:

Núm.	Fet imposable	Import	Bonif.
1	I.B.I. Urbana	424,16 €	0 €

Data/hora generació del document: 21/06/2009 18:29:08. Operació núm.: 02009000907

[HTML](#) [PDF](#) [Inici](#)

Fig. 8.2.19 Detall d'un document. doConsultaDocumentsIngressos



JUSTIFICANT DE PAGAMENT

L'Ajuntament de Ripolllet

JUSTIFICA

Que de la documentació que consta en aquest Ajuntament resulta que el rebut indicat a continuació ha estat cobrat:

Contribuent:
46608461L
ANTONIO BUIZA GRILLO

Document:
Referència: **1455557**
Exercici: **2008**
Concepte d'ingrés: **IB.I. Impost sobre bens immobles de nat. urbana**
Objecte tributar: **BI /**

Ref. Cadastral:	
Domicili:	CE
VALOR CATASTRAL:	
VALOR SOL:	
VALOR CONSTRUCC:	

Estat document: **Cobrat en Entitat Bancària (V)**
Total cobrat: **424,16 €**
Total document: **424,16 €**


Detalls del document:

Núm.	Fet imposable	Import	Bonif.
1	IB.I. Urbana	424,16 €	0 €

Balmes, 2
Tel. 93 504 60 00
Fax. 93 580 81 48
08291 Ripolllet
CIF P-08-17900 D

Fig. 8.2.20 Justificant en format HTML. doGenerateDocument

Firmado y todas las firmas son válidas. Panel de firma



JUSTIFICANT DE PAGAMENT

L'Ajuntament de Ripolllet

JUSTIFICA

Que de la documentació que consta en aquest Ajuntament resulta que el rebut indicat a continuació ha estat cobrat:

Contribuent:
46608461L
ANTONIO BUIZA GRILLO

Document:
Referència: **1455557**
Exercici: **2008**
Concepte d'ingrés: **IB.I. Impost sobre bens immobles de nat. urbana**
Objecte tributar: **BI /**

Ref. Cadastral:	
Domicili:	CE
VALOR CATASTRAL:	
VALOR SOL:	
VALOR CONSTRUCC:	

Estat document: **Cobrat en Entitat Bancària (V)**
Total cobrat: **424,16 €**
Total document: **424,16 €**

Detalls del document:

Núm.	Fet imposable	Import	Bonif.
1	IB.I. Urbana	424,16 €	0 €

Balmes, 2
Tel. 93 504 60 00
Fax. 93 580 81 48
08291 Ripolllet
CIF P-08-17900 D

Signat electrònicament per:
Ajuntament de Ripolllet
(RIPOLLET CAT)
Data: 16.06.2009

Data/hora generació del document: 15/06/2009 20:34:30. Operació nòm.: 0200000080

Fig. 8.2.21 Justificant en format PDF Signat. doGenerateDocument

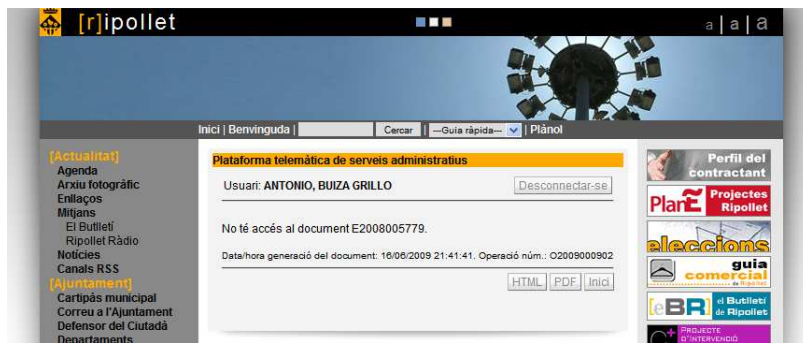


Fig. 8.2.22 Error quan s'intenta accedir a document no vinculat

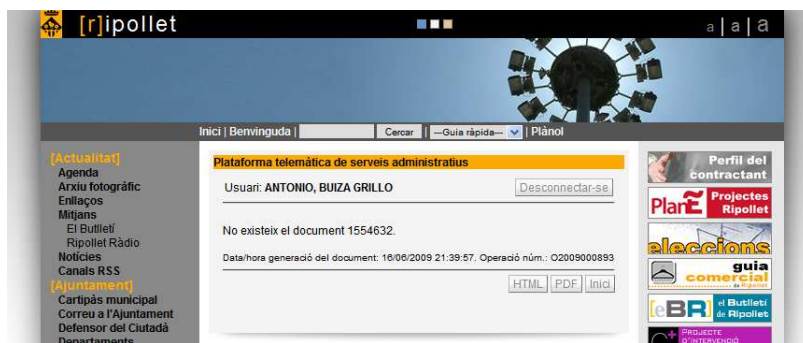


Fig. 8.2.23 Error quan s'intenta accedir a un document inexistent



Fig. 8.2.24 Pantalla de tancament de sessió voluntària. doLogOut

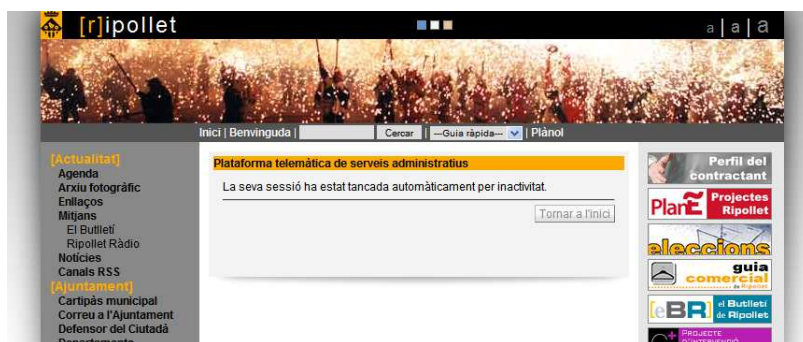


Fig. 8.2.25 Pantalla de tancament de sessió per inactivitat. doLogOut

8.3 Detalls tècnics d'implementació

8.3.1 Configuració IIS / .NET Framework

8.3.1.1 Habilitar canal SSL

Per a establir un canal segur de comunicacions, *https*, entre el client i l'*Internet Information Server*, necessitarem:

- Disposar d'un certificat de dispositiu segur.
- Disposar dels certificats de clau pública de tota la ruta de certificació del certificat anterior.
- Tenir drets d'administració sobre el servidor virtual a configurar.

Passos a seguir:

1) Instal·lar els certificats de clau pública de la ruta de certificació i el de dispositiu segur.

Obrim la consola *mmc* amb el procediment: Inicio→Ejecutar→*mmc*. Després afegim el component *Certificados* del compte *Equipo local*.

Expandir l'arbre *Certificados*→*Entidades emisoras raiz de confianza*→*Certificados*, i en el menú contextual d'aquest node seleccionar *Todas las tareas*→*Importar...*

Incloure els certificats arrels de certificació necessaris.

Després expandir l'arbre *Certificados*→*Personal*→*Certificados*, i en el menú contextual d'aquest node seleccionar *Todas las tareas*→*Importar...*

Aquesta vegada incloure només el certificat de dispositiu segur.

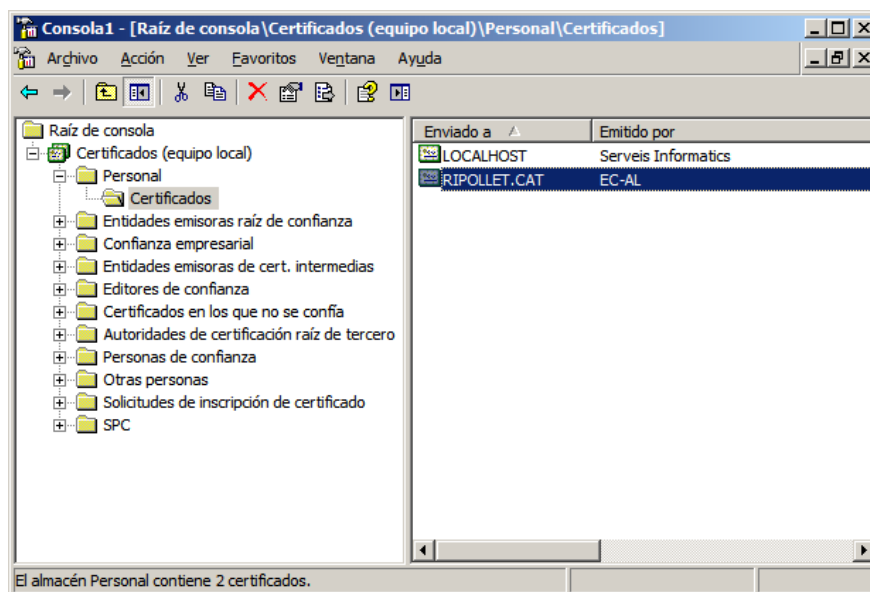


Fig. 8.3.1 Consola MMC amb component de certificats

Un cop està instal·lat el certificat de dispositiu segur, comprovem que no surt cap advertència de confiança i ens assegurem que disposa de clau privada mirant si surt la icona de la clau abaix de tot.

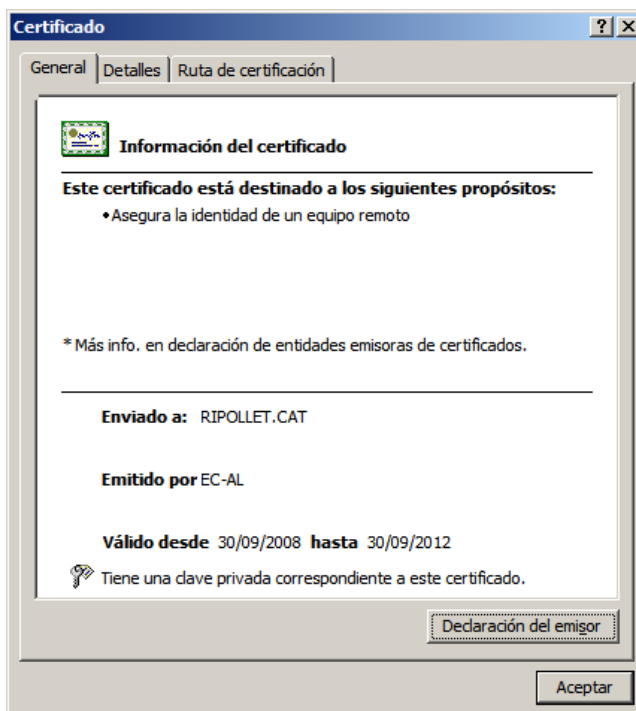


Fig. 8.3.2 Propietats del certificat.

2) Habilitar el canal segur SSL a *Internet Information Server*.

Obrim la consola *mmc* novament amb el procediment: Inicio→Ejecutar→mmc. Després afegim el component *Servicios de Internet Informtion Server*.

Expandim l'arbre d'administració fins trobar el servidor virtual que volem activar l'accés segur i seleccionem en el menú contextual l'opció *Propiedades*.

Anem a la pestanya *Seguridad de directorios* i fem clic al botó *Certificado de servidor*. Fem clic a botó *Siguiente* de l'assistent i seleccionem l'opció *Asignar un certificado ya existente*.

Si hem fet correctament el pas 1), al llistat se'ns mostrarà el nostre certificat de dispositiu segur, el seleccionem, fem clic a *Siguiente* i després a *Finalizar*.

Ara en la pestanya *Seguridad de directorios*, fem clic a *Modificar* i activem la casella *Requerir canal seguro (SSL)*.

A partir d'aquest moment, el servidor virtual configurat només servirà les pàgines a través d'un canal segur *https/ssl*. El port de servei es pot canviar a la pestanya *Sitio web*.

8.3.1.2 Requerir certificats de client i configuració de la CTL

Per a que *IIS* requereixi, en un determinat *webform*, l'entrega d'un certificat de client hem visualitzar les propietats de l'arxiu des de la consola *mmc* amb el component *Servicios de Internet Information Server*.

Dins les propietats, a la pestanya *Seguridad de archivo*, fer clic al botó *Modificar* i activar les caselles *Requerir canal seguro (SSL)* i *Requerir certificados de cliente*.

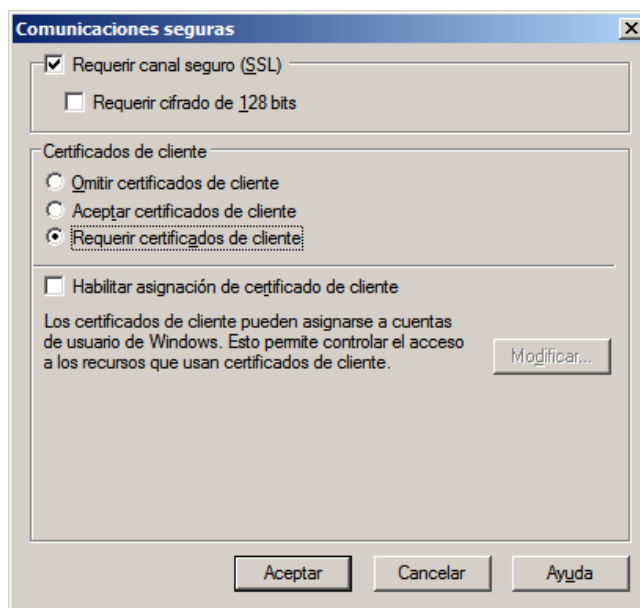


Fig. 8.3.3 Pantalla de comunicaciones seguras de IIS

Ara si tot va bé, l'execució d'aquest *webform* només es realitzarà si el client presenta un certificat de client, però abans de provar-ho hem de configurar una *Certificate Trusted List* amb les arrels de certificació dels certificats que volem acceptar per la nostra aplicació.

Ho farem de la següent manera:

S'han d'importar els certificats de clau pública de cada una de les entitats de certificats que volem reconèixer. Això es farà des de la consola *mmc* amb el component de *Certificados*. Igual que en el punt anterior, els hem d'importar contra

Certificados→*Entidades emisoras raiz de confianza*→*Certificados* però aquesta vegada a mesura que anem introduint-les hem d'obrir cada un dels certificats, revisar que no hi hagi advertències, anar a la pestanya *Detalles*, fer clic al botó *Modificar propiedades* i revisar que la casella *Autenticación del servidor* es troba marcada.

Un cop hem acabat d'introduir totes les arrels de certificació i revisat les seves propietats podem configurar la *CTL* a IIS.

Anem a les propietats del servidor virtual adequat i fem clic a la pestanya *Seguridad de directorios* i després al botó *Modificar*. Activem la casella *Habilitar la lista de certificados de confianza*. Després fem clic a *Nueva*, després iniciem l'assistent i fem clic a *Agregar desde almacén*. Al llistat hem de localitzar les arrels configurades en el pas anterior. Les seleccionem una per una i anem configurant-les a la nova *CTL*.

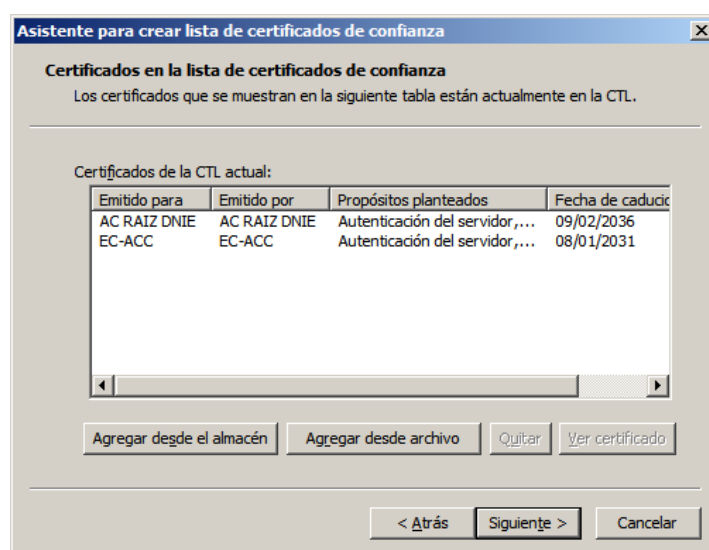


Fig. 8.3.4 Pantalla d'assistent de creació de la CTL

Finalment donem un nom a la nova *CTL* i ho acceptem tot. Si tot ha anat bé ara els navegadors utilitzats per fer les proves demanaran un certificat d'usuari i es mostrarà un llistat amb els certificats emesos per les *CAs* configurades a la *CTL*.

8.3.1.3 Registrar un component a la GAC

Hi ha diferents formes d'enregistrar un component compilat en una *DLL* a la *GAC* o *Global Assembly Cache* del *.NET Framework*.

Es pot fer des de la consola de comandes *cmd* i la comanda *gacutil* del *SDK* del *.NET Framework* o bé fent servir novament la consola *mmc* però aquesta vegada amb el component *.NET Framework XX Configuration*.

Només hem d'expandir l'arbre de consola fins *.NET Framework XX Configuration*→*Mi PC*→*Cache de ensamblados*, i en el menú contextual d'aquesta seleccionar l'opció *Agregar*. Després naveguem fins on hem guardat l'arxiu *DLL*, el seleccionem i fem clic a *Abrir*.

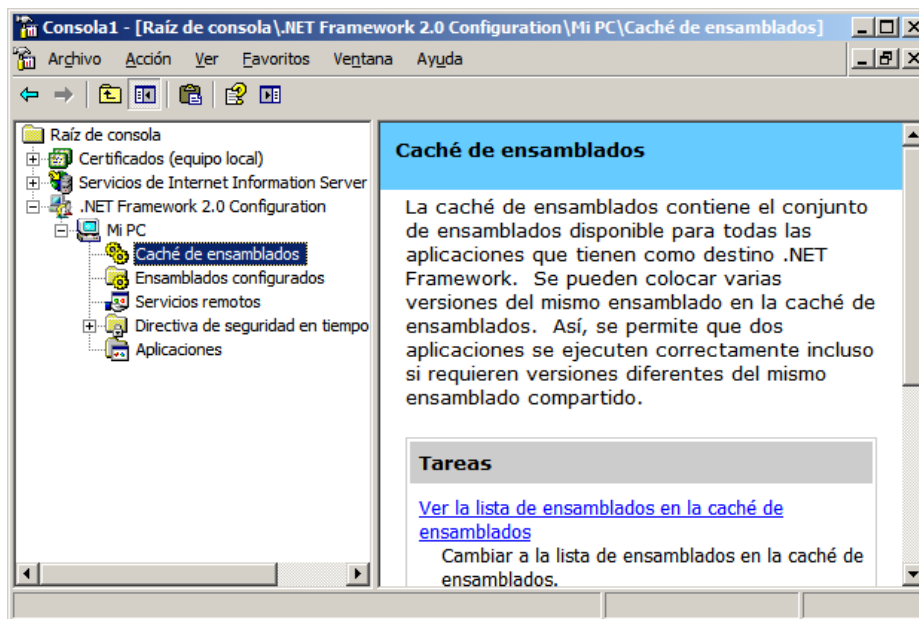


Fig. 8.3.5 Consola mmc amb component .NET Framework Configuration

8.3.1.4 Referències a components en arxiu web.config

L'arxiu *web.config* el genera automàticament l'*IDE* del *Visual Studio Web Developer* però és pràctic saber com validar un component enregistrat a la GAC per a que sigui possible referenciar-lo des del nostre projecte.

Per trobar el literal amb la referència del component que volem enllaçar farem servir la comanda *gacutil*. Aquesta la trobarem al directori *bin* de l'*SDK* de *.NET* que estiguem utilitzant.

En el directori */bin*, executarem la comanda *gacutil /l > gac.txt*. Aquesta ens deixa la llista de components enregistrats a la GAC a l'arxiu *gac.txt*. Ara cal editar aquest arxiu i copiar el literal amb la referència del component que volem enllaçar. Vegem la figura:

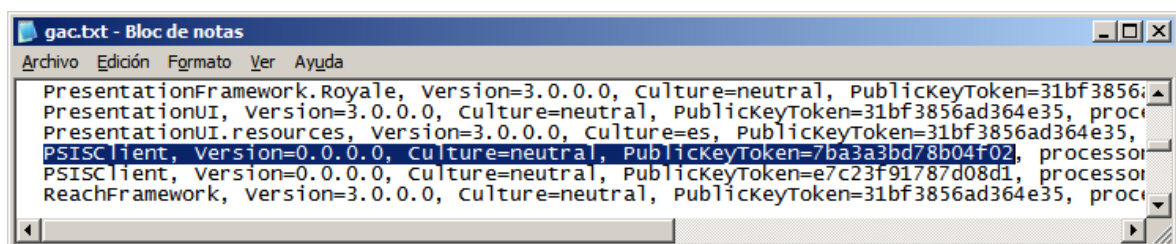


Fig. 8.3.6 Selecció del literal del component a l'arxiu gac.txt

Ara copiem aquest literal, fins l'última coma i ho introduïm a l'arxiu *web.config* de la següent manera:

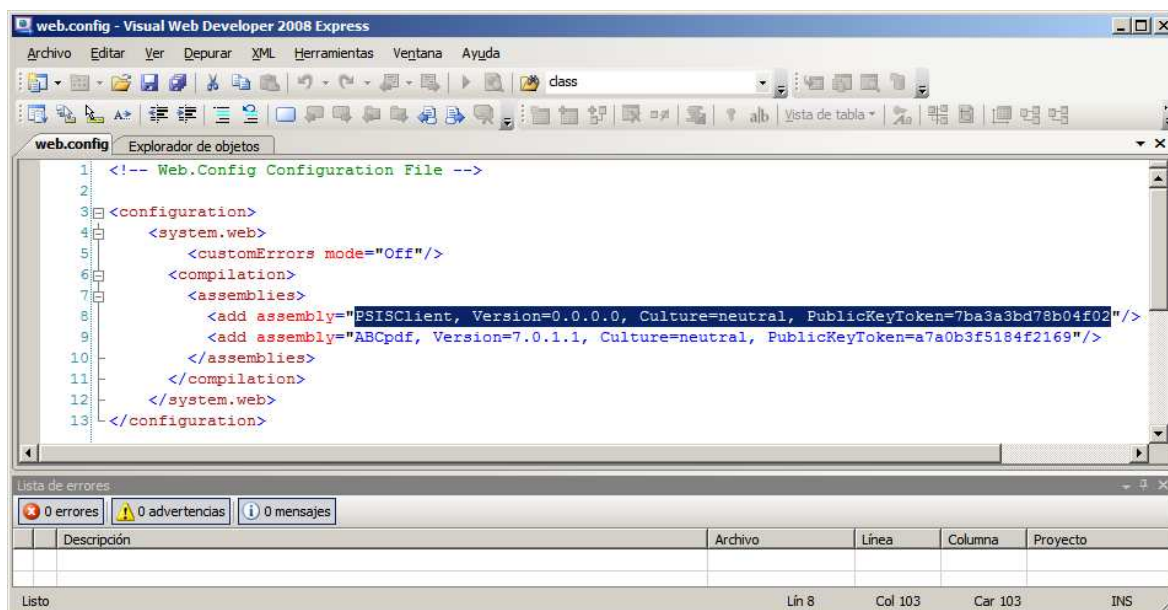


Fig. 8.3.7 Inclusió de la referència a l'arxiu web.config

Fent això ja serà possible importar i utilitzar els recursos del control des del nostre projecte.

8.3.2 Validació de certificats amb CATCert/PSIS

En aquest apartat il·lustrarem amb uns exemples de codi en C# el comportament de la plataforma PSIS de CATCert en quant a la validació de certificats digitals.

8.3.2.1 Invocació senzilla del servei

Per al correcte funcionament de l'exemple hem d'invocar el mètode *doCertificateValidation* passant com a paràmetre un objecte *byte[]* que conté l'stream binari del certificat a validar. Per exemple el *HttpClientCertificate.Certificate*.

```
using net.catcert.psis;

public class CPSISUtilities {

    public VerifyResponse doCertificateValidadion(byte[] bCert) {

        // Utilitats
        Utils utils = new Utils();

        // Còpia del certificat de client
        byte[] certificate = bCert;

        // Inicialització del client
        digitalSignatureService proxy = new digitalSignatureService();

        // Indicar l'adreça del servidor
        // Pre-Produccio -> "https://psisbeta.catcert.net/psis/catcert-test/dss";
        proxy.Url = "https://psis.catcert.net/psis/catcert/dss";

        // Composició del missatge
        X509DataType x509DataType = new X509DataType();
```

```

Object[] certificates = new Object[1];
certificates[0] = certificate;

ItemsChoiceType[] certificateType = new ItemsChoiceType[1];
certificateType[0] = ItemsChoiceType.X509Certificate;

x509DataType = new X509DataType();
x509DataType.Items = certificates;
x509DataType.ItemsElementName = certificateType;

SignatureObjectTypeOther other = new SignatureObjectTypeOther();
other.X509Data = x509DataType;

SignatureObjectType signature = new SignatureObjectType();
signature.Item = other;

// Creació del missatge DSS
VerifyRequest requestDocument = new VerifyRequest();
requestDocument.SignatureObject = signature;

// Execució del servei
VerifyResponse responseDocument = proxy.verify(requestDocument);
return (responseDocument);
}
}

```

Fig. 8.3.8 Codi que implementa la invocació senzilla a la plataforma PSIS

El mètode retorna un objecte *VerifyResponse* definit al *namespace* de *PSIS*. A l'apartat següent expliquem com tractar aquest objecte.

8.3.2.2 Processament de la resposta

En aquest apartat explicarem amb un exemple de codi com fer el tractament de l'objecte *VerifyResponse* retornat de la invocació al servei de validació de *PSIS*.

Definirem la classe *CPSISResultsParser* per al tractament d'objectes *VerifyResponse* de la següent manera:

```

public class CPSISResultsParser {

    private String sResultMajor;
    private String sResultMinor;
    private String sResultMessage;

    public String resultMajor {
        get { return (sResultMajor); }
    }

    public String resultMinor {
        get { return (sResultMinor); }
    }

    public String resultMessage {
        get { return (sResultMessage); }
    }

    public CPSISResultsParser(VerifyResponse oVR) {
        if (oVR != null) {
            sResultMajor = getValue(oVR.Result.ResultMajor);
            sResultMinor = getValue(oVR.Result.ResultMinor);

            if (oVR.Result.ResultMessage != null)
                sResultMessage = getValue(oVR.Result.ResultMessage.ToString());
        }
    }

    private String getValue(String sInput) {
        if (!String.IsNullOrEmpty(sInput)) {

```

```

String[] sRet;
sRet = sInput.Split(':');
if (sRet.Length > 1) return (sRet[sRet.Length - 1]);
else return (String.Empty);
} else {
return (String.Empty);
}
}

public bool isValid {
get {
bool bRet = false;

if ((!String.IsNullOrEmpty(resultMajor)) && (resultMajor.ToUpper() ==
"SUCCESS")) {
if ((!String.IsNullOrEmpty(resultMinor)) && (resultMinor.ToUpper() ==
"DEFINITIVE")) {
bRet = true;
}
}

return (bRet);
}
}
}
}

```

Fig. 8.3.9 Codi que implementa la classe CPSISResultsParser

Un detall important és que el constructor d'aquesta classe rep com a paràmetre un objecte *VerifyResponse*.

Observem que els valors de retorn del objecte *VerifyResponse* que ens indica que el certificat enviat a *PSIS* és vàlid són: *resultMajor*="SUCCESS" i *resultMinor*="DEFINITIVE". Qualsevol altra combinació ens dona error i així s'ha implementat en el codi de la propietat *isValid* d'aquesta classe.

8.3.2.3 Invocació amb OptionalInputs

Veiem ara el codi que s'ha d'afegir al ja introduït a l'apartat 8.3.2.1 per realitzar una invocació amb *OptionalInputs*. Això ens permet demanar més dades del procés de validació. En aquest cas ens interessa el nom, cognoms i DNI del titular del certificat. És molt pràctica aquesta consulta doncs evita escanejar els camps del certificat d'usuari fent recerca de les dades.

```

using net.catcert.psis;

public class CPSISUtilities {

public VerifyResponse doCertificateValidadion(byte[] bCert) {
// Utilitats
Utils utils = new Utils();

// Copia del certificat de client
byte[] certificate = bCert;

// Inicialització del client
digitalSignatureService proxy = new digitalSignatureService();

// Indicar l'adreça del servidor
// Pre-Produccio -> "https://psisbeta.catcert.net/psis/catcert-test/dss";
proxy.Url = "https://psis.catcert.net/psis/catcert/dss";

// Composició del missatge
X509DataType x509DataType = new X509DataType();

```

```

Object[] certificates = new Object[1];
certificates[0] = certificate;

ItemsChoiceType[] certificateType = new ItemsChoiceType[1];
certificateType[0] = ItemsChoiceType.X509Certificate;

x509DataType = new X509DataType();
x509DataType.Items = certificates;
x509DataType.ItemsElementName = certificateType;

SignatureObjectTypeOther other = new SignatureObjectTypeOther();
other.X509Data = x509DataType;

SignatureObjectType signature = new SignatureObjectType();
signature.Item = other;

// Creació del missatge DSS
VerifyRequest requestDocument = new VerifyRequest();
requestDocument.Profile = "urn:oasis:names:tc:dss:1.0:profiles:XSS";
requestDocument.SignatureObject = signature;
requestDocument.OptionalInputs = doBuildOptionalInputs();

// Execució del servei
VerifyResponse responseDocument = proxy.verify(requestDocument);

return (responseDocument);
}

private OptionalInputs doBuildOptionalInputs() {
    OptionalInputs optInputs = new OptionalInputs();
    ItemsChoiceType6[] types6 = new ItemsChoiceType6[1];

    Object[] objects = new Object[1];

    optInputs.ItemsElementName = types6;
    optInputs.Items = objects;

    // Colleccio dels atributs a demanar
    string[] attributeDesignators = new string[]
    {
        // GivenName
        "urn:oasis:names:tc:dss:1.0:profiles:XSS:certificateAttributes:SubjectDistingishedName:givenName",
        // Surname
        "urn:oasis:names:tc:dss:1.0:profiles:XSS:certificateAttributes:SubjectDistingishedName:surname",
        // KeyOwnerNIF
        "urn:catcert:psis:certificateAttributes:KeyOwnerNIF",
        // SerialNumber
        "urn:oasis:names:tc:dss:1.0:profiles:XSS:certificateAttributes:SerialNumber",
        // IssuerCA
        "urn:catcert:psis:certificateAttributes:issuerCA",
    };

    AttributeDesignator[] ads = new
AttributeDesignator[attributeDesignators.Length];

    types6[0] = ItemsChoiceType6.ReturnX509CertificateInfo;
    objects[0] = ads;

    for (int i = 0; i < attributeDesignators.Length; i++) {
        string attributeDesignator = attributeDesignators[i];

        AttributeDesignator ad = new AttributeDesignator();
        ad.Name = attributeDesignator;

        ads[i] = ad;
    }
    return optInputs;
}
}

```

Fig. 8.3.10 Codi que implementa la invocació amb OptionalInputs a la plataforma PSIS

En aquest exemple, s'ha introduït un mètode privat per a la generació de l'objecte *OptionalInputs*. Aquest objecte es genera amb es paràmetres per a demanar les dades *givenName*, *surname*, *KeyOwnerNIF*, *SerialNumber* i *issuerCA*.

8.3.2.4 Processament de la resposta amb *OptionalOutputs*

A continuació veurem com tractar la resposta *VerifyResponse* amb la recuperació dels *OptionalOutputs*.

```
public class CPSISResultsParser {

    private String sResultMajor;
    private String sResultMinor;
    private String sResultMessage;
    private String sGivenName;
    private String sSurname;
    private String sKeyOwnerNIF;
    private String sSerialNumber;
    private String sIssuerCA;

    public String resultMajor {
        get { return (sResultMajor); }
    }

    public String resultMinor {
        get { return (sResultMinor); }
    }

    public String resultMessage {
        get { return (sResultMessage); }
    }

    public String givenName {
        get { return (sGivenName); }
    }

    public String surname {
        get { return (sSurname); }
    }

    public String keyOwnerNIF {
        get { return (sKeyOwnerNIF); }
    }

    public String serialNumber {
        get { return (sSerialNumber); }
    }

    public String issuerCA {
        get { return (sIssuerCA); }
    }

    public CPSISResultsParser(VerifyResponse oVR) {
        if (oVR != null) {
            sResultMajor = getValue(oVR.Result.ResultMajor);
            sResultMinor = getValue(oVR.Result.ResultMinor);

            // PROCESSA OPTIONALOUTPUTS
            doParseOptionalOutputs(oVR);

            if (oVR.Result.ResultMessage != null)
                sResultMessage = getValue(oVR.Result.ResultMessage.ToString());
        }
    }

    private void doParseOptionalOutputs(VerifyResponse oVR) {
        String sAttributeName, sAttributeValue;

        foreach (object o in oVR.OptionalOutputs.Items) {
```

```

if (o.GetType() == typeof(X509CertificateInfo)) {
    X509CertificateInfo ci = (X509CertificateInfo)o;

    foreach (net.catcert.psis.Attribute a in ci.Attribute) {
        sAttributeName = getValue(a.Name);
        sAttributeValue=a.AttributeValue[0].ToString();

        switch (sAttributeName) {
            case "givenName": sGivenName = sAttributeValue; break;
            case "surname": sSurname = sAttributeValue; break;
            case "KeyOwnerNIF": sKeyOwnerNIF=sAttributeValue; break;
            case "SerialNumber": sSerialNumber=sAttributeValue; break;
            case "issuerCA": sIssuerCA=sAttributeValue; break;
        }
    }
}

private String getValue(String sInput) {
    if (!String.IsNullOrEmpty(sInput)) {
        String[] sRet;
        sRet = sInput.Split(':');
        if (sRet.Length > 1) return (sRet[sRet.Length - 1]);
        else return (String.Empty);
    } else {
        return (String.Empty);
    }
}

public bool isValid {
    get {
        bool bRet = false;

        if ((!String.IsNullOrEmpty(resultMajor)) && (resultMajor.ToUpper() ==
"SUCCESS")) {
            if ((!String.IsNullOrEmpty(resultMinor)) && (resultMinor.ToUpper() ==
"DEFINITIVE")) {
                bRet = true;
            }
        }

        return (bRet);
    }
}
}

```

Fig. 8.3.11 classe *CPSISResultsParser* amb nous atributs i tractament d'*OptionalOutputs*

A diferència de la classe *CPSISResultsParser* de l'exemple 8.3.2.2, aquesta disposa de nous atributs per al tractament de les dades demanades amb la invocació amb *OptionalInputs* de l'exemple 8.3.2.3.

S'afegeix també un mètode privat per al tractament dels valors retornats per l'objecte *OptionalOutputs*. Aquest és el *doParseOptionalOutputs*.

8.3.3 Generació dinàmica de PDF amb webSupergoo.ABCPdf

En aquest apartat es pretén demostrar la versatilitat del component *ABCPdf* com a generador de contingut PDF dinàmic.

8.3.3.1 Exemple bàsic “Hello world”

Mostrarem a continuació com utilitzar el component per a la generació de contingut PDF senzill:

```
using WebSupergoo.ABCpdf7;
using WebSupergoo.ABCpdf7.Objects;

public class PDFDummie {
    public void doHelloWorld() {
        Doc theDoc = new Doc();
        theDoc.FontSize = 50;
        theDoc.AddText("Hello World");
        theDoc.Save("c:\\HelloWorld.pdf");
        theDoc.Clear();
    }
}
```

Fig. 8.3.12 Exemple de codi que genera un arxiu PDF amb “Hello World”

El codi anterior produeix el següent arxiu PDF:

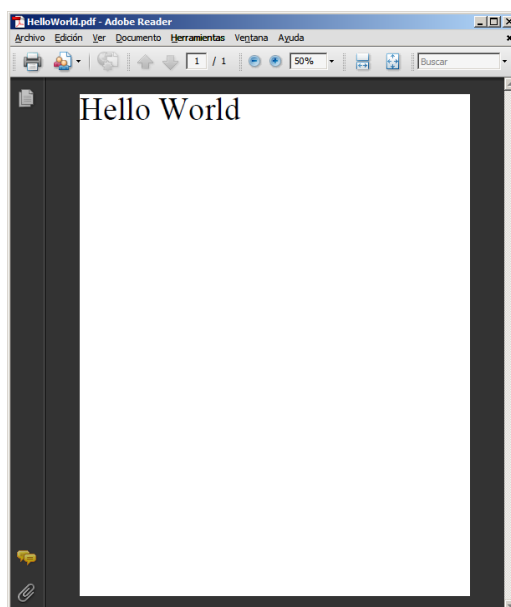


Fig. 8.3.13 Document PDF senzill produït amb el codi anterior

8.3.3.2 Renderització HTML en PDF

En aquest exemple, mostrarem les possibilitats de renderització HTML del component.

```
using WebSupergoo.ABCpdf7;
using WebSupergoo.ABCpdf7.Objects;

public class PDFDummie {
    public void doRenderHTML() {
        Doc theDoc = new Doc();
        theDoc.Rect.Inset(72, 144);
        theDoc.AddImageUrl("http://etse.uab.cat/");
        theDoc.Save("c:\\HTML.pdf");
        theDoc.Clear();
    }
}
```

Fig. 8.3.14 Exemple de codi que renderitza contingut HTML dins un arxiu PDF

El codi anterior produeix el següent arxiu PDF:

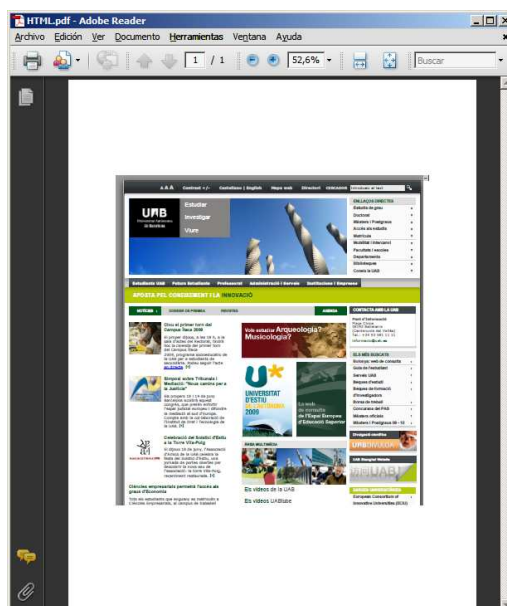


Fig. 8.3.15 Document PDF amb HTML renderitzat produït amb el codi anterior

8.3.3.3 Signatura electrònica amb certificat digital

Finalment, l'exemple següent, mostra com generar dinàmicament contingut i com estampar una signatura digital:

```
using WebSupergoo.ABCpdf7;
using WebSupergoo.ABCpdf7.Objects;

public class PDFDummie {
    public void doSignSample(String sSignaturePath, String sPassword) {
        Doc theDoc = new Doc();
        theDoc.FontSize = 50;
        theDoc.AddText("Contingut signat electrònicament");

        // AFEGIR SIGNATURA ELECTRÒNICA
        InteractiveForm form = new InteractiveForm(theDoc);
        form.AddSignature("410 0 550 110", "Signature",
            sSignaturePath, sPassword, "Ajuntament de Ripollet", "Ripollet",
            "\nSignat electrònicament per:\nAjuntament de Ripollet\n({0})\nData:
{1:dd.MM.yyyy}");

        Signature theSig = (Signature)theDoc.Form["Signature"];

        if (theSig == null) throw new Exception("*** ERROR: theSig is null.");
        theSig.Sign(sSignaturePath, sPassword);

        theDoc.Save("C:\\Signed.pdf");
        theDoc.Clear();
    }
}
```

Fig. 8.3.16 Exemple de codi que genera un arxiu PDF amb "Hello World"

Aquest exemple de codi requereix crear un objecte *InteractiveForm*, la referència a aquest la trobem al projecte *Annotations* –inclòs amb el control–.

El mètode s'invoca passant com a paràmetres la ruta absoluta a l'arxiu del certificat digital –arxiu *.pfx*– i el password d'aquest.

El resultat produït és el següent:

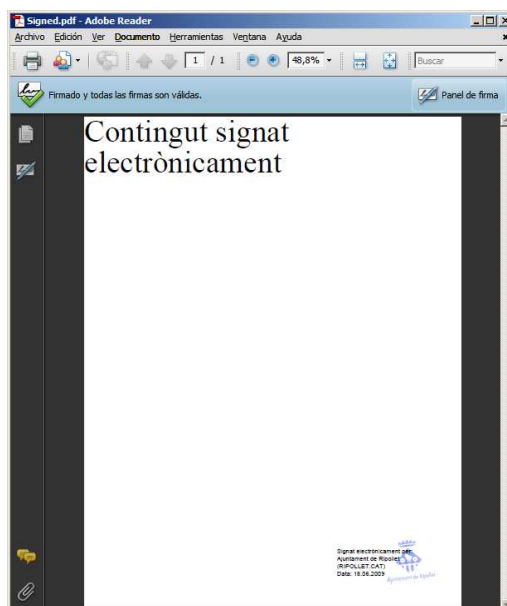


Fig. 8.3.17 Document PDF signat electrònicament produït amb el codi anterior

8.3.4 Enviament d'arxius amb NeatUpload

NeatUpload és el control *.NET* utilitzat en la tramesa d'arxius adjunts en aquest projecte. Aquest control de *Brett Development* és gratuït i a continuació mostrem uns exemples amb l'objectiu d'explicar la integració d'aquest.

8.3.4.1 Integració del component

La interfície de comunicació d'aquest control la trobem disponible dins la caixa d'eines del dissenyador de formularis de l'IDE *.NET* que estiguem fent servir; en el nostre cas el *Microsoft Visual Web Developer 2008 Express*.

La tècnica d'integració realitzada al projecte ha estat la de dissenyar un *webform* independent per al tractament de la càrrega de l'arxiu. Aquest es troba incrustat dins un *Iframe* i es comunica amb la plataforma utilitzant la serialització d'objectes a la sessió web del client al igual que es fa des d'altres punts d'enllaç de comunicacions entre els diferents mòduls.

A continuació es mostra un parell d'imatges que il·lustren la integració del component amb l'IDE:

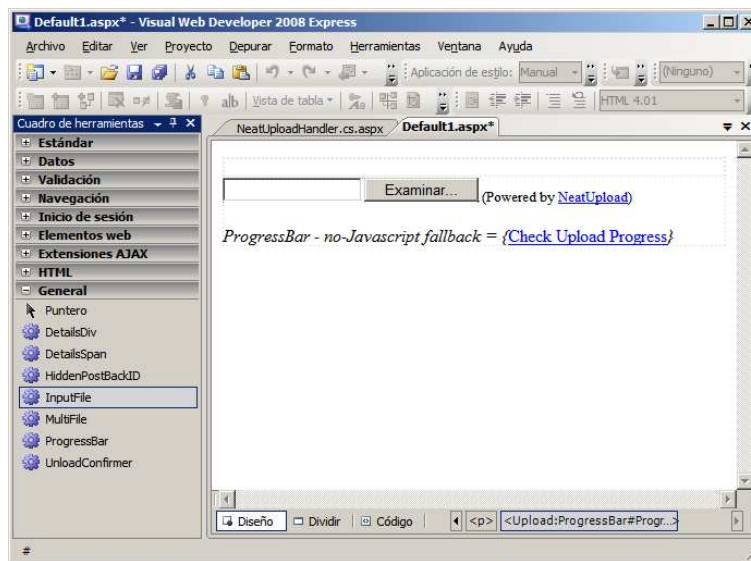


Fig. 8.3.18 IDE Visual Web Developer 2008 EE amb el control NeatUpload integrat

Els dos controls s'integren dins un mateix formulari i el procés d'enviament comença quan es fa el submit d'aquest.

8.3.4.2 Exemple d'enviament

El procés d'enviament i d'actualització de la barra de progrés es realitza automàticament per els recursos que aquest component inclou. No obstant per personalitzar l'estil i facilitar-ne la integració amb el nostre projecte hem adaptat l'arxiu */NeatUpload/Progress.aspx*.

Quan, finalment, es completa la càrrega de l'arxiu, quan les dades del formulari han arribat al servidor, el codi per fer-ne el tractament de l'arxiu rebut és el següent:

```
// ...

if (NeatIF.HasFile) {
    // DESHABILITITO ELS CONTROLS
    NeatIF.Enabled = false;
    submitButtonId.Enabled = false;

    // CALCULO EL PATH ABSOLUT TEMPORAL DE L'ARXIU
    String sPathName = sDirectory + "\\\"
        + sFileName;

    // MOC L'ARXIU
    try { NeatIF.MoveTo(sPathName, MoveToOptions.None); }
    catch { throw new Exception("Excepció provocada intentant moure l'arxiu.
Possiblement ja existeix.<br>["+ sPathName +"]"); }
}

// ...
```

Fig. 8.3.19 Codi per al tractament de l'arxiu rebut a través de NeatUpload

En el codi anterior, *NeatIF* és el nom del control *InputFile* de *NeatUpload*. Veiem que la lògica es basa en la utilització del mètode *MoveTo* d'aquest control. Així de senzill.

8.3.4.3 Càlcul del HASH MD5

Encara que el càlcul del hash MD5 no correspon a les funcionalitats del control que es comenta en aquest apartat, hem cregut convenient mostrar l'exemple en aquest apartat ja que guarda una estreta relació en el procediment de càrrega de l'arxiu en aquest projecte.

Per calcular aquest hash hem utilitzat un objecte *MD5CryptoServiceProvider* de la llibreria *System.Security.Cryptography* del *.NET Framework*.

```
// ...  
  
using System.Security.Cryptography;  
  
// ...  
  
public String getMD5Hash(String sPathName) {  
    MD5CryptoServiceProvider oMD5 = new MD5CryptoServiceProvider();  
    FileStream oFS = new FileStream(sPathName, FileMode.Open, FileAccess.Read,  
    FileShare.Read, 8192);  
    Byte[] bMD5 = oMD5.ComputeHash(oFS);  
  
    return BitConverter.ToString(bMD5).Replace("-", ""); ;  
}  
  
// ...
```

Fig 8.3.20 Codi que calcula el hash MD5 utilitzant l'objecte *MD5CryptoServiceProvider*

El mètode anterior rep com a paràmetre el path absolut de l'arxiu i retorna una cadena de caràcters amb els 32 dígit hexadecimals corresponents al càlcul del MD5.

Resum

PTSA és el projecte resultat del desenvolupament de la Plataforma Telemàtica de Serveis Administratius proposada com a iniciativa per l'Ajuntament de Ripollet. Amb l'objectiu de promoure la utilització dels canals de comunicació segurs entre el ciutadà i l'Administració, aporta els serveis d'acreditació, registre i consulta.

Integrada amb el portal *ripollet.cat*, implementa les tecnologies per al tractament de l'acreditació d'usuaris mitjançant certificació digital i la generació dinàmica de documents signats electrònicament.

Dins el marc d'execució de la llei 11/2007 d'accés electrònic dels ciutadans als serveis públics, és un gran primer pas en el llarg camí amb destinació una Administració electrònica completa i de qualitat.

Resumen

PTSA es el proyecto resultado del desarrollo de la Plataforma Telemática de Servicios Administrativos propuesta como iniciativa por el Ayuntamiento de Ripollet. Con el objetivo de promover el uso de los canales de comunicación seguros entre el ciudadano y la Administración, aporta los servicios de acreditación, registro y consulta.

Integrada con el portal *ripollet.cat*, implementa las tecnologías para el tratamiento de la acreditación de usuarios mediante certificación digital y la generación dinámica de documentos firmados electrónicamente.

En el ámbito de ejecución de la ley 11/2007 de acceso electrónico de los ciudadanos a los servicios públicos, se trata de un primer gran paso en el largo camino con destino una Administración electrónica completa y de calidad.

Abstract

PTSA is the project obtained as a result of the ICT Platform Development of the Administrative Services as an initiative of Ripollet City Council. This project emerged in order to encourage the use of reliable communication channels between the citizen and the Civil Service. Moreover, it contributes the registering, accreditation and consulting process.

Integrated into *ripollet.cat* web site, it implements technologies for the user accreditation treatment through digital certification and dynamic generation of electronic signed documents.

In the implementation field of 11/2007 law about the citizens' electronic access to Public Services, it is a big first step forward in the long way to a complete and quality electronic Administration.