



UNIVERSITAT OBERTA DE CATALUNYA

ESTUDIS D'ENGINYERIA EN INFORMÀTICA

Projecte Fi de Carrera

APLICACIONS WEB PER TREBALL COL·LABORATIU

**“DISSENY I PROTOTIPATGE D'UN SISTEMA
D'EVENTS PER A DISPOSITIUS MÒBILS EN EL
TREBALL EN GRUP”**

Memòria

Consultor: Fatos Xhafa

Alumne: Xavier Rivadulla Diaz

E-mail: xrivadulla@uoc.edu

LLICÈNCIA

(Creative Commons)

Aquest treball està subjecte - excepte que s'indiqui el contrari- en una llicència de Reconeixement-NoComercial-SenseObraDerivada 2.5 Espanya de Creative Commons.

Podeu copiar-lo, distribuir-lo i transmetre'ls públicament sempre que **citeu l'autor i l'obra, no es faci un ús comercial i no es faci còpia derivada**.

La llicència completa es pot consultar en

<http://creativecommons.org/licenses/by-nc-nd/2.5/es/deed.ca>

A la Susagna, el Marc, la Laia i l'Abril per tot el suport que m'han donat.

ÍNDEX

1. Introducció	7
2. Estructura de la memòria	7
3. Descripció del Projecte	8
3.1 Objectius	8
3.2 Resultats Esperats	8
3.3 Motivació	9
3.4 Metodologia emprada	9
3.5 Tecnologia emprada	9
3.6 Justificació de les principals tecnologies	10
4. Estimació i planificació del Projecte	11
4.1 Planificació temporal	11
4.2 Fites principals	11
4.3 Diagrama Gantt de temporització de tasques	11
5. Organització del Projecte	12
5.1 Relació de tasques	12
5.2 T1 Elecció proposta	12
5.3 Fase 1	12
5.3.1 T2 Elaboració del pla de treball.....	12
5.3.2 T3 Instal·lació i documentació de les tecnologies.....	12
5.3.3 T3.2 Estudi i instal·lació servidor Linux-Cherokee/Apache-PHP	12
5.3.4 T3.2 Estudi i instal·lació de la base de dades noSQL MongoDB	12
5.3.5 T4 Anàlisi de requeriments i disseny	12
5.3.6 Lliurable.....	12
5.4 Fase 2	13
5.4.1 T5 Estudi de patrons de programació que puguin ser aplicables al projecte	13
5.4.2 T6 Estudi de sistemes d'events i integració amb dispositius mòbils des de plataformes web pel treball en grup.....	13
5.4.3 T7 Proves de concepte	13
5.4.4 T8 Implementació pràctica del prototip.....	13
5.4.5 T8.1 Implementació gestió d'usuaris i grups.....	13
5.4.6 T8.2 Implementació gestió events	13
5.4.7 T7.3 Implementació gestió subscripció events	13
5.4.8 T8.4 Implementació sistema d'enviament events.....	13
5.4.9 T9 Inici proves de funcionament	13
5.4.10 T9.1 Proves gestió d'usuaris i grups.....	13
5.4.11 T9.2 Proves gestió events.....	13
5.4.12 T9.3 Proves gestió subscripció events.....	13
5.4.13 T9.4 Proves motor events	14
5.4.14 T10 Desenvolupament i prototipatge.....	14
5.4.15 Lliurables	14
5.5 Fase 3	14
5.5.1 T11 Documentació del projecte	14
5.5.2 T12 Publicació del prototip.....	14
5.5.3 T13 Lliurament final	14
5.5.4 Lliurables.....	14
6. Equip de treball i implicats	15

7. Anàlisi de riscos.....	15
8. Pla de qualitat	16
8.1 Política.....	16
8.2 Responsables.....	17
8.3 Processos de qualitat.....	17
8.3.1 Processos de qualitat de la totalitat del projecte.....	17
8.3.2 Processos específics de cada activitat	17
8.4 Estàndards que s'han d'aplicar	17
8.4.1 Documentació	17
8.4.2 Programari	18
8.5 Criteris d'acceptació	18
8.6 Criteris de qualitat global	18
9. Anàlisi de requeriments.....	19
9.1 Mòdul de gestió d'usuaris/grups.....	19
9.2 Mòdul de gestió d'events	20
9.3 Mòdul de gestió de subscripció d'events	21
9.4 Mòdul sistema d'enviaments d'events.....	21
10. Consideracions sobre els requeriments.....	22
11. Plataforma i eines de desenvolupament	23
11.1 Introducció	23
11.2 Instal·lació i documentació del entorn.....	24
11.2.1 easyPHP	24
11.2.2 mongoDB	27
11.2.3 Eclipse.....	29
11.2.4 Egit	29
11.2.5 Navegadors Web	33
12. Patrons de disseny emprats en el desenvolupament.....	34
12.1 Introducció	34
12.2 Model-vista-controlador	34
12.3 Singleton.....	35
12.4 Front Controller	35
13. Anàlisi del sistema	36
13.1 Introducció	36
13.2 Arquitectura del sistema	36
13.3 Diagrama de components.....	37
13.3.1 Capa de presentació.....	37
13.3.2 Capa de negoci.....	38
13.3.3 Capa integració	38
13.3.4 Diagrama de components.....	38
13.4 Model de dades.....	40
13.4.1 Diagrama de dades.....	40
13.4.2 MongoDB	41
13.5 Casos d'ús	48
13.5.1 Introducció	48
13.5.2 Casos d'ús de gestió del sistema	48
13.5.3 Casos d'ús gestió d'events	49
13.5.4 Casos d'ús per sistemes externs	50
14. Interfícies.....	51
14.1 Interfícies gràfica	51
14.1.1 Llistes	51

14.1.2	Grups	52
14.1.3	Usuaris.....	52
14.1.4	Context.....	53
14.1.5	Objectes.....	53
14.1.6	Retroaccions.....	54
14.1.7	Events.....	54
14.1.8	Subscripcions.....	55
14.2	Interfícies de servei web	56
14.2.1	PULL.....	56
14.2.2	Observador remot	57
14.2.3	Subscripcions remotes	57
14.3	Access via navegador mòbil	58
15.	Conclusions	59
16.	Possibles millores del sistema	60
17.	Vocabulari	61
18.	Bibliografia	62
18.1	Eines	62
18.2	Llibreries externes utilitzades	62
18.3	Informació	62
18.4	Articles	62

1. Introducció

Aquest memòria ha estat realitzada per Xavier Rivadulla Díaz, alumne dels estudis d'Enginyeria en Informàtica de la UOC on s'exposa la proposta presentada i acceptada pel tutor Fatos Xhafa.

Aquest projecte es realitzarà dins l'àrea d'Aplicacions Web per al Treball Col·laboratiu i de les propostes disponibles per aquesta àrea s'ha elegit fer : **“Disseny i prototipatge d'un sistema d'events per a dispositius mòbils en el treball en grup.”**.

Un aspecte important dels sistemes online de treball en grup és l'anomenat "awareness", és a dir mantenir informats els membres del grup de tot allò que ocórrer en l'espai de grup. La premissa és que si els membres del grup estan informats sobre el que passa dins l'espai del grup, com ara lectura de documents, modificació de documents, nous documents, noves fites del calendari, etc., doncs això pot incrementar la productivitat del grup i per tant els resultats acadèmics. Aquest enfocament ja s'aplica en sistemes web. Aquest projecte tractaria d'estudiar com estendre aquest model a dispositius mòbils per tal d'aplicar així el paradigma "anytime, anywhere", és a dir estar informat en tot moment i en qualsevol lloc.

2. Estructura de la memòria

El capítols que aquesta memòria conté poden ser estructurats de la següent manera:

- Descripció del projecte
- Planificació, organització i anàlisi de requeriments del projecte
- Anàlisi del sistema: Arquitectura i disseny del projecte
- Conclusions i apèndix.

Queden fora d'aquesta memòria els manuals d'usuari i proves d'avaluació del prototipus, que es documenten a banda.

3. Descripció del Projecte

3.1 Objectius

Es pretén fer una proposta d'arquitectura i prototipatge d'un sistema que permeti definir un model d'esdeveniments dins un espai de treball en grup que sigui aplicable per dispositius mòbils, i que a banda es pugui integrar amb plataformes web pel treball en grup.

Aquesta proposta d'arquitectura i prototipatge cobrirà els següents aspectes:

- Alta disponibilitat: l'arquitectura tindrà un disseny que asseguri la continuïtat del servei tot i la fallada d'alguna de les seves parts.
- Escalabilitat: l'arquitectura permetrà redimensionar els seu tamany depenent de les exigències demandades.
- Modularitat: l'arquitectura permetrà afegir noves funcionalitats, per exemples noves connexions a plataformes d'enviament de missatges, de forma senzilla.
- Seguretat: l'arquitectura assegurarà que els usuaris rebin només els avisos als quals s'han subscrit evitant rebre alarmes no desitjades.
- Independència de dispositiu mòbil: el projecte tindrà en compte els diferents tipus de dispositiu mòbils i assegurarà una cobertura universal.
- Obert a altres sistemes: el disseny a de garantir que el sistema pugui se emprat per tercers de forma fàcil, de manera que es pugui convertir en un HUB (eix) per altres sistemes.

3.2 Resultats Esperats

S'espera obtenir un coneixement ampli d'eines informàtiques Open Source disponibles al mercat, i veure el seu comportament en un entorn real, encara que limitat, per extreure les conclusions oportunes. En concret, s'espera:

- Desenvolupar el disseny/arquitectura proposat en el present document emprant les eines Open Source indicades.
- Implementació del prototip limitat.
- Comprovació de la integració de la plataforma desenvolupada en un entorn mòbil i amb plataformes de tercers, si es possible.
- Generar un procés i manual d'ús referent al funcionament d'aquest prototip, així com les valoracions de la plataforma dissenyada, destacant els punts dèbils o a millorar.

3.3 Motivació

La principal motivació del estudiant es aprofundir en una sèrie de coneixements en l'àrea mòbil, àrea en la que porta treballant professionalment des de fa uns anys, per tal d'oferir una solució Open Source que pugui fàcilment fer-se servir en altres projectes.

També hi ha un gran motivació per construir un sistema que sigui molt estàndard i on sigui fàcil la connexió amb qualsevol sistema d'usuaris/grups que haguin de ser notificats dels events als quals estan subscrits. L'ambició es tenir una API pública que permeti fer fàcil aquesta integració.

Òbviament les proves sobre dispositius mòbils tenen un cost i es farà el possible per tal de testejar el projecte, tot i que no es descarta que no sigui possible.

Per altra banda també resulta molt engrescador fer tot el cicle de vida d'un projecte, des de la tria fins el lliurament, ja que resulta complicat des del àmbit professional estar present en tots les etapes d'un projecte.

3.4 Metodologia emprada

Per aquest projecte s'utilitzarà una metodologia molt comuna, l'anomenada en cascada o clàssica. Aquesta defineix el projecte en si com un tot dividit en tasques i subtasques que es realitzaran en ordre i en el qual no se'n iniciarà una sense haver acabat l'anterior. Per altra banda només es disposa d'un únic recurs per fer el projecte, per la qual cosa la seqüencialitat es inevitable.

3.5 Tecnologia emprada

La tecnologia que s'utilitzarà durant el projecte serà:

- Ordinador personal iMac amb Mac OS X 10.7 amb Parallels Desktop per tal de tenir un entorn de màquines virtual on poder executar la distribució Linux Fedora.
- Altres ordinadors amb una distribució Linux per tal de demostrar la robustesa i escalabilitat del projecte.
- Ordinador amb Windows 7 per tal de fer el desenvolupament i provar la plataforma en entorn Windows.
- S'avaluarà Cherokee per tal de tenir un servidor web. També es farà ús del servidor web Apache per provar la portabilitat a altres servidors web.
- Mòduls de PHP pel servidor web per tal de poder programar el projecte.
- MongoDB per tal de tenir una solució per la persistència de dades.
- Gestor de control de versions.
- Programes d'edició.
- Programes ofimàtics per realitzar la documentació.

3.6 Justificació de les principals tecnologies

- Distribució Linux Fedora: Un dels principals requeriments del projecte es emprar Software Lliure, per tant la tria del sistema operatiu queda justificada per aquesta decisió.
- Servidor web Cherokee: aquest servidor web permet alta disponibilitat, balanceig i alta velocitat, per tant resulta una bona tria si es volen assolir els objectius del projecte.
- PHP: La curta durada del projecte fa que la tria del llenguatge de programació no pugui ser molt gran i s'hagi de triar el que el desenvolupador coneix més. Per altra banda PHP ha arribat a un grau de maduresa que ens permet trobar-lo en grans projecte, per la qual cosa la tria es considera correcta.
- MongoDB: la persistència de dades es un dels punts més crítics del projecte. Es vol poder tenir una gran flexibilitat i per altra banda es volen complir els objectius d'alta disponibilitat, balanceig de carga... Totes aquestes característiques les compleix MongoDB.

4. Estimació i planificació del Projecte

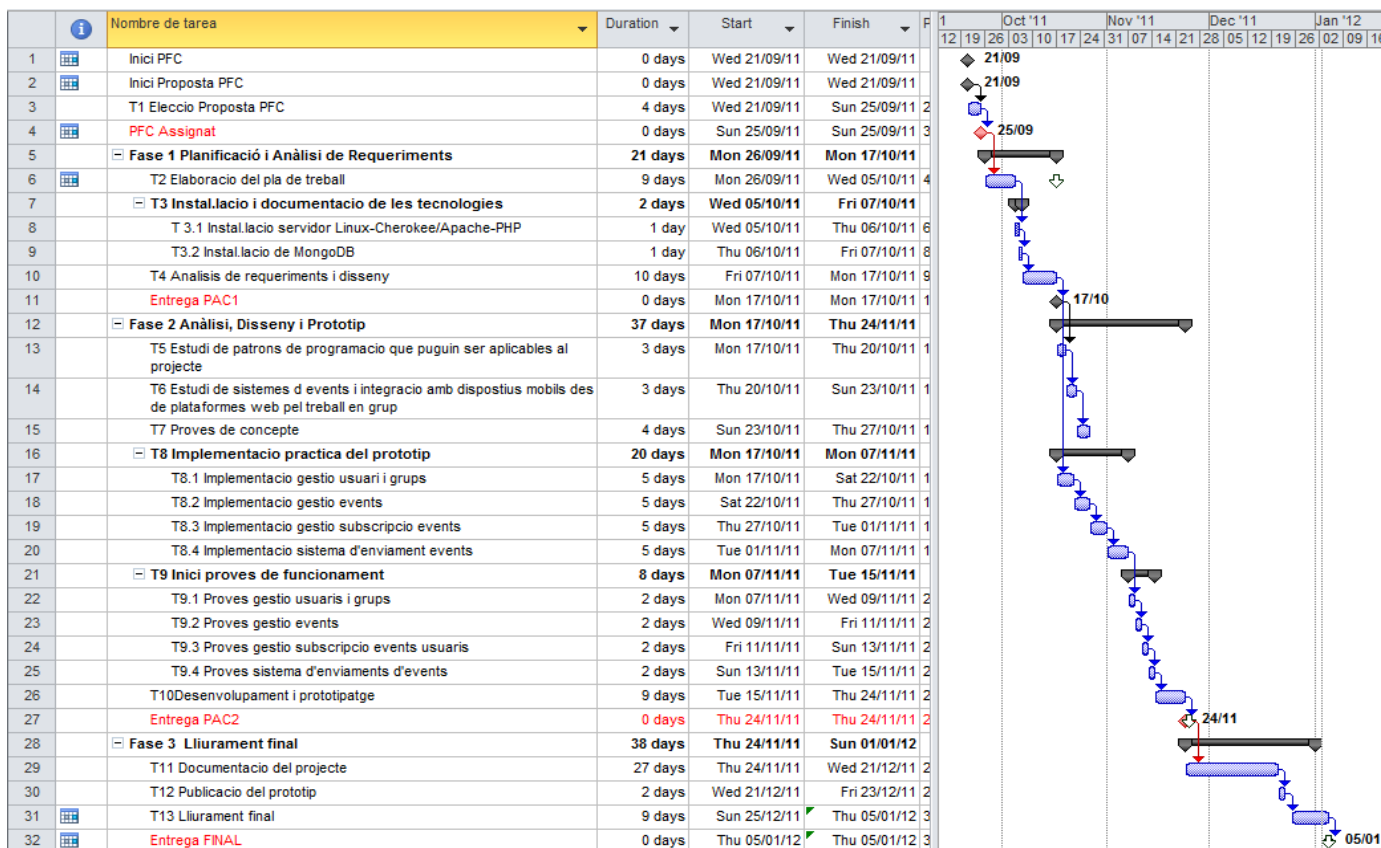
4.1 Planificació temporal

La estimació en hores de les tasques, la planificació temporal i les fites de control que s'estableixen es mostren en el diagrama de Gantt de l'apartat 3.3.

4.2 Fites principals

Data	Descripció Fita
21/09/2011	Inici PFC
21/09/2011	Formalització de la proposta
25/09/2011	PFC assignat
17/10/2011	PAC 1: Planificació i anàlisi de requeriments
25/11/2011	PAC 2: Desenvolupament del projecte
05/01/2012	PAC 3: Lliurament Final

4.3 Diagrama Gantt de temporització de tasques



5. Organització del Projecte

5.1 Relació de tasques

Al llarg del projecte s'aniran realitzant les tasques que hem vist temporitzades en el diagrama de GANTT. Aquestes seran:

5.2 T1 Elecció proposta.

Triar una de les propostes de treball disponibles a l'àrea i acordar-la amb el consultor.

5.3 Fase 1

5.3.1 T2 Elaboració del pla de treball

El document resultant haurà de deixar reflectit els objectius, els resultats esperats i la planificació de les tasques que es duran a terme per desenvolupar el Projecte.

5.3.2 T3 Instal·lació i documentació de les tecnologies.

En aquesta fase es realitza un estudi de les diferents tecnologies Open Source que s'utilitzaran per tal d'implementar el projecte. Es tracta de fer la instal·lació de cada una de les eines per tal de conèixer el seu funcionament i la funcionalitat que ofereixen.

5.3.3 T3.2 Estudi i instal·lació servidor Linux-Cherokee/Apache-PHP

Instal·lació i configuració de l'entorn per realitzar la programació del projecte.

5.3.4 T3.2 Estudi i instal·lació de la base de dades noSQL MongoDB

Instal·lació i configuració de l'entorn on tindrem la persistència de dades del projecte.

5.3.5 T4 Anàlisi de requeriments i disseny

Recopilació de la informació trobada en els punts anteriors. Aquest document ha d'incorporar a més els requisits i el disseny del prototip a implementar.

5.3.6 Lliurable

PAC-1. Planificació del projecte, descripció, objectius, abast del projecte i anàlisi de requeriments.

5.4 Fase 2

5.4.1 T5 Estudi de patrons de programació que puguin ser aplicables al projecte

Recerca i estudi de patrons de programació que puguin ser útils en els mòduls que formaran part del projecte

5.4.2 T6 Estudi de sistemes d'events i integració amb dispositius mòbils des de plataformes web pel treball en grup

Recerca i estudi d'altres sistemes que siguin semblants al del projecte i recerca i estudi de la integració amb plataformes mòbils, per tal de poder validar el sistema.

5.4.3 T7 Proves de concepte

Recerca de frameworks que puguin accelerar el desenvolupament del projecte. També es faran petits prototipus per tal d'avaluar la millor manera d'implementar els patrons seleccionats.

5.4.4 T8 Implementació pràctica del prototip

En les següents subtasques tindrem la implementació del codi del projecte.

5.4.5 T8.1 Implementació gestió d'usuaris i grups

Una de les parts principals del projecte es tenir definits grups d'usuaris que volen estar informats de tot el que passa. En aquesta tasca implementarem la gestió d'aquest usuaris i grups.

5.4.6 T8.2 Implementació gestió events

L'altra gran part del projecte es tenir definits events als quals els usuaris dels grups es podran subscriure. Aquesta tasca implementarà la gestió d'aquests events.

5.4.7 T7.3 Implementació gestió subscripció events

Un cop implementat la gestió de usuaris i events tenim que implementar la subscripció de usuaris a events. D'aquest feina s'encarregarà aquesta subtasca.

5.4.8 T8.4 Implementació sistema d'enviament events

Per últim els usuaris han de rebre els events cada cop que es produeixen. De l'enviament dels events ens ocuparem en aquesta subtasca.

5.4.9 T9 Inici proves de funcionament

En aquesta tasca, i subtasques, farem els testos funcionals del sistema, per tal de verificar que la implementació es congruent amb les especificacions.

5.4.10 T9.1 Proves gestió d'usuaris i grups

Proves de la gestió d'usuaris i grups.

5.4.11 T9.2 Proves gestió events

Proves de la gestió d'events.

5.4.12 T9.3 Proves gestió subscripció events

Proves de la gestió subscripció events.

5.4.13 T9.4 Proves motor events

Proves del motor d'events.

5.4.14 T10 Desenvolupament i prototipatge

Recopilació de la documentació generada en els punts anteriors. En aquest documentació també hi haurà la primera versió del codi del prototip i el guió de proves que s'hauran realitzat.

5.4.15 Lliurables

PAC-2. Desenvolupament i Prototipatge.

5.5 Fase 3

5.5.1 T11 Documentació del projecte

En aquesta tasca es generarà tota la documentació necessària per la entrega final del projecte, així com el necessari per la defensa virtual del mateix.

Donat que la documentació es va generant durant tot el projecte aquesta tasca farà la recopilació de la mateixa, de manera que estigui revisada i es pugui tenir tota en un únic document de forma cohesionada.

5.5.2 T12 Publicació del prototip

La intenció es que el sistema pugui ser utilitzable, de manera que un cop acabat es buscarà una manera de publicar el prototip, tot i que no es considerarà obligatori.

5.5.3 T13 Lliurament final

Lliurament de tota la documentació generada

5.5.4 Lliurables

- Memòria
- Presentació per la defensa virtual
- Software Desenvolupat: codi complet, dades, scripts de creació, etc.
- Manuals d'Administrador i Usuari.
- Desplegament: Publicació web en servidor públic. (opcional)

6. Equip de treball i implicats

Tenint en compte la naturalesa del projecte, ja que es tracta d'un PFC, els diferents rols de l'equip de treball i dels implicats en el projecte hauran d'esser assumits per l'alumne i el consultor, i en alguns casos seran ficticis.

	Descripció	Rol assumit per
Comitè de direcció	Pren les decisions últimes. Aprova el projecte.	Consultor
Comitè executiu	Responsable de les diferents àrees (TIC, màrqueting, ventes...)	Per l'alumne per l'àrea TIC (la resta queda exclosa del PFC)
Director del projecte	Estableix pla de treball, assignació de recursos, pla de qualitat ...)	Per l'alumne
Desenvolupadors	Equip de desenvolupadors tècnic.	Per l'alumne
Client	Estableix funcionalitats.	Per l'alumne
Departament de màrqueting	Accions de màrqueting per promocionar el sistema	Fictici
Proveïdors de serveis	Proporciona serveis per enviar missatges.	Fictici
Patrocinador		Fictici

7. Anàlisi de riscos

Donat que tot projecte, tot i que existeix un esforç de planificació per tal d'evitar retards, pot tenir una sèrie de imprevistos val més identificar-los per tal de poder actuar en cas de que es materialitzin.

Risc 1	Malaltia
Descripció	En cas de malaltia es podria retardar el desenvolupament del projecte
Impacte	Qualitat del treball realitzat, donada la falta de temps. No poder complir plaços d'entrega
Probabilitat	Mitja
Acció	Augmentar el numero d'hores per recuperar el temps perdut
Risc 2	Viatge per feina
Descripció	La feina actual m'obliga a viatger com a mínim un dia cada dues setmanes.
Impacte	Qualitat del treball realitzat, donada la falta de temps. No poder complir plaços d'entrega
Probabilitat	Mitja
Acció	Recuperar el temps durant el temps del viatge, i planificar per tal de reduir l'impacte.

Risc 3	Problemes amb l'equip informàtic
Descripció	Problemes de hardware, de malfuncionament del equip....
Impacte	Qualitat del treball realitzat, donada la falta de temps. No poder complir plaços d'entrega
Probabilitat	Baixa.
Acció	Fer còpies de seguretat, tenir un pla de contingències.
Risc 4	Problemes en el test als mòbils
Descripció	No trobar una plataforma que permeti l'enviament de missatges gratuïts.
Impacte	Problemes per demostrar la funcionalitat del projecte en events que hagin de ser enviats per SMS.
Probabilitat	Alta
Acció	Trobar alternatives que demostrin que el projecte funcioni per exemple via Twitter.
Risc 5	Conciliació familiar
Descripció	Donat que aquest projecte es realitza amb paral·lel amb la vida familiar poden sorgir incompatibilitats d'horaris, tot i la planificació. La probabilitat més alta es que es treballi la majoria dels dies fins a ben entrada la nit.
Impacte	Falta de hores, son.
Probabilitat	Alta
Acció	Prendre cafè per aguantar hores nocturnes.

8. Pla de qualitat

8.1 Política

El pla de qualitat té com objectiu assegurar de manera contínua la qualitat dels productes i serveis que es volen desenvolupar. Tot i que en el present projecte no s'espera que puguin haver molts canvis en el pla de treball a causa dels resultats del desenvolupament o per altres companyies que hi participen, establir el pla permetrà controlar de forma més eficient les desviacions temporals que s'hi puguin presentar.

Els objectius del pla serà:

- Marcar les pautes que garanteixin les expectatives de qualitat del producte resultant del projecte
- Descobrir desviacions del pla quan s'originin
- Facilitar la gestió de manera que es puguin prendre accions correctores, si es necessari, tan aviat com sigui possible.

8.2 Responsables

Donat que el projecte es un PFC l'únic responsable serà l'alumne. Això farà que la mateixa persona acabi tenint diversos rols ja que serà el responsable de qualitat i també el desenvolupador, per exemple. Com a conseqüència això farà més difícil el seguiment del pla de qualitat.

8.3 Processos de qualitat

8.3.1 Processos de qualitat de la totalitat del projecte

Es consideren les tasques següents:

- Coordinar les activitats d'assegurament de la qualitat i tasques tècniques.
- Comprovar al principi de cada fita la viabilitat de les activats planificades
- Verificar en les diferents fites si els resultats són coherents, complets, consistents i correctes, i també si es compleixen els requisits i se segueixen els estàndards establerts.
- Informar el Comitè Executiu del projecte de les activitats d'assegurament de la qualitat, i també dels riscos que tenen.
- Organitzar auditories internes per a cada lliurable del projecte.
- Ajudar a establir i mantenir l'informe d'anomalies.
- Ajudar a trobar solucions als problemes i supervisar que se segueixen en la seva resolució.
- Comprovar que es segueixen els procediments de control de qualitat.

8.3.2 Processos específics de cada activitat

S'han identificat les tasques específiques següents de cada activitat:

- Ajudar a assegurar que cada requisit s'implementa de manera correcta.
- Assegurar un acoblament mínim entre components, cosa que s'aconsegueix des del disseny arquitectònic, de manera que permeti un disseny detallat independent.
- Revisar la metodologia d'avaluació i també els procediments de prova per a assegurar que cada fase de prova és preparada correctament.
- Col·laborar en la preparació d'activitats de proves del sistema per a assegurar que s'adeqüen als plans i procediments de prova.

8.4 Estàndards que s'han d'aplicar

8.4.1 Documentació

Es seguirà els estàndards de la UOC, basant-se en els exemples que pugui aportar el consultor i els apunts d'altres assignatures.

La documentació del usuari haurà de ser comprensible per a persones amb poca experiència tecnològica.

8.4.2 Programari

S'haurà de seguir un estàndard a nivell de documentació de codi i de proves unitàries que permeti la comprensió del projecte.

8.5 Criteris d'acceptació

Característica	Criteri	Mètode
El programari ha de satisfer els requisits d'alt nivell definits.	Matriu de traçabilitat de requisit-funcionalitat.	Comprovar els 5 requisits principals, i una part aleatòria del 50% restant.
El programari ha de ser robust	Pla de proves funcionals sense errors. Superar les proves d'estrès.	Comprovar que el prototipus lliurat està sense errors. Proves aleatòries sobre 5 casos d'ús. Passar escenaris d'estrès amb una eina adient.
El programari ha de ser escalable i mantenible	Seguiment dels estàndards de programació i documentació.	Comprovar amb una eina la documentació del programari (comentaris) i la documentació tècnica.

8.6 Criteris de qualitat global

- Temps: el projecte ha de seguir la planificació del pla de projecte. Es toleren retards intermedis, sempre que no afectin la planificació global.
- Abast: en el cas de necessitar revisar l'abast del projecte, és necessari l'acord de totes les parts del projecte.

9. Anàlisi de requeriments

Partint del flux estàndard de comunicació on un emissor envia un missatge per un canal a un receptor, s'ha intentat dissenyar una solució basada en aquest flux on cada part sigui fàcilment configurable i ampliable, de manera que sigui relativament senzill afegir-ne de noves, o adaptar les existents, sense haver de modificar la resta.

Així doncs partint del flux estàndard de comunicació (on les fletxes representen el canal) tindríem:



En el nostre sistema el disseny seria:



on la fletxa entre missatge i usuari seria el mitjà de transport : SMS, MMS, Tweet

Per tal de poder manegar aquest flux tindrem els següents mòduls:

9.1 Mòdul de gestió d'usuaris/grups

Descripció

El client ens parla d'un sistema en treball en grup, es obvi que el grup estarà format per usuaris que seran els que rebran finalment el missatge en el seu dispositiu mòbil.

Tal com hem vist en el flux el usuaris seran els receptors del missatge, i tot i que se'ns demana que aquest arribi a un dispositiu mòbil també hem de pensar en un sistema que sigui flexible i pugui arribar en diversos mitjans de publicació.

Funcionalitats bàsiques

- Afegir grups i usuaris del grup
- Consultar grups i usuaris
- Modificar grups i usuaris del grup
- Eliminar grups i usuaris del grup

9.2 Mòdul de gestió d'events

Descripció

Els events els podem definir com el canvi d'estat d'un objecte. Per tant en la definició podem veure que tenim dos conceptes clau: canvi d'estat i objecte.

Un objecte serà l'ens que el sistema observa de forma periòdica esperant un canvi d'estat. Els objectes seran referenciats per una URI (identificador uniforme de recursos), de manera que podrem accedir a objectes en local o remot.

Un canvi d'estat es produirà quan una dels atributs que defineixen un objecte canvia. El sistema gestionarà aquests atributs mitjançant les condicions que s'avaluen per detectar els canvis dels objectes.

Aquestes atributs podran ser: llegir, escriure, crear, esborrar i modificar. El sistema permetrà definir-los de forma lliure, mitjançant les condicions, de manera que se'n puguin afegir-ne de nous de forma fàcil

Cada objecte tindrà associat un observador que informará del canvi d'estat. Aquest observador seran programes que s'executaran per detectar els canvis.

Donat el flux de comunicació estàndard, i ja que l'event es el que inicia la comunicació, podríem dir que l'event serà l'emissor.

Per tal de poder agrupar els objectes que conceptualment pertanyen al mateix "lloc" tenim el context. Així doncs tots els objecte que pertanyen a una aula (per exemple l'Aula PFC), per exemple fòrums de discussió, taulell d'anuncis, tweeter, documentació..., tindran com a nexa la pertinença al context "Aula PFC".

Per altra banda, i per tal de que l'usuari pugui realitzar certes accions quan rep els missatges, tenim associat als objectes les retroaccions. Això permet definir com accedir directament a la resposta en un fòrum, un tweet...

Funcionalitats bàsiques

Submòdul objectes

- Afegir, consultar, modificar i eliminar objectes

Submòdul observador

- Afegir, consultar, modificar i eliminar observador

Submòdul events

- Afegir, consultar, modificar i eliminar events, on els events aniran definits per l'associació d'un objecte a un observador.

Submòdul context

- Afegir, consultar, modificar i eliminar context.

Submòdul retroaccions

- Afegir, consultar, modificar i eliminar retroaccions.

9.3 Mòdul de gestió de subscripció d'events

Descripció

El sistema ha de mantenir informat els membres del grup de tot allò que ocórrer en l'espai del grup. Fins ara hem vist els grups i els events, però encara ens falta poder saber de que volen estar informats els membres del grup, es a dir a quins events es poden subscriure. Aquesta subscripció es podrà fer a nivell de grup, o bé a nivell d'usuari. Tot i que el client no ho ha sol·licitat la subscripció a nivell d'usuari es considera una funcionalitat necessària, ja que un usuari podria tenir interessos diferents als del grup. En cas de incongruències sempre es tindrà en compte les preferències del usuari per damunt de les del grup.

Per tal que les notificacions arribin als usuaris caldrà definir el canal. Aquests canals vindran definits dins el sistema per l'acció associada al event. Una cua de missatges serà el lloc on s'emmagatzemaran els missatges a enviar.

Per tant en la subscripció tindrem els següents actors: grups i usuaris, events i canals. Podem dir que en aquest mòdul es on instanciem el flux de comunicació.

Aquesta subscripció podrà ser en temps real, ho anomenarem PUSH, o bé la informació la rebrà l'usuari quan aquest la demani, cosa que anomenarem en direm PULL.

Funcionalitats bàsiques

Submòdul subscripció grups

- Subscripció del grup a events
- Consultar subscripcions del grups
- Desubscripció del grup a events

Submòdul subscripció usuaris

- Subscripció del usuari a events
- Consultar subscripcions del usuari
- Desubscripció del usuari a events

9.4 Mòdul sistema d'enviaments d'events.

Descripció

Cada cop que un observador detecta un canvi d'un atribut en un objecte es produeix un event que cal notificar al grup o als usuaris. L'observador no tindrà capacitat per enviar missatges, l'únic que podrà fer es encuar el missatge en una cua del sistema d'enviament d'events que serà el que s'encarregarà d'enviar el missatge. En el cas que el missatge falli, per exemple perquè hi ha un error d'entrega, el sistema seguirà una política de reintents per tal de fer tot lo possible per fer arribar el missatge.

Funcionalitats bàsiques

- Enviar missatge
- Reencuar missatge enviat
- Afegir, consultar, modificar política de reintents

10. Consideracions sobre els requeriments

El cost del projecte es el preu de la matrícula de l'assignatura i el cost d'oportunitat de poder fer altres activitats.

Queda fora de l'abast del projecte el manteniment i les possibles millores que puguin realitzar-se del prototipus desenvolupat

També queda fora del abast d'aquest projecte l'autenticació i autorització al sistema, es a dir el sistema no demanarà en cap moment credencials per fer-lo servir, ni tampoc proporcionarà mecanismes per permetre o denegar l'accés a les diverses parts dels mòduls.

La consideració anterior implica que la subscripció d'events a un grup podrà ser feta per qualsevol usuari.

Per últim tot el control de despeses associada a l'ús de canals que siguin tarifats queda també exclòs.

11. Plataforma i eines de desenvolupament

11.1 Introducció

Donat que el sistema té com a mínim requisit disposar d'un servidor web que permeti executar PHP i una base de dades mongoDB, i per altra banda com a exercici per demostrar aspectes multiplataforma del desenvolupament, aquest s'ha fet finalment sobre Windows 7 amb un paquet WAMP anomenat easyPHP i la versió de mongoDB per Windows.

Cal esmentar que tot i la tria de Windows 7 com a sistema operatiu, el sistema també s'instal·larà sota una distribució Linux per tal de demostrar, tal com dèiem, el vessant multiplataforma.

Donat que del paquet easyPHP només calia el servidor web, que ha sigut finalment doncs un Apache, el servidor MySQL s'ha aturat. Per altra banda ha calgut buscar i instal·lar el mòdul de mongo per PHP per tal de tenir totes les funcionalitats necessàries.

Com a front end de mongoDB sota Windows s'ha fet servir una eina anomenada mongoVUE , tot i que també s'ha fet servir la shell que mongo incorpora.

Com a editor del codi font s'ha fet servir Eclipse amb les eines de desenvolupament en PHP.

S'ha triat com a sistema de control de versions Git. Aquest control de versions que va ser desenvolupat per Linus Torvalds i que es fa servir pel control de versions del kernel de Linux, s'ha integrat a Eclipse per mitjà de l'afegit EGit.

Per fer el disseny dels diagrames s'ha fet servir MagicDraw UML Personal Edition amb la llicència de la UOC.

Per últim dir que donat que s'ha treballat simultàniament en dos ordinadors diferents, un iMac amb MacOSX i un portàtil HP amb Windows 7, s'ha utilitzat Dropbox per tal de fer la sincronització entre els dos equips. Això ha garantit una redundància en la còpia de seguretat, ja que Dropbox emmagatzema versions dels fitxers que s'hi desen i per altra banda des del iMac s'ha fet servir l'eina per defecte de còpia de seguretat: TimeMachine.

11.2 Instal·lació i documentació del entorn

11.2.1 easyPHP

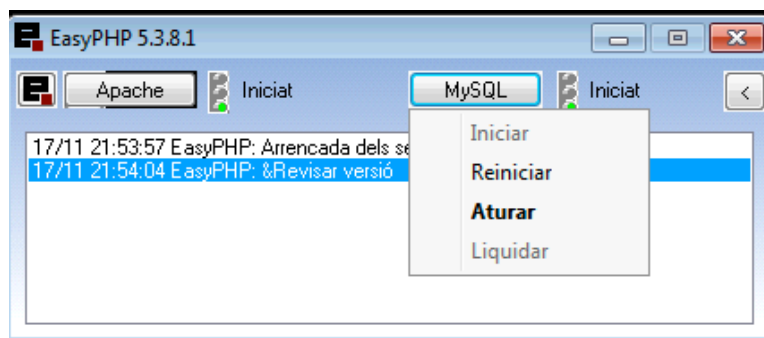
El paquet seleccionat per tenir servidor web amb els mòduls de PHP ha sigut easyPHP. Per instal·lar-lo cal anar a <http://www.easyphp.org/> i baixar-se la última versió de la secció download. Pel desenvolupament del projecte hem fet servir la versió 5.3.8.1. Aquesta versió proporciona :

- PHP 5.3.8 VC9
- Apache 2.2.21 VC9
- MySQL 5.5.16
- PhpMyAdmin 3.4.5
- Xdebug 2.1.2

Un cop instal·lat ja el podem executar i ens apareixerà una icona en l'àrea de notifikacions.

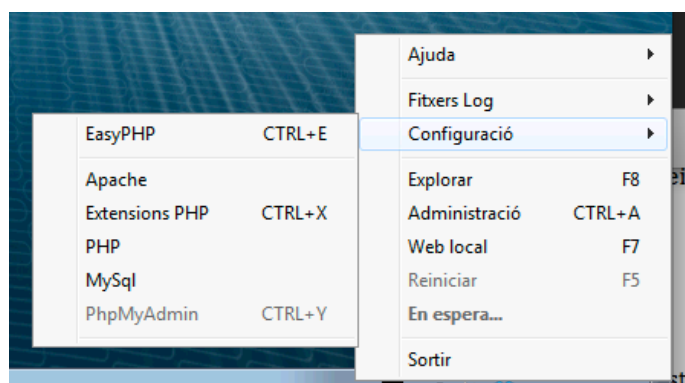


Clicant a sobre ens apareix una finestra des de la qual podem aturar el MySQL

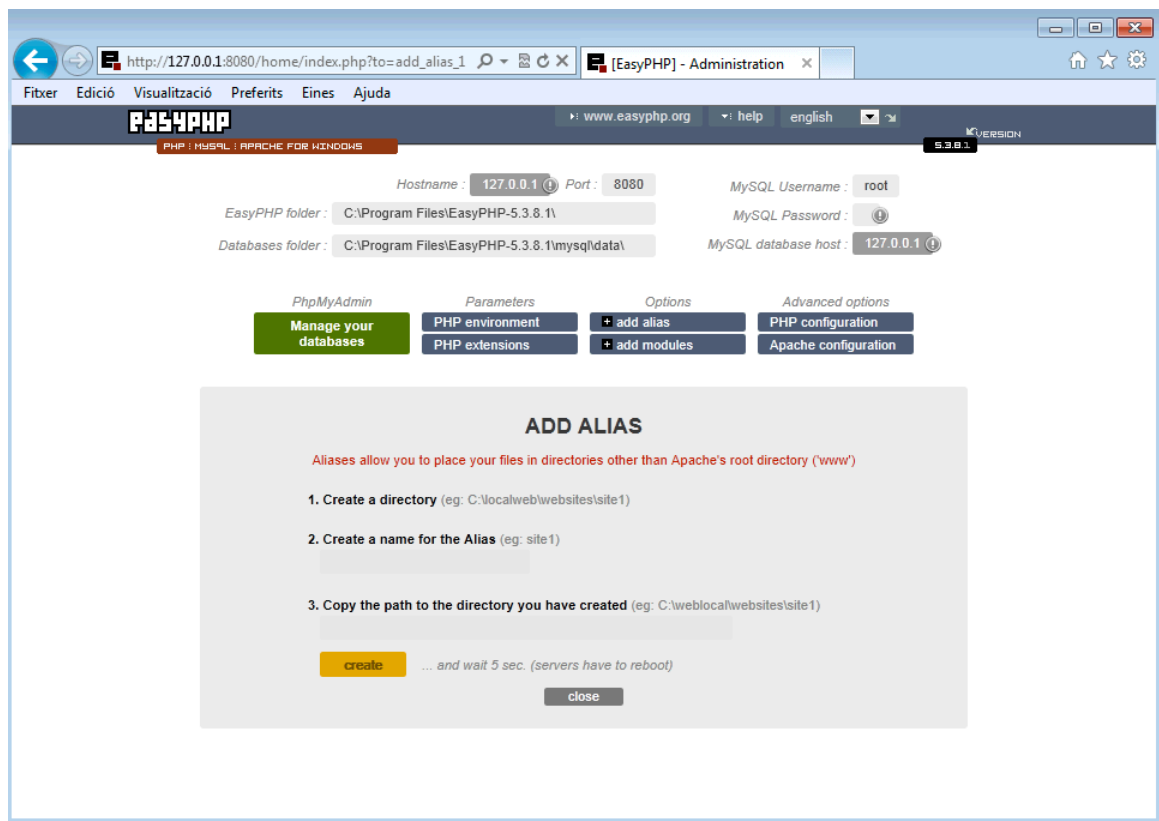


Per defecte easyPHP no porta el mòdul de mongoDB per instal·lar-lo cal anar a <http://www.php.net/manual/en/mongo.installation.php#mongo.installation.windows> i baixar la DLL corresponent a la versió de PHP 5.3. Per instal·lar-la només ens cal deixar la DLL php_mongo.dll en el directori `C:\Program Files\EasyPHP-5.3.8.1\php\ext`. Reiniciat l'apache ja tindrem a la nostra disponibilitat totes les funcions de Mongo.

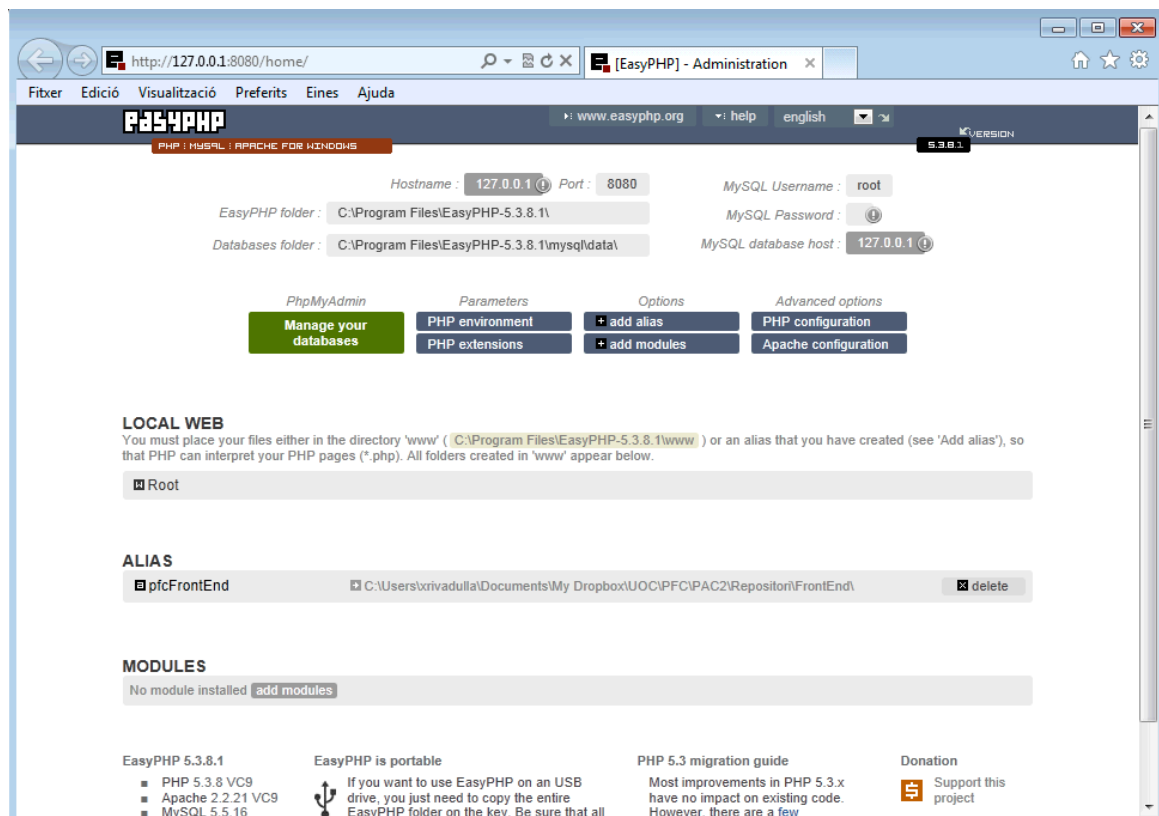
Per últim cal dir-li on tenim les pàgines que cal que serveixi el servidor. Per configurar-lo fem clic sobre la icona de l'àrea de notifikacions i seleccionem *Configuració->EasyPHP*



Aquesta acció ens obre un navegador web des d'on podem afegir una *Alias*.



Amb el que configurarem el nom de l'alias i el directori on hi hauran els fitxers.



Des d'aquest moment ja podem accedirà http://127.0.0.1:8080/nom_alias per accedir a les nostres pàgines.

En el cas del PFC aquest són els alies creats:

pfFrontEnd

C:\Users\xrivadulla\Documents\My Dropbox\UOC\PFC\PAC2\Repositori\FrontEnd\

Observer

C:\Users\xrivadulla\Documents\My Dropbox\UOC\PFC\PAC2\Repositori\Observer\

Dispatcher

C:\Users\xrivadulla\Documents\My Dropbox\UOC\PFC\PAC2\Repositori\Dispatcher\

On :

- A pfFrontEnd tenim la interfície web
- A Observer el sistema per detectar events
- A Dispatcher el sistema per enviar missatges

Per altra banda també hem configurat el php editant el fitxer php.ini i hem afegit la línia:

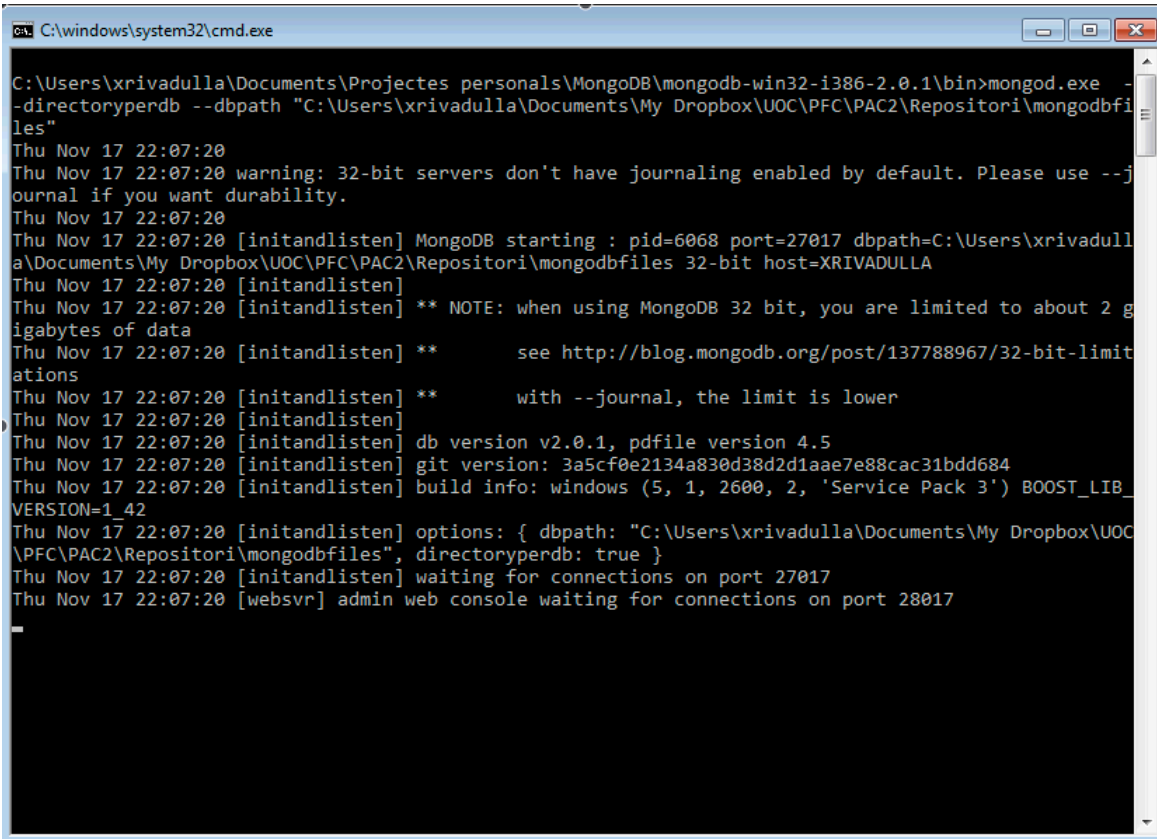
```
include_path = ".;C:\php\pear;C:\Users\xrivadulla\Documents\My  
Dropbox\UOC\PFC\PAC2\Repositori\libs;C:\Users\xrivadulla\Documents\My  
Dropbox\UOC\PFC\PAC2\Repositori\Observer\Observers;C:\Users\xrivadulla\Documents\My  
Dropbox\UOC\PFC\PAC2\Repositori\Dispatcher\class"
```

D'aquesta manera no tindrem problemes a l'hora d'incloure fitxers dins d'altres. Això es especialment útil quan volem fer servir un fitxer que conté una classe dins d'una altra, per exemple.

11.2.2 mongoDB

La persistència de dades les deixem en mans de mongoDB. Per tal de instal·lar la base de dades cal anar a <http://www.mongodb.org/> i a la secció de download triar la versió que correspongui. En el nostre cas hem fet servir la versió 2.0.1 per Windows 32-bit.

Instal·lar mongoDB es simplement descomprimir el fitxer zip i executar la comanda mongod.exe.



```

C:\windows\system32\cmd.exe
C:\Users\xrivadulla\Documents\Projectes personals\MongoDB\mongodb-win32-1386-2.0.1\bin>mongod.exe --
--directoryperdb --dbpath "C:\Users\xrivadulla\Documents\My Dropbox\UOC\PFC\PAC2\Repositori\mongodbfiles"
Thu Nov 17 22:07:20
Thu Nov 17 22:07:20 warning: 32-bit servers don't have journaling enabled by default. Please use --j
ournal if you want durability.
Thu Nov 17 22:07:20
Thu Nov 17 22:07:20 [initandlisten] MongoDB starting : pid=6068 port=27017 dbpath=C:\Users\xrivadulla
\Documents\My Dropbox\UOC\PFC\PAC2\Repositori\mongodbfiles 32-bit host=XRIVADULLA
Thu Nov 17 22:07:20 [initandlisten]
Thu Nov 17 22:07:20 [initandlisten] ** NOTE: when using MongoDB 32 bit, you are limited to about 2 g
igabytes of data
Thu Nov 17 22:07:20 [initandlisten] **      see http://blog.mongodb.org/post/137788967/32-bit-limit
ations
Thu Nov 17 22:07:20 [initandlisten] **      with --journal, the limit is lower
Thu Nov 17 22:07:20 [initandlisten]
Thu Nov 17 22:07:20 [initandlisten] db version v2.0.1, pdfile version 4.5
Thu Nov 17 22:07:20 [initandlisten] git version: 3a5cf0e2134a830d38d2d1aae7e88cac31bdd684
Thu Nov 17 22:07:20 [initandlisten] build info: windows (5, 1, 2600, 2, 'Service Pack 3') BOOST_LIB_
VERSION=1_42
Thu Nov 17 22:07:20 [initandlisten] options: { dbpath: "C:\Users\xrivadulla\Documents\My Dropbox\UOC
\PFC\PAC2\Repositori\mongodbfiles", directoryperdb: true }
Thu Nov 17 22:07:20 [initandlisten] waiting for connections on port 27017
Thu Nov 17 22:07:20 [websvr] admin web console waiting for connections on port 28017
-

```

De totes maneres per facilitar l'ús hem creat un fitxer .bat on passem alguns paràmetres extres. En concret dins el fitxer .bat tenim :

```

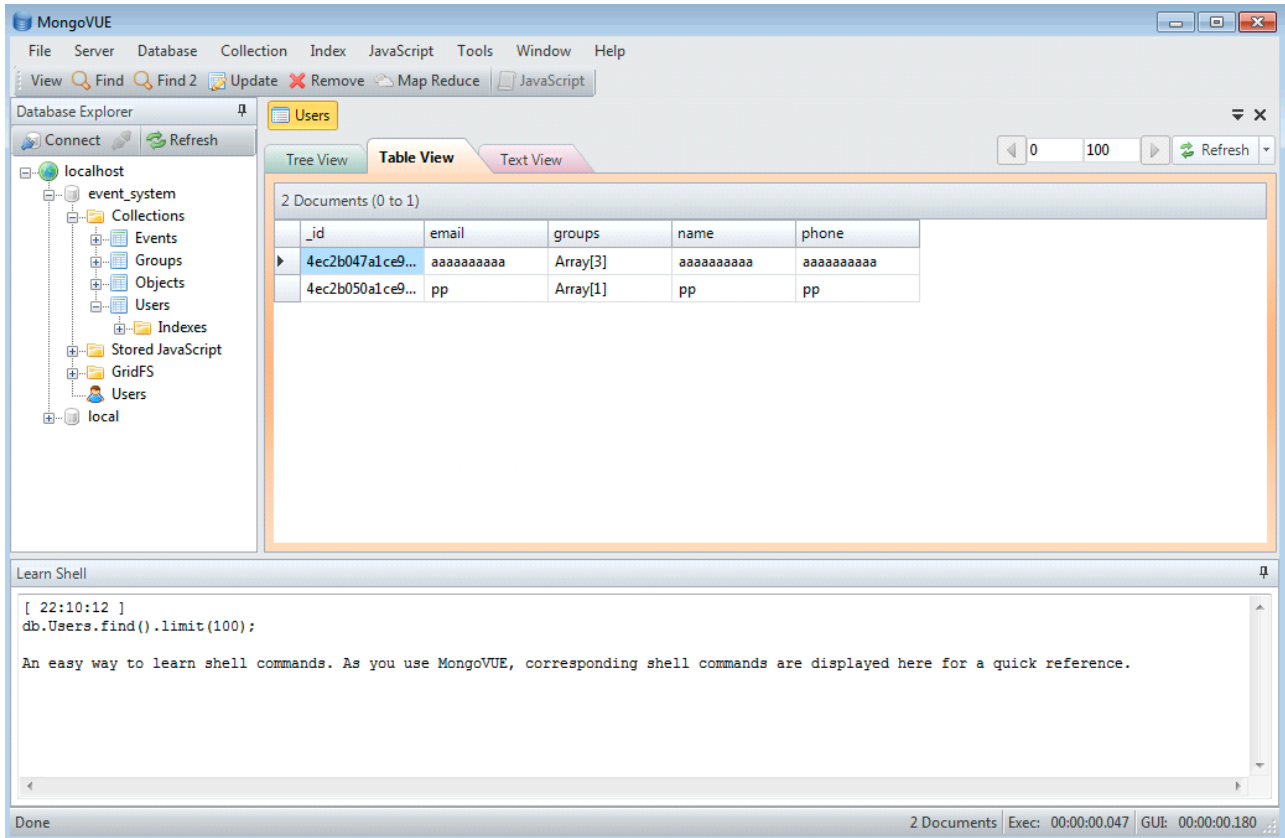
mongod.exe --directoryperdb --dbpath "C:\Users\xrivadulla\Documents\My
Dropbox\UOC\PFC\PAC2\Repositori\mongodbfiles"

```

El paràmetre --directoryperdb el que fa es generar un directori per cada base de dades, en comptes de tenir-les a totes en un únic directori.

El paràmetre --dbpath indica on ha de generar aquest fitxers. Hem triat un directori de Dropbox per tal de poder fer servir les dades des d'altres entorns de desenvolupament i poder fer còpies de seguretat.

Com a front end de mongoDB hem fet servir MongoVUE. Per instal·lar-lo cal anar a <http://www.mongovue.com/> i baixar-lo de la secció download. Per instal·lar-lo només cal executar el instal·lable.



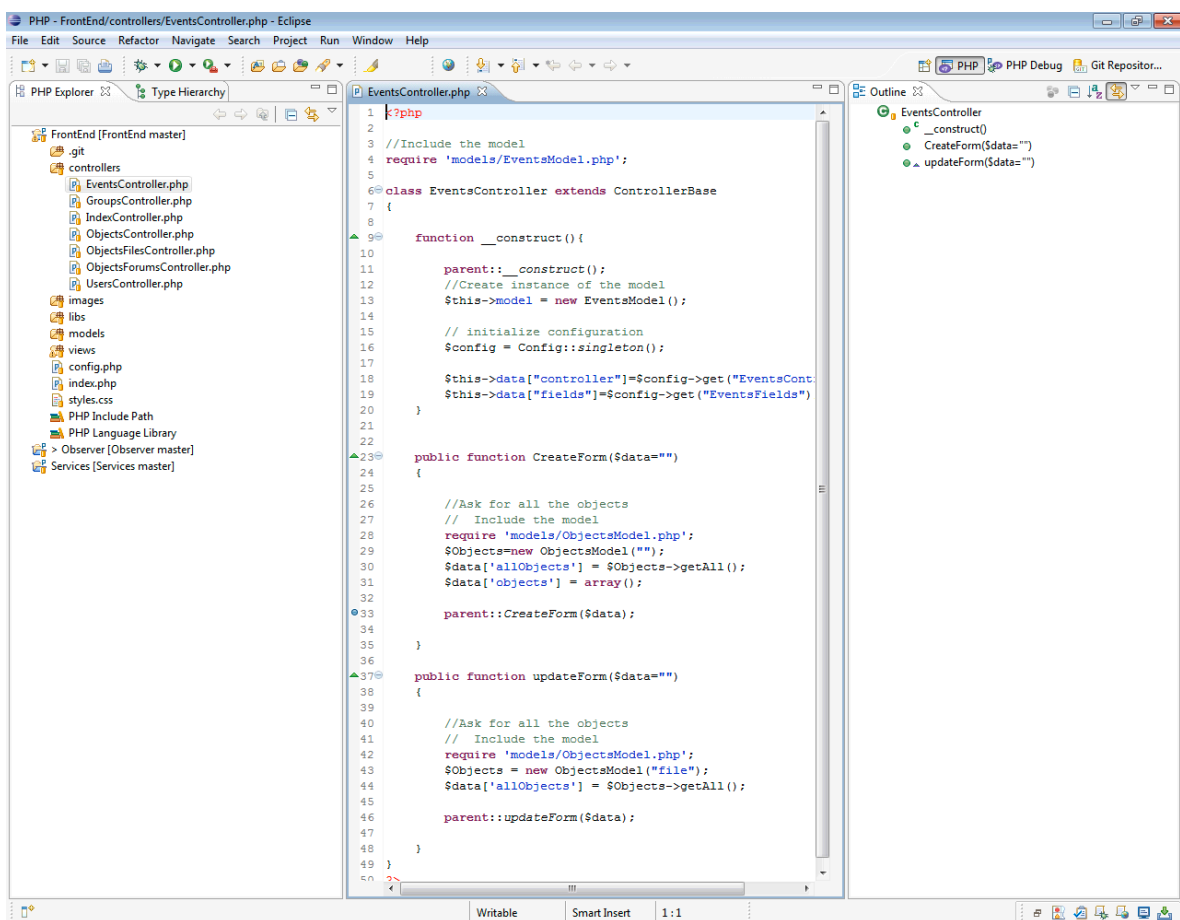
Aquesta aplicació només permet obrir tres vistes, ja que es l'edició gratuïta, però es considera suficient per tal de fer petites tasques de gestió de les dades.

11.2.3 Eclipse

Com editor de codi font em seleccionat Eclipse amb l'afegit per programar amb PHP que s'anomena PDT (PHP Development Tools).

Per tal de instal·lar-lo cal anar a <http://eclipse.org/pdt/downloads/> i triar la darrera versió. Com qualsevol Eclipse instal·lar-lo es simplement descomprimir el fitxer i executar el executable anomenat eclipse.

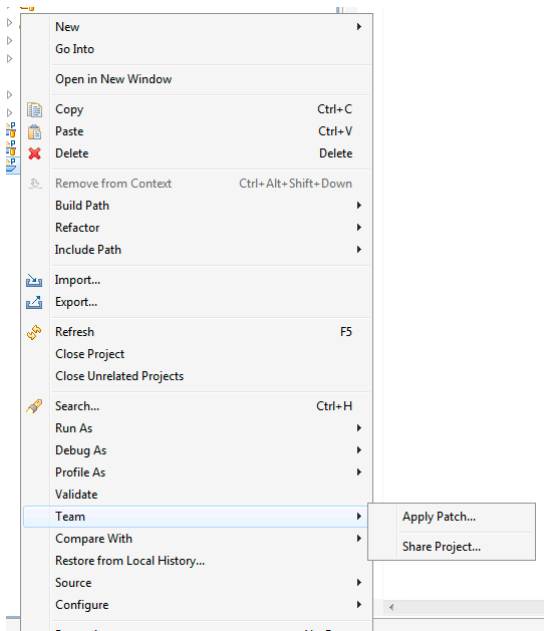
Tot el desenvolupament ha sigut fet des d'aquesta aplicació, inclosos els petits detalls de disseny que s'han afegit, com és la creació del CSS.



11.2.4 Egit

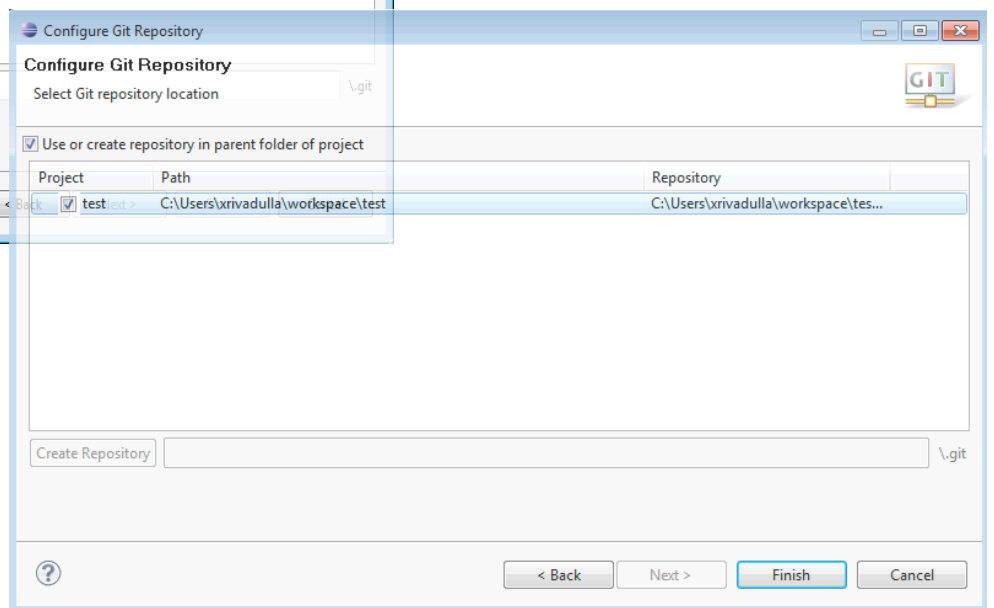
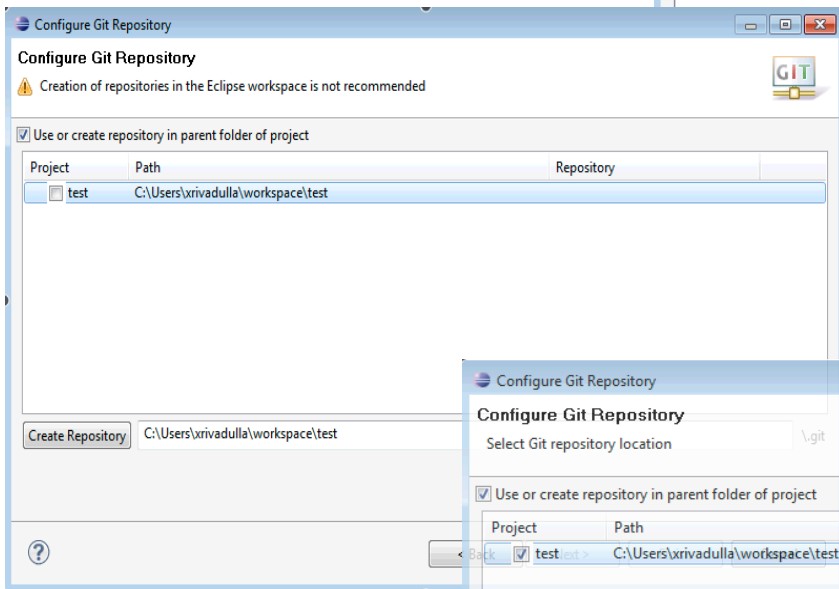
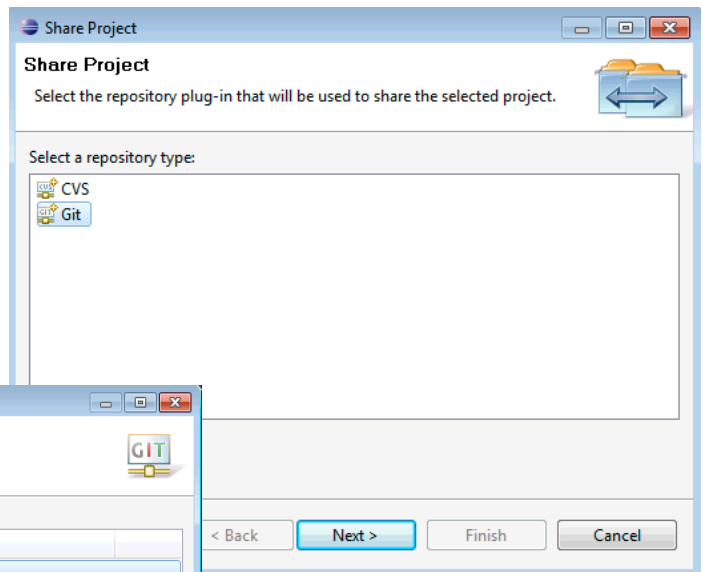
Per tal de poder desenvolupar amb control de versions hem instal·lat l'afegit EGit per Eclipse que es pot trobar a <http://eclipse.org/egit/>. Per instal·lar-lo només cal anar, des del dins de Eclipse, a l'opció *Install new software* i afegir la url d'instal·lació.

Un cop instal·lat el Egit, i després de crear un projecte de PHP, que de nou te el seu directori dins un directori de Dropbox, per posar-lo sota control de versió només cal :

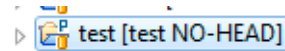


Seleccionar el projecte i fent clic amb el botó de la dreta anar a Team->ShareProject

Seleccionem Git com a tipus de repositori.



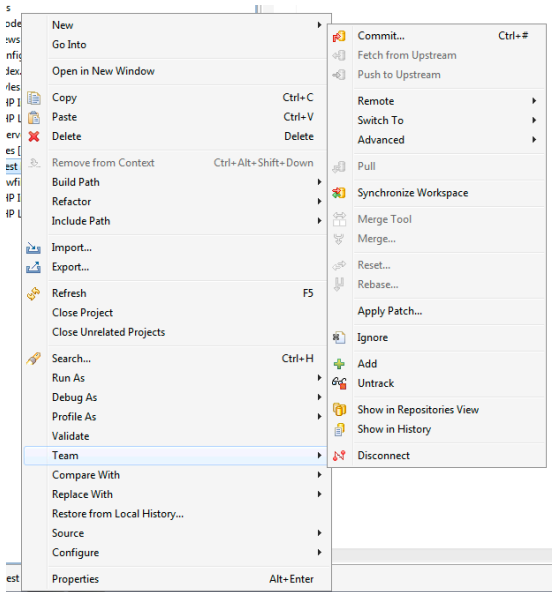
Ara ja podem crear el repositori seleccionat el projecte i fent clic a "Create repository"



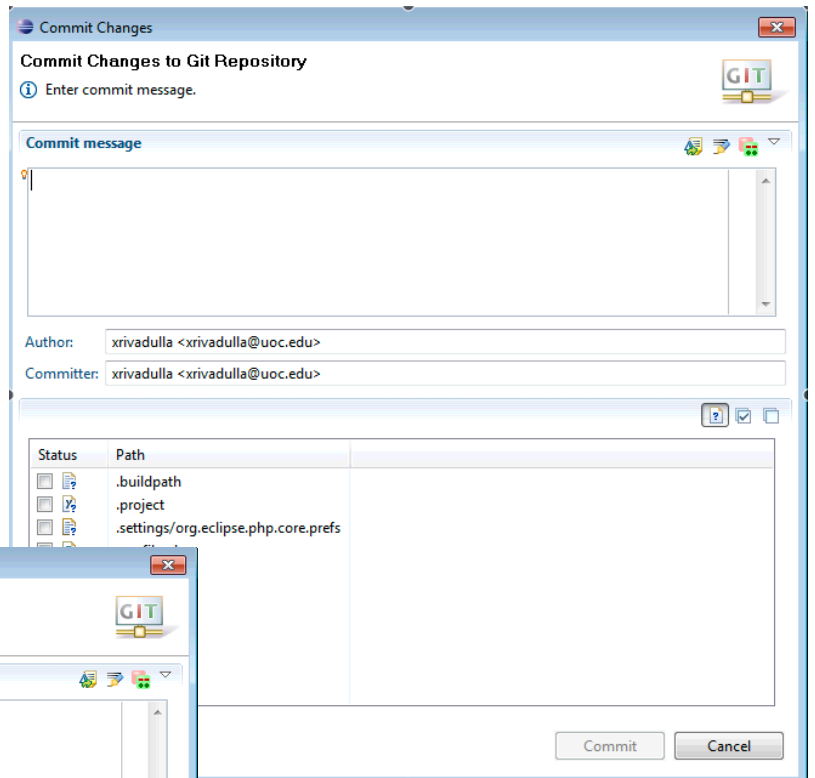
Si ens fixem en el projecte el veurem que esta en *NO-HEAD*

Això es així per que el repositori encara esta buit, cal fer un commit per tal de començar a fer-ne ús. En altres sistema de control de versions el commit es coneix com a check-in i en català, segons el termcat (http://www.termcat.cat/ca/Diccionaris/En_Linia/123/Cerca/), en podem dir *pujar*. La definició que ens proporciona el termcat es :

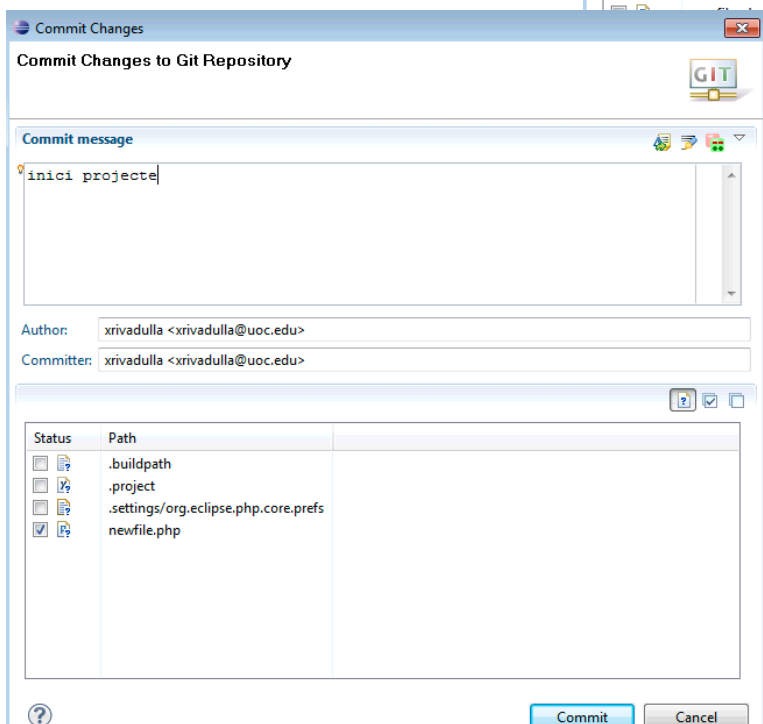
Desar una nova versió al repositori d'un sistema de control de versions amb l'objectiu de permetre a tots els usuaris del sistema l'accés a aquesta nova versió.



Per fer el commit caldrà fer de nou clic amb el botó de la dreta i si seleccionem la opció Team. Veurem que el desplegable es diferent. Seleccionem la primer opció anomenada Commit.

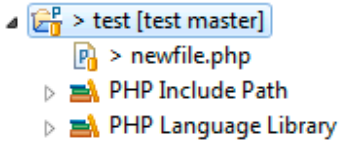
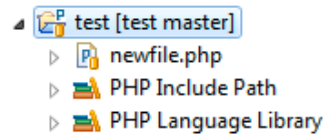


En la finestra que se'ns obre podem seleccionar els fitxer als que volem fer el commit.



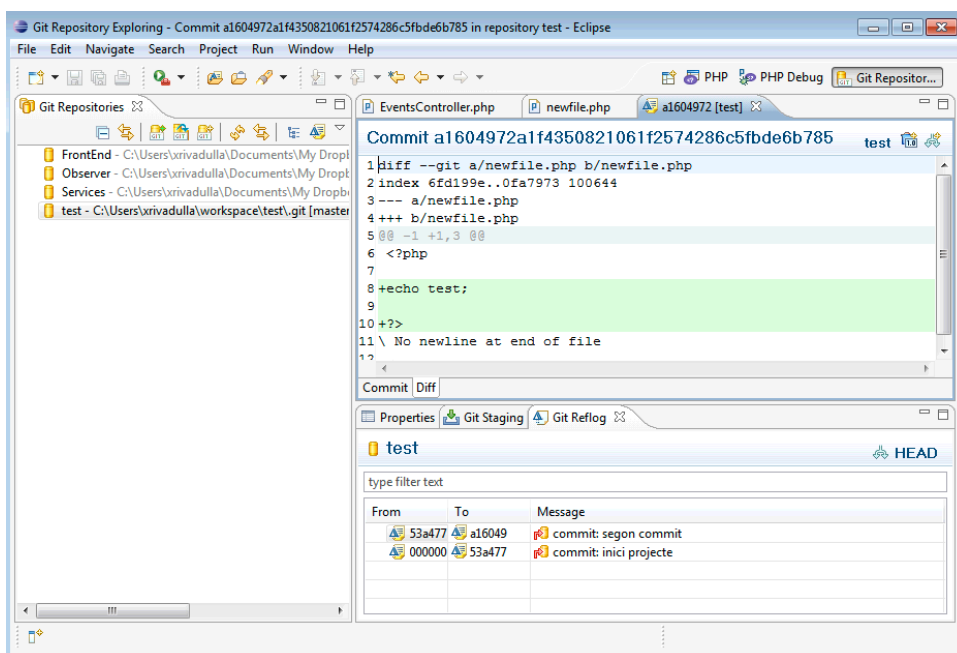
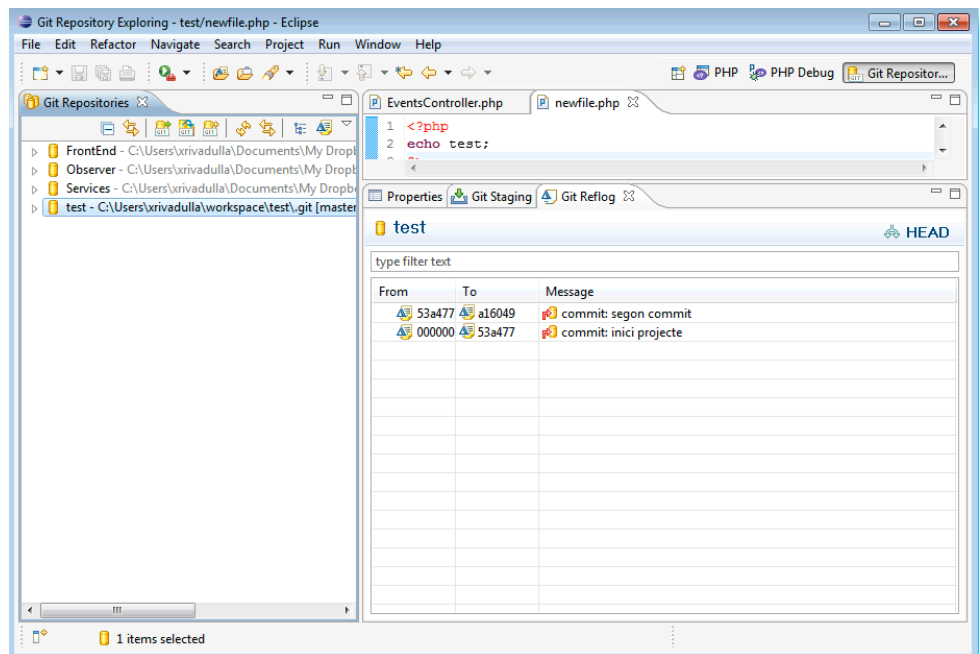
Escrivim un missatge que ens indiqui el motiu i ja podem fer clic en el botó commit.

Si ens fixem en el projecte veurem que ara ja no surt *NO-HEAD*, ara ens apareix la paraula *master*



Cada cop que modifiquen un fitxer apareixerà un símbol > que indica que el fitxer esta modificat respecte a la versió de l'últim commit.

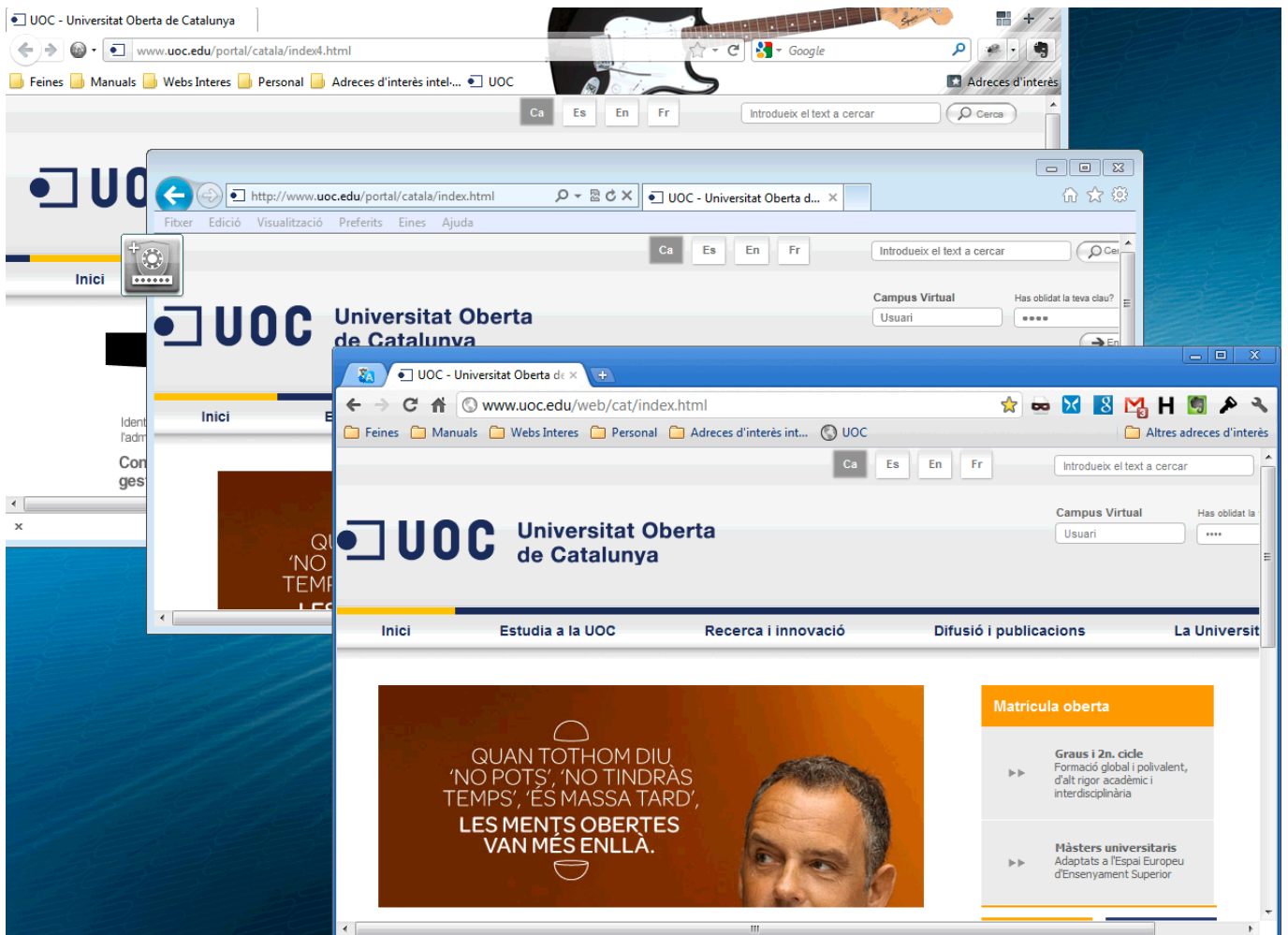
EGit també ens proporciona una nova perspectiva des d'on podem treure una llista dels commits realitzats...



... i mirar quins canvis es van introduir en els fitxers.

11.2.5 Navegadors Web

Per tal de poder desenvolupar les interfícies web s'ha fet servir els tres navegadors que actualment són més importants: Chrome, Firefox i Explorer, en les seves darreres versions. De totes maneres la interfície web no fa ús de javascripts o altres tipus de tecnologies que puguin ser considerades un risc per la verificació funcional de la interfície, per la qual cosa es pot dir que funcionarà en altres navegadors.



12. Patrons de disseny emprats en el desenvolupament

12.1 Introducció

Tal com trobem a la Wikipedia:

En l'enginyeria de programari, un **patró de disseny** és una solució general a un problema comú i recurrent en el disseny de programari. Un patró de disseny no és un disseny acabat que es pot transformar directament en codi; és una descripció o plantilla per resoldre un problema que es pot utilitzar en moltes situacions diferents.

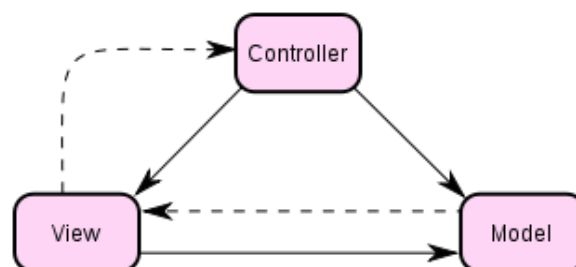
De l'estudi dels patrons i l'anàlisi del projecte els següent patrons que descrivim a continuació han estat seleccionats.

12.2 Model-vista-controlador

Tal com veurem més endavant, aquest patró s'ha fet servir especialment en els apartats de disseny d'interfícies web.

El patró ens permet fer una separació del nostre sistema en 3 capes.

- En la capa de model posarem tot el que té ha veure amb la persistència de dades. Emmagatzema i recupera informació sense que importi si la persistència es fa des d'una base de dades, fitxers de text, XML...
- La capa de vista presenta la informació obtinguda a la cap de model per tal que l'usuari pugui interactuar.
- El controlador es la capa que, depenen del que vol l'usuari, demana a la capa model les dades i crida la vista que li correspongui.



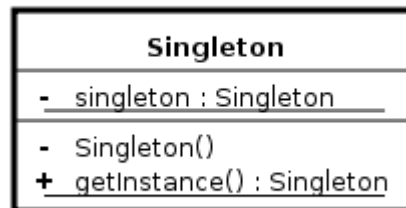
Podem obtenir més informació del patró a

<http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>

12.3 Singleton

Aquest patró ens serà útil en diversos casos. El seu ús ens permet restringir la creació d'objectes a un de sol per classe.

Això ens es d'utilitat per tal d'establir una única connexió a la base de dades o crear una única vegada una instància d'una classe on tenim dades de configuració.



Podem obtenir més informació del patró a http://en.wikipedia.org/wiki/Singleton_pattern

12.4 Front Controller

Aquest patró ens permet centralitzar tota les funcionalitats de les interfícies web en una única pàgina, de manera que en comptes de tenir diverses pàgines per cada acció només en tenim una de sola que s'encarrega de decidir el que cal cridar cada cop.

Podem obtenir més informació del patró a <http://java.sun.com/blueprints/patterns/FrontController.html>

13. Anàlisi del sistema

13.1 Introducció

En aquesta secció analitzarem les diferents parts del sistema i les seves funcionalitats, els actors i casos d'ús que el sistema haurà de tenir en compte i les interfícies que haurem de crear per tal d'interactuar amb el sistema.

13.2 Arquitectura del sistema

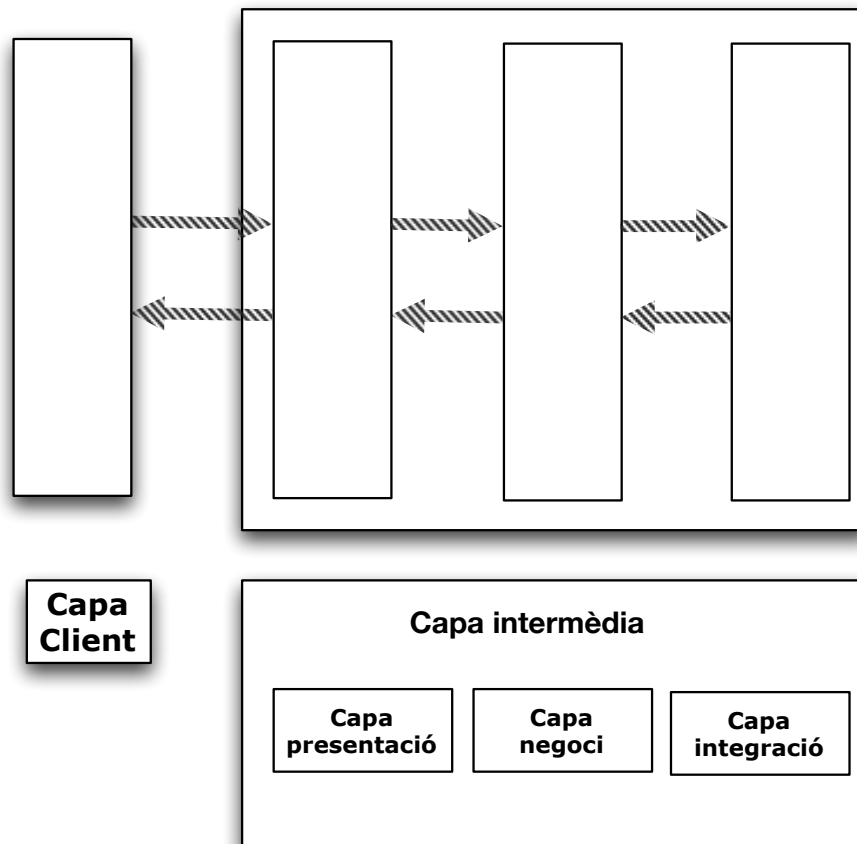
El sistema el podem dividir en dos grans parts.

- Per una part tenim el sistema amb el que interactuarà l'usuari. Aquesta part es l'encarregada de configurar el sistema. Es on podem gestionar els usuaris, grups, events, objectes i subscripcions. També tindrem en compte que certes parts del sistema podran ser utilitzades des de sistemes externs.
- Per altra banda tenim el sistema que s'encarrega de controlar els objectes que generen events i enviar de forma adequada les notificacions als usuaris. De nou tindrem en compte que sistemes externs poden tenir detecció d'events i voldran fer servir el nostre.

El sistema el podem dissenyar en termes d'arquitectura de la següent manera.

- Una capa client que es la que interactuarà amb el sistema. Aquesta capa client serà un client web que tindrà accés a la capa de presentació de la capa intermèdia. També dins d'aquesta capa tindrem el client de SMS que mostrarà els missatges rebuts al usuari, o el navegador del mòbil... es a dir el client que rebrà/consultarà les notificacions.
- La capa presentació: aquesta capa serà la que gestionarà tota la interfície web amb els client web. Dins la capa tindrem un patró model-vista-controlador que la modelarà. També per altra banda dins aquesta capa tindrem la capa de presentació de les notificacions(nosaltres l'anomenem missatgeria), com per exemple l'enviament de SMS, pàgines per dispositius mòbils amb els events, centre de notificacions de tercers...
- La capa de negoci serà la capa amb tota la lògica que s'encarrega de gestionar la supervisió dels events. Haurà d'accedir a la capa de presentació per tal d'accedir a les dades necessàries per consultar els events que cal controlar, quins usuaris han de rebre les notificacions...
- La capa d'integració serà la que proveirà de dades a la capa de negoci, per exemple les cues d'enviament.

El següent esquema il·lustra l'arquitectura descrita.



13.3 Diagrama de components

13.3.1 Capa de presentació

La capa de presentació la formarien els següents packages: View, Controller i Model. Aquest paquets modelen el patró model-vista-controlador per la interfície gràfica.

Dins de la cap View tindrem els paquets:

- VUsers: vista per tractar usuaris i grups
- VSubscriptions: vista per tractar subscripcions
- VExternalSubscriptions: vista per tractar subscripcions gestionades per tercers. En comptes d'una sortida "gràfica" aquí tindrem una API que tercers podran fer servir per tal de subscriure a usuaris externs a events que controla el sistema.
- VEvents: vista per tractar events

Dins la capa controller tindrem:

- CUsers: controlador d'usuaris i grups
- CSubscriptions: controlador de subscripcions
- CExternalSubscriptions: controlador de subscripcions gestionades per tercers.
- CEvents: controlador d'events

Dins la capa model tindrem:

- MUsers: model per accedir a les dades d'usuaris i grups
- MSubscriptions: model per accedir a les dades de subscripcions
- MEvents: model per accedir a les dades d'events

També considerem dins d'aquesta capa el paquet Messaging que s'encarrega de l'enviament "efectiu" de missatges. De moment aquest package només té un component, anomenat Messaging, per enviar missatges però en podríem tenir diversos depenent de si tenim, per exemple, diverses connexions amb operadors o agregadors de missatgeria SMS, o si volem fer arribar els missatges com a notificacions d'altres sistemes, o presentar-les en una pagina per dispositiu mòbil..

13.3.2 Capa de negoci

La capa de negoci la formen els paquets Observer i Dispatcher, que són els encarregats de manegar els events i enviar missatges.

Dins Dispatcher tenim els components:

- Dispatcher: que s'encarrega de fer arribar els missatges als usuaris
- Pull: que permet consultar els missatges dels usuaris
- QueueManager: que gestiona la cua de missatges

Dins d'Observer tenim els components:

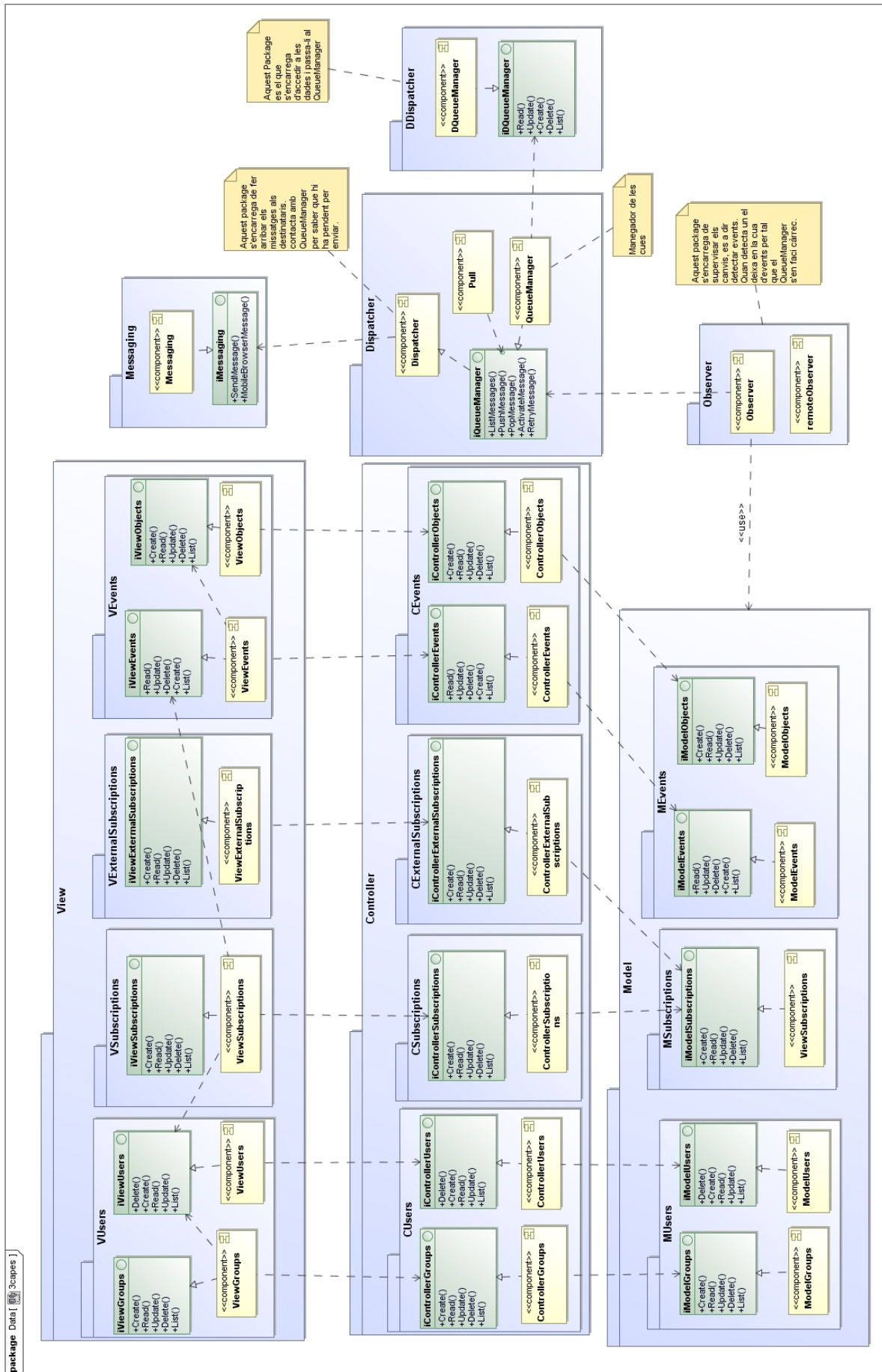
- Observer: que vigila els events controlats pel sistema
- remoteObserver: es un component accessible des de fora i que permet rebre notificacions d'events produïts en sistemes externs.

13.3.3 Capa integració

Per últim la cap d'integració la forma el paquet DDispatcher que dona servei a Dispatcher.

13.3.4 Diagrama de components

En el següent diagrama de components tenim representada l'arquitectura descrita. Per tal de no tenir un gràfic molt gran i il·legible s'han obviat les variables d'entrada i sortida de cada operació.

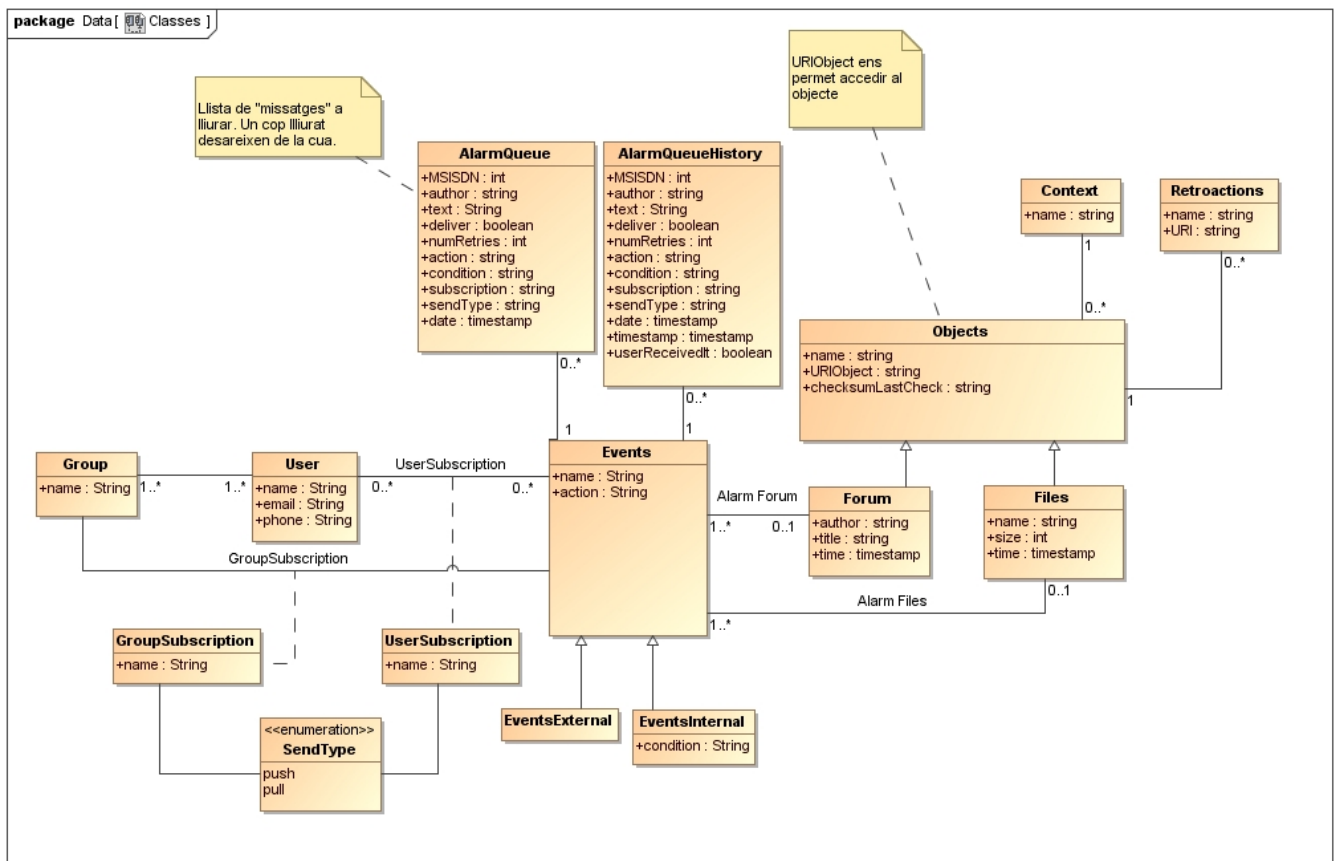


13.4 Model de dades

13.4.1 Diagrama de dades

Tot i que per la persistència de dades no s'ha escollit un sistema de gestió de base de dades relacional, podem fer servir un diagrama de classes per representar la informació que volem emmagatzemar. Es considera que tan en SQL com en NO SQL el que s'acaba representant són relacions i, tot i que a nivell d'implementació hi han grans canvis, a nivell de disseny comparteixen molts punts en comú.

En el diagrama podem veure les dues parts del sistema. Per una banda tenim les classes de grups, usuaris, objectes, events, context, retroaccions i subscripció que ens permeten modelar les dades de configuració del sistema, i per altra banda tenim la cua dels missatges a enviar i la cua amb missatges ha enviats, es a dir l'històric.



La modelització dels events no es prou acurada, ja que les relacions s'haurien de fer sobre EventsInterns i EventsExterns, però s'ha fet sobre Events per tal de no complicar el model. Precisament EventsInterns i EventsExterns ens permetrà gestionar events que es controlen des del sistema o bé notifikacions d'events de sistemes externs.

13.4.2 MongoDB

Donat que hem escollit mongoDB per la persistència de dades abans de parlar de la implementació feta cal fer una petita introducció a les bases de dades orientades a document.

En primer lloc cal dir que canvia una mica el lèxic:

- Les base de dades tenen col·leccions en comptes de taules
- Els registres s'anomenen documents
- Els elements dels documents s'anomenen camps com a les base de dades relacionals.
- Els documents no estan prefixats amb un "create table", això vol dir que podem tenir documents dins una mateixa col·lecció on els "camps" són diferents, es a dir que la base de dades es "schema-free"
- Les col·leccions, igual que en les base de dades relacionals, tenen índex que ens permeten accelerar les consultes, restringir repeticions de dades...

Així doncs pel projecte s'ha creat la base de dades *event_system* amb les següent col·leccions:

13.4.2.1 Events

En aquesta col·lecció tenim informació d'events:

- o `_id` : identificador únic
- o `name` : nom del event. Té una finalitat purament informativa.
- o `action` : es el nom del mètode que s'executa si la condició es certa. Serà un mètode del paquet Messaging.
- o `condition` : nom del mètode que s'ha d'executar per saber si un event s'ha produït. Si no apareix es perquè l'event es extern, sinó es un dels mètodes de un observador.
- o `object` : id del objecte que l'observador observa.
- o `Origin`: pot ser external o internal i ens diu si l'event es controlat per una plataforma externa.

Hem posat un índex únic a *name* per tal de no tenir repeticions.

Exemple:

```
/* 0 */
{
  "_id" : ObjectId("4ef09062a1ce9fa022000013"),
  "action" : "MobileBrowserMessage",
  "condition" : "fileModified",
  "name" : "Memòria PFC modificada",
  "object" : "4ef09052a1ce9f080f000000b",
  "origin" : "internal"
}
```

13.4.2.2 Groups

En aquesta col·lecció tenim informació dels grups d'usuaris:

- `_id` : identificador únic
- `name` : nom del grup

Hem posat un índex únic a `name` per tal de no tenir repeticions.

Exemple:

```
/* 0 */
{
  "_id" : ObjectId("4ec92a03a1ce9f9408000000"),
  "name" : "Grup 1"
}
```

13.4.2.3 Users

En aquesta col·lecció tenim informació dels usuaris.

- `_id` : identificador únic
- `email` : correu electrònic
- `groups` : grups als que pertany
- `name` : nom
- `phone` : numero de telèfon.
-

Hem posat un índex únic a `name` per tal de no tenir problemes de repeticions.

Exemple:

```
/* 0 */
{
  "_id" : ObjectId("4ec92a54a1ce9f9408000002"),
  "email" : "user1@uoc.edu",
  "groups" : ["4ec92a03a1ce9f9408000000", "4ec92a0ba1ce9f9408000001"],
  "name" : "Usuari 1",
  "phone" : "686881111"
}
```

13.4.2.4 Objects

En aquesta col·lecció tenim informació dels objectes:

- `_id` : identificador únic
- `name`: nom del event. Té una finalitat purament informativa
- `URIObject` : identificador per accedir al recurs. Per exemple una URL o un path a un fitxer.
- `type` : tipus d'objecte. En el prototipus només tenim Files o Fòrums però òbviament en podríem tenir tants com volguéssim.
- `context`: id del context al que pertany l'objecte
- `checksumLastCheck` : camp que ens permet identificar canvis en l'estat del objecte observat
- `lastInfo`: document amb la última informació que hem llegit del objecte.

Hem posat un índex únic a `name` i `type` per tal de no tenir repeticions.

Exemple:

```
/* 0 */
{
  "URIObject"      :      "http://groups.google.com/group/mongodb-
user/feed/rss_v2_0_msgs.xml",
  "_id" : ObjectId("4eefd16da1ce9fec23000001"),
  "checksumLastCheck" : "8df1f9c326e3ae9aab89c82873db81fd",
  "context" : "4efa3fb8a1ce9fdc05000001",
  "lastInfo" : {
    "time" : "20120101224914",
    "title" : "Re: [mongodb-user] Re: URGENT - Could Not Handle Heavy
Load",
    "author" : "(scotthernan...@gmail.com (Scott Hernandez))"
  },
  "name" : "Forum mongodb",
  "type" : "Forums"
}
```

13.4.2.5 Retroactions

En aquesta col·lecció tenim informació de les accions que els usuaris podran efectuar sobre els objectes:

- `_id` : identificador únic
- `name`: nom de la retroacció. Té una finalitat purament informativa
- `URI` : identificador per accedir a la acció. Per exemple una URL o un path a un fitxer executable.
- `object` : id del objecte al que pertany la retroacció

Hem posat un índex únic a `name` i `object` per tal de no tenir repeticions.

Exemple:

```
/* 0 */
{
  "_id" : ObjectId("4efba4a2a1ce9fe81a000001"),
  "name" : "Answer",
  "URI" : "http://blblabla",
  "object" : "4eefd16da1ce9fec23000001"
}
```

13.4.2.6 Context

En aquesta col·lecció tenim informació sobre els context dels objectes:

- `_id` : identificador únic
- `name`: nom del context.
-

Hem posat un índex únic a `name` per tal de no tenir repeticions.

Exemple:

```
/* 0 */
{
  "_id" : ObjectId("4efa3fb8a1ce9fdc05000001"),
  "name" : "Aula 1 PFC"
}
```

13.4.2.7 AlarmQueue

En aquesta col·lecció tenim informació dels missatges que s'han de lliurar:

- `_id` : identificador únic
- `eventId` : identificador del event
- `MSISDN` : numero de telèfon al que hem d'enviar el missatge.
- `autor`: nom del autor del missatge
- `text` : text que enviem
- `deliver` : amb un booleà marquem si el missatge s'ha d'enviar o no. Si el missatges es un PUSH valdrà cert sinó fins que l'usuari no faci un PULL valdrà fals.
- `numRetries` : comptador per gestionar una política de reintents basada en les vegades que s'ha intentat enviar un missatge i ha fallat.
- `action`: acció que s'ha d'executar
- `condition`: condició que va provocar l'event
- `subscription`: subscripció associada
- `sendType`: tipus d'entrega, pot ser push o pull
- `date`: data associada al event
- `dateCreate`: data en que es va inserir l'alarma.

Hem posat un índex únic a `eventId` i `MSISDN` per tal de no tenir problemes de spam.

Exemple:

```
/* 0 */
{
  "_id" : ObjectId("4f00e57ca1ce9f841c000002"),
  "eventId" : "4ef09062a1ce9fa022000013",
  "MSISDN" : "666123321",
  "author" : null,
  "text" : "Memòria (xrivadulla_PAC3).docx",
  "deliver" : false,
  "numRetries" : 0,
  "action" : "MobileBrowserMessage",
  "condition" : "fileModified",
  "subscription" : "Memories PFC",
  "sendType" : "PULL",
  "date" : "20111219174529",
  "dateCreate" : "20120102000012"
}
```

13.4.2.8 AlarmQueueHistory

En aquesta col·lecció tenim informació dels missatges que s'han lliurat:

- `_id` : identificador únic
- `eventId` : identificador del event
- `MSISDN` : numero de telèfon al que hem d'enviar el missatge.
- `autor`: nom del autor del missatge
- `text` : text que enviem
- `deliver` : amb un booleà marquem si el missatge s'ha d'enviar o no. Si el missatges es un PUSH valdrà cert sinó fins que l'usuari no faci un PULL valdrà fals.
- `numRetries` : comptador per gestionar una política de reintents basada en les vegades que s'ha intentat enviar un missatge i ha fallat.
- `action`: acció que s'ha d'executar
- `condition`: condició que va provocar l'event
- `subscription`: subscripció associada
- `sendType`: tipus d'entrega, pot ser push o pull
- `date`: data associada al event
- `dateCreate`: data en que es va inserir l'alarma.
- `timestamp`: data en que es va enviar l'alarma.
- `userReceivedIt`: camp per saber si l'usuari va rebre el missatge o bé si el sistema ha anul·lat l'enviament.

Exemple:

```
/* 0 */
{
  "_id" : ObjectId("4f00e57ca1ce9f841c000003"),
  "eventId" : "4eefd178a1ce9fec23000002",
  "MSISDN" : "686123123",
  "author" : "(scotthernan...@gmail.com (Scott Hernandez))",
  "text" : "Re: [mongodb-user] Re: URGENT - Could Not Handle Heavy Load",
  "deliver" : true,
  "numRetries" : 0,
  "action" : "SendMessage",
  "condition" : "NewPostForum",
  "subscription" : "Nou misstage Forum mongodb",
  "sendType" : "PUSH",
  "date" : "20120101224914",
  "dateCreate" : "20120102000012",
  "timestamp" : "20120102112639",
  "userReceivedIt" : true
}
```

13.4.2.9 Subscriptions

En aquesta col·lecció tenim informació de les subscripcions.

- `_id` : identificador únic
- `name`: nom de la subscripció. Té una finalitat purament informativa
- `events` : event al que pertany la subscripció
- `users | groups`: usuari o grup al que esta subscrit.
- `sendtype` : pot ser PULL o PUSH
- `type` : tipus de subscripció. Pot ser `groups`, `users` o `remoteUser`
- `origin`: pot ser `internal` o una cadena de caràcters que identifica un sistema extern.

Hem posat un índex únic a `users`, `groups`, `type` i `name` per tal de no tenir problemes de repeticions.

Exemple:

```

/* 0 */
{
  "_id" : ObjectId("4efc8ac3a1ce9fc00b000001"),
  "events" : {
    "_id" : ObjectId("4ef06fdaa1ce9fa02200000c"),
    "action" : "SendMessage",
    "condition" : "NewPostForum",
    "name" : "New message PHP Forum coding => General",
    "object" : "4ef06fc9a1ce9fd406000000",
    "type" : "internal"
  },
  "name" : "Nou missatge php foro",
  "sendtype" : "PUSH",
  "type" : "users",
  "users" : {
    "_id" : ObjectId("4eefd133a1ce9fec23000000"),
    "email" : "revolucio@gmail.com",
    "groups" : ["4eefd12aa1ce9fac19000000"],
    "name" : "Xavier Rivadulla",
    "phone" : "686123123"
  },
  "origin" : "internal"
}
/* 5 */
{
  "_id" : ObjectId("4efc8a05a1ce9fc012000000"),
  "name" : "Memories PFC",
  "sendtype" : "PULL",
  "type" : "groups",
  "events" : {
    "_id" : ObjectId("4ef09062a1ce9fa022000013"),
    "action" : "MobileBrowserMessage",
    "condition" : "fileModified",
    "name" : "Memòria PFC",
    "object" : "4ef09052a1ce9f080f00000b"
  },
  "groups" : {
    "_id" : ObjectId("4eefd12aa1ce9fac19000000"),
    "name" : "Grup 1"
  },
  "origin" : "internal"
}

```

13.5 Casos d'ús

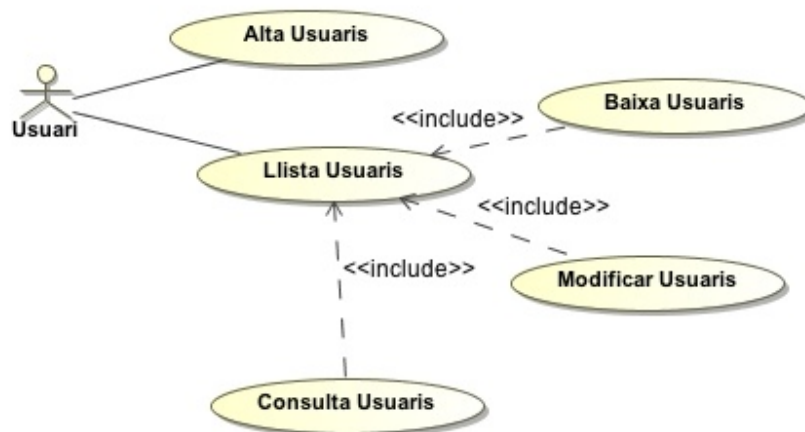
13.5.1 Introducció

En aquest apartat discutirem sobre els casos d'ús del sistema. Com ja hem comentat els sistema esta dividit en dos grans parts: la part de gestió i la de gestió d'events. Això fa que puguem parlar dels casos d'ús seguint aquesta divisió. Per altra banda també veurem els casos d'ús que sistemes externs poden fer sobre el nostre sistema.

13.5.2 Casos d'us de gestió del sistema

La gestió de cada element gestionable via interfície sempre serà idèntica. Es basarà en un CRUD per cadascun, es a dir que es podrà crear, consultar, actualitzar i esborrar, a banda de fer una llista d'elements per facilitar la selecció dels elements.

Presentem com a cas d'ús la gestió d'usuaris.

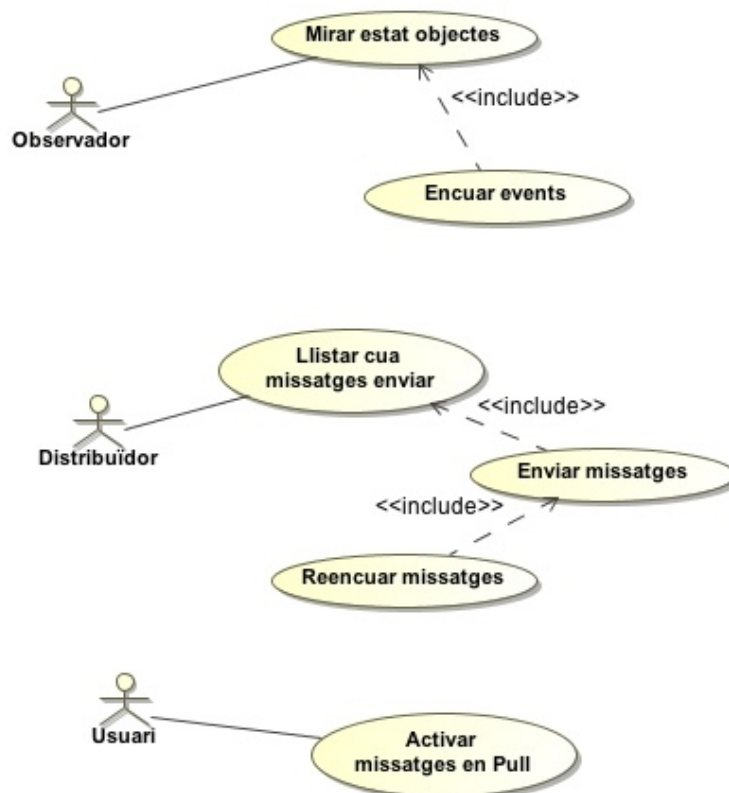


Donat que hem decidit no fer una gestió d'autenticació i autorització l'únic actor del sistema serà l'usuari, tal com podem veure en el diagrama .

13.5.3 Casos d'ús gestió d'events

En la gestió d'events trobem tres actors:

- Observador: es el programa que de forma periòdica revisa l'estat dels objectes i actua en cas de detectar un nou event.
- Distribuïdor: es el programa dispatcher que s'encarrega d'enviar els missatges que estan pendents de ser enviats. En cas que l'enviament falli també ha d'aplicar la política de reintents.
- Usuaris: els usuaris poden tenir missatges pendents d'enviar però que necessiten de l'acció del usuari per ser enviats, es a dir que són missatges de tipus *pull*.



Per tal de tenir més clar els sistema de gestió d'events esposarem una sèrie de casos i com els tractarà els sistema.

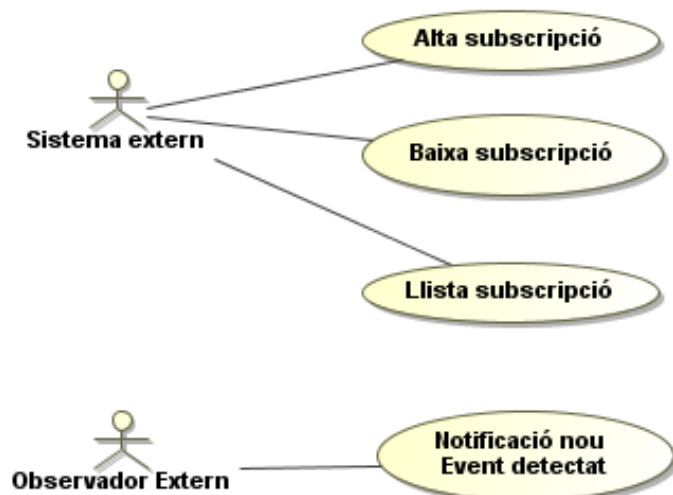
1. *L'usuari es subscriu per rebre alertes automàticament, mitjançant un SMS, cada cop que es publica un nou comentari en el Fòrum XXX.*

- L'usuari el tenim donat d'alta en el sistema.
- Haurà seleccionat dins de la llista de subscripcions possibles les que es relacionen amb els events del Fòrum XXX i haurà triat rebre les alertes de forma automàtica. Això fa que es creï un nou registre a Subscriptions.
- El procés observador que vigila els canvis en el sistema revisa periòdicament si hi ha algun canvi. Quan un nou event es detectat es registra en la cua i es marca per entregar.

- El dispatcher de forma periòdica revisa tots els missatges encuats que s'han d'enviar. En el cas que l'entrega no hagi sigut possible aplicarà la política de reintents associada.
2. *L'usuari es subscriu per rebre alertes manualment, mitjançant un SMS, cada cop que es publica un nou comentari en el Fòrum XXX.*
- L'usuari el tenim donat d'alta dins el sistema.
 - Haurà seleccionat dins de la llista de subscripcions possibles les que es relacionen amb events del Fòrum XXX i haurà triat rebre les alertes de forma manual. Això fa que es creï un nou registre a UserSubscription.
 - El procés observador que vigila els canvis en el sistema revisa periòdicament si hi ha algun canvi. Quan un nou event es detectat es registra en la cua associada al event però no es marca encara per entregar. Si ja existeix el missatge encuat, per tal de no enviar diversos missatges a l'usuari amb la mateixa informació, no s'encauarà.
 - El procés per activar l'enviament del missatge s'activarà quan l'usuari enviï l'ordre al sistema. A efectes pràctics podem suposar que això es pot fer, per exemple, per les següents raons:
 - L'usuari envia un SMS a un numero curt amb una paraula clau determinada. El parser de missatges detecta que l'usuari vol rebre la notificació i la activa a la cua. Suposarem que el mètode per fer aquesta activació es cridar una URL.
 - L'usuari marca en una pàgina web rebre les notificacions. De nou això provocarà que l'activació es faci cridant una URL d'activació.
 - El dispatcher de forma periòdica revisa tots els missatges encuats i que s'han d'enviar. En el cas que l'entrega no hagi sigut possible aplicarà la política de reintents associada.

13.5.4 Casos d'us per sistemes externs

En aquest cas tenim dos actors. Per una banda *Sistema extern* es l'actor que gestiona les subscripcions. Aquestes subscripcions no es relacionen amb usuaris del sistema sinó del sistema Extern. *Observador Extern* s'encarregarà d'enviar les notificacions de nous events que gestioni el sistema extern.



14. Interfícies

14.1 Interfícies gràfica

Les interfícies gràfica d'usuari serà la part on els usuaris podran interactuar amb els sistema. En el prototipus la llengua escollida es l'anglès per tal de fer l'eina el més universal possible. Queda com a millora permetre múltiples llengües.

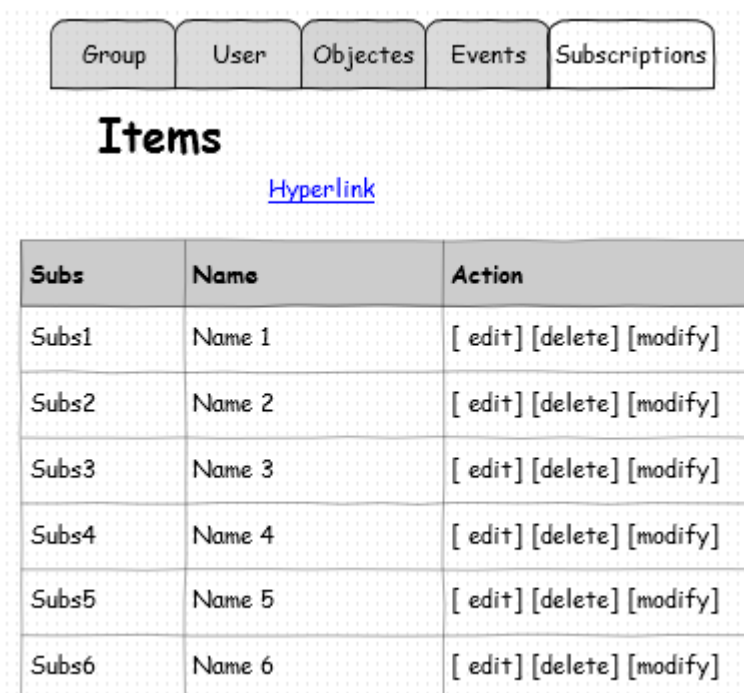
Donat que no es objectiu del prototipus una interfície visualment atractiva les pantalles seran lo més simples possibles.

Les interfícies gràfiques, tal com em vist en l'arquitectura, permetran gestionar els següents elements:

- Grups
- Usuaris
- Context
- Objectes (fòrums i fitxers)
- Retroaccions
- Events
- Subscripcions (usuaris i grups)

14.1.1 Llistes

Les llistes seran l'acció principal de gestió, ens mostraran un llistat de tots els elements i per cadascun ens permetrà editar, esborrar o modificar-lo. Per altra banda des de la llista se'ns permetrà afegir nous elements.



Subs	Name	Action
Subs1	Name 1	[edit] [delete] [modify]
Subs2	Name 2	[edit] [delete] [modify]
Subs3	Name 3	[edit] [delete] [modify]
Subs4	Name 4	[edit] [delete] [modify]
Subs5	Name 5	[edit] [delete] [modify]
Subs6	Name 6	[edit] [delete] [modify]

14.1.2 Grups

La gestió de grups serà molt simple ja que només tenim un camp. La següent interfície representa les altes i modificacions.

The image shows a mobile application screen titled "Group". It features a single text input field labeled "Name" with the placeholder text "text". Below the input field is a grey button labeled "Ok".

14.1.3 Usuaris

La gestió de usuaris serà molt simple ja que només tenim el camp nom, correu, telèfon i el de pertinença a grups. La següent interfície representa les altes i modificacions.

The image shows a mobile application screen titled "User". It features three text input fields labeled "Name", "Email", and "Phone", each with the placeholder text "text". Below these fields is a section labeled "Group" with four checkboxes, each followed by a group name: "Group 1", "Group 2", "Group 3", and "Group 4". At the bottom of the screen is a grey button labeled "Ok".

14.1.4 Context

La gestió de context serà molt simple ja que només tenim un camp. La següent interfície representa les altes i modificacions.

Context

Name

Prototipus de la interfície 'Context'. El títol és 'Context'. Hi ha un camp de text etiquetat 'Name' amb el valor 'text'. A sota hi ha un botó 'Ok'.

14.1.5 Objectes

A la gestió d'objectes tenim el camp nom, URI del objecte i el context. La següent interfície representa les altes i modificacions.

Tindrem diversos objectes accessibles depenent del tipus que volem administrar. En el prototipus tenim dues opcions per objecte fitxer i fòrums.

Objects

Name

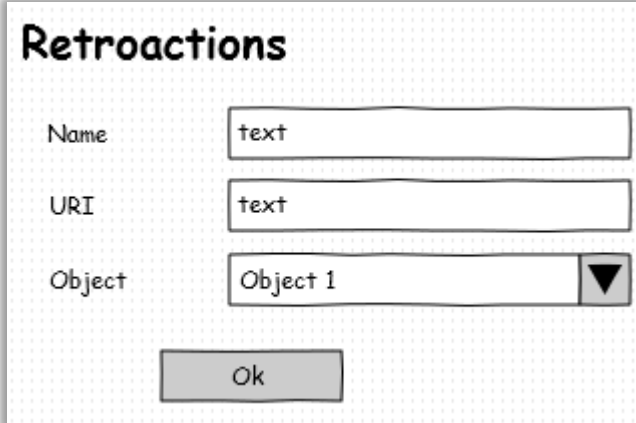
URI Object

Context ▼

Prototipus de la interfície 'Objects'. El títol és 'Objects'. Hi ha tres camps: 'Name' amb el valor 'text', 'URI Object' amb el valor 'text', i 'Context' amb el valor 'Context 1' i una fletxa cap avall. A sota hi ha un botó 'Ok'.

14.1.6 Retroaccions

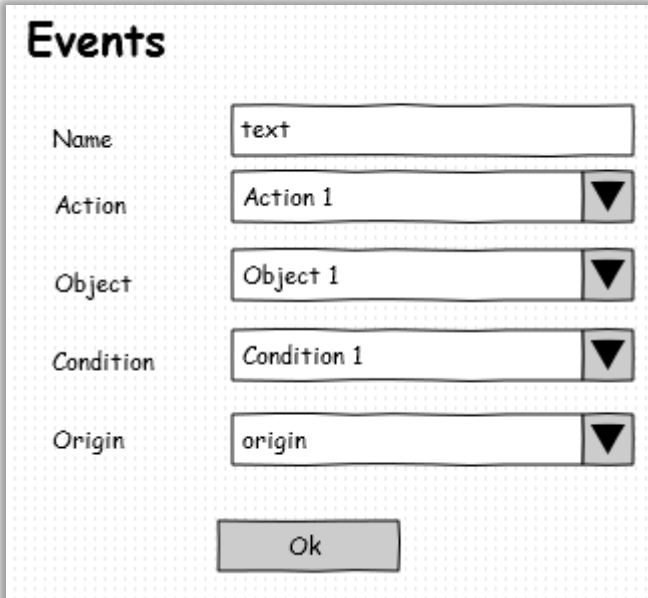
A la gestió d'objectes tenim el camp nom , URI i l'objecte. La següent interfície representa les altes i modificacions.



The screenshot shows a form titled "Retroactions" on a dotted background. It contains three input fields: "Name" with the value "text", "URI" with the value "text", and "Object" with a dropdown menu showing "Object 1". Below the fields is an "Ok" button.

14.1.7 Events

A la gestió d'events haurem de seleccionar la condició, l'acció, l'objecte i l'origen que l'event gestiona. Posarem un nom per qüestions informatives.



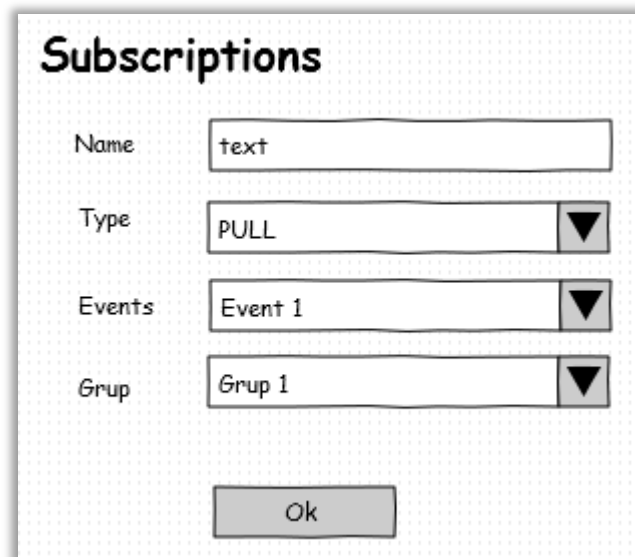
The screenshot shows a form titled "Events" on a dotted background. It contains five input fields: "Name" with the value "text", "Action" with a dropdown menu showing "Action 1", "Object" with a dropdown menu showing "Object 1", "Condition" with a dropdown menu showing "Condition 1", and "Origin" with a dropdown menu showing "origin". Below the fields is an "Ok" button.

14.1.8 Subscripcions

Les subscripcions les tindrem separades segons si les volem fer per usuaris o grups.

A la gestió de subscripcions haurem d'escollir el tipus , l'event i el grup (o usuari) i posar un nom per temes informatius.

L'exemple següent correspondria a la subscripció de grups.



The image shows a mobile application form titled "Subscriptions". The form is set against a dotted grid background. It contains the following elements:

- Name:** A text input field containing the word "text".
- Type:** A dropdown menu with "PULL" selected and a downward arrow on the right.
- Events:** A dropdown menu with "Event 1" selected and a downward arrow on the right.
- Grup:** A dropdown menu with "Grup 1" selected and a downward arrow on the right.
- Ok:** A rectangular button with the text "Ok" centered inside.

14.2 Interfícies de servei web

A banda de les interfícies gràfica es proporcionaran interfícies de servei web per interactuar amb el sistema. Podríem pensar en aquestes interfícies com les API's a tercers que el sistema ofereix.

14.2.1 PULL

Aquesta interfície permetrà a l'usuari demanar al sistema que li envii un missatge pendent, es a dir que li permetrà fer un PULL dels missatges encuats.

Des d'on es crida la interfície queda fora de l'abast d'aquest projecte. Dos possibilitats serien:

- L'usuari envia un missatge (SMS) i el sistema l'interpreta com a una petició de PULL i fa la crida corresponent
- L'usuari fa click en una opció d'una pàgina web que desencadena la crida a la interfície.

De moment la interfície esta implementada d'una manera molt senzilla. Per cridar-la cal accedir de la següent manera:

<http://domini/directoriDispatcher/pull.php?id=idmissatge>

Passant el id del missatge aquest s'activa per ser enviat.

A banda de l'activació de cada missatge els sistema també implementarà una interfície per activar tots les alarmes que estiguin pendents d'un usuari, entre unes dates determinades. Per cridar-la cal accedir de la següent manera:

http://domini/directoriDispatcher/pull.php?MSISDN=num_tel&start_date=datainici&end_date=date_fi

També podríem pensar en accedir a una pàgina del component Messaging que crides la URL per obtenir tots els missatges PULL i els mostres en aquesta pàgina.

Per cridar-la cal accedir de la següent manera:

http://domini/directoriDispatcher/pull.php?MSISDN=num_tel&start_date=datainici&end_date=date_fi&sendList

Com a millora podríem definir la interfície en REST i discutir com hauria de ser la seguretat per accedir a la interfície. Però aquest són qüestions que el projecte no pretén cobrir.

14.2.2 Observador remot

Aquesta interfície sorgeix de la necessitat de que no tots els events els podrem detectar de forma eficient des del sistema i que tercers ho poden fer millor, però aquest tercers no tenen capacitat per gestionar aquestes notificacions.

D'aquesta manera pot ser que un servei de fòrum ens avisi cada cop que hi ha una nou comentari en un fòrum, o que un taulell ens avisi quan es puja un nou document.

La interfície que s'ha creat es molt simple i està dins el Package Observer.

<http://domini/directoriObserver/remoteObserver.php?eventId=id&info=text>

Com podem veure els dos paràmetres d'entrada són l'identificador d'event i un text.

El fet de tenir un identificador d'event vol dir que prèviament em de donar-los d'alta en el sistema. Això permetrà identificar l'event que s'ha produït, a banda que ens permet saber quins events externs té el sistema.

El paràmetre info ens passarà un text que pot ser, per exemple, el que rebrà l'usuari.

14.2.3 Subscripcions remotes

Una altra qüestió interessant al integrar tercers es saber que hem de fer amb els usuaris. Idealment haurien d'estar donats d'alta en el sistema, per això implica una redundància de dades que potser no interressi als tercers que vulguin integrar el nostre sistema.

Les subscripcions remotes intenten solucionar el problema permetent associar usuaris que no estan en el sistema amb events, que si que estan.

En aquest cas tenim que oferir una interfície que permeti manegar de forma completa les subscripcions, es a dir que permeti fer altres, baixes, consultes i modificacions.

La interfície serà :

http://domini/directoriFrontEnd/remoteSubscriptions.php?user=12:677123123:NomUser&eventId=12&action=create&name=test1&sendtype=PULL&_id=4eca7d28a1ce9fcc0e000014

on tindriem que :

- user : seria la informació del usuari. Podriem pensar en que ens passessin una cadena de caràcters separats per dos punts amb el id d'usuari del sistema extern, el numero de telèfon i el nom. Es a dir una cadena del tipus idUser:numTel:Nom
- eventId: seria l'identificador de un dels events que ja existeixen
- action: l'acció que es vol fer llista, crear, llegir, actualitzar i esborrar
- name: un nom per qüestions informatives
- sendtype: o bé PULL o PUSH
- _id=identificador de subscripció

Ens quedaria, de nou, la possibilitat de crear la interfície en REST i discutir el tema de la seguretat. Per exemple per validar el numero de telèfon hauríem d'implementar un doble opt-in. El primer opt-in s'haurà efectuat al demanar la subscripció per part de l'usuari en el sistema extern. Llavors al arribar-nos la petició podríem emmagatzemar les dades de la subscripció en una col·lecció de la base de dades de *peticions per comprovar*. Llavors per fer el doble opt-in enviaríem un SMS que un cop l'usuari respongués afirmativament faria passar la subscripció de la col·lecció *de peticions per comprovar* a la definitiva. De totes maneres aquests són aspectes que el projecte no contempla.

14.3 Access via navegador mòbil

Ja hem comentat de la possibilitat de tenir una pàgina, que es mostraria en un navegador d'un terminal mòbil, les alarmes de tipus PULL que té l'usuari.

A banda d'aquesta pàgina en podríem tenir:

- Una pàgina que mostri una alarma determinada
- Una pàgina que mostri qualsevol tipus d'alarma entre unes dates determinades
- Una pàgina que mostri l'historial d'alarmes rebudes

15. Conclusions

Un cop ha finalitzat la realització de totes les activitats definides en el Pla de Treball podem donar per finalitzat el Projecte.

Aquest PFC mica a mica s'ha anat centrant més en explorar les possibilitats teòriques que un sistema d'events hauria de tenir, que no pas en construir un prototipus robust que permeti ser considerat una versió beta. De fet el prototipus es més una prova de concepte (o *proof of concept*, com es el agrada dir als anglosaxons) per demostrar si les idees són possibles o no.

Una de les conclusions a les que he arribat es que els sistema hauria de donar molta importància al fet que tercers el puguin fer servir. El fet de convertir-se en un HUB (eix) on sistemes externs en puguin fer un ús per enviar alarmes crec que obre moltes possibilitats i pot assegurar l'èxit del sistema.

Un dels principals reptes que té el sistema es detectar canvis en els objectes, es a dir adonar-se que un event s'ha produït. La delegació d'aquesta feina faria el sistema més precís. Òbviament això faria augmentar la necessitat de dissenyar de forma més robusta i segura les interfícies que fan servir els tercers.

Considero que cada cop la diferència entre dispositius mòbils i no mòbils s'anirà fent més difosa. Que la plataforma sigui prou flexible per ser accessible des de qualsevol dispositiu es el que ha de ser important. De fet ara un correu electrònic o una pàgina web pot ser accessible des d'un terminal mòbil, per tant aquesta distinció entre mòbil i no mòbil ja no es tan important. És per aquest raó que de fet durant el desenvolupament de la plataforma el concepte "mòbil" no ha sigut lo més important del projecte.

Per altra banda durant la recerca del projecte m'he trobat sistema d'alarmes comercials que estan pensats per fer-se servir durant catàstrofes i que garanteixen alta disponibilitat en els condicions més exigents. Això m'ha fet adonar de la importància de desenvolupar un sistema que des del punt de vista de l'arquitectura sigui molt robust. Malauradament no he pogut fer test exhaustius del sistema per tal de verificar la solidesa de la solució davant de problemes, però crec que no estic massa lluny d'un sistema robust.

El projecte també m'ha permès entrar en contacte amb les base de dades noSQL i l'experiència ha sigut molt positiva. Em queda el dubte de com respondria en un entorn real de producció, però el que es clar es que el fet que sigui schema-free dóna un gran grau de llibertat per desenvolupar amb un cost reduït.

16. Possibles millores del sistema

Donat que el prototipus creat es més una prova de concepte que un producte final les millores possibles són diverses.

Se m'acut fer mòduls per CMS, com Joomla, per tal d'afegir la possibilitat de fer servir el sistema en sistemes externs

Tampoc ens podem oblidar de l'administració via web. Sol passar que després de fer una cosa la tornaries a fer completament diferent, i aquest es el cas. A banda de la necessitat d'afegir autenticació i autorització, si l'hagués de refer la faria més intuïtiva i mes "contextual", es a dir que no hi hagués d'una forma tan marcada, en forma de menús, les diferents opcions, sinó que aquestes apareguessin en el seu lloc de forma natural. Per exemple des del punt de vista de l'usuari un cop autenticat sortiria les subscripcions que té, podria consultar per cada una les alarmes que té pendents (mitjançant una nova finestra però sense recarregar la pàgina existent), podria cancel·lar les subscripcions, programar rebre les alarmes en uns dies i hores preestablertes (per tal de no rebre determinades alertes el cap de setmana, per exemple).

Probablement a nivell de perfils d'usuaris podríem parlar d'usuaris finals (es a dir subscriptors), sistemes externs (plataformes externes que poden manegar les subscripcions als seus objectes, els seus usuaris...) i administrador del sistema (que podria crear nous usuaris, nous objectes, events....)

Per altra banda la interacció des del navegador mòbil del usuari també obre tot un camp de possibilitats que en el prototipus només s'intueix. Caldria implementar una autenticació i autorització de manera que els usuaris tinguessin un lloc on consultar tota la informació que tenen associada, i poguessin gestionar-la de forma més eficient. Fins i tot podríem generar aplicacions per PC, Android o iOS per fer aquest interacció.

També es podria generar els serveis web fent servir REST i no pas el pas de paràmetres via GET. Això permetria generar una API més entenedora i més usable.

17. Vocabulari

En el present document han anat apareixent una sèrie de termes que formaran part del vocabulari del projecte. La present taula servirà per fixar-los.

Terme	Definició
Usuari	Cadascuna de les persones que fan ús del sistema.
Grup	Associació d'usuaris.
Dispositiu mòbil	Aparells de petit tamany amb capacitat de processament i connexió permanent o intermitent a una xarxa informàtica.
Event	Canvi en l'estat d'un atribut d'un objecte.
Missatge	Informació que l'emissor envia al receptor.
Canal	Mitjà de transmissió dels missatges.
Estat	Configuració única d'informació. En el projecte associem un estat amb un atribut. L'estat correspondria a un dels atributs donat un instant de temps.
Objecte	Cadascun dels elements que seran observats.
Atributs del objecte	Cadascuna de les propietats d'un objecte que poden ser observables.
Observador	Programa que comprova si un objecte a tingut algun canvi en un atribut
Subscripció	Lligam entre grups o usuaris i objectes que farà que els primers rebin informació cada cop que es produeixi un event.
Desubscripció	Anul·lació del lligam entre grups o usuaris i objectes.
Cua	Magatzem ordenat del sistema d'enviaments d'events.
URI	Uniform Resource Identifier, identificador uniforme de recursos. Consta d'un identificador del mètode d'accés al recurs i un nom del recurs.
PUSH	Modalitat d'enviament que es dona quan es produeix l'event sobre l'objecte.
PULL	Modalitat d'enviament que es dona quan l'usuari demana consultar els events produïts sobre l'objecte.

18. Bibliografia

Aquests són les fonts d'informació utilitzades durant el projecte.

18.1 Eines

- Pencil project <http://pencil.evolus.vn/>
- Eclipse for PHP developers: <http://www.eclipse.org/>
- MongoVUE : <http://www.mongovue.com/>
- MongoHub: <http://mongohub.todayclose.com/>
- EasyPHP : <http://www.easyphp.org/>
- MagicDraw UML: <http://www.magicdraw.com>

18.2 Lliberies externes utilitzades

- HAWHAW: <http://hawhaw.de>
- Simplepie: <http://simplepie.org>

18.3 Informació

- Wikipedia.
- Termcat : http://www.termcat.cat/ca/Diccionaris_En_Linia/123
- MongoDB schema design: <http://www.mongodb.org/display/DOCS/Schema+Design>
- Git with Eclipse : <http://www.vogella.de/articles/EGit/article.html>
- PHP Manual: <http://us.php.net/manual/en>

18.4 Articles

- Introducción a MVC con PHP, segunda parte: <http://www.jourmoly.com.ar/introduccion-a-mvc-con-php-segunda-parte/>
- Clean and pure CSS FORM design: <http://woork.blogspot.com/2008/06/clean-and-pure-css-form-design.html>