



Monitorització d'alt nivell per el computing de l'experiment ATLAS (LHC)

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

Alejandro Guinó Feijóo

i dirigit per

Diego Javier Mostaccio Mancini

Bellaterra, 16 de Juny de 2010



El sotasignat, Diego Javier Mostaccio Mancini, professor de la Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

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

I per tal que consti firma la present.

Signat: Diego Javier Mostaccio Mancini
Bellaterra, 16 de Juny de 2010



Helping to turn information into knowledge

El sotassinat, Dr. Xavier Espinal Curull, de l'empresa PIC (*Port d'Informació Científica*)

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat en l'empresa sota la seva supervisió mitjançant un conveni entre PIC (*Port d'Informació Científica*) i DACSO (*Departament d'Arquitectura de Computadors i Sistemes Operatius*) de la Universitat Autònoma de Barcelona.

Així mateix, l'empresa en té coneixement i dóna el vist-i-plau al contingut que es detalla en aquesta memòria.

Signat: Dr. Xavier Espinal Curull
Cerdanyola, 16 de Juny de 2010

Campus UAB Edifici D – 08193 Bellaterra – Barcelona – Spain – Tel +34 93 581 41 09 – Fax +34 93 581 41 10
<http://www.pic.es>



Generalitat de Catalunya
**Departament d'Innovació,
Universitats i Empresa**



Ciemat
Centro de Investigaciones
Energéticas, Medioambientales
y Tecnológicas

UAB
Universitat Autònoma
de Barcelona



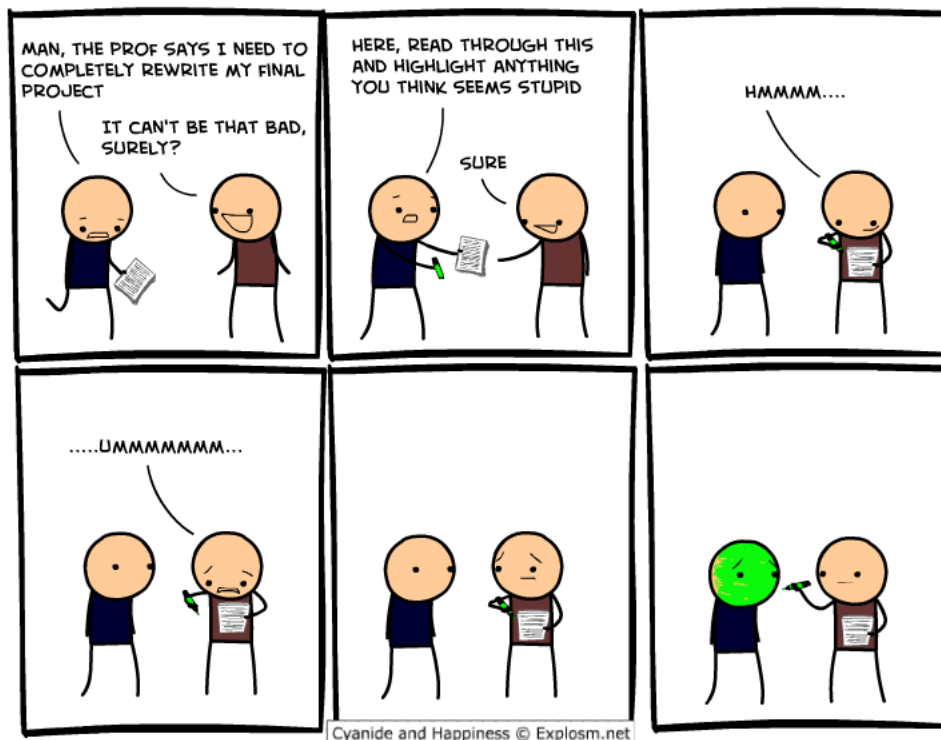
Acknowledgments

En primer lloc, agrair a la meva mare la seva paciència.

En segon lloc, agrair al personal del PIC i als companys projectistes el suport rebut sempre que he necessitat quelcom.

En última instància, donar les gràcies al Dr. Xavier Espinal per fer possible aquest projecte.

Gràcies a tots.



Índex

1	Justificació del Projecte	8
1.1	Proposta de Projecte	10
2	Avantprojecte	12
2.1	Objectius concrets	12
2.1.1	Monitorització del processament de dades al Grid	12
2.1.2	Monitorització de la distribució de dades al Grid	15
2.1.3	Serveis de monitorització central	16
2.2	Planificació inicial	17
2.2.1	Recursos materials	17
2.2.2	Planificació temporal	18
2.3	Objectius Personals	19
2.4	Fonaments Teòrics	20
2.4.1	Estat del art	20
2.4.2	Valoració d'alternatives	20
2.5	Metodologia	24
2.6	Planificació de tasques	25
2.7	Plataforma de desenvolupament	27
2.8	Anàlisi de riscos	28
3	Anàlisi de Requeriments	30
3.1	Requeriments Funcionals	30
3.2	Requeriments No Funcionals	31
3.3	Estudi dels Casos d'Us	32
3.3.1	Diagrama de Casos d'ús	32
3.3.2	Identificació dels actors	33
3.3.3	Identificació dels Casos d'ús	33
3.4	Diagrama de Flux	35

4 Disseny	37
4.1 Diagrames de Seqüència	37
4.1.1 Scripts	37
4.1.2 Sistema web	38
4.2 Diagrames d'Estats	40
4.2.1 Col·lector Jobs	41
4.2.2 Col·lectors SAM/Dades	42
4.2.3 Sistema web	43
4.3 Diagrames de Classes	44
4.4 Estructura de la BBDD	46
5 Implementació	47
5.1 Pla d'Implementació	47
5.2 Fases de la implementació	47
5.3 Interfície d'usuari	51
5.4 Dificultats sorgides a la implementació	53
5.4.1 Col·lectors	53
5.4.2 Web	53
5.4.3 Problemes puntuals d'Implementació a qualsevol punt de l'entorn del sistema	54
6 Proves i/o Tests	56
6.1 Pla de test	56
7 Conclusions	59
A Glossari de Termes	61
B Esquema ER de la BBDD	65
C Paper presentat a CHEP 2010	69

Índex de taules

2.1	Llistat de tasques i duració	18
-----	--	----

Índex de figures

1.1	Model en capes o <i>tiers</i>	9
1.2	Sala de servidors al CERN	11
2.1	Estats d'un job (part 1)	13
2.2	Estats d'un job (part 2)	14
2.3	Gràfic de Panda Monitor	15
2.4	Estats d'una transferència (part 1)	15
2.5	Estats d'una transferència (part 2)	16
2.6	Part d'una taula amb alguns <i>checks</i>	17
2.7	Diagrama de Gantt	19
2.8	Gràfic de Panda generat amb RRD	21
2.9	Gràfic generat amb Open Flash Chart	22
2.10	Gràfic generat amb solució basada en JavaScript	23
2.11	Gràfic generat amb GWT	24
3.1	Diagrama de casos d'ús	32
3.2	Visió interna del cas d'ús Monitor1	34
3.3	Visió interna del cas d'ús Monitor2	34
3.4	Visió interna del cas d'ús Manteinance	34
3.5	Visió interna del cas d'ús Application feed	35
3.6	Visió interna del cas d'ús DDBB feed	35
3.7	Diagrama de flux	36
4.1	Diagrama de seqüència dels <i>scripts</i>	38
4.2	Diagrama de seqüència del sistema web (part 1)	39
4.3	Diagrama de seqüència del sistema web (part 2)	40
4.4	Diagrama d'estats del col·lector Jobs	41
4.5	Diagrama d'estats dels col·lectors SAM/Dades	42

4.6	Diagrama d'estats del sistema web	43
4.7	Diagrama de classes (resumit)	44
4.8	Diagrama de classes (Jobs)	45
5.1	Representació de la primera etapa de la interfície web	49
5.2	Representació (resumida) de l'estructura global	50
5.3	Visió d'un dels gràfics (Jobs)	51
5.4	Visió d'un clic a un dels punts del estat <i>Running</i>	52
5.5	Detall de la taula de <i>checks</i> SAM	52

Capítol 1

Justificació del Projecte

El gran col·lisionador d'Hadrons del CERN (LHC) [5] forma part d'un dels projectes científics més importants fets fins avui dia. Aquesta enorme "màquina", creada amb la finalitat d'esbrinar els fonaments físics del nostre univers, genera una ingent quantitat de dades (15 milions de Gigabytes anualment) les quals han de ser accessibles per a multitud de científics d'arreu del món (uns 5000 aproximadament) per tal de que aquests les puguin analitzar i extreure'n conclusions. La infraestructura necessària per a que això sigui possible es el WLCG (Worldwide LHC Computing Grid [10]), una estructura de recursos computacionals que engloba més de 170 centres en 34 països amb 100.000 CPUs encarregades de processar i analitzar les dades (més de 10 Petabytes) per tal d'acabar emmagatzemant-les. Aquesta estructura es troba distribuïda en un model de capes o "tiers" on podem trobar quatre nivells:[16]

- El nivell 0 (Tier-0 al CERN) on les dades son filtrades (amb valors de 1GB/s fins a 1 PB/s) i emmagatzemades en cinta en cru, essent sotmeses a un pre-processament abans de ser distribuïdes a nivells inferiors.
- El nivell 1 (Tier-1) format per onze grans centres d'arreu del món que disposen d'una gran capacitat d'emmagatzemament, computació i suport "round-the-clock" per el WLCG (suport continu independent al horari). Aquests

centres reprocessen les dades a mesura que el coneixement del detector augmenta. També d'encarreguen de posar a disposició del nivell subsegüent les dades rebudes del nivell 0, augmentant la disponibilitat de les mateixes en gran mesura.

- El nivell 2 (Tier-2) disposa de centres adequats per a les tasques d'anàlisi de caire més específic (instituts de física) i generació de simulacions MonteCarlo.
- El nivell 3 (Tier-3) format per petits instituts, permet als científics accedir a aquestes dades i anàlisis.

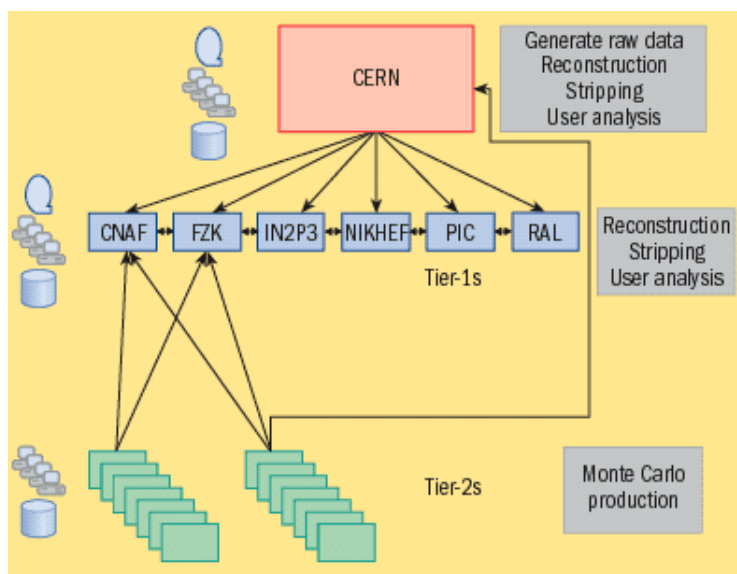


Figura 1.1: Model en capes o *tiers*

En tot aquest complicat entramat de computació i fluxos de dades, el PIC (Port d'Informació Científica) es un dels onze centres Tier-1 el qual dona serveis a tres dels quatre experiments relatius al LHC (ATLAS, CMS i LHCb.) i, com a tal, requereix un alt grau de monitorització de cadascun dels aspectes relatius a cada experiment i la seva activitat diària.[4] En el cas de l'experiment ATLAS, les

dades sobre el rendiment del centre PIC, es troben fragmentades en diverses planes web que obtenen les seves dades directament del nivell 0, fent difícil el seu seguiment i la posterior pressa de decisions en cas d'errors, fallades o problemes diversos. A més, el PIC és el centre que actua com a canal per als Tier-2 associats, porta única a tots els centres d'Espanya i Portugal.

Així doncs el projecte que s'exposa en les pàgines següents tracta de fer més portable la tasca de monitorització relativa al processament i recepció/entrega de dades de l'experiment ATLAS, reunint en una sola plana les fonts de informació sobre el mateix, fent possible un seguiment del rendiment d'aquest centre i dels centres associats (Tier-2s) molt més acurat, permetent reaccionar als múltiples imprevistos que la complexa estructura descrita anteriorment pot patir.

1.1 Proposta de Projecte

Per a solucionar el problema de la fragmentació de la informació es proposa: Construir una sèrie d'aplicacions, que anomenarem *col·lectors*, que s'encarreguin de l'obtenció de les dades. Aquestes dades es guardaran en una BBDD local amb la finalitat de construir una plana que permeti representar-les de la millor de las maneres per a la nostra tasca de monitorització, seguint de prop els possibles incidents a partir d'una estructura top-down (HLM)[21].

Pros: Es disposa de certa llibertat a l'hora d'establir els procediments/tecnologies que portin a la correcta consecució del projecte tot i tenir una sèrie de condicions relatius a tecnologies emprades amb èxit fins aleshores en projectes semblants.

Contres: Les diverses planes que contenen la informació necessària per a la correcta monitorització obtenen les seves dades del Tier-0 (nivell 0 o CERN), el qual restringeix l'accés a les seves BBDD. La raó es que aquestes BBDD estan destina-

des a ser emprades per temes de producció i, degut a la gran càrrega que aquestes suporten, el seu ús per a tasques de monitorització està molt limitat. El primer repte rau en la necessitat d'obtenir i emmagatzemar aquestes dades. Així doncs caldrà obtenir-les per altres vies menys invasives i transparents a les BBDD del CERN.



Figura 1.2: Sala de servidors al CERN

Capítol 2

Avantprojecte

2.1 Objectius concrets

Relacionat amb l'activitat del PIC i els sites associats (6 Tier-2s) es pretén obtenir informació de control relativa al experiment ATLAS de:

2.1.1 Monitorització del processament de dades al Grid

Treballs enviats al Grid de simulació d'esdeveniments, processat i re-processat de dades reals. Aquestes dades les obtindrem del sistema de monitorització Dashboard (Arda Dashboard Production System Monitoring) i del sistema de monitorització Panda (Production ANd Distributed Analysis system).[12]

- Les figures següents, parts d'una taula única del sistema Dashboard, mostren els estats dels *clouds* (un *cloud* està format pel Tier-1 i els seus Tier-2s associats). ATLAS utilitza el mètode dels treballs *pilots* (petits treballs que comproven tant la idoneïtat del node en termes de programari i de variables d'entorn com de disponibilitat de les dades). Els noms de les columnes corresponen a diferents estats dels *jobs* (treballs) pilots. Els diferents estats són [9]:

- **Defined:** *Job* inserit a la BBDD de Panda.
- **Assigned:** *Job* subscrits a un site.
- **Waiting:** Els fitxers d'entrada no estan llestos.
- **Activated:** Esperant peticions *pilot*.
- **Sent:** Enviat a un *worker node*.
- **Running:** En procés per un *worker node*.
- **Holding:** Afegint fitxers de sortida als datasets DQ2.
- **Transferring:** Els fitxers de sortida s'estan movent de T2 a T1.
- **Finished:** *Job* finalitzat correctament.
- **Failed:** *Job* finalitzat incorrectament degut a errors.

<i>cloud</i>	<i>defined</i>	<i>assigned</i>	<i>waiting</i>	<i>activated</i>
TRIUMF	0	0	0	1960
None	4	13	32	6878
NDGF	0	0	0	0
BNL	0	1	0	156
SARA	0	1286	0	245
CNAF	0	0	0	416
PIC	0	0	0	1127
CERN	0	10	0	51
ASGC	0	0	0	119
LYON	0	0	0	0
FZK	0	0	0	2646
RAL	0	4	0	2029
<i>total</i>	4	1314	32	15627

Figura 2.1: Estats d'un job (part 1)

- En la primera figura (figura 2.1) es pot observar com el *cloud* PIC, presenta 1190 *jobs* esperant a ser atesos.
- En la segona (figura 2.2), són visibles els valors restants que defineixen l'estat d'un *job*. El cloud PIC disposa de 1742 *jobs* en execució, 313 acabant d'ésser processats, 2269 en espera de ser transferits d'un Tier 2 al PIC, 9 finalitzats correctament i 54 finalitzats amb errors. L'eficiència en aquest punt

temporal es només d'un 14,3%. Quelcom dins el *cloud* PIC està provocant moltes fallades.

<i>running</i>	<i>holding</i>	<i>transferring</i>	<i>success</i>	<i>failure</i>	<i>efficiency</i>
1966	520	765	3074	15	99.5%
1827	405	2127	2096	140	93.7%
3326	105	34	1803	31	98.3%
383	1031	24	281	55	83.6%
1742	629	1164	23	150	13.3%
885	35	1909	110	0	100%
1190	313	2269	9	54	14.3%
22	1	0	40	0	100%
27	396	0	30	0	100%
26	608	18	20	0	100%
2793	215	9094	11	3	78.6%
16	345	7	7	0	100%
14203	4603	17411	7504	448	94.4%

Figura 2.2: Estats d'un job (part 2)

- Panda (Production ANd Distributed Analysis system) [9] es un altre dels sistemes de monitorització disponibles. Aquest sistema s'ha desenvolupat per a ATLAS per tal d'assolir els requeriments d'un sistema de gestió de càrregues de treball, orientat a dades per a la producció i el processament d'anàlisis distribuïts, capaç d'operar sobre els nivells de processat de dades del LHC. De la multitud de informació que presenta, certs aspectes són força útils per a la monitorització del cloud, com el que podem veure en la figura 2.3
- En períodes d'activitats corren de l'ordre de 50.000 *jobs* concurrentment a tot el WLCG per ATLAS.

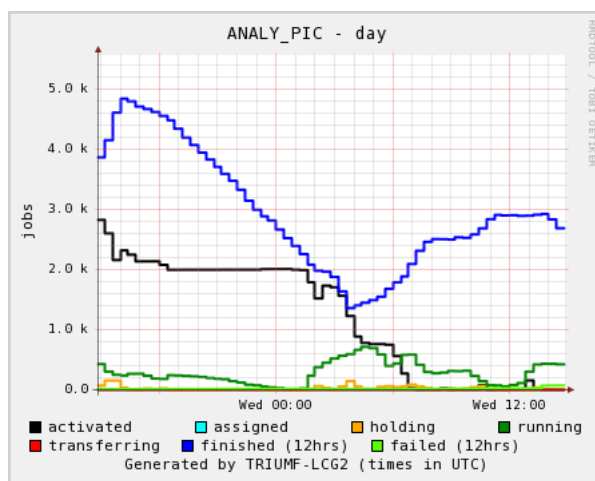


Figura 2.3: Gràfic de Panda Monitor

2.1.2 Monitorització de la distribució de dades al Grid

La distribució de dades, tant reals com simulades, son de vital importància i la seva monitorització és crítica. L'objectiu final de l'estructura WLCG es possibilitar l'accés als científics de totes les dades generades per les col·lisions i les dades simulades. Les figures següents, parts d'una taula única, mostren els *rates* de transferències per cada *cloud*, el número de fitxers i les eficiències.

Cloud	Transfers		
	Efficiency	Throughput	Successes
ASGC	100%	3 MB/s	156
BNL	33%	205 MB/s	3857
CERN	100%	68 MB/s	1164
CNAF	99%	7 MB/s	369
FZK	99%	108 MB/s	8537
LYON	99%	102 MB/s	2387
NDGF	100%	3 MB/s	187
PIC	100%	14 MB/s	604
RAL	100%	14 MB/s	2374
SARA	100%	31 MB/s	663
TRIUMF	100%	21 MB/s	3478

Figura 2.4: Estats d'una transferència (part 1)

- En la primera figura (figura 2.4) es pot observar com el *cloud* PIC, presenta una eficiència del 100% (aproximadament), amb una taxa de transferència de 14MB/s i amb 604 fitxers transferits.

Registrations		Errors		
Datasets	Files	Transfer	Registration	Services
33	156	0	0	0
76	3850	7981	0	0
35	1162	0	0	0
69	369	2	0	0
52	8542	77	0	0
54	2391	28	0	0
32	186	0	0	0
48	600	1	0	0
53	2370	1	0	0
70	663	1	0	0
138	3471	3	0	0

Figura 2.5: Estats d'una transferència (part 2)

- En la segona figura (figura 2.5) es pot observar com s'han registrat correctament 600 fitxers i 48 datasets. També es visible un error de transferència. Tot i presentar un 100% en l'anterior figura, sembla que algun servei no ha acabat de transferir un arxiu correctament.

2.1.3 Serveis de monitorització central

Des del CERN s'envien contínuament *checks* (comprovacions) per tal de monitoritzar el serveis a tots els sites de WLCG. Aquestes comprovacions les realitza una eina anomenada SAM (Service Availability Monitoring) la qual s'encarrega de testar totes les capes del WLCG:

- Sistema d'informació: BDII (Berkeley Databases Information Index)
- Serveis de computació: Resource Broker (RB), Computing Elements (CE), Worker Nodes (WNs), accounting, etc.
- Serveis de Storage (emmagatzematge): Storage Element (SE), Storage Resource Manager (SRM), Catàlegs de dades (LFC, local file system).

- Serveis de transferència de dades: FTS (File Transfer Service)

SiteName	NodeName	Status	atlas								
			js	wn	rgma	bi	esh	swdir	atlas-local-file-access	atlas-swdirspace	
BIFI	ce-egee.bifi.unizar.es	WARN	ok	info	ok	ok	ok	ok	ok	warn	ok
CESGA-EGEE	ce2.egee.cesga.es	OK	ok	info	ok	ok	ok	ok	ok	warn	ok
IFIC-LCG2	ce01.ific.uv.es	OK	ok	info	warn	ok	ok	ok	ok	warn	ok
IFIC-LCG2	ce03.ific.uv.es	OK	ok	info	warn	ok	ok	ok	ok	ok	ok
IFIC-LCG2	ce04.ific.uv.es	OK	ok	info	warn	ok	ok	ok	ok	ok	ok
LIP-Coimbra	grid006.gridc.lip.pt	OK	ok	info	warn	ok	ok	ok	ok	ok	ok
LIP-Lisbon	ce02-atlas.lip.pt	OK	ok	info	warn	ok	ok	ok	ok	error	ok
MA-01-CNRST	ce1.cnrst.magrid.ma	WARN	ok	info	ok	ok	ok	ok	ok	warn	ok
NCG-INGRID-PT	ce01.ncg.ingrid.pt	OK	ok	info	warn	ok	ok	ok	ok	ok	ok
UAM-LCG2	grid003.ft.uam.es	OK	ok	info	warn	ok	ok	ok	ok	ok	ok
ifae	ifaece01.pic.es	OK	ok	info	warn	ok	ok	ok	ok	ok	ok
ifae	ifaece02.pic.es	OK	ok	info	warn	ok	ok	ok	ok	ok	ok
pic	ce05.pic.es	OK	ok	info	warn	ok	ok	ok	ok	error	ok
pic	ce06.pic.es	OK	ok	info	warn	ok	ok	ok	ok	ok	ok
pic	ce07.pic.es	OK	ok	info	warn	ok	ok	ok	ok	ok	ok

Figura 2.6: Part d'una taula amb alguns *checks*

2.2 Planificació inicial

2.2.1 Recursos materials

Fet un primer estudi del que es demana i del que es pretén construir, les necessitats materials d'aquest projecte s'estima que seran, inicialment les següents:

- Una màquina per tal de desenvolupar el projecte.
- Una màquina (servidor físic) que contingui: els col·lectors, la BBDD i la plana web (amb els servidors de suport).

2.2.2 Planificació temporal

Dins de les previsions inicials, es realitza una estimació temporal del que ha durar el projecte i com s'han de distribuir, en primera instància, les diverses tasques que es preveu tindrà el mateix.

Tasques	Duració	Inici	Fi
Planificació Inicial	5 dies	26/10/09	30/10/09
Proposta i inici prototip	25 dies	02/11/09	04/12/09
Documentació (1 ^a fase)	5 dies	30/11/09	04/12/09
Generació del col·lector de "Dades"	10 dies	07/12/09	18/12/09
Adaptació a GWT de "Dades"	9 dies	21/12/09	31/12/09
Proposta inicial de plana	23 dies	01/12/09	31/12/09
Documentació (2 ^a fase)	20 dies	18/01/10	12/02/10
Generació del col·lector "SAM"	10 dies	04/01/10	15/01/10
Adaptació a GWT de la part "SAM"	10 dies	25/01/10	05/02/10
Documentació (3 ^a fase)	5 dies	22/03/10	26/03/10
Generació del col·lector VO-box	10 dies	15/02/10	26/02/10
Adaptació a GWT de la part "VO-box"	10 dies	01/03/10	12/03/10
Documentació (4 ^a fase)	5 dies	19/04/10	23/04/10
Completar i complementar	25 dies	15/03/10	16/04/10
Detalls de la capa de presentació	10 dies	26/04/10	07/05/10
Documentació (5 ^a fase)	5 dies	10/05/10	14/05/10
Detalls globals	33 dies	17/05/10	30/06/10

Taula 2.1: Llistat de tasques i duració

A continuació es mostra un diagrama de Gantt amb la planificació gràfica de les tasques anteriors:

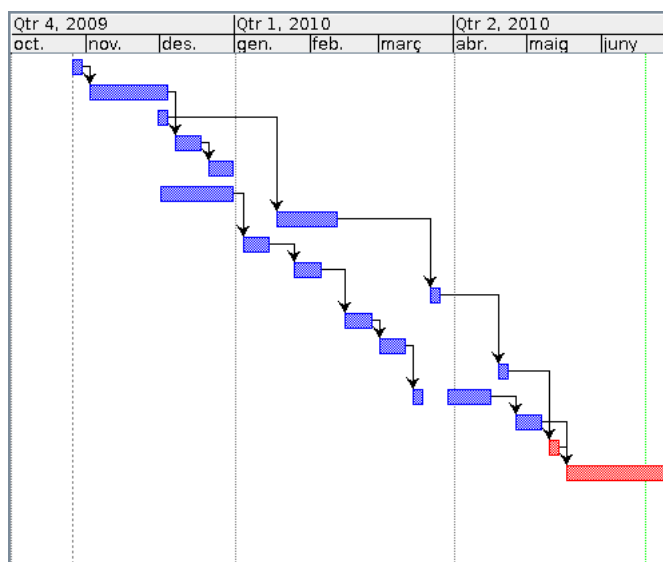


Figura 2.7: Diagrama de Gantt

2.3 Objectius Personals

Un cop plantejats els objectius del projecte i definida la situació del mateix, cal estudiar quina ha estat la motivació personal, el rerefons, que m'ha situat al capdavant del mateix. Són diverses les raons que m'han portat a liderar el desenvolupament d'aquest projecte, entre les que cal destacar l'oportunitat de treballar en col·laboració amb el personal del Port d'Informació Científica, on he pogut assistir a demostracions tecnològiques, i aprendre més enllà del mateix projecte en si, adquirint valuosa experiència pel dia en que formi part d'un entorn de treball definitiu.

Aquest projecte, m'ha ofert l'oportunitat d'enllaçar els meus coneixements amb el projecte científic més important d'Europa i potser del món (LHC) que, tot i trobar-me en un pla molt secundari del mateix, em permet gaudir de l'honor de participar-hi i col·laborar-hi. Respecte del punt de vista formatiu del projecte,

destacaria la importància d'haver interactuat amb tecnologies totalment noves per a mi, assolint un grau de coneixement profund de les mateixes fins al punt de poder comptar amb aquests nous coneixements per a nous reptes professionals. Així doncs, en resum, aquest projecte m'ha ofert: experiència laboral, nous coneixements i una visió més realista del desenvolupament d'un projecte informàtic en el món real.

2.4 Fonaments Teòrics

2.4.1 Estat del art

La idea de generar aplicacions col·lectores de dades no es quelcom nou. Existeixen diverses aplicacions en camps tan diversos com la informació personal o la informació topogràfica. Aquest projecte respon a la mateixa necessitat que la majoria d'aquestes aplicacions (tenir la informació a disposició del usuari/client) però amb una visió final específica: monitoritzar un sistema. Aquesta necessitat provoca que el projecte no només es preocupi per aquesta obtenció i emmagatzematge de dades, sinó que defineix la representació de tota aquesta informació en un sol entorn que permeti la correcta gestió del sistema del que s'obtenen.

2.4.2 Valoració d'alternatives

Abans de prendre una decisió respecte de les tecnologies que s'acabaran emprant, es fa un petit estudi de diverses possibles tecnologies candidates. Si bé encara no s'ha parlat dels requeriments del projecte, es possible avançar que no existeix cap que explícitament marqui quines son les tecnologies referència. Les alternatives presentades són les següents:

Front-end

Aquí parlarem d'allò que l'usuari acabarà veient, de la part "web" del projecte. Inicialment, cal destacar que totes les opcions presentaran PHP com a llenguatge web base per tal de construir els enginys gràfics, tant per la seva ampla difusió, suport, modularitat, dinamisme, eficiència i acceptació com per què es una eina OpenSource.

PHP + RRDtool[15]: RRDtool es una eina OpenSource amb capacitat de realitzar logs, ja que incorpora una base de dades pròpia i específica, així com de construir gràfics basats en sèries temporals. Primera alternativa a tenir en compte degut al seu ampli ús en projectes molt relacionats amb el PIC com són Ganglia, Cacti o Panda. Aquesta proposta era un de les més lògiques per ser força coneguda pel client i disposar de bona "reputació". El problema rau en el poc dinamisme que les gràfiques ofereixen (gràfiques 2D en format imatge) i la pèrdua de granularitat en el temps.

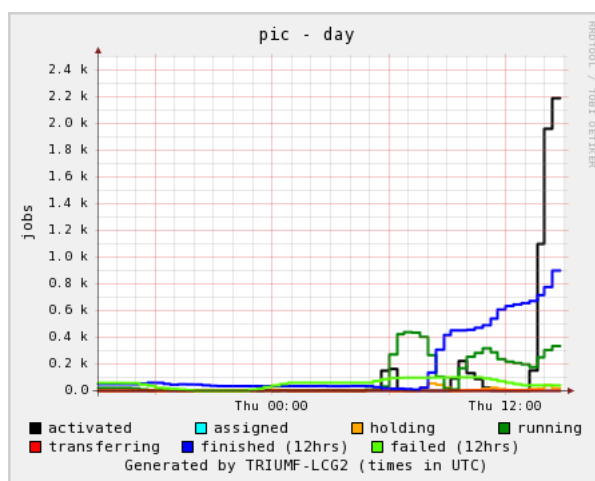


Figura 2.8: Gràfic de Panda generat amb RRD

PHP + Open Flash Chart[2]: Open Flash Chart es una eina OpenSource que permet crear gràfiques via Flash, configurables a través de fitxers JSON, i integrar-les en diversos llenguatges de programació pels que disposa llibreries específiques. Si bé els resultats són força espectaculars i les possibilitats de dinamisme dels gràfics son molt elevades, la proposta presenta una certa complexitat donat el nombre de tecnologies que es concentren en tant poc espai, fent la codificació i les posteriors modificacions, tasques no trivials. A més, el projecte es una iniciativa particular de dubtosa continuïtat i de difícil suport.

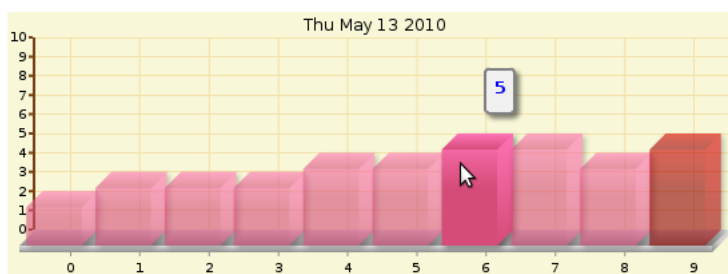


Figura 2.9: Gràfic generat amb Open Flash Chart

PHP + JavaScript: Diverses són les solucions construïdes sobre JavaScript, moltes d'elles espectaculars, basat en el paradigma de l'orientació a objectes, amb una sintaxi semblant a C i molt emprat avui en dia en el món web. Si bé és una proposta atractiva, sobretot per la integració, pels resultats i per l'ordre dels seus algorismes, les diverses eines desenvolupades amb aquest llenguatge són privatives. En cas de no emprar aquestes eines, hauríem de codificar nosaltres des de 0 les gràfiques i això, amb el temps i el personal de que disposa el projecte, no és acceptable.

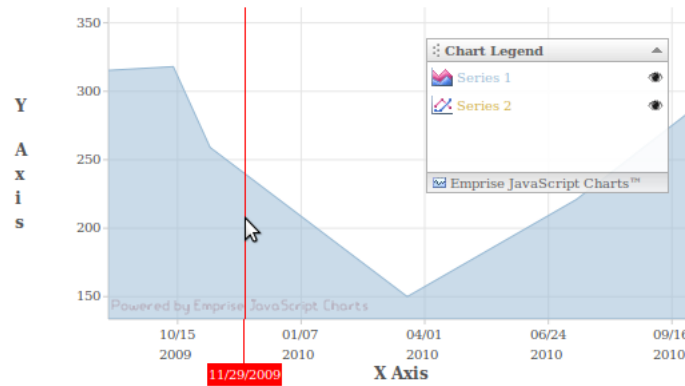


Figura 2.10: Gràfic generat amb solució basada en JavaScript

GWT: GWT (Google Web Toolkit) és un framework que permet desenvolupar aplicacions web via Java, les quals seran interpretades i transformades en planes web HTML + JavaScript. Bàsicament, ens permet construir aplicacions AJAX (Asynchronous JavaScript and XML) les quals aporten beneficis com els de proporcionar més lògica a la part client o ostensibles millores en els procediments de peticions a servidor entre d'altres. El problema d'AJAX ha estat sempre la dificultat de programar, la complexitat de les crides al servidor o la falta d'entorn de programació adients. GWT elimina tots aquests problemes. Gràcies a la capacitat d'integració a IDEs (Interactive Development Environments) tant importants com Eclipse o com Netbeans soluciona el problema del marc de treball i gràcies a emprar Java ens permet, en cas que sigui necessari, oblidar-nos tant de JavaScript com de HTML, així com dels nombrosos problemes que dona la depuració d'entorns web i la gestió dels famosos RPC (Remote Procedure Calls o crides remotes a servidor) per tal d'obtenir recursos del mateix. A més, ofereix un servidor lleuger per tal de provar l'aplicació, cosa que ens deslliura d'haver de muntar i configurar un entorn client-servidor per tal de realitzar tests i ofereix la garantia Google, com en tots els productes, visible en les continues actualitzacions, les millores de rendiment i l'expansió de possibilitats.

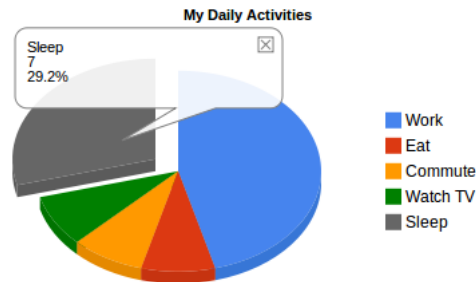


Figura 2.11: Gràfic generat amb GWT

Back-end

C: Llenguatge robust, molt utilitzat per a realitzar *scripts* com els que cal desenvolupar, clar, ordenat i de ràpida execució per la seva natura de llenguatge compilat, es una bona alternativa. El problema de C és que la mateixa natura de baix nivell que el fa ràpid, el fa complicat, factor de pes tenint en compte les restriccions de personal i de temps.

Python: Sense abandonar els valors de C, reduint una mica el de rapidesa, Python ofereix simplicitat, orientació a objecte i una API increïblement rica, equipada amb funcionalitats de gestió de xarxa, de fitxers, de threads, etc. Sobretot, cal destacar la facilitat amb la que permet interactuar amb entorns complexos, com es un fitxer XML o una connexió a base de dades, i treure'n tot el suc amb relativament poques línies de codi, mantenint un nivell d'eficiència elevat.

2.5 Metodologia

La metodologia que s'utilitzarà per a dur a terme aquest projecte es basa en la coneguda per Cicle de vida amb prototipatge: *Aquesta consisteix en crear una primera versió del programari o d'una part d'aquest i ensenyar-la als usuaris per tal que la critiquin i proposin millores. Aquest procés es repeteix tantes vegades com calgui.*[1]

Així doncs, es proposarà crear prototips amb les diverses alternatives presentades, tant dels col·lectors com de la part visible (web) on s'han posat de manifest els pros i els contres de cada una per tal d'acabar establint el model actual com a definitiu i extensible, discutint amb el client diversos punts de vista tals com l'eficiència, el coneixement de les tecnologies emprades, la usabilitat, la reutilització, la flexibilitat o, fins i tot, l'impacte visual.

2.6 Planificació de tasques

La planificació de tasques (així com la seva durada) ja apareix llistada a l'apartat anterior (Planificació temporal). Aquí llistarem doncs de nou les tasques i procedirem a explicar de forma somera, la funció de cadascuna:

- **Planificació Inicial:** Un cop pactada l'execució del projecte, es plantegen una sèrie de reunions amb el responsable del PIC per tal de llimar els requeriments funcionals del projecte i obtenir tota la informació possible relativa a l'obtenció de les dades per als col·lectors.
- **Proposta i inici prototip (col·lector mínim de *Jobs* + representació web):** Un cop coneguts els paràmetres bàsics del projecte, es procedeix a implementar un prototip bàsic de col·lector en diversos llenguatges. El mateix es fa amb la gestió local de les dades (BBDD) i amb la capa de presentació. S'estableixen aquí les tecnologies finals a aplicar al projecte.
- **Documentació (1^a fase):** Generació de la documentació relativa als aspectes bàsics del projecte.
- **Generació del col·lector de "Dades":** Desenvolupament del col·lector de la font Dashboard (monitor) relatiu a la informació que fa referència a la gestió dels fluxos de dades que fa el PIC.
- **Adaptació a GWT de "Dades":** Representació de les dades obtingudes tan a nivell gràfic com a nivell descriptiu.

- Proposta inicial de plana: Primera execució conjunta de tot el sistema. El sistema ha de ser viable a tots els efectes excepte a nivell de presentació.
- Documentació (2^a fase): Generació dels següents aspectes de la documentació segons els passos assolits fins aquest moment.
- Generació del col·lector "SAM": Desenvolupament del col·lector de la font SAM, relativa a serveis que ofereix el PIC.
- Adaptació a GWT de la part "SAM": Representació de les dades obtingudes tan a nivell gràfic com a nivell descriptiu. Integració amb tot allò fet fins al moment. Revisió de la presentació de les dades.
- Documentació (3^a fase): Generació dels següents aspectes de la documentació segons els passos assolits fins aquest moment.
- Generació del col·lector VO-box: Desenvolupament del col·lector de la font VO-box, amb informació relativa als *Jobs* que es corren en els hosts suportats per la infraestructura del PIC.
- Adaptació a GWT de la part "VO-box": Representació de les dades obtingudes tan a nivell gràfic com a nivell descriptiu. Integració amb tot allò fet fins al moment. Revisió de la presentació de les dades.
- Documentació (4^a fase): Generació dels següents aspectes de la documentació segons els passos assolits fins aquest moment.
- Completar i complementar: Aquesta és la primera de tres fases destinades a complementar la feina bàsica del que es la base del projecte. En les converses inicials, es va fer palès que “les necessitats” bàsiques del projecte es podien assumir amb el temps i els recursos que el PFC marca. Així doncs, aquesta tasca permet, en primer lloc, eixugar els retards en altres tasques (detallats en l’anàlisi de riscos) i, al mateix temps, pensar a dur a terme ampliacions que facin el sistema més potent, efectiu i, probablement, estètic.

- Detalls de la capa de presentació: Durant tot el projecte, la principal preocupació es la usabilitat, la portabilitat i la flexibilitat.
- Per desgràcia, en parlar d'un entorn web, la presentació del que es codifica **no sempre es la mateixa** depenent del navegador emprat. Es evident que, de cara a una presentació en societat del sistema, ens caldrà depurar aspectes de disseny visual per tal de que aquest no es menystingui.
- Documentació (5^a fase): Fase final de la documentació. La memòria hauria de trobar-se acabada o pendent d'aprovació per falta de petits detalls.
- Detalls globals: Qualsevol feblesa en l'estructura del projecte des de qualsevol punt de vista ha de ser resolta en aquesta fase.

2.7 Plataforma de desenvolupament

Un cop avaluades totes les possibilitats, aquest es l'esquema tecnològic que s'emprarà per tal de desenvolupar el projecte:

Generació dels col·lectors

- Tecnologies/llenguatges de programació: Python versió 2.6, MySQL 5.1.30.
- Paquets "extres" emprats: PyCurl, MySQLdb.
- Entorn de desenvolupament:
 - SO: Ubuntu 9.04
 - IDE: Eclipse 3.5.1 + PyDev

Generació dels col·lectors

- Tecnologies/llenguatges de programació:

- Entorn web o front-end: AJAX via GWT (JavaScript, XML, Java) sobre servidor web (Apache), XHTML i CSS.
- Gestió de les dades o back-end: JSP (GWT) sobre servidor amb capacitat per a l'execució de servlets (Tomcat, Jetty o similar)
- Entorn de desenvolupament:
 - IDE: Eclipse 3.5.1 + Google Plugin for Eclipse (SDK + Google engine)

Sistema Operatiu

- Linux Ubuntu 9.04 (opcionalment si s'acaba emprant una virtual machine s'emprarà el Red Hat Scientific Linux proveït pel CERN)

2.8 Anàlisi de riscos

- Riscs de tecnologia: Els col·lectors empen Python, un llenguatge de programació robust i conegut molt apropiat per a la generació de scripts. En el cas de l'entorn web, es confia en el llenguatge base de tota pàgina web (XHTML).
- Riscs relatius a persones: Aquest apartat esdevé desert ja que només una persona es troba encarregada de la gestió i desenvolupament de tot el projecte.
- Riscs organitzacionals: Aquest apartat tampoc presenta riscos donat que el PIC es una entitat organitzacional no de lucre, basada en la recerca científica.
- Risc d'eines: El desenvolupament de tot el sistema s'esta realitzant sota eines de provada efectivitat, gratuïtes (algunes amb llicència GNU) i d'ampla difusió entre el sector.

- Risc de requeriments: Els requeriments, pactats amb el client, només poden patir canvis relativament importants en l'apartat de la capa de presentació.
- Riscs d'estimació: L'estimació temporal de la durada de la generació de cada col·lector es pot arribar a veure afectada de manera greu si el model d'obtenció de les dades presenta problemes de difícil solució. Bàsicament existeix la idea que de les 4 fonts, 3 han d'aportar les dades en format XML i una d'elles via SNMP però es quelcom que encara no està segur. Si aquests pressuposts canviessin, els col·lectors es podrien tornar més complexos i, per tant, endarrerir la seva generació. Es per això que, tot i la estricta estimació temporal inicial, s'ha destinat força temps a tasques com *Detalls de presentació* o *Detalls globals* les quals poden ser àmpliament retallades en favor de qualsevol dels col·lectors afectats. No s'espera trobar grans problemes en la generació del entorn web un cop establert el prototip. En qualsevol cas, existeix una altre tasca (*Completar i complementar*) que permetria eixugar qualsevol problemàtica al respecte. Tot i les previsions inicials, cal tenir molt en compte que aquest és, sens dubte, el punt que presenta més incògnites i, per tant riscos, degut a la necessitat d'usar molt temps a la investigació de les tecnologies escollides, la falta de fonts definides per als col·lectors (base del projecte) i la possibilitat de trobar errors que puguin retardar les tasques més de l'estimat entre d'altres contingències.

Capítol 3

Anàlisi de Requeriments

L'Anàlisi de requeriments estableix i descriu a nivell teòric, de forma el més completa possible, el funcionament global del sistema. Podem dividir aquesta visió del sistema en dues categories: [11]

- Requeriments funcionals: Descriuen el comportament o funcionalitats del programari.
- Requeriments no funcionals: Restriccions de tipus quantificable imposades pel client o pel problema i que descriuen condicions generals que aquest ha de complir.

A mode d'introducció descriptiva, el projecte, des del punt de vista funcional, té com a missió obtenir dades i visualitzar-les a diversos nivells, permetent a l'usuari controlar els processos que aquestes descriuen.

3.1 Requeriments Funcionals

- Obtenció de les dades base – El sistema ha de recol·lectar una sèrie de dades bàsiques que permetin generar estructures d'anàlisi de les mateixes (com son gràfics) i emmagatzemar-les de manera que es pugin consultar sempre que sigui necessari a mode d'històric.

- Visualització de dades clau – Les dades obtingudes en el procés anterior s'han de representar de manera que permetin al seu usuari, a primer cop d'ull, obtenir una visió global del que representen per tal de mantenir un cert control i seguiment.
- Capacitat d'aprofundir en les dades – Si la visió anterior revela que els processos que generen les dades es troben en un estat no desitjat, caldrà aprofundir i veure quina es l'arrel del problema.

3.2 Requeriments No Funcionals

- Requeriments de Rendiment – Inicialment no es pretén establir restriccions ni de capacitat ni sobre el comportament d'execució del sistema. En qualsevol cas, es desitja que; el sistema sigui escalable, estructurat, ampliable i de manteniment "senzill".
- Restriccions de Disseny:
 - **Acompliment dels estàndards:** Es demana que la part web de l'aplicació segueixi els estàndards web actuals per tal de que la seva representació en els diversos navegadors reporti els menys problemes possibles.
 - **Limitacions de hardware:** En principi no s'ha establert cap restricció sobre el maquinari.
 - **Recuperació i fiabilitat davant errors:** El nivell de control d'errors del sistema ha de ser prou complet per permetre una ràpida intervenció i recuperació. Cal recordar que aquest es un sistema que funcionarà 24 hores, 7 dies a la setmana.
 - **Seguretat:** Les dades emprades per a popularitzar la BBDD s'obtenen a través de Internet per mitjans públics, a l'abast de tothom. De la mateixa manera, les representacions d'aquestes dades es trobaran obertes

a qualsevol que les vulgui consultar. La seguretat de les màquines que contenen el projecte i de la xarxa on es troben situades queda fora de l'àmbit d'aquest projecte.

3.3 Estudi dels Casos d'Us

Per tal de millorar l'especificació dels requeriments funcionals s'ha definit un diagrama de casos d'ús el qual ens permetrà veure més clarament les interaccions entre actors o entitats externes al nostre sistema que hi participin d'alguna manera, i casos d'ús o seqüències d'esdeveniments generades per aquests agents externs [17][18].

3.3.1 Diagrama de Casos d'ús

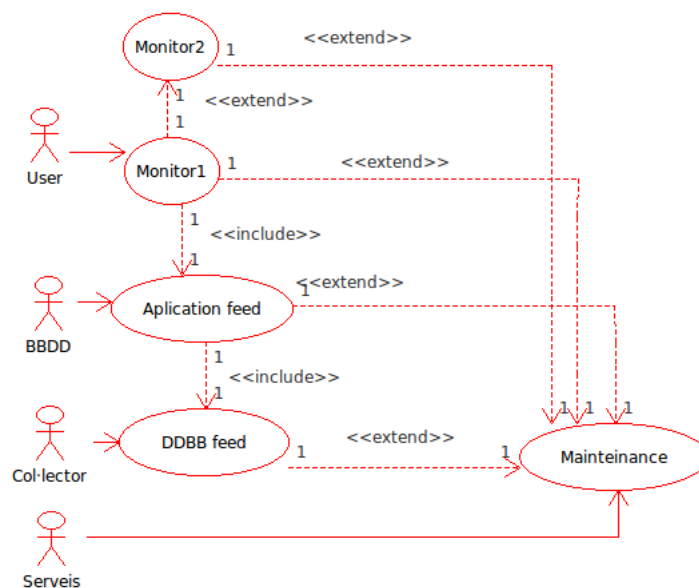


Figura 3.1: Diagrama de casos d'ús

Un usuari ha de poder consultar el nostre sistema per tal comprovar l'estat del *cloud*. En resposta a la petició de consulta, aquest ha de mostrar un primer nivell d'informació via interfície web. En cas de necessitar més informació, si l'usuari interacciona amb el sistema, ha de poder arribar a un segon nivell d'informació, més concret. Per a que això sigui possible, un sistema de col·lectors amb col·laboració amb un sistema gestor de bases de dades, han de proveir al sistema de les dades necessàries per a construir estructures informatives. El sistema pot arribar a requerir manteniment tant preventiu com adaptatiu. Això correrà a càrrec de l'usuari serveis.

3.3.2 Identificació dels actors

- User – L'usuari base de l'aplicació, el "gestor" (*shifter* o encarregat de la gestió dels Tier 2 o bé personal del PIC relatiu a la gestió de l'experiment ATLAS), s'encarregarà de visualitzar les dades de primer nivell i, en cas necessari, interactuar amb l'entorn per tal d'esbrinar quin o quins son els factors que han portat a les dades base a reflectir els valors actuals.
- BBDD – O base de dades, proveirà d'informació tant a l'entorn bàsic (dades de primer nivell) com al nivell d'anàlisi.
- Serveis – L'usuari Serveis és l'usuari o conjunt d'usuaris que s'encarregaran del posterior manteniment del sistema.
- Col·lector – Aquest actor representa a un conjunt de scripts que obtindran i proveiran a la base de dades de les dades que després estaran a disposició del sistema.

3.3.3 Identificació dels Casos d'ús

- Monitor1 – Monitor1 representa l'esdeveniment de *cridar* l'aplicació web per tal que les dades de primer nivell es representin en forma d'estructures

que permetin una visualització de l'estat dels processos que representen. L'usuari ha de poder obtenir una primera visió global de l'estat del *cloud*.



Figura 3.2: Visió interna del cas d'ús Monitor1

- Monitor2 – Monitor2 representa les seqüències d'esdeveniments que es poden produir un cop l'actor *User* pren la decisió d'interactuar amb l'entorn. Les dades obtingudes permeten profunditzar en el valors del primer nivell.



Figura 3.3: Visió interna del cas d'ús Monitor2

- Maintenance – Aquest cas d'ús reflecteix les operacions de manteniment (canvis, millores, depuració d'errors, etc.) i control del sistema.

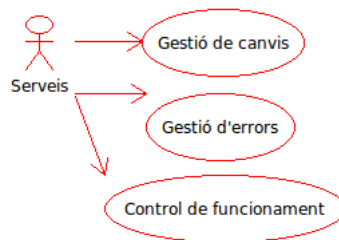


Figura 3.4: Visió interna del cas d'ús Maintenance

- Application feed – En aquest cas d'ús es defineixen els procediments entre la base de dades i l'aplicació per tal d'obtenir les dades que necessita.



Figura 3.5: Visió interna del cas d'ús Application feed

- DDBB feed– Les operacions per tal de popularitzar de manera regular la base de dades transcorren en la definició d'aquest cas d'ús.



Figura 3.6: Visió interna del cas d'ús DDBB feed

3.4 Diagrama de Flux

Per tal de representar el funcionament del sistema d'una manera global i per tal de donar suport als diagrames anteriors, s'ha definit un diagrama de flux. A grans trets, podem observar les diverses relacions i connexions entre les operacions que se succeeixen un cop iniciat el procés. Si bé la operació *Càrrega de dades* parteix d'inici, caldria deixar constància que és una operació que s'executa de manera continua, independentment de si hi han peticions HTTP o no.

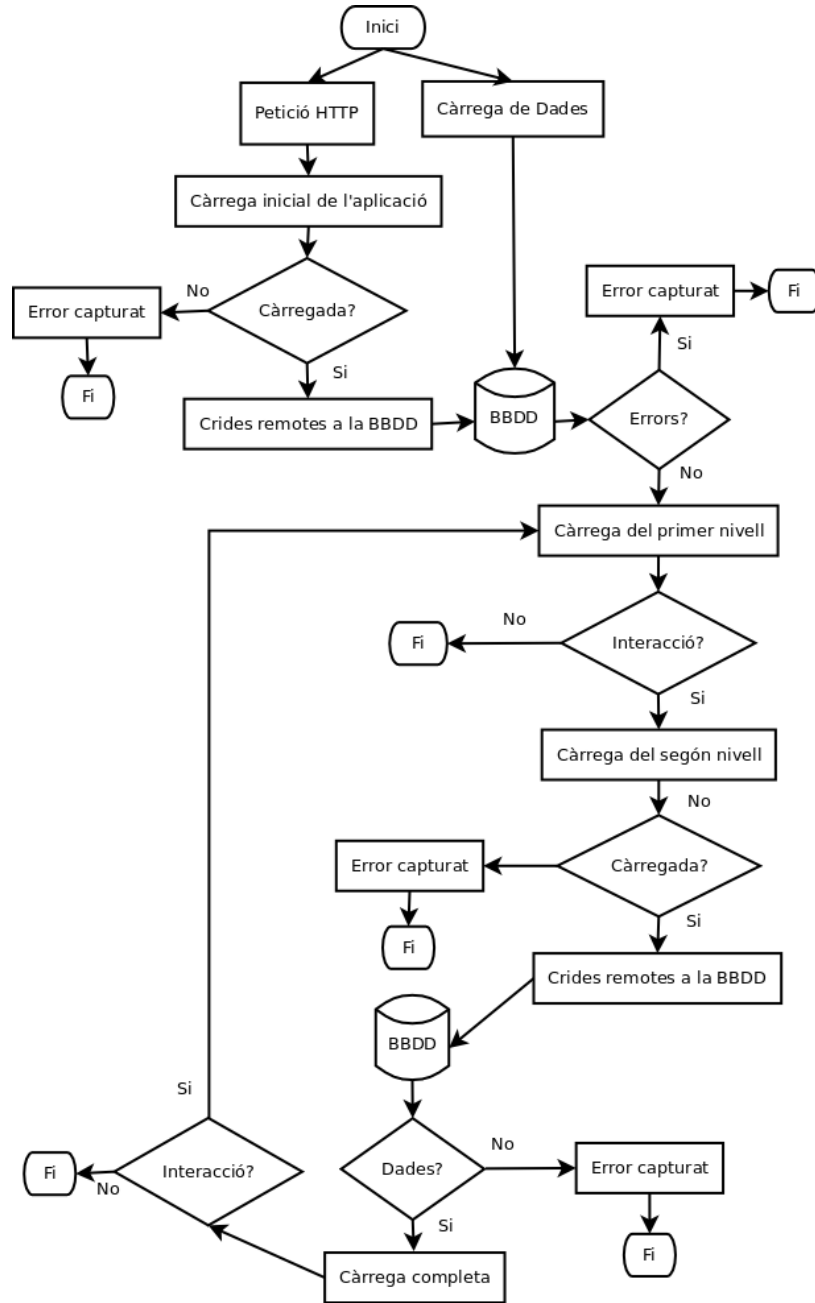


Figura 3.7: Diagrama de flux

Capítol 4

Disseny

La fase de disseny ens permet aplicar de manera *pràctica* i detallada les assumpcions obtingudes a la fase d'anàlisi, tenint en compte problemàtiques relatives a aspectes tècnics. Els models que es detallen aquí son l'avantsala de l'arquitectura que presentarà el sistema.

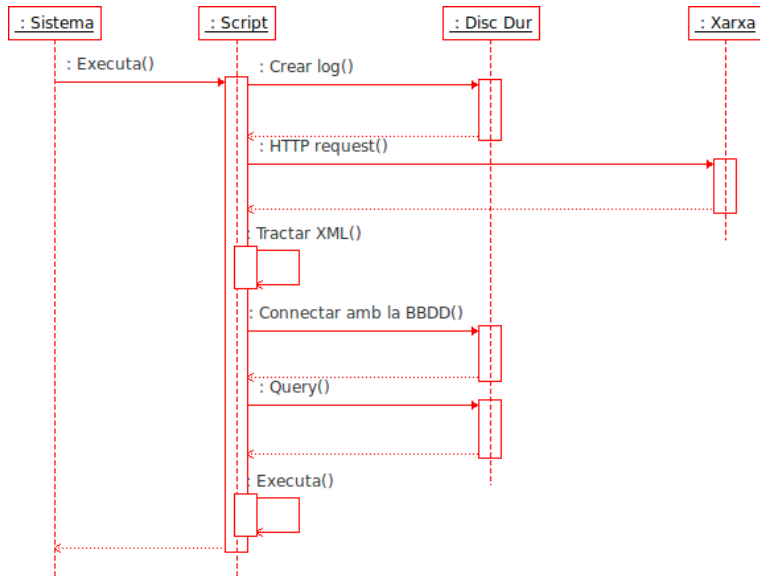
4.1 Diagrames de Seqüència

Un diagrama de seqüència mostra la interacció d'un conjunt d'objectes en una aplicació a través del temps i es modela per a cada mètode de la classe. En el nostre cas modelarem tant la seqüència d'esdeveniments que afecten als col·lectors (que aquí anomenarem *scripts*) com dels esdeveniments que genera el sistema web.

Amb aquests diagrames, començarem a perfilar les operacions que ens caldrà definir de manera més explícita més endavant.

4.1.1 Scripts

En paraules, el sistema manté els *scripts* en execució, com un procés més. Aquest *script* mostra un comportament a nivell de Thread i, cada 15 o 30 minuts depenent

Figura 4.1: Diagrama de seqüència dels *scripts*

del col·lector, es crida a si mateix. En aquesta crida/execució, cada *script* genera un log (només la primera vegada i en cas que no estigués ja creat). A continuació realitza una sèrie de peticions HTTP en les que ha d'obtenir, com a resultat, un recurs en format XML. Posteriorment ha de tractar aquest recurs per tal d'obtenir les dades necessàries. Finalment aquestes dades s'han d'acabar emmagatzemant a una base de dades (operació que necessita d'una connexió i una sèrie de *queries* per tal de garantir la correctesa de la inserció).

4.1.2 Sistema web

Donada la complexitat del sistema web, el dividirem en dos etapes (i per tant dos diagrames). El primer diagrama representarà el funcionament estructural del que s'ha anomenat amb anterioritat *primer nivell de granularitat* (recordar l'abstracció feta tant al avantprojecte com a l'anàlisi) mentre que el segon representarà la capacitat d'interacció amb l'entorn o segon nivell de granularitat.

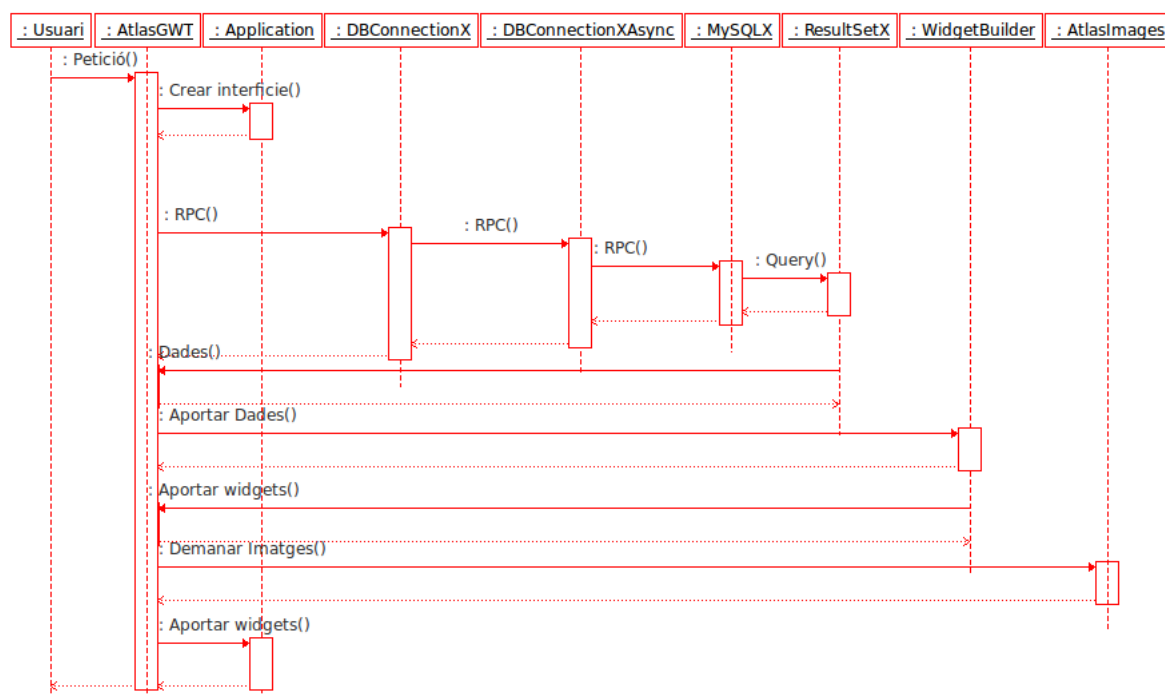


Figura 4.2: Diagrama de seqüència del sistema web (part 1)

Quan un usuari fa una petició per tal de visualitzar l'entorn web (figura 4.2), el primer punt que es crea és la interfície bàsica (elements estàtics). Inmediatament es generen crides asíncrones per tal d'obtenir els elements dinàmics (*widgets*). Durant aquest procés, les imatges (pre-processades i pre-carregades) s'envien a les seves respectives localitzacions. Un cop els *widgets* són creats, es col·loquen a la interfície.

La figura 4.3 mostra que succeeix quan l'usuari clica sobre algun dels elements (*widgets*) per tal d'obtenir més informació. Aquest element, detecta la selecció i crida a una classe gestora (abans anomenades *listeners* o escoltadors i que en la penúltima versió del SDK han adoptat el nom de *handlers* o manegadors) que genera una crida asíncrona per tal d'obtenir dades. Al mateix temps genera un element del tipus popup al que li passarà les dades un cop el procediment les serveixi.

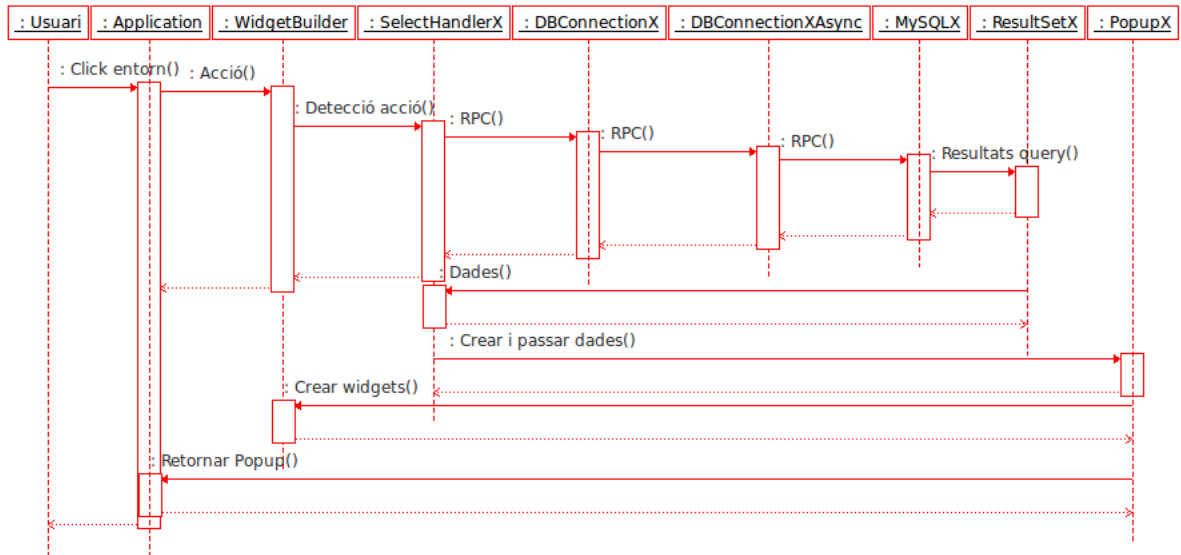


Figura 4.3: Diagrama de seqüència del sistema web (part 2)

Aquestes dades s'empraran per generar nous widgets que es col·locaran dins el popup. Aquest popup es retornarà al context de *Application* (interfície). Eventualment, es possible que es regenerin *widgets* anteriorment creats en el primer nivell. És important veure que un cop feta la primera càrrega de l'entorn, *Application* pren el control envers la classe *AtlasGWT*, punt d'entrada inicial.

4.2 Diagrames d'Estats

Per tal de tenir una visió més acurada del diagrama de flux vist a la part de l'anàlisi de requeriments, construirem un diagrama d'estats, separant el funcionament de cada un dels col·lectors, del sistema web. A grans trets, aquests son els estats per els que passen ambdós subsistemes:

4.2.1 Col·lector Jobs

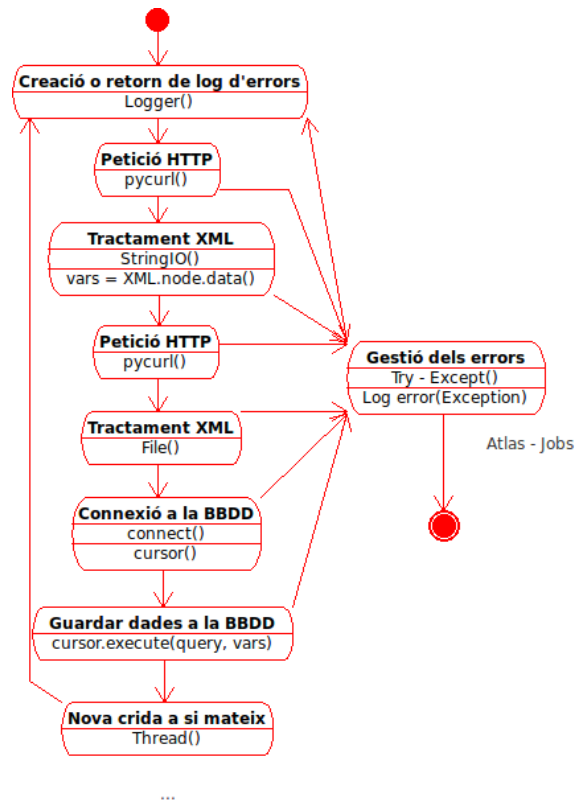


Figura 4.4: Diagrama d'estats del col·lector Jobs

El col·lector *jobs*, un cop en execució, genera un log (només la primera vegada i en cas que no estigués ja creat) a través de l'eina `logger()`. A continuació, realitza una petició HTTP via la funció incorporada *Curl*, que consisteix a executar una ordre del tipus `curl -H 'Accept: text/xml' 'url'`. Un cop servida la petició, es duu a terme un primer processament de la informació, emmagatzemant en memòria el fitxer XML en format de cadena (`StringIO`) per tal de processar-lo i obtenir els valors. Acte seguit, realitzem una nova crida HTTP per obtenir un nou fitxer XML. Aquest fitxer l'emmagatzemarem a disc donat que la seva extensió fa massa difícil el tractament en memòria dinàmica. Un cop processats tots els

valors obtinguts, els enviarem al sistema gestor de bases de dades al que, previamente, ens haurem connectat. El col·lector, en acabar, es cridarà a si mateix en un interval de 15 minuts. Aquest col·lector disposarà d'un sistema de control i gestió d'errors en sintonia amb el log creat al inici.

4.2.2 Col·lectors SAM/Dades

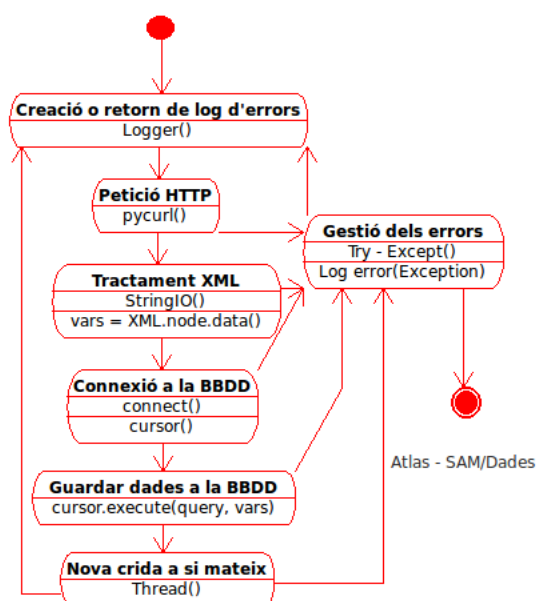


Figura 4.5: Diagrama d'estats dels col·lectors SAM/Dades

Els col·lectors SAM i Dades presenten una estructura semblant a la de Jobs amb la diferència de que no els cal realitzar dues peticions HTTP. Això es així perquè tota la informació que cal per cobrir les necessitats d'aquests col·lectors, es troba en un sol XML. Així doncs, son versions *reduïdes* del col·lector Jobs. La diferència bàsica entre els tres radica en el tractament de les dades de cada XML (donat que cada XML es troba estructurat de manera diferent).

4.2.3 Sistema web

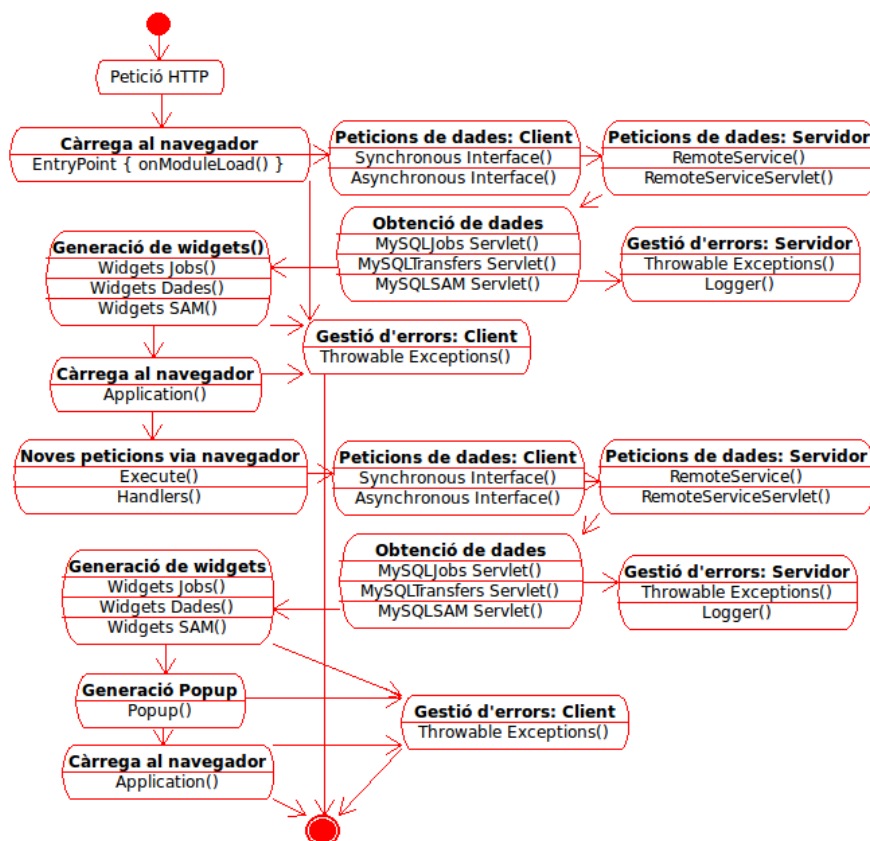


Figura 4.6: Diagrama d'estats del sistema web

El sistema web, un cop invocat, carrega una primera interfície estàtica de manera síncrona a través de AtlasGWT (OnModuleLoad()) mentre, de manera asíncrona, es realitza una petició de dades que creua el llindar d'un client fins a un servidor (callback). Aquest gestiona la petició i retorna les dades. Ambdós moviments tenen control d'errors. En el cas d'errors al client, es procedeix a finalitzar amb un missatge d'error. En cas d'errors al servidor, el control d'errors acaba transmetent aquest error al client (a més d'emmagatzemar el problema en un log al servidor). Un cop servides les dades, es generen els *widgets* (contingut dinàmic) i finalitza

la primera execució. A partir d'aquest moment, Application (que disposa de tot el contingut carregat) esdevé el punt central de gestió, generant les noves peticions de dades i les noves creacions de contingut dinàmic.

4.3 Diagrames de Classes

Aquest tipus de diagrames estàtics, permeten descriure l'estructura del sistema mostrant la interacció entre les diverses classes que el formen. Per tenir una visió global del sistema, s'ha creat un diagrama de classes que representa un resum de l'estructura total (format per 40 classes i de difícil representació).

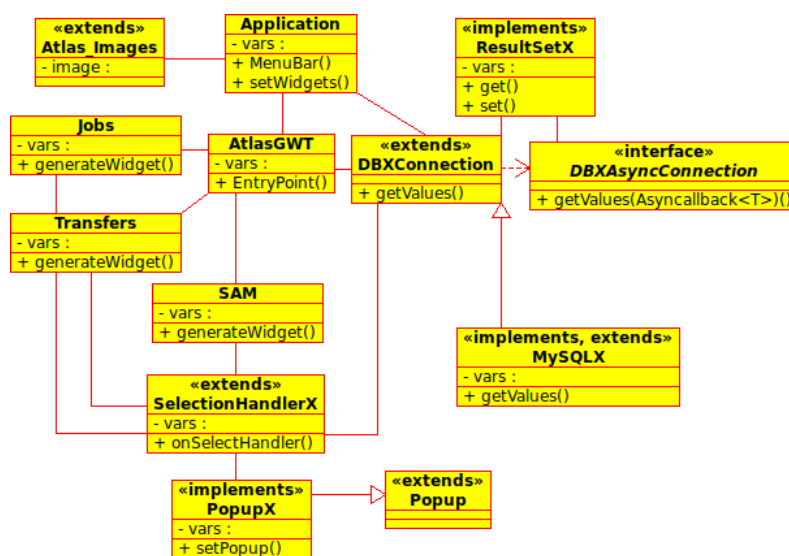


Figura 4.7: Diagrama de classes (resumit)

Vista l'estructura global, es fa palès que existeix un submodel d'estructura que comparteixen els tres elements pels quals es generen dades (Jobs, Dades i SAM). Entrar en detall amb un d'ells, permetrà conèixer com es qualsevol dels altres dos. L'únic que canviarà seran els atributs i els mètodes (però no les funcionalitats).

Així doncs, es representarà Jobs amb tota la seva magnitud com a últim esquema, el de més baix nivell, pas previ a la implementació.

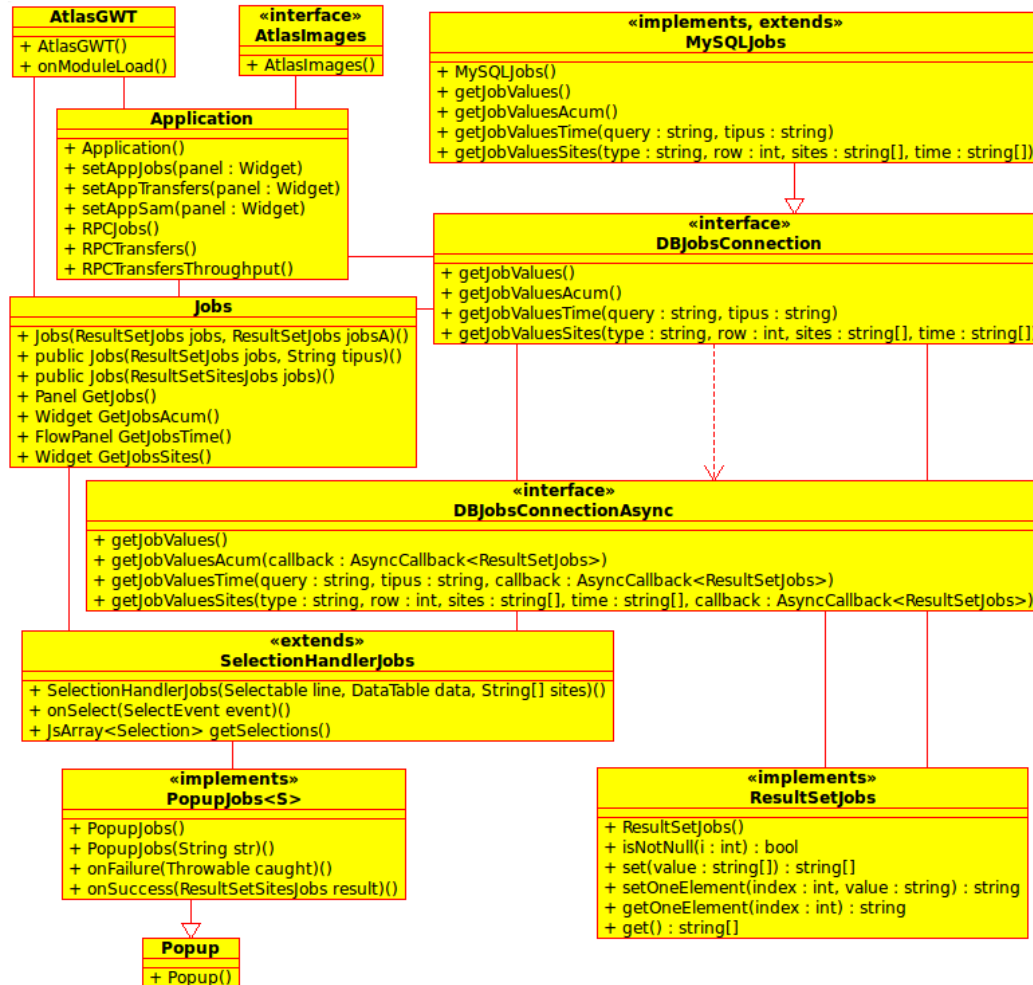


Figura 4.8: Diagrama de classes (Jobs)

4.4 Estructura de la BBDD

La BBDD està formada per 27 taules gestionades pels motor MySQL i InnoDB segons si presenten relacions que necessiten integritat referencial o no. Si bé el màxim interès recau en la funció d'històric i de *font de dades* per sobre de les necessitats relacionals, s'ha de tenir en compte que com més informació es vol, més relacions s'acabaran establint, si no ara, en un futur. Ara per ara, les funcionalitats relatives a Jobs i Dades requereixen de relacions mentre que SAM no (tot i que podria fer-ho). La causa es troba en la necessitat de lligar cada punt d'un gràfic amb les dades que representa en el següent nivell de granularitat. En el cas de SAM no és necessari perquè:

- Els valors de primer nivell s'insereixen al mateix temps que els de segon nivell.
- En ser la representació una taula, la selecció de l'últim valor de SAM ha de correspondre amb l'últim valor de segon nivell.
- Les insercions son cada 30 minuts.

El diagrama entitat-relació de la base de dades (sencera) es troba detallada en l'apèndix B.

Capítol 5

Implementació

5.1 Pla d'Implementació

Tal i com està planificat (avantprojecte, requeriments i disseny en parlen), existeixen dues fases d'implementació: en la primera (derivada de l'aprovació del prototipus) es pretén codificar el que s'ha anomenat *primer nivell de monitorització* en el que s'han de mostrar, en format de gràfic, les dades necessàries per tal de realitzar un control inicial de la dinàmica dels *sites* i un de segon nivell per tal de puntualitzar quins son el punt/s que provoquen que les dades del primer nivell mostrin incoherències o errors.

5.2 Fases de la implementació

Primera etapa

- Creació dels col·lectors amb les següents funcionalitats bàsiques:
 - Creació de log per tal de conèixer els possibles errors que puguin sorgir. Aquí es fa referència a tots aquells errors que van més enllà de la pròpia aplicació com poden ser la falta d'espai en disc, un error de

connexió a la base de dades o bé un problema de xarxa que no permeti l'obtenció d'un recurs de la mateixa.

- Obtenció del primer recurs en format XML. S'emprarà la funcionalitat que el paquet PyCurl ofereix (emular l'ordre curl) a Python, evitant cridar a aquesta ordre continguda al sistema. Així es desvincula el col·lector del sistema operatiu.
 - Gestió del recurs XML. La funcionalitat `xml.dom.minidom` permet al desenvolupador moure's per l'estructura del fitxer XML des de l'arrel fins la més profunda de les seves fulles.
 - Establiment de connexió amb la base de dades. MySQLdb proporciona els mitjans per tal de generar una connexió i els cursors per tal d'executar queries.
 - Emmagatzematge de valors.
- Creació de la interfície bàsica:
 - Generació d'una primera versió del front-end (part visual) del entorn web.
 - Generació dels procediments per tal d'obtenir la informació de la base de dades. Aquí es construeixen els ponts cap a les dades: els RPC. Les dades del servidor de base de dades s'obtenen via `servlets`, s'emmagatzemen amb un objecte serialitzable i es retornen al client.
 - Desenvolupament dels procediments d'obtenció de `widgets` (gràfics, taules, elements dinàmics depenent del paquet `visualization`)
 - Primera idea del que ha de ser l'estructura final.

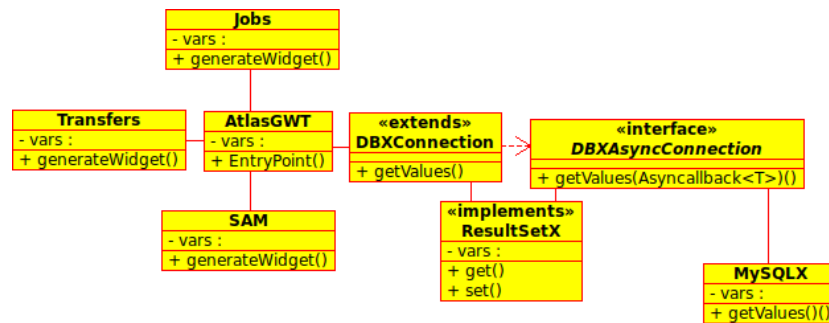


Figura 5.1: Representació de la primera etapa de la interfície web

Segona etapa

- Obtenció de la versió definitiva dels col·lectors:
 - Obtenció del nou recurs XML.
 - Gestió del nou recurs XML.
 - Establiment de les noves connexions amb la base de dades.
 - Emmagatzematge de valors.
- Els col·lectors (versió definitiva) arriben a gestionar dos fitxers XML de manera seqüencial, seguint el mateix mètode i procediments tant en la primera etapa com en la segona.
- Obtenció de la interfície definitiva:
 - Desenvolupament final del front-end (part visual) del entorn web.
 - Desenvolupament dels nous procediments d'obtenció de widgets.
 - Primera idea d'estructura final.
- La robustesa del model inicial permet repetir procediments i emprar alguns objectes de la primera etapa per a la segona. La diferència entre els dos models rau en com es representen els nous objectes: un popup.

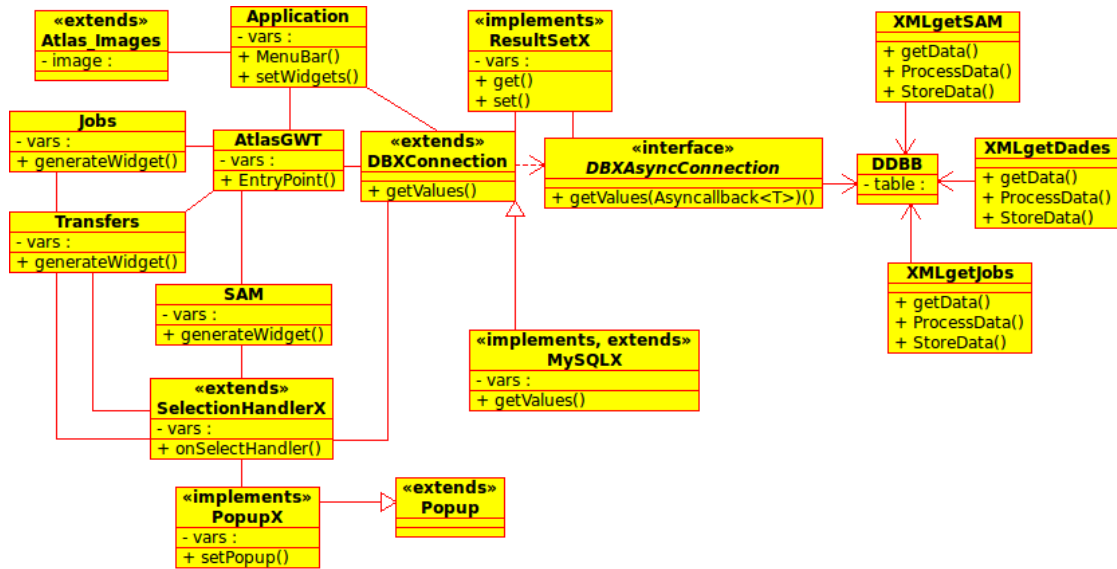


Figura 5.2: Representació (resumida) de l'estructura global

5.3 Interfície d'usuari

Tot i ser una versió “Beta”, que probablement acabarà patint canvis visuals ja es disposa d'una proposta ferma d'interfície. Les dades dels diversos jobs, transferències i checks del cloud son visibles a primer cop d'ull. Es mostren, inicialment, els últims 15 valors dels estats *efficiency*, *activated*, *running* i *failure* per a jobs i els últims 15 valors de *files*, *files errors* i *efficiency* de les transferències, continguts als sistema, i la taula amb el valor global de l'últim conjunt de checks als serveis de cada site.

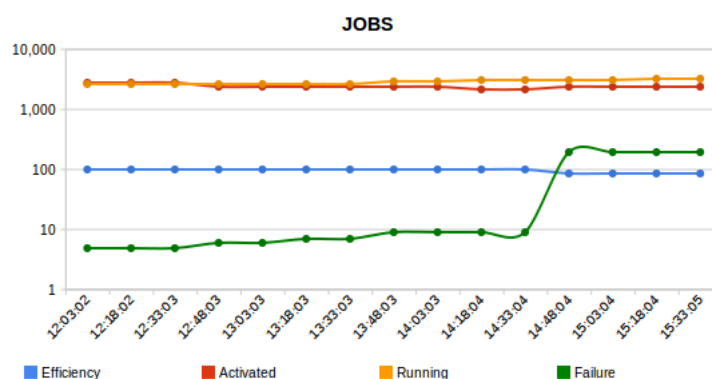


Figura 5.3: Visió d'un dels gràfics (Jobs)

Cada gràfic permet (clicant a qualsevol dels seus punts) obtenir els valors de tots els sites del estat concret.

En el cas de la taula de checks (SAM), si es clica sobre el valor de la fila, s'obté informació de cadascun dels serveis.

L'entorn permet a més:

- Veure gràfiques de Panda i del DDM (ja creades).
- Obtenir una visió temporal de l'estat dels jobs i les transferències en trams de 1, 4, 12 i 24 hores.
- Obtenir una visió del throughput de les transferències.

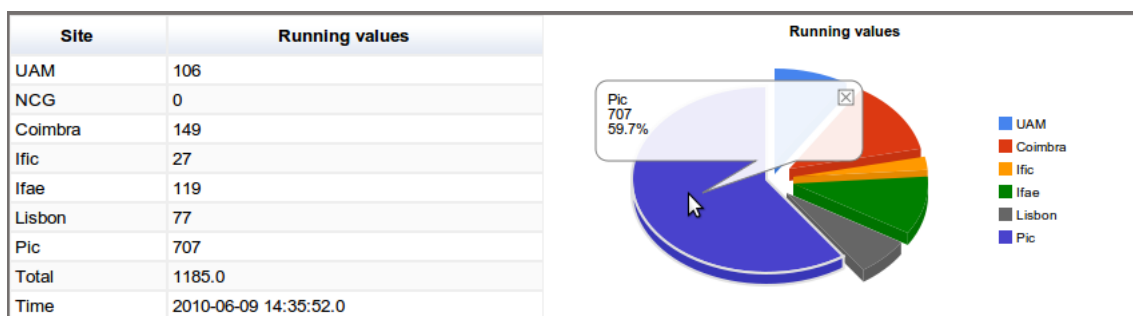


Figura 5.4: Visió d'un clic a un dels punts del estat *Running*

SAM

Site	Status
PIC	error
IFAE	error
IFIC	ok
Coimbra	ok
Lisbon	error
NCG	ok
UAM	ok

Figura 5.5: Detall de la taula de *checks* SAM

- Veure el valor acumulat de tots els jobs/ transferències fins al moment.

5.4 Dificultats sorgides a la implementació

5.4.1 Col·lectors

- Recollida de dades – El primer problema (i el més important) era com obtenir les dades base, les dades que havien de servir per formar qualsevol taula/gràfic/figura/informe. La primera idea es basava en realitzar queries a la BBDD de producció del nivell 0 (CERN) però es va desestimar ràpidament per que el nostre sistema té una importància de segon ordre i, per tant, no hauria de competir per recursos destinats a sistemes de primer ordre. Així doncs, ens calia trobar fonts de dades amb accés a aquest primer nivell a les que si es pogués accedir. Un cop trobades calia saber el format/forma d'accés a aquestes dades. Després de mails, peticions i preguntes, es van establir 4 sites com a possibles fonts de dades. Aquestes dades arribarien en format XML i d'altres via SMTP. Un cop depurada aquesta primera tria, es van establir 3 fonts de dades i totes via XML, una de les quals requereix autorització expressa i, ara per ara, es fa a través d'una màquina secundària.
- Tractament dels XML – Superat aquest aspecte, calia veure com fer el tractament d'aquest tipus de dades, fet (un dels fets si més no) important per tal de decidir el llenguatge dels col·lectors. Python vas ser l'escollit, llenguatge mai emprat pel desenvolupador del projecte. A favor de Python hi ha les diverses raons ja discutides a l'apartat de l'avantprojecte i que és un llenguatge àmpliament emprat en l'entorn WLCG.

5.4.2 Web

- Presentació dels resultats – La monitorització dels diferents *sites* requeria gràfics. Un cop provades diverses opcions, es va optar per la més dinàmica, per aquella que oferia més possibilitats envers les altres (tant d'interactivitat com de implementacions futures) la més espectacular, la de millor rendiment però una de les més complexes alhora de modificar, basada en GWT,

un entorn relativament desconegut pel desenvolupador i relativament nou en el món del desenvolupament (però en constant creixement).

- GWT – Com ja s'ha dit en el punt anterior, és en si mateix un rept. El cas es que un dels problemes rau en la poca informació que hi ha (per ser una tecnologia relativament nova) i per la informació *out of date* (per ser una tecnologia amb updates cada mes o dos mesos)
- Connexió amb la BBDD: els RPC – Els procediments remots són la base de la connexió entre el model de client/servidor que GWT segueix. La seva comprensió, així com els seus límits, són bàsics per avançar en aquest projecte. Per obtenir dades de la BBDD, cal establir un pont entre les classes del servidor que poden executar consultes a la BBDD i les classes del client que empren aquests resultats.
- Estructura del projecte – Un cop aprovat el prototipus, calia començar des de 0 a codificar el que seria la versió definitiva. Calia decidir una estructura que permetés que l'aplicació fos eficient alhora que es mantenien els paradigmes de l'orientació a objecte i una certa lògica respecte del que es buscava.

5.4.3 Problemes puntuals d'Implementació a qualsevol punt de l'entorn del sistema

- Els XML han de contenir informació relativa a cada paquet que s'afegeix. Per exemple: `curl -H 'Accept: text/xml' 'http://dashb-atlas-data.cern.ch/dashboard/request.py/site' > transfers.xml`
- Alguns XML (SAM) no presentaven cap valor en algunes branques, deixant només el tag d'inici `<Efficiency>` sense tancar quan el valor no s'ha computat correctament, provocant errors al intentar gestionar aquesta branca.

- Per defecte, MySQL emprava el motor MyISAM, que no permet (de manera nativa) l'ús de *Foreign Keys*, entre d'altres elements del entorn relacional. Així doncs, certes taules tenen un motor diferent (InnoDB) i, per tant, una gestió dels recursos particular.
- Els gràfics emprats no pertanyen de manera nativa a GWT, son un paquet afegit amb la seva documentació a part.
- Segons quins canvis a GWT, requereixen que el desenvolupador toqui el fitxer XML del projecte (que descriu, bàsicament, totes les herències)
- Les dades que es reben/envien via RPC han de ser serialitzables

Capítol 6

Proves i/o Tests

6.1 Pla de test

Per tal d'anar provant les funcionalitats afegides segons el pla d'implementació, s'han dissenyat una sèrie de casos de test que s'han aplicat al llarg del desenvolupament per tal de comprovar que el sistema sigui efectiu davant de les situacions d'error previstes.

1. Comprovació de PyCurl

- Verificació de la funcionalitat que PyCurl ofereix i que no és altre que l'obtenció d'un recurs en format XML via petició HTTP.
 - Execució de l'entorn un cop definida la URL i l'escriptura de la petició en format text/XML
 - Comprovar que existeix un fitxer XML a la ruta establerta que contingui la petició esmentada segons l'anterior URL
- Resultats: PyCurl respon correctament i obté el fitxer demanat en el format desitjat.

2. Control i gestió de les dades

- Cas per tal de conèixer si el tractament i emmagatzematge de les dades es correcte.
 - Execució del entorn (assumint que el cas anterior fos correcte)
 - Comprovar que els valors a MySQL responen als esperats.
- Resultats: MySQL respon com s'esperava i les dades tractades son les mateixes que les emmagatzemades.

3. Assaigs per tal de testar el comportament a nivell de *threads* de l'entorn

- Cas per comprovar si el *script* es comporta com un *thread* i realitza les crides tal i com estan programades.
 - Execució del entorn.
 - Comprovar els valors temporals via un `timer()`.
- Resultats: L'entorn realitza les crides en l'interval temporal definit.

4. Control d'errors `try/except`

- Cas per comprovar que tots els possibles errors amb els que el *script* es pot trobar, són capturats i loguejats.
 - Execució del entorn en condicions de privació de:
 - * Xarxa.
 - * Connexió a la base de dades.
 - * Capacitat de disc.
- Resultats: Els `try/except` es troben col·locats correctament i el log registra totes les entrades.

5. Control de la representació gràfica

- Cas per comprovar, en la primera part de la implementació, la correcta visualització dels valors a representar.

- Execució del entorn web.
- Comprovació de valors, posicions i interactivitat dels *widgets*.
- Resultats: Els gràfics mostren els valors adequats a la petició.

6. Control de la representació gràfica (fase 2)

- Cas per comprovar, relatiu al segon nivell de la implementació, la correcta visualització dels valors a representar.
 - Execució del entorn web.
 - Comprovació de valors, posicions i interactivitat dels *widgets*.
- Resultats: Els gràfics mostren els valors adequats a la petició.

7. Control d'errors: taules buides

- Cas per comprovar que si no tenim cap valor a les taules de la base de dades, el sistema es capaç de reaccionar davant aquest error.
 - Execució del entorn web.
 - Comprovació que els *widgets* mostren valors 0 i les taules mostren valors n/a.
- Resultats: Si no hi ha dades disponibles, la interfície ho reflecteix.

8. Control d'errors: connexió a la BBDD no disponible

- Cas per comprovar que si la instància de la base de dades està caiguda, el sistema és capaç de reaccionar davant aquest error.
 - Execució del entorn web.
 - Comprovació que el sistema respon amb un missatge d'error adequat, anunciant el problema.
- Resultats: Si la connexió no es pot establir, un missatge anuncia l'error de connexió.

Capítol 7

Conclusions

Un cop finalitzat el projecte, son diverses les conclusions que se'n poden treure. En primer lloc, com a estudiant, estic content per haver superat amb èxit un repte d'aquesta magnitud, content per què avui puc dir que soc coneixedor de tres noves tecnologies (LaTeX, Python i GWT), afegibles a les ja conegudes, que em permeten ésser més competitiu d'ara en endavant i orgullós per què la feina està ben feta, tenint com a prova d'aquesta afirmació, el vistiplau del personal del Port d'Informació Científica (PIC), el vistiplau dels delegats de diversos Tier-2, el vistiplau del meu director de projecte i la sensació pròpia que genera el treball ben fet. Del projecte en si mateix, les conclusions fan referència a la planificació temporal. Aquesta planificació ha estat parcialment modificada en desestimar l'execució d'una de les tasques planificades i, com en la majoria de projectes, per les petites variacions que ha patit. Les tasques relatives a l'apartat VO-BOX es van cancel·lar per no reunir les característiques necessàries per formar part del projecte, tot i estar planificades inicialment. El temps relatiu a aquestes tasques es va destinar a tasques posteriors i, en concret, a aquelles destinades a eixugar els petits retards que es poguessin generar durant el desenvolupament d'altres. Això i certs imprevistos tals com parades no programades, errors de difícil i llarga solució, petites modificacions en aspectes molt concrets del disseny i molt de temps dedicat a investigació, permeten concloure que les planificacions temporals son

sempre del tipus “best effort”, o aproximacions/esquemes del que en realitat acabarà succeint. Per sort, la disposició de certes tasques en el temps per contenir aquests possibles problemes, ha estat una idea força encertada. També m'agradaria destacar que aquest es un projecte necessari. Les fortes necessitats de tenir en control tots els fluxos de dades del *cloud*, requerien d'una eina que facilités la pressa de decisions. Cal recordar que son moltes les dades que es mouen al llarg del dia (i la nit) i son moltes i molt disperses les fonts que indiquen el seu estat.

Per últim, a l'apartat laboral i tal i com ja he dit a l'apartat d'objectius personals, l'experiència al PIC ha estat d'allò més enriquidora, obtenint coneixements que han anat més enllà d'aquells relatius al propi projecte. He pogut gaudir d'un entorn laboral real, interactuant amb grans professionals que m'han facilitat la meva tasca sempre que ha estat precís i que han respectat tant la meva feina com les diverses decisions preses.

Apèndix A

Glossari de Termes

AJAX – [19] Ajax, acrònim de Asynchronous JavaScript And XML, es una tècnica de desenvolupament web per a crear aplicacions interactives o RIA (Rich Internet Applications). Aquestes s’executen en el client, es a dir, al navegador dels usuaris mentre es manté la comunicació asíncrona amb el servidor en segon pla. D’aquesta forma es possible realitzar canvis sobre las pàgines sense necessitat de refrescar-les, el que significa augmentar la interactivitat, velocitat i usabilitat de les aplicacions.

ATLAS –[4] o A Toroidal LHC ApparatuS, es un dels experiments de partícules físiques al LHC del CERN. Aquest detector té com a missió la descoberta de quelcom nou via les col·lisions d’extraordinària energia de protons. Aquest detector aprendrà sobre les forces bàsiques que donen forma al Univers des del principi dels temps i que han de determinar el seu destí. Entre les possibles incògnites a descobrir trobem; el origen de la massa, les dimensions “extres”, la unificació de les forces fonamentals en un model i l’evidència de matèria obscura.

CERN – [5] o the European Organization for Nuclear Research, es un dels centres científics de recerca més grans i més respectats del planeta. Les seves investigacions es focalitzen en la física fonamental, cercant tot allò relatiu al Univers. A tra-

vés de diversos instruments de gran complexitat, estudien els constituents bàsics de la matèria; les partícules fonamentals. Estudiant que succeeix quan aquestes col·lideixen, els físics aprenen sobre les diverses lleis que regulen la natura.

CMS – [6] o Compact Muon Solenoid, es un altre dels experiments de partícules físiques al LHC del CERN. El CMS està designat per detectar un rang molt ampli de partícules i fenòmens que es puguin rebel·lar durant les col·lisions d'alta energia de protons.

Dashboard Data Monitoring – [3] El projecte Dashboard es desenvolupa per el CERN IT/GS group i es part activa del projecte EGEE (Enabling Grids For E-Science). Aquest software permet obtenir informació relativa als jobs de la grid i les transferències de diverses fonts i localitzacions, presentant a l'usuari final del grid, una representació processada de les dades via web.

Grid – [20] La computació grid es una tecnologia innovadora que permet utilitzar de forma coordinada tot tipus de recursos (entre ells còmput, emmagatzemament i aplicacions específiques) que no estan subjectes a un control centralitzat. En aquest sentit es una nova forma de computació distribuïda, en la qual els recursos poden ésser heterogenis.

GWT – [13] o Google Web Toolkit és un framework creat per Google que permet ocultar la complexitat de diversos aspectes de la tecnologia AJAX. És compatible amb diversos navegadors, la qual cosa és notori ja que cada navegador sol necessitar codi específic per a assolir un front-end correcte en una aplicació web. El concepte de Google Web Toolkit és bastant senzill, bàsicament el que s'ha de fer és crear el codi en Java usant qualsevol entorn de desenvolupament (IDE) de Java i el compilador ho traduirà a HTML i Javascript.

InnoDB – [22] és un motor de base de dades per a MySQL el gestor de base

de dades distribuït per MySQL AB. El principal avantatge respecte a altres sistemes proporcionats per MySQL és que proporciona transaccions que compleixen amb ACID (similar al que fa PostgreSQL) i també suporta claus foranies (integritat referencial declarativa).

LHC – [7] o Large Hadron Collider, es un instrument científic gegant prop de Ginebra, que s'expandeix entre Suïssa i França a 100 m sota terra. Es un accelerador de partícules de prop de 27 Km de circumferència emprat per els físics per a estudiar les partícules conegudes més petites, els blocs fonamentals de la matèria.

MyISAM – [23] és el sistema d'emmagatzemament per defecte del sistema gestor de base de dades MySQL. Està basat en el sistema ISAM de IBM però proporciona múltiples i útils extensions.

Panda – [8] o the Production ANd Distributed Analysis system es un sistema desenvolupat per a ATLAS per tal d'assolir els requeriments d'un sistema de gestió de càrregues de treball, orientat a dades per a la producció i el processament d'anàlisis distribuïts, capaç d'operar sobre els nivells de processat de dades del LHC i ser interoperables amb els diferents grids que comprenen el LHC (EGEE, OSG, NDGF).

PHP – [14] o Hypertext PreProcessor, es un llenguatge de scripting molt emprat, de propòsit general, especialment dissenyat per al desenvolupament web amb capacitat per a ser incrustat sobre HTML.

SAM - Portal que permet medir la disponibilitat dels diversos sites dintre del Grid.

RPC – [24] o Remote Procedure Call, de l'anglès Crida a Procediment Remot, és una tecnologia que permet a un programa d'ordinador fer que una subrutina o procediment s'executi en un altre espai d'adreces (habitualment en un altre or-

dinador en una xarxa compartida) sense que el programador hagi de programar explícitament els detalls d'aquesta interacció remota. És a dir, el programador escriuria essencialment el mateix codi tant si la subrutina és local al programa executant, o remota.

Widgets – [25] es una petita aplicació o programa, usualment presentat en arxius o fitxers petits que son executats por un motor de widgets o Widget Engine. Entre els seus objectius estan els de donar fàcil accés a funcions freqüentment emprades i proveir de informació visual.

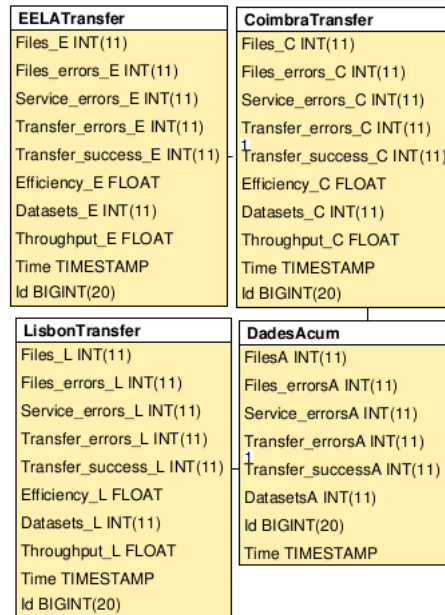
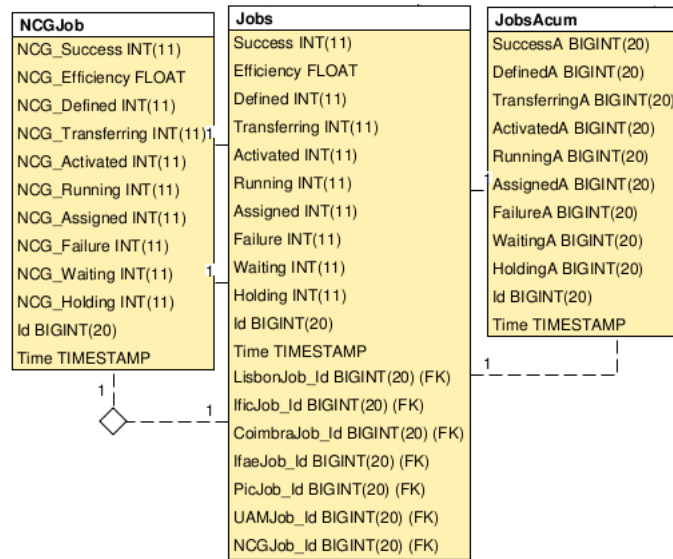
WLCG – [10] o Worldwide LHC Computing Grid, es una infraestructura Grid de col·laboració global formada per multitud de centres per tot el globus. Existeix per tal de distribuir, emmagatzemar i analitzar les ingents quantitats de dades generades per el LHC.

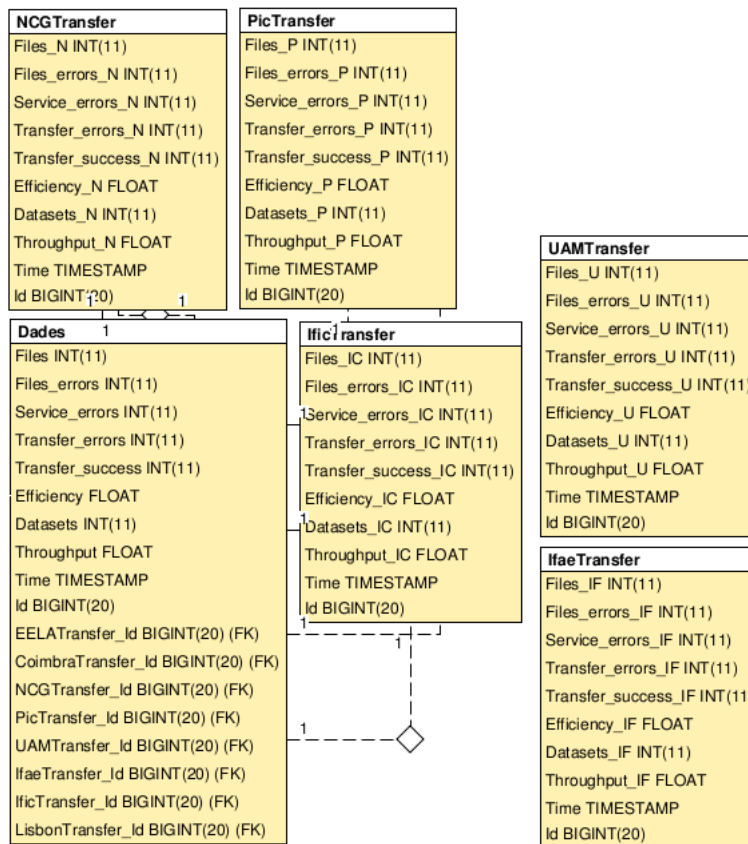
Apèndix B

Esquema ER de la BBDD

LisbonJob	IficJob	CoimbraJob
Lisbon_Success INT(11)	Ific_Success INT(11)	Coimbra_Success INT(11)
Lisbon_Efficiency FLOAT	Ific_Efficiency FLOAT	Coimbra_Efficiency FLOAT
Lisbon_Defined INT(11)	Ific_Defined INT(11)	Coimbra_Defined INT(11)
Lisbon_Transferring INT(11)	Ific_Transferring INT(11)	Coimbra_Transferring INT(11)
Lisbon_Activated INT(11)	Ific_Activated INT(11)	Coimbra_Activated INT(11)
Lisbon_Running INT(11)	Ific_Running INT(11)	Coimbra_Running INT(11)
Lisbon_Assigned INT(11)	Ific_Assigned INT(11)	Coimbra_Assigned INT(11)
Lisbon_Failure INT(11)	Ific_Failure INT(11)	Coimbra_Failure INT(11)
Lisbon_Waiting INT(11)	Ific_Waiting INT(11)	Coimbra_Waiting INT(11)
Lisbon_Holding INT(11)	Ific_Holding INT(11)	Coimbra_Holding INT(11)
Id BIGINT(20)	Id BIGINT(20)	Id BIGINT(20)
Time TIMESTAMP	Time TIMESTAMP	Time TIMESTAMP

IfaeJob	PicJob	UAMJob
Ifae_Success INT(11)	Pic_Success INT(11)	UAM_Success INT(11)
Ifae_Efficiency FLOAT	Pic_Efficiency FLOAT	UAM_Efficiency FLOAT
Ifae_Defined INT(11)	Pic_Defined INT(11)	UAM_Defined INT(11)
Ifae_Transferring INT(11)	Pic_Transferring INT(11)	UAM_Transferring INT(11)
Ifae_Activated INT(11)	Pic_Activated INT(11)	UAM_Activated INT(11)
Ifae_Running INT(11)	Pic_Running INT(11)	UAM_Running INT(11)
Ifae_Assigned INT(11)	Pic_Assigned INT(11)	UAM_Assigned INT(11)
Ifae_Failure INT(11)	Pic_Failure INT(11)	UAM_Failure INT(11)
Ifae_Waiting INT(11)	Pic_Waiting INT(11)	UAM_Waiting INT(11)
Ifae_Holding INT(11)	Pic_Holding INT(11)	UAM_Holding INT(11)
Id BIGINT(20)	Id BIGINT(20)	Id BIGINT(20)
Time TIMESTAMP	Time TIMESTAMP	Time TIMESTAMP





NCGAtSAM	LisbonAtSAM	IfaeAtSAM	CoimbraAtSAM
NCGGngRbt01 VARCHAR(6)	LisbonGngRbt01 VARCHAR(6)	IfaeGngRbt01 VARCHAR(6)	CoimbraGngRbt01 VARCHAR(6)
NCGFrontSquid01 VARCHAR(6)	LisbonGngRbt02 VARCHAR(6)	IfaeGngRbt02 VARCHAR(6)	CoimbraFrontSquid01 VARCHAR(6)
NCGLcgtag01 VARCHAR(6)	LisbonFrontSquid01 VARCHAR(6)	IfaeFrontSquid01 VARCHAR(6)	CoimbraLcgtag01 VARCHAR(6)
NCGJob01 VARCHAR(6)	LisbonFrontSquid02 VARCHAR(6)	IfaeFrontSquid02 VARCHAR(6)	CoimbraJob01 VARCHAR(6)
NCGSwdir01 VARCHAR(6)	LisbonLcgtag01 VARCHAR(6)	IfaeLcgtag01 VARCHAR(6)	CoimbraSwdir01 VARCHAR(6)
NCGLcgcp VARCHAR(6)	LisbonLcgtag02 VARCHAR(6)	IfaeLcgtag02 VARCHAR(6)	CoimbraLcgcp VARCHAR(6)
NCGLcgcr VARCHAR(6)	LisbonJob01 VARCHAR(6)	IfaeJob01 VARCHAR(6)	CoimbraLcgcr VARCHAR(6)
NCGLcgdel VARCHAR(6)	LisbonJob02 VARCHAR(6)	IfaeJob02 VARCHAR(6)	CoimbraLcgdel VARCHAR(6)
NCGStatus VARCHAR(6)	LisbonSwdir01 VARCHAR(6)	IfaeSwdir01 VARCHAR(6)	CoimbraStatus VARCHAR(6)
Id BIGINT(20)	LisbonSwdir02 VARCHAR(6)	IfaeSwdir02 VARCHAR(6)	Id BIGINT(20)
	LisbonLcgcp VARCHAR(6)	IfaeLcgcp VARCHAR(6)	
	LisbonLcgcr VARCHAR(6)	IfaeLcgcr VARCHAR(6)	
	LisbonLcgdel VARCHAR(6)	IfaeLcgdel VARCHAR(6)	
	LisbonStatus VARCHAR(6)	IfaeStatus VARCHAR(6)	
	Id BIGINT(20)	Id BIGINT(20)	

UAMAtSAM	IficAtSAM	SAM	PicAtSAM
UAMGngRbt01 VARCHAR(6)	IficGngRbt01 VARCHAR(6)	Id BIGINT(20)	PicGngRbt05 VARCHAR(6)
UAMFrontSquid01 VARCHAR(6)	IficGngRbt03 VARCHAR(6)	pic_st VARCHAR(6)	PicGngRbt06 VARCHAR(6)
UAMLcgtag01 VARCHAR(6)	IficGngRbt04 VARCHAR(6)	ifae_st VARCHAR(6)	PicGngRbt07 VARCHAR(6)
UAMJob01 VARCHAR(6)	IficFrontSquid01 VARCHAR(6)	ific_st VARCHAR(6)	PicFrontSquid05 VARCHAR(6)
UAMSwdir01 VARCHAR(6)	IficFrontSquid03 VARCHAR(6)	coimbra_st VARCHAR(6)	PicFrontSquid06 VARCHAR(6)
UAMLcgcp VARCHAR(6)	IficFrontSquid04 VARCHAR(6)	lisbon_st VARCHAR(6)	PicFrontSquid07 VARCHAR(6)
UAMLcgcr VARCHAR(6)	IficLcgtag01 VARCHAR(6)	ncg_st VARCHAR(6)	PicLcgtag05 VARCHAR(6)
UAMLcgdel VARCHAR(6)	IficLcgtag03 VARCHAR(6)	uam_st VARCHAR(6)	PicLcgtag06 VARCHAR(6)
UAMStatus VARCHAR(6)	IficLcgtag04 VARCHAR(6)		PicLcgtag07 VARCHAR(6)
Id BIGINT(20)	IficJob01 VARCHAR(6)		PicJob05 VARCHAR(6)
	IficJob03 VARCHAR(6)		PicJob06 VARCHAR(6)
	IficJob04 VARCHAR(6)		PicJob07 VARCHAR(6)
	IficSwdir01 VARCHAR(6)		PicSwdir05 VARCHAR(6)
	IficSwdir03 VARCHAR(6)		PicSwdir06 VARCHAR(6)
	IficSwdir04 VARCHAR(6)		PicSwdir07 VARCHAR(6)
	IficLcgcp VARCHAR(6)		Picfts VARCHAR(6)
	IficLcgcr VARCHAR(6)		PicLcgcp VARCHAR(6)
	IficLcgdel VARCHAR(6)		PicLgcr VARCHAR(6)
	IficStatus VARCHAR(6)		PicLgdel VARCHAR(6)
	Id BIGINT(20)		PicStatus VARCHAR(6)
			Id BIGINT(20)

Apèndix C

Paper presentat a CHEP 2010

El CHEP (Conference on Computing in High Energy and Nuclear Physics) es una serie de conferències per a físics i professionals de la Informàtica relacionats amb les altes energies i les físiques nuclears, la informàtica i les tecnologies de la informació. A continuació, es mostra el *abstract* (resum) presentat a la conferència que es durà a terme a Taiwan del 18 al 22 d'Octubre:

Title: Integrated experiment activity monitoring for wLGC sites based on GWT

Abstract content LHC is well on the way since early November 2009 and experiments are computing and delivering data almost uninterruptedly. The wLGC Tier-1s started the operations in 2006 but they are now in full production since the LHC start. One of the peculiarities about the Tier-1s is that big fraction of them give support to several LHC experiments: ATLAS, CMS, LHCb or ALICE; each one of these having its intrinsic workflows for the different fundamental experiment activities: data simulation, real data reprocessing, user analysis and data distribution. On top of that, wLGC project is constantly monitoring the performance of the sites by sending job probes either from a general point of view (experiment transversal probes) or experiment-oriented probes. The consequence is that sites end up looking at a plethora of monitoring pages without having an overall inte-

grated view, which would make life easier to site-admins and experiment experts at the sites. The nature of ATLAS Computing Model, based on computing clouds formed by a Tier-1 and some Tier-2s, lead us to develop the monitoring for an entire ATLAS cloud, the PIC cloud, that have 6 Tier-2s associated. In this paper an approach is described to merge all monitoring daily needs for an LHC experiment like ATLAS from the cloud point of view. The main idea is to develop a set of collectors to gather information for data distribution, data processing and wLCG tests (Service Availability Monitoring), store them in specific databases, wrap the results and show it in a single High Level Monitoring (HLM) page. Once having a HLM one can investigate further by interacting with the front-end which is fed by the stats stored at the databases. The project was developed using a browser-based development kit (Google Web Toolkit) which produces a high-performance AJAX front-end application written in JAVA, that runs across all major browsers, relying on a system of collectors that parses the information into a SQL database.

Bibliografía

- [1] Ingeniería del software. ciclo de desarrollo del software. Last Update - June 7, 2008, Url - http://acacha.org/wiki_privada/index.php/Usuario:Sergi/Temari/Tema_48_Ingenier%C3%ADa_del_%22software%22._Ciclo_de_desarrollo_del_%22software%22._Tipos_de_ciclos_de_desarrollo.#Prototipat, June 2010. (Last Access).
- [2] Open flash chart. Last Update - n/a, Url - <http://teethgrinder.co.uk/open-flash-chart-2/>, June 2010. (Last Access).
- [3] CERN. Atlas dashboard. Last Update - On demand, Url - <http://dashb-atlas-data.cern.ch/dashboard/request.py/site>, June 2010. (Last Access).
- [4] CERN. ATLAS Experiment. Last Update - n/a, Url - <http://atlas.ch/>, June 2010. (Last Access).
- [5] CERN. CERN in a nutshell. Last Update - 2008, Url - <http://public.web.cern.ch/public/en/About/About-en.html>, June 2010. (Last Access).
- [6] CERN. The cms experiment. Last Update - 2008, Url - <http://cms.web.cern.ch/cms/Detector/WhatCMS/index.html>, June 2010. (Last Access).

- [7] CERN. The large hadron collider. Last Update - 2008, Url - <http://public.web.cern.ch/public/en/LHC/LHC-en.html>, June 2010. (Last Access).
- [8] CERN. Panda monitor. Last Update - On demand, Url - <http://panda.cern.ch:25980/server/pandamon/query?dash=prod>, June 2010. (Last Access).
- [9] CERN. The panda production and distributed analysis system (wiki). Last Update - June 10, 2010, Url - <https://twiki.cern.ch/twiki/bin/view/Atlas/PanDA>, June 2010. (Last Access).
- [10] CERN. Worldwide LHC Computing Grid. Last Update - May 2009, Url - <http://lcg.web.cern.ch/LCG/public/>, June 2010. (Last Access).
- [11] UAB (CVC). Anàlisi de requeriments del software. Last Update - n/a, Url - <http://www.cvc.uab.es/shared/teach/a20363/introAN.pdf>, June 2010. (Last Access).
- [12] Benjamin Gaidioz. Arda dashboard production system monitoring. Last Update - 2007, Url - <http://dashb-build.cern.ch/build/nightly/doc/guides/production/html/user/index.html>, June 2010. (Last Access).
- [13] Google. Google web toolkit. Last Update - 2010, Url - <http://code.google.com/intl/ca/webtoolkit/overview.html>, June 2010. (Last Access).
- [14] The PHP Group. Php. Last Update - 2008, Url - <http://php.net/index.php>, June 2010. (Last Access).
- [15] Tobias Oetiker. Rrdtool. Last Update - September 12, 2009, Url - <http://oss.oetiker.ch/rrdtool/index.en.html>, June 2010. (Last Access).

- [16] PIC. Port d'informació científica. Last Update - n/a, Url - "http://www.pic.es/research_project_detalle.gsp?id=1", June 2010. (Last Access).
- [17] Reyes Grangel Seguer. Diagrama de casos d'ús. Last Update - November 2007, Url - http://mermaja.act.uji.es/itis/IS31/is310708_tema3.pdf, June 2010. (Last Access).
- [18] Toni Sellarès. Definició de requeriments, anàlisi i disseny orientats a objectes. Last Update - n/a, Url - <http://ima.udg.edu/~sellaes/ETIG-ES/RADoo.pdf>, June 2010. (Last Access).
- [19] Wikipedia. AJAX. Last Update - June 10, 2010, Url - <http://es.wikipedia.org/wiki/AJAX>, June 2010. (Last Access).
- [20] Wikipedia. Computación grid. Last Update - June 5, 2010, Url - http://es.wikipedia.org/wiki/Computaci%C3%B3n_grid, June 2010. (Last Access).
- [21] Wikipedia. Hierarchical linear modeling. Last Update - June 17, 2010, Url - http://en.wikipedia.org/wiki/Hierarchical_linear_modeling, June 2010. (Last Access).
- [22] Wikipedia. Innodb. Last Update - October 30, 2009, Url - <http://ca.wikipedia.org/wiki/InnoDB>, June 2010. (Last Access).
- [23] Wikipedia. Myisam. Last Update - October 30, 2009, Url - <http://ca.wikipedia.org/wiki/MyISAM>, June 2010. (Last Access).
- [24] Wikipedia. Remote procedure call. Last Update - April 19, 2010, Url - <http://php.net/index.php>, June 2010. (Last Access).
- [25] Wikipedia. Widget. Last Update - June 1, 2010, Url - <http://php.net/index.php>, June 2010. (Last Access).

Resum

In this project an approach is described to merge all monitoring daily needs for an LHC experiment like ATLAS from the cloud point of view. The main idea is to develop a set of collectors to gather information for data distribution, data processing and wLCG tests (Service Availability Monitoring), store them in specific databases, wrap the results and show it in a single High Level Monitoring (HLM) page. Once having a HLM one can investigate further by interacting with the front-end which is fed by the stats stored at the databases.

Aquest projecte descriu la fusió de les necessitats diàries de monitorització del experiment ATLAS des del punt de vista del cloud. La idea principal es desenvolupar un conjunt de col·lectors que recullin informació de la distribució i processat de les dades i dels test de wlcg (Service Availability Monitoring), emmagatzemant-la en BBDD específiques per tal de mostrar els resultats en una sola pàgina HLM (High Level Monitoring). Un cop aconseguit, l'aplicació ha de permetre investigar més enllà via interacció amb el front-end, el qual estarà alimentat per les estadístiques emmagatzemades a la BBDD.

Este proyecto describe la fusión de las necesidades diarias de monitorización del experimento ATLAS desde el punto de vista del cloud. La idea principal es desarrollar un conjunto de colectores que recojan información de la distribución y procesado de los datos y los tests wlcg (Service Availability Monitoring), almacenándola en BBDD específicas para mostrar los resultados en una sola página HLM (High Level Monitoring). Una vez conseguido, la aplicación debe permitir investigar mas allá vía interacción con el front-end, el cual estará alimentado por las estadísticas almacenadas en la BBDD.