



S.D.U.W.

Segmentacions dinàmiques sobre usuaris d'un motor de blogging basat en Wordpress

Memòria de Projecte Final de Grau

Grau Multimèdia
Enginyeria web

Autor: Marc Benet Surroca

Consultor: Ignasi Lorente Puchades

Professor: Carlos Casado Martínez

Crèdits

Reconeixement-NoComercial-CompartirIgual 3.0



Dedicatòria

Pere, Carme i Guillem,

Sou els meus grans patrocinadors i motivadors

Res d'això seria possible sense vosaltres.

Cristina,

Sense la teva inestimable ajuda i suport

ara mateix no estaria on estic.

Magdalena,

Les teva energia, positivisme i ganes de viure m'han donat

l'ultima empenta per acabar aquesta gran fita de la meva vida.

Abstract

L'analítica web i el comportament d'usuaris és un pilar bàsic pel model de negoci de màrqueting i publicitat que es porta a terme avui en dia. Avui en dia, si es vol assegurar un bon resultat d'una campanya publicitària, s'han de saber els gustos i el comportament dels sectors de la població interessats en els productes que s'ofereixen. Només d'aquesta manera s'obtindrà una reacció positiva sobre els anuncis que arriben als consumidors.

Normalment es podrien segmentar aquests segments de població a base de recollir dades sobre diferents campanyes publicitàries enviades amb anterioritat, de les quals s'ha pogut recollir un cert comportament de les persones en vers el material rebut, o bé, a partir d'enquestes o promocions de serveis gratuïts a canvi dels quals s'ha rebut una direcció de correu electrònic i dades de contacte.

Així doncs aquest Treball de Final de Grau pretén donar una solució d'un caire diferent a la demanda d'informació esmentada anteriorment, el que es proposa és recollir l'analítica web que ens proporciona un *blog* d'una temàtica o temàtiques concretes per tal d'obtenir i segmentar les dades dels seus usuaris a partir de la seva activitat i gustos dels quals han deixat constància en el *blog*.

Tot el projecte estarà desenvolupat sobre una aplicació web que respondrà a les següents característiques:

- Segmentacions dinàmiques sobre una base de dades d'usuaris d'un *blog* basat en el CMS Wordpress a partir d'un formulari amb clàusules dinàmiques.
- Búsqueda i obtenció de resultats de les segmentacions.
- Generació d'estadístiques sobre diferents atributs dels usuaris del *blog*.

L'aplicació resultant del projecte estarà desenvolupada sota el framework Symfony 2 en llenguatge PHP. Aquest servirà de base per dur a terme connexions amb altres tecnologies que permetran resoldre les gestions amb la base de dades (MySQL), Javascript amb la llibreria jQuery i el conegut *Frontend Framework* Foundation que optimitza el llenguatge HTML per tal que pugui ser visualitzat fàcilment per a usuaris de *Smartphones* i tauletes.

Paraules clau: Treball de Final de Grau, Segmentacions dinàmiques, framework, blog, Analítica web

Índex

Crèdits	1
Abstract.....	3
1. Introducció	8
2. Descripció	9
3. Objectius	10
3.1. Generals.....	10
3.2. Específics.....	10
4. Continguts	11
4.1. Eines de les quals disposa l'aplicació.....	11
5. Metodologia.....	12
6. Arquitectura de l'aplicació.....	13
6.1. Estructura bàsica de desenvolupament.....	13
6.2. Comunicació.....	14
6.3. Escalabilitat.....	15
7. Plataforma de desenvolupament.....	15
7.1. Màquina virtual.....	15
7.2. Control de versions.....	16
7.3. Projecte.....	17
8. Planificació	18
9. Procés de desenvolupament.....	20
9.1. Anàlisi previ.....	20
9.2. Anàlisi tècnic	20
9.3. Disseny.....	20

Segmentacions dinàmiques sobre usuaris d'un motor de blogging basat en Wordpress	
9.4. Desenvolupament.....	20
9.5. Bugfixing.....	20
9.6. Finalització.....	21
10. Frameworks i llibreries utilitzades.....	22
10.1. Symfony 2.....	22
10.2. jQuery.....	22
10.3. Foundation.....	22
10.4. API Google Charts.....	23
10.5. DataTables.....	23
11. Diagrames UML	24
11.1. Diagrama de casos d'ús	24
11.2. Diagrama de classes	25
12. Wireframes	25
13.1. Entrada.....	25
13.2. Homepage.....	26
13.3. Cerca Usuaris.....	26
13.4. Llista Segments.....	27
13.5. Formulari Segment.....	27
13. Perfils d'usuari.....	28
13.1. Supervisor.....	28
13.2. Administrador de seleccions.....	28
13.3. Conclusions.....	29
14. Usabilitat i UX.....	30
14.1. Navegació.....	30

14.2. Sitemap.....	30
14.3. DCU.....	31
15. Seguretat.....	32
16. Tests.....	33
17. Versions de l'aplicació.....	34
18. Requisits d'instal·lació.....	35
19. Instruccions d'instal·lació.....	36
20. Instruccions d'ús.....	38
20.1. Cerca Usuaris.....	38
20.2. Crear Segment.....	39
20.3. Elimina Segment.....	40
21. Funcionament Intern de l'aplicació.....	41
21.1. Com es genera internament una segmentació?.....	41
21.2. Com s'obtenen les gràfiques del reporting?.....	41
21.3. Com s'efectua la cerca de contactes?.....	42
22. Bugs	43
23. Projecció a futur.....	44
23.1. Inmediat.....	44
23.2. Pròxim.....	45
23.3. Llunyà.....	46
24. Pressupost.....	46
25. Anàlisi de mercat	47
26. Conclusions.....	48

Figures i taules

Il·lustració 1. Mètode cascada	13
Il·lustració 2. Arquitectura web	14
Il·lustració 3. Entorn de treball	16
Il·lustració 4. Bitbuket.com	17
Il·lustració 5. Planificació de tasques	19
Il·lustració 6. Diagrama de Gantt	20
Il·lustració 7. Casos d'ús	25
Il·lustració 8. Diagrama de Classes.....	26
Il·lustració 9. Wireframe login	27
Il·lustració 10. Wireframe Home.....	28
Il·lustració 11. Wireframe cerca Usuaris.....	28
Il·lustració 12. Wireframe Llistat segments.....	29
Il·lustració 13. Wireframe Formulari segments.....	29
Il·lustració 14. Diagrama 2 passos de seguretat	34
Taula 1. Llista de bugs	43
Taula 2. Pressupost de RRHH	46
Taula 3. Pressupost no RRHH	46

1. Introducció

Tota empresa que es dediqui a l'explotació d'una base de dades de clients per a fins publicitaris, sempre està buscant noves vies per incrementar el volum de dades i la seva varietat. Com a mètode tradicional ens podem suposar que disposen de mètodes per poder inserir dades de clients a una base de dades de contactes molt gran sobre la qual hi envien campanyes publicitàries.

Normalment disposen de les següents possibilitats:

1. Comprar bases de dades externes.
2. Subcontractar bases de dades de tercers per efectuar l'enviament de campanyes publicitàries.
3. La compra i obtenció de direccions a partir de promocions i ofertes especials a canvi de les dades de contacte.

Cada vegada més s'inverteixen recursos en poder cercar clients nous als quals poder fer arribar els productes que es volen oferir. Mitjançant l'ús de l'aplicació que es proposa en aquest projecte es vol donar una via de prospecció sobre superfícies diferents a les tradicionals per aquestes empreses.

El fet de poder segmentar usuaris d'un blog segons els seus comentaris o interessos obre les portes a una nova manera de cercar contactes de clients potencials nous des del lloc on mostren els seus interessos i la seva cara més informal, un blog.

2. Descripció del Projecte

Com bé indica el nom del projecte, el seu fi principal és poder segmentar i filtrar la informació que desprèn una base de dades d'un motor de blogging basat en Wordpress.

L'aplicació web constarà d'una eina basada en un formulari amb clàusules dinàmiques, mitjançant les quals es podrà segmentar els usuaris a partir de la informació que hi hagin deixat prèviament al blog. Per poder evaluar aquestes segmentacions es podrà filtrar per diferents atributs definits en el propi blog.

També oferirà una secció on efectuar cerques sobre les segmentacions creades anteriorment i sobre el nom o direcció electrònica dels usuaris del blog.

Finalment, hi haurà la possibilitat d'obtenir estadístiques en forma de gràfics sobre els diferents camps per fer les segmentacions disponibles.

Cal afegir que prèviament a poder recopilar totes les dades cal que el blog sobre el qual s'extrauran les dades hi hagi una certa activitat prèvia per part dels usuaris.

2.1. Exemple de treball sobre l'aplicació

Abans de fer qualsevol segmentació l'usuari de l'aplicació vol saber en quin tant percent han aparegut tots els tags del blog sobre tots els posts que s'hi ha efectuat. L'usuari podria accedir a la secció d'estadístiques i fer una consulta ràpida i visual del camp consultat amb un resultat gràfic en forma de formatge.

Un cop consultada la informació, a la secció de segmentacions pot elaborar la selecció que més l'hi convingui tenint en compte els resultats anteriors.

Com a últim pas, es pot fer una cerca sobre la segmentació efectuada anteriorment i obtenir els noms d'usuari i les direccions de correu electrònic corresponents als usuaris inclosos a dins del segment. D'aquesta manera se'n pot valorar la fiabilitat.

Respectant la LOPD, tot anunciant haurà d'informar prèviament a qualsevol usuari que sigui susceptible a rebre qualsevol campanya publicitària que a partir d'aquest moment la seva direcció de correu electrònic entrarà a la seva base de dades de clients provinent del *blog*. Al mateix temps ha de donar dret a l'usuari a poder-se donar de baixa, sense condicions, d'aquest servei.

3. Objectius

Els objectius d'aquest projecte s'entendran com en un projecte real; es planteja una manera de funcionar per part d'un hipotètic client, i com a empresa s'han de desenvolupar els objectius generals i específics. D'aquests objectius s'extrauran un seguit de requisits i pautes a desenvolupar per tal de donar solució a la demanda que es planteja.

3.1. Generals

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

- a) Facilitat de la lectura i guardat de les dades que construeixen les segmentacions.
- b) Fer més senzilla i amigable la gestió de la informació sobre el comportament dels usuaris d'un motor de *blogging*.
- c) Obtenir gràfiques i anàlisi del contingut de la base de dades d'usuaris.
- d) Aprendre a fer servir un *framework* per desenvolupar el programari.

3.2. Específics

Aquí es defineixen els objectius més concrets que ajudaran a arribar al punt final del projecte.

- a) Obtenir informació addicional i detallada sobre els gustos i comportament d'usuaris d'un *blog*.
- b) Poder realitzar cerques dins de la base de dades des de diferents seccions de l'aplicació web per obtenir-ne resultats semblants amb diferent tipus de sortida (gràfica o dades de text)
- c) Generar gràfics resultants de l'anàlisi de la base de dades en temps real.
- d) Permetre la categorització de la informació disposada a partir de les segmentacions generades per l'usuari de la plataforma.
- e) Visualització de forma ràpida de les diferents segmentacions generades i del nombre d'usuaris del *blog* als quals afecta.

4. Continguts

Una vegada analitzats els problemes a resoldre i els objectius a realitzar, se n'han extret els requisits que haurà de complir l'aplicació. Així doncs a continuació s'aporta la documentació conceptual prèvia a l'anàlisi més tècnic, l'últim pas abans d'entrar en la branca més orientada al programari.

4.1. Eines de les quals disposa l'aplicació

En primera instància, abans de descriure les funcions és important delimitar els rols que tindran els usuaris de l'aplicació. Cadascun dels rols tindrà un seguit de funcions associades, i tots ells seran compatibles entre ells.

Cal esmentar abans de continuar que, donades les dades personals que es detallen dins de l'aplicació, no serà possible accedir-hi sense autenticar-se i tenir assignat un rol prèviament.

Descripció dels rols:

Rol supervisor: aquest perfil d'usuari serà el que tindrà accés a la part d'estadístiques de l'aplicació, serà capaç de veure els gràfics i les estadístiques de la base de dades d'usuaris de Wordpress.

Rol seleccionador: aquest perfil d'usuari, permetrà efectuar segmentacions sobre a base de dades de contactes de Wordpress sobre els camps que desitgi, observar-ne el nombre de resultats i fer operacions sobre elles.

Rol de cerca: mitjançant aquest rol, els usuaris que el tinguin assignat podran efectuar cerques sobre diferents camps de la base de dades o segmentacions per tal de veure'n els usuaris que responen a la cerca realitzada.

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 que es poden adoptar per dur a terme el procés, en aquest cas, donat els requeriments temporals i personals dels quals es disposa s'ha optat pel model en cascada.

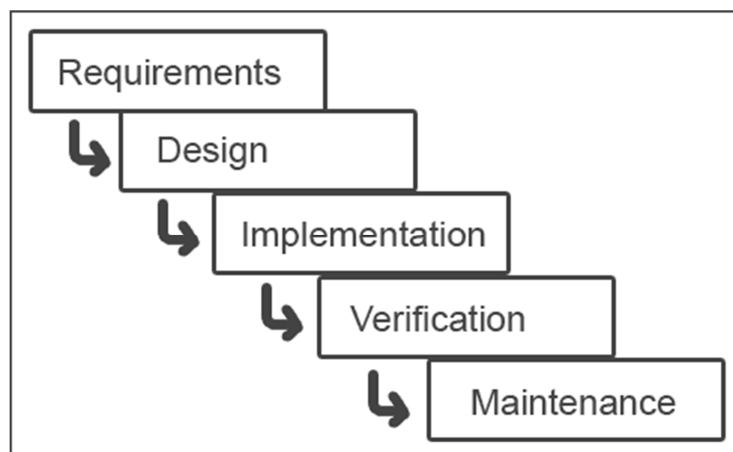
Aquest model es basa en un procés seqüencial de desenvolupament en el que les tasques són vistes com graons dependents un de l'altre, vistos en el *Planning* com si fossin una cascada d'aigua. Les fases que el caracteritzaran són l'anàlisi de les necessitats prèvies, el disseny, la implementació, les proves de validació, la integració i el manteniment.

Aquesta metodologia segueix uns principis bàsics:

- El projecte està dividit en fases seqüencials, amb certa superposició i dependència entre les fases.
- Es dóna una especial importància a la planificació, els horaris i les dates, així com als pressupostos i execució de tot el sistema d'una sola vegada.
- Es manté un estricte control durant tota la vida del projecte a través de la utilització d'una àmplia documentació escrita. 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.



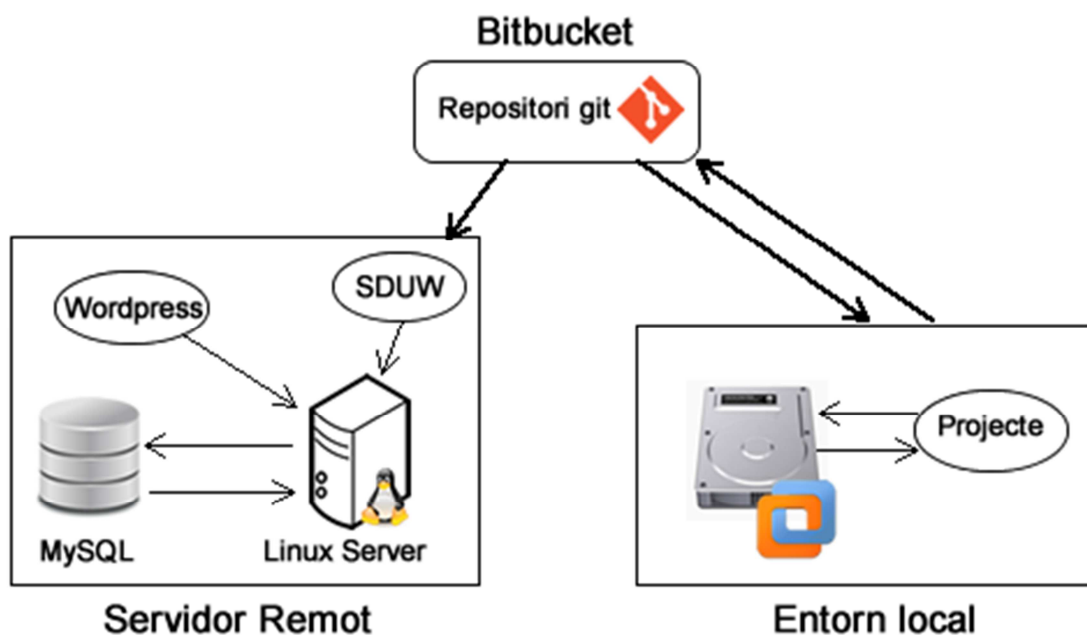
Il·lustració 1. Mètode cascada

6. Arquitectura de l'aplicació

Definir l'arquitectura web és un procés en el qual és necessari dissenyar l'entorn sota el qual l'aplicació funcionarà i donarà servei als usuaris. A part de garantir que l'aplicació funcionarà correctament, ha de ser un entorn controlat per que els desenvolupadors puguin treballar amb comoditat i contemplar la futura escalabilitat.

6.1. Estructura de desenvolupament

A la il·lustració següent es mostra 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.



Il·lustració 2. Arquitectura web

La plataforma d'entorn local, que és on treballaran els desenvolupadors, serà ampliada a la secció "7."

El servidor remot serà on l'aplicació es desplegarà en el seu entorn de producció i serà utilitzada per als seus usuaris finals. En aquest cas concret l'aplicació té el requisit de que haurà d'estar allotjada en el mateix servidor que el Wordpress al qual ha de fer les consultes. Per aquest motiu queda subjecte a les dependències del blog. El blog de Wordpress sobre el qual haurà de treballar l'aplicació està allotjat en una màquina amb sistema operatiu basat en "Linux" on hi té configurat els següent programari:

- *Apache*, software que permet utilitzar la màquina per servir webs als usuaris.
- PHP, el llenguatge de codi interpretat sota el qual estarà desenvolupada gran part de l'aplicació i és sobre el qual es basa Symfony2.
- GIT, sistema de control de versions, ens servirà per poder descarregar el codi destinat a desplegar la nostra aplicació.

Pel que fa al Sistema Gestor de Bases de Dades, es farà servir MySQL, ja que és sobre el qual s'ha desenvolupat el motor de blogging. És de tipus relacional i s'hi allotjaran les taules referents a l'aplicació web que es desenvolupa.

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

6.2 Comunicació

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

Seguint el diagrama anteriorment mencionat, l'usuari accedirà al servidor remot per consultar i treballar amb la aplicació web, i aquesta serà la única manera d'accedir 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) es tingui accés per la seva consulta, donat que sinó fos d'aquesta manera, suposaria un forat de seguretat important, tot i està protegit amb contrasenya.

El servidor remot tindrà accés al repositori, però només de 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.

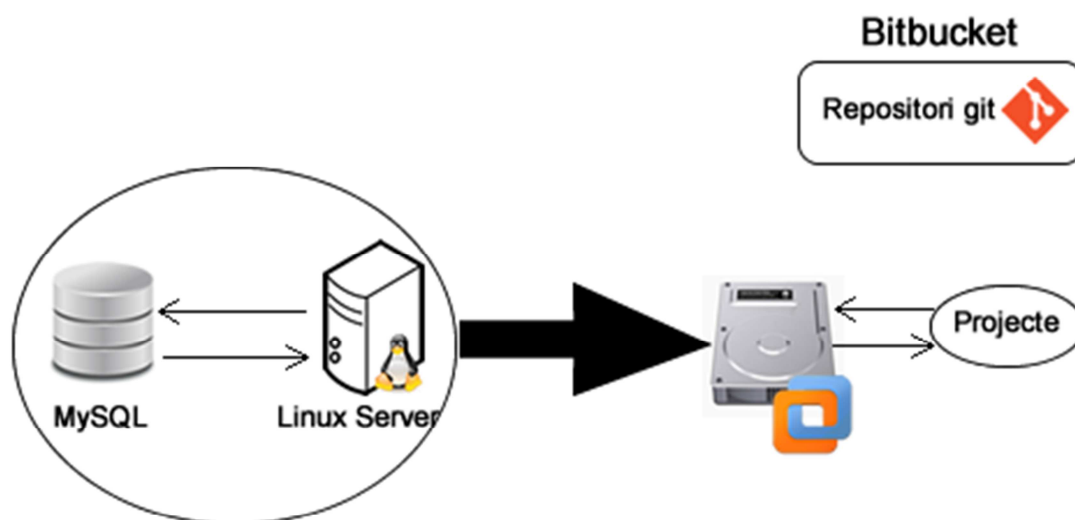
6.3. Escalabilitat

Tot i que l'aplicació 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 la cobertura de l'aplicació a altres motors de blogging basats en Drupal, o bé altres blogs personalitzats. En cas que fos així s'hauria de treballar en un Desacoblament dels paràmetres de configuració de la l'aplicació per tal de que es puguin variar per adaptar-la a diferents plataformes.

7. Plataforma de desenvolupament

L'entorn de desenvolupament d'aquesta aplicació web es farà sobre sistema operatiu Windows7. Havent definit l'entorn de producció en el punt anterior, s'opta per utilitzar una màquina virtual per emular el mateix entorn de treball que existirà en el moment de desplegar l'aplicació al servidor de producció. D'aquesta manera obtenim una arquitectura web quasi bé idèntica per poder-hi fer proves i desenvolupar en un entorn el més similar possible els diferents mòduls de l'aplicació.



Il·lustració 3. Entorn de treball

7.1. Màquina virtual

En el món de la informàtica i sobretot en el món del desenvolupament de programari és important tenir en compte la compatibilitat del software en diferents sistemes i en cas de aplicacions web també cal tenir en compte els navegadors així com la combinació de tots dos elements.

A part, per raons de comoditat, a l'hora de generar codi en un entorn local controlat, ha d'assegurar que aquest producte que s'està desenvolupant, 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, dins del mateix ordinador, reserva una sèrie de recursos, els quals interpreta 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 conviure.

En aquest 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 s'eviten problemes d'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

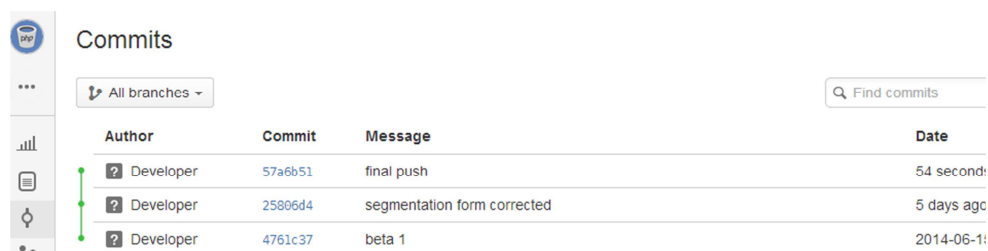
En tot projecte de desenvolupament de software mitjanament professional és necessari incorporar un sistema de control de versions.

Aquests tipus de programari permeten gestionar i emmagatzemar tots els canvis que s'han efectuat sobre el codi del projecte, d'aquesta manera es pot generar versions o revisions del projecte que son simplement estats del producte que s'està desenvolupant en un moment en concret de la seva creació.

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.
- **Còpia de seguretat:** al final aquests sistemes funcionen com una còpia 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 treballen 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*. Aquest web permet la creació de repositoris 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.



Author	Commit	Message	Date
Developer	57a6b51	final push	54 seconds
Developer	25806d4	segmentation form corrected	5 days ago
Developer	4761c37	beta 1	2014-06-11

Il·lustració 4. Bitbucket.com

7.3. Projecte

A l'hora de desenvolupar el codi és normal ajudar-se dels anomenats IDE (plataformes integrades per a desenvolupament). Aquests tipus de 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 SublimeText.

En el mercat actual hi ha molts IDE preparats per treballar amb PHP com podrien ser Eclipse, Netbeans, PHPStorm, etc. Però fent una recerca per Internet he trobat aquest que de sèrie porta una llibreria de desenvolupament PHP força interessant, a més es poden adjuntar *plugins* per tal de fer-lo més complet i personalitzable. A més consumeix molt pocs recursos i manté l'estructura del projecte.

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

- **Editor de codi intel·ligent:** *SublimeText* interpreta el 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.
- **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 molt important que cal abordar des de el primer moment del projecte per no perdre el rumb i aconseguir un cicle de creació i desenvolupament sense desviacions.

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

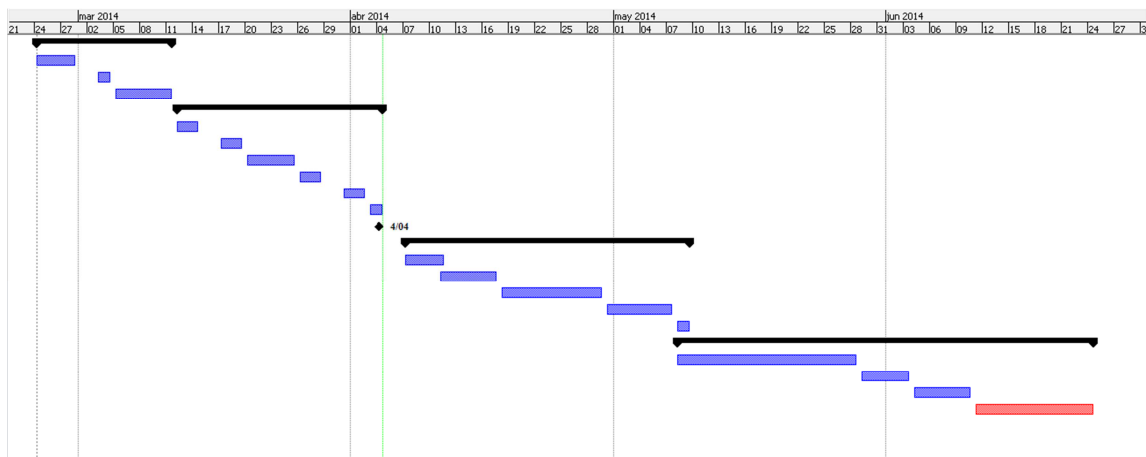
8.1. Planificació inicial

Des de la data d'inici del projecte (24 de febrer) fins a la data d'entrega (24 de juny).

		Nombre	Duració	Inicio	Terminado
1		Planificació	12 days?	24/02/14 8:00	11/03/14 17:00
2		Recull d'informació	5 days?	24/02/14 8:00	28/02/14 17:00
3		Definició del projecte	2 days?	1/03/14 8:00	4/03/14 17:00
4		Documentació PAC1	5 days?	5/03/14 8:00	11/03/14 17:00
5		Definició	18 days?	12/03/14 8:00	4/04/14 17:00
6		Requisits	3 days?	12/03/14 8:00	14/03/14 17:00
7		Arquitectura web	3 days?	15/03/14 8:00	19/03/14 17:00
8		Wireframes	4 days?	20/03/14 8:00	25/03/14 17:00
9		Disseny	3 days?	26/03/14 8:00	28/03/14 17:00
10		Diagrama de classes	3 days?	29/03/14 8:00	2/04/14 17:00
11		Bases de dades	2 days?	3/04/14 8:00	4/04/14 17:00
12		Documentació PAC2	0 days?	4/04/14 8:00	4/04/14 8:00
13		Desenvolupament	25 days?	7/04/14 8:00	9/05/14 17:00
14		Preparació workspace	5 days?	7/04/14 8:00	11/04/14 17:00
15		Estructura MVC	5 days?	11/04/14 8:00	17/04/14 17:00
16		Controladors	8 days?	18/04/14 8:00	29/04/14 17:00
17		Vistes	6 days?	30/04/14 8:00	7/05/14 17:00
18		Documentació PAC3	2 days?	8/05/14 8:00	9/05/14 17:00
19		Finalització	34 days?	8/05/14 8:00	24/06/14 17:00
20		Desenvolupament de persistència	15 days?	8/05/14 8:00	28/05/14 17:00
21		Bugfixing	4 days?	29/05/14 8:00	3/06/14 17:00
22		Desplegament	5 days?	4/06/14 8:00	10/06/14 17:00
23		Finalització de la memòria	10 days?	11/06/14 8:00	24/06/14 17:00

Il·lustració 5. Planificació de tasques

A continuació el diagrama de Gantt associat al projecte:



Il·lustració 6. Diagrama de Gantt

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

En la **planificació**, es simulen les reunions inicials amb el client final per arribar a un consens i obtenir un anàlisi de requisits clars. En aquesta part el projecte comença a agafar un rumb concret.

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

L'etapa de **desenvolupament** és 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 començar a cobrar vida i al llarg de les entregues arribarà a una fase Beta.

La **finalització** és 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 anirà 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 compromès.

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.

Aquest 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 acabar d'afinar i posar cara l'anàlisi tècnic previ. Aquest punt no només té com a objectiu acabar d'especificar el contingut de l'aplicació, sinó que és molt útil alhora de detectar errors en la definició, ja que és en aquest punt en que les idees comencen a agafar forma.

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

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. Resolució de errors o *Bugfixing*

Un cop s'entregui una versió estable cal que agents externs als desenvolupadors facin proves sobre la web final i detectin tota mena d'errors i inconsistències.

Moltes vegades aquest punt també es pot considerar un estudi de la usabilitat, molta de la informació obtinguda es podrà utilitzar per detallar millores i possibilitats noves de l'aplicació.

9.6. Finalització

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

En aquest punt és 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.

10. Frameworks, API's i llibreries utilitzades

10.1. Symfony2

A l'hora de desenvolupar un producte web en PHP professionalment és difícil no pensar en fer servir un un *framework*. *Symfony*, en la seva segona versió, 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, l'enviament de correus electrònics i a seguir el patró MVC (model – vista – controlador) amb un gran resultat.

Finalment, cal esmentar, que gracies a la gran comunitat que té al 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

Avui en dia, si es vol introduir interacció, dinamisme i AJAX en una plana web, és necessari utilitzar el llenguatge *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:** es delega la responsabilitat sobre la compatibilitat entre 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 el codi, a la vegada el fa mes intuïtiu i llegible.
- **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. Foundation

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 *Zurb* 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 web. 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.

El mateix *framework* permet definir diferents amplades de columnes segons la mida de la pantalla que visualitza el contingut. De fet el propi Foundation incorpora la filosofia *Mobile-First*, que consisteix en el fet de que la plana web es dissenyi en primera instància sobre una visualització de telèfon mòbil, i a partir d'aquesta modificar-la per ser visible correctament en pantalla gran. No al revés, com es pensa habitualment.

10.4 API Google Charts

Per finalitzar, també he fet servir la API "Google Charts" que es tracta d'una llibreria escrita en HTML5 i Javascript que proporciona Google a partir de la càrrega d'una llibreria gràfica que proporcionen a la seva web. Soporta tot tipus de gràfics, en concret per el projecte que es desenvolupa interessa només els gràfics de formatgets per tal de mostrar una informació amigable del contingut de la nostra base de dades.

10.5. DataTables

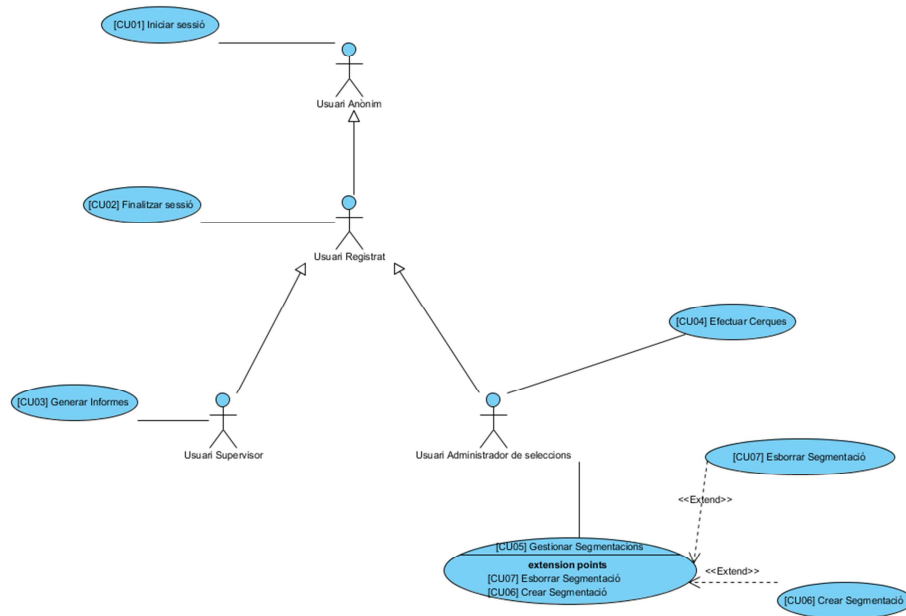
Per poder mostrar unes taules que es puguin pàginar en cas que hi hagi un nombre de resultats elevat, permetin ordenar les columnes i fer cerques sobre els resultats mostrats en la taula, he utilitzat aquest plugin basat en jQuery que permet afegir les propietats esmentades anteriorment a qualsevol taula estàtica.

11. Diagrames UML

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

11.1. Diagrama de casos d'ús

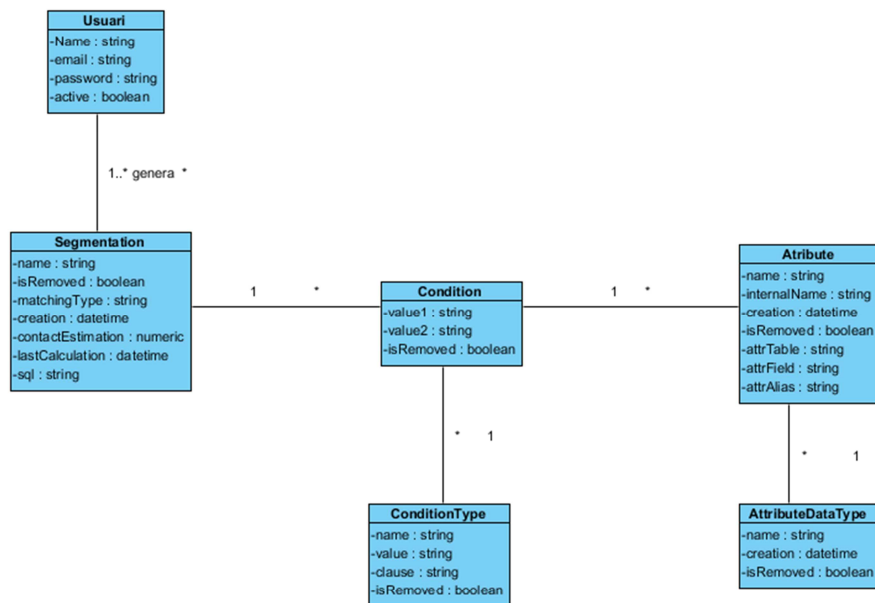
Aquest diagrama serveix per a modelitzar els casos d'ús de l'aplicació web i relacionar-los amb els actors del sistema.



Il·lustració 7. Diagrama de casos d'us

11.2. Diagrama de classes

En aquest diagrama es modelitzen les classes i objectes així com les seves relacions fent un èmfasis 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 fet de que s'utilitza un ORM (Doctrine), que entre les seves funcions hi és el fet de transformar les classes definides en taules i relacions a la base de dades.



Il·lustració 8. Diagrama de classes

12. Wireframes

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.

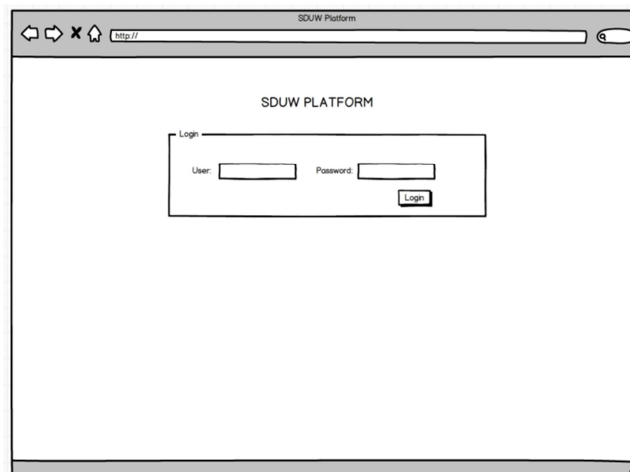
Tal i com hem fet en aquest cas, s'utilitzen normalment per a il·lustrar la interfície que veurà l'usuari final, sense necessitat d'haver desenvolupat el software. Aquesta disposició d'imatges defineixen bastant bé com serà cadascuna de les vistes que hi haurà en la següent etapa de disseny i maquetació.

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

- **Entrada:** login
- **Home page:** Reporting
- **Cerca d'usuaris:** formulari de cerca
- **Llistat de segments:** llistat de les segmentacions existents
- **Formulari de segment:** generació de les segmentacions

12.1 Entrada

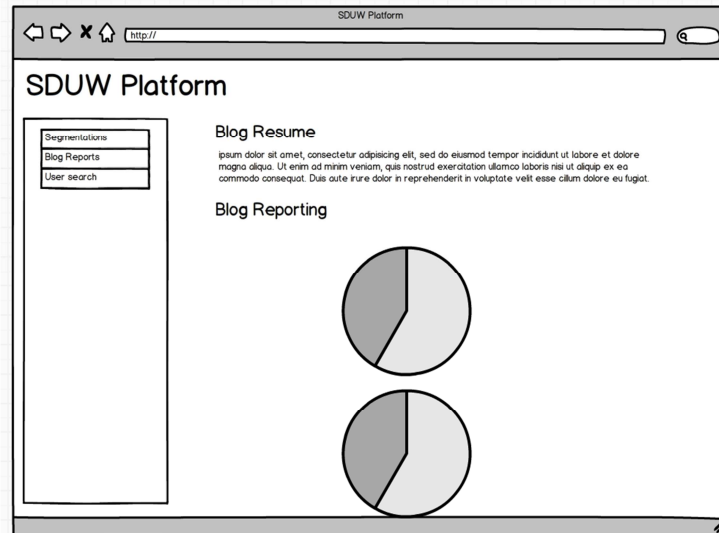
En la entrada de l'aplicació disposarem d'un *Login* típic on es demana l'usuari i la clau de l'usuari. Donat que l'aplicació serà d'accés privat no hi ha opció de registre.



Il·lustració 9. Wireframe Login

12.2. Homepage

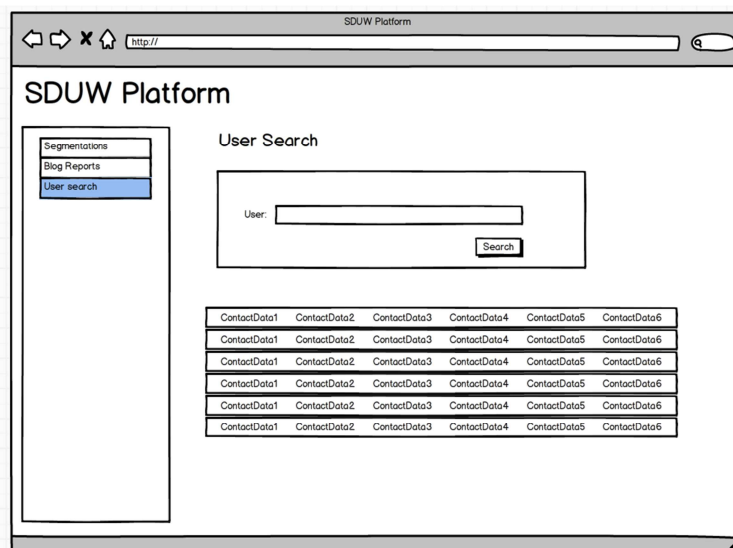
En aquesta secció es mostra un resum de l'activitat diària i general del blog en forma de gràfiques i resums. D'aquesta manera es pot supervisar de forma genèrica les dades abans d'aplicar qualsevol segment.



Il·lustració 10. Wireframe Home-Reporting

12.3. Cerca d'usuaris

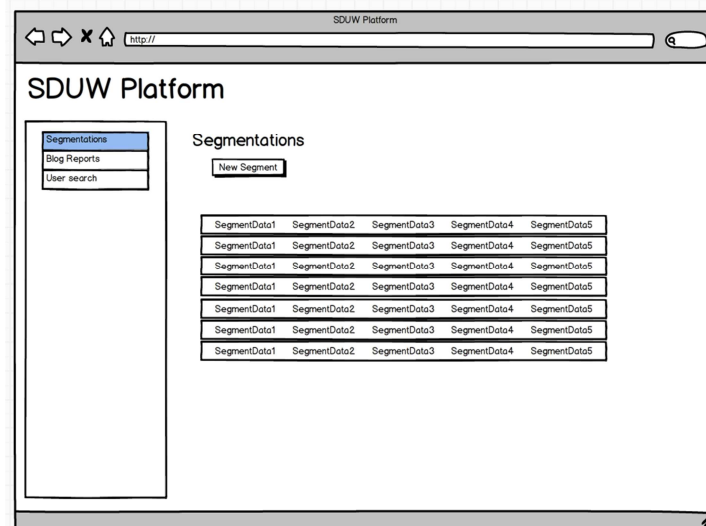
En aquest apartat es mostrarà un formulari per buscar usuaris del blog i obtindrà les seves informacions bàsiques en forma de llistat.



Il·lustració 11. Wireframe Cerca d'usuaris

12.4. Llistat de Segments

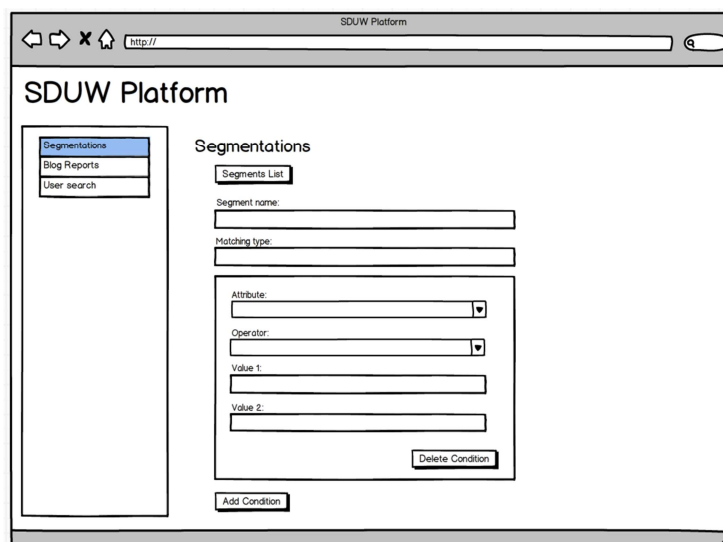
En aquesta pàgina es mostra el llistat de segmentacions amb el seu nombre de contactes estimat juntament amb el tipus de relació que mantenen amb les condicions i la data de creació.



Il·lustració 12. Wireframe Llistat de segments

12.5. Formulari de Segment

Aquí es mostrarà totes les entrades per poder generar una segmentació sobre els usuaris del blog, donant la opció de crear i esborrar condicions i modificar les dades dinàmicament abans de guardar el segment.



Il·lustració 13. Wireframe Formulari de segment

13. Perfils d'usuari

En la fase del Disseny Centrat en l'Usuari (DCU), un component força important és l'anàlisi de usuaris. Aquest anàlisi d'usuari proporciona detalls sobre qui farà servei del producte, identifica rols i defineix les característiques de l'usuari. Com ara:

- 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 l'aplicació que es desenvolupa, s'han definit dos grups d'usuaris diferents:

- Supervisor.
- Administradors de seleccions.

A continuació l'avaluació de cadascun d'aquests per tal de treure'n el màxim profit del plantejament del DCU.

13.1. Supervisor

13.1.1. Context

Seràn els usuaris encarregats de supervisar el treball sobre la plataforma d'usuaris del *blog* i avaluar-ne les dades. Els perfils d'usuari que supervisen els continguts de l'aplicació, no tenen perquè ser capaços de efectuar-ne seleccions.

13.1.2. Factors demogràfics

Els perfils professionals de caps de projecte solen ser en la seva majoria homes en una edat sobre els 30 anys cap a munt. Molts d'ells no tenen coneixements tècnics, ja que en un percentatge elevat tenen estudis de direcció de negocis que no tenen perquè incloure coneixements necessaris o bàsics per fer funcionar el motor de seleccions de l'aplicació.

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 gestions directives a totes hores i en qualsevol dia de la setmana s'acostuma a portar un seguiment des de mòbil o tauleta.

13.2. Administrador de seleccions

13.2.1. Context

Seràn els usuaris encarregats de fer les seleccions sobre la base de dades d'usuaris del blog. Tindran permès crear i editar-ne seleccions i efectuar-ne cerques a partir d'elles.

13.2.2. Factors demogràfics

En aquest cas el perfil d'usuari és més tècnic, sol ser un estudiant d'enginyeria o un enginyer format, amb certa experiència amb bases de dades i de les exigències dels clients que sol·liciten informació sobre la base de dades del blog.

13.2.3. Entorn

L'entorn i abast per utilitzar aquesta aplicació acostuma a ser en oficina, tot i que donada les necessitats diverses podrien treballar des de casa o en qualsevol altre lloc que no sigui la oficina mitjançant un dispositiu mòbil o tauleta.

13.3. Conclusions

Tot i que els perfils son bastant diferents un de l'altre, és dedicarà el màxim esforç de desenvolupament cap a la part dels Administradors de seleccions, donat que són els que faran servir l'aplicació en un rang més extens i on basaran la majoria de la seva feina.

El rol de Supervisor serà un suport extern i farà poc ús de l'aplicació a la vegada que tindrà un accés molt superficial 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 d'un dispositiu mòbil o tauleta.

14. Usabilitat i UX

És interessant tenir en compte l'experiència de l'usuari en el moment que està fent servir l'aplicació web. S'ha de plantejar com fer òptima la navegabilitat i tot l'entorn que genera l'experiència del treball durant el funcionament de l'aplicació.

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 pràctica i intuïtiva, així com l'accés a totes les seves seccions sigui clar i sense obstacles.

14.1. Navegació

L'esquema de navegació de què disposa el programari que es desenvolupa és bàsicament global, navegarà entre Seleccions, Criteris, Informes i Cerca.

La profunditat màxima de l'aplicació serà de tres nivells. On la secció de més profunditat serà la de crear segmentacions.

En tot moment disposarem d'un *breadcrumb* que ens informarà des de la part superior en quina secció, subsecció i/o fitxa ens trobem. D'aquesta manera l'usuari serà conscient de la seva situació i a la vegada podrà desfer el camí fàcilment.

14.2. Sitemap

El sitemap de l'aplicació serà el següent; consta d'una llista de pàgines accessibles pels usuaris. En el cas actual s'organitza en un format de llista donada la simplicitat de l'organització de l'aplicació:

- Home
 - Login
 - Logout
 - Cerca
 - Seleccions
 - Eliminar
 - Nova
 - Selecció de condicions
 - Informes

14.3. DCU

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

Part de l'anàlisi de les seccions anteriors es parla molt de com s'està dissenyant aquesta experiència, així com la avaluació de grups de rols que farà servir l'eina.

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
- Avaluació heurística

15. Seguretat

L'aplicació ha de ser segura tant per l'usuari que la fa servir com per 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.

En el cas de l'aplicació que es desenvolupa, es desplega el codi sobre el mateix servidor on s'allotja el motor de *blogging* basat en *Wordpress*. Al tractar-se d'un servidor configurat prèviament, la seguretat d'aquest servidor raurà sobre l'equip de desenvolupament i manteniment del *blog*.

15.1. Seguretat de l'aplicació

La seguretat en l'aplicació anirà de la mà del *framework* Symfony2 que ens dona un seguit d'eines per a controlar tot aquest àmbit.

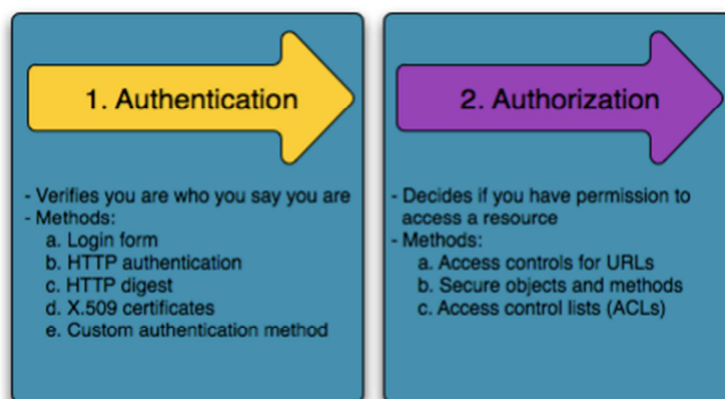
Un cop configurada aquesta seguretat, ens proporcionarà una validació en dos passos: Autenticació i Autorització. A continuació s'en detallen aquests passos.

En el primer pas del procés, el sistema de seguretat identificarà qui és l'usuari requerint que aquest aportí algun tipus d'identificació. Aquest procés s'anomena autenticació, la seva finalitat bàsica és esbrinar qui és l'usuari que es connecta i si és un usuari vàlid. En l'aplicació que es presenta, aquesta funció la efectua el *Login*.

Una vegada el sistema ha autenticat i validat l'usuari, el següent pas és determinar si l'usuari validat té privilegis per a executar una acció determinada. En l'aplicació es controlarà aquest accés a partir de la definició dels rols d'usuari.

Exemple:

Un usuari efectua un Login a l'aplicació correctament, i en la selecció de les segmentacions li apareixen només aquells projectes dels quals tingui privilegis de visualització i treball. Encara que d'alguna manera aconseguís accedir a una segmentació a la qual no estigui en el seu àmbit de treball, l'hi serà impossible realitzar cap acció i el sistema donarà una excepció del tipus "Autorització denegada".



Il·lustració 14. Diagrama de 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. Aquesta situació, en gran part dels casos és inevitable.

Per poder minimitzar aquestes fonts d'errors durant el desenvolupament del projecte, farem ús de dues eines, que tot i aplicar-se en moments diferents de la construcció de la web, tenen com a principal objectiu trobar excepcions i mancances al codi. Amb els resultats obtinguts els programadors podran ser capaços de resoldre els errors abans de publicar el producte final.

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 sí sola 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 i 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.

A mesura que el desenvolupament de l'aplicació avanci, si s'és capaç de comprovar petits fragments de codi que no han de ser molt més complexos, al final es pot assegurar que en quant a codi sempre s'obtindrà la resposta desitjada.

17. Versions de l'aplicació

Les versions en el software tracten d'assignar etiquetes úniques a les seves versions, donada una etiqueta o versió en un nombre, nom o codificació es pot entendre en quina etapa del desenvolupament es troba. Existeix la possibilitat de referir-se 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'establiran tres versions diferents però totalment operatives de l'aplicació. A continuació es detallen les característiques:

17.1. Versió Alpha

- Sistema d'autenticació: registre, login, logout.
- Sistema d'autorització: control de permisos
- Sistema bàsic de seleccions i criteris
- Interfície bàsica sense estils

17.2. Versió Beta

- Correcció de bugs de la versió anterior
- Integració de la cerca
- Integració dels informes
- Millores en interacció bàsica
- Control d'errors

17.3. Versió 1.0

- Correcció de bugs de la versió anterior
- Textos correctes i ajudes en l'aplicació
- Millora en la interfície i la interacció amb jQuery
- Establiment dels estils CSS finals

18. Requisits d'instal·lació

Per tal d'instal·lar el software és necessari un entorn de servidor web LAMP (Linux, Apache, MySQL, PHP). 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 basat en Linux [Ubuntu Server / Devian]
- PHP [5.3 ~ 5.4]
 - Extensions necessàries:
 - pdo_mysql
- Apache
- MySQL
- Composer

No es necessari que les versions siguin exactes, sempre i quant s'asseguri que es poden complir els mateixos requisits esmentats anteriorment.

19. Instruccions d'instal·lació

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

Nota: és important que el *virtual host* del servidor web Apache apunti directament a la carpeta `web/app.php`

El primer pas és configurar l'arxiu situat a `app/config/parameters.yml`, on haurem disposar de les dades de connexió a la nostra base de dades MySQL (que serà la mateixa que el blog) i les URL dels arxius en format `.cdn` del qual en té dependència l'aplicació.

```
parameters:
  database_driver: pdo_mysql
  database_host: /*wp_db_host*/
  database_port: /*wp_db_port*/
  database_name: /*wp_db_name*/
  database_user: /*wp_db_user*/
  database_password: /*wp_db_psw*/
  mailer_transport: smtp
  mailer_host: 127.0.0.1
  mailer_user: null
  mailer_password: null
  locale: en
  secret: 92c62bb398e794b63aec5548d89d0ed38fd8fd1d
  # CDNS
  cdn_jquery_script:
  '//ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js'
  cdn_jqueryui_script:
  'https://ajax.googleapis.com/ajax/libs/jqueryui/1.9.0/jquery-ui.min.js'
  cdn_jqueryui_css:
  'https://ajax.googleapis.com/ajax/libs/jqueryui/1.9.0/themes/smoothness/jquery-ui.css'
  cdn_datatables_script:
  '//ajax.aspnetcdn.com/ajax/jquery.dataTables/1.9.4/jquery.dataTables.min.js'
  cdn_datatables_css:
  '//ajax.aspnetcdn.com/ajax/jquery.dataTables/1.9.4/css/jquery.dataTables.css'
```

Un cop configurat el fitxer de paràmetres caldrà instal·lar les llibreries de tercers que utilitza l'aplicació (anomenades *vendors*) mitjançant el control de dependències de *Composer*.

```
$ php composer.phar install
```

La mateixa comanda ens neteja la cache de l'aplicació i la deixa neta per començar a funcionar de zero.

Després publicarem els arxius públics com CSS, JavaScript i imatges amb la següent comanda:

```
$ php app/console assets:install web --symlink
```

Acte seguit haurem de generar la base de dades i carregar les dades de configuració. Amb la següent instrucció generem la base de dades:

```
$ php app/console doctrine:schema:create
```

Quan tinguem l'esquema de la base de dades creat, necessitarem un usuari per poder inserir, en cas contrari, el Firewall del Symfony configurat amb el *Bundle 'FOS User Bundle'* no ens deixarà accedir a ninguna altre ruta que no sigui el login. Haurem d'executar les següents instruccions i seguir les indicacions:

```
$ php app/console fos:user:create /*genera usuari*/
$ php app/console fos:user:promote /*dota de rol al usuari*/
```

Com a últim pas ens queda inserir les següents dades de configuració a la base de dades de l'aplicació:

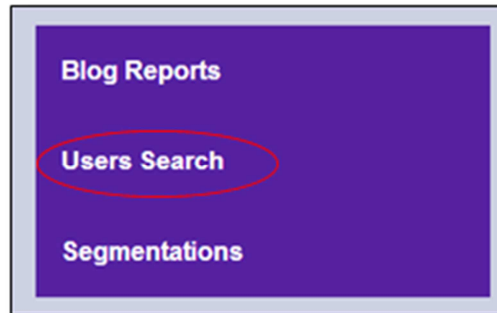
```
INSERT INTO `sduw_attribute_data_type` (`id`, `name`, `creation`,
`is_removed`) VALUES
(1, 'string', '2014-06-15 00:00:00', 0),
(2, 'integer', '2014-06-15 00:00:00', 0),
(3, 'boolean', '2014-06-15 00:00:00', 0),
(4, 'behavior', '2014-06-15 00:00:00', 0),
(5, 'datetime', '2014-06-15 00:00:00', 0);

INSERT INTO `sduw_condition_type` (`id`, `attribute_data_type_id`, `name`,
`value`, `clause`, `is_removed`, `creation`) VALUES
(1, 1, 'empty', 'empty', '__ATTRIBUTE_FIELD__ IS NULL OR __ATTRIBUTE_FIELD__ =
'', 0, '2014-06-15 00:00:00'),
(2, 1, 'not empty', 'not_empty', '__ATTRIBUTE_FIELD__ IS NOT NULL AND
__ATTRIBUTE_FIELD__ <> ''
\r\n', 0, '2014-06-15 00:00:00'),
(3, 1, 'equal', 'equal', 'LOWER(__ATTRIBUTE_FIELD__) = '__PARAM_1_VALUE__',
0, '2014-06-15 00:00:00'),
(4, 1, 'not equal', 'not_equal', 'LOWER(__ATTRIBUTE_FIELD__) <>
 '__PARAM_1_VALUE__', 0, '2014-06-15 00:00:00'),
(5, 1, 'contains', 'contains', 'LOWER(__ATTRIBUTE_FIELD__) LIKE
'%__PARAM_1_VALUE__%', 0, '2014-06-15 00:00:00'),
(6, 1, 'not contains', 'not_contains', 'LOWER(__ATTRIBUTE_FIELD__) NOT LIKE
'%__PARAM_1_VALUE__%', 0, '2014-06-15 00:00:00'),
(7, 1, 'starts with', 'starts_with', 'LOWER(__ATTRIBUTE_FIELD__) LIKE
 '__PARAM_1_VALUE__%', 0, '2014-06-15 00:00:00'),
(8, 1, 'ends with', 'ends_with', 'LOWER(__ATTRIBUTE_FIELD__) LIKE
'%__PARAM_1_VALUE__', 0, '2014-06-15 00:00:00'),
(9, 5, 'less or equal than', 'less_or_equal_than', '__ATTRIBUTE_FIELD__ <=
 '__PARAM_1_VALUE__', 0, '2014-06-15 00:00:00'),
(10, 5, 'higher or equal than', 'higher_or_equal_than', '__ATTRIBUTE_FIELD__
 >= '__PARAM_1_VALUE__', 0, '2014-06-15 00:00:00'),
(11, 5, 'not between', 'not_between', '__ATTRIBUTE_FIELD__ NOT BETWEEN
 '__PARAM_1_VALUE__' AND '__PARAM_2_VALUE__', 0, '2014-06-15 00:00:00'),
(12, 5, 'between', 'between', '__ATTRIBUTE_FIELD__ BETWEEN
 '__PARAM_1_VALUE__' AND '__PARAM_2_VALUE__', 0, '2014-06-15 00:00:00');
```

20. Instruccions d'ús

20.1. Cerca d'usuari

En primera instància ens haurem de dirigir a la secció coneguda com 'Cerca d'usuaris' a partir del menú de navegació lateral de l'aplicació.



A continuació se'ns mostrarà una pàgina amb un formulari, que un cop completat i enviat ens retornarà una llista amb tots els usuaris que coincideixen amb la cadena de text introduïda.

Resultat obtingut:

Show Search:

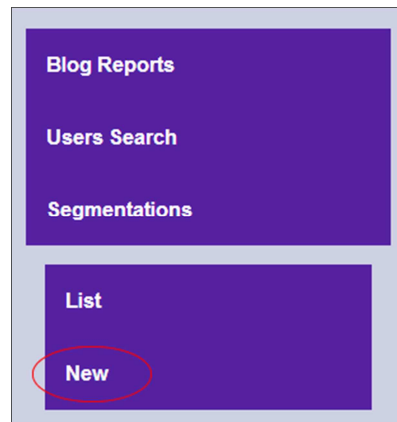
entries

id	login	nice name	display name	email	url	registered	status
4	marc2	marc2	marc marc	mбенет@gmail.com		2014-06-22 15:24:16	Active
2	marc	marc	Marc Benet Surroca	mбенетсу@uoc.edu		2014-03-07 14:15:06	Active

Showing 1 to 2 of 2 entries ◀ Previous Next ▶

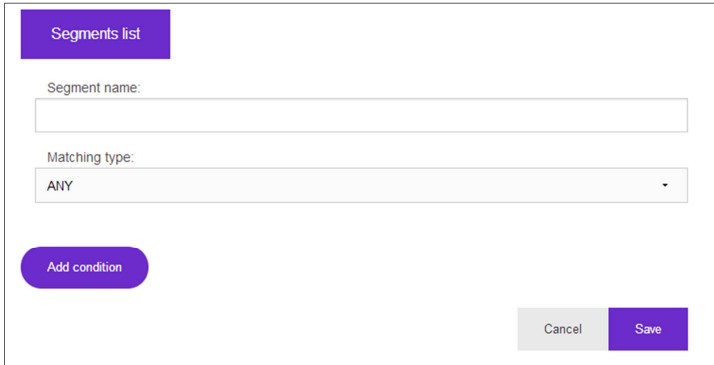
20.2 Creació d'una segmentació

Per a poder crear una segmentació nova, caldrà accedir a la seva plana a partir del menú de navegació.

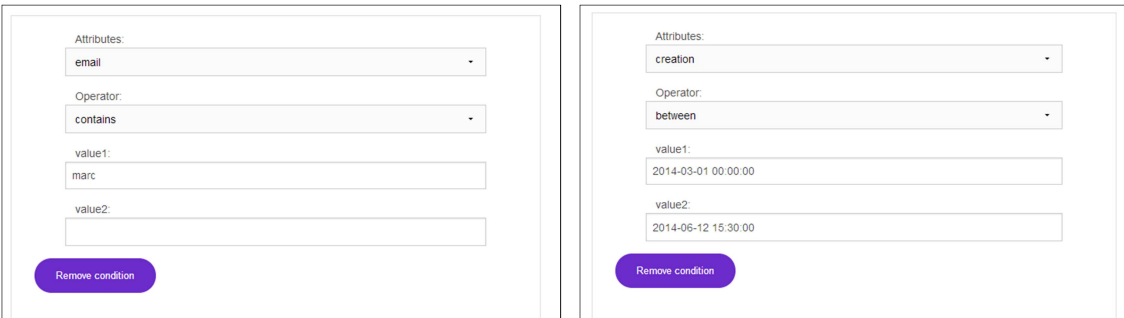


Un cop a dins de la plana se'ns mostra un menú bàsic on es demana el nom i el tipus d'enllaç que tindran les condicions de la segmentació. Podem afegir tantes condicions com vulguem per poder obtenir la segmentació desitjada.

Primera vista del formulari:

A form titled 'Segments list' with a purple header. It contains a text input for 'Segment name', a dropdown menu for 'Matching type' with 'ANY' selected, a purple 'Add condition' button, and 'Cancel' and 'Save' buttons at the bottom right.

Exemples de condicions:

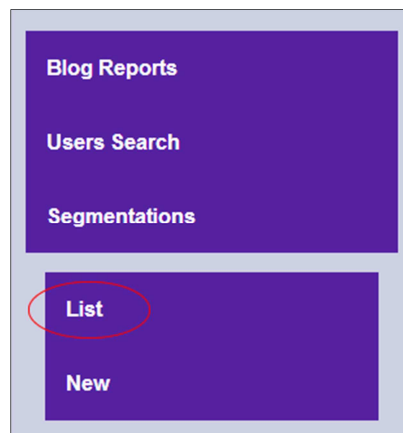
Two side-by-side screenshots of condition configuration forms. The left form shows 'Attributes: email', 'Operator: contains', 'value1: marc', and a 'Remove condition' button. The right form shows 'Attributes: creation', 'Operator: between', 'value1: 2014-03-01 00:00:00', 'value2: 2014-06-12 15:30:00', and a 'Remove condition' button.

Exemple de condició sobre usuaris amb correu electrònic que conté la paraula marc i exemple sobre la data de creació d'un usuari que estigui entre el dia 01/03 i el 12/06.

Exemple de condició sobre els usuaris que hagin comentat després del dia 01/03 i exemple sobre els usuaris que hagin postejat amb un tag igual a 'general'.

20.3 Eliminar una Segmentació

Per poder eliminar-ne una, haurem d'accedir a la llista on s'ens mostren totes les segmentacions, en aquesta taula es pot trobar una columna que s'anomena 'Actions'. En aquesta columna apareix la icona de borrar una segmentació. Quan apremem el botó, aquesta segmentació desapareixerà immediatament de la taula i quedarà marcada com esborrada a la base de dades.



Show: Search:

entries

id	name	matching type	number of users	last calculation	actions
3	testman	ANY	0	2014-06-24 13:28:44	
2	test 1	ANY	0	2014-06-24 13:28:44	
1	xaaaxo	ANY	0	2014-06-24 13:28:44	

Showing 1 to 3 of 3 entries ◀ Previous Next ▶

21. Funcionament intern de l'aplicació

Com s'ha detallat en apartats anteriors, l'aplicació en si consta de poques taules a la base de dades comparat amb les que pot arribar a tenir un blog basat en Wordpress. El framework Symfony2 identifica les seves entitats com a taules en la base de dades, en el cas del SDUW totes les entitats definides menys una són per definir les segmentacions sobre el blog.

L'entitat Usuari extén de la classe abstracte 'FOSUser' que incorpora el bundle 'FOSUserBundle' i gestiona la autenticació i l'autorització dels usuaris de la plataforma.

21.1. Com es genera internament una Segmentació?

Una Segmentació estarà relacionada amb una sèrie de 1 a N condicions. Aquestes condicions constaràn d'un tipus de condició i un atribut. L'atribut també constarà d'un tipus per a ell mateix.

L'atribut representa el camp de la taula de la base de dades del blog que es vol consultar el valor. Dins de la entitat d'atribut es guarda el nom, el nom de la taula on es troba i el nom de la columna de la taula que representa a la base de dades.

El tipus de condició ve representat en el formulari de segmentacions com el operador, aquest operador definirà de quina manera s'efectuarà la comparació entre el valor introduït com a 'valor' en el formulari i el atribut que es vol consultar. Dins d'aquesta entitat es guardarà una clàusula SQL amb unes 'literals' que es substituiràn mitjançant expressions regulars per els valors de l'atribut i el valor introduït per l'usuari al formulari a l'hora d'executar la query resultant del procés de creació de la segmentació.

El tipus d'atribut dictarà els tipus de condició disponibles que tindrà cada atribut, ja que no podrà ser la mateixa clàusula de comparació entre una columna que representi dates i una altre que representi una cadena de text.

Un cop definida la segmentació s'utilitzarà un servei que generi la SQL resultant de les relacions de la segmentació amb les entitats esmentades anteriorment i la executarà per tal d'averiguar el nombre de contactes resultants.

21.2. Com s'obtenen les gràfiques del reporting?

Al no constar les entitats del blog de Wordpress al projecte del SDUW, s'ha optat per utilitzar un servei de Symfony2 que generi les sentències SQL necessàries per obtenir els resultats desitjats a la base de dades de Wordpress i tractar la informació mitjançant la API de GoogleCharts per tal de mostrar les dades en forma de gràfiques.

La informació que mostra aquest apartat és el nombre de comentaris per post, el nombre de vegades que s'ha utilitat cada *tag* i el nombre posts per dia.

21.3 Com s'efectua la cerca de contactes?

La idea és molt semblant a la esmentada en la secció del reporting, utilitzant una altre funció del mateix servei que connecta amb les taules del blog, es proporciona la cadena que ha introduït el usuari en el formulari i s'efectua una selecció sobre la taula d'usuaris a partir del valor introduït. Es construeix una sentència SQL a partir les dades que ha facilitat el usuari i es retornen els resultats a la vista per poder generar la taula resultant.

22. 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.

22.1 Llistat de bugs

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

Configurar gestió d'usuaris i firewall	Resolt
Configurar els formularis de condicions amb javascript	Resolt
Guardar correctament la eliminació de segmentacions	Resolt
Restringions de tipus de selectors al formulari	Resolt
No s'aplica els rols a les diferents seccions de l'aplicació	Resolt
Introduïr un 'datepicker' quan es dades siguin dates	Pendent
Amagar els inputs no necessaris al formulari de seleccions	Pendent
Calcular el nombre de contactes d'una segmentació	Pendent

Taula 1. Llistat de Bugs

23. Projecció de 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, per aquest motiu, hi haurà elements que fallin o siguin millorables.

A nivell de projecció 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.

23.1. 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.

- Codi de l'aplicació
 - o Millor tractament d'errors
 - o Efectuar el càlcul de contactes de la segmentació
 - o Validar entitats, taules i relacions de la base de dades
- Funcionalitats
 - o Poder editar una segmentació
 - o Poder recalcular una segmentació
 - o Introduir una pàgina de detall de l'usuari del blog
 - o Exportar usuaris segons cerques o segments
 - o Ampliar el reporting
- Aspecte visual
 - o Afegir un logotip a l'aplicació
 - o Millorar els estils de les taules i la seva paginació
- Usabilitat
 - o Textos d'ajuda per tota l'aplicació
 - o Introducció de tutorials per saber com navegar per l'aplicació

23.2. 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 una secció d'estadístiques més àmplia
- Millorar la transversalitat de l'aplicació, afegint enllaços a altres seccions més detallades en les parts que calgui de l'aplicació.
- Afegir una cerca avançada de contactes segons el seu comportament
- Afegir cerca d'usuaris del blog a partir de les segmentacions
- Moure l'aplicació a un servidor dedicat.
- Tenir una base de dades pròpia i connectar-se remotament a la base del client.

23.3. Futur llunyà

Aquest rang inclouria de 6 mesos a un any, segons el feedback que es vagi obtenint es podria modificar aquest full de ruta, però els objectius principals serien:

- Introduir funcionalitats per efectuar la mateixa feina amb blogs generats amb Drupal.
- Generalitzar les segmentacions per poder-les adaptar a qualsevol base de dades (amb prèvia configuració).
- Incorporar una plataforma d'enviament de campanyes publicitàries a partir de les seleccions.

Com s'ha esmentat anteriorment, l'important en aquest pla de futur és anar creixent aprofitant el feedback obtingut per els usuaris de l'aplicació per tal de crear una aplicació més robusta, d'aquesta manera serà més fàcil adaptar-se a les necessitats del mercat on es vol introduir l'aplicació.

24. Pressupost

A continuació 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	Descripció
Cap de projecte	30.000€	Necessari durant tota la durada del projecte
Programador	21.000€	Només serà necessari durant 4 mesos
Maquetador	18.000€	Només serà necessari durant els últims 3 mesos
Dissenyador	22.000€	Participarà en el projecte durant dos mesos

Taula 2. Recursos humans

Descripció del pressupost sobre recursos físics i digitals:

Concepte	Cost	Descripció
Servidor	0€	A càrrec de la empresa que allotja el blog
Domini	15€ mensuals	-
Local i despeses	600€ mensuals	Inclòs factures, local, equips, etc.

Taula 3. Valanç de recursos no humans

Establerta la durada inicial del projecte de 6 mesos, el cost aproximat en recursos humans seran 30.166€ i 3.690€ en material.

Una vegada entregat el projecte, serà necessari presentar un nou pla de desenvolupament de millores i manteniment de l'aplicació, amb un pla adicional de despeses econòmiques associat.

25. Anàlisi de mercat

Amb l'evolució de les telecomunicacions i internet en la qual està immersa la societat, cada vegada és més important per part d'empreses dedicades al màrqueting el fet de recopilar dades sobre els internautes. Cada vegada més les dades que s'exigeixen són més complexes i les empreses d'anàlisi web no només recopilen "clickstream", sino que cada vegada busquen dades més completes sobre el comportament i gustos dels internautes.

Amb aquesta aplicació s'intenta oferir una solució integrada en un propi *blog* que ja estigui funcionant per tal de recopilar dades sobre els usuaris que hi han interactuat d'alguna manera. Segmentar-los al gust del client i generar-ne informes i estadístiques.

Actualment no hi ha cap empresa gran amb un preu molt competitiu que ofereixi aquest tipus d'aplicació a mida, per tant és una bona oportunitat de crear més diversificació de mercat. Donat al baix cost del projecte es podria sortir a un preu força atractiu.

26. Conclusions

Després de tants anys a la UOC havia arribat un punt on les pràctiques em resultaven moltes vegades repetitives, amb la mateixa estructura i a vegades no trobava la seva utilitat. Molt probablement aquesta sensació és el fruit de molts anys esperant per poder arribar aquest moment compaginant-lo amb diverses feines i en la majoria del temps fent jornada completa i sacrificant temps personal per passar amb la família i amics per poder arribar a aquest moment.

La impressió que esmentava anteriorment s'ha esvaït al realitzar aquest últim projecte, al efectuar aquest treball final de grau, he estat sol davant la planificació i execució d'un projecte, per aquest motiu he hagut de recórrer als coneixements que havia adquirit involuntàriament en totes les pràctiques i PACs cursades i utilitzar-los transversalment per poder dur a terme aquest projecte.

Finalitzar el treball final de grau m'ha donat la oportunitat de trencar amb la rutina del dia a dia al lloc de feina i poder idear, crear un projecte de zero i participar-ne en totes i cadascuna de les etapes de planificació i execució. Al mateix temps he aprofitat l'avinentesa per arrencar un repte personal i formar-me de manera autodidacta en un *framework* basat en PHP anomenat Symfony2 per tal d'assolir el meu objectiu final. Gràcies a aquest fet he descobert una nova manera de programar orientada a objectes i respectant el Model – Vista – Controlador que de ben segur m'obrirà moltes portes professionals de cara al futur.

Durant tot aquest semestre, el SDUW ha sofert varies modificacions i replantejaments donat el poc temps que he disposat mentrestant cursava unes altres tres assignatures que m'han esgarrapat bastant de temps. M'he trobat amb problemes i els he hagut de solucionar adoptant altres vies les quals m'han portat a més carrerons sense sortida, però en acabar puc afirmar que em sento satisfet amb el què he aconseguit i que fins hi tot em sento amb ganes d'acabar el treball que em queda per fer en l'aplicació com a hobby i repte personal.

Amb tots els sentiments i conceptes esmentats anteriorment vull concloure reiterant la meva satisfacció personal per haver realitzat un projecte final de grau que trobo molt necessari i útil de cara a consolidar tots els coneixements adquirits des del principi fins al final. Sobre tot vull destacar el fet de poder gaudir del desenvolupament d'un projecte de principi a fi sense cap interferència, és una situació idíl·lica que espero que pugui repetir en algun moment durant la meua vida laboral.

Annex 1. Lliurables del projecte

Codi

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

- Symfony 2.2
- Codi desenvolupat pel projecte
- Arxius Javascript
- Awesome Font Icons
- Plantilles de totes les vistes de l'aplicació

../PAC_FINAL_prj_BenetSurroca_Marc/

Documentació

Presentació lliure en format Power Point

../PAC_FINAL_prs_BenetSurroca_Marc.pptx

Autoinforme d'avaluació

../PAC_FINAL_auto_BenetSurroca_Marc.pdf

Annex 2. Codi font (extractes)

A continuació recorreré els diferents controladors que hi ha en cada Bundle per tal d'il·lustrar el funcionament de l'aplicació.

CoreBundle

En aquest paquet de codi s'efectuen les funcions internes i del homepage de l'aplicació, així com la definició de serveis interns.

Default Controller:

```
namespace Sduw\CoreBundle\Controller;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
class DefaultController extends Controller
{
    public function indexAction()
    {
        $dbConnection = $this->get('database_connection');
        $generalInfo = $this->get('sduw.wordpress_service')->getGeneralResume($dbConnection);
        $topComment = $this->get('sduw.wordpress_service')->getTopCommentPosts($dbConnection);
        $topTags = $this->get('sduw.wordpress_service')->getTopTags($dbConnection);
        $postsPerDay = $this->get('sduw.wordpress_service')->getNumberPostsPerDay($dbConnection);

        return $this->render('SduwCoreBundle:Default:index.html.twig',
            array('generalInfo' => $generalInfo,
                'topComment' => $topComment,
                'topTags' => $topTags,
                'postsPerDay' => $postsPerDay ));
    }
}
```

WordpressQueryService:

```
namespace Sduw\CoreBundle\Services;
use Doctrine\ORM\EntityManager;
class WordpressQueryService
{
    public function __construct(){}

    public function getGeneralResume($dbConnection)
    {
        $result = array();

        $sql = "SELECT count(*) FROM wp_users";
        $query = $dbConnection->prepare($sql);
        $query->execute();
        $result['userNumber'] = $query->fetchColumn();

        $sql = "SELECT user_registered FROM wp_users ORDER BY user_registered DESC LIMIT 1";
        $query = $dbConnection->prepare($sql);
        $query->execute();
        $result['lastUser'] = $query->fetchColumn();

        $sql = "SELECT wp_posts.post_date, wp_users.user_login FROM wp_posts INNER JOIN wp_users ON wp_posts.post_author = wp_users.ID ORDER BY wp_posts.post_date DESC LIMIT 1";
        $query = $dbConnection->prepare($sql);
        $query->execute();
        $sqlResult = $query->fetchAll();
    }
}
```

```

    $result['lastPost'] = $sqlResult[0];
    return $result;
}

public function getUsersByName($dbConnection, $compareString)
{
    $compareString = mysql_escape_string($compareString);
    $sql = "SELECT * FROM wp_users WHERE user_login LIKE
'%$compareString%'";

    $query = $dbConnection->prepare($sql);
    $query->execute();

    return $query->fetchAll();
}

public function getTopCommentPosts($dbConnection)
{
    $sql = "SELECT wp_posts.ID, wp_posts.post_title, count(*) AS
number_comments
FROM wp_comments
LEFT JOIN wp_posts ON wp_comments.comment_post_ID =
wp_posts.ID
GROUP BY wp_posts.ID, wp_posts.post_title
ORDER BY number_comments DESC";
    $query = $dbConnection->prepare($sql);
    $query->execute();
    return $query->fetchAll();
}

public function getTopTags($dbConnection)
{
    $sql = "SELECT wp_term_taxonomy.count, wp_terms.name
FROM wp_term_taxonomy
LEFT JOIN wp_terms ON wp_term_taxonomy.term_id =
wp_terms.term_id
WHERE wp_term_taxonomy.taxonomy <> 'category'
ORDER BY wp_term_taxonomy.count, wp_terms.name DESC";
    $query = $dbConnection->prepare($sql);
    $query->execute();
    return $query->fetchAll();
}

public function getNumberPostsPerDay($dbConnection)
{
    $sql = "SELECT DATE(post_date) as order_day, count(id) AS count
FROM wp_posts
GROUP BY order_day";
    $query = $dbConnection->prepare($sql);
    $query->execute();
    return $query->fetchAll();
}

//NOT TESTED
public function calculateSegment($dbConnection, $segmentation)
{
    $query = '';
    $selectStatement = "count(*)";
    $allowedMatchingTypes = array('UNION', 'INTERSECT'); // maybe except
in the future

    $ruleMatchingType = strtoupper($segmentation->getMatchingType());
    if (!in_array($ruleMatchingType, $allowedMatchingTypes))
        $ruleMatchingType = 'UNION';

    $segmentQueryString = '';
    if ($query != ''){
        $query .= ' ' . $ruleMatchingType . ' ' ;
    }

    //retrieve conditions of segmentation
    $conditions = $segmentation->getConditions();
    $conditionsAsString = array();
    // get segmentation query
    foreach ($conditions as $condition)
    {

```

```

removed conditions      if (false === $condition->getIsRemoved()) { // clear
removed conditions      $conditionsAsString[] = $this->
>conditionToSql($condition);
    }
    }
    // link segmentation conditions function of the segmentation
matching type          if (count($conditionsAsString) > 0){
    switch (strtoupper($segmentation->getMatchingType())){
    case 'ANY':
        // instead of OR clauses, UNION subqueries are
used to improve the query performance.
        foreach ($conditionsAsString as $cond)
        {
            if (trim($segmentQueryString) != ''){
                $segmentQueryString .= ' UNION ';
            }
            $segmentQueryString .= ' ( '; // wrap
segmentation condition
                $segmentQueryString .= 'SELECT ' .
$selectStatement . ' FROM "wp_users" '.implode(" ", $leftJoins).' WHERE ( ';
                $segmentQueryString .= $cond ;
segmentation condition
                $segmentQueryString .= ') ' ; // wrap
                }
            $segmentQueryString = ' ( ' . $segmentQueryString
. ' ) ' ; // wrap the entire segmentation
            break;
        case 'ALL':
            $conditionsGlue = ' AND ';
            // glue conditions
            $conditionsAsString = implode($conditionsGlue,
$conditionsAsString);
            $segmentQueryString .= ' ( '; // wrap segmentation
condition
                $segmentQueryString .= 'SELECT '
$selectStatement . ' FROM ' . $this->customerDefaultSchema . '"contact"
'.implode(" ", $leftJoins).' WHERE ( ' . $conditionsAsString . ') ' ;
            $segmentQueryString .= ' ) ' ; // wrap segmentation
condition
            break;
        } // switch (strtoupper($segmentation-
>getMatchingType())){
    } // if (count($conditionsAsString) > 0){
        $query .= $segmentQueryString;
    }
    return $query;
}

//NOT TESTED
protected function conditionToSql($condition){
    // return a string similar to: CONDITION_1 BETWEEN 1 AND 3

    $conditionAsQuery = '';

    // get condition attribute
    $attribute = $condition->getAttribute();
    $attrField = '';
    $attrTableName = $attribute->getAttrTable();
    $attrInternalName = $attribute->getInternalName();

    $attrField .= '"' . $attrTableName . '";
    $attrField .= '"' . $attribute->getAttrField() . '";

```

```

    $attrDataType = $attribute->getAttributeDataType();
    $dataTypeName = $attrDataType->getName();
    $customerDataType = $attrDataType->getBaseType();

    $conditionType = $condition->getConditionType();
    $conditionValue = $conditionType->getValue();

    // get get clause pattern from condition type
    $clausePattern = $conditionType->getClause();

    //get the prefix of the query depending of the parameters
    $finalClausePattern = $this->getClausePattern($attrInternalName,
    $dataTypeName, $customerDataType, $conditionValue, $lists);

    switch($dataTypeName)
    {
        case 'date':
            if($attrIsCustom === true)
                $attrField = "(TO_DATE('.$attrField.', 'YYYY-MM-DD'))";

            if(substr($conditionValue, 0, 6) === 'from_x'){
                $param1 = $condition->getValue1();
                $param2 = '';
            }
            else{
                $param1 = date('Y-m-d', strtotime($condition-
>getValue1()));
                $param2 = date('Y-m-d', strtotime($condition-
>getValue2()));
            }
            break;
        case 'datetime':
            if($attrIsCustom === true)
                $attrField = "(TO_TIMESTAMP('.$attrField.', 'YYYY-MM-DD
HH24:MI:SS'))";

            if(substr($conditionValue, 0, 6) === 'from_x'){
                $param1 = $condition->getValue1();
                $param2 = '';
            }
            else{
                $param1 = date('Y-m-d H:i:s', strtotime($condition-
>getValue1()));
                $param2 = date('Y-m-d H:i:s', strtotime($condition-
>getValue2()));
            }
            break;
        case 'integer':
            $param1 = $condition->getValue1();
            $param2 = $condition->getValue2();

            if($conditionValue != "empty")
                $attrField = $attrField."::bigint";
            elseif ($conditionValue != "not_empty")
                $attrField = $attrField."::bigint";
            else
                $attrField = $attrField."::varchar";
            break;
        case 'string': // compare strings in lower case
            $param1 = strtolower($condition->getValue1());
            $param2 = strtolower($condition->getValue2());
            break;
        default:
            $param1 = $condition->getValue1();
            $param2 = $condition->getValue2();
            break;
    }
    // replace __ATTRIBUTE_FIELD__ in clause pattern
    $clausePattern = str_replace('__ATTRIBUTE_FIELD__', $attrField,
    $clausePattern);
    // replace __PARAM_1_VALUE__ in clause pattern
    if (! is_null($condition->getValue1() && $condition->getValue1() != ''
))
        $clausePattern = str_replace('__PARAM_1_VALUE__', $param1,
    $clausePattern);
    // replace __PARAM_2_VALUE__ in clause pattern

```

```

))
    $clausePattern = str_replace('__PARAM_2_VALUE__', $param2,
    $clausePattern);

    $conditionAsQuery = str_replace('__FINAL_CLAUSE__',
    $clausePattern, $finalClausePattern);

    return $conditionAsQuery;
}
}

```

Segmentations Bundle

En aquest paquet s'estableix tota la gestió de segmentacions. També es poden observar procediments que es criden per AJAX i el control d'usuaris per rols.

```

namespace Sduw\SegmentationBundle\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\HttpFoundation\RedirectResponse;
use JMS\SecurityExtraBundle\Annotation\Secure;

use Sduw\SegmentationBundle\Entity\Segmentation;
use Sduw\SegmentationBundle\Form\SegmentationType;

/**
 *
 * Segmentations Controller
 *
 */
class SegmentationsController extends Controller
{
    /**
     * @Secure(roles="ROLE_SEGMENT")
     * Lists all Segmentation entities.
     */
    public function indexAction()
    {
        $em = $this->getDoctrine()->getManager();

        //show only the segmentations that have not been removed
        $entities = $em->getRepository('SduwSegmentationBundle:Segmentation')->getNotRemoved('DESC');

        return $this->render('SduwSegmentationBundle:Segmentations:index.html.twig',
        array('entities' => $entities));
    }

    /**
     * @Secure(roles="ROLE_SEGMENT")
     * Displays a form to create a new Segmentation entity.
     */
    public function newAction()
    {
        $segmentation = new Segmentation();
        $em = $this->getDoctrine()->getManager();
        $request = $this->getRequest();

        $form = $this->createForm(new SegmentationType($em, $this->get('database_connection'), $this->get('security.context')), $segmentation);
        $operatorEquivalenceTable = $this->get('sduw.segmentation_service')->getOperatorSelectors();

        return $this->render('SduwSegmentationBundle:Segmentations:new_segment.html.twig', array(
        'entity' => $segmentation,
        'segmentation_form' => $form->createView(),
        'operatorEquivalenceTable' => $operatorEquivalenceTable,
        'segmentation_form_action' => 'sduw_segmentations_create',

```

```

    ));
}
/**
 * @Secure(roles="ROLE_SEGMENT")
 * Creates the segment and redirects to the list
 */
public function createAction()
{
    // @todo: check roles
    $em = $this->getDoctrine()->getManager();
    $segmentation = new Segmentation();
    $request = $this->getRequest();
    $session = $request->getSession();

    $conditionsErrors = array();
    $conditionsArr = array();

    $segmentationType = new SegmentationType($em, $this-
>get('database_connection'), $this->get('security.context'));
    $form = $this->createForm($segmentationType, $segmentation);
    $form->bind($request);

    if ($form->isValid())
    {
        $segmentationFields = $request->request-
>get('sduw_segmentationbundle_segmentationtype');
        $errors = 0;
        // validate criteria
        // check if exists another criterion with the same name
        $duplicatedSegment = $em-
>getRepository('SduwSegmentationBundle:Segmentation')-
>findOneByName($segmentationFields['name']);
        if ($duplicatedSegment){
            $session->getFlashBag()->set('warning', 'DUPLICATED NAME');
            $errors++;
        }

        if ($errors == 0)
        {
            foreach ($segmentation->getConditions() as $condition)
            {
                $conditionTypeName = $condition->getConditionType()-
>getValue();

                // set datetime by user timezone to fields of type
                datetime
                if ($condition->getAttributeDataType() == 'datetime') {
                    if ($condition->getValue1() != null) {
                        $datetime = new \DateTime($condition-
>getValue1());
                        $condition->setValue1($datetime->format('Y-m-d
H:i:s'));
                    }
                    if ($condition->getValue2() != null) {
                        $datetime = new \DateTime($condition-
>getValue2());
                        $condition->setValue2($datetime->format('Y-m-d
H:i:s'));
                    }
                }

                // for this reason, reset this criterion to the condition
                $condition->setSegmentation($segmentation);
            }

            // CALCULATE CONTACTS HERE //

            $segmentationContacts = 'hola perico';
            // if the result query is empty or null
            if (!$segmentationContacts) {
                $session->getFlashBag()->set('warning', 'WRONG SEGMENT' );
            }
            else {
                //we do not have the tool yet
                $segmentation->setContactsEstimation(0);
            }
        }
    }
}

```



```

        $em->persist($segmentation);
        $em->flush();
    }
}
else {
    $session->getFlashBag()->set('error', 'ERROR VALIDATING THE
FORM' );
}
}
return $this->redirect($this->generateUrl('sduw_segmentations_list'));
}

/**
 * @Secure(roles="ROLE_SEGMENT")
 * Deletes the segment, receives the id by post
 * is called by ajax
 */
public function deleteAction()
{
    $request = $this->getRequest();
    //if it's called by ajax
    if ($request->isXmlHttpRequest())
    {
        $em = $this->getDoctrine()->getManager();
        $parameters = $request->request->all();
        //marks as removed the entity
        $segmentId = $parameters['id'];
        $segmentationEntity = $em->getRepository('SduwSegmentationBundle:Segmentation')->findOneById($segmentId);
        $segmentationEntity->setIsRemoved(true);
        $em->persist($segmentationEntity);
        $em->flush();

        $return = json_encode('ok');

        return new Response($return, 200);
    }
    else {
        throw new \Exception("Error!! Invalid Request", 1);
    }
}
}
}
}

```

ReportBundle

En aquest paquet hi seràn en un futur totes les funcions de generar reports, actualment allotja la cerca d'usuaris.

```

namespace Sduw\ReportBundle\Controller;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
class ReportController extends Controller
{
    public function userSearchAction()
    {
        $usersFound = array();
        $searchFlag = false;
        $request = $this->getRequest();
        //checking if is post
        if ($request->isMethod('POST'))
        {
            $postParams = $request->request->all();
            $stringToSearch = $postParams['contactSearch'];
            //get the database information
            $dbConnection = $this->get('database_connection');

```

```
        //searches the contacts through the service
        $usersFound = $this->get('sduw.wordpress_service')->
>getUsersByName($dbConnection, $stringToSearch);
        //flags the search
        $searchFlag = true;
    }
    return $this->render('SduwReportBundle:Report:users_search.html.twig',
array('usersFound' => $usersFound, 'searchFlag' => $searchFlag));
}
}
```

Annex 3. Llibreries i codi extern utilitzat

A continuació el codi de llibreries de tercers utilitzades en el desenvolupament que s'allotja en el fitxer composer.json.

```
"php": ">=5.3.3",
"symfony/symfony": "~2.4",
"doctrine/orm": "~2.2,>=2.2.3",
"doctrine/doctrine-bundle": "~1.2",
"twig/extensions": "~1.0",
"symfony/assetic-bundle": "~2.3",
"symfony/swiftmailer-bundle": "~2.3",
"symfony/monolog-bundle": "~2.4",
"sensio/distribution-bundle": "~2.3",
"sensio/framework-extra-bundle": "~3.0",
"sensio/generator-bundle": "~2.3",
"incenteev/composer-parameter-handler": "~2.0",
"friendsofsymfony/user-bundle": ~2.0@dev
```

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

- Symfony 2.2
- Doctrine 2.2
- FosUserBundle 2.0

A part també s'ha d'esmentar el framework jQuery de Javascript i la API de Google de gràfiques per part de les vistes de l'aplicació.

Annex 4. 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.

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.

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.

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.

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

Annex 5. Bibliografia (webgrafia)

Varis autors (2001) **Creative Commons**. <http://www.creativecommons.org>

Consultat el 12/03/2014

Gabor Ivan (2010). "**LAMP on a Virtual Box**" <http://lamp.gixx-web.com/chapter1.html>

Consultat el 16/03/2014

Anil Desai (2011). "**What is a virtual Machine?**"

<http://searchservirtualization.techtarget.com/definition/virtual-machine>

Consultat el 24/03/2014

Tad Chef (2014). "**Advanced User Segmentation with WordPress, Google Analytics + SiteApps**" <http://blog.ahrefs.com/responsive-seo-advanced-user-segmentation-wordpress-google-analytics-siteapps/>

Consultat el 25/03/2014

Varis autors (2014). "**Wordpres Codex**" <http://codex.wordpress.org/>

Consultat el 25/03/2014

PMQuality (2013). "**Que es el diseño UX?**"

<http://pmqlinkedin.wordpress.com/about/que-es-el-diseno-ux/>

Consultat el 05/04/2014

SensioLabs (2013). "**The Symfony Cookbook**"

<http://symfony.com/doc/2.1/book/security.html>

Consultat el 06/04/2014

Fabien Potencier (2013). "**Security**" <http://symfony.com/doc/2.1/book/security.html>

Consultat el 06/04/2014

Fabien Potencier (2013). "**Testing**" <http://symfony.com/doc/master/book/testing.html>

Consultat el 06/04/2014

Sensiolabs (2014). "**How to Embed a Collection of Forms**"

http://symfony.com/doc/current/cookbook/form/form_collections.html

Consultat el 13/04/2014

Christophe Coevoet (2014). "**FOSUser Bundle documentation**"

<https://github.com/FriendsOfSymfony/FOSUserBundle/blob/master/Resources/doc/index.md>

Consultat el 12/04/2014