

**Gestió de Col·lectors Virtuals de Dades**  
*Administració de Web i comerç electrònic*

**Autor:** Xavier Sardà Pérez

**Consultor UOC:** Francisco Javier Noguera Otero

**Tutor empresa GTD:** Francisco Javier Varas Noriega

18/01/2013



Aquest treball està publicat sota una llicència:

**Creative Commons Attribution-ShareAlike 3.0<sup>1</sup>**

---

1 Podeu trobar els detalls de la llicència a la pàgina web: <http://creativecommons.org/licenses/by-sa/3.0/>

## Resum Executiu

Aquesta memòria correspon al projecte d'aplicació professional del Màster en Programari Lliure de la UOC en l'àmbit de l'Administració Web i Comerç Electrònic.

El document explica de forma detallada les fases seguides per a la implementació d'un sistema complet d'informació que inclou una aplicació web i un servei web per a la consulta d'informació d'un gran nombre de dispositius (el servei web implementat persegueix oferir les mateixes dades i les operacions per a la seva modificació que l'aplicació web). Aquests dispositius són elements que aporten un gran nombre de dades que l'aplicació ha de ser capaç de gestionar a través d'interfícies gràfiques o serveis web.

El projecte s'ha de situar en un entorn on l'empresa que ha desenvolupat el projecte és també responsable d'un projecte germà amb el qual es comparteix el magatzem de dades principals; aquest projecte germà té altres requeriments que imposen certes solucions com es podrà veure al llarg de la memòria però també ha permès compartir tasques i reaprofitar coneixements adquirits en un projecte per a l'altre.

Els requeriments del client en aquest projecte, tot i tenir unes funcionalitats complexes, s'han centrat en els aspectes no funcionals del sistema (rendiment, facilitat d'ús de l'aplicació web, interoperabilitat amb els seus sistemes actuals...). Això ha fet que, tot i haver-hi una gran càrrega de desenvolupament, bona part del focus del projecte s'hagi dirigit a l'estudi d'eines i components que facilitessin el satisfer aquestes necessitats.

Com s'ha comentat, tant l'aplicació com el servei web han de tenir un rendiment d'acord amb les expectatives del client i per això s'han introduït tecnologies que permeten ajustar de forma adequada el rendiment de tots els components. D'entre tots els paquets de programari utilitzats per aconseguir un rendiment acceptable destaquen el framework de desenvolupament web, GWT, la base de dades de càrrega del sistema, Cassandra, i el servidor que s'utilitzarà per al desplegament de l'aplicació web i del servei web, Jetty.

Tots aquests components es detallen amb més o menys detall dins del document junt amb el procés que s'ha seguit per al desenvolupament del projecte, lleugerament diferent als estàndards de la UOC ja que l'empresa on s'ha desenvolupat el projecte utilitza metodologies i estàndards molt concrets.

El projecte, tot i haver d'encaixar en un sistema corporatiu fet a mida persegueix construir un sistema integrat per components i utilitzar eines de programari lliure per gestionar el projecte de forma eficient i demostrar, en certa manera, la viabilitat de solucions basades en aquest tipus de programari per a entorns corporatius. Aquests entorns de vegades poden mostrar-se reticents a l'adopció d'aquestes solucions i, per aquest motiu, també es destaca en aquesta memòria la importància del programari lliure en el projecte i es detallen els components utilitzats per a futures referències.

# Índex

<b>Resum Executiu.....</b>	<b>3</b>
<b>Índex.....</b>	<b>4</b>
<b>Introducció.....</b>	<b>6</b>
L'entorn del projecte.....	6
El projecte.....	7
El programari lliure en el projecte.....	8
Estat de l'art.....	8
Estructura de la memòria.....	9
<b>Estudi de requeriments i disseny preliminar.....</b>	<b>11</b>
Requeriments funcionals.....	11
Requeriments no funcionals.....	11
Disseny preliminar.....	12
Selecció tecnològica inicial.....	13
<b>Anàlisi i disseny de la solució tècnica.....</b>	<b>16</b>
Aplicació Web.....	16
Client.....	16
Tecnologia.....	17
Servidor.....	18
Contenedor de Servlets i servidor web.....	18
Implementacions i components addicionals.....	19
Tasques programades.....	20
Servei web.....	21
Base de dades de configuració.....	21
Base de dades de càrrega.....	21
Diagrames.....	22
<b>Desenvolupament.....</b>	<b>24</b>
Utilització d'eines de control de versions distribuïdes.....	24
Implementació de proves unitàries.....	24
Utilització d'eines de suport per a la gestió del cicle de vida.....	25
Utilització d'eines d'integració contínua.....	25
<b>Implantació i desplegament.....</b>	<b>27</b>
Monitorització del consum de recursos de les implementacions.....	27
Utilització d'eines de suport per al desplegament automatitzat de servidors.....	28
Diagrames.....	29
Entorn de Desenvolupament.....	30
Entorn d'Integració.....	31
Entorn de Pre-producció i Producció.....	32

Captures de l'aplicació.....	33
<b>Manteniment.....</b>	<b>38</b>
<b>Relació de Software Lliure utilitzat en el projecte.....</b>	<b>39</b>
Java (OpenJDK).....	39
Servidor web Jetty.....	39
Eclipse 4.....	40
Google Web Toolkit (GWT).....	40
Git.....	40
Maven.....	41
Hudson.....	41
Quartz.....	42
Cassandra.....	42
MySQL.....	43
Apache Shiro.....	43
Spring JDBC.....	43
Astyanax.....	44
JUnit.....	44
<b>Conclusions i treballs a futur.....</b>	<b>45</b>
Innovació i recursos.....	45
Cassandra i el seu entorn.....	45
Programari lliure, comunitats i fundacions.....	46
Objectius assolits i millores potencials.....	47
<b>Referències bibliogràfiques.....</b>	<b>49</b>

## Introducció

El present projecte es situa dins d'un marc d'actuació corporatiu per al client pel qual s'està realitzant aquest encàrrec. Així doncs, sense una petita introducció a aquest marc seria impossible justificar algunes de les nebuloses o indefinicions que es poden donar en l'actual projecte.

## L'entorn del projecte

Tal i com s'introdueix en el paràgraf anterior, el projecte s'emmarca dins d'un pla estratègic d'actuació del client<sup>2</sup> pel que fa als seus serveis d'informació. Aquest pla d'actuació comporta la renovació de bona part del seu software de gestió i control de la seva activitat principal. L'àmplia renovació del programari que el client ha projectat ens proporciona una gran flexibilitat a l'hora de definir la nostra arquitectura però també hem de ser conscients de que hi ha serveis amb els que s'haurà d'interactuar que encara no es troben totalment definits i que no es troben sota el nostre control.

El projecte s'ha desenvolupat en una empresa anomenada *GTD Sistemas de Información*. En aquesta empresa, fundada fa 25 anys a Barcelona, el gruix principal d'activitats estan relacionades amb els sectors aeroespacial i de defensa però, com en el cas d'aquest projecte, també es realitzen projectes en altres àmbits com l'energètic o el sector públic. Amb la seu central a Barcelona, l'empresa té altres seus a Madrid i Càdis dins de l'estat i té presència a altres països europeus com França (incloent-hi la Guaiana Francesa) i Alemanya.

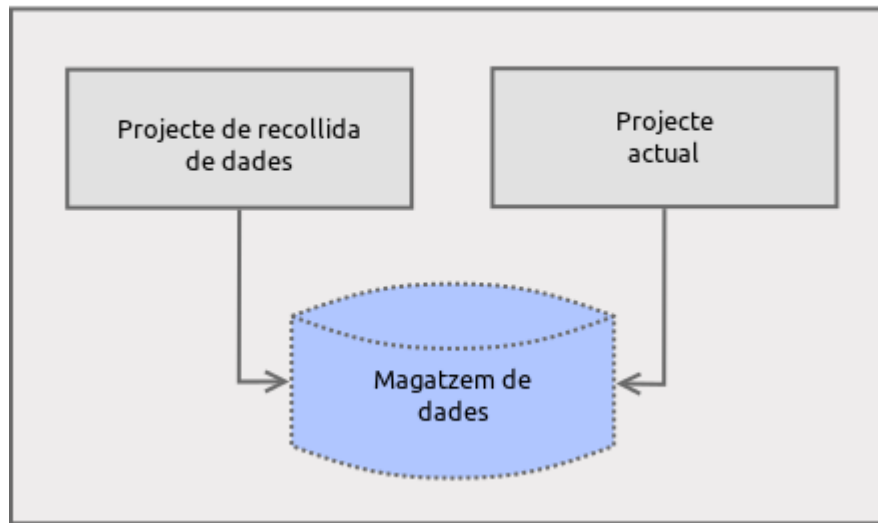
Pel que fa a l'empresa GTD, la implementació que es tracta en aquest document no és l'única que es durà a terme per a aquest client durant l'execució del projecte. GTD, a part del present projecte, també és responsable de la implementació d'un mòdul de gran impacte en el projecte que es presenta en aquesta documentació.

El projecte paral·lel (o germà) del que estem parlant tracta la recollida massiva d'informació d'un gran nombre de dispositius que en aquests moments escapen al sistema de control del client. Donat el gran nombre de dispositius i el nombre de comunicacions que aquests requeriran, el sistema de control capaç de connectar-se amb ells imposarà certes condicions pel que fa a l'emmagatzematge de dades i els serveis amb els que comunicar-se. Per posar un parell d'exemples: En l'àmbit dels magatzems de dades haurem de recollir informació d'una base de dades concebuda per a tractar grans volums d'informació. En l'àmbit de les implementacions d'interfícies gràfiques seran necessàries funcionalitats per a la gestió de serveis que normalment escapen de l'àmbit de les aplicacions web corporatives.

---

<sup>2</sup> Per les polítiques internes de l'empresa *GTD Sistemas de Información* (empresa on s'ha desenvolupat el projecte), el nom del client ha de mantenir-se a l'anonimat i, per tant, encara que resulti repetitiu, ens referirem a ell durant tot el document com "el client".

Projectes assignats a GTD



## El projecte

Tornant al projecte que estem tractant directament, aquest pretén oferir mitjançant una aplicació web i la implementació d'un servei web (SOAP) la possibilitat de gestionar la informació i el comportament del software encarregat de la gestió dels dispositius que ara es troben fora de l'abast del sistema de gestió del client.

Les funcionalitats d'aquest paquet de programari es defineixen en funció de les operacions concretes que siguin possibles realitzar amb els dispositius que actualment no es poden comunicar de forma remota amb el sistema (les funcionalitats no depenen directament dels dispositius que sí es connecten actualment al sistema tot i que el client desitja que les funcionalitats disponibles siguin el més similars possibles entre tots els seus dispositius). Aquests requeriments funcionals, a l'inici del projecte, no poden ser definits amb total claredat donat l'estat canviant de l'entorn del projecte (explicat a l'apartat anterior) i, per aquest motiu, és necessari plantejar un sistema el més modular possible.

A part d'aquest requeriment intrínsec donat per l'estat dels serveis d'informació del client també cal contemplar altres preocupacions expressades pel client que no tenen relació directa amb els requeriments funcionals.

Les preocupacions més importants expressades per part del client han estat les següents:

- Controlar el cost de la plataforma resultant en operació.
- Controlar el cost de l'escalabilitat del sistema.

En la primera d'aquestes qüestions cal tenir en compte que la utilització de software lliure pot reduir de manera considerable el cost de la plataforma en comparació amb plataformes de software privatiu que utilitzen en altres departaments de l'empresa.

En el cas del control del cost de l'escalabilitat del sistema, un dels aspectes que es tindrà

més en compte serà la tria dels components que millor facilitin la migració a maquinari més potent i la portabilitat a entorns descentralitzats.

## El programari lliure en el projecte

Tal i com començàvem a introduir en el punt anterior, el programari lliure és un dels punts on més focus es posarà en el projecte. La limitació en el cost de les llicències i la portabilitat que ens proporciona segons el tipus de programari són els millors avals per tal de satisfer les expectatives del client i dotar a l'empresa de competències en paquets de software lliure o de codi obert de cara al futur.

Per posar un exemple concret<sup>3</sup>, la implementació del costat del servidor de l'aplicació web es realitzarà amb el llenguatge de programació **Java** sobre la màquina virtual d'OpenJDK. Oracle, tot i haver generat bastanta polèmica amb la compra de Sun Microsystems, va establir com a implementació de referència de la versió 7 de Java la implementació oberta de l'OpenJDK. Aquesta tecnologia ens permet executar el mateix programari tant en una màquina d'una sola CPU, 1 GigaByte de RAM i una unitat d'emmagatzematge S-ATA de 5400 rpm com en un servidor de gran capacitat amb 4 CPUs de diversos nuclis, 16 GigaBytes de RAM i unitats de discs durs configurats en RAID.

## Estat de l'art

Les funcionalitats que es volen implementar en aquest projecte, i sobretot el còm es volen implementar, tenen diversos factors que situen el projecte en una posició avantguardista respecte a l'estat de l'art en el món de la programació web (dins d'entorns corporatius).

Per un costat, la implementació de RIAs (Rich Internet Application), tot i ser un estàndard assumit pels usuaris de xarxes socials i altres serveis web avançats, no es un àmbit en el que les tecnologies de la informació s'hagin prodigat en els entorns corporatius. El caire conservador de les polítiques dels departaments de TI de les empreses i la relativa joventut de les eines amb les que es solen implementar aquest tipus d'aplicació allunya moltes vegades els usuaris corporatius dels beneficis que aquest tipus d'aplicació pot aportar a les interfícies gràfiques i a la interacció.

Per l'altre costat, tot i ser **Apache Cassandra**<sup>4</sup> un motor de persistència amb el suport d'alguns socis importants com ara Facebook o la Apache Software Foundation i ser NoSQL un àmbit en el que moltes empreses estan confiant el seu magatzem de dades, sembla que a les empreses "tradicionals" (o no relacionades directament amb Internet) encara són reticents a fer el canvi de mentalitat que hi ha al darrera d'aquest moviment que s'allunya dels sistemes gestors de bases de dades relacionals.

---

3 La relació completa del programari lliure o de codi obert que s'utilitza en aquest projecte es pot trobar a l'apartat previ a les conclusions d'aquest mateix document, en ell es llista tots els elements de software lliure utilitzats, una breu introducció a cadascun d'ells, la seva funcionalitat al projecte i la llicència sota la qual s'utilitzen.

4 Es pot trobar informació de la base de dades no relacional Cassandra a la secció on es detalla el programari de codi obert utilitzat en el projecte.



Tot això, en un projecte on es pretén donar una solució el més econòmica, mantenible, escalable i fiable a un sistema que ha de gestionar un volum de dades que es podria qualificar de Big Data pel volum de dades generat i pel curt espai de temps en el que es vol processar aquesta informació, situa el projecte com a referència en la consolidació de les tecnologies comentades en un entorn on fins ara no hi tenien cabuda. Es tracta doncs d'un repte per al client perquè ha de ser capaç d'adoptar aquestes tecnologies i qualificar el seu personal per a utilitzar i mantenir el programari que es generi en el projecte i, un altre repte per a GTD a l'hora de generar el coneixement necessari per aportar una implementació estable amb els components que s'ha decidit incorporar.

## Estructura de la memòria

El projecte no segueix una metodologia estàndard per al desenvolupament d'aplicacions convencionals o web sinó que segueix una metodologia basada en els estàndards de l'Agència Espacial Europea (ESA) i el Nivell 3 del CMMI<sup>5</sup>.

S'ha adoptat aquesta metodologia ja que és normativa interna de l'empresa i el client d'aquest projecte va expressar el seu interès per la maduresa en la gestió que demostrava el seguiment d'estàndards tant avançats com els mencionats anteriorment.

L'adaptació d'aquesta metodologia al desenvolupament del projecte es pot consultar a l'annex I d'aquest document<sup>6</sup> però destacarem que, en relació a la metodologia proposada des de la Universitat, el canvi més important és l'eliminació de l'estudi de viabilitat en favor d'un procés d'estudi de requisits i disseny preliminar.

D'una banda aquest canvi es justifica donat que en el moment que es va presentar l'oferta comercial que ha generat aquest encàrrec ja es va realitzar un anàlisi de viabilitat previ en certa profunditat. De l'altra, també es considera més adequat el realitzar un procés de disseny amb un parell d'iteracions amb visibilitat per part del client (d'aquesta manera es poden ajustar les expectatives de tots els *stakeholders*) que no pas un procés més opac d'anàlisi i disseny que donaria per resultat un disseny poc flexible després d'un procés llarg i detallat. La primera opció, proposar un primer acostament amb visibilitat per al client, ens facilita la involucració del client en el disseny, construcció, comprensió del sistema i presa de decisions.

Explicada la metodologia del projecte ja podem passar a explicar l'estructura de la memòria:

- A l'inici d'aquest document podem trobar el resum executiu del mateix. En aquest resum podem extreure una idea general del contingut del document i el projecte. Abans del resum executiu podem trobar la llicència sota la qual es publica aquesta documentació i la taula de versions del document.
- A continuació podem trobar l'Índex del document i aquest breu apartat introductori.

---

5 El nivell tres del CMMI certifica que els processos de desenvolupament de l'empresa estan definits i que aquests són respectats en els projectes desenvolupats a l'empresa. Es pot trobar més informació sobre el CMMI a les fonts relacionades amb el CMMI a l'apartat de referències bibliogràfiques.

6 Aquest document es troba a la mateixa carpeta que aquesta memòria, a l'arxiu *TFM\_PAC4\_ANNEXI.pdf* i és el resultat de la Prova d'Avaluació Continuada 1 de l'assignatura de Treball de Final de Màster.

- Tot seguit trobem els apartats d'Estudi de requeriments i disseny preliminar i d'Anàlisi i disseny de la solució tècnica. El primer d'aquests dos apartats és el que reflexa el treball fet en lloc de l'estudi de viabilitat, provocat pel canvi principal en la metodologia que comentàvem en la secció anterior. El segon és el apartat on es consoliden les decisions tècniques i es concreten els dissenys dels components a implementar.
- Un cop completats aquest capítols trobarem els apartats destinats a desenvolupament, implantació i desplegament i manteniment. La informació continguda en aquests apartats fan referència a eines utilitzades i qüestions generals sobre l'àmbit de les seccions excepte l'apartat sobre manteniment. L'apartat sobre el manteniment no està massa desenvolupat per no haver-se iniciat el procés associat al manteniment en el moment d'entrega d'aquesta documentació. Tot i així, a la secció destinada a aquest àmbit s'han descrit les accions realitzades per a facilitar el treball de l'equip que s'encarregui de les tasques de manteniment en el futur.
- Al final del cos principal del document ens trobarem les conclusions extretes del projecte i, prèviament, un llistat dels components de programari lliure o de codi obert utilitzats en el present projecte.
- Per tancar aquesta documentació podrem trobar la secció de Referències bibliogràfiques.

## Estudi de requeriments i disseny preliminar

Els requeriments del client en aquest projecte es van capturar a la fase d'oferta. Aquests requeriments, van lligats amb el projecte paral·lel que es dedica a la comunicació amb els dispositius dels quals es gestiona la informació en el projecte objecte d'aquest document.

Independentment de que els requeriments es recollissin en una fase principalment comercial, s'han reescrit per tal d'acostar-los a l'equip responsable de la implementació del projecte.

### Requeriments funcionals

Gestionar el nombre de col·lectors virtuals dels dispositius a gestionar: Aquests col·lectors són els homòlegs de dispositius físics que l'empresa té repartits pel territori per tal de controlar grups de dispositius agrupats geogràficament. Per fer el més realista possible l'emulació d'aquests equips s'ha de proporcionar un sistema el més fàcil possible per a que el client creï el nombre de col·lectors virtuals que més li convingui per integrar en el seu sistema actual.

Gestió dels dispositius assignats als col·lectors: El sistema ha d'oferir una altra interfície gràfica per tal de definir quins dispositius han d'agrupar-se sota quins col·lectors. Al tractar-se d'una tasca amb un volum de dades bastant gran es possible que s'ofereixi al client una interfície el més adequada possible per al tractament d'aquest volum de dades.

Gestió de les tasques automatitzades: El sistema, com a part intrínseca de la gestió de la informació dels dispositius, incorpora un sistema de planificació de tasques. Aquestes tasques s'han de poder consultar i modificar des del sistema plantejat

Gestió dels usuaris del sistema: El sistema ha de permetre gestionar els usuaris del sistema i, probablement, restringir l'accés dels usuaris a certes funcionalitats segons el seu perfil.

### Requeriments no funcionals

El sistema ha de poder configurar-se en alta disponibilitat: Al tractar-se d'un sistema amb un volum de dades molt elevat en comparació amb la xarxa de dispositius físics que té l'empresa, en el moment que entri en producció serà un actiu molt important i aquest ha de poder-se configurar de forma que s'asseguri la seva disponibilitat per sobre de tot.

La usabilitat del sistema ha de ser elevada: Comptant que es tracta d'un programari que ha de ser l'eina de treball principal per algunes persones de la organització del client, s'ha remarcat diverses vegades que les interfícies d'usuari del sistema han de ser el més agradables i senzilles a la vista per evitar fatigues innecessàries.

El sistema ha de mostrar un tauler de control amb la informació més rellevant del sistema de forma compacta: L'aplicació ha de poder mostrar la informació més important en un tauler de control (*dashboard*) on es puguin identificar ràpidament els trets més importants de l'estat del sistema.

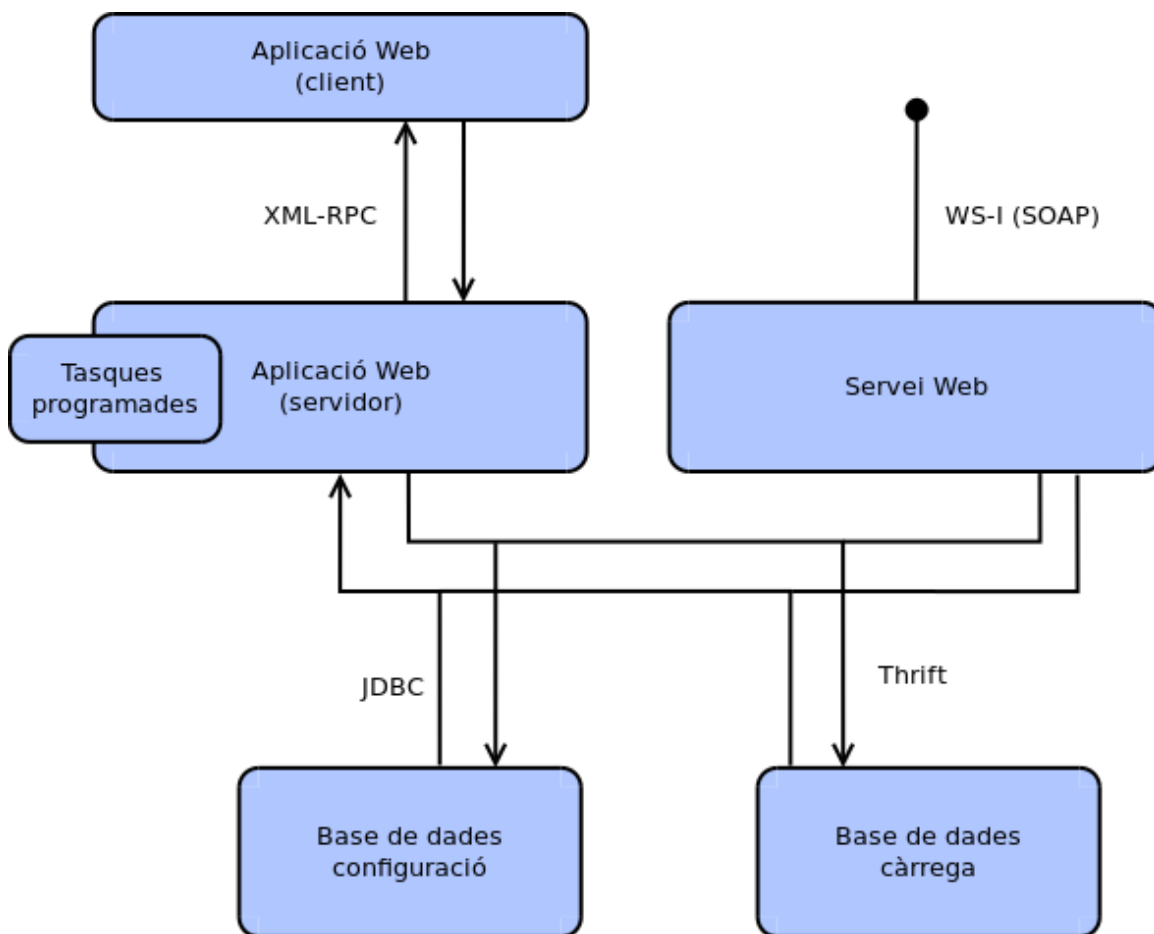
Es podria considerar aquest requeriment com una extensió de l'anterior però la insistència del client en aquest aspecte ha provocat que se'n definís un de independent per a fer el seguiment d'aquesta característica del sistema.

El sistema d'identificació d'usuaris s'ha de poder integrar amb el sistema corporatiu del client: Un dels principals objectius del client és integrar aquest nou sistema de la forma més transparent possible amb el seu sistema actual. S'ha de tenir en compte que el sistema que el client té actualment en producció és un sistema fet a mida per diverses companyies i que la comptabilitat amb d'altres sistemes no està garantida.

Per aquest motiu s'ha de procurar que el sistema d'identificació d'usuaris sigui el més flexible possible.

## Disseny preliminar

Donats els requeriments capturats per part de l'usuari podem definir un disseny preliminar que el que es mostra en el següent diagrama:



## Components

- **Aplicació Web (client):** Haurà de ser compatible amb els navegadors majoritaris i proporcionar una interfície el més usable possible.
- **Aplicació Web (servidor):** Encarregat de satisfer les peticions del client per a mostrar les dades del sistema. Aquesta informació pot provenir tant de la base de dades de configuració com de la base de dades de càrrega.
- **Tasques programades:** El sistema ha d'incloure algun component que sigui capaç de disparar i gestionar tasques periòdicament.
- **Servei web:** El servei web es construirà per satisfer les peticions del sistema de gestió actual del client. L'objectiu d'aquest servei és oferir una API amb la qual el client pugui integrar el nou sistema sense modificar cap dels dos sistemes (estrictament s'haurà de modificar el seu sistema actual però s'intenta que aquests canvis es puguin aplicar mitjançant complements o proxies que no comportin canvis en profunditat de les seves aplicacions).
- **Base de dades de configuració:** Aquesta base de dades es destinarà a emmagatzemar la configuració del sistema i altres dades que pugui ser necessàries per al sistema.
- **Base de dades de càrrega:** Aquesta base de dades no sorgeix de l'estudi dels requeriments sinó que es tracta d'una exigència provinent dels projectes externes que s'estan duent a terme.

## *Selecció tecnològica inicial*

Per satisfer tots els blocs definits en el disseny preliminar s'ha realitzat una selecció prèvia de tecnologies per tal de fixar les bases del projecte. Aquestes eines s'han triat seguint els requisits del projecte de costos, fiabilitat, mantenibilitat i escalabilitat.

### **Aplicació web (client)**

L'aplicació client ha de permetre una interacció dinàmica amb l'usuari i, alhora, oferir una capacitat alta per evolucionar les funcionalitats oferides. A més, s'ha de tenir en compte que el consum de recursos del servidor ha de ser el més reduït possible.

Amb aquests paràmetres s'han descartat alternatives basades en servidors d'aplicacions molt complets com podria ser JBoss en favor de servidors més simples com Apache Tomcat o Jetty. Les alternatives descartades per aquest motiu han estat els frameworks Seam i Struts.

Pels mateixos paràmetres s'han avaluat GWT, Vaadin (basat en l'anterior), Grails i Adobe Flex. Donada la exigència de donar suport per a diferents plataformes (sistemes operatius i navegadors) i la dificultat per mantenir l'aplicació s'ha decidit descartar Flex. Grails ha estat descartat per una manca de coneixement de l'equip de desenvolupament i per estar per

sota de l'escollit finalment en quant a documentació i volum de la comunitat<sup>7</sup> que hi ha al darrere.

La tecnologia escollida per a aquest component ha estat doncs GWT (davant de Vaadin per no estar aquest recolzat per una gran empresa com sí ho està GWT) pels coneixements previs de l'equip de desenvolupament, l'alt grau d'integració amb eines com els IDEs o eines de suport a la gestió del cicle de vida del software, la documentació disponible i la gran comunitat de desenvolupadors que estan utilitzant actualment aquest framework.

### **Aplicació web (servidor)**

La tecnologia a nivell de servidor estava fixada per motius de plataforma i coneixements de l'equip de desenvolupament a Java. Java ens ofereix una gran flexibilitat a l'hora de desenvolupar el sistema (es pot treballar en diferents arquitectures i en diferents sistemes operatius sense que això afecti al projecte) i una integració total amb les eines de gestió que s'utilitzen al projecte.

En principi els mecanismes de comunicació amb l'aplicació client que ofereix la tecnologia triada en el punt anterior semblen suficients per a complir amb els requisits del projecte.

Afegir que, donat l'anunci de la finalització del suport a la versió 6 de Java per part d'Oracle (amb excepció d'aquells que tinguin una llicència comercial), s'ha decidit apostar per la implementació de referència de la versió 7 de Java (recordem que Oracle va definir l'OpenJDK com a implementació de referència per a aquesta versió de Java). El projecte, donat que utilitza el sistema de comunicació entre client i servidor estàndard de GWT, utilitzarà un contenidor de servlets per a donar resposta a les peticions del client. Aquest servidor de servlets serà Jetty perquè ofereix un bon rendiment tant en el processament de les peticions dinàmiques com en l'entrega de contingut estàtic. L'eficiència de Jetty en aquestes dues tasques ens permet estalviar-nos una capa de servidors HTTP per a continguts estàtics com podria ser l'Apache Web Server

### **Tasques programades**

La selecció tecnològica inicial d'aquest component ha estat influenciada per la tecnologia destinada triada per a la part de servidor de l'aplicació web. Al ser Java la tecnologia escollida per a implementar la part de servidor de l'aplicació web s'han filtrat components tecnològics que utilitzessin o fossin compatibles amb aquest entorn d'execució.

D'entre totes les opcions que es van començar a avaluar, el més madur i que més possibilitats oferia en quant a integració, API i mantenibilitat era Quartz. Aquest framework ens ofereix un sistema clusteritzable, tolerant a fallades i integrable amb els principals servidors web i d'aplicació que existeixen a l'entorn de Java.

### **Servei web**

El servei web també és un requeriment extern del client per tal de poder integrar el sistema desenvolupat en aquest projecte amb el seu sistema de gestió. El servei web a

---

7 Farem una reflexió en profunditat sobre la comunitat que hi ha al darrere de GWT a l'apartat del Conclusions ja que aquest ha estat definitivament un punt decisiu en l'elecció del framework de desenvolupament de l'aplicació web.

desenvolupar és un servei web SOAP i, per a la seva implementació s'ha triat en aquesta fase el framework JAX-WS per la seva gran integració amb altres eines que s'utilitzen a l'empresa.

Actualment, l'empresa no té cap sistema equivalent al que s'està desenvolupant (té una xarxa de dispositius físics repartits pel territori però no tenen un sistema capaç d'emular aquesta xarxa de forma centralitzada en el seu centre de processament). Per aquest motiu la definició del servei web corre a càrrec de l'empresa GTD i seran els sistemes de gestió del client els que s'hauran d'adaptar a aquest un cop el sistema estigui implantat. Cal destacar que la possibilitat de gestionar les funcionalitats que implementa el projecte de forma programàtica va ser una de les inquietuds que el client va expressar en les fases inicials del projecte.

### **Base de dades de configuració**

En tractar-se d'una base de dades que ha de donar suport a la configuració del sistema i que haurà d'emmagatzemar un volum de dades relatiu, s'ha triat MySQL com motor per a aquesta funcionalitat.

MySQL té una quantitat de documentació disponible de forma pública a Internet i hi ha tants desenvolupadors que donen suport a aquest sistema gestor de base de dades que, al no tenir cap requeriment especial fa complicat posar en dubte aquesta elecció.

### **Base de dades de càrrega**

La base de dades de càrrega no és un component que es pugui definir des d'aquest projecte sinó que ens ve donat per el projecte que tracta la comunicació amb els dispositius. Aquest projecte paral·lel necessita una gran capacitat d'entrada de dades i ha escollit Apache Cassandra per solucionar aquest necessitat.

És un factor de risc el treballar amb aquest tipus d'emmagatzematge però també és un dels valors afegits del projecte pel que suposa per al client adoptar aquest tipus de tecnologia i per a GTD per a generar coneixement sobre Cassandra dins de l'equip del projecte.

## Anàlisi i disseny de la solució tècnica

Un cop validada la captació i interpretació dels requeriments i la fase de disseny preliminar amb el client, arriba el moment de realitzar l'anàlisi en profunditat de la plataforma que es vol construir i definir un disseny per implementar el sistema en qüestió.

Aquest mecanisme iteratiu que permet una major transparència amb el client a nivell d'anàlisi i disseny és molt beneficiosa ja que permet alinear els objectius del client amb els de l'equip del projecte en un estadi molt embrionari del projecte.

Com a resultat de la validació del client de les idees expressades en el capítol anterior, a continuació es detalla l'anàlisi detallat de components i un possible disseny per a la seva implementació.

### Aplicació Web

L'aplicació web, tal i com comentàvem a l'inici, ha de poder gestionar diversos aspectes del sistema.

D'una banda és important que l'aplicació pugui satisfer els requeriments funcionals (gestionar col·lectors d'informació, assignació de dispositius a aquests col·lectors, gestió de tasques programades i gestió d'usuaris).

De l'altra, també hem d'aconseguir que la sensació de l'usuari davant de l'aplicació web sigui el més satisfactòria possible. En aquesta línia, a la fase de disseny preliminar, ja es van triar les eines necessàries per a facilitar la comunicació asíncrona entre el client i el servidor (dins l'aplicació web) però en aquesta fase entrarem més en detall en aquest punt i en com aconseguir-ho.

#### *Client*

L'aplicació web, a nivell de client/navegador, és el punt més visual del projecte i que requereix més treball a nivell gràfic. L'objectiu des d'un bon principi es proporcionar al client un entorn el més usable possible i amb un disseny agradable per als usuaris.

Tot i aquesta "imposició" sobre la interfície gràfica del projecte descriurem un a un els requeriments funcionals del projecte per a assegurar-nos de que tots queden satisfets des de l'anàlisi i el disseny del sistema.

#### Gestió de col·lectors

Els gestors de col·lectors requerirà un disseny del model de dades i la implementació d'operacions senzilles de creació, consulta, modificació i esborrat de col·lectors (operacions **Create Read Update Delete** en Anglès).

L'accés a la informació dels col·lectors i les seves operacions disponibles ha de ser en un format el més fàcil de reconèixer i sense una gran quantitat de passos entre les operacions de consulta i les de modificació i les de creació, modificació o esborrat.



### Assignació de dispositius a col·lectors

Les operacions que s'han de fer sobre els dispositius són les d'assignació d'un dispositiu a un col·lector, la des-assignació, el canvi d'assignació d'un col·lector a un altre.

En si les operacions no són complicades d'implementar però s'ha de tenir en compte que la quantitat de dades que s'han de manejar en aquesta secció són de l'ordre dels centenars de milers. Per aquest motiu és necessari no només implementar les funcionalitats requerides sinó acordar amb el client la interfície més adequada per a gestionar aquest volum de dades amb la màxima eficiència possible.

### Gestió de tasques programades

Les tasques programades són un altre exemple d'operacions CRUD a implementar. L'única complicació d'aquest apartat és accedir a les dades d'aquestes operacions. Per a la implementació de les tasques programades s'utilitzarà un framework amb el que s'haurà d'establir la connexió adequada per a implementar les operacions comentades.

El volum de dades en aquest paquet no hauria de ser un problema, per tant, les preocupacions expressades en la documentació referida al requeriment de l'assignació de dispositius a col·lector no apliquen en aquest requeriment.

### Gestió d'usuaris

La gestió d'usuaris d'aquest projecte només té la complicació implícita d'haver de gestionar els permisos per als diferents perfils d'usuaris que utilitzaran l'aplicació. Per tant, en quant a la implementació, s'haurà de cuidar la creació, modificació, consulta, eliminació i actualització dels rols d'usuaris.

## **Tecnologia**

A nivell tecnològic, la part client de l'aplicació web, s'implementarà en GWT però amb alguns afegits necessaris per complir totes les expectatives.

GWT no ofereix un procediment senzill per a mostrar un tauler de control prou flexible com el que aquest projecte demanda. A més, s'ha de tenir en compte que un dels millors formats amb el que es pot mostrar més quantitat d'informació fàcilment identificable per a usuaris finals són gràfics i aquest també és un camp on GWT no proveeix una solució prou avançada.

Pels motius exposats en el paràgraf anterior s'ha decidit escollir una llibreria addicional per a satisfer totes dues exigències. Entre totes les opcions disponibles s'han triat GXT de Sencha<sup>8</sup> i SmartGWT d'Isomorphic Software<sup>9</sup>.

GXT proveeix un gran ventall d'extensions sobre GWT però amb un model de llicències dual comercial-GPL. És la llibreria que millor soluciona les necessitats del projecte però s'hauria d'adaptar el model de llicència de tot el projecte per utilitzar la versió GPL o bé adquirir una llicència comercial per a aquest software.

---

8 Pàgina web de Sencha: <http://www.sencha.com/>

9 Pàgina web d'Isomorphic Software: <http://www.smartclient.com/>

SmartGWT és una segona opció que ens permet la implementació de taulers de control molt similars als que ens ofereix el programari de Sencha però la part gràfica no està tant ben treballada com en el cas de GXT. Un punt a favor d'SmartGWT és que la versió de programari lliure dels seus productes és LGPL, amb lo qual el model de llicències del projecte no hauria d'alterar-se per a aquest framework.

Amb tot, a nivell tècnic és decideix que la utilització de GXT, propietat de Sencha, és la més beneficiosa per al projecte i qualsevol decisió administrativa en contra d'aquest fet provocaria la revisió de la planificació per l'augment de complexitat en quant al desenvolupament.

## *Servidor*

La implementació del component relatiu al servidor te dues línies d'actuació. Per un costat tenim la tasca de servir els continguts de l'aplicació web (HTML, JS, imatges, animacions...) i per l'altra la implementació dels procediments que l'aplicació client cridarà de forma asíncrona.

A continuació es detallen les decisions més importants en ambdós àmbits ja que cadascun ha estat objecte d'estudi en profunditat.

## **Contenedor de Servlets i servidor web**

Pel que fa a la responsabilitat de servir els continguts, en molts projectes, no es quelcom massa rellevant (moltes vegades es tria Apache com a servidor HTTP o Tomcat com a contenidor de servlets per defecte) i, en aquest cas, és important reflexionar una mica sobre aquest component perquè en funció del seu rendiment l'usuari final pot acabar tenint la sensació de que l'aplicació no és tot el fluïda que hauria de ser. En el nostre cas d'estudi és important servir ràpidament els continguts estàtics (aquí s'inclouen els arxius JavaScript) però també és important poder processar eficientment les peticions asíncrones amb el servidor.

La major part de la comunicació entre el client i el servidor s'estima que es realitzarà mitjançant crides asíncrones però alguna d'aquestes crides pot "bloquejar" l'aplicació fins a la tornada per evitar concurrències indesitjables. Per aquest motiu l'eficiència en el processament d'aquestes peticions també és important.

Per mantenir l'equilibri entre totes dues funcionalitats s'ha identificat el servidor HTTP i contenidor de Servlets Jetty (projecte sota l'empara de la Eclipse Foundation des de la versió 7). Jetty és un servidor web i contenidor de servlets eficient que requereix molts pocs recursos per a funcionar, de fet, s'utilitza moltes vegades en software embarcat (l'entorn de programació d'Eclipse l'utilitza per alguns tests sobre servidors de servlets) però també serveix per a grans desplegaments de software com ho demostra el fet de que és el servidor web amb el que Google desplega la seva oferta de PaaS<sup>10</sup>, *App Engine*.

---

10 PaaS és un acrònim en Anglès que vol dir Platform as a Service, un concepte associat al Cloud Computing, tot i que un nivell per sobre del IaaS (Infrastructure as a Service) i un per sota del SaaS (Software as a Service).

Un cop fixada la tecnologia que s'anava a utilitzar al projecte, Java, va ser complicat identificar alternatives vàlides per a la implementació de la part servidora de l'aplicació. Abans de detectar la limitació tecnològica que va imposar GTD (i que va quedar reflectida a l'oferta comercial) s'havien identificat diverses alternatives com Python o el framework de desenvolupament Rails de Ruby. Totes dues opcions eren bastant atractives, tenint un petit avantatge Python per la major maduresa del llenguatge i les eines amb suport disponible per a aquest. D'altres opcions més lleugeres i amb més potencial per a la gestió d'un gran nombre de crides concurrents com Node.js van ser descartades pel grau de coneixement de l'equip de desenvolupament sobre aquesta tecnologia i per no estar dins de les tecnologies que l'empresa pren en consideració per als projectes.

## Implementacions i components addicionals

La implementació dels processos que el client cridarà de forma asíncrona és un altre tema a detallar. Aquests processos han de realitzar tasques tant diferents com comunicar-se amb el framework de tasques programades, inserir registres a la base de dades de configuració o recuperar dades de la base de dades de càrrega.

Per cadascuna d'aquestes tasques identificades és important crear uns procediments el més estables, mantenibles i eficients possible. Per aquest motiu s'ha intentat recórrer a components estables. La llista definida a continuació detalla els components interns necessaris per a la implementació de les funcionalitats detectades a la presa de requeriments i una tria tecnològica basada en els criteris exposats en aquest apartat<sup>11</sup>:

- **Connexió amb el component de tasques programades:** Per a realitzar aquesta tasca s'ha optat per connectar directament amb la API del framework que s'utilitzarà. Ho detallarem més endavant però el programari escollit per a aquest component és **Quartz** i s'utilitzaran les **llibreries del mateix paquet** per tal d'accedir i modificar les dades i el comportament d'aquest.  
L'alternativa a aquest mètode era recuperar la informació de Quartz llegint i modificant directament les configuracions de la base de dades desplegada per a Quartz. Aquest procediment s'ha estimat que requereix un coneixement en profunditat del model de dades del paquet de programari i que, per aquest motiu, queda completament desaconsellat des del punt de vista tècnic.
- **Connexió amb la base de dades de configuració:** Per accedir i modificar les dades de configuració serà necessari fer consultes contra una base de dades relacional. L'accés a bases de dades és un dels àmbits en que més hi ha per a escollir però, de la mateixa manera com ens hem preocupat d'escollir el millor servidor web, amb l'accés a la base de dades farem exactament el mateix.  
L'accés a la base de dades des de Java té un protagonista clar: Hibernate. Aquest framework proporciona una implementació de la JPA i és utilitzant en molts projectes Java. Tot i així, per l'experiència que GTD té en projectes anteriors amb Hibernate, s'ha arribat a la conclusió que pot arribar a donar problemes de rendiment

---

11 Les referències al programari lliure utilitzat en aquest projecte s'han agrupat en una secció independent per tal de centralitzar la informació d'aquests.

(realitza de forma automàtica massa consultes per a completar els objectes Java). Com el rendiment és un dels aspectes que el client ha destacat en tot el procés s'ha descartat Hibernate com a framework per a facilitar l'accés a la base de dades i s'ha optat per una opció més simple com és **Spring JDBC**.

Spring JDBC ens proporciona un accés senzill a la base de dades amb la llibertat per a dissenyar les consultes manualment i un rendiment més que acceptable.

- **Connexió amb la base de dades de càrrega:** Treballar amb tecnologies relativament noves té un handicap important, i és que el suport per a comunicar-se amb elles acostuma a ser bastant experimental. El cas de Cassandra no és una excepció i les llibreries i clients per a connectar-s'hi encara es troben en estadis bastant embrionaris. Tot i així, s'ha trobat un client que permet establir connexions a clústers de Cassandra amb una fiabilitat bastant elevada. El client del que estem parlant es **Astyanax** i és un projecte sorgit de l'empresa nord-americana Netflix. Aquest projecte té el seu origen en un projecte de software lliure impulsat de forma personal com és **Hector**, un altre dels clients avaluats per a cobrir aquesta necessitat. Hector és un bon client de Cassandra però amb un ritme d'evolució menor (és normal, es tracta d'un projecte personal i Astyanax sorgeix de les necessitats de Netflix per treballar amb Cassandra) i amb algunes funcionalitats menys que Astyanax.
- **Gestió d'autenticació i autorització d'usuaris:** L'aplicació ha de gestionar la identificació dels usuaris i les funcionalitats a les que tenen accés, per tant, és necessari un component que faciliti aquesta tasca al desenvolupament. S'han avaluat diverses eines per a aquest propòsit però moltes d'elles formaven part de frameworks més complexes i no era senzill el fet de desacoblar-les, per tant, l'elecció ha quedat reduïda a **Apache Shiro** (prèviament JSecurity). Apache Shiro és un gestor d'autenticació i autorització que permet definir diversos orígens de dades i aquests poden ser o bases de dades o altres sistemes de gestió d'usuaris com servidors LDAP o Directoris Actius.

En principi, el desenvolupament de la part del servidor s'hauria de dur a terme amb les eines mencionades tot i que aquesta llista està subjecte a canvis en funció de l'avanç del projecte.

## Tasques programades

Com ja s'havia introduït en seccions anteriors, el framework de gestió de tasques programades serà Quartz. Aquesta llibreria ens permet gestionar les tasques amb diferents instàncies de la llibreria clusteritzades, gestió de fallades de les diferents instàncies, connexió a la seva API a través de codi Java per a gestionar les tasques i integració amb els principals contenidors de servlets i servidors d'aplicacions Java.

Així com en d'altres aspectes del projecte la tecnologia que s'utilitza té un component d'innovació, Quartz és un producte que s'utilitza en importants projectes de software lliure (Pentaho o JBoss, per exemple) i la seva maduresa està garantida.

## Servei web

El servei web que ha d'oferir la funcionalitat de part de l'aplicació web també necessitarà part dels components comentats anteriorment. Com es tracta d'oferir funcionalitats similars, és molt possible que la implementació sigui exactament la mateixa.

Independentment de les necessitats que pugui compartir amb la banda de servidor de l'aplicació web, el servei web té una necessitat independent: La llibreria d'implementació del servei web (no implementarem el servei web des de zero havent-hi llibreries madures que es dediquen exclusivament a això).

En la elecció de la llibreria més adient per a la implementació del servei web s'han avaluat principalment dues alternatives JAX-WS i Axis 2. Ambdues llibreries ofereixen un nivell de maduresa prou elevat per al projecte però al final s'ha imposat JAX-WS per un motiu molt concret, el nivell d'integració que pot arribar a oferir amb Maven. Les implementacions de serveis web amb Axis també es poden gestionar amb Maven però la facilitat amb la que es pot dur a terme aquesta tasca amb JAX-WS ha pesat més que qualsevol altre paràmetre (a nivell de funcionalitat queden al mateix nivell, és a dir, que no s'ha perdut funcionalitat pel fet d'escollir JAX-WS davant d'Axis).

## Base de dades de configuració

La base de dades de configuració del sistema serà implementat per un sistema gestor de base de dades relacional clàssic. L'avaluació d'alternatives en aquest punt s'ha limitat voluntàriament per tal d'evitar un pseudo estudi de l'estat de l'art de les bases de dades relacionals.

En l'avaluació s'han tingut en compte principalment PostgreSQL i MySQL. Tot i oferir totes dues les funcionalitats requerides per al projecte s'ha optat per MySQL pel coneixement que té l'equip del projecte d'aquest software de base de dades.

Cal comentar que aquest punt podria estar a l'aire perquè l'equip d'operacions del client està més habituat al treball amb bases de dades d'Oracle i podria donar-se el cas que el projecte s'acabés implementant amb Oracle.

En qualsevol cas, cal destacar que la utilització d'SpringJDBC per a la comunicació amb els servidors de bases de dades poden facilitar enormement el canvi entre els diferents motors de persistència comentats en aquest apartat.

## Base de dades de càrrega

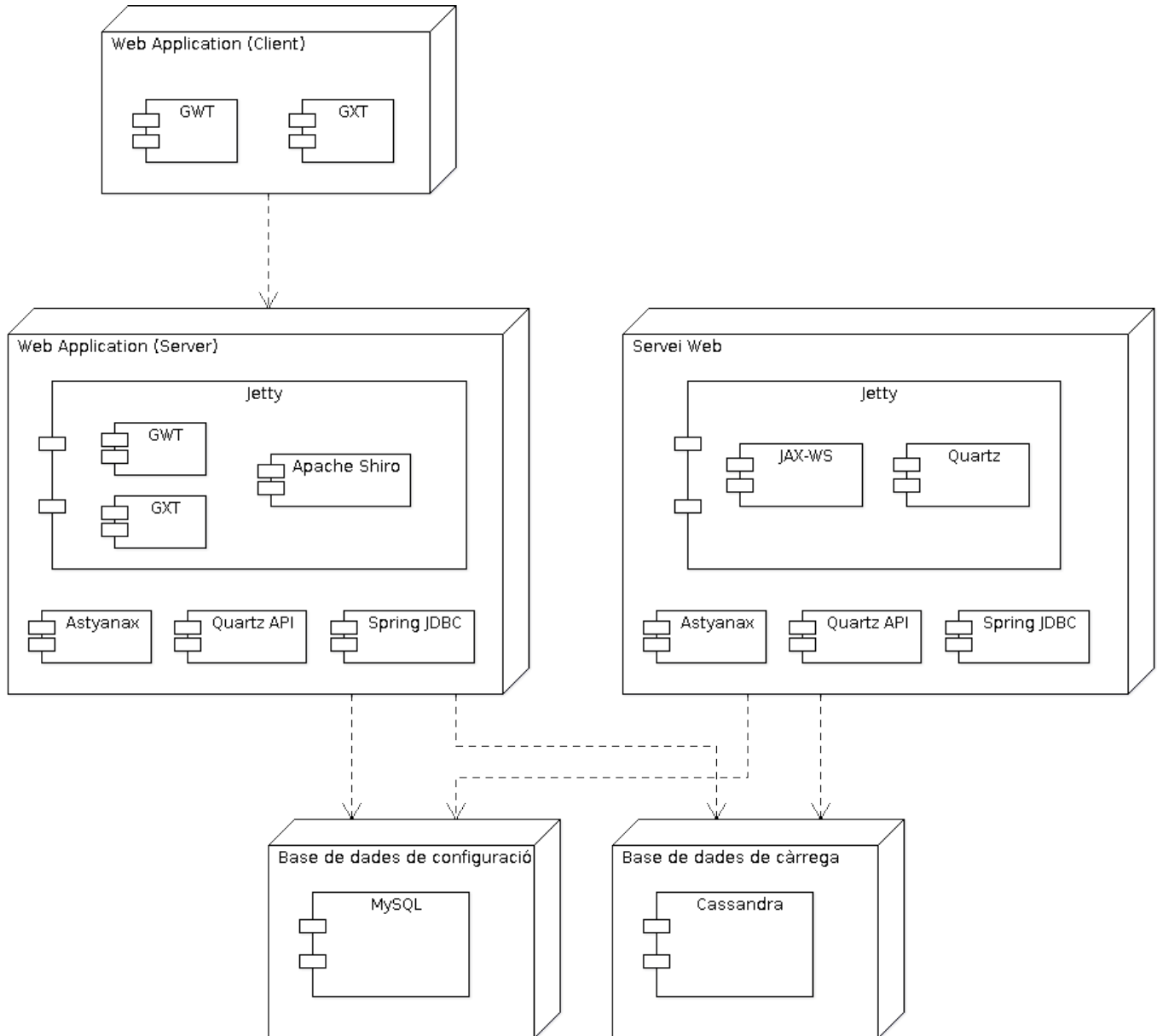
La base de dades de càrrega, de la qual som només un dels clients, ve donada per factors externs a aquest projecte (ja hem comentat anteriorment que la base de dades de càrrega s'ha definit en un projecte paral·lel per a suportar un gran volum d'entrades d'informació).

El magatzem de dades que s'utilitzarà serà doncs, Cassandra. Cassandra és una base de dades que té l'origen en Facebook però que actualment depèn de la Apache Software Foundation i DataStax, l'empresa que ofereix suport comercial i versions empresarials del

software. Tot i tenir un bon nombre de seguidors, Cassandra és una tecnologia molt jove i tot el que està relacionat amb ella té un component d'experimental que el projecte ha de tenir en compte com a risc.

## Diagrames

A continuació es mostra el diagrama de components que respon a l'anàlisi realitzat de tots els components del sistema:



Un canvi substancial que es pot observar en aquest diagrama és el canvi d'ubicació del motor d'execució de tasques programades. Aquest component s'ha decidit moure'l del servidor de l'aplicació web al servidor que allotja el servei web per motius de càrrega del sistema.

La utilització de l'aplicació web, com ja hem comentat abastament, comporta un elevat nombre de peticions asíncrones. Si tenim en compte que aquesta aplicació ha de ser utilitzada per diversos usuaris alhora, podem acabar tenint un servidor amb una gran consum de recursos (principalment d'entrada/sortida) i un altre amb una càrrega molt reduïda. Si a aquest fet li sumem la facilitat per escalar (en nombre de servidors) el servei web i que aquesta escalabilitat és compatible amb els mecanismes d'escalat que ens ofereix Quartz, la decisió de moure aquest component queda completament justificada.

## Desenvolupament

El procés de desenvolupament per a aquest projecte ha tractat d'aplicar un conjunt de bones pràctiques per tal de donar resposta a l'interès del client en l'excel·lència que se suposa en la gestió dels projectes a una empresa certificada per al nivell 3 del CMMI.

Aquestes bones pràctiques han estat recolzades amb la implantació de noves eines de suport. Tant les bones pràctiques incloses en el procés de desenvolupament com les eines adoptades per fer-les possibles es detallen a continuació:

### Utilització d'eines de control de versions distribuïdes

Cada cop està més generalitzada la utilització d'eines de control de versions per als projectes informàtics. Tot i així, l'adopció d'aquestes eines per part d'algunes empreses ha estat relativament lenta i no sempre s'han posat al dia pel que fa a l'evolució del programari que ens proveeix aquesta funcionalitat.

El cas d'aquest projecte no és diferent, l'empresa, tot i tenir definit en els seus processos que la utilització d'eines de control de versions és quelcom indispensable en els projectes que es duen a terme a qualsevol dels departaments, té Subversion -i no alternatives més actuals- definit com eina oficial per a complir aquest objectiu.

Subversion és una eina amb un gran potencial i ha demostrat les seves fortaleses en molts projectes des de la seva primera versió publicada a l'Octubre del 2000 però, a dia d'avui, hi ha un conjunt d'eines que sota el paradigma del control de versions distribuït poden oferir un millor rendiment en equips de projectes des-localitzats o en mobilitat (com és el cas d'aquest projecte). Un dels aspectes en que GIT és superior és a l'hora de gestionar branques i tags. Subversion només proveeix la possibilitat de tenir tantes còpies com branques o tags es necessitin al projecte mentre que GIT les gestiona de forma interna a base de diferencials. Això suposa un estalvi en espai de disc i una reducció del temps destinat a realitzar aquestes operacions que en funció de la mida del projecte poden arribar a ser rellevants.

Pels motius exposats en el darrer paràgraf s'ha decidit utilitzar un repositori GIT per a la gestió de versions. L'adopció de GIT com a eina de control de versions no és senzilla ni en equips on es coneixen altres eines de control de versions com Subversion o CVS però s'ha considerat adequat per tal de millorar les competències de l'equip i gaudir de les funcionalitats que els sistemes de control de versions distribuïts ens poden oferir.

### Implementació de proves unitàries

Un altre dels aspectes en el que s'ha volgut introduir al projecte com a novetat dins de l'empresa és la implementació de proves unitàries per als components del sistema.

Al tractar-se d'un projecte amb un llenguatge de programació únic (Java, tot i que el framework GWT acaba traduint el codi a HTML i Javascript), l'eina escollida per a la implementació de les proves unitàries ha estat JUnit. JUnit proporciona una herència de



classes molt senzilla per a la implementació de les proves i aquestes són fàcilment executables tant des del entorn integrat de desenvolupament (en el nostre cas Eclipse) com des d'altres eines que s'han utilitzat al projecte (el projecte ha utilitzat Maven com a eina de suport al cicle de vida del software).

Les proves unitàries desenvolupades al llarg del projecte es redueixen a components que no requereixen interacció amb altres paquets de programari (formin part del programari desenvolupat dins del projecte o components independents utilitzats com per exemple el sistema gestor de base de dades). Les proves implementades que requereixen la interacció entre diferents components formen part de les proves d'integració, però aquestes es detallen en apartats posteriors.

## Utilització d'eines de suport per a la gestió del cicle de vida

Algunes de les tasques que més temps “fan perdre” en un projecte són les relacionades amb la compilació, empaquetatge i distribució dels paquets de software. Aquests processos acostumen a ser tasques repetitives en les que és molt senzill introduir-hi errors humans.

Tant pel fet de ser tasques repetitives com per la repercussió que poden tenir els errors humans, és aconsellable la utilització d'eines de suport a la gestió del cicle de vida del programari. Aquestes eines faciliten la gestió del cicle de vida i alleugereixen la càrrega de feina de l'equip de desenvolupament fent possible la delegació d'aquestes tasques a processos automatitzats.

Com ja s'ha comentat anteriorment, amb l'excepció de certs matisos, el projecte s'ha implementat completament en Java. Això ens ha donat la possibilitat d'escollir Maven com a eina de suport al cicle de vida del programari, delegant així també un tema clau com és la gestió de dependències.

Alternatives com Gradle o Apache Ant van ser descartades, la primera pel grau de maduresa extra que aportava Maven respecte Gradle (molt valorat per part de l'empresa) juntament amb el coneixement que podien tenir altres companys de l'eina i la segona per la simplicitat de Maven en la definició de tasques predefinides (que es va valorar més que la flexibilitat que aportava Ant).

Cal comentar que tot i l'estructura rígida que utilitza per defecte Maven als projectes gestionats amb aquesta eina, s'ha pogut comprovar satisfactòriament la seva eficiència en projectes Java de plataforma (llibreries), serveis web (SOAP implementats amb JAX-WS) i projectes amb el framework GWT (que requereixen unes eines concretes per al desenvolupament i compilació).

## Utilització d'eines d'integració contínua

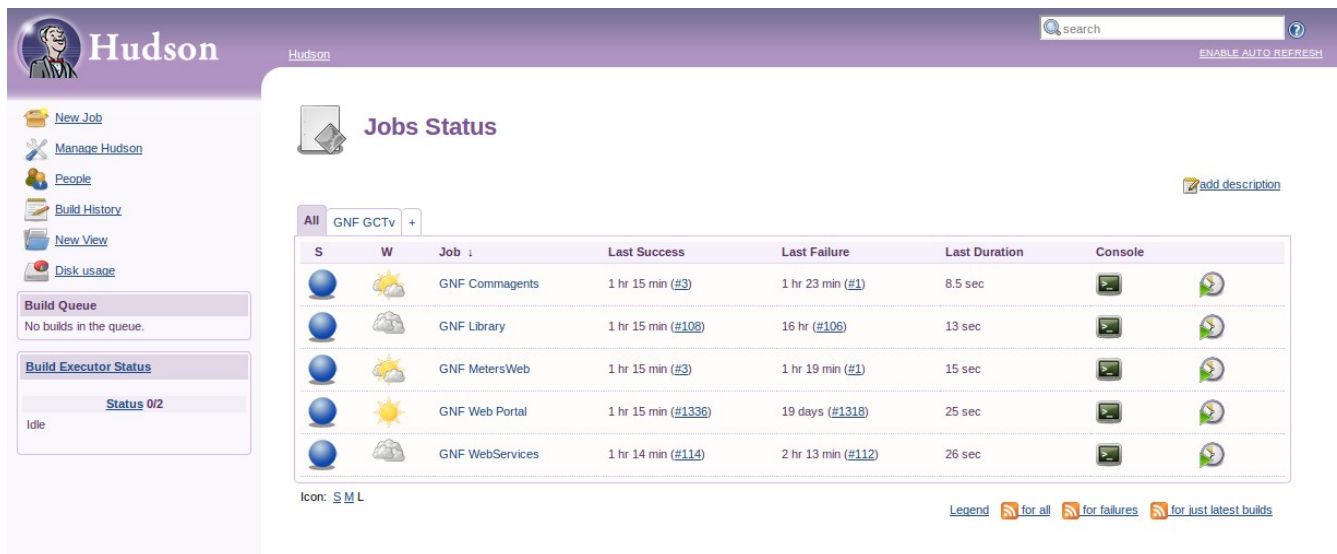
Les eines d'integració contínua permeten a l'equip de projecte saber en tot moment l'estat del projecte. Aquesta pràctica, a GTD, era molt poc utilitzada, fins el punt que únicament s'estava utilitzant en un projecte per petició expressa d'un client.

En el present projecte s'han utilitzat eines d'integració contínua, bàsicament, per

monitoritzar en tot moment que els canvis en el codi no provocaven cap error. Així doncs, els tres projectes dels que es compona el projecte s'han configurat en un servidor d'integració continua, en concret Hudson (dins de l'entorn de desenvolupament).

Hudson ens permet en un cop d'ull veure l'estat de les darreres "builds" del projecte i, gràcies a la utilització de JUnit i Maven, executar una bateria de proves unitàries i d'integració abans d'acceptar una "build" concreta.

Les bateries de proves unitàries s'implementen utilitzant JUnit tal i com s'ha comentat en l'apartat d'Implementació de proves unitàries. Les proves d'integració s'han implementat utilitzant les funcionalitats que aporta Maven per a satisfer aquest objectiu; aquestes funcionalitats permeten executar proves implementades amb frameworks d'implementació de proves unitàries contra paquets compilats i desplegats de forma temporal per la mateixa eina. Aquesta és una gran funcionalitat per al projecte ja que evita que l'equip de desenvolupament hagi d'adquirir competències en un altre framework d'implementació de proves d'integració.



The screenshot shows the Hudson web interface. On the left is a navigation menu with options like 'New Job', 'Manage Hudson', 'People', 'Build History', 'New View', and 'Disk usage'. The main area is titled 'Jobs Status' and shows a table of jobs. The table has columns for 'S' (Status), 'W' (Weather icon), 'Job', 'Last Success', 'Last Failure', 'Last Duration', and 'Console'. The jobs listed are GNF Commagents, GNF Library, GNF MetersWeb, GNF Web Portal, and GNF WebServices. Below the table, there are icons for 'S M L' and a legend for 'for all', 'for failures', and 'for just latest builds'.

S	W	Job	Last Success	Last Failure	Last Duration	Console
		GNF Commagents	1 hr 15 min (#3)	1 hr 23 min (#1)	8.5 sec	
		GNF Library	1 hr 15 min (#108)	16 hr (#106)	13 sec	
		GNF MetersWeb	1 hr 15 min (#3)	1 hr 19 min (#1)	15 sec	
		GNF Web Portal	1 hr 15 min (#1336)	19 days (#1318)	25 sec	
		GNF WebServices	1 hr 14 min (#114)	2 hr 13 min (#112)	26 sec	

*Instal·lació de Hudson per al projecte*

## Implantació i desplegament

El desplegament del sistema ha constatat de quatre entorns tals com:

- **Desenvolupament:** L'entorn de desenvolupament consta d'un servidor sense una configuració de maquinari especialitzada per a cap component. Aquest servidor allotjava tots els serveis necessaris per al desenvolupament i altres eines importants per a l'equip com el software de suport al desplegament de servidors, el servidor de monitorització<sup>12</sup> o un repositori de Maven intern.
- **Integració:** Aquest entorn s'ha desplegat en un hardware similar al de producció però sense la redundància ni el nombre de servidors dels entorns productiu o pre-productiu per a la base de dades de càrrega.
- **Pre-producció:** L'entorn de pre-producció és un entorn idèntic al productiu en el que es poden provar els paquets de software desenvolupats en les mateixes condicions que en l'entorn de producció. Mateix nombre de màquines amb les mateixes característiques.
- **Producció:** L'entorn productiu és l'entorn que es posa al servei del client per tal de satisfer les seves necessitats.

Al final de la secció s'ha preparat un sub apartat amb els diagrames de desplegament dels entorns de desenvolupament, integració i un diagrama per als entorns de pre-producció i producció.

Per a facilitar les operacions en els diferents entorns i mantenir la qualitat en tots ells amb independència del nombre de vegades que calgués reconfigurar-los es van adoptar algunes eines que facilitessin les tasques d'instal·lació, configuració i mesura del rendiment de certs paquets de programari. A continuació hi ha una explicació del software utilitzat així com els motius principals per a la seva adopció.

## Monitorització del consum de recursos de les implementacions

Un dels requeriments no funcionals del sistema és el rendiment, aguantar el volum de trànsit expressat pel client en el moment de l'oferta. Recordem que el projecte persegueix la possibilitat de gestionar un elevat nombre de dispositius des d'una localització central, per aquest motiu és molt important saber en tot moment si estem provocant un augment dels recursos necessaris de forma involuntària.

Per tenir informació sobre aquest fet s'ha decidit dotar als servidors d'un sistema de monitorització per a poder comprovar el nivell de consum de recursos de cadascun d'ells. Aquest sistema és útil a tots els entorns, fins i tot a desenvolupament o integració on es pot comprovar el consum de recursos per a les tasques de compilació, empaquetat i proves (sent aquesta darrera la més representativa per comprovar l'augment del consum de

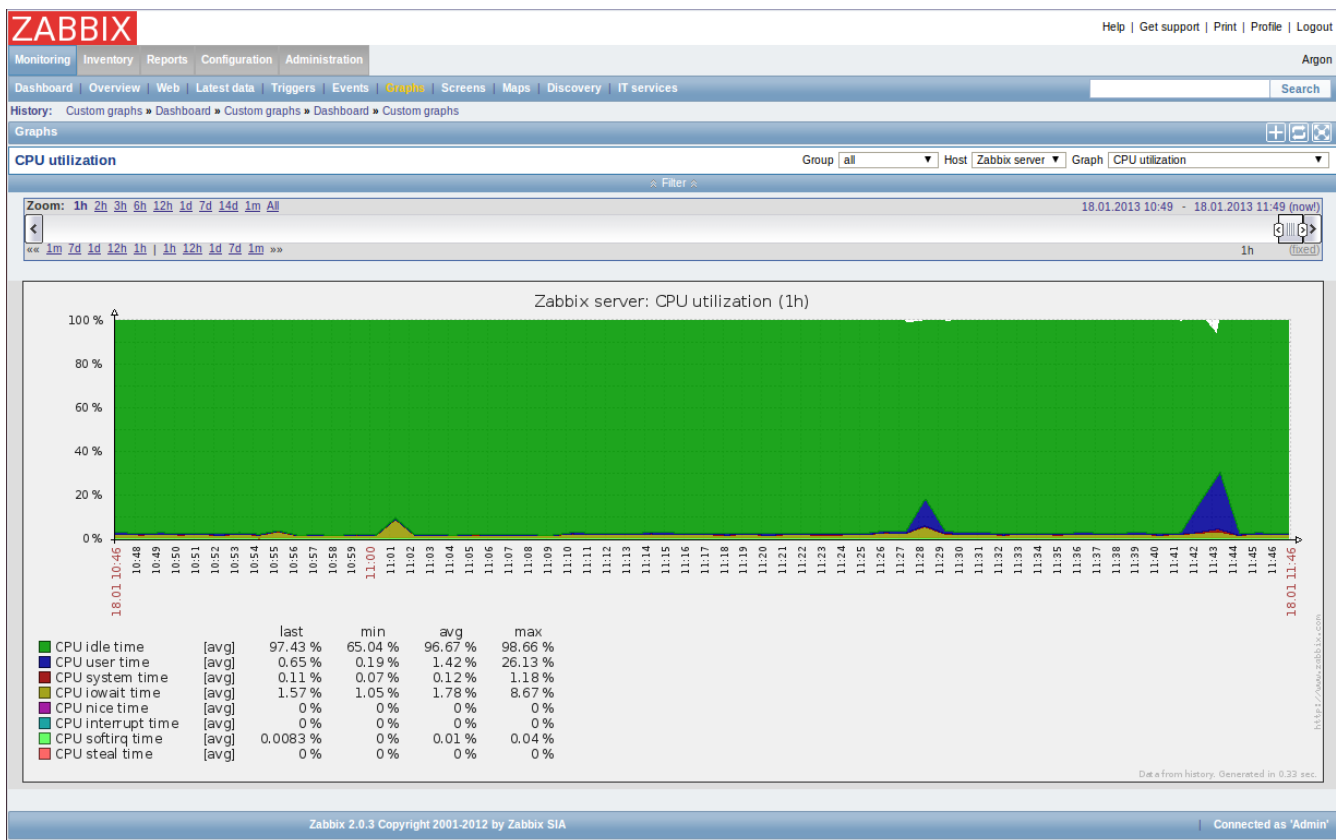
---

12 Tant el servidor de monitorització com el software de suport al desplegament de servidors estan explicats en aquest mateix apartat.

recursos en cas de que s'hagi produït).

Els beneficis d'aplicar aquest tipus d'eina en entorns de producció o pre-producció són motiu de consens dins de la comunitat i no cal explicar els motius de la seva adopció. Més aviat hauria de ser el contrari, la no adopció d'eines de monitorització en entorns productius hauria de generar la necessitat de documentar l'absència d'aquestes.

L'eina utilitzada per a tal efecte ha estat Zabbix perquè ens proporciona la flexibilitat que una solució de codi obert proveeix en quant al nombre de servidors que podem gestionar i un cost controlat de la solució. Alternatives de programari privat havien estat descartades per un model de llicències basades en el nombre de processadors o de nuclis del processador.



*Gràfic d'estat de la CPU del servidor de desenvolupament*

## Utilització d'eines de suport per al desplegament automatitzat de servidors

Amb un nombre variable i canviant de servidors, una de les tasques més pesades per dur a terme és la gestió del desplegament d'aquests. Per aquest motiu s'han avaluat solucions

que facilitessin les operacions de personalització de servidors en funció de certs paràmetres.

En aquest àmbit, el projecte s'ha decantat pel desplegament de Puppet, una eina també de codi obert que permet definir l'estat dels servidors, modificar les seves característiques i corregir les desviacions que es puguin produir.

Puppet treballa amb un DSL<sup>13</sup> que cal entendre per tal de realitzar les operacions requerides. Aquest DSL s'utilitza en una sèrie de fitxers en els que es defineix l'estat de cada servidor (o grup de servidors). El format en el que s'emmagatzema aquest DSL és text pla, format que ens ha facilitat la inclusió d'aquests perfils de servidors en el nostre sistema de control de versions.

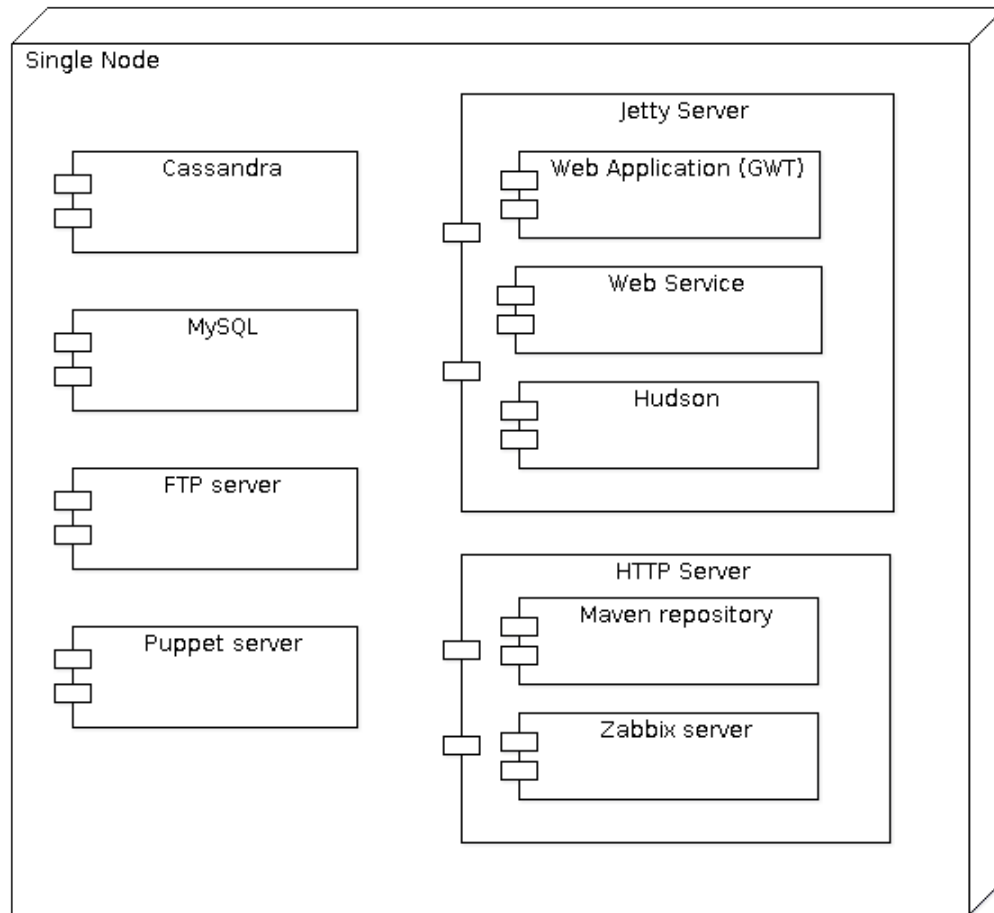
## Diagrames

Per poder reproduir els documents a una mida adequada per a la seva lectura s'han separat en pàgines diferents. De la mateixa manera, per no sobrecarregar el diagrames s'han amagat les relacions entre els diferents components ja que aquestes es poden observar de forma molt més clara al diagrama de la secció d'anàlisi i disseny tècnic.

### *Entorn de Desenvolupament*

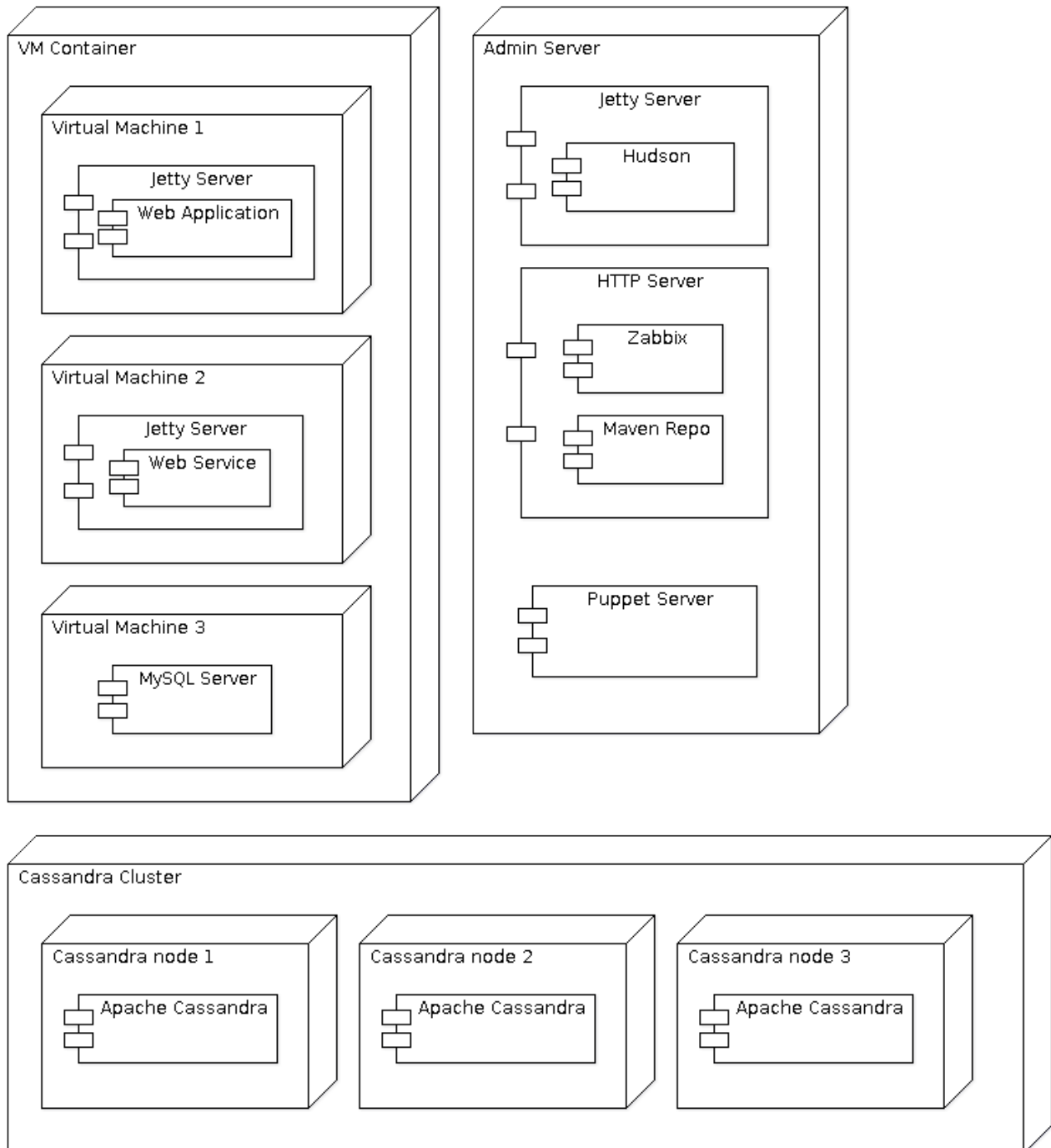
---

13 DSL: Domain Specific Language. Els DSL acostumen a ser pseudo-llenguatges de programació que es defineixen per tal de facilitar l'ús d'una certa tecnologia a un grup d'usuaris als que no se'ls suposa un gran domini del llenguatge de programació del que sorgeix aquesta tecnologia.

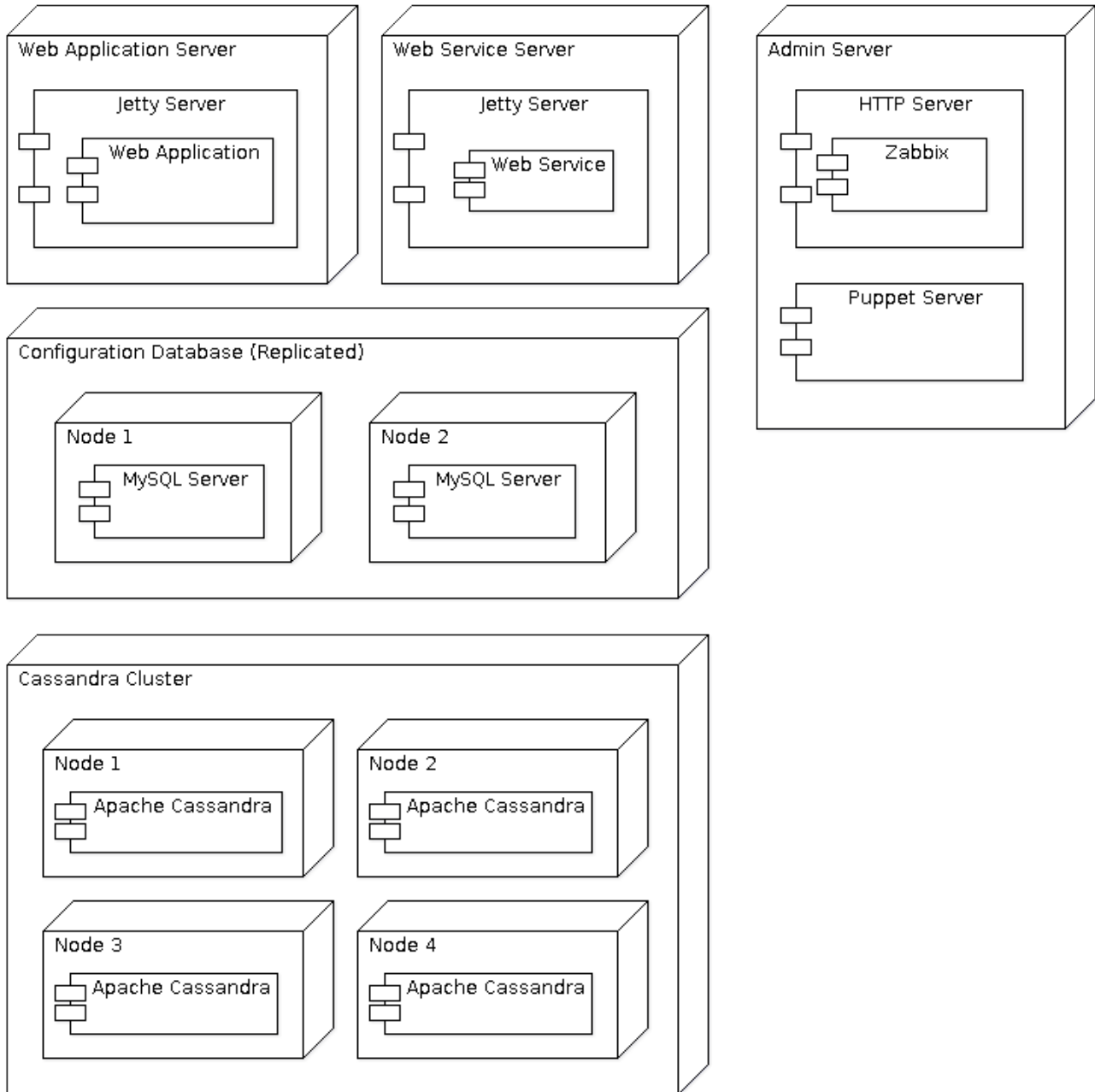


Tant en aquest diagrama com en els següents no s'han indicat la instal·lació dels agents de monitorització ni del sistema Puppet en tots els servidors físics i les màquines virtuals.

## Entorn d'Integració



### Entorn de Pre-producció i Producció





## Captures de l'aplicació



The screenshot shows the login interface for the GTD application. At the top center is the 'gtd' logo, which consists of a dark blue circle containing the lowercase letters 'gtd' in white. To the right of the logo, the text 'Benvinguts a l'aplicació de GTD' is displayed. Below the logo and text is a login form. The form has a grey header bar with the word 'Entra' in white. Inside the form, there are two input fields: the first is labeled 'Àlies' and the second is labeled 'Contrasenya'. Below these fields is a grey button with the text 'Entra' in white.

*Formulari d'identificació*

**gtd** GTD test

Welcome xavi

Tauler Gestió de DCs Gestió de comptadors Gestió de tasques programades Users

**Dashboard Info**  
There you will find interesting information about the system status

**Recent logins**

Àlies	Date	IP Address
xavi	Fri Jan 18 09:19:48 CET 2013	127.0.0.1
xavi	Fri Jan 18 09:18:16 CET 2013	127.0.0.1
xavi	Fri Jan 18 09:04:36 CET 2013	127.0.0.1
xavi	Fri Jan 18 09:02:40 CET 2013	127.0.0.1
xavi	Wed Jan 16 17:51:52 CET 2013	127.0.0.1

**Monitoring data**

**User connection Check**

**HTTP Request**

**Main section**  
Concentrators  
Meters  
Scheduled tasks  
Users

**Contact**  
Administrator  
Sysadmin  
Maintainer

**Information**  
This website is optimized for screen resolution: 1280x1024.

**gtd** Developed and maintained by:  
**GTD Sistemas de Información**

*Dashboard de l'aplicació*

The screenshot shows a web application interface for 'GTD test'. At the top left is the 'gtd' logo. To the right, it says 'Welcome xavi' with a 'Log out' button. Below this is a navigation menu with items: 'Tauler', 'Gestió de DCs', 'Gestió de comptadors', 'Gestió de tasques programades', and 'Users'. The main content area is titled 'Lista de DCs' and includes a 'Nou DC' button. A table lists two DCs with columns for 'Id', 'Etiqueta', 'Info', and four action buttons: 'Meters', 'Veü', 'Edita', and 'Elimina'. The footer contains four sections: 'Main section' (Concentrators, Meters, Scheduled tasks, Users), 'Contact' (Administrator, Sysadmin, Mantainer), 'Information' (This website is optimized for screen resolution: 1280x1024.), and a 'gtd' logo with the text 'Developed and maintained by: GTD Sistemas de Información'.

gtd GTD test

Welcome xavi [Log out](#)

Tauler Gestió de DCs Gestió de comptadors Gestió de tasques programades Users

### Lista de DCs [Nou DC](#)

Id	Etiqueta	Info				
ZIV0000036292	4CCT-EA6-334104BC	Size 1	Meters	Veü	Edita	Elimina
ZIV0000036293	4CCT-EA6-334104BC	Size 2	Meters	Veü	Edita	Elimina

---

**Main section**  
Concentrators  
Meters  
Scheduled tasks  
Users

**Contact**  
Administrator  
Sysadmin  
Mantainer

**Information**  
This website is optimized for screen resolution: 1280x1024.

gtd  
Developed and maintained by:  
**GTD Sistemas de Información**


*Lilstat de concentradors*

**Data Concentrator Details**

<b>Id</b>	ZIV0000036292	<b>Type</b>	4CCT-EA6-334104BC
<b>Manufacturing year</b>	2011	<b>Equipment type</b>	concentrator
<b>Protocol</b>	ISDIP-3.0-2010/09	<b>Communications</b>	
<b>DC IP Address</b>	10.177.9.130	<b>WS port</b>	8080
<b>IP Mask</b>	255.255.255.240	<b>Gateway IP</b>	10.177.9.129
<b>DHCP</b>	<input type="checkbox"/>	<b>Priority</b>	<input checked="" type="checkbox"/>
<b>NTP Server IP</b>		<b>FTP retry attempts</b>	null
<b>Time btwn FTP retries</b>	300	<b>Sync Meters</b>	<input type="checkbox"/>
<b>Max time deviation</b>	600	<b>Reset message</b>	<input type="checkbox"/>
<b>#Meters in reports</b>	2000	<b>Time send request</b>	3600
<b>Time discon meter</b>	86400	<b>Retry discon meter</b>	5
<b>Time retry interval</b>	120	<b>Meter reg data</b>	010000600100ff02010000600
<b>Report format</b>	0	<b>S26 content</b>	
<b>Values check delay</b>	0	<b>Timeout prime fw upgrade</b>	300
<b>IP STG</b>		<b>URL STG</b>	
<b>Password STG</b>		<b>User reports</b>	
<b>IP Reports</b>		<b>User DC Upgrade</b>	
<b>Password Reports</b>		<b>User Meter Upgrade</b>	
<b>IP DC Upgrade</b>		<b>User Cycles</b>	
<b>Password DC Upgrade</b>		<b>Folder Cycles</b>	
<b>IP Meter Upgrade</b>			
<b>Password Meter Upgrade</b>			
<b>IP Cycles</b>			
<b>Password Cycles</b>			

Developed and maintained by:  
**TD Sistemas de formación**

Formulari de modificació d'un concentrador de dades

GTD testWelcome xavi [Log out](#)

[Tauler](#) [Gestió de DCs](#) [Gestió de comptadors](#) [Gestió de tasques programades](#) [Users](#)

### Llista d'usuaris

Id	Nom	Rol			
1	gnf	admin	<a href="#">Ve</a>	<a href="#">Edita</a>	<a href="#">Elimina</a>
3	xavi	admin	<a href="#">Ve</a>	<a href="#">Edita</a>	<a href="#">Elimina</a>
4	jordi	admin	<a href="#">Ve</a>	<a href="#">Edita</a>	<a href="#">Elimina</a>
5	jordi2	viewer	<a href="#">Ve</a>	<a href="#">Edita</a>	<a href="#">Elimina</a>
6	xavi2	manager	<a href="#">Ve</a>	<a href="#">Edita</a>	<a href="#">Elimina</a>

### Nou Usuari

Àlies

Contrasenya

Role [Select a Role ▼](#)

[Crea](#)


### Llista de rols

Id	Rols	Permisos			
1	admin	manage_dcs, manage_meters, manage_users, manage_scheduled, view_dcs, view_meters, view_users, view_scheduled	<a href="#">Ve</a>	<a href="#">Edita</a>	<a href="#">Elimina</a>
2	manager	manage_dcs, manage_meters, manage_scheduled, view_dcs, view_meters, view_users, view_scheduled	<a href="#">Ve</a>	<a href="#">Edita</a>	<a href="#">Elimina</a>
3	viewer	view_dcs, view_meters, view_users, view_scheduled	<a href="#">Ve</a>	<a href="#">Edita</a>	<a href="#">Elimina</a>

**Main section**  
Concentrators  
Meters  
Scheduled tasks  
Users

**Contact**  
Administrator  
Sysadmin  
Mantainer

**Information**  
This website is optimized for screen  
resolution: 1280x1024.



Developed and  
maintained by:  
**GTD Sistemas de  
Información**

*Pàgina de gestió d'usuaris*

## Manteniment

El projecte tot just comença la seva etapa de manteniment en el moment de l'entrega de la versió final d'aquest document. Com GTD serà l'encarregat del manteniment del sistema resultant s'ha tingut en ment en totes les fases del projecte el facilitar al màxim les tasques que hagin de realitzar els companys de l'equip de manteniment.

Una de les accions que s'han pres per materialitzar l'objectiu definit en el paràgraf anterior és la factorització de les operacions de recopilació d'informació. Tant les consultes a la base de dades com altres processos d'obtenció d'informació s'han inclòs en una llibreria independent de l'aplicació web i del servei web, que pot ser utilitzada des d'ambdós projectes.

Aquesta organització dels projectes (aplicació web, servei web i llibreria comuna), a part de reduir a un únic punt les operacions d'accés a les dades, també facilita la gestió de l'equip de manteniment, definint de forma molt més clara els perfils que s'han d'encarregar dels diversos sub-projectes en els que s'ha acabat dividint el projecte.

Una altra de les accions preses en aquest sentit és la utilització de Maven i la creació d'un repositori intern per a GTD. Aquesta mesura, tot i que sembla que va en línia amb altres aspectes del projecte també impacta positivament sobre la mantenibilitat del programari.

Per una banda, el projecte pot ser compilat i empaquetat per qualsevol persona amb accés al repositori intern (hi han artefactes que només es poden trobar al repositori intern per motius de llicències o perquè són artefactes generats de forma interna). Per una altra banda, la gestió de dependències permet fàcilment eliminar o afegir noves dependències mitjançant la modificació d'un fitxer de text (també sota el control de versions), cosa que redueix el volum i la complexitat del desplegament del projecte.

Per millorar la mantenibilitat global del codi font s'ha optat per mantenir la documentació en el format Javadoc. Aquest format permet generar una documentació (tècnica) bàsica per a no desenvolupadors i permet als desenvolupadors futurs aprofitar tots els beneficis d'aquest conegut format de documentació (integració amb IDEs, consulta de la documentació mitjançant interfície web...).

## Relació de Software Lliure utilitzat en el projecte

### Java (OpenJDK)

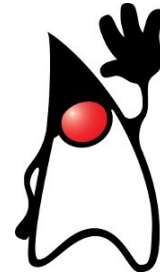
Versió: 7 update 9

Llicència: GNU GPLv2 (amb algunes excepcions)

L'ampli coneixement de Java a l'empresa ha fet d'aquest el llenguatge protagonista del projecte. Juntament amb el grau de coneixement d'aquesta tecnologia també s'ha tingut en compte la gran diversitat d'eines disponibles i la facilitat que Java ens aporta a l'hora de treballar en diverses plataformes en temps de desenvolupament sense que això afecti al resultat final.

Amb l'adquisició per part d'**Oracle** de **Sun Microsystems** (empresa propietària de Java i MySQL entre d'altres) **OpenJDK** ha esdevingut, per ordre de la mateixa Oracle, la implementació de referència de Java i, per tant, es pot considerar OpenJDK com una plataforma de confiança per a implementar aplicacions corporatives (fet que no es donava en versions anteriors).

- Pàgina web d'Oracle sobre tecnologia Java: [enllaç](#)
- Pàgina web d'OpenJDK: [enllaç](#)
- Declaració d'OpenJDK com a implementació de referència: [enllaç](#)



### Servidor web Jetty

Versió: 8.1.8

Llicència: dual, Apache License 2.0 o Eclipse Public License 1.0

Jetty és un projecte sota l'empara de la **Eclipse Foundation** des de 2009 (prèviament havia estat part de Codehaus i, en les seves etapes inicials com a projecte a Sourceforge). Aquest servidor web proveeix un gran rendiment en l'entrega de continguts estàtics superior a altres paquets de programari similars que també poden actuar com a contenidors de servlets, com per exemple **Tomcat**.

Per tenir una idea dels beneficis que Jetty pot aportar, aquest és el servidor utilitzat per Google en la seva oferta comercial de PaaS, **Google App Engine** i alhora és un servidor web lleuger que es pot encastar a l'Eclipse per a realitzar proves en una estació de treball estàndard.

Hi ha una empresa que actualment dona suport comercial per al desplegament, implementació i personalització de Jetty anomenada Webtide.

- Pàgina web del projecte Jetty a l'Eclipse Foundation: [enllaç](#)
- Pàgina web del projecte Jetty a Codehaus: [enllaç](#)



- Pàgina web de Webtide: [enllaç](#)

## Eclipse 4

Versió: 4.2 (Juno)

Llicència: Eclipse Public License 1.0

La versió més recent de l'entorn integrat de desenvolupament Eclipse ha estat l'editor per defecte del projecte. A part de la gran **integració** amb **Java** i totes les eines com **JUnit**, **Maven** o **Hudson**, Eclipse també té una gran integració amb el framework de desenvolupament web **GWT**. Aquesta integració ve de la mà de Google, qui proveeix un plugin per a Eclipse de les seves tecnologies més populars (Android, Google App Engine, GWT i altres).

- Pàgina web d'Eclipse: [enllaç](#)
- Pàgina web del complement d'Eclipse per a Maven: [enllaç](#)
- Pàgina web del complement de Maven per a Eclipse: [enllaç](#)
- Pàgina web del complement de Google: [enllaç](#)



## Google Web Toolkit (GWT)

Versió: 2.5

Llicència: Apache Software License 2.0

El framework de desenvolupament web triat per al projecte ha estat GWT. GWT ens proporciona la facilitat d'implementar una aplicació web amb un gran nombre de **crides asíncrones** per obtenir les dades des del servidor.

La facilitat principal de GWT davant d'altres alternatives radica en la capacitat del framework d'oferir, **en un mateix llenguatge** la implementació de codi de part del client (s'executa en el navegador) i de part del servidor (s'executa com a part del Servlet). Un dels trets diferencials d'aquest framework és la seva integració amb Eclipse i el plugin per a integrar el desenvolupament de l'aplicació amb l'eina de gestió de cicle de vida de Maven.

- Pàgina web del framework: [enllaç](#)
- Pàgina web del plugin de Maven per a projectes en GWT: [enllaç](#)



## Git

Versió: 1.7.10

Llicència: GNU GPLv3





El sistema de control de versions utilitzat per al desenvolupament del nucli de Linux també ha estat l'eina utilitzada per al control de versions del projecte. GIT ens ha proporcionat la possibilitat de **treballar remotament**, des de diferents ubicacions sense perdre l'historial de cap membre de l'equip i amb una gestió de **branques** molt més eficient que no pas Subversion o CVS.

Per no haver de gestionar els repositoris de forma local i tenir disponible el repositori en tot moment (o pràcticament en tot moment) es va decidir des d'un principi escollir un proveïdor de repositoris GIT privats. En aquesta decisió va influir molt la dificultat de comunicació amb GitHub que l'empresa havia tingut en un projecte anterior i es va optar per Bitbucket, un servei de l'empresa Atlassian, especialistes en SaaS en l'àmbit de la gestió de projectes de software.

- Pàgina web de GIT: [enllaç](#)
- Pàgina web de Bitbucket, el proveïdor de repositoris GIT del projecte: [enllaç](#)
- Pàgina web de l'empresa Atlassian: [enllaç](#)

## Maven

Versió: 3.0.4

Llicència: Apache Software License 2.0

The logo for Maven, featuring the word "maven" in a bold, lowercase, sans-serif font. The letter "a" is colored orange, while the remaining letters "m", "v", "e", "n" are black.

Maven ha estat una de les **innovacions** més importants a l'empresa en quant a projectes Java. Maven és una eina que ens ha ajudat a controlar el **cicle de vida** del programari que s'anava desenvolupant. Gràcies a Maven ha estat senzill **compilar, empaquetar, executar tests unitaris**, configurar els projectes en servidors d'integració contínua i altres tasques menors com la generació de paquets amb el codi font o la documentació.

Alguns complements de Maven, com ara Surefire, ens han permès aprofitar el màxim el coneixement de l'equip de desenvolupament en eines utilitzades en el projecte, com JUnit. En el cas concret de Surefire i JUnit, la utilització de Maven ens ha permès implementar proves d'integració aprofitant els coneixements de l'equip en JUnit i no haver de formar a tot l'equip en una altra eina d'implementació de proves d'integració.

- Pàgina web de Maven: [enllaç](#)

## Hudson

Versió: 3.0.0-RC4

Llicència: Eclipse Public License 1.0



Hudson era un servidor d'integració contínua sota l'empara de l'empresa Sun Microsystems. Amb la compra d'aquesta per part d'**Oracle**, Hudson va patir una època complicada en la que es va dividir en dos projectes **Hudson** i **Jenkins**. En un primer moment semblava que la **comunitat** optava preferentment per seguir fent aportacions a Jenkins (fork impulsat per la comunitat davant la manca de resposta d'Oracle a les seves peticions)

però l'empresa de Califòrnia va decidir relançar Hudson cedint el projecte a la **fundació Eclipse**, on sembla que l'avanç del projecte ha millorat.

La versió utilitzada durant el projecte no està considerada com a una versió estable (es tracta de la release candidate 4) però els avantatges que te a nivell d'usabilitat davant de la versió estable de Jenkins ens va fer decantar per aquesta opció.

- Pàgina web de Hudson: [enllaç](#)
- Pàgina web de Hudson dins de l'Eclipse Foundation: [enllaç](#)
- Pàgina web de Jenkins: [enllaç](#)

## Quartz



Versió: 2.1.6

Llicència: Apache Software License 2.0

Funcionalment un dels problemes més importants a resoldre són les **tasques programades** o automatitzades. És tant probable que s'hagi d'efectuar una operació el mateix dia de cada mes que cada dia exceptuant els festius o els caps de setmana. Realment, el nombre de frameworks o llibreries que proporcionen la **flexibilitat** necessària per gestionar aquesta heterogeneïtat de configuracions no és massa elevat i Quartz ens ho permet amb un grau de maduresa molt elevat.

Un altre punt a favor de Quartz és la gran **documentació** publicada en cada versió (la versió utilitzada és la 2.1.6 però es pot trobar la documentació de qualsevol versió amb un esforç mínim a la pàgina web del projecte). A nivell corporatiu el nombre i el renom d'aplicacions i llibreries que integren Quartz ens oferien la **credibilitat** necessària per apostar per aquest framework.

- Pàgina web de Quartz: [enllaç](#)

## Cassandra



Versió: 1.1.5

Llicència: Apache Software License 2.0

Cassandra és una base de dades noSQL sorgida des de **Facebook** i que ara mateix està impulsada per la **Apache Software Foundation**. Cal destacar en aquest punt que tot i estar sota l'empara de la ASF, una empresa amb una oferta comercial basada en Cassandra és la que porta el gruix del desenvolupament, **DataStax**. Lluny de ser una situació positiva, dona l'efecte que el domini de DataStax del desenvolupament de Cassandra empitjora la qualitat de la documentació i que moltes vegades és complicat trobar la informació al portal web del projecte lliure.

La necessitat d'utilitzar Cassandra ve donada per un projecte paral·lel però l'aprenentatge de Cassandra ha estat una de les grans apostes de l'empresa GTD per a aquest projecte.

- Pàgina web del projecte Apache Cassandra: [enllaç](#)

- Pàgina web de la companyia DataStax: [enllaç](#)
- Pàgina web impulsada per DataStax per a la comunitat d'usuaris de Cassandra: [enllaç](#)

## MySQL

Versió: 5.5

Llicència: GNU GPLv2

La base de dades **relacional** destinada a emmagatzemar la configuració i guardar certes dades que requereixen un control detallat de les transaccions s'ha implementat amb MySQL.

MySQL és un sistema gestor de bases de dades que, juntament amb Java i Hudson ha passat de pertànyer a Sun Microsystems a Oracle recentment. Sembla que aquest fet no ha afectat a l'estabilitat del projecte gràcies al gran nombre de participants en el projecte lliure (com les empreses Percona i InnoDB, que participen activament en la comunitat).



- Pàgina web de MySQL: [enllaç](#)
- Pàgina web de MySQL dins d'Oracle: [enllaç](#)
- Pàgina web de Percona: [enllaç](#)
- Pàgina web de InnoDB a InnoDB: [enllaç](#)

## Apache Shiro

Versió: 1.2.1

Llicència: Apache Software License 2.0

Shiro és una llibreria utilitzada per a facilitar la gestió d'usuaris a l'aplicació web. Amb Shiro ha estat fàcil implementar un sistema d'usuaris, amb permisos que es pugui integrar amb el sistema d'identificació d'usuaris del client i que alhora proveeixi un rendiment acceptable. Un dels problemes principals de Shiro per a la seva adopció com a tecnologia de referència en quant a **autorització** i **autenticació** és l'estat embrionari de la seva documentació.



- Pàgina web d'Apache Shiro: [enllaç](#)

## Spring JDBC

Versió: 3.1.2

Llicència: Apache Software License 2.0

Spring és un dels frameworks més coneguts i més utilitzats en Java. D'aquest projecte, ara sota l'empareda de l'empresa VMWare, s'ha adoptat el component per a facilitar l'accés a la



base de dades.

Cal comentar que part del desenvolupament s'ha realitzat contra una base de dades Oracle, per si el client optava finalment per imposar-la, i part s'ha realitzat contra una base de dades MySQL. Gràcies a Spring JDBC no hi ha hagut cap problema a l'hora de migrar d'una a l'altra sense problemes.

- Pàgina web d'Spring JDBC: [enllaç](#)

## Astyanax

Versió: 1.0.6

Llicència: Apache Software License 2.0

Donada la necessitat d'utilitzar Cassandra, era necessari triar una bona llibreria per poder inserir, modificar i consultar dades sense haver de passar pel protocol natiu Thrift que proporciona Cassandra.

Astyanax, a més de tenir el suport d'una empresa com Netflix al darrere, proporciona funcionalitats que altres client per a Cassandra no oferien i el nivell de maduresa del Astyanax també semblava, a simple vista, major que el d'altres alternatives com Hector (projecte del qual Astyanax es declara hereter).

- Pàgina web d'Astyanax: [enllaç](#)
- Pàgina web d'Hector: [enllaç](#)
- Pàgina web de programari de codi lliure de Netflix: [enllaç](#)

## JUnit

Versió: 4.10

Llicència: Common Public License 1.0

El framework utilitzat per a la implementació de les proves unitàries (i d'integració gràcies a plugins concrets de Maven) ha estat JUnit. Aquest framework és una referència en l'àmbit de les proves unitàries i s'ha portat amb èxit a altres llenguatges com C (CUnit), PHP (PHPUnit), Ada (AUnit) o Python (PyUnit).

JUnit és un estàndard de facto amb una gran integració amb eines com Maven, Hudson o Eclipse, tres de les eines utilitzades en el projecte i amb les que s'ha aconseguit un alt grau de productivitat.

- Pàgina web de JUnit: [enllaç](#)



## Zabbix

Versió: 2.0.0

Llicència: GNU GPLv2



El sistema de monitorització utilitzat al projecte ha estat Zabbix. Zabbix és un sistema de monitorització que destacava el 2010 per la facilitat d'implementació de disparadors en funció de paràmetres de monitorització i de definició de mètriques personalitzades en base a l'execució d'scripts. A dia d'avui aquestes funcionalitats s'han incorporat a bona part de la competència però Zabbix proveeix un nivell de maduresa en aquestes aspectes molt més destacable.

- Pàgina web de Zabbix: [enllaç](#)

## Puppet

Versió: 3.0.1

Llicència: Apache Software License 2.0

Puppet és un sistema d'ajuda a la gestió de servidors. S'utilitza per a garantir l'estat d'un servidor, per exemple, assegurant l'existència d'un usuari concret o que un paquet determinat estigui instal·lat. Puppet no és l'únic programari, Chef és un altre projecte amb característiques similars, utilitzat per empreses com RightScale per a la gestió del desplegament de màquines virtuals. En aquest projecte s'ha optat per Puppet per l'extensa funcionalitat que proveeix el seu pseudo-llenguatge, enlloc de valorar més positivament la possibilitat d'implementar scripts en BASH que proporciona Chef.



- Pàgina web de Puppet: [enllaç](#)
- Repositori de Puppet a GitHub: [enllaç](#)

## Conclusions i treballs a futur

Com s'ha comentat abastament en el present document, aquest projecte té un component innovador molt elevat per al client i per a l'empresa GTD: D'un costat, la incorporació de tecnologies no utilitzades anteriorment per GTD i de l'altra la introducció d'un producte innovador en l'entorn del client.

### Innovació i recursos

La introducció de noves tecnologies a GTD ha posat de manifest la conveniència d'incorporar millores en el procés de producció de programari com ara eines de suport al cicle de vida o servidors d'integració continua. Les millores en quant a eficiència i rendiment en el desenvolupament evidencien la necessitat latent d'aquests canvis en el procés productiu.

El procés d'aprenentatge d'aquestes eines és reduït molt en el moment d'adoptar-les en un projecte real i els beneficis s'han començat a notar tot just a les primeres entregues de software (al cap d'un parell de mesos de començar el projecte).

L'adopció de nous frameworks de desenvolupament web a GTD també ha donat un resultat satisfactori. El rendiment de l'aplicació resultant és correcte i l'aparença és molt més actual que no pas les darreres aplicacions de gestió implementades per la mateixa empresa. Tot i així, també cal destacar que s'han trobat a faltar segons quines eines per a facilitar el desenvolupament de les típiques operacions d'alta, baixa, accés i modificació d'entitats a la base de dades.

El nivell de documentació oficial de GWT és molt elevat (tot i haver utilitzat una versió en desenvolupament, la 2.5, fins ben entrat el projecte) i el suport de la comunitat a través del grup oficial a Google Groups de GWT o el grup de Barcelona GDG (Google Developer Group) funciona francament bé; resol els problemes i ho fa, de mitjana, en períodes molt curts temps. Existeixen altres canals de suport de la comunitat per a aquest framework de desenvolupament però potser caldria destacar la qualitat i la quantitat de les qüestions relacionades amb GWT presents a la comunitat d'StackOverflow, un altre dels recursos que més ha ajudat a resoldre els problemes que anaven sorgint a mesura que el desenvolupament avançava.

També m'agradaria destacar com s'ha posat de manifest el fet de que és possible portar un projecte en el que l'excel·lència tecnològica i de gestió eren un valor afegit de cara al client de forma pràcticament íntegra amb software lliure. Entorn integrat de desenvolupament, servidor d'integració continua, sistema de control de versions, frameworks, bases de dades o sistemes de monitorització, tots ells projectes de codi obert. Casi tot el que s'ha utilitzat ha estat software lliure i el resultat amb tots els components ha estat més que satisfactori.

### Cassandra i el seu entorn

Les conclusions presentades fins al moment estan relacionades amb l'empresa GTD però

també cal analitzar el principal factor d'innovació a nivell tecnològic del projecte, la introducció de Cassandra en un entorn productiu.

Si bé és cert que Cassandra és un projecte de codi obert que es troba sota l'empara de l'Apache Software Foundation, el gruix del desenvolupament el porta a terme l'empresa nord-americana DataStax; empresa que ofereix serveis de consultoria i formació relacionats amb Cassandra i productes comercials inspirats en aquesta base de dades noSQL amb diferents millores (com la d'un gestor que permet conèixer l'estat del clúster de forma visual).

Aquesta concentració del desenvolupament per part de DataStax afecta greument a la qualitat de la documentació de la versió lliure, que en molts casos és incompleta i que en molts altres es troba completament obsoleta. La versió que manté l'empresa de la documentació s'ha anat allunyant de la versió lliure i està administrada de forma que invita els lectors a optar per les seves solucions no lliures del producte.

Un altre punt en contra de l'adopció de Cassandra és la maduresa de les llibreries que s'utilitzen per a comunicar-se amb ella. Casi totes aquestes llibreries són de codi obert però l'única que s'ha trobat realment compatible amb un entorn empresarial ha estat Astyanax, que curiosament també desenvolupa majoritàriament una sola empresa (en aquest cas Netflix). La qualitat d'Astyanax és bastant elevada però la documentació disponible i la resposta de la comunitat en la resolució d'aquests dubtes és bastant deficient. De fet, un dels millors canals per a obtenir suport d'Astyanax és la llista de correu del projecte de Cassandra on sembla que s'ha posicionat com la llibreria predominant en Java.

## Programari lliure, comunitats i fundacions

Personalment el projecte m'ha brindat la possibilitat d'interactuar amb comunitats de software lliures molt actives com la de GWT o Cassandra amb diferents graus de satisfacció, com es pot desprendre de les exposicions anteriors. Igualment, m'ha permès arribar a la conclusió que el lideratge empresarial d'alguns projectes de codi obert poden afectar greument la percepció de qualitat del producte de codi obert per afavorir solucions tancades basades en la mateixa base. No és un cas representatiu perquè en el projecte s'ha tocat un framework que justament evidencia el contrari però la imatge d'Apache Cassandra respecte la imatge de DataStax Cassandra és un bon exemple del que s'intenta transmetre.

Una altra reflexió personal que el projecte m'ha provocat està relacionada amb la gran tasca que les fundacions com la d'Eclipse o la d'Apache fan per al programari lliure. Aquestes fundacions, tot i semblar únicament directors de projectes de programari lliure o de codi obert, fan una tasca importantíssima tant de promoció com de suport a aquests projectes i, el més important per a mi, aporten estàndards al món de les llicències de programari lliure.

Sense aquestes fundacions seria impensable que, amb tots els components que s'han utilitzat, només hi hagin quatre llicències diferents en tot el projecte. A nivell corporatiu em sembla una tasca bàsica per promocionar l'ús del programari lliure. Sense un model de llicències mínimament unificat l'adopció de programari lliure requereix el coneixement d'un gran nombre de llicències mentre que en aquest projecte pràcticament hagués estat

possible fer l'anàlisi del model de llicències amb coneixements de la General Public License, l'Apache Software License i la Eclipse Public License.

Dins de l'experiència amb la comunitat de GWT he pogut descobrir que el màxim òrgan de govern del projecte inclou responsables de variants comercials del producte (representants de Vaadin i Sencha ocupen llocs en el comitè de coordinació del projecte) i que això, segurament redunda en una millor col·laboració entre tots els actors.

Per destacar quelcom negatiu, a part del que ja s'ha expressat sobre Cassandra, cal comentar que la tria de productes concrets de programari lliure en cada un dels casos ha costat més del que hauria de ser necessari. Projectes amb molt poques diferències però que viuen per separat (com Hudson i Jenkins), eines de suport al cicle de vida (Maven, Gradle, Ant...), diferents versions del mateix (GWT, Vaadin, GXT, SmartGWT...). En resum, que de vegades no és fàcil identificar les millors alternatives i de vegades és complicat entendre les diferències entre alguns projectes (tot i saber la història prèvia d'adquisicions com és el cas de Hudson per part d'Oracle).

## Objectius assolits i millores potencials

Els objectius personals plantejats a l'inici del projecte s'han assolit satisfactòriament. Amb aquestes paraules vull expressar la convicció de que aquest projecte ha servit per consolidar coneixements adquirits al llarg de tot el màster i per adquirir noves competències tant a nivell tècnic com a nivell de gestió en l'àmbit del programari lliure.

Si tenim en compte la planificació inicial del projecte però, és inevitable pensar en les desviacions que s'han produït en els objectius inicials. Aquestes desviacions afecten a dos àmbits principalment: la documentació per a administradors i les tasques de disseny d'arquitectura i dels components de software.

La documentació que havia de generar-se per als administradors de sistemes del client no ha arribat a temps degut a les desviacions en tasques d'auto-formació en diverses tecnologies (GWT i Cassandra principalment). Això considero que, tot i ser una manca lleu del projecte, afecta a la qualitat global de l'entrega ja que redueix la independència del client a l'hora de gestionar el producte entregat.

En relació a les tasques de disseny de l'arquitectura i dels components de software, cal comentar que aquestes tasques tot i tenir una càrrega important a l'inici de cada fase, s'han diluït en el temps en favor d'un mecanisme iteratiu molt més àgil. Realment és una desviació que, sota el meu punt de vista, ha millorat la qualitat del producte final, però no deixa de ser destacable el fet que aquestes tasques hagin pogut fer-se de forma iterativa sense afectar les dates d'entrega del projecte.

El projecte, en l'estat actual té bastants eixos sobre els que es podrien aplicar millores. Un d'ells és en l'apartat gràfic. Tot i tenir en compte la usabilitat del sistema, la manca d'un dissenyador professional d'interfícies ha afectat l'aspecte final de l'aplicació (el disseny s'ha anat fent a mesura que es desenvolupava l'aplicació i s'afegien noves funcionalitats).

Un aspecte millorable de l'aplicació relacionat amb el disseny però amb prou importància per comentar-lo de forma independent és la restricció de funcionalitats en funció del perfil



d'usuari. A dia d'avui es procedeix a desactivar certes funcionalitats per a determinats perfils d'usuari però això podria ser millorable en el futur amb nous mecanismes de bloqueig de funcionalitats.

Un altre eix de millora clara és la unificació de les comunicacions de la pàgina web i l'API. En aquests moments el mecanisme d'intercanvi entre la pàgina web i el servidor no passa pel servei web implementat. Donat que les funcionalitats d'un i altre han de coincidir, fer que la pàgina web obtingués les dades d'aquesta API enlloc d'implementar els seus propis mecanismes d'interacció amb el servidor seria un pas en la millora de la mantenibilitat del sistema.

## Referències bibliogràfiques

Pàgina web del CMMI Institute a <http://cmmiinstitute.com/>

European Cooperation for Space Standardization Standard (6 de Març de 2009).  
*ECSS-E-ST-40C*.

Hewitt, E. (Novembre de 2010). *Cassandra: The definitive Guide*, Editorial O'Reilly.

Capriolo, E. (Juliol de 2011). *Cassandra High Performance Cookbook*, Editorial Packt Publishing.

Moser, M. Obrien, T. (2011). *The Hudson Book*, Oracle Inc.

