

# laSalle

UNIVERSITAT RAMON LLULL

**Escola Tècnica Superior d'Enginyeria La Salle**

Treball Final de Màster

Màster Universitari en Enginyeria Informàtica i la seva gestió

**Disseny, implementació i implantació d'un  
servidor AS2**

Alumne  
*Joaquim Cirac del Toro*

Professor Ponent  
*Albert Fornells Herrera*

---

# ACTA DE L'EXAMEN DEL TREBALL FI DE CARRERA

---

Reunit el Tribunal qualificador en el dia de la data, l'alumne

D. Joaquim Cirac del Toro

va exposar el seu Treball de Fi de Carrera, el qual va tractar sobre el tema següent:

Disseny, implementació i implantació d'un servidor AS2

Acabada l'exposició i contestades per part de l'alumne les objeccions formulades pels Srs. membres del tribunal, aquest valorà l'esmentat Treball amb la qualificació de

Barcelona,

VOCAL DEL TRIBUNAL

VOCAL DEL TRIBUNAL

PRESIDENT DEL TRIBUNAL

## ABSTRACT

L'intercanvi de documents entre empreses és un procés dut a terme de manera habitual des de molts anys enrere. Avui en dia i des de fa relativament pocs anys ha començat a les empreses una transició cap el món de la informàtica, aquest procés ha fet necessària la comunicació entre empreses a través de documents electrònics.

L'empresa eDiversa s'encarrega de facilitar i donar suport a l'intercanvi electrònic de documents entre els seus clients fent servir diferents tecnologies, entre les quals destaquen SMTP (*Simple Mail Transfer Protocol*), FTP (*File Transfer Protocol*) i X400. Aquest projecte consisteix en la implementació d'una nova tecnologia dins de la infraestructura de la empresa que permetrà oferir un nou mitjà de comunicació als clients. Aquesta nova tecnologia és AS2 (*Applicability Statement 2*), que és un tipus de comunicació basada en HTTP (*Hypertext Transfer Protocol*), i que es caracteritza per l'enviament i recepció de missatges a través d'un entorn web per encriptar i signar els missatges, així com registrar els missatges de servei per confirmar la seva correcta recepció.

Per tant es dissenyarà, implementarà i implantarà un nou servidor dins de l'empresa que permetrà enviar i rebre missatges de servidors AS2 d'altres xarxes i empreses. Tot i tractar-se d'un projecte intern permetrà oferir un nou servei als clients.

## RESUM

Quan el món empresarial es va modernitzar per començar a aprofitar les avantatges de la informàtica i d'Internet les empreses van començar a substituir els faxos i els documents impresos per documents electrònics. Per fixar aquests documents va sorgir un estàndard anomenat EDI (*Electronic Data Interchange*) que indicava el format que aquests fitxers electrònics havien de tenir, per tal de facilitar enormement l'automatització i integració dels processos gestors.

Fent servir el format EDI, l'empresa eDiversa s'encarrega d'interconnectar clients i permetre que aquests enviïn i rebin comandes, factures, i altres tipus de documents necessaris per dur a terme les transaccions comercials entre clients i proveïdors. Els clients de l'empresa tenen una compta FTP que fan servir per enviar els seus documents i rebre els documents entrants. També es fa servir la tecnologia SMTP (AS1 o *Applicability Statement 1*, que és el predecessor de AS2) per dur a terme intercanvis entre clients que fan servir una estació implementada per l'empresa anomenada comeDi, que permet una fàcil gestió de les entrades i sortides de missatges. Per comunicar clients amb usuaris de xarxes externes, és a dir, usuaris que no són de eDiversa, es fa servir la tecnologia X400.

Recentment ha sorgit un altre mètode d'intercanvi electrònic anomenat AS2, que es basa en fer servir HTTP per enviar i rebre informació, de la mateixa manera que s'envien variables a una pàgina web i aquesta ens retorna una resposta HTML. La diferència és que en aquest cas s'envia a un servidor web un document en format MIME (*Multipurpose Internet Mail Extensions*) i unes capçaleres HTTP. Aquest servidor processa aquesta informació i genera una resposta fent servir el mateix format MIME i, d'aquesta manera, es poden automatitzar enviaments i rebudes de missatges i es pot construir una plataforma basada en la tecnologia AS2. Aquesta plataforma ha de ser capaç de funcionar de manera desatesa i permetre fer arribar un document a un destinatari AS2 extern sense que el client de la nostra empresa hagi de fer cap canvi o tràmit.

La finalitat d'aquest projecte és construir una plataforma dins de l'empresa que permeti als usuaris de la nostra xarxa fer servir la tecnologia AS2 de manera totalment transparent per poder enviar i rebre documents. Per tant el projecte és intern dins de l'empresa però oferirà un nou servei a tots els nostres clients.

## ÍNDEX

1- Introducció	pàg.6
1.1-Objectius	pàg.7
1.2-Estructura de la memòria	pàg.8
2- Fonaments teòrics	pàg.9
2.1- Glossari	pàg.9
2.1.1- MINE	pàg.9
2.1.2- X400	pàg.9
2.1.3- Partner	pàg.10
2.1.4- POST	pàg.10
2.1.5- SMTP	pàg.11
2.1.6- DNS i MDN	pàg.11
2.1.7- 3DES	pàg.11
2.1.8- CCED	pàg.12
2.2- El format EDI	pàg.12
2.3- AS2	pàg.13
3- Disseny	pàg.16
3.1- Requeriments del projecte	pàg.16
3.2- Diagrama de blocs	pàg.19
3.3- Diagrama de casos d'ús	pàg.20
3.3.1- Diagrama de casos d'ús de la infraestructura de l'empresa	pàg.20
3.3.2- Diagrama de casos d'ús del servidor AS2	pàg.24
3.4- Diagrama entitat-relació	pàg.27
3.5- Diagrama de seqüència	pàg.28
3.5.1- Diagrama de seqüència del procés general	pàg.28
3.5.2- Diagrama de seqüència del servidor AS2	pàg.29
3.6- Diagrama de classes	pàg.29
3.6.1- Diagrama de classes dels blocs Client i Servidor	pàg.30
3.6.1- Diagrama de classes del bloc CCED AS2	pàg.31
4- Implementació	pàg.33
4.1- Eines emprades	pàg.33
4.1.1- PERL	pàg.33
4.1.2- PHP	pàg.35
4.1.3- Apache	pàg.36
4.1.4- MySQL	pàg.36
4.1.5- Crontab	pàg.39
4.2- Implementació dels mòduls	pàg.40
4.2.1- Mòdul PERL: procés CCED AS2	pàg.40
4.2.2- Mòdul PHP: llibreria AS2	pàg.46
4.2.3- Procés de reenviament	pàg.48
4.2.4- Clients genèrics PHP	pàg.51
5- Avaluació del servidor AS2	pàg.53
5.1- Eines emprades per avaluar el servidor AS2	pàg.53
5.1.1- ComeDi	pàg.53
5.1.2- Mendelson AS2 Free Client	pàg.53
5.1.3- Logs del sistema	pàg.54
5.1.4- Logs del mòdul PHP AS2	pàg.54
5.1.5- PhpMyAdmin	pàg.54
5.2- Proves	pàg.55
5.2.1- Enviament de missatge AS2 amb MDN asíncron	pàg.56

## Disseny, implementació i implantació d'un servidor AS2

5.2.2- Enviament de missatge AS2 amb MDN síncron	pàg.59
5.2.3- Enviament de missatge AS2 erroni	pàg.59
5.2.4- Recepció d'un missatge AS2 amb MDN asíncron	pàg.63
5.2.5- Recepció d'un missatge AS2 amb MDN síncron	pàg.65
5.2.6- Recepció de missatges erronis	pàg.66
5.3- Conclusions del procés d'avaluació	pàg.67
6- Estudi econòmic	pàg.69
7- Conclusions i línies de futur	pàg.70
8- Bibliografia	pàg.72

## ÍNDIX DE FIGURES

Figura 1- Esquema de l'enviament d'un missatge AS2	pàg.7
Figura 2- Esquema de l'enviament d'un missatge a AS2	pàg.14
Figura 3- Esquema de rebuda d'un missatge AS2	pàg.14
Figura 4- Esquema de la infraestructura de l'empresa	pàg.17
Figura 5- Esquema del servidor AS2	pàg.17
Figura 6- Diagrama de blocs del servidor AS2	pàg.19
Figura 7- Diagrama de casos d'ús de la infraestructura de l'empresa	pàg.20
Figura 8- Diagrama de casos d'ús del servidor AS2	pàg.24
Figura 9- Diagrama entitat-relació de la base de dades	pàg.27
Figura 10- Diagrama de seqüència del procés general	pàg.28
Figura 11- Diagrama de seqüència del servidor AS2	pàg.29
Figura 12- Diagrama de classes dels blocs Client i Servidor amb atributs i funcions	pàg.30
Figura 13- Diagrama de classes dels blocs Client i Servidor amb relacions	pàg.31
Figura 14- Diagrama de classes de CCED AS2	pàg.32
Figura 15- Crida a un servidor HTTP en PERL	pàg.35
Figura 16- Taula comparativa de gestors de bases de dades	pàg.38
Figura 17- Exemple de Crontab	pàg.39
Figura 18- Fitxer de configuració del CCED AS2	pàg.42
Figura 19- Exemple d'un missatge d'entrada en format XML	pàg.43
Figura 20- Exemple de les capçaleres HTTP d'un missatge AS2	pàg.44
Figura 21- Funció handlemdn()	pàg.47
Figura 22- Part 1 de la funció de reenviament de missatges	pàg.49
Figura 23- Part 2 de la funció de reenviament de missatges	pàg.50
Figura 24- Exemple de comunicació de PERL amb PHP	pàg.51
Figura 25- Client genèric PHP	pàg.51
Figura 26- Exemple de missatge en format EDI	pàg.57
Figura 27- Log de la cua AS2_IN de CCED AS2	pàg.58
Figura 28- Traça de missatges a la taula tr_msg	pàg.58
Figura 29- Log de la cua RAW_AS2 de CCED AS2	pàg.58
Figura 30- Traça de missatges de la taula tr_intercambio	pàg.59
Figura 31- Log de la cua AS2_OUT de CCED AS2	pàg.59
Figura 32- Log de la cua AS2_IN de CCED AS2 amb problema d'entrega	pàg.61
Figura 33- Part 2 de log de la cua AS2_IN de CCED AS2 amb problema d'entrega	pàg.61
Figura 34- Log OUT_AS2 en cas de reenviament incorrecte	pàg.62
Figura 35- Traça de missatge reenviat incorrecte de la taula tr_msg	pàg.62
Figura 36- Traça de missatge reenviat incorrecte de la taula tr_intercambio	pàg.62
Figura 37- Enviament de missatge amb MDN asíncron de Mendelson	pàg.63
Figura 38- Log de la cua RAW_AS2 per missatge entrant	pàg.64
Figura 39- XML de sortida de AS2_OUT	pàg.64
Figura 40- Recepció del MDN asíncron a Mendelson	pàg.65
Figura 41- Log events.log	pàg.66
Figura 42- Recepció del MDN negatiu a Mendelson	pàg.67
Figura 43- Taula resum de l'estudi econòmic	pàg.69
Figura 44- Gràfic de estudi econòmic	pàg.69

## 1- INTRODUCCIÓ

Actualment a nivell estatal existeixen relativament poques empreses d'intercanvi electrònic de documents. No obstant, existeix molta competència en el sector ja que una sola empresa pot cobrir un gran nombre de clients, i molt rarament un client contractarà més d'un proveïdor de serveis EDI. Per tant és molt important oferir constantment noves funcionalitats i actualitzar contínuament l'estructura interna per poder mantenir el nivell de la competència.

El format EDI és un estàndard que permet generar fitxers electrònics totalment vàlids que poden substituir des de factures fins a xecs per tal de ser tractats sense intervenció de personal. Aquest tipus de fitxer de text va sorgir davant la necessitat d'establir un estàndard en el format dels documents electrònics que les empreses fan servir. La seva estructura ve definida per una sèrie de tags que defineixen parts dins del document i fixen informació com les dades d'emissor i receptor, el tipus de document, informació de les parts involucrades i el cos del document, entre altres. Aquest format pot ser enviat fent servir diferents tecnologies, com el FTP, SMTP (AS1), X400, HTTP (AS2).

- FTP<sup>[8]</sup> és un estàndard d'intercanvi de fitxers entre un client i un servidor. El servidor té un o més directoris on el client pot desar i recuperar fitxers de tot tipus. Normalment aquests directoris es fan servir de manera similar a una bústia de correu, on tenim un directori on el client deixa els fitxers que vol enviar i un altre directori on el client recull els fitxers que li han enviat a ell.
- SMTP<sup>[7]</sup> és un estàndard d'enviament de fitxers a correu electrònic. A diferència del FTP on hi ha un client i un servidor, en SMTP la connexió és entre dos clients, on un envia un correu electrònic a l'altre. Les capçaleres i la part del cos del missatge que pertanyen a l'estàndard actuen com un sobre que indica tant el seu origen com el seu destí.
- X400<sup>[10]</sup> és un estàndard d'intercanvi de fitxers entre xarxes, que actualment comença a estar en desús però que encara es fa servir en el món EDI.
- HTTP<sup>[9]</sup> és l'estàndard que regeix el funcionament les pàgines web. En aquest cas ens centrem en una de les funcionalitats d'aquest estàndard que permet enviar i rebre informació entre un client i un servidor web. Aprofitant aquesta funcionalitat es va definir un protocol d'intercanvi de fitxers fent servir la tecnologia HTTP i que es va anomenar AS2.

AS2<sup>[3]</sup> és un mètode d'intercanvi de documents basat en AS1 (SMTP) i que fa servir HTTP per dur a terme aquestes comunicacions. La figura 1 descriu a grans trets el procés d'enviament d'un missatge entre dos usuaris AS2.



## Disseny, implementació i implantació d'un servidor AS2

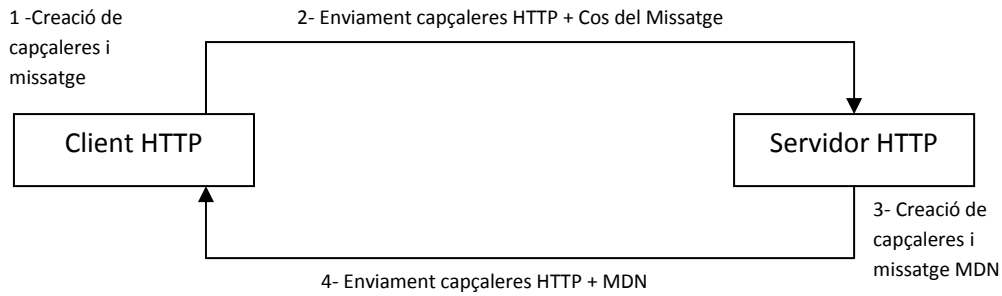


Figura 1 - Esquema de l'enviament d'un missatge AS2

1 - L'emissor crea un document en format AS2, que consta d'unes capçaleres HTTP i un cos fent servir el format MIME<sup>[3]</sup>.

2 - L'emissor obre una connexió HTTP amb una adreça URL corresponent al servidor web del receptor, puja les capçaleres HTTP anomenades anteriorment i envia pel mètode POST el cos del missatge en format MIME.

3 - El servidor web del receptor rep aquesta informació i la processa i genera una resposta (MDN) que indica si el missatge ha estat rebut correctament o no.

4 - Aquesta resposta és un missatge AS2 que el emissor rep de manera idèntica que en el procés d'enviament del missatge original. Un missatge MDN pot ser síncron si es produeix dins de la mateixa connexió HTTP, o asíncron si es produeix en diferents connexions.

Per tant el problema que es presenta a la nostra empresa és el següent: els clients demanen poder enviar i rebre missatges fent servir la tecnologia AS2 i cal que nosaltres donem suport a aquesta demanda creixent.

### 1.1 Objectius

L'objectiu del projecte és permetre als clients de l'empresa enviar i rebre documents a través d'AS2.

Els sub-objectius del projecte són:

- Aconseguir implementar una infraestructura capaç de comunicar-se amb clients AS2 fent servir totes les possibles característiques i funcionalitats que l'estàndard AS2 ofereix.
- Integrar aquesta infraestructura dins de l'empresa per tal que un client la pugui fer servir de manera totalment transparent.
- Afegir eines de traçabilitat i seguiment de missatges, necessàries per poder respondre a dubtes i peticions de clients, així com per poder resoldre incidències.
- Crear una eina d'administració per poder gestionar el nou servei AS2 de manera fàcil i des de qualsevol ordinador de l'oficina.

## Disseny, implementació i implantació d'un servidor AS2

Aquest projecte es desenvolupa internament a l'empresa, és a dir, no es ven com a producte a cap client, però implantar-lo dins de l'empresa permetrà oferir als clients un nou servei que suposa una avantatge competitiva davants d'altres empreses. A més cal remarcar que tot i ser un projecte intern de l'empresa jo he de dur a terme totes les fases de disseny, implementació i implantació del servidor AS2.

### 1.2 Estructura de la memòria

La memòria del projecte es pot dividir en 8 capítols. El primer capítol descriu el marc que envolta el projecte, el problema que cal resoldre, com es vol resoldre, les parts implicades en el desenvolupament del projecte i quins objectius es volen assolir amb aquest.

El capítol 2 explica la part teòrica que envolta el projecte.

El capítol 3 planteja el disseny de la solució proposada per resoldre el problema.

El capítol 4 descriu la implementació de les parts descrites a la fase de disseny.

El capítol 5 avalua el resultat de la fase d'implementació, mostrant les diferents proves realitzades i exposant les conclusions extretes de les mateixes.

El capítol 6 resum el temps emprat en cadascuna de les fases de desenvolupant i el cost associat a cada fase.

El capítol 7 conclou amb i planteja futures fites no incloses al projecte que podrien ser afegits posteriorment.

Finalment es mostra la bibliografia del projecte, on s'indica les fonts d'informació que s'han fet servir per dur a terme el seu desenvolupament.

## 2- FONAMENTS TEÒRICS

Aquest apartat defineix el context i marc teòric que envolta el projecte. Primerament es mostra un glossari de tots els elements que intervenen en el context teòric i una explicació de cada un d'ells. A continuació s'explica el format que l'empresa fa servir per intercanviar els documents, el format EDI. Seguidament es concreta el funcionament de l'estàndard AS2 i dels elements que fa servir en el procés d'enviament de fitxers. A continuació es planteja la problemàtica de la conversió dels missatges de servei que els diferents estàndards implicats al projecte fan servir. Finalment es resum el procés complet d'enviament d'un missatge a través de totes les tecnologies comentades anteriorment.

### 2.1- Glossari

Aquest apartat descriu tots els elements que intervenen en els apartats posteriors de tal manera que es pugui entendre correctament el context que envolta al projecte.

#### 2.1.1- MIME

MIME<sup>[3]</sup> és l'acrònim anglès de *Multi-Purpose Internet Mail Extensions*, "Extensions de correu Internet multipropòsit".

És un estàndard per adjuntar arxius a missatges de correu electrònic, com ara gràfics, documents de processadors de text, arxius de so, etc. Quan s'envien arxius binaris, aquests es converteixen (codifiquen) a text usant l'estàndard MIME, el qual es pot veure en format text tot i ser incompreensible. L'estàndard MIME és una manera d'especificar tant el tipus d'arxiu que s'envia com el mètode que s'hauria d'usar per retornar-lo al seu format original. A més de programari de correu electrònic, l'estàndard MIME s'usa per identificar els arxius que s'envien a clients Web, nous formats de fitxers es poden acomodar simplement actualitzant la llista de browsers de parells de tipus MIME i el programari apropiat per a gestionar cada tipus.

La IETF (Internet Engineering Task Force) va començar a desenvolupar aquesta norma al 1991 i des de 1994 totes les extensions MIME són especificades de forma detallada en diversos documents oficials disponibles a Internet (RFC 2045-2049).

Actualment cap programa de correu electrònic o navegador d'Internet pot considerar-se complet si no accepta MIME en les seves diferents formes (text i formats d'arxiu).

#### 2.1.2- X400

X400<sup>[10]</sup> és un estàndard del model de referència d'Interconnexió de Sistemes Oberts OSI, per a l'intercanvi de correu electrònic (aleshores se'n deia Missatge Interpersonal o IPMs) desenvolupat pel Sector de Normalització de les Telecomunicacions de la UIT amb el permís de l'ISO des de l'any 1984 .

Com va passar amb la majoria dels estàndards OSI de Nivell d'aplicació no va suportar la competència amb el protocol similar Internet, en aquest cas el SMTP. El correu X400 va arribar a tenir una base d'usuaris relativament àmplia, especialment a Europa, sobretot en entorns corporatius i d'investigació. El model de correu era més robust i complet que l'equivalent d'Internet. El seu sistema d'adreces de correu era massa complicat per a l'època, encara que

## Disseny, implementació i implantació d'un servidor AS2

molt més potent. Com tots els estàndards OSI, aquest era el recomanat per les companyies telefòniques (a l'època i a Europa gairebé totes eren monopolis estatals) que ofertaven unes tarifes de connexió excessives. Per aquests motius l'estàndard OSI no va tenir gran acceptació. No obstant això encara es fa servir el correu X400 en algunes aplicacions sectorials que requereixin major seguretat i integritat (com aplicacions militars), i és el model que hi ha per sota d'aplicacions relativament populars com Lotus Notes.

Algunes característiques excel·lents de X400 eren la separació entre contingut i sobre, les adreces estructurades, la possibilitat de contingut multimèdia (a MIME) i el tractament integral del xifrat i autenticació.

### 2.1.3- Partner

Representa un client AS2, a cada estació cal tenir representat tant el partner destí com el partner local, és a dir, l'emissor. La manera de representar els partners pot variar enormement depenent del software de gestió emprat. En el nostre cas un partner ve determinat per un arxiu .conf i un directori amb el seu nom que contindrà el seu certificat o clau privada depenent de si es local o no.

Els atributs que determinen un partner venen descrits a l'arxiu .conf corresponent i són:

- Nom: nom del partner. No és vital que sigui únic, ja que el seu ús es local.
- Id: és l'identificador del partner dins de la xarxa AS2, per tant cal que sigui únic i que el servidor AS2 on s'envia el missatge tingui configurat un partner amb aquesta mateixa id.
- Key/certificat: una transferència AS2 fa servir normalment un xifrat 3DES i una signatura digital fent servir un certificat. Per tant quan un partner és local cal tenir una key privada al seu directori corresponent, en canvi quan un partner no és local només cal tenir el seu certificat (clau pública) per validar la seva firma.

### 2.1.4- POST

Quan es vol enviar informació a una pàgina web es pot fer per 2 mètodes: GET i POST. El mètode GET consisteix a enviar variables i el seu valor fent servir l'adreça URL de la pàgina web:

```
http://www.exemple.es/index.php?var1=1&var2=2
```

En aquest exemple s'estarien enviat dues variables a la pàgina web index.php que es troba al servidor representat per l'adreça "http://www.exemple.es". En aquest exemple s'estarien enviant les variables "var1", amb valor "1", i la variable "var2" amb valor "2". El principal problema del mètode GET és que les variables i els seus valors són visibles per qualsevol usuari, cosa que crea un greu forat de seguretat.

El mètode POST, en canvi, envia les variables de manera oculta, cap usuari pot saber el seu valor ni el seu nom. A més el fet de no incloure el valor de les variables a la URL permet enviar variables que tenen salts de línia i altres caràcters especials que si s'enviessin per mètode GET

Disseny, implementació i implantació d'un servidor AS2

provocarien un error al navegador. Quan el servidor PHP rep aquestes variables per POST les converteix en un missatge de tipus AS2.

#### 2.1.5- SMTP

SMTP (*Simple Mail Transfer Protocol*)<sup>[7]</sup> és un estàndard que defineix l'enviament de fitxers a través del correu electrònic. Les capçaleres i la part del cos del missatge que pertanyen a l'estàndard actuen com un sobre que indica tant el seu origen com el seu destí, a més donen informació necessària per poder llegir el contingut del missatge, ja que aquest es sol encriptar.

Aquest estàndard fa servir un tipus de missatge especial, anomenat missatge de servei, que serveix per indicar l'estat d'un missatge enviat anteriorment. Els missatges de servei indiquen si un missatge ha arribat correctament al destinatari o no, i si el destinatari ha pogut obrir aquest missatge.

#### 2.1.6- DNS i MDN

Són els dos tipus de missatges de servei que fa servir l'estàndard SMTP. Dins d'aquests missatges hi ha un element que indica el missatge original al qual fan referència, indicant el seu estat.

El missatge DSN (*Delivery Status Notification*) pot ser de dos tipus: positiu i negatiu. Un DNS positiu indica que el missatge al qual fa referència s'ha entregat correctament al destinatari, mentre que un DNS negatiu indica que hi ha hagut problema i no s'ha pogut entregar el missatge, a més es sol incloure el tipus d'error i una descripció del mateix al cos del missatge de servei.

El missatge MDN (*Message Disposition Notification*) es fa servir per notificar a l'emissor que el receptor del missatge ha pogut obrir el missatge correctament. Normalment l'enviament d'un missatge SMTP té dues fases, primerament es desa el missatge a la bústia SMTP del receptor i es genera el DSN notificant que s'ha pogut entregar el missatge, un cop el client receptor es connecta i es descarrega el missatge de la seva bústia es genera el missatge MDN notificant que el receptor ha pogut obrir el missatge.

AS2 ha heretat els missatges de servei de SMTP però amb alguns canvis: el missatge DSN desapareix, ja que no existeix cap bústia on es guardi el missatge fins que el destinatari el descarregui. El missatge MDN es manté però se li suma la funcionalitat que abans tenia el DNS d'indicar si hi ha hagut algun error o no. Per tant en enviar un missatge AS2 es pot rebre un MDN positiu si tot ha anat correctament o un MDN negatiu indicant l'error que hi ha hagut en el procés d'enviament que ha impedit fer arribar el missatge al receptor.

#### 2.1.7- 3DES

3DES (*Triple DES*) és un algoritme d'encriptació que consisteix en aplicar tres cops l'algoritme DES (*Data Encryption Standard*). Aquest algoritme va sorgir quan es va detectar que el mètode DES era insuficient per evitar un atac de força bruta sobre el missatge.

## Disseny, implementació i implantació d'un servidor AS2

En el cas de AS2 aquest algoritme s'aplica per xifrar el cos del missatge fent servir la clau pública del receptor del missatge. D'aquesta manera només el receptor real del missatge, que tindrà la clau privada, podrà llegir el missatge.

### 2.1.8- CCED

Dins de l'empresa és el nom que es dona a un procés que dirigeix un servei i gestiona el comportament de totes les parts implicades en aquest servei. A la infraestructura interna de l'empresa trobem molts CCED diferents:

- El CCED central, o procés central, és l'encarregat de dirigir tot el flux de missatges que passen per la xarxa de l'empresa. És qui s'encarrega de determinar a on cal enviar cada missatge entrant per tal que es pugui tractar adequadament. Compta amb nombroses cues de missatges, depenent de la seva procedència, i crida a diferents subrutines que gestionen cadascuna d'aquestes cues.
- El procés CCED que gestiona cada servei, com per exemple el CCED\_WEB que s'encarrega d'atendre els missatges que es volen pujar a l'aplicació web de l'empresa. El seu comportament és semblant al del CCED central però el seu objectiu és el d'actuar com a pont d'enllaç entre el CCED central i el servei que gestiona. Per tant el servidor AS2 haurà de comptar amb un "CCED\_AS2" que dirigeixi el seu comportament, ja que així ho fixa l'empresa.

### 2.2- El format EDI

EDI (*Electronic Data Interchange*), o Intercanvi Electrònic de Dades, es defineix com la transmissió d'un conjunt de dades coherents en un format normalitzat entre les estacions informàtiques participants en unes determinades transaccions. També podem dir que és una comunicació electrònica entre dos equips que es transmeten documents electrònics eliminant l'ús del paper i automatitzant els processos de comerç electrònic entre empreses públiques o privades amb una mínima intervenció humana.

L'EDI permet incorporar un procediment de transmissió d'informació per estalviar:

- Temps: ja que la informació viatja per xarxes de telecomunicacions.
- Errors: ja que pot ser un procés automatitzat.
- Diners: ja que s'estalvia la impressió dels documents.

El format normalitzat que es fa servir actualment a Espanya segueix la norma EDIFACT (*Electronic Data Interchange for Administrator Commerce and Transport*). Aquesta norma va ser creada per les Nacions Unides l'any 1986 per crear una norma de sintaxis internacionalment acceptable per la transmissió de missatges electrònics. EDIFACT ha sigut adoptada per la Organització Internacional de Normalització com la norma ISO 9735. La norma EDIFACT estableix:

- Un conjunt de regles de sintaxis per l'estructura de les dades.
- Un protocol d'intercanvi interactiu.

## Disseny, implementació i implantació d'un servidor AS2

- Missatges estàndards que permeten intercanvis entre diferents països y diferents tipus d'indústries.

El format AS2 crea un embalatge electrònic que pot contenir tota mena de informació en format text, per tant el fet de que en aquest cas es facin servir documents EDI no limita el seu ús només a aquest format.

### 2.3- AS2

AS2<sup>[3]</sup> és un protocol de comunicació creat a partir del model SMTP (AS1). A diferència d'aquest altre model, AS2 fa servir HTTP com a mitjà de transferència de missatges.

El procés és molt semblant a obrir una pàgina web amb el navegador:

Quan es vol enviar un missatge, el client AS2 encapsula aquest missatge fent servir el format MIME i s'envia per POST a una url fent servir unes capçaleres HTTP que determinen dades referents al missatge AS2.

El servidor AS2, que és un servidor web comú, rep aquesta informació per POST i l'extreu de l'embalatge per poder tractar-la. Com a resposta es genera un MDN, que és un missatge de servei que indica si el missatge ha estat rebut correctament o no.

Aquesta és la idea bàsica d'una transferència AS2. Ara es concretarà més detalladament aquest procés.

El procés d'enviament i validació és el següent:

- 1 - Primerament el missatge s'encapsula en un contenidor MIME. Seguidament es generen unes capçaleres HTTP, que indiquen dades com la mida del missatge, el tipu de xifrat, dades MIME per poder ser llegit, els partners implicats, etc. Aquestes capçaleres s'envien a una la url del receptor del missatge (es carreguen amb un HTTPRequest), seguidament s'envia per POST el missatge MIME a aquesta mateixa url, que prèviament ha estat signat i xifrat, fent servir la clau privada local.

- 2 - El receptor del missatge rep el paquet MIME per post, el desxifra i comprova la firma fent servir la nostra clau pública. Si tot ha estat correcte, envia un missatge MDN positiu a l'emissor per notificar la correcte entrega, si hi ha hagut algun error, envia un MDN negatiu. Normalment els missatges MDN es signen però no es xifren.

Per tal de poder entendre correctament el funcionament del servidor AS2, cal concretar exactament el camí que seguirà un missatge dins de la infraestructura interna de l'empresa. Cal deixar clar que un client de l'empresa enviarà un missatge cap a un destinatari AS2 de la mateixa manera que envia un missatge cap a SMTP, X400 o FTP, sense fer cap canvi, de fet un client no cal ni que sàpiga que el missatge serà enviat per AS2. La infraestructura de l'empresa no es canviarà per adaptar-se a l'implantació del nou servidor AS2, el servidor AS2 s'ha de

## Disseny, implementació i implantació d'un servidor AS2

dissenyar per poder funcionar en el context descrit a continuació, que és inalterable i fixat per l'empresa.

- Enviar missatge a AS2:

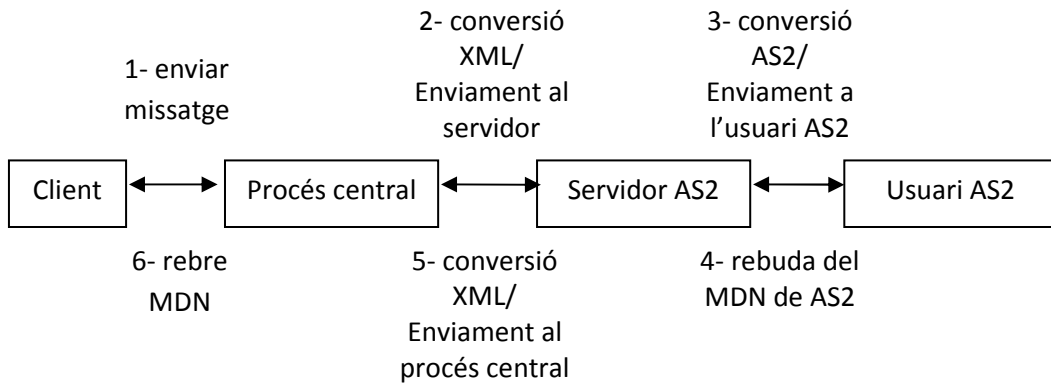


Figura 2 – Esquema d'enviament d'un missatge a AS2.

A la figura 2 es pot observar com, ja sigui per SMTP/X400 o FTP, els missatges entrants enviats pels clients de l'empresa acaben passant per un procés central, que gestiona totes les cues d'entrada, valida els missatges, i els entrega a les cues sortints corresponents. Per tant aquest procés llegirà el missatge entrant, determinarà que l'interlocutor destí és un usuari AS2 i el col·locarà a la cua de sortida corresponent.

Des d'aquesta cua s'envien els missatges a un directori del servidor AS2, ja convertits a format XML, on un procés anomenat CCED els llegirà, validarà novament (des de punt de vista AS2, no es repeteixen les mateixes validacions), i de ser tot correcte enviarà el missatge fent servir el client AS2. Ja sigui de forma síncrona o asíncrona es rebirà un MDN (positiu o negatiu) i es deixarà a una cua de sortida del bloc AS2, el CCED llegirà aquest MDN i generarà un missatge XML cap el procés central, que enviarà aquest missatge de servei a l'origen del missatge original.

- Missatge de AS2 a SMTP/X400/FTP:

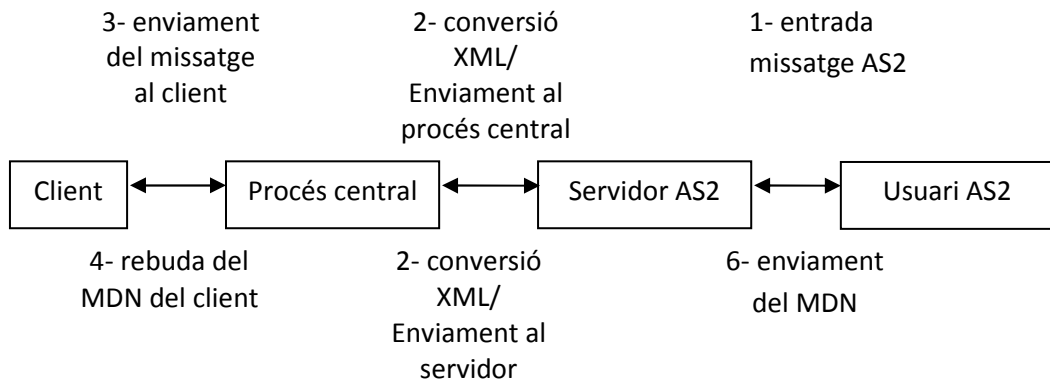


Figura 3 - Esquema de la rebuda d'un missatge de AS2.

A la figura 3 es pot observar el procés de recepció d'un missatge procedent d'un usuari AS2. Cal distingir dos casos: MDN síncron o asíncron.



## Disseny, implementació i implantació d'un servidor AS2

- Si el MDN es síncron: un usuari AS2 envia un missatge al nostre servidor AS2, a les capçaleres d'aquest missatge s'indica que el MDN ha de ser síncron, és a dir, s'ha de respondre a la mateixa connexió HTTP. El bloc AS2 crida a un script que fa algunes validacions al missatge entrant i genera un MDN positiu o negatiu depenent del resultat d'aquest. El MDN es generat com un HTTPResponse, per tant, s'envia a la mateixa connexió HTTP. El missatge és enviat a la cua de missatges entrants de AS2, el CCED el converteix a format XML i l'envia al procés central, que finalment el fa arribar al destinatari.
- Si el MDN es asíncron: un usuari AS2 envia un missatge al nostre servidor AS2, les capçaleres del missatge indiquen que requereix un MDN asíncron. El bloc AS2 deixa el missatge al directori d'entrada de AS2, el CCED el recull, fa les validacions i el converteix en XML abans d'enviar-lo al procés central, que el farà arribar al destinatari. El CCED esperarà a rebre tots els missatges de servei corresponents per poder generar el MDN AS2, ja que cal recordar que dins d'un missatge AS2 poden haver més d'un missatge EDI, i per tant cal esperar a tots els DSN/MDN EDI corresponents.

### 3- DISSENY

En aquest capítol es descriurà el disseny de tot el servidor AS2. Primerament es definiran els requeriments que cal complir, que han estat fixats per l'empresa. Seguidament es mostrarà i explicarà el diagrama de blocs que representa a grans trets tot el conjunt del projecte. Després es detallarà els diagrames de casos d'ús, començant primer per un diagrama que engloba tota la infraestructura de l'empresa i després un altre que concreta només la part del servidor AS2. Seguidament es descriurà el diagrama entitat-relació de la base de dades que fa servir el servidor AS2, així com una breu explicació de la funció de cada taula. Finalment es mostraran els diagrames de classes de les parts implicades.

#### 3.1- Requeriments del projecte

Un servidor AS2 pot actuar de moltes maneres diferents. Per exemple podríem tenir un servidor funcional però que requereix ser operat de manera manual, o en canvi un servidor que actués de manera autònoma però que no comptés amb totes les funcionalitats que especifica l'estàndard. Aquests requeriments normalment són fixats pel client que encarrega el projecte, en aquest cas el client és la pròpia empresa ja que es tracta d'un projecte intern.

L'empresa ha fixat els següents requeriments:

- El servidor ha de ser totalment autònom, no ha de requerir la supervisió de cap persona que supervisi les seves accions.
- El servidor ha de ser totalment funcional, és a dir, ha de poder interpretar i executar totes les possibles casuístiques que l'estàndard especifica.
- El servidor ha d'estar integrat dins de l'estructura interna de l'empresa, és a dir, ha de comunicar-se adientment amb la resta de servidors i processos de l'empresa.
- El procés d'enviament d'un missatge cap a un usuari de AS2 ha de ser idèntic pel client que qualsevol altre tipus d'enviament.
- Els missatges que entren i surten del servidor han d'estar correctament controlats, i s'ha de poder veure una llista dels missatges i del seu estat (correctes, incorrectes, esperant missatge de servei, etc).
- El servidor ha de poder funcionar tant si es troba a la mateixa màquina que la resta de processos com si està en una màquina diferents.

A més de les característiques del servidor, l'empresa també fixa les tecnologies i eines que s'han de fer servir a l'hora de la seva implementació. Tot i que alguna d'elles pot variar si això suposés un avantatge. Tot i això el projecte té alguns requeriments que no han estat fixats per l'empresa, sinó que són necessitats de disseny. Per entendre aquestes necessitats cal plantejar primer què suposa la implantació d'aquest nou servidor dins de l'empresa.

Es vol incorporar una nova tecnologia a l'empresa per tal que els nostres clients puguin enviar missatges fent-la servir. De la mateixa manera que es poden enviar missatges per correu

## Disseny, implementació i implantació d'un servidor AS2

electrònic o fent servir una compta FTP, l'objectiu principal d'aquest projecte és que un client de l'empresa pugui enviar informació a un usuari AS2 sense que hagi de fer cap mena de canvi a la seva estació client.

Per tal d'aconseguir això cal dissenyar un "servidor AS2", que faci de pont entre aquests usuaris AS2 externs i la resta de la infraestructura de l'empresa. Tot i anomenar-se "servidor AS2" no es tracta d'una màquina física independent, es tracta d'un conjunt de mòduls que es poden allotjar a qualsevol màquina i fàcilment es puguin configurar per treballar conjuntament amb la resta de l'empresa.

Per tal d'entendre el disseny del projecte, primer es descriurà l'estructura global que l'envolta.

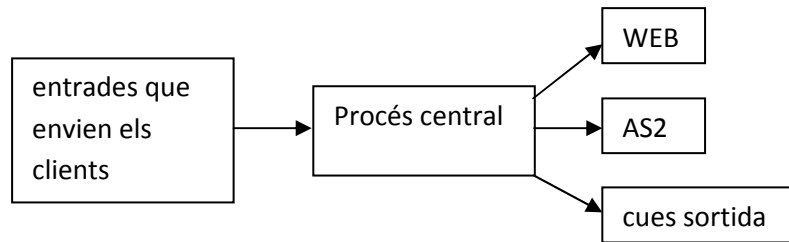


Figura 4 - Esquema de la infraestructura de l'empresa.

Com es pot veure a la figura 4, el bloc AS2 és dissenyat com un gateway de la infraestructura interna de l'empresa per on es podran encaminar missatges de sortida, però a més a més també serà una nova font d'entrada de missatges. Dic que està dissenyat com un gateway ja que és un dels possibles punts per on un missatge que un client de l'empresa envia pot sortir fora cap a un destinatari extern. Altres mecanismes tipus gateway de l'empresa serien per exemple X400, que es fa servir per comunicar-se amb altres empreses amb xarxes EDI, o la WEB, on usuaris poden accedir per consultar i generar documents.

A la figura 4 podem observar com tenim diferents d'entrades enviades pels clients de l'empresa, independentment de l'estàndard que fan servir per enviar els missatges. Totes aquestes entrades passen per un procés central que determina a quin bloc de sortida han d'anar a parar. Si el missatge ha de ser enviat a algun destinatari aquest anirà a les cues de sortida dels diferents clients de l'empresa. Si en canvi aquest missatge va a la pàgina web de l'empresa on els clients poden visualitzar-lo aquest serà enviat al servidor WEB. Si es tracta d'un missatge adreçat a un destinatari AS2 el missatge serà enviat al servidor AS2.

El servidor AS2 té dues fases:

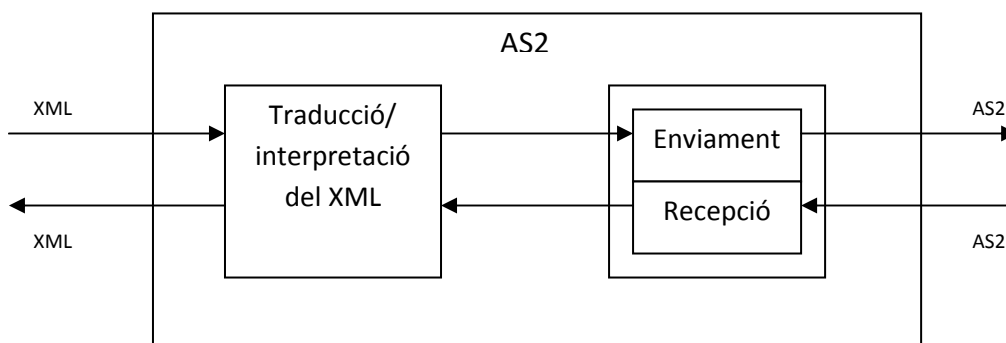


Figura 5 - Esquema del servidor AS2

## Disseny, implementació i implantació d'un servidor AS2

A la figura 5 es poden observar aquestes dues fases i com es comuniquen entre elles.

- Traducció/interpretació del XML: serveix com enllaç entre la part d'enviament i recepció de missatges AS2 i la resta de la infraestructura interna de l'empresa. Rep missatges en format XML i els interpreta i envia a la part d'enviament per poder ser enviats en format AS2. De la mateixa manera rep missatges AS2 de la part de recepció i els converteix i envia al procés central en format XML.
- Enviament/recepció: s'encarrega de rebre missatges entrants en format AS2 enviats per usuaris externs i d'enviar missatges de sortida cap a AS2.

El disseny es divideix en aquestes 2 fases ja que la intenció es deixar la part d'enviament i recepció com una caixa negra, que constituirà el core i la api amb les funcionalitats necessàries per dur a terme totes les tasques i complir tots els requeriments que marca l'estàndard. El motiu d'aquest modelat de caixa negra és que la part d'enviament i recepció està ubicada a uns directoris diferents que la fase de traducció i interpretació i per facilitar la gestió i la actualització del conjunt és molt més fàcil editar només la part de traducció mentre que el core amb les funcionalitats bàsiques resta sense modificar.

Per tant, després de veure tot el context que envolta el servidor AS2, podem veure que aquest projecte no és implementar un client AS2, es tracta d'integrar un nou servei dins de la infraestructura EDI de l'empresa. Aquest fet té un impacte especialment important en els missatges de servei de AS2 i el seu tractament dins de la xarxa de l'empresa. Un missatge AS2 rebut serà convertit i tractat com un missatge X400 o SMTP. En fer això sorgeixen 2 problemes:

- L'estàndard AS2 permet enviar més d'un arxiu dins d'un sol missatge: és a dir una sola transmissió AS2 pot contenir per exemple 4 missatges EDI, per tant cal esperar a que cada un d'aquests missatges (que poden anar a diferents receptors) sigui confirmat per enviar el MDN AS2 a l'emissor.
- A SMTP i X400 existeixen 2 missatges de servei, a AS2 només 1: a SMTP i a X400 quan s'envia un missatge, es genera un missatge de servei anomenat DSN (DR a X400) que notifica que el missatge ha estat rebut a la xarxa destí sense errors (en cas d'error serà un DSN/DR negatiu), a més un cop l'usuari destí obre el missatge es genera un MDN (RR a X400) per tal de notificar que l'usuari final ha rebut el missatge correctament. A AS2 aquest últim tipus de missatge de servei no existeix i s'anomena MDN al DSN de SMTP.

Per tant cal prendre una decisió per solucionar aquest intercanvi. Hem decidit que en el cas dels missatges AS2 s'ignorarà el MDN/RR de SMTP o X400 i es notificarà només l'entrega del missatge fent servir el MDN de AS2.

A més cal solucionar un altre problema, els missatges MDN de AS2 poden ser de 2 tipus:

- Síncron: si es respon en la mateixa connexió HTTP en la qual s'envia el missatge original. En principi el missatge de servei és recollit pel servidor que va generar el missatge original i no es crea cap missatge físic representat aquest missatge de servei, ja que ja es disposen de totes les dades per poder relacionar-lo amb el missatge original al qual contesta.

## Disseny, implementació i implantació d'un servidor AS2

- Asíncron: si es respon amb una connexió HTTP posterior a l'enviament del missatge original. Com que no es duu a terme en una connexió diferent cal crear-lo en forma de fitxer físic i enviar-lo al procés d'enviament/recepció de missatges per poder relacionar-lo amb el missatge original del qual notifica l'estat.

Però per qualitat de servei cal fer arribar un missatge de servei al nostre client sempre, notificant si el missatge ha arribat correctament al seu destinatari o no. Per solucionar això es decideix que qualsevol missatge de servei serà tractat internament com asíncron. És a dir, que quan el nostre servidor AS2 envia un missatge a un destinatari i especifica que la resposta a aquest missatge ha de tenir resposta síncrona espera la resposta del MDN síncron a la mateixa connexió HTTP, però quan el rep genera i propaga aquest missatge de servei cap el procés central de l'empresa per fer-lo arribar al nostre client, com si es tractés d'un missatge de servei asíncron.

### 3.2- Diagrama de blocs

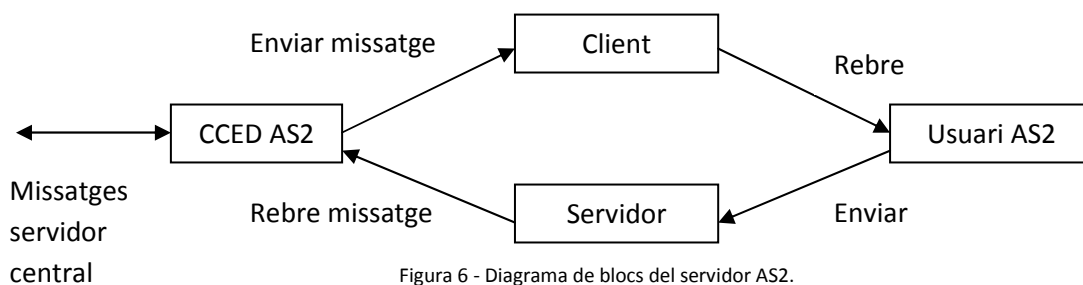


Figura 6 - Diagrama de blocs del servidor AS2.

A la figura 6 podem tornar a observar l'estructura detallada anteriorment a la figura 4, separant els blocs funcionals.

Tenim la part de traducció i interpretació dels XML, representada pel bloc CCED AS2, que s'encarrega d'actuar com un pont entre el servidor AS2 i la resta de l'empresa. Per altra banda tenim la fase d'enviament i recepció, representat pels blocs Client i Servidor, que s'encarrega d'enviar i rebre missatges respectivament.

Quan un usuari de AS2 envia un missatge, aquest és rebut pel Servidor, que el tracta i l'envia al bloc CCED AS2.

Si el bloc CCED AS2 vol enviar un missatge a un usuari AS2, l'envia al bloc Client que envia el missatge en forma de variable per POST a la pagina web (servidor) del destinatari AS2. Qui s'encarrega d'executar el client PHP és el bloc CCED AS2, ja que és qui actua com a pont entre l'empresa i la resta del servidor AS2. A continuació detallaré més a fons el funcionament del bloc CCED AS2 i la seva interacció amb la resta de blocs.

El bloc CCED AS2 rep missatges del procés central en format XML, fent servir les dades d'aquest XML envia una petició d'enviament de missatge AS2 al bloc Client. El Client rep aquestes dades i envia un missatge en format AS2 al bloc Usuari AS2. De la mateixa manera, el bloc Usuari AS2 pot enviar un missatge en format AS2 al bloc Servidor, aquest bloc llegirà el missatge i el deixarà a una cua d'entrada del bloc CCED AS2. El bloc CCED AS2 llegirà el

## Disseny, implementació i implantació d'un servidor AS2

missatge en format AS2 de la cua i el convertirà en format XML per enviar-lo al servidor central.

Cal tenir en compte que un missatge de servei rep el mateix tractament que un missatge normal. Per tant el cicle complet d'enviament d'un missatge consisteix en l'enviament d'aquest i la posterior rebuda del missatge de servei corresponent, ja sigui positiu o negatiu.

### 3.3- Diagrama de casos d'ús

Primerament es descriurà el diagrama de casos d'ús de tota la infraestructura del sistema, on es concretarà a més quina part és la implementada per aquest projecte. A continuació es mostrarà el diagrama de la part referent al servidor AS2, és a dir, la part que s'ha implementat per aquest projecte.

#### 3.3.1- Diagrama de casos d'ús de la infraestructura de l'empresa

La figura 7 mostra el diagrama de casos d'ús que engloba tant el projecte AS2 com la resta d'elements que formen la infraestructura de l'empresa. La part implementada per aquest projecte consisteix en els casos d'ús "Enviar AS2", "Crea XML", "Validar dades" i "Enviar client AS2". La resta de casos d'ús no formen part del projecte en sí però ha calgut que siguin actualitzats per configurar adequadament la seva comunicació i comportament amb el servidor AS2.

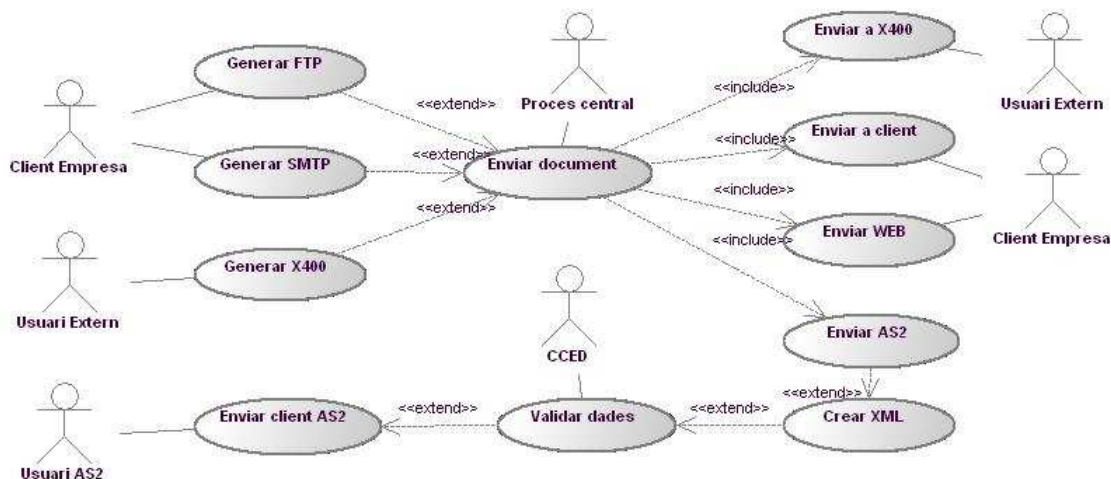


Figura 7 - Diagrama de casos d'ús de la infraestructura de l'empresa.

Com es pot veure, de tot el tràfic d'entrada que es rep, una part pot anar al bloc AS2, i aquest bloc s'encarrega de fer arribar el tràfic als usuaris AS2.

· Taules dels casos d'ús:

Nom:	Generar FTP
Descripció:	El client crea un fitxer i el puja a la seva compta FTP de l'empresa per ser enviat.
Actors principals:	Client Empresa

Disseny, implementació i implantació d'un servidor AS2

Actors secundaris:	-
Precondicions:	El client fa servir el software de l'empresa o un software equivalent.
Postcondicions:	El missatge és guardat a la bústia FTP del client
Flux principal:	<ol style="list-style-type: none"> <li>1- El client selecciona un missatge per enviar.</li> <li>2- El client prem el botó d'enviar.</li> <li>3- El programa client es connecta a la compta FTP corresponent a la seva bústia.</li> <li>4- El programa envia per FTP el missatge i aquest és guardat a la bústia.</li> </ol>
Excepcions:	<ol style="list-style-type: none"> <li>3a- Error de connexió.</li> <li>3b- El programa torna a connectar posteriorment.</li> </ol>

Nom:	Generar SMTP
Descripció:	El client fent servir el software de l'empresa envia un missatge a un interlocutor extern.
Actors principals:	Client Empresa
Actors secundaris:	-
Precondicions:	El client fa servir el software de l'empresa i prem el botó d'enviar missatge.
Postcondicions:	El missatge és dipositat a la cua SMTP.
Flux principal:	<ol style="list-style-type: none"> <li>1- El client selecciona un missatge i prem el botó traduir.</li> <li>2- Posteriorment va al directori de missatges traduïts, en selecciona un i prem enviar.</li> <li>3- El programa afegeix les capçaleres SMTP adients i envia el missatge a l'adreça SMTP de l'empresa.</li> <li>4- El missatge es rebut i dipositat a la cua de missatges SMTP.</li> </ol>
Excepcions:	<ol style="list-style-type: none"> <li>2a- Error de traducció.</li> <li>2b- Es mostra l'error a l'usuari i es torna al pas 1.</li> </ol>

Nom:	Generar X400
Descripció:	El MTA de l'empresa rep un missatge X400 d'un client extern.
Actors principals:	Usuari extern
Actors secundaris:	-
Precondicions:	L'usuari extern ha enviat un missatge a través de X400.
Postcondicions:	El missatge és guardat a la cua de X400
Flux principal:	<ol style="list-style-type: none"> <li>1- El software client de l'usuari extern converteix el missatge EDI a un missatge x400.</li> <li>2- El software del client envia el missatge a l'adreça del MTA de l'empresa.</li> <li>3- El MTA de l'empresa rep el missatge.</li> <li>4- El missatge es converteix a EDI de nou i es guardat a la cua X400.</li> </ol>
Excepcions:	-

Nom:	Enviar document
Descripció:	Procés central envia un document a un dels diferents gateway de l'empresa.
Actors principals:	Client empresa / usuari extern.
Actors secundaris:	Procés central

## Disseny, implementació i implantació d'un servidor AS2

Precondicions:	El missatge ha estat enviat per SMTP, FTP o X400.
Postcondicions:	El missatge és enviat al gateway corresponent.
Flux principal:	<ol style="list-style-type: none"> <li>1- Es verifica el missatge EDI.</li> <li>2- Es verifica que els interlocutors existeixin i tinguin permisos.</li> <li>3- Es pugen les dades dels missatges a la BBDD.</li> <li>4- Fent servir la BBDD es determina quin tipus d'usuari és el receptor i es decideix el gateway.</li> <li>5- El missatge és enviat al gateway corresponent.</li> </ol>
Excepcions:	<ol style="list-style-type: none"> <li>1a- Error en l'estructura del EDI</li> <li>1b- S'envia un missatge de servei negatiu indicant l'error a l'emissor.</li> <li>2a- Error de permisos o de no existència d'interlocutor.</li> <li>2b- S'envia un missatge de servei negatiu indicant l'error a l'emissor.</li> </ol>

Nom:	Enviar a X400
Descripció:	El missatge és preparat i enviat a un destinatari X400
Actors principals:	Usuari Extern
Actors secundaris:	Procés central
Precondicions:	Arriba un missatge de sortida al MTA de l'empresa
Postcondicions:	El missatge és enviat al MTA del destinatari
Flux principal:	<ol style="list-style-type: none"> <li>1- Arriba un missatge de sortida al MTA de l'empresa.</li> <li>2- El missatge es converteix al format X400</li> <li>3- El missatge s'envia a l'adreça del MTA del destinatari.</li> </ol>
Excepcions:	-

Nom:	Enviar a client
Descripció:	El missatge es fa arribar al client de l'empresa
Actors principals:	Client Empresa
Actors secundaris:	Procés central
Precondicions:	Hi ha un missatge pendent d'enviar al client
Postcondicions:	El missatge arriba al client
Flux principal:	<ol style="list-style-type: none"> <li>1- El procés central agafa el missatge de la cua de sortida.</li> <li>2- El missatge es converteix a format SMTP o no depenent del mètode d'entrega.</li> <li>3- Si el mètode d'entrega és SMTP s'envia el missatge a l'adreça SMTP del client.</li> <li>4- Si el mètode d'entrega és FTP es deixa el missatge a la bústia FTP del client.</li> <li>5- El missatge arriba al client.</li> </ol>
Excepcions:	-

Nom:	Enviar Web
Descripció:	El missatge arriba al servidor web de l'empresa
Actors principals:	Client Empresa
Actors secundaris:	Procés central
Precondicions:	Hi ha un missatge pendent d'enviar a la web
Postcondicions:	El missatge arriba a la cua d'entrada del servidor web
Flux principal:	<ol style="list-style-type: none"> <li>1- El procés central agafa el missatge de la cua de sortida.</li> <li>2- El missatge es converteix en format XML.</li> </ol>



## Disseny, implementació i implantació d'un servidor AS2

	3- El missatge s'envia al servidor web. 4- El missatge arriba a la cua d'entrada del servidor web.
Excepcions:	-

Nom:	Enviar AS2
Descripció:	El missatge s'envia al servidor AS2
Actors principals:	Usuari AS2
Actors secundaris:	Procés central
Precondicions:	Hi ha un missatge pendent d'enviar a un usuari AS2
Postcondicions:	El missatge arriba al servidor AS2.
Flux principal:	1- El procés central agafa el missatge de la cua de sortida. 2- El missatge es converteix a format XML. 3- El missatge s'envia al servidor AS2. 4- El missatge arriba a la cua d'entrada del servidor AS2.
Excepcions:	-

Nom:	Validar dades
Descripció:	Es verifica que el contingut del missatge rebut és correcte.
Actors principals:	CCED
Actors secundaris:	-
Precondicions:	Arriba un missatge entrant
Postcondicions:	El missatge es verifica com a correcte o no.
Flux principal:	1- El CCED obre un missatge entrant. 2- Es comprova que les dades dels partners i del missatge EDI siguin correctes. 3- Es dona el missatge com a vàlid.
Excepcions:	2a- El missatge es detecta com erroni. 2b- S'envia un missatge de servei negatiu cap a l'emissor del missatge.

Nom:	Enviar client AS2
Descripció:	El missatge s'envia al servidor web del client AS2
Actors principals:	Usuari AS2
Actors secundaris:	-
Precondicions:	Hi ha un missatge pendent de ser enviat al client AS2
Postcondicions:	El missatge arriba al usuari AS2 correctament
Flux principal:	1- Es converteix el missatge, afegint les capçaleres HTTP corresponents i encapsulant-lo en format MIME. 2- S'envia el missatge a la url del servidor web del usuari AS2 en forma de variables passades per mètode POST. 3- El usuari AS2 rep aquestes dades i contesta amb un missatge de servei.
Excepcions:	-

### 3.3.2- Diagrama de casos d'ús del servidor AS2

La figura 8 mostra el diagrama de casos d'ús del servidor AS2, mostrant amb més detall tots els casos que formen el conjunt dels blocs CCED AS2, Client i Servidor descrits al diagrama de blocs de la figura 6.

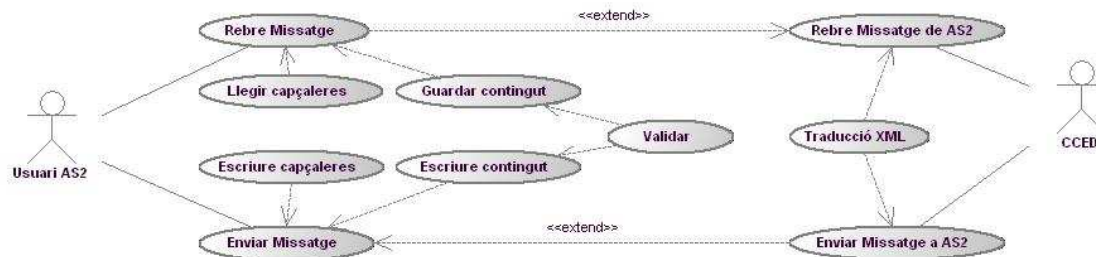


Figura 8 - Diagrama de casos d'ús del servidor AS2.

· Taules dels casos d'ús:

Nom:	Enviar Missatge a AS2
Descripció:	Un missatge d'entrada en format XML es prepara per ser enviat a AS2
Actors principals:	CCED
Actors secundaris:	-
Precondicions:	Hi ha un missatge a la cua d'entrada de AS2
Postcondicions:	La informació necessària per enviar el missatge està llesta.
Flux principal:	<ol style="list-style-type: none"> <li>1- Extreure informació del XML.</li> <li>2- Buscar a la base de dades la relació dels punts operacionals amb els partners corresponents.</li> <li>3- Buscar a la base de dades la configuració del partner.</li> <li>4- La informació necessària per enviar el missatge està llesta.</li> </ol>
Excepcions:	<ol style="list-style-type: none"> <li>2a- El partner no existeix.</li> <li>2b- Es retorna un missatge de servei negatiu per indicar l'error.</li> </ol>

Nom:	Traducció XML
Descripció:	Procés tant d'extreure informació d'un XML com de generar un XML nou a partir de les dades donades.
Actors principals:	CCED
Actors secundaris:	-
Precondicions:	Cal generar un XML o llegir la informació d'un XML
Postcondicions:	Dades llegides del XML o XML generat correctament.
Flux principal:	<ol style="list-style-type: none"> <li>1- Obrir el XML, per escriptura si es vol generar, o per lectura si es vol llegir.</li> <li>2- Guardar la informació del XML a les variables en el cas de lectura, o guardar la informació de les variables al XML en cas d'escriptura.</li> <li>3- Dades llegides del XML o XML generat correctament.</li> </ol>
Excepcions:	<ol style="list-style-type: none"> <li>1a- Format del XML incorrecte, el XML no es pot llegir</li> <li>1b- Es retorna un missatge de servei negatiu per indicar l'error.</li> </ol>

Nom:	Validar
------	---------

Disseny, implementació i implantació d'un servidor AS2

Descripció:	Procés que verifica que el contingut del missatge i la informació referent als partners és correcta.
Actors principals:	CCED
Actors secundaris:	-
Precondicions:	Missatge d'entrada o de sortida requereix ser validat.
Postcondicions:	La informació és validada.
Flux principal:	<ol style="list-style-type: none"> <li>1- Rebre informació a ser validada.</li> <li>2- Comprovar l'estructura del missatge EDI.</li> <li>3- Comprovar que els punts operacionals del missatge tenen un partner corresponent</li> <li>4- La informació és validada.</li> </ol>
Excepcions:	2,3a – Qualsevol error genera un missatge negatiu.

Nom:	Guardar contingut.
Descripció:	Procés que a partir d'un missatge AS2 obté el seu contingut (payload)
Actors principals:	CCED
Actors secundaris:	-
Precondicions:	Es necessita obtenir el contingut d'un missatge AS2.
Postcondicions:	El contingut del missatge és guardat a un fitxer.
Flux principal:	<ol style="list-style-type: none"> <li>1- Un missatge AS2 arriba al servidor.</li> <li>2- S'obtenen les capçaleres del missatge, obtenint la informació necessària per poder llegir-lo.</li> <li>3- El missatge es desencripta i es comprova la signatura digital.</li> <li>4- Fent servir la informació de les capçaleres es llegeix el cos del missatge.</li> <li>5- El contingut del missatge es guarda a un fitxer.</li> </ol>
Excepcions:	<ol style="list-style-type: none"> <li>3a- Error en la signatura digital</li> <li>3b- S'envia un missatge de servei negatiu notificant l'error.</li> </ol>

Nom:	Escriure contingut
Descripció:	Construeix un missatge AS2 afegint al cos del missatge el contingut que es vol enviar.
Actors principals:	CCED
Actors secundaris:	-
Precondicions:	Es vol enviar un missatge a AS2
Postcondicions:	El missatge AS2 és construït
Flux principal:	<ol style="list-style-type: none"> <li>1- Es configuren les capçaleres HTTP del missatge.</li> <li>2- Es construeix el cos MIME del missatge.</li> <li>3- S'afegeix el contingut del missatge al cos.</li> <li>4- El missatge AS2 és construït.</li> </ol>
Excepcions:	-

Nom:	Llegir capçaleres
Descripció:	Captura i llegeix les capçaleres HTTP.
Actors principals:	CCED
Actors secundaris:	-
Precondicions:	Arriba un missatge HTTP al servidor
Postcondicions:	Les capçaleres són llegides.

## Disseny, implementació i implantació d'un servidor AS2

Flux principal:	<ol style="list-style-type: none"> <li>1- Informació HTTP arriba al servidor.</li> <li>2- Es capturen les capçaleres a partir de l'objecte HTTPRequest.</li> <li>3- Les capçaleres es llegeixen.</li> </ol>
Excepcions:	-

Nom:	Escriure capçaleres
Descripció:	Construeix i escriu les capçaleres HTTP necessàries per poder enviar el missatge AS2.
Actors principals:	CCED
Actors secundaris:	-
Precondicions:	Cal enviar un missatge AS2.
Postcondicions:	Capçaleres HTTP creades i pujades al servidor web destinatari.
Flux principal:	<ol style="list-style-type: none"> <li>1- S'obtenen dades del missatge AS2 que es vol enviar.</li> <li>2- A partir d'aquestes dades es construeixen les capçaleres HTTP.</li> <li>3- Fent servir una instrucció CURL es pugen aquestes capçaleres al servidor destí.</li> </ol>
Excepcions:	-

Nom:	Enviar missatge
Descripció:	Procés que engloba tot el conjunt d'instruccions necessàries per enviar el missatge.
Actors principals:	CCED
Actors secundaris:	Usuari AS2
Precondicions:	Cal enviar un missatge a un usuari AS2
Postcondicions:	Missatge enviat correctament.
Flux principal:	<ol style="list-style-type: none"> <li>1- A partir de la informació del missatge es construeixen les capçaleres.</li> <li>2- S'envien les capçaleres al servidor HTTP de l'usuari AS2.</li> <li>3- Es construeix el cos del missatge i s'envia al servidor de l'usuari AS2.</li> <li>4- Missatge enviat correctament</li> </ol>
Excepcions:	-

Nom:	Rebre missatge
Descripció:	Procés que engloba tot el conjunt d'instruccions necessàries per rebre el missatge.
Actors principals:	CCED
Actors secundaris:	Usuari AS2
Precondicions:	Arriba un missatge al servidor AS2
Postcondicions:	Missatge rebut correctament i guardat a la cua d'entrada
Flux principal:	<ol style="list-style-type: none"> <li>1- Arriben capçaleres HTTP al servidor, a més d'informació passada per mètode POST.</li> <li>2- Es llegeixen i interpreten les capçaleres.</li> <li>3- Es rep el cos del missatge, es comprova la seva signatura i es descripta.</li> <li>4- S'obté la informació útil dins del cos del missatge.</li> <li>5- El missatge es guarda a la cua d'entrada.</li> </ol>
Excepcions:	-

## Disseny, implementació i implantació d'un servidor AS2

Nom:	Rebre Missatge de AS2
Descripció:	El missatge rebut pel servidor AS2 s'interpreta i es prepara per ser enviat al procés central.
Actors principals:	CCED
Actors secundaris:	-
Precondicions:	Hi ha un missatge d'entrada a la cua corresponent al servidor AS2.
Postcondicions:	El missatge s'envia al procés central.
Flux principal:	<ol style="list-style-type: none"> <li>1- Fent servir les capçaleres HTTP s'obté informació dels partners.</li> <li>2- Es comprova que la informació és correcta.</li> <li>3- Es construeix un XML on s'inclou el cos del missatge i la informació que el procés central necessita per poder interpretar-lo.</li> <li>4- S'envia el XML al procés central.</li> </ol>
Excepcions:	<ol style="list-style-type: none"> <li>2a- La informació dels partners no és correcta.</li> <li>2b- S'envia un missatge de servei negatiu a l'emissor del missatge.</li> </ol>

### 3.4- Diagrama entitat-relació

La figura 9 detalla el diagrama entitat-relació de la base de dades MySQL que fa servir el servidor AS2. La base de dades es fa servir per traçabilitat dels missatges, és a dir saber quins missatges s'han enviat, quan, i quin és el seu estat i paràmetres de configuració. A més es fa servir per la conversió dels missatges entre el format intern de la empresa i el format AS2 i per gestionar tot el servei.

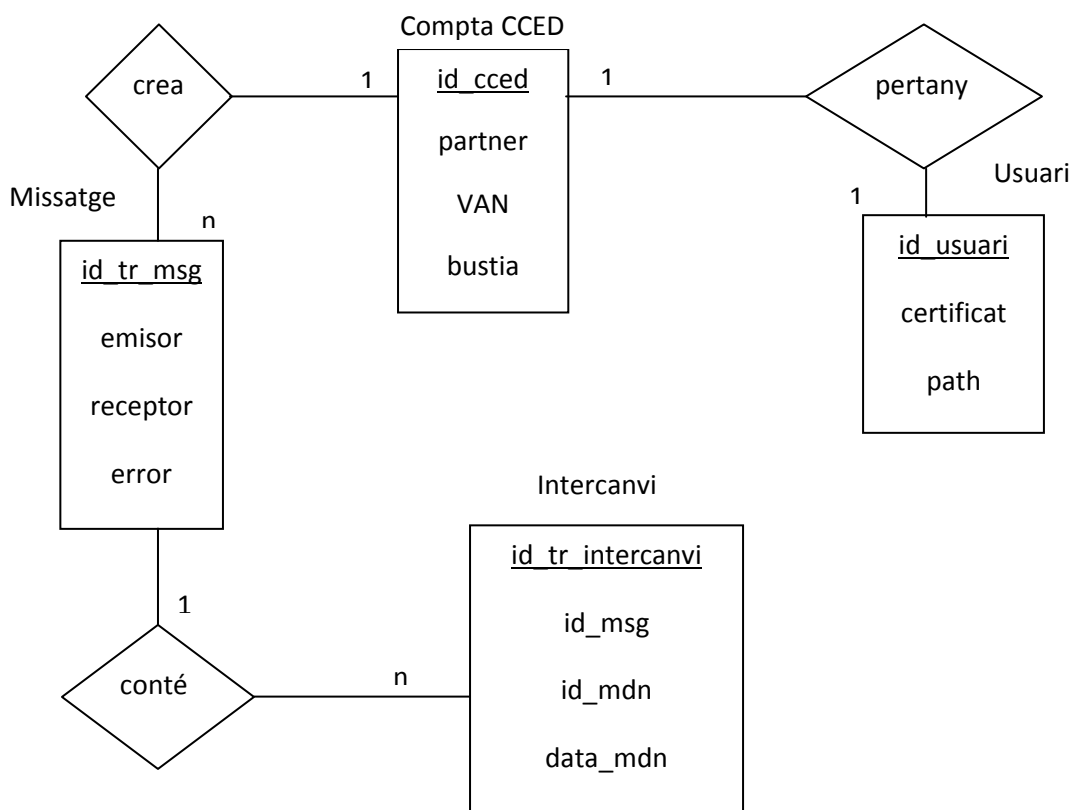


Figura 9 - Diagrama entitat-relació de la base de dades.

## Disseny, implementació i implantació d'un servidor AS2

- La taula de Usuari conté la informació referent als partners, incloent la seva configuració, certificats, etc.
- La taula de Compta CCED conté la relació entre un partner de AS2 i la seva compta d'usuari intern dins de l'empresa, per tal que el procés central pugui fer arribar els seus missatges en ambdós sentits.
- La taula de Missatge representa tota la informació necessària d'un missatge AS2, sense mostrar el seu contingut, aquesta informació es fa servir per poder assegurar entrega de missatges i notificar errors.
- La taula Intercanvi correspon a un missatge EDI contingut dins d'un missatge AS2. Aquesta taula és necessària ja que el RFC que defineix AS2 permet que dins d'un missatge AS2 es puguin incloure múltiples documents i cal tenir constància de tots ells i de les seves respostes per poder respondre el missatge AS2 original.

### 3.5- Diagrama de seqüència

En aquest apartat es mostraran els diagrames de seqüència del projecte.

Primerament es mostrarà el diagrama de seqüència de l'enviament d'un missatge des d'un client de l'empresa fins a l'usuari AS2 final, passant per tots els processos que intervenen en l'enviament. Després es mostrarà el diagrama referent al mateix procés d'enviament però entrant en més detall a la part del servidor AS2.

#### 3.5.1- Diagrama de seqüència del procés general

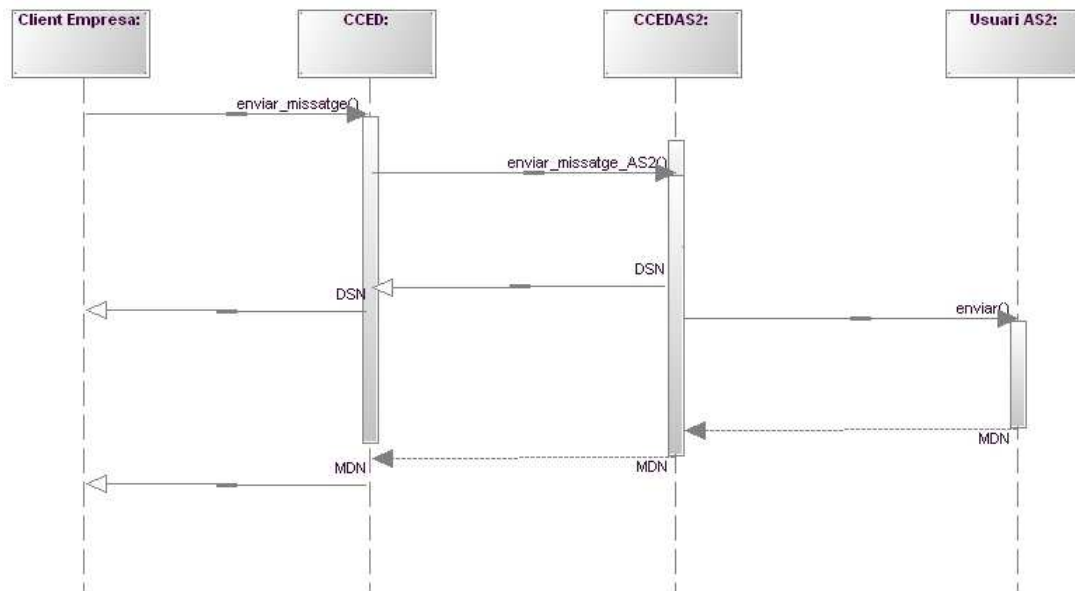


Figura 10 - Diagrama de seqüència del procés general.

La figura 10 mostra procés d'enviar un missatge des d'un client de l'empresa fins al destinatari AS2 destí, rebent el missatge de servei corresponent. El procés de rebre és exactament igual,

## Disseny, implementació i implantació d'un servidor AS2

però el sentit dels missatges va de Usuari AS2 a Client Empresa. Es pot observar com el client executa l'enviament d'un missatge, que pot ser a través de diferents tecnologies però que sempre acabarà arribant al procés central CCED. El procés central CCED veurà que el destinatari del missatge és un usuari AS2 i per tant enviarà aquest missatge, ara en format XML, al procés CCED AS2. En rebre el missatge el procés CCED AS2 genera el missatge de servei DSN, indicant que el missatge ha arribat correctament al servidor AS2 i seguidament envia el missatge en format AS2 a al destinatari.

Ja sigui de manera síncrona o asíncrona l'usuari AS2 contestarà al missatge anterior amb un MDN en format AS2. El CCED AS2 rep aquest missatge i el converteix en XML per enviar-lo al procés central CCED. El procés central converteix aquest MDN a un missatge de servei, del format adient depenent del tipus de tecnologia emprada inicialment, i l'envia a l'emissor del missatge original.

### 3.5.2- Diagrama de seqüència del servidor AS2

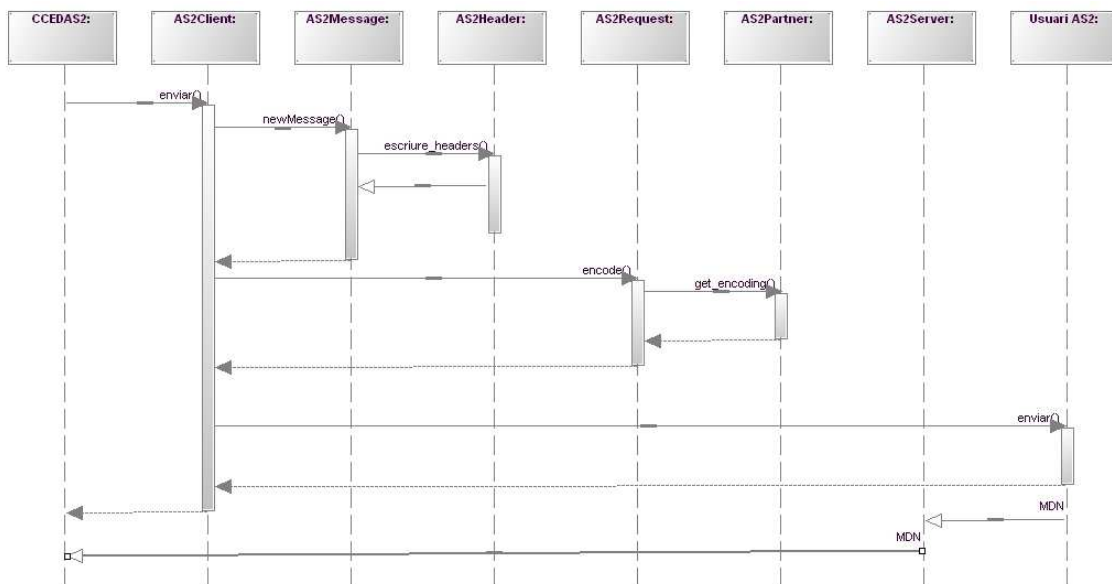


Figura 11 - Diagrama de seqüència del servidor AS2.

La figura 11 mostra el procés d'enviar un missatge un cop aquest arriba al servidor AS2. El procés de rebre un missatge és igual, només canvia el sentit dels missatges i que en comptes d'escriure headers es llegeixen, i en comptes de codificar el missatge es decodifica.

Es poden observar tots els passos que es segueixen per convertir un missatge en format XML a format AS2 i enviar-lo, passant per les diferents classes que trobem a la fase d'enviament i recepció de missatges. Cal tenir en compte que el procés de encoding pot contenir també el procés de firma digital si així està configurat.

### 3.6- Diagrama de classes

En aquest apartat es mostraran els diagrames de classes del servidor AS2.

## Disseny, implementació i implantació d'un servidor AS2

Primerament es mostraran dues versions del diagrama de classes del blocs Client i Servidor, la primera versió especifica cada classe, amb els seus atributs i funcions, però no especifica les relacions entre classes ja que farien el diagrama massa complicat. La segona versió ensenya les relacions entre les classes citades al primer diagrama. Finalment es mostra el diagrama de classes complet del bloc CCED AS2.

### 3.6.1- Diagrama de classes dels blocs Client i Servidor

Per tal de facilitar la seva lectura, la figura 12 mostra el diagrama de les classes, els seus atributs i les seves funcions, però sense mostrar les seves relacions ja que no es podria llegir correctament:

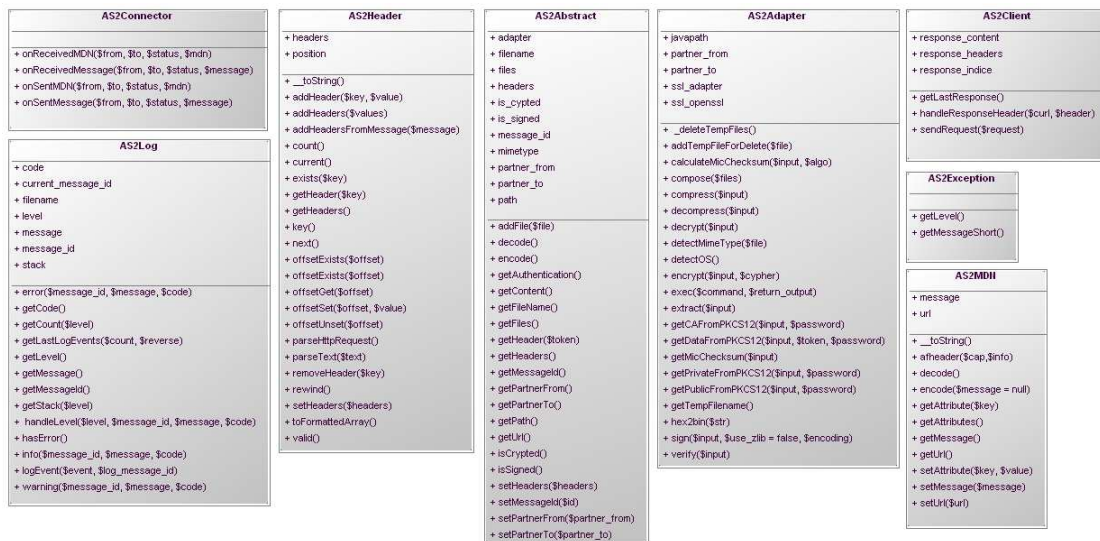


Figura 12 - Diagrama de classes del blocs Client i Servidor amb atributs i funcions.

A continuació es mostren les relacions entre les classes a la figura 13:



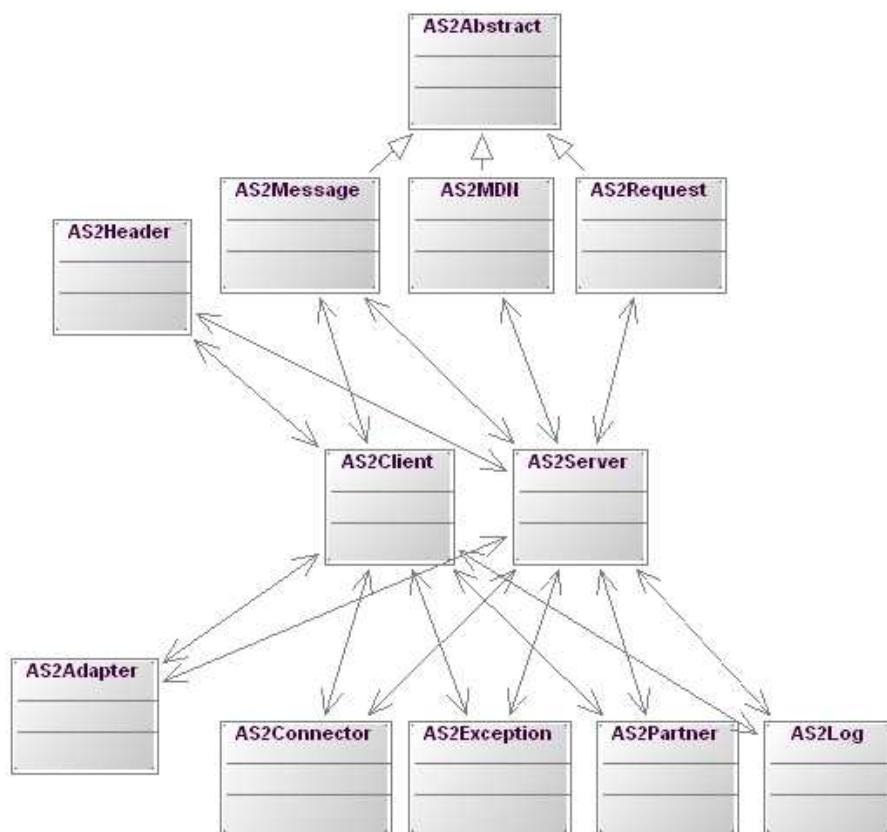


Figura 13 - Diagrama de classes del blocs Client i Servidor amb relacions.

Com es pot observar la classe AS2Abstract és la base de la qual hereten AS2Message, AS2MDH i AS2Request. Aquestes 3 classes juntament amb AS2Header són les principals que intervenen en el procés d'enviament, gestionat per la classe AS2Client, i recepció de missatges, gestionada per la classe AS2Server.

Les altres classes tenen funcions de suport per les classes AS2Client i AS2Server. Les classes AS2Adapter i AS2Connector tenen funcions que ajuden a establir la connexió HTTP amb un servidor destí i a controlar la connexió que un client extern por fer al nostre servidor. La classe AS2Partner té funcions per recuperar informació dels diferents partners que tenim emmagatzemats, entre aquesta informació té especial importància el certificat del partner. Finalment les classes AS2Exception i AS2Log tenen una funció informativa, on una controla els possibles errors del bloc AS2 i l'altra genera un log amb aquests error i altra informació important.

### 3.6.2- Diagrama de classes del bloc CCED AS2

Com es pot observar a la figura 14 la part referent al bloc CCED AS2 és força senzilla pel que fa a les classes. Tenim una classe CCED que gestiona tot el procés de les cues d'entrada i sortida i l'execució de subrutines, una classe Cues on trobem les subrutines corresponents a les cues d'entrada, sortida i RAW (entrada en format AS2). Finalment la classe Funcions conté tot tipus de funcions auxiliars que es fan servir a les subrutines de les cues.

## Disseny, implementació i implantació d'un servidor AS2

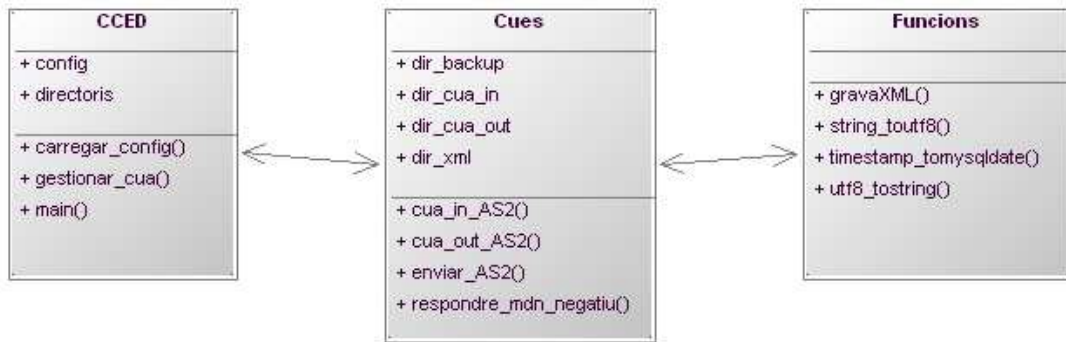


Figura 14 - Diagrama de classes de CCED AS2

## 4- IMPLEMENTACIÓ

En aquest capítol s'explica com s'han implementat tots els mòduls funcionals especificats a la fase de disseny. A més es llisten i expliquen totes les eines emprades per implementar aquestes parts.

La implementació d'aquest projecte està dividida en dues parts, el bloc CCED AS2 descrit a la fase de disseny, implementat en PERL, que s'encarrega de la comunicació amb el procés central de l'empresa, la traducció i tractament dels XML i gestiona i dirigeix l'execució dels blocs Client i Servidor, implementats en PHP, que s'encarreguen d'enviar i rebre missatges a usuaris AS2 externs, el procés de rebuda de missatges és automàtic, però el procés d'enviament ha de ser activat pel bloc CCED AS2.

Per tal de comunicar-se amb el procés central i el bloc PHP, el bloc PERL compta amb 3 cues, dues d'entrada i una de sortida. Fent servir FTP el procés central deixa els missatges que vol enviar a AS2 a la cua d'entrada corresponent del bloc PERL, aquest llegeix i interpreta els missatges i fent servir el bloc PHP els envia a AS2. Quan el bloc PHP rep un missatge que prové de AS2 el deixa a la cua d'entrada corresponent del bloc PERL, aquest el llegeix i el converteix i posteriorment el deixa a la seva cua de sortida. Allà s'executa una subrutina que, fent servir FTP, envia el missatge al procés central.

### 4.1- Eines emprades

A continuació es llistaran les eines que s'han fet servir per implementar el projecte. Es donarà una explicació del context històric que envolta a cada tecnologia i es donarà una explicació sobre el seu funcionament, els seus avantatges i inconvenients. Es recalcarà sobretot el motiu pel qual s'ha escollit aquesta tecnologia per sobre d'altres de característiques similars.

En el cas de PERL, PHP i Apache, s'han escollit aquestes tecnologies per requeriment de l'empresa, ja que són les que han estat emprades anteriorment en projectes similars. A més l'empresa compta amb força experiència en el seu ús i configuració, fet que pot escurçar enormement la fase de implementació.

En el cas de MySQL l'empresa ha deixat que s'esculli el gestor de base de dades amb millors prestacions per ajudar a la implementació del projecte. Tot i que s'ha buscat prioritzar escollir un gestor gratuït per no encarir el projecte.

El cas de Crontab és diferent a la resta de tecnologies. En l'entorn de UNIX existeixen molt pocs programes de tipus Cron, i Crontab ha demostrat al llarg dels anys ser un programa fiable, fàcil de configurar i amb una gran capacitat d'executar i controlar tot tipus de scripts. Aquests motius han fet que Crontab hagi estat escollit de manera gairebé automàtica per controlar l'execució periòdica dels scripts.

#### 4.1.1- PERL

Perl<sup>[5]</sup> és un llenguatge de programació dissenyat per Larry Wall. El llenguatge de programació Perl va ser alliberat el 18 de desembre de 1987 i està inspirat en els llenguatges awk, C, sed,

## Disseny, implementació i implantació d'un servidor AS2

shell entre d'altres. Fins 1991, l'única documentació de Perl era una simple pàgina de manual d'Unix. El 1991 es va publicar *Programming Perl* i immediatament es va convertir en referència de fet del llenguatge. Al mateix temps, el número de la versió de Perl va avançar a 4, no per significar un gran canvi al llenguatge, sinó per identificar la versió que estava documentada al llibre.

Perl és el llenguatge que fem servir a l'empresa per programar scripts i rutines, així com els programes centrals. En el meu cas l'he fet servir per implementar el programa central que s'encarrega de fer de pont entre el servidor AS2 i la resta de l'empresa.

El principal motiu per fer servir Perl és que és un dels millors llenguatges de programació pel tractament de text, gràcies a les seves expressions regulars, per exemple:

```
$line =~ /Original-Message-ID:\s(.+)\r\n/;
```

On \$line conté tot el contingut d'un missatge AS2. Amb aquesta instrucció estem trobant el patró "Original-Message-ID: " i guardant tot el que hi ha entre aquest patró i el següent sal de línia a una variable \$1.

També es poden fer servir expressions regulars per detectar si un patró existeix dins d'un fitxer, per exemple:

```
if ($line =~ /Content-Type: message\vdisposition-notification/){ ## ES MDN ##}
```

O per fer substitucions:

```
$orig_msg_id=~s/(\\;|\\')/\\$1/g;
```

En aquesta instrucció es busquen dins de la variable \$orig\_msg\_id tots els ";" i els "\", i es substitueixen per "\\;" i "\\\" respectivament, o el que és el mateix, s'escapen.

A part de les funcionalitats vistes en aquests exemples les expressions regulars poden fer-se servir per molts més casos, i són una eina increïblement potent, de fet altres llenguatges com per exemple PHP ha adoptat aquestes funcionalitats i aquesta sintaxi en les seves últimes versions.

A més de les expressions regulars, Perl compta amb un mòdul anomenat CPAN que permet afegir a Perl altres mòduls que implementen funcionalitats de tota mena, per exemple existeixen mòduls per llegir i escriure XML, per connectar amb tot tipus de bases de dades, per tractament de pàgines web, etc. Cada un d'aquests mòduls es descarrega i es manté actualitzar d'un o més repositoris fent servir la instrucció de shell "cpan".

El fet de comptar amb aquesta possibilitat d'obtenir llibreries ja provades i funcionals facilita molt la implementació de nous scripts, sobretot quan aquests han de dur a terme operacions en camps amb els quals no estem massa familiaritzats.

Per últim, he escollit Perl per fer el programa central ja que dóna moltes funcionalitats per implementar programes stand-alone (es poden executar a qualsevol lloc del sistema sense dependre d'allò que els envolta) i poder controlar el seu funcionament per mitjà de crons (programa que executa un determinat script en el moment de temps desitjat).

#### 4.1.2- PHP

PHP<sup>[1][2]</sup> és un llenguatge de programació interpretat utilitzat per a generar pàgines web dinàmiques. S'executa a la banda del servidor, per tant al navegador web la pàgina ja li arriba en format HTML i no podem veure'n el codi php. Tot i que menys popular, també pot ser usat per generar interfícies de comandes i aplicacions executables amb interfície gràfica. Té versions tant per Windows com GNU/Linux i altres.

PHP es distribueix sota la llicència PHP, que la Free Software Foundation qualifica com a programari lliure. Les sigles són un acrònim recursiu de PHP: Hipertext Preprocessor, tot i que originalment eren les sigles de Personal Home Page Tools. PHP és extremadament modularitzat. En tipus de dades, sintaxi i funcions s'assembla molt al llenguatge de programació C. I des de la versió 5, que inclou un major suport a l'orientació a objectes, també s'assembla al C++. PHP pot ser inclòs dins el codi HTML indicant-ho a l'interpret mitjançant les etiquetes `<? i ?>` o `<?php i ?>`. Aleshores, quan el llenguatge sigui interpretat s'executarà el codi entremig de les etiquetes i generarà com a sortida una codi en HTML, que serà barrejat amb la pàgina HTML on hagués estat inclòs.

El core del programa que interpreta i envia els missatges AS2 està implementat en PHP. El principal avantatge d'aquest llenguatge de programació és que està altament especialitzat per la creació i gestió de pàgines web. Un altre gran avantatge afegit és que un usuari final mai podrà veure el codi PHP des de la seva estació client, ja que aquest codi s'interpreta i executa a la part del servidor i només el resultat (en mode text pla, xml, html, o altre format) és vist pel client.

Una altra gran avantatge de PHP és la facilitat per execució des de altres programes i el control de la interacció entre aquests. Per exemple en el cas d'aquest projecte el procés central fet en PERL crida a un PHP que alhora executa funcions del core de AS2, tot aquest procés és controlat des de el procés central i si alguna part va malament i es dona un error el procés fet en PERL rep l'error, l'interpreta i actua d'una manera o una altra en funció del problema.

```
$responde = POST $url, Content => [ from => $partner_from,to => $partner_to,edi => $edi, id => $id];  
$sortida=$browser->request($responde)->as_string;  
if($sortida=~"/Fatal error/i){  
# Determinar tipus d'error
```

Figura 15 - Crida a un servidor HTTP en PERL.

En aquesta part de codi que mostra la figura 15, feta en PERL, es pot veure com es genera una estructura de dades concreta per poder fer una crida a un servidor HTTP on es troba el codi PHP que es vol executar. Aquesta crida es fa gràcies a un mòdul de PERL pensat per aquesta funció i que facilita enormement la feina. Després de l'execució d'aquest PHP es captura la resposta i fent servir expressions regulars es pot comprovar fàcilment si hi ha hagut algun problema, i si aquest fos el cas comprovar què ha fallat.

#### 4.1.3- Apache

Apache HTTP Server<sup>[9]</sup> és un servidor HTTP de codi obert multiplataforma desenvolupat per Apache Software Foundation. Quan va començar el seu desenvolupament l'any 1995, es basava inicialment en el codi del popular NCSA HTTPd 1.3, però més tard es reescriuria completament.

Des d'abril de 1996 Apache ha estat el servidor HTTP més popular a la World Wide Web; des de març de 2006, tot i això, ha experimentat una declinació de la seva quota de mercat, perduda en major part contra Microsoft Internet Information Services i .NET, fet servir per alguns dels grans proveïdors de blogs. En octubre de 2007 Apache va servir el 27,73% de tots els llocs web encara que recentment ha tingut un petita remuntada. El 30 de novembre de 2007, Apache servia el 50,76% de tots els llocs web. Això és un increment del 3% des d'octubre, mentre que Microsoft va perdre un 2%.

Per tal de poder executar PHP només cal tenir l' intèrpret instal·lat a la màquina i vincular d'alguna manera el script amb el path a aquest binari. Però si PHP es fa servir per creació i gestió de pàgines webs cal tenir un servidor HTTP per poder generar les pàgines web.

A més en el cas d'aquest projecte el servidor Apache té especial importància:

La base de la tecnologia AS2 és que fa servir HTTP com a mitjà de comunicació per poder enviar i rebre missatges. Per enviar un missatge cal executar el PHP adient i enviar al servidor HTTP del client la informació, però quan som nosaltres qui rebem un missatge és el nostre servidor HTTP qui veritablement el rep i crida al nostre script PHP per iniciar tot el flux d'instruccions que constitueixen el core del bloc AS2.

A més les bases per poder construir, interpretar, i llegir el contingut d'un missatge AS2 les dona el servidor HTTP, per mitjà dels headers que indiquen totes les variables i camps necessàries per poder llegir el bloc MIME on es troba el contingut del missatge.

#### 4.1.4- MySQL

MySQL és un sistema de gestió de bases de dades relacional (anglès RDBMS - Relational DataBase Management System) multi-fil (multithread), multiusuari, que usa el llenguatge SQL (Structured Query Language).

Al contrari d'altres projectes com el servidor de HTTP Apache, MySQL pertany a, i està esponsoritzat per, l'empresa Sueca MySQL AB. La companyia desenvolupa i manté el programari en la versió lliure i la comercial. Ofereix suport als clients, assistència tècnica i documentació.

MySQL ha esdevingut molt popular gràcies a la seva velocitat en executar consultes i el seu suport de forma nativa per part del llenguatge PHP (fins a la versió 4.X d'aquest llenguatge ja que a partir de la versió 5 deixa d'estar-ho), en l'elaboració d'aplicacions web, en l'entorn del programari lliure.

Es pot fer ús de MySQL en aplicacions de tota mena (web, d'escriptori o d'altres) de forma lliure i gratuïta sota les condicions de la llicència GPL. Si es vol integrar MySQL com a part d'un producte privatiu cal adquirir una llicència d'ús específica per a aquest propòsit.

Disseny, implementació i implantació d'un servidor AS2

MySQL és un dels components de l'arquitectura LAMP (de Linux - sistema operatiu) i la WAMP (de Windows - sistema operatiu). Les tres sigles que segueixen (A,M i P) fan referència al conjunt format per Apache (servidor web), MySQL (base de dades) i PHP (llenguatge de programació). Una plataforma per a la construcció de llocs web utilitzant programari lliure.

El 16 de gener del 2008, Mysql AB fou adquirida per Sun Microsystems.

· Motius pels quals s'ha escollit MySQL:

Avui en dia es poden trobar un gran nombre de gestors de bases de dades, molts d'ells gratuïts, tot i això cadascun té unes característiques en concret i sol ser més adient per un cas d'ús en concret.

Excel:

- Possibilitat de presentar les dades de forma estètica.
- Els fulls poden emmagatzemar dades numèriques.
- Ús de les ordres i les eines de Microsoft Excel per a executar càlculs amb les seves dades.
- Permet sumar files i columnes.
- Utilització de les cel·les per a realitzar avaluacions d'una mateixa funció amb diferents valors.
- Escriptura en els formats \*.xls, \*.csv i \*.xlsx.

Tot i aquestes funcionalitats no compta amb una eina per dur a terme cerques, ni organitzar taules amb PK i FK, ni relacions de cap tipus.

Access:

Microsoft Access és un programa gestor de bases de dades relacional creat per Microsoft per a ús personal i de petites empreses. És un component del paquet Microsoft Office. Per treballar en una base de dades cal, en primer lloc, carregar el programa i, després definir la base de dades o utilitzar-ne una de construïda anteriorment.

Tot i donar opcions per crear i gestionar base de dades no dona facilitat per fer servir la base de dades en un entorn web i enviar query des de una aplicació externa.

MySQL:

- Molt ràpid en entorn web.
- Si la base de dades és petita és molt eficient.
- Problemes d'eficiència en bases de dades grans.
- Fa servir el llenguatge SQL per les query.
- Permet accés des de aplicacions externes fent servir query SQL.
- Compatibilitat amb PHP.

PostgreSQL:

- Funciona igual de ràpid amb bases de dades grans o petites.

## Disseny, implementació i implantació d'un servidor AS2

- Consumeix molts recursos.
- Gestor objecte – relació.
- Pot definir tipus propis (vectors).
- Pot fer/definir stored procedures.
- Pot generar regles : combinació de triggers amb stored procedures.

### Oracle:

- Primer gestor en tenir base de dades relacionals.
- Permet mirroring: funcionalitat per generar còpies de seguretat.

### Informix:

- Permet fer compactació d'índexs: tècnica usada per optimitzar consultes. Es crea un arbre segons la semblança dels índexs.

### SQL server:

- Suport de transaccions.
- Escalabilitat, estabilitat i seguretat integrades en Directori Actiu.
- Suport per procediments emmagatzemats.
- Inclou també un potent entorn gràfic d'administració, que permet l'ús comandaments DDL i DML de manera gràfica.
- Permet treballar en mode client-servidor on la informació i dades viuen en el mateix servidor i els terminals o clients de la xarxa només accedeixen a la informació.
- A més a més, permet administrar informació d'altres servidors de dades.

	Query	Accessible	Ràpid	Segur	Gratis	Comentaris
Excel			X			No és relacional
Access	X		X			
MySQL	X	X	X	X	X	Molt eficient en BD petites
PostgreSQL	X	X	X	X	X	Pensat per BD grans
Oracle	X	X	X	X	X	Permet mirroring
Informix	X	X	X	X		Compactació índexs
SQL server	X	X	X	X		Moltes funcionalitats

Figura 16 - Taula comparativa de gestors de bases de dades.

A la figura 16 es pot observar com MySQL és el gestor que funciona de manera més ràpida i eficient en un entorn web amb una base de dades de grandària relativament petita com és el cas de l'aplicació actual. A més PHP dóna total suport pel llenguatge SQL i la connexió amb el gestor MySQL.



## Disseny, implementació i implantació d'un servidor AS2

Tot i que altres gestors podrien haver funcionat igualment, MySQL és el gestor més senzill que satisfà les condicions mínimes per poder-lo fer servir pel projecte, per tant haver-ne fet servir un de més complicat només encariria el projecte i provocaria que es perdés massa temps en rutines de connexió, actualització, inserció o eliminació de dades.

### 4.1.5- Crontrab

Els programes tipus cron serveixen per programar l'execució de tasques en un determinat moment. Aquest tipus de programa es troba en entorns de tipus Unix i es solen fer servir sobretot per l'execució d'scripts de backup, de control del sistema i de control d'aplicacions.

En el meu cas he fet servir el cron "Crontab"<sup>[11]</sup>, que ve per defecte amb la majoria de distribucions Unix, per programar l'execució d'un script que controla una cua del projecte. La sintaxi de programació es força senzilla, com es mostra a la figura 17:

```
01 * * * * root nice -n 19 run-parts /etc/cron.hourly
50 0 * * * root nice -n 19 run-parts /etc/cron.daily
22 4 * * 0 root nice -n 19 run-parts /etc/cron.weekly
42 4 1 * * root nice -n 19 run-parts /etc/cron.monthly
```

Figura 17 - Exemple de Crontab.

Primerament es posen els paràmetres que determinen el moment de temps, un asterisc significa tots i d'esquerra a dreta els paràmetres són minut, hora, dia, mes, any, dia de la setmana (Dilluns, Dimarts...). Després dels paràmetres de temps es posa la comanda de Shell que es vol executar. Els paràmetres de temps accepten també operacions de tipus \*/5 per exemple, que voldria dir sempre que la unitat es divisible per 5, per exemple si es posa com a primer paràmetre significaria cada 5 minuts.

En el cas que un missatge AS2 no es pugui entregar al destinatari per algun motiu es rep un error HTTP que especifica el tipus d'error. un error de tipus 4XX sol ser donat perquè el nostre client AS2 no pot trobar la URL del servidor AS2 destí, en aquest cas l'error es nostre i segurament cal arreglar algun problema amb la configuració del partner o amb la sortida a l'exterior del client AS2, sigui com sigui, és un error que només es dona una vegada, o poc sovint, i que un cop solucionat és poc probable que es repeteixi.

En el cas d'un error tipus 5XX el problema sol estar en el servidor AS2 destinatari, pot ser que el servidor estigui en manteniment, o tingui un problema tècnic d'algunes tipus. En aquests casos seguir intentant entregar el missatge cada vegada que el programa miri la cua de sortida no es gens òptim, i és indesitjable ja que ompliria el log de l'aplicació innecessàriament. De la mateixa manera no seria correcte donar el missatge com erroni i notificar-ho a l'emissor, ja que ni el nostre client ni l'emissor del missatge ha comés cap error com per rebre una notificació d'entrega negativa.

Per solucionar això el que he fet és capturar el missatge d'error HTTP i buscar el codi d'error que conté, en el cas que el codi d'error sigui un 5XX el missatge es mou a una cua auxiliar que està gestionada per un script que s'executa cada 15 minuts.

## Disseny, implementació i implantació d'un servidor AS2

Aquest script auxiliar és força senzill, el seu funcionament és el següent:

- 1- En ser executat mira el directori de la cua auxiliar per si hi ha algun missatge.
- 2- En el cas d'haver algun arxiu, el programa consulta un arxiu auxiliar per veure si ja l'ha tractat abans, de no ser així, crea una entrada a l'arxiu auxiliar amb un contador a 1.
- 3- Si el programa troba a l'arxiu auxiliar el nom del missatge incrementa el comptador.
- 4- En qualsevol dels casos anteriors, el missatge és reenviat a la cua de sortida de AS2 per tornar a intentar a enviar-lo, si es torna a produir un error 5XX serà enviat de nou a la cua auxiliar.
- 5- En el moment que el comptador de reenviaments del missatge, que està al fitxer auxiliar, sigui 3 el missatge serà considerat com un error, serà notificat com a tal a l'emissor del missatge i es traurà del circuit de cues.

És a dir, un missatge serà considerat com a erroni si durant uns 45 minuts aproximadament ha estat impossible fer-lo arribar al destinatari, sense que l'error sigui nostre.

### 4.2- Implementació dels mòduls

A continuació es detalla la implementació dels mòduls PERL i PHP.

Es començarà detallant el mòdul PERL, representat per tots els scripts que formen el procés CCED AS2, explicant quins scripts el formen, quina és la seva funció, com estan implementats, i com interactuen entre ells.

#### 4.2.1- Mòdul PERL: procés CCED AS2

Aquest mòdul és l'encarregat de comunicar la plataforma AS2, feta en PHP, amb el procés central de la infraestructura de l'empresa. Per dur a terme aquesta comunicació el mòdul PERL compta amb 3 cues, és a dir, directoris on es dipositen i recullen fitxers.

La cua AS2\_IN és el directori on el procés central deixa fitxers en format XML. Cada un d'aquests fitxers és un missatge que es vol fer arribar a un usuari AS2 extern i conté totes les dades necessàries per que el mòdul PERL els pugui interpretar correctament.

La cua AS2\_OUT és el directori on el mòdul PERL deixa els missatges en format XML que es volen fer arribar al procés central. Cada un d'aquests fitxers XML representa un missatge que ha arribat de un usuari AS2 extern i que s'ha de fer arribar a un client intern de l'empresa.

La cua RAW\_AS2 és el directori on el bloc PHP deixa els missatges en format AS2 que arriben al servidor AS2. Per cada missatge es crea un mínim de 2 fitxers, un que conté les capçaleres HTTP del missatge i un altre que conté el cos. Si el missatge tenia algun tipus de contingut també s'inclou a un fitxer a part. Totes aquestes dades són necessàries per tal que el mòdul PERL els pugui interpretar i tractar.

Disseny, implementació i implantació d'un servidor AS2

El mòdul consta de 4 scripts fets en PERL: `cced_as2.pl`, `colas_as2.pl`, `fuc_as2.pl` i `copy_as2.pl`. A continuació es descriuran per separat cadascun d'aquests scripts.

`cced_as2.pl`:

És l'script principal que controla i es comunica amb la resta d'scripts. La seva estructura bàsica és un bucle que va comprovant cadascuna de les cues d'entrada o sortida de missatges i executa el procés corresponent depenent de la cua que calgui tractar.

Per controlar la seva execució i la seva aturada es fa servir un arxiu programat en bash que l'executa, redirigint tota la sortida que es pugui generar a un fitxer de log. Quan es vol aturar el procés, es crida el fitxer bash i aquest genera un fitxer auxiliar al directori on es troba `cced_as2.pl`, quan l'script detecta aquest fitxer duu a terme una parada controlada, assegurant que no es perdi informació.

Per executar aquest fitxer bash es fa servir la següent sintaxi:

```
/etc/init.d/cced_as2 status
```

Mostra si el procés `cced_as2.pl` està funcionant o parat.

```
/etc/init.d/cced_as2 start
```

Executa el procés `cced_as2.pl`

```
/etc/init.d/cced_as2 stop
```

Genera el fitxer auxiliar al directori del procés per tal que aquest s'aturi de manera controlada.

Quan el procés s'executa llegeix un fitxer de configuració que determina el seu comportament, el fitxer és el següent:

## Disseny, implementació i implantació d'un servidor AS2

```
[colas_activas]
cola_as2_in=1
cola_as2_out=1
cola_raw_as2=1
[DEBUG]
screen=0
timer=0
[colas]
dir_cola_as2_in=/data/COLAS/AS2/IN_AS2
dir_cola_as2_out=/data/COLAS/AS2/OUT_AS2
dir_cola_raw_as2=/data/CCED_AS2/RAW_AS2
[path]
error=/data/backup/COLAS/as2/errores
backup=/data/backup/COLAS/as2/backups
filestopcced_as2=/data/CCED_AS2/STOPCCED_AS2
reenvio=/data/scripts-control/colas/fixters_moguts/AS2
[mail]
smtpserver=xx.xx.xx.xx
alertascced_as2="xxx\@ediversa.com,xxx\@ediversa.com"
[cced]
ftpservers=xx.xx.xx.xx
user=ftpccedas2
passwd=ftpccedas214
[web]
dominio=as2.ediversa.net
[altres]
nom_maquina=pegasus
version=v5.00
temps_chekprocesoscolas=15
max_missatges_cua=20
[mysql]
user=root
database=ccedas2
host=localhost
password=xxxxxx
```

Figura 18 – Fitxer de configuració del CCED AS2.

El fitxer determina diferents variables que el `cced_as2.pl` llegeix i serveixen per configurar tot tipus de paràmetres, les variables es reparteixen en diferents blocs:

A *colas\_activas* es determina quines cues seran o no controlades pel procés, és a dir, quines cues executaran la seva subrutina.

A *debug* es determinen els paràmetres per poder provar i arreglar errors en el propi procés `cced_as2.pl`.

A *colas* s'indiquen els paths absoluts de cadascuna de les cues actives.

A *path* s'indiquen la resta de paths absoluts que el programa necessita per executar-se. Entre aquests paths estan el d'error, on es mouen els fitxers incorrectes, el de backup, on es guarda copia de tot fitxer tractat, el de reenviament, on es mouen els fitxers per ser reenviats posteriorment i finalment el path del fitxer auxiliar que indica l'aturada del procés.

A *mail* es troba l'adreça IP del servidor smtp i les adreces email a on enviar alertes.

A *cced* hi ha les variables necessàries per la connexió FTP amb el procés central per tal d'enviar-li els missatges que entren per AS2.

A *web* hi ha el domini web del servidor AS2.

## Disseny, implementació i implantació d'un servidor AS2

A *altres* hi trobem informació sobre la versió de l'script i de la màquina on es troba. A més a més s'hi determina el temps d'espera entre cues i el nombre màxim de missatges d'una mateixa cua que seran tractats abans de passar a la següent cua. Això es fa per evitar que alguna de les cues entri en *cpu starvation* si les altres cues tenen molt de tràfic.

Finalment a *mysql* hi trobem totes les variables necessàries per poder connectar a la base de dades.

Un cop s'han actualitzat totes les variables especificades en aquest fitxer de configuració, el procés *ccd\_as2.pl* fa una sèrie de tests per assegurar-se que tots els serveis estan llestos per poder tractar les cues. Entre altres mira que els servidors web i FTP responguin correctament.

Un cop fet això s'entra en el bucle principal, on a cada iteració es mira una de les cues actives. Quan es comprova que en alguna cua hi apareixen missatges que cal tractar s'executa la subrutina *adient*, localitzada a l'script *colas\_as2.pl*.

### colas\_as2.pl

En aquest script trobem les subrutines que tracten cadascuna de les cues, així com subrutines auxiliars relacionades amb el tractament d'aquestes.

Pel correcte funcionament de la plataforma són necessàries 3 cues: *AS2\_IN*, *RAW\_AS2* i *AS2\_OUT*. I per tant tenim una subrutina que tracta cadascuna d'aquestes cues.

La cua *AS2\_IN* gestiona els missatges que usuaris del nostre sistema envien cap a usuaris *AS2*. Aquests missatges estan en format XML amb la forma següent:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ms c="EDI">
  <cfg>
    <id_cced>12073</id_cced>
    <id_as2></id_as2>
    <po_emisor>8420418000003</po_emisor>
    <po_receptor>8714253049021</po_receptor>
    <date>110211115725</date>
    <mdn></mdn>
  </cfg>
  <edi><![CDATA[UNA:+.?
'UNB+UNOC:3+8420418000003:14+8714253049021:14+110211:1157+1157'UNH+1157+DESADV:D:01B:UN:EA0
07'BGM+351+110150+9'DTM+137:20110209:102'DTM+2:20110217:102'RFF+ZZZ:EANNL3'RFF+ON:P1100615'NAD
+BY+8714253049021::9'NAD+SU+8420418000003::9'NAD+DP+8712423001374::9'TOD+4'CPS+1'CPS+2+1'PAC+80'
PCI+33E'GIN+BJ+384204180000000059'LIN+1++84204180002038:SRV'PIA+1+200070-006:SN'IMD+F++:::MARQUES
DE MOJA BRUT'QTY+12:480'ALI+++66'DTM+94:20110204'UNT+22+1157'UNZ+1+1157']]></edi>
</ms>
```

Figura 19 – Exemple d'un missatge d'entrada en format XML.

En el missatge XML s'especifiquen camps necessàries per saber de qui ve i a qui va adreçat el missatge, així com el contingut del mateix, en format EDI. El tag *<edi>* esta dins d'un tag *<xml CDATA* per tal que els caràcters especials del format edi no donin problemes a l'hora de llegir el XML.

Els missatges entrant poden ser de 2 tipus, com es mostra a l'atribut *c* del tag *<ms>*. Els missatges de tipus EDI són missatges que s'han d'enviar a un destinatari i dels quals cap

## Disseny, implementació i implantació d'un servidor AS2

esperar un missatge de servei, mentre que els missatges de tipus MDN són missatges de servei que s'envien al destinatari AS2 com a resposta d'un missatge AS2 previ enviat pel mateix.

Un cop llegit el missatge es comprova que l'emissor i el receptor constin a la base de dades, en cas negatiu es mou el missatge a errors i es contesta amb un missatge de servei negatiu. En cas positiu s'executa la subrutina d'enviament de missatge a AS2 que fa servir al client genèric PHP.

Si en intentar fer arribar el missatge al destinatari AS2 es troba algun error es comprova que l'error sigui del servidor web destí i no del nostre. Si és així normalment significarà que per algun motiu no permanent el servidor destí no està disponible en aquests moments i el missatge passa a ser gestionat per l'script de reenviament de missatges, que serà explicat posteriorment.

La cua RAW\_AS2 gestiona els missatges que arriben des de AS2 i s'han de fer arribar als nostres clients. Cada missatge que arriba genera 2 o més fitxers a la cua:

- Fitxer .as2: aquest fitxer és el missatge sencer, sense tractar, que arriba al nostre servidor web. En el cas que el missatge entrant sigui un missatge de servei aquest fitxer serà llegible ja que els missatges de servei no es solen encriptar. Si no es tracta d'un missatge de servei no estarà en format llegible i el procés no el tractarà.
- Fitxer .headers: aquest fitxer conté totes les capçaleres HTTP referents al missatge que han estat rebudes pel servidor web. Un exemple seria el que es mostra a la figura 20:

```
User-Agent: Mozilla/4.0 [en] (WinNT; I)
Accept: image/gif, */*
Host: as2test.ediversa.net
Content-Type: multipart/signed; micalg=sha1; protocol="application/pkcs7-signature"; boundary="----
_Part_57_325453957.1302687666756"
AS2-From: "8714253049021"
AS2-To: "8435178700025"
AS2-Version: 1.1
Message-ID: <1387749322.411302687666758.JavaMail.is76acc@romulus>
Content-Length: 2051
Via: 1.0 osiris:3128 (squid/2.5.STABLE5)
X-Forwarded-For: 141.93.168.43
Cache-Control: max-age=259200
Connection: keep-alive
```

Figura 20 – Exemple de les capçaleres HTTP d'un missatge AS2.

Són importants els paràmetres AS2-From i AS2-To, que determinen l'emissor i el receptor, el Message-ID, que és el identificador del missatge i que en enviar un missatge de servei del mateix haurà d'aparèixer com Original-Message-ID. La variable Content-Type i boundary són les variables que calen per poder llegir el cos del missatge en format MIME.

· Fitxer .payload\_X: aquest fitxer només apareix si el missatge no és de servei i conté el cos del missatge llegit per a ser llegit, és a dir, descriptat i amb totes les comprovacions de seguretat fetes. Com que l'estàndard AS2 permet enviar més d'un fitxer dins d'un mateix missatge AS2 pot ser que apareguin nombrosos payloads, que es diferenciaran amb un número seqüencial al final del nom del fitxer.

## Disseny, implementació i implantació d'un servidor AS2

Primerament la subrutina de la cua mira el fitxer .headers per comprovar que l'emissor i el receptor del missatge són correctes. En cas negatiu es mou el missatge a errors i es contesta amb un missatge de servei negatiu fent servir el client genèric PHP, ja que el missatge de servei s'envia a AS2.

Per determinar si el missatge és de servei o no es mira el fitxer .as2, si aquest es llegible i conté la línia "*Content-Type: message/disposition-notification*" es tracta d'un missatge de servei, en cas contrari es tracta d'un missatge normal.

El missatge és convertit en format XML fent servir les dades de la base de dades. Si el missatge és de servei es relaciona aquest missatge amb el missatge original que es va enviar a AS2, i en el XML s'inclou la referència d'aquest. En el cas que sigui un missatge normal s'agafa el contingut del payload i es posa en el tag <edi>. Si hi hagués més d'un payload es genera un XML per cadascun d'aquests i s'esperaria un missatge de servei per cada un. Un cop format el XML aquest s'envia a la cua AS2\_OUT per poder ser enviat al procés central.

Ja implementades les dues cues d'entrada per on entren els missatges al mòdul PERL, ara cal implementar una cua de sortida per poden enviar els missatges de sortida cap al servidor central. Aquesta és la subrutina que s'encarrega de la cua AS2\_OUT.

La cua AS2\_OUT duu a terme una sincronització FTP, on el cced\_as2 es connecta per FTP al procés central i li envia de cop tot el contingut del directori AS2\_OUT. Aquest contingut són els fitxers XML generats per al subrutina de la cua RAW\_AS2 i que ja estan llestos per ser tractats pel procés central.

A part de les subrutines de les cues, hi ha 2 subrutines auxiliar més que fan servir les cues:

La subrutina enviar\_as2() s'encarrega de executar el client genèric PHP per tal d'enviar tant missatges normals com de servei a un destinatari AS2, a més a més és qui captura els possibles errors HTTP i qui envia els missatges al procés de reenviament en el cas que sigui necessari.

La subrutina respondre\_mdn\_negatiu() retorna un missatge de servei negatiu en format XML al procés central si hi ha hagut algun problema amb el missatge enviat per aquest.

Ara ja tenim implementats els processos que gestionen les 3 cues, però cal encara especificar les funcions auxiliars de les quals depenen aquests processos.

### fuc\_as2.pl

Aquest script està format per subrutines auxiliars que fan servir la resta d'scripts. La majoria d'elles són subrutines de tractament de dates, ja que el tipus de data que Unix fa servir (nombre de segons des de l'any 1700) no és un format massa pràctic i cal convertir-lo a altres formats, com per exemple el tipus DATETIME de MySQL, que té la forma "AAA-MM-DD HH:MM:SS".

Altres funcionalitats s'encarreguen de la conversió de la codificació de caràcters UTF-8 a la codificació ISO-8859-1. Això és necessari ja que el català i el castellà fan servir més caràcters

Disseny, implementació i implantació d'un servidor AS2

que l'anglès per exemple, i si posem caràcters amb accents a un document UTF-8 sense fer la conversió no es veuran o es veurà un caràcter estrany al seu lloc.

Els XML que es fan servir per passar la informació entre el procés central i el CCED de AS2 estan en format ISO-8859-1 per tal de conservar aquests caràcters especials sense haver de fer conversions. En canvi els documents que surten del bloc AS2 en PHP estan codificats en UTF-8, ja que per defecte es la codificació estàndard i així assegurem que qualsevol missatge entrat es veurà correctament.

copy\_as2.pl

Aquest script conté les subrutines de còpia de missatges, cal destacar sobretot una d'elles: la subrutina copytodir(from, to).

Aquesta subrutina és la que faig servir per moure fitxers d'un directori a un altre i abans de moure duu a terme comprovacions de seguretat.

Això és molt important ja que en procés de desenvolupament o proves es poden cometre errors de programació, i si en fer una instrucció "mv" de shell es comet algun error pots malmetre greument tot el sistema de fitxers del sistema, i fins i tot deixar la màquina no operativa.

Un exemple d'això seria una instrucció del tipus "mv \$dir/\$fitxer\* \$directoridesti":

A simple vista sembla que tot sigui correcte, però si en fase de proves per exemple es comet un error i les variables \$dir i \$fitxer no són plenes voldria dir que en realitat estem fent un "mv /\* \$directoridesti", és a dir estem movent tots els directoris de la màquina a \$directoridesti, malmetent totalment tot el sistema de fitxers i els directoris del sistema operatiu, i deixant la màquina totalment malmesa.

Per evitar això la subrutina fa un filtratge del directori destí i el directori origen a l'hora de fer la còpia. Només mourà els fitxers si els directoris origen i destí estan dins d'una llista de directoris permesos.

Amb això ja s'ha especificat tota la implementació del mòdul PERL. És a dir, tenim un programa que sap interpretar els missatges que venen del procés central, sap com enviar missatges correctes a aquest mateix procés central, interpreta correctament els missatges que venen del mòdul PHP i sap com executar el client PHP per enviar missatges cap a AS2. Ara només cal especificar el mòdul PHP que s'encarrega d'enviar i rebre els missatges a AS2, el procés que s'encarrega de gestionar el reenviament de missatges i els clients PHP genèrics que el mòdul PERL fa servir per enviar missatges a AS2 a través del mòdul PHP.

#### 4.2.2- Mòdul PHP: llibreria AS2

El mòdul de PHP està basat en una llibreria de lliure distribució feta en PHP que cobreix les funcions bàsiques d'un servidor AS2. Per tal de poder incorporar AS2 a la infraestructura de



## Disseny, implementació i implantació d'un servidor AS2

l'empresa no hi havia prou amb les funcionalitats bàsiques, o sigui que ha calgut ampliar la llibreria.

El principal problema era la comunicació AS2 mitjançant MDN asíncrons: com està explicat anteriorment, AS2 pot enviar els missatges de servei a la mateixa connexió HTTP (síncron) o amb una connexió HTTP posterior (asíncron). La llibreria contemplava poder rebre aquests MDN asíncrons però no deixava enviar-los, només tenia implementada la resposta MDN síncrona. Per poder solucionar aquest problema vaig implementar la següent funció:

```
public static function handlemdn($part_to, $part_from, $orig, $error){
    try{
        $params = array('partner_from' => $part_from,
            'partner_to' => $part_to);
        $e=new AS2Message("we",$params);
            $e->afheader('Disposition-Notification-Options', 'signed-receipt-protocol=optional, pkcs7-
            signature; signed-receipt-micalg=optional, sha1');
        $mdn = $e->generateMDN();
        $mdn->setAttribute('original-message-id', $orig);
        $mdn->setMessage("Message received");
        if($error == 1){
            $mdn->setAttribute('disposition-modifier', "error");
            $mdn->setAttribute('disposition-type', 'failed');
            $mdn->setMessage("Error processing the message.");
        }
        $mdn->encode($e);

        foreach($mdn->getHeaders() as $key => $value) {
            AS2Log::info(false, "HEADERS MDN: $key => $value\n");
        }

        $mdn->setUrl($mdn->getPartnerTo()->send_url);
        AS2Log::info(false, "URL MDN SEND: ".$mdn->getUrl()."\nCONTENT: ".$mdn->getContent()."\n");
        $client = new AS2Client();
        $result = $client->sendRequest($mdn);
        if ($result['info']['http_code'] == '200'){
            AS2Log::info(false, 'An AS2 ASYNC MDN has been sent MANUALLY!!!');
        }
        else{
            AS2Log::error(false, 'An error occurs while sending MDN message : ' . $result['info']['http_code']);
        }
    }catch(Exception $e){
        throw $e;
    }

    return (1);
}
```

Figura 21 – Funció handlemdn().

Amb aquesta funció que mostra la figura 21 la llibreria permet crear des de zero un MDN asíncron amb totes les dades necessàries per què qualsevol programa AS2 pugui reconèixer el seu format com a vàlid i relacionar-lo amb el seu missatge original fent servir el "original-message-id". A més permet indicar errors si el missatge ha estat incorrecte.

Fent servir aquesta funció podem ampliar notòriament les funcionalitat de la llibreria AS2, ja que aquesta està programada per fer-se servir com una tecnologia P2P, és a dir, entre 2 usuaris finals. En comptes d'això la nostra empresa la fa servir com un gateway d'interconnexió entre N usuaris nostres i un usuari final o una altra empresa que tingui M usuaris.

## Disseny, implementació i implantació d'un servidor AS2

Un altre element important que implica aquesta subrutina és que permet expandir l'abast dels missatges de servei:

Normalment en AS2 el missatge de servei o MDN només serveix per confirmar que el missatge ha arribat correctament i està ben format segons l'estàndard. En canvi, en poder enviar missatges de servei en qualsevol moment, la nostra empresa fa servir els MDN asíncrons com una confirmació de que el missatge EDI que està contingut dins del missatge AS2 ha arribat no solament a la nostra plataforma AS2 sinó també al client final, i a més a més, ha passat totes les proves de control del format EDI.

Aquest control de l'estructura del missatge EDI només seria possible si es fes servir MDN asíncron fent servir el mètode anterior. Per tal de poder estendre aquesta funcionalitat al mode de resposta síncrona vaig modificar la llibreria PHP per tal que executés un fitxer PERL abans de respondre de manera síncrona al missatge, enviant un missatge de servei positiu o negatiu en funció de la variable de retorn de dit script PERL.

Ara ja tenim implementat tot el servidor AS2, és a dir, podem enviar missatges des de el procés central cap a un usuari AS2 i de la mateixa manera rebre missatges provinents de AS2. Només queda especificar dos processos: el reenviament de missatges i els clients PHP. El reenviament de missatges és un procés que s'encarrega d'atendre missatges que per algun motiu no s'han pogut fer arribar al destinatari. Els clients PHP són emprats pel mòdul PERL per enviar peticions d'enviament de missatge al mòdul PHP.

### 4.2.3- Procés de reenviament

Quan s'intenta enviar un missatge a un servidor HTTP extern poden sorgir errors. Aquests errors venen identificats amb un codi que va des de el 0 fins més enllà del 500.

Els codis entre 400 i 499 solen identificar un error de connexió per part de l'emissor, com per exemple posar una adreça incorrecta. En canvi si l'error es superior a 500 vol dir que el servidor al qual intentem connectar té algun problema, com per exemple estar en manteniment. Quan es dona aquest últim cas el procés CCED de AS2 mou els fitxers a una cua auxiliar on un programa PERL s'encarregarà d'intentar reenviar aquests missatges, i durà un control sobre el nombre d'intents de reenviament efectuats sobre cada missatge.

Els missatges que arriben a aquesta cua auxiliar estan sempre en format XML, ja que són missatges que vénen del procés central i s'han de fer arribar a un destinatari AS2.

L'script PERL que controla els reenviaments fa servir un fitxer auxiliar per controlar el nombre de reenviaments que es fa de cada missatge:

Fitxer1.xml 1
Fitxer2.xml 2
Fitxer3.xml 1

## Disseny, implementació i implantació d'un servidor AS2

En aquest exemple es poden veure 3 fitxers XML, el primer i el tercer han estat reenviats 1 vegada, mentre que el segon ja ha estat reenviat 2 cops.

Amb aquest exemple serà més fàcil entendre el comportament de l'script:

```
opendir (DIR,$pathcola) or print "No se pudo acceder al directorio";
while($file= readdir DIR){
    next if( -d $file);          # cuando es un directorio
    next if($file=~/\..swp/);    # cuando se editan o crean ficheros temporales en la COLA (con el vi p.ej)
    next if($file=~/\..\/);
    next if($file !~/\..xml/);

    open(DAT, "$pathcola/aux.txt") or print ("Error abriendo archivo\n");
    my $line=join("", <DAT>);
    close(DAT);
    if ($line =~ /\$file\:\(d)/){
        if($1 > 2){ # MAX Reenvios -> mover a errores.
            print LOG "AS2: $ahora Mover a errores y enviar MDN negativo, File: $file\n";
            $line =~ s/\$file\:\(d)/;/;
            open(DAT, ">$pathcola/aux.txt") or print ("Error abriendo archivo\n");
            print DAT $line;
            close(DAT);

            ### GENERAR MDN NEGATIVO ###
            open(DAT, "$pathcola/$file") or print ("Error abriendo archivo\n");
            my $xml=join("", <DAT>);
            close(DAT);
            $xml =~ /<id_cced>(.)</id_cced>/;
            my $id_cced=$1;
            $xml =~ /<date>(.)</date>/;
            my $timestamp=$1;
            my $xml_cced='<?xml version="1.0" encoding="ISO-8859-1"?>
<ms c="MDN">
<cfg>
    <id_cced>'. $id_cced.'</id_cced>
    <id_as2></id_as2>
    <po_emisor></po_emisor>
    <po_receptor></po_receptor>
    <date>'. $timestamp.'</date>
    <stat>KO</stat>
    <mdn></mdn>
    <error>Reenviado 3 veces por cron e imposible de entregar</error>
</cfg>
<edi></edi>
</ms>';

            open(DAT, ">$pathout/$file.DSN") or print ("Error abriendo archivo\n");
            print DAT $xml_cced;
            close(DAT);

            if (! -d "$patherrores$dirhoy"){
                `mkdir $patherrores$dirhoy`;
            }
            `mv $pathcola/$file $patherrores$dirhoy/`;
            #####
        }
    }
}
```

Figura 22 – Part 1 de la funció de reenviament de missatges.

```
    }else{
        $count=$1+1;
        print LOG "AS2: $ahora File: $file | count: $count\n";
        $line =~ s/$file:\d/$file\;$count/;
        open(DAT, ">$pathcola/aux.txt") or print ("Error abriendo archivo\n");
        print DAT $line;
        close(DAT);
        `mv $pathcola/$file $pathdesti`;
    }
}
}
}
}
```

Figura 23 – Part 2 de la funció de reenviament de missatges.

Aquesta és només una part del codi, però serà suficient per entendre el seu comportament. Com es pot veure es fa una iteració mirant si hi ha algun arxiu XML a la cua auxiliar, depenent de l'arxiu auxiliar es poden donar 3 casos:

- 1- El fitxer XML apareix al fitxer auxiliar i té un comptador de reenviaments superior a 2. En aquest cas el fitxer ha estat reenviat massa vegades sense èxit, per tant es considera que és un fitxer erroni, es mou al directori d'errors i es contesta a l'emissor amb un missatge de servei negatiu.
- 2- El fitxer XML apareix al fitxer auxiliar i té un comptador de reenviaments inferior o igual a 2. En aquest cas només cal incrementar el comptador i tornar a intentar reenviar el document. Cal entendre que el procés de reenviar com a tal el duu a terme el procés CCED AS2, per tant aquest script l'únic que fa es reencuar el fitxer XML a la cua d'entrada de AS2.
- 3- El fitxer XML no apareix al fitxer auxiliar. En aquest cas vol dir que aquesta és la primera vegada que s'intenta reenviar el fitxer, per tant, a més d'enviar al fitxer a la cua d'entrada de AS2, es crea una entrada al fitxer auxiliar que conté el nom del document XML i posa el comptador de reenviaments a 1.

L'únic que queda es executar aquest script PERL de manera desatesa. Per poder fer això cal programar una entrada al CRON del sistema per tal que el procés s'executi cada X minuts. Per exemple programem que s'executi cada 15 minuts.

```
*/15 * * * * /data/scripts-control/colas/colareenviarAS2.pl > /dev/null
```

D'aquesta manera per que es doni un fitxer com a erroni el servidor destí ha d'estar no disponible durant més de 45 minuts (3 reenviaments).

Ara ja tenim especificats tots els elements del servidor tret d'un: els clients genèrics PHP. A continuació es detallarà aquesta darrera part que s'encarrega de comunicar el mòdul PERL amb el mòdul PHP.

## Disseny, implementació i implantació d'un servidor AS2

### 4.2.4- Clients genèrics PHP

Per poder enviar missatges a AS2 cal algun mètode de comunicació entre PERL i la llibreria PHP. Per fer això vaig programar 2 clients genèrics en PHP que fan totes les operacions necessàries per enviar un missatge AS2 i un missatge de servei a AS2.

Quan el procés CCED de AS2 vol enviar un missatge fa una crida fent servir al subrutina enviar\_AS2(). Dins d'aquesta subrutina s'executa les següents línies de codi:

```
my $browser = LWP::UserAgent->new();
my $responde="";
if($modo eq 'MDN'){
    $responde = POST $url, Content => [ from => $partner_from,
    to => $partner_to, orig => $edi, id => $id, error => $error];
}
else{
    $responde = POST $url, Content => [ from => $partner_from,
    to => $partner_to, edi => $edi, id => $id];
}
$sortida=$browser->request($responde)->as_string;
```

Figura 24 – Exemple de comunicació de PERL amb PHP.

Fent servir el mòdul de PERL LWP::UserAgent s'envia una petició HTTP a l'adreça URL on es troba el client adient, passant-li per POST les variables que necessita. Amb la variable \$sortida posteriorment es comprova si hi ha hagut algun error.

El client genèric que envia un missatge AS2 té la següent forma:

```
<?php
include_once('include.inc');
if($_POST['from'] != ""){

    $params = array('partner_from' => $_POST['from'],
    'partner_to' => $_POST['to']);

    $tmp_file = AS2Adapter::getTempFilename();
    echo "Temp file: $tmp_file\n";
    file_put_contents($tmp_file, $_POST['edi']);

    //AS2Log::info(false, "Cliente generico PHP: mensaje de $_POST['from'] a $_POST['to']\n->contenido:
    $_POST['edi']\n");

    file_put_contents("log_clientcced", $_POST['edi']);

    $message = new AS2Message(false, $params);
    $message->addFile($tmp_file);
    $message->encode($_POST['id']);

    echo "Ini sendRequest\n";
    $client = new AS2Client();
    $result = $client->sendRequest($message);
    echo "Fi sendRequest\n";
    //var_dump($result['response']);

    if($result['response'] != null){
        AS2Server::creaMDNSync($result['response']);
    }
    //echo "Fi CreaMDNSYNC!\n";
}
else{
    echo "No se puede ejecutar este cliente sin las variables necesarias!!\n";
}
?>
```

Figura 25 – Client genèric PHP.

## Disseny, implementació i implantació d'un servidor AS2

Com es pot veure rep les variables referents a l'emissor, al receptor, i al propi missatge per POST, i executa la funció `sendRequest()` després de preparar el missatge per poder fer-lo arribar al destinatari AS2.

Aquí es poden veure els passos necessaris per enviar un document a AS2. Primerament s'agafen les dades dels partners emissor i receptor, seguidament es posa el contingut del missatge a un fitxer auxiliar. Després es crea un objecte de tipus `AS2Message` donant els paràmetres dels partners i se li afegeix el fitxer auxiliar com a document. Finalment es fa un procés de `encode()` que és on s'encripta el missatge i es signa, si així ho requereixen els paràmetres dels partners, fent servir els certificats.

El client genèric per enviar MDN a AS2 és pràcticament idèntic a l'anterior, amb l'única diferència que es crida la funció `handlemdn()` citada anteriorment. Quan es crida al client genèric de MDN s'envien uns paràmetres diferents a quan es crida al client genèric de missatges:

- No s'envia contingut del fitxer, ja que un MDN no envia ningun document.
- S'envia una variable que determina si el MDN ha de ser positiu o negatiu.
- S'envia l'explicació de l'error per ser inclòs al cos del MDN, en cas que aquest sigui negatiu.
- S'envia el `message-id` del missatge original per tal que al cos del MDN pugui aparèixer el camp `original-message-id` i així el servidor AS2 que rebí aquest MDN podrà relacionar-lo.

Amb l'especificació d'aquest darrer element ja s'ha completat la implementació de tot el projecte. Ara cal avaluar que els resultats obtinguts d'aquesta fase d'implementació compleixen els requeriments i objectius fixats a la fase de disseny.

## 5 - AVALUACIÓ DEL SERVIDOR AS2

L'objectiu d'aquest capítol és avaluar el servidor AS2.

Per tal de considerar el servidor AS2 funciona correctament, aquest ha de poder enviar i rebre tot tipus de missatge AS2 (amb MDN síncron o asíncron), gestionant correctament els missatges de servei. A més ha de mantenir un historial d'aquests missatges, indicant la data i hora del seu moment d'enviament o recepció, l'estat en el que es troben, i el control dels errors.

Altres requisits que es van fixar a la part de disseny són:

- L'enviament de missatges a AS2 de manera totalment transparent per part d'un client de la nostra empresa. Aquest requeriment està inclòs en el disseny.
- Els MDN síncrons han d'arribar a l'emissor del missatge com si es tractés d'un MDN asíncron per tal de notificar si el missatge ha arribat correctament o no al destinatari.
- La gestió del servidor amb una eina web d'administració.

En els següents apartats es mostraran totes les proves que s'han fet per validar que el servidor AS2 funciona correctament i que compleix els objectius i requeriments que es van fixar a la fase de disseny. Al final d'aquest capítol es llistaran les conclusions extretes de la fase d'avaluació.

### 5.1- Eines emprades per avaluar el servidor AS2

Per poder dur a terme tots els test de la manera més fiable farà servir les següents eines:

#### 5.1.1- ComeDi

ComeDi és un software que la nostra empresa ofereix als seus clients per gestionar enviaments, traducció i rebuda de missatges EDI. El programa envia els fitxers al procés central de l'empresa per SMTP i rep de la mateixa manera els missatges de servei.

Fent servir aquest programa simulem al 100% l'entorn d'execució real de bloc AS2, a més facilita enormement la traducció dels fitxers i la comprovació de la seva integritat.

#### 5.1.2- Mendelson AS2 Free Client

En l'etapa de desenvolupament normalment feia les proves d'enviament i recepció entre dues instàncies de la nostra plataforma AS2. Per tal d'assegurar que el funcionament de la nostra plataforma compleix els estàndards com qualsevol altre programa comercial farà servir un programa de distribució gratuïta però que molts usuaris fan servir en entorn real de treball.

## Disseny, implementació i implantació d'un servidor AS2

El programa client de Mendelson està implementat en Java i té dues versions, la gratuïta i la de pagament, la versió gratuïta només té implementades les funcionalitats bàsiques, i ha de ser manipulada manualment, enviant missatges un per un, la versió de pagament és molt completa i és feta servir per moltes empreses importants, però el seu preu és extremadament alt.

Tot i això serà molt útil per simular a un usuari extern AS2 que no fa servir el nostre software. El seu funcionament és simple i la seva interfície gràfica ajudarà a veure ràpidament el resultat de les proves.

### 5.1.3- Logs del sistema

Tot i que normalment no serà necessari consultar-los, els logs de `httpd_error` i `httpd_access` que es troben al directori `/var/log/httpd/` de qualsevol sistema UNIX mostren informació que ens pot interessar per detectar errors a les proves.

El log `httpd_error` indica els errors de HTTP, amb el seu error code i la causa d'aquest. El log `httpd_access` indica quins arxius del nostre sistema han estat executats per Apache.

Cal tenir en compte que si es fa una connexió HTTPS (connexió segura xifrada amb certificat) els logs que cal consultar són `ssl_httpd_error` i `ssl_httpd_access`, tot i que els seus noms poden ser diferents depenent de la màquina.

### 5.1.4- Log del mòdul PHP AS2

La llibreria implementa un log on es mostra informació de l'activitat del bloc PHP AS2. En la fase de desenvolupament es va ampliar notòriament la funcionalitat d'aquest bloc i a més a més vaig ampliar la informació mostrada a aquest log.

El log és el fitxer `events.log` que està a la carpeta `logs` dins del directori arrel del bloc AS2. Entre la informació que vaig afegir al log serà especialment útil la impressió dels headers i cos de tots els missatges que entren i surten de AS2, ja que ens permetran observar en quin moment es detecta un error.

### 5.1.5- PhpMyAdmin

PhpMyAdmin és un visor de bases de dades implementat en PHP que permet veure ràpidament taules i el seu contingut, entre altres coses. El faré servir per comprovar que tota la traçabilitat dels missatges es fa correctament.



## 5.2- Proves

Les proves consistiran en enviar diferents tipus de fitxers amb diferents configuracions de partners per tal de poder comprovar el correcte funcionament de totes les casuístiques possibles.

Tot i que es podria provar també d'enviar des de totes les possibles fonts d'origen, no es farà ja que el procés central CCED de l'empresa funciona correctament i sempre deixarà el mateix a la cua d'entrada: un XML. L'origen del missatge mai afectarà de cap manera el comportament del bloc AS2.

Les proves consistiran en enviaments de missatges desde ComeDi i el client Mendelson per simular l'enviament de missatges des de un client intern o un usuari extern, respectivament. S'enviaran diferents tipus de fitxers, configurant els partners de maneres diverses per poder provar tots els casos d'execució.

Per tant les possibles casuístiques que cal comprovar són:

- Enviament d'un missatge correcte amb MDN síncron. En la mateixa connexió se'ns contestarà amb un MDN positiu síncron que haurem de propagar i fer arribar al l'emissor.
- Enviament d'un missatge incorrecte amb MDN síncron. En la mateixa connexió se'ns contestarà amb un MDN negatiu síncron que haurem de propagar i fer arribar al l'emissor.
- Enviament d'un missatge correcte amb MDN asíncron.
- Enviament d'un missatge incorrecte amb MDN asíncron.
- Rebuda d'un missatge de servei MDN asíncron incorrecte estructuralment.
- Rebuda d'un missatge de servei MDN asíncron correcte estructuralment però amb original-message-id desconegut.
- Rebuda d'un missatge correcte amb MDN síncron. En la mateixa connexió contestarem amb un MDN positiu síncron.
- Rebuda d'un missatge incorrecte amb MDN síncron. En la mateixa connexió contestarem amb un MDN negatiu síncron.
- Rebuda d'un missatge correcte amb MDN asíncron.
- Rebuda d'un missatge incorrecte amb MDN asíncron.
- Rebuda d'un missatge correcte des de el punt de vida AS2 amb MDN síncron, però amb el contingut EDI amb format incorrecte. En la mateixa connexió contestarem amb un MDN positiu síncron.
- Rebuda d'un missatge AS2 amb MDN asíncron que conté diversos EDI, control de tots els missatges de servei relacionats amb aquest missatge i l'enviament final del MDN.

## Disseny, implementació i implantació d'un servidor AS2

- Rebuda d'un missatge AS2 amb MDN asíncron que conté diversos EDI on algun dels missatges que conté és incorrecte, control de tots els missatges de servei relacionats amb aquest missatge i l'enviament final del MDN.
- Rebuda d'un missatge AS2 amb MDN asíncron que conté diversos EDI on tots els missatges que conté són incorrectes, control de tots els missatges de servei relacionats amb aquest missatge i l'enviament final del MDN.

Però a la pràctica moltes d'aquestes casuístiques queden provades en provar-ne d'altres:

- 1- Enviar un missatge des de comeDi a un usuari AS2 amb MDN síncron o asíncron és idèntic des de el punt de vista del nostre servidor AS2, ja que qui contesta de manera diferent és l'usuari extern. Per tant provant només un dels dos casos ja validem tots dos.
- 2- Enviar un missatge incorrecte des de comeDi a un usuari AS2 és per definició impossible. ComeDi controla i valida els missatges que envia per tal que siguin estructuralment correctes. A més el procés central de l'empresa fa la validació de que els punts operacionals implicats en el missatge siguin correctes i es puguin comunicar entre ells. L'únic error que es podria tenir és que per algun motiu el missatge no es pogués entregar al servidor receptor tot i ser correcte, on el servidor enviarà el missatge a la cua de reenviament per ser enviat posteriorment. Tot i que en principi no poden arribar missatges incorrectes des de els nostres usuaris interns, provar aquest cas forçant un error servirà per comprovar que el servidor està preparat contra aquest contratemps.

Per tant les proves que es faran seran:

- Enviament de missatge AS2 amb MDN asíncron.
- Enviament de missatge AS2 amb MDN síncron, que queda provat amb l'anterior.
- Enviament de missatge AS2 erroni, forçant un error de reenviament.
- Recepció de missatge AS2 amb MDN asíncron.
- Recepció de missatge AS2 amb MDN síncron.
- Recepció de missatges erronis.

### 5.2.1- Enviament de missatge AS2 amb MDN asíncron

La prova consisteix en enviar un missatge EDI des de comeDi a un punt operacional configurat com un usuari AS2 i comprovar que es rep correctament el missatge de servei asíncron. Com que el missatge que s'envia es correcte el missatge de servei ha de ser positiu. La configuració de si es requereix MDN síncron o asíncron es fa a la base de dades del servidor AS2, però per requeriment de l'empresa els nostres clients sempre requeriran MDN asíncron.

## Disseny, implementació i implantació d'un servidor AS2

Primerament cal configurar el partner a ComeDi, es crea un interlocutor amb el punt operacional on volem enviar i es configura que el procés central el tracti com un interlocutor AS2.

Seguidament enviem el missatge EDI:

```
UNA:+.?  
'UNB+UNOC:3+8420418000003:14+8714253049021:14+110304:1304+1  
204'UNH+1204+DESADV:D:01B:UN:EAN007'BGM+351+110150+9'DTM+  
137:20110209:102'DTM+2:20110217:102'RFF+ZZZ:EANL3'RFF+ON:P11  
00615'NAD+BY+8714253049021::9'NAD+SU+8420418000003::9'NAD+D  
P+8712423001374::9'TOD+4'CPS+1'CPS+2+1'PAC+80'PCI+33E'GIN+BJ+38  
4204180000000059'LIN+1+++8420418002038:SRV'PIA+1+200070-  
006:SN'IMD+F+++::MARQUES DE MOJA  
BRUT'QTY+12:480'ALI+++66'DTM+94:20110204'UNT+22+1204'UNZ+1+1  
204'
```

Figura 26 – Exemple de missatge en format EDI.

Aquest és un EDI de tipus DESADV, com que el tipus de l'EDI no afecta a les proves mentre sigui correcte farà servir diferents EDI.

Un cop el missatge arriba al procés central aquest detecta que l'usuari destí, que té el PO (punt operacional) 8714253049021, pertany a un usuari AS2 i l'envia a la cua d'entrada del nostre servidor AS2.

El procés CCED de AS2 detecta aquest nou missatge i el processa:

## Disseny, implementació i implantació d'un servidor AS2

```

---13:05:33-----<OPEN>-----[AS2_IN|AS2_OUT|RAW_AS2]]
PATH_ARCHIVO: /data/COLAS/AS2-int/IN_AS2/12156-ev00000002.xml
$VAR1 = {
  'cfg' => {
    'id_cced' => '12156',
    'po_emisor' => '8420418000003',
    'date' => '110304130529',
    'po_receptor' => '8714253049021'
  },
  'edi' => 'UNA:+.?'
  \UNB+UNOC:3+8420418000003:14+8714253049021:14+110304:1304+1204\UNH+1204+DESADV:D:01B:UN:EAN007\BGM+
  351+110150+9\DTM+137:20110209:102\DTM+2:20110217:102\RFF+ZZZ:EANL3\RFF+ON:P1100615\NAD+BY+87142530
  49021::9\NAD+SU+8420418000003::9\NAD+DP+8712423001374::9\TOD+4\CPS+1\CPS+2+1\PAC+80\PCI+33E\GIN+BJ+3
  8420418000000059\LIN+1++8420418002038:SRV\PIA+1+200070-006:SN\IMD+F++::MARQUES DE MOJA
  BRUT\QTY+12:480\ALI+++66\DTM+94:20110204\UNT+22+1204\UNZ+1+1204\,
  'c' => 'EDI'
  };
  12156-ev00000002.xml ---> EDI
  id_cced: 12156
  id_as2:
  emisor: 8420418000003
  receptor: 8714253049021
  data: 2011-03-04 13:05:29
  edi: UNA:+.?
  \UNB+UNOC:3+8420418000003:14+8714253049021:14+110304:1304+1204\UNH+1204+DESADV:D:01B:UN:EAN007\BGM+35
  1+110150+9\DTM+137:20110209:102\DTM+2:20110217:102\RFF+ZZZ:EANL3\RFF+ON:P1100615\NAD+BY+8714253049021::
  9\NAD+SU+8420418000003::9\NAD+DP+8712423001374::9\TOD+4\CPS+1\CPS+2+1\PAC+80\PCI+33E\GIN+BJ+38420418000000
  0059\LIN+1++8420418002038:SRV\PIA+1+200070-006:SN\IMD+F++::MARQUES DE MOJA
  BRUT\QTY+12:480\ALI+++66\DTM+94:20110204\UNT+22+1204\UNZ+1+1204\
  &enviar_as2(8435178700025, 8714253049021, UNA:+.?
  \UNB+UNOC:3+8420418000003:14+8714253049021:14+110304:1304+1204\UNH+1204+DESADV:D:01B:UN:EAN007\BGM+35
  1+110150+9\DTM+137:20110209:102\DTM+2:20110217:102\RFF+ZZZ:EANL3\RFF+ON:P1100615\NAD+BY+8714253049021::
  9\NAD+SU+8420418000003::9\NAD+DP+8712423001374::9\TOD+4\CPS+1\CPS+2+1\PAC+80\PCI+33E\GIN+BJ+38420418000000
  0059\LIN+1++8420418002038:SRV\PIA+1+200070-006:SN\IMD+F++::MARQUES DE MOJA
  BRUT\QTY+12:480\ALI+++66\DTM+94:20110204\UNT+22+1204\UNZ+1+1204\, EDI, 12156-ev00000002-110304130533.as2)
----->CLOSE<-----

```

Figura 27 – Log de la cua AS2\_IN de CCED AS2.

El procés CCED AS2 llegeix el XML, processa el missatge i envia al bloc PHP la petició d'enviar-lo.

← T →	id_tr_msg	id_emisor	id_receptor	id_msg	fecha	tipo_mdn	error
<input type="checkbox"/>	186	8420418000003	8714253049021	12080-ev00000002-110215130509.as2	2011-02-15 13:05:09	ASYNCR	0
<input type="checkbox"/>	187	8420418000003	8714253049021	12080-ev00000002-110215130708.as2	2011-02-15 13:07:08	ASYNCR	0
<input type="checkbox"/>	188	8420418000003	8714253049021	12080-ev00000002-110215130757.as2	2011-02-15 13:07:57	ASYNCR	0
<input type="checkbox"/>	189	8420418000003	8714253049021	12080-ev00000002-110215131035.as2	2011-02-15 13:10:35	ASYNCR	0
<input type="checkbox"/>	190	8420418000003	8714253049021	12156-ev00000002-110304130533.as2	2011-03-04 13:05:33	ASYNCR	0

Figura 28 – Traça de missatges a la taula tr\_msg.

A l'última línia d'aquesta imatge del PhpMyAdmin on es mostra una part de la taula tr\_msg podem veure la traçabilitat d'aquest missatge.

Passats uns segons el servidor rep un missatge entrant.

```

---14:24:03-----<OPEN>-----[AS2_IN|AS2_OUT|RAW_AS2]]
PATH_ARCHIVO: /data/CCED_AS2-int/RAW_AS2/20110209142403430_141.93.32.47.as2.header
data_file: /data/CCED_AS2-int/RAW_AS2/20110209142403430_141.93.32.47.as2.header
partner_from: 8714253049021 -> emisor: 8714253049021
partner_to: 8435178700025 -> receptor: 8435178700025
buzon:
tipo_mdn: ASYNCR
+ES MDN -> ORIG-MESSAGE-ID: 12156-ev00000002-110304130533.as2
----->CLOSE<-----

```

Figura 29 – Log de la cua RAW\_AS2 de CCED AS2.

A "ORIG-MESSAGE-ID" podem veure com és un MDN que correspon al missatge que hem enviat anteriorment.

←T→	id_tr_intercambio	id_tr_msg	id_documento	nombre_mdn	fecha_mdn	id_cced
<input type="checkbox"/>	41	156	11950-ev00000002-110202123708.as2	20110202123747480_141.93.32.47.as2	2011-02-02 12:37:52	11950
<input type="checkbox"/>	42	157	11951-ev00000002-110202124833.as2	20110202124910949_141.93.32.47.as2	2011-02-02 12:49:17	11951
<input type="checkbox"/>	43	158	12056-ev00000002-110209142325.as2	20110209142403430_141.93.32.47.as2	2011-02-09 14:24:03	12056

Figura 30 – Traça de missatges a la taula tr\_intercambio.

Aquesta és una part de la taula tr\_intercambio, on es pot veure a la última línia com s'han introduït correctament les dades del MDN. Un cop processat aquest missatge entrant, es crea un XML a la cua de sortida per poder fer arribar el missatge de servei a l'emissor del missatge original.

```

---14:24:08-----<OPEN>-----[AS2_IN|AS2_OUT|RAW_AS2]]
|FILES move to 'CCED':
/usr/bin/ncftpput -a -u ftpccedas2-int -p ftpccedas2-int14 -DD 62.97.114.52 / /data/COLAS/AS2-int/OUT_AS2/*
/data/COLAS/AS2-int/OUT_AS2/12056-ev00000002-110209142325.as2.DSN:^M...0000002-110209142325.as2.DSN: ETA:
0:00 256.00/276.00 B 5.55 MB/s ^M...0000002-110209142325.as2.DSN: ETA: 0:00 276.00/276.00 B 42.09 kB/s
^M...int/OUT_AS2/12056-ev00000002-110209142325.as2.DSN: 276.00 B 42.09 kB/s
...int/OUT_AS2/12056-ev00000002-110209142325.as2.DSN: 276.00 B 42.09 kB/s
|File[1]: /data/COLAS/AS2-int/OUT_AS2/12056-ev00000002-110209142325.as2.DSN
----->CLOSE<-----
    
```

Figura 31 – Log de la cua AS2\_OUT de CCED AS2.

Aquí el missatge s'entrega al procés central per fer-lo arribar a l'emissor, en aquest cas nosaltres. A comeDi es rep el missatge de servei confirmant que la prova és correcta.

### 5.2.2- Enviament de missatge AS2 amb MDN síncron

El procés de prova és idèntic, ja que la única diferència és que el servidor AS2 de l'usuari receptor ens contestarà a la mateixa connexió HTTP, però el procés que el nostre servidor AS2 duu a terme per tractar aquesta casuística és el mateix que per un MDN síncron.

### 5.2.3- Enviament de missatge AS2 erroni

En aquesta prova forçaré un error d'enviament del missatge. Normalment aquest error no seria possible, o altament improbable, però forçar-lo ens permet comprovar que el servidor AS2 està preparat davant aquest imprevist. Per forçar aquesta casuística s'enviarà un missatge AS2 correcte des de comeDi però es configurarà el seu partner incorrectament per tal que el sistema detecti un error i envii el missatge a la cua de reenviament. Un cop el missatge sigui reenviat 3 cops sense èxit es generarà un missatge de servei negatiu que ha d'arribar fins l'emissor del missatge.

Com s'ha pogut veure abans, si el missatge va d'un usuari nostre a un usuari AS2 les proves són idèntiques tant si el missatge de servei es síncron com asíncron.

## Disseny, implementació i implantació d'un servidor AS2

Cal recordar que si es troba un error al intentar enviar el missatge i aquest error es de tipus 5XX el missatge serà enviat a la cua de reenviament. Si el missatge és reenviat 3 cops sense èxit aleshores serà tractat com un cas erroni normal.

Per tant si fem la prova del cas on l'error és 5XX en realitat estem comprovant que tots 2 casos són correctes.

Igual que en l'enviament d'un missatge correcte s'efectuen tots els passos necessaris per que el missatge arribi a la plataforma AS2. Cal tenir en compte que si el missatge no arriba al bloc AS2 vol dir que el procés central CCED l'ha rebutjat, i que per tant no és un exemple de test vàlid.

Un cop el missatge arriba a la cua d'entrada de AS2 el procés CCED AS2 el processa i observem la següent informació al log.

## Disseny, implementació i implantació d'un servidor AS2

```
---13:06:03-----<OPEN>-----[AS2_IN|AS2_OUT|RAW_AS2]]
PATH_ARCHIVO: /data/COLAS/AS2-int/IN_AS2/12387-ev00000002.xml
$VAR1 = {
    'cfg' => {
        'id_cced' => '12387',
        'po_emisor' => '8420418000003',
        'date' => '110412130548',
        'po_receptor' => '8714253049021'
    },
    'edi' => 'UNA:+.?'
}
\UNB+UNOC:3+8420418000003:14+8714253049021:14+110412:1305+1226\UNH+1226+DESADV:D:01B:UN:EAN007\BGM
+351+110150+9\DTM+137:201102091200:203\DTM+2:20110217:102\RFF+ZZZ:EANL4\RFF+ON:P1100615\NAD+BY+871
4253049021::9\NAD+SU+8420418000003::9\NAD+DP+8712423001374::9\TOD+4\CPS+1\CPS+2+1\CPS+3+2\PAC+80\PCI
+33E\GIN+BJ+38420418000000059\LIN+1+++8420418002038:SRV\PIA+1+200070-006:SN\IMD+F+++::MARQUES DE MOJA
BRUT\QTY+12:480\ALI+++66\DTM+94:20110204\UNT+23+1226\UNZ+1+1226",
    'c' => 'EDI'
};
12387-ev00000002.xml ---> EDI
id_cced: 12387
id_as2:
emisor: 8420418000003
receptor: 8714253049021
data: 2011-04-12 13:05:48
edi: UNA:+.?'
\UNB+UNOC:3+8420418000003:14+8714253049021:14+110412:1305+1226\UNH+1226+DESADV:D:01B:UN:EAN007\BGM+35
1+110150+9\DTM+137:201102091200:203\DTM+2:20110217:102\RFF+ZZZ:EANL4\RFF+ON:P1100615\NAD+BY+8714253049
021::9\NAD+SU+8420418000003::9\NAD+DP+8712423001374::9\TOD+4\CPS+1\CPS+2+1\CPS+3+2\PAC+80\PCI+33E\GIN+BJ+38
420418000000059\LIN+1+++8420418002038:SRV\PIA+1+200070-006:SN\IMD+F+++::MARQUES DE MOJA
BRUT\QTY+12:480\ALI+++66\DTM+94:20110204\UNT+23+1226\UNZ+1+1226'
&enviar_as2(8435178700025, 8714253049021, UNA:+.?'
\UNB+UNOC:3+8420418000003:14+8714253049021:14+110412:1305+1226\UNH+1226+DESADV:D:01B:UN:EAN007\BGM+35
1+110150+9\DTM+137:201102091200:203\DTM+2:20110217:102\RFF+ZZZ:EANL4\RFF+ON:P1100615\NAD+BY+8714253049
021::9\NAD+SU+8420418000003::9\NAD+DP+8712423001374::9\TOD+4\CPS+1\CPS+2+1\CPS+3+2\PAC+80\PCI+33E\GIN+BJ+38
420418000000059\LIN+1+++8420418002038:SRV\PIA+1+200070-006:SN\IMD+F+++::MARQUES DE MOJA
BRUT\QTY+12:480\ALI+++66\DTM+94:20110204\UNT+23+1226\UNZ+1+1226', EDI, 12387-ev00000002-
110412130603.as2@as2.ediversa.net)
ERROR: el HTTPRequest a la url http://as2test.ediversa.net/clientcced.php no ha finalizado correctamente
SORTIDA: HTTP/1.1 200 OK
Connection: close
Date: Tue, 12 Apr 2011 11:06:03 GMT
Server: Apache/2.2.3 (CentOS)
Content-Length: 451
Content-Type: text/html; charset=UTF-8
Client-Date: Tue, 12 Apr 2011 11:06:07 GMT
Client-Peer: 62.97.114.59:80
Client-Response-Num: 1
X-Powered-By: PHP/5.2.10

Temp file: /tmp/as2file_econQB
Ini sendRequest
<br />
<b>Fatal error</b>: Uncaught exception 'AS2Exception' with message 'HTTP Error Code :
403(url:http://rubidium.ah.nl:4080/as2-acc/invoke/wm.EDIINT/receiveeeee)' in /data/WEB/AS2-int/lib/AS2Client.php:92
Stack trace:
#0 /data/WEB/AS2-int/www/clientcced.php(54): AS2Client-&sendRequest(Object(AS2Message))
#1 {main}
  thrown in <b>/data/WEB/AS2-int/lib/AS2Client.php</b> on line <b>92</b><br />
```

Figura 32 – Log de la cua AS2\_IN de CCED AS2 amb problema d'entrega.

```
&copytodir('/data/COLAS/AS2-int/IN_AS2/12387-ev00000002.xml','/data/scripts-control/colas/fixters_moguts/AS2-int/') if
(12387-ev00000002.xml != '')
TROBAT_FROM: /data/COLAS/AS2
TROBAT_TO: /data/scripts-control/colas/fixters_moguts/AS2
F.FROM:/data/COLAS/AS2-int/IN_AS2/12387-ev00000002.xml
F.TO :/data/scripts-control/colas/fixters_moguts/AS2-int/
DIRECTORI DESTI: /data/scripts-control/colas/fixters_moguts/AS2-int/
----->CLOSE<-----
```

Figura 33 – Part 2 de log de la cua AS2\_IN de CCED AS2 amb problema d'entrega.

## Disseny, implementació i implantació d'un servidor AS2

Normalment no apareix tanta informació al log, però per comprovar els tests de validesa posem una variable anomenada \$DEBUG a 1 per tal que es mostri informació extra.

Com es pot veure el missatge es processa correctament, però quan el clientcced.php (client genèric PHP per enviar missatges) intenta enviar el missatge aquest retorna un error 403, ja que l'adreça a la qual intenta enviar el missatge no existeix.

Cal aclarir que per forçar aquesta casuística he modificat l'adreça a una que no tingui sentit i he modificat el "if" que comprova si l'error es 5XX per tal que doni TRUE si l'error es 4XX. Això s'ha fet així perquè replicar un error 5XX no és trivial, cal donar de baixa servidors, o parar serveis que són necessaris per altres programes, per tant és molt més fàcil tocar un parell de línies de codi i no afectar a la resta del sistema.

Per tant, tal com està ara mateix programat el procés CCED AS2, el missatge serà mogut a la cua de reenviament, i en intervals d'aproximadament 15min s'intentarà reenviar un total de 3 vegades, apareixen 3 vegades el mateix error pel log ja que no modifiquem l'adreça.

Un cop el script de reenviament falla el tercer reenviament genera el missatge de servei negatiu a la cua de sortida de AS2.

```
---13:10:05-----<OPEN>-----[AS2_IN|AS2_OUT|RAW_AS2]]
|FILES move to 'CCED':
/usr/bin/ncftpput -a -u ftpccedas2-int -p ftpccedas2-int14 -DD 62.97.114.52 / /data/COLAS/AS2-int/OUT_AS2/*
/data/COLAS/AS2-int/OUT_AS2/12387-ev00000002.xml.DSN: ^M...AS2/12387-ev00000002.xml.DSN: ETA: 0:00
256.00/311.00 B 5.55 MB/s ^M...AS2/12387-ev00000002.xml.DSN: ETA: 0:00 311.00/311.00 B 425.96 kB/s
^M/data/COLAS/AS2-int/OUT_AS2/12387-ev00000002.xml.DSN: 311.00 B 425.96 kB/s
/data/COLAS/AS2-int/OUT_AS2/12387-ev00000002.xml.DSN: 311.00 B 425.96 kB/s
|File[1]: /data/COLAS/AS2-int/OUT_AS2/12387-ev00000002.xml.DSN
----->CLOSE<-----
```

Figura 34 – Log OUT\_AS2 en cas de reenviament incorrecte.

El missatge XML arriba al procés central CCED i aquest l'envia a la nostra estació comeDi, on veiem que el missatge queda marcat com rebutjat.

		id_tr_msg	id_emisor	id_receptor	id_msg	fecha	tipo_mdn	error
<input type="checkbox"/>		216	8420418000003	8714253049021	12387-ev00000002-110412130603.as2@as2.ediversa.net	2011-04-12 13:06:03	ASYNCR	1
<input type="checkbox"/>		217	8420418000003	8714253049021	12387-ev00000002-110412130703.as2@as2.ediversa.net	2011-04-12 13:07:03	ASYNCR	1
<input type="checkbox"/>		218	8420418000003	8714253049021	12387-ev00000002-110412130803.as2@as2.ediversa.net	2011-04-12 13:08:03	ASYNCR	1
<input type="checkbox"/>		219	8420418000003	8714253049021	12387-ev00000002-110412130903.as2@as2.ediversa.net	2011-04-12 13:09:03	ASYNCR	1

Figura 35 – Traça de missatge reenviat incorrecte de la taula tr\_msg.

Aquí podem observar com la base de dades té enregistrades 4 entrades pel missatge 12387-ev00000002, l'enviament original i els 3 reenviaments, i com totes elles han tingut un error.

		id_tr_intercambio	id_tr_msg	id_documento	nombre_mdn	fecha_mdn	id_cced
<input type="checkbox"/>		101	216	12387-ev00000002-110412130603.as2@as2.ediversa.net	NULL	NULL	12387
<input type="checkbox"/>		102	217	12387-ev00000002-110412130703.as2@as2.ediversa.net	NULL	NULL	12387
<input type="checkbox"/>		103	218	12387-ev00000002-110412130803.as2@as2.ediversa.net	NULL	NULL	12387
<input type="checkbox"/>		104	219	12387-ev00000002-110412130903.as2@as2.ediversa.net	NULL	NULL	12387

Figura 36 – Traça de missatge reenviat incorrecte de la taula tr\_intercambio.

A la taula tr\_intercambio es pot confirmar com ningun dels 4 enviaments ha rebut cap mena de MDN, ja que no s'han pogut fer arribar al receptor.



#### 5.2.4- Recepció d'un missatge AS2 amb MDN asíncron

En aquesta prova enviarem un missatge des de Mendelson per simular l'enviament d'un missatge des de un usuari AS2 extern. El missatge estarà configurat per demanar un MDN asíncron. Aquest missatge ha d'arribar fins a un client intern correctament i un missatge de servei positiu ha d'arribar al nostre programa Mendelson.

Com que ara els que rebem el missatge són nosaltres, qui envia és un usuari AS2 extern, per simular això faré servir Mendelson AS2.

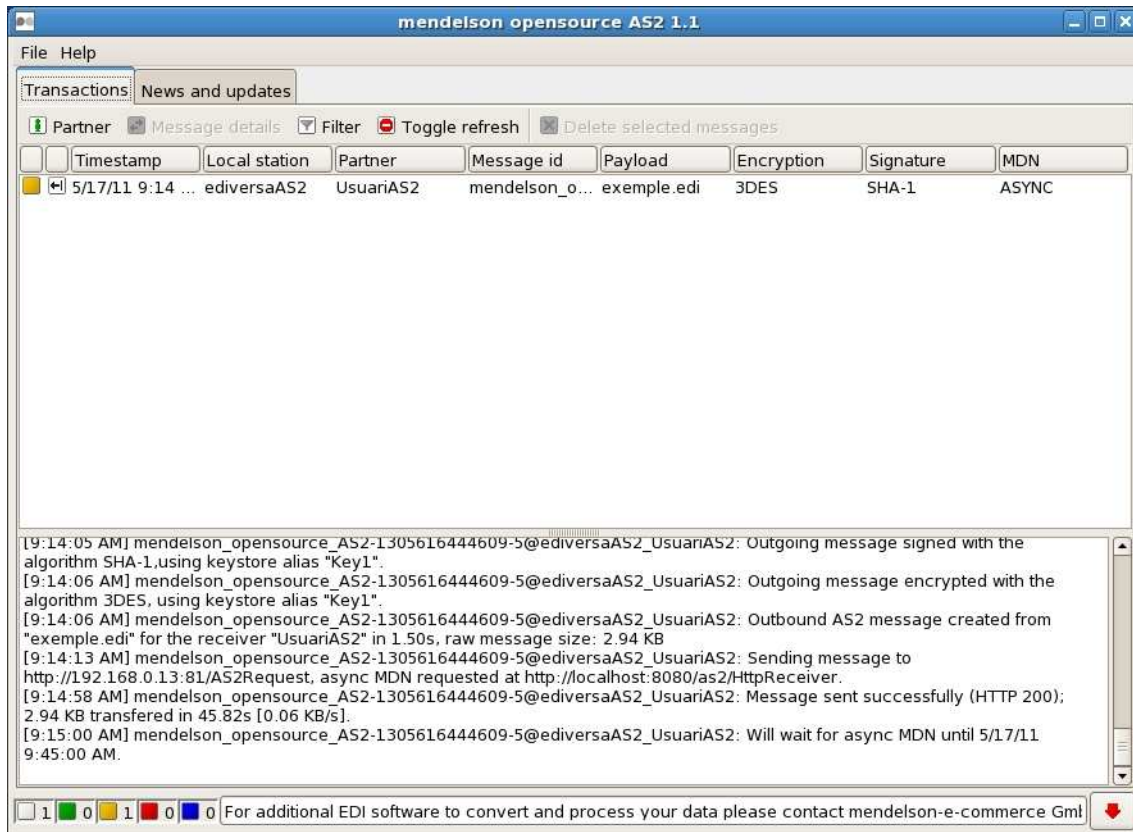


Figura 37 – Enviament de missatge amb MDN asíncron desde Mendelson.

Fent servir Mendelson AS2 enviem el missatge al nostre servidor AS2. Com es pot veure Mendelson es queda esperant a que arribi el MDN asíncron.

Al directori RAW\_AS2 podem veure com el missatge ha arribat i s'ha pogut comprovar la signatura i descriptar.

```
[root@centosKim RAW_AS2]# ll
total 24K
drwxr-xr-x 8 root root 4,0K may 13 12:50 ..
-rw-rw-rw- 1 apache apache 879 may 13 12:55 20110513125554857_192.168.0.13.as2.header
-rw-rw-rw- 1 apache apache 3,0K may 13 12:55 20110513125554857_192.168.0.13.as2
-rw-rw-rw- 1 apache apache 2,6K may 13 12:55 20110513125554857_192.168.0.13.as2.decrypted
-rw-rw-rw- 1 apache apache 212 may 13 12:56 20110513125554857_192.168.0.13.as2.payload_0
drwxr-xr-x 2 apache apache 4,0K may 13 12:56 .
```

Al CCED AS2 podem veure com el missatge entrant de la cua RAW\_AS2 es processa:

## Disseny, implementació i implantació d'un servidor AS2

```
(cola_raw_as2)
---13:27:28-----<OPEN>-----[AS2_IN|RAW_AS2]]
PATH_ARCHIVO: /data/CCED_AS2-int/RAW_AS2/20110513132718900_192.168.0.13.as2.header
data_file: /data/CCED_AS2-int/RAW_AS2/20110513132718900_192.168.0.13.as2.header
Numero Payload: 1 -> Numero Payload real: 1
partner_from: ediversaAS2 -> emisor: POAS2
partner_to: UsuariAS2 -> receptor: 8435178700308
buzon: ev00000004
tipo_mdn: ASYNC
TROBAT_FROM: /data/CCED_AS2
TROBAT_TO: /data/backup/
F.FROM:/data/CCED_AS2-int/RAW_AS2/20110513132718900_192.168.0.13.as2*
F.TO :/data/backup/COLAS/as2/backups/20110513/
DIRECTORI DESTI: /data/backup/COLAS/as2/backups/20110513/
mv /data/CCED_AS2-int/RAW_AS2/20110513132718900_192.168.0.13.as2* /data/backup/COLAS/as2/backups/20110513/
----->CLOSE<-----
```

Figura 38 – Log cua RAW\_AS2 per missatge entrant.

I com a resultat a la cua AS2\_OUT trobem el següent XML:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ms c="EDI">
<cfg>
    <id_cced></id_cced>
    <id_as2>114</id_as2>
    <po_emisor>POAS2</po_emisor>
    <po_receptor>8435178700308</po_receptor>
    <date>110513132728</date>
    <stat></stat>
    <error></error>
    <mdn>ASYNC</mdn>
</cfg>
<edi><![CDATA[UNA:+.? 'UNB+UNOC:3+POAS2:ZZ+8435178700308:14+110513:1224+844'UNH+844+GENRAL:D:96A:UN:EAN003'B
GM+14E::9+1234+9'DTM+137:20110513:102'NAD+MS+POAS2::9'NAD+MR+POAS2KIM::9'FTX+AAI+++test'UNT+7+844'UNZ+1+844'
]]></edi>
</ms>
```

Figura 39 – XML de sortida a AS2\_OUT.

Aquest XML és enviat al procés central CCED i aquest al nostre comeDi, on podem veure com arriba el missatge. El missatge de servei d'aquest missatge arriba en format XML a la cua d'entrada del CCED AS2. El procés d'enviament d'aquest fitxer és idèntic a l'enviament de qualsevol fitxer, només varia el contingut. Al client Mendelson podem veure com el missatge de servei ha arribat correctament.

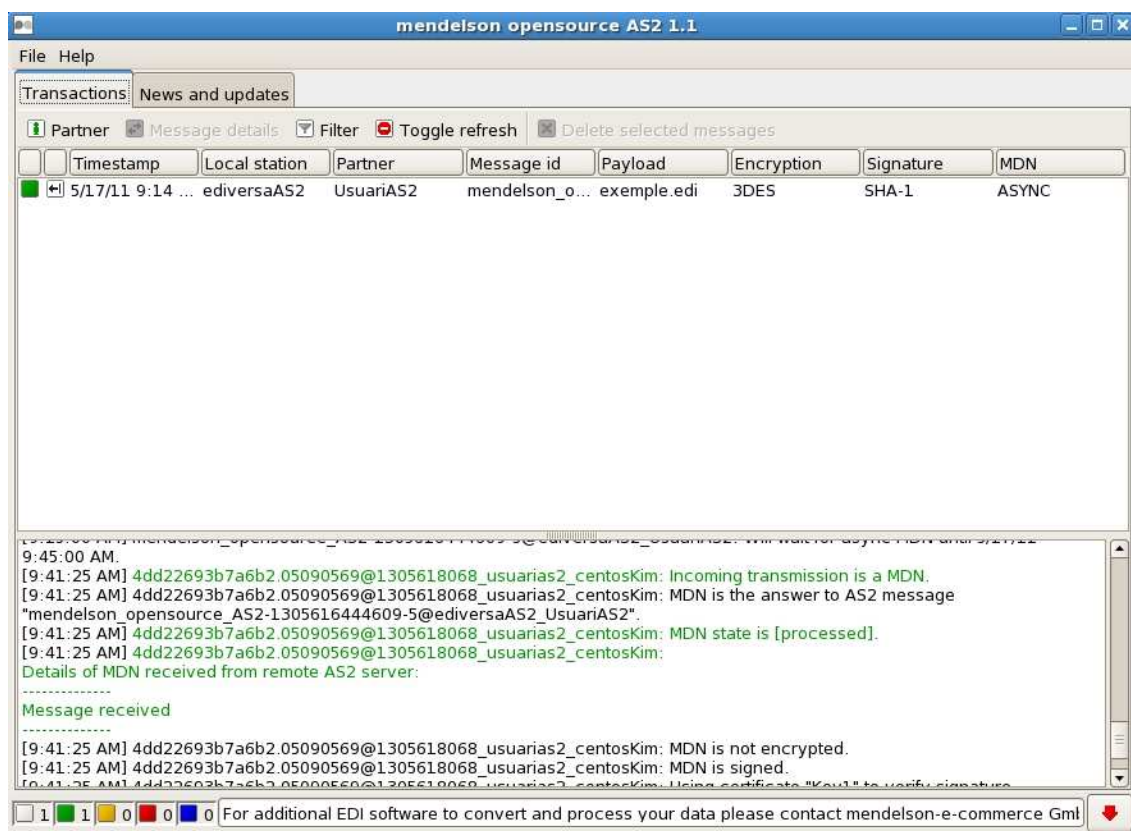


Figura 40 – Recepció del MDN asíncron a mendelson.

### 5.2.5- Recepció d'un missatge AS2 amb MDN síncron

En aquesta prova s'enviarà un missatge des de Mendelson cap a un client intern de l'empresa, configurant que es requereixi un MDN síncron. Per tant el nostre servidor AS2 ha de contestar en la mateixa connexió HTTP amb un MDN positiu.

Des de el punt de vista de l'usuari extern, representat en aquest cas pel programa Mendelson AS2, no hi ha diferència entre l'enviament síncron i l'asíncron. La diferència és com tracta aquest missatge el nostre servidor.

Un cop el servidor rep el missatge i detecta que cal respondre de manera síncrona, executa un script que comprova l'estructura del EDI que s'ha rebut. En funció de la variable de retorn d'aquest script es contesta amb un MDN positiu o negatiu.

Si el MDN és positiu el missatge es dipositarà a la cua RAW AS2 i serà tractat com un missatge asíncron, amb l'única diferència que no s'esperarà un missatge de servei procedent del procés central, ja que aquest MDN s'ha fet arribar a l'usuari AS2 abans que el missatge original arribi al destinatari.

## Disseny, implementació i implantació d'un servidor AS2

```
[2010-08-20 13:21:36] mendelson_opensource_AS2-1282303295310-47@ediversaAS2_mycompanyAS2 : (INFO) Incoming transmission is a AS2 message, raw message size: 2.74 KB.

[2010-08-20 13:21:36] mendelson_opensource_AS2-1282303295310-47@ediversaAS2_mycompanyAS2 : (INFO) SaveMessage-> FILENAME= 20100820132136209_127.0.0.1.as2.decrypted

[2010-08-20 13:21:36] mendelson_opensource_AS2-1282303295310-47@ediversaAS2_mycompanyAS2 : (INFO) SaveMessage DECRYPTED-> ../messages/_rawincoming/20100820132136209_127.0.0.1.as2.decrypted

[2010-08-20 13:21:36] mendelson_opensource_AS2-1282303295310-47@ediversaAS2_mycompanyAS2 : (INFO) SaveMessage PAYLOAD-> ../messages/_rawincoming/20100820132136209_127.0.0.1.as2.decrypted

[2010-08-20 13:21:36] mendelson_opensource_AS2-1282303295310-47@ediversaAS2_mycompanyAS2 : (INFO) AS2 message is encrypted.

[2010-08-20 13:21:36] mendelson_opensource_AS2-1282303295310-47@ediversaAS2_mycompanyAS2 : (INFO) The data has been decrypted using the key "mycompanyAS2".

[2010-08-20 13:21:36] mendelson_opensource_AS2-1282303295310-47@ediversaAS2_mycompanyAS2 : (INFO) AS2 message is signed.

[2010-08-20 13:21:40] mendelson_opensource_AS2-1282303295310-47@ediversaAS2_mycompanyAS2 : (INFO) The sender used the algorithm "sha1" to sign the message.

[2010-08-20 13:21:40] mendelson_opensource_AS2-1282303295310-47@ediversaAS2_mycompanyAS2 : (INFO) Using certificate "ediversaAS2" to verify signature.

[2010-08-20 13:21:40] mendelson_opensource_AS2-1282303295310-47@ediversaAS2_mycompanyAS2 : (INFO) Incoming transmission is a Message.

[2010-08-20 13:21:41] mendelson_opensource_AS2-1282303295310-47@ediversaAS2_mycompanyAS2 : (INFO) 1 payload(s) found in incoming transmission.

[2010-08-20 13:21:41] mendelson_opensource_AS2-1282303295310-47@ediversaAS2_mycompanyAS2 : (INFO) Payload #1 : 0 KB / "testmen.txt".

[2010-08-20 13:21:41] mendelson_opensource_AS2-1282303295310-47@ediversaAS2_mycompanyAS2 : (INFO) SaveMessage-> FILENAME= 20100820132136209_127.0.0.1.as2.payload_0

[2010-08-20 13:21:41] mendelson_opensource_AS2-1282303295310-47@ediversaAS2_mycompanyAS2 : (INFO) SaveMessage PAYLOAD-> ../messages/_rawincoming/20100820132136209_127.0.0.1.as2.payload_0

[2010-08-20 13:21:43] mendelson_opensource_AS2-1282303295310-47@ediversaAS2_mycompanyAS2 : (INFO) An AS2 SYNC MDN has been sent:
```

Figura 41 – Log events.log.

Aquesta és part del log events.log del mòdul PHP. Es pot veure com un cop rebut el missatge el descripta, comprova la signatura i després de comprovar si és correcte o no envia el MDN síncron.

### 5.2.6- Recepció de missatges erronis

En aquesta prova es força un error en rebre un missatge des de un usuari AS2 extern. Per tant el servidor AS2 ha de contestar amb un MDN negatiu, que ha de ser interpretat com a tal pel programa Mendelson.

Qualsevol missatge incorrecte provinent d'un client de l'empresa és filtrat pel procés central i mai arribaria al servidor AS2. De tota manera aquest cas es controla i en cas d'haver un problema es contesta amb un MDN indicant el problema.

## Disseny, implementació i implantació d'un servidor AS2

En el cas d'un enviament incorrecte d'un usuari extern, l'error es detecta, ja sigui al procés central o al servidor AS2, i es contesta amb un MDN negatiu.

Fent servir el client Mendelson simulem aquest error, i el servidor AS2 ens contesta amb un MDN negatiu.

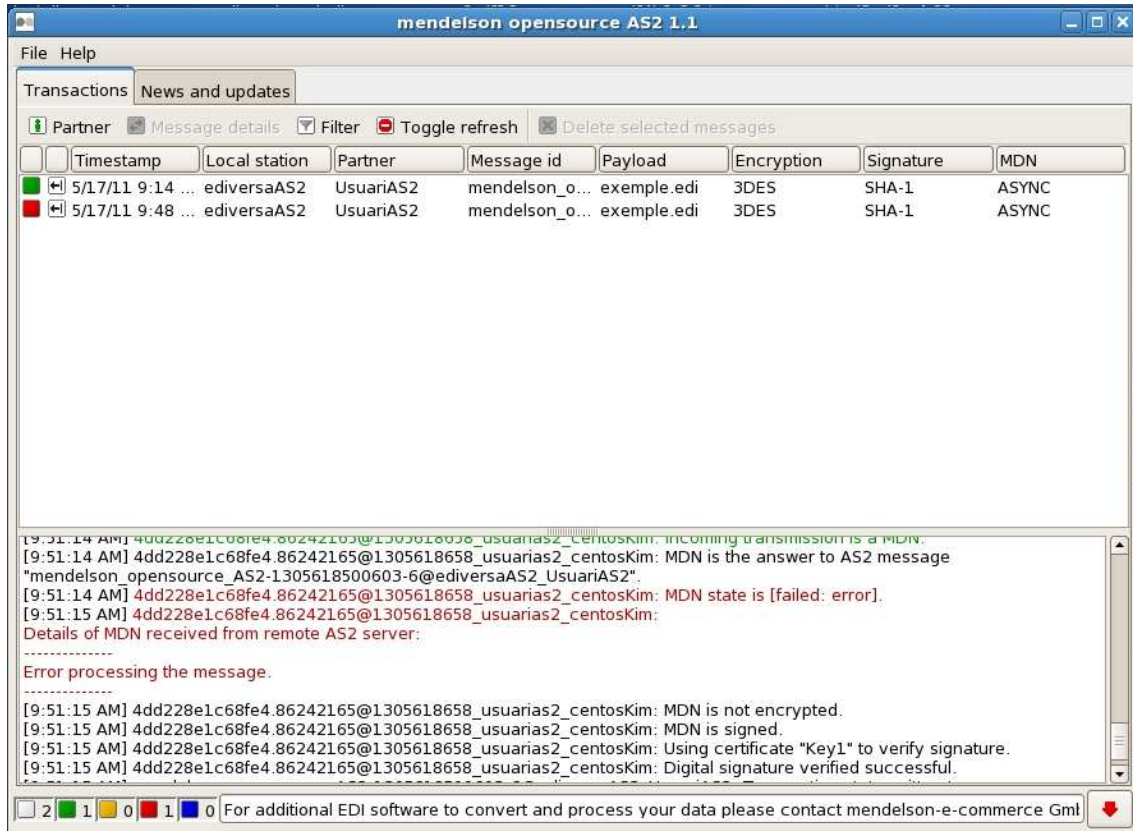


Figura 42 – Recepció del MDN negatiu a mendelson.

Es pot veure com el programa reconeix el MDN com a incorrecte, això és molt important ja que el contingut del missatge de servei no és rellevant. El que és important és una part del cos del missatge que indica que el seu estat és "failed: error".

En aquest exemple el missatge es asíncron, tot i que si fos síncron el procediment es idèntic.

### 5.3- Conclusions del procés d'avaluació

Com s'ha pogut veure el servidor ha passat correctament totes les proves realitzades, per tant es pot assegurar que el seu funcionament és correcte, ja que pot enviar i rebre tota mena de missatges AS2 i respondre adequadament amb els missatges de servei. A més, des de el punt de vista d'un client de l'empresa, tots els missatges són tractats com qualsevol altre tipus de missatge, sense saber si va a AS2 o a algun altre servidor.

Gràcies a les proves s'ha pogut comprovar també que el seu rendiment i temps d'execució és satisfactori, ja que duu a terme totes les tasques quasi instantàniament.

## Disseny, implementació i implantació d'un servidor AS2

Tot i això els companys de l'empresa m'han remarcat que la seva gestió és encara una mica massa complexa per un usuari que no és familiar amb la tecnologia AS2. Per tant més endavant hauré de millorar l'eina d'administració del servidor per tal que qualsevol persona de l'empresa pugui gestionar-lo de manera fàcil i ràpida.

## 6- ESTUDI ECONÒMIC

En aquest capítol es mostra la quantitat de temps invertida per poder desenvolupar totes les fases del projecte. A més es dona una estimació del preu del projecte basant-se en les hores emprades a cada fase de desenvolupament.

	# hores	Cost/hora	Cost total
<b>Estudi tecnològic</b>	20 hores	0 euros/hora	0 euros
<b>Anàlisi funcional</b>	60 hores	40 euros/hora	2400 euros
<b>Anàlisi tecnològic</b>	80 hores	25 euros/hora	2000 euros
<b>Programació i proves</b>	360 hores	25 euros/hora	9000 euros
<b>Memòria</b>	30 hores	0 euros/hora	0 euros

Cost total = 13400 euros

Figura 43 – Taula resum de l'estudi econòmic.

Cal tenir en compte que aquest projecte ha estat desenvolupat per l'ús intern dins de l'empresa, i per tant aquest valor de cost no ha estat pagat per cap client i és només orientatiu.

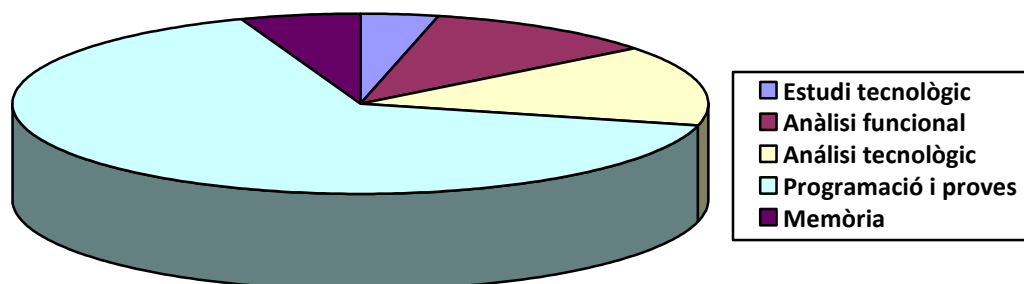


Figura 44 – Gràfic de estudi econòmic.

A la figura 43 podem observar com la major part del temps invertit ha estat en la programació dels mòduls i en l'avaluació del correcte comportament del projecte. Tot i això s'ha dedicat una bona quantitat de temps a l'anàlisi tecnològic, escollint les eines adients per poder implementar el servidor, i en l'estudi tecnològic, on s'ha après a utilitzar i comprendre aquestes tecnologies emprades.

## 7- CONCLUSIONS I LÍNIES DE FUTUR

Actualment les empreses es comuniquen entre elles fent servir missatges en format electrònic, existeixen diferents formats, un dels més emprats és el format EDI. La nostra empresa es dedica a donar suport als seus clients per fer arribar missatges en format EDI al destinatari desitjat. Aquest missatges poden enviar-se fent servir gran nombre de protocols, i l'empresa es veu forçada a donar suport a la gran majoria d'aquests. Un nou protocol que ha sorgit per enviar tot tipus de missatges, entre ells EDI, és el protocol AS2, que fa servir el protocol HTTP per enviar missatges. Amb l'aparició d'aquest nou protocol l'empresa es veu obligada a donar-ne suport, per poder oferir aquest servei als clients és necessari el desenvolupament intern d'un servidor AS2.

Aquest projecte ha abordat l'anàlisi, disseny, implementació i implantació d'aquest servidor AS2 dins de la infraestructura interna de l'empresa, per poder oferir als nostres clients un nou servei: enviar i rebre missatges de destinataris AS2. El servidor AS2 es va dissenyar dividint el procés d'enviament i recepció en dues fases, en una primera fase es feia una traducció i interpretació dels missatges provinents del servidor central de l'empresa i a una segona fase s'enviava o rebia el missatge d'un servidor AS2. Per la primera fase es va decidir fer servir el llenguatge de programació PERL, ja que és el que l'empresa fa servir per implementar tasques d'aquest tipus, per la segona fase en canvi es va fer servir PHP, ja que és un llenguatge més adient per implementar aplicacions de tipus HTTP.

El resultat del projecte ha estat molt satisfactori, ja que a la fase d'avaluació del servidor AS2 es va determinar que complia amb tots els requeriments fixats a les fases d'anàlisi i disseny i que per tant està llest per posar-lo a funcionar en l'entorn de producció de l'empresa. En el procés de desenvolupament del projecte han aparegut alguns problemes, sobretot relacionats amb les tecnologies emprades i dificultats de disseny que es van poder solucionar adequadament.

Abans de desenvolupar el servidor AS2 havia fet només petits scripts simples en PERL, en canvi per poder dur-lo a terme he hagut d'ampliar notòriament els meus coneixements sobre aquest llenguatge de programació i les eines que aporta. Vaig aconseguir implementar el mòdul de PERL amb menys problemes dels esperats, tot i tractar-se d'un llenguatge de programació nou per a mi, gràcies a que PERL es fa servir molt a l'empresa i vaig tenir força exemples i ajuda dels meus companys.

En altres projectes anteriors ja havia treballat força fent servir PHP. Tot i això fins ara només l'havia fet servir per implementar pàgines web. Per poder implementar el mòdul PHP del servidor AS2 va ser necessari ampliar els coneixements sobre les funcionalitats que el llenguatge aporta per poder tractar capçaleres i crides HTTP, i poder entendre i modificar la llibreria AS2.

Al tractar-se d'un servidor que actua com un gateway i que depèn del procés central de l'empresa, ha estat molt important coordinar les fases de disseny, programació i test amb la resta de companys del departament de sistemes, ja que per tal que els mòduls implementats poguessin ser executats i pogués anar fent tests parcials de la meva feina necessitava que el procés central estigués preparat per tractar el nou servidor AS2. Durant la fase d'avaluació es



## Disseny, implementació i implantació d'un servidor AS2

va fer que les proves fossin fetes amb col·laboració dels companys del departament de sistemes i les seves opinions sobre el funcionament del servidor han estat positives.

Com a línies de futur m'agradaria millorar la interfície web que ajuda a gestionar el servidor, ja que ara mateix és molt simple i aporta poques funcionalitats. Ara mateix només es poden modificar paràmetres sobre els partners, i només es pot fer si entens correctament el funcionament del protocol AS2. Per tant cal implementar una interfície més entenedora, amb tutorials bàsics per tal que qualsevol persona de l'empresa pugui gestionar-lo sense problemes.

Ara mateix el servidor està optimitzat per l'enviament i recepció de documents EDI, però al tractar-se d'una plataforma AS2 es poden enviar tot tipus de documents, com per exemple PDF firmats. En un futur no molt llunyà és possible que s'hagin d'implementar nous mòduls del servidor per poder atendre aquest tipus de tasques, tenint això ja en compte el disseny de la plataforma es va fer el més estructurat possible, ajudant i facilitant enormement el desenvolupament futur de nous mòduls i ampliacions. Aquestes noves tasques que pot dur a terme el servidor consistirien en poder enviar diferents tipus de fitxers, per tal de poder complir aquestes demandes caldria implementar noves subrutines al CCED AS2 que s'encarreguessin de tractar aquests nous formats i enviar-los a AS2. De la mateixa manera caldria modificar el mòdul PHP ja que ara mateix duu a terme comprovacions de validesa de missatges a nivell EDI.

## 8- BIBLIOGRAFIA

- [1] PHP GROUP (2009, 21 Maig). *PHP: Hypertext Preprocessor* [en línia]. [Consultat: 20 Agost 2010]. Disponible a Internet: <http://www.php.net/>
- [2] GRUPO DESARROLLOWEB (2009, 12 Maig). *PHP a fondo* [en línia]. [Consultat: 25 Agost 2010]. Disponible a Internet: <http://www.desarrolloweb.com/php/>
- [3] D. Moberg, R. Drummond (2005, Juliol). *MIME-Based Secure Peer-to-Peer. Business Data Interchange Using HTTP, Applicability Statement 2 (AS2)* [en línia]. [Consultat: 12 Agost 2010]. Disponible a Internet: <http://www.rfc-editor.org/rfc/rfc4130.txt>
- [4] Sebastien Mallot (2010, 23 Gener). *AS2secure Manual*. [en línia]. [Consultat : 10 Agost 2010]. Disponible a Internet: <http://www.as2secure.com/manual>
- [5] Tom Christiansen (1998, 15 Juliol). *Learning Perl*. O'Reilly.
- [6] David Sklar, David Trachtenberg (2003). *PHP Cookbook*. SURREY.
- [7] Philip Hazel (2007, 1 Abril). *The Exim SMTP Mail Server*. Independent Pub Group.
- [8] B Cher Gruppe (2006, 2 Maig). *FTP-Server*. Bertrams Print on Demand.
- [9] Charles Auds (2000, Novembre). *Linux Apache Web Server Administration*. Sybex Inc.
- [10] Kille Steve (2004, Agost). *Implementing X.400 And X.500: The Pp And Quipu Systems*. Artech Huo.
- [11] Livres Groupe (2010, 27 Juliol). *Comandes Unix*. Books LLC.