



Universitat Autònoma  
de Barcelona

# Pàgina web pel Club Bellaterra

Memòria del projecte  
d'Enginyeria Tècnica en  
Informàtica de Gestió  
realitzat per

*Eric Martínez Mariné*

i dirigit per

*Daniel Blabia Girau*

**Escola d'Enginyeria**

Sabadell, *Setembre* de 2010

[El/La] sotasignat, ***Daniel Blabia Girau***  
professor[/a] de l'Escola d'Enginyeria de la UAB,

**CERTIFICA:**

Que el treball al que correspon la present  
memòria ha estat realitzat sota la seva  
direcció per en ***Eric Martínez Mariné***

I per a que consti firma la present.

Sabadell, ***Setembre*** de ***2010***

-----  
Signat: ***Daniel Blabia Girau***

## Índex

1. Introducció	
1.1 Descripció.....	5
1.2 Objectius .....	5
1.3 Estat de l'art .....	6
1.4 Motivacions personals .....	7
1.5 Estructura de la memòria .....	8
2. Anàlisi i planificació	
2.1 Requeriments funcionals.....	9
2.2 Requeriments no funcionals .....	10
2.3 Mètode de desenvolupament.....	12
2.4 Eines de desenvolupament.....	15
2.5 Riscos.....	16
2.6 Planificació temporal.....	16
3. Marc teòric	
3.1 Apache HTTP Server.....	19
3.2 HTML.....	19
3.3 CSS.....	20
3.4 Javascript .....	21
3.5 jQuery.....	22
3.6 MySQL.....	22
3.7 PHP.....	24
4. Disseny	
4.1 Casos d'ús.....	26
4.2 Arquitectura del sistema .....	33
4.3 Base de dades.....	41
4.4 Documentació sobre el disseny.....	44
5. Proves .....	47
6. Conclusions	
6.1 Assoliment d'objectius .....	49
6.2 Línies d'ampliació.....	49
7. Bibliografia.....	51

# Resum introductori

Aquesta memòria explica el desenvolupament del projecte de final de carrera d'Enginyeria Tècnica en Informàtica de Gestió proposat pel meu director de projecte.

El projecte sorgeix per a oferir als socis d'aquesta entitat esportiva un nou servei per tal de que puguin sentir-se més pròxims al Club.

Per a realitzar el projecte s'han seguit totes les fases necessàries, sent una part molt important la anàlisi del mateix i la definició dels objectius i requeriments abans de començar a programar, segons les necessitats del client i dels futurs usuaris. El projecte es començava de zero.

# 1. Introducció

## 1.1 Descripció

El Club Bellaterra fou la primera entitat esportiva de Bellaterra, fundada el 1934 per la iniciativa i inquietud de diversos veïns per tenir un lloc on trobar-se, fer esport i passar una bona estona. Sent fidels a aquells emocionants inicis, el Club Bellaterra, segueix i seguirà sent un punt de trobada, i ara més que mai, una referència per a la pràctica esportiva i saludable, ja que comptarà amb els recursos necessaris per esdevenir un espai dinàmic, actiu i accessible.

El Club Bellaterra és un club amb molta força i importància en l'àmbit social d'aquest petit municipi. Ara fa poc que el Club ha acabat un procés de reconstrucció i modernització i com que els socis són els propietaris del club, se'ls vol brindar un servei de gran utilitat per a ells.

Ja és sabut que avui en dia resulta molt important per a qualsevol empresa, societat o, com en el cas del projecte, per a una entitat esportiva tenir un lloc web per tal d'oferir un bon ventall d'avantatges al col·lectiu que en forma part. A més, això pot obrir fronteres, podent aconseguir nous socis i també nous patrocinadors, la qual cosa ajudaria a que el Club pogués seguir creixent, especialment ara que s'ha realitzat la reconstrucció íntegra del Club.

Per aquests motius es vol que aquesta web sigui d'un ús simple, amigable i confortable. Es vol mantenir informant a tot visitant dels serveis i instal·lacions que ofereix el Club, a més de comunicar notícies d'interès, per tal d'atreure, tan a nous visitants, com als socis, els quals tindran accés a zones reservades per a ells. Un fòrum, un taulell d'anuncis, un sistema de reserva de pistes per a pàdel i tennis on-line i una galeria de fotografies d'activitats realitzades en l'entorn del Club.

## 1.2 Objectius

En aquest projecte es busca, sobretot, que els socis del Club facin ús de la web. Per tant, és important que s'animin a entrar a la web, que vegin que el seu ús és simple, i que hi participin en la part que només ells utilitzaran, la Zona Socis.

Per tant els principals objectius serien:

Pàgina inicial en la qual es pugui intuir tot el contingut web.

Fent dos o tres clics de ratolí un usuari ha de poder trobar allò que busca.

Actualització de la informació que apareixerà en la web.

Aconseguir que gran part dels socis es registrin i facin ús de la Intranet.  
L'accés a la Intranet només pot ser realitzat per a socis que es registrin.

Per a què això sigui realitat seria important que a través d'un menú desplegable, el qual donaria més vistositat, es pogués accedir a tot el contingut.

Els usuaris que poden entrar a la web seran de totes les edats i no es pot saber si estan gaire acostumats a navegar per Internet, i menys, a utilitzar una Intranet. Així doncs el Club els informarà, en cas de ser necessari, de com fer-ho, i a la web també trobaran indicacions per tal de no resultar complicat.

### 1.3 Estat de l'art

Aquesta entitat esportiva disposava fins el passat mes de març d'una pàgina web estàtica i molt simple en la qual: en l'apartat obres es poden veure els vídeos de les obres de remodelació del recinte esportiu; al de notícies alguna notificació purament informativa; al nou club hi ha imatges de com serà el club una vegada finalitzades les obres. En la figura 1 en tenim una captura.

#### Estem treballant amb la nova pàgina web



**Carrer Lluís d'Abalo s/n**  
(actualment tenim les oficines al **Centre Cívic de Bellaterra**)  
Plaça Maragall  
elclub@club-bellaterra.com + 93 580 25 42

Figura 1

A partir del març i amb la idea de com seria finalment la web al finalitzar el projecte, el Club va afegir contingut per tal d'anar informant als visitants d'algunes notícies i activitats a realitzar-se. En la figura de 2 tenim la captura d'imatge de la pàgina inicial



**Figura 2**

## 1.4 Motivacions

La principal motivació per a realitzar aquest projecte va ser el fet de poder crear una pàgina web, negociant els requeriments directament amb el client, semblant de si es tractés d'un primer contacte amb el món laboral.

El projecte es desenvolupa en PHP i MySQL en la part de la Intranet. D'aquests llenguatges no en tenia una gran base abans, per tant abans de començar he hagut de buscar tutorials i exemples abans de poder començar. La part estàtica serà en HTML, a més de funcions de Javascript per tal de donar-li més

vistositat.

## **1.5 Estructura de la memòria**

A partir d'aquest primer capítol que serveix d'introducció al projecte, en seguiran sis més.

En el segon capítol es realitzarà la fase d'anàlisi i la planificació del projecte.

En el tercer capítol hi haurà el marc teòric on s'explicarà amb detall les tecnologies utilitzades per a desenvolupar aquest projecte.

En el quart capítol s'hi trobarà la fase del disseny amb els casos d'ús, la base de dades i la interfície.

En el cinquè capítol es tractarà aspectes de la implementació, com ara algunes proves dutes a terme.

En el sisè capítol hi ha les conclusions del projecte, en les quals es trobaran: els assoliments dels objectius, modificacions en els requeriments i en la planificació i possibles ampliacions en el projecte.

Per acabar, en el setè capítol hi ha la bibliografia amb enllaços a diferents webs que han servit de referència i ajuda per desenvolupar el projecte.



## **2. Anàlisi i planificació**

### **2.1 Requeriments funcionals**

#### **Intranet**

Per accedir a la Intranet l'usuari cal que s'identifiqui. Per a controlar-ho es farà servir la variable SESSION, la qual serveix per guardar la informació referent a l'usuari.

En cas de ser un usuari que accedeix per primera vegada, per tant que no estarà registrat, caldrà fer una petició de registre, la qual serà enviada als administradors per afegir l'usuari a la base de dades.

Si l'usuari desitja finalitzar la sessió haurà de fer el log out corresponent matant la variable SESSION.

#### **Fòrum i taulell d'anuncis**

Un cop iniciada la sessió l'usuari podrà accedir-hi per llegir els diferents temes oberts i si ho desitja citar-ne a un mitjançant l'identificador de respostes o començar un nou tema. La id corresponent a la taula Socis de la base de dades, serà la que permetrà identificar qui edita els missatges.

Es faran consultes de inserció i de cerca.

#### **Galeria**

Aquí els usuaris amb sessió ja iniciada podran visionar les diferents galeries de fotos, accedint als arxius del servidor on estaran allotjats els diferents àlbums. Les fotos es visionaran mitjançant funcions de Javascript.

#### **Sistema de reserves on-line**

Els usuaris que hagin iniciat sessió podran accedir-hi per consultar la disponibilitat de les pistes. Es realitzaran consultes per veure l'estat de cada pista al llarg de tota la setmana.

Es mostraran els registres de les reserves, si n'hi ha, a la pàgina. Si l'usuari vol demanar una pista farà una consulta per inserir un nou registre per mitjà de les llistes desplegable.

### **Interacció en les diferents pàgines**

La interacció per les diferents pàgines es farà mitjançant els enllaços creats al menú o en el contingut de les diferent pàgines.

Es podrà interactuar per les diferents pàgines de la Intranet sempre i la variable SESSION segueixi activa.

Aquesta variable es mantindrà fins que es tanqui el navegador, o es premi l'enllaç logout.

## **2.2 Requeriments no funcionals**

### **Intranet**

Per accedir a la Intranet caldrà iniciar sessió. Per tal de fer-ho caldrà accedir al camp del menú Zona Socis on es demanarà el log in o el registre en cas de que l'usuari no s'hagi registrat anteriorment. En cas de fer el registre l'usuari omplirà un petit formulari amb les seves dades, les quals arribaran

### **Notícies**

Obertes a tots els visitants de la web, els quals podran llegir el registre de l'última notícia a la pàgina principal. Des de aquesta última notícia podran accedir a l'arxiu de totes les altres notícies anteriors per llegir-ne el contingut de les mateixes.

### **Fòrum i taulell d'anuncis**

Quan un usuari ja està identificat podrà llegir els missatges i si ho desitja respondre el missatge, citant-lo, i apareixent aquest com a última resposta a la pàgina principal del fòrum o del taulell d'anuncis.

A més podrà començar un nou tema triant-ne el títol.

## **Galeria**

L'usuari ja identificat podrà veure els diferents àlbums i les seves fotos.

Al fer clic a una foto d'un àlbum, l'usuari veurà la foto ampliada i podrà seguir veient les següents i/o anteriors fent clic al botó de següent o d'anterior respectivament.

Serà una manera fàcil i atractiva per veure les fotos.

## **Interacció en les diferents pàgines**

La interacció es farà la majoria de vegades pel menú i no caldrà buscar molt per trobar la informació desitjada.

Un cop identificat a la Zona Socis es pot tornar a la estàtica o pública de la web a través del menú, per després tornar a la Intranet, sempre i quan no hagi caducat la sessió, si l'usuari ho desitja.

## **Sistema de reserves on-line**

Primerament l'usuari identificat haurà d'elegir quina pista vol per a la reserva.

N'hi haurà quatre: tres de pàdel i una de tenis.

Llavors, podrà observar si hi ha alguna reserva per aquesta setmana, en la pista que hagi elegit. El següent serà que aquest usuari seleccioni el dia de la setmana i la hora en la que vol realitzar aquesta reserva.

Estar controlat que cap usuari pugui reservar més d'una pista en la mateixa hora i el mateix dia.

## **Condicions d'ús i funcionament de la Intranet**

Els usuaris s'hauran de registrar a la web, a la pàgina Zona Socis. Només seran donats d'alta sempre i quan siguin socis del Club. Llavors rebran un correu electrònic per tal de que ja puguin accedir a la seva conta. És important que la direcció de correu electrònic sigui correcta.

Un cop registrats de forma correcta, els usuaris podran accedir a tot el

contingut de Zona Socis: el sistema de reserves de pistes on-line, el fòrum, el taulell d'anuncis i la galeria de fotos.

Els usuaris que vulguin reservar pista podran triar de les hores disponibles de les quatre diferents pistes que hi ha. Només necessitarà seleccionar l'hora del dia que ell prefereixi.

Pel que fa al fòrum i al taulell d'anuncis, tot i aparentment semblar iguals, el contingut per ambdós serà diferent. En el fòrum es tractaran temes del club, d'esport, d'oci o de qualsevol tipus que no sigui en benefici comercial o en forma de patrocini. En canvi, al taulell d'anuncis els usuaris podran escriure sobre informació referent a anuncis comercials, ja siguin per anunciar una venda o oferir-se per a donar classes entre altres. Es podrà sancionar el fet de no respectar això últim, sobretot si és de forma reiterativa.

Els dos tindran un comú, el respecte envers la resta d'usuaris sigui quina sigui la raça, sexe o orientació política. Qui no mostri respecte podrà ser sancionat fins a perdre el seu accés a la Zona Socis.

Per últim, en la galeria de fotos els socis podran veure fotos de diferents actes o activitats organitzades pel Club. En cas de que algun soci desitgi aportar material fotogràfic per a la web, ho podrà fer posant-se en contacte amb el Club, ja sigui a través de la pàgina de contacte o dirigint-se a les oficines del Club. En tot cas queda prohibida la comercialització i l'ús indegut de les fotografies de la galeria ja que un cop publicades a la web són propietat del Club.

## **2.3 Mètodes de desenvolupament**

El terme cicle de vida del software descriu el desenvolupament del software, des de la fase inicial fins a la fase final. El propòsit d'aquest programa és definir les diferents fases intermèdies que es requereixin per a validar el desenvolupament de l'aplicació i poder garantir que el software compleixi els requisits per a l'aplicació.

Els models prescriptius de procés es van proposar originalment per a ordenar el caos del desenvolupament del software. Aquests models convencionals proveeixen unes estructures útils pel treball de la enginyeria del software, proporcionant un camí raonablement efectiu a seguir als equips de software.

Cada model descriu un flux de treball, és a dir, la forma en la qual els processos es relacionen entre si.

Per a realitzar el projecte es van tenir en comte dos models diferents de desenvolupament del software: el model seqüencial, o model de cascada, i el model iteratiu.

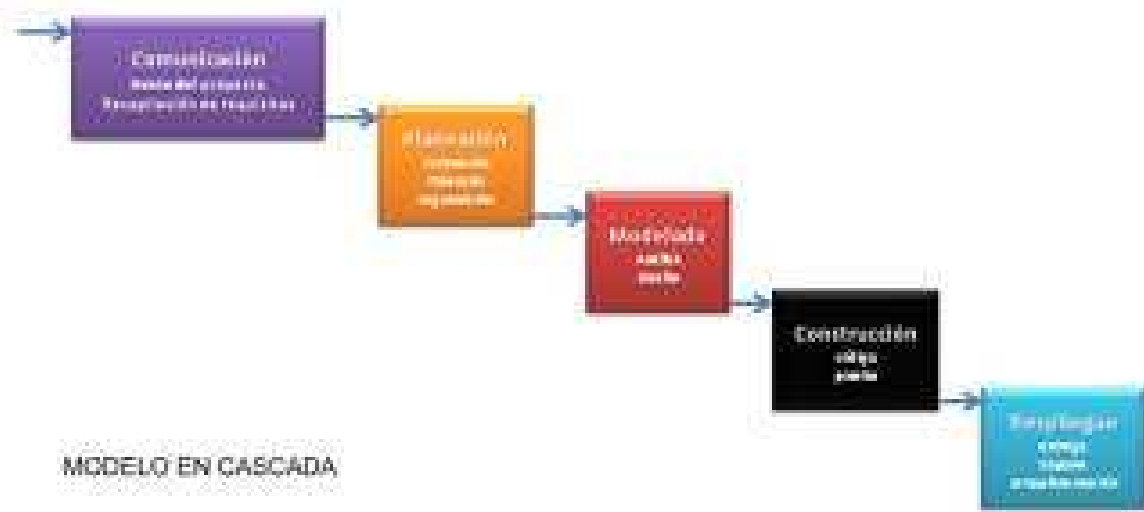
El model seqüencial es pot utilitzar quan:

- Els requisits del problema s'entenen raonablement.
- Per a fer adaptacions o millores ben definides a un sistema existent.
- Limitat en projectes nous a desenvolupar, només si presenten uns requisits ben definits i estables al llarg del procés de desenvolupament.

Aquest model pot presentar problemes ja que:

- Normalment els projectes reals no acostumen a seguir un flux seqüencial. Té certes iteracions indirectes, el qual pot arribar a confondre el desenvolupador.
- Frequentment és difícil pel client establir tots els requisits des d'un principi.
- El client ha de tenir paciència ja que la versió que funcioni estarà llesta amb el projecte ben avançat.

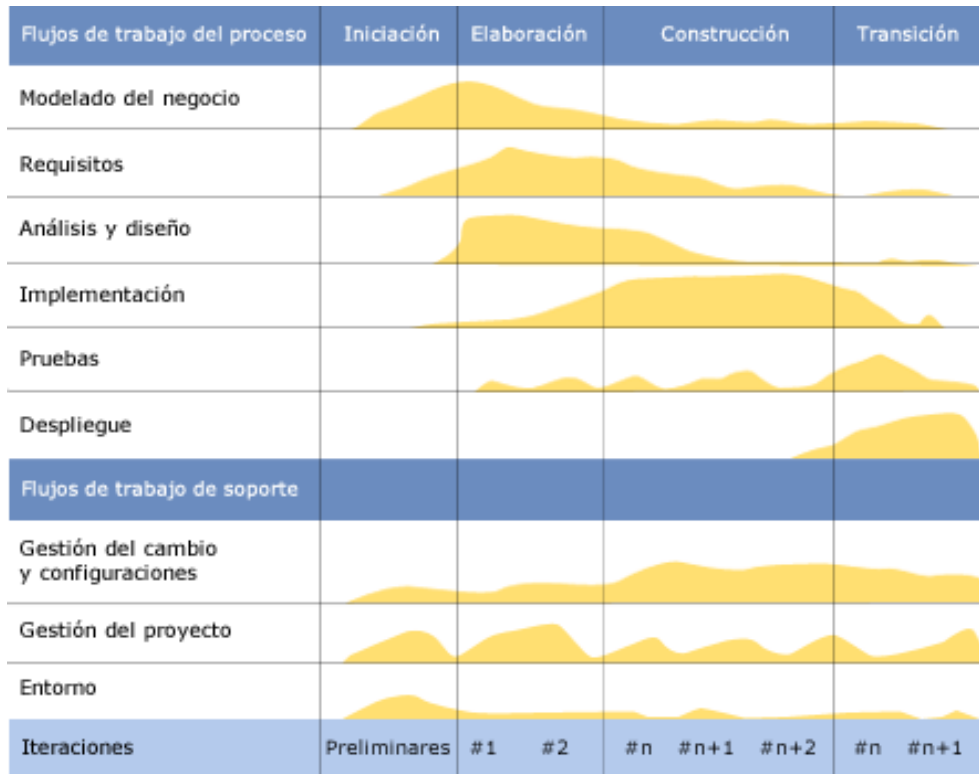
Un exemple gràfic d'aquest model seria la figura 2.1.



**Figura 2.1**

El model iteratiu es basa en separar el desenvolupament per etapes, anomenades iteracions. En la figura 2.2 es podrà observar un exemple gràfic on es veurà que al inici les fases més rellevants seran les d'anàlisi i disseny, per

anar deixant pas a les d'implementació i de proves, a mesura que anava avançant el procés de desenvolupament del software.



**Figura 2.2**

Aquest model permet més flexibilitat amb canvis en els requisits i permet millorar l'expectativa del producte final. En contra, té els inconvenients d'incrementar el risc en errar en l'estimació del temps de desenvolupament del projecte respecte el previst inicialment.

Havent vist les condicions anteriors sobre els dos models, s'ha decidit de fer ús del model seqüencial. Aquest model anirà bé per tal d'evitar el risc d'allargar el temps de desenvolupament del projecte. Un altre factor a tenir en comte, és la de separar les fases de forma clara, ja que al ser una sola persona qui ho portarà a terme, en facilitarà la feina. Finalment, el fet de tenir un client amb uns requeriments ben definits i amb poques variants i requeriments opcionals, ha permès de tirar-ho endavant amb el procés seqüencial.

## 2.4 Eines de desenvolupament

Els llenguatges de programació que es faran servir en el projecte són: HTML, PHP, MySQL i Javascript, que en el cas del menú desplegable es farà per mitjà de llibreries jQuery.

El fet de decantar-se per a aquests llenguatges és que són llenguatges de software lliure, exemptes de pagar llicències, que són còmodes d'utilitzar i a la alta fiabilitat que donen.

En el desenvolupament del projecte es farà servir un servidor Apache, ja que a més de ser gratuït, és fàcil de fer servir i molt segur.

Per tal de configurar i posar en marxa l'entorn d'aplicació d'una manera ràpida i fiable, s'instal·larà el Appserv, la qual és una eina de OpenSource per a Windows. Aquesta porta tot un paquet amb les principals eines que faran falta per al desenvolupament de la pàgina web del projecte. Appserv, doncs facilitarà la instal·lació de Apache, MySQL i PHP en la qual aquestes aplicacions es configuren de forma automàtica. L'únic que s'haurà de realitzar és crear una contrasenya per l'usuari *root* (administrador) per a la base de dades de MySQL. A més incorpora el phpMyAdmin per tal de gestionar la base de dades en MySQL, permetent crear, modificar, eliminar i consultar totes les taules de manera fàcil i ràpida. Serà de gran ajuda per a comprovar que les consultes que es voldran fer al desenvolupar el projecte, ja que en cas d'error, facilita localitzar-lo amb èxit.

La versió utilitzada del Appserv per a desenvolupar el projecte ha estat la 2.5.9, i a més, inclou la versió 2.2.4 de l'Apache, la 5.2.3 de PHP, la 5.0.45 de MySQL i, finalment, la 2.10.2 de phpMyAdmin.

Alhora de modificar i depurar contingut de la web es farà servir una eina anomenada Firebug. Es tracta d'una extensió de Firefox creada i dissenyada especialment per desenvolupadors i programadors web. És un paquet d'utilitats amb el qual es pot analitzar, editar monitoritzar i depurar el codi font, CSS, HTML i Javascript d'una pàgina web de manera instantània. Amb aquesta eina també es podran guardar els canvis que es realitzin. Firebug està encapsulat en forma de plug in o complement de Mozilla, és OpenSource, lliure i de distribució gratuïta. La versió de Firebug utilitzada per a desenvolupar el projecte és la 1.5.4.

Amb el Firebug hi ha hagut un molt bon suport, però per la part del projecte feta en PHP i MySQL s'ha escollit una IDE, per tal d'aprofitar que completen el codi i ressaltin errors en la sintaxi, al contrari d'algun editor de text com el Bloc de Notes. S'ha escollit l'Eclipse en detriment de NetBeans. Els dos són els IDEs

gratuïts més populars i el fet d'elegir l'Eclipse rau en motius de ser una eina més usada en empreses, per tant més pròxima al món laboral. Alhora de desenvolupar el projecte s'ha emprat l'Eclipse per a fer més emena la interacció entre les diferents carpetes i arxius de la pàgina web, pel fet de tenir-los ja carregats com a un projecte.

## 2.5 Riscos

Alhora de realitzar aquest projecte existeixen alguns riscos a tenir en comte, els quals es detallen a continuació:

- Planificació temporal optimista: és un risc amb probabilitat alta. A no ser que s'allargui en excés no comportarà a greus problemes.
- Canvi de requisits: aquest és risc relativament probable, a més aquests canvis serien més aviat algunes opcions a tenir en comte. Els canvis en tot cas no haurien d'afectar massa a la planificació.
- Dificultat per accedir als *stakeholders*: aquest risc de succeir podria fer augmentar el temps del projecte, sobretot alhora de realitzar l'anàlisi.
- No es fa correctament la fase de test: aquest seria un risc crític, ja que al tractar-se d'una web l'accés a les diferents pàgines i, sobretot, la Intranet, la qual hauria de funcionar sense errades.

## 2.6 Planificació







El projecte es desenvoluparà del 15 de març de 2010 al 13 de setembre de 2010.

La part d'anàlisi i del disseny es realitzaran en el període del 15 de març al 28 d'abril. Entre mig d'aquest període hi haurà una parada degut a la Setmana Santa. La dedicació, en aquest cas, serà d'unes 20 hores setmanals.

Les següents parts es realitzaran entre el 21 de juny de 2010 i el 13 de setembre de 2010. Aquest espai de temps que va de l'Abril fins el juny és degut a motius laborals i acadèmics.

A continuació, es presenten el nom dels recursos pel projecte en la figura 2.3:



		Nombre	Tipo
1		Analista	Trabajo
2		Programador	Trabajo
3		Director de projecte	Trabajo
4		Cap de projecte	Trabajo
5		Tècnic de proves	Trabajo

**Figura 2.3**

Les tasques del projecte, amb la seva duració, i la planificació temporal del mateix, mitjançant el Diagrama de Gantt, es presenten a continuació en la figura 2.4 i la 2.5:

	Nombre	Duración	Nombres del Recurso
1	<input type="checkbox"/> <b>Web Club Bellaterra</b>	<b>80,5 days</b>	
2	<input type="checkbox"/> <b>Anàlisi</b>	<b>11,5 days</b>	
3	Anàlisi de requisits	4 days	Analista
4	Anàlisi del Club Bellaterra	6 days	Analista
5	Documentació de l'anàlisi	1 day	Analista
6	Aprovació de l'anàlisi	0,5 days	Analista[50%];Cap de projecte[50%]
7	<input type="checkbox"/> <b>Disseny</b>	<b>10,5 days</b>	
8	Disseny de la base de da	3 days	Analista[50%];Programador[50%]
9	Disseny de la part estàtic	3 days	Analista[50%];Programador[50%]
10	Disseny de la intranet	5 days	Analista[50%];Programador[50%]
11	Documentació del disseny	2 days	Analista[50%];Programador[50%]
12	Aprovació del disseny	0,5 days	Analista[50%];Cap de projecte[50%]
13	<input type="checkbox"/> <b>Desenvolupament</b>	<b>37,5 days</b>	
14	Preparació de l'entorn de	3 days	Programador
15	Configuració accés intranet	3 days	Programador
16	Desenvolupament web	6 days	Programador
17	Desenvolupament intranet	25 days	Programador
18	Aprovació del desenvolup	0,5 days	Analista[50%];Cap de projecte[50%]
19	Proves	4 days	Tècnic de proves
20	Redacció de la memòria	14 days	Cap de projecte
21	Tancament del projecte	0,5 days	Cap de projecte[50%];Director de projecte[50%]
22	Defensa del projecte	0,5 days	Cap de projecte[50%];Director de projecte[50%]

**Figura 2.4**

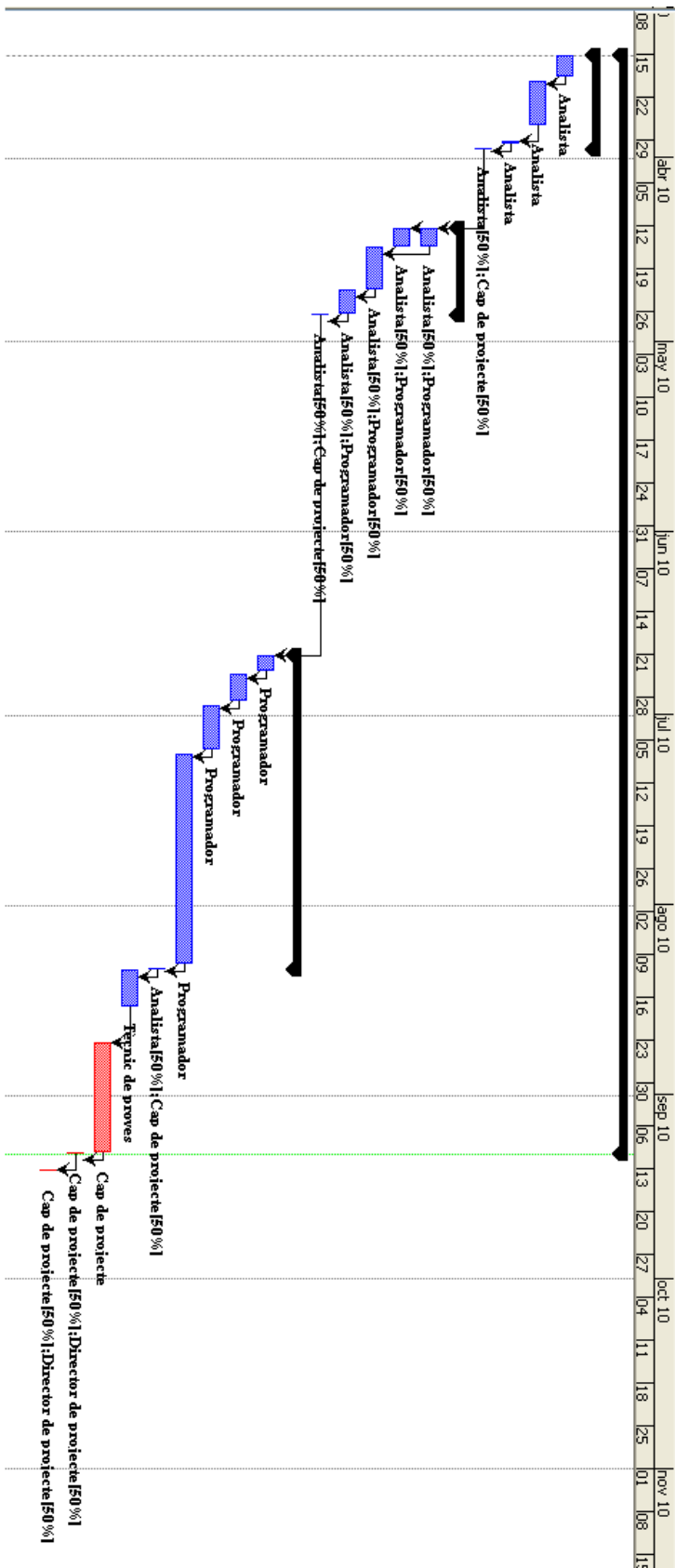


Figura 2.5

## 3. Marc teòric

### 3.1 Apache HTTP Server

Apache HTTP Server és un servidor web HTTP de codi obert per a tot tipus de plataformes (Unix, Microsoft, Macintosh, entre d'altres), que implementa el protocol HTTP/1.1 i la noció de lloc virtual.

És un software col·laboratiu robust, ple de funcionalitats i disponible gratuïtament que permet amb poc temps tenir un servidor web funcionant en qualsevol ordinador.

És utilitzat principalment per a enviar pàgines web estàtiques i dinàmiques a la World Wide Web. Moltes aplicacions web estan dissenyades amb característiques pròpies d'aquest servidor web.

Apache és utilitzat per a moltes tasques on el contingut necessita ser posat en marxa de forma segura i fiable. Un clar exemple seria el d'un usuari que comparteix arxius des del seu ordinador personal cap a Internet. Un usuari que té Apache instal·lat en el seu escriptori pot col·locar arbitràriament arxius a l'arrel de documents de Apache, des de on poden ser compartits.

Els programadors d'aplicacions web a vegades utilitzen la versió local del Apache amb la finalitat de poder previsualitzar i provar el codi mentre aquest es va desenvolupant (com en el cas d'aquest projecte).

Aquest software permet tenir varis *hosts* funcionant simultàniament, mitjançant *virtual hosts* configurables en l'arxiu de configuració d'apache "httpd.conf".

### 3.2 HTML (HyperText Markup Language)

HTML és el llenguatge de predominant per a l'elaboració de pàgines web. Aquest llenguatge no només en defineix l'estructura, sinó també la semàntica. Encara que a l'usuari se li pugui mostrar unes dades de forma idèntica utilitzant etiquetes diferents, cada etiqueta porta un concepte associat que ajuda a que usuaris no humans, puguin entendre'n el contingut i interpretar-lo degudament. Aquest llenguatge es basa en la definició d'etiquetes, escrites entre els símbols "<" i ">". Les etiquetes poden contenir diferents atributs per distingir-los, definir-ne el comportament, donar-li cert format... A més també poden contenir text i altres etiquetes, aquestes etiquetes per tal de saber on comencen i on acaben s'han d'escriure utilitzant una etiqueta d'obertura i una de tancament, quedant de la següent manera "<etiqueta>text</etiqueta>".

## Editors

Aquest llenguatge pot ser creat i editat en qualsevol editor de textos bàsic, com per exemple "bloc de notes" en Windows, "gedit" en Linux o qualsevol altre editor que admeti text sense format.

A part, existeixen altres editors més complexos per la creació de pàgines web amb característiques WYSIWYG (What You See Is What You Get). Aquest tipus d'editors permeten veure el resultat del que s'està editant en temps real a mesura que es va editant el document. Alguns exemples d'aquests editors serien: Macromedia Dreamweaver o Microsoft FrontPage. El principal problema d'aquests és que no acostumen a tenir en compte la semàntica web ni tampoc respecten la separació entre el contingut i la presentació.

Aquests editors estan pensats per separar l'estructura de la presentació, i tenir en compte la semàntica web per d'aquesta manera solucionar el problema dels anteriors. El problema que tenen aquests editors és que s'ha de tenir clara des d'un inici l'estructura que tindrà el document, i sovint tenen un llenguatge propi. Aquests problemes fan que requereixin un període d'adaptació i que a vegades resulti difícil d'utilitzar.

## La importància del marcat HTML en el posicionament en els buscadors

En els punts anteriors s'ha remarcat la importància d'una bona estructura HTML.

Encara que els buscadors no facilitin dades sobre el procés de registre dels llocs web, el que sí que és clar és que en més del 90 per cent dels casos aquest s'efectua de forma automàtica. Els encarregats d'aquesta tasca són programes anomenats *spiders*. Aquests programes són cecs a tot el contingut visual de la nostra pàgina (imatges, vídeos, Flash...), per tant en el que es fixen és en el contingut HTML i aquesta és una de les coses que tindrà en compte alhora d'indexar una pàgina web.

Per tant, és molt útil per tal de poder ésser localitzat per als buscadors utilitzar codi HTML.

## 3.3 CSS

CSS, que significa fulles d'estil en cascada, és un llenguatge utilitzat per definir la presentació d'una pàgina web. L'objectiu d'aquests llenguatges és la separació entre l'estructura i significat d'una pàgina web, de la seva aparença, fent així possible visualitzar una mateixa pàgina de diferents maneres, sense tenir-ne que modificar el codi HTML.

Aquesta separació entre el codi HTML i la presentació de la pàgina millora l'eficiència a l'hora de modificar qualsevol de les dues coses, ja que el codi està més ordenat.

CSS ens ofereix tres camins diferents per aplicar les regles d'estil a una pàgina web.

- **Fulla d'estil externa:** aquesta fulla està guardada en un arxiu diferent a l'arxiu on es guarda el codi HTML. Aquesta és la manera més òptima perquè separa completament el codi HTML de les regles que defineixen l'estil.
- **Fulla d'estil interna:** és una fulla d'estil que està incrustada dins del document HTML. En general, s'utilitza només si es vol proporcionar alguna característica a una pàgina web en un arxiu, per exemple, si s'està enviant alguna dada a la pàgina web.
- **Estil en línia:** és un mètode per inserir directament dins l'etiqueta HTML, el llenguatge d'estil de la pàgina. Aquesta no és la manera més adequada, ja que converteix en una tasca més complexa arreglar algun error en el codi, al ser aquest més llarg. Pot ser fet servir si es necessita fer algun retoc puntual.

### 3.4 Javascript

Javascript és un llenguatge de scripting, utilitzat per a accedir a objectes en l'aplicació. Principalment, s'utilitza integrat en un navegador web permeten el desenvolupament d'interfícies d'usuari millorades i pàgines web dinàmiques.

Javascript ha tingut influència de múltiples llenguatges i es va dissenyar amb una sintaxis similar al llenguatge de programació de Java, tot i que més fàcil de fer servir per a persones que no programen.

Tots els navegadors moderns interpreten el codi de Javascript, o sigui amb Internet Explorer 6 o anteriors no es podria a no ser que s'afegeixin unes certes condicions i instruccions en el codi que així permetin la seva visualització. Per a interactuar en la pàgina web es fa ús d'una implementació del llenguatge DOM per a Javascript.

Un inconvenient que pot tenir aquest llenguatge, és el fet de que es necessita escriure un codi bastant llarg per a algunes determinades accions, la qual cosa no passa amb el següent llenguatge que es tractarà, en el cinquè apartat d'aquest capítol.

### 3.5 jQuery

jQuery és una llibreria de JavaScript ràpida i concisa que simplifica el tractament d'elements HTML, tractament de events, animació i interaccions AJAX. jQuery no ofereix cap funcionalitat extra que no es pogués fer únicament amb JavaScript, però en simplifica molt l'ús. Tant és així que per exemple, una acció tant recurrent com pot ser una petició AJAX i posar el contingut en un element HTML, que en Javascript ocupa unes 10-15 línies com a mínim, en jQuery en pot resoldre en una sola línia utilitzant només dues funcions.

Una de les principals característiques que permet aquesta simplificació del codi és que qualsevol funció sempre en retorna l'objecte per d'aquesta manera poder encadenar varies funcions que es vulguin aplicar a un mateix element una rere l'altre.

jQuery doncs, consisteix en un únic fitxer Javascript que conté les funcionalitats comunes de DOM, events, efectes i AJAX.

### 3.6 MySQL

MySQL és un sistema de gestió de base de dades relacional amb llicència GNU que s'executa com a un servidor *multithread* i multi-usuari.

Una de les principals característiques de MySQL és que et permet escollir entre varis sistemes d'emmagatzemament, podent coexistir en una mateixa base de dades taules de més d'un sistema.

Existeixen diverses APIs, que no són més que un conjunt de trucades a certes biblioteques que ofereixen l'accés a certs serveis des dels seus processos com ara, funcions ja implementades i les quals estalvien al programador començar de zero; a aplicacions en diversos llenguatges de programació per a accedir a les bases de dades de MySQL. Entre ells C, C++, C#, Pascal, Delphi, Java, Perl, PHP o Python, entre d'altres. Com es deia abans, cada un té la seva API per connectar amb la base de dades. En java, per exemple, es pot fer d'una altre manera, agregant una llibreria dins del projecte, anomenada jdbc (Java DataBase Connectivity).

Els sistemes d'emmagatzemament són: MyISAM, InnoDB, Memory, Merge i Federated entre altres. Per poder escollir adequadament el sistema a utilitzar cal conèixer les principals característiques que puguin afectar al rendiment.

## **MyISAM**

- És el sistema que té el maneig d'auto increments més flexible i eficaç de tots els sistemes d'emmagatzemament.
- Suporta índexs FULL TEXT.
- Múltiples sentències SQL poden llegir simultàniament d'una taula, però al escriure cadascuna bloqueja la taula sencera mentre dura l'acció.
- No suporta transaccions.

## **Merge**

- Una taula MERGE es una col·lecció de taules MyISAM idènticament estructurades.
- Lògicament, una consulta en una taula MERGE actua com una consulta en cadascuna de les taules MyISAM que la formen.
- Són més lentes llegint índexs, ja que es tenen que buscar índexs de múltiples taules.

## **InnoDB**

- Suporta transaccions.
- Proporciona un sistema d'auto recuperació després d'una fallada del servidor MySQL o del host on s'està executant.
- Múltiples sentències SQL poden llegir simultàniament d'una taula, al escriure cadascuna bloqueja només la fila afectada mentre dura l'acció.
- Suporta claus externes i integritat referencial, incloent events en cascada, eliminació i actualització.

## **Memory**

- Té un rendiment molt ràpid.
- Els continguts de les taules Memory no perduren després d'una fallada o del reinici del sistema. Si que ho fa l'estructura.
- Al utilitzar memòria, no són adequades per grans continguts.
- No poden contenir camps de tipus TEXT ni BLOB.

## **Federated**

Aquest sistema d'emmagatzemament és un sistema nou, inclòs en la versió 5 de MySQL. Aquest sistema permet a MySQL server utilitzar taules d'altres servidors MySQL per ser utilitzades com si fossin del propi servidor. Una de les grans avantatges que aporta, es que en una sola consulta pots accedir a la

informació de varis servidors, però al ser un sistema nou, encara no està del tot ben optimitzat.

- No suporta transaccions.
- No s'efectua cap mena de blocatge a les taules.

### 3.7 PHP (PHP: Hypertext Processor)

#### Breu història introductòria

PHP és un llenguatge de *scripting*. La majoria de la seva sintaxi està treta d'altres llenguatges com C, Java i Perl. L'objectiu d'aquest llenguatge de programació es permetre als desenvolupadors web codificar ràpidament pàgines generades dinàmicament.

La primera versió de PHP va néixer l'any 1995, sota el nom de PHP/FI. Inicialment era un recull d'*scripts* de Perl, amb la finalitat de controlar els accessos a una web personal. L'autor d'aquest codi, que va anar augmentant la col·lecció de codi amb implementacions de C per connectar a base de dades per exemple, va fer públic el codi font perquè tothom el pogués utilitzar. Probablement aquest naixement tant personal i enfocat a una finalitat concreta sigui la que va marcar-ne el seu objectiu de facilitar la codificació de pàgines web, així com la seva poca consistència, en els inicis, com a llenguatge de programació.

L'any 1998 va aparèixer la versió 3, anomenada ja com a PHP, el seu punt més fort era la gran quantitat de compatibilitats amb bases de dades, protocols i API's, el que va provocar que molta gent s'apuntés a crear mòduls per a PHP. L'aparició de la versió 4 va suposar importants canvis. L'objectiu d'aquesta versió va ser millorar el rendiment de les aplicacions complexes i millorar-ne la modularitat del codi de PHP, ja que fins llavors no estava preparat per suportar extensions externes de manera eficaç. PHP5 va significar un canvi considerable al introduir un nou model de programació orientada a objectes, més estricta i eficaç tot mantenint el màxim possible la compatibilitat amb la versió anterior.

No és estrany llegir opinions negatives que diuen que PHP és un llenguatge poc rigorós, per inexperts... PHP des del seu naixement va ser creat per facilitar la programació de pàgines web dinàmiques, mentre que altres llenguatges com ara Java poden esdevenir molt feixucs d'aprendre per algú que mai no hagi programat, PHP és just el contrari, resulta molt senzill començar-lo a utilitzar a un nivell bàsic, només cal anar inserint etiquetes PHP que continguin



petits *scripts* enmig d'una pàgina HTML. A més, per aquest perfil de gent, PHP té un altre gran avantatge, i és que no cal declarar variables prèviament a utilitzar-les, ni definir-los cap tipus de dades, tot i que internament PHP adjudica un tipus de dades, que pot variar, a totes elles.

Aquest tipus de pàgines, a més a més, acostumen a tenir totes el gran inconvenient de no tenir nitidesa del codi, i de no ser escalables. PHP, però, permet estructures molt més complexes que no pas aquest tipus de pàgines, com per exemple des de la versió 5 una efectiva programació orientada a objectes que entre altres coses permet aplicar correctament el patró de disseny MVC (Model Vista Controlador). Aquest patró de disseny simplifica el desenvolupament i el manteniment del software al separar l'aplicació en tres components lògics:

- **Model:** aquesta capa és l'encarregada de la lògica de negoci i l'accés a les dades emmagatzemades. En definitiva es un conjunt de mètodes encapsulats en classes segons la seva naturalesa que en permeten la reutilització en qualsevol punt de l'aplicació.
- **Vista:** aquesta capa, en aplicacions web, seria el que es considera com a disseny web, és a dir l'encarregada de la sortida de l'aplicació, tot el que visualitzarà l'usuari.
- **Controlador:** aquesta és la capa principal, aglutina el flux d'execució del programa. és l'encarregada de recollir les dades d'usuari, comunicar-se amb els models i finalment donar la informació a la vista perquè la mostri a l'usuari.

## Framework

Els *frameworks* són un conjunt d'eines amb la finalitat de disminuir la càrrega dels desenvolupadors al aportar tot un seguit d'eines utilitzables en qualsevol tipus de projecte i independentment de la seva mida. Les primeres versions públiques i les més utilitzades de frameworks en PHP van ser CakePHP, Symfony i Zend Framework les quals van aparèixer l'any 2005.

A part de proporcionar als desenvolupadors eines per facilitar la programació d'una manera modular i configurable amb eines de filtratge, registre d'errors, arxius de configuració... també ajudar al programador a separar les diferents parts del codi en les tres capes corresponents a una arquitectura MVC i a que aquestes estiguin ben organitzades o a gestionar les direccions de les diferents pàgines de l'aplicació.

# 4. Disseny

## 4.1 UML: Casos d'ús

### Introducció

La finalitat en la qual es crea aquest apartat de la memòria del projecte, és definir les funcionalitats del sistema (casos d'ús) i el context d'aquest (la interacció existent entre entitats externes o els actors amb la pàgina web).

A continuació es presentaran els diferents diagrames de casos d'ús per als actors, a més de descriure un resum de les seves accions dins de cada cas d'ús.

### Actors

N'hi haurà quatre tipus:

- **Usuari no identificat (visitant):** aquest actor podrà veure el contingut públic de la web, però per accedir a la intranet (Zona Socis) necessitarà registrar-se.
- **Usuari identificat:** aquest actor, a més de poder fer el mateix que l'usuari no identificat, podrà accedir al contingut de la intranet.
- **Usuari autor:** usuari igual que l'usuari identificat, la diferència serà que aquest haurà participat, ja sigui actuant en el fòrum, el taulell d'anuncis o fent alguna reserva de pista.
- **Servidor:** seran accions que el servidor realitzarà de forma automàtica.

### Casos d'ús

El nombre de casos d'ús serà d'un total d'onze:

- **Donar-se d'alta**

L'usuari inserirà les dades que se li demanen per tal de poder visitar i/o participar en la Zona Socis.

- **Iniciar sessió**

L'usuari haurà de identificar-se i així podrà accedir a la Zona Socis.

- **Finalitzar sessió**

L'usuari podrà donar per finalitzada la seva sessió i per a poder tornar a la Zona Socis necessitarà començar una nova sessió.

- **Visualitzar contingut públic**

L'usuari estigui identificat o no, podrà navegar pel contingut públic de la web. Això es podrà fer amb el menú desplegable, el qual permetrà accedir a les diferents pàgines de la web.

- **Enviar consulta al club**

Qualsevol usuari, sigui un usuari registrat o no, podrà posar-se en contacte de forma virtual amb el Club, a través de la pàgina de contacte.

- **Llegir del fòrum o del taulell d'anuncis**

Cas d'ús que serà molt freqüent entre els usuaris identificats, per tal de poder saber de quins temes han escrit els usuaris. Es podrà veure en quina data es va escriure el missatge, qui ho va fer, les respostes que té i qui li ha respòs, a més de veure en la pàgina principal del fòrum, qui és l'últim que usuari que ha respòs.

- **Escriure un missatge en el fòrum o en el taulell d'anuncis**

En aquest cas d'ús un usuari podrà escriure un nou missatge, guardant-se la el seu nom d'usuari, el títol del missatge, el missatge que hagi escrit i la data del moment en que l'ha escrit.

- **Respondre un missatge en el fòrum o en el taulell d'anuncis**

Aquest cas d'ús permetrà continuar amb un tema iniciat per algun altre usuari, o el mateix usuari. Es farà citant aquell missatge i escrivint el missatge corresponent. Es guardaran el nom d'usuari de qui respongui, a més de la data, la incrementació en una unitat de les respostes del missatge original, i la data de la última resposta.

- **Consulta reserves de les pistes**

En aquest cas d'ús l'usuari ja haurà elegit la pista en la qual vol fer aquesta consulta. Llavors podrà veure si hi ha alguna reserva en la present setmana.

- **Reservar pista**

L'usuari ja haurà vist les reserves realitzades i podrà decidir fer una reserva. Per això, seleccionarà el dia de la setmana i la hora per tal de fer una reserva per ell.

- **Visualitzar la galeria de fotografies**

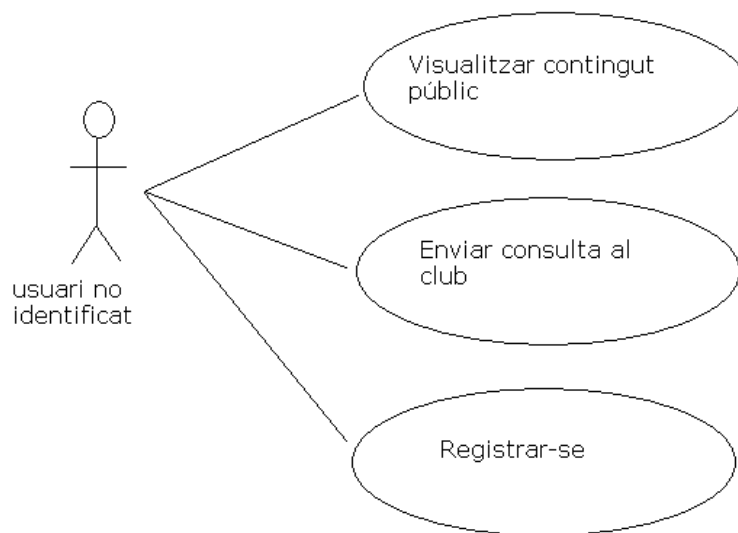
El darrer cas d'ús, permetrà als usuaris veure les diferents galeries de fotografies. A més, fent clic sobre la portada de cada galeria, aquesta es farà més gran i així podrà navegar per aquest àlbum avançant o retrocedint les fotografies i movent-les de posició.

## Vistes dels casos d'ús

A continuació es mostren les diferents vistes funcionals del sistema.

### Situació del visitant o usuari no identificat

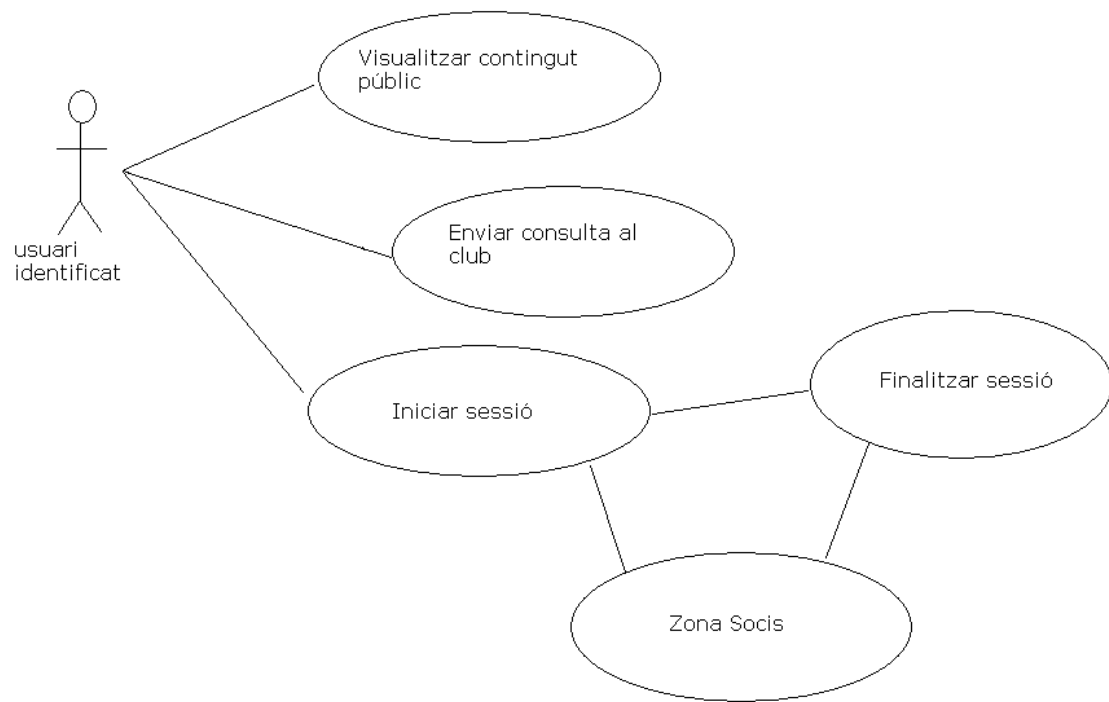
En la figura 4.1 veiem el diagrama de casos d'ús que fan referència a les accions que podrà realitzar un usuari que no estigui identificat.



**Figura 4.1**

### Situació de l'usuari que s'identifica

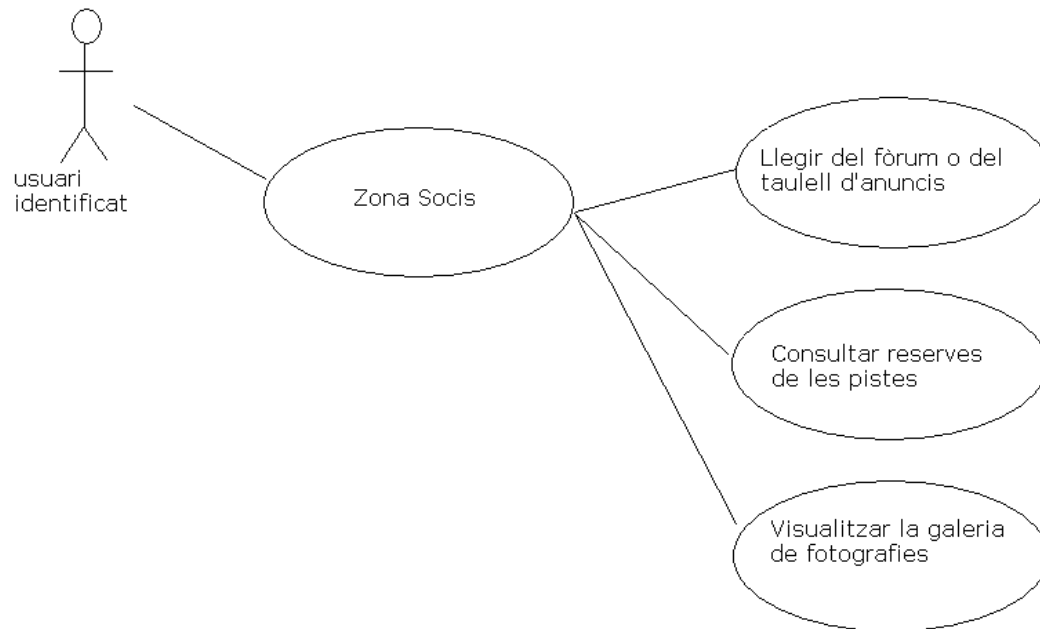
En la figura 4.2 podem veure el diagrama de casos d'ús d'un usuari que s'identifica amb el seu nom d'usuari i el seu password.



**Figura 4.2**

## Situació de l'usuari identificat en la intranet

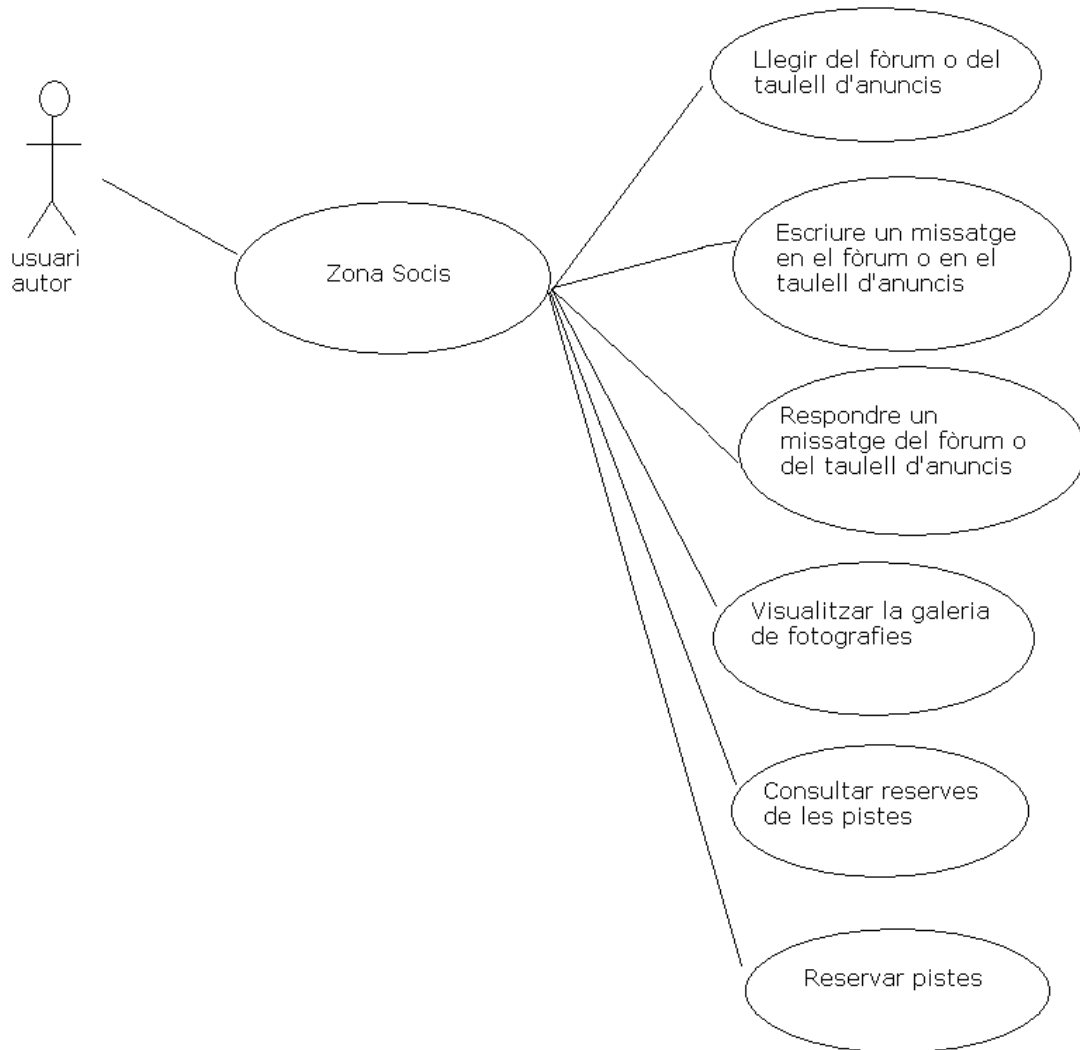
En la figura 4.3 podem veure el diagrama de classes d'ús d'un usuari ja identificat interactuant dins de la intranet.



**Figura 4.3**

## Situació de l'usuari autor en la intranet

En aquest diagrama de casos d'ús, de la figura 4.4 se'ns mostra totes les accions que podrà realitzar un usuari identificat en la Zona Socis. En aquest cas l'usuari serà actor, o sigui participarà.



**Figura 4.4**



## 4.2 Arquitectura del sistema

Aquest és un projecte iniciat des de zero, amb el qual l'anàlisi del mateix és un pas molt important per tal desenvolupar-lo.

### Part estàtica de la web

Al ser contingut estàtic, el llenguatge base serà el HTML.

Es crearan diferents pàgines per mostrar el contingut requerit pel client. Com que és un nou club, el qual ha canviat la imatge i s'ha modernitzat, es vol mostrar tal amb la pàgina web i així es busca que compleixi algunes de les seves característiques. Es vol que sigui per tothom i fàcilment accessible.

Per a fer-ho de fàcil accés s'ha decidit crear un menú principal, el qual serà desplegable quan hi passem el ratolí per sobre. Ens apareixeran les diferents opcions.

Aquest menú està fet sobre una llista, amb les seves respectives subllistes. Les llistes es creen amb els tags `<li>` per obrir-la i `</li>` per tancar-la. Després es crearan les subllistes amb els tags `<ul>` per obrir-la i `</ul>` per tancar-la.

En aquest tros de codi podem veure un fragment del menú creat amb els tags `<li>` i `<ul>`.

```
<div id="menu">

  <ul id="nav">
    <li><a href="inici.html">Inici &nbsp;</a></li>
    <li><a href="#">&nbsp;</a>El Club&nbsp;</a>
      <ul class="submenu">
        <li><a href="Presentacio.html">Presentació</a></li>
        <li><a href="Contacte.html">Contacte</a></li>
        <li><a href="Historia.html">Història</a></li>
        <li><a href="Horaris.html">Horaris</a></li>
        <li><a href="Quotes.html">Quotes</a></li>
      </ul>
    </li>
  </ul>
```

Les etiquetes <a href ...> ens a serveixen per a indicar a quina pàgina enllacen.

Les etiquetes <div> ens indiquen com es diu la part de la pàgina en la qual se li aplicarà el CSS. En la fulla d'estil de CSS estaran descrites les característiques que se li donarà a la presentació de la web.

A continuació un tros de codi de CSS, que és el que fa referència al tros de codi anterior.

```
#menu {  
  
top:25%;  
background-image:url(nav_bg.png);  
background-repeat:repeat-x;  
height:30px;  
width:868px;  
margin:auto;  
padding-left:70px;  
  
}
```

Per tal de que la pàgina HTML sàpiga anar a buscar l'arxiu de CSS, farà falta posar una sentència. Caldrà situar-la entre el <head> i el </head>.

```
<script type="text/javascript" src="menu/menu.js"></script>
```

Llavors ja es tindria el menú fet, però faltaria alguna cosa més per tal que fos desplegable al passar-hi el ratolí per sobre. Per això es farà servir el llenguatge de Javascript. En aquest cas s'ha fet servir una llibreria de jQuery, per tal de no haver de posar tantes línies de codi, a més de ser un codi eficient i molt professional.

A continuació el codi que s'ha fet servir pel menú.

```
function mainmenu(){  
$(" #nav ul ").css({display: "none"});  
$(" #nav li").hover(function(){  
    $(this).find('ul:first:hidden').css({visibility:  
"visible",display: "none"}).slideDown(400);
```

```

    },function(){
        $(this).find('ul:first').slideUp(400);
    });
}
$(document).ready(function(){
    mainmenu();
});

```

Com es pot veure és un codi ben curt, ja que amb Javascript s'haurien d'haver escrit moltes línies de codi.

A més, el jQuery també té la seva part de CSS per a determinar el fons del menú, el tamany de la lletra i el color, etc.

Per a fer ús del jQuery, el qual recordem és una llibreria que extén de Javascript, el que hem de fer és incorporar-la amb el nostre projecte i cridar-la mitjançant la següent sentència.

```
<script language="javascript" src="jquery-1.3.min.js"></script>
```

## Part dinàmica de la web

El llenguatge principal en aquesta part serà el PHP, per tal d'interactuar amb la base de dades, de llenguatge MySQL. També hi hauran els llenguatges emprats en la part estàtica.

Per tal de realitzar-la, una cosa molt important és la de controlar l'accés a la part dinàmica o d'intranet.

Com se sap només si podrà accedint a través d'un inici de sessió, el qual es farà fent entrant les dades a l'usuari en un formulari.

Si és la primera vegada que accedeix s'haurà de registrar per a donar-se d'alta. Això generarà una consulta d'inserció a la taula socis.

```
INSERT INTO `clubbellaterra`.`socis` (`id`, `nom`, `cognoms`, `nick`,
`password`, `numeroSoci`, `email` )
```

```
VALUES (
NULL, 'jordi', 'garcia garcia', 'jordi20', MD5( 'jordi' ), '55', 'jordi20@gmail.com'
);
```

Cal destacar la funció MD5, la qual permet assegurar la contrasenya, així els administradors no podran accedir-hi.

MD5 és un algoritme de reducció criptogràfica de 128 bits molt utilitzat. És molt utilitzat ja que és ràpid, fàcil i potent, i a més garanteix una certa seguretat de no veure la contrasenya interceptada.

Si ja està registrat només necessitarà identificar-se introduint el seu nom d'usuari i el password. Llavors es generaria una consulta a la base de dades per tal de comprovar si les dades entrades al formulari coincideixen amb algun usuari de la taula socis.

Aquí una cosa important, si realment existeix l'usuari, caldrà guardar les variables, de manera que durant tota la sessió de l'usuari estiguin actives i així poder accedir a tot el contingut protegit de la part dinàmica.

El codi següent crea les variables de sessió

```
$_SESSION["uid"] = $_REQUEST["nick"];  
$_SESSION["pwd"] = md5($_REQUEST["pass"]);
```

Llavors farem ús de les variables següents per fer una consulta a la base de dades.

```
$uid = $_SESSION["uid"];  
$pwd = $_SESSION["pwd"];
```

La següent línia de codi permetrà mantenir la sessió iniciada a totes les pàgines.

```
<?php session_start(); // obre o inicia la sessió  
?>
```

En aquest moment l'usuari ja està identificat i podrà accedir al contingut de la intranet.

## Galeria de fotografies

Les fotografies contingudes en diferents àlbums no faran ús de connexió a base de dades, degut a que si a la base de dades s'hi carreguen fotografies, amb un pes important, o moltes fotografies, llavors faria que anés lenta la connexió.

Per tant, per aconseguir veure la galeria i que a més, tingui alguns efectes amb més vistositat, es farà ús de Javascript.

Un problema alhora d'implementar la segona funció en Javascript, recordem que el menú també ho és, era que en una mateixa pàgina només hi pot haver una funció de Javascript.

La solució ha estat crear un iframe, el qual és una pàgina que es crida per a carregar-se en una part de la pàgina principal.

Amb això ja no hi hauria el problema anterior.

Aquesta és la sentència que ens permet fer el que s'ha mencionat anteriorment.

```
<iframe src="gallery_img.html" width="100%" height="900px">
```

Un cop a la pàgina que contindrà el Javascript i la galeria, en mostrem una part de codi en la qual es veu la ruta per a carregar la imatge, el títol i una funció que es farà al fer clic sobre la fotografia.

Aquesta serà `hs.expand()`, la qual farà ampliar el tamany de la fotografia.

```
<a id="thumb1" href="highslide/images/large/770083.jpg"
class="highslide"
      title="Fotografia 1"
      onclick="return hs.expand(this, config1 )">
  
</a>
```

Una altra cosa important és que quan es fa clic a una fotografia ens apareixeran unes petites icones. Seran les que permetran avançar i retrocedir entre les fotografies, tancar-la o moure-la.

El següent és un fragment de codi que mostra com retrocedir a la anterior

fotografia i també fer *play*.

```
'<div class="highslide-controls"><ul>' +
    '<li class="highslide-previous">' +
        '<a href="#"
title="{hs.lang.previousTitle}">' +
            '<span>{hs.lang.previousText}</span></a>' +
        '</li>' +
        '<li class="highslide-play">' +
            '<a href="#"
title="{hs.lang.playTitle}">' +
                '<span>{hs.lang.playText}</span></a>' +
            '</li>' +
        '<li class="highslide-pause">' +
            '<a href="#"
```

Aquesta pàgina també tindrà la seva pròpia fulla d'estils de CSS.

## **Fòrum i taulell d'anuncis**

Les pàgines que ens mostraran el fòrum i el taulell d'anuncis seran molt semblants. El funcionament de les dues serà el mateix. Escriure un missatge, veure tots els que hi ha escrits, veure qui els ha escrit, etc. Però la diferència entre les dos seran dues.

Primera, el contingut del fòrum serà, en general, per debatre temes relacionats amb el Club, i el taulell d'anuncis per a fer anuncis de compra venda, per exemple.

La segona, és la que tindrà a veure amb la programació, ja que aniran a dues taules diferents segons a la pàgina on ens trobem.

La pàgina principal ens mostrarà una taula amb el títol dels missatges que hi hagin, l'autor i la data del missatge, el número de respostes i la data en què es va publicar la última resposta al missatge.

Per a veure això cal fer una consulta de la taula del fòrum o del taulell d'anuncis, per a veure tot els registres guardats a la base de dades.

Des de la pàgina d'inici es podrà anar a escriure un nou missatge i a llegir els que ja hi ha.

Si es desitja escriure un nou missatge s'omplirà el formulari amb el títol del

missatge i el contingut. L'usuari ja estarà identificat amb la gràcies a la variable SESSION.

Llavors s'introduirà en la base de dades el missatge. Es comprovarà que en el text del missatge no s'ha inclòs codi en HTML, mitjançant htmlentities, funció que ho controla. Aquí es veu com s'utilitza.

```
$missatge = htmlentities($missatge);
```

Si no s'ha posat títol es gravarà per defecte un "Sense títol".

En canvi, si es desitja llegir els missatges, fent clic al títol hi accedirem al agafar la id del missatge que hem seleccionat. Podrem llegir el missatge i les seves respostes, fent una consulta per obtenir la id, el títol, l'autor, el missatge i la data.

Llavors si no ens retorna un valor nul, mirant el nombre de registres que ens retorna.

```
if(mysql_num_rows($rs)>0)
```

Després amb el resultat de cada registre es recorrerà un bucle per tractar cada registre obtingut en la consulta anterior.

```
while($row = mysql_fetch_assoc($rs))
```

Alhora de mostrar les variables, es farà ús de la funció parsearTags, per parsejar el que són lletres de símbols (els claudàtors de citar), i la funció de HTML nl2br.

Aquesta funció es fa servir ja que HTML no entén els salts de línia (Enters), els quals es fan amb el tag <br>. El que fa nl2br és reemplaçar els salts de línia normals pel tag <br>. Així s'aconsegueix amb menys línies de codi, veure el resultat de forma més clara i entenedora.

Llavors si es cita el missatge escriurem el missatge agafant com a referència la id del missatge al que respondrem.

## Reserva de pistes

La pàgina de reserves de pista servirà per veure les reserves que hi ha fetes i per reservar una hora de les que hi hagi disponibles.

Primerament, haurem de seleccionar una de les quatre pistes (tenis, padel1, padel2, padel3).

Seguidament, anirem a una altra pàgina on se'ns mostraran les reserves que s'han realitzat per aquesta setmana. Per a obtenir la setmana actual, s'utilitza la funció `mktime()`; la qual ens retornarà la data. Passant-li el mètode `date()`, la obtindrem en el format de data `YYYY-MM-DD`.

Així i amb un bucle que anirà controlant el temps fins a una setmana obtenim el primer i l'últim de la setmana.

Aquí un fragment del codi com a exemple.

```
$primer_dia = mktime();
$ultim_dia = mktime();
while(date("w", $primer_dia) != 1) {
    $primer_dia -= 3600;
}
while(date("w", $ultim_dia) != 0) {
    $ultim_dia += 3600;
}

$dilluns=date("Y-m-d", $primer_dia);
```

El següent pas serà fer una consulta entre aquests dos dies (el dilluns i el diumenge), per tal d'obtenir les dades que ens interessin per mostrar, el dia i la hora que tenen reserves.

Si l'usuari desitja fer una reserva no té més que seleccionar el dia i hora dels camps desplegable, i aquesta reserva es farà efectiva.

Es faran dues comprovacions en la base de dades:

- La primera una clau única a pista, data i hora, per tal d'assegurar fer una inserció a la base de dades amb les tres camps emplenats i cap de buit.
- La segona, també una clau única de soci, dia i hora, per tal d'assegurar-nos que cap usuari reserva una mateixa hora i dia, ja que això seria incongruent.



### 4.3 Base de dades

La base de dades del Club Bellaterra, es presenta com un conjunt de cinc taules, de les quals quatre estaran indexades pel mateix camp, el camp id de la taula socis, que farà de referència per saber quin usuari realitza alguna acció que quedi gravada en els registres de la base de dades.

La taula de socis contindrà la informació amb la qual un soci es registre i se li demana en el formulari.

La taula de notícies contindrà nova informació que es mostrarà a la pàgina inicial de la web, la qual serà pública.

La taula de reserves ens guardarà les hores i les dates de les reserves que es realitzin en alguna de les quatre pistes, que estaran declarades com a un camp 'enum' = tenis, padel1, padel2, padel3. Es creen unes comprovacions, abans ja explicades en el punt 4.2, UNIQUE les quals no permetran errors al inserir dades en la base de dades.

La taula del fòrum guardarà els missatges escrits, amb el seu autor, títol, respostes, identificador, i data del missatge i de la última resposta. Els camps respostes i identificador comencen a 0 per a cada registre. El de respostes es va incrementant en una unitat per a cada resposta. El de identificador prendrà el valor de la id del missatge al qual ens referenciem per a escriure-li una resposta.

La taula del taulell d'anuncis tindrà els mateixos camps que la del fòrum.

A continuació una gràfica de la taula en la figura 4.5

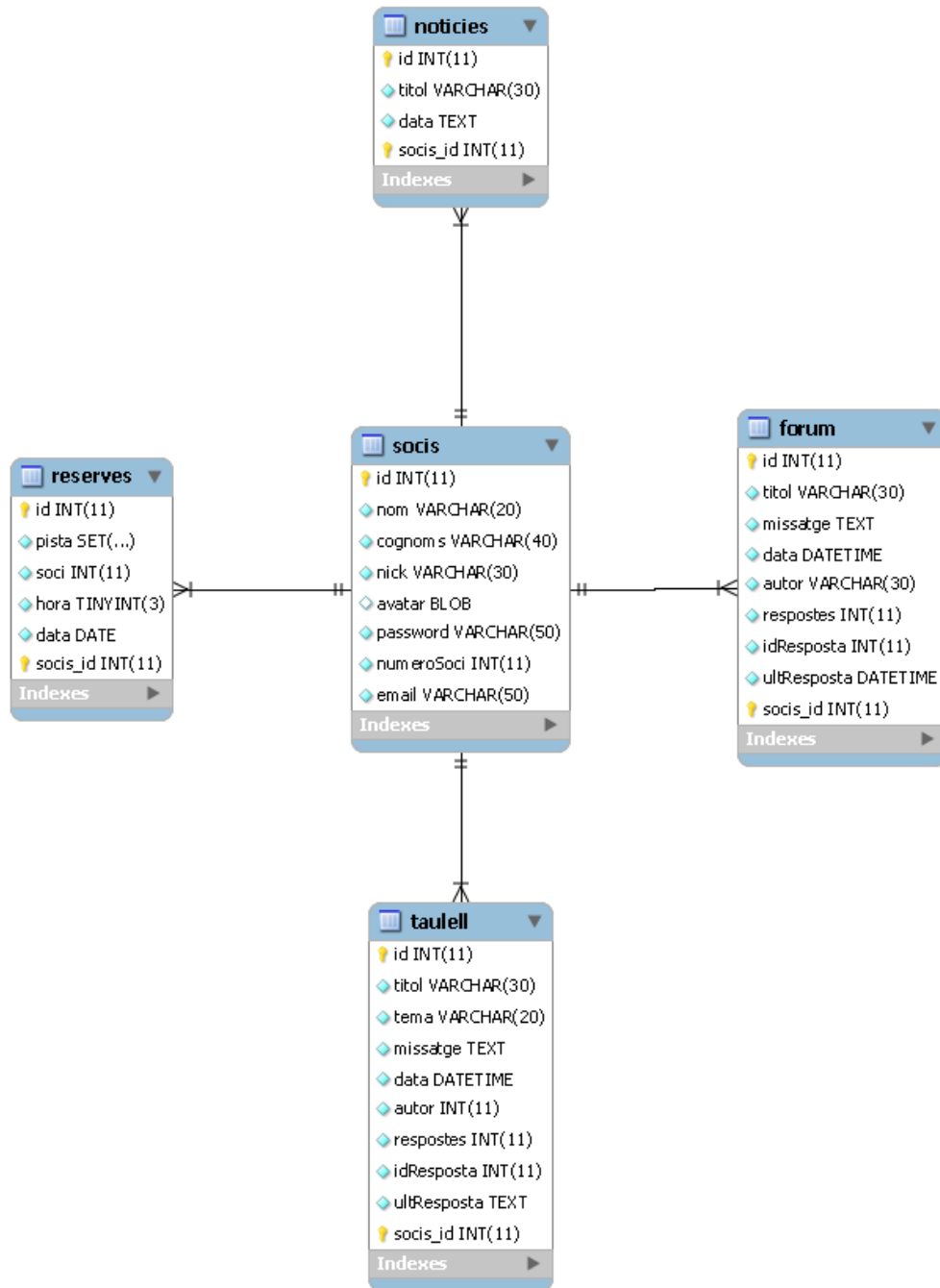


Figura 4.5

#### 4.4 Documentació sobre el disseny

El Club Bellaterra ja tenia planejada una estratègia de disseny per dur a terme alhora de remodelar el club.

Es van dissenyar algunes directrius per identificar el club.

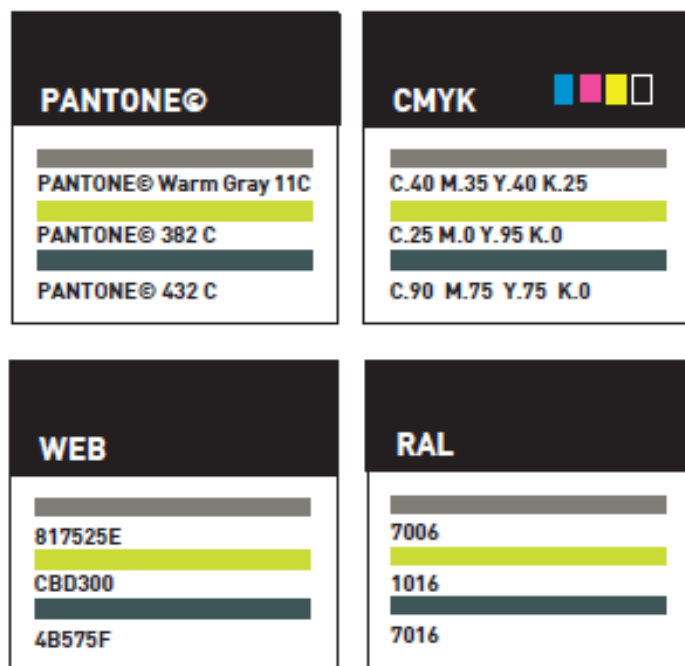
En la figura 4.6 es mostra el nou logotip del club



**Figura 4.6**

Els colors també es vol que siguin identificatius.

En la figura 4.7 es mostren aquests colors.



**Figura 4.7**

Per això s'han creat aquestes taules amb les referències per a usar aquests colors en la web. No es pretén que només es facin servir aquests colors, sinó de que apareguin en algunes pàgines pel fet de resultar familiars al visitar la web.

El logotip ja fa ús d'aquests colors i ja comença a fer-se familiar al utilitzar la web, ja que forma part de la capçalera de la web en totes les pàgines, juntament amb una imatge del club.

En la figura 4.8 es mostra la capçalera.

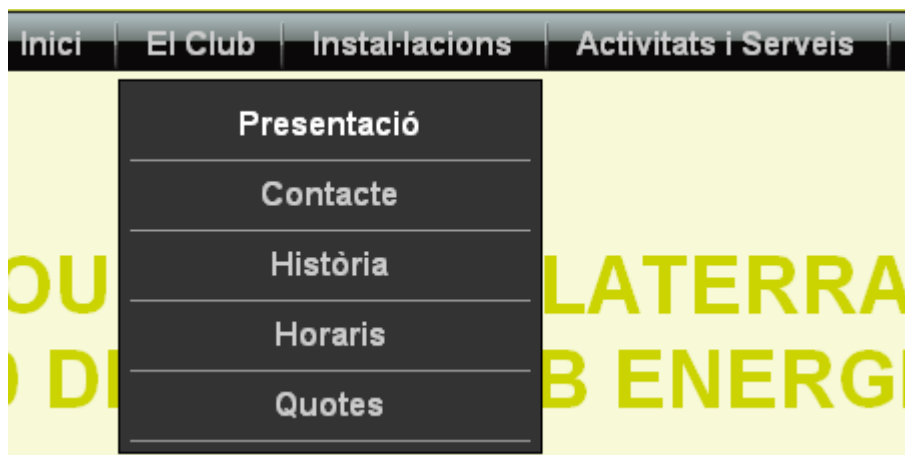


**Figura 4.8**

En la figura 4.9 es mostra una part del menú principal i en la figura 4.10 un fragment d'aquest menú desplegat.

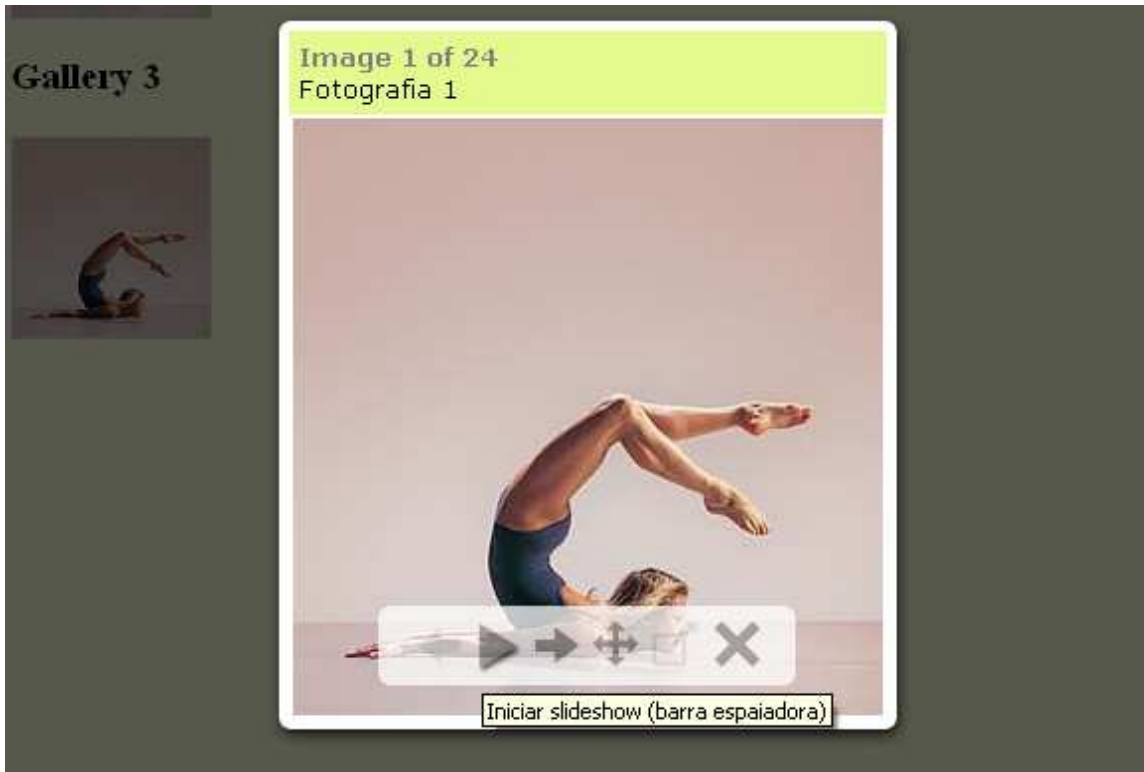


**Figura 4.9**



**Figura 4.10**

En la figura 4.11 es mostra una imatge de la galeria de fotografies



**Figura 4.11**

## 5. Proves

Al ser aquesta pàgina web pública i la possibilitat de crear-hi una conta, és susceptible a rebre usuaris malintencionats, per això es faran algunes comprovacions per tal de no tenir problemes.

Les proves sobretot es troben en els formularis, on s'envia informació a la base de dades.

Primerament, cal realitzar proves per comprovar el funcionament correcte de l'aplicació com ara les que s'han mencionat anteriorment. Htmlelities per evitar codi HTML, md5 per assegurar i encriptar les contrasenyes, les comprovacions UNIQUE per tal de passar les dades que es necessiten per a un registre, etc.

Llavors cal conèixer una mica alguns altres tipus d'atacs que pot patir la web, per si aquesta responia correctament.

S'ha provat el SQL injection, comprovant que no s'introdueixi codi SQL en les consultes realitzades al servidor a la base de dades. S'han hagut de provar totes les entrades.

També la fuga d'informació, ja que petits errors programant en PHP poden fer que hi hagi un error de seguretat i es mostrin rutes d'arxius i a vegades també petites parts del codi.

Una altra prova realitzada és la de Cross Site Scripting (XSS), la qual és prevenir l'atac de introduir codi de script, javascript per exemple, en la pàgina web. Per a controlar i detectar aquestes possibles entrades de codi de script, s'ha fet servir la següent funció:

```
function Seguretat1(Comentaris)
{
    var regexp = /<script || script>/;
    if ((regexp.test(Comentaris))){
        alert("Ha introduït codi potencialment perillós, se'l
```

```

redireccionara.");
        location.href='ZonaSocis.php';
    }
}

```

També s'ha comprovat que les camps que emplenava l'usuari per a identificar-se es feien correctament i de manera raonable, introduint només caràcters alfanumèrics.

```

function ValidarNom(Nom)
{
    if (Nom.length>30 || Nom.length<3)
    {
        alert("NOM: Ha de tenir menys de 30 caracters i més de 3.
En té " + Nom.length+ ".");
        return(true);
    }

    var regexp=/^\w+$</;
    if (!(regexp.test(nom)))
    {
        alert("NOM: Nomes permet caracters alfanumerics,
caracters introduits incorrectes.");
    }else{
        //alert("NOM: Correcte, el nom es acceptat");
    }
}

```



# 6. Conclusions

## 6.1 Assoliment d'objectius

El fet de saber si s'han assolit els objectius d'aquest projecte, resulta un aspecte certament difícil de valorar en primera instància.

Cal tenir en compte que encara no s'ha posat en marxa la pàgina web, un cop això sigui una realitat es podrà saber millor si s'han assolit els objectius.

El que si es té present és que s'han seguit els requisits establerts pel client i s'espera que un cop tancat el projecte, aquest s'implementi en el servidor del Club Bellaterra i els usuaris del club en puguin fer ús i també ho sàpiguen fer correctament.

## 6.2 Línies d'ampliació

Hi ha diverses línies d'ampliació per aquest projecte, que en cas de ser requerit es podria estudiar de cara a un futur a curt o mig termini.

Es podria tenir la possibilitat d'afegir un enllaç a la part dinàmica de la web a les webcams del club. Això permetria que els socis poguessin veure si fa un bon clima per a la practica de l'esport a les pistes que estan a l'aire lliure.

Una altra idea seria la de que els usuaris poguessin enviar arxius fotogràfics a les galeries de fotografies. Aquesta és una proposta però, que no es va tenir en compte en cap moment per a realitzar-la ja que primer es vol esperar a observar com funciona l'entorn web i de la intranet.

La última idea per a ampliar, seria crear una botiga virtual, on els usuaris poguessin fer el típic "carro de la compra" amb els productes esportius que el club té a la venda pel públic. Aquesta via s'havia pensat en un inici, tot i que per motius de seguretat el client del club va decidir no implementar-la d'inici i seguir

fent la venda de la botiga física en les instal·lacions del club. Si fos oportú també es podria estudiar de implementar-ho.

# 7. Bibliografia

## Fonts electròniques - Pàgines webs de referència

- Wikipedia. Enllaç a: <http://en.wikipedia.org>
- PHP Manual. Enllaç a: <http://www.php.net>
- MySQL Reference Manual. Enllaç a: <http://dev.mysql.com/doc/refman/5.0/es/index.html>
- Apache Software Foundation. Enllaç a: <http://www.apache.org/>
- Javascript DOM. Enllaç a: <http://krook.org/jsdom/>
- jQuery API. Enllaç a: <http://api.jquery.com/>
- HTML w3schools.com Tutorial. Enllaç a: <http://www.w3schools.com/html/>
- Club Ciudad Diagonal. Enllaç a: <http://www.clubciudaddiagonal.com/>
- Foros del web. Enllaç a: <http://www.forosdelweb.com/>