

SMAC: “Social Media en l’atenció al client”

Memòria de Projecte de Final de Grau

Grau Multimèdia
Enginyeria web

Autor: Joan Madrid López

Consultor: Ignasi Lorente Puchades

Professor: Carlos Casado Martínez

20 / 06 / 2013

Crèdits

Reconeixement-NoComercial-CompartirIgual 3.0



Dedicatòria

*Zoe,
que tanta ajuda m'ha aportat
tant a nivell personal
com de coneixement per al projecte.*

Abstract

El Social Media, és un àmbit que es refereix a aquella interacció mitjançant plataformes socials que facilita l'intercanvi de informació. I aquest concepte és un gran denominador comú entre totes les empreses que volen tenir presència online. Normalment nosaltres coneixem una d'aquestes parts, on molts negocis i marques es donen a conèixer, fan promocions què, entre d'altres paraules, s'anomena publicitat.

Però aquí no s'acaba la seva utilitat, ja que coneixent i investigant una mica aquest mercat, o simplement fer una senzilla cerca a Google, hom es pot adonar que l'atenció al client mitjançant aquestes plataformes és una tendència a l'alça.

Portar tota aquesta gestió per una gran empresa acaba suposant una petita ampliació al departament d'atenció al client, un nou desenvolupament a l'eina que facin servir on els costos poden ser poc representatius; però quan parlem de petites i mitjanes empreses aquestes normalment no poden assolir els costos.

Aquest Treball de Fi de Grau pretenen formular una resposta per a aquesta demanda mitjançant l'anàlisi i desenvolupament d'un projecte per a crear una eina que permeti gestionar l'atenció al client de les xarxes socials, en quan a recepció, emmagatzemament, categorització i resposta de totes les peticions generades. Transversalment a aquest fet, permetrà generar estadístiques útils per a informes de Community Managers i interessats d'aquestes dades.

Tot aquest projecte estarà desenvolupat en una aplicació web que tindrà aquestes característiques:

- Importació automàtica de dades de les xarxes socials més utilitzades (Twitter i Facebook), categorització i gestió d'un gran volum de dades.
- Formació de tickets de consulta per l'atenció al client, gestió de perfils i definició d'estat.
- Generació d'estadístiques per a responsables de la gestió de les xarxes socials de la empresa o marca en qüestió

Estarà desenvolupat sota el framework Symfony 2 en llenguatge PHP, que servirà de base per a connexió d'altres tecnologies que permetin resoldre cada problema de la millor manera: MySQL i MongoDB però a gestió de dades, Javascript amb la llibreria jQuery per a la implementació de la interfície i LESS per a la definició d'estils.

Paraules clau: Treball de Fi de Grau, Social Media, atenció al client, desenvolupament, programari, pàgina web, xarxes socials

Índex

Crèdits	2
Dedicatòria	3
Abstract	4
Figures i taules.....	9
1. Introducció	10
2. Descripció.....	11
2.1. Exemple de treball real	11
2.1.1. Alimentació de dades.....	11
2.1.2. Categorització i desambiguació.....	12
2.1.3. Resolució d'incidències.	12
2.1.4. Anàlisi	12
3. Objectius.....	14
3.1. Generals	14
3.2. Específics	14
4. Continguts	15
4.1. De quines eines disposa l'aplicació?	15
4.2. Exemple d'escenari d'ús de l'aplicació web	16
5. Metodologia	19
6. Arquitectura de l'aplicació	20
6.1. Estructura bàsica de desenvolupament.....	20
6.2. Comunicació.....	22
6.3. Escalabilitat	22
6.4. Escenari final	23
7. Plataforma de desenvolupament.....	24
7.1. Màquina virtual	24
7.2. Control de versions	25
7.3. Projecte	26
8. Planificació	27
9.1. Planificació inicial	27
9. Procés de desenvolupament.....	29
9.1. Anàlisi previ	29
9.2. Anàlisi tècnic.....	29

9.3. Disseny	29
9.4. Desenvolupament	29
9.5. <i>Bugfixing</i>	29
9.6. Finalització.....	30
10. Frameworks, APIs i llibreries utilitzades.....	31
10.1. Symfony 2.....	31
10.2. jQuery.....	31
10.3. Bootstrap.....	31
10.4. Twitter API.....	32
10.5. HighCharts.....	33
11. Diagrames UML.....	34
11.1. Diagrama de casos d'ús.....	34
11.2. Diagrama de classes.....	39
12. <i>Mockups</i>	40
12.1. Entrada.....	41
12.2. Projecte.....	42
12.3. Missatges.....	43
12.4. Tickets.....	44
12.5. Informes.....	45
13. Perfils d'usuari.....	46
13.1. Community Manager's.....	46
13.1.1. Context.....	46
13.1.2. Factors demogràfics.....	46
13.1.3. Entorn.....	46
13.1.4. Productes semblants.....	47
13.2. Atenció al client.....	47
13.1.1. Context.....	47
13.1.2. Factors demogràfics.....	47
13.1.3. Entorn.....	47
13.1.4. Productes semblants.....	47
13.3. Conclusions.....	48
14. Usabilitat i UX.....	49
14.1. Navegació.....	49
14.2. <i>Sitemap</i>	49

14.3. DCU.....	50
15. Seguretat	51
15.1. Seguretat del servidor	51
15.2. Seguretat de l'aplicació	52
16. Tests	53
16.1. Unit testing.....	53
16.2. Selenium IDE	54
17. Versions de l'aplicació	56
17.1. Versió Alpha	56
17.2. Versió Beta	56
17.3. Versió 1.0.....	56
18. Requisits d'instal·lació	57
19. Instruccions d'instal·lació	58
20. Instruccions d'ús.....	60
20.1. Creació d'un usuari.....	60
20.2. Creació i selecció de projecte.....	61
20.3. Configuració del projecte	62
20.4. Classificant els primers missatges	64
20.5. Treballant amb Tickets	65
20.6. Resultats i informes.....	67
20.7. Canvi de password	68
21. Bugs	69
21.1. Llistat de bugs.....	69
22. Projecció a futur	70
22.1. Futur immediat.....	70
22.2. Futur pròxim.....	70
22.3. Futur llunyà	71
23. Pressupost	72
24. Anàlisi de mercat.....	73
24.1. Estudi de la competència	74
25. Màrqueting i vendes	75
25.1. Pla de Branding	75
25.2. Pla de promoció	75
25.3. Política de preus.....	75

25.4. Estratègia de venta.....	76
26. Conclusions.....	77
Annex 1. Lliurables del projecte.....	78
Codi	78
Documentació	78
Autoinforme d'avaluació.....	78
Annex 2. Codi font (extractes).....	79
Annex 3. Llibreries i codi extern utilitzat.....	91
Annex 4. Captures de pantalla	92
Annex 6. One-page business plan	99
Annex 7. Glossari.....	100
Annex 8. Bibliografia (webgrafia).....	102

Figures i taules

Il·lustració 1. Cas real de SAC	13
Il·lustració 2. Mètode cascada	19
Il·lustració 3. Arquitectura web	20
Il·lustració 4. Arquitectura final	23
Il·lustració 5. Entorn de treball	24
Il·lustració 6. Captura de Bitbuket.org	25
Il·lustració 7. Planificació de tasques	27
Il·lustració 8. Diagrama de Gantt	28
Il·lustració 9. Exemple.....	33
Il·lustració 10. Casos d'ús	34
Il·lustració 11. Classes de SMAC	39
Il·lustració 12. Mockup de Login	41
Il·lustració 13. Mockup de projecte.....	42
Il·lustració 14. Mockup de missatges	43
Il·lustració 15. Mockup de ticket.....	44
Il·lustració 16. Mockup de informe	45
Il·lustració 17. Datacenter vigilat	51
Il·lustració 18. Diagrama dels 2 passos de seguretat.....	52
Taula 1. Exemple de taula de bugs.....	69
Taula 2. Llistat de bugs.....	69
Taula 3. Pressupost de RRHH.....	72
Taula 4. Pressupost de material	72
Taula 5. Estudi de la competència	74
Taula 6. Preus de SMAC	75

1. Introducció

Exemple:

Imaginem: una dona va a fer la seva compra de la setmana a un supermercat Caprabo i aquell dia la responsable de la fruiteria l'atén molt malament, i no només això sinó que a més les peces de fruita que agafa estan totes en mala qualitat.

En aquesta situació es presenten possibilitats:

- 1. La dona torna a la tenda i es queixa al responsable de la tenda, o directament a la persona que la va atendre.*
- 2. La dona truca a l'atenció al client de Caprabo i presenta una queixa.*
- 3. La dona es connecta a la pàgina del Facebook o al Twitter de Caprabo i llença una queixa al públic, ja que en aquell moment està molt enfadada, de manera que tothom pugui llegir lo dolents que son tots els fruiters del Caprabo.*

Cada vegada més ens trobem en situacions com aquesta, i com molts poden arribar a dir, és el doble fil de l'espasa que s'anomena Social Media, és a dir, promocionar-se a aquestes xarxes socials (i públiques), on pots arribar a una gran quantitat de gent, però a la vegada tothom ho pot fer, ja que els missatges son públics per a tots.

Els encarregats de gestionar aquest flux d'informació a les xarxes socials de cara a una empresa són els Community Managers. Aquests responsables actuen com a gestors i auditors de les xarxes socials, i que en aquest últim període està essent una professió emergent.

A grans trets podríem dividir les seves tasques en:

- Gestionar campanyes de promocions, publicitat i generar valor i bona imatge de la marca o empresa que representen.
- Auditar, generar informes i estadístiques importants per als altres departaments: Com ha reaccionat la comunitat davant d'un aspecte rellevant o un succès? Quan ha augmentat la comunitat després de l'anunci publicitari a la TV?
- Gestionar el feedback que produeix la comunitat afiliada a les xarxes socials.

Aquesta última és la que importa per al projecte, ja que intentarà donar resposta i resoldre els requisits dels problemes en què es troben dia a dia aquests treballadors.

2. Descripció

SMAC és un compost amb les paraules "Social Media en l'atenció al client", donat que respon a grans trets el producte com serà i aporta la possibilitat de pronunciació necessària per ser una marca o producte en un futur.

2.1. Exemple de treball real

No hi ha millor manera de entendre les necessitats que ha de resoldre el software SMAC com conèixer com treballa actualment un Community Manager d'una prestigiosa agència de Social Media, la qual no pot ser nombrada per privacitat i protecció.

El Community Manager de l'agència (a partir d'ara, CM) s'encarrega d'auditar diàriament tota la retroalimentació que han generat l'usuari cap a un compte concret. Un compte per un CM pot representar una empresa, o si aquesta és molt gran, una marca.

Definiré pas a pas, quin és el procediment seguit en l'actualitat alhora de resoldre i generar informes SAC (Servei d'atenció al client) en aquesta agència i bàsicament en totes les empreses que formaran part del futur mercat per aquesta aplicació.

2.1.1. Alimentació de dades

Primer de tot els CM s'encarreguen d'importar dades de Facebook, Twitter (i d'altres xarxes socials) a unes fulles d'Excel. En aquest moment han d'omplir un seguit de columnes d'informació:

- a. Data d'entrada: moment en què s'insereix el missatge
- b. Plataforma: xarxa social de procedència (Twitter, Facebook, Tuenti, etc..)
- c. Usuari: Nom de la persona o nick que té a la xarxa
- d. Tipologia: pot ser en aquest cas una queixa, demanda d'informació, elogi, o corporatiu
- e. Categoria i subcategoria: es una associació de temàtica del missatge, molt interessant per generar estadístiques. Per exemple si seguim amb l'exemple del supermercat podria ser: Web, Cupons descompte, Empleats, Productes. Com a subcategoria de productes: Fruiteria, Carnisseria, Begudes, etc.
- f. Resolució: en cas que sigui una demanda d'informació o incidència, si ha pogut ser resolta amb autonomia del CM o si ha hagut de recorre a preguntar al Centre d'Atenció al Client.
- g. Satisfacció: si han resolt la incidència, han rebut feedback del usuari? Pot ser "positiu", "negatiu", o "no respon"
- h. Eficiència: efectivitat en la resolució d'incidències. En aquest cas pot ser 1 si s'ha resolt la incidència en una iteració de pregunta-resposta o 2 si han seguit dues o més iteracions i comunicacions amb l'usuari per respondre.
- i. Estat: en quin estat es troba la incidència. Per exemple: oberta, tancada, en progrés, reoberta, etc.
- j. Data de tancament de la incidència.
- k. SAC: es marca si l'usuari precisa una trucada del Centre d'Atenció al Client oficial.
- l. Per últim tenim tota la informació i cos del missatge.

Aquesta tasca es fa com a mínim un cop al dia, però s'acostuma a fer durant tot el dia per donar la màxima atenció i rapidesa possible en les queixes.

2.1.2. Categorització i desambiguació

Els CM un cop reben tota la informació procedeixen a dividir aquestes capsules d'informació. Suposant que només es dediquen a la resolució d'incidències i atenció al client, separaran aquelles mencions d'usuaris que facin elogis al compte, o siguin missatges corporatius o dels mitjans de comunicació. Aquesta separació es fa segons la tipologia explicada en el punt 2.1.1.

2.1.3. Resolució d'incidències.

L'acció anterior ens deixa només amb aquelles incidències que cal resoldre. El CM ha estat format per a resoldre aquells problemes i dubtes que siguin comuns i no calgui cap tipus d'informació privilegiada. Normalment el client s'encarrega de donar aquesta formació per evitar-se els costos de l'Atenció al Client tradicional.

Exemple:

Caprabo ha contractar a l'agència SMAC perquè un grup de 2 Community Managers els hi portin el compte de Twitter de CapraboOnline. Aquests CM han sigut formats per resoldre els típics dubtes que es poden trobat en la tenda on-line d'aquest supermercat: pagaments amb targeta, quan arriba la comanda, que fan si han perdut el usuari o la clau de pas, etc.

Es possible que si una queixa d'un usuari es referent a un cobrament indegut a la seva targeta de crèdit, els CM han de contactar amb el departament d'Atenció al Client de Caprabo ja que ells tindran accés a dades confidencials de pagament.

Sigui el camí que sigui, que s'hagi emprat per resoldre la incidència, s'ha de marcar el seu estat de resolució i les dates. També serà important indicar el nivell de satisfacció i retroalimentació de l'usuari, així com l'eficiència i els recursos emprats per al CM per atendre la petició.

2.1.4. Anàlisi

Finalment, a final de mes, amb totes aquestes dades del Excel, es generen estadístiques que serviran per analitzar i treure'n conclusions. També, i amb gran importància, serà donar al client els resultats obtinguts.

Tota aquesta informació analitzada pot ser molt interessant en moltes àrees, per exemple per reduir costos segons sigui més barat l'Atenció al Client habitual, o la gestió i auditoria amb CM. També pot ser important per un departament de màrqueting, per veure com ha sigut rebuda una promoció o novetat.



II-lustració 1. Cas real de SAC

3. Objectius

Els objectius d'aquest projecte s'han d'entendre com un projecte real: el client ens planteja i ens introdueix la seva manera de treballar, i ara és el moment com a empresa consulta desgranar els objectius generals com específics dels quals es podran extreure més endavant un seguit de requisits i pautes reals per a desenvolupar.

3.1. Generals

Els objectius generals definiran les grans pautes a seguir dins del procés.

- a) Facilitat la lectura i guardat de dades de les consultes i missatges d'atenció al client que provinguin de les xarxes socials i d'aquesta manera eliminar la inserció manual i la possibilitat d'errada humana.
- b) Millorar la gestió d'incidències, tant per la banda dels Community Managers alhora de categoritzar-les i gestionar-les, com per la part de l'Atenció al Client, per poder accedir a aquelles on sigui requerida la seva ajuda i poder establir una comunicació i interacció amb els CM.
- c) Obtenir gràfiques i anàlisis, així com dades rellevants, dels resultats generats en tot el procés de les consultes.

3.2. Específics

Per altra banda aquí definirem aquells objectius més específics que ajudaran a arribar al punt cap on ens dirigim.

- a) Obtenir informació addicional dels usuaris que realitzen consultes a través de les APIs de *Twitter* i *Facebook* (biografia, gustos, origen, etc.)
- b) Visualitzar en una base de dades, en temps real, l'estat de les consultes, així com tot el procés seguit des de el principi de la incidència.
- c) Permetre la categorització de les consultes a través de categories, subcategories a elecció de l'usuari del producte (per exemple, tipologia, categoria i subcategoria)
- d) Poder realitzar cerques dins de la base de dades, des de diferents departaments d'una mateixa empresa o des de client-agència.
- e) Visualització, de forma ràpida i senzilla, l'estat de les diferents consultes dels usuaris.
- f) Permetre l'assignació d'interlocutors a les diverses fitxes d'incidència de les consultes. Per exemple, si la consulta està relacionada amb el departament corporatiu, assignar-la al responsable de l'àrea.
- g) Generar gràfics de resultats en funció de les categories escollides. Per exemple: barrejant tipologies amb subcategories en temps real.

4. Continguts

Fins ara hem definit primerament la problemàtica que pretenem resoldre, hem analitzat les necessitats d'una branca determinada d'una agència de publicitat dedicada al Social Media.

D'aquí hem extret els requisits que haurà de complir l'aplicació i hem analitzat els objectius primaris i secundaris que ha de donar resposta SMAC.

En aquest punt caldrà aportar la documentació conceptual prèvia a l'anàlisi més tècnic i serà l'última frontera abans d'endinsar-se en el mont del programari.

4.1. De quines eines disposa l'aplicació?

Abans de descriure les funcions de l'aplicació és interessant fer una agrupació prèvia: els rols. Cada rol de l'aplicació disposarà d'un seguit de funcions, i aquest rols a més a més son compatibles.

SMAC disposarà dels següents rols:

Usuari anònim: aquest perfil serà el referit aquells usuaris externs a l'aplicació o que encara no han iniciat sessió. Aquest usuari només disposarà d'una funcionalitat que serà la de "Login", és a dir, autenticar-se a l'aplicació

Usuari registrat: aquest perfil és el bàsic que tindran tots els usuaris de l'aplicació. Només en aquest rol disposaràs de funcions per fer "Logout", és a dir, deixar l'aplicació i la funció de modificació de dades personals, on podràs canviar el teu nom, la teva descripció així com l'avatar (la fotografia que identificarà el teu usuari).

Usuari gestor: aquest usuari serà el cap del projecte, el gestor, el responsable de l'àrea que serà capaç de crear un projecte, gestionar el seu accés a les API's de *Twitter* i *Facebook* i finalment podrà adjuntar cada usuari de l'aplicació al seu projecte.

Aquest rol és important ja que des d'aquí es planificarà qui i com treballarà en cada projecte. Per projecte podem entendre una empresa, o si és una agència que porta varies marques o productes, es podrien crear varis projectes.

Usuari analista: aquest usuari normalment serà compartit amb altres rols, i és el que tindrà permisos d'accés a les estadístiques i a la generació d'analítiques i informes. Donat que l'enfoc de SMAC és per empreses petites i mitjanes, normalment no tindran un departament tan gran, i aquest rol serà compartit amb d'altres, però en un cas més complex l'aplicació seria capaç de donar-hi solució.

Usuari CM: aquest serà el rol del Community Manager, que s'encarregarà de classificar els *tweets* o missatges de *Facebook* entrants, podrà decidir si son elogis, queixes o sol·licituds d'ajuda o d'informació. També podrà marcar la seva categoria i tipologia a motius de posterior anàlisi i resum.

També tindrà la funció de generar tickets. Un ticket considerem que és una tasca que assignarà un Community Manager a partir d'un missatge que necessiti una resolució. Aquesta resolució pot ser que un usuari hagi fet una petició d'ajuda, preguntat una qüestió o simplement que demani que es resolgui alguna incidència.

Alguns tickets podran ser resolts pels mateixos Community Manager, en canvi n'hi haurà altres, que per uns motius concrets, no pugui ser el CM sinó una altra persona de l'aplicació que hagi de participar, per tant es podrà assignar a qualsevol usuari involucrat en el projecte.

Usuari AC: entenem com al departament d'atenció al client. Aquest rol serà normalment optatiu, i es donarà en el cas que l'empresa que faci servir l'aplicació tingui distinció entre els CM i el departament de atenció al client. En aquest cas, el rol de AC permetrà només veure tickets, de manera que la seva funció serà assistir al CM en la resolució d'aquests.

4.2. Exemple d'escenari d'ús de l'aplicació web

Per tal de veure més clar tots aquests termes i entendre millor la filosofia de l'aplicació, a continuació exposaré un cas imaginari però amb components que es podrien arribar a donar en un cas real.

Exemple:

L'agència de publicitat "JML" ha obert un nou departament de xarxes socials que pretén donar servei al seus clients per tal de gestionar la comunitat de Twitter i Facebook.

Un dels clients més grans de l'agència, ECI (una empresa que es dedica a la venda de productes cosmètics online) decideix ampliar el seu negoci a internet i crear unes contes de xarxes socials per a fer promocions, concursos i informar als usuaris de ofertes i descomptes. A la vegada els usuaris tenen un canal directe de comunicació amb ECI, que aprofiten per exposar les seves queixes, formular preguntes, bàsicament fer-ho servir com un canal d'atenció al client més ràpid que la típica trucada telefònica.

D'aquí neix la necessitat per part de l'agència que gestiona les xarxes socials (JML) d'adquirir un software capaç de gestionar aquestes peticions d'atenció al client a través de les xarxes socials: l'aplicació web SMAC.

El departament encarregat de ECI consta d'aquests recursos:

- *Pep García: el CEO de JML, màxim responsable de l'agència.*
- *Albert Pont: el responsable del projecte i del client d'ECI.*
- *Marta López: una Community Manager assignada a ECI.*
- *Alba Grau: una altra CM assignada a temps parcial al projecte*
- *Departament d'atenció al client: ECI disposa del seu propi departament d'atenció al client que disposa de les seves pròpies eines i accés a informació dels usuaris.*

Els tècnics de SMAC (la nostra aplicació) aconsellen a JML les següents pautes per a començar a treballar.

Exemple:

1. *Crear un compte a Albert Pont amb els següents rols: Usuari registrat, Usuari gestor.*
2. *Crear un compte a Pep García: Usuari registrat, Usuari analista*

3. *Crear un compte a Marta López i Alba Grau: Usuari registrat, Usuari CM*
4. *Crear un compte al departament de atenció al client d'ECI: Usuari registrat, Usuari AC.*
5. *Albert Pont crea el Projecte ECI*
 - 5.2. *Gestiona les dades del projecte i configurar les API's de les xarxes socials*
 - 5.3. *Adjunta tots els demés usuaris esmentats al projecte ECI creat al pas 5.*

Un cop en aquest punt tots els actors ja estan capacitats per començar a treballar amb SMAC. En el següent exemple detallarem el que podria ser una rutina d'un CM.

Exemple:

L'Alba Grau comença la seva jornada laboral i decideix entrar a SMAC:

1. *Fa login amb el seu usuari i clau.*
2. *Selecciona el projecte ECI (com que esta a temps parcial està adjuntada a altres projectes)*
3. *Accedeix a la secció de missatges i li apareixen 5 piulades de Twitter i procedeix a classificar-les.*
4. *Totes son elogis dels productes d'ECI però un dels missatges és un usuari que es queixa que no ha rebut una comanda que va fer fa una setmana. Sembla molt enfadat.*
5. *L'Alba marca la tipologia del missatge com a "Queixa", la categoria "Compra online" i la subcategoria "Perfum de dona", també marca la casella de "generar ticket".*
6. *Accedeix a continuació a la secció "Tickets" i escriu un comentari: "Sembla molt furios aquest usuari, hauríem de resoldre-ho ASAP", marca el ticket com a "Molt prioritari" i ho assigna al usuari de Atenció al Client de ECI.*

Seguint amb els exemples, suposarem que entra en escena un tècnic d'atenció al client de la empresa ECI que té accés al compte de SMAC.

Exemple:

1. *El tècnic es connecta a SMAC*
2. *Entra a "Tickets" la única secció on té accés.*
3. *Llegeix el ticket que li ha assignat l'Alba, donat que té accés a les dades de les comandes, busca la informació de l'usuari i veu que no hi ha cap problema simplement que està tardant més de lo pensat per una vaga de transportistes.*
4. *Actualitza el ticket "Alba, diga-li a l'usuari que tot està correcte, donat el retràs li pots oferir un codi (ZDF56) que pot canviar per un descompte d'un 20% en la següent compra". Assigna el ticket novament a l'Alba i marca l'estat com a "Resolt".*

A final de més el CEO de l'empresa JML decideix veure com va el projecte ECI:

Exemple:

1. *Pep García es connecta a SMAC*
2. *Accedeix a la secció "Informes" i selecciona el projecte ECI i el mes en curs.*

3. *Visualitza el nombre de resolucions de tickets, el grau de satisfacció de l'usuari i el temps que han trigat de mitja en contestar les peticions dels usuaris.*
4. *Veu la gràfica comparativa dels mesos anteriors d'elogis arribats a Twitter i Facebook.*
5. *Amb aquestes dades analitza l'impacte de la campanya publicitària que van llençar el mes passat i veu que realment s'ha incrementat el nombre d'elogis i d'usuaris que es connecten mitjançant les xarxes socials*
6. *Tot un èxit a la companyia, ara cal preparar un informe per la reunió de la setmana que ve.*

5. Metodologia

La metodologia en el desenvolupament de software es considera un marc de treball per a estructurar, planificar i controlar el procés de desenvolupament del producte. Existeixen moltes filosofies o prototip de metodologies que es poden adoptar i en aquest cas, donat els requeriments temporals i l'ordre lògic s'ha optat pel model en cascada o *waterfall*.

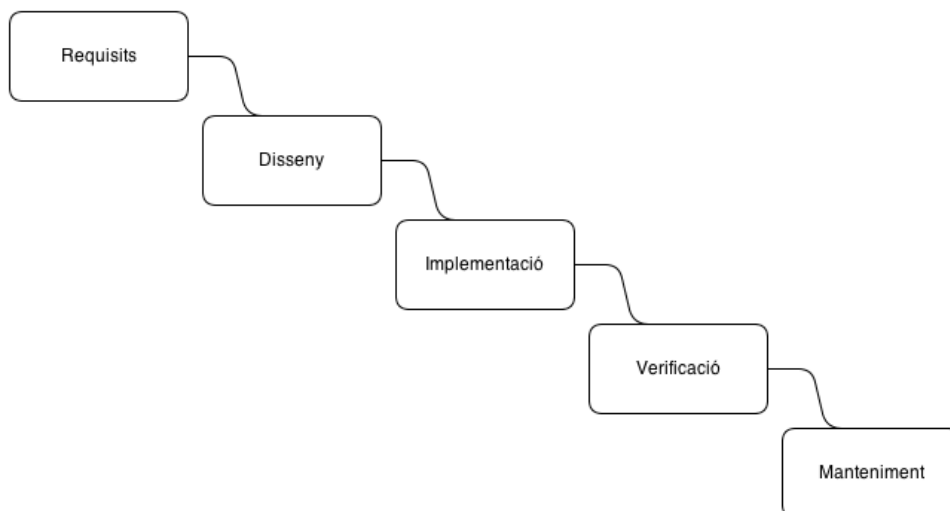
Aquest model és un procés seqüencial de desenvolupament en el que les tasques son vistes cap a baix (com en una cascada d'aigua, d'aquí el seu nom) a través de les fases d'anàlisi de les necessitats, el disseny, la implementació, les proves (validació), la integració i el manteniment.

Aquesta metodologia segueix uns principis basics:

- El projecte esta dividit en fases seqüencials, amb certa superposició i *splashback*¹ entre les fases.
- Es dona una especial importància a la planificació, els horaris i les dates, així com als pressupostos i execució de tot el sistema una sola vegada.
- Es manté un estricte control durant tota la vida del projecte a través de la utilització d'una amplia documentació escrita, així com a través de comentaris i aprovacions per part de l'usuari. De la mateixa manera que la tecnologia utilitzada alhora de planificar les fases.

Vist això, es pot observar com aquesta metodologia encaixa molt bé amb les necessitats que exposa el model de Treball de Fi de Grau en el qual ha d'encaixar.

És també important adonar-se'n que aquestes pautes entenen perfectament el desenvolupament i planificació a partir d'un sol recurs durant tot el procés, i que els consultors actuaran com el client final i a la vegada com a cap que validarà tota la feina feta per a seguir avançant entre les fases.



II-lustració 2. Mètode cascada

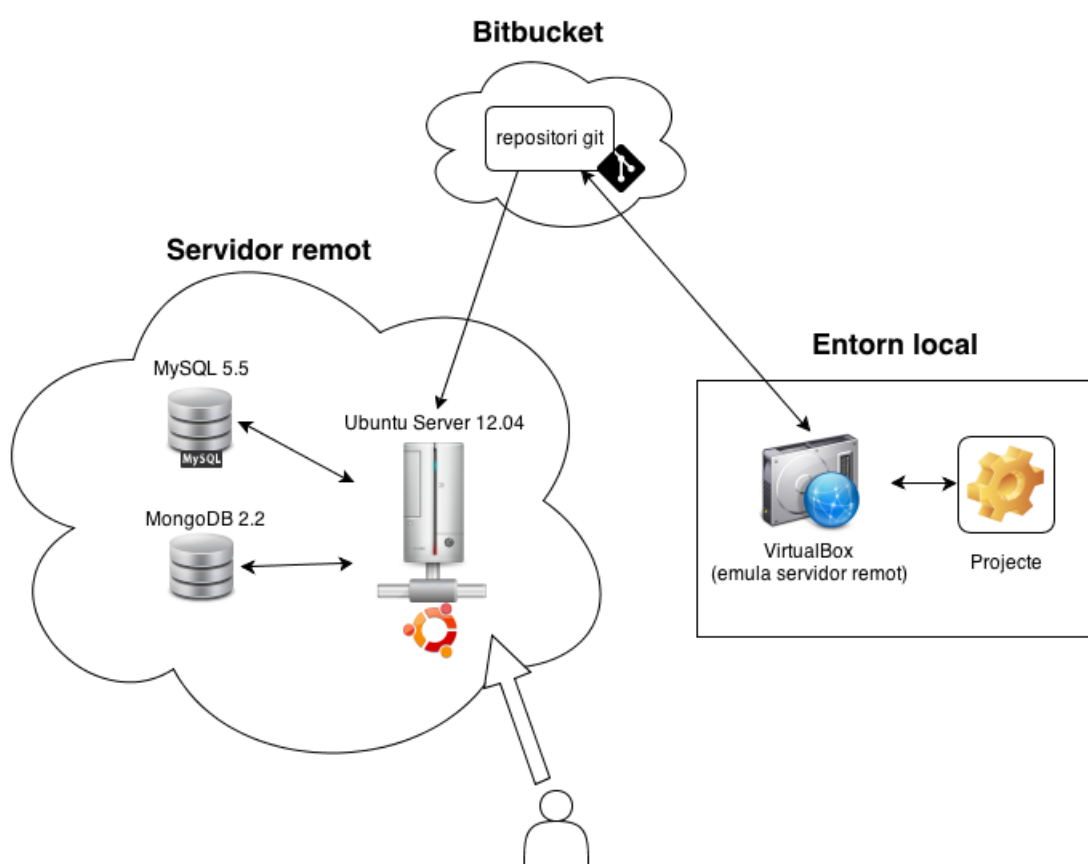
¹ Quan parlem de *splashback* ens referim que una fase pot esquitxar a una altra, en el sentit que afecta en algunes parts del desenvolupament d'aquesta.

6. Arquitectura de l'aplicació

Definir l'arquitectura web és un procés en el qual s'ha de crear l'entorn sota el qual l'aplicació funcionarà i donarà servei a tots els usuaris. Apart del simple fet que l'aplicació funcioni correctament, ha de ser un entorn controlat pel treball dels desenvolupadors i en un futur ha de permetre l'escalabilitat, és a dir creixement del hardware si la demanda dels usuaris així ho requereix.

6.1. Estructura bàsica de desenvolupament

Es mostra a continuació l'esquema de l'estructura inicial del servidor remot i com interacciona amb els altres elements importants en tot el cicle de desenvolupament i manteniment.



II-Il·lustració 3. Arquitectura web

La part de l'entorn local és on treballaran els desenvolupadors, que serà ampliada a la secció "7. Plataforma de desenvolupament".

El servidor remot serà on la nostra aplicació estarà en producció serà utilitzada per als usuaris i clients finals, per tant serà un servei que estarà en unes màquines dedicades específicament per a aquest fet i estaran connectades directament a Internet.

Aquest servidor remot, serà un ordinador dedicat exclusivament a l'aplicació web SMAC i estarà configurat sota la distribució de *Ubuntu Server 12.04*. S'ha triat aquesta distribució de Linux en concret per un seguit de motius:

- LTS (Long Term Support): *Ubuntu* es compromet amb els seus usuaris d'atorgar manteniment i suport del seu software durant els 5 anys posteriors de la seva sortida
- Gran comunitat darrere. *Ubuntu* disposa d'una gran comunitat darrere, de manera que es molt senzill trobar documentació, actualitzacions i bons programes.
- *Ubuntu* té una empresa darrere, pel que el seu creixement va vinculat a uns canons empresarials i de seriositat imperativa. La distribució *Debian* per exemple està feta per usuaris voluntaris, i tot i ser una de les més estables, sempre va endarrerida en quan a les versions de molts programes.

Aquest servidor disposarà també d'un seguit de mòduls per a que pugui funcionar la nostra aplicació:

- *Apache*, el software que permet utilitzar la màquina per a servir webs als usuaris.
- PHP, el llenguatge de codi interpretat sota el qual estarà desenvolupada gran part de l'aplicació i òbviament es el llenguatge utilitzat per *Symfony 2*.
- GIT, es un sistema de control de versions. En aquest servidor només s'utilitzarà per agafar el codi destinat a producció del servidor de repositori.

Cadascun d'aquests mòduls estarà configurat amb un seguit de llibreries i paràmetres de manera que funcioni de manera òptima i sigui capaç de comunicar-se amb els altres servidors i tecnologia.

Per altra banda disposarem d'una base de dades *MySQL*. Aquesta base de dades es del tipus relacional i servirà per emmagatzemar totes les dades dinàmiques de l'aplicació.

També tindrem instal·lat una base de dades *MongoDB*², que es del tipus documental. Aquest tipus de base de dades ens aporta un gran rendiment per tractar grans volums de dades heterogènies.

Per altra banda, tindrem una compta creada als servidors de *Bitbucket*³. Aquests servidors disposen de manera gratuïta de repositori de GIT privats per a poder treballar. En aquest repositori tindrem tot el historial de desenvolupament de codi de l'aplicació, de manera que ens suposarà una font de seguretat coma copia de seguretat a la xarxa, així com un manteniment del repositori totalment controlat.

² *MongoDB* (de "*humongous*"), es una base de dades documental i de codi obert, que es una de les més importants en el món de les bases de dades no relacionals.

³ *Bitbucket* forma part de la xarxa de productes que ofereix l'empresa *Atlassian* que dona suport en totes les fases de desenvolupament d'una aplicació.

6.2. Comunicació

La comunicació entre els servidors serà configurada de manera que contempli la màxima seguretat en totes les bandes i restringir al mínim la interacció entre totes les parts afectades: servidors, clients i desenvolupadors.

Seguint el diagrama anteriorment mencionat, el usuari accedirà al servidor remot per consultar i treballar amb la aplicació web, i aquesta serà la única manera d'accés des de l'exterior. Les bases de dades tindran restringit el seu ús de manera que només des de l'entorn local (el mateix servidor remot) tingui accés per la seva consulta, donat que sinó fos d'aquesta manera, suposaria un forat de seguretat important, tot i està protegit amb *password*.

El servidor remot tindrà accés al repositori, però només del tipus "lectura", ja que en el moment que hi hagi nou codi per ser atès a producció, podrà descarregar-se els canvis i nous fitxers i incorporar-los immediatament a la versió funcionant que els usuaris consulten. Aquest procés no hauria de ser superior als 5-10 minuts, de manera que es minimitza la finestra de temps en la qual l'aplicació deixa de ser funcional.

6.3. Escalabilitat

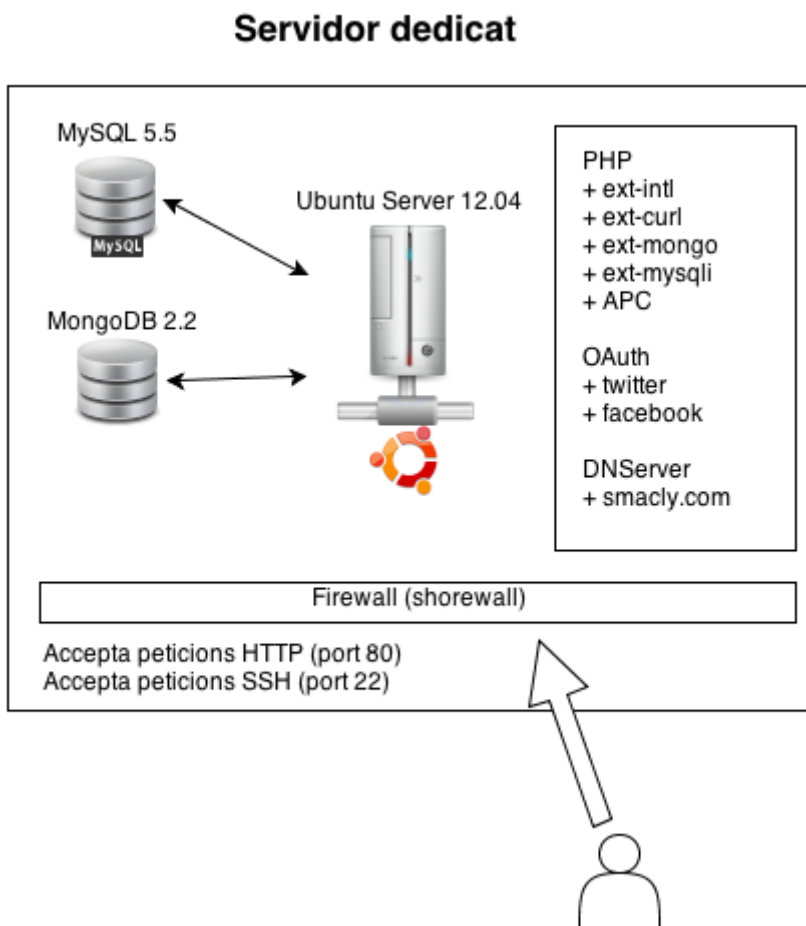
Tot i que l'aplicació SMAC és només un prototip del que podria ser en un futur, i el volum de dades i d'usuaris concurrents es mínima no ha calgut pensar en un sistema escalable que suporti moltes visites i volums de dades molt grans.

De totes maneres, part de l'anàlisi tecnològic que s'ha portat a terme, ha sigut pensar en maneres d'ampliar el hardware en cas que apareguessin els anomenats "colls d'ampolla", és a dir, en el moment que una part dels components del servidor provoqui un límit alhora de servir l'aplicació a tots els usuaris a la vegada.

- **Sistema de cues:** en el moment que alguns processos suposin temps i recursos molt costosos per les màquines, es podria instaurar un procés d'encuat, on es prioritzeu alguns processos, de manera que mai es pogués provocar un col·lapse del sistema per concurrència de processos llençats per l'usuari.
- **Desacoblament:** el *frontend* (la part visual de l'eina) i el *backend* (la part de gestió i processos) no han d'estar unides, sinó que mitjançant serveis i *API's* es pot arribar a desacoblar de manera que l'impacte que sofreixi un, no afecti a l'altre. Si els processos estan descentralitzats a altres màquines, pot arribar a ser molt senzill escalar el hardware d'aquestes màquines per atendre un gran volum de peticions.
- **MongoDB:** aquesta base de dades està pensada per un gran volum de dades i ja ve preparada per ser estesa i clonada entre altres servidors per una millora de la disponibilitat i concurrència.

6.4. Escenari final

Després de treballar amb l'arquitectura a nivell de desenvolupament durant el transcurs d'aquests, cal veure com quedaria finalment l'entorn de producció final:



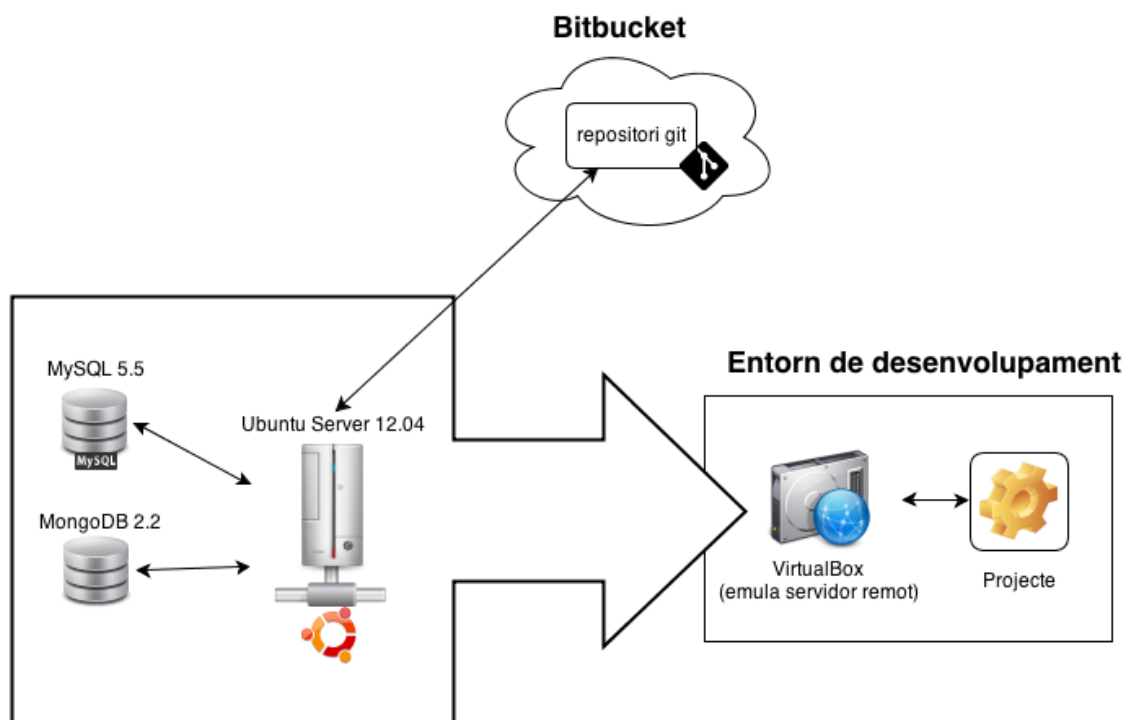
Il·lustració 4. Arquitectura final

Alhora de tenir un servidor accessible de cara al públic cal tenir en compte certs paràmetres a més a més dels requisits necessaris de l'aplicació que ja contemplem als apartats anteriors:

- **Extensions de PHP:** cal tenir instal·lades un seguit d'extensions de PHP que ens permeten connectar-nos amb el software de bases de dades que tenim instal·lat, connectar amb el sistema de cache APC que fa servir Symfony2, així com habilitar les llibreries necessàries per connectar-nos a serveis externs (com les API's de Twitter o Facebook) mitjançant cURL
- **Tallafocs:** disposem d'un Firewall que ens protegeix de peticions no esperades de l'exterior i limita l'accés només a peticions web (mitjançant HTTP) al port 80.
- **Servidor de DNS:** el servidor també actuarà en funcions de DNS ja que rebrà les peticions orientades des de el domini que s'ha adquirit per l'aplicació: smacly.com.
- **Llibreria OAuth:** necessària per autenticacions externs a les API's de Twitter i Facebook.

7. Plataforma de desenvolupament

L'entorn de desenvolupament d'aquesta aplicació web no es trivial, donada la complexitat d'aquesta, ha calgut realitzar un anàlisi tècnic de tecnologies i software necessari per crear tot l'entorn. En el punt 6 anterior s'ha detallat com seria l'arquitectura web des de el punt de vista de producció, i ara profunditzaré més en l'aspecte del desenvolupament, on els programadors crearan tot el codi.



Il·lustració 5. Entorn de treball

7.1. Màquina virtual

En el món de la informàtica i sobretot en el món de la programació moltes vegades, per no dir sempre, és important tenir en compte la compatibilitat del software en diferents sistemes i en cas de aplicacions webs també cal tenir en compte els navegadors així com la combinació de tots dos elements.

A part per raons de comoditat, quan un programador desenvolupa codi en un entorn local controlat ha d'assegurar que aquest producte que està generant, funcionarà igual que en el servidor on hi serà al final.

Aquests dos problemes exposats tenen una solució: les màquines virtuals.

Una màquina virtual es un software que simula una computadora en tots els sentits, en altres paraules, que dins del mateix ordinador amb un sistema operatiu concret, pot existir una emulació d'un o més màquines amb diferents sistemes operatius.

Una de les característiques més importants es que aquestes emulacions virtuals son abstractes i tenen els recursos limitats, donat que si fes servir el 100% dels recursos no serien capaços de conèixer.

En el nostre cas, la importància d'utilitzar una màquina virtual és que podem emular amb la màxima exactitud l'entorn real on, quan l'aplicació estigui acabada o presentable, hi figurarà en un futur pròxim.

Així evitem problemes de incompatibilitat d'alguns recursos entre Windows (l'entorn real del programador) i Linux (l'entorn de producció del servidor). Aquest fet es degut a que per motius de rendiment més del 90% dels servidors reals webs son Linux i per tant el software d'aquest sistema operatiu alhora de programar i executar codi ha viscut una evolució molt més constant.

7.2. Control de versions

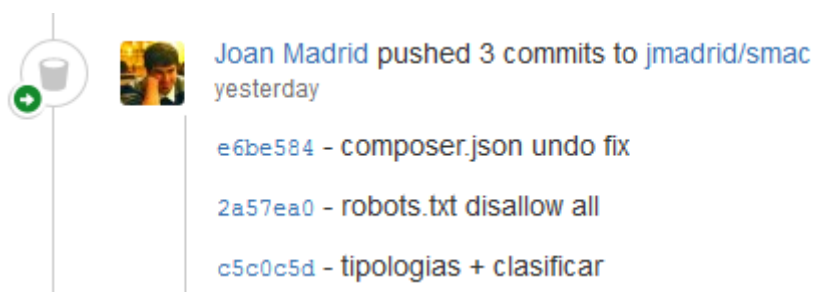
En tot projecte de desenvolupament de software mitjanament professional és necessari incorporar un sistema de control de versions.

Aquests softwares et permeten gestionar i guardar tots els canvis que s'han efectuat sobre el codi del projecte, d'aquesta manera es generen versions o revisions del projecte que son simplement estats del producte que s'està desenvolupant en un moment en concret de la seva producció (normalment en un moment estable).

Aquests sistemes ens aporten un seguit d'avantatges:

- **Historial:** donat que es guarden tots els canvis que s'han anat fent en el projecte sempre hi ha la possibilitat de recapitular, consultar o tirar cap enrere en cas de necessitat.
- **Copia de seguretat:** al final aquests sistemes funcionen com una copia de seguretat si ho tens descentralitzat de l'entorn local de treball.
- **Treball en equip:** és una de les característiques més importants, donat que quan sobre el mateix codi hi treballa més d'una persona és imprescindible que hi hagi un control de solapaments i un sistema d'integració i resolució de conflictes.

En el cas concret d'aquest projecte es farà servir la versió gratuïta de *Bitbucket*. Aquests web permet la creació de repositori privats i emmagatzematge de codi en els seus servidors mitjançant Git. Donat les necessitats inicials on només un programador interaccionarà amb el codi és més que suficient i fiable.



II-Ilustració 6. Captura de Bitbuket.org

7.3. Projecte

Al final el projecte no és més que un conjunt de fitxers de text que seran interpretats per un servidor i aquest mitjançant peticions HTTP generaran pàgines webs per l'usuari.

Alhora de desenvolupar el codi és normal ajudar-se dels anomenats *IDE* (plataformes integrades per a desenvolupament). Aquests software ens aporten moltes avantatges i ens permeten treballar en un entorn integrat d'eines que faciliten la feina al programador.

El nostre codi serà majoritàriament PHP i es per això que s'ha decidit utilitzar *PHPStorm*.

En el mercat actual hi ha molts IDE preparats per treballar amb PHP com podrien ser *Eclipse*, *Netbeans*, *Dreamweaver* i un llarg etcètera. Però fent una recerca per Internet he trobat aquest que està totalment enfocat al desenvolupament PHP i per això està molt més centrat en aquesta tasca i prescindeix de molts mòduls que no són necessaris.

En la creació de SMAC ens ajudarem d'aquest programa i en treure'm partit, entre d'altres coses en aquestes tecnologies:

- **Editor de codi intel·ligent:** *PHPStorm* interpreta el teu codi i l'entén de manera que es capaç de donar-te suport i ajudar-te alhora de relacionar funcions, donar-te informació del llenguatge PHP i avisar-te en cas d'errors.
- **Qualitat:** gràcies a eines integrades, aquest IDE porta un control de qualitat de codi, de manera que et suggereix bones practiques de programació i alhora t'avisa de possibles incompatibilitats. També dona suport a la documentació de codi seguint varis estàndards instaurats.
- **Entorn de desenvolupament:** aquest software es comunica directament amb GIT, Composer, PHP i altres eines necessàries, de manera que ja porta un entorn integrat per treballar conjuntament, sense necessitat d'anar intercanviant d'eines per cada acció requerida.
- **Editor de *HTML/CSS/Javascript*:** no cal oblidar que no només es programarà en PHP ja que al ser una aplicació web caldrà integrar el codi amb les vistes. Aquestes vistes estaran creades amb llenguatges *HTML*, estructurades i decorades amb *CSS* i la interacció vindrà de la mà del *Javascript*.

8. Planificació

Una bona planificació és una tasca imperativa que cal abordar des de el primer moment del projecte per no perdre el rumb i aconseguir un cicle de creació i desenvolupament robust i sense desviacions.

En un cas real com aquest, aquesta planificació sofrirà variacions per causes externes, i això serà tasca del Cap de Projecte de redirigir esforços o refer aquesta planificació per a que encaixi en el marc temporal.

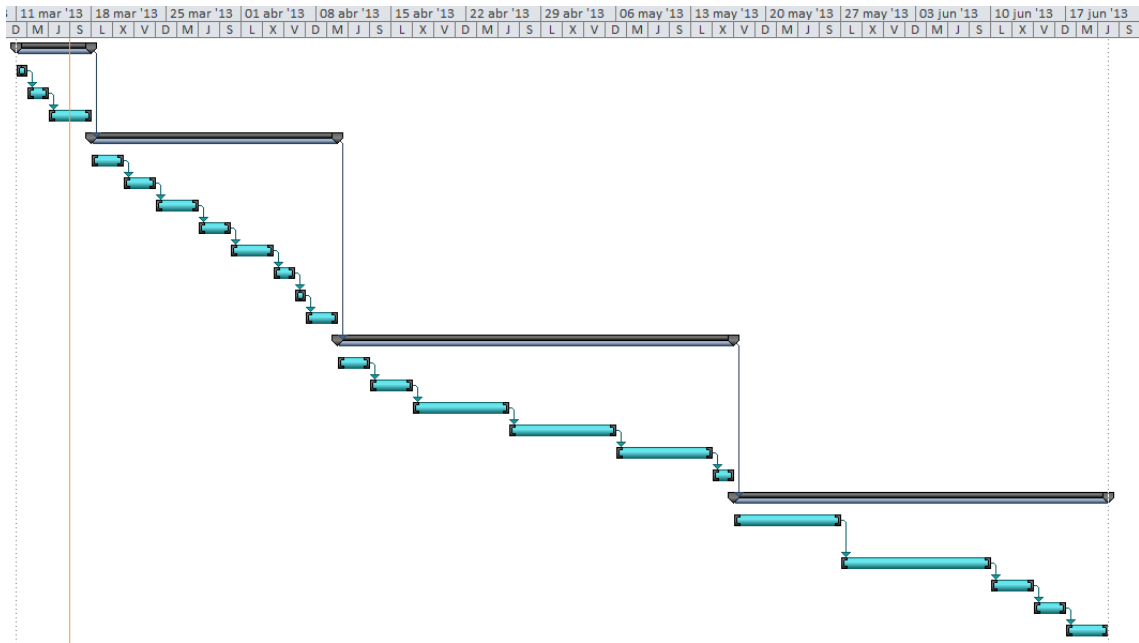
9.1. Planificació inicial

El següent llistat exposa les tasques per la creació d'un diagrama de Gantt, des de la data d'inici del projecte (11 de març) fins a la data d'entrega (20 de juny).

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
Planificació	7 días	lun 11/03/13	dom 17/03/13	
Recull d'informació	1 día	lun 11/03/13	lun 11/03/13	
Definició del projecte	2 días	mar 12/03/13	mié 13/03/13	2
Documentació PAC1	4 días	jue 14/03/13	dom 17/03/13	3
Definició	23 días	lun 18/03/13	mar 09/04/13	1
Requisits	3 días	lun 18/03/13	mié 20/03/13	
Arquitectura web	3 días	jue 21/03/13	sáb 23/03/13	6
Wireframes	4 días	dom 24/03/13	mié 27/03/13	7
Disseny	3 días	jue 28/03/13	sáb 30/03/13	8
Diagrama de classes	4 días	dom 31/03/13	mié 03/04/13	9
Esudi de tecnologies	2 días	jue 04/04/13	vie 05/04/13	10
Bases de dades	1 día	sáb 06/04/13	sáb 06/04/13	11
Documentació PAC2	3 días	dom 07/04/13	mar 09/04/13	12
Desenvolupament	37 días	mié 10/04/13	jue 16/05/13	5
Preparació workspace	3 días	mié 10/04/13	vie 12/04/13	
Preparació servidor	4 días	sáb 13/04/13	mar 16/04/13	15
Estructura MVC	9 días	mié 17/04/13	jue 25/04/13	16
Controladors	10 días	vie 26/04/13	dom 05/05/13	17
Vistes	9 días	lun 06/05/13	mar 14/05/13	18
Documentació PAC3	2 días	mié 15/05/13	jue 16/05/13	19
Finalització	35 días	vie 17/05/13	jue 20/06/13	14
Desenvolupament de persistència	10 días	vie 17/05/13	dom 26/05/13	
Programació de servei	14 días	lun 27/05/13	dom 09/06/13	22
Bugfixing	4 días	lun 10/06/13	jue 13/06/13	23
Desplegament	3 días	vie 14/06/13	dom 16/06/13	24
Finalització de memòr	4 días	lun 17/06/13	jue 20/06/13	25

II-lustració 7. Planificació de tasques

I a continuació exposo el Diagrama de Gantt generat amb les seves dependències.



Il·lustració 8. Diagrama de Gantt

Es pot observar que s'ha dividit el projecte en 4 grans tasques, que coincideixen amb la majoria de tasques naturals d'un projecte de desenvolupament de software.

En la **planificació**, s'emula el tracte amb el client, les reunions inicials per arribar a un consens i obtenir un anàlisi de requisits clars. És la part on el projecte començarà a agafar un rumb molt concret.

La **definició** és la part més formal on es plasmaran les idees i les pautes en esquemes de requisits i diagrames. És en aquest moment on el departament tècnic s'incorporarà i entendre el problema a resoldrà, de manera que es pugui començar a plantejar una solució d'acord amb el que s'ha demanat.

L'etapa de **desenvolupament** serà aquella on es comença a crear el producte seguint totes les pautes de la planificació i la definició del projecte. En aquest moment el programa començarà a cobrar vida i al llarg de les entregues es podrà arribar a una fase Beta.

La **finalització** serà la conclusió del treball realitzat. S'acabaran de tancar temes del desenvolupament i es passarà a provar i analitzar el producte per a obtenir la màxima qualitat abans de ser entregat.

9. Procés de desenvolupament

El procés de desenvolupament va molt lligat a la planificació, ja que es imprescindible ser conscient en tot moment en quin punt del projecte ens trobem i fer el possible per seguir els pronòstics alhora que som responsables de la feina a la que ens hem compromesos.

9.1. Anàlisi previ

L'elaboració d'aquest document juntament amb els entregables de les PAC formen part d'un anàlisi previ i necessari, ja que abans de picar una línia de codi cal tenir molt clar molts dels aspectes que s'estan tractant del tipus tècnic però també conceptual.

En el nostre cas treballem com si es tractés d'un projecte real on hi ha un client que ens demana un seguit de requisits i que primerament els analistes funcionals intenten desxifrar i posar en ordre de manera que l'equip tècnic sigui capaç d'interpretar-ho i fer-ho realitat.

9.2. Anàlisi tècnic

Un cop obtingut tots els requeriments, és hora que els analistes tècnics es posin a treballar amb els diagrames que representaran d'una manera comprensible als desenvolupadors el que el client en un principi ha especificat de manera caòtica i amb les seves pròpies paraules.

Aquests diagrames de classes i de casos d'ús ja donen moltíssima informació i es gairebé un punt on els programadors podrien començar a plantejar-se com resoldre el problema i per on atacar-lo.

9.3. Disseny

En aquest punt entra el disseny conceptual i de pantalles. Cal preparar *wireframes* i *mockups* per acabar d'afinar i posar cara el anàlisi tècnic previ. Aquest punt no només és per acabar d'especificar el contingut de l'aplicació, sinó que es molt útil alhora de detectar errors en la definició, ja que és en aquest punt en el que es comencen a veure les coses clares.

També caldrà començar a pensar en el disseny final de la pàgina web: colors, formes, tipografies i textos. Donat que en una futura versió s'hi incorporaria un estudi de disseny d'interfícies per a fer una versió millorada, bé caldrà sortir a producció amb un prototip agradable i usable.

9.4. Desenvolupament

La part més llarga i difícil. Cal codificar i fer que funcioni tots els requisits i indicacions que han estat aprovades fins aleshores. Tot ha de funcionar a la perfecció com s'ha demanat i s'han de minimitzar els errors i *bugs*.

9.5. Bugfixing

Un cop s'entregui una versió estable cal que agents externs als desenvolupadors, ja que aquests poden estar viciats de l'aplicació, facin proves sobre la web final i detectin tant errors com inconsistències conceptuals.

Moltes vegades aquest punt també és pot considerar un estudi de la usabilitat i la interacció i moltes vegades es poden detallar millores i possibilitats noves a l'aplicació.

9.6. Finalització

Un cop arreglat els errors i acabats els últims detalls cal posar en marxa l'aplicació. S'ha de muntat tot a un servidor real i remot, de manera que els usuaris reals puguin accedir i puguin treballar amb ella.

En aquest punt es on comença una etapa de manteniment i millora per als desenvolupadors, que vetllaran per arreglar tots els errors i aplicar les millores que els analistes i els clients creguin adient.

També aquí comença tot un nou camí per al *marketing*, la publicitat i el departament de vendes. En aquest punt ja poden començar a pensar en negoci i en com accedir a nous mercats i fer arribar l'aplicació al màxim de gent, a la vegada que en poden treure un benefici substancial.

10. Frameworks, APIs i llibreries utilitzades

10.1. Symfony 2

Alhora de desenvolupar un producte web en PHP professionalment normalment no es pot pensar sense un *framework* darrere. Symfony i la seva versió 2 és un dels més avançats i complets. Entre tots els competidors aquest destaca per seguir unes bones practiques diferenciades, una comunitat darrere molt gran i una gran filosofia que ha fet que darrerament la elecció d'un bon *framework* hagi quedat limitada a un parell d'opcions.

Gracies a les llibreries que ja porta inclusives al nucli, ens facilita la feina de connexió amb bases de dades, al enviament de *emails* i a seguir el patró MVC (model – vista – controlador) amb un gran resultat.

A part de les facilitats alhora de desenvolupar també ens ofereix un gran rendiment amb els seus sistemes de cache, que a la vegada son totalment extensibles i abstractes per treballar amb qualsevol altre sistema.

Finalment, cal esmentar, que gracies a la gran comunitat que té darrere, podem trobar una gran quantitat (i de gran qualitat) de *Bundles* o paquets de codi amb noves funcionalitats, totalment gratuïtes i llestes per ser utilitzades.

10.2. jQuery

Treballar amb *Javascript* és imprescindible avui en dia en qualsevol aplicació web: interacció, dinamisme i AJAX son algunes de les necessitats que ens resol el *Javascript*. Però més enllà de voler reinventar la roda, ja existeixen llibreries amb una comunitat gegant darrere i un gran equip de desenvolupadors que ens donen les eines necessàries per no haver-nos de preocupar de compatibilitat entre navegadors, en elaborar funcions comunament utilitzades o tenir la possibilitat de fer servir *plugin* i complements que altra gent ha preparat per la comunitat.

Una de les llibreries mes avançades i utilitzades es *jQuery* que destaca per les següents característiques:

- **Compatibilitat entre navegadors:** no ens hem de preocupar que funcioni en totes les versions i sistemes operatius, això ja se'n encarreguen els seus creadors.
- **Motor molt potent de selecció del DOM:** per moure'ns a través dels elements HTML cal un seguit de regles per poder orientar els scripts.
- **AJAX:** *jQuery* ja porta incorporat un motor per a fer peticions AJAX asíncrones.
- Gran varietat de **plugins**
- **Efectes i animacions** per als nostres elements HTML
- **Control d'esdeveniments** del client (el usuari i el seu navegador)

10.3. Bootstrap

De la mateixa manera que ens basem en un bastiment de PHP per treballar amb el llenguatge de servidor i aprofitem la potencia de *jQuery* per al *Javascript* a nivell de client, també farem ús d'un *framework* de CSS.

Bootstrap es un projecte de l'empresa de Twitter que es tracta d'una col·lecció de eines per a crear pàgines webs i aplicacions web, conté HTML i CSS en forma de plantilles i pautes per a generar fàcilment estructures webs. Entre els seus elements destaquem els formularis, botons, menús de navegació i altres elements d'interfície que es fan servir normalment com icones.

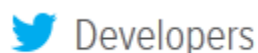
Una de les potències que ens dona aquest grup d'eines és el sistema de graella alhora de maquetar la pàgina web, que es tracta bàsicament d'una tècnica que es divideix la web en un seguit de columnes de manera que tots els elements de la web tindran una amplada N, on aquest nombre serà un nombre variable de columnes.

Un altre element es el disseny responsiu, es a dir que segons els dispositiu o el tamany del navegador de l'usuari la web s'adapta per a fer una navegació més còmode.

Finalment destacar que també incorpora un seguit de *plugins* que s'integren amb jQuery que li donen interacció i bons efectes a tots aquests elements que he comentat anteriorment.

10.4. Twitter API

Twitter proporciona als desenvolupadors un seguit de API per a que es pugui treballar amb totes les dades que genera aquesta plataforma. En el meu cas he fet servir la "REST API v1.1". (<https://dev.twitter.com/docs/api/1.1>)



Mitjançant el protocol REST⁴ podem fer connexions al servidor de Twitter un cop autenticats. Twitter ens proporciona un seguit de URL que podem cridar segons la informació que volem accedir.

El procés que fem servir per a realitzar tota la connexió és el següent:

1. Ens hem donat d'alta com a aplicació a la base de dades de Twitter i ens han proporcionat unes dades identificatives per fer connexions segures al seu sistema (un codi d'usuari i un codi secret).
2. Demanem a l'usuari que connecti el seu compte de Twitter amb l'aplicació:
 - a. Enviem una sol·licitud a la pàgina de connexió de Twitter amb les nostres dades
 - b. L'usuari fa login amb la seva compta i accedeix a donar permisos a l'aplicació
 - c. Twitter redirigeix a l'aplicació amb un codi d'accés de l'usuari. Donat que nosaltres necessitem fer connexions moltes vegades sense demanar permís a l'usuari cada vegada, tornem a sol·licitat a Twitter unes dades més permissives (User Token) i les enregistrem a la nostra base de dades.
3. Ara tenim les dades de l'usuari que identifiquen a l'eina i a l'usuari permeten una connexió directa.

⁴ Arquitectura de software que s'utilitza sobretot per les API web. Es fan connexions a servidors remots mitjançant el protocol HTTP.

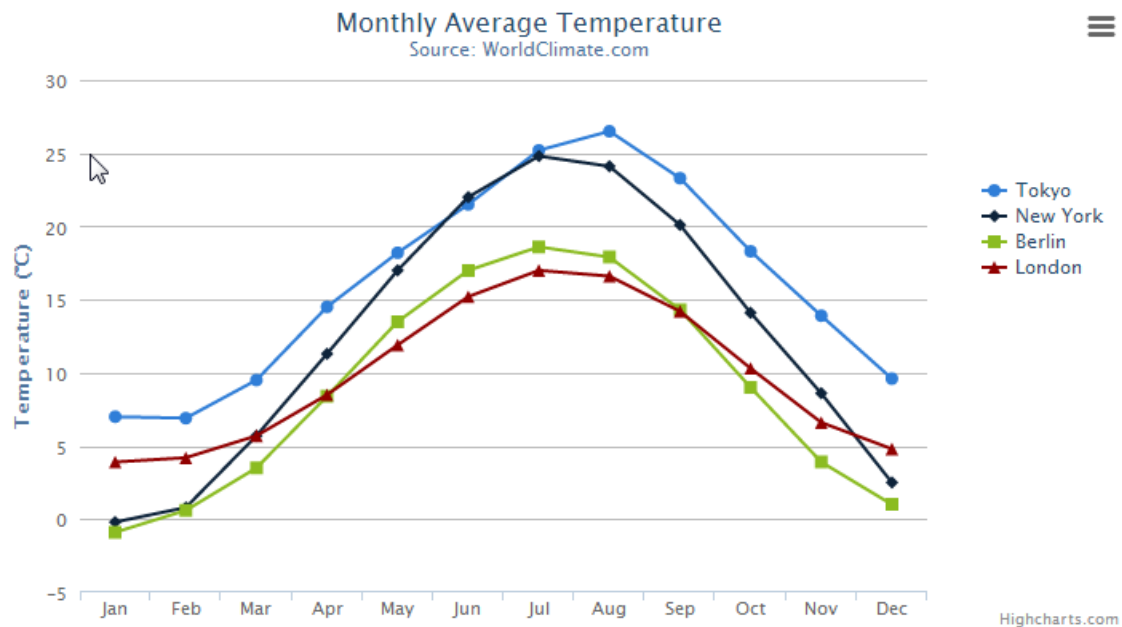
- Ara ja podem sincronitzar missatges de l'usuari amb les dades obtingudes en aquest procés d'autenticació: enviem una petició REST contra la API de Twitter i obtenim en JSON les dades sol·licitades.

10.5. HighCharts

Per acabar, també hem fet servir "Highcharts 3.0" que es una llibreria escrita purament en HTML5 i Javascript que ofereix d'una manera intuïtiva, gràfics interactius per una aplicació web. Entre d'altres, suporta gràfics de línia, de barres, de àrees, de columnes, rodones, angulars entre d'altres.

Mitjançant la API que ens proporciona aquesta llibreria podem generar aquests gràfics mitjançant un recull de dades que podem extreure de les bases de dades de l'aplicació:

- Nombre de missatges totals
- Organitzacions per categories, subcategories, tipologies
- Distincions de les fonts: Twitter, Facebook, Manual, etc..



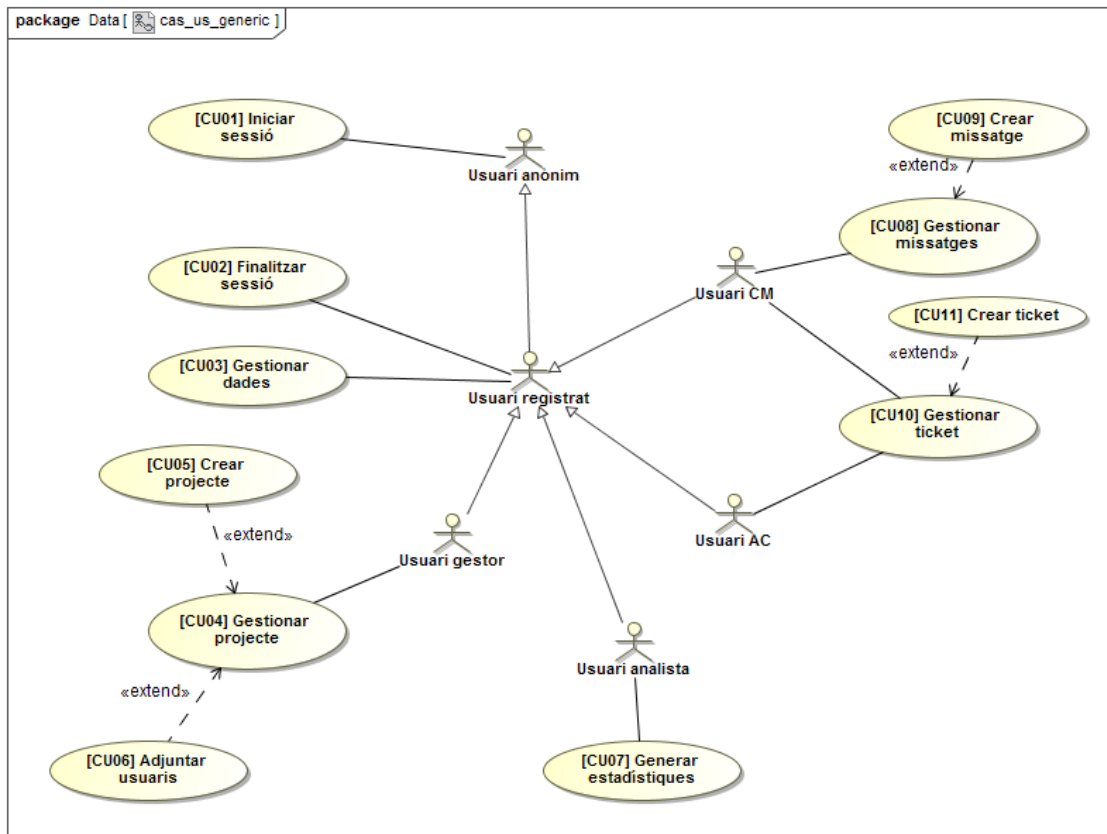
II-lustració 9. Exemple

11. Diagrames UML

Com a part de l'anàlisi funcional i tècnic es preparen un seguit de diagrames que ajuden a la definició del producte i donen pautes clares als desenvolupadors.

11.1. Diagrama de casos d'ús

Aquest diagrama serveix per a modelitzar els casos d'ús de l'aplicació web i relacionar-los amb els actors del sistema.



II-lustració 10. Casos d'ús

Aquest es el llistat de casos d'ús relacionats amb algunes captures de l'aplicació:

[CU01] Iniciar sessió



SMAC

Usuario:

Contraseña:

No cerrar sesión

[• Crear una cuenta](#)

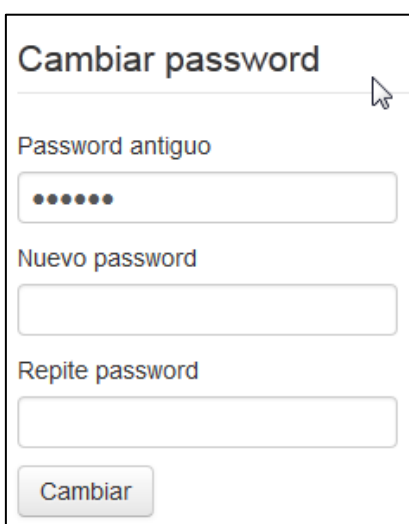
[CU02] Finalitzar sessió



Joan Madrid ▾

- Modificar datos
- Salir**

[CU03] Gestionar dades



Cambiar password

Password antiguo

Nuevo password

Repite password

[CU04] Gestionar projecte

SMAC UOC Atenció al Client

Gestionar Usuarios Catego

Name
UOC Atenció al Client

Description
Servei d'atenció al client online de la UOC

Editar

[CU05] Crear projecte

SMAC

Name
[Input field]

Description
[Input field]

Crear

- [Volver al listado de proyectos](#)

[CU06] Adjuntar usuarios

SMAC

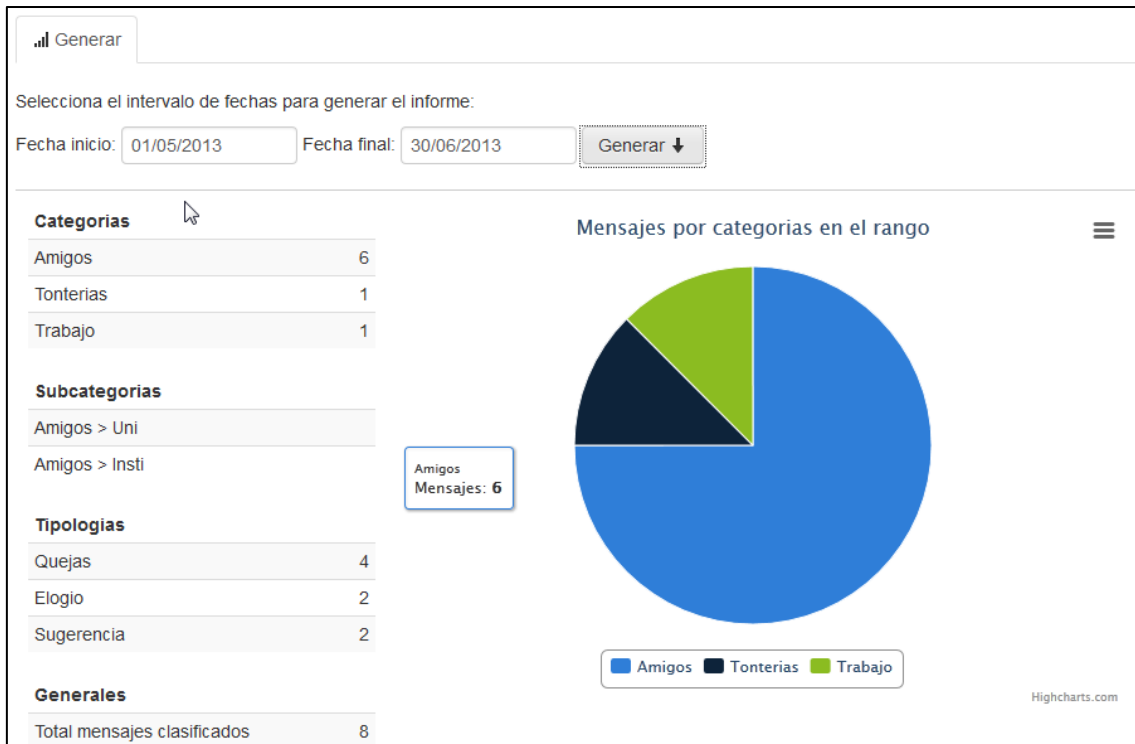
Gestionar Usuarios Categorias Tipologías

Adjuntar los usuarios...

Usuario	Gestor	Analista	C.M.	A.C.	
Joan Madrid(joanmadrid@uoc.edu)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>

Selecciona usuario... [Dropdown] Añadir

[CU07] Generar estadístiques



[CU08] Gestionar missatges

Clasificar Archivo

Twitter Publico

Cristina Pérez 19/05/2013 18:23

@UOCestudiant bona tarda se sap ja si psicopedagogia pasará a ser master? O com es fara? GRACIES

Clasificar

Categoría Subcategoria Tipología Genera ticket Ok

[CU09] Crear missatge

User

Text

Visibility

Publico

Source

Manual

Crear

[CU10] Gestionar *ticket*

The screenshot shows a web interface for managing a ticket. At the top left, there is a tab labeled "Gestionar". The ticket details are as follows:

- Ticket**: #3
- Creado**: 25/05/2013 23:08
- Estado**: Abierto
- Asignado**: nombre1 apellido1

On the right side, under "Generado del mensaje:", it shows the user "jaumegoma" and the time "25/05/2013 14:13". Below this is a message snippet: "Grande! RT @joanmadrid: Dicen que en @ulabox los que empaquetan son antiguos jugadores profesionales del tetrís. GJ http://t.co/LWycGyNKhH".

Below the details, there are three buttons: "Cerrar ticket", "Comentar", and "Asignar". The "Asignar" button is active, and a dropdown menu is open, showing two options: "pe in" and "nombre2 apellido2".

Below the dropdown, there are two messages from "nombre1 apellido1":

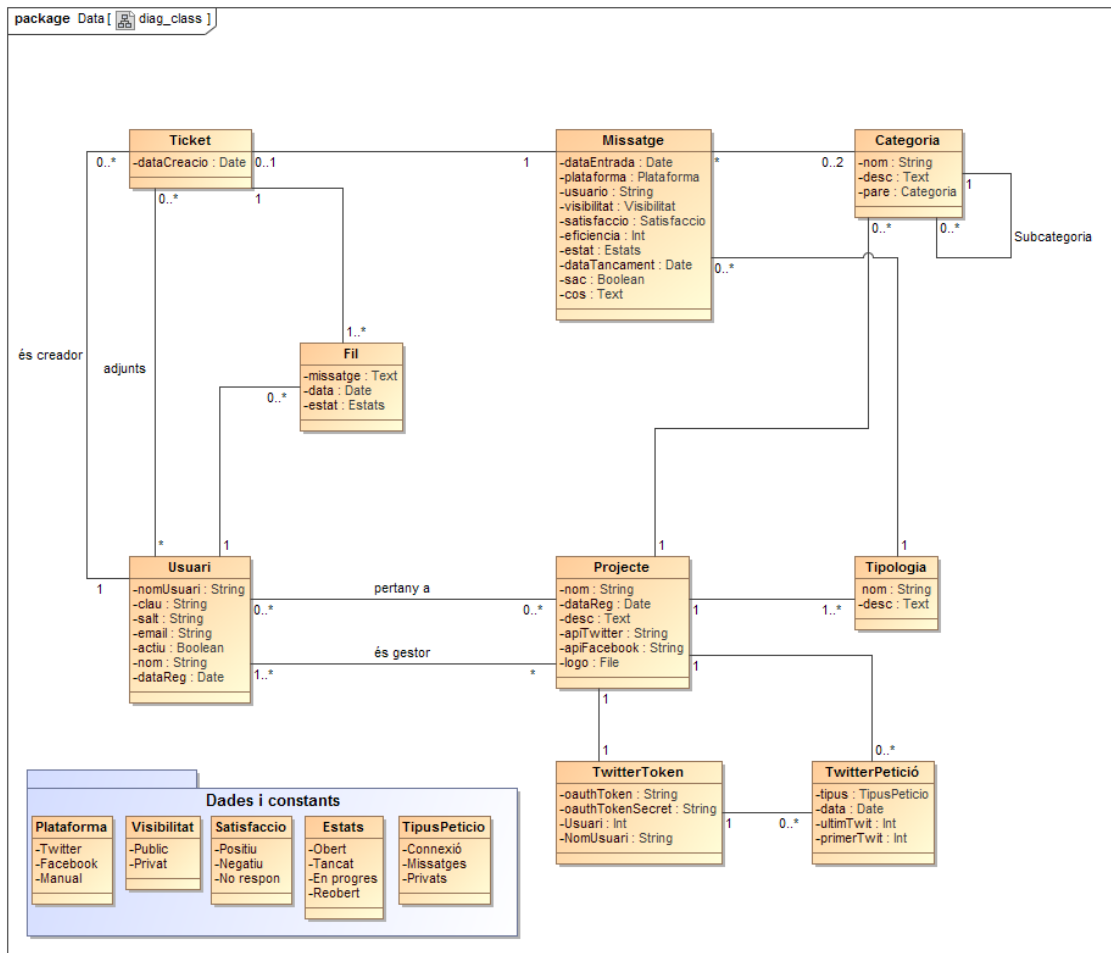
- Message 1: "nombre1 apellido1 (28/05/2013 23:08): holo"
- Message 2: "nombre1 apellido1 (28/05/2013 23:35): uyeeea"

[CU11] Crear *ticket*

The screenshot shows a form for creating a ticket. At the top, there is a text input field containing the text "través de DM para poder informarte. Gracias". Below this is another text input field containing the text "ia". To the right of this field is a dropdown menu with a downward arrow. To the right of the dropdown is a button labeled "Genera ticket" with a small icon of a ticket. To the right of the "Genera ticket" button is an "Ok" button. A mouse cursor is pointing at the "Genera ticket" button.

11.2. Diagrama de classes

En aquest diagrama es modelitzen les classes i objectes així com les seves relacions fent un èmfasi en els aspectes estructurals. En una aplicació web com aquesta, un diagrama de classes ens dona molta idea de com seran les classes donat el ORM ⁵que s'utilitza (*Doctrine*).



II-Il·lustració 11. Classes de SMAC

⁵ *Object-relational mapping*, es una tècnica per convertir dades entre llenguatges de programació i bases de dades reals. Es una abstracció per poder alternar entre sistemes diferents sense necessitat de refactoritzar el codi.

12. *Mockups*

A continuació s'han plantejat uns *mockups* d'una primera versió de l'aplicació. Aquests mapes o diagrames no son més que prototips de les pantalles de l'aplicació molt visuals que donen una idea general de la distribució de cadascun dels elements web que incorporarà.

Tal i com hem fet en aquest cas, s'utilitzen normalment per a il·lustrar la interfície que veurà l'usuari final, sense necessitat d'haver desenvolupat el software. A diferència d'un *wireframe*, que es molt més conceptual i arquitectònic, aquesta disposició d'imatges defineixen bastant bé com serà cadascuna de les vistes que hi haurà en la següent etapa de disseny i maquetació.

S'han dividit aquestes pantalles per zones de l'aplicació que reuneixen i agrupen un mateix concepte:

- **Entrada:** login, recuperar contrasenya, selecció de projecte.
- **Projecte:** gestió del projecte
- **Missatges:** rebuda y classificació de missatges
- **Tickets:** gestió i resolució de *tickets* i tasques
- **Informes:** generació de resums i informes estadístics.

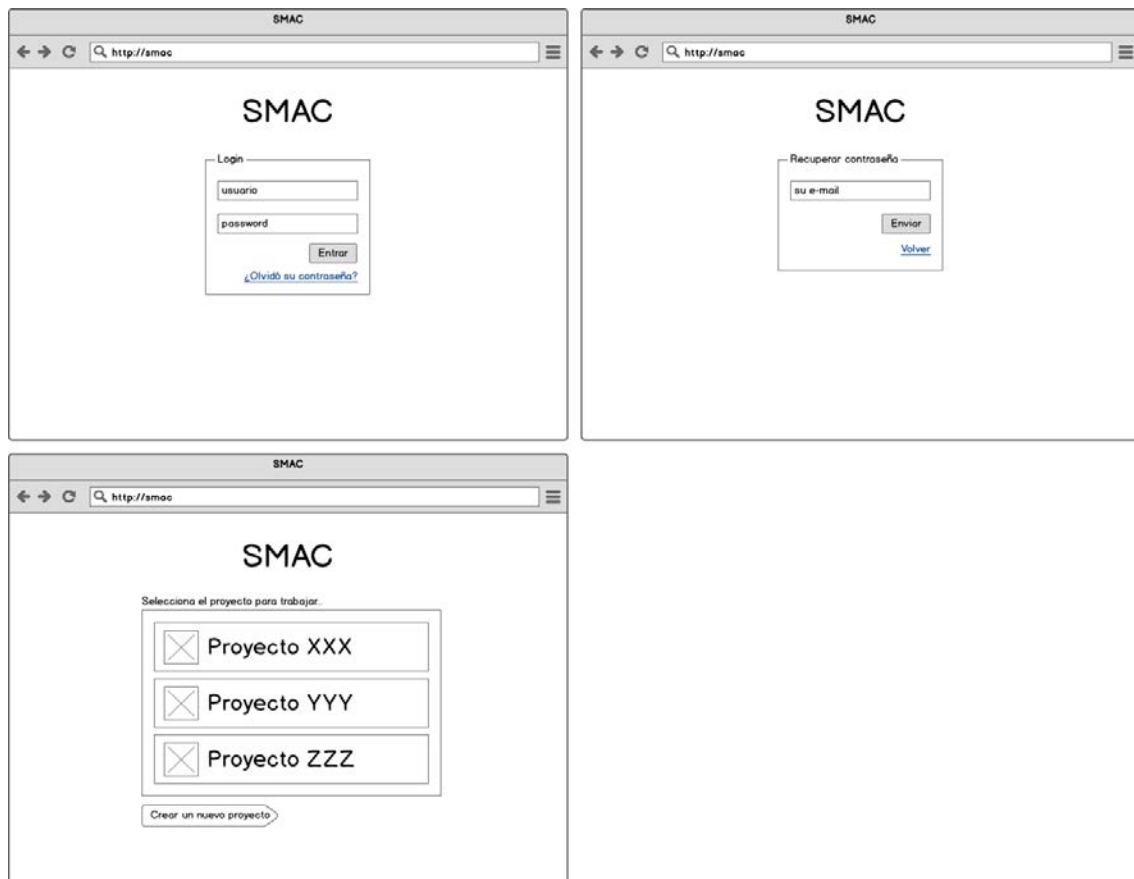
12.1. Entrada

En la entrada de l'aplicació disposarem d'un Login típic on es demana l'usuari i la clau de l'usuari. Donat que l'aplicació serà d'accés privat no hi ha opció de registre, ja que des de la web comercial es faria aquesta gestió després del pagament i la signa de contracte i terminis.

Un usuari pot gestionar un nombre indefinit de projectes, però cada vegada treballarà en un en concret, per això triaria al iniciar en quina instancia vol treballar en aquest moment.

En qualsevol moment l'usuari podrà decidir alternar de projecte tornant a aquesta pantalla, que simplificaria el procés de sortir de l'aplicació i tornar a entrar.

La maqueta de totes les pantalles de la entrada és diferent a la resta de l'aplicació, donat que fins que no hi ha accés no té sentit utilitzar el disseny funcional de la web, apart que des de el punt de vista de disseny i decoratiu es molt més simple i fàcil de fer servir, donat que els elements per interactuar son mínims.



II-lustració 12. Mockup de Login

12.2. Projecte

Una de les grans seccions serà destinada al projecte, en singular, donat que anteriorment l'usuari haurà escollit en quina instància vol treballar.

En aquest punt es poden gestionar totes les dades associades del projecte, des de el nom fins la imatge, passant per la gestió de les API's de xarxes socials.

També, els usuaris que disposin dels permisos oportuns, podran afegir altres usuaris al projecte. La idea en aquesta secció es que només un usuari gestor tingui accés i pugui decidir quins treballadors seran assignats en cada projecte.

The screenshot shows a web browser window with the URL 'http://smac'. The page title is 'SMAC Marca X' and the navigation menu includes 'Mensaje', 'Tickets', and 'Informes'. The user is logged in as 'Usuario' and can click 'Salir'. The main content area is titled 'Gestionar Usuarios' and contains a form for project configuration. The form fields are: 'Nombre del proyecto' (Text Field), 'Logotipo' (Browse for Files with a 'Browse' button), 'Descripción' (Text Area), 'API Twitter' (ON/OFF toggle), and 'API Facebook' (OFF toggle). A 'Guardar' button is at the bottom.

The screenshot shows the 'Usuarios' section of the SMAC application. It features a search field and a list of 'Usuarios disponibles' (Usuario 01 to Usuario 14). A large arrow points to the 'Usuarios adjuntos' table, which lists users and their assigned roles. The table has columns for 'Usuario', 'Gestor', 'Analista', 'CM', and 'AC'. The first row (Usuario AA) has all roles checked. The second row (Usuario BB) has 'Analista' checked. The third row (Usuario CC) has 'Analista' checked. The fourth row (Usuario DD) has 'Analista' checked. The fifth row (Usuario EE) has 'Analista' checked. The sixth row (Usuario FF) has 'Analista' checked. The seventh row (Usuario GG) has 'Analista' checked. The eighth row (Usuario HH) has 'Analista' checked. There is an 'Añadir usuario' button at the bottom left.

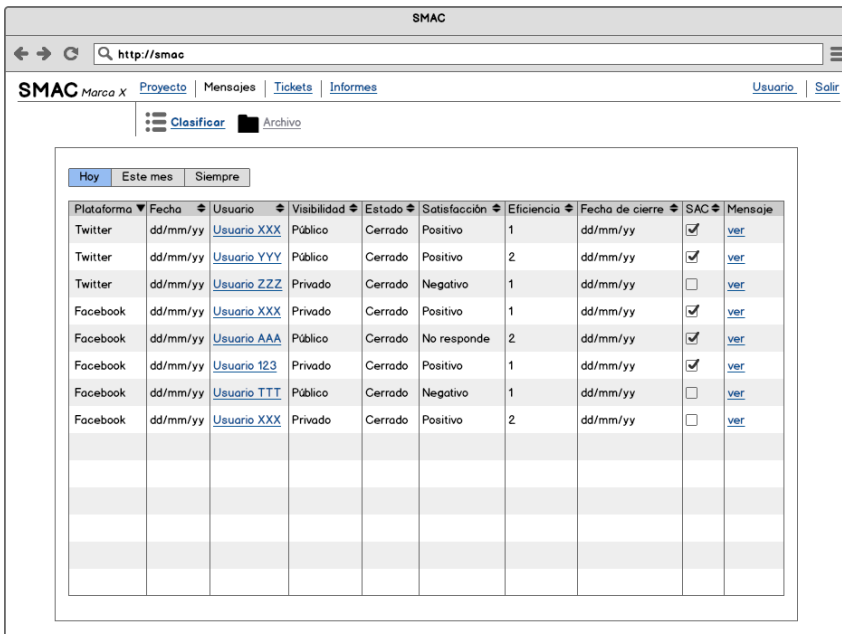
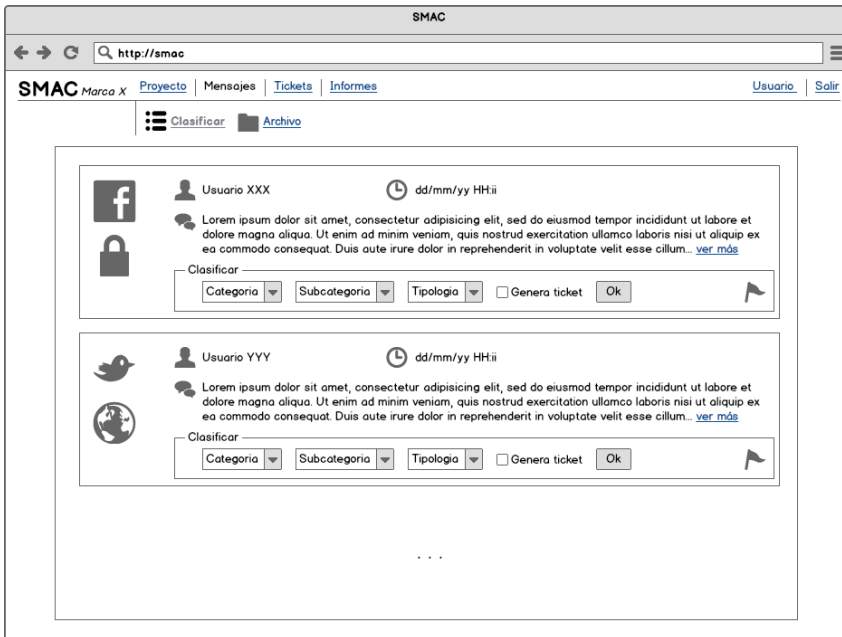
Usuario	Gestor	Analista	CM	AC
Usuario AA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Usuario BB	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Usuario CC	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Usuario DD	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Usuario EE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Usuario FF	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Usuario GG	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Usuario HH	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

II-Il·lustració 13. Mockup de projecte

12.3. Missatges

En aquesta secció el sistema haurà generat els missatges que, mitjançant les *API's* de les xarxes socials, haurà rebut i emmagatzemat. Aquest missatges, en la primera versió de l'aplicació, serà necessària la intel·ligència humana per a poder classificar-los i poder definir dades com la tipologia, la categoria i la subcategoria.

Aquest es un punt on, normalment el Community Manager, decidirà si un missatge necessita ser associat a un ticket, donat que l'usuari que l'ha generat sol·licita informació, es queixa o genera algun feedback que necessita ésser resolt.



Il·lustració 14. Mockup de missatges

12.4. Tickets

Des de l'apartat anterior, s'hauran decidit quins missatges rebuts generaran un ticket. Aquesta associació significa que el gestor o Community Manager ha de resoldre un problema o una petició, i per això es tractarà totalment per separat.

En aquest punt es té una visió global de l'estat de tots els tickets actius, amb la seva prioritat i un petit extracte o sumari de la informació del missatge original.

Dins de cadascun dels tickets, hi haurà una pantalla on estarà ampliada aquesta informació resum i on es tindran el controls per contactar amb altres usuaris de atenció al client per a poder resoldre els dubtes. Aquest ticket anirà sofrint canvis d'estat, de prioritat i apart se li aniran afegint nous comentaris a mesura que sigui necessari.

Ticket	Fecha creación	Prioridad	Sumario	Estado	SAC
XZ12	dd/mm/yy	5	Lorem ipsum dolor sit amet elit sed do eiusmod...	Abierto	<input checked="" type="checkbox"/>
XZ13	dd/mm/yy	5	Lorem ipsum dolor sit amet elit sed do eiusmod...	Cerrado	<input checked="" type="checkbox"/>
XZ15	dd/mm/yy	4	Lorem ipsum dolor sit amet elit sed do eiusmod...	En progreso	<input type="checkbox"/>
XZ16	dd/mm/yy	4	Lorem ipsum dolor sit amet elit sed do eiusmod...	Reabierto	<input type="checkbox"/>
XZ19	dd/mm/yy	3	Lorem ipsum dolor sit amet elit sed do eiusmod...	Resuelto	<input type="checkbox"/>
XZ22	dd/mm/yy	2	Lorem ipsum dolor sit amet elit sed do eiusmod...	Abierto	<input checked="" type="checkbox"/>
XZ32	dd/mm/yy	1	Lorem ipsum dolor sit amet elit sed do eiusmod...	Abierto	<input type="checkbox"/>
XZ55	dd/mm/yy	1	Lorem ipsum dolor sit amet elit sed do eiusmod...	Abierto	<input checked="" type="checkbox"/>

Ticket: XZ16

Fecha de creación: dd/mm/yyyy

Prioridad: Muy Alta (5)

Estado actual: Abierto

Asignado a: [UsuarioAC](#)

Sumario:
 Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Comentario:

Asignar a:

Prioridad:

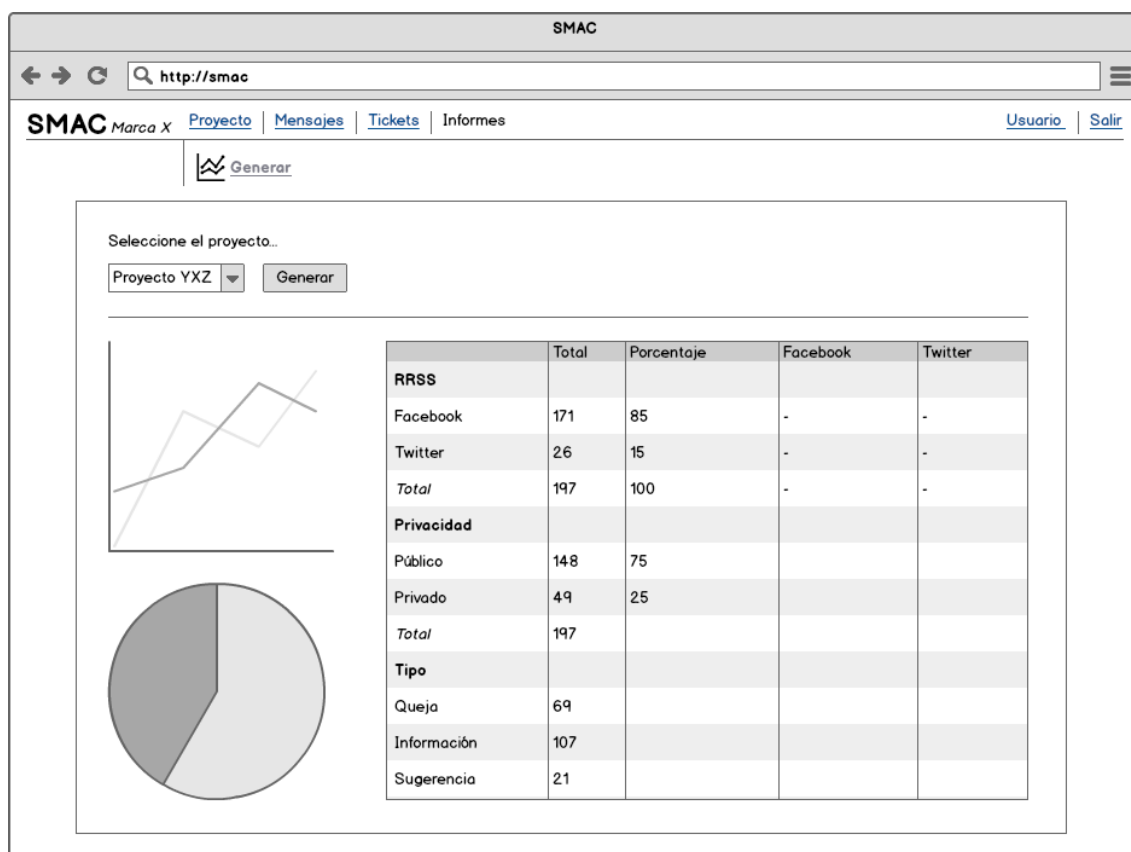
Nuevo estado:

II-lustració 15. Mockup de ticket

12.5. Informes

Aquesta secció és el que engloba tota la informació generada durant tot el procés de treball en l'aplicació. Aquí podem crear un informe que recollirà totes les dades, agrupades per mesos naturals on es detallarà el total de totes les tipologies, categories i subcategories.

També hi haurà un resum d'aquell tickets que s'han generat juntament amb la seva quota de resolució. En aquest punt es faran uns càlculs de iteracions de eficiència, resolució i satisfacció final de l'usuari que ha generat cada missatge.



Il·lustració 16. Mockup de informe

13. Perfils d'usuari

Un component essencial de la fase del Disseny Centrat en l'Usuari (DCU) es l'anàlisi de usuaris, que proporciona detalls sobre qui farà servei del producte. Aquest anàlisi d'usuari identifica roles i defineix les característiques de l'usuari:

- el nivell de coneixement
- experiència i habilitat amb productes similars
- el seu entorn
- freqüència d'ús
- tipus de hardware, software i tecnologies que fa servir

Tot això es necessari si es vol evitar dissenyar sense un procés formal que tingui en compte els demes, ja que sinó es pot caure en l'error de definir per un mateix.

En el nostre cas hem trobat que en aquesta aplicació conviuen dos grups d'usuaris diferents:

- Community Manager's
- Administratius d'atenció al client

A continuació s'avaluarà cadascun d'aquests grups de manera que es pugui treure profit de la informació alhora de plantejar el DCU.

13.1. Community Manager's

13.1.1. Context

És una branca professional que esta creixent exponencialment durant els últims anys. Son els encarregats de gestionar les xarxes socials i el Social Media d'una marca o una empresa en concret.

13.1.2. Factors demogràfics

Donada la recent aparició d'aquests rols encara no hi ha dades oficials del Ministeri o de sindicats, pel que en aquest punt, em baso en l'experiència personal i professional.

60% son dones i 40% son homes

Edats compreses entre 18 i 27 de mitja

Més d'un 70% porten més d'un projecte o treballen per varies marques o agència.

El 75% son titulats universitaris superiors

D'aquests titulats un 40% son de Periodisme, un 40% de Comunicació i Relacions Publicques i la resta (20%) d'altres titulacions universitàries o enginyeries.

13.1.3. Entorn

L'entorn i abast per utilitzar aquesta aplicació acostuma a ser en oficina, tot i que donada la necessitats d'atendre peticions d'usuaris a totes hores i en qualsevol dia de la setmana s'acostuma a teletreballar, per tant els equips son tan professionals, com personals inclús adaptats en mòbil o *tablet*.

Podríem destacar que una gran quantitat d'agències prefereixen l'ús de ordenadors amb el sistema operatiu.

13.1.4. Productes semblants

Els CM estan acostumats a fer ús d'una gran varietat d'eines per a fer el seu dia a dia més senzill. Moltes d'aquestes eines tenen a veure amb la monitorització i gestió de les xarxes socials.

13.2. Atenció al client

13.1.1. Context

Un tècnic d'atenció al client sol ser un nom a comercials i administratius de teleoperacions. És un camp molt estès professionalment tant en el nostre país com en altres països amb una renda més baixa, en el que s'estan donant casos de externalització d'empreses per abaratir costos.

13.1.2. Factors demogràfics

Normalment la gent que es dedica en aquest camp professional solen estar definits pels següents factors demogràfics:

85% son dones i 15% son homes

Edats compreses entre 25 i 45 de mitja

Un 85% treballen per una mateixa marca o empresa

Un 5% tenen estudis universitaris superiors o enginyeries

13.1.3. Entorn

Aquests operadors acostumen a treballar en grans empreses dedicades amb equipació telefònica específica. En casos de marques o empreses més senzilles, pot arribar a ser un petit departament associat o annex.

Els equips de treball solen ser ordenadors senzills amb porques prestacions i normalment es fa tota la feina a la oficina.

13.1.4. Productes semblants

Aquest perfil té poca experiència en productes tecnològics semblants i normalment se'ls hi dedica un recurs de formació previ abans de fer servir qualsevol programa informàtic donada la seva situació de inexperiència amb altres programari.

13.3. Conclusions

Tot i que els perfils son bastant diferents un de l'altre, és dedicaran el màxim esforç de desenvolupament cap a la part de Community Manager's i tècnics, donat que son els que faran servir l'aplicació en un rang més extens i on basaran la majoria de la seva feina.

El rol de Atenció el Client serà un suport extern i farà poc ús de l'aplicació a la vegada que tindrà un accés molt restringit a aquesta.

Per la part tecnològica s'haurà de tenir molt en compte la compatibilitat de l'aplicació web tant en sistemes Windows com en OSX i els seus navegadors. Seria molt interessant tenir en compte un accés des de mòbil o *tablet*.

L'aplicació pot fer servir un llenguatge tècnic i unes eines professionals, donat que seran les companyies que apostin per aquest producte i paguin el seu preu, seran interessats en documentar-se i en formar-se en l'aplicació.

14. Usabilitat i UX

Es interessant tenir en compte l'experiència de l'usuari en el moment que esta fent servir l'aplicació web SMAC, com podem fer optima la seva navegabilitat i tot l'entorn que genera aquesta experiència de treball.

Per altra banda, la usabilitat anirà molt lligada a una bona experiència d'usuari tot i no ser el mateix concepte, caldrà tenir en compte que l'aplicació sigui practica i intuïtiva així com l'accés a totes les seves seccions sigui clar i sense obstacles.

14.1. Navegació

L'esquema de navegació que disposa l'aplicació es bàsicament global, navegarà entre les quatre grans seccions: projecte, missatges, tickets i informes.

La profunditat màxima serà 3, donada per una gran secció de les anterior mencionades, dins d'aquesta disposarem d'un seguit d'accions: gestionar, administrar, adjuntar o generar i finalment en un tercer nivell tindriem una fitxa detall concreta com per exemple podria ser una fitxa d'un ticket amb tota la seva informació associada.

En tot moment disposarem d'un *breadcrumb*, o conegut com fil d'Arianna, que ens informarà en tot moment que en la part superior ens informarà en quina secció, subsecció i/o fitxa ens trobem, de manera que serem conscients de la nostra situació i a la vegada podrem desfer el camí fàcilment.

14.2. Sitemap

El *sitemap* de l'aplicació serà el següent que consta d'una llista de pàgines accessibles pels usuaris. En el nostre cas ho organitzem en un format de llista donada la simplicitat (a propòsit) de l'aplicació:

- SMAC
 - Login
 - Logout
 - Gestió de dades d'usuari
 - Selecció de projecte
 - Crear projecte
 - Projecte
 - Gestionar
 - Usuaris
 - Missatges
 - Classificar
 - Fitxa de missatge
 - Arxiu
 - Fitxa de missatge
 - Tickets
 - Gestionar
 - Fitxa de ticket
 - Informes
 - Generar informe

14.3. DCU

En les primeres etapes de l'aplicació hem de preveure com avaluarem el disseny centrat en l'usuari, quines tècniques podrem aplicar per millorar l'experiència del client.

Part de l'anàlisi de les seccions anteriors ens parlem molt de com s'està dissenyant aquesta experiència, així com lavaluació de grups de rols que faran servir l'eina. Tot i que tota aquesta part esmentada formarà part de la prèvia.

Una vegada l'aplicació sigui operativa, en una primera versió *alpha*, es podran aplicar noves tècniques per a fer una segona versió amb millores en aquest camp.

- Tests amb usuaris reals i entrevistes
- Avaluació heurística

15. Seguretat

L'aplicació ha de ser segura tant com per l'usuari que la fa servir així com les dades que aquest aporta al sistema, però també ho ha de ser pels desenvolupadors i creadors d'aquesta, que han de protegir el software que han desenvolupat així com l'accés al seu codi. Per això dividirem aquest apartat en dos gran termes, la part aplicació de l'usuari i la part servidor de dades.

15.1. Seguretat del servidor

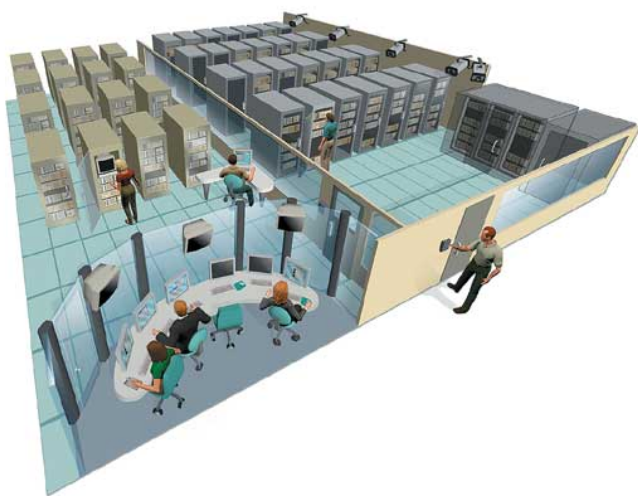
En l'apartat "6. Arquitectura de l'aplicació" detallo amb un gràfic com serà l'entorn de producció de SMAC i explico quina serà la comunicació entre els servidors. És un exemple de com podem protegir que les dades puguin ser accedides des de l'exterior en entorns no controlats.

És molt important per nosaltres protegir les dades que els clients insereixen a les nostres bases de dades per això es contractarà un servei professional de servidors dedicats que ens aporten les següents característiques tant físicament com d'aspecte digital o virtual.

Firewall en hardware: estem acostumats a fer servir tallafocs de programari, que son programes que filtren l'accés no desitjat, tant del nostre ordinador cap a fora com a la inversa. En el món dels servidors, existeix el concepte de tallafocs per hardware, que consten d'aparells normalment connectats als *routers* que aporten una nova capa de seguretat en la xarxa, filtren paquets i examinant capçaleres sospitoses.

D'aquesta manera podem evitar atacs del tipus *DDoS*⁶ i altres situacions que podrien fer fallida en el nostre servidor.

Datacenters vigilats: els *datacenters* son els llocs físics on estan els servidors, és a dir, on estarà l'ordinador que dona el servei web. Aquests *datacenters* estan dotats de una monitorització les 24h al dia, on disposen de tècnics presents 24/7 també per qualsevol incidència. Els servidors estan protegits de pics de corrent així com baixades de tensió amb una triple toma elèctrica. Els edificis tenen accés controlat i servei de seguretat.



II-Il·lustració 17. Datacenter vigilat

⁶ Distributed denial of service attack, son intents aconseguir denegar l'accés a un servidor mitjançant la creació de moltíssimes peticions a aquest amb l'únic objectiu de col·lapsar-lo.

15.2. Seguretat de l'aplicació

La seguretat en l'aplicació anirà de la mà del *framework Symfony* que ens dona un seguit d'eines per a controlar tot aquest àmbit.

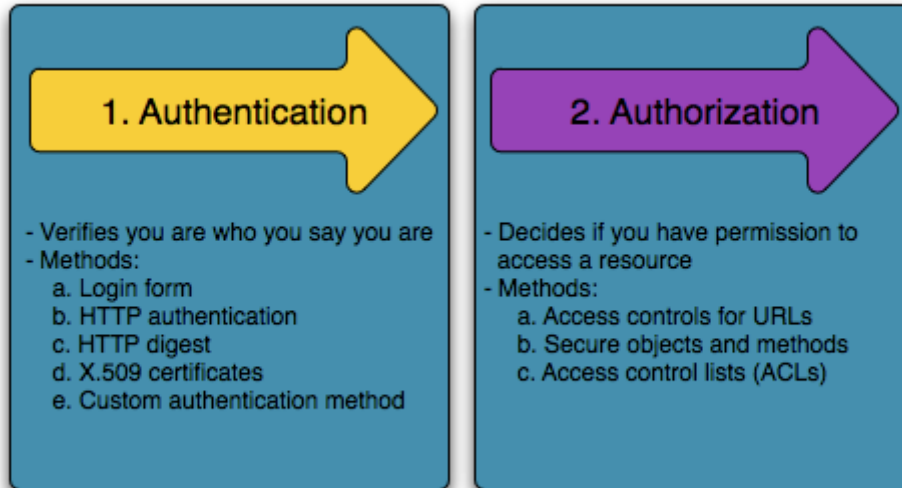
Constarà en un procés de dos passos on l'objectiu serà prevenir a un usuari a accedir a un recurs del qual no en té accés.

En el primer pas del procés, el sistema de seguretat identificarà qui és cada usuari requerint que aquest aportï algun tipus d'identificació. Aquest procés s'anomena **autenticació** i significa que el sistema intenta esbrinar qui ets. En el nostre cas aquest procés serà el Login.

Una vegada el sistema sap qui ets, el següent pas és determinar si hauries de tenir accés a segons quin recurs, aquesta part del procés s'anomena **autorització**, que significa que el sistema comprovarà si tens privilegis per a executar una acció determinada.

Exemple:

Alba Grau efectua un Login a l'aplicació correctament, i en la selecció del projecte li apareixen només aquells projectes dels quals tingui privilegis de visualització i treball. Encara que d'alguna manera aconseguís accedir a un projecte al qual no estigui adjuntada, li serà impossible realitzar cap acció i el sistema donarà una excepció del tipus "Autorització denegada".



II-lustració 18. Diagrama dels 2 passos de seguretat

16. Tests

En l'etapa de desenvolupament és molt comú generar errors o programar de manera que no es tinguin en compte totes les possibilitats de l'usuari final, i aquests és un fet que no es pot canviar.

Però si que disposem de mecanismes per a minimitzar aquesta font d'errors i assegurar-nos al màxim que el producte que entregarem al client o publicarem a Internet tindrà minimitzada aquesta problemàtica.

En el nostre cas del desenvolupament farem ús de dues eines que tot i aplicar-se en moments diferents de la construcció de la web tenen com a objectius trobar forats i problemes de manera que els programadors puguem ser capaços de resoldre abans que arribi al públic.

16.1. Unit testing

Cada vegada que un programador escriu una línia de codi, potencialment s'estan afegint nous *bugs*, per tal de construir aplicacions fiables serà necessari fer comprovacions dels nostre codi a nivell funcional però també a nivell de codi.

Symfony2 integra una llibreria independent de tests a nivell de codi anomenada *PHPUnit* que per si sol ja es un *framework* dedicat a aquest àmbit.

El Unit *Testing* normalment serà un test contra una classe específica de PHP, on comprovarem la veracitat de comandes unitàries.

Exemple:

Suposem que tenim una classe "Calculadora" on tindrem funcions de suma, resta i multiplicació. Crearem un PHPUnit per aquesta classe on comprovéssim sempre que per exemple 2+3 son 5. Sembla obvi que sempre serà cert, però pot arribar un punt del desenvolupament que una dependència d'aquesta classe canviï i tot que la resposta ha de seguir sent la mateixa, per una raó desconeguda o bug, això no sigui així.

Donat que el PHPUnit sempre l'executarem podríem detectar-ho fàcilment i ens adonaríem que situacions obvies moltes vegades poden fallar.

Òbviament es un exemple increïblement simple, però a mesura que el desenvolupament de l'aplicació avanci, si som capaços de comprovar petits fragments de codi que no han de ser molt més complexos, al final podrem estar segurs que en quant a codi sempre obtindrem el resultat que esperem.

16.2. Selenium IDE

Per altra banda, tot i que el codi no tingui errors i tot sembla que vagi bé poden sorgir errors funcionals a les vistes, o retorns de funcions que tot i que tornin coses lògiques a nivell de codi, per una persona humana siguin res potes totalment inadequades.

Hom podria pensar que la única manera d'assegurar-nos que això no passi es fer proves i més proves del codi que anem generant. Aquesta afirmació es certa i necessària, els programadors sempre hauran de provar el codi que generen, però moltes vegades això no es suficient, donat que moltes parts ja acabades i funcionals no es proven cada vegada per falta de temps o simplement perquè els programadors acaben "viciats" de la seva eina i a vegades no entenen que un usuari extern pot ser capaç d'accedir o realitzar accions que no espera.

Per això existeixen programes que ens ajuden en aquesta tasca. En SMAC farem servir *Selenium IDE*, que es un sistema integrat de desenvolupament que actua com una extensió per al navegador *Firefox* que ens permet guardar accions definides en scripts que permeten "gravar" el comportament d'una persona a la nostra web i reproduir-ho automàticament.

D'aquesta manera podem definir i automatitzar un seguit de tasques que executariem automàticament cada vegada que volguéssim comprovar la nostra aplicació. Aquestes tasques es componen d'accions i respostes esperades, si una resposta no es la esperada el sistema avisarà que hi ha un error.

Exemple:

Posem a gravar amb el Selenium IDE un procés de login, on entrem amb l'usuari "admin" i la clau "123123" que en principi sempre existirà al sistema, quan accedim la web retorna un "Benvingut Admin!".

Aquest petit test quedaria gravat com una tasca que executariem sempre prèviament a altres. Si en algun moment la resposta no es "Benvingut Admin!" sinó que és "Error: usuari invàlid", Selenium faria saltar un avís que no s'ha trobat el que esperava.

Es per això que de moment em planificar les següents tasques per a SMAC en un entorn controlat on tinguem dades de prova fixades (fixtures⁷), aquestes tasques les dividirem en els següents blocs.

Bateria 1

Nota: s'executarà el paquet 1 de fixtures amb usuaris de prova

1. Login amb un usuari de prova "Usuario1".
2. Creació del projecte "Proyecto X", omplir les dades bàsiques.
3. Assignar al projecte els usuaris "Usuario 2" (CM) i "Usuario 3" (AC).
4. Logout de "Usuari 1".

⁷ Les "fixtures" son dades de prova que podem forçar en un entorn de desenvolupament per a poder aplicar-hi sistemes com el Selenium o proves amb persones reals.

Bateria 2

Nota: s'executarà el paquet 2 de fixtures amb missatges per al "Proyecto X", també inclourà categories i tipologies associades.

1. Login amb "Usuario 2".
2. Accés al "Proyecto X", a la secció missatges.
3. Classificació del "Mensaje 1" amb "Categoria 1", "Subcategoria 2", "Tipologia 1".
4. Marcar el missatge com a ticket i guardar la tramesa.
5. Logout de "Usuario 2".

Bateria 3

1. Login amb "Usuario 3".
2. Accés al "Proyecto X", a la secció tickets.
3. Obrir el "Ticket 1", escriure un comentari, canviar la prioritat i assignar-ho a un "Usuario 2".
4. Logout amb "Usuario 3".

Bateria 4

Nota: s'executarà el paquet 3 de fixtures amb usuari analista i dades de missatges i tickets resolts.

1. Login amb "Usuario analista"
2. Accés al "Proyecto X", a la secció informes.
3. Generar informe del mes actual i esperar un seguit de dades.
4. Logout amb "Usuario analista".

17. Versions de l'aplicació

El versionat en el software es una realitat que tracta d'assignar etiquetes úniques a versions d'aquest. Donada una etiqueta o versió en un nombre, nom o codificació podem entendre en quina etapa del desenvolupament ens trobem, podem referir-nos temporalment a diferents etapes i revisar quins canvis, millores i correccions hi ha hagut entre una versió i una altra.

Durant aquesta etapa de desenvolupament en el termini del projecte de final de grau s'alliberaran tres versions diferents però totalment operatives de l'aplicació i a continuació es detallaran les característiques (referides comunament com a *features*):

17.1. Versió Alpha

- Sistema d'autenticació: registre, *login*, *logout*.
- Sistema d'autorització: control de projectes i adjunts
- Sistema bàsic de missatges: inserció manual
- Sistema de *ticketing*
- Interfície bàsica sense estils

17.2. Versió Beta

- Correcció de *bugs* de la versió anterior
- Integració de API de *Twitter* per a crear nous missatges
- Integració de API de *Facebook* per a crear nous missatges
- Millores en interacció bàsica
- Control d'errors
- Interfície nova amb Bootstrap



17.3. Versió 1.0

- Sistema de generació d'informes i resums
- Correcció de *bugs* de la versió anterior
- Textos correctes i ajudes en l'aplicació
- Millora en la interfície i la interacció amb *jQuery*

18. Requisits d'instal·lació

Per tal d'instal·lar el software de SMAC és necessari un entorn de servidor web LAMP (*Linux, Apache, MySQL, PHP*) juntament amb una base de dades *MongoDB*. A part d'això també es necessari complir amb els requisits mínims que proposa el *framework* *Symfony2*.

Entrant en detall i per tal que l'aplicació funcioni perfectament en un entorn controlat, que es igual a l'entorn que s'ha fet servir per al desenvolupament serà necessari:

- Servidor Ubuntu Server [12.04.2 LTS]
- PHP [5.3.10-1ubuntu3.6 with Suhosin-Patch (cli)]
 - Extensions necessàries:
 - ext-mongo
 - ext-intl
 - ext-fileinfo
 - pdo_mysql
 - ext-curl
 - apc
- Apache [2.2.22 (Ubuntu)]
 - mod_rewrite activat
- MongoDB [Shell version: 2.4.3]
- MySQL [Server version: 5.5.31-0ubuntu0.12.04.1 (Ubuntu)]
- Composer

No es necessari que les versions siguin exactes tot i que es recomanable fer servir al menys la indicada o una superior, ja que es possible que s'utilitzi alguna característica que no estigui disponible en una versió anterior.

19. Instruccions d'instal·lació

Una vegada disposem d'un servidor amb els requeriments esmentats anteriorment, haurem de disposar del codi de l'aplicació en la del Apache on apunti el domini o la ip publica en la que vulguem que sigui visible.

Nota: es important que el virtual host de l'Apache apunti directament a la carpeta web/app.php

Primer de tot caldrà instal·lar les llibreries de tercers que utilitza l'aplicació (anomenades *vendor*) mitjançant el control de dependències de *Composer*

```
php composer.phar install
```

Seguidament haurem de configurar l'arxiu situat a *app/config/paràmetres.yml*, on haurem disposar de les dades de connexió a la nostre base de dades MySQL

```
parameters:
  database_driver:  pdo_mysql
  database_host:   127.0.0.1
  database_port:   ~
  database_name:   smac
  database_user:   root
  database_password: ~

  mailer_transport: smtp
  mailer_host:      127.0.0.1
  mailer_user:      ~
  mailer_password: ~

  locale:          es
  secret:          ThisTokenIsNotSoSecretChangeItMmmLater
```

També farem lo mateix a l'arxiu *app/config/config.yml* per les dades de connexió de MongoDB:

```
#mongodb
doctrine_mongodb:
  connections:
    default:
      server: mongodb://localhost:27017
      options: {}
  default_database: smac
  document_managers:
    default:
      auto_mapping: true
```

A continuació generarem els arxius de cache de manera que l'aplicació funcioni més ràpidament

```
php app/console cache:clear --env=prod
```

Després publicarem els arxius públics com *css*, *javascript* i imatges. Donat que el CSS es generarà compilant els arxius LESS automàticament, cal avisar al mòdul "Assetic" que ho compili prèviament.

```
php app/console assets:install web  
php app/console assetic:dump --env=prod
```

Finalment generarem automàticament les dades necessàries perquè l'aplicació funcioni correctament (a la base de dades):

```
php app/console doctrine:fixtures:load
```

20. Instruccions d'ús

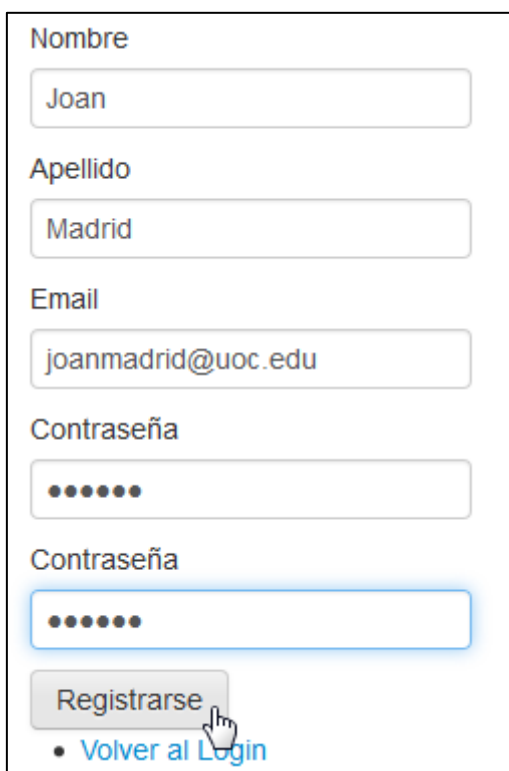
20.1. Creació d'un usuari

Primerament ens haurem de registrar a l'aplicació web amb un nou usuari. Amb aquest nou usuari podrem gestionar tants projectes com vulguem.

Primerament, des de la pantalla inicial, farem clic a "Crear una cuenta"



A continuació omplim tots els camps del formulari i premem el botó "Registrarse", d'aquesta manera ja tindrem creat un usuari a l'eina.

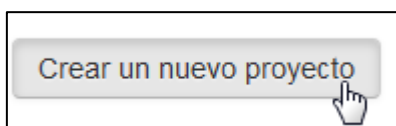
Aquesta imatge mostra un formulari de registre completat. Els camps són: "Nombre" amb el valor "Joan", "Apellido" amb "Madrid", "Email" amb "joanmadrid@uoc.edu", i dos camps "Contraseña" amb punts ocults. A la part inferior hi ha un botó gris "Registrarse" i un enllaç de text blau "Volver al Login". Una cursor de ratolí està posat sobre el botó "Registrarse".

20.2. Creació i selecció de projecte

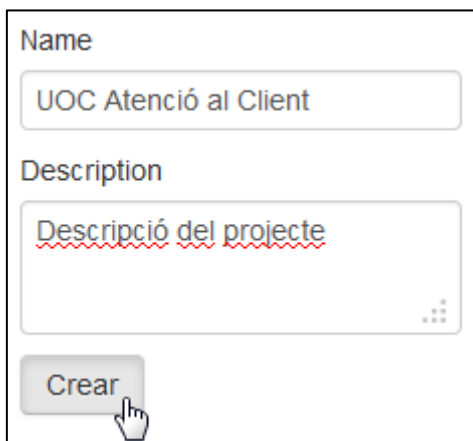
Podem tornar a la pantalla inicial i entrar amb les nostres dades d'usuari prèviament creades (email i password). Un cop a dins, podem seleccionar un projecte ja creat fent clic al nom del projecte del llistat.



En cas que no tinguem cap projecte podem crear-ne un de nou prement el botó corresponent:



Un cop dins, omplirem els camps de text, que son obligatoris i crearem el projecte.

A screenshot of a form for creating a new project. It has two text input fields. The first is labeled "Name" and contains the text "UOC Atenció al Client". The second is labeled "Description" and contains the text "Descripció del projecte" with a red wavy underline. Below the description field is a "Crear" button with a mouse cursor pointing at it.

Tant si hem creat un de nou, com si hem seleccionat un prèviament creat, accedirem dins de l'eina en l'àmbit del projecte creat, és a dir, tota la feina que fem farà referencia a aquest projecte.

Podem saber en qualsevol moment, en quin projecte estem treballant mirant la capçalera i podem canviar aquest àmbit fent clic al títol principal de la pàgina:



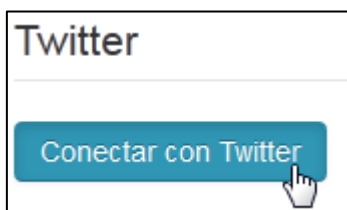
20.3. Configuració del projecte

Un cop dins de l'aplicació web, amb un àmbit seleccionat podrem configurar el projecte.

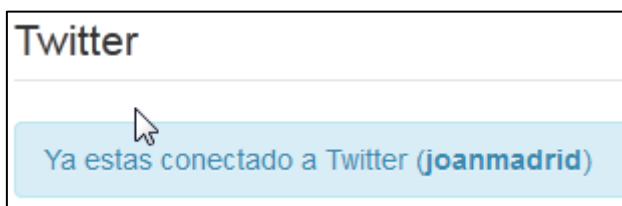
Accedim a la secció "Gestionar".



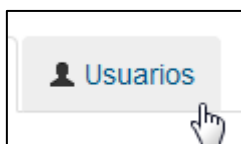
Primer de tot podem connectar-lo amb les API's que estiguin disponibles. Segons la xarxa social que seleccionem, aquest pas pot variar, però al final sempre ens demanaran el nostre usuari i una petició que haurem de validar.



Si tot ha funcionat correctament, tornarem a la pàgina de l'eina i ens avisarà que hem connectat correctament.



Ara, accedim a la següent secció "Usuarios".

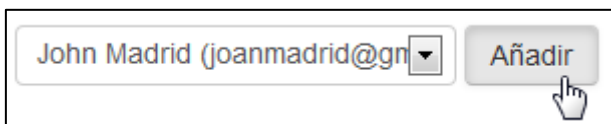


Des d'aquesta secció podem afegir usuaris. Veurem un llistat dels usuaris afegits i podrem: afegir, modificar permisos o treure'n.

Per modificar permisos hem de fer clic a les caixetes de selecció i a continuació donar-li al botó de guardar.



Si volem afegir un nou usuari, haurem de desplegar el selector i seleccionar a un altre usuari.



A form with a text input field containing "John Madrid (joanmadrid@gn)" and a dropdown arrow on the right. To the right of the input field is a button labeled "Añadir". A mouse cursor is pointing at the "Añadir" button.

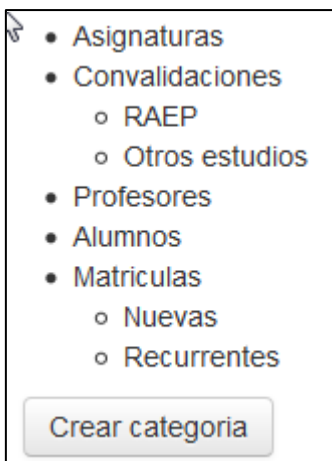
Un cop tenim aquests passos fets, podem continuar a configurar les nostres "Categorías" i "Tipologías" que ens ajudaran a determinar com podem classificar els missatges entrats de l'eina. El procediment es el mateix per els dos i només varia en el cas de categorías donat que permet escollir una categoría pare per a que sigui una subcategoría d'aquesta.

Primer accedim a la secció:



A navigation menu with two items: "Categorías" and "Tipologías". The "Categorías" item is highlighted with a grey background and a mouse cursor is pointing at it.

Un cop dins veurem el llistat de categorías.

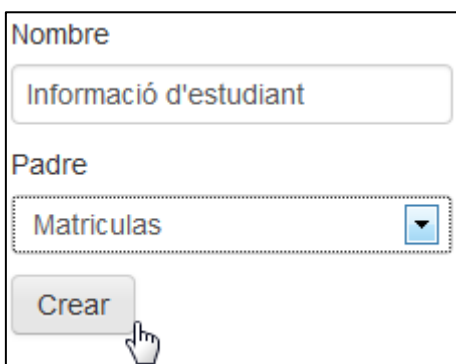


A list of categories with a "Crear categoría" button at the bottom. The categories are:

- Asignaturas
- Convalidaciones
 - RAEP
 - Otros estudios
- Profesores
- Alumnos
- Matriculas
 - Nuevas
 - Recurrentes

Crear categoría

Podem crear-ne de noves anant a la secció de creació. Omplim les dades i opcionalment seleccionem si es tracta d'una categoría filla, és a dir, triem el seu pare.



A form for creating a category. It has two sections:

Nombre
Informació d'estudiant

Padre
Matriculas

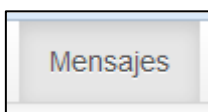
Crear

A mouse cursor is pointing at the "Crear" button.

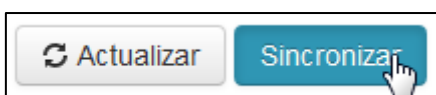
En aquest cas, informació d'estudiant es una subcategoría de "Matriculas".

20.4. Classificant els primers missatges

Al apartat de "Mensajes" trobarem un seguit d'eines.



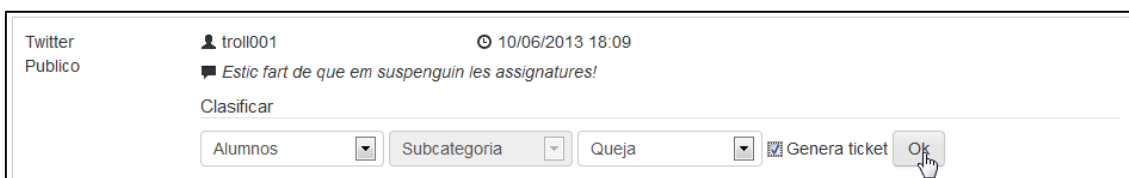
Per classificar missatges, necessitem que hi hagin alguns creats. Podem crear-ne manualment omplint un senzill, tot i que en una situació d'utilitat real segurament voldrem sincronitzar automàticament a través de les API's connectades de les xarxes socials. Mitjançant el botó "Sincronizar" podem rebre nous missatges.



Un cop clicat, ens avisarà de quants missatges ha sincronitzat i automàticament s'actualitzarà la secció, de la mateixa manera que si haguéssim clicat a "Actualizar".

Cada missatge està compostat d'una fitxa on es detallen alguns elements d'aquest:

- Origen: creat manualment, sincronitzat de Twitter, Facebook, etc..
- Visibilitat: públic o privat (segons la xarxa social)
- Usuari: usuari que ha generat el missatge
- Data: moment de la creació del missatge a la xarxa social.
- Text: cos del missatge
- Clasificar: opcions que disposem per a classificar el missatge. Aquí veurem el resultat d'haver configurat el projecte prèviament.



A la part inferior podem seleccionarà primer de tot la categoria. A continuació, i en el cas de que tingui subcategories, escollirem una. A continuació podrem escollir una tipologia que haurem configurat al projecte.

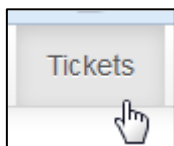
Finalment, ens permet marcar si aquest missatge genera ticket. Un ticket es necessari si d'un missatge se'n deriva una tasca, com per exemple consultar amb el servei d'atenció del campus.

Els missatges classificat aniran desapareixent del llistat. Tot i això podem seguir l'històric a través de la secció "Archivo", on també podrem tornar qualsevol missatge a la classificació en cas d'equivocació.



20.5. Treballant amb Tickets

Canviem de secció i anem a la de "Tickets".



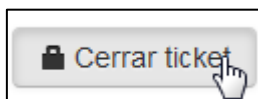
Aquí disposarem d'un llistat de tickets pendents de resoldre. Entenem que un ticket normalment es genera a partir d'un missatge que precisa fer una tasca, a part de la classificació.

Des de el llistat podem veure una visió global dels tickets, tot i això, podem accedir a un ticket en concret per veure'l en detall i poder gestionar-lo.

D'un ticket podem veure:

- ID: nombre identificador del ticket
- Creado: data de creació del ticket
- Estado: estat del ticket, pot estar Obert i Tancat.
- Asignado: usuari actual que està assignat a la resolució del ticket.
- Cos del ticket
- Accions

Entre les accions podem tancar el ticket, suposant que ja ha estat resolt o que no es necessària la resolució.



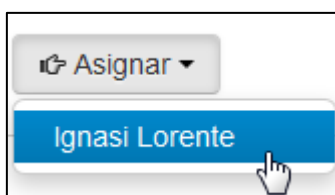
Mitjançant aquesta acció tornarem al llistat i el ticket ja no hi serà.

També podem comentar, els comentaris poden ser útils per mantenir un petit diari de la resolució o per comunicar-se amb altres persones de l'equip que participin en el ticket.



The screenshot shows a ticket management interface. At the top, there are three buttons: "Cerrar ticket" (with a lock icon), "Comentar" (with a speech bubble icon), and "Asignar" (with a person icon and a dropdown arrow). Below these buttons, a comment from "Joan Madrid (10/06/2013 19:51):" is visible, with the text "S'haura de resoldre quan abans millor, el pobre noi porta temps demanant ajuda amb el tema :(". Underneath the comment is a text input field labeled "Mensaje" containing the text "Com va el tema?". Below the input field is an "Enviar" button.

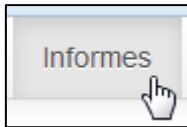
Finalment, podem assignar el ticket a un altre usuari del projecte, de manera que ens pugui ajudar a resoldre'l o simplement que vulguem repartir la feina.



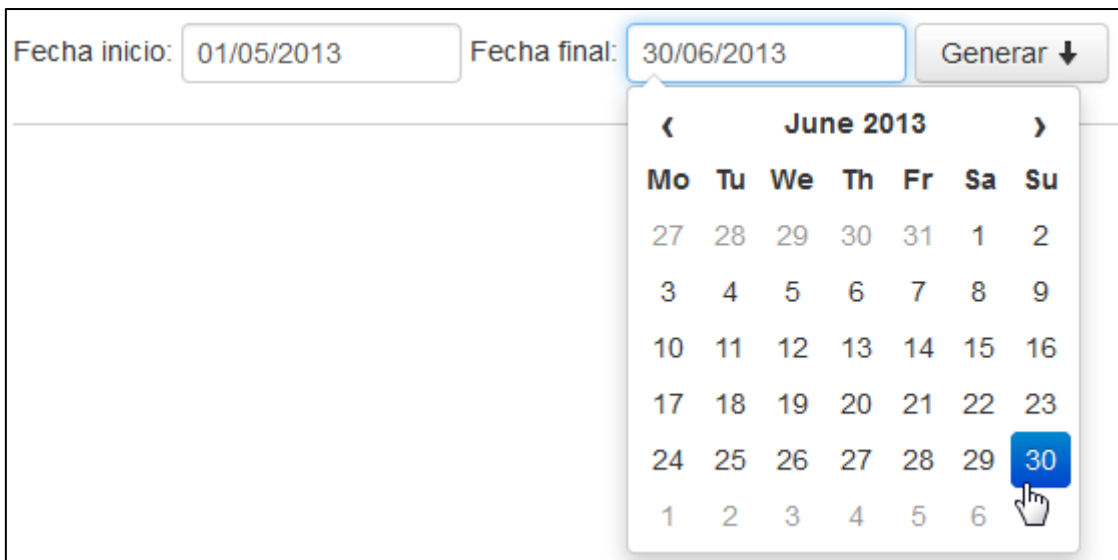
The screenshot shows a close-up of the "Asignar" button from the previous image. The dropdown menu is open, displaying a list of users. The user "Ignasi Lorente" is highlighted in blue, and a mouse cursor is pointing at it.

20.6. Resultats i informes

Finalment, després de mesos de feina volem veure i generar estadístiques o informes sobre els resultats i la classificació. Ho farem a partir de la secció "Informes".



En aquesta secció disposarem d'un formulari on podem indicar la data inicial i la data final del rang de dies que volem generar les estadístiques, de manera que puguem generar informes totalment adaptats a la temporalitat que es desitja.

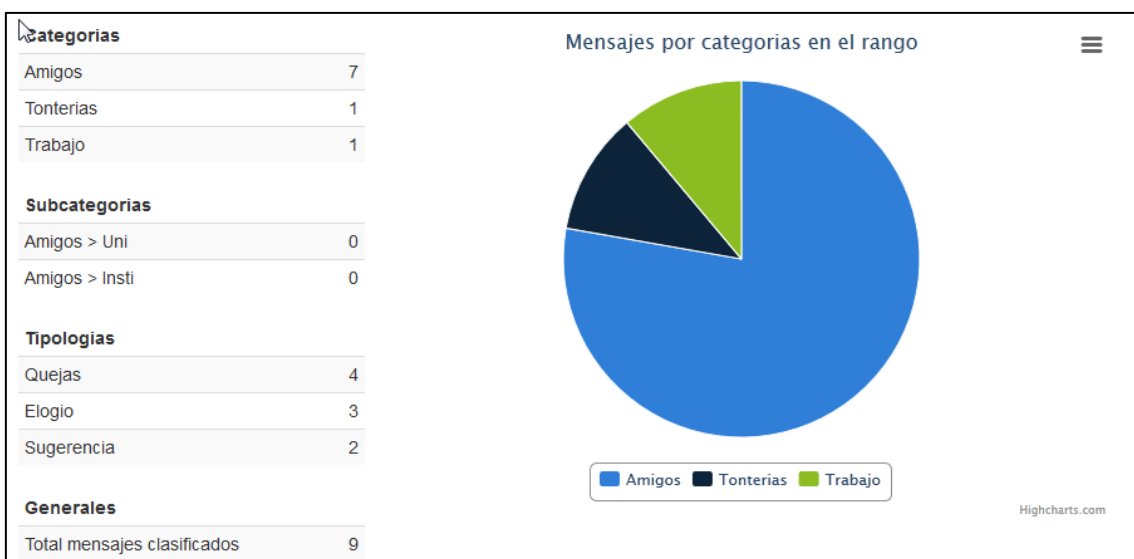


Fecha inicio: 01/05/2013 Fecha final: 30/06/2013 Generar ↓

Calendar de June 2013:

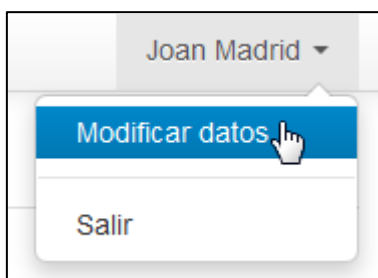
Mo	Tu	We	Th	Fr	Sa	Su
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	

Podem navegar entre dies, mesos i anys mitjançant el selector de dates. Un cop seleccionades les dates farem clic al botó "Generar", i després d'un procés d'uns segons se'ns generarà automàticament un informe amb estadístiques i gràfiques que ens ajudaran a veure el treball realitzat.

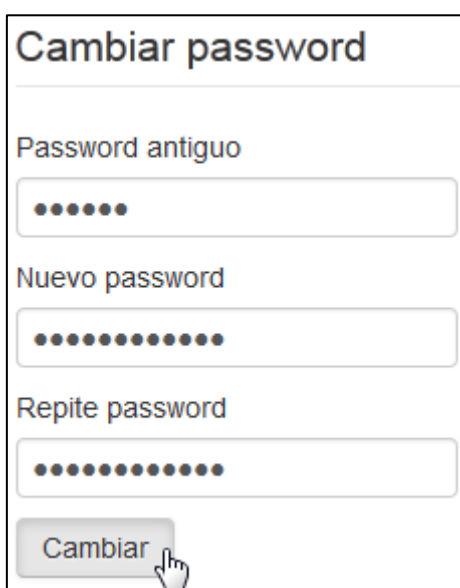


20.7. Canvi de password

En el cas que vulguem canviar la paraula clau del nostre usuari podem accedir a les preferències d'usuari fent clic sobre el nostre nom, a dalt a la dreta de l'aplicació.



Un cop aquí dins, haurem d'omplir el formulari amb la paraula de pas actual (per seguretat) i a continuació escriure dues vegades la nova clau.

A screenshot of a form titled "Cambiar password". It contains three input fields: "Password antiguo" (with 6 dots), "Nuevo password" (with 10 dots), and "Repíte password" (with 10 dots). At the bottom, there is a "Cambiar" button with a mouse cursor pointing to it.

Una vegada haguem sortit de l'aplicació i tornem a entrar ho haurem de fer amb la nova paraula de pas que prèviament hem modificat.



21. Bugs

Tot i l'esforç per part dels programadors, analistes i totes les tècniques que hem descrit anteriorment per fer test i per evitar bugs, podem estar segurs que es inevitable que apareguin errors i comportaments no esperats.

Es per això que hem de focalitzar els nostres esforç en organitzar un bon sistema de resolució de problemes i incidències ja que seria molt greu pensar que el sistema no en tindrà.

Donada la poca grandària del projecte i que només hi participa un mateix programador ens hem decantat per fer servir *Trello*⁸ alhora d'organitzar-nos.

Les columnes que farem servir seran: *Ideas*, *Known Issues* (*bugs* i problemes detectats), *In Progress* (tasques que s'estan arreglant), *Testing* (tasques acabades que cal provar bé), *Done* (tasques acabades i provades que entraran en la següent versió publica).

Exemple:

<i>Ideas</i>	<i>Known Issues</i>	<i>In progress</i>	<i>Testing</i>	<i>Done</i>
<i>Poder crear comandes i automatismes per categoritzar automàticament</i>	<i>Dona error al clicar el enllaç de "Dades personals"</i>	<i>Problema al inserir tickets manualment</i>	<i>Error al configurar la API de twitter</i>	<i>Error al fer login</i>

Taula 1. Exemple de taula de bugs

21.1. Llistat de bugs

A continuació detallo l'estat actual, en el moment de la entrega, quin es l'estat dels bugs:

Activar Firewall al servidor i protegir ports	Resolt
Configurar .htpasswd per a restringir l'accés	Resolt
Les "x" dels missatges d'ajuda no funcionen	Resolt
Configurar el virtualhost amb el domini smacly.com	Resolt
Guardar registre en els tickets: assignacions, canvis d'estats i comentaris	Resolt
Fallen botons d'assignació de missatges un cop es recarrega fent servir "Actualizar"	Resolt
No sincronitza bé missatges nous a Twitter	Resolt
Les opcions de categories i tipologies han d'estar restringides per projecte	Resolt
Activar la secció (menú i submenú) segons estat actiu	Resolt
Ordenar per data d'arribada els missatges	Resolt
Al crear un nou projecte automàticament et canviï el context en el que estàs	Resolt
La sessió del projecte expira massa ràpid	Pendent
No aplica permisos d'usuari a les seccions	Pendent
Error en generació de reports	Pendent
No permetre navegar sense sessió de projecte	Pendent

Taula 2. Llistat de bugs

⁸ *Trello* es una aplicació online que ens permet gestionar tasques mitjançant llistats i etiquetes. Seria la idea d'una paret plena de *post-its* de colors (cadascun d'un tema diferent) ordenats en columnes, portat al mon digital (trello.com)

22. Projectió a futur

Podríem dir que l'aplicació es troba en un estat de "beta permanent" que comunament s'utilitza avui en dia per definir aquelles aplicacions, web normalment, que estant en constant canvi i creixement per oferir millores, i donat el seu estat evolutiu constant, és possible que hi hagi algunes coses que fallin o siguin millorables.

A nivell de projectió de futur clarament podem veure tres nivells d'aproximació: futur immediat, futur pròxim, futur llunyà. Aquestes etapes defineixen el camí que es vol arribar en el projecte, tot i que com més temporalment allunyar estigui és més fàcil de caure en desviacions.

22.1. Futur immediat

En una primera etapa de futur immediat que començaria des de el moment de l'entrega del projecte fins la durada d'uns pocs mesos, s'atendran aquelles idees i mòduls que o han faltat programar per falta de temps o es consideraven innecessaris per una primera "release" de l'aplicació.

- A nivell de codi de l'aplicació
 - Tractament d'error
 - Millorar seguretat d'usuari
 - Validar entitats (taules i relacions de base de dades)
 - Optimitzar velocitat de l'aplicació
- A nivell de funcionalitats
 - Permetre mostrar logotip per projecte
 - Es pugui sortir del vincle amb la API de Twitter
 - Vista prèvia dels tickets al llistat
- A nivell visual
 - Afegir logotips de API's i xarxes socials
 - Afegir logotip de l'aplicació
 - Modificar colors i estructura segons imatge de marca
- A nivell d'usabilitat
 - Textos d'ajuda per tota l'aplicació
 - Breadcrumb

22.2. Futur pròxim

Si enfoquem a un futur més pròxim, i pensem en coses bàsiques per d'un cost d'hores de desenvolupament altes podríem discernir:

- Afegir API de Facebook a l'aplicació
- Capacitat de respondre missatges (twits i facebook) desde la web
- Poder filtrar tickets per assignats i projectes
- Sincronització automàtica de missatges
- Validació de formularis a nivell de client
- Millorar feedback de l'aplicació

22.3. Futur llunyà

Finalment, enfocant a un futur llunyà que podria tractar d'uns 6 mesos a un any, veiem que podríem seguir aquest camí i enfocar:

- Limitat certes accions de l'aplicació, diferents comptes premium (monetitzar)
- Permetre filtres intel·ligents per classificar missatges
- Vistes transversals d'ajuda per usuaris de missatges (informació extra)
- Exportació de dades en formats de treball (excel per exemple)
- Millorar generació d'informes i permetre personalització

Un bon pla de futur es poder créixer amb els nostres usuaris. Mitjançant el seu feedback siguem capaços d'enfocar la dedicació i esforç a crear una eina millor i més útil de manera que gràcies al seus "know how" puguem evolucionar l'aplicació perquè s'adapti a les necessitats del mercat on volem penetrar.

23. Pressupost

S'ha fet la divisió de pressupostos d'acord amb els costos dels recursos humans durant tota la etapa d'anàlisi i desenvolupament, i del cost de software, hardware i altre material.

Concepte	Sou brut anual	Comentari
Cap de projecte	30.000€	Necessari durant els 6 mesos del projecte
Director de producte	27.000€	Necessari durant els 6 mesos del projecte
Programador	24.000€	Només serà necessari durant els 4 últims mesos
Dissenyador	22.000€	Estarà dos mesos en salari
Maquetador	22.000€	Només serà necessari els 3 últims mesos

Taula 3. Pressupost de RRHH

Per la part de recursos físics i digitals:

Concepte	Cost	Comentari
Servidor	12€ mensuals	Kemsirve mKS 2G
Llicència PHPStorm	100€	Pagament únic
Domini .es	10€ mensuals	-
Local i despeses	450€	Inclòs electricitat, equips, etc.

Taula 4. Pressupost de material

Donat que la durada inicial del projecte serà de 6 mesos, el cost aproximat en recursos de persones seran **47.500€** i **1932€** en material.

Una vegada arribada la entrega del projecte serà necessari aixecar una ronda de inversors i presentar un nou pla de despeses econòmiques i traçar un nou pla de desenvolupament de millores i manteniment de l'aplicació.

24. Anàlisi de mercat

Els departaments d'atenció al client porten anys donant servei als usuaris. Va començar com un àmbit intern i petit, on la tenda del barri atenia les consultes post-venta fins a passat pel correu postal i l'atenció telefònica finalment.

Amb la popularització dels ordinadors i Internet es van obrir nous canals, com el correu electrònic o els formularis de contacte via web.

Actualment podem afirmar que un nou canal ha arribat per a quedar-se, i les xarxes socials son una via més per atendre les consultes dels usuaris o clients.

Petites i grans empreses s'han adonat de la necessitat d'estar a les xarxes socials, guanyar visibilitat i empatitzant amb els seus clients. Aquests, a la vegada, s'han adonat que poden parlar amb les seves marques/empreses preferides, poden preguntar-li, suggerir i inclús queixar-se.

En aquest context, mesurar la interacció de la marca amb els seus usuaris en medis socials es converteix en una necessitat per a l'empresa, que podrà determinar el temps, els recursos i la formació necessària per a oferir un bon servei al client, així com conèixer l'èxit de les seves inversions. En resum, es tracta d'obtenir resultats.

Es per aquesta raó que SMAC ofereix una solució integrada per a petites i mitjanes empreses, que interactuen en les xarxes socials i desitgen mesurar els resultats dels seus esforços de l'Atenció al Client en les xarxes socials.

24.1. Estudi de la competència

A continuació una taula amb la principal competència i un anàlisi:

	Serveis	Idioma	Suport	Pricing
Zendesk	- Conversacions amb organització interna i externa - Compatible amb múltiples dispositius mòbils i portàtils - Creació d'informes personalitzats	ES	- Web - Telèfon - Correu - Chat - Facebook - Twitter	En funció del nº d'agents i serveis: · 24-99\$ lic/mes
Freshdesk	- Funcionalitats molt personalitzables - Pla de preus molt flexible - Personalització de camps en els tickets - Sugerència intel·ligents en funció del contingut	ES (en funció del pla de pagament escollit)	- Correu - Telèfon - Facebook - Twitter	En funció del nº d'agents i serveis: · 16-40\$ lic/mes · Versió gratuïta limitada · Opció de passes temporals
Conversocial	- Assignació de diferents rols i permisos als agents - Integració dels grups de treball amb múltiples opcions	EN	- Facebook - Twitter	En funció del nombre de missatges entrants: mínim 200\$ al mes per 1.500 missatges.

Taula 5. Estudi de la competència

25. Màrqueting i vendes

25.1. Pla de Branding

L'esperit de SMAC és el de convertir-se en un company de viatge de les petites i mitjanes marques que s'aventuren en el món de les xarxes socials. SMAC hauria de ser percebut com una eina molt senzilla de fer servir, amigable i amb un equip humà darrere disposat a resoldre qualsevol dubte o consulta i que parlar el teu mateix idioma. En altres paraules, SMAC és una eina humana.

El nom de SMAC neix de les sigles de *Social Media Atención al Cliente*, que defineix explícitament l'objecte del servei. Al ajuntar les sigles, obtenim un nom senzill de pronunciar, fàcil de recordar i atractiu, de manera que els nostres futurs clients el recordin.

Els colors que donaran vida a SMAC seran el taronja i el gris mig. Després d'observar les webs de la competència, veiem que els tons grisos, blaus i verds son els que més imperen, pel que intentarem allunyar-nos d'aquesta imatge. Escollim un color sobri i elegant com el gris però amb tons ataronjats i càlids per denotar la sensació de proximitat amb els nostres possibles clients.

25.2. Pla de promoció

La forma de promocionar el producte serà contactant directament amb agències de *social media*, consultories i clients finals amb presència de xarxes socials.

Per portar a terme això, serà precís establir un llistat de petites i mitjanes empreses que ja disposin d'una pagina oficial de *Facebook*, així com agència i consultories.

Es tractaria de contactar amb els *brand managers* i *Community managers* de les comptes seleccionades, oferint una prova del nostre producte. Se'ls dirigiria a un vídeo curt i un document explicatiu de les característiques i possibles funcionalitats i utilitats de l'eina i, en cas que fos necessari, una formació presencial inicial.

25.3. Política de preus

Els preus seran sensiblement diferents a la nostra competència. Si bé és cert que s'estableixen diferents plans en funció de si disposen amb més o menys funcionalitats, inicialment també hi hauria diferents tarifes en funció del nombre d'agents a la xarxa i del nombre de missatges rebuts.

Tipus de pla	Nombre d'agents	Quantitat de missatges	Preu
Free	1	Fins a 100 missatges rebuts	0€/mes
Simple	Fins a 5	Fins a 500 missatges rebuts	10€/mes
Plus	De 6 a 20	Fins a 1500 missatges	18€/mes
Premium	Més de 20	Fins 5000 missatges	35€/mes

Taula 6. Preus de SMAC

25.4. Estratègia de venda

A més a més dels plans de promoció, en una primera fase de llançament ens interessaria donar visibilitat al projecte i guanyar imatge al mercat. D'aquesta manera, en un primer moment, tractaríem de col·laborar amb empreses oferint el producte de forma gratuïta a canvi de contar amb una retroalimentació de cara a detectar bugs i implementar millores al producte.

Més endavant, al sortir de la fase beta, es proposaria un mes de prova gratuïta aquells que estiguessin interessats. D'aquesta manera podem seguir obtenint feedback i comptar amb una base de dades de possibles clients.

Seria interessant també realitzar una petita inversió de Facebook Ads i Google Ads de cara a arribar a un públic molt més especialitzat, que potser podria estar interessats.

26. Conclusions

Després de tants anys com a estudiant de la UOC haig de confessar que havia arribat a un punt on moltes de les pràctiques que havia d'entregar em resultaven pesades o moltes vegades no els hi trobava raó de ser, per el meu cap passava alguna vegada el pensament "això no em serveix de res", segurament fruit del cansament donat que sempre he compaginat els estudis amb una jornada laboral completa.

Aquesta confessió l'escric perquè d'alguna manera la realització d'aquest projecte de final de grau m'ha servit per adonar-me'n que estava equivocat. Quan per primera vegada els consultors m'han deixat sol davant del "perill" per enfrontar-me a un problema que jo havia de crear, modificar i donar-li la dificultat que jo cregués perquè fos competent, ha sigut en aquest període que m'he adonat i he valorat realment tot l'ensenyament que he absorbit durant aquests anys.

La finalització d'aquest projecte m'ha donat l'essència que inclús moltes vegades es perd al dia dia de la feina, el poder crear, pensar un projecte des de zero, participar en totes les etapes: des de les primeres pinzellades a nivell d'esbossos i idees, passant per la planificació del projecte i el anàlisis de les tecnologies fins arribar al punt de desenvolupament i producció. En aquest transcurs he anat fent servir els coneixements de quasi totes les assignatures que d'alguna manera m'ha servit com de redempció dels meus pensaments.

SMAC ha sigut una eina que per primera vegada he creat jo de principi a fi, on m'he barallat amb mi mateix per escollir la millor tecnologia, en planificar les fites d'entrega. A la vegada he agafat el rol del client, on demana tots els requeriments i pantalles que m'agradaria veure. Un altre jo, que feia de cap de projecte lluitava amb el client per analitzar i consensuar els requisits per donar-li forma, restringir algunes de les idees i fer-les mes plausibles de cara a un desenvolupament web. Finalment el meu tercer jo més programar i enginyer s'ha posat mans a l'obre per crear aquesta aplicació.

Durant tot aquest semestre el projecte ha sofert moltes modificacions, a mesura que avançava i em trobava en problemes he hagut d'agafar altres rutes i camins, en algunes m'he tronar a entrebancar però, ara que ja he acabat puc afirmar que em sento satisfet amb aquesta creació, fins a un punt que més endavant m'agradaria continuar-la com a un projecte propi i qui sap si el dia de demà aconseguiré algun client real.

Amb tot això puc concloure i reiterar la meva satisfacció personal i agrair l'oportunitat (obligada) de la realització d'un projecte de final de grau que trobo molt necessari i útil per qualsevol estudiant de grau, des de els que mai han tingut experiència laboral real com els que com jo, portem anys treballant però mai havíem pogut gaudir d'un projecte de principi a fi.

Annex 1. Lliurables del projecte

Codi

Codi de l'aplicació (sense llibreries de tercers). Inclou:

- Symfony 2.2
- Codi desenvolupat pel projecte
- LESS, arxius Javascript i imatges de la web
- Plantilles de totes les pantalles de l'eina

/PAC_FINAL_prj_MadridLópez_Joan/

Documentació

Presentació lliure en format Power Point

PAC_FINAL_prs_MadridLópez_Joan.pptx

Autoinforme d'avaluació

PAC_FINAL_auto_MadridLópez_Joan.pdf

Dades de connexió

PAC_FINAL_online_MadridLópez_Joan.txt

Annex 2. Codi font (extractes)

El primer extracte és per una de les parts més complexes del projecte que ha sigut connectar-me amb la API de Twitter mitjançant el seu servei REST, a través d'un bundle que s'ha creat per Symfony únicament per aquesta tasca. La complexitat ha vingut donada per haver d'integrar en una sola acció 3 components: el meu projecte, el servei de Twitter per Symfony i la API.

La primera funció es la que crida l'acció de l'usuari quan fa clic al botó per connectar-se a Twitter per primera vegada:

```
/**
 * Conecta por primera vez con Twitter y pide los token
 *
 * @return RedirectResponse
 */
public function connectAction()
{
    $em = $this->getDoctrine()->getManager();
    $twitter = $this->get('fos_twitter.service');
    $request = $this->getRequest();
    $session = $request->getSession();

    //petición de la URL para conexión del servicio
    $authURL = $twitter->getLoginUrl($request);
    $response = new RedirectResponse($authURL);

    //almaceno información de la petición en la base de datos
    $proj = $session->get('smac_context');
    $project = $em->getRepository('SmacProjectBundle:Project')->find($proj);
    $twitterRequest = new TwitterRequest();
    $twitterRequest->setDate(new \DateTime());
    $twitterRequest->setType(TwitterRequest::TWITTER_REQUEST_TYPE_CONNECT);
    $twitterRequest->setProject($project);
    $em->persist($twitterRequest);
    $em->flush();

    return $response;
}
```

A continuació la funció que rep les dades de l'API una vegada que l'usuari ha introduït les seves dades i ha permès la connexió entre SMAC i Twitter:

```
/**
 * Twitter devuelve el Token del usuario aceptado y lo guardamos
 en base de datos
 *
 * @return RedirectResponse
 * @throws
 \Symfony\Component\HttpKernel\Exception\NotFoundHttpException
 */
public function connectCallbackAction()
{
    $request = $this->getRequest();
    $token = $request->query->get('oauth_token');
```

```

$verifier = $request->query->get('oauth_verifier');
$session = $request->getSession();
$em = $this->getDoctrine()->getManager();

$proj = $session->get('smac_context');

//verificamos el token recibido con el servicio
$twitter = $this->get('fos_twitter.service');
$accessToken = $twitter->getAccessToken($token, $verifier);

//creo la entidad Token para guardarlo
$twitterToken = new TwitterToken();
$twitterToken->setOauthToken($accessToken['oauth_token']);
$twitterToken-
>setOauthTokenSecret($accessToken['oauth_token_secret']);
$twitterToken->setUserId($accessToken['user_id']);
$twitterToken->setScreenName($accessToken['screen_name']);

//lo asocio al proyecto
$project = $em->getRepository('SmacProjectBundle:Project')-
>find($proj);

if (!$project) {
    throw $this->createNotFoundException('El proyecto no
existe');
}

//lo asocio al proyecto
$project->setTwitterToken($twitterToken);

//lo guardamos en la base de datos
$em->persist($twitterToken);
$em->persist($project);
$em->flush();

$this->get('session')->getFlashBag()->add(
    'success',
    'Conexión con Twitter correcta'
);

return $this->redirect($this-
>generateUrl('smac_project_edit'));
}
    
```

Finalment com a part d'aquest "pack", l'acció que executa la sincronització de missatges des de l'API fins a la nostra base de dades documental MongoDB. Guardem un registre de l'últim missatge sincronitzar per evitar repeticions i peticions innecessàries:

```

/**
 * Sincroniza con mensajes de Twitter, segun las ultimas
peticiones.
 * Si $type es "old", sincronizará hacia atras
 * Si $type es "new", sincronizará mensajes nuevos
 *
 * @param $type
 * @return Response
 */
public function syncAction($type)
    
```



```

    {
        $request = $this->getRequest();
        $session = $request->getSession();
        $twitter = $this->get('fos_twitter.api');
        $twitter->host = "https://api.twitter.com/1.1/";
        $params = array();
        $em = $this->getDoctrine()->getManager();
        $dm = $this->get('doctrine_mongodb')->getManager();

        $proj = $session->get('smac_context');
        $project = $em->getRepository('SmacProjectBundle:Project')-
>find($proj);
        $token = $project->getTwitterToken();

        //miro la ultima sync
        $result = $em->getRepository('SmacAPIBundle:TwitterRequest')-
>findLastRequest($project, $token,
TwitterRequest::TWITTER_REQUEST_TYPE_MENTIONS);

        $first = $result['first'];
        $last = $result['last'];

        if ($type == 'old') {
            $param_name = 'max_id';
            $param_value = $last;
        } elseif ($type == 'new') {
            $param_name = 'since_id';
            $param_value = $first;
        }

        if ($result) {
            $params[$param_name] = (string)$param_value;
        }

        //conecto con la API de twitter para pedir menciones de
        timeline
        $twitter->setOAuthToken($token->getOAuthToken(), $token-
>getOAuthTokenSecret());
        $response = $twitter->get(
            'statuses/mentions_timeline',
            $params
        );

        //guardo mensajes en mongo
        $lastId = 0;
        $firstId = 0;
        $count = 0;

        if (count($response)>0) {
            foreach ($response as $tweet) {
                if ($count==0) {
                    $firstId = $tweet->id_str;
                }
                $message = new Message();
                $message->setProject($proj);
                $message->setUsername($tweet->user->name);
                $message->setUser($tweet->user->screen_name);
                $message->setDateCreated(new \DateTime($tweet-
>created_at));
                $message->setDateReg(new \DateTime());
                $message->setRaw(json_encode($tweet));
            }
        }
    }

```

```
        $message->setSource(Message::MESSAGE_SOURCE_TWITTER);
        $message->setText($tweet->text);
        $message-
>setVisibility(Message::MESSAGE_VISIBILITY_PUBLIC);
        $count++;
        $dm->persist($message);

        $lastId = $tweet->id_str;
    }

    $dm->flush();

    //logueo el request y last ID para futuras sync
    $twitterRequest = new TwitterRequest();
    $twitterRequest->setDate(new \DateTime());
    $twitterRequest-
>setType(TwitterRequest::TWITTER_REQUEST_TYPE_MENTIONS);
    $twitterRequest->setProject($project);

    if ($type == 'old') {
        $twitterRequest->setLast(($lastId > 0 ? $lastId :
$param_value));
        $twitterRequest->setFirst(($firstId > $first ?
$firstId : $first));
    } elseif ($type == 'new') {
        $twitterRequest->setFirst(($firstId > 0 ? $firstId :
$param_value));
        $twitterRequest->setLast(($lastId < $last || $last ==
0 ? $lastId : $last));
    }

    $twitterRequest->setToken($token);
    $em->persist($twitterRequest);
    $em->flush();

}

    return new Response(json_encode(array('synced'=>$count)), 200,
array('Content-Type'=>'application/json'));
}
```

Passant de bundle, ens trobem amb una de les accions insígnia de l'aplicació la modificació de missatges rebuts per la sincronització a les xarxes socials. Aquesta classificació emmagatzema i actualitza un document en MongoDB i si cal generar un registre a MySQL per un ticket. Aquesta acció es crida mitjançant AJAX:

```

    /**
     * Clasifica un mensaje, actualizando todos los datos mediante
    AJAX
     *
     * @return Response
     * @throws
    \Symfony\Component\HttpKernel\Exception\NotFoundHttpException
     */
    public function updateAction()
    {
        $request = $this->getRequest();
        $form = $request->request;

        //recibo todos los datos por POST
        $mId = $form->get('message-id');
        $cat = $form->get('category');
        $subcat = (int)$form->get('subcategory');
        $typ = $form->get('typology');
        $genTicket = $form->has('ticket');

        //obtengo el mensaje desde MongoDB
        $message = $this->get('doctrine_mongodb')
            ->getRepository('SmacMessageBundle:Message')
            ->find($mId);

        if (!$message) {
            throw $this->createNotFoundException('Mensaje no
    encontrado');
        }

        //actualizo los datos y guardo
        $dm = $this->get('doctrine_mongodb')->getManager();
        $message->setDateClassified(new \DateTime());
        $message->setCategory($cat);
        $message->setTypology($typ);
        if ($subcat > 0) {
            $message->setSubcategory($subcat);
        }
        $dm->persist($message);
        $dm->flush();

        //si me piden generar ticket genero entidad y la persisto
        if ($genTicket) {
            $request = $this->getRequest();
            $session = $request->getSession();
            $proj = (int)$session->get('smac_context');
            $em = $this->getDoctrine()->getManager();

            $project = $em-
    >getRepository('SmacProjectBundle:Project')->find($proj);
            $user = $this->get('security.context')->getToken()-
    >getUser();

            $ticket = new Ticket();
            $ticket->setUser($user);
    
```

```
$ticket->setAssigned($user);
$ticket->setDateReg(new \DateTime());
$ticket->setMessageUuid($mId);
$ticket->setProject($project);
$ticket->setStatus(Ticket::TICKET_STATUS_OPENED);
$em->persist($ticket);
$em->flush();
}

//devuelvo un "ok" para el ajax
return new Response(json_encode(array('saved'=>true)), 200,
array('Content-Type'=>'application/json'));
}
```

Per la banda de la vista, mostro el codi que executa en Javascript quan fem clic al botó de "classificar". Primer comprova que tots els camps necessaris estiguin plens i després executa la crida AJAX esperant una resposta:

```
/**
 * Clasificar mensaje
 */
$('body').on('click', '.classify-message .classify .btn',
function(e) {
    e.preventDefault();
    var ref = $(this);
    var message = $(this).closest('.classify-message');

    //si esta desactivado no avanza
    if($(this).hasClass('disabled'))
        return false;

    //si no ha seleccionado categoria no avanza
    var category = message.find('select[name="category"]');
    if (category.val()==0) {
        category.addClass('error');
        return false;
    } else {
        category.removeClass('error');
    }

    //si no ha seleccionado categoria no avanza
    var typology = message.find('select[name="typology"]');
    if (typology.val()==0) {
        typology.addClass('error');
        return false;
    } else {
        typology.removeClass('error');
    }

    var id = message.attr('data-id');
    var subcategory = message.find('select[name="subcategory"]');
    var genTicket = message.find('input[name="ticket"]');

    var data = [];
    data.push({'name':'message-id', 'value':id});
    data.push({'name':'category', 'value':category.val()});
    var attr = subcategory.attr('disabled');
```

```
        if (typeof attr !== 'undefined' && attr !== false && attr !==
'disabled') {
            data.push({'name':'subcategory',
'value':subcategory.val()});
        }

        if (genTicket.is(':checked')) {
            data.push({'name':'ticket', 'value':1});
        }

        data.push({'name':'typology', 'value':typology.val()});

        $.ajax({
            type      : "POST",
            url       : Routing.generate('smac_message_update'),
            data      : data,
            beforeSend: function() {
                //ref.addClass('disabled');
                message.slideUp();
            },
            success   : function(ret) {

            }
        });
    });
});
```

Ara ens trobem a la part de generació del informe entre dues dates. En aquestes tres funcions que us mostro a continuació fem les consultes necessàries tant a MySQL com a MongoDB per obtenir totes les dades i les classifiquem amb unes funcions que hem creat per la ocasió.

Es pot observar com dins del codi hi ha marcats uns "@todo" que es un marcador dinàmic que permeten configurar quasi tots els editors professionals (en aquest cas PHPStorm) de manera que pots guardar-te trossos de codi o fragments incomplets per a futures millores:

```
/**
 * Genera un informe comprendido entre dos fechas de todos los
 mensajes clasificados
 *
 * @return \Symfony\Component\HttpFoundation\Response
 */
public function generateAction()
{
    $request = $this->getRequest();
    $form = $request->request;
    $session = $request->getSession();

    //recibo las dos fechas
    $ini = $form->get('date-ini');
    $end = $form->get('date-end');

    $dini = \DateTime::createFromFormat('d/m/Y', $ini);
    $dend = \DateTime::createFromFormat('d/m/Y', $end);

    $proj = (int)$session->get('smac_context');

    //todos los mensajes clasificados entre las fechas
    $messages = $this->get('doctrine_mongodb')
        ->getManager()
        ->getRepository('SmacMessageBundle:Message')
        ->findClassifiedBetweenDates($proj, $dini, $dend);

    $classify = null;

    if (count($messages)>0) {
        $em = $this->getDoctrine()->getManager();
        //numero de tickets clasificados por categorias
        //...por subcategorias
        //...por tipologias
        //...por source
        //totales
        $classify = $this->classify($messages);
        $categories = $em-
>getRepository('SmacProjectBundle:Category')->findFromProject($proj);
        $subcategories = $em-
>getRepository('SmacProjectBundle:Category')-
>findSubcategoriesFromProject($proj);
        $typologies = $em-
>getRepository('SmacProjectBundle:Typology')->findFromProject($proj);

        //por visibilidad
        //@todo

        //tickets generados
    }
}
```

```

        //@todo

        //tickets resueltos
        //@todo

        //mensajes sincronizados
        //@todo

        //API Chart
        $ob = new Highchart();
        $ob->chart->renderTo('linechart');
        $ob->title->text('Mensajes por categorias en el rango');
        $ob->plotOptions->pie(
            array(
                'allowPointSelect' => true,
                'cursor'          => 'pointer',
                'dataLabels'       => array('enabled' => false),
                'showInLegend'     => true
            )
        );
        $ob->chart->width(600);
        $data = array();
        foreach ($categories as $category) {
            $data[] = array($category->getName(),
                $classify['categories'][$category->getId()]);
        }

        $ob->series(array(array('type' => 'pie', 'name' =>
'Mensajes', 'data' => $data)));

        //render final
        return $this->render(
            'SmacReportBundle:Default:report.html.twig',
            array(
                'ini' => $ini,
                'end' => $end,
                'classify' => $classify,
                'categories' => $categories,
                'subcategories' => $subcategories,
                'typologies' => $typologies,
                'chart' => $ob
            )
        );
    } else {
        return $this-
>render('SmacReportBundle:Default:report_no_results.html.twig');
    }
}

/**
 * Función que itera todos los mensajes y los va clasificando
segun tipos
 *
 * @param $messages
 * @return array
 */
private function classify($messages)
{
    $ret = array();
    $ret['categories'] = array();
}

```

```
$ret['subcategories'] = array();
$ret['typologies'] = array();
$ret['source'] = array();
$ret['total'] = count($messages);

foreach ($messages as $msg) {
    $this->increment($ret['categories'], $msg->getCategory());
    $this->increment($ret['subcategories'], $msg->
>getSubcategory());
    $this->increment($ret['typologies'], $msg->getTypology());
    $this->increment($ret['source'], $msg->getSource());
}

return $ret;
}

/**
 * Incrementa el valor de un array o lo crea si no existe
 *
 * @param $arr
 * @param $key
 */
private function increment(&$arr, $key)
{
    if (array_key_exists($key, $arr)) {
        $arr[$key]++;
    } else {
        $arr[$key] = 1;
    }
}
}
```


Els tickets de l'eina estan compostats per dues entitats (o taules), la primera compon el ticket unitàriament i les altres (relació 1 a N) son fils o "threads" d'accions que es fan sobre el ticket: comentar, canvis d'estat, assignacions entre usuaris, de manera que puc portar un registre de tot el que passa en una zona on interactuen varis usuaris.

Aquesta funció en concreta es la de comentar un ticket, però les altres son quasi idèntiques: obtinc el ticket, creo un nou fil i l'omple amb les dades i informació suficient:

```
/**
 * Comento un ticket y creo un Thread para guardar registro
 *
 * @param $id
 * @return \Symfony\Component\HttpFoundation\RedirectResponse
 * @throws
 \Symfony\Component\HttpKernel\Exception\NotFoundHttpException
 */
public function commentAction($id)
{
    $em = $this->getDoctrine()->getManager();
    $request = $this->getRequest();
    $form = $request->request;

    $ticket = $em->getRepository('SmacTicketBundle:Ticket')->find($id);
    $user = $this->get('security.context')->getToken()->getUser();

    if (!$ticket) {
        throw $this->createNotFoundException('Ticket no encontrado');
    }

    //creo la entidad Thread con toda la información
    $thread = new TicketThread();
    $thread->setDateReg(new \DateTime());
    $thread->setUser($user);
    $thread->setText($form->get('message'));
    $thread->setTicket($ticket);
    $thread->setType(TicketThread::TYPE_COMMENT);

    $em->persist($thread);
    $em->persist($ticket);
    $em->flush();

    $this->get('session')->getFlashBag()->add(
        'success',
        'Comentario registrado correctamente'
    );

    return $this->redirect($this->generateUrl('smac_ticket_view',
    array('id'=>$id)));
}
```

Finalment podem veure en el bundle d'usuari com registrem a un nou usuari al sistema. Es interessant denotar que es fa servir el sistema de Symfony per encriptar passwords i generar camps "salt" que proporcionen una capa més de seguretat i validació d'usuaris:

```
/**
 * Registro un usuario en el sistema
 *
 * @return
 \Symfony\Component\HttpFoundation\RedirectResponse|\Symfony\Component\
HttpFoundation\Response
 */
public function registerAction()
{
    $request = $this->getRequest();

    $user = new User();
    $form = $this->createForm(new UserType(), $user);

    if ($request->getMethod() == 'POST') {
        $form->bind($request);
        if ($form->isValid()) {
            $encoder = $this->get('security.encoder_factory')
                ->getEncoder($user);
            $user->setSalt(md5(time()));
            $encodedPwd = $encoder->encodePassword(
                $user->getPassword(),
                $user->getSalt()
            );
            $user->setPassword($encodedPwd);
            $em = $this->getDoctrine()->getManager();
            $em->persist($user);
            $em->flush();

            return $this->redirect($this-
>generateUrl('smac_user_register_landing'));
        }
    }

    return $this->render(
        'SmacUserBundle:Default:register.html.twig',
        array(
            'form' => $form->createView()
        )
    );
}
```

Annex 3. Llibreries i codi extern utilitzat

A continuació exposo el codi que pertany a l'arxiu *composer.json* que pertany al gestor de llibreries i codi extern de l'aplicació, en aquest llistat hi figuren totes les llibreries que fa servir l'aplicació.

```
"php": ">=5.3.3",
"symfony/symfony": "2.2.*",
"doctrine/orm": "~2.2,>=2.2.3",
"doctrine/doctrine-bundle": "1.2.*",
"twig/extensions": "1.0.*",
"symfony/assetic-bundle": "2.1.*",
"symfony/swiftmailer-bundle": "2.2.*",
"symfony/monolog-bundle": "2.2.*",
"sensio/distribution-bundle": "2.2.*",
"sensio/framework-extra-bundle": "2.2.*",
"sensio/generator-bundle": "2.2.*",
"jms/security-extra-bundle": "1.4.*",
"jms/di-extra-bundle": "1.3.*",
"doctrine/doctrine-fixtures-bundle": "dev-master",
"twitter/bootstrap": "dev-master",
"leafo/lessphp": "dev-master",
"jquery/jquery": "1.9.1",
"doctrine/mongodb-odm": "1.0.*@dev",
"doctrine/mongodb-odm-bundle": "3.0.*@dev",
"friendsofsymfony/jsrouting-bundle": "~1.1",
"kriswallsmith/assetic": "v1.1.0-alpha4",
"kertztwitteroauth": "dev-master",
"friendsofsymfony/twitter-bundle": "dev-master",
"ob/highcharts-bundle": "dev-master",
"friendsofsymfony/facebook-bundle": "1.1.*"
```

Tot i que bàsicament, si n'agrupen les llibreries per diferents paquets de software estariem parlant de:

- Symfony 2.2
- Doctrine 2.2
- Twitter Bootstrap 2.3.2
- LessPHP
- JQuery 1.9.1
- JSRoutingBundle 1.1
- Twitter OAuth
- Twitter Bundle
- Highcharts
- Facebook Bundle

Annex 4. Captures de pantalla

Entrada a l'aplicació, pantalla de Login:



The screenshot shows the SMAC login interface. At the top, the text "SMAC" is displayed in a large, bold, black font. Below this, the label "Usuario:" is followed by a white text input field. Underneath, the label "Contraseña:" is followed by another white text input field. A checkbox with a checkmark is labeled "No cerrar sesión". Below the checkbox is a grey button with the text "Acceder". At the bottom, there is a blue link that says "Crear una cuenta".

Selecció de projecte:



The screenshot shows the SMAC project selection interface. At the top, the text "SMAC" is displayed in a large, bold, black font. Below this, the text "Selecciona un proyecto para empezar a trabajar..." is shown. A list of four project options is presented, each with a right-pointing chevron icon: "UOC Atenció al Client", "Twitter de SMAC", "Personal", and "Proyecto personal". A mouse cursor is hovering over the "UOC Atenció al Client" option. Below the list is a grey button with the text "Crear un nuevo proyecto".

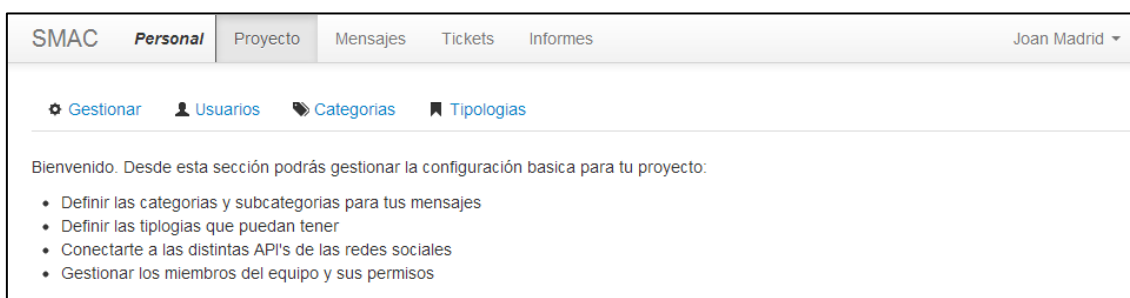
Crear un projecte nou:



The form is titled "SMAC" and contains the following elements:

- A "Name" label followed by a text input field.
- A "Description" label followed by a text area.
- A "Crear" button.
- A link: [Volver al listado de proyectos](#)

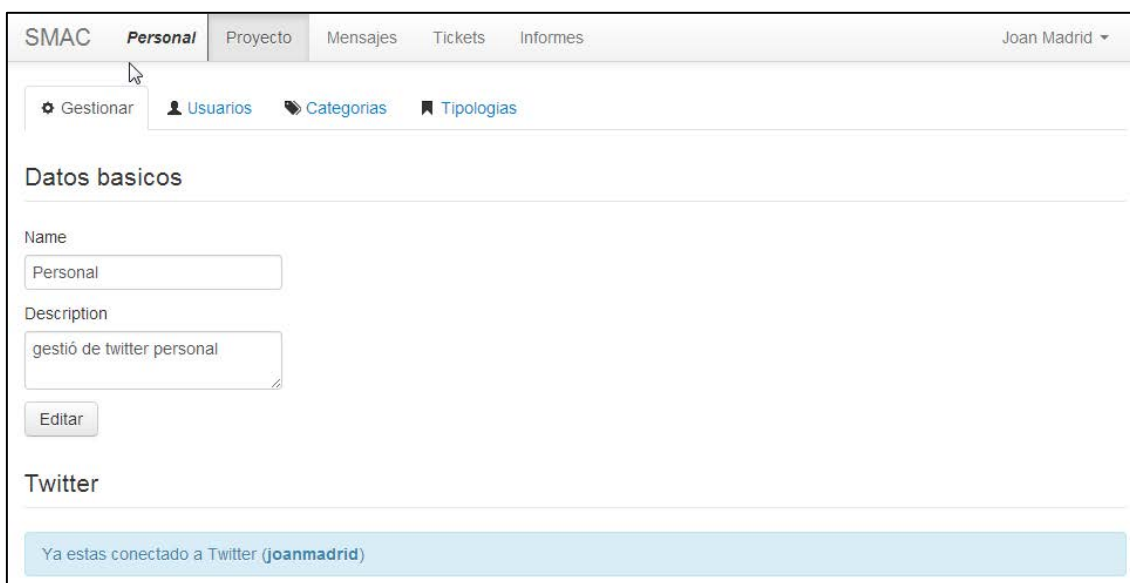
Entrada a l'aplicació:



The dashboard shows the following structure:

- Header: SMAC, Personal, Proyecto, Mensajes, Tickets, Informes, Joan Madrid
- Navigation: [Gestionar](#), [Usuarios](#), [Categorias](#), [Tipologias](#)
- Message: Bienvenido. Desde esta sección podrás gestionar la configuración básica para tu proyecto:
- List of tasks:
 - Definir las categorías y subcategorías para tus mensajes
 - Definir las tipologías que puedan tener
 - Conectarte a las distintas API's de las redes sociales
 - Gestionar los miembros del equipo y sus permisos

Gestionar proyecto:



The project management page shows the following details:

- Header: SMAC, Personal, Proyecto, Mensajes, Tickets, Informes, Joan Madrid
- Navigation: [Gestionar](#), [Usuarios](#), [Categorias](#), [Tipologias](#)
- Section: Datos básicos
- Form fields:
 - Name: Personal
 - Description: gestió de twitter personal
- Button: Editar
- Section: Twitter
- Status: Ya estas conectado a Twitter (joanmadrid)

Gestió d'usuaris del projecte:

The screenshot shows the SMAC interface with the 'Personal' profile and 'Proyecto' tab selected. The 'Usuarios' sub-tab is active. Below the navigation bar, there are tabs for 'Gestionar', 'Usuarios', 'Categorías', and 'Tipologías'. The main content area displays a table of users with columns for 'Usuario', 'Gestor', 'Analista', 'C.M.', 'A.C.', and action buttons. Two users are listed: Joan Madrid and Ignasi Lorente. At the bottom, there is a search input field containing 'John Madrid (joanmadrid@gr)' and an 'Añadir' button.

Usuario	Gestor	Analista	C.M.	A.C.	
Joan Madrid(joanmadrid@uoc.edu)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>
Ignasi Lorente(groomete@uoc.edu)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/>

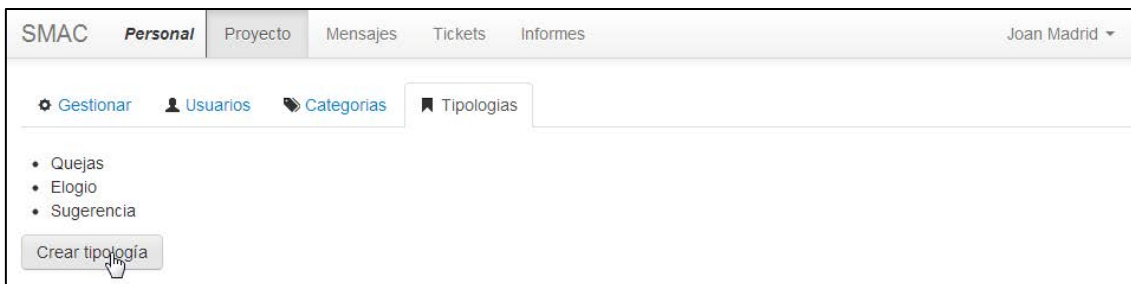
Gestió de categories del projecte:

The screenshot shows the SMAC interface with the 'Personal' profile and 'Proyecto' tab selected. The 'Categorías' sub-tab is active. Below the navigation bar, there are tabs for 'Gestionar', 'Usuarios', 'Categorías', and 'Tipologías'. The main content area displays a list of categories: 'Amigos' (with sub-items 'Uni' and 'Insti'), 'Tonterias', and 'Trabajo'. At the bottom, there is a 'Crear categoría' button.

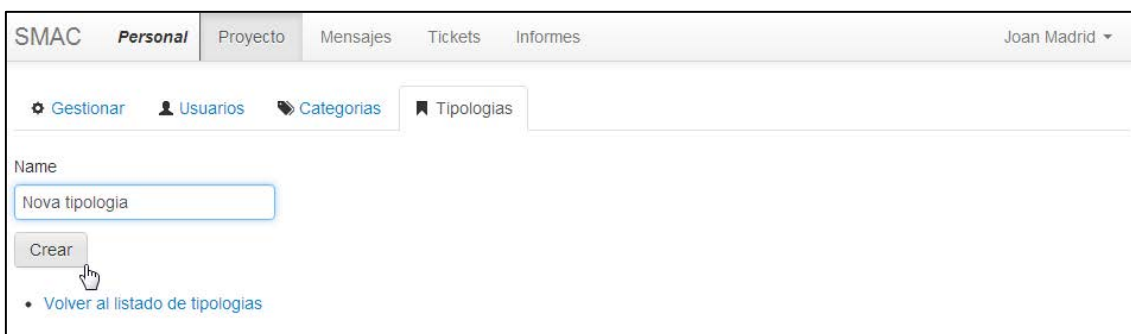
Nova categoria de projecte:

The screenshot shows the SMAC interface with the 'Personal' profile and 'Proyecto' tab selected. The 'Categorías' sub-tab is active. Below the navigation bar, there are tabs for 'Gestionar', 'Usuarios', 'Categorías', and 'Tipologías'. The main content area displays a form for creating a new category. The 'Nombre' field contains 'Nova categoria'. The 'Padre' dropdown menu is set to 'Amigos'. At the bottom, there is a 'Crear' button and a link to 'Volver al listado de categorías'.

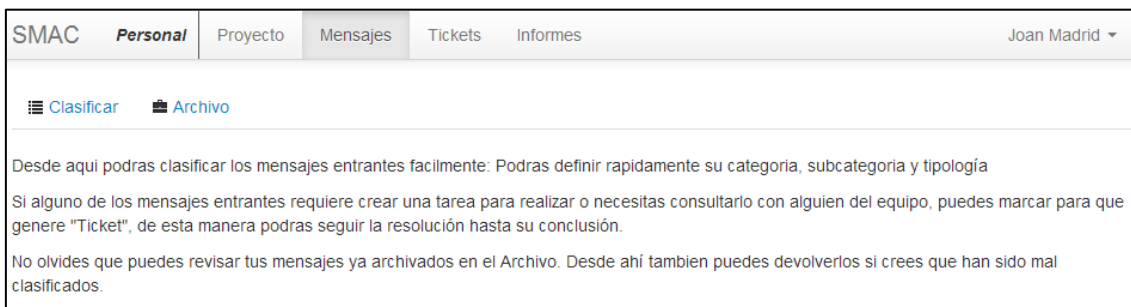
Gestió de tipologies del projecte:



Nova tipologia del projecte:



Secció de missatges:



Classificar missatges:

Arxiu de missatges classificats:

Plataforma	Fecha entrada	Fecha clasificado	Usuario	Visibilidad	Categoría	Subcategoría	Tipología	Ticket	Acciones
Twitter	26/05/2013 15:49	26/05/2013 16:05	Zoespinosa	Publico	Amigos	-	Elogio	-	🗑️
Twitter	26/05/2013 15:49	26/05/2013 16:05	xrsox	Publico	Amigos	-	Sugerencia	-	🗑️ Devolver a clasificar
Twitter	26/05/2013 15:49	26/05/2013 16:06	xrsox	Publico	Amigos	-	Quejas	-	🗑️
Twitter	26/05/2013 15:49	01/06/2013 22:58	xrsox	Publico	Amigos	-	Quejas	-	🗑️
Twitter	26/05/2013 15:49	26/05/2013 16:07	Chronoexpres	Publico	Trabajo	-	Quejas	-	🗑️
Twitter	26/05/2013 15:49	01/06/2013 22:58	Zoespinosa	Publico	Amigos	-	Sugerencia	-	🗑️
Twitter	26/05/2013 15:49	01/06/2013 22:58	Chronoexpres	Publico	Tonderias	-	Elogio	-	🗑️
Twitter	26/05/2013 15:49	01/06/2013 22:58	Chronoexpres	Publico	Amigos	-	Quejas	-	🗑️
Twitter	26/05/2013 15:49	02/06/2013 12:24	Chronoexpres	Publico	Amigos	-	Elogio	-	🗑️

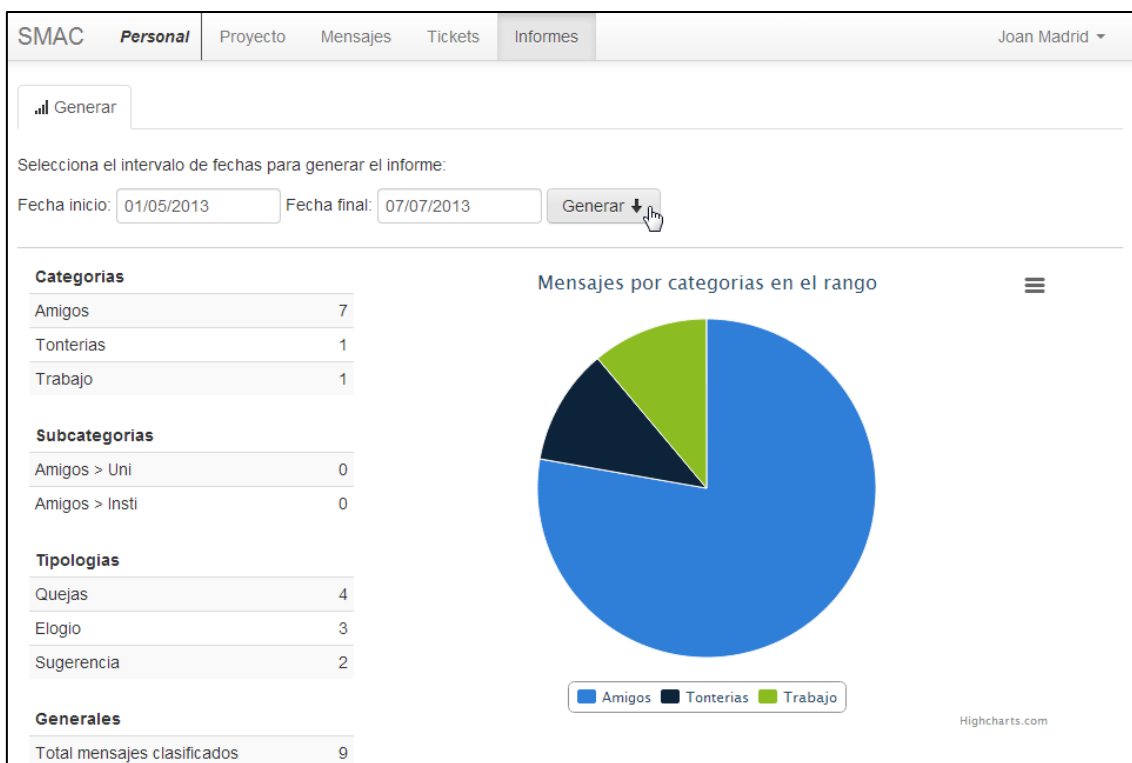
Secció de tickets (llistat):

SMAC		Personal	Proyecto	Mensajes	Tickets	Informes	Joan Madrid ▾
✕ Gestionar							
Tickets pendientes por resolver							
Ticket	Fecha entrada	Sumario	Estado	Acciones			
#1	26/05/2013 16:05	→ Generado desde clasificación	Abierto	👁			
#3	02/06/2013 12:24	→ Generado desde clasificación	Abierto	👁	Ver ticket		

Vista de ticket (gestió):

SMAC		Personal	Proyecto	Mensajes	Tickets	Informes	Joan Madrid ▾
✕ Gestionar							
Ticket	#3	Generado del mensaje:					
Creado	02/06/2013 12:24	👤 Chronoexpres	🕒	26/05/2013 15:49			
Estado	Abierto	🗨 @joanmadrid Por favor, mandanos el número de envío a través de DM para poder informarte.					
Asignado	Joan Madrid	Gracias.					
🗑 Cerrar ticket	💬 Comentar	👤 Asignar ▾					
Mensaje							
Nou comentar!							
Enviar							

Generació d'informes:



Gestió de preferències d'usuari (canvi de clau de pas):

The screenshot shows the SMAC user preference interface for changing a password. At the top, there are navigation tabs: SMAC, Personal, Proyecto, Mensajes, Tickets, and Informes. The user name 'Joan Madrid' is visible in the top right. The main heading is 'Cambiar password'. Below this, there are three password input fields: 'Password antiguo', 'Nuevo password', and 'Repite password'. Each field contains a series of dots representing masked characters. Below the input fields, there is a 'Cambiar' button with a hand cursor icon pointing to it.

Annex 6. One-page business plan

Nom comercial: SMAC

Resum comercial: eines de gestió i dinamització d'atenció a l'usuari de xarxes socials.

Model de negoci: *freemium*, oferint serveis bàsics gratuïts però un sistema de pagament per a una cartera més completa de clients

Expertise: un equip format per un expert en noves tecnologies amb un ampli coneixement i experiència en aplicacions web i nombrosos estils; l'altre membre de l'equip es tracta d'un professional en el *social media*.

Producte i serveis: un únic servei amb diferents de pla de pagament i funcionalitats.

Mercat: s'enquadra fins del mercat de serveis, d'àmbit nacional, enfocat a consumidors finals (per a *pymes*) o proveïdors (en el cas d'agències). Mercat de solucions multimèdia per a petites i mitjanes empreses.

Competència: tipus de mercat amb competència perfecta, existeixen multitud d'empreses que ofereixen serveis similars o molt semblants. La competència amb la que contem com a referència són *Freshdesk* i *Zendesk*, empreses molt similars, joves i fresques.

Pla de màrqueting: els objectius del pla de màrqueting en un primer lloc estan centrats en la visibilitat i obtenció de *leads* dels serveis a través dels plans de promoció. Més endavant contariem amb objectius de vendes, mensuals i fidelització dels que ja són clients. En resumides contes, es pretén oferir una solució integrada a les necessitats de mesura d'atenció a l'usuari en xarxes socials.

Inversió inicial i costos de curt i mitjà termini: es plantegen uns costos inicials de 60.000€ per als 6 primers mesos de projecte, contant els salaris, servidors, llicències i despeses d'oficina.

DAFO:

- **Debilitats:** l'estructura i recursos de l'empresa són limitats, de manera que les actualitzacions i millores podrien no ser immediates.
- **Amenaces:** competència nombrosa i molt assentada en el mercat. Existeixen empreses que porten anys tractant amb AC i podrien presentar actualitzacions dels seus sistemes centrats en xarxes socials. Els preus podrien ser més competitius en contra d'una estructura major.
- **Fortaleses:** gran coneixement del mercat, proximitat amb el possible client (tracte familiar) i llengua castellana nativa.
- **Oportunitats:** sistema de mercat proper i en alça en aquest tipus d'iniciatives, necessitat existent però no explotada i poc coneguda. Aposta per als negocis nacionals i no en les empreses estrangeres.

Annex 7. Glossari

Apache és un software de servidor web dels més usats avui en dia. Mitjançant mòduls es pot estendre la seva funcionalitat per suportar altres tecnologies com el PHP.

API és una interfície de programació d'aplicacions que bàsicament defineix una llibreria de funcions per a fer servir un servei per agents de tercers.

backend, en el món web es denomina aquella part més de gestió i oculta del usuari corrent on normalment s'efectuen operacions amb bases de dades i altres serveis.

Bitbucket és un servei web de la companyia Atlassian que permet, gratuïtament, disposar de repositori de control de versions per a codis de programació i scripts.

breadcrumb és una tècnica emprada en la usabilitat on informa a l'usuari en tot moment en quina secció es troba de la web. També es coneix com la tècnica del "fil d'Arianna".

bug és un defecte o error en un element de software (programari) generat normalment per una mala programació o descuit del programador.

bugfixing és la tasca que s'acostuma a desenvolupar en un procés de software on gent encarregada de la qualitat del producte es repassa tota l'aplicació intentant generar errors (bugs) per a que els programadors puguin resoldre'ls.

bundle en el món de la programació es un paquet de llibreries i funcionalitats agrupades que aporten un servei determinat a l'aplicació, però que a la vegada es pot exportar a altres aplicacions sense que variï el seu funcionament.

datacenter és la localització física dels servidors que allotgen webs. Normalment es tracten d'edificis amb tot tipus d'acondicionaments per a mantenir l'estabilitat dels servidors i procurar que aquests estiguin operatius el major % de temps possible.

DNS o Domain Name System és un sistema distribuït de nominalització d'ordinadors, serveis i altres recursos a internet de manera que direccionen un domini cap a unes direccions IP concretes.

Doctrine és un conjunt de llibreries que estan enfocades la persistència de serveis i dades. Segueix el patró ORM (Object Relational Mapper) que abstrau una base de dades independentment del tipus que sigui, de manera que el programador no s'ha de preocupar dels detalls i característiques de cadascuna d'elles.

framework és un conjunt de llibreries i funcionalitats de software que aporten normalment una manera de programa seguida de patrons de programari.

frontend es denomina a la part de la web visible a l'usuari que engloba tot el que seria el disseny i la part més vistosa d'aquesta.

git és un sistema distribuït de control de versions totalment gratuït i de llicència oberta utilitzat per fer un seguiment de versionat d'arxius de codis de programació.

jQuery és una de les llibreries més conegudes de Javascript que aporta moltes funcionalitats de selecció, animació i control que fa que programar en aquest llenguatge sigui molt més fàcil i intuïtiu.

LESS és un llenguatge dinàmic de fulles d'estil. Bàsicament permet afegir elements de la programació bàsiques (variables, controls de flux, funcions) en el àmbit de les fulles d'estil de manera que per un programador sigui molt més fàcil seguir una lògica semblant a la dels scripts. Al final s'acaba compilant per formar un arxiu CSS normal.

mockup és un esbós o prototip d'un producte que es vol dissenyar. En el mon web acostumen a ser prototips de cadascuna de les seccions de l'aplicació.

MongoDB és una base de dades que segueix el patró documental (i no relacional) on bàsicament es poden crear milions de documents amb un alt rendiment de lectura i escriptura però que a la vegada no disposa d'operacions relacionals.

MySQL és un sistema de base de dades relacional i molt utilitzat en qualsevol àmbit del mon del software.

REST és un estil d'arquitectura distribuïda de software molt comunament utilitzat en el món del desenvolupament d'API's. Bàsicament és el protocol de comunicació entre dues aplicacions.

Annex 8. Bibliografia (webgrafia)

- Varis autors (2001) **Creative Commons**. <http://www.creativecommons.org>
Consultat el 12/03/2013
- Begoña Oliver (2012) About.com "¿Qué es un Community manager y qué puede aportar a la empresa?" <http://tendenciasweb.about.com/od/el-trabajo-y-la-web/a/Que-Es-Un-Community-Manager-Y-Que-Puede-Aportar-A-La-Empresa.htm>
Consultat el 15/03/2013
- Gabor Ivan (2010). "LAMP on a Virtual Box" <http://lamp.gjxx-web.com/chapter1.html>
Consultat el 16/03/2013
- Anil Desai (2011). "What is a virtual Machine?" <http://searchservervirtualization.techtarget.com/definition/virtual-machine>
Consultat el 24/03/2013
- Derek Robert Price (2006). "CVS – Concurrent Versions System" <http://cvs.nongnu.org/>
Consultat el 24/03/2013
- Fundación CTIC (2003). "Perfiles de grupos de usuarios" <http://www.uiaccess.com/justask/es/users.html>
Consultat el 05/04/2013
- Alejandro Castells (2013). "Usabilidad vs User Experience vs UI (User Interface)" <http://blog.opicos.com/category/ux-usabilidad/>
Consultat el 05/04/2013
- Vangie Beal (2010). "The Differences and Features of Hardware and Software Firewalls" http://www.webopedia.com/DidYouKnow/Hardware_Software/2004/firewall_types.asp
Consultat el 06/04/2013
- Fabien Potencier (2013). "Security" <http://symfony.com/doc/2.1/book/security.html>
Consultat el 06/04/2013
- Fabien Potencier (2013). "Testing" <http://symfony.com/doc/master/book/testing.html>
Consultat el 06/04/2013
- Selenium IDE Dev's (2009). "The Selenium IDE Webpage" <http://docs.seleniumhq.org/projects/ide/>
Consultat el 06/04/2013
- (2011). "The Web Robots Pages" <http://www.robotstxt.org/>
Consultat el 20/04/2013
- Twitter Dev Team (2013). "Twitter API Documentation" <https://dev.twitter.com/docs/twitter-libraries>
Consultat el 29/04/2013

· Gonzalo Ayuso (2011). "**How to configure Symfony's Service Container to use Twitter API**" <http://gonzalo123.com/2013/02/04/how-to-configure-symfonys-service-container-to-use-twitter-api/>

Consultat el 30/04/2013

· FTASSI (2012). "**Work with bootstrap, assetic and less**"

<http://www.screenfony.com/blog/howto-symfony-bootstrap-assetic-less>

Consultat el 01/05/2013

· Vince Verber (2011). "**FULL guide for integrating twitter and facebook with fos userbundle in symfony 2.1**" <https://coderwall.com/p/qkdzca>

Consultat el 15/05/2013

· Scott Ansil (2012). "**[FOSTwitterBundle]: Simple way to tweet**"

<http://aulatic.16mb.com/wordpress/2011/09/fostwitterbundle-simple-way-to-tweet/>

Consultat el 30/06/2013