

# **Escola Universitària Politécnica de Mataró**

Centre adscrit a:



**UNIVERSITAT POLITÈCNICA  
DE CATALUNYA**

**Enginyeria Informàtica de Gestió**

**COMUNITAT MIRTH**

**VERSIÓ 1.0**

**Memòria**

**FELIPE CORTÉS SAYAVERA  
PONENT: EDUARD DE BRU DE SALA**

**TARDOR 2011**



**TecnoCampus  
Mataró-Maresme**



## **Dedicatòria**

A la meva dona **Verónica** per suportar el pes dels meus somnis.

**Als meus pares** pels valors que m'han inculcat i ensenyar-me a valorar-me tal com soc.

**Als meus germans** per la seva valentia a l'hora de prendre decisions.

**Als professors i companys** que m'han ensenyat que la formació no té edat.

**Als meus companys de feina** per valorar la cooperació per sobre de la individualitat.

**El meu fill** per ensenyar-me a valorar el temps.

**El sr. Manel Domingo Falcón** per ser exigent amb les meves possibilitats.



## **Agraïments**

No seria just si no dediques aquest espai a tots aquells docents que en algun moment m'han donat el seu suport i els seus ànims sobretot quant les coses no surten com s'espera ,  
graciés a tots.



## **Resum**

L'objectiu d'aquest projecte es crear una xarxa social on el flux de arxius sigui intens, la creació de grups de treball on no només es pugi compartir coneixements amb preguntes i respostes si no el poder compartir arxius comuns de grups. El aspirant tindrà com objectiu el aprenentatge del llenguatge php i de la eina Wordpress, i com des de un llenguatge no preparat a ser orientat a objectes es pot aconseguir orientar-lo amb una mica de imaginació.

## **Resumen**

El objetivo de este proyecto es crear una red social donde el flujo de archivos sea intenso, la creación de grupos de trabajo donde no sólo se pueda compartir conocimientos con preguntas y respuestas si no el poder compartir archivos comunes de grupos. El aspirante tendrá como objetivo el aprendizaje del lenguaje php y de la herramienta wordpress, y como desde un lenguaje no preparado a ser orientado a objetos se puede conseguir orientarlo con un poco de imaginación.

## **Abstract**

The objective of this project is to create a social network where the flow is heavy files, creating working groups where he can not only share knowledge with questions and answers, if not the common file sharing groups. The applicant will target the language learning php and wordpress tool, and since a language is not prepared to be guided to orient objects can be achieved with a little imagination.





# Índex.

Índex de taules.....	VII
Glossari de termes. ....	IX
1. Objectius.....	1
1.1. Propòsit. ....	1
1.2. Finalitat. ....	1
1.3. Objecte. ....	1
1.4. Abast. ....	1
2. Planificació (Abans de començar). ....	3
2.1. Establir Objectius amb el Client. ....	3
2.1.1 Establir uns objectius reals.....	3
2.1.2 Establir els objectius SMART. ....	3
(Específics, Mesurables, Arribables, Realistes, Temps).....	3
2.1.3 Definir els objectius per escrit.....	4
2.1.4 Crear un pla d'acció.....	4
2.1.5 Mostrar continuïtat.....	5
2.2. La matriu urgent/important. ....	5
2.3.La taula de tasques. ....	6
3. Avantprojecte. ....	9
3.1.Fases.....	9
3.1.1 Projecte assignat. ....	9
3.1.2 Recollida d'informació anàlisi funcional.....	9
3.1.3 Selecció eines a utilitzar i material informàtic. ....	10
3.1.4 Formació sobre la eina. ....	10
3.1.5 Formació sobre el llenguatge a utilitzar. ....	11
3.1.6 Programació dels Requisits explícits per el client. ....	11
3.1.7 Documentació i redacció de la memòria. ....	12
3.1.8 Tester de funcionalitats i retocs finals. ....	12
3.1.9 Fer presentació del projecte i retocs en la memòria. ....	13
3.2 Cronograma. ....	14
4.Col·laborar en una comunitat. ....	15
4.1 Que es la col·laboració.....	15

4.2 Quin es el valor afegit d'aquesta comunitat?.....	16
4.2.1 Abans de continuar coneixem una mica més la plataforma Mirth.....	17
4.2.2 Perquè surt la idea de crear una comunitat ?.....	19
4.2.3 Quina es la motivació de cara l'usuari ?.....	19
4.2.4 Quina es la motivació de cara l'estudiant ?.....	19
5. Que es la eina Wordpress.....	21
5.1 Aprofundim en Wordpress i els seus components.....	21
5.1.2 Base de dades inicial del sistema.....	22
5.1.3 Sistema tecnològic i funcional de la eina.....	25
5.1.4 Com es controlen els arxius.....	26
5.1.5 Sistema de puntuació.....	28
5.1.6 Els fluxos d'arxius entre usuaris.....	29
6. Les parts de l'aplicació.....	30
6.1 La explicació dels menús.....	30
6.1.1 El menú d'Inici.....	30
6.1.2 El menú d'Activitat.....	32
6.1.3 El menú Contactes.....	35
6.1.4 Barra lateral.....	36
6.1.5 Barra Superior.....	36
6.2 La explicació dels directoris i filosofia d'objecte en php.....	38
6.2.1 Dins del propi tema Comunitat Mirth.....	38
6.2.2 Filosofia d'objecte en php.....	39
6.3 Perfils d'usuari i la seva activitat.....	40
6.3.1 Perfils d'usuaris.....	40
6.3.2 Activitats per usuaris diferenciadores.....	42
6.3.3 Activitats comuns.....	44
7. Possibles millores de la Aplicació.....	51
7.1 Millores de funcionalitats.....	51
7.2 Millores no funcionals.....	52
8. Conclusions del Estudiant.....	53
8.1 Sobre el treball realitzat.....	53
8.2 Sobre la formació obtinguda.....	55

9. Bibliografia i webs consultades. ....	56
9.1 Bibliografia .....	56
9.2 Webs consultades.....	57
10. Cost del prototip. ....	59
10.1. Costos de recursos humans. ....	59
10.2. Amortització equips, instrumental i software. ....	60
10.3. Cost total. ....	61



## Índex de figures.

Fig. 3.2 Dies Treball	14
Fig. 3.2-1 Cronograma Organitzatiu	14
Fig. 4.2.1 Arquitectura Mirth (documentació oficial).	17
Fig. 4.2.1-1 Estructura de Canals	18
Fig. 5.1 Menú Mirth “Tema propi”	22
Fig. 5.1.2 Model BBDD BuddyPress i Wordpress	23
Fig. 5.1.3 Arquitectura Utilitzada	25
Fig. 5.1.4 Model de Entrada de arxius	27
Fig. 5.1.5 Taules wp-users Wordpress	28
Fig. 6.1.1 Pantalla Inici	31
Fig. 6.1.1-1 Pujar arxiu	31
Fig. 6.1.2 Pantalla Activitat	32
Fig. 6.1.3 Pantalla Contactes	35
Fig. 6.1.4 Barra Lateral	36
Fig. 6.3.1 Possible millora creixement usuari	41
Fig. 6.3.2 Creacions PDF	43
Fig. 6.3.3 Pantalla Puntuacions	48
Fig. 6.33-1 Pantalla Àlbum	51
Fig. 8.1 Modularització	53



## **Índex de taules.**

Taula 2.2 Matriu urgent important	5
Taula 2.3 : Tasques i Temps.	7
Taula 8.1 Comparatives PHP - JAVA	54





## **Glossari de termes.**

EUPMT Escola Universitària Politècnica de Mataró

PFC Projecte Final de Carrera

TFC Treball Final de Carrera

CCI Centre de Competències de Integració

OO Orientació a Objectes

BBDD Bases de Dades

HTML HyperText Markup Language

CSS Cascading Style Sheet

TCM Tecnocampus Mataró



# **1. Objectius.**

## **1.1. Propòsit.**

El propòsit es generar una comunitat de usuaris enfocat en la plataforma de Mirth, actualment no existeix cap tipus de know-how per a desenvolupadors de aquesta plataforma sent de pagament qualsevol formació que es volguí obtenir.

## **1.2. Finalitat.**

La finalitat real d'aquest projecte es facilitar les eines necessàries per la creació i formació de programadors i empreses que desenvolupant amb la eina Mirth, fent grups de formació, compartir arxius per grups, o fins i tot compartir-ho amb contactes, creant entorns de treball definits per el usuari.

## **1.3. Objecte.**

En aquest document el que s'intenta es exposar alguns dels aspectes més teòrics de la organització i abast de un TFC poden agafar com exemple el anàlisis adjuntats en aquest document.

Secundàriament s'explicarà la visió del treball realitzat per la creació de una comunitat virtual.

## **1.4. Abast.**

Els requisits demanats per el client son sense cap dubte l'Abast d'aquest projecte, en alguns casos per la relació "temps – requisits" es impossible arribar-hi, fet que s'explicarà en aquest document, per aquest motiu el projecte és més un prototipus o versió inicial 1.0.



## 2. Planificació (Abans de començar).

### 2.1. Establir Objectius amb el Client.

La definició dels objectius es una etapa fonamental, aquest procés comença amb un estudi del que es desitja fer i finalitza amb un treball eficaç, sempre que es puguí aquest objectius en de estar formalitzats amb paper, amb anàlisis previ i sense buits.

Podíem definir cinc etapes clares per definir aquest objectius.

#### 2.1.1 Establir uns objectius reals.

Quant succeeix això s'han de fixar objectius que estiguin d'acord amb les prioritats de la seva vida, per aconseguir-ho, haurà de poder complir-los, un consell es intentar escriure que es mes valuós i què importància té per el estudiant, i el client, d'aquesta manera ens evitarem entrar en objectius pràcticament inassolibles.

#### 2.1.2 Establir els objectius SMART.

**(Específics, Mesurables, Arri bables, Realistes, Temps).**

Existeixen moltes interpretacions del acrònim SMART, però en un principi fonamental es el següent:

**Specific (eSpecífics):** Un objectiu específic a de estar definit de manera clara i precisa respecte a la aplicació, en aquest cas un dels objectius clars que es va deixar des de un bon principi va se que la comunitat havia de tenir la possibilitat de crear grups de idees.

**Measurable (Mesurables):** Els objectius han de tenir algun tipus de mesura, hem de creure que algunes funcionalitats no tenen la mateixa feina així doncs s'han de establir mesures com dates, o temps en hores de treball, a més de saber quin beneficií obtindrem del temps, en hem de fer preguntes com, es gran el benefici en comparació al pes del treball ?, tot això ens servirà per poder redactar un bon pressupost.

**Achievable (Assolibles):** Si resulta que el projecte es massa difícil o per fer una sola tasca necessitem molt de temps, l'únic que aconseguirem serà perdre confiança en nosaltres mateixos, en definitiva s'ha de buscar un equilibri entre dificultat i resultat.

**Realistic (Realistes):** Els objectius han de coincidir amb els estudis realitzats i la orientació del projectista, amb la finalitat de obtenir bons resultats en el projecte.

**Time bound (Limitats en el Temps):** S'ha de procurar establir uns objectius que siguin directament proporcional amb el temps del projecte, entre altres coses no es pot pretendre arribar més enllà del temps que disposem, en aquest cas observem que per la realització del projecte disposem exactament de 4 mesos, si adjuntem alguns requisits no funcionals com “s’ha de fer amb aquest llenguatge” o “s’ha de utilitzar aquesta eina”, llavors trobem que el temps de formació augmentarà mentre que el temps de creació disminuirà considerablement, per aquest motiu s’ha de identificar bé els objectius dels quals arribarem i quins no.

*NOTA:*

*Les definicions de objectius no es limiten en una simple declaració de intencions.*

*Sense una definició rigorosa i precisa del seu objectiu, i una consciència clara de les raons que ho motiven, les seves possibilitats de èxit es redueixen considerablement.*

### 2.1.3 Definir els objectius per escrit.

La primera entrevista que es va realitzar amb el Client i el meu Ponent es van definir una sèrie de objectius o requeriments tals que alguns van quedar clar i altres no, la pràctica de escriure els objectius i deixar notes com per exemple (**Faré** en comptes de **M’agradaria**) fan que els objectius siguin més clars i motivadors, en les reunions s’ha de intentar escriure tot allò que sembla que pot ser important, frases com la aplicació haurà de fer...., son dignes de atenció.

### 2.1.4 Crear un pla d’acció.

Molt sovint, aquesta etapa no s’inclou en el procés de definició dels objectius , el pla d’acció no es una altre cosa que intentar escriure molt resumidament (com si fossin títols) en un full totes les passes a seguir, aquest full a de acompanyar el Enginyer en la creació del projecte taxant les fases ja fetes per assegurar-se el seu seguiment.

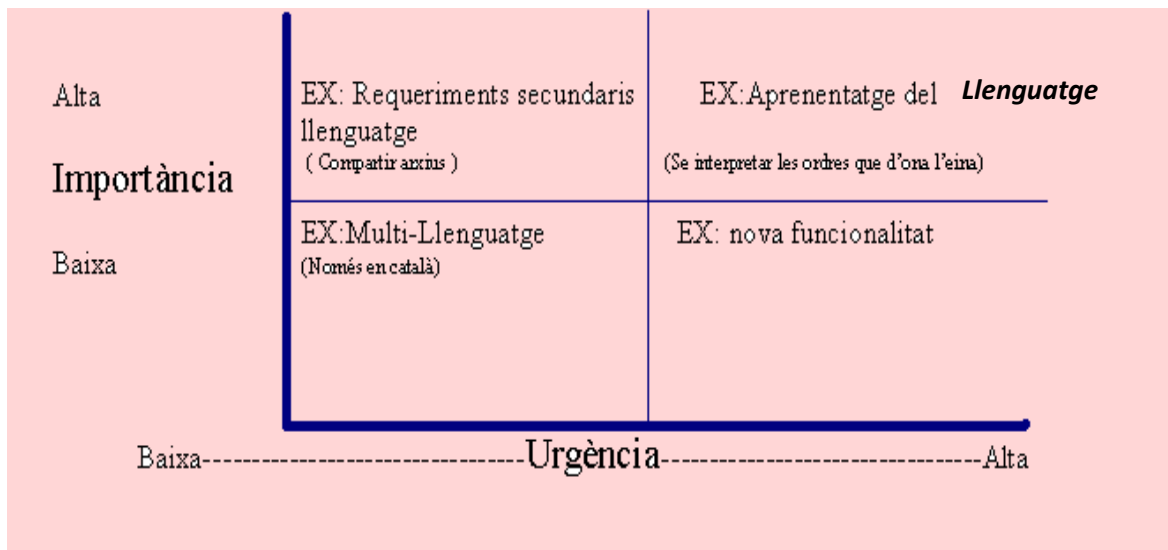
### 2.1.5 Mostrar continuïtat.

Les reunions amb el client han de ser sovint, si una cosa no queda clara doncs fer una reunió amb el client per descriure aquells dubtes i buits que no queden ben definits, es l'escenari ideal, a passar que per un TFC es poden obviar algunes reunions.

## 2.2. La matriu urgent/important.

Per poder gestionar el temps de manera eficaç i desenvolupar bé un projecte, es necessari dedicar un temps a tasques que siguin importants, i no solament urgents.

- Les activitats importants contribueixen, amb el seu resultat, a la realització dels objectius.
- Les activitats urgents, son aquelles que tenen o requereixen la atenció immediata, de tant en tant estan associades a problemes que s'ha de resoldre o situacions que no son del tot agradables.



**Taula 2.2: Matriu Urgent Important**

Les reunions amb el client son de vagades molt abstractes els clients expliquen que es el que desitgen però amb un àmbit de importància molt lineal, per exemple en el cas que ens avarca de cara al client es tant important poder crear un grup de idees com poder canviar d'idioma en qualsevol moment, però per nosaltres (s'entén l'espitant) la necessitat es equiparada amb la importància, per aquest motiu els requeriments s'han de puntuar el

exemple clar es intentar fer una puntuació del 1 al 5 podent d'aquesta manera veure i avaluar quins son els graus dels requeriment i per quin hauríem de començar.

En la matriu urgent - important, hi ha 4 rectangles que es defineixen com:

**Activitats Urgents alta e Importants alta:** Son aquelles que s'han de fer al principi del projecte, un dels requeriments era que s'havia de fer amb la eina Wordpress que esta creada amb llenguatge PHP doncs una de les activitats importants es poder adaptar aquesta eina i formar-se en el llenguatge de creació.

**Activitats Urgents alta e Importants baixa:** Aquestes activitats normalment acostumen a ser **urgents però no importants** una nova funcionalitat que en un moment donat al client creu que seria bona aplicar-la i que no ha estat comptabilitzada la podríem anotar com una interrupció.

**Activitats Urgents baixa e Importants alta,** el exemple clar es la funcionalitat de poder pujar arxius per compartir , hem de entendre que no es un requisit pròpiament comptabilitzat o dit directament per el client però si creem una comunitat per poder compartir coneixement es necessari poder compartir arxius.

**Activitats Urgents baixa e Importants baixa,** de vegades per el temps que ens dona el client hi ha requisits que no arribarem, al ser una versió 1.0 uns dels requisits que es poden desmarcar en aquesta línia es el multi-llenguatge, dependrà de l'estudií de l'ús de la mateixa, la assignació de nous llenguatges.

### 2.3.La taula de tasques.

Una vegada definit bé els requisits tan els funcionals com els no funcionals,(entenem funcionals definits com a "l'aplicació hauria de fer..." i els no funcionals definits com a "l'aplicació s'hauria de fer amb ...") , es important crear un sistema de actuació, aquest el definim amb un diagrama de Gantt, per començar el primer que vaig fer es crear una llista amb les tasques a realitzar.

Per cada tasca, s'ha de indicar la data de inici, duració estimada, i el tipus si es paral·lela o seqüencial, si son seqüencials s'ha de indicar a què etapa depèn.



D'aquesta manera el que vaig obtenir es una llista com la següent, hem de deixar clar les dates de inici i de fi del projecte, llavors calcularem des de el dia 3 de octubre del 2011 fins el dia 31 de gener del 2012, un total de 122 dies o dir d'una altre manera, 17 setmanes i 3 dies, per suposat això va variar en dependència de les necessitats, problemes etc...

**Taula 2.3: Tasques i Temps**

Tasca	Inici	Duració	Tipo	Depèn de...
Anàlisi Funcional	Setmana [1..3]	21 dies	Seqüencial	-----
Selecció eines a utilitzar i material informàtic	Setmana 4	7 dia	Seqüencial	1
Formació sobre la eina	Setmana [5..8]	21 dies	Seqüencial	1
Formació sobre el llenguatge a utilitzar	Setmana[9..10]	15 dies	Paral·lela	3
Programació dels Requisits explícits per el Client	Setmana [11..15]	30 dies	Seqüencial	4
Documentació i redacció de la memòria	-----	7 dies	Paral·lela	----
Tester de	Setmana[16..17]	15 dies	Seqüencial	5

funcionalitats i retocs finals				
Fer presentació del projecte i retocs en la memòria.	Setmanes 17	5 dies	Paral·lela	6
TOTALS	SETMANES 17	122 DIES		

### 3. Avantprojecte.

Abans de començar amb el projecte s'ha de definir bé en el context de l'Avantprojecte, es van definir les passes comentades en el punt anterior.

L'avantprojecte a de definir de forma general les diferents fases o actuacions que es planifiquen per el projecte COMUNITAT MIRTH, com a planificació de forma genèrica i general agafarem les esmentades en el punt 2.3-La taula de tasques.

#### 3.1.Fases.

##### 3.1.1 Projecte assignat.

Abans de fer la primera entrevista amb el client vaig poder veure la documentació del projecte assignat, en aquest sentit el concepte era "Xarxa Social", a partir d'aquí vaig entendre que les dimensions que avarca una xarxa social son bastant amples.

No es un tema nou i existeixen diferents documents de estudis del que s'ha de fer per crear una xarxa social, però una de les intencions del client es poder mantenir la aplicació amb un mínim d'esforç.

##### 3.1.2 Recollida d'informació anàlisi funcional.

La recollida d'informació té com a finalitat reduir les possibles errades a l'hora de definir el projecte s'ha de perdre un temps a entrevistar-se amb el client, i saber que es el que desitja, no només ens trobarem coses que el client desitja sinó que el client acostuma a demanar recordem els requisits funcionals i no funcionals.

#### NOTA:

*Una vegada entrevistat el client als meus ulls el concepte de "Xarxa Social" crec que no es l'adequat per el futur del projecte, així doncs decidim conjuntament canviar el concepte per comunitat, ja que està definida per un cercle molt concret d'usuaris, com exemple podríem veure la població espanyola de 48 milions no tots estan a l'atur, els aturats comparteixen un mateix vincle l'atur doncs això es una comunitat.*

### Etapes de recollida d'informació:

- Cerca d'informació inicial tecnològica
- Cerca de productes semblants al projecte a desenvolupar

### Objectius de recollida d'informació:

- Tenir una visió general del projecte a desenvolupar
- Detectar les necessitats dels diferents perfils
- Identificar les tecnologies més adients per preparar-ho.

### 3.1.3 Selecció eines a utilitzar i material informàtic.

Durant aquesta fase es dur a terme la primera implementació i construcció.

**Prototip Inicial:** Gràcies a la premissa de que el projecte s'ha de realitzar de manera que en un futur sigui lo mes mantenible possible, es va decidir crear la aplicació **en Java**, el motiu no va ser un altre que la seva **fàcil implementació i manteniment**, a l'hora de penjar-ho en un hosting de lloguer no hauríem de tenir problemes ja que cada vegades son

més els que suporten jsp i tenen possibilitats de configuració amb biblioteques com, exemple podem trobar el java-hosting.es

En definitiva la implementació orientada a objectes faria que el projecte fos fàcilment modificable en un futur sense haver de dependre de altres eines, amb possibilitats de introduir biblioteques i aplicacions externes, no necessàriament fetes en java, es va presentar a finals de setembre un prototip funcionant fet amb aquest llenguatge.

**Prototip segons requisits no funcionals:** Després de la primera presentació el client decideix que la aplicació o comunitat es faci amb una eina anomenada **Wordpress**.

### 3.1.4 Formació sobre la eina.

El període de formació d'aquesta eina es realment senzill si el que es vol es fer es una aplicació a nivell usuari, sense activitat de programació, el gran ventall de plugins disponibles per aquesta eina es realment gran.

**El problema recau** si el que es vol es modificar amb codi propi i requeriments demanats per el client, el fet de enfocar els requeriments funcionals a les demandes de clients fa que

el codi sigui embolicat de fer funcionar. Entre altres coses per la navegació de la maquina central de Wordpress.

### 3.1.5 Formació sobre el llenguatge a utilitzar.

El llenguatge que utilitza aquest tipus de eina es php, en les versions més antigues encara no existia la possibilitat de crear classes, a partir de la versió 5 es poden generar, no oblidem que estem parlant de classes i no de objectes.

Realment l'aprenentatge de aquest llenguatge una vegada fet java no es gaire complicat fora de saber la nomenclatura que ja es una aspecte important a tenir en conte, els dies dedicats en veure el seu funcionament es realment curt ja que es un llenguatge de sentencies , les confusions es poden generar entre classes de php i del llenguatge més utilitzat en vida formativa de la universitat (OO), ja que php interpreta el arxiu tal com es si hi ha una funció pública la executarà la primera vegada que entri i a l'arxiu, es cridi o no des de fora aquesta funció.

En aquest TFC s'ha intentat orientar a classes a passar de que es en alguns casos quasi era impossible.

### 3.1.6 Programació dels Requisits explícits per el client.

Es van definir en un principi diferents requisits que el client volia de cara a la creació de la aplicació que explico de forma molt general, i alguns que no apareixen per ser obvis com per exemple la creació de un avatar per el usuari, entre altres vaig trobar en la primera reunió.

Funcionals (La aplicació ha de fer)

- Alta Clients
  - Definicions de Perfils.
    - (Usuaris, Empreses)
- Perfil Client
  - Definició de perfil
  - Automatitzar el Currículum
  - .....
- Multimèdia
  - Incrustar el canal de CCI de youtube en la aplicació.
  - Possibilitat de descarregar documents pdf des de la aplicació
  - Possibilitat de pujar documents des de un comentari.
  - .....
- Comunicació
  - Creació de grups de idees.
  - Recerca de grups, membres.

- Envio de missatges privats.
  - Creació de curssets per programadors i principiants (Empreses)
  - Creació de ofertes de treball (Empreses)
  - Notificacions de acceptació de contactes.
  - Puntuacions de les aportacions a la comunitat.
  - .....
- Navegabilitat
    - Possibilitat de canvi de password
    - Possibilitat de canvi de idioma
    - Possibilitat de descarregar el currículum.
    - .....

No cal dir que entre tots els requeriments esmentats com a aspirant he de seleccionar aquells que crec son més importants per el funcionament de la aplicació, buscar el equilibri entre temps i requisits.

### 3.1.7 Documentació i redacció de la memòria.

Durant tot el procés de creació es van prenen notes que acostumen a ser molt informals, es obvi que aquestes notes tenen les seves funcions, com a aspirant a Enginyer aconsello que es mantingui una llibreta durant tot el procés que anomenarem la llibreta de bitàcola, més que no pas unes fulles que acostumen a ser un problema a l'hora de ordenar-ho, es important escriure la data i el temps de realització, una vegada s'hagi de documentar la memòria utilitzarem aquesta guia o llibreta de bitàcola per gestionar bé una memòria.

### 3.1.8 Tester de funcionalitats i retocs finals.

Els testers s'han de fer de manera gradual, un requisit implementat i provat es millor que 4 requisits implementats i provats a ala vegada, els testers s'han de fer per funcionalitats, sempre podrem fer un tester al finalitzar tota la aplicació, es necessari omplir la nostra bbdd per garantir l'èxit, abans de fer els retocs finals s'ha de fer una backup de la aplicació finalitzada, els retocs finals acostumen a ser molt ràpids e imprecisos.

### **3.1.9 Fer presentació del projecte i retocs en la memòria.**

Arribant aquest punt m'he de plantejar des de quin punt he de fer la meva presentació, el rumb que agafí pot ser definitiu de cara a la nota, com que no es un projecte que jo hagi plantejat he de ser realista amb la idea plantejada per el client

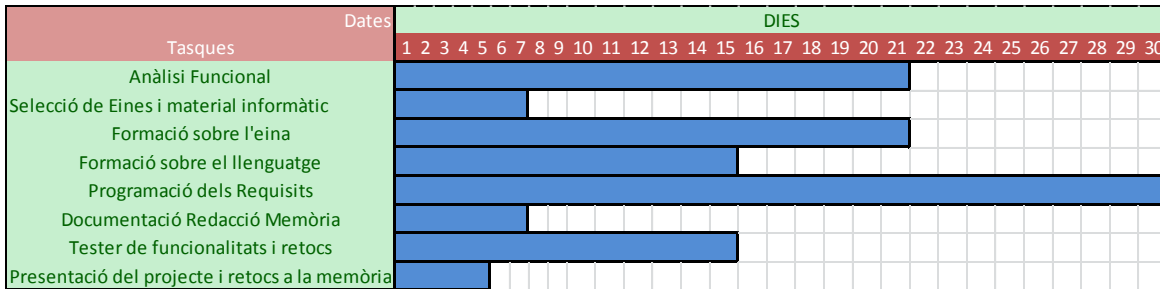
El repàs a la memòria es essencial ja que sempre et pots trobar sorpreses que son dignes de fer-les conèixer.

### 3.2 Cronograma.

**Figura 3.2: Dies de treball**

#### Dies de treball

En el diagrama següent trobem les intensitats de les diferents fases en temps podem observar que la implementació pròpiament dita ens treu 30 dies de feina calculades a 4 hores diàries, càlcul principal de 120 hores.

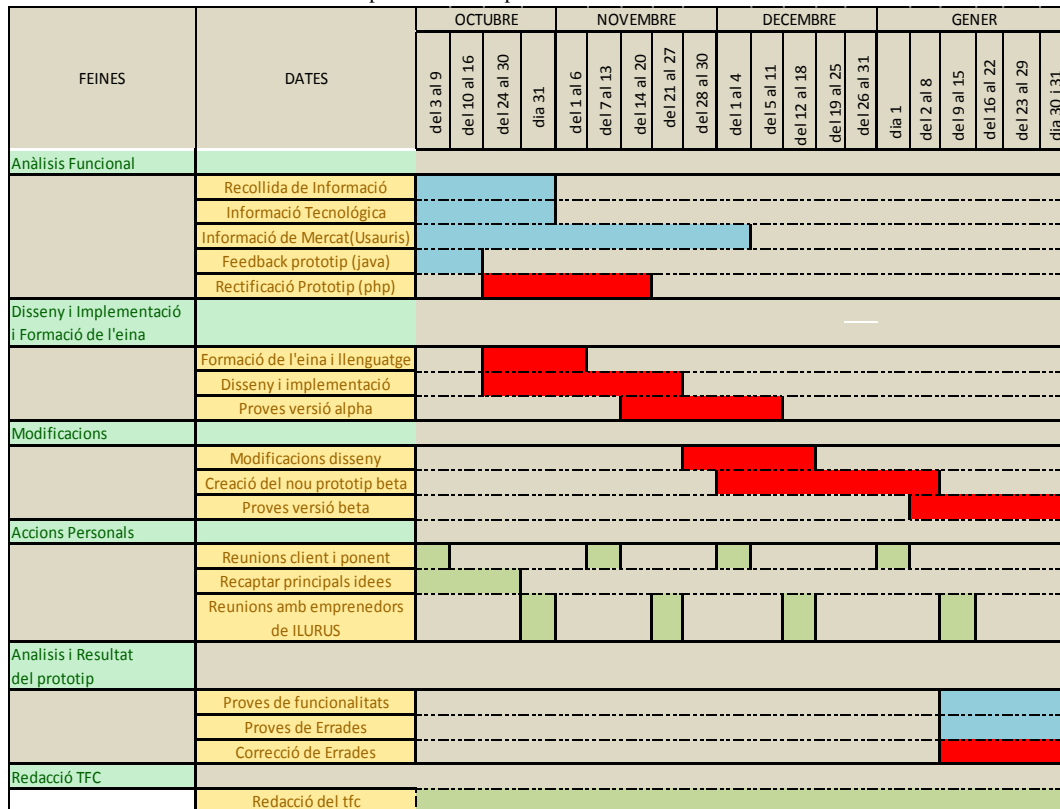


**Figura 3.2-1: Grantt**

#### Figura de organitzatiu Gantt (Inicial)

En aquest diagrama observem dues coses:

- 1- les intensitats de elaboració marcades en vermell, algunes tan importants com la creació del prototip, amb menys intensitat marcades en blau, i les mes distretes com reunions amb el client o amb altres emprenedors.
- 2- La durabilitat de la feina i la seva dispersió en el temps





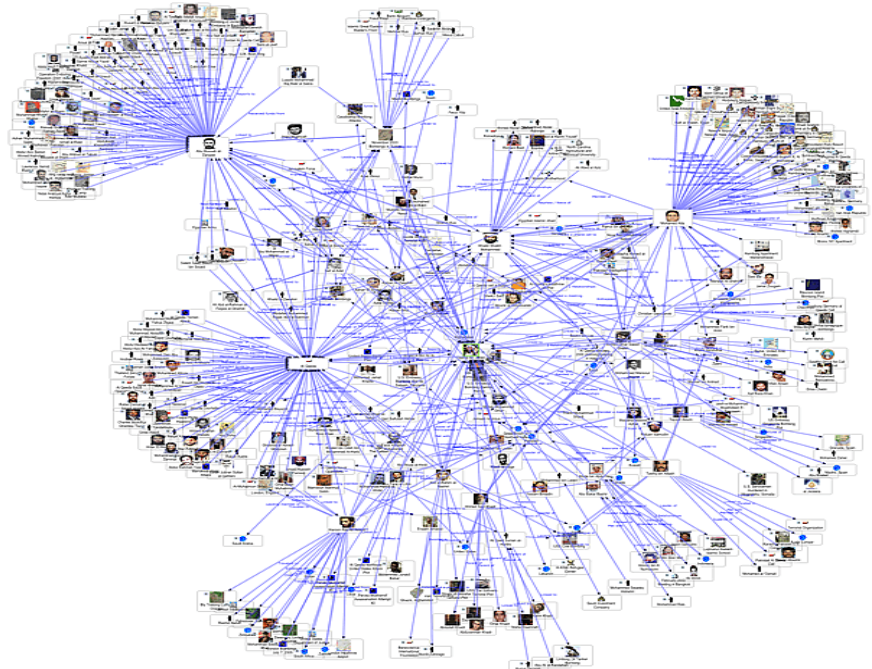
## 4.Col·laborar en una comunitat.

### 4.1 Que es la col·laboració

**Treballar en conjunt amb altres persones en una tasca comuna**, aquest es el significat més general per la col·laboració, la manca de documentació gratuïta i traspàs de coneixements en aquesta eina fa sorgir la necessitat de crear una comunitat per a la creació de grups de idees o simplement crear comentaris per aquest perfil de usuaris.

Aquesta va ser la principal premissa que va canviar el projecte de nom, després de entrevistar-me amb el client i explicar les seves necessitats es va arribar a la conclusió que no s'estava demanant una xarxa social sinó que el que es volia realment era una comunitat, no va necessària la meua

explicació, el client després de rumiar es va donar conte que s'avia de canviar de nom només al examinar el requisits que estava demanant. Gràficament la diferencia com podem observar en la imatge no existeix un cercle comú per els usuaris sinó que cada usuari té els seus interessos en els quals els pot compartir o no amb altres usuaris creant de vegades cercles de



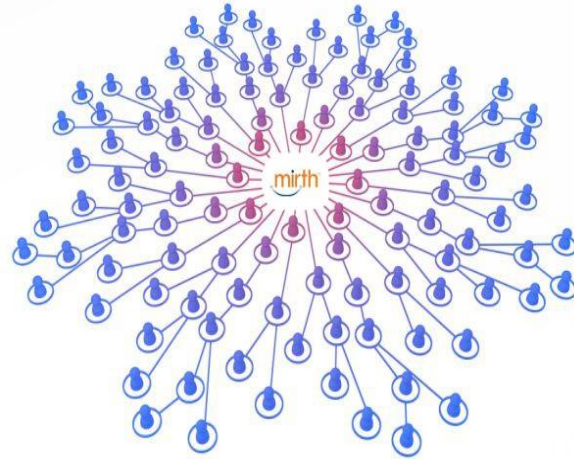
amistat, o simplement de comunicació sense un tema en concret, una altre diferencia es que les comunitats son eines que faciliten la col·laboració sent aquestes definides en el àmbit de "treball", en resum amb al aspecte més lúdic diríem que una xarxa social no existeix la col·laboració sinó la comunicació sense més preàmbuls ni tema concret.

Si que es cert que cada vegada més les aplicacions de xarxes social intenten administrar comunitats creades per els usuaris, ¿podem interpretar d'aquesta manera que les xarxes socials ja adquireixen la possibilitats de crear comunitats ?, si sens dubte però de moment no existeixen comunitats professionals ja que en el més ampliàt significat relacionem xarxa social amb oci.

¿Llavors com definiríem el projecte que s'ha desenvolupat ?, diríem que es una comunitat que ens permet crear diferents comunitats dins d'aquesta, un exemple crear una branca que només parlés de com instal·lar les eines necessàries.

Com he comentat les comunitats han de ser atractives per els usuaris així doncs si creem una comunitat de programadors hem de contemplar la possibilitat dels diferents llenguatges.

En algunes comunitats ja creades com [www.lawebdelprogramador.com](http://www.lawebdelprogramador.com) , o [www.forosdelweb.com](http://www.forosdelweb.com), ens trobem que els grups o temes ja estan creats, tals com php, css, etc..., i dins de aquest les derivades d'aquest llenguatges com el cas de css trobaríem el html ja que estan lligats, la intenció es que els usuaris puguin crear aquest grups de treball, que no estiguin definits d'entrada així doncs hi existeix la possibilitat de crear cercles de treballs de col·laboració, però sempre amb un mateix tema la dedicació a la plataforma mirth, en la primera versió la part de administració hem definit que sigui l'usuari que ha creat el grup qui tingui aquesta potestat, més endavant explicarem com.



En definitiva una comunitat genera sentiment de pertinença, els que formen part d'aquesta es senten identificats amb el grup, aporten alguna cosa i també reben alguna cosa d'aquesta, existeix un valor afegit. No totes les xarxes socials compleixen aquest paràmetres. Per tant tota comunitat intrínsecament es una xarxa social, però no succeeix així en sentit contrari, no totes les xarxes socials son una comunitat.

## 4.2 Quin es el valor afegit d'aquesta comunitat?

Amb la eina open source de Mirth no existeixen gaires traspassaments de coneixements, ni tan sols algunes bases de tutories tot bàsicament es basa en el fòrum creat per la empresa en qüestió (<http://www.mirthcorp.com/community/forums>) , si es desitja anar més enllà de simples preguntes i respostes necessites la compra del suport, definiríem que sí, que la eina Mirth es open source i el seu model de negoci es basat en el suport.

#### 4.2.1 Abans de continuar coneixem una mica més la plataforma Mirth.

La integració de la clínica electrònica amb el resta de sistemes hospitalaris o sanitaris, es el principal objectiu de aquesta plataforma.

La idea principal de Mirth es entendre que es una caixa negra amb regles i transformadors, que permeten la adaptació dels missatges HL7 a altres sistemes. Es un dispositiu intermedi que evita haver de desenvolupar altres canals de adaptació en les aplicacions a integrar.

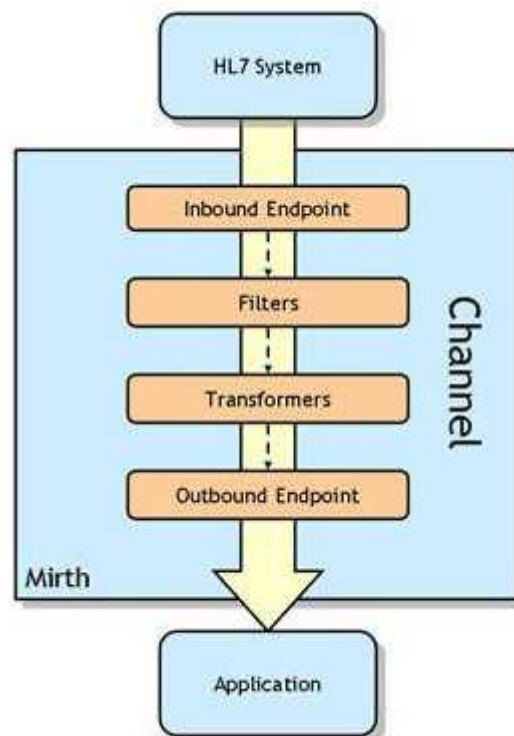
Existeixen moltes eines que poden oferir part dels serveis però intentaré comentar las raons que porten a que moltes integracions acabin utilitzen com a eina Mirth.

**Es obvi que la formació sobre la eina no es abast de aquest TFC, però si que crec important donar una pinzellada a la eina per poder conèixer-la.**

Disposa de una interfaz HL7 de plataformes de creuament i codi obert, permetin el enviament bidireccional entre sistemes i aplicacions sobre múltiples capes de transport, es pot crear connexions HL7 on transformar/adaptar els missatges entre dominis.

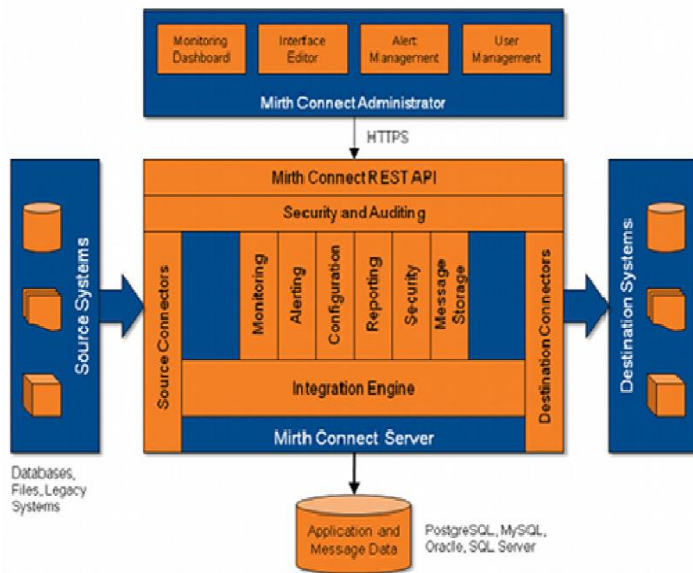
Utilitza canals per connectar els sistemes HL7 que estableixen diferents punts de adaptació.

Els nivells de adaptació segons la plana oficial son els següents.



**Figura 4.2.1** Arquitectura Mirth (documentació oficial).

#### 4.2.1-1 Estructura dels canals. I característiques.



En aquesta arquitectura basada en canals, es mostra de forma molt bàsica com es capaç de tractar o transformar la informació que arriba a la plataforma.

- Per una part es configura el tipus de entrada (Source Systems)
- Es configura un tipus destí (Destination System)
- Tot això o podem veure recolzat amb una eina de administració via web, ( Mirth Connect Administrator) amb una base de dades que es pot crear amb diferents motors, (Mirth Connect Server).

En definitiva es un estàndard de la interpretació de la comunicació entre diferents aplicacions del sistema sanitari / hospitalari.

Mirth	
HL7 v2.x	X
HL7 v3	X
DICOM (MWL, MPPS, Print, Query/Retrieve, Storage)	Q/R
XML	X
ODBC Natiu <sup>28</sup>	
JDBC Natiu <sup>28</sup>	X
Flat-files	X
EDIFACT	X
X12/HIPAA	X
Tipo de licencia	Open-source
Precio base( sin add-ons)	\$0
<b>Plataformas</b>	
Windows	X
Linux	X
Otras	Varios (Java VM)
<b>Arquitectura</b>	
Topologia	Hub-and-Spoke
Suporte SOA	X
Orquestación como servicio	
Transformaciones	XSLT, Java, TCL, Python, Javascript
Schemas de validación	
Content-based routing	
Comunicación bidireccional	X
Servicios de alertas	X (E-mail)
<b>Interface</b>	
Interface de configuración	Desktop
Interface de monitorización	Desktop
Mapper de mensajes	Scripting
Exportacion de mensajes	PDF/RTF
Vista Dashboard	X

Existeixen moltes altres aplicacions estandarditzades com Bitzalk, Iguana, Corepoint, etc., però no es objecte de aquest projecte ensenyar les diferències de les eines. En el esquema podem veure les característiques resumides d'aquesta eina que inspira aquest projecte.

#### 4.2.2 Perquè surt la idea de crear una comunitat ?.

El CCI-TCM treballa per facilitar la integració, la interoperativitat i l'ús d'estàndards en el Sector Sanitari en general i, especialment, en els Sistemes Públics de Salut.

Un dels projectes és l'entorn d'integració desenvolupat pel CCI-TCM. El principal objectiu d'aquest projecte és crear un entorn obert d'eines per facilitar la integració i la interoperabilitat de dispositius mèdics, serveis, i processos assistencials amb els sistemes d'informació sanitaris. L'entorn desenvolupat està basat en l'**EAI (Enterprise Application Integration)** Mirth Connect, el qual està dissenyat sota l'arquitectura SOA. Els treballs realitzats sobre l'entorn han estat dotar-lo d'eines de normalització, visualització, i post processat de fitxers de dispositius mèdics.

Una de les realitats que existeixen en aquest entorn és la manca de informació addicional per als programadors que utilitzen aquesta eina, a nivell Europeu no existeix cap comunitat que doni suport als desenvolupadors que la utilitzen.

Des de el CCI-TCM es va plantejar la possibilitat de crear un cercle de usuaris que treballen a l'àrea de la sanitat desenvolupant aplicacions amb aquesta eina.

#### 4.2.3 Quina és la motivació de cara l'usuari ?.

El cercle de usuaris que utilitzaran aquesta comunitat no és gaire gran, no podem pensar en un principi que serà una aplicació amb milions de usuaris, entre altres coses perquè el cercle de especialització que abarca el tema és alt.

Però per una altra banda sabem que no existeix cap tipus de know-how semblant a el que es vol desenvolupar, així doncs fa pensar que comunitats molt especialitzades no són un cercle gran de usuari però sí de intensitat d'ús, aquest és el principal objectiu la intensitat d'ús.

#### 4.2.4 Quina és la motivació de cara l'estudiant ?.

Quant es planteja un tema com aquest el fet que està dirigit per un públic molt concret ja és una motivació afegida, he de dir que els usuaris que utilitzaran aquesta eina són particularment desenvolupadors de software o entenedors de la programació bàsica per aquest motiu és un valor afegit que el usuari siguin capaços de entendre el funcionament de la eina sense necessitat de tenir sempre el cap la premissa de que sigui molt intuïtiva, les aplicacions que es desenvolupen per un usuari inexpert acostumen a ser molt simples perquè es confon la simplicitat amb la facilitat d'ús.

Una altre aspecte es el requisit no funcional que ha plantejat el client, el fet de aprendre un llenguatge nou sempre es una motivació extra, seré capaç de arribar-hi s'hi he d'aprendre tot això? , es una pregunta que correspon al aspirant contestar amb el seu esforç.

La eina utilitzada es una de les més espargides per tota la web a l'hora de crear blocs, convertir la funcionalitat de un bloc en una xarxa social o comunitat es un punt molt valorable no només aplicant alguns plugins que podem descarregar sinó condicionar-la i adaptar-la als requeriments del client es sense dubte el esforç i la motivació més gran.

## 5. Que es la eina Wordpress.

WordPress es un sistema de gestió de continguts, enfocat a la creació de blocs, esta desenvolupat en php y mysql, té una llicència GPL i codi obert modificable, te com a fundador a Matt Mullenweg.

És el CMS més popular les causes del seu creixement son entre altres coses la seva llicència, i la seva facilitat d'ús (nivell usuari), a més de la seva gran comunitat de desenvolupadors i dissenyadors, que s'encarreguen de desenvolupar-lo en general, creant plugins i temes per la comunitat, a Agost d'aquest mateix any (2011) el seu ratin d'us es del 14,7% de totes les planes webs que corren per Internet.

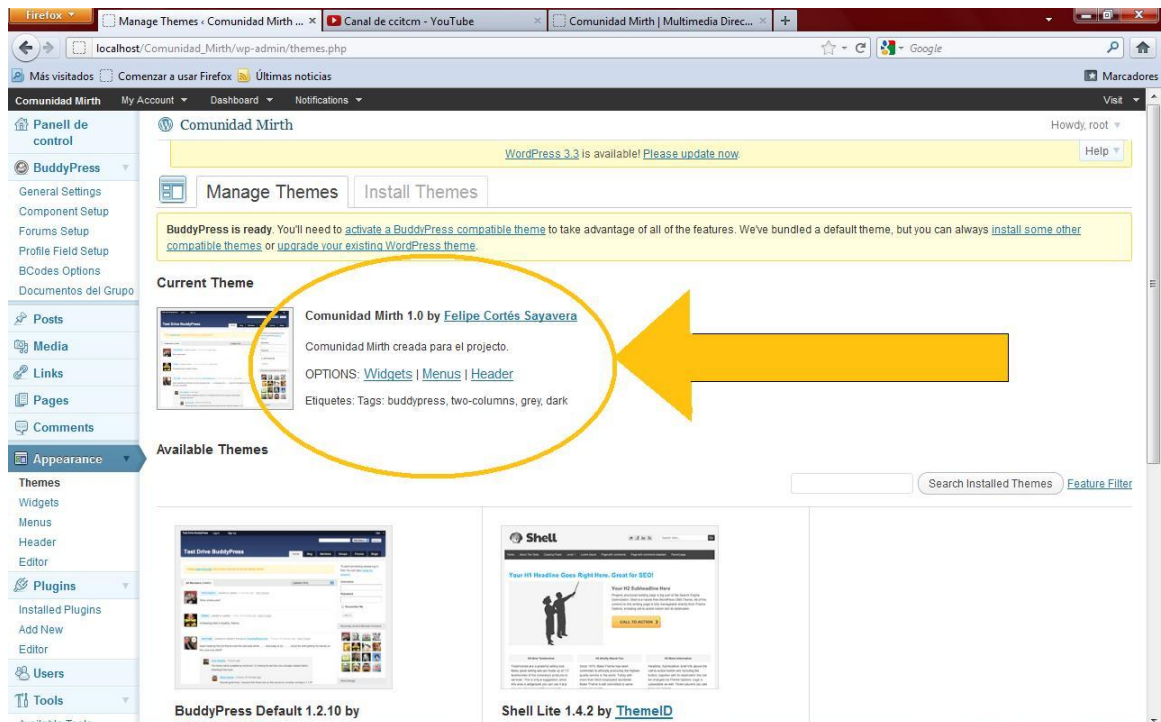
### 5.1 Aprofundim en Wordpress i els seus components

Realitzant un estudi de la seva arquitectura de aquest gestor de continguts i el plugin necessari Buddypress per la realització de la xarxa social / comunitat, es poden diferenciar tres tipus de components:

- a) **Core:** Es el motor essencial de la eina, es qui defineix i implementa les funcions bàsiques, com pot ser les entrades en blocs.
- b) **Plugins:** Son funcionalitats que s'afegeixen o modifiquen el comportament del core, en un principi sense cap modificació la incursió del plugin de buddypress incorpora funcionalitats socials.
- c) **Interfícies:** Es tracta de la part que els Enginyers anomenen com la capa de presentació, però que en aquest tipus de eines no son ni molt menys el que entenem com capa de presentació.

Es basen en plantilles o "templates" que defineixen com es mostrarà la informació per la pantalla, en aquest projecte es va definir un tema propi o plantilla, creat per l'aspirant, com a aspirant entenc la capa de presentació com una capa fàcilment modificable en un futur, no cal dir que per que això es compleixi no ha de tenir cap tipus de codificació de la eina, no es el cas en aquest tipus de aplicacions o cms ja que ens trobem molta codificació dins de aquest temes, a pesar que s'ha intentat estructura per una futura modificació, com es pot veure en el menú de administració de la mateixa eina, el core de wordpress reconeix el tema creat per l'aspirant.





**Figura 5.1 Model Administració tema propi.**

Aquesta tecnologia proporciona una base molt sòlida per el desenvolupament de la aplicació que volem fer, però no cal dir que no s'adapta ni molt menys a les necessitats del client i del projecte a desenvolupar, per aquest motiu s'ha de modificar i és aquí on tenim el problema afegit.

### 5.1.2 Base de dades inicial del sistema.

Les taules que ens genera la eina CMS més el plugin de buddypress son un total de 18, per poder portar a terme el projecte requerit per el client aquestes es van elevar fins a 46, com podem veure aquesta tecnologia ofereix una solució molt genèrica on si que es cert que adquirim una petita eina social per el nostre objectiu principal que es poder adaptar-la per adquirir noves funcionalitats.



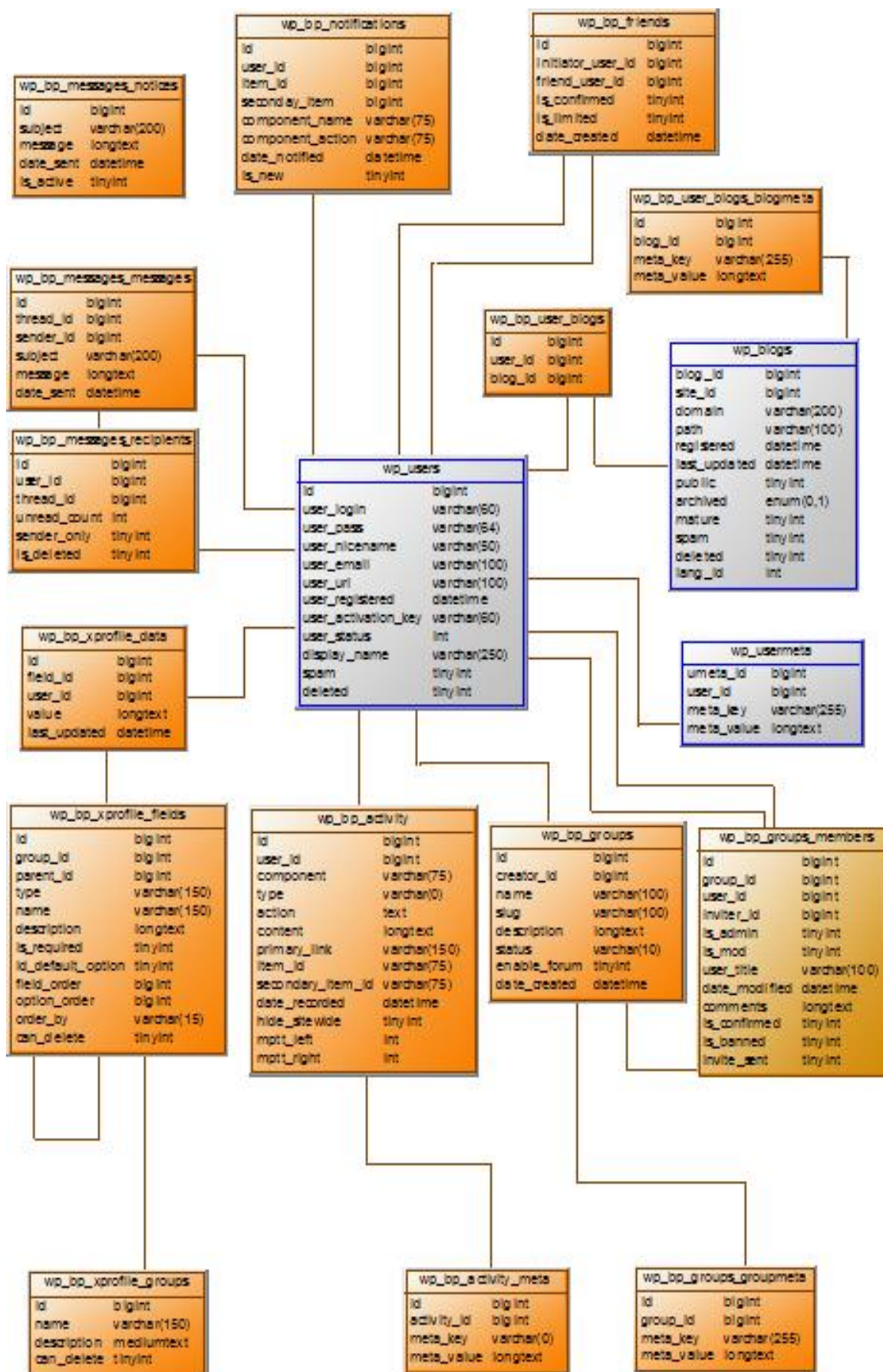
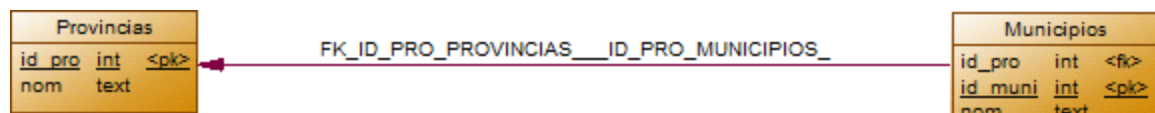


Figura 5.1.2 BBDD de la eina Wordpress i Buddypress

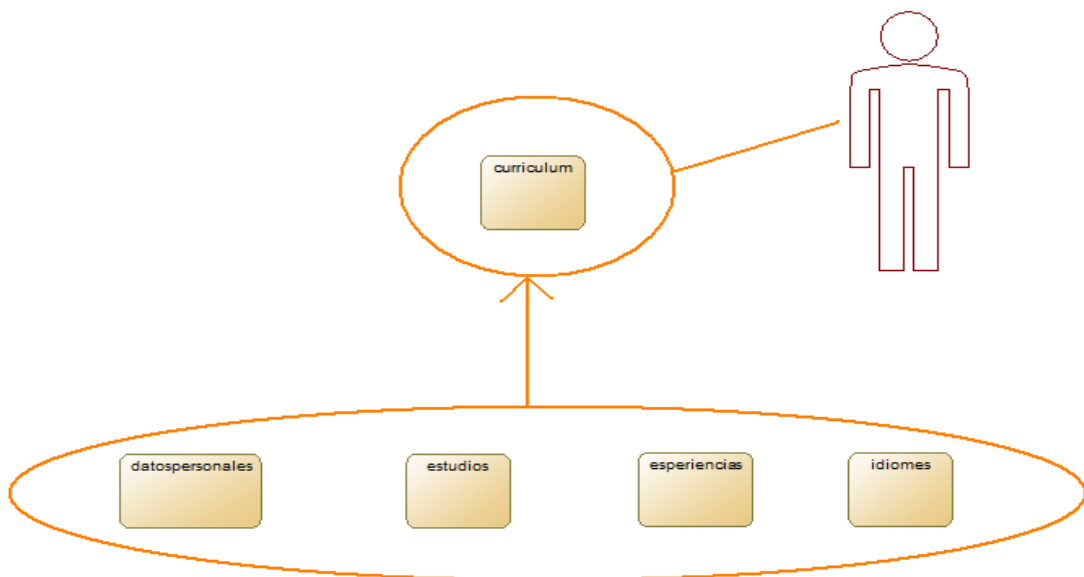
Aquesta base de dades a sigut adaptada per poder arribar a bastir les noves funcionalitats que es requerien, en aquest cas les taules les podem definir de dos tipus bàsics, **taules de magatzematge primari** i **taules de magatzematge de la aplicació**.

Aquestes taules que s'han afegit tenen funcions molts diferents explicarem las seves funcionalitats amb un exemple:

- **Taules de magatzematge primari:** Son aquelles taules que bàsicament serveixen per bastir de dades la eina, però que tenen poques modificacions o aquestes son nul·les, en un exemple podríem posar la taula de “províncies”, que no tenen cap tipus de modificació una vegada plena de dades o la seva modificació esta subjecte a una acció poc probable, però facilitarà en un futur al programador, normalment aquestes taules son modificades per el administrador de bbdd.



- **Taules de magatzematge secundari o de la aplicació:** Son aquelles que tenen un grau de modificació o inserció alta, podríem parlar com es el cas de la taula “datospersonales”, la seva modificació esta subjecte a l’usuari i es pressuposa una inserció i modificació alta.

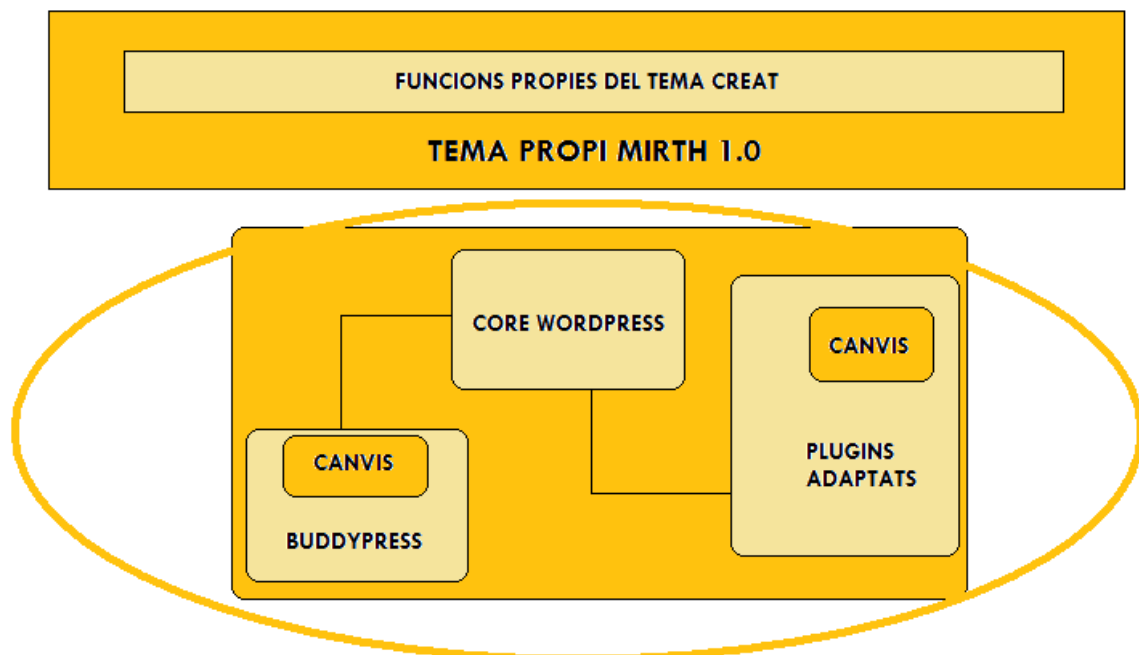


### 5.1.3 Sistema tecnològic i funcional de la eina.

Dins dels sistema tecnològic proporcionat per la eina s'han agut de fer diferents canvis, que son els següents:

- **Core de Buddypress:** S'ha agut de modificar aquest motor, entre altres coses podríem definir un inici fàcil i simple on veurem un petit menú (modificable en un futur) on ens ensenya el nostre espai amb els arxius que hem pujat i els links més importants.
- **Plugins adaptats:** S'han adaptats els plugins proporcionats per la comunitat Wordpress, tant per obtenir un aspecte adequat o el seu propi funcionament.
- **Funcions pròpies:** Les funcions pròpies son moltes, totes adaptades dins del tema, on podrem destacar el dipòsit d'arxius.
- **Interfície:** La interfície gràfica ha estat dissenyada pensant en els colors de la marca Mirth, "darkorange" i "darkblue", la modificació a estat a partir del tema bàsic del plugin Buddypress, però creant un tema nou.

Podem observar en el següent esquema els components que comparteixen la eina, per poder diferenciar els modificats s'ha canviat el color.



**Figura 5.1.3** Arquitectura Utilitzada

Observem amb cercle el motor propi de la eina.

### 5.1.4 Com es controlen els arxius.

Vaig decidir crear una sèrie de funcions per poder gestionar-ho, uns dels aspectes més comuns a l'hora de crear un magatzem es saber diferenciar els usuaris i els seus arxius per aquest motiu, es van fer dos tipus de carpetes, una anomenada Uploads, i un altre anomenada Uploads\_Grupos, dins de aquestes carpetes es gestiona el dipòsit, segons el següent.

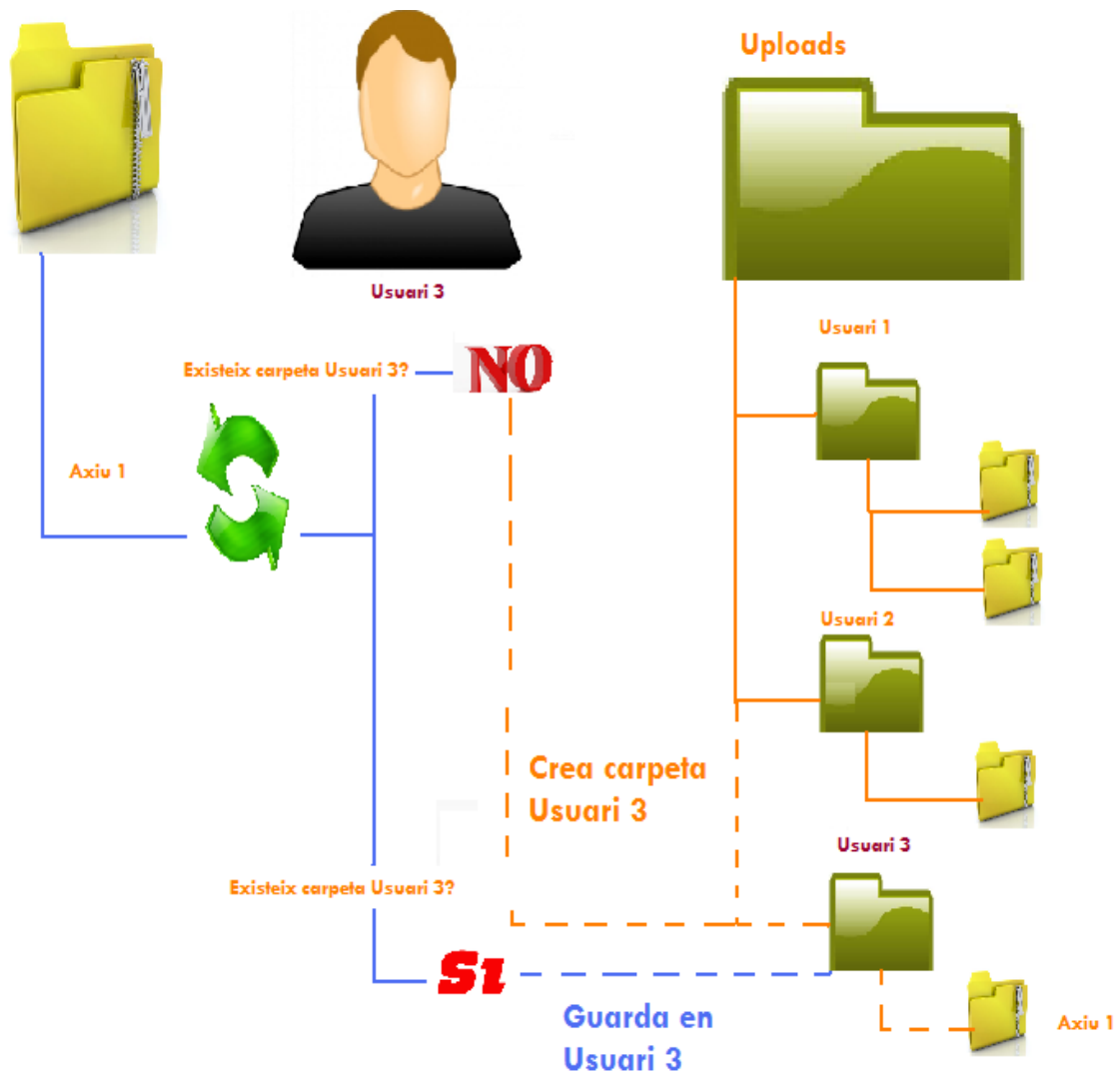
*La reflexió que vaig arribar com a aspirant va ser.....*

Sabem que els noms de usuari no es poden repetir ja que existeix un control a l'hora de donar-se d'alta, doncs bé si tenim que pujar arxius, i el servidor a de treballar per què hem de fer que la base de dades també treballi , ja que a de guardar el arxiu complet com a BLOB i convertit a binari.

Doncs bé una vegada pujat es controla simplement per arxius i rutes, o sigui si dins de la carpeta "Uploads", existeix una carpeta amb el nom del usuari doncs el guarda allà sinó simplement crea la carpeta amb el nom del usuari i després el guarda, en el cas que aquell usuari ja hagués pujat un arxiu amb el mateix nom i tipus, simplement el guardaria igual ja que s'afegeix un identificador al principi. Un altre tema es a l'hora de llegir els arxius del usuari, la classe creada només fa una recerca en el directori "Uploads" si existeix la carpeta amb aquell nom d'Usuari i té arxius només carrega a la interfície el nom del arxiu, es va habilitar un botó que referència a una altre classe per descarregar-ho.

El control de grups es exactament igual, la carpeta en aquest cas es la habilitada com a "Uploads\_Grupos", la diferencia es que tots els participants del grup poden descarregar els arxius compartits i poden pujar de nous.

Sobre el control de arxius no s'ha habilitat cap tipus de restricció que no sigui la de el pes, que es de 10 megabytes, però s'ha deixat definit un atribut type que deixa veure de quin tipus es (xml, pdf, doc, js....), només s'ha de crear una nova condició de entrada , observem el següent esquema reverenciat a la manera de fer de el reposi tori del usuari.



**Figura 5.1.4 Model de Entrada de arxius**

Si en qualsevol moment el usuari es quedés sense arxius dins de la seva carpeta no vaig crear convenient borra la carpeta buida, no es va definir per que dins del meu punt de vista no era un requisit important o es previsible que es defineixi més endavant, es va tenir el mateix criteri per els arxius compartits de grup, per que tot això funcioni no cal dir que es va crear una funció que retira els espais en blanc i els caràcters “\_” ja que el renom del arxiu el va amb identificador seguit d’aquest caràcter “\_” la possibilitat de tenir algun que contingui el caràcter “\_” podria esdevenir equivocacions.

### 5.1.5 Sistema de puntuació.

El sistema preguntes i respostes dins dels fòrums tenen un objectiu clar que es orientar al usuari en els dubtes plantejats, però per altre banda sabem que no sempre es pot fer cas de les respostes que podem veure en fòrums, per aquest motiu s'ha definit un sistema de puntuació.

Dins de cada comentari es mostrarà la puntuació del autor, d'aquesta manera podrem veure la mitjana que té la persona que ens ha contestat.

¿Com es calcula aquesta mitjana?, s'han creat diferents atributs dins de la taula wp-users, definida per la eina wordpress, per aquest objectiu en qüestió s'ha creat els camps ("puntuacion", "PreguntasPuntuadas" i "MediaTotal"), en els quals alberguen, en el primer es la suma de totes les puntuacions al usuari, el segon son el total de les respostes que han estat puntuades i el tercer es la divisió entre el primer i el segon, he cregut convenient fer una puntuació rodona per aquest motiu sempre es fa un arrodoniment, per exemple el resultat de la puntuació es 6,3 la puntuació del usuari serà un 6, si es un 6,5 la puntuació serà un 7.

Camp	Tipus	Nul	Predeterminat	Extra
<b>ID</b>	Bignit(20)	NO		Auto_Increment
<b>User_login</b>	Varchar(60)	NO		
<b>User_Pass</b>	Varchar(64)	NO		
<b>User_Nicename</b>	Varchar(50)	NO		
<b>User_Email</b>	Varchar(100)	NO		
<b>User_URL</b>	Varchar(100)	NO		
<b>User_Registered</b>	Datetime	NO	0000-00-00 00:00:00	
<b>User_Activation_Key</b>	Varchar(60)	NO		
<b>User_Status</b>	Int(11)	NO	0	
<b>Display_name</b>	Varchar(250)	NO		
<b>Nivel</b>	Int(11)	SI	NULL	
<b>Puntuacio</b>	Int(11)	NO	0	
<b>PreguntasPuntuadas</b>	Bigint(20)	NO	0	
<b>MediaTotal</b>	Double	NO	0	

**Figura 5.1.5 Taula wp-users Wordpress**

### **5.1.6 Els fluxos d'arxius entre usuaris**

La aplicació dona la possibilitat de crear i cercar usuaris, aquells que son contactes nostres podrem gaudir dels arxius que ells tenen en la seu espai, això el que intenta es que el sistema de compartir arxius sigui àgil ja que el atractiu de ser amic o tenir un contacte entre altres coses es oferir els arxius que tenen pujats.

No només tenim els arxius pujats per els nostres contactes, més avançat el projecte es va valorar la possibilitat de adjuntar arxius a comentaris, per fer més concreta la resposta amb un manual adjunt o un document explicatiu.

## 6. Les parts de l'aplicació.

Per poder establir una explicació i així sigui molt més fàcil la avaluació crec necessari desmantellar la aplicació per poder fer una explicació raonable dels seus menús, i enllaços, obviarem la part de l'alta de usuari ja que no té gaire diferenciable de les aplicacions que ens trobarem en la xarxa, en aquest document no s'intenta exposar el codi sinó explicar que fa l'aplicació per aquest motiu obviarem molt codi de implementació..

### 6.1 La explicació dels menús.

Dins de aquest apartat s'explicarà la funcionalitat de cada menú així com la inclusió de alguns dels codis més significatius a l'hora d implementar, abans de començar s'ha de aclarir que algunes variables estan definides en el arxiu functions.php que carrega wordpress al inici de la aplicació algunes com `BASE_PATH` que son cridades i alberguen la ruta de la base dels arxius, es necessari per una futura modificació no s'hagi de revisar tots els arxius en recerca de rutes absolutes.

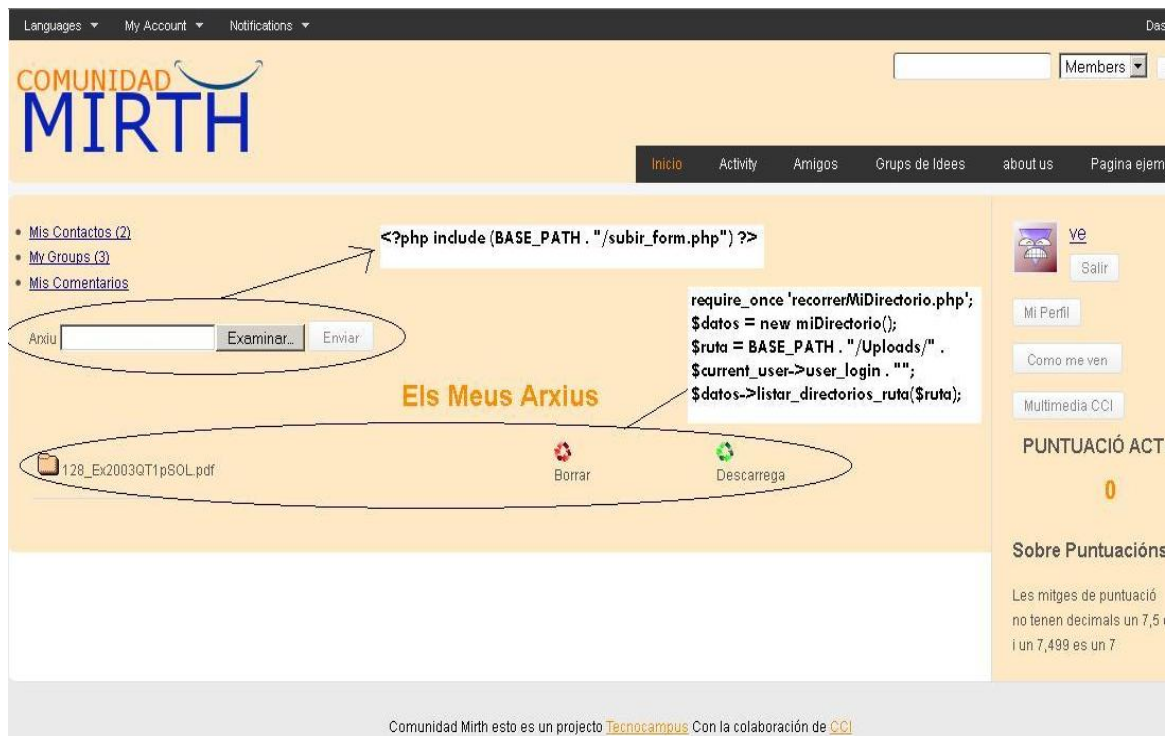
#### 6.1.1 El menú d'Inici.

Una vegada que hem entrat amb el nostre usuari i contrasenya el primer que veurem serà el menú d'inici, aquest menú es caracteritza perquè es el nostre espai de memòria que albergarà els arxius que compartirem amb els nostres contactes, des de aquí podrem pujar els arxius al nostre espai i els veurem sempre que entrem en el sistema, per a cada arxiu s'habilitarà la possibilitat de esborrar-ho o fer una descarrega del arxiu, en la part superior s'aniran creant enllaços als nostres comentaris, grups, contactes etc...

Dins d'aquest apartat trobarem el formulari per poder pujar arxius, aquest formulari es cridarà fent una inclusió de un arxiu php, també podrem veure tots els arxius que hem pujat al nostre espai de directori, això es farà a través de una classe implementada per l'estudiant.

Si observem la següent imatge de la pantalla de inici, en la part de pujar el arxiu he de dir que només es una inclusió de un formulari creat per l'estudiant, per suposat serà aquest arxiu qui farà el inclusió necessària de la classe que s'encarregarà de crear el directori si no existeix o guardar el arxiu en aquest en cas contrari.





**Figura 6.1.1** Pantalla Inici

Si observem el codi del formulari



Podrem veure el que he denominat com la orientació a objectes i els seus mètodes en php

Si mirem lo primer que fa es dir al sistema que per poder executar aquest codi es necessari el arxiu subir.php, immediatament després es crea una variable de objecte anomenada \$datos. El més interessant de tot es que en aquest formulari no es crida a cap arxiu php sinó que es crida a un mètode del objecte \$datos, mètode anomenat subir().

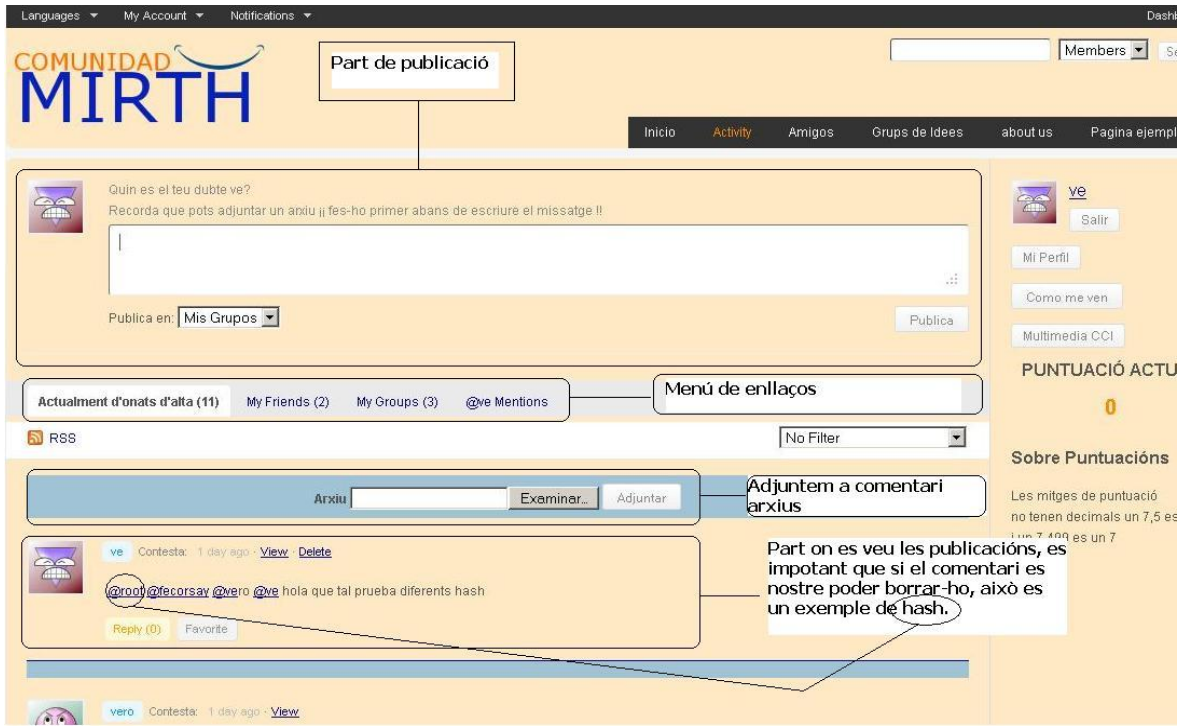
```

<?php
require(BASE_PATH."/subir.php");
$datos= new subirArchivo();
?>
  <div id="formulario">
    <form action="<?php $datos->subir() ?>"
method="post" enctype="multipart/form-data" name="form">
      <label for="archivo">Arxiu</label>
      <input name="archivo" type="file" id="archivo" />
      <input name="boton" type="submit" id="boton"
value="Enviar" />
    </form>
  </div>
<?php if($datos->page_path!=""){ ?>
<script type="text/javascript">
  alert ("L'arxiu a de ser inferior a 10 megas");
  location.href="<?php echo $datos->page_path ?>";
</script>
<?php } ?>

```

**Figura 6.1.1-1** Codi PHP  
formulari pujar arxiu

## 6.1.2 El menú d'Activitat.



**Figura 6.1.2 Pantalla Activitat**

Es aquí on podrem publicar les nostres preguntes i obtenir les diferents respostes dels usuaris, trobem en un principi la area on publiquem les nostres preguntes, amb un petit menú inferior que ens donarà la opció de veure els amics, grups, i mencions al nostre

usuari, dins de aquesta part trobarem la possibilitat de puntuar les respostes que ens han facilitat, però no podrem repetir la puntuació a una mateixa resposta ja que la varem puntuar abans, també es dona la facilitat al usuari de esborrar la seva pregunta.

He de dir que les mencions al nostre usuari es una influencia directa de twitter, en que simplement posant el símbol arroba seguit del nom del usuari farem un enllaç directe aquest comentari des de el nostre propi menú.

En una última instancia el client va demanar que si fos possible s'afegeixin arxius no només per grups o per usuaris sinó que també per comentaris.

- Apartat Amics: Aquest apartat es un dels més interessants, veurem totes les converses dels amics que tenim en la nostre llista de contactes, també veurem els arxius que han pujat en les converses per poder descarregar.
- Apartat Grups: Es molt similar a l'apartat anterior però entre els grups que nosaltres estem inscrits o treballem en ells.
- Els meus favorits: Son totes aquelles converses que hem marcat com a favorites, d'aquesta manera tindrem un enllaç directe a totes aquelles converses que en resolt alguna vegada els nostres dubtes.
- @usuari Mencions: Son aquelles mencions que han posat els usuaris amb el nostre nom, els famosos enllaços ràpids.

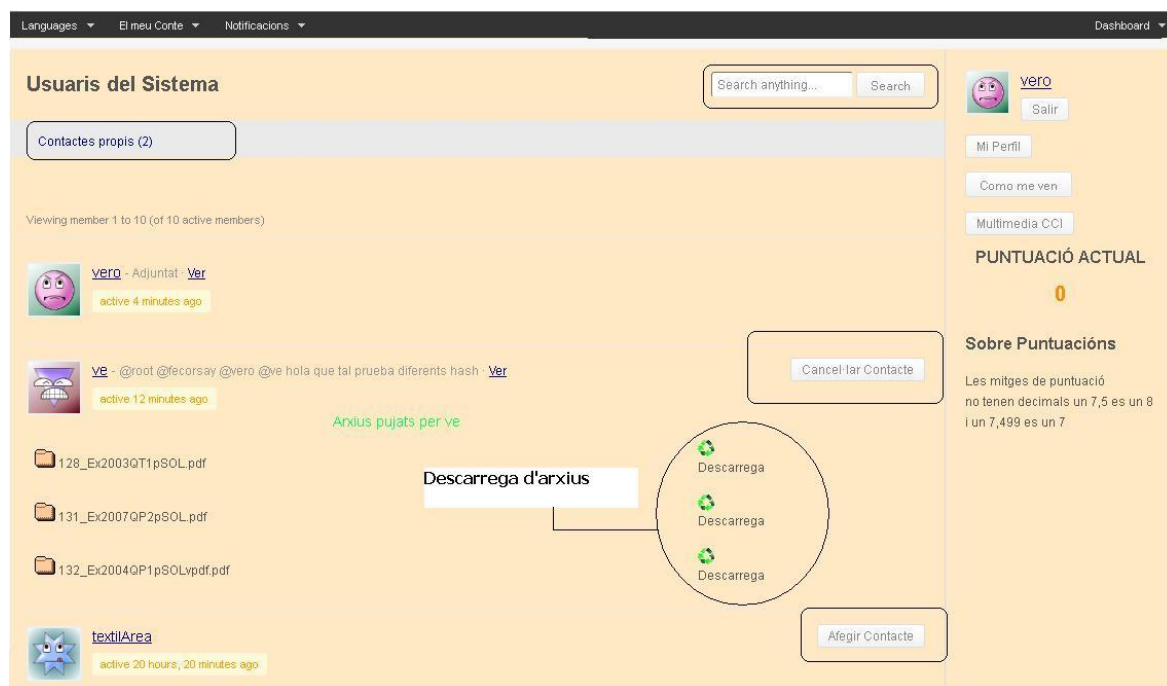
Quant un usuari respon a una pregunta generada per un altre usuari el primer podrà veure la resposta amb la puntuació de la persona que ha respòs, també es genera una taula del 1 al 10 per poder puntuar amb un sol clic, podrà tornar a replicar o fins i tot marcar-la com a favorita per tenir-la guardada dins de la llista de favorites del usuari.



### 6.1.3 El menú Contactes.

Dins de aquest apartat podrem observar els diferents usuaris que estan donats d'alta al sistema, podrem fer una recerca per nom, l'atractiu es que podrem enviar sol·licituds d'amistat a tot aquell que no sigui contacte nostre, que aquest veure en l'apartat de notificacions que explicarem més endavant, s'ha habilitat un sistema per poder veure i descarregar els arxius dels nostres contactes a més de poder cancel·lar la nostre amistat.

Des de el botó “Afegir Contacte” enviarem una notificació al usuari que veure en el seu espai de notificacions que s'ha sol·licitat una amistat, aquest usuari podrà refusar la invitació o acceptar-la.



**Figura 6.1.3** Pantalla Contactes

### 6.1.4 Barra lateral.



Dins d'aquesta barra cal una menció especial a l'apartat de Multimedia CCI, es va crear un apartat propi per demanda del client, era un requisit poder fer visualitzacions de vídeos explicatius, i descàrregues de arxius propis diferenciant els arxius de grup o de comentaris.

S'ha partit de la base que el client ja té un canal fet al youtube i que és aquest el que normalment s'està mantenint, apart d'això també es important estalviar espai en el servidor per aquest motiu s'ha creat un enllaç al canal de CCI, visualitzant els vídeos que alberga aquest canal, en la nostre aplicació.

El meu perfil actua cridant a diferents formularis que ens facilitaran la feina a l'hora de implementar el nostre currículum, una vegada fet e implementat

podrem observar el resultat cridant la opció de "Com hem veuen les empreses", que executarà un mètode per obrir les dades en format pdf.

Aquesta barra ens visualitzarà i ens acompanyarà en totes les navegacions possibles, ja que ens informarà de la nostre puntuació segons les rebudes per els usuaris de les nostres respostes.

### 6.1.5 Barra Superior.

La barra superior ens facilitarà filtrar tota la aplicació, on tenim diferents enllaços a tot el motor de Buddypress i amb les modificacions requerides, així podrem observar que contempla la possibilitat de canviar de llenguatge (preparat per futures versions), El nostre conte, o les notificacions.

- **Llenguatges:** s'ha volgut deixar preparat el plugin de llenguatges per les futures modificacions que poden esdevenir, en un principi s'ha creat els espais de català, castellà i anglès.
- **El meu Conte:** és aquí on està la major part de la aplicació:

- **Activitat:** Son tots aquells comentaris o activitats que s'han generat entre els quals es pot filtrar per:
  - Personal, els que l'usuari a fet.
  - Friends, els que els seus contactes han realitzat
  - Groups, aquells comentaris nous que s'han fet els grups que pertany
  - Favorites, aquelles converses que s'han seleccionat com a favorites
  - Hash, son tots aquelles mencions que s'han fet del usuari en la aplicació.
- **Profile:** Possible millora en posteriors versions, crec que es important deixar preparat el perfil públic de l'usuari, i la seva posterior edició, en canvi es important deixar la possibilitat de canviar la imatge del avatar a l'usuari.
- **Messages:** son els missatges privats entre usuaris, en aquest menu trobarem la safata d'entrada, la safata de sortida i compondre un missatge.
- **Friends:** Aquí trobarem una recerca dels nostres amics, com les sol·licituds que hem rebut d'amistat que estiguin pendents.
- **My Groups:** Son tots aquells grups dels quals som part, s'ha quedat per implementar l'apartat de invitacions per visualitzar totes les invitacions de grups que tenim.
- **Album:** Dins de totes les funcionalitats possibles s'ha creat la possibilitat de generar àlbums de imatges dels usuaris, per poder compartir imatges, els filtres possibles son, publiques, tots els usuaris poden veure-les, membres registrats, solament amics o simplement privades.
- **Settings:** S'ha dividit en tres parts, la primera simplement ens deixarà canviar el password o el mail de notificacions, per possibles millores seria important afegir el formulari que crea el currículum en aquest apartat, un apartat de notificacions per configurar les notificacions que ens arriben o esborrar el conte.
- **Log Out:** Sortir de la aplicació.
- **Notificacions:** Si tenim una notificació pendent serà aquest petit apartat el que ens farà un avís previ per estar sempre actualitzar, de les nostres invitacions a grups, missatges privats etc...

## 6.2 La explicació dels directoris i filosofia d'objecte en php.

Els directoris que s'han creat per part de l'estudiant per portar a terme una bona organització son varis que intentarem desgranar en aquest apartat, així com alguns dels arxius més importants.

### 6.2.1 Dins del propi tema Comunitat Mirth.

- **Documents:** Es aquells directori que conté els arxius pujats per CCI creant d'aquesta manera un espai apart per els arxius que es podran descarregar del client, crec convenient comentar que una vegada pujat el arxiu no caldrà fer res per la futura descarrega ja que s'ha creat un sistema automàtic de creació de botons.

```

$directorio = dirname(__FILE__);
$path = str_replace("multimedia", "Documentos/", $directorio);
$directorio = dir($path);
require("download.php");
$datos = new Documentos();
while ($archivo = $directorio->read()) {
    if ($datos->cargado != "si" && $archivo != "." && $archivo != "..") {
        $archivo = utf8_encode($archivo);
        ?>
        <tr>
            </br>
            <form id="documento" name="documento" method="post" action="<?php $datos-
>verDocumento() ??">
                <input type="hidden" id="id" name="id" value="<?php echo $archivo ??" />
                <input type="hidden" id="ver" name="ver" value="si" />
                <input type="submit" value="<?php echo $archivo ??" />

            </form>
            </br>
        </td>
    </tr>

    <?php
}
}
$directorio->close();
?>

```

- **Índex:** Es en aquest directori on guardarem tres arxius php diferenciadors, el que es carregarà si l'usuari no està donat d'alta, el que es carrega si que està donat d'alta, i un últim arxiu que genera la recerca de tots els arxius que l'usuari a pujat a l'aplicació.
- **PDF:** Documents de la biblioteca necessària que ens facilitarà la feina de crear arxius amb format pdf al vol.



- **Perfil:** Es el directori on guardarà els formularis necessaris per la creació de un perfil d'usuari així com el seu currículum.
- **Uploads:** Es el directori base d'arxius pujats al servidor, es crea subdirectoris amb el nom del usuari on serà aquest els que albergaran els diferents arxius pujats per l'usuari.
- **Uploads Grups:** Fa la mateixa funció que el directori Uploads però en comptes de ser usuaris son grups de usuaris, els subdirectoris es crea a partir del nom del grup.
- **Uploads Comentaris:** Arxius que seran pujats per cada comentari, es fa un canvi de nom i es guarda posant el identificador del comentari al principi del arxiu.
- **Altes:** Programes, funcions i diferents arxius encarregats de l'alta correcte del usuari.
- **Multimedia:** Espai encarregat de portar el canal de CCI a la nostre aplicació així com facilitar el formulari de la descarrega dels arxius.

Aquets son els directoris més significatius de la aplicació, alguns d'ells contenen fitxers, que a la seva vegada contenen classes necessàries per la implementació de la aplicació, com per exemple l'últim comentat "Multimedia", conté l'arxiu download.php que té una classe que simplement conté un única funció, es important saber que es creen classes perquè d'aquesta manera facilitarà les futures ampliacions només ampliant les classes amb mètodes, l'única funció que té esta creada per visualitzar el document seleccionat per el usuari.

### 6.2.2 Filosofia d'objecte en php

La creació de pàgines web dinàmiques son dividides per funcionalitats això sembla una mica estrany però queda més clar si diem que la creació de les pagines les pensem com l'alta, l'inici, la pagina de amics, etc...

Amb això vull explicar que si valorem la pàgina de inici de un usuari i tenim diferents perfils de usuari podem pensar que la diferencia entre un usuari (normal) i un usuari (empresa) segurament sigui mínima, la crida de un menú o de un boto que no apareix en el usuari normal en canvi si apareix en el usuari empresa..., totes aquestes modificacions es poden diferenciar si mitjancem classes.

La filosofia no es crear codi molt llarg, que de vegades no entenem perquè la complexitat es molt gran, entre altres coses gracies a la quantitat de variables que necessitem per

generar la funcionalitat que volem, a part de que en la majoria de aplicacions fetes en aquest llenguatge trobem que la variable “action” del formulari sempre crida a un arxiu php creixen de manera exponencial la quantitat de arxius que emmagatzemarem en el nostre servidor i fomentant la poca mantenibilitat.

En aquesta aplicació he volgut fer que això quedi de manera una mica més clara i que la complexitat estigui en el mètode que cridem, fent que la crida sigui clara i senzilla com exemple mirem el codi del índex quant l’usuari està donat d’alta.

```
require_once 'recorrerMiDirectorio.php';
$datos=new miDirectorio();
$ruta=BASE_PATH . "/Uploads/" . $current_user->user_login."";
$datos->listar_directorios_ruta($ruta);
```

Si observem aquest codi la complexitat no la veiem en cap moment simplement avisem que necessitem una classe i instanciem un nou objecte, cridant a un mètode i passant la variable ruta, serà aquest objecte el que crearà tot el demès perquè la aplicació funcioni, per suposat crea els formularis de borra i descarregar l’arxiu.

Imaginem que una pàgina conté diferents formularis que tots es dediquen o tenen un objectiu comú que podríem denominar com a exemple donar d’alta a un usuari, doncs bé si creem un objecte de la classe alta només hauríem de cridar al mètode en qüestió per cada formulari, això ens resoldria que només amb un sol arxiu tindríem tota la alta de usuaris feta.

Que hem aconseguit ?, doncs bé graciés a la instancia del objecte només hem de cridar aquest mètode des de fora sense preguntar-nos com ho fa, simplement ho fa, aquesta es la filosofia que s’ha seguit.

## 6.3 Perfils d’usuari i la seva activitat.

En aquest apartat s’explicarà els dos perfils d’usuaris que van quedar, ja que en un principi es va valorar la possibilitat de crear tres, també s’explicarà alguns dels casos d’ús més importants que marca l’estudiant.

### 6.3.1 Perfils d’usuaris.

En aquesta aplicació en un principi es va valorar la possibilitat de tres usuaris diferents, el programador, l’usuari principiant i l’empresa, després de diferents reunions amb el client, es va valorar la possibilitat de eliminar el programador.

Aquesta eliminació va estar realitzada ja que per manca de temps no es va crear un sistema de oposicions, en un principi el usuari principiant podia pujar a programador després de una sèrie de puntuacions, la avantatge principal era que els programadors podrien realitzar tasques que no es van acabar de definir, per aquest motiu no té sentit si a l'hora de la presentació tots dos usuaris poden realitzar els mateixos cassos d'us sense diferències.

De totes maneres l'estudiant ha deixat definit aquest aspecte per si el client vol incloure-ho més endavant, amb els seus fitxers i objectes, una de les possibles millores podria ser que el programador pot incloure codi amb un editor de la aplicació.

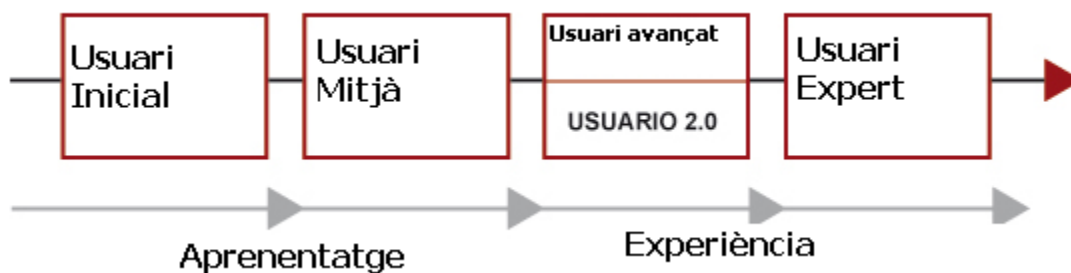
Els usuaris que s'han mantingut son l'empresa i l'usuari referent a la persona física, ja que la diferència es que les empreses poden incloure ofertes de treball i l'usuari normal no, a més de que un usuari s'ha de poder inscriure en una oferta.

Per poder diferenciar usuari dins de la nostra aplicació s'ha creat un camp a la taula wp-user anomenat "nivel", aquest atribut conté una numeració que ens indicarà si es tracta de una usuari o de una empresa, per diferenciar-los només hem de cridar diferents índex per cada un d'ells.

```

        If ($usuario[0]->nivel == 3){
            Include (BASE_PATH."indexEmpresa.php");
        }
        Else if $usuario[0]->nivel != 3){
            Include (BASE_PATH ."indexUsuario.php");
        }
    
```

Podríem definir en una possible millora una evolució d'aquests usuaris segons la seva estructura de coneixement i segons la seva experiència en diferents contextos d'ús, així doncs en posteriors versions es podrien tenir en conte la estructura següent:

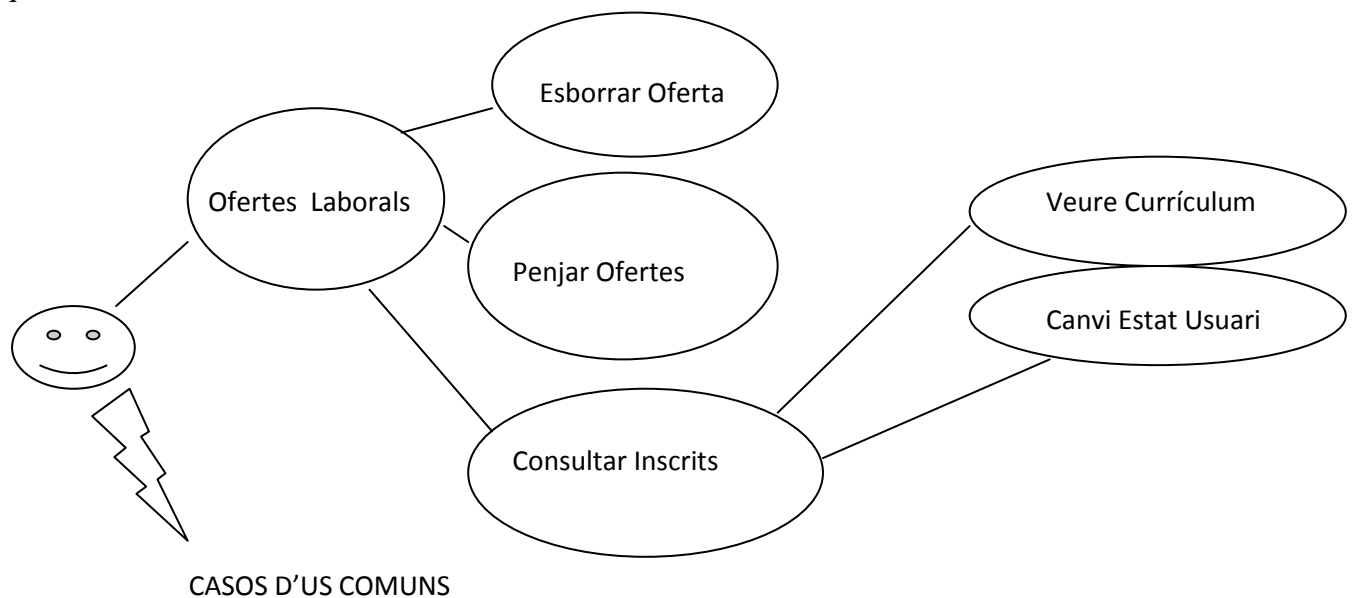


**Figura 6.3.1** Possible millora creixement del usuari

### 6.3.2 Activitats per usuaris diferenciadores.

#### **Empresa:**

Per diferenciar la activitat entre empresa i usuari s'ha creat algunes diferències dintre del ventall del programari realitzar, les tasques comuns com activitats, creació de grups, missatges entre usuaris i tot el programari necessari per poder portar a terme la comunitat, que per un altre costat es comú per tots dos usuaris es donarà una petita pinzellada més endavant, es obvi que la importància queda reflectida en aquells casos d'ús diferenciables, que estan relacionat amb les ofertes laborals i els cursets i seminaris.



- **Ofertes Laborals:** Aquest grup es divideix en diferents subgrups, significativament es comentarà el cas d'ús consulta dels currículums dels usuaris Inscrits que conté potser una de les característiques més interessants.

Els currículums de usuaris no es guarden dins del servidor, d'aquesta manera s'intentarà no acumular documents innecessaris.

Quant la empresa vol veure un currículum inscrit en la seva oferta el que es genera es una consulta a la bbdd creant el pdf al vol i generant el codi necessari per la descarrega d'aquest, això a estat possible gràcies al aprenentatge de la biblioteca "fpdf" que es pot descarregar gratuïtament aquí ( <http://www.fpdf.org/> ).

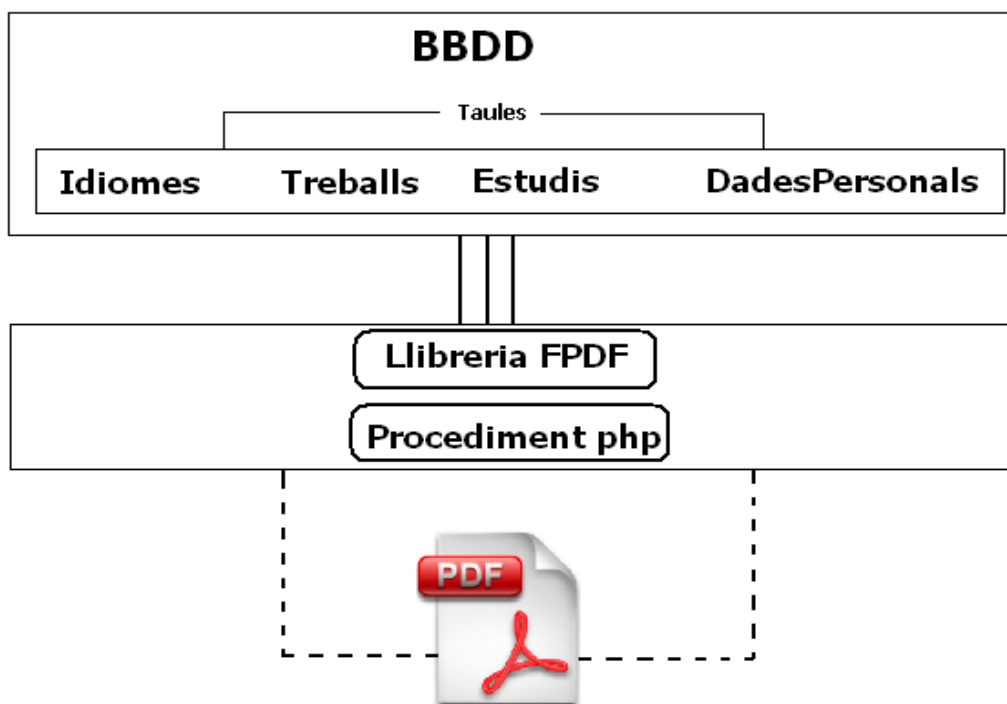
FPDF és una classe escrita en PHP que permet generar documents PDF directament des de PHP, la F de FPDF significa Free (gratis i lliure), es pot utilitzar per a qualsevol propòsit i modificar al nostre gust per satisfer les necessitats plantejades. FPDF té altres avantatges: funcions d'alt nivell. Aquesta és una llista de les seves principals característiques:

- Elecció de la unitat de mesura, format de pàgina i marges
- Gestió de capçaleres i peus de pàgina
- Salt de pàgina automàtic
- Salt de línia i justificació del text automàtics
- Admissió d'imatges (JPEG, PNG i GIF)
- Colors
- Enllaços
- Admissió de fonts TrueType, Type1 i codificació
- Compressió de pàgina

FPDF no necessita de cap extensió per a PHP (excepte zlib per activar la compressió i GD per a suport a GIF) i funciona amb PHP4 ( $\geq 4.3.10$ ) i PHP5.

Existeix una gran varietat de documentació completa des de la mateixa pàgina web a l'àrea de descàrrega.

Un exemple gràfic de la utilització o funcionalitat de aquest cas d'ús utilitzant la llibreria seria el següent:



**Figura 6.3.2** Creació de un pdf

**Usuari:**

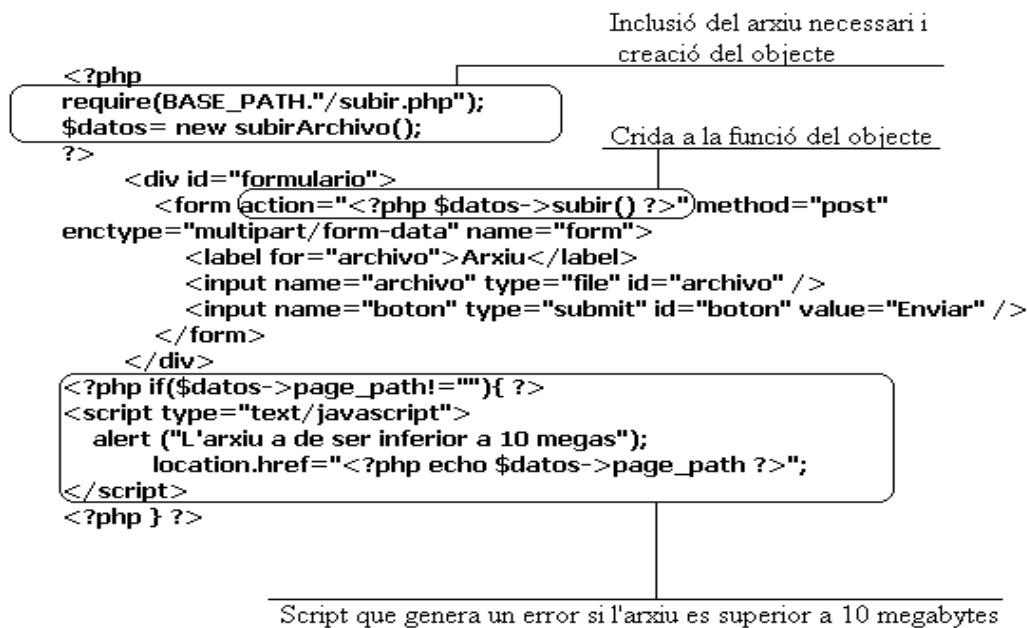
Els usuaris poden inscriure's en les ofertes generades per les empreses, s'ha habilitat un apartat dins del inici principal on es poden visualitzar totes les empreses que actualment estan donades d'alta en el sistema, dins de cada empresa estaran totes les ofertes laborals que l'empresa té pujades al servidor.

La consulta de la situació de la seva candidatura es essencial per aquest motiu l'usuari podrà veure en quin moment del procés està, per què un usuari estigui millor considerat de cara a les empreses ha de tenir el currículum complet, per que això sigui un valor afegit ha de realitzar la implementació que es troba en l'apartat de perfil que inclou diferents formularis per facilitar la visualització del currículum, amb la línia de successió següent, **Dades Personals, Estudis, Experiència e Idiomes.**

S'ha permès que el usuari pugui visualitzar el seu perfil en format currículum, d'aquesta manera podrà saber de quina manera el veuen les empreses, abans de inscriure's en qualsevol candidatura, seguint el mateix exemple de creació esposat en el punt anterior.

**6.3.3 Activitats comuns.**

**El dipòsit d'arxius de usuari:** Es el nostre espai dins de la aplicació, ens facilitarà la salvaguarda de documents dins del servidor, la programació es realment senzilla, s'ha creat un arxiu php que proporcionarà el formulari sense haver de picar el codi dins de la pàgina de inici, aquest es anomenat "subir-form.php", i conté la següent codificació:



Com podem observar en la codificació anterior el formulari té com a acció una funció del objecte \$datos, en el trasbalsament de arxius no s'ha contemplat la possibilitat de diferenciar els tipus de arxiu però si que s'ha habilitat la variable per una futura millora.

Per poder diferenciar els arxius per futures descarregues o simplement per esborrar el arxiu adequat quan l'usuari ho demana, necessitem la ajuda de la bbdd, amb un atribut auto increment que s'emmagatzemarà en una taula i després es canvia el nom de l'arxiu pujat amb aquest atribut.

Així doncs si un fitxer pujat al servidor amb nom xxxxx.xxx, quedarà finalment com a ide\_xxxxx.xxx, on ide serà el identificador del arxiu, per suposat no seria possible si no tenim un petit procés que evitarà errors de nomenclatura en els arxius.

```

private function quitarEspacios($arxiu){
    $this->nombre_archivo="";
    for ($i = 0; $i < strlen($arxiu); $i++) {
        if ($arxiu[$i] != ' ' && $arxiu[$i] != "_") {
            $this->nombre_archivo.=$arxiu[$i];
        }
    }
}

```

Com podem observar es una funció privada de classe que només treu els espais i els símbols “\_” al nom del arxiu, d'aquesta manera ens assegurem que quant fem un recorregut no tindrem problemes a l'hora de trobar el nostre arxiu.

**El dipòsit d'arxius de grup:** La creació de un espai per guardar arxius de grup, era primordial a l'hora de realitzar intercanvi de coneixement, però a diferencia del anterior, no era un dipòsit individualitzat sinó que era compartit per diferents usuaris, la principal

diferencia es que tots els usuaris que estan inscrits en aquest grup poden pujar arxius, amb la mateixa filosofia del punt anterior sabrem que arxiu es de quin usuari, guardan tots els arxius en contes de la carpeta del servidor “Uploads/NomUsuari”, guardarla en “Uploads\_Grupos/NomDelGrup”.

Saber diferenciar els propietaris dels arxius ens facilitarà les futures modificacions com també en aqueta versió poder esborrar el arxiu de propietat i no un arxiu aliè.

**El recorregut:** Per visualitzar els noms de fitxers que tenim en els directoris tant en el de grup com en el índex es va crear diferents funcions, el espai propi es recorregut amb una classe anomenada “miDirectorio” que està situada dins de la carpeta Índex, aquesta classe conté una funció “**function listar\_directorios\_ruta(\$ruta)**”, que ens retornarà la implementació dels formularis amb els noms dels arxiu, el boto per esborrar o el de descarregar, d’aquesta manera només cridant a una funció es aquesta la que crea els formularis necessaris per cada un dels arxius.

Els canvis referents als arxius de grups son importants, per començar s’ha de transmetre a la capçalera un altre atribut com es el nom del grup, **function listar\_directorios\_ruta (\$ruta,\$nomGrup)**, però la complexitat no acaba aquí sinó que com he comentat en el paràgraf anterior, el mètode crea els formularis necessaris amb els botons necessaris per descarregar i esborrar l’arxiu, però en aquest cas hem de tenir en conte el propietari del arxiu, per què sinó ho es, doncs ens evitarem el formulari de esborrar.

```

$this->resultado = $wpdb->get_results("select * from arxius_autor where idArxiu=" . $this->arxiuActual . " ");
if ($this->resultado[0]->idAutor == $this->current_user->ID) {
    ?>
    <form id="eliminar" name="eliminar" method="post" action="<?php $this->descarga(); ?>"
    onsubmit="document.eliminar.env_report.value=window.location.href">
        <input type="hidden" name="env_report" id="env_report" value=""/>
        <input type="hidden" id="enviar" name="enviar" value="borrar"/>
        <input type="hidden" id="ruta" name="ruta" value="<?php echo $this->ruta ?>"/>
        <input type="hidden" id="file" name="file" value="<?php echo $this->file ?>"/>
        <input type="image" src="<?php echo site_url() . "/wp-content/themes/Comunidad-Mirth/_inc/images/RecycleRed.ico" ?>"/>
    </form>
    Borrar
    <?php } echo "</td>"; ?>

```

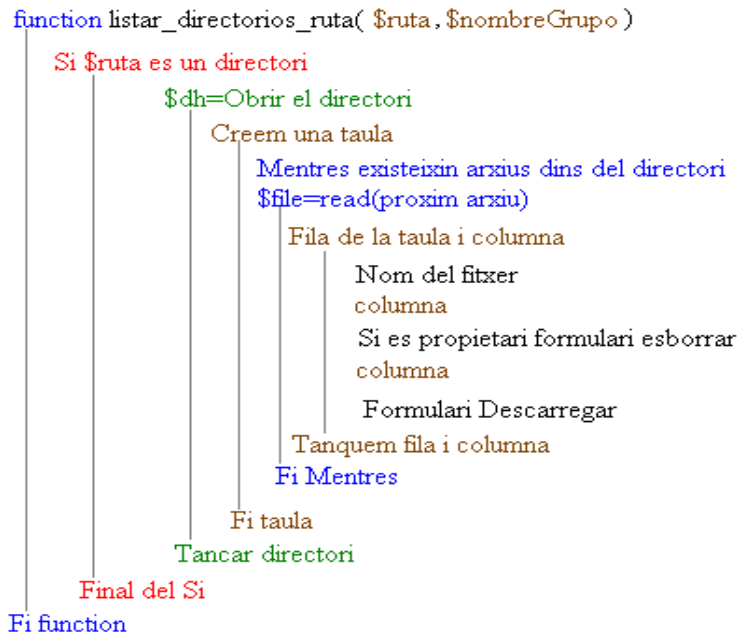
Variable resultado: Es guarda el resultat del select de la bbdd.

If: comparem si el propietari del arxiu es el actual.  
Si no ho es sortim, si ho es executem

Onsubmit: Creat per facilitar el refresc de la pàgina.  
Variable hidden env: On es guarda el resultat del onsubmit  
Variable hidden enviar: Que he de fer ? esborrar.  
Variable hidden ruta: Ruta del Arxiu.  
Variable hidden file: Nom del arxiu.



Perquè puguem veure la estructura d'aquesta funció creada per el alumne, de una manera gràfica, s'ha creat el següent esquema:



Com es de entendre l'últim cas d'us que va plantejar client va se la possibilitat de adjuntar arxius als comentaris realitzats per els usuaris, amb la mateixa estructura es va completar aquest cas d'us, però en vers de canviar de nom amb un **identificador es va recolzar amb el identificador del comentari.**

**Les puntuacions:** Dins del sistema de puntuacions s'ha valorat tres escenaris possibles dins de la aplicació.

**Figura 6.3.3 Pantalles Puntuacions**



**El primer,** es sense que l'usuari hagi puntuat el comentari, sortirà una fila que contindrà les possibles puntuacions del 1 al 10, que l'usuari amb un sol clic la enviarà per la valoració del comentari.



**El segon,** escenari es després de puntuar-lo, sortirà un comentari recordant que ja es va puntuar aquest i amb la puntuació que es va donar en aquell moment, com a possible millora podria ser el canvi de puntuació.



**El tercer,** es important saber que els comentaris propis no es poden valorar, imaginem que un usuari realitza 10, i en cada un d'ells s'assigna la major puntuació, tindriem un usuari que cap altre la valorat i en canvi obté una puntuació màxima, per

aquest motiu es va arribar a la conclusió que els propis usuaris no es poden valorar a ells mateixos.

Cada un d'aquestes versions de puntuació tenen un programari emmagatzemat bastant important però per evitar que la memòria sigui il·legible simplement donarem unes pinzellades de com un objecte \$puntuar ens facilitarà la vida, tots els objectes classe i demès que s'ha programat persegueixen aquest objectiu.

```

if ($yalohepuntuado == false && $escomentarioMio==false && $current_user->ID!=0) {
    for ($i = 1; $i < 11; $i++) {
        $content .=<tr>;
        $content .=<form id="puntuacion" name="puntuacion" method="post" action=" . $puntuar->puntuarComentario() . ">;
        $content .=<input type="hidden" name="Autor" id="Autor" style="color:darkorange;" value=" . $nombreDelAutor . "/>;
        $content .=<input type="hidden" name="IdComentario" id="IdComentario" style="color:darkorange;" value=" . $comment->id . "/>;
        $content .=<input type="hidden" name="Puntuacion" id="Puntuacion" style="color:darkorange;" value=" . $i . "/>;
        $content .=<input type="submit" name="Puntuar" id="Puntuar" style="color:darkorange;" value=" . $i . "/>;
        $content .=</form>;
        $content .=</tr>;
    }
} else if ($yalohepuntuado==true) {
    $content .=<tr>;
    $content .=<br>;
    $content .=<(Ya puntuastes este comentario con un: ' . $misPuntuaciones[$fila]->puntuacion.')>;
    $content .=</tr>;
}
else if ($escomentarioMio==true){
    $content .=<tr>;
    $content .=<br>;
    $content .=<(No se pueden puntuar los propios comentarios)>;
    $content .=</tr>;
}
}

```

Com podem observar en el primer if, es comprova si ja s'ha puntuat, si es un comentari propi i l'última opció que respon a \$current\_user->ID!=0 es perquè el usuari ha de estar donat d'alta en el sistema, ja que la eina Wordpress assigna un 0 a tots els usuari que no han fet lógin a la aplicació.

**Els contactes:** Quina es la motivació per enviar invitacions de amistat?,

Doncs bé saber que entre altres coses compartirem tots els nostres arxius amb els nostres contactes en l'apartat de contactes trobarem una llista de tots els usuaris que actualment estan donats d'alta, però amb la avantatge que els que son de la nostre llista de contactes podem descarregar els arxius que tinguin en el seu directori, per saber quins son només hem de cridar a la classe miDirectorio que en un principi s'utilitzava per donar suport a recórrer el directori del usuari però ara canviant la ruta del magatzem, obtindrem els arxius dels nostres contactes.

Una vegada obtenim si es contacte nostre o no, que no entrarem en codi extens, observem que amb un simple condicional obtindrem el resultat:

```

if($EsAmigo==false){
    echo "<h4> No es contacte teu no pots visualitzar els arxius </h4>";
}
else {
    echo"<h4 align='center' style='color:#00FF66'>Arxius pujats per " . $members_template->member->fullname. "</h4>";
    $ficherosAmigos->listar_directorios_ruta($ruta);
    unset($identificadorAmigos[$j]);
    $identificadorAmigos=array_values($identificadorAmigos);
}
}

```

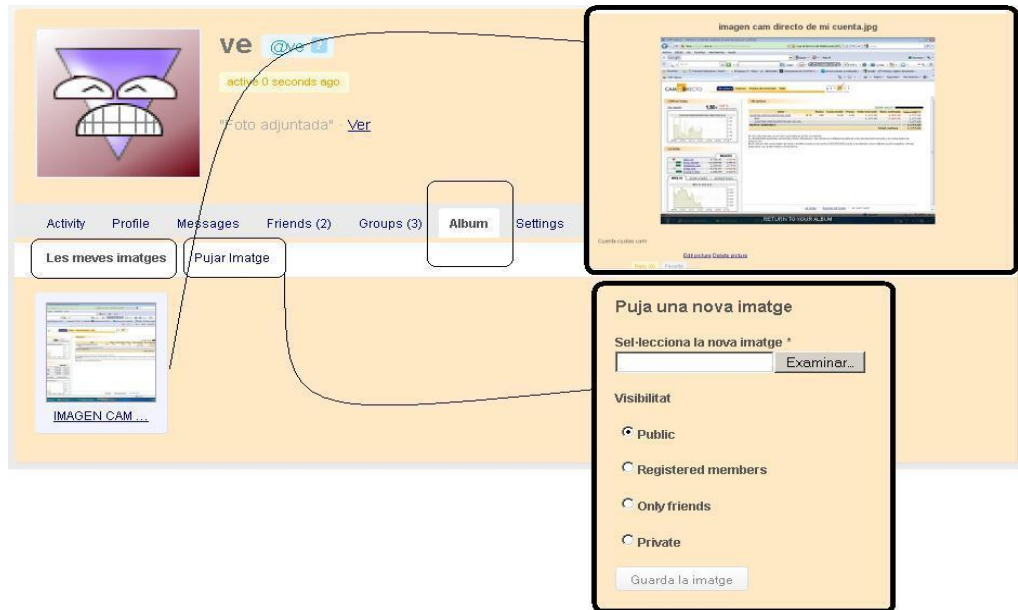
Si observem el últim codi de la pàgina anterior, veurem que el objecte \$ficherosAmigos, crida a la mateixa funció que es crida en el índex del usuari però canviant la ruta del directori d'aquesta manera:

```
$ficherosAmigos = new miDirectorio();  
$ruta = BASE_PATH . "/Uploads/" . $result[0]->user_nicename."/";
```

On això està dins de un bucle que genera un nou objecte per cada amic nostre, si no fos d'aquesta manera es cridaria a la mateixa funció i aniríem carregant tots els fitxers dels nostres amics en un sol espai, i el que realment interessa es saber de diferenciar els arxius per els nostres contactes.

**Altres Activitats:** Son aquestes activitats totes les demès que fan el funcionament de la comunitat, com enviar missatges privats, enviar mencions a usuaris o el fet de crear grups entre altres, a destacar he de comentar que s'ha habilitat un àlbum de imatges per usuaris, d'aquesta manera no solament es podrà compartir coneixement sinó que pot mostrar les seves imatges, compartir-les amb els seus amics o no, ja que s'ha definit quatre forma de pujar les imatges.

1. **Imatges públiques:** Qualsevol usuari pot visualitzar aquestes imatges, fins i tot aquells no estan donats d'alta en el sistema.
2. **Usuaris Registrats:** Aquestes imatges les veuran els usuaris que estiguin registrats en el sistema.
3. **Solament Amics:** Només seran visualitzades per els nostres contactes.
4. **Privades:** Només la podrà visualitzar el propietari de la imatge.

**Figura 6.3.3-1 Pantalla Àlbum**

En la barra de feines que hi ha dins de l'espai de l'usuari, trobem el Àlbum, si obrim es generarà les nostres imatges que hem pujat al servidor mitjançant el formulari de pujada, que ens donarà la possibilitat de adjuntar un comentari o descripció, la seva visibilitat, i si volem que es puguin fer comentaris de aquesta imatge des de la aplicació.

## 7. Possibles millores de la Aplicació.

Realitzant aquesta aplicació van sortir diferents requisits que es van anar ampliant conforme va creixent, aquests requisits es podrien dividir en dues parts.

### 7.1 Millores de funcionalitats.

En un futur seria important considerar la creació de blocs per part del usuari, aquest espais estarien destinats a veure un perfil més concret, de moment s'ha creat un enllaç que ens dirà quins comentaris ha generat aquest usuari, els seus amics, els seus grups etc..., però en un futur seria interessant ampliar aquesta informació.

Un altre aspecte son els fòrums, ja que en aquest moment es pot donar amb la creació de grups, però seria interessant crear temes tancats per part de l'administrador o inclús poder donar més importància a la creació de grups creats per el client.

## 7.2 Millores no funcionals.

Seria interessant en un futur canviar la versió del plugin de buddypress, però per poder realitzar aquesta tasca s'ha de tenir en conte tots els canvis realitzat per l'estudiant en el motor de buddypress.

Crear una nova línia de anàlisis del programari, durant el període de aprenentatge de la eina wordpress, així com del llenguatge, es va definir una manera de actuar el més semblant a la orientació objecte però sense que el llenguatge estigui preparat, es complexa arribar a una funcionalitat semblant a la que ofereix el llenguatge java.

## 8. Conclusions del Estudiant.

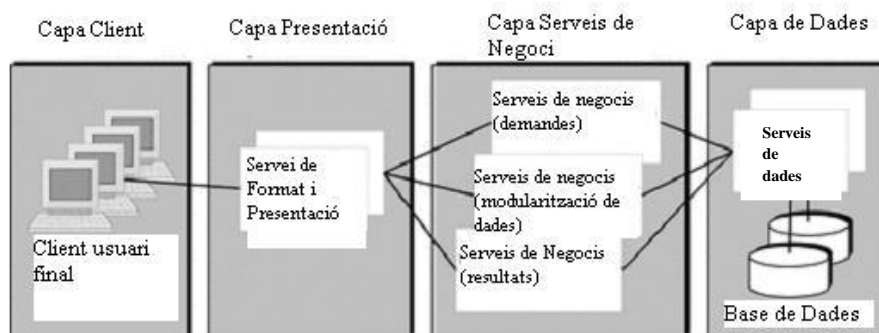
Les conclusions que s'han extret de la realització d'aquest TFC son extenses, però amb el mateix criteri que he intentat no introduir comentaris innecessaris dins de la memòria i obtenir una memòria fàcil d'entendre i llegir realitzaré aquest apartat dividit en dos punts.

### 8.1 Sobre el treball realitzat.

**Modularització:** entenem la separació modular d'un sistema o aplicació sigui quina sigui la seva naturalesa o propòsit. Concretament en aquestes conclusions, es tracta d'un anàlisi entre dues tecnologies de desenvolupament web, considerem el terme Modularització com la separació en capes definides en un model MVC (Model Vista-Controlador). La modularitat d'un sistema té vital importància en l'aspecte de la consistència, robustesa, mantenibilitat i altres aspectes que detallarem més endavant.

Possiblement en aquesta característica es basin altres que veurem posteriorment per la qual cosa hem de prestar especial atenció. Atenent a la definició i estructura de PHP i Java podem dir que hi ha una gran diferència entre ambdós en aquest àmbit.

La tecnologia Java usada en qualsevol portal web té una estructura clarament diferenciada, podent diferenciar amb facilitat el model MVC amb els seus diferents mòduls:



**Figura 8.1** Modularització

Pel que fa a PHP, podem dir que perdem una mica la pista de la modularitat que hem destacat en java ja que totes les capes lògiques són implementades en un mateix arxiu.

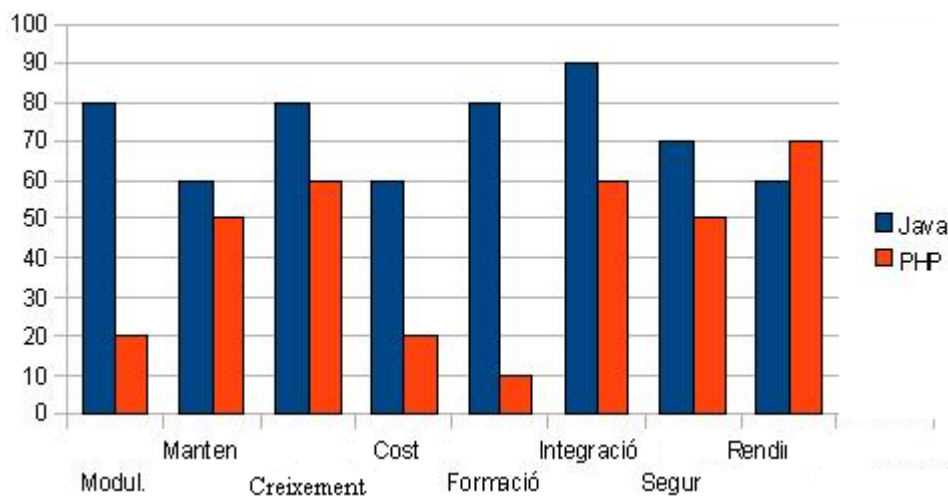
**El manteniment:** El manteniment del sistema és una part fonamental en el cicle de vida de qualsevol projecte que estiguem tractant, i està estretament relacionada amb la tecnologia que hem triat en l'etapa de disseny.

Per realitzar l'anàlisi de les tècniques que estem tractant ens hem de remuntar de nou al punt anterior per aconseguir treure una conclusió ferma. Un sistema en què hi hagi una estructura clara dels seus components serà més fàcilment mantenible en un futur ja que serà necessari el seguiment d'una metodologia ja definida.

Per experiència sabem que una vegada finalitzat el desenvolupament, pot passar que les millores del sistema no siguin implementades per el programador original i sí per una altra persona o empresa externa. El fet que PHP sigui una tècnica poc estructurada i que el desenvolupador no sigui ple coneixedor de l'estratègia seguida en la programació original, pot donar origen a un empobriment del codi, repercutint normalment al seu rendiment.

Potser un dels avantatges de PHP davant Java sigui en qüestió de rendiment ja que el primer és molt menys pesat, el que produeix una sensació a l'usuari de rapidesa i major usabilitat.

**Taula 8.1 Comparativa PHP – JAVA**



- Referència: Web oficial php, Wikipedia

Un altre dels grans problemes que he trobat a estat la poca facilitat a l'hora de supervisar el codi, com a exemple veurem dues capçaleres de mètodes.

**Php:** públic function XXXXX (\$id){ }

**Java:** públic ArrayList XXXXX (int id){ }

Podem observar que en el primer cas no sabem el retorn del mètode, clar si ha retorn perquè això ho haurem de veure repassant tot el codi de la funció, tampoc sabem de quin tipus es la variable que hem de passar, això ens complica la vida a l'hora de adaptar el codi de la eina a utilitzar.



En el segon cas no només sabem que fa un retorn sinó que ens retornarà un ArrayList a més sabem que el tipus de variable que s'ha de passar al mètode es un int, no cal dir que a l'hora de mantenir-ho només hem de fer una ullada a la capçalera per saber quin es el mètode que hem de tocar.

Per el meu parer aquest projecte o treball si es vol fer en java, s'hauria de fer un petit estudi de la facilitat de manteniment i distribuir la feina en diferents projectistes per tal de que el projecte final sigui una eina fàcil de mantenir.

A manera de conclusió crec que la idea del client es bona ja que no existeix cap tipus de xarxa social o comunitat relacionada amb el tema que es tracta, però crec que la seva decisió de realitzar aquesta aplicació amb aquest llenguatge i aquesta eina es per obtenir uns resultats ràpids i poder arribar a temps, (entre altres coses perquè la corba de aprenentatge del llenguatge es ràpid, no així de la eina).

Si això es volgués mantenir segurament el llenguatge a utilitzar seria un altre que tingues molta més modularitat.

## 8.2 Sobre la formació obtinguda.

Si alguna cosa es característica de les formacions relacionades amb les TIC, és que el procés de aprenentatge mai s'acaba, no només s'ha de formar quan estudies sinó que també quan surts al mercat laborals. És important saber tenir versatilitat per enfrontar-se a nous reptes i noves tecnologies per aquest motiu puc dir que aquest projecte m'ha aportat formació en llenguatges nous i eines que fins ara no havia manipulats.

També apreciar la formació que fins ara he obtingut, i corroborant això que sempre m'han dit, una aplicació mal estructurada “ si la comparem amb altres llenguatges “ es més difícil de modificar o mantenir.

Algunes de les eines que he utilitzat son, jquery ( biblioteca de javascript ), ajax, php, mysql, html, css, fpdf,javascript, y a més si fiquem la primera versió implementada en java com a prova pilot podria afegir, jsp, jstl, struts, etc..., si que es cert que potser la complicació ha estat intentar implementar la filosofia de objectes en php.

## 9. Bibliografia i webs consultades.

### 9.1 Bibliografia

Amb el gran ventall de possibilitats que existeixen per obtenir el mateix resultat electrònicament via web es pot entendre que la bibliografia consultada no sigui gaire extensa.

**Quijano, Andrés**

(Programación web Java) – 1ª ed. Buenos Aires MP Edicions, 2006. 352 pàgines  
ISBN:978-526-391-5.

**Cevallos, Francisco Javier**

(JAVA 2 Interfaces gráficas y aplicaciones para Internet) – 3ª ed. RA-MA editorial 687  
páginas Madrid, 2008 ISBN:978-84-7897-859-5.

**Heuter, Oliver**

(PHP Introducción y primeros pasos) Col·lecció : TechNote 221 pàgines ISBN:978-2-  
7460-5019-8

**Heuter, Oliver**

(PHP 5.3) Col·lecció Recursos Informaticos 497 pàgines ISBN:978-2-7460-6666-3

**Aubry, Cristophe**

(PHP 5.3) Col·lecció Soluciones Informaticas 406 pàgines ISBN: 978-2-7460-5326-7

## 9.2 Webs consultades

Entre altres aspectes es quasi bé impossible enumerar totes les pàgines webs consultades en la realització del TFC així que es numera les 10 més importants.

<http://www.forosdelweb.com/>

Comunitat de informàtica útil ja que té els seus temes ven diferenciats.

<http://www.lawebdelprogramador.com/>

Web que ens proporciona tant tutorials com informació de les diferents eines informàtiques a més de tenir un sistema de preguntes i respostes.

<http://www.desarrolloweb.com/>

Espai que conté diferent informació sobre tot basat en php

<http://foros.cristalab.com/>

Aquesta web està destinat exclusivament a php i mysql.

<http://www.webestilo.com/>

Es destina a albergar diferents temes entre els quals està php.

<http://php.net/manual/>

La pàgina més completa que he trobat, conté la informació necessària per poder entendre els mètodes que conte la eina de php.

<http://wordpress.org/>

Web relacionada amb la eina de Wordpress.

<http://es.buddypress.org/home/>

Web oficial de buddypress i temes relacionats amb aquest plugin.

<http://buddypress-es.org/>

Comunitat en castellà del plugin de buddypress.

[http://codex.wordpress.org/es:Referencia\\_de\\_Funciones/](http://codex.wordpress.org/es:Referencia_de_Funciones/)

Referència de totes les funcions disponibles del motor de Wordpress.

## **Escola Universitària Politécnica de Mataró**

Centre adscrit a:



**UNIVERSITAT POLITÈCNICA  
DE CATALUNYA**

**Enginyeria Informàtica de Gestió**

**COMUNITAT MIRTH**

**VERSIÓ 1.0**

**Cost econòmic**

**FELIPE CORTÉS SAYAVERA  
PONENT: EDUARD DE BRU DE SALA**

**TARDOR 2011**

## 10. Cost del prototip.

### 10.1. Costos de recursos humans.

<u>Concepte</u>	<u>Hores</u>	<u>Preu/hora (€)</u>	<u>Total (€)</u>
Estudi i documentació (Enginyer júnior) (25% ja que la formació obtinguda serà amortitzada en altres projectes)	25 % de 50 hores  13	35	455
Desenvolupament (Enginyer sènior)	200	40	8000
Hores de proves (Enginyer júnior)	40	15	600
Redacció memòria (Documentalista)	50	15	750
<b>TOTAL RECURSOS HUMANS</b>			<b>9805,00</b>

## 10.2. Amortització equips, instrumental i software.

<u>Equip utilitzat</u>	<u>Hores d'utilització</u>	<u>Preu/hora</u>	<u>Total</u>
<u>Equips i programari informàtic</u>			
* Ordinador	340	0,11	38
Software Neetbeans 7.0	200	0,0	0
Software MySQL	100	0,00	0
Software PHP	200	0,00	0
Software Ofimàtica	50	0,00	0
Software Wordpress	50	0,00	0
Software Buddypress	50	0,00	0
OpenOffice	50	0,00	0
<b>TOTAL AMORTITZACIONS</b>			<b>38,00</b>

\* Ordinador : com a regla empírica es considera que els ordinadors es canvien cada tres a cinc anys, amb una utilització de 40 hores setmanals ens donarà una vida útil de la computadora de uns 8320 hores, calculada a 4 anys.

El preu mig de una pc està sobre els 900 euros surtin a 0.11 cèntims la hora de utilització, si calculem les 340 hores son 38 euros.

**10.3. Cost total.**

Concepte	Preu
Costos de recursos humans	<b>9805,00€</b>
Amortització equips, instrumental i software	<b>38 €</b>
<b>TOTAL</b>	<b>9843,00€</b>