TFC-GNU/Linux
Open Workload Scheduler

Memòria

Lluís Cozar Pérez
ETIG

Ignasi Rius Ferrer
Consultor

Data Lliurament 13/01/2013

Dedicatòria i agraïments

Al meu fill Mario, per tot. Per tot el temps que no he pogut dedicar-te com tu et mereixes. Per la teva paciència i comprensió (encara que d'algunes vegades picaves molt a la porta!). Gracies fill meu, ets el meu millor regal.

A la meva companya Isabel. No cal afegir molt més. Tretze anys d'esforç són molts, cap dels dos ha flaquejat i sempre han sigut ànims i facilitats per la teva part. Això és de tot dos, mai t'ho podré agrair. Gràcies.

A la meva família, pares, sogres, germanes, cunyats, cosins... En algun moment heu hagut de participar en aquesta aventura i us dono les gràcies també.

Molt especialment al meu pare Lluis. Després de tant de temps estic molt a prop d'aconseguir-ho. Es això que tu volies des de que era petit. Teníem pocs recursos llavors però finalment el teu fill tindrà una carrera universitària. Gracies per ser tan constant.

Resum

El present document té com objectiu detallar el procés que s'ha dut a terme per la implementació d'un projecte de gestió de la càrrega batch d'una companyia real amb un sistema d'informació heterogeni, mitjançant l'ús d'eines, llibreries i software opensource. Resultat d'aquest treball, apareixen diferents components i aplicacions que queden batejades sota el nom de OWS (Open Workload Scheduler).

Per la implementació d'aquests elements s'ha escollit Java com a llenguatge de programació, afegint l'ús de l'arquitectura J22EE mitjançant l'ús del framework Struts per a la vessant web de l'aplicació.

En primer lloc, s'ha realitzat una primera fase d'anàlisi i disseny per determinar l'arquitectura de sistemes a utilitzar, els frameworks sota els quals recolzar-nos mitjançant l'estudi de diferents alternatives, les funcionalitats que es requerien en aquesta aplicació, la definició de l'entorn de desenvolupament i altres aspectes importants dins del context de la solució.

A continuació, la fase d'implementació, ens ha permès obtenir com a productes diferents elements (scheduler, software del node/agent, API per línia de comandes i aplicació web per la gestió interactiva de la càrrega batch).

Tots aquestes fases i punts queden reflectits amb tot detall en els següents epígrafs del document.

Paraules clau: TFC, Java, J2EE, MVC, JSP, scheduler, Apache Active MQ, cron4j, MySQL, framework, opensource.

Àrea TFC: GNU/Linux

Índex de continguts

.

Índex de figures

Descripció del TFC

Estem immersos dins d'un procés recurrent d'innovació tecnològica on, cada cop més, es maneguen grans volums d'informació i on la interactivitat amb l'usuari i l'oferta d'informació, premia per sobre de qualsevol altre terme. En definitiva, l'escenari actual ens obliga a fer èmfasi en la capacitat *on-line* de les aplicacions.

Però dins d'aquest paradigma, existeix, en l'ombra, quelcom desconegut per la immensa majoria, que no parem a reflexionar que tota aquesta informació resident ens els diferents sistemes d'informació que ens envolten, requereix de manteniment, processament i estudi. I tot això, com és evident, és necessari fer-ho amb unes eines específiques que apliquin ordre i atenent a una planificació per calendari i a la vegada dinàmica.

En aquest punt, és on apareixen els sistemes de planificació de treballs, utilitzats generalment en les grans corporacions que busquen l'explotació de les dades del seu sistema informàtic amb l'objectiu d'obtenir valor de negoci de les seves dades o exclusivament un manteniment d'aquestes. En aquest cas, la diferència dels sistemes més interactius, no es premia la velocitat sinó la qualitat de la dada obtinguda. És per aquest motiu que aquestes tasques es processen en *background* o més comunament dit sota processos "batch" on no existeix interacció amb l'usuari.

El *leitmotiv* del desenvolupament d'aquest TFC rau envers a la meva vinculació professional com a responsable del producte de planificació i execució de la càrrega batch d'una companyia real. La necessitat de justificació recurrent del perquè del software utilitzat en aquesta companyia (desenvolupat a mida, amb un gestor de bbdd propietari i el qual descansa sota una arquitectura z/OS) em porta a plantejar-me la possibilitat d'avaluar que ofereixen altres arquitectures menys monopolitzades i de caràcter obert.

En aquest sentit, l'abast d'aquest treball és centra en el desenvolupament d'un projecte hipotètic de transformació d'un producte de planificació de treballs batch multiplataforma (com podria ser Control-M, Zeke, TWS o la pròpia solució feta a mida existent a la companyia referenciada), cap a solucions basades en sistemes GNU i desenvolupaments sota estàndards J2EE.

Així doncs, els objectius principals que persegueix aquest TFC són:

- La implementació d'una aplicació d'orquestració, planificació i execució de treballs multi-plataforma sota estàndards J2EE recolzant-nos en

eines, llibreries i/o frameworks opensource, que ens permetin emular les funcionalitats bàsiques d'un planificador de mercat.

- La creació de l'arquitectura de sistemes necessària pel seu funcionament, simulant l'escenari fictici d'un sistema d'informació format per 1 sistema servidor GNU/Linux i 2 sistemes esclaus amb sistemes operatius GNU/Linux i Windows respectivament (tots 3 virtualitzats).

Ambdues tasques formarien el producte entregable que estaria orientat a oferir una solució de planificació i orquestració de treballs amb costos reduïts i basats en sistemes opensource.

Els usuaris de la solució podran planificar, amb la freqüència i horaris predefinits, treballs de manteniment contra les diferents plataformes del seu sistema mitjançant funcionalitats específiques. La visualització de la planificació es realitzarà mitjançant l'aplicació web que permetrà el seu seguiment així com la realització d'operatives de re-execució, finalització o cancel·lació de treballs.

Finalment s'hauran de resoldre altres temes colaterals com la comunicació amb les plataformes on s'executaran els processos planificats, de tal manera que puguem fer transparent a l'orquestrador el sistema operatiu del sistema remot.

L'aplicació (tant la vessant web com la part més 'core') es dissenyarà e implantarà sota estàndards J2EE tal que ens permetin utilitzar els diferents frameworks i llibreries de suport existents a la xarxa. D'altra banda existeixen molts recursos documentals en relació a aquest estàndard i això afavorirà el seu reaprofitament per a la implementació de la solució.

Estat de l'art del món de la planificació

Quan parlem del món del processament batch podríem pensar, lícitament, que es tracta d'un tema dirigit a les grans organitzacions (bancs, companyies d'assegurances, governs, etc...) que requereixen del tractament de grans volums d'informació per tal d'obtenir indicadors, operatives pròpies d'aplicacions que s'han de realitzar de manera nocturna pel seu tancament, o d'altres funcions molt llunyanes de les petites i mitjanes companyies.

Però si ho mirem des d'una òptica més oberta, totes les companyies tenen necessitats de planificació de treballs. Quina entitat no requereix de planificar el seu procés de backup diari, l'enviament d'un *mailing* periòdic i/o tasques de manteniment pròpies del sistema? Cap. Això si, aquestes necessitats i en aquests escenaris, queden satisfetes per eines especialistes en d'altres d'operatives i que no fan més que encapsular el seu propi planificador per calendaritzar l'execució de les seves feines.

És evident que els clients d'aquestes solucions que mencionem són les grans corporacions. Són elles les que es plantegen l'adquisició d'un producte de mercat per tal de resoldre transversalment l'organització de les tasques en background de manera planificada, amb calendari i d'acord amb unes polítiques de priorització i paral·lelització alineades amb la potencia de càlcul del seu sistema, la disponibilitat dels seus subsistemes i de possibles acords de nivell de servei.

Per tant, podem dir que el mercat ja és madur, si tenim en compte que les arquitectures Mainframe han sigut les destinatàries d'aquestes solucions. Amb l'aparició i penetració fa ja uns anys dels sistemes opensource, les companyies de software de planificació han ampliat el seu abast, dotant de noves funcionalitats al seu executor per tal que traspassessin les antigues fronteres dels sistemes centrals.

En aquesta línia, les solucions de diferents companyies que podem mencionar com a productes amb més expansió a nivell mundial són les següents:

- **Control-M de BMC** <http://www.bmc.com/products/control-m/control-m-workload-automation.html>
- **Tivoli Workload Scheduler** (TWS o abans batejat com a OPC) de IBM <http://www-142.ibm.com/software/products/es/es/tivoworksche/>

- **CA-7 de Computer Associates**
  <http://www.ca.com/us/products/detail/CA-7-Workload-Automation.aspx.>
- **Zeke de ASG**
  <http://www.asg.com/Products/View/ASG-Zeke.aspx>
- **Opswise Automation Center de Stonebranch.**
  <http://www.stonebranch.com/products/automation-center/>
- Etc...

Totes aquestes solucions es diferencien entre elles per la quantitat d'extensions i prestacions que ofereixen en relació a la solució integral del batch (fixació de prioritats de treballs per l'acompliment dels acords de nivell de servei, connectors de diferents plataformes –Linux, Windows, SAP, etc) que fan que els diferents clients apostin per una determinada solució o un altre en funció de la complexitat del seu sistema d'informació i les seves necessitats.

Per a finalitzar amb les referències a les eines de mercat, cal dir que són productes molt complexos i que requereixen generalment de grans grups especialitzats, tant per a la incorporació dels treballs al procés de planificació com per al seguiment i manteniment de la càrrega batch que realitzen els departaments d'operació i explotació informàtica.

Si analitzem ara el segment de solucions opensource, apareixen tímidament algunes eines com SuperScheduler o Cron4j que conceptualment persegueixen els mateixos objectius que les eines de pagament i poden ser referents en aquest àmbit.

El que si podem afirmar és que existeixen diversos frameworks i/o llibreries opensource que ofereixen funcionalitats que ens doten de per si de les funcions bàsiques de planificació i que permeten utilitzar-les com a extensions a integrar dins del nostre projecte de creació d'un sistema de planificació (Quartz, JDring, MyBatchFramework, etc...).

Seran aquests últims elements els que utilitzarem al nostre TFC per tal d'implementar el nostre propi planificador multiplataforma.

Objectius generals i específics

Els objectius generals que persegueix aquest TFC són:

- Obtenir un producte de planificació de treballs multiplaforma (amb un abast concret) desenvolupat sota estàndards J2EE.

- Construcció d'un sistema d'informació virtualitzat, constituït per un servidor GNU/Linux on s'instal·larà la vessant d'orquestració i dos agents amb diferents tipologies de sistemes operatius, de tal manera que puguem certificar la integració del diferents components i el seu funcionament.

Més concretament i en aquest ordre, els objectius específics del nostre pla de treball són:

1. Avaluar els diferents frameworks i/o llibreries opensource existents (preferiblement java) que ens permetin resoldre de manera àgil les casuístiques més comunes relatives a la planificació de treballs i fer les extensions necessàries que donin cobertura a l'abast definit :

    a. Planificació i assignació de la data i hora d'execució del treball.
    b. Anul·lació de treballs ja planificats prèviament.
    c. Orquestració de la càrrega (sol·licitud d'execució).

2. Fer un estudi dels diferents frameworks J2EE opensource existents a la xarxa que ens permetin implementar una solució lleugera de "comunicació" mitjançant sockets TCP/IP per tal de resoldre, tant la vessant del servidor (atendrà la petició de solució de submit per part de l'orquestrador), com la del client (farà la sol·licitud d'execució).

3. Seleccionar l'arquitectura de sistemes a utilitzar més apropiada que permeti simular el sistema d'informació on implantar la nostra solució.

4. Implementar la solució de planificació sota estàndards J2EE i que estarà constituïda per:

    a. L 'orquestrador (el que anomenem workload scheduler). El *cor* de la solució.
    b. Els processos dinàmics de planificació i desplanificació.
    c. La interfície web per la interacció amb l'usuari relativa al seguiment de la planificació i les operatives on-line.

   d. Els servidors i client TCP/IP per la comunicació entre les plataformes.

5. La instal·lació de la infraestructura virtual que constituirà el nostre sistema d'informació:

   a. GNU/Linux on s'instal·larà l'orquestrador.
   b. GNU/Linux com a node de la xarxa.
   c. Windows 7 com a node de la xarxa.

Planificació

1. Recursos

El nombre de recursos per a la realització del projecte serà d'un FTE assumint diferents rols en funció de la fase de projecte en la que es trobi el desenvolupament del treball.

Es comptarà també amb el suport del consultor de la UOC assignat per al seguiment del TFC.

2. Calendari

Les fites definides per la UOC envers al seguiment de l'avaluació continuada atenen a les següents:

	Data inici	Data venciment
PAC-1	01/10/2012	14/10/2012
PAC-2	15/10/2012	04/11/2012
PAC-3	05/11/2012	02/12/2012
PAC-4	03/12/2012	23/12/2012
Entrega memòria final	24/12/2012	13/01/2013
Entrega vídeo presentació	14/01/2013	17/01/2013
Tribunals Virtuals	21/01/2013	25/01/2013

3. Tasques a desenvolupar

A continuació es detallen el conjunt de tasques vinculades al TFC amb les seves planificacions i esforços corresponents.

	Nombre de tarea	Duración	Comienzo	Fin
1	**TFC-GNU/Linux OWS**	**121 días**	**lun 01/10/12**	**vie 25/01/13**
2	**Pla de treball**	**14 días**	**lun 01/10/12**	**dom 14/10/12**
3	Definicio del TFC	3 días	lun 01/10/12	mié 03/10/12
4	Objectius	5 días	jue 04/10/12	lun 08/10/12
5	Planificació del projecte	4 días	mar 09/10/12	vie 12/10/12
6	Maquetació PAC-1	2 días	sáb 13/10/12	dom 14/10/12
7	*Entrega PAC-1*	0 días	dom 14/10/12	dom 14/10/12

	Nombre de tarea	Duración	Comienzo	Fin
8	**Anàlisi i disseny**	**21 días**	**lun 15/10/12**	**dom 04/11/12**
9	**Anàlisi**	**9 días**	**lun 15/10/12**	**mar 23/10/12**
10	Requeriments funcionals	3 días	lun 15/10/12	mié 17/10/12
11	Diagrama casos d'us	3 días	jue 18/10/12	sáb 20/10/12
12	**Anàlisi de tecnologies opensource**	**2 días**	**dom 21/10/12**	**lun 22/10/12**
13	Frameworks de planificació	1 día	dom 21/10/12	dom 21/10/12
14	Suport a la construcció de socket,s TCP/IP	1 día	lun 22/10/12	lun 22/10/12
15	*Revisió anàlisi amb consultor TFC*	0 días	mar 23/10/12	mar 23/10/12
16	**Disseny**	**11 días**	**mié 24/10/12**	**sáb 03/11/12**
17	Disseny entitat-relació	2 días	mié 24/10/12	jue 25/10/12
18	Disseny arquitectura aplicació	4 días	vie 26/10/12	lun 29/10/12
19	Disseny arquitectura infraestructura	2 días	mar 30/10/12	mié 31/10/12
20	*Revisió disseny amb consultor TFC*	1 día	jue 01/11/12	jue 01/11/12
21	Actualització Memòria (PAC-2)	3 días	vie 02/11/12	dom 04/11/12
22	*Entrega PAC-2*	0 días	dom 04/11/12	dom 04/11/12
23	**Implementació**	**29 días**	**lun 05/11/12**	**dom 02/12/12**
24	**Instal·lació infraestructura virtual**	**2 días**	**lun 05/11/12**	**mar 06/11/12**
25	Server (GNU/Linux)	1 día	lun 05/11/12	lun 05/11/12
26	Esclau (Ubuntu)	1 día	mar 06/11/12	mar 06/11/12
27	Esclau Windows 7	1 día	mar 06/11/12	mar 06/11/12
28	**Instal·lació entorn desenvolupament**	**2 días**	**mié 07/11/12**	**jue 08/11/12**
29	Instal·lacio software	1 día	mié 07/11/12	mié 07/11/12
30	Creació BBDD	1 día	jue 08/11/12	jue 08/11/12
31	*Checkpoint amb consultor TFC*	0 días	jue 08/11/12	jue 08/11/12
32	**Desenvolupament**	**20 días**	**vie 09/11/12**	**mié 28/11/12**
33	**Implementació workload scheduler**	**6 días**	**vie 09/11/12**	**mié 14/11/12**
34	Dispatxer	4 días	vie 09/11/12	lun 12/11/12
35	Submit	2 días	mar 13/11/12	mié 14/11/12
36	**Interface web**	**7 días**	**jue 15/11/12**	**mié 21/11/12**
37	Llistat de tasques	2 días	jue 15/11/12	vie 16/11/12
38	Criteris de selecció	2 días	sáb 17/11/12	dom 18/11/12
39	Finalització procès	1 día	lun 19/11/12	lun 19/11/12
40	Rearranc on-demand	1 día	mar 20/11/12	mar 20/11/12
41	Anul·lació	1 día	mié 21/11/12	mié 21/11/12
42	**Processos dinàmics**	**3 días**	**jue 22/11/12**	**sáb 24/11/12**
43	Planificació	2 días	jue 22/11/12	vie 23/11/12
44	Desplanificació	1 día	sáb 24/11/12	sáb 24/11/12
45	**Tractament execucions**	**4 días**	**dom 25/11/12**	**mié 28/11/12**
46	Modul server connectivitat	2 días	dom 25/11/12	lun 26/11/12
47	Mòdul client connectivitat Linux	1 día	mar 27/11/12	mar 27/11/12
48	Módul client connectivitat Windows	1 día	mié 28/11/12	mié 28/11/12
49	*Checkpoint amb consultor TFC*	0 días	mié 28/11/12	mié 28/11/12
50	Actualització Memòria (PAC-3)	4 días	jue 29/11/12	dom 02/12/12
51	*Entrega PAC-3*	0 días	dom 02/12/12	dom 02/12/12

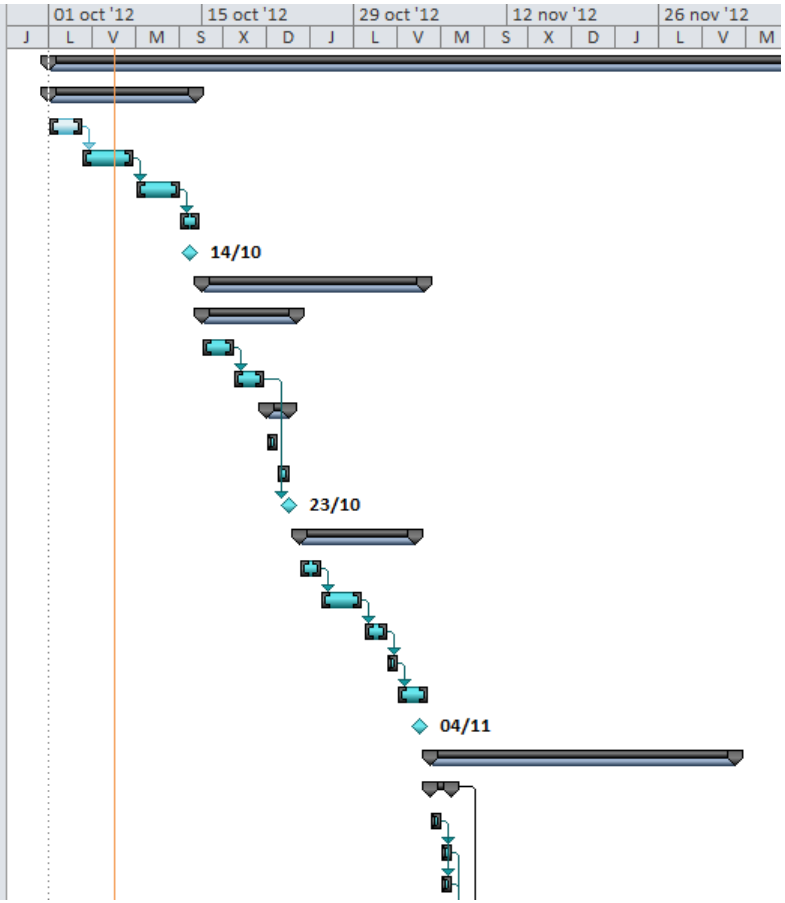
	Nombre de tarea	Duración	Comienzo	Fin
52	**Resultats i anàlisi del TFC**	**21 días**	**lun 03/12/12**	**dom 23/12/12**
53	Proves	4 días	lun 03/12/12	jue 06/12/12
54	Manual de instal·lació	2 días	vie 07/12/12	sáb 08/12/12
55	Manual d'usuari	4 días	sáb 08/12/12	mié 12/12/12
56	Actualització Memòria	4 días	jue 13/12/12	dom 16/12/12
57	*Revisió memòria amb consultor TFC*	0 días	lun 17/12/12	lun 17/12/12
58	Correccions i maquetació final memòria	6 días	mar 18/12/12	dom 23/12/12
59	*Entrega PAC-4*	0 días	dom 23/12/12	dom 23/12/12
60	**Entrega final**	**21 días**	**lun 24/12/12**	**dom 13/01/13**
61	Redacció presentació	10 días	lun 24/12/12	mié 02/01/13
62	*Revisió presentació amb consultor TFC*	0 días	mié 02/01/13	mié 02/01/13
63	Preparació entrega final	10 días	jue 03/01/13	sáb 12/01/13
64	Edició video presentació	10 días	jue 03/01/13	sáb 12/01/13
65	*Entega memòria final*	0 días	sáb 12/01/13	sáb 12/01/13
66	**Video presentació**	**4 días**	**lun 14/01/13**	**jue 17/01/13**
67	*Revisió video presentació amb consultor TFC*	0 días	lun 14/01/13	lun 14/01/13
68	Correccions video presentació	2 días	lun 14/01/13	mar 15/01/13
69	Preparació entrega video presentació	2 días	mié 16/01/13	jue 17/01/13
70	*Entrega video presentació*	0 días	jue 17/01/13	jue 17/01/13
71	Tribunals virtuals	5 días	lun 21/01/13	vie 25/01/13

Figura 1 – Tasques TFC

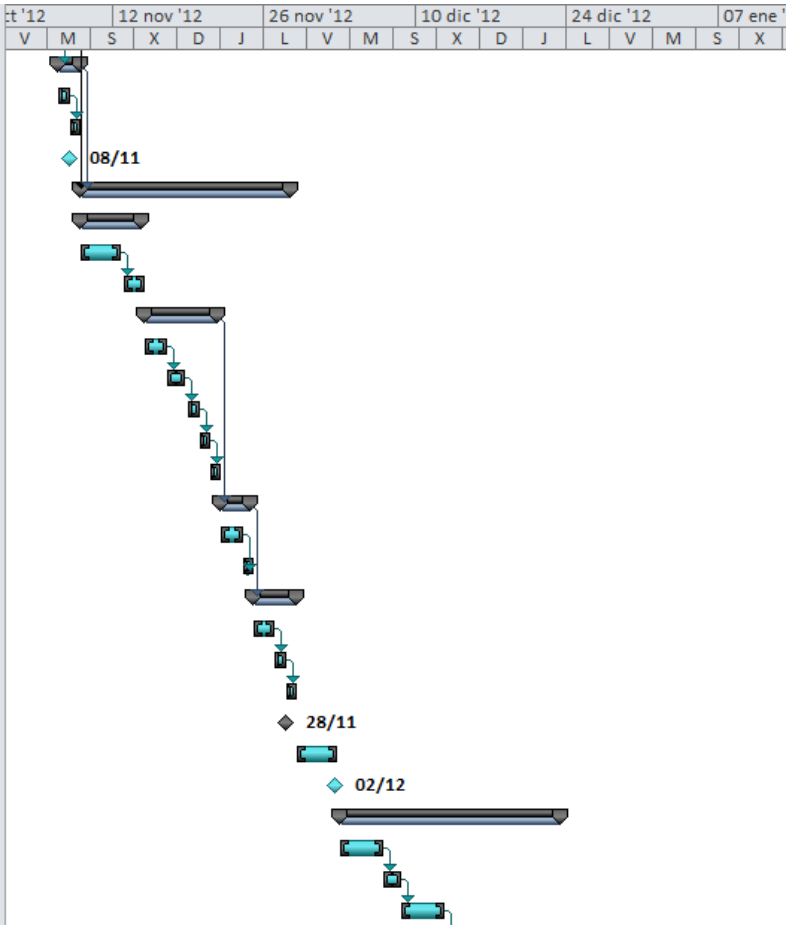
Dintre de les tasques pròpies del projecte i durant tot el cicle de vida s'han afegit diferents checkpoints amb el consultor del TFC amb l'objectiu de determinar possibles desviacions respecte a la planificació inicial establerta.

4. Diagrama de Gantt

	Nombre de tarea	Duración	Comienzo	Fin
1	**TFC-GNU/Linux OWS**	**121 días**	**lun 01/10/12**	**vie 25/01/13**
2	**Pla de treball**	**14 días**	**lun 01/10/12**	**dom 14/10/12**
3	Definicio del TFC	3 días	lun 01/10/12	mié 03/10/12
4	Objectius	5 días	jue 04/10/12	lun 08/10/12
5	Planificació del projecte	4 días	mar 09/10/12	vie 12/10/12
6	Maquetació PAC-1	2 días	sáb 13/10/12	dom 14/10/12
7	*Entrega PAC-1*	0 días	dom 14/10/12	dom 14/10/12
8	**Anàlisi i disseny**	**21 días**	**lun 15/10/12**	**dom 04/11/12**
9	**Anàlisi**	**9 días**	**lun 15/10/12**	**mar 23/10/12**
10	Requeriments funcionals	3 días	lun 15/10/12	mié 17/10/12
11	Diagrama casos d'us	3 días	jue 18/10/12	sáb 20/10/12
12	**Anàlisi de tecnologies opensource**	**2 días**	**dom 21/10/12**	**lun 22/10/12**
13	Frameworks de planificació	1 día	dom 21/10/12	dom 21/10/12
14	Suport a la construcció de socket,s TCP/IP	1 día	lun 22/10/12	lun 22/10/12
15	*Revisió anàlisi amb consultor TFC*	0 días	mar 23/10/12	mar 23/10/12
16	**Disseny**	**11 días**	**mié 24/10/12**	**sáb 03/11/12**
17	Disseny entitat-relació	2 días	mié 24/10/12	jue 25/10/12
18	Disseny arquitectura aplicació	4 días	vie 26/10/12	lun 29/10/12
19	Disseny arquitectura infraestructura	2 días	mar 30/10/12	mié 31/10/12
20	*Revisió disseny amb consultor TFC*	1 día	jue 01/11/12	jue 01/11/12
21	Actualització Memòria (PAC-2)	3 días	vie 02/11/12	dom 04/11/12
22	*Entrega PAC-2*	0 días	dom 04/11/12	dom 04/11/12
23	**Implementació**	**29 días**	**lun 05/11/12**	**dom 02/12/12**
24	**Instal·lació infraestructura virtual**	**2 días**	**lun 05/11/12**	**mar 06/11/12**
25	Server (GNU/Linux)	1 día	lun 05/11/12	lun 05/11/12
26	Esclau (Ubuntu)	1 día	mar 06/11/12	mar 06/11/12
27	Esclau Windows 7	1 día	mar 06/11/12	mar 06/11/12



	Nombre de tarea	Duración	Comienzo	Fin
28	**Instal·lació entorn desenvolupament**	**2 días**	**mié 07/11/12**	**jue 08/11/12**
29	Instal·lacio software	1 día	mié 07/11/12	mié 07/11/12
30	Creació BBDD	1 día	jue 08/11/12	jue 08/11/12
31	*Checkpoint amb consultor TFC*	0 días	jue 08/11/12	jue 08/11/12
32	**Desenvolupament**	**20 días**	**vie 09/11/12**	**mié 28/11/12**
33	**Implementació workload scheduler**	**6 días**	**vie 09/11/12**	**mié 14/11/12**
34	Dispatxer	4 días	vie 09/11/12	lun 12/11/12
35	Submit	2 días	mar 13/11/12	mié 14/11/12
36	**Interface web**	**7 días**	**jue 15/11/12**	**mié 21/11/12**
37	Llistat de tasques	2 días	jue 15/11/12	vie 16/11/12
38	Criteris de selecció	2 días	sáb 17/11/12	dom 18/11/12
39	Finalització procès	1 día	lun 19/11/12	lun 19/11/12
40	Rearranc on-demand	1 día	mar 20/11/12	mar 20/11/12
41	Anul·lació	1 día	mié 21/11/12	mié 21/11/12
42	**Processos dinàmics**	**3 días**	**jue 22/11/12**	**sáb 24/11/12**
43	Planificació	2 días	jue 22/11/12	vie 23/11/12
44	Desplanificació	1 día	sáb 24/11/12	sáb 24/11/12
45	**Tractament execucions**	**4 días**	**dom 25/11/12**	**mié 28/11/12**
46	Modul server connectivitat	2 días	dom 25/11/12	lun 26/11/12
47	Mòdul client connectivitat Linux	1 día	mar 27/11/12	mar 27/11/12
48	Módul client connectivitat Windows	1 día	mié 28/11/12	mié 28/11/12
49	*Checkpoint amb consultor TFC*	0 días	mié 28/11/12	mié 28/11/12
50	Actualització Memòria (PAC-3)	4 días	jue 29/11/12	dom 02/12/12
51	*Entrega PAC-3*	0 días	dom 02/12/12	dom 02/12/12
52	**Resultats i anàlisi del TFC**	**21 días**	**lun 03/12/12**	**dom 23/12/12**
53	Proves	4 días	lun 03/12/12	jue 06/12/12
54	Manual de instal·lació	2 días	vie 07/12/12	sáb 08/12/12
55	Manual d'usuari	4 días	sáb 08/12/12	mié 12/12/12



	Nombre de tarea	Duración	Comienzo	Fin
56	Actualització Memòria	4 días	jue 13/12/12	dom 16/12/12
57	*Revisió memòria amb consultor TFC*	0 días	lun 17/12/12	lun 17/12/12
58	Correccions i maquetació final memòria	6 días	mar 18/12/12	dom 23/12/12
59	*Entrega PAC-4*	0 días	dom 23/12/12	dom 23/12/12
60	**Entrega final**	**21 días**	**lun 24/12/12**	**dom 13/01/13**
61	Redacció presentació	10 días	lun 24/12/12	mié 02/01/13
62	*Revisió presentació amb consultor TFC*	0 días	mié 02/01/13	mié 02/01/13
63	Preparació entrega final	10 días	jue 03/01/13	sáb 12/01/13
64	Edició video presentació	10 días	jue 03/01/13	sáb 12/01/13
65	*Entega memòria final*	0 días	sáb 12/01/13	sáb 12/01/13
66	**Video presentació**	**4 días**	**lun 14/01/13**	**jue 17/01/13**
67	*Revisió video presentació amb consultor TFC*	0 días	lun 14/01/13	lun 14/01/13
68	Correccions video presentació	2 días	lun 14/01/13	mar 15/01/13
69	Preparació entrega video presentació	2 días	mié 16/01/13	jue 17/01/13
70	*Entrega video presentació*	0 días	jue 17/01/13	jue 17/01/13
71	Tribunals virtuals	5 días	lun 21/01/13	vie 25/01/13

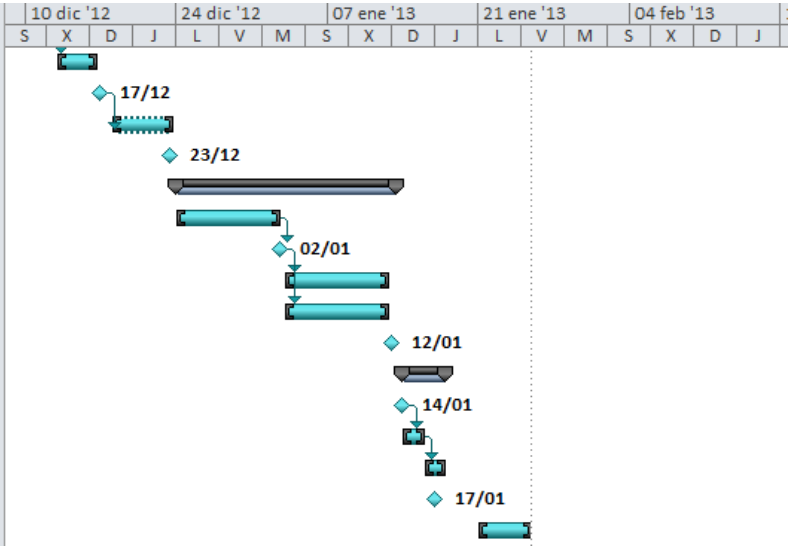


Figura 2 – Diagrama de Gantt

Anàlisi

1. Descripció

Des d'una òptica general el sistema ha de permetre automatitzar, de manera senzilla i eficient, la càrrega de processos background que apareixen de manera recurrent dins de qualsevol companyia.

L'eina, ha de dotar, mitjançant la vinculació d'un calendari a la tasca a executar, del mecanisme de planificació periòdica del treball en qualsevol node/servidor existent en el diagrama topològic del sistema d'informació de la corporació.

Així mateix, ha de disposar de funcionalitats orientades a la gestió i seguiment de la càrrega batch més pròpies d'un departament d'explotació informàtica i que facilitin operacions tals com el re-arranc, atur preventiu i execució immediata de qualsevol tasca de la planificació existent.

El següent esquema representa el context d'execució a cobrir per la solució:

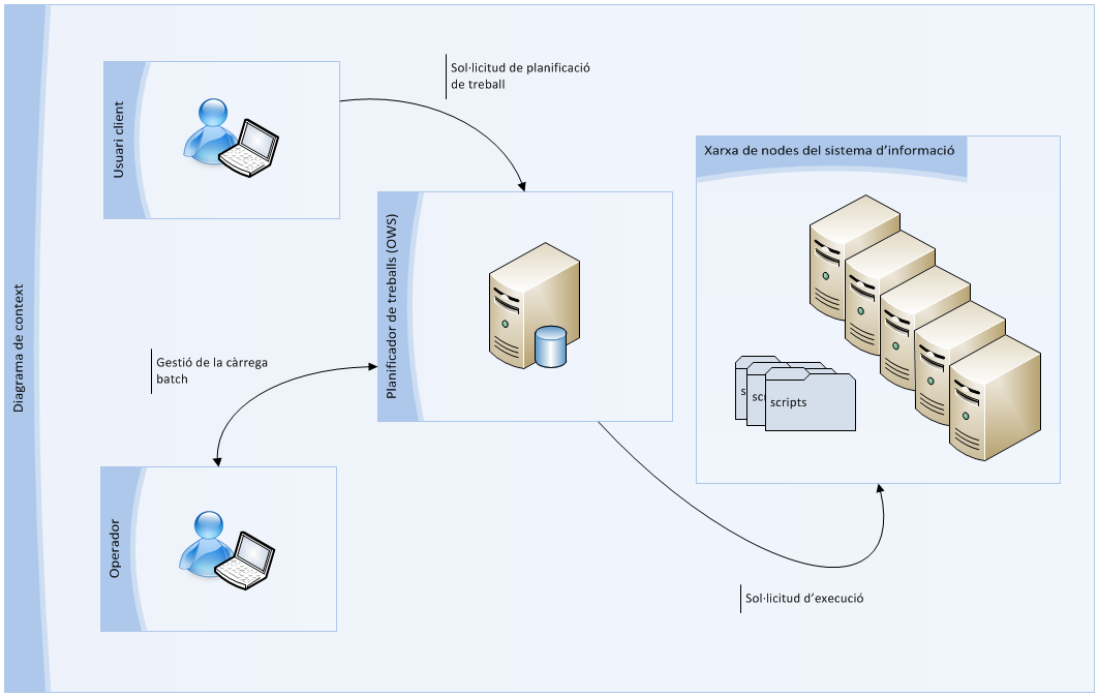


Figura 3 – Diagrama de context

2. Anàlisi funcional

En aquest apartat es descriuen el conjunt de funcionalitats que la solució OWS ha d'oferir, així com els actors que les utilitzaran.

2.1. Actors i funcions principals

A diferència d'un sistema real de planificació, on apareixen diferents actors en tot el cicle de vida, simplificarem l'escenari identificant exclusivament dos actors:

- L'usuari que assumirà les funcions de l'usuari client, el qual sol·licita la planificació del seu treball.

- L'operador de gestió del batch, que vetlla pel seguiment i interactua amb els treballs planificats mitjançant operatives definides per actuar en cas d'error o necessitat d'intervenció.

Cal dir però, que aquesta abstracció serà lògica i no serà recollida per tant a nivell d'aplicació. Ometrem doncs la gestió d'usuaris, permisos, etc... que no seran objecte d'implementació per motius de temps.

Des del punt de vista de l'usuari, les funcions proveïdes per l'eina seran:

- Procediments de gestió de la planificació

  Entenen-les com les funcionalitats que utilitzarà com a norma general l'usuari client per la incorporació d'una tasca a la planificació, la desplanificació o modificació de tasques ja existents.

- Funcions de seguiment i gestió de la càrrega batch.

  Es tracta d'aquells serveis orientats principalment per a la gestió i seguiment de la planificació que apareixen en el dia a dia de l'operador i que li permeten interactuar amb els treball planificats. S'aborden des de la llista de treballs planificats i podem esmentar els diferents canvis d'estats, simulacions, sol·licitud d'execucions, etc... com a funcions més rellevants.

  Cal dir que algunes d'aquestes funcions poden ser explotades també des de l'òptica de l'usuari client.

2.2. Casos d'ús

Seguirem les convencions i estructures establertes al "Mòdul 4. Recollida i documentació de requisits" de l'assignatura "Enginyeria del Programari I":

Per a la descripció dels diferents casos d'ús:

- Nom d'actors en **negreta.**
- Altra terminologia de l'usuari en cursiva.
- Referència a d'altres casos d'ús subratllades.

La documentació textual i estructura seguirà el següent patró:

- **Capçalera**. Número i nom del cas d'ús, resum de la funcionalitat, paper del cas d'ús dins de la solució, casos d'ús relacionats i actors.

- **Cos**. Pre-condicions i post-condicions del sistema, entrades/sortides i alternatives de procés i excepcions.

- **Final**. Hipòtesis, comentaris per clarificar les qüestions que apareguin.

2.2.1. Procediments de gestió de la planificació

Descripció dels casos d'ús

Cas d'ús numero 1 : Planificació d'un treball	
Resum de la funcionalitat	Permet a l'usuari injectar un nou treball a la planificació amb o sense un calendari específic
Paper dins el treball de l'usuari	**Acció principal que realitza el usuari**
Actors	**Usuari client**
Casos d'ús relacionats	
Precondició	El treball no s'ha planificat prèviament.
Postcondició	S'incorpora l'activitat a la planificació
L'**usuari,** mitjançant una utilitat standalone, sol·licita al sistema la incorporació d'una nova tasca a planificar informant de les dades relatives al servidor on s'executarà, el patern corresponent al calendari de planificació del treball, les opcions d'execució, la línia de comandament amb la utilitat, funció o programa a executar i finalment i de manera opcional la ràfega amb els paràmetres si escau.	
Alternatives d'excepció i excepcions	Les dades proporcionades per l'**usuari són** invàlides, incompletes o sintàcticament errònies o bé es produeix un error en la connexió i/o inserció en la BBDD.

Cas d'ús numero 2 : Desplanificació d'un treball	
Resum de la funcionalitat	Permet a l'usuari eliminar un treball de la planificació
Paper dins el treball de l'usuari	Acció esporàdica que realitza l' **usuari.**
Actors	**Usuari client**
Casos d'ús relacionats	
Precondició	El treball s'ha planificat prèviament i existeix a la BBDD.
Postcondició	S'elimina l'activitat de la planificació
L'**usuari,** mitjançant una utilitat standalone, sol·licita al sistema l'eliminació d'una tasca planificada prèviament informant exclusivament del valor que identifica unívocament un treball dins de la planificació. Abans d'executar l'operació es demanarà confirmació.	

Alternatives d'excepció i excepcions	Les dades proporcionades per l'**usuari són** invàlides, incompletes o sintàcticament errònies o bé es produeix un error en la connexió i/o esborrat del registre en la BBDD.

Cas d'ús numero 3 : Modificació del calendari d'un treball	
Resum de la funcionalitat	Permet a l'usuari modificar el calendari previ d'un treball existent a la planificació actual
Paper dins el treball de l'usuari	Acció esporàdica que realitza l' **usuari.**
Actors	**Usuari client**
Casos d'ús relacionats	
Precondició	El treball s'ha planificat prèviament i existeix a la BBDD.
Postcondició	Es modifica el calendari de l'activitat dins la planificació
L'**usuari client,** mitjançant una utilitat standalone, sol·licita al sistema la modificació d'una tasca planificada prèviament informant del valor que identifica unívocament el treball dins de la planificació així com el nou patern corresponent al nou calendari que sobreescriurà a l'actual. Abans d'executar l'operació es demanarà confirmació.	
Alternatives d'excepció i excepcions	Les dades proporcionades per l'**usuari són** invàlides, incompletes o sintàcticament errònies o bé es produeix un error en la connexió i/o esborrat del registre en la BBDD.

Diagrama dels casos d'ús

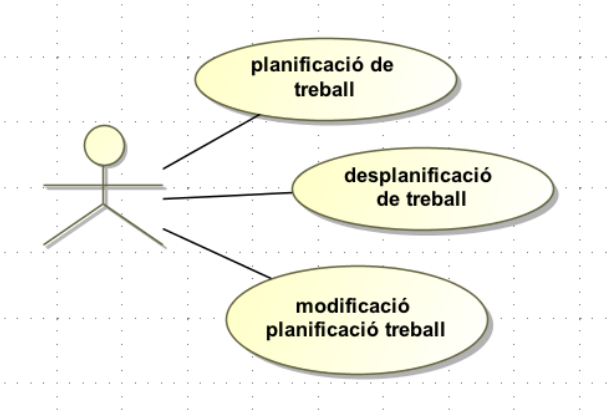


Figura 4 – Casos d'ús dels procediments de gestió de la planificació

2.2.2. Funcions de seguiment i gestió de la càrrega batch

Descripció dels casos d'ús

Cas d'ús numero 4 : Llistat de treballs planificats	
Resum de la funcionalitat	Permet a l'**usuari** llistar les tasques que ha injectat prèviament mitjançant el cas d'ús Planificació d'un treball. Es permet filtrar la vista mitjançant uns criteris de selecció que condicionaran els resultats que s'oferiran.
Paper dins el treball de l'usuari	Acció principal que realitza el **usuari**
Actors	**Operador i Usuari client**
Casos d'ús relacionats	
Precondició	
Postcondició	Es recuperen tots els treballs planificats prèviament per l'**usuari** que segueixen els criteris de selecció establerts.
L'**operador o usuari client** accedeix a l'aplicació i pot establir els criteris de selecció dels treballs a localitzar acotant els resultats a mostrar. Si no informa de cap valor es mostraran tots els treballs vigents a la planificació. Els criteris de selecció que s'identifiquen inicialment són: identificador treball, estat, data/hora injecció del treball a l'entorn, patern de planificació, servidor/ip	
Alternatives d'excepció i excepcions	Els criteris de selecció no atenen a cap registre coincident a la BBDD i per tant s'haurà de notificar a l'**usuari** o bé es produeix un error en la connexió i/o consulta a la BBDD

Cas d'ús numero 5 : Consulta de les dades generals d'una tasca	
Resum de la funcionalitat	Permet a **l'operador o usuari client** visualitzar el detall d'una tasca determinada que hagi estat donada d'alta prèviament mitjançant el cas d'ús Planificació d'un treball i a partir dels resultats obtinguts al cas d'ús Llistat dels treballs planificats.
Paper dins el treball de l'usuari	Acció principal que realitza el **usuari**
Actors	**Operador i Usuari client**
Casos d'ús relacionats	

Precondició	El treball ha estat planificat prèviament.
Postcondició	Es recuperen totes les dades registrades per a una tasca determinada.
L'**operador o usuari client** selecciona una tasca i recupera tota la informació vinculada a aquesta: identificador treball, estat actual, data/hora injecció del treball a l'entorn, patern de planificació, servidor/ip, comanda a executar, data ultima execució i últim return code.	
Alternatives d'excepció i excepcions	Es produeix un error en la connexió i/o consulta a la BBDD

Cas d'ús numero 6 : Canvi d'estat	
Resum de la funcionalitat	Permet a **l'operador** realitzar, per a una tasca registrada prèviament mitjançant el cas d'ús Planificació d'un treball, un canvi d'estat cap als diferents estadis reconeguts per l'aplicació.
Paper dins el treball de l'usuari	Acció principal que realitza el **usuari**
Actors	**Operador**
Casos d'ús relacionats	
Precondició	El treball ha estat planificat prèviament.
Postcondició	Es modifica l'estat de la tasca.
L'**operador** selecciona una tasca i informa l'estat desitjat a partir de la llista d'estats autoritzats: Fin-OK, Fin-KO, Fin-NUL, Pendent. L'estat de la tasca condicionarà el seu tractament posterior.	
Alternatives d'excepció i excepcions	Es produeix un error en la connexió i/o modificació a la BBDD

Cas d'ús numero 7 : Forçar execució immediata	
Resum de la funcionalitat	Permet a l'**operador** sol·licitar, per a una tasca registrada prèviament mitjançant el cas d'ús Planificació d'un treball, l'execució immediata del treball fent cas omís al seu calendari establert.
Paper dins el treball de l'usuari	Acció principal que realitza el **usuari**
Actors	**Operador**
Casos d'ús relacionats	

Precondició	El treball ha estat planificat prèviament.
Postcondició	Es sol·licita l'execució immediata del treball.
L'**operador** selecciona una tasca i dona l'ordre d'execució immediata del treball fent omissió del patern relatiu al calendari origen sol·licitat i/o a l'estat de la tasca. S'acompanyarà d'un canvi d'estat del treball cap a la transició tipificada com "En execució". Cal dir que aquesta operació no invalida la seva execució posterior un cop arribat el moment que estableix el calendari establert en temps de planificació.	
Alternatives d'excepció i excepcions	Es produeix un error en la connexió i/o modificació a la BBDD o bé

Cas d'ús numero 8 : Simulació calendari	
Resum de la funcionalitat	Permet a **l'operador i usuari** client sol·licitar, per a una tasca registrada prèviament mitjançant el cas d'ús Planificació d'un treball, obtenir la propera data/hora prevista d'execució atenent al calendari indicat en el instant de planificació.
Paper dins el treball de l'usuari	Acció principal que realitza el **usuari**
Actors	**Operador i Usuari client**
Casos d'ús relacionats	
Precondició	El treball ha estat planificat prèviament.
Postcondició	Es retorna la propera data/hora d'execució prevista.
L'**operador o usuari client** selecciona una tasca i sol·licita la simulació del calendari. El sistema retornà la següent data/hora prevista d'execució a partir del patern de calendari vinculat a la tasca.	
Alternatives d'excepció i excepcions	Es produeix un error en la connexió i/o consulta a la BBDD

Diagrama dels casos d'ús

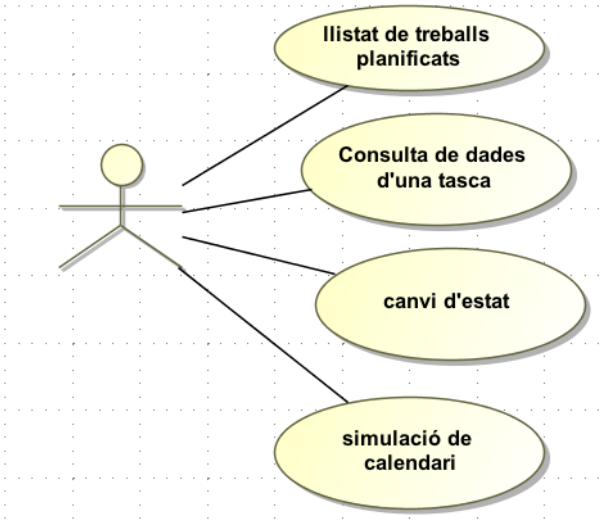


Figura 5 – Casos d'ús de les funcions de seguiment i gestió de la càrrega batch

3. Ampliació per futures versions

Aquesta versió del OWS, l'hem de veure com una versió incipient (potser fins i tot com una prova de concepte), la qual passa per alt moltes funcionalitats que podrien complimentar aquesta solució fins fer-la d'allò més complexa i robusta de cara a complir totes les expectatives que una companyia pugui tenir.

A continuació es detallen algunes línies de treball que podrien ampliar l'abast inicial de la solució en aquest sentit:

- Incorporació del concepte de prerequisits i dependències entre processos (tant lògics –procés a procés- com físics –pel fet de generar un fitxer o recurs que consumeix un altre).
- Priorització de tasques.
- Seguretat lògica.
- Identificació de recursos i serialització.
- Control de la concurrència de processos.
- Històric d'execucions.
- Estadístiques.
- Etc...

4. Anàlisi de tecnologies opensource

Per donar cobertura als diferents àmbits de la solució analitzarem diversos components ja existents a la xarxa que treballen alguns dels aspectes que perseguim en aquest TFC. D'aquesta manera podrem minimitzar el temps de desenvolupament d'algunes peces i reaprofitar les metodologies i mecanismes que aquests tipus de frameworks i llibreries ens aporten.

4.1. Frameworks de planificació

La funció més important del nostre TFC és la del propi scheduler, és a dir, la del component que, donat un conjunt de treballs, sigui capaç de donar l'ordre d'execució sol·licitada d'acord amb el calendari sol·licitat.

A continuació farem referència a dues solucions open source i a les seves característiques principals que resolen aquestes i d'altres funcionalitats relacionades.

4.1.1. Quartz

Es tracta d'un "servei" de planificació de jobs open source amb el recolzament de l'empresa nord-americana Terracota, la qual ofereix solucions Enterprise de planificació basades sota aquest component.

La versió no corporativa ofereix el producte natiu per ser utilitzat com:

- Un component incrustat en altre aplicació standalone.
- Instanciat per un application server.
- Com un component standalone e independent amb la seva pròpia JVM.
- Inclús pot ser instanciat en format clúster per dotar la solució de balanceig de càrrega i alta disponibilitat.

Com a característiques diferència ls de la versió pública podem destacar:

- Possibilitat de gestionar combinacions amb diferents directives de planificació.
- Tots els jobs s'implementen mitjançant una Java interface (Job interface) amb la possibilitat d'extendre les seves capacitats originals.
- Existència del concepte de triggers, que capturen esdeveniments quan els jobs s'inicien o finalitzen.

- La finalització dels jobs s'acompanyen amb un JobCompletionCode sobre el qual podem decidir si automatitzar el següent pas (anul·lació, rearranc, etc...)
- Persistència de la planificació i la configuració mitjançant diversos mecanismes (RAM –volàtil- o en BBDD mitjançant JDBC).
- Etc...

La solució es complementa amb d'altres serveis (aquest de pagament) com "Quartz Manager" que ofereixen una solució complerta de monitorització i control en temps real, mitjançant una interfície gràfica que permet la visibilitat de la carrega, estat i activitat.

URL : <http://www.quartz-scheduler.org/overview/features>

4.1.2. Cron4j

És un scheduler pròpiament dit per a la plataforma Java, molt similar al cron d'Unix i que pot ser integrat sota una aplicació aliena que persegueixi resoldre aspectes d'execucions recurrents de tasques.

Es tracta d'un software lliure sota llicència LGPL i, si més no, no disposa de gran publicitat, té funcionalitats simples i força interessants que, combinades entre sí, poden utilitzar-se per cobrir molts dels aspectes que apareixen en el paradigma de la gestió dels processos en background.

Com a característiques més destacables enumerem:

- Existència de components tipus ProcessTask o Task, que ens permetran executar processos de sistema directament o bé extendre la classe Java vinculada amb les funcions i operacions necessàries per satisfer els objectius més comuns.
- Patrons de planificació similars al format de cron.
- Possibilitat d'incorporar listeners per tal d'interceptar esdeveniments d'arranc, finalització OK o KO.
- Arranc manual de les tasques.
- Planificació, desplanificació i replanificació.
- Redirecció stdin, stdout, stderr cap a sortides/entrades no estàndard.
- Predicció de calendari
- Parser de patrons
- Etc...

URL : < http://www.sauronsoftware.it/projects/cron4j/>

4.2. Suport a la missatgeria

L'element diferenciador del nostre TFC en relació a les solucions de planificació descrites anteriorment és la visió multiplataforma.

Aquesta funcionalitat permet reconèixer i comunicar la xarxa de nodes del nostre sistema d'informació, de manera que, les ordres d'execució dels comandaments que emet el node on resideix el planificador de treball, traspassen les fronteres del servidor de manera transparent, fent-la arribar als diferents nodes com si la sol·licitud es realitzés en local.

4.2.1. Apache ActiveMQ

Aquesta funció la resoldrem mitjançant la utilització d'un sistema broker de missatgeria asíncrona (MOM – Message Oriented Middleware) entre aplicacions sobre la qual recolzarem les converses de la nostra xarxa de nodes reconeguts.

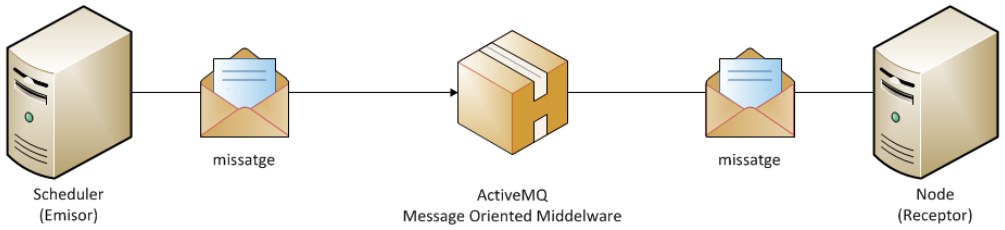


Figura 6 – Diagrama concepte Active-MQ

En aquest sentit la solució open source Apache Active MQ (URL : <http://activemq.apache.org/>) ens proporciona funcionalitats molt interesants que seran aplicables en la nostra solució:

- Comunicació asíncrona entre aplicacions considerades com a consumidores i les receptores.
- Suporta diferents llenguatges (Java, C, C++, Ruby, Perl, etc...)
- Compatible amb la majoria d'applications servers com jBoss, GlassFish o WebLogic.
- Suporta diferents protocols (TCP, HTTP, SSL, etc...).
- Disposa d'una interfície d'administració.
- Clustering, persistència, etc...

Disseny

1. Descripció

Per donar resposta a les necessitat funcionals descrites anteriorment segmentarem el problema a resoldre en quatre capes/sub-aplicacions:

- Una tasca servidora que s'encarregarà de rebre les peticions de planificació i d'executar les tasques en les dates i hores convingudes en les plataformes que formen el sistema d'informació.

- Una sèrie de funcionalitats standalone (executables per línia de comandament) que ens permetran introduir nous treballs a la planificació, amb i sense calendari, i realitzar desplanificacions de treballs ja existents.

- Una aplicació web per la visualització de la càrrega batch existent i la interacció amb certes funcionalitats per forçar la finalització i/o anul·lació dels treballs.

- Un client i diversos servidors (1 per sistema operatiu/tipus) que ens permeti sol·licitar l'execució de treballs a les diferents plataformes.

La integració d'aquestes peces donarà lloc al que anomenem OWS (Open Workload Scheduler) que es traduirà en la nostra solució opensource de planificació multi plataforma.

A continuació, analitzarem l'arquitectura d'aplicació necessària per a la implementació de les peces esmentades anteriorment així com l'arquitectura de sistemes necessària per escenificar la posta en marxa de la solució en el nostre sistema d'informació.

2. Arquitectura d'aplicació

El desenvolupament d'aquesta eina es dissenyarà i implementarà seguint les especificacions definides per la plataforma J2EE, justificant la nostra elecció mitjançant els següents arguments:

- Existència de comunitats i infinitat de recursos documentals al respecte.
- Gran varietat de frameworks i utilitats que augmenten les utilitats base de la plataforma J2EE.
- Ens permet gaudir d'un nivell d'abstracció envers a tasques de baix nivell.

- Llenguatge de programació és Java, clarament enfocat a la programació orientada a objecte i afavoreix, entre d'altres, el reaprofitament del codi font.

D'altra banda utilitzarem, per la construcció de la vessant web de la solució, el patró de disseny MVC (Model-View-Controller) que divideix les dades de l'aplicació, la interfície d'usuari i la lògica de control en tres capes totalment diferenciades, de tal forma que, davant un canvi en un component d'una capa no afecta al la resta de components que formen les altres capes.

A la següent il·lustració es representen les capes i relacions entre el model, la vista i el controlador.

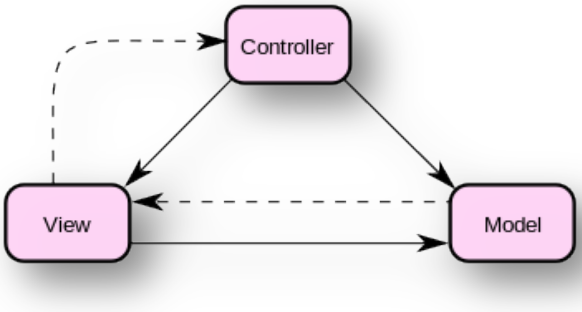


Figura 7 – Representació del model MVC

2.1. Diagrama arquitectura

El següent diagrama il·lustra l'arquitectura amb la que es portarà a terme el desenvolupament de la vessant web de l'aplicació OWS.

Ens recolzarem amb el framework Struts 2 que implementa el patró MVC :

- La vista s'implementa mitjançant la tecnologia JSP (Java Server Page) i taglibs que aporta el propi struts.

- La part del controlador es configura a partir del fitxer struts.config.xml on es coordinen les activitats a executar i es realitza la gestió d'errors.

- Finalment, la part del model és on s'implementa la lògica de negoci i és on s'accedeix a la base de dades.

Figura 8 – Diagrama arquitectura aplicació

2.2. Frameworks escollits

Els frameworks escollits per a resoldre l'arquitectura de l'aplicació són:

- **Struts**. Com a framework de distribució lliure per la Apache Software Foundation que implementa el patró d'arquitectura MVC en Java, i que simplificarà la implementació de la nostra aplicació, separant per una banda la gestió del workflow de l'aplicació, el model d'objectes de negoci i finalment la generació de la interfície d'usuari.

- **Cron4j**. Serà utilitzat per al suport de les tasques pròpies de planificació de treballs donada la seva senzillesa i la possibilitat d'incrustar-lo dins del nostre aplicatiu web per a la resolució de les tasques que es deriven envers aquest punt. La possibilitat d'interceptar els canvis d'estat i d'altres punts del cicle de vida de les tasques ens permetran donar persistència a les sol·licituds, accions, i en general, a la planificació sol·licitada.

- **Apache ActiveMQ**. Tindrà per objectiu funcionar com a broker per a la comunicació asíncrona entre el nostre scheduler i la xarxa de nodes que formen el sistema d'informació resolent les peticions d'execució de treballs així com el retorn de la resposta i stdout/err.

2.3. Entorn d'execució

A continuació detallarem el software necessari per a la definició de l'entorn d'execució sota el qual descansarà la vessant web i la persistència de la nostra solució OWS :

- El contenidor per aplicacions J2EE seleccionat serà Tomcat en la seva versió 6.0.35. S'integra perfectament amb l'entorn de desenvolupament i, al tractar-se d'un software amb un ús tant extès facilita la cerca d'informació i suport.

- Per a la persistència de dades escollirem MySQL en la serva versió Community Server 5.5.28 com a gestor de bases de dades de lliure distribució i ens recolzarem en les eines gratuïtes per la seva administració (MySQL Workbench) així com de tota la informació existent a la xarxa i del suport de tota la comunicat.

2.4. Entorn de desenvolupament

Com a eina de desenvolupament hem seleccionat Eclipse en la serva versió Eclipse Java EE IDE for Web Developers (Juno Service Release 1) considerada com a referent d'eines de desenvolupament Java i per la seva capacitat de creixement mitjançant plugins de tercers.

D'altra banda, ens hem decantat per aquesta elecció donat la seva capacitat d'integració de l'entorn de desenvolupament amb el application server Tomcat que utilitzarem també com a contenidor d'aplicacions web.

2.5. Disseny de la BBDD

Per tal de cobrir la necessitat de persistència de les dades que manega la solució de planificació es requereix una BBDD constituïda per les següents taules.

2.5.1. Taula Jobs

En aquesta taula s'emmagatzemaran, de manera unívoca, cadascun dels treballs sol·licitats per l'usuari client i que de manera conjunta formaran part de la càrrega de planificació existent.

S'enregistraran, per cadascun dels treballs, les dades de la seva planificació així com les relatives als seus canvis d'estat (transicions). Així mateix es representaran les relacions dels treballs amb la taula de nodes que representen el nostre sistema d'informació i que explotarem per la comunicació dels nodes entre la xarxa.

2.5.2. Taula Nodes

Representa els nodes existents en el nostre S.I. Per cadascun d'ells emmagatzemarem la seva descripció lògica així com la seva adreça IP, necessària pel reconeixement i enviament d'informació mitjançant la nostra xarxa Apache Active-MQ. Cada treballs es relacionaria de manera unívoca amb un node de la xarxa.

2.5.3. Taula Estado

Es tracta de la taula d'estats possibles que per els quals poden passar els treballs planificats. La relació amb el treball és 1 a n.

2.5.4. Taula Transacciones

És la taula que representa la història relativa al job en qüestió. És necessari pel seguiment i traçabilitat del cicle de vida de cadascun dels items de la nostra càrrega batch. Existiran tantes transicions vinculades amb un job com canvis d'estats s'hagin donat per aquell treball.

2.5.5. Taula Executions

És la taula que representa la sortida de les execucions (stdout/err) relatives al job en qüestió. Existiran tantes execucions vinculades amb un job com execucions consolidades s'hagin donat per aquell treball.

2.6. Model E/R

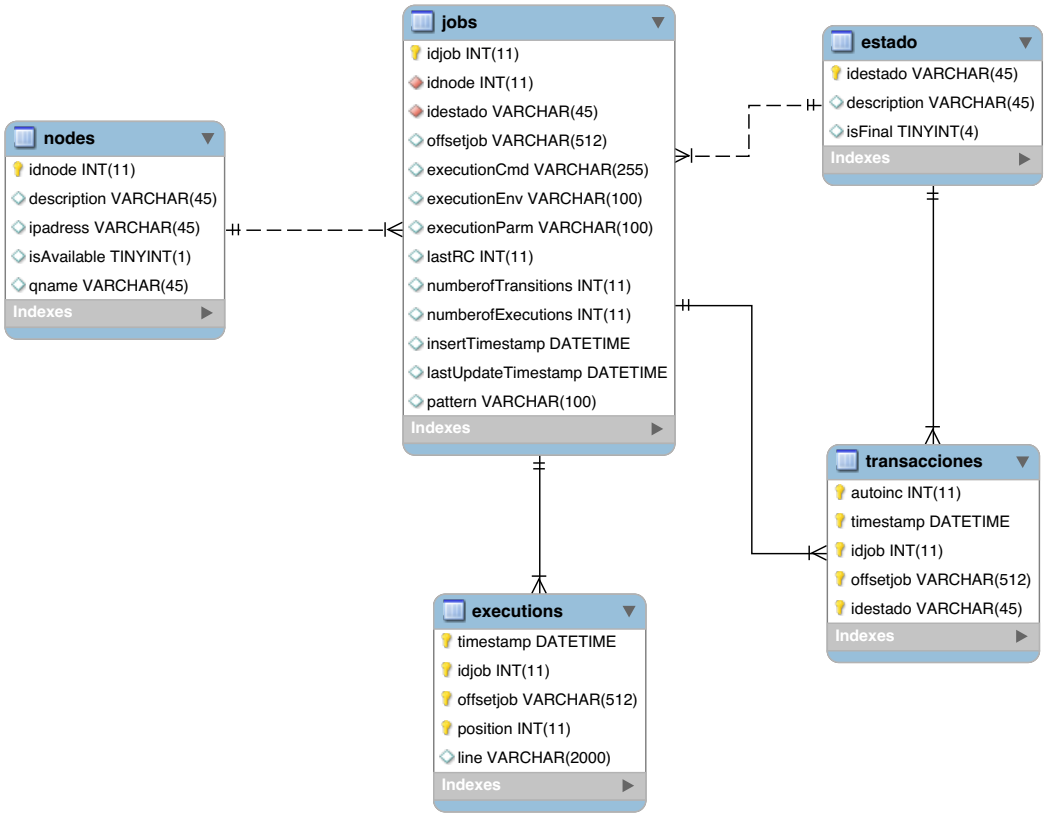


Figura 9 – Model entitat-relació

2.7. Diagrama de classes

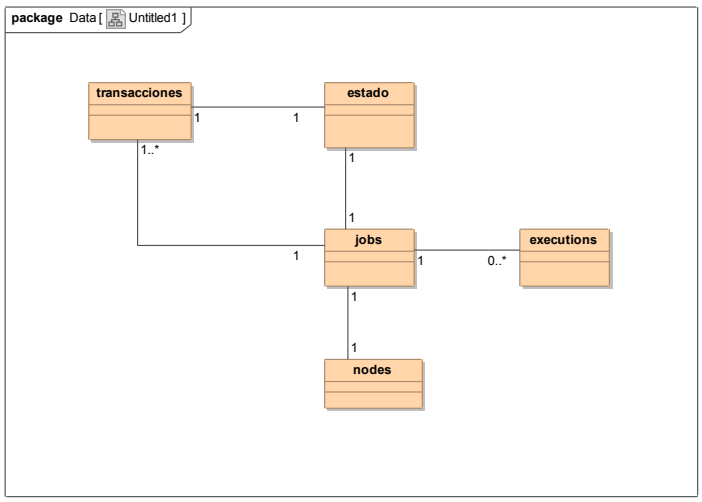


Figura 10 – Diagrama de classes

2.8. Interfície gràfica

A continuació passem a detallar el disseny client envers a la implementació de la interfície de les pantalles de l'aplicació.

Hem d'entendre aquestes propostes com a no definitives i que poden sofrir modificacions envers el disseny així com la incorporació i/o eliminació d'informació.

2.8.1. Consola principal

Les consola principal que es mostra a continuació és una aproximació de la versió final on apareixen tots els treballs que formen la planificació i des d'on l'operador podrà interactuar mitjançant les operatives descrites anteriorment.

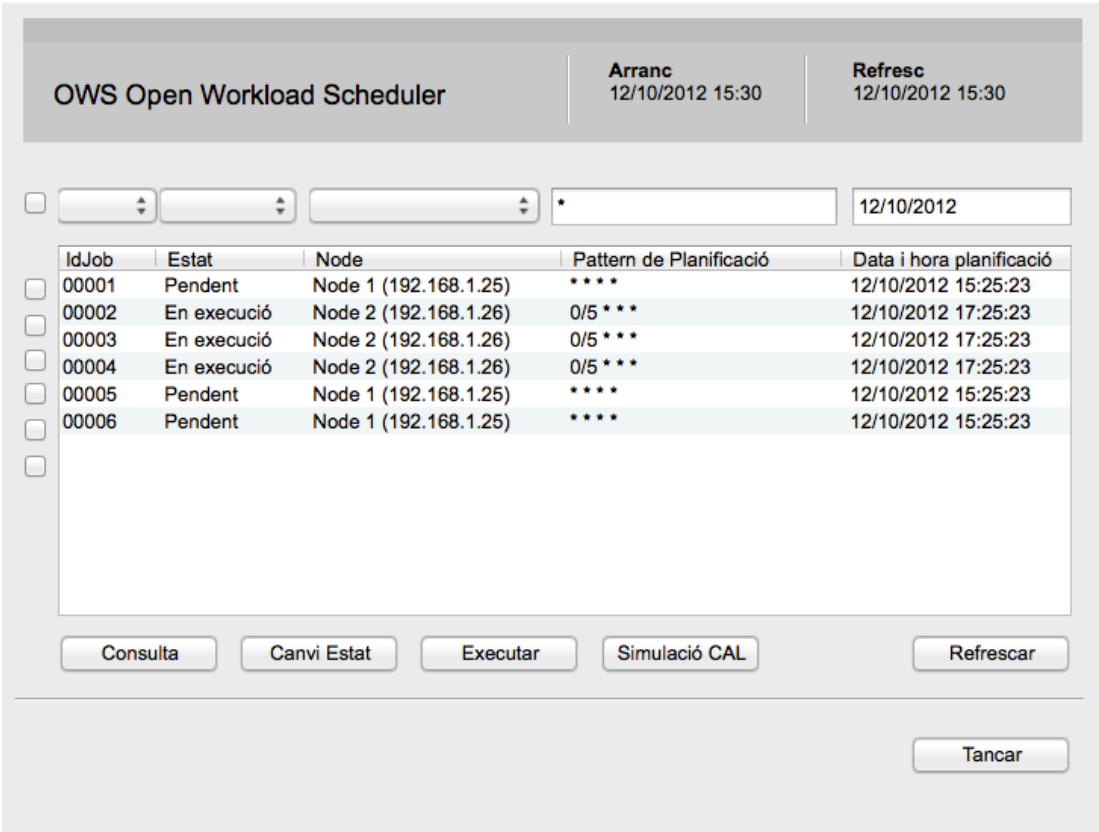


Figura 11 – Consola

2.8.2. Consulta del treball

L'operador, prèvia selecció del treball, pot consultar totes les dades relatives al treball planificat així com el detall on apareixen les diferents transicions per les quals ha passat.

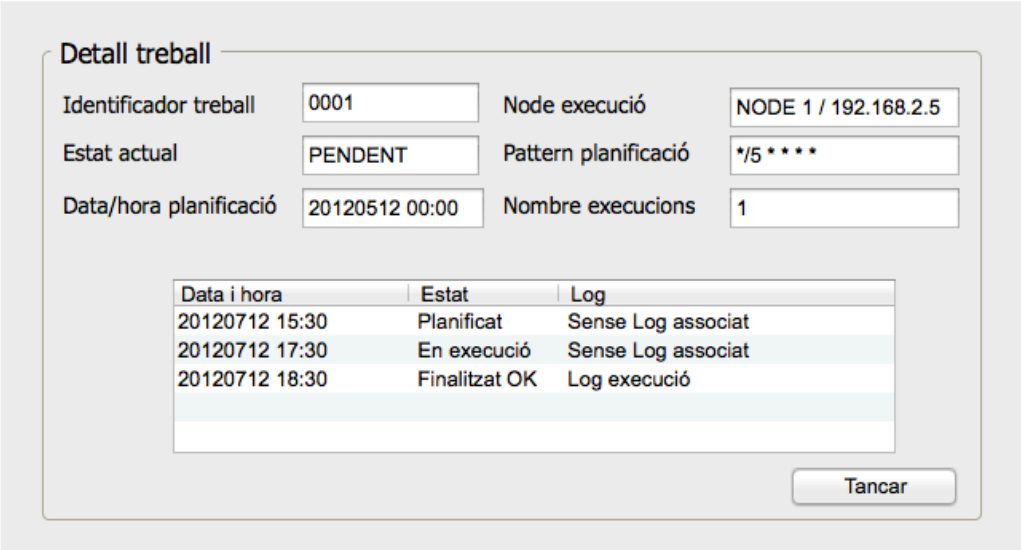


Figura 12 – Detall del treball

2.8.3. Canvi d'estat

Aquesta operativa, permetrà a l'operador, forçar un canvi d'estat davant incidents o necessitats pròpies de la gestió del batch (indisponiblitat de nodes de la xarxa, inhibició de la planificació per petició, etc...).

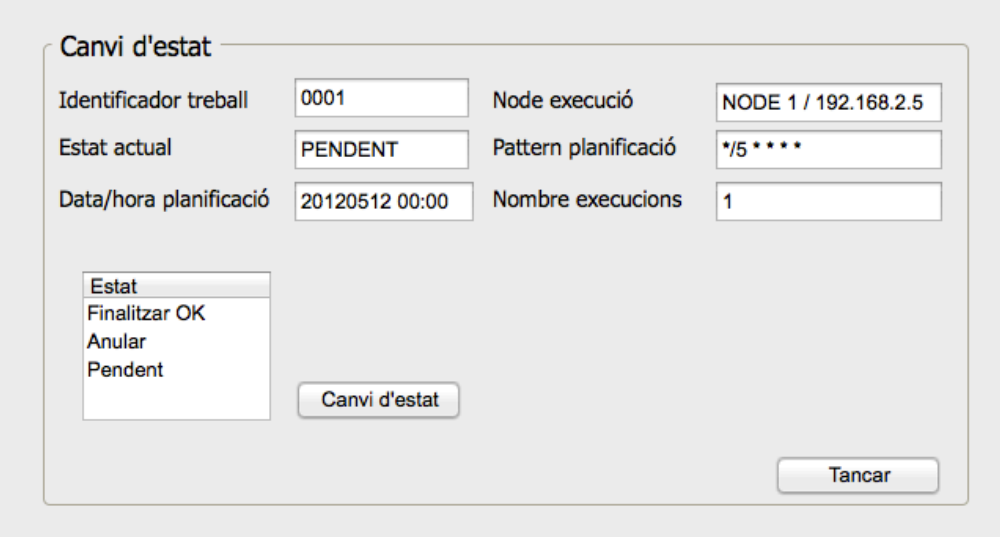


Figura 13 – Canvi d'estat

2.8.4. Simulació de calendari

L'operador podrà obtenir, en un llenguatge natural, quina és la següent data d'execució del treball d'acord amb el calendari vinculat amb aquest.

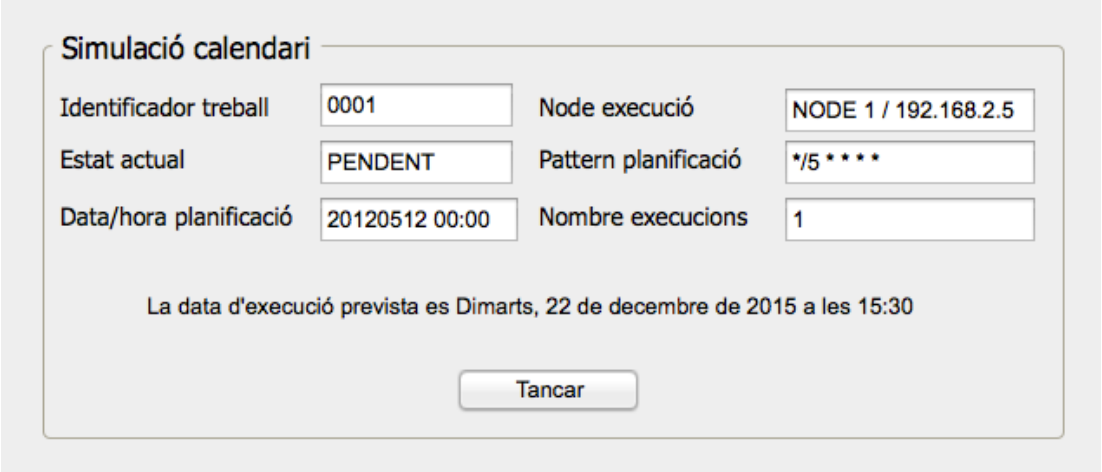


Figura 14 – Simulació de calendari

3. Arquitectura de la infraestructura

A continuació detallarem el sistema d'informació on s'implantarà la nostra solució de planificació. Es tracta d'un sistema amb pocs nodes a la xarxa, però amb diferents tipologies de sistemes operatius, on podem percebre el caire multiplataforma i el nivell d'abstracció que, en aquest sentit, ha d'oferir el programari desenvolupat.

La connectivitat entre els diferents elements implicats es realitzarà mitjançant un router tradicional amb connexió a internet recolzant-nos en DHCP per l'assignació dinàmica d'IP,s (les adreces representades són, per tant, temptatives).

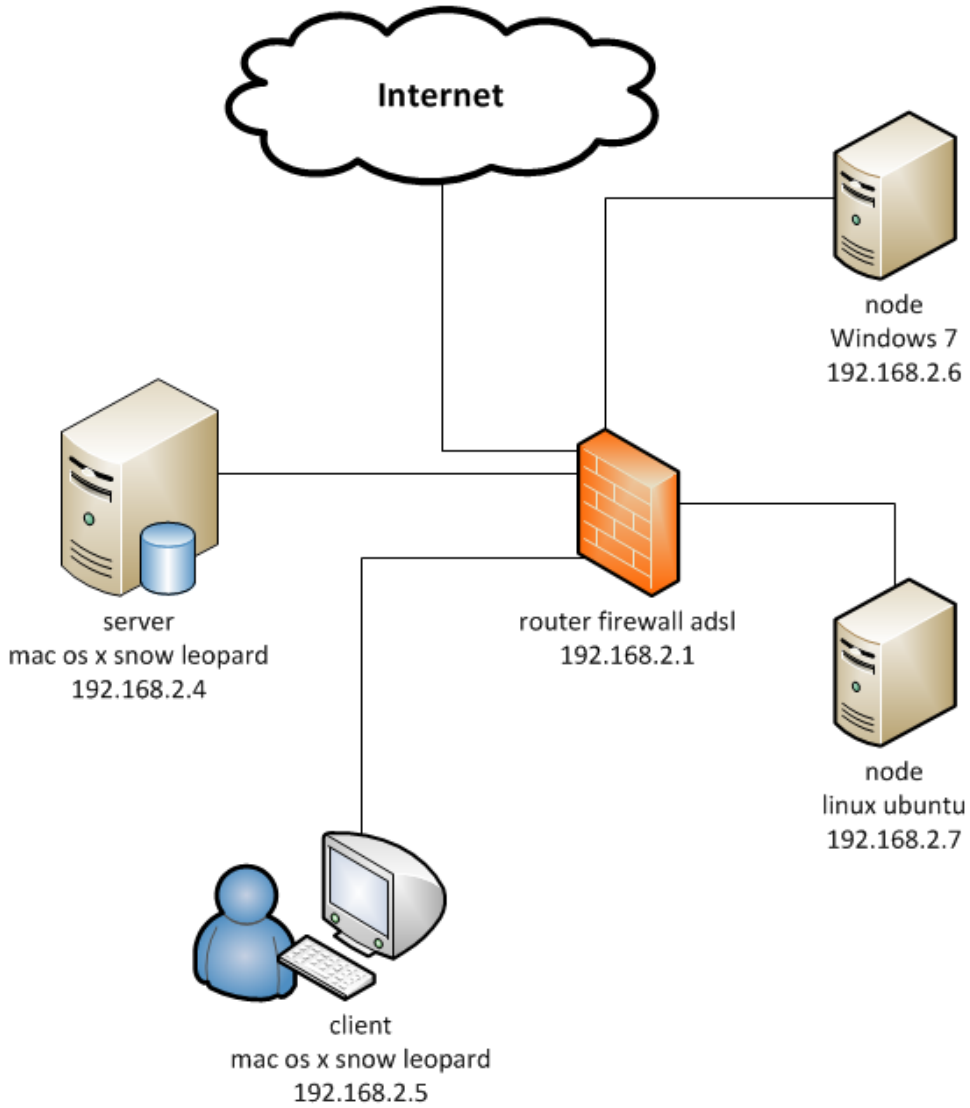


Figura 15 – Diagrama d'arquitectura

3.1. Server

Es tracta de la peça més important de la solució. En ell residirà tant el servidor d'aplicacions com el servidor de BBDD i el middleware Apache ActiveMQ necessari per la comunicació amb la resta de nodes de la xarxa.

Així mateix existirà la instància del servidor standalone de planificació i execució que consumirà la informació de la BBDD per tal d'incorporar, eliminar o desactivar qualsevol treball de la planificació en curs.

A nivell d'execució, un cop arribada l'hora d'execució per cadascun dels treballs, el server transformarà la sol·licitud en un missatge cap al node receptor de la sol·licitud, indicant el command a executar mitjançant la xarxa Apache MQ.

3.2. Client

Es tracta del node de la xarxa on es realitzaren les tasques identificades com executables per l'usuari client. Aquestes tasques es traduiran tècnicament en insercions, modificacions o eliminació del registres en la BBDD relatius a les sol·licituds de planificació realitzades.

En aquest cas, la connexió JDBC contra la BBDD des de Java és l'únic requeriment per part d'aquest node.

3.3. Node

Es tracta dels nodes "esclaus" de la solució que es troben sempre a l'espera de la recepció dels missatges via ActiveMQ amb les sol·licituds per part del node Server (considerat com l'orquestador de la solució).

Un cop arriben els missatges de sol·licitud d'execució de les tasques, aquests són interceptats i traduïts en l'execució del component sol·licitud per part del node.

Com es pot apreciar els nodes són de sistemes operatius diferents i per tant, aprofitant la universalitat de la JVM el component desenvolupat ha de ser compatible en tots ells, tot i que internament s'hagin de gestionar certes excepcionalitats vinculades amb el mètode d'interacció amb el sistema operatiu per la sol·licitud d'execució.

Implementació

1. Canvis en el disseny

Durant el procés d'implementació de l'eina han sorgit una sèrie de canvis respecte al disseny inicial que detallem a continuació. Tots aquests han sigut actualitzats als epígrafs anteriors.

1.1. Struts 2 vs Struts 1.3

En la fase de disseny vàrem seleccionar la versió 1.3 de Struts com a framework per el desenvolupament de la vessant web de la solució. Un cop arribat el moment d'implementació i d'aprofundir en la seva utilització, apreciem que, amb diferència de versions anteriors, no és permès l'ús de datasources i finalment apostem per utilitzar la versió 2 de Struts, més actual, i sota la qual hem pogut implantar l'accés a BBDD mitjançant DAOs especialitzats.

Una altre possibilitat era utilitzar nous frameworks per l'accés a BBDD (com hibernate) però que incrementaven la dedicació inicial en el desenvolupament i finalment hem descartat.

1.2. Incorporació de la taula d'execucions

Com a millora, hem incorporat una nova taula d'execucions per tal de representar la sortida estàndard (stdout i stderr) de cadascuna de les execucions dels treballs planificats. D'aquesta manera disposarem d'una historia amb el resultat de les diferents execucions realitzades en el temps.

1.3. Altres canvis

S'han reconduït altres conceptes, de menor envergadura, com el canvi de nomenclatura de certes taules i la incorporació/eliminació de claus primàries i foràanies inicialment no identificades.

Així mateix, hem variat l'enfoc de la funció de canvi d'estat "seleccionable" representada en la interfície gràfica per tal de resoldre-la com a funcions específiques en la vessant web de l'aplicació.

2. Estat de la implementació

A data d'avui, es troben implementats tots els casos d'ús definits a la fase d'anàlisi i disseny i són 100% funcionals.

No obstant, queden petits matisos a millorar envers l'entrega final relacionats amb la millora de documentació de cadascuna de les classes i mètodes implementats dins del codi que anirem resolent durant les properes setmanes.

3. Desenvolupament de la fase d'implementació

Per tal de donar forma a la nostra proposta he establert diverses línies de treball que he desenvolupat d'acord amb el que es descrivia en el pla de treball.

- Instal·lació de la infraestructura virtual i software base
- Implementació del 'core' de la solució (el motor de planificació i d'execució)
- Implementació dels procediments sobre el tractament de sol·licitud d'execucions i recuperació de les sortides (a instal·lar en els agents)
- Implementació dels processos "standalone" de planificació, desplanificació i canvi de pattern.
- Implementació de la vessant web de l'aplicació.

4. Infraestructura virtual i software base

Mitjançant el software de virtualització Parallels Desktop 7 per Mac, he configurat dues màquines de sistemes operatius Windows 7 y Linux Ubuntu respectivament.

Figura 16 – Màquines virtuals

Una tercera màquina que representarà un altre node del nostre sistema d'informació serà física amb sistema operatiu Mac OS X Snow Leopard 10.7.5 (iMac 24").

Finalment, el nostre server serà també físic amb sistema operatiu Mac OS X Snow Leopoard 10.7.5 (MacBook Pro) i actuarà també en mode agent per tal de donar més contingut i joc a la proposta.

5. Software base

A continuació detallarem el software base necessari que hem instal·lat pel funcionament de la nostra solució. Diferenciarem el software a instal·lar en la vessant que representa els nodes/agents del nostre sistema d'informació i el de la vessant del servidor on s'executarà el nostre scheduler.

5.1. Servidor

- Contenidor J2EE Apache Tomcat 6.0.36
  (URL : <http://tomcat.apache.org/download-60.cgi>)
- MySQL versió Community Server 5.5.28
  (URL : <http://www.mysql.com/downloads/mysql/>)
- Java Runtime Environement 7
  (URL :
  <http://www.oracle.com/technetwork/java/javase/downloads/index.html>)
- Apache Active MQ (URL : <http://activemq.apache.org/download.html>)

5.2. Agents

- Apache Active MQ (URL : <http://activemq.apache.org/download.html>)
- Java Runtime Environement 7
  (URL :
  <http://www.oracle.com/technetwork/java/javase/downloads/index.html>)

5.3. Consideracions

Les instal·lacions de tots els productes s'han realitzat sense cap mena de parametrització especial.

En el cas de la instal·lació del Apache MQ, hem hagut d'incorporar un arxiu JAR específic en la llibreria (ows.jar) de la nostra solució al directori lib/ d'instal·lació del producte. Aquest component conté els beans necessaris que

representen els missatges que viatgen per la xarxa MQ de manera que puguin ser reconeguts pel producte i els seus consumidors.

6. Diagrama de funcionament de la solució implementada

El següent diagrama representa les diferents peces implementades dins de la solució i que es detallaran a continuació.

El seu objectiu és dotar al lector d'una visió d'alt nivell sobre la interacció dels diferents components en el cicle de vida de la planificació, sol·licitud, execució, recuperació de sortides estàndard, gravació i visualització dels treballs planificats.

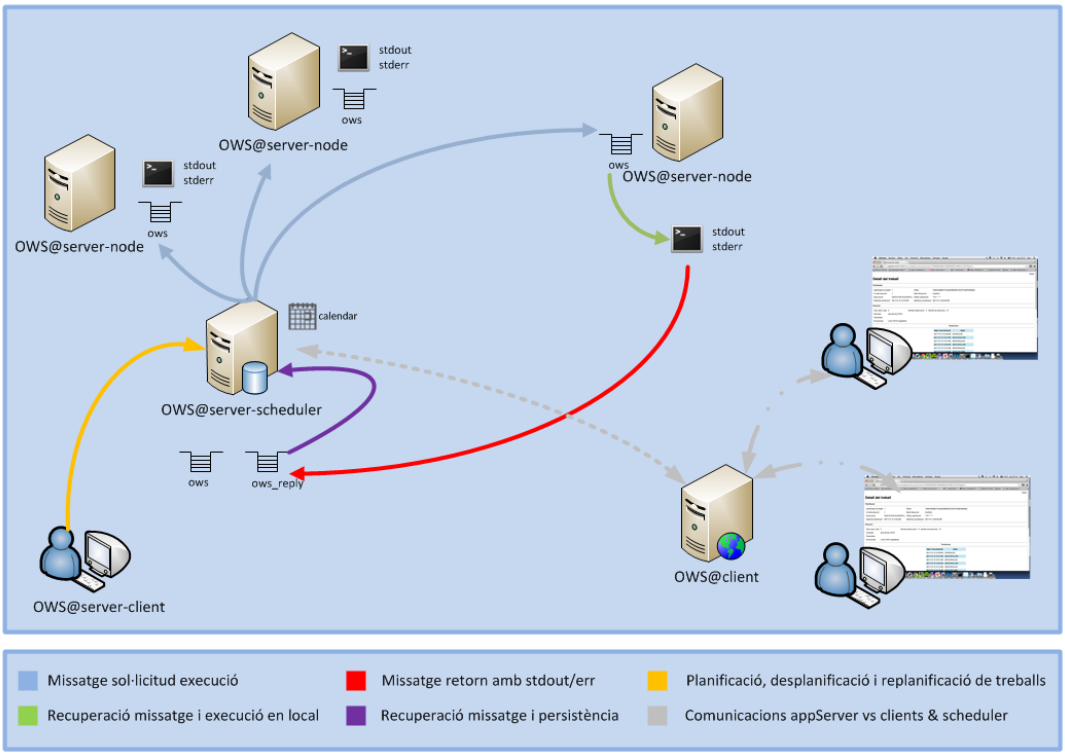


Figura 17 – Diagrama de funcionament

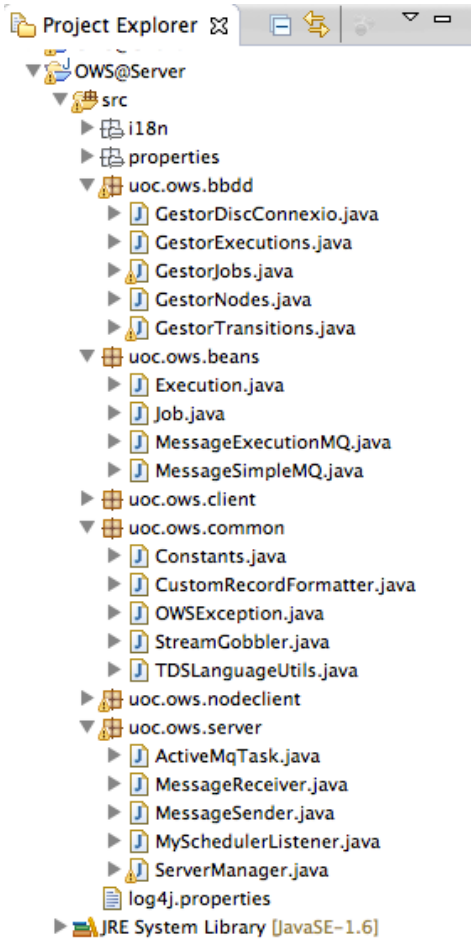
7. Desenvolupament OWS@server-scheduler

El OWS@server-scheduler es tracta de la peça 'core' de la solució que treballarà com a motor de planificació i d'execució.

Característiques generals :

- Es tracta d'una peça standalone que no requereix cap servidor o contenidor d'aplicacions.
- Descansa sota qualsevol plataforma amb JRE instal·lat (Java Runtime environament).
- Incorpora les llibreries cron4j com a recolzament pels processos de planificació, execució i intercepció d'esdeveniments dels jobs.
- Utilitza les API,s facilitades pel producte Apache Active MQ per l'emissió i recepció dels missatges.
- Per a la connexió amb la BBDD MySQL, utilitza el connector java proporcionat per oracle.

7.1. Visió d'estructura del projecte



L'estructura del projecte s'ha segregat d'acord a diferents packages orientats a resoldre diferents casuístiques.

i18n

És on ubiquem els missatges parametritzats d'acord amb l'estàndard i18n.

properties

En ells apareixen els diferents arxius de propietats necessaris per la parametrització d'accés a la BBDD (url, user/pwd) i d'altres com el fitxer utilitzat per a la injecció dinàmica de treballs.

uoc.ows.bbdd

Es tracta del Package on estan ubicats els elements on es concentra la lògica d'accés a les diferents taules que formen la BBDD. Cal dir que el GestorDiscConnexio es la superclasse a partir de la qual hereten els

diferents gestors existents per tal de potenciar la reutilització del codi.

uoc.ows.beans

És on s'ubicaran els beans necessaris en la part standalone de la solució. Cal destacar que l'exportació d'aquest Package és el que dona forma a l'arxiu ows.jar comentat prèviament i que s'incorpora a llibreria d'objectes del Apache Active MQ.

uoc.ows.common

Aquest package es pot considerar com el package de suport utilitzat des d'altres classes de la solució. En ells s'ubiquen des de classes relatives a constants generals, tractament d'excepcions, utilitats de lectura i control de missatges (reutilitzada de l'assignatura de Tècniques de desenvolupament de programari) fins a temes més específics per formateig del log o d'altres.

uoc.ows.server

L'ànima del OWS. El main ServerManager.java és el coordinador de tota la solució i es recolza en la resta de classes per determinar els canvis d'estat dels treballs, l'enviament i recuperació de missatgeria.

La resta de classes (plegades) representen els packages de la vessant del node client i/o els processos standalone que descriurem més endavant.

7.2. Cicle de funcionament i interrelació amb la resta de components

A continuació explicarem el funcionalment del cicle de treball del component ServerManager per tal d'escenificar el seu funcionament i la interrelació amb la resta de classes i serveis de la solució.

Fase d'inicialització

Aquesta fase és l'encarregada de :

- Establir la connexió inicial amb la BBDD.
- Inicialitzar el log i formatejar-lo apropiadament.
- Incorporar el component Listener al nostre scheduler. Aquest serà l'encarregat de detectar i actuar en conseqüència davant :
  - L'arranc de qualsevol tasca.
  - La fase de finalització correcta de la tasca
  - La fase de finalització failed de la tasca
- Determinar la modalitat d'arranc en funció dels paràmetres

- COLD. En fred i que suposa la inicialització prèvia de la BBDD (taules de jobs, execucions i transicions respectant les definicions d'estat i configuració dels nodes).
- HOT. En calent (defecte) i que respecta la persistència dels registres anteriors existents a la BBDD. Modificarà massivament l'estat dels jobs existents a estat PENDING per tal que siguin replanificats un cop finalitzada la fase d'inicialització.

Fase de treball

En aquest punt entrem en un 'bucle infinit' amb operacions repetitives i seqüencials en cada volta del cicle. Per evitar un consum excessiu de CPU s'incorporen 'delays' de 10 segons entre cada volta de cicle.

Cada cicle esta constituït de :

a) Subfase de gestió de jobs pendents de tractar

   En primera instància recuperem els jobs en estat PENDING per tal d'incorporar-los a la planificació. Aquestes sol·licituds apareixen a la BBDD després de l'execució de la funció de la planificació de treballs standalone o bé com a petició d'incorporació d'un treball desplanificat prèviament des de la vessant web.

   Tots els candidats s'incorporen a la planificació, canvien d'estat i passen a ser executants tan d'hora es compleixi la seva data d'execució prevista.

   Per tots aquells jobs en estat DESCHEDULING es sol·licita la desplanificació i s'inhibeixen les següents execucions. Aquests no desapareixen com a registres a la BBDD i es poden re-planificar des de la vessant web a petició.

   Els jobs en estat RESCHEDULING, passen a ser actualitzats per tal de refrescar el nou pattern de planificació indicat en la funció standalone destinada per aquesta funció.

   Finalment, els jobs en estat EXECUTE atenen a sol·licituds d'execució immediata (sense respectar el seu calendari de planificació) on el nostre scheduler rebrà l'ordre d'execució i sol·licitarà immediatament l'ordre.

b) Subfase de recuperació d'execucions/respostes dels nodes externs

En cada cicle, recuperem de la cua (ows_reply) totes les respostes i notificacions rebudes dels agents per tal d'actualitzar oportunament els resultat de les execucions així com la sortida stdout/err corresponent a cadascuna de les sol·licituds enviades prèviament.

c) Com a últim pas, generem un missatge tipus heartbeat per el control de vivència del server.

7.3. Diagrama d'estats

El següent diagrama d'estats, escenifica la màquina d'estats que implementa el planificador i pels quals pot arribat a passar un treball dins del seu cicle de vida.

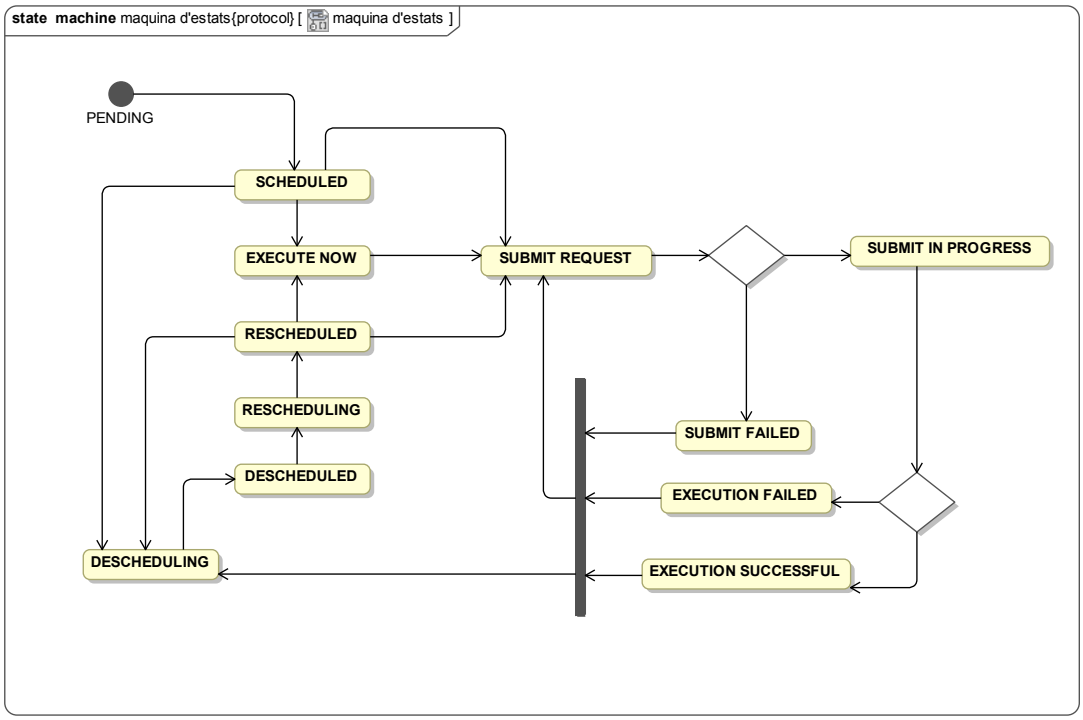


Figura 18 – Diagrama d'estats

A continuació donarem una petita descripció del significat de cada estat.

Estat	Descripció
PENDING	Estat inicial. S'ha demanat la planificació del treball i està pendent d'incorporar-lo a la planificació real.
SCHEDULED	El job es troba planificat a l'espera de data/hora d'execució.
SUBMIT REQUEST	Ha arribat l'hora d'execució del treball. Primera fase.
SUBMIT FAILED	Fase submissió del treball fallida (no s'ha pogut encuar el

	treball al node executor)
SUBMIT REQUEST	Fase de submit del treball OK (node receptor amb missatge).
EXECUTION FAILED	Execució de treball incorrecta (return code no es 0).
EXECUTION SUCCESFUL	Execució del treball OK (return code 0)
DESCHEDULING	Sol·licitada la desplanificació del treball (encara no consolidada)
DESCHEDULED	Treball desplanificat
RESCHEDULING	Sol·licita de replanificació (canvi de pattern).
RESCHEDULED	Canvi de pattern de planificació aplicat.
EXECUTE NOW	Sol·licitada l'execució immediata del treball.

7.4. Particularitats

La sol·licitud d'execució per part del planificador s'efectua mitjançant l'emissió d'un missatge asíncron cap el node involucrat. Un cop aquest el recupera dins del seu cicle, executa la funció i retorna un registre amb l'estat de la finalització i els registres stdout/err pertinents (si escau).

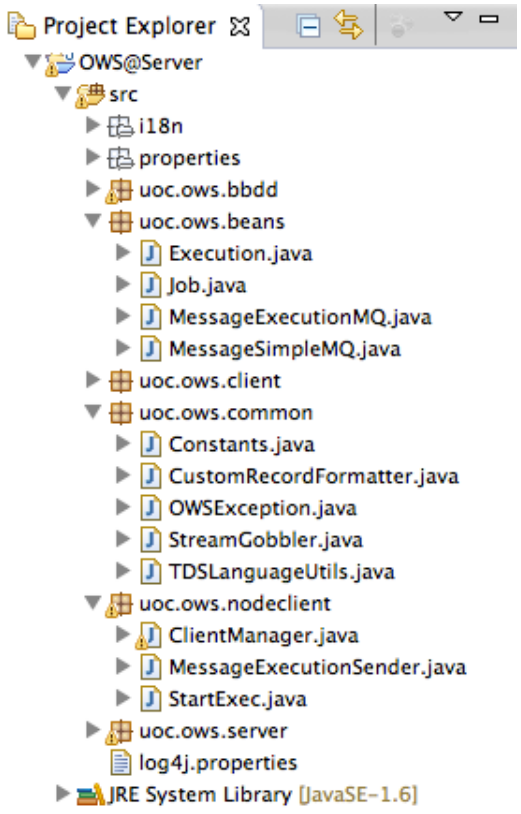
8. Desenvolupament OWS@server-node

El OWS@server-node es tracta de la peça a incorporar en cada node agent de la solució.

Característiques generals :

- Es tracta d'una peça standalone que no requereix cap servidor o contenidor d'aplicacions.
- Descansa sota qualsevol plataforma amb JRE instal·lat (Java Runtime environment).
- Utilitza les API,s facilitades pel producte Apache Active MQ per l'emissió i recepció dels missatges.

8.1. Visió d'estructura del projecte



uoc.ows.beans
Ídem que la vessant server.

uoc.ows.common
Ídem que la vessant server

uoc.ows.nodeclient
El main ClientManager.java és el coordinador de la vessant dels agents i es recolza en la resta de classes per recuperar les ordres d'execució, la gestió de l'execució així com la recuperació i enviament del stdout/err de cada sol·licitud.

8.2. Cicle de funcionament i interrelació amb la resta de components

A continuació explicarem el funcionalment del cicle de treball del component ClientManager per tal d'escenificar el seu funcionament i la interrelació amb la resta de classes i serveis de la solució.

Fase d'inicialització

Aquesta fase és l'encarregada de :

- Establir la connexió inicial i creació de sessió amb l'Apache Active MQ.
- Inicialitzar el log i formatejar-lo apropiadament.

Fase de treball

En aquest punt entrem en un 'bucle infinit' amb operacions repetitives i seqüencials en cada volta del cicle. Per evitar un consum excessiu de CPU s'incorporen 'delays' de 10 segons entre cada volta de cicle.

Cada cicle esta constituït per una única subfase de recuperació del missatge, execució del treball i enviament de resposta i stdout/err.

Per a cada missatge recuperat, s'analitza el seu contingut i es sol·licita l'execució del treball sol·licitat. Les sortides estàndards es redirigeixen oportunament i són convertides a missatges de retorn cap a la cua ows_reply existent al node manager on està ubicat el scheduler i on seran tractats oportunament en el seu cicle de treball.

El node emissor del missatge on injectar la resposta forma part del propi missatge fet que permetria la comunicació i existència de diversos planificadors sol·licitant accions contra un mateix node.

8.3. Particularitats

Com s'ha comentant anteriorment aquesta vessant requereix del jar ows.jar dins de les llibreries del Apache Active MQ per el seu correcte funcionament.

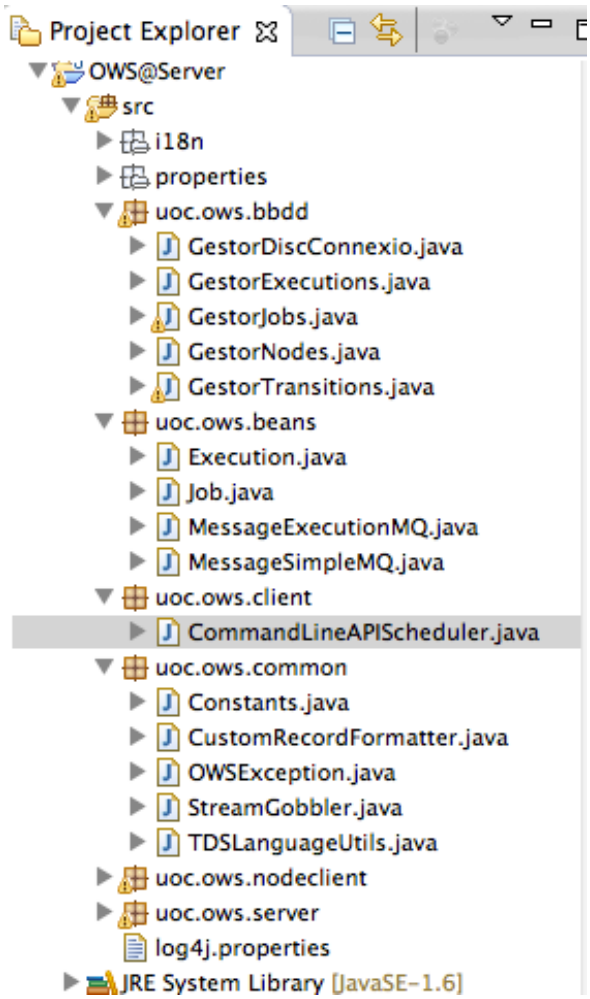
9. Desenvolupament OWS@server-client

El OWS@server-client es tracta de la API que ens permetrà injectar treballs a la planificació, desplanificar treballs existents o modificar el pattern del calendari de qualsevol treball planificat prèviament.

Característiques generals :

- Es tracta d'una peça standalone que no requereix cap servidor o contenidor d'aplicacions.
- Descansa sota qualsevol plataforma amb JRE instal·lat (Java Runtime environament).
- Per a la connexió amb la BBDD MySQL, utilitza el connector java proporcionat per oracle.

9.1. Visió d'estructura del projecte



uoc.ows.bbdd
Ídem que la vessant server.

uoc.ows.beans
Ídem que la vessant server.

uoc.ows.common
Ídem que la vessant server

uoc.ows.client
El main CommandLineAPIScheduler.java és el API estàndard per al tractament de les altes, baixes i modificacions dels treballs planificats.

9.2. Cicle de funcionament i interrelació amb la resta de components

A continuació explicarem el funcionalment del cicle de treball del component CommandLineAPIScheduler per tal d'escenificar el seu funcionament i la interrelació amb la resta de classes i serveis de la solució.

Fase principal

És l'encarregada de:

- Recuperar la modalitat d'execució
- Bifurcar cap a les funcions especifiques en funció del mode sol·licitat :
    - **Planificació del treball.** Es recolza en l'existència de l'arxiu de propietats scheduleJob.prop on s'incorporen les dades necessàries per a la planificació del treball. Un cop carregades les variables s'insereix a BBDD un registre en estat PENDING*.
    - **Desplanificació.** Es recolza en l'existència de l'arxiu de propietats scheduleJob.prop on s'incorporen les dades necessàries per a la desplanificació del treball. S'actualitza l'estat del job indicat per DESCHEDULING*
    - **Replanificació.** Es recolza en l'existència de l'arxiu de propietats scheduleJob.prop on s'incorporen les dades necessàries per a la replanificació del treball. S'actualitza l'estat del job indicat per RESCHEDULING*

*Veure OWS-scheduler-server per veure el tractament d'aquests estats.

El node emissor del missatge on injectar la resposta forma part del propi missatge fet que permetria la comunicació i existència de diversos planificadors sol·licitant accions contra un mateix node.

9.3. Particularitats

En la prova de concepte aquest component s'executa des de la infraestructura que representa el server.

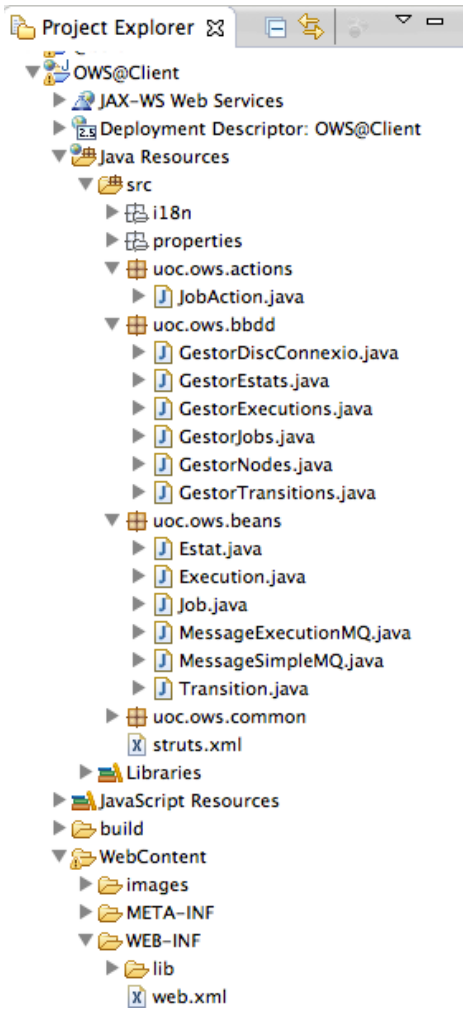
10. Desenvolupament OWS@client

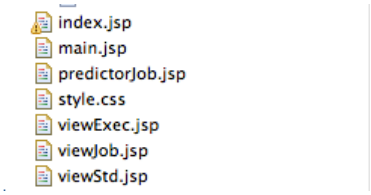
El OWS@client es tracta de la peça de la solució que dona forma a l'aplicació web amb la qual l'operador i client de la solució podrà interactuar.

Característiques generals :

- Es tracta d'una peça que requereix el servidor Tomcat com a contenidor d'aplicacions per a funcionar.
- Descansa sota qualsevol plataforma amb JRE instal·lat (Java Runtime environament).
- Per a la connexió amb la BBDD MySQL, utilitza el conector java proporcionat per oracle.
- Utilitza el framework struts com a element per a la implementació de la solució web sota estàndard MVC.

10.1. Visió d'estructura del projecte





uoc.ows.actions
Es tracta de la classe on s'implementen totes les funcions que són invocades com a actions identificades en el fitxer struts.xml on s'escenifica la seqüencia de navegació.

uoc.ows.bbdd
Ídem que la vessant server amb certes particulars i funcions especifiques en el tractament i accés a la BBDD.

uoc.ows.beans
Ídem que la vessant server.

uoc.ows.common
Ídem que la vessant server.

Els elements *.jsp són els jsp que donen lloc

a la navegació de l'eina.

10.2. Cicle de navegació

En aquest representarem el workflow que dona forma a la navegació d'aplicació per tal de comprendre les diferents funcionalitats que s'ofereixen.

Pantalla principal (main.jsp).

El que implementa aquest jsp es la pantalla principal de l'aplicació on es presenten tots els treballs planificats de manera seqüencial. Per cadascun del jobs podem interactuar amb una sèrie de funcions que ens permeten :

a) Obtenir el detall i transicions en el temps del treball en qüestió.
b) Desplanificar el treball
c) Retornar un treball a la planificació (prèviament desplanificat)
d) Sol·licitar l'execució immediata
e) Simular el calendari de les següents execucions
f) Recuperar el detall i història de les diferents execucions

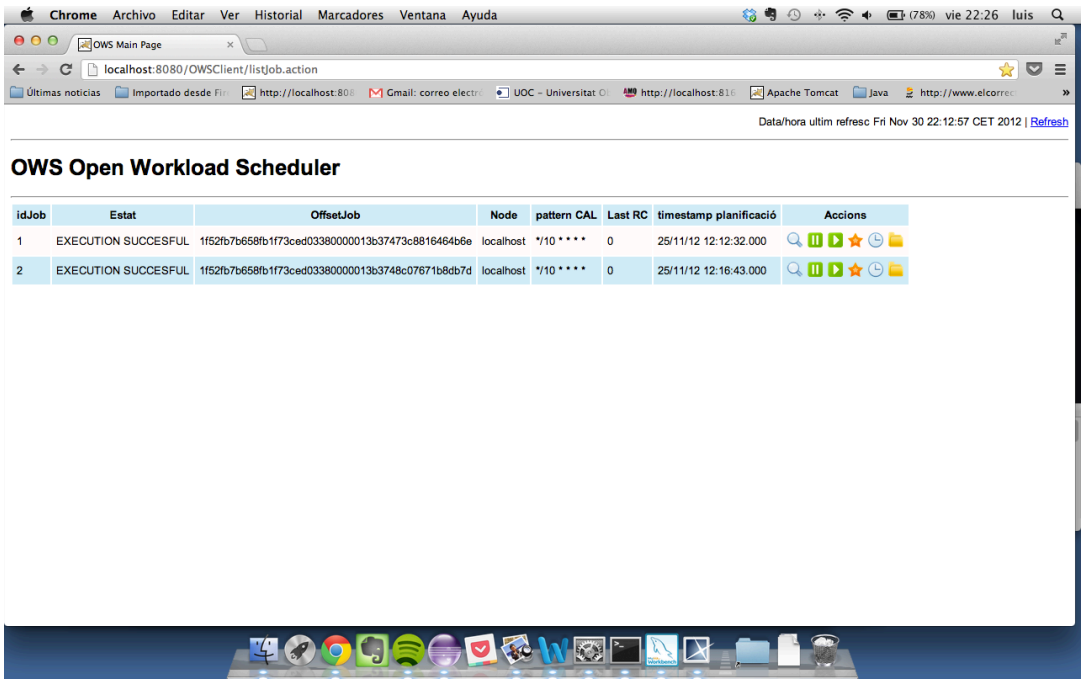


Figura 19 – Pantalla principal OWS

Detall del job seleccionat (viewJob.jsp).

Ens retorna el detall del job seleccionat junt amb la història de les seves transicions.

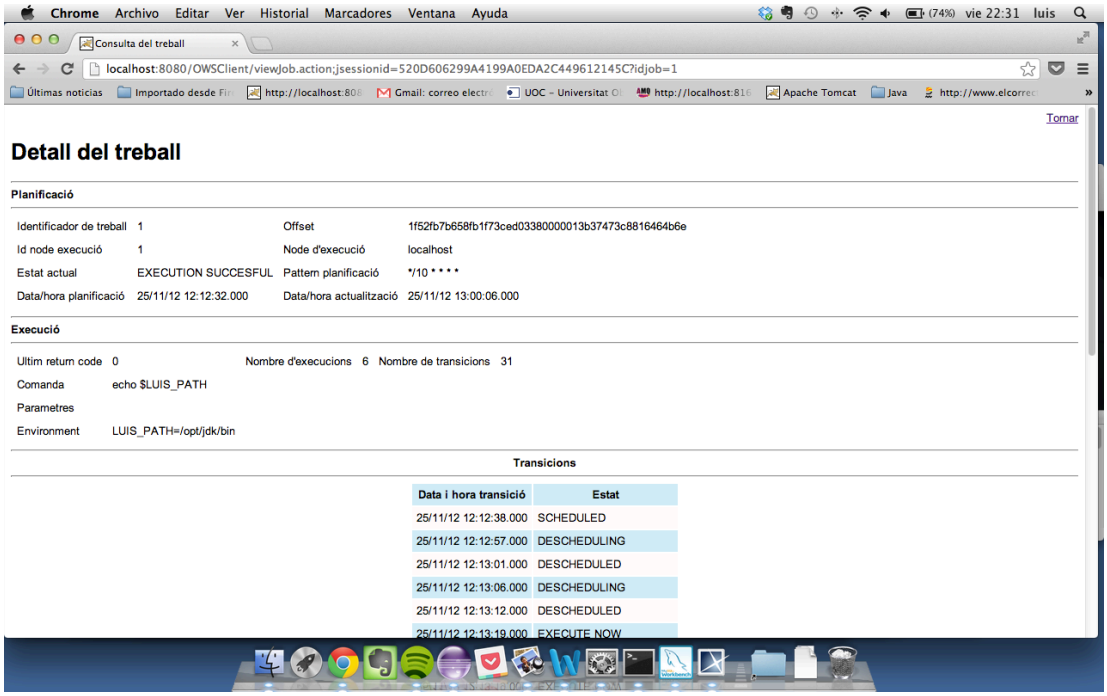


Figura 20 – Pantalla de detall OWS

Desplanificar el treball seleccionat

Provoca un canvi d'estat del job a DESCHEDULING.

Planificar el treball seleccionat

Provoca un canvi d'estat del job a PENDING.

Execució immediata del treball seleccionat

Provoca un canvi d'estat del job a EXECUTE NOW.

Simulació del calendari del job seleccionat (predictorJob.jsp).

Ens retorna el detall del job seleccionat junt amb la predicció en format textual de les següent 5 execucions previstes.

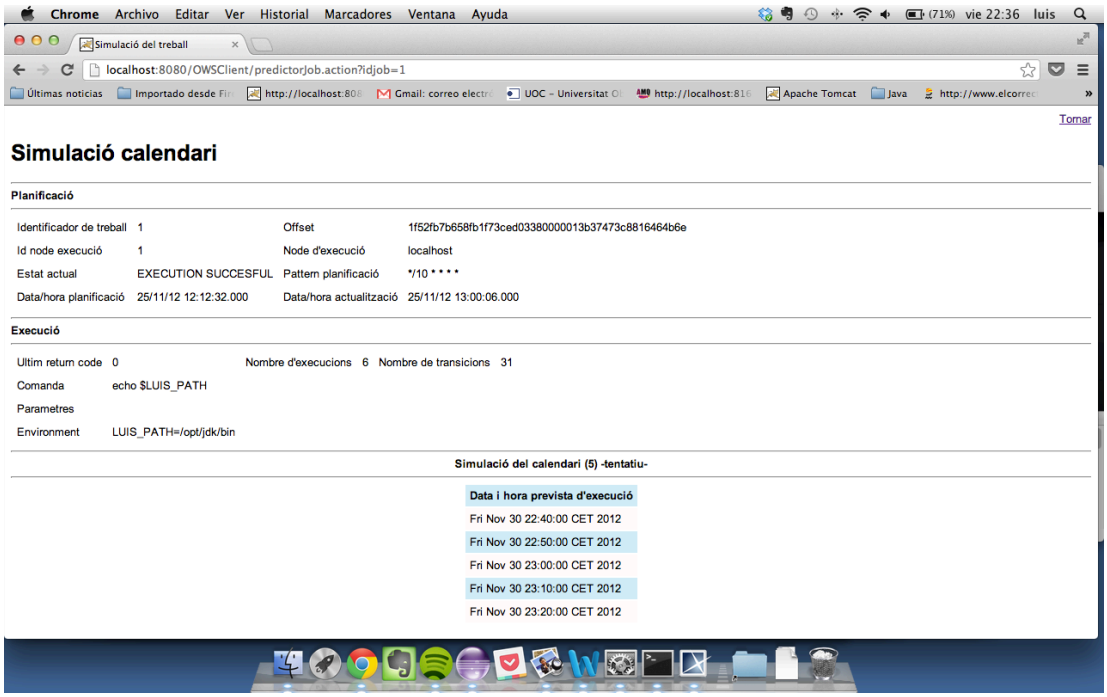


Figura 21 – Pantalla de simulació de calendari OWS

Històric d'execucions (viewExec.jsp / viewStd.jsp).

Ens retorna el detall del job i de les seves execucions. Es pot navegar finalment en profunditat per veure la sortida del stdout / stderr resultat de l'execució.

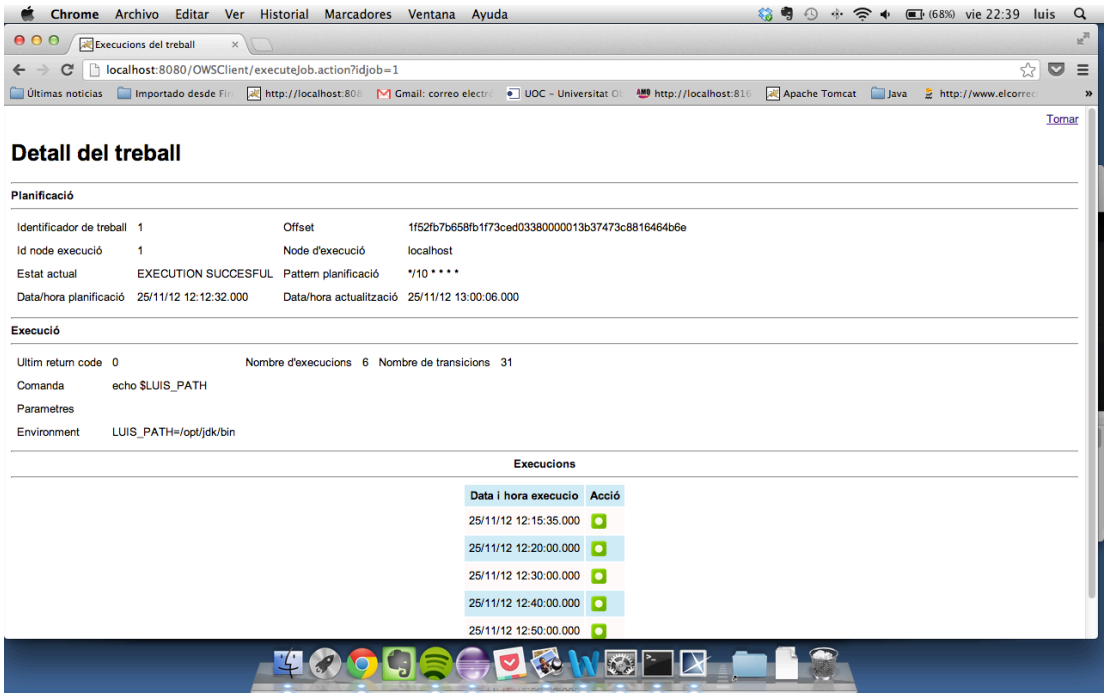


Figura 22 – Pantalla d'històric d'execucions OWS

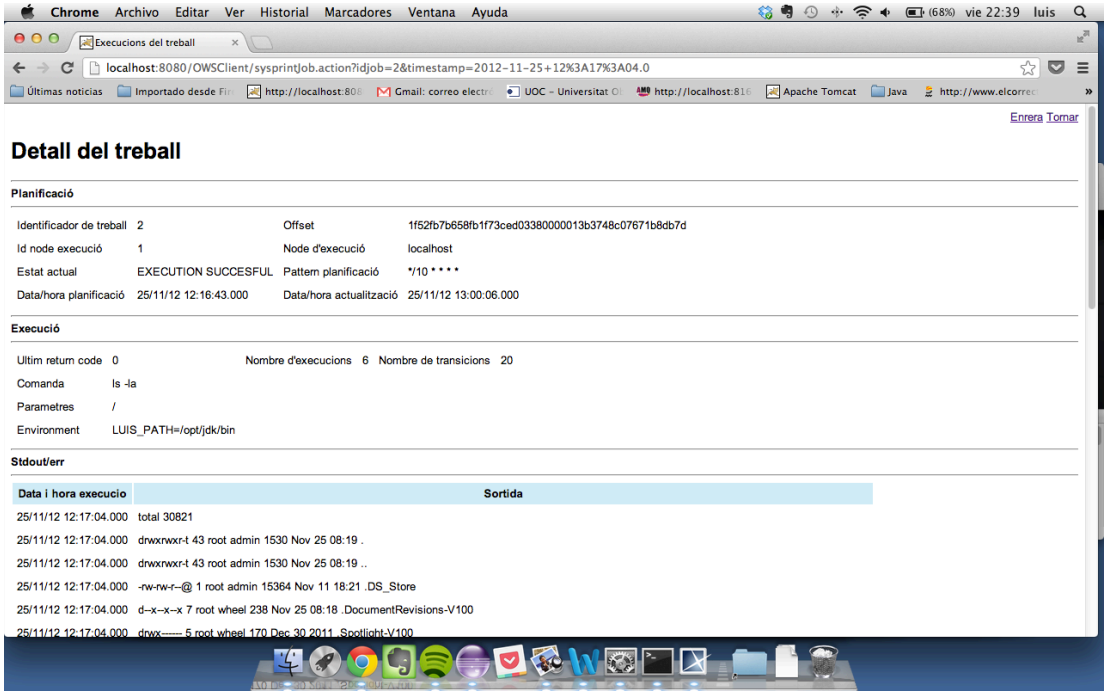


Figura 23 – Pantalla de detall de l'execució (stdout/err)

11.Problemes i lliçons apreses

Dins de tot el període de desenvolupament no m'he trobat amb una complicació tècnica excessiva en termes generals. Les dimensions de la comunitat Java i la gran quantitat d'informació envers els diferents frameworks i utilitats seleccionades existents a la xarxa han sigut de gran utilitat davant de petites dificultats o necessitat de referències.

D'altra banda, la selecció del Apache Active MQ com a MOM realment ha simplificat la missatgeria asíncrona de la solució i m'ha permès guanyar molt de temps envers a aquest tema, el que he aprofitat per a poder dedicar-hi a altres aspectes o punts que no havia previst inicialment.

Realment, tot i que he dedicat forces hores en la implementació, estic força content amb el resultat i ho considero un primer pas per a la construcció seria d'un planificador basat en plataformes distribuïdes.

Resultats i conclusions del TFC

El tret més característic del producte obtingut és, sense dubte, el seu caràcter multiplataforma i la portabilitat de solució, facilitant la seva implantació en entorns d'arquitectures heterogènies.

La seva implantació, ara que tenim el producte funcionant, podem afirmar que aportaria molts beneficis operacionals a nivell corporatiu tals com els següents:

- Autoservei en el servei de sol·licitud de planificació de treballs periòdics per part dels usuaris del sistema d'informació.
- Execució automàtica i recurrent de treballs de manera automatitzada.
- Desplanificació i replanificació de treballs existents.
- Històric de transicions de la càrrega batch.
- Històric d'execucions de qualsevol treball executat (stdout/err) de manera centralitzada.
- Seguiment on-line de càrrega de treballs.
- Execucions on-demand (a petició i fora de termini del calendari).
- Etc...

Estic convençut de la utilitat del software obtingut i dels beneficis que podrien gaudir petites i mitjanes empreses amb la seva implantació. Cal destacar però, que per motius obvis, no s'han pogut implementar totes les funcionalitats que serien desitjables i que caldria una inversió una mica més extensa per tal de gaudir d'un producte realment competitiu dins del mercat.

Des d'un punt de vista personal, l'elecció d'aquest TFC, m'ha permès reafirmar la possibilitat d'implementar una solució de planificació real i funcionant basada en diferents productes opensource, donant lloc a un producte que podria competir amb solucions de pagament existents en el mercat i orientades principalment al públic objectiu que representen les grans companyies.

D'altra banda, m'ha permès ampliar i consolidar el meu coneixement en el món Java i posar en pràctica coneixements adquirits durant la carrera relatius a l'enginyeria del programari, programació orientada a objectes i base de dades.

La elecció de la tecnologia J2EE, els diferents frameworks i el producte de comunicació asíncrona han sigut clau en el desenvolupament del projecte, molt d'ells considerats estàndards de mercat. Tots ells han afavorit el desenvolupament així com la resolució dels problemes que han aparegut durant tot el cicle de vida del projecte donada la extensa comunitat que hi dona suport.

Conclouria finalment, que entrego aquest TFC amb la percepció d'haver assolit els objectius definits al inici del treball, amb la experiència enriquidora d'haver realitzar un projecte de "transformació tecnològica" que ben bé podria ser real i amb la satisfacció de veure funcionant aquesta idea com a cloenda dels meus estudis universitaris.

Glossari

Leitmotiv. Motivació principal
FTE. Full time equivalent.
Middleware. És un software que ofereix els seus serveis desde el propi sistema operatiu.
Frameworks. Conjunt de llibreries o classes reutilitzables que implementen una estructura estàndard d'aplicació.
Utilitat Standalone. Es tracta d'un programa sense interfície gràfica i que no requereix cap altre subsistema que el propi sistema operatiu per funcionar.
JRE. Java Rutime Environment.

Bibliografia consultada

JIM KEOGH- *J2EE Manual de referencia.* McGraw Hill.

SANCHEZ ALLENDE, J., HUECAS FERNÁNDEZ-TORIBIO G., FERNÁNDEZ MANJÓN B., MORENO DÍAZ, P.- *Java 2 2ª Edición.* McGraw Hill.

CAMPDERRICH, BENET. *Enginyeria del programari; Anàlisi orientada a objectes* . UOC. (material de l'assignatura Enginyeria del Programari).

MIGUEL ARLANDY RODRIGUEZ.[online] *Introducción a Apache ActiveMQ.* http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=ActiveMQ

PHIL HANNA – *Manual de referència JSP.* McGraw Hill.

CHUCK CAVANESS – *Programming Jakarta Struts.* O'REILLY.

HERBERT SCHILDT – *The complete reference.* McGraw Hill.

Annexos

1. Guia d'instal·lació

1.1. Estructura i contingut del software entregat

El software entregat (format zip), un cop descomprimit conté la següent estructura de carpetes:

Path	Descripció
ows@jars/	Conté l'arxiu ows.jar necessari per la configuració de la instancia del Apache Active MQ.
OWS@Client/	Conté els fonts i l'arxiu war que representen la vessant web de la solució.
OWS@Server/	Conté els fonts i jars executables del software standalone, el scheduler i els software client a instal·lar en els diferents nodes.
scripts/	Scripts d'ajuda per l'arranc dels components.

1.2. Creació del schema de BBDD

Com a primer punt de la instal·lació caldrà la definició del schema de BBDD en el SGBD MySQL. Per realitzar aquesta operació es facilita l'arxiu **ows_model.sql** existent al package **OWS@server\src\sql** que ha de ser executat per un usuari amb permisos d'administrador.

Aquesta operació es pot realitzar mitjançant la línia de comandes que proporciona el propi producte MySQL o bé amb el software MySQLWorkbench descarregable des de la web http://dev.mysql.com/downloads/workbench/.

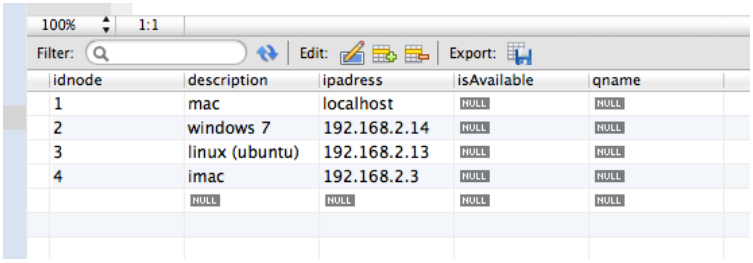
Un cop executat l'arxiu sql en qüestió apareixen les següents estructures:

- El schema ows_scheduler.
- Les taules
  - estado (amb els registres tipus pre-insertats)
  - executions (buida)
  - jobs (buida)
  - nodes (amb registres d'exemple)
  - transacciones (buida)

1.3. Configuració de la taula nodes

Cada registre de la taula nodes representa un node o servidor del nostre sistema d'informació. Abans de començar a operar és necessari representar en aquesta taula cadascun d'ells indicant :

- Description. Una petita descripció del nom del node.
- ipAdress. Amb l'adreça IP que representa el node dins de la xarxa.
- isAvailable. No cal informar-lo (valor per una possible extensió).
- Qname. No cal informar-lo (valor per una possible extensió).



idnode	description	ipadress	isAvailable	qname
1	mac	localhost	NULL	NULL
2	windows 7	192.168.2.14	NULL	NULL
3	linux (ubuntu)	192.168.2.13	NULL	NULL
4	imac	192.168.2.3	NULL	NULL
	NULL	NULL	NULL	NULL

Figura 24 – Taula nodes exemple

1.4. Configuració accés a la BBDD

Dins de l'arxiu configuration.properties existent al path **src\properties** de la carpeta OWS@server és on es troba la configuració per l'accés a la BBDD.

```
url = jdbc:mysql://localhost/ows_scheduler
schema = ows_scheduler
username = root
password =
```

on

url, correspon a la cadena de connexió a la BBDD.
schema, correspon al schema utilitzat
username, a l'usuari utilitzat per a la connexió
password, a la pwd vinculada amb l'usuari de connexió.

En cas de no configurar-se apropiadament aquests paràmetres es produiran errors en el procés de connexió a la BBDD.

1.5. Instal·lació i configuració Apache ActiveMQ

Tal i com es descriu en epígrafs anteriors únicament cal descarregar el software Apache Active MQ de la URL: <http://activemq.apache.org/download.html> i realitzar la instal·lació sense cap mena de parametrització especial en tots els nodes del sistema d'informació (nodes agents i server inclòs).

Únicament s'ha de tenir present que cal incorporar l'arxiu ows.jar existent a la carpeta ows@jars del entregable dins de la llibreria **/lib** on s'instal·la el producte per defecte.

Per finalitzar, l'arranc del producte s'ha de realitzar d'acord a les especificacions pròpies que proveeixen dins del paquet d'instal·lació del producte **(HOME_APACHEMQ/bin/activemq start**).

1.6. Arranc dels agents (OWS@server-node)

Un cop instal·lat l'Apache Active MQ, ja ens trobem en disposició d'arrancar el software a cadascun dels nodes. A la carpeta **scripts/** apareixen el arxius exemple per l'arranc de les diferents instàncies en funció del seu sistema operatiu (**scripts/node*.sh**) .

```
cd ..
cd OWS@Server
java -jar OWS@server-node.jar
```

Figura 25 – Exemple nodeMac.sh

1.7. Arranc dels scheduler (OWS@server-scheduler)

El arranc del scheduler és similar a la vessant del node. Cal arrancar en aquest cas l'arxiu **scripts/scheduler.sh** en el node de la xarxa que representi l'agent planificador.

Per defecte, l'arranc es realitza en mode COLD. Cal modificar el script referenciat si volem un arranc en modalitat HOT.

1.8. Configuració i arranc de la vessant web (OWS@client)

Prèvia a la instal·lació de la vessant web de l'aplicació, cal configurar l'accés a la base de dades tal i com es descriu a l'epígraf "Configuració Accés a la

BBDD" d'aquesta guia d'instal·lació. En aquest cas, l'arxiu de configuració resideix a l'arxiu **Java Resources/src/properties/configuration.properties** del Web Archive OWS@Client.war existent al path del entegrable **OWS@Client**.

Un cop realitzada aquesta acció, es pot realitzar el *deploy* de l'aplicació en el contenidor d'aplicacions J2EE Tomcat. Cal copiar doncs, l'arxiu OWS@Client.war en el HOME_TOMCAT/webapps on HOME_TOMCAT correspon al directori d'instal·lació del producte. Tot seguit cal iniciar Tomcat i ja podrem accedir a la vessant web de l'aplicació (URL: <http://localhost:8080/OWSClient/listJob.action>)*[1].

[1] Cal incorporar Jobs a la planificació prèviament per a la visualització de la llista.

2. Guia d'usuari

2.1. Planificació de treballs

La utilitat executable per a la planificació d'un treball atén al script **scripts/cmdLineAPIscheduler.sh**

En aquest cas cal tenir cura dels paràmetres a informar en funció de l'acció, tant en la vessant d'invocació, com dels paràmetres de l'arxiu de propietats.

Aquesta modalitat atén al següent format d'invocació

```
# scheduling jobs
java -jar OWS@server-client.jar sch
/src/properties/schedulejob.prop
```

on

sch, correspon a la modalitat de "planificació"
/src/properties/scheduleJob.prop, correspon a l'arxiu de propietats necessari per a la planificació del treball.

Exemple invocació entorn Windows

```
node = 10.211.55.3
executionCmd = cmd.exe /C dir C:\
executionEnv =
executionParms =
pattern = */10 * * * *
```

on

node, correspon al node de la xarxa on executar el treball (ha d'existir a la taula de nodes).
executionCmd, el comandament a executar en la plataforma corresponent al node.
executionEnv, paràmetres d'entorn
executionParms, paràmetres de la invocació
pattern, calendari de planificació en format "cron"

2.2. Desplanificació de treballs

La utilitat executable per a la desplanificació d'un treball atén al script **scripts/cmdLineAPIscheduler.sh**

En aquest cas cal tenir cura dels paràmetres a informar en funció de l'acció, tant en la vessant d'invocació, com dels paràmetres de l'arxiu de propietats.

Aquesta modalitat atén al següent format d'invocació

```
# scheduling jobs
java -jar OWS@server-client.jar desch
/src/properties/schedulejob.prop
```

on

desch, correspon a la modalitat de "desplanificació"
/src/properties/scheduleJob.prop, correspon a l'arxiu de propietats necessari per a la desplanificació del treball.

```
Idjob = 1
```

on

idjob, correspon al identificador del job a desplanificar.

2.3. Modificació del calendari d'un treball

La utilitat executable per a la modificació del calendari d'un treball atén al script **scripts/cmdLineAPIscheduler.sh**

En aquest cas cal tenir cura dels paràmetres a informar en funció de l'acció, tant en la vessant d'invocació, com dels paràmetres de l'arxiu de propietats.

Aquesta modalitat atén al següent format d'invocació

```
# scheduling jobs
java -jar OWS@server-client.jar modify
/src/properties/schedulejob.prop
```

on

modify, correspon a la modalitat de “modificació del calendari”
/src/properties/scheduleJob.prop, correspon a l’arxiu de propietats necessari per a la modificació del calendari del treball.

```
Idjob = 1
pattern = */5 * * * *
```

on

idjob, correspon al identificador del job a desplanificar.
pattern, calendari de planificació (nou) en format “cron”

2.4. OWS@client – Menú principal

L’accés a l’aplicació es realitza mitjançant la següent URL: <http://localhost:8080/OWSClient/listJob.action> que dona lloc a la pantalla principal.

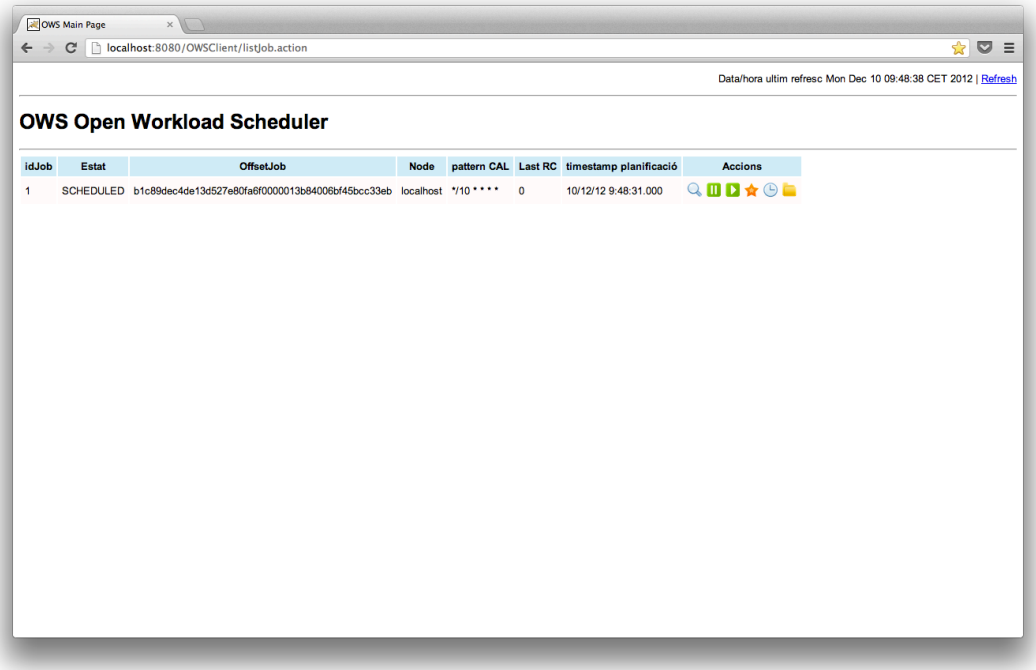


Figura 26 – Menú principal

on apareixen els treballs planificats amb informació bàsica de primer nivell :

- **Idjob**. Identificador numèric del job (únic i irrepetible).

- **Offsetjob**. Identificador intern del treball dins del scheduler.
- **Node**. Ipadress/dns on s'executa el job planificat.
- **Pattern CAL**. Patró de calendari en format cron vinculat amb el treball.
- **Timesteamp de planificació**. Timestamp d'incorporació a la cartera de jobs planificats.

Com es pot apreciar, en últim lloc, apareix una botonera d'accions a la dreta que dona accés a les següents operatives:

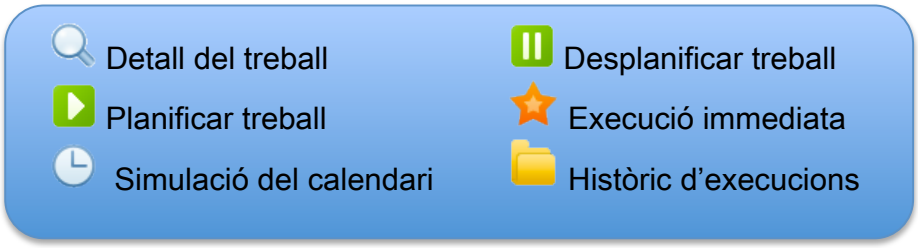


Figura 27 – Accions

2.5. OWS@client – Detall del treball

Dins d'aquesta operativa es poden apreciar en detall total els atributs i valors vinculats amb el treball planificat tant en termes relatius de la pròpia planificació (identificador, estats, timestamps, etc...), com en termes d'execució (últim return code, nombre d'execucions, transicions, comandaments, paràmetres, etc...).

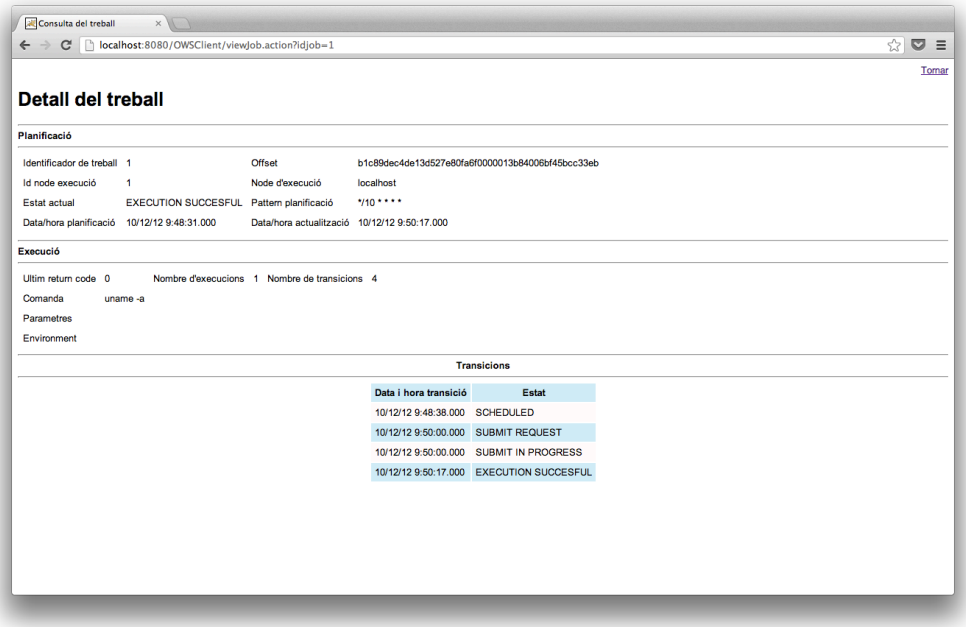


Figura 28 – Detall del treball

Finalment, es pot apreciar una taula de transicions amb la cronologia dels canvis d'estat del treball.

2.6. OWS@client – Desplanificar el treball

Aquesta operativa, similar a la que s'ofereix des de línia de comandaments, permet desplanificar un treball planificat prèviament.

La seva execució comporta el canvi d'estat del treball a DESCHEDULING que serà interceptat pel scheduler i li donarà curs, deixant el treball desplanificat temporalment.

2.7. OWS@client – Planificar el treball

Facilitem, amb aquesta funció, la replanificació d'un job prèviament desplanificat, incorporant-lo de nou a la cartera de jobs vigents.

La seva execució comporta el canvi d'estat del treball en estat DESCHEDULED a RESCHEDULING que serà interceptat pel planificador que li donarà curs, deixant el treball en estat planificat.

2.8. OWS@client – Execució immediata del treball

La necessitat d'execució immediata d'un treball sense tenir en compte el seu pattern de calendari queda resolta amb aquesta funció.

La seva execució comporta el canvi d'estat del treball a EXECUTE NOW que serà interceptat pel planificador que donarà curs a la seva execució de manera immediata.

2.9. OWS@client – Simulació del calendari del treball

Tal i com es pot apreciar, el pattern de calendari de cada treball està expressat en format "cron".

Si volem conèixer en un llenguatge natural quines seran les següents previsions d'execucions, es pot realitzar mitjançant aquesta funció que oferirà una simulació de la data/hora de les següents 5 execucions previstes.

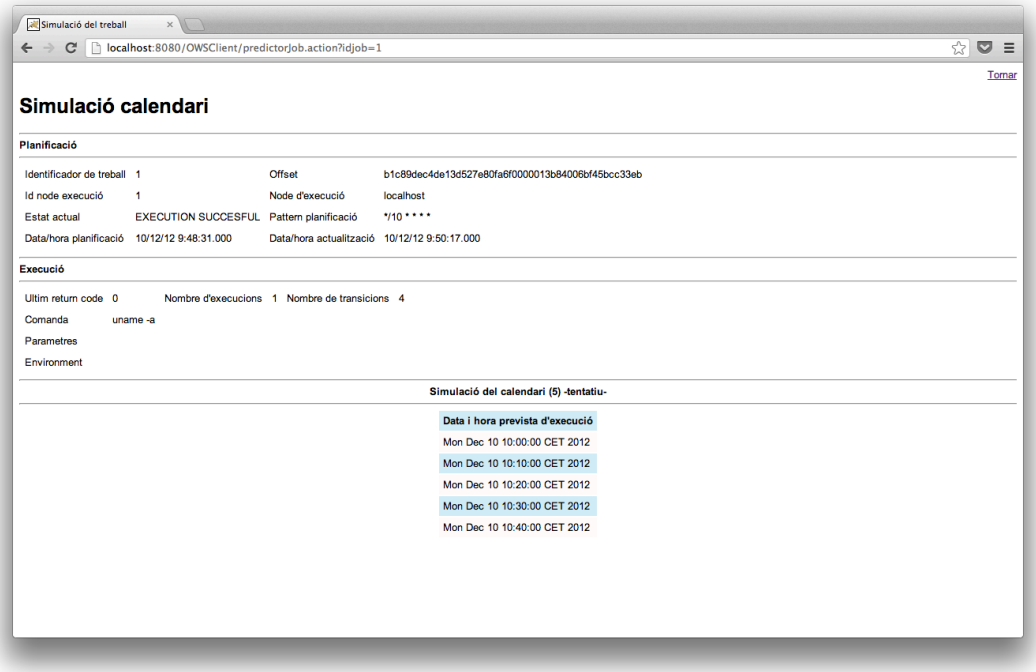


Figura 29 – Simulació del calendari

2.10. OWS@client – Històric d'execucions

La funcionalitat d'històric d'execucions permet, per a cada execució consolidada en el temps, veure el seu timestamp d'execució així com el detall propi de la sortida del component executat i que, generalment, queda reflectit en les sortides estàndard stdout i stderr.

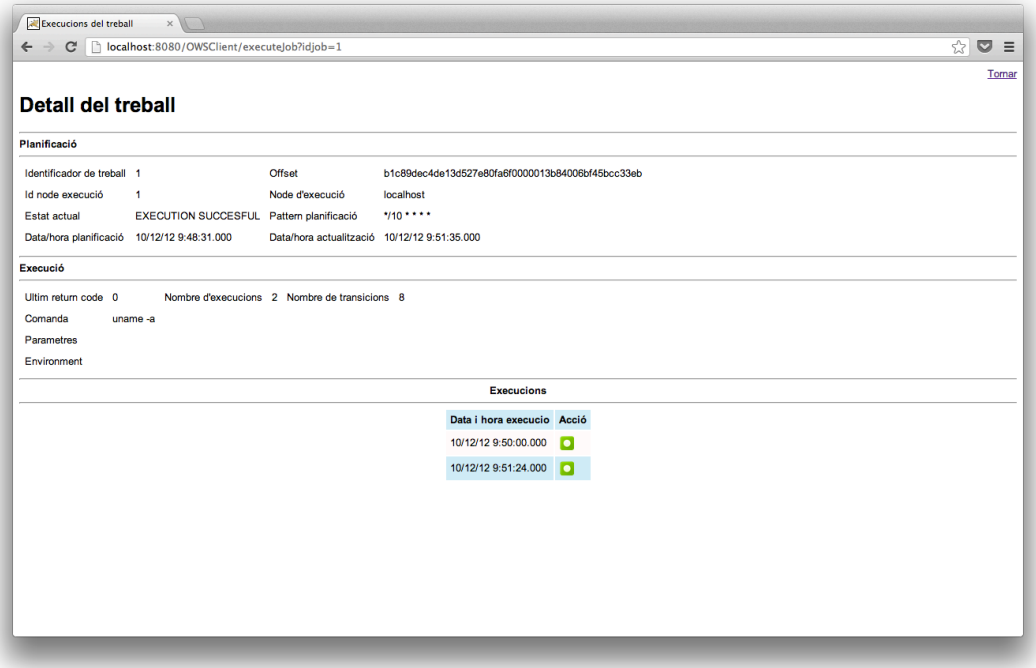


Figura 30 – Execucions del treball

Podem navegar en profunditat dins de la taula d'execucions i accedir per a cada instància a la sortida estàndard de la pròpia execució.

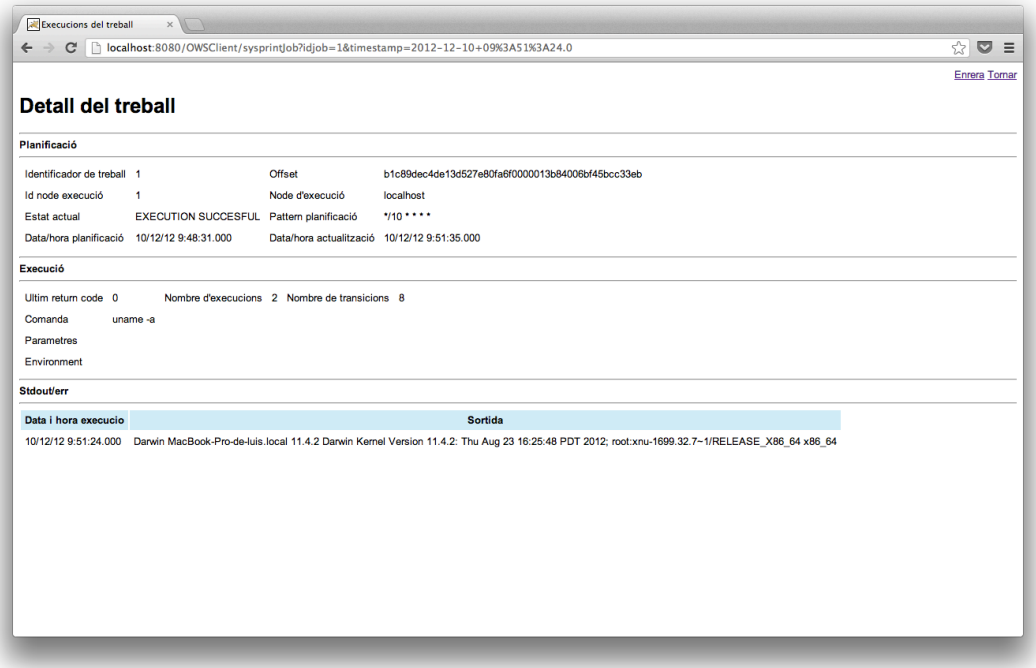


Figura 31 – Sortida de la execució del treball