

# Aplicació web en xarxa per a gestió i coordinació d'empresa de lleure

Memòria del projecte  
d'Enginyeria en Informàtica  
realitzat per

XAVIER SANGLAS MOLERA

i dirigit per

ORIOL RAMOS TERRADES

Bellaterra, 30 de juny de  
2014







El sotasignat, Oriol Ramos Tenades  
professor/a de l'Escola d'Enginyeria de la UAB,

**CERTIFICA:**

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en/na Xavier Sangla Molera

I per a que consti firma la present.

Signat:  .....



El sotasignat, Anna Cots Feixas  
de l'empresa, Actua Serveis Scp.

CERTIFICA

Que el treball a què correspon aquesta memòria ha estat realitzat en l'empresa sota la seva supervisió mitjançant conveni DE COOPERACIÓ EDUCATIVA

firmat amb la Universitat Autònoma de Barcelona.

Així mateix, l'empresa en té coneixement i dóna el vist-i-plau al contingut que es detalla en aquesta memòria.

Signat: Anna Cots Feixas



centells, 17 de juny del 2004.



NIF: J-65.758.245





## AGRAÏMENTS

En primer lloc, vull agrair al meu director de projecte Oriol Ramos, tot el seu suport i dedicació al llarg d'aquests mesos. Des del principi, per accedir a dirigir el meu projecte i, sobretot, durant la realització del mateix, per els seus consells, el seu recolzament i també la seva infinita paciència. Sense la seva contribució m'hauria set totalment impossible realitzar aquest projecte.

En segon lloc, a l'empresa ACTUASERVEIS SCL, més concretament, als seus responsables Anna i Dani. Gràcies per a donar-me la idea inicial que ha desembocat en la creació d'aquest projecte. Agrair el tracte i les facilitats que sempre m'han ofert quan els he necessitat.

També vull agrair als meus pares Amadeu i Silvia per tots els esforços i sacrificis que han fet perquè jo hagi arribat fins aquí i que mai els hi sabré agrair suficientment. També per educar-me en la cultura de l'esforç i el treball. La lluita per aconseguir allò que em proposi. Per no permetre'm llençar la tovallola en els moments més difícils i animar-me a tirar endavant quan més dur semblava. De veritat, gràcies.

A la meva germana Silvia, una incansable lluitadora de la vida a la que mai li recordo l'increïblement important que és per mi. A pesar de les dificultats, ni s'imagina la de vegades que la miro com a l'exemple a seguir ni la força que em dóna veure-la riure cada dia.

Agrair també a tots el meus familiars i amics. Malgrat els molts moments importants on no he pogut estar present, han seguit al meu costat allargant la mà quan ho he necessitat o arrencant-me aquell somriure quan tan em feia falta. No vull escriure cap dels seus noms perquè, segurament, em deixaria algú. A tots, gràcies.

Finalment, m'agradaria tenir un moment de record per tots aquells que he perdut pel camí i que, des d'algun lloc, segur que em veuen. Que sapiguen que no us oblidó.



# TAULA DE CONTINGUTS

<b>1. Introducció .....</b>	<b>1</b>
1.1. Discapacitat i societat .....	1
1.2. ACTUASERVEIS SCP.....	2
1.3. Objectius i Motivació.....	4
1.4. Estructura del document.....	5
<b>2. Anàlisi de requeriments .....</b>	<b>7</b>
2.1. Perfil d'usuaris.....	7
2.2. Requeriments Funcionals i no Funcionals.....	8
2.3. Anàlisi de riscos .....	10
2.4. Planificació de Tasques .....	12
2.5. Pressupost .....	15
<b>3. Estat de la qüestió .....</b>	<b>17</b>
3.1. Arquitectures d'aplicació web.....	17
3.2. Tecnologies Web i Bases de dades.....	19
3.3. Frameworks per a Aplicacions Web .....	25
3.3.1. El Framework CakePHP .....	26
3.4. El paquet XAMPP.....	27
3.4.1. Descripció .....	27
3.4.2. Funcionament de CakePHP dins XAMPP.....	29
<b>4. Descripció de l'aplicació .....</b>	<b>33</b>
4.1. Descripcions prèvies.....	34
4.2. Diagrama Casos d'Ús.....	34
4.2.1 Casos d'ús del subsistema CRUD.....	36
4.2.2. Casos d'ús del subsistema Control de serveis.....	39
4.3. Diagrama de Classes.....	45

<b>5. Disseny de l'aplicació .....</b>	<b>49</b>
5.1. Diagrama Entitat-Relació.....	49
5.2. Model Relacional de Taules .....	55
5.3. Vistes de l'aplicació .....	59
<b>6. Implementació.....</b>	<b>71</b>
6.1. Programari instal·lat.....	71
6.2. Creació de la base de dades i taules a través de phpMyAdmin.....	72
6.3. Creació dels índexs de restricció de les taules .....	73
6.4. Creació dels directoris necessaris .....	74
6.5. Creació dels models en CakePHP .....	75
6.6 Creació dels controladors i les vistes en CakePHP .....	76
6.7. Creació d'un sistema d'autenticació amb nom d'usuari i contrasenya .....	77
<b>7. Tests i Avaluació .....</b>	<b>81</b>
<b>8. Conclusions .....</b>	<b>85</b>
<b>Referències .....</b>	<b>87</b>
<b>Índex de figures.....</b>	<b>89</b>





# 1. Introducció

## 1.1. Discapacitat i societat

El col·lectiu social de persones amb discapacitat tan física com intel·lectual o sensorial, han obtingut poca atenció per part de les institucions i grans empreses, més ara en el context social i econòmic en el que ens trobem.

Tot i això, sempre hi ha hagut moviments que, des de fa temps, treballen per aconseguir la total integració d'aquestes persones. Una integració que va des de l'adequació d'elements quotidians com pot ser un simple llibre o pel·lícula, passant per l'habilitació de l'espai urbà com poden ser les rampes d'accés a edificis i, acabant amb la pròpia acceptació d'una societat sovint exclouent amb les minories.

Només cal anomenar el moviment paralímpic o la ONCE (Organització Nacional dels Cecs d'Espanya) com a coneguts paradigmes d'institucions que vetllen per aconseguir que no només els discapacitats s'integrin en el funcionament natural de la societat, sinó també per acostar a la societat a aquestes realitats existents en el dia a dia de moltes persones.

També cal prestar atenció a organitzacions i fundacions d'àmbit més local, que treballen de manera anònima per oferir tot tipus de serveis pensats per a persones amb necessitats especials.

Anem a veure unes quantes dades de l'Institut d'Estadística de Catalunya (IDESCAT) sobre la discapacitat a Catalunya [1].

### Persones amb discapacitat reconeguda 2012

PROVÍNCIA	Físics motòrics	Físics no motòrics	Visuals	Auditius	Intel·lectual	Malalts mentals	No hi consta	Total
Barcelona	130.215	91.438	26.908	21.217	32.323	73.359	500	<b>375.960</b>
Girona	15.571	6.548	2.113	1.684	5.580	5.700	50	<b>37.246</b>
<b>TOTAL</b>	<b>145.786</b>	<b>97.986</b>	<b>29.021</b>	<b>22.901</b>	<b>37.903</b>	<b>79.059</b>	<b>550</b>	<b>413.206</b>

Catalunya	172.071	114.156	33.985	26.754	47.980	94.492	696	<b>490.134</b>
-----------	---------	---------	--------	--------	--------	--------	-----	----------------

## Població per províncies 2012

ZONA	Barcelona	Girona	Lleida	Tarragona	Catalunya
HABITANTS	<b>5.552.050</b>	<b>761.627</b>	443.032	814.199	7.570.908

### Conclusions extretes

Si sumem les poblacions de Barcelona i Girona (radi d'acció d'ACTUASERVEIS SCP), obtenim 6.313.677 habitants, o el que és el mateix el 83,4% de la població de Catalunya.

Tenint en compte les dades de les dues taules anteriors tenim que un 6,4% dels catalans pateixen algun tipus de discapacitat.

A la zona de Barcelona i Girona el percentatge no varia pràcticament i observem que un 6.5% dels barcelonins i gironins pateixen algun tipus de discapacitat i que formen part del més del 84% de discapacitats de Catalunya.

### 1.2. ACTUASERVEIS SCP

ACTUASERVEIS SCP, és una empresa dedicada a oferir serveis interdisciplinaris i amb una àmplia experiència en la cura i tracte a persones amb necessitats especials (gent amb discapacitats físiques o intel·lectuals, persones de la tercera edat, malalts d'Alzheimer, etc.) [2].

Té la seva seu ubicada a Centelles, població d'uns 7000 habitants localitzada al sud de la Comarca d'Osona, aproximadament a uns 50 km al nord de Barcelona.

Els serveis a discapacitats té un fort pes en la dinàmica del dia a dia de ACTUASERVEIS SCP, que centra el seu radi d'acció en les províncies de Barcelona i Girona, amb l'objectiu a mig termini, d'arribar a tot el territori de Catalunya.

L'empresa organitza els seus usuaris depenent del grau o tipus de necessitats que requereixen. Per tant, trobem un sistema de classificació d'usuaris que es centra en tres grans grups:



- **Tercera edat**, on hi entrarien aquelles persones que, degut al lògic deteriorament a causa de l'edat, necessiten d'una assistència suplementària amb més o menys regularitat.

- **Discapacitats**, grup del que en formen part persones amb algun tipus de deficiència sigui física, intel·lectual o sensorial. Estan, a la vegada, organitzats en tres subgrups segons el grau de d'assistència que necessitin. De menys a més atenció requerida hi trobem:

- Intermitents: amb una atenció just quan es necessiti, no sempre necessària.

- Limitats extens: amb una atenció regular i continua.

- Generalitzats: amb una atenció constant i d'alta intensitat.

A més, trobem els Intermitents Sènior i els Limitats Extens Sènior que mantenen les característiques descrites en els seus corresponents subgrups però que, a més, estan en una avançada edat.

- **Malalts Mentals**, grup on podem trobat totes aquelles persones que, sense tenir cap discapacitat de les comentades anteriorment, pateixen d'alguna malaltia mental com, per exemple, serien la depressió, ansietat o certs tipus de fòbies severes.

Com a serveis que ofereix l'empresa en trobem dos de clarament diferenciats:

**Sortides**: Organitzades per a grups d'usuaris segons els tipus de necessitats que acabem d'anomenar. Solen ser activitats que duren varis dies o, fins i tot, setmanes. En aquestes sortides s'hi fan visites a diferents indrets, activitats d'oci, jocs,... Estan tutelades per varis monitors el número del qual dependrà del número d'usuaris i de les necessitats que aquests precisin.

**Acompanyaments**: Serveis d'atenció personalitzada que es duen a terme entre un sol treballador i un sol usuari. Acostumen a ser serveis que poden durar des de hores fins a tota una jornada. Aquí podem trobar des d'acompanyar a una visita al metge, vigilàncies en hospitals o auxiliar a una persona en tasques domèstiques com el vestir-se o incorporar-se del llit.

A través de la responsable de la direcció de projectes d'ACTUASERVEIS, se'ns va demanar algun sistema per a poder millorar i modernitzar el mètode de treball que utilitza l'empresa tot acomplint les necessitats concretes que ella requereix.

### 1.3. Objectius i Motivació

Amb això, el nostre **principal objectiu** és crear una aplicació web que gestioni la base de dades de l'empresa per tal de millorar la coordinació de tots els involucrats en aquesta tasca i, així, obtenir-ne un millor funcionament que, alhora, es pugui traduir en un millor servei per als seus usuaris i clients.

Un cop definit l'objectiu principal anem a veure els objectius secundaris que ens permetran aconseguir-lo:

#### Planificació eficaç d'activitats

Aquesta aplicació, servirà per planificar temporalment les sortides i activitats relacionades amb l'empresa, mantenir un control de les necessitats concretes dels diferents grups d'usuaris.

#### Control i seguiment dels actors implicats

També hem de poder facilitar el fer un seguiment del rendiment dels treballadors i el seu tracte amb aquests usuaris, facilitar l'organització logística de l'empresa i millorar la comunicació entre l'empresa i els centres o particulars que contractin els seus serveis.

#### Motivació

La principal motivació que ens porta a realitzar el projecte, és la de oferir un salt de qualitat en l'organització de l'activitat diària d'una empresa dedicada a una tasca, que creiem, molt necessària. Això es traduirà, segurament, en una millor atenció als usuaris i treballadors d'ACTUASERVEIS SCP.

A més, a tret personal, la motivació es encara més gran al tractar-se d'una empresa que es troba al meu mateix poble i així col·laborar en la feina com la que duen a terme persones tan properes. Tots hauríem d'arribar a la tercera edat o podríem patir algun tipus de discapacitat o malaltia mental. Per tant, tots podríem acabar necessitant algun tipus d'atenció especial.

## 1.4. Estructura del document

Per finalitzar aquesta primera part introductòria anem a explicar, de manera mot resumida, el contingut dels següents capítols que trobarem en aquesta memòria:

**Capítol 2:** Es farà un anàlisi dels requeriments del projecte. Des del perfil d'usuari que es preveu que utilitzarà la nostra aplicació, passant pels requeriments funcionals i no funcionals de la mateixa, l'anàlisi de riscos i la planificació de les tasques. Finalment, desglossarem el pressupost previst per a la realització del projecte.

**Capítol 3:** En aquest capítol, es descriurà l'estat actual de les tecnologies que utilitzarem. Descriurem els elements que necessitarem i aprofundirem, a un nivell raonable, en el funcionament dels Frameworks d'aplicacions web i el XAMPP, paquet de programes que serà la base del nostre treball.

**Capítol 4:** Es recordaran breument els conceptes relacionats amb l'empresa, ja mencionats en la introducció i necessaris per entendre'n el funcionament general. Es definiran els casos d'ús i es mostrarà el Diagrama de Classes.

**Capítol 5:** Vinculat estretament amb el capítol anterior, ja que de la informació exposada, es definiran el Diagrama ER, amb la definició de les Entitats i les seves Relacions, i el model relacional de taules, amb la descripció de cadascun dels camps que contindrà cada taula.

**Capítol 6:** Capítol on s'explicarà, pas a pas, tot el procediment que hem seguit per a implementar l'aplicació web a fi de complir els requisits especificats.

**Capítol 7:** S'explicaran els tests i les proves d'avaluació de l'aplicació que demostrin l'assoliment de cadascun dels casos d'ús descrits en el capítol 4.

**Capítol 8:** Capítol final on es mostraran les conclusions extretes en la realització del projecte.

**Altres:** Després del capítol 8, es mostren les referències consultades i un índex de les figures utilitzades per il·lustrar alguns punts concrets de la memòria. També comentar, que en la còpia digital que s'entregarà a Gestió Acadèmica després de la defensa del projecte, inclourem el codi utilitzat i organitzat tal i com s'explicarà en aquesta memòria.



## 2. Anàlisi de requeriments

En aquest capítol, primer analitzarem quin tipus d'usuari acabarà utilitzant la nostra aplicació web i les característiques que esperem d'aquest usuari. Seguidament, estudiarem quins requisits tenim a l'hora de dissenyar l'aplicació i, llavors, ens centrarem en la planificació temporal de les tasques a realitzar. Finalment, mostrarem un pressupost orientatiu dels costos del nostre projecte.

### 2.1. Perfil d'usuaris

Anem a analitzar quin serà el perfil d'usuari de la nostra aplicació per a tenir en compte les característiques i requeriments que aquesta haurà de preveure i complir.

Ens trobem davant d'un usuari amb un nivell d'estudis mitjà o alt, amb una relativa experiència en l'administració i organització de gran quantitat de dades. De fet, la nostra aplicació serà una millora directe del sistema utilitzat fins ara per ACTASERVEIS SCP, cosa que reduirà el temps d'aprenentatge si el comparem amb el temps que podríem haver necessitat si es tractés d'un usuari que comencés a treballar des de zero amb un aplicatiu que suposés un sistema de funcionament totalment desconegut.

A més, el nostre prototip d'usuari, comptarà amb un grau mitjà o alt en el maneig de les noves tecnologies, especialment, d'un ordinador. L'aparició de fenòmens com les xarxes socials o l'augment de possibilitats que ha aportat l'increment de la velocitat de processament dels ordinadors i de la transferència de dades per Internet, ha fet elevar considerablement el nivell mitjà de l'habilitat de la població en l'ús dels aparells electrònics, especialment, els més tradicionals tipus ordinador o telèfon mòbil.

Aquestes característiques, ens portaran a idear una aplicació que s'assembli el és possible a entorns de treball gràfics prou coneguts pel ciutadà mitjà actual. Aquests entorns poden ser lleugerament complexes com per a permetre funcionalitats prou avançades, però sense perdre mai de vista la practicitat que ha popularitzat l'ús de les tecnologies en tasques clàssicament fetes de manera més manual o primitiva.

Aquests punts s'hauran de tenir en compte, especialment, a l'hora de definir els requeriments no funcionals de la nostra aplicació i, en conseqüència, els funcionals. Anem, doncs, a descriure'ls.

## 2.2. Requeriments Funcionals i no Funcionals

### Requeriments funcionals de la nostra aplicació

#### ***RF1: Controlar tots els elements que componen l'aplicació de manera específica (CRUD).***

Com en tota base de dades, hem d'assegurar un funcionament correcte de cadascun dels seus elements tan de manera individual com tenint en compte la relació que pugui haver-hi entre cada un d'ells. Així hem de poder realitzar les manipulacions bàsiques de creació, edició, eliminació o cerca.

#### ***RF2: Disposar d'un sistema de classificació dels usuaris i les seves necessitats concretes.***

El fet de treballar amb usuaris de tanta diversitat, ens obliga a preveure algun tipus de classificació basada en els conceptes que hem definit en el capítol 1.

#### ***RF3: Control, seguiment i valoració dels treballadors de l'empresa.***

Un altre requisit, és el fet de poder controlar tan el número com la "qualitat" els serveis que realitzen els treballadors a través de l'empresa i avaluar-ne el rendiment de cara a futures contractacions.

#### ***RF4: Control i seguiment de les entitats implicades com els centres o associacions d'usuaris.***

Seguint la direcció de l'anterior requeriment, se'ns demana que l'aplicació pugui tenir un sistema de control d'activitats i serveis que demanden els centres o persones que contacten amb l'empresa.

#### ***RF5: Definició i acotament dels diferents tipus de servei: Sortides i Acompanyaments.***

Com hem comentat anteriorment, ens trobem amb dos grans serveis ben diferenciats. Com a tals, necessiten de sistemes de control i gestió diferents i per separat.

***RF6: Control i càlcul de les despeses de Quilometratge dels Acompanyaments.***

Un dels trets que diferencien els acompanyaments, és la necessitat de calcular les despeses extres que puguin suposar per a l'empresa. Per això necessitem algun mètode per a computar-los.

***RF7: Permetre el funcionament en xarxa per a varis administradors alhora.***

Tot i tractar-se d'una empresa relativament petita, aquesta disposa d'un mínim de 3 treballadors que han de poder utilitzar la nostra aplicació alhora. És per això que necessitarem d'alguna arquitectura que ens permeti complir amb aquesta necessitat.

Requeriments no funcionals:

***RNF1: Funcionament fàcil i intuïtiu.***

Com en la pràctica totalitat d'aplicacions o programari en general, un funcionament amb aquestes característiques redueix, en principi, la corba d'aprenentatge i fa millorar la productivitat dels treballadors d'administració de l'empresa.

***RNF2: Estructura conseqüent alhora d'integrar els elements en els menús de l'aplicació.***

Seguint amb relació a l'anterior requisit no funcional, volem concedir els principals avantatges del disseny centrat en l'usuari de les aplicacions web 2.0, permetent així una gran facilitat en l'accés a les accions i seccions que necessiti tractar el treballador administratiu.

***RNF3: Maximitzar l'accés i comunicació dels diversos elements mitjançant els enllaços web.***

Un altre requisit que va en consonància amb els dos anteriors. Seguint el disseny de varis wikis, blogs i xarxes socials, volem proveir d'un sistema d'enllaços dins de l'aplicació que acceleri encara més la velocitat d'accés a cadascun dels elements que compondran la nostra aplicació. És un sistema que, tot i trencar la jerarquia de menús, ofereix dreceres ràpides cap diferents punts de l'aplicació.

***RNF4: Maximitzar la compatibilitat entre sistemes operatius de l'aplicació.***

Un requisit quasi obligatori ja que la portabilitat en aplicacions és un de les característiques més importants en el seu desenvolupament. Hem d'intentar que l'aplicació pugui funcionar tant en sistemes operatius Windows, com basats en Linux o, fins i tot, MacOS.

**RNF5: Permetre cert grau d'actualització i ampliació de cara a funcionalitats futures.**

És possible que les característiques dels usuaris canviïn, el catàleg de serveis augmenti o el funcionament de l'empresa simplement variï per tal d'oferir un millor servei. Per tant, hem de tenir la previsió de que es puguin afegir funcionalitats a l'aplicació i aquestes s'han de poder integrar de manera que no afecti al funcionament de l'aplicació original.

### 2.3. Anàlisi de riscos

Per a l'anàlisi de riscos, hem de tenir en compte quins són els actius del projecte.

Per una banda podríem definir els actius propis de l'aplicació que serien les mateixes dades de la Base de dades i la manera en què es mostraran en pantalla per a que l'administrador del sistema faci funcionar l'aplicació web.

A més, tenim l'administrador del sistema del qual se n'ha d'avaluar el nivell de coneixement del funcionament de l'aplicació (amb la corba d'aprenentatge lògica), l'ús més o menys destre de l'ordinador on s'executarà l'aplicació o el grau de coneixement de la pròpia empresa.

D'aquests actius en podem preveure els següents riscos:

**Risc en la seguretat de l'aplicació:** No podem permetre que qualsevol persona o element extern tingui accés a poder manipular l'aplicació. Un sistema d'identificació amb clau tipus LOGIN tan pels administradors de l'aplicació, com de cara al servidor de la xarxa podria minimitzar la probabilitat d'aparició d'aquest risc.

**Risc en la pèrdua de dades:** Tan per error en el hardware (del tipus sobreescalfament del PC, falla en la corrent elèctrica,...), en el software (desconfiguracions vàries) o degut a elements externs (petit incendi, aiguats,...). Més enllà del control rutinari de les instal·lacions de l'empresa, fet que sobrepassa la nostra funció, realitzar còpies de seguretat (BACKUPS) de les dades de l'aplicació regularment i tenir-los emmagatzemats en dispositius externs o en el núvol, seria una bona manera de reduir aquest risc.

**Risc en la redundància no controlada o inconsistència de les dades:** Un dels principals riscos en la gestió de les bases de dades. Per exemple, evitar que un treballador o usuari s'assigni o



s'apunti a 2 esdeveniments simultanis o que hi hagi 2 usuaris o treballadors duplicats o 2 esdeveniments on no poden ser-hi,... Això ens portaria a un increment innecessari del treball de l'aplicació i dels administradors i una pèrdua d'espai de memòria d'emmagatzemament. Realitzar un bon disseny del software abans d'implementar l'aplicació i fer-ne tests i avaluació després, ens ha de permetre controlar aquest risc.

**Risc en les pròpies funcionalitats de l'aplicació:** Hem d'intentar eliminar pràcticament els errors de funcionament automàtic de l'aplicació (per exemple, cometre un error en el càlcul de quilometratge o una mala classificació d'usuaris). Novament, la realització de tests i avaluació de l'aplicació ens minimitzaria els efectes d'aquest risc.

**Risc en l'ús erroni de l'aplicació per part dels administradors:** Ens referim a l'ús erroni involuntari de l'aplicació. Evitar, per exemple, que l'administrador cliqui en un botó per error o confusió, insereixi alguna dada equivocada,... Tenir en compte la col·locació correcta dels menús, botons d'acció, enllaços a altres funcionalitats, la validació de dades entrades amb format específic com DNIs, número de Seguretat Social, e-mails,...

**Risc en l'increment desmesurat del volum d'informació:** Hem d'evitar que l'aplicació pugui quedar obsoleta o desfasada degut al gran volum de dades que manegi. Per evitar col·lapsar l'aplicació necessitem algun mètode que ens asseguri les necessitats de l'empresa en, almenys, varies dècades de treball (altes d'usuaris, de treballadors, més serveis,...)

Per altra banda, podem definir com a actius de l'empresa el conjunt d'usuaris, centres i treballadors que compondran les dades de la nostra aplicació. En quan als riscos que podem preveure d'aquests actius tenim:

**Risc en la protecció de dades personals de la base de dades:** La llei Orgànica de Protecció de dades de caràcter personal a través de l'agència Catalana de Protecció de Dades, s'encarrega de garantir el compliment de la legislació vigent sobre protecció de dades de caràcter personal i controlar-ne l'aplicació, drets d'informació, accés, rectificació, cancel·lació i oposició. Per això, la nostra aplicació al contenir dades privades, ha de tenir un cert control en l'accés d'aquestes dades. Per això algun sistema d'identificació amb clau d'accés podria protegir aquestes dades de tercers.

Aquests actius, també es veuen afectats de manera directa o indirecta per algun risc dels que hem comentat anteriorment com la redundància i consistència de les dades o l'error de funcionalitats com seria en el càlcul de quilometratges o distribució d'usuaris en grups.

Per representar aquests riscos d'una manera més gràfica, els mostrem a la següent taula-resum (**Figura 1**), on establim una relació entre la probabilitat que considerem que tenen de produir-se i la gravetat de l'impacte que tindrien en cas de donar-se:

RISC	PROBABILITAT	IMPACTE
Seguretat de l'aplicació	MITJANA	ALT
Pèrdua de dades	BAIXA	ALT
Inconsistència de dades	BAIXA	ALT
Funcionalitats de l'aplicació	BAIXA	MITJÀ
Ús erroni	MITJANA	BAIX
Volum d'informació	MITJANA	BAIX
Protecció dades personals	MITJANA	ALT

Figura 1. Taula-Resum de l'Anàlisi de Riscos

## 2.4. Planificació de Tasques

Hem organitzat la tasca a realitzar per a aquest projecte en mòduls de treball que definirem mitjançant 5 fases seqüencials. Les descrivim a continuació:

### Fase 1: Anàlisi previ del projecte.

En aquesta fase ens hem dedicat a recollir els requeriments que se'ns demanava des de l'empresa ACTUASERVEIS SCP. Se'ns van descriure les necessitats que tenien tan en l'organització de les dades amb les que treballaven, com amb l'organització que aquestes necessitaven. També ens van descriure les característiques de l'entorn de treball on es faria servir l'aplicació web (xarxa local, número d'ordinadors,...).

L'altre gran punt de treball d'aquesta fase va ser analitzar les tecnologies disponibles per a dur a terme el nostre projecte. Vam estudiar totes les possibilitats lògiques per a fer la base de dades en quan a utilització o no de Frameworks, llenguatges de programació, gestors de base de dades, servidors web,...

## Fase 2: Disseny del Software

La segona fase estava encarada en tot el que està relacionat en la definició del sistema en si un cop analitzades les dades extretes en la primera fase. Un cop conegudes les necessitats de la nostre aplicació i els actors que hi participarien, vam dissenyar teòricament la nostre base de dades, utilitzant els mètodes conceptuals i els diagrames corresponents: Anàlisi de casos d'ús, Diagrama ER, Diagrama de Classes i, finalment, el model relacional de les Taules que contindria la nostra Base de dades.

## Fase 3: Programació i Implementació del Software

A partir del disseny anterior, vam passar a començar la codificació de tot el sistema. Des de la instal·lació del programari necessari, passant per la creació de les taules tal i com les havíem previst en el disseny anterior fins a la programació pròpiament dita de l'aplicació web.

Finalment, es va fer una instal·lació d'una versió Alfa amb les funcionalitats bàsiques per tal de fer un petit test de funcionament i posterior recollida d'incidents.

## Fase 4: Versió Beta

Estretament lligada a l'anterior fase, es va fer el mateix però amb una versió Beta de l'aplicació, que ja incloïa totes les funcionalitats demanades. També vàrem fer una recollida d'incidents.

## Fase 5: Desplegament / Manteniment.

Aquesta fase incloïa la instal·lació de la versió definitiva de l'aplicació i un seguiment del seu funcionament que va més enllà de la data d'entrega d'aquest projecte. A més, no es descarten possibles actualitzacions o ampliacions de les funcionalitats de l'aplicació.

Aquestes fases les detallem en el següent Diagrama de Gantt (**Figura 2**):

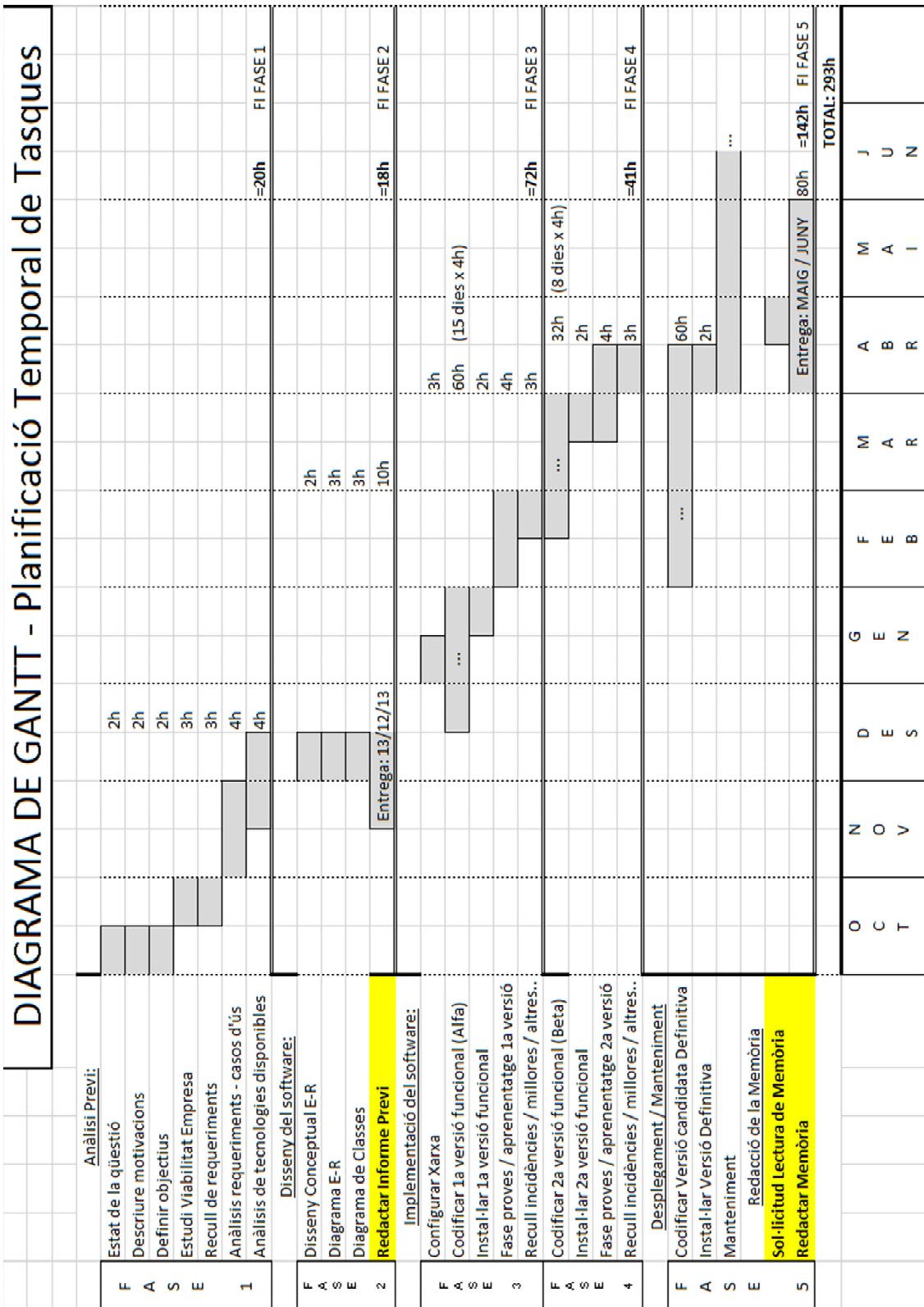


Figura 2. Diagrama de Gantt

## 2.5. Pressupost

Veient la planificació anterior anem a resumir les estimacions de pressupost en la construcció del nostre programa en la següent taula:

CONCEPTE	HORES DEDICADES	PREU / HORA o COST	TOTALS
Anàlisi	20 hores	30 € / hora	600 €
Disseny	18 hores	30 € / hora	540 €
Implementació			
1a FASE	68 hores	30 € / hora	5.010 €
2a FASE	37 hores		
3a FASE	62 hores		
Aprenentatge	8 hores	20 € / hora	160 €
Programari	--	0 €	0 €
TOTAL	213 hores	--	6.310 €

Per tant, si comptéssim tots els costos **d'anàlisi** (Estudi viabilitat, requeriments, estat de la qüestió,...), **disseny** (disseny conceptual E-R, Diagrames de classes, definició de vistes,...), **implementació** (codificar l'aplicació, instal·lació, fase de proves) i l'**aprenentatge** (explicació del funcionament de l'aplicació, fase de proves, recull d'incidències,...), ens dona un pressupost estimat de **6.310 €**.

Cal tenir en compte que, al treballar amb programari lliure i gratuït ens permet no haver de pagar cap cost addicional pel material utilitzat.



### 3. Estat de la qüestió

Aquest capítol està dedicat a analitzar l'estat actual de les bases de dades i les aplicacions web. També farem un breu anàlisi de les tecnologies disponibles i dels elements necessaris per a realitzar el nostre projecte. També comentarem les opcions triades per a dur-lo a terme.

#### 3.1. Arquitectures d'aplicació web

Degut a la necessitat de disposar d'una aplicació que pugui ser utilitzada al mateix temps per varis ordinadors connectats en xarxa, utilitzarem alguna solució que ens permeti treballar en el que s'anomena arquitectura **client-servidor**.

Aquesta arquitectura es basa en la relació establerta entre el **servidor**, que ofereix un recurs de qualsevol tipus (físic, de programari, de dades,...), i el **client** perquè aquest en tregui un profit o avantatge.

El **servidor** el podríem definir com a una computadora on s'executaria l'aplicació web que ens ocupa i que, en el nostre cas, es tractaria d'una espècie d'híbrid entre **servidor web** i **servidor de base de dades** ja que, per una banda, emmagatzemarà documents tipus html o arxius d'imatge, clars exemples de material web i, per l'altra, proveirà d'un servei de base de dades [3].

Els **clients** seran cadascun dels ordinadors que utilitzin a les oficines d'ACTUASERVEIS per a utilitzar aquesta aplicació. Mitjançant adreces IP, aquests clients es comunicaran al servidor per tal d'emprar el servei que aquest ofereix (aplicació web) de manera remota. Bàsicament, les comunicacions seran, o bé, peticions dels clients, o bé, les respostes a aquestes peticions. Il·lustrem aquesta arquitectura a la **Figura 3**, que ve a continuació:

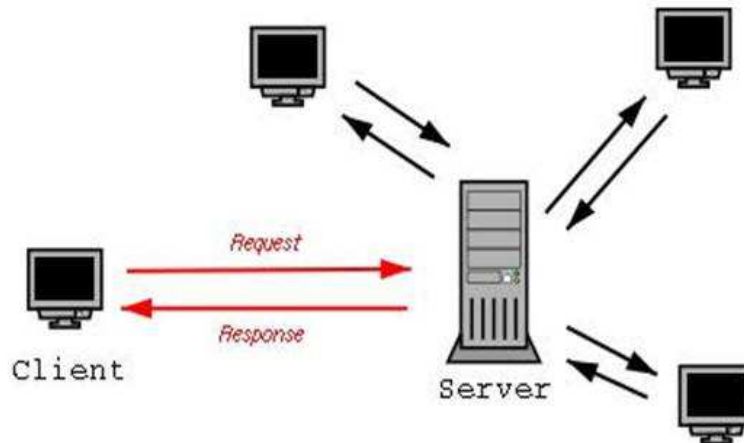


Figura 3. Esquema client-servidor amb fluxos de petició i resposta

Per exemplificar una mica més, altres tipus de servidor habituals que podem trobar serien el **servidor d'impressores** (que controla diverses impressores i els fa arribar les peticions de màquines de la mateixa xarxa), **servidor de correu electrònic** (emmagatzema, envia, rep i realitza operacions relacionades amb el correu electrònic per als clients de la xarxa) o el **servidor de telefonia** (resposta automàtica de trucades, emmagatzema missatges de veu o encamina les trucades a diferents clients de la xarxa).

Per altra banda, l'aplicació gestionarà una base de dades i ens hem decantat per el **patró d'arquitectura de software MVC** (Model-Vista-Controlador).

Encara que originalment el model MVC va ser ideat per a aplicacions d'escriptori, ha estat àmpliament adaptat com a arquitectura per dissenyar i implementar aplicacions web en els principals llenguatges de programació.

En quan a la programació utilitzant el patró MVC, tal i com il·lustrarem a la **Figura 4**, aquesta es basa en separar l'aplicació en tres parts principals [4]:

**Model:** Representaria la informació amb la qual el sistema opera. Per tant, és la part que s'encarrega de l'accés a les dades tan a través de consultes com a actualitzacions o a la implementació dels privilegis d'accés a les pròpies dades.

**Vista:** És la part que s'encarrega de representar el Model en un format adequat per a interactuar-hi. Usualment es pot veure com a la interfície d'usuari.



**Controlador:** És el component encarregat de respondre als esdeveniments (accions de l'usuari), és a dir, a la lògica d control de l'aplicació. Per exemple, editar un registre, eliminar-lo, crear-lo,...

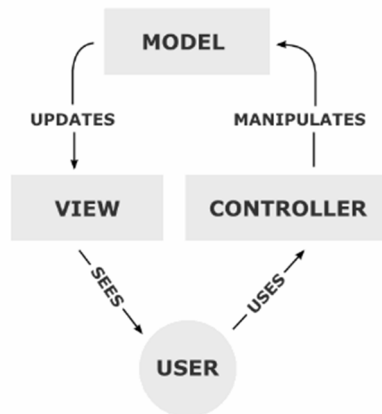


Figura 4. Esquema de col·laboració entre els components MVC

Per a implementar el patró d'arquitectura MVC, utilitzarem el que es coneix per Framework i que explicarem detingudament en els següents capítols.

### 3.2. Tecnologies Web i Bases de dades

La nostra aplicació web gestionarà una base de dades en una xarxa local. Per tant, necessitem poder comptar amb una sèrie d'elements que comentem a continuació, exposant-ne les alternatives òptimes a utilitzar en cada cas i la justificació de la opció final triada:

- **Servidor Web:** o servidor HTTP, és un programa capaç de processar alguna acció mitjançant connexions bidireccionals (en el nostre cas) en un sistema amb l'esquema CLIENT – SERVIDOR. S'acostuma a presentar en forma d'aplicacions executades en un navegador web.

Hi ha multitud de servidors web per a poder elegir [5]. En el nostre cas, no és previst que hàgim de treballar amb un volum desmesurat de tràfic de dades per això, en principi, podríem utilitzar qualsevol dels servidors més utilitzats actualment.

Comencem parlant de la primera opció considerada, el **Windows Server**. Necessari si es treballa amb tecnologia .NET, no és molt adequat per a llocs webs grans. Dels seus avantatges en

destaquen la fàcil configuració i instal·lació o la integrabilitat en tots els productes de la família Microsoft. En la resta dels aspectes no sembla millorar cap de les següents alternatives [6].

Parlem ara de **Lighttpd** que, com el seu nom indica, té com a principal avantatge la rapidesa amb la que treballa. Aquest fet va suposar que, en un principi, l'adoptessin llocs webs com Youtube o Wikipedia. A pesar d'això, la falta d'estabilitat i maduresa del projecte, així com la falta de *saber-fer* (transferència de competències, coneixements, tecnologies, mètodes de fabricació, mostres de fabricació,...) fa que presenti alguns errors greus (*memory leak*, per exemple) i que, aquests, es tardin a solucionar.

La rapidesa també és una dels grans punts forts de **Cherokee**, servidor molt senzill de configurar que sembla que durant els últims temps, a partir del 2011, hagi quedat estancat per la falta de desenvolupament dels seus responsables. Tot i això, és una alternativa realment a tenir en compte per característiques com l'actualització sense parada de servei o el rendiment general del mateix servidor.

Un altre opció a tenir en compte és **NGINX**, un servidor molt interessant ja que uneix una bona velocitat d'execució, una comunitat d'usuaris prou activa, un bon nivell en la documentació del projecte i, sobretot, una flexibilitat que li permet suportar les configuracions més habituals.

Finalment, parlarem de l'opció que hem cregut més convenient. El servidor **Apache**. És un servidor tan per plataformes Unix, Microsoft Windows o Macintosh i des de 1996 és, segons Netcraft, el servidor HTTP més utilitzat arreu del món, arribant a la seva màxima popularitat al 2005 on es calcula que el 70% de les pàgines web l'utilitzaven (**Figura 5**):

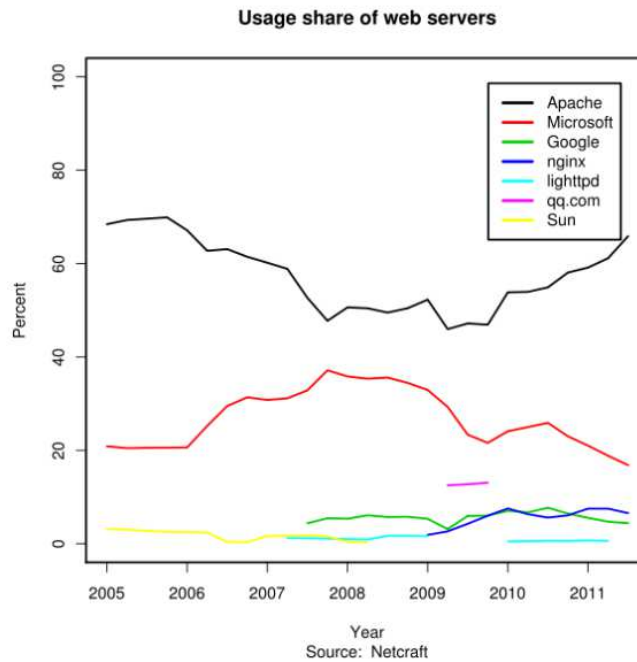


Figura 5. Gràfic d'utilització de Servidors Web

A més, té característiques altament configurables per a bases de dades com l'autenticació o la negociació de continguts. És el servidor que disposa de la major documentació i flexibilitat de configuració. Com a desavantatge es podria indicar que, consumeix molts recursos sobretot si treballa amb un volum alt de tràfic de dades.

Com a curiositat, deu el seu nom a la tribu Apache que va ser la última en rendir-se davant del que seria el futur govern d'Estats Units. El seu creador va voler fer el símil amb aquesta tribu de la que se'n reconeixen la fermesa i energia però no així l'agressivitat dels seus membres [7].

- **Navegador Web:** programa informàtic capaç de mostrar documents d'hipertext, generalment escrits en HTML, que es troben en un servidor web o, en el nostre cas, en un servidor local.

Una de les maneres més pràctiques d'oferir accés a una base de dades és implementar-la com a una aplicació web. Aquest tipus d'aplicacions han guanyat popularitat últimament gràcies a que l'usuari pot tractar i manipular, de manera intuïtiva, les dades emmagatzemades dins de la base de dades com si d'una pàgina web d'Internet es tractés, ja que se sol utilitzar un navegador web com a client.

En aquest cas, la elecció del navegador web no ha de ser nostra ja que un dels nostres objectius és permetre que l'usuari esculli el navegador web que utilitzi abans de la implementació de la nostra aplicació. Només comentar els principals navegadors que hi ha actualment en ús com són Google Chrome, Mozilla Firefox, Ópera, Safari o Internet Explorer:

### Estadístiques d'ús de navegadors web [8]

Navegadors	abril del 2009	abril de 2010	abril de 2011
Internet Explorer	66,82%	60,65%	55,11%
Mozilla Firefox	22,05%	24,52%	21,63%
Google Chrome	1,23%	6,13%	11,94%
Safari	8,23%	4,65%	7,15%
Opera	0,70%	3,15%	3,25%
Netscape	0,69%	--	--
Altres	0,26%	0,85%	0,91%

Font: wikipedia

- **Llenguatge de programació:** Per a crear la interfície web d'usuari, trobem també molts llenguatges de programació que s'utilitzen per al disseny web d'una Base de Dades (PHP, C, C++, Perl o Ruby).

En principi, l'elecció d'un o altre llenguatge, no ens limitaria a l'hora de programar més enllà del coneixement previ que podem tenir d'un o altre. Però basant-nos en opinions d'usuaris, experiència personal prèvia i les característiques de cadascun d'ells, ens hem decantat per considerar **Ruby i PHP** com a les alternatives més sòlides.

El llenguatge de programació **Ruby** va ser creat per Yukihiro "Matz" Matsumoto l'any 1993. És un llenguatge de guions totalment orientat a objectes. Està molt orientat al tractament de fitxers i per manteniment del sistema [9]. És simple, extensible i portable. El principal desavantatge que trobem en utilitzar aquest llenguatge és el poc ús que té en àmbits laborals de programació si el comparem amb els altres dos candidats pel fet de tenir una corba d'aprenentatge més gran.

**El PHP (Personal Home Page)** va ser creat l'any 1994. L'origen i causa de la seva creació podria tractar-se com un altre punt a favor en la nostra elecció. I és que, el seu creador Rasmus Lerdof, buscava una solució ràpida a la sobrecàrrega que patia el servidor web on allotjava la seva pàgina personal i currículum. Tan sols un parell d'anys després, 15.000 pàgines web l'utilitzaven, xifra

que es veuria quadruplicada dos anys més tard, el 1998. Es calcula que actualment, PHP està present d'una manera o altre en el 24% de pàgines web del món.

Més enllà de les xifres, PHP ens aporta una sèrie d'avantatges. És un llenguatge totalment lliure i obert, la seva corba d'aprenentatge és molt baixa, els entorns de desenvolupament són ràpids i senzills de configurar, té un desplegament fàcil (a partir de paquets totalment autoinstal·lables), una comunitat d'usuaris molt gran i en constant desenvolupament i, a més de tot això, compta amb un accés molt senzill a bases de dades.

En aquest capítol, també podríem considerar la possibilitat de programar en **AJAX**. Són les sigles de Asynchronous Javascript And Xml, (JavaScript asíncron i XML), , una llibreria programada en Javascript, que permet actualitzar continguts web sense haver de tornar a carregar la pàgina, ja que treballa des del costat del client. Això obre la porta a pàgines web interactives. Sens dubte seria una bona solució per al projecte que ens ocupa ja que permet, a més, poder treballar conjuntament amb PHP. Tot i això hem descartat aquesta possibilitat, ja que hem considerat que amb PHP tindríem la tecnologia suficient com per a resoldre el problema. No descartem utilitzar-lo en un futur per si s'ha de millorar, canviar o estendre alguna funcionalitat.

- **Sistema Gestor de Base de Dades dinàmica:** o SGBD, és un programa que s'encarrega de l'escriptura, recuperació, edició i esborrat de les dades dins de la nostra Base de Dades [10]. Perquè un programa SGBD sigui fiable ha de complir amb les propietats conegudes com ACID:

(A)atomicitat: Una transacció o finalitza correctament o no deixa rastre de la seva execució.

(C)onsistència: La concurrència de transaccions no pot produir resultats anòmals.

(I)solament o Aïllament: Cada transacció s'ha d'executar com si fos l'única en aquell moment.

(D)efinitivitat: Si es confirma una transacció, el resultat ha de ser definitiu i no es pot perdre.

Com a primera alternativa, trobem **Oracle**. És el SGBD més emprat a nivell mundial. Multiplataforma, fàcil d'utilitzar i configurar, bon rendiment en les transaccions, gestió de seguretat, portabilitat i una llista de prestacions més, són les claus per fer-lo el SGBD més utilitzat. Com a desavantatges trobem el seu elevat preu o la quantitat d'errors que contenen les versions més recents.

Per altra banda tenim **PostgreSQL**. També multiplataforma, té una major seguretat que l'anterior. Permet la gestió de varis usuaris i els permisos assignats a cadascun d'ells, l'ús d'arrays com a tipus de dades, la possibilitat de crear tipus propis, és gratuït i està especialment dissenyat per a ambients d'alt volum d'informació. Amb això, no té una velocitat de resposta massa ràpida i això fa que sigui un SGBD poc utilitzat.

Un altre possibilitat seria **SQL Server**. És un SGBD comercial tot i que compta amb una versió gratuïta que limita el número de connexions simultànies. Conté un potent entorn gràfic per a l'administració, té gran estabilitat, escalabilitat i seguretat. Tot i això, consumeix gran quantitat de memòria RAM tan en la instal·lació com en la utilització del software. La relació qualitat-preu és molt baixa i no té una bona implementació de tipus de dades.

Finalment, anem a analitzar l'opció triada. El **MySQL**. És un SGBD gratuït, multiplataforma, amb gran velocitat al realitzar operacions, no necessita grans requeriments de sistema, facilitat d'instal·lació i té varis modes d'emmagatzemament (InnoDB, MyISAM,...). Com a contres trobem que no és molt intuïtiu, no té suport o no permet un mode d'autenticació local. De cara a l'usuari final, aquests desavantatges no tenen una influència directe en el funcionament de l'aplicació.

Aquesta tria, ens porta a comentar que l'eina que utilitzarem per a gestionar l'administració de MySQL a través del navegador serà **phpMyAdmin**, eina escrita en PHP per tal de crear o eliminar Bases de Dades; crear, eliminar i alterar taules; afegir, esborrar o editar camps,... en definitiva executar qualsevol sentència SQL. També ens permet administrar claus en camps, administrar privilegis, exportar dades en diversos formats i està disponible en 50 idiomes. Es troba disponible sota la llicència GPL i va ser avaluat com el millor eina d'administració de base de dades en la comunitat de descàrregues de SourceForge.net. Aquest projecte es troba vigent des de l'any 1998.

### 3.3. Frameworks per a Aplicacions Web

Últimament, ha crescut de manera important l'ús de Frameworks per al disseny i implementació d'aplicacions web.

Framework és un terme adoptat de l'anglès i equival a entorn de treball o, també, marc de treball. Inclou tots aquells recursos, la metodologia en la seva disposició i la seva interfuncionalitat que determinin el desenvolupament organitzat d'un projecte de programari. Pot incloure programari de suport, llibreries de codi, llenguatges de programació, i programari extra que ajudi a organitzar, desenvolupar i integrar components diversos d'un altre projecte de programari.

Un Framework representa una arquitectura de software que modela les relacions generals de les entitats del domini, i dona una estructura i una metodologia de treball, que estén o fa servir les aplicacions del domini.

Dels avantatges que podem tenir utilitzant un Framework en destaquen el ràpid desenvolupament de les aplicacions, la reutilització dels components del software (el Framework es considera un dels paradigmes de la reutilització), la facilitat en l'aprenentatge del seu ús o la flexibilitat de programació que proporciona (degut a la seva compatibilitat amb la majoria de llenguatges de programació).

Com en l'anterior apartat, trobem que podem elegir entre diferents tipus de Frameworks gratuïts que puguin utilitzar MVC i que estiguin escrits en PHP, el llenguatge que utilitzarem [11]. Tot ells tenen una sèrie de pros en comú com hem comentat en la descripció dels avantatges d'utilitzar Frameworks en el paràgraf anterior. Per tant, ens hem centrat en els desavantatges que suposaria utilitzar-ne un o altre. De totes les opcions disponibles, en destaquem les que, probablement, puguin ser les més indicades per als nostres requeriments:

**Zend:** Seria un clar exemple de bon candidat a utilitzar. Però és un Framework gran, pesat, gasta molta memòria i té una gran quantitat d'inclusions (jQuery, java,...) que fan que consumeixi encara més recursos.

**Yii:** Tot i ser un Framework molt complet, el principal problema és el temps d'aprenentatge. Aprendre a utilitzar un Framework no necessàriament és quelcom trivial i aquest cas n'és un clar exemple. A més, la dependència ja intrínseca en els Frameworks (que fa que quan escrivim un programa per determinat Framework sigui impossible migrar a un altre sense haver de reescriure gairebé tot el codi) és, en Yii, encara més evident.

**Symfony:** Gran part de la velocitat d'aquest Framework es deu a un ús important de la memòria cau de manera que, quan s'està desenvolupant amb ell, tendeix a ser una mica carregós el procés de neteja de la memòria, més encara, al haver-se de realitzar cada cert temps. A més, els processos de Symfony utilitzen massa memòria.

**CakePHP:** És el Framework pel qual ens hem decidit. El seu principal problema és el possible excés de consultes, ja que en projectes en els quals el nombre de taules és molt elevat i hi ha moltes relacions, l'ORM integrat (Object-relational mapping, tècnica emprada per a convertir les dades de llenguatges de programació orientats a objecte en la seva representació en base de dades relacional) pot arribar a fer excessives consultes. Com que el nostre nombre previst de taules no serà excessivament gran, podem considerar que és un problema de qual podem minimitzar-ne els danys.

En el següent apartat, aprofundirem en la descripció del Framework CakePHP.

### 3.3.1. El Framework CakePHP

El **Framework CakePHP** està escrit, com indica el seu nom, amb el llenguatge PHP. És un software lliure i de codi obert que compta amb un conjunt propi de llibreries i classes. Està inspirat en un altre Framework per a aplicacions web anomenat Ruby on Rails.

Va ser creat a partir d'una versió mínima d'un Framework per a aplicacions en PHP, desenvolupat el 2005 per Michal Tatarynowicz. Més tard, publicaria el Framework sota llicència del MIT creant, així, una comunitat de desenvolupadors que ara mantenen el Framework sota el nom de CakePHP [12].

Des de llavors, CakePHP ha vist créixer considerablement la seva popularitat degut a una sèrie de factors que l'han convertit en un dels Frameworks més utilitzats en el desenvolupament d'aplicacions web.



Figura 6. Logo de CakePHP



Dels factors més destacats, podem dir que és una eina que té una corba d'aprenentatge relativament senzilla i que disposa d'una configuració i futur desplegament molt simple ja que, normalment ve inclòs dins d'un paquet de programari de l'estil **XAMPP** (que comentarem en el següent capítol de la memòria) que requereix simplement executar un instal·lador per a tenir-lo operatiu i en funcionament.

Un altre avantatge que trobarem és que es tracta d'un Framework que disposa d'una comunitat de desenvolupadors molt activa i que sempre està en constant expansió oferint actualitzacions, millores, exemples d'implementació i tot tipus de documentació.

CakePHP treballa, a més, seguint un patró de disseny de software bastant estès com és el MVC (Model – Vista - Controlador) que, com hem dit, serà el que utilitzarem. Al tractar-se d'un Framework, CakePHP ja disposa de les eines necessàries per permetre'ns una programació molt ben estructurada basada en un arbre de directoris predefinit que disposa els elements i els tracta com a Model, Vista o Controlador depenent del directori i les convencions que hem de seguir quan es creen (més endavant, comentarem aquestes convencions juntament amb l'arbre de directoris del Framework CakePHP, inclòs en el paquet d'aplicacions XAMPP).

### 3.4. El paquet XAMPP

Anem ara a veure aquest tipus de programari, molt popularitzat els últims anys, capaç d'encapsular el sistema de gestió de Base de Dades, un servidor web i empaquetar, a més, varis components amb els principals llenguatges de programació com els que hem definit abans, per a ser utilitzats de manera combinada i sota compatibilitats que assegurin un perfecte funcionament de tots aquests elements.

#### 3.4.1. Descripció

El Framework CakePHP ve inclòs en un paquet d'aplicacions anomenat XAMPP que disposa de tots els elements necessaris per a implementar l'aplicació web desitjada.



Figura 7. Logo del projecte XAMPP

Per entendre ràpidament quines característiques ens ofereix el paquet XAMPP tan sols cal analitzar les sigles del seu nom [13].

La primera lletra del nom del paquet es refereix a la compatibilitat en quan al sistema operatiu pel qual ha estat dissenyat. Per tant, LAMPP seria compatible amb Linux, WAMPP compatible amb Windows i, en el cas que ens ocupa, la X de XAMPP indica que és compatible amb qualsevol sistema operatiu. La resta d'inicials són comunes en totes les varietats del paquet.

La segona lletra, la A, es refereix al servidor HTTP que inclou, en aquest cas el servidor Apache, servidor que hem indicat amb anterioritat, que serà el que utilitzarem en el desenvolupament de l'aplicació.

Per acabar amb l'anàlisi dels components del paquet XAMPP comentarem el significat de les dues P's. En aquest cas, es refereixen als dos llenguatges que integra com són el PHP i el Pearl.

Nosaltres utilitzarem el PHP, llenguatge d'ús general originalment creat per al disseny web. És un dels primers llenguatges del tipus que es coneix com a "del costat del servidor". Aquest tipus de llenguatges funcionen interpretant un script (seqüència de comandes) en el servidor web per a generar pàgines HTML com a resposta (en el cas de CakePHP arxius amb extensió .ctp, que tenen un format que el propi Framework entén i mostra com si es tractés d'un HTML).

Finalment, ens falta indicar, que el Sistema Gestor de Base de Dades (SGBD) que utilitza el XAMPP és el que hem triat, el MySQL, un dels més utilitzats en el desenvolupament de bases de dades i del que en manejarem l'administració amb phpMyAdmin a través de pàgines web.

En conclusió, amb el paquet software XAMPP tenim els avantatges i els components suficients per a implementar la nostra aplicació web, ja que disposem de:

- Facilitat d'aprenentatge i d'instal·lació
- Compatibilitat amb els sistemes operatius principals (Windows, Linux i MacOS)
- Servidor Apache (el més utilitzat)

- SGBD MySQL (dels més populars)
- phpMyAdmin per a gestionar la Base de Dades.
- Implementació estructurada i organitzada (treballar amb Framework seguint MVC)
- Possibilitat d'afegir nous mòduls (com per exemple per treballar amb AJAX)

### 3.4.2. Funcionament de CakePHP dins XAMPP

Com hem comentat anteriorment, per al correcte maneig de l'arquitectura MVC en el Framework CakePHP dins del paquet XAMPP, hem de col·locar els elements en el directori corresponent del paquet tal i com marquen els convenis de funcionament [14]. Així doncs, hem de tenir en compte l'estructura de carpetes dins del directori principal de XAMPP, que resumim a la següent **Figura 8**:

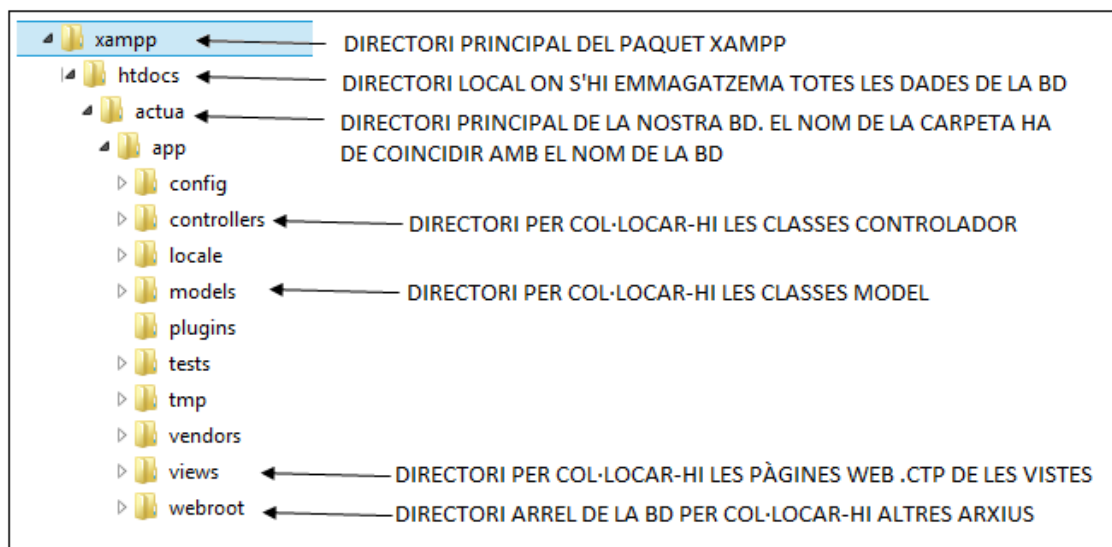


Figura 8. Vista de l'arbre de directoris principal de XAMPP per col·locar els elements de l'arquitectura MVC

Seguint aquesta pauta de directoris, el CakePHP pot treballar dins de XAMPP identificant els tres elements MVC (Model, Vista, Controlador). Això succeeix perquè la creació de les classes que necessitem són una extensió de les classes pròpies que venen definides per defecte: AppModel, AppController i AppView.

Llavors, per accedir a la vista de la pàgina principal de la nostra aplicació només cal escriure la direcció <http://localhost/biblioteca> (si la base de dades s'anomenés "biblioteca" i treballéssim des del servidor. Sinó, des del client, escriuríem la IP del servidor enlloc de "localhost").

Normalment, l'arxiu que conté el codi de la pàgina principal s'anomena "home.ctp" i ha d'estar col·locada dins la carpeta "views\pages" com la resta de pàgines que vulguem mostrar i que no utilitzin directament cap model o controlador.

D'altra banda, la manera d'accedir als mètodes que implementaran les accions del controlador d'un model determinat és escrivint a la barra de direccions del navegador el nom de la classe del model seguit de la barra invertida "\" i el nom del mètode del controlador. En alguns casos, es poden passar arguments al mètode del controlador mitjançant la pròpia barra de direccions. Si, per exemple, volem eliminar un client que té una id = 8 i el mètode "delete" està correctament implementat, accedirem al mètode del controlador des de la barra de direccions passant-li com a argument el número 8. És a dir: <http://localhost/biblioteca/client/delete/8>

A part d'això, a l'hora de crear les classes pròpies, s'han de seguir una sèrie de convencions definides pel Framework CakePHP, tan en el nom de les taules, com pels noms de les classes dels models, controladors i vistes [15]. Les descrivim a continuació:

#### CONVENCIONS EN MODELS I TAULES

**1) NOM DE LES TAULES I ELS MODELS:** Les taules han de tenir el nom en plural i el caràcter subratllat (underscored) si volem utilitzar varies paraules: Per exemple, una taula anomenada "bon\_clients". Pels noms de les classes dels models, s'utilitza el singular amb *CamelCased* (treure l'espai entre paraules i que aquestes comencin amb majúscules). Per tant, la classe model de l'anterior taula seria "BonClient". Aquesta classe es guardarà en un fitxer anomenat "bonclient.php" a la carpeta "models".

**2) NOM DE LES TAULES I ELS MODELS II:** Quan les taules són fruit d'una relació N a N, s'utilitzarà el caràcter subratllat (underscored) per separar els noms en plural de les dues classes dels models relacionats, sempre en ordre alfabètic. Per exemple, una taula que relacioni un model anomenat "Excursio" amb un d'anomenat "Client" s'anomenarà "clients\_excursions".

**3) ELS CAMPS DE LES TAULES:** Per convenció, hi ha d'haver-hi un camp anomenat "id" que serà la clau primària de cada taula, normalment un número amb autoincrement. A més, si un camp

té per nom més d'una paraula, aquest s'escriurà en minúscula i caràcter subratllat. Per exemple, "primer\_cognom".

**4) CLAUS EXTERNES:** Els camps que es relacionen amb una altra taula, són reconeguts per defecte si s'escriu utilitzant el singular de la taula seguit de caràcter subratllat i "id". Així, un camp "client\_id" serà reconegut com clau externa d'una taula que estigui relacionada amb una d'altra anomenada "clients".

### CONVENCIONS EN CONTROLADORS

**1) NOM DEL CONTROLADOR:** El de la classe ha de ser en plural, CamelCased i afegint "Controller" al final. Per tant, una classe controlador per a una classe model "ClientsExcursions" s'anomenarà ClientsExcursionsController i es guardarà a la carpeta "controllers" amb el nom clients\_excursions\_controller.php

**2) MÈTODE INDEX:** Si no es diu el contrari, quan des de la direcció del navegador no especifiquem cap mètode, s'executarà el mètode obligatori index(). Això vol dir que és el mateix escriure <http://localhost/biblioteca/client> que <http://localhost/biblioteca/client/index>

### CONVENCIONS EN VISTES

**1) NOM DE LES VISTES:** Les vistes es classifiquen en carpetes que es diuen exactament igual que les taules a les que pertanyen i dins la carpeta "views". Dins de cada carpeta es crearà un fitxer .ctp que portarà per nom del mètode del controlador al que fan referència (amb underscoring si s'escau). Per exemple. El mètode add() del controlador "Client", buscarà l'arxiu de la vista que haurà d'estar guardat a la ruta de carpeta "\views\clients\add.ctp" quan s'escriui pel navegador <http://localhost/biblioteca/clients/add>



## 4. Descripció de l'aplicació

Com hem explicat en el capítol **1.2. ACTUASERVEIS SCP**, la nostra aplicació ha d'oferir una solució per gestionar la base de dades d'aquesta empresa. ACTUASERVEIS SCP s'encarrega, mitjançant treballadors, d'oferir serveis de varis tipus a persones amb un diferent grau de discapacitat.

Aquestes persones, anomenades usuaris, es classifiquen depenent d'aquest grau de discapacitat. Podem trobar els següents tipus d'usuaris:

- **Tercera edat:** Usuaris sense cap tipus de discapacitat més enllà del que suposen les limitacions típiques de la seva avançada edat.
- **Discapacitats:** usuaris amb algun tipus de deficiència sigui física, intel·lectual o sensorial. Estan, a la vegada, organitzats en tres subgrups segons el grau de d'assistència que necessitin:
  - Intermitents: amb una atenció just quan es necessiti, no sempre necessària.
  - Limitats extens: amb una atenció regular i continua.
  - Generalitzats: amb una atenció constant i d'alta intensitat.

A més, trobem els Intermitents Sènior i els Limitats Extens Sènior que mantenen les característiques descrites en els seus corresponents subgrups però que, a més, estan en una avançada edat.

- **Malalts Mentals:** Usuaris que pateixen d'alguna malaltia mental com, per exemple, serien la depressió, ansietat o certs tipus de fòbies severes.

Per una altra banda, els serveis de l'empresa que haurà de controlar la nostra aplicació, són dos:

**Sortides:** Organitzades per a grups d'usuaris segons els tipus de necessitats que acabem d'anomenar. Solen ser activitats que duren varis dies o, fins i tot, setmanes i tutelades per varis monitors.

**Acompanyaments:** Serveis d'atenció personalitzada que es duen a terme entre un sol treballador i un sol usuari.

## 4.1. Descripcions prèvies

Per tal de dissenyar els diagrames corresponents de la nostra aplicació, anem a definir els següents conceptes:

- \* **USUARI:** Persona amb un grau variable de discapacitat física o intel·lectual, que participa en les activitats d'ACTUASERVEIS que controlarem des de l'aplicació: Sortides i Acompanyaments.
- \* **TREBALLADOR:** Monitor responsable dels usuaris en les activitats organitzades per l'empresa.
- \* **CENTRES:** Entitats que demanden sortides o acompanyaments per als seus usuaris. En aquesta categoria cal tenir en compte també que pot tractar-se d'usuaris particulars.
- \* **LLOCS:** Espais físics on es poden dur a terme les SORTIDES (cases rurals, els propis centres, llocs de particulars,...)
- \* **SORTIDES:** Organitzades per a grups mitjanament grans, de fins a 30 usuaris. Segons els número d'usuaris i el grau d'atenció necessari, es necessitarà un número de treballadors o un altre.
- \* **ACOMPANYAMENT:** Servei individual que realitza un treballador per a un usuari concret (Vetlla hospitalària, companyia, urgències,...)
- \* **ADMINISTRADOR DE L'APLICACIÓ:** Actor principal del nostre projecte. És l'encarregat de portar l'administració i la gestió de la l'empresa a través de l'aplicació web i la Base de Dades.

## 4.2. Diagrama Casos d'Ús

Analitzant els casos d'ús de la nostra aplicació web, veiem que podem dividir-los en dos subsistemes, per tal de satisfer els requeriments descrits en el capítol **2. Anàlisi de requeriments**.

Per una banda, el primer subsistema que anomenarem CRUD, és l'encarregat de contenir totes les operacions bàsiques referents als elements que compondran la nostra base de dades. Aquí hi entra la creació, edició, consulta i eliminació de tots els elements de l'aplicació: tan dels llocs físics (Centres i Allotjaments), com del individus que hi interactuen (Treballadors i Usuaris).

Aquest primer subsistema ve descrit directament per l'assoliment dels requeriments funcionals **RF1** (Controlar tots els elements que componen l'aplicació de manera específica (CRUD)), **RF2** (Disposar d'un sistema de classificació dels usuaris i les seves necessitats concretes), **RF3**



(Control, seguiment i valoració dels treballadors de l'empresa) i **RF4** (Control i seguiment de les entitats implicades com els centres o associacions d'usuaris).

Per altra banda, l'administrador ha de poder portar un control dels serveis que presta l'empresa. Des de la creació d'una Sortida amb totes les necessitats dels seus Usuaris i llista de Treballadors, fins a un Acompanyament individual per part d'un Treballador i el càlcul de les facturacions en quilometratges.

En aquest cas, el segon subsistema ens permet assolir els requeriments funcionals **RF5** (Definició i acotament dels diferents tipus de servei: Sortides i Acompanyaments) i **RF6** (Control i càlcul de les despeses de Quilometratge dels Acompanyaments).

Veiem tot el que acabem d'explicar, resumit a la **Figura 9**:

### DIAGRAMA DE CASOS D'ÚS

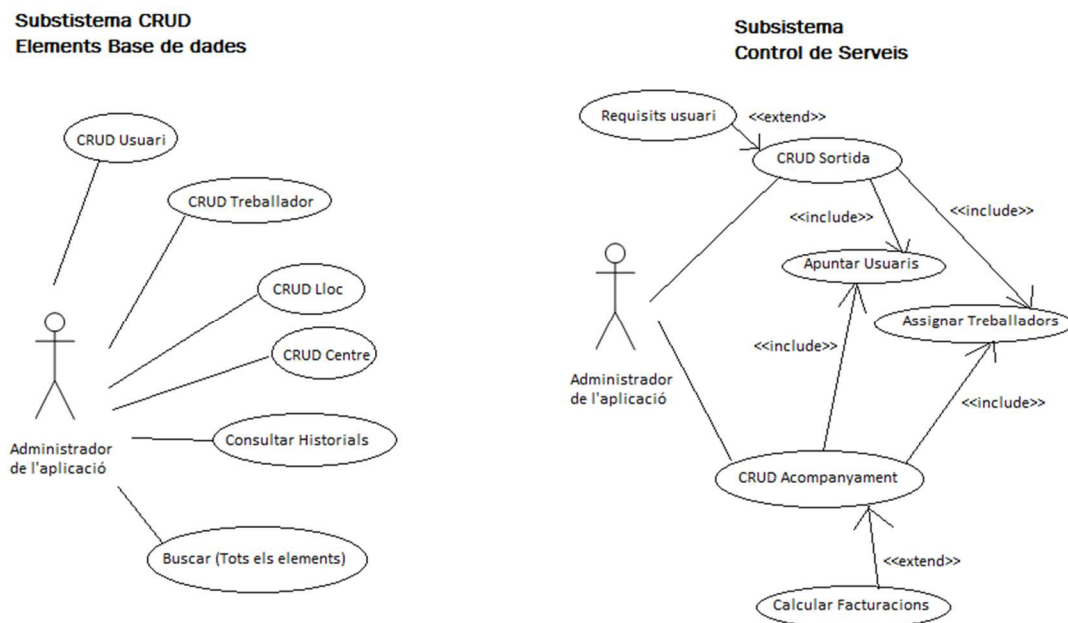


Figura 9. Diagrama casos d'ús

A continuació desglossem els casos d'ús fent-ne una descripció més detallada.

#### 4.2.1 Casos d'ús del subsistema CRUD

CAS D'ÚS: Donar d'alta un element			
Versió	1.0	Data	JUNY 2014
Autors	Autor del projecte		
Descripció	Aquest cas d'ús descriu com l'administrador de l'aplicació crea cadascun dels elements que componen el nostre sistema: CENTRES, LLOCS, USUARIS i TREBALLADORS. Depenent de cada element s'introdueixen unes dades o altres.		
Actors	Administrador de l'aplicació		
Precondició			
Flux Principal	<ol style="list-style-type: none"> <li>1. L'Administrador insereix les dades al formulari.</li> <li>2. El sistema comprova que les dades tenen format correcte.</li> <li>3. El sistema crea i guarda el nou element.</li> </ol>		
Fluxos alternatius	<u>Dades incorrectes</u> Si el sistema detecta alguna dada errònia (per exemple el format d'un DNI), es mostra un missatge d'error en el propi formulari demanant que es torni a entrar les dades.		
Postcondició	Les dades introduïdes són correctes		
Requeriments No Funcionals	<u>RNF1</u> : Funcionament fàcil i intuïtiu. <u>RNF2</u> : Estructura conseqüent alhora d'integrar els elements en els menús de l'aplicació.		
Prioritat	Normal		

CAS D'ÚS: Modificar un element			
Versió	1.0	Data	JUNY 2014
Autors	Autor del projecte		
Descripció	Aquest cas d'ús descriu com l'administrador de l'aplicació modifica algun dels elements que hagi estat creats en el nostre sistema: CENTRES, LLOCS, USUARIS o TREBALLADORS.		
Actors	Administrador de l'aplicació		
Precondició	Que l'element hagi estat creat correctament		
Flux Principal	<ol style="list-style-type: none"> <li>1. l'Administrador modifica les dades del formulari.</li> <li>2. El sistema comprova que les dades tenen format correcte.</li> <li>3. El sistema guarda els canvis en l'element.</li> </ol>		
Fluxos alternatius	<u>Dades incorrectes</u> Si el sistema detecta alguna dada errònia (per exemple el format d'un DNI), es mostra un missatge d'error en el propi formulari demanant que es torni a entrar les dades.		
Postcondició	Les dades introduïdes són correctes		
Requeriments No Funcionals	<u>RNF1</u> : Funcionament fàcil i intuïtiu. <u>RNF2</u> : Estructura conseqüent alhora d'integrar els elements en els menús de l'aplicació.		
Prioritat	Normal		

CAS D'ÚS: Eliminar un element			
Versió	1.0	Data	JUNY 2014
Autors	Autor del projecte		
Descripció	Aquest cas d'ús descriu com l'administrador de l'aplicació elimina algun dels elements que hagi estat creats en el nostre sistema: CENTRES, LLOCS, USUARIS o TREBALLADORS.		
Actors	Administrador de l'aplicació		
Precondició	Que l'element hagi estat creat correctament		
Flux Principal	<ol style="list-style-type: none"> <li>1. l'Administrador clica en l'opció d'eliminar.</li> <li>2. El sistema comprova que l'element estigui creat.</li> <li>3. El sistema elimina l'element.</li> </ol>		
Fluxos alternatius			
Postcondició	L'element s'elimina		
Requeriments No Funcionals	<u>RNF1</u> : Funcionament fàcil i intuïtiu. <u>RNF2</u> : Estructura conseqüent alhora d'integrar els elements en els menús de l'aplicació.		
Prioritat	Normal		

CAS D'ÚS: Buscar un element			
Versió	1.0	Data	JUNY 2014
Autors	Autor del projecte		
Descripció	Aquest cas d'ús descriu com l'administrador de l'aplicació busca algun dels elements que hagi estat creats en el nostre sistema: CENTRES, LLOCS, USUARIS o TREBALLADORS.		
Actors	Administrador de l'aplicació		
Precondició	Que l'element hagi estat creat correctament		
Flux Principal	<ol style="list-style-type: none"> <li>1. l'Administrador escriu la paraula clau.</li> <li>2. El sistema busca l'element i el mostra si existeix.</li> </ol>		
Subfluxos			
Fluxos alternatius	<u>Element inexistent</u> Si el sistema detecta que l'element no existeix, mostra un missatge per pantalla i ofereix a l'administrador la possibilitat de tornar a introduir una paraula clau diferent.		
Postcondició	Si existeix, es mostren les dades de l'element (incloent un petit historial dels últims serveis en que ha estat involucrat)		
Requeriments No Funcionals	<u>RNF1</u> : Funcionament fàcil i intuïtiu. <u>RNF2</u> : Estructura conseqüent alhora d'integrar els elements en els menús de l'aplicació. <u>RNF3</u> : Maximitzar l'accés i comunicació dels diversos elements mitjançant els enllaços web.		
Prioritat	Normal		

#### 4.2.2. Casos d'ús del subsistema Control de serveis

CAS D'ÚS: Crear una sortida			
Versió	1.0	Data	JUNY 2014
Autors	Autor del projecte		
Descripció	Aquest cas d'ús descriu com l'administrador de l'aplicació crea una nova sortida amb les dades necessàries. Aquesta acció inclou, després, apuntar els usuaris i assignar-ne els treballadors. Finalment, per cada usuari es defineixen els requeriments individuals per a aquella sortida.		
Actors	Administrador de l'aplicació		
Precondició			
Flux Principal	<ol style="list-style-type: none"> <li>1. L'Administrador dona d'alta la sortida.</li> <li>2. El sistema comprova que les dades són correctes.</li> <li>3. El sistema crea la sortida.</li> <li>4. L'Administrador apunta els Usuaris.</li> <li>5. El sistema crea la relació corresponent entre elements.</li> <li>6. Per cada usuari, en defineix els requeriments.</li> <li>7. El sistema crea la relació corresponent entre elements.</li> <li>8. L'Administrador assigna els Treballadors</li> <li>9. El sistema crea la relació corresponent entre elements.</li> </ol>		
Subfluxos			
Fluxos alternatius	<p><u>Element no disponible</u></p> <p>El sistema no permet que un treballador o un usuari estiguin en dues sortides al mateix temps.</p>		
Postcondició	La sortida es crea, els usuaris s'apunten amb els requisits de cadascun i els treballadors hi són assignats		
Requeriments No Funcionals	<p><u>RNF1</u>: Funcionament fàcil i intuïtiu</p> <p><u>RNF2</u>: Estructura conseqüent alhora d'integrar els elements en els menús de l'aplicació</p> <p><u>RNF3</u>: Maximitzar l'accés i comunicació dels diversos elements mitjançant els enllaços web.</p>		
Prioritat	Normal		

CAS D'ÚS: Modificar una sortida			
Versió	1.0	Data	JUNY 2014
Autors	Autor del projecte		
Descripció	Aquest cas d'ús descriu com l'administrador de l'aplicació modifica quelcom d'una sortida creada. Aquesta acció inclou, des de canviar alguna dada de la pròpia sortida fins a desapuntar un usuari o desassignar-ne algun treballador. També es poden modificar els requisits de l'usuari		
Actors	Administrador de l'aplicació		
Precondició	Que la sortida hagi estat creada		
Flux Principal	<ol style="list-style-type: none"> <li>1. L'Administrador modifica la sortida.</li> <li>2. El sistema comprova que les dades són correctes.</li> <li>3. El sistema modifica la sortida.</li> <li>4. L'Administrador desapunta els Usuaris.</li> <li>5. El sistema elimina la relació corresponent entre elements.</li> <li>6. Per cada usuari, pot modificar els requeriments.</li> <li>7. El sistema modifica els requeriments.</li> <li>8. L'Administrador desassigna els Treballadors</li> <li>9. El sistema elimina la relació corresponent entre elements.</li> </ol>		
Subfluxos			
Fluxos alternatius	<p><u>Nova disponibilitat</u></p> <p>Quan es desapunta o es desassigna algun element, apareix un missatge de confirmació i aquest element passa a estar disponible per a altres sortides dels mateixos dies.</p>		
Postcondició	La sortida es modifica, els usuaris es desapunten, s'eliminen els requisits de cadascun i els treballadors són desassignats.		
Requeriments No Funcionals	<p><u>RNF1</u>: Funcionament fàcil i intuïtiu.</p> <p><u>RNF2</u>: Estructura conseqüent alhora d'integrar els elements en els menús de l'aplicació.</p> <p><u>RNF3</u>: Maximitzar l'accés i comunicació dels diversos elements mitjançant els enllaços web.</p>		
Prioritat	Normal		

CAS D'ÚS: Eliminar una sortida			
Versió	1.0	Data	JUNY 2014
Autors	Autor del projecte		
Descripció	Aquest cas d'ús descriu com l'administrador de l'aplicació elimina una sortida creada. En conseqüència, la relació dels usuaris i els seus requisits i els treballadors amb la sortida s'eliminen.		
Actors	Administrador de l'aplicació		
Precondició	Que la sortida hagi estat creada		
Flux Principal	<ol style="list-style-type: none"> <li>1. L'Administrador elimina la sortida.</li> <li>2. El sistema comprova que la sortida existeix.</li> <li>3. El sistema elimina la sortida.</li> <li>4. El sistema elimina la relació corresponent entre elements.</li> </ol>		
Subfluxos			
Fluxos alternatius	<p><u>Nova disponibilitat</u></p> <p>Quan s'elimina una sortida, apareix un missatge de confirmació i els elements relacionats a ella passen a estar disponibles per a altres sortides dels mateixos dies.</p>		
Postcondició	La sortida es modifica, els usuaris es desapunten, s'eliminen els requisits de cadascun i els treballadors són desassignats.		
Requeriments No Funcionals	<p><u>RNF1</u>: Funcionament fàcil i intuïtiu.</p> <p><u>RNF2</u>: Estructura conseqüent alhora d'integrar els elements en els menús de l'aplicació.</p> <p><u>RNF3</u>: Maximitzar l'accés i comunicació dels diversos elements mitjançant els enllaços web.</p>		
Prioritat	Normal		

CAS D'ÚS: Crear un acompanyament			
Versió	1.0	Data	JUNY 2014
Autors	Autor del projecte		
Descripció	Aquest cas d'ús descriu com l'administrador de l'aplicació crea un nou acompanyament amb les dades necessàries.		
Actors	Administrador de l'aplicació		
Precondició			
Flux Principal	<ol style="list-style-type: none"> <li>1. L'Administrador omple el formulari de l'acompanyament.</li> <li>2. El sistema comprova que les dades són correctes.</li> <li>3. El sistema crea la l'acompanyament.</li> <li>4. El sistema crea la relació corresponent entre elements.</li> </ol>		
Subfluxos			
Fluxos alternatius			
Postcondició	L'acompanyament es crea amb les seves dades i la corresponent tupla usuari / treballador		
Requeriments No Funcionals	<p><u>RNF1</u>: Funcionament fàcil i intuïtiu</p> <p><u>RNF2</u>: Estructura conseqüent alhora d'integrar els elements en els menús de l'aplicació</p> <p><u>RNF3</u>: Maximitzar l'accés i comunicació dels diversos elements mitjançant els enllaços web.</p>		
Prioritat	Normal		



CAS D'ÚS: Modificar un acompanyament			
Versió	1.0	Data	JUNY 2014
Autors	Autor del projecte		
Descripció	Aquest cas d'ús descriu com l'administrador de l'aplicació modifica les dades d'un acompanyament creat.		
Actors	Administrador de l'aplicació		
Precondició	Que l'acompanyament hagi estat creat		
Flux Principal	<ol style="list-style-type: none"> <li>1. L'Administrador modifica el formulari d'acompanyament.</li> <li>2. El sistema comprova que les dades són correctes.</li> <li>3. El sistema modifica l'acompanyament.</li> <li>4. El sistema modifica, si cal, la relació entre elements.</li> </ol>		
Subfluxos			
Fluxos alternatius			
Postcondició	L'acompanyament es modifica.		
Requeriments No Funcionals	<p><u>RNF1</u>: Funcionament fàcil i intuïtiu.</p> <p><u>RNF2</u>: Estructura conseqüent alhora d'integrar els elements en els menús de l'aplicació.</p> <p><u>RNF3</u>: Maximitzar l'accés i comunicació dels diversos elements mitjançant els enllaços web.</p>		
Prioritat	Normal		

CAS D'ÚS: Eliminar un acompanyament			
Versió	1.0	Data	JUNY 2014
Autors	Autor del projecte		
Descripció	Aquest cas d'ús descriu com l'administrador de l'aplicació elimina un acompanyament creat. En conseqüència, la relació entre usuari i treballador també s'elimina.		
Actors	Administrador de l'aplicació		
Precondició	Que la sortida hagi estat creada		
Flux Principal	<ol style="list-style-type: none"> <li>1. L'Administrador elimina l'acompanyament.</li> <li>2. El sistema comprova que l'acompanyament existeix.</li> <li>3. El sistema elimina l'acompanyament.</li> <li>5. El sistema elimina la relació corresponent entre elements.</li> </ol>		
Subfluxos			
Fluxos alternatius	<u>Nova disponibilitat</u> Quan s'elimina una acompanyament, apareix un missatge de confirmació.		
Postcondició	L'acompanyament s'elimina.		
Requeriments No Funcionals	<u>RNF1</u> : Funcionament fàcil i intuïtiu. <u>RNF2</u> : Estructura conseqüent alhora d'integrar els elements en els menús de l'aplicació. <u>RNF3</u> : Maximitzar l'accés i comunicació dels diversos elements mitjançant els enllaços web.		
Prioritat	Normal		

CAS D'ÚS: Consultar facturacions calculades			
Versió	1.0	Data	JUNY 2014
Autors	Autor del projecte		
Descripció	Aquest cas d'ús descriu com l'administrador de l'aplicació consulta les facturacions mensuals de cada treballador, calculades a partir dels quilometratges anotats al crear un acompanyament		
Actors	Administrador de l'aplicació		
Precondició	Que l'acompanyament hagi estat creat		
Flux Principal	1. L'Administrador consulta les facturacions d'un treballador 2. El sistema les calcula a partir de la informació entrada.		
Subfluxos			
Fluxos alternatius			
Postcondició	L'administrador veu les facturacions mensuals del treballador		
Requeriments No Funcionals	<u>RNF1</u> : Funcionament fàcil i intuïtiu. <u>RNF2</u> : Estructura consegüent alhora d'integrar els elements en els menús de l'aplicació. <u>RNF3</u> : Maximitzar l'accés i comunicació dels diversos elements mitjançant els enllaços web.		
Prioritat	Normal		

### 4.3. Diagrama de Classes

Abans de veure el diagrama de classes voldríem comentar que, per a entendre el funcionament intern del Framework, vàrem fer un anàlisi de les classes ja implementades que utilitzava per a funcionar i mostrar, segons el cas, la informació de manera automàtica en pantalla (veure capítol **3.4.2. Funcionament de CakePHP dins XAMPP**). Vàrem veure la manera com aquestes classes, que ja vénen creades pel propi Framework, es relacionaven amb les que havíem creat nosaltres per al projecte.

El resultat de l'anàlisi d'aquestes relacions es pot veure en el següent diagrama de classes (**Figura 10**):

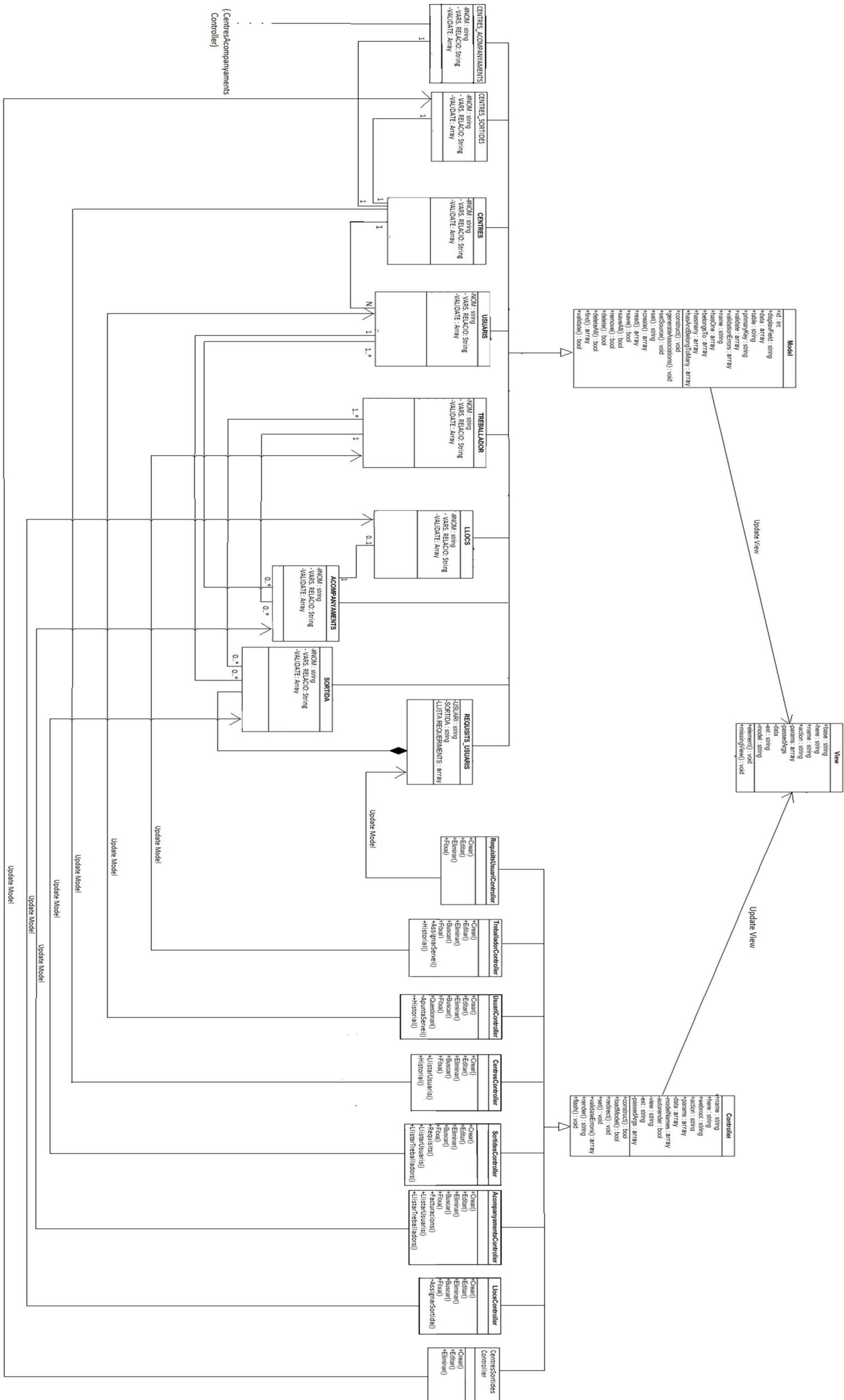


Figura 10. Diagrama de classes del Framework i l'aplicació Web

## 5. Disseny de l'aplicació

En aquest capítol, primerament mostrarem el diagrama d'Entitat – Relació i el Model Relacional de Taules que hem obtingut de tota la informació explicada fins a aquest punt. Explicarem detalladament tot el referent a les característiques dels camps de les taules i, finalment, mostrarem les vistes de l'aplicació a mode d'exemple per a il·lustrar cadascun dels casos d'ús definits en el capítol **4.2. Casos d'Ús**.

### 5.1. Diagrama Entitat-Relació

Analitzant el diagrama de classes anterior i els requeriments de l'aplicació descrits abans hem dissenyat el diagrama Entitat – Relació que mostrem a la següent **Figura 11**:

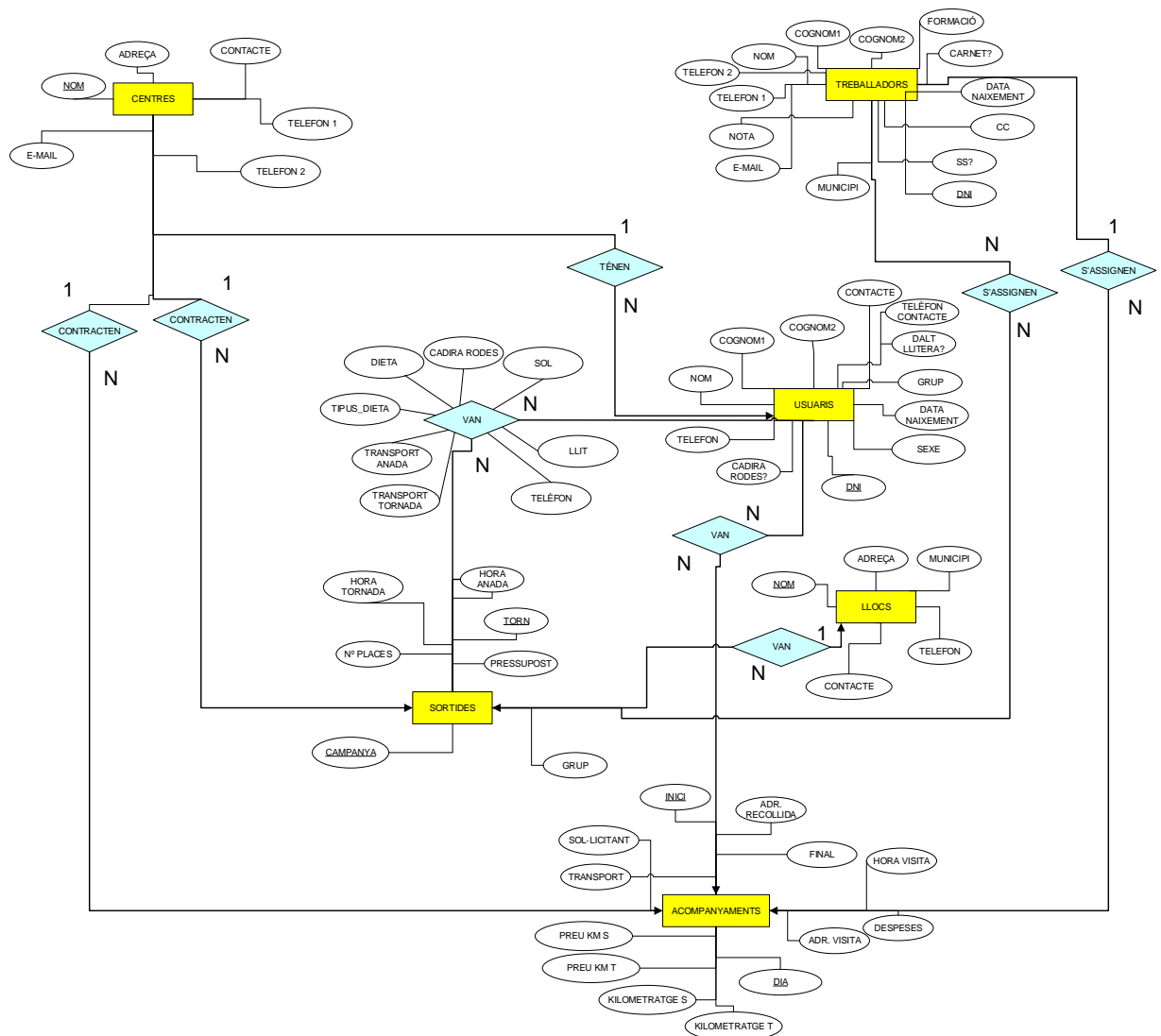


Figura 11. Diagrama ER

Anem ara a definir quines són les entitats i les relacions que hem obtingut:

**ENTITATS:**

\* **USUARIS:** Persona amb cert grau de discapacitat que pot contractar tan un servei d'acompanyament com apuntar-se a una sortida. La informació que en volem controlar:

NOM I COGNOMS: Nom complet de l'usuari.

CONTACTE: El nom del representant legal de l'usuari (per exemple un familiar directe).

TELÈFON DEL CONTACTE: Telèfon de contacte del representant legal.

DALT LLITERA: Per indicar si l'usuari pot ser capaç de dormir en la llitera de dalt, que normalment es troba a certa alçada, sense que s'hagi de patir per la seva seguretat física.

GRUP: Sol definir el tipus i grau de discapacitat. Per raons d'organització i practicitat, l'empresa sol agrupar usuaris del mateix grup sempre que sigui possible.

DATA NAIXEMENT: El dia que va néixer l'usuari.

SEXE: Si es tracta d'un home o d'una dona.

DNI: Número del document nacional d'identitat.

CADIRA DE RODES: Si l'usuari requereix d'una cadira de rodes per desplaçar-se.

TELÈFON: En alguns casos d'usuaris amb força autonomia, aquests disposen de telèfon mòbil personal.

\* **TREBALLADORS:** Persona contractada per l'empresa i encarregada de la vetlla dels usuaris. Pot assignar-se tan a un acompanyament com a una sortida.

NOM I COGNOMS: Nom complet del treballador

FORMACIÓ: Estudis que ha realitzat

CARNET: Si disposa de carnet de conduir

DATA NAIXEMENT: El dia que va néixer el treballador

CC: Número del Compte Corrent bancari

SS: Número de la Seguretat Social

DNI: Número del document nacional d'identitat

MUNICIPI: Municipi de residència del treballador

E-MAIL: Direcció electrònica.

NOTA: Mesura per puntuar el rendiment del treballador. El valor pot ser de 0 a 10.

TELEFON 1: Telèfon de contacte

TELÈFON 2: En alguns casos, un segon telèfon de contacte per si no es pot contactar a través del TELÈFON 1.

\* **CENTRES:** Institució que s'encarrega d'organitzar activitats per als usuaris i que contacten amb l'empresa ACTUASERVEIS SCL per tal de demanar-ne els serveis. També considerarem un centre anomenat "Particular" per aquells usuaris que no pertanyin a cap centre i contractin algun servei a través d'un particular. En volem guardar la següent informació:

NOM: El nom complet del centre

ADREÇA: Adreça física o legal.

CONTACTE: Nom de la persona que contactarà amb ACTUASERVEIS SCL en nom del centre.

TELÈFON 1: Telèfon de contacte.

TELÈFON 2: En alguns casos, un número de telèfon alternatiu.

E-MAIL: Direcció electrònica del centre.

\* **LLOCS:** Es refereix a lloc, si n'hi ha, on un grup de sortida s'allotjarà o farà la majoria d'activitats.

NOM: Nom del lloc (per exemple, el nom de la casa de colònies).

ADREÇA: adreça física o legal.

MUNICIPI: Municipi on es troba.

TELÈFON: Telèfon de contacte

CONTACTE: Nom de la persona del lloc que contactarà amb ACTUASERVEIS SCL.

\* **SORTIDES:** Un dels principals serveis que ofereix ACTUASERVEIS SCL. Consisteixen en un grup de fins a 30 usuaris i els monitors necessaris per satisfer les seves necessitats. El número

d'aquests, varia en funció del número i necessitats dels usuaris que el componen. Solen organitzar-se en tres èpoques molt concretes de l'any com són Nadal, Setmana Santa i Estiu. Tot i això, excepcionalment, es poden organitzar sortides en altres dates. Volem guardar-ne:

**TORN:** O el nom de la sortida. Sol portar el nom de la població on es va i, addicionalment, se'ls hi assigna una lletra en ordre alfabètic que representa el número de sortida que hi ha en aquelles dates en concret. Per exemple, un segon grup que vagi a Tarragona per Setmana Santa tindria el nom de TARRAGONA B. No s'ha de confondre amb el l'entitat LLOC ja que, per exemple, a TARRAGONA el grup A pot anar a un lloc i el grup B pot anar a un altre.

**DATA ANADA I TORNADA:** Els dies on comença i acaba la sortida.

**NÚMERO DE PLACES:** Quantitat d'usuaris apuntats a la sortida.

**CAMPANYA:** Si es tracta d'una sortida de Nadal, Setmana Santa, Estiu o altres dates.

**GRUP:** O també, anomenat per l'empresa, col·lectiu. És el Nom del tipus d'usuari que més abunda en el grup. Se solen agrupar els usuaris depenent del grau de discapacitat però, en ocasions, per necessitats organitzatives, s'ajunten usuaris de grups diferents. Per exemple, pot haver-hi una sortida amb usuaris Discapacitats Intermittents i alhora amb usuaris Discapacitats Limitats Extens.

**PRESSUPOST:** Per casos d'emergència o imprevistos, es sol entregar una quantitat variable d'euros als treballadors que hauran de justificar a la tornada en el cas que n'hagin de fer ús.

\* **ACOMPANYAMENTS:** És l'altre principal servei que ofereix l'empresa. En aquest cas, es tracta d'una activitat on només hi ha implicat un treballador i un usuari. Pot tractar-se d'un servei de poques hores o de tota una jornada laboral. Per exemple, es pot tractar d'una vetlla hospitalària o d'un acompanyament a una visita al metge o, simplement, per oci. Volem controlar-ne:

**TIPUS:** De quin tipus d'acompanyament es tracta. Pot ser: Acompanyament, Acompanyament mèdic, Acompanyament Oci, Urgència o Vetlla Hospitalària.

**DIA:** Dia on comença el servei.

**INICI:** Hora concreta on començar el servei per part del treballador.

**FINAL:** Hora concreta on acaba el servei per part del treballador.



ADREÇA DE RECOLLIDA: Si existeix, adreça on s'ha de reunir el treballador amb l'usuari per començar el servei.

ADREÇA DE VISITA: Adreça on s'efectuarà el servei.

HORA DE VISITA: Si es tracta, per exemple, d'un acompanyament mèdic és l'hora de visita que té l'usuari amb el metge.

DESPESES: Costos extraordinaris relacionats directa o indirectament amb el servei. Per exemple, si s'ha de comprar una ampolla d'aigua per l'usuari o el treballador.

TRANSPORT: Tant d'anada com de tornada ja que sempre és el mateix. Pot ser Anar a peu, Transport públic, Taxi o Cotxe particular.

KILOMETRATGE SERVEI: Número de kilòmetres, si s'escau, que s'ha fet durant el servei si s'ha utilitzat com a transport el cotxe particular del treballador.

KILOMETRATGE TREBALLADOR: Número de kilòmetre, si s'escau, que ha fet el treballador per desplaçar-se, amb el seu cotxe, cap a realitzar el servei.

PREU KILOMETRATGE SERVEI: Preu que paga l'empresa al treballador per cada kilòmetre que realitza de servei. Hi ha un preu fix estipulat per l'empresa, però es pot canviar per raons concretes.

PREU KILOMETRATGE TREBALLADOR: Igual que l'anterior, però per cada kilòmetre que ha fet el treballador per desplaçar-se, amb el seu cotxe, cap a realitzar el servei.

Cal indicar en aquest punt que, a petició de l'empresa, cadascuna d'aquestes entitats definides haurà de tenir un apartat destinat a poder escriure-hi algun tipus d'observació.

#### **RELACIONS:**

- \* **CENTRES – USUARIS:** Relació 1 a N, que indica que un centre pot tenir varis usuaris.
- \* **CENTRES – SORTIDES:** Relació 1 a N, que indica que cada centre pot contractar varies sortides.
- \* **CENTRES – ACOMPANYAMENTS:** Relació 1 a N, que indica que cada centre pot contractar varis acompanyaments.
- \* **USUARIS – SORTIDES:** Relació N a N, que indica que varis usuaris alhora es poden apuntar a varies sortides.

En aquesta relació hem d'indicar que, per cada usuari apuntat a una sortida, necessitarem saber-ne una sèrie de requisits en aquells casos on sigui necessari:

DIETA: Si l'usuari segueix alguna dieta específica.

TIPUS DIETA: En cas de seguir una dieta, s'indica quina. Per exemple: Diabètica, celíaca,...

SOL: Indica si l'usuari té prou autonomia com per anar sol en un moment determinat a realitzar alguna acció que el separi momentàniament del grup.

TRANSPORT ANADA: Si l'usuari compta amb transport d'anada cap al punt de reunió de la sortida el dia que aquesta comença.

TRANSPORT TORNADA: El mateix per al dia de tornada.

A més, es pren de nou la següent informació que l'empresa ja té de l'usuari des de que es va inscriure al sistema. Són característiques de l'usuari que, de manera temporal, poden haver canviat del que podríem anomenar el seu "estat normal". Per exemple si l'usuari ha patit algun tipus de lesió que l'obliga a traslladar-se amb cadira de rodes, ha perdut el telèfon que utilitza normalment i el seu responsable n'hi entrega un de nou,... Són les característiques següents:

LLIT: Si pot dormir a la llitera de dalt.

CADIRA RODES: Si necessita cadira de rodes per desplaçar-se.

TELÈFON: El número de telèfon que, si és el cas, portarà l'usuari durant la sortida.

\* **USUARIS – ACOMPANYAMENTS:** Relació 1 a N, que indica que un sol usuari pot apuntar-se a varis acompanyaments.

\* **TREBALLADORS – SORTIDES:** Relació N a N, que indica que varis treballadors alhora poden assignar-se a vàries sortides.

\* **TREBALLADORS – ACOMPANYAMENTS:** Relació 1 a N, que indica que un sol treballador pot assignar-se a varis acompanyaments.

\* **SORTIDES – LLOCS:** Relació 1 a 1, que indica que, a cada sortida, es va a algun lloc a allotjar-se.

## 5.2. Model Relacional de Taules

Del diagrama ER anterior, hem seguit el procés per a la creació i definició de cadascuna de les taules que necessitem per a contenir la informació i la relació que hi ha entre elles, per tal de cobrir tots els requeriments definits anteriorment. A continuació, mostrem a la **Figura 12**, el model relacional de taules obtingut:

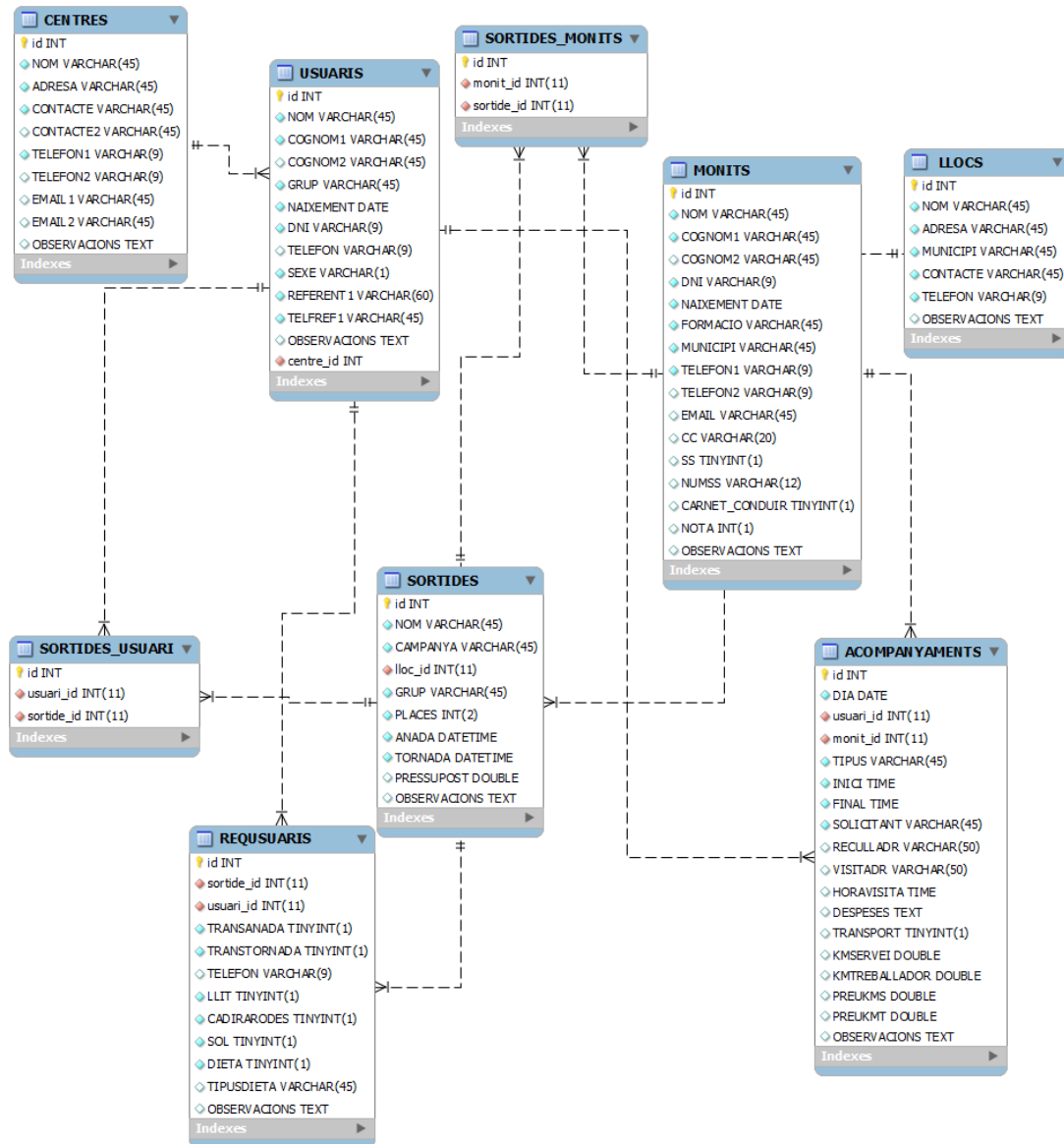


Figura 12. Model Relacional de Taules

Com ja s'ha explicat en l'apartat **3.4.2. Funcionament de CakePHP dins XAMPP**, les convencions de CakePHP ens exigeixen com a clau primària un camp numèric (enter de ) amb autoincrement anomenat "id" per a cadascuna de les taules creades. Paral·lelament, i per garantir la integritat

de les dades, hem assignat la restricció UNIQUE a aquell camp o conjunt de camps de cada taula que preveiem que serien la nostra clau primària i que definiria inequívocament cada registre introduït a la base de dades.

Anem ara a veure, més detalladament, les característiques dels camps de cada taula. S'especifica el tipus del camp, el rang de valors que pot tenir, si té alguna propietat específica i una breu descripció:

#### TAULA CENTRES

<u>NOM</u>	<u>TIPUS</u>	<u>RANG VALORS</u>	<u>PROPIETATS</u>	<u>DESCRIPCIÓ</u>
id	INT(11)	Fins a 2147483647	AUTOINCREMENT PRIMARY KEY	Identificador
NOM	VARCHAR(45)	Cadena de 45 caràcters	UNIQUE	Nom del centre
ADRESA	VARCHAR(45)	Cadena de 45 caràcters	NOT NULL	Adreça del centre
CONTACTE	VARCHAR(45)	Cadena de 45 caràcters	NOT NULL	Contacte principal
CONTACTE2	VARCHAR(45)	Cadena de 45 caràcters	NULL	Contacte alternatiu
TELEFON	VARCHAR(9)	9 caràcters numèrics	NOT NULL	Telèfon principal
TELEFON2	VARCHAR(9)	9 caràcters numèrics	NULL	Telèfon alternatiu
EMAIL1	VARCHAR(45)	Cadenes de caràcters	NOT NULL	E-mail principal
EMAIL2	VARCHAR(45)	Cadenes de caràcters	NULL	E-mail alternatiu
OBSERVACIONS	TEXT	Fins 65.535 caràcters	NULL	Comentaris varis

#### TAULA USUARIS

<u>NOM</u>	<u>TIPUS</u>	<u>RANG VALORS</u>	<u>PROPIETATS</u>	<u>DESCRIPCIÓ</u>
id	INT(11)	Fins a 2147483647	AUTOINCREMENT PRIMARY KEY	Identificador
NOM	VARCHAR(45)	Cadena de 45 caràcters	NOT NULL	Nom de l'usuari
COGNOM1	VARCHAR(45)	Cadena de 45 caràcters	NOT NULL	Primer cognom
COGNOM2	VARCHAR(45)	Cadena de 45 caràcters	NULL	Segon cognom
GRUP	VARCHAR(45)	Veure <b>1.2.</b> <b>ACTUASERVEIS SCP</b>	NOT NULL	Tipus Discapacitat
NAIXEMENT	DATE	Qualsevol data vàlida	NOT NULL	Data naixement
DNI	VARCHAR(9)	9 caràcters format DNI	UNIQUE	Número DNI
TELEFON	VARCHAR(9)	9 caràcters numèrics	NULL	Telèfon usuari
SEXE	VARCHAR(1)	M / F	NOT NULL	Sexe de l'usuari
REFERENT	VARCHAR(60)	Cadena de 60 caràcters	NOT NULL	Responsable legal

TELFREF	VARCHAR(45)	Cadenes de caràcters	NOT NULL	Telèfon referent
OBSERVACIONS	TEXT	Fins 65.535 caràcters	NULL	Comentaris varis
centre_id	INT(11)	Fins a 2147483647	FOREIGN KEY	De Taula CENTRES

#### TAULA MONITS (TREBALLADORS)

<u>NOM</u>	<u>TIPUS</u>	<u>RANG VALORS</u>	<u>PROPIETATS</u>	<u>DESCRIPCIÓ</u>
id	INT(11)	Fins a 2147483647	AUTOINCREMENT PRIMARY KEY	Identificador
NOM	VARCHAR(45)	Cadena de 45 caràcters	NOT NULL	Nom del monitor
COGNOM1	VARCHAR(45)	Cadena de 45 caràcters	NOT NULL	Primer cognom
COGNOM2	VARCHAR(45)	Cadena de 45 caràcters	NULL	Segon cognom
DNI	VARCHAR(9)	9 caràcters format DNI	UNIQUE	Número DNI
NAIXEMENT	DATE	Qualsevol data vàlida	NOT NULL	Data naixement
FORMACIO	VARCHAR(45)	Cadena de 45 caràcters	NOT NULL	Tipus de formació
MUNICIPI	VARCHAR(45)	Cadena de 45 caràcters	NOT NULL	Lloc de residència
TELEFON1	VARCHAR(9)	9 caràcters numèrics	NOT NULL	Telèfon monitor
TELEFON2	VARCHAR(9)	9 caràcters numèrics	NULL	Telèfon alternatiu
EMAIL	VARCHAR(45)	Cadena de 45 caràcters	NULL	E-mail personal
CC	VARCHAR(20)	20 caràcters numèrics	NULL	Compte bancari
SS	TINYINT(1)	Si / No	NULL	Se sap el número?
NUMSS	VARCHAR(12)	12 caràcters format SS	NULL, UNIQUE	Numero Seg. Social
CARNET	TINYINT(1)	Si / No	NULL	Carnet de conduir
NOTA	INT(1)	0 - 9	NULL	Nota rendiment
OBSERVACIONS	TEXT	Fins 65.535 caràcters	NULL	Comentaris varis

#### TAULA LLOCS

<u>NOM</u>	<u>TIPUS</u>	<u>RANG VALORS</u>	<u>PROPIETATS</u>	<u>DESCRIPCIÓ</u>
id	INT(11)	Fins a 2147483647	AUTOINCREMENT PRIMARY KEY	Identificador
NOM	VARCHAR(45)	Cadena de 45 caràcters	UNIQUE	Nom del lloc
ADRESA	VARCHAR(45)	Cadena de 45 caràcters	NOT NULL	Adreça del lloc
MUNICIPI	VARCHAR(45)	Cadena de 45 caràcters	NOT NULL	Municipi del lloc
CONTACTE	VARCHAR(45)	Cadena de 45 caràcters	NOT NULL	Contacte principal
TELEFON	VARCHAR(9)	9 caràcters numèrics	NOT NULL	Telèfon principal
OBSERVACIONS	TEXT	Fins 65.535 caràcters	NULL	Comentaris varis

### TAULA SORTIDES

<u>NOM</u>	<u>TIPUS</u>	<u>RANG VALORS</u>	<u>PROPIETATS</u>	<u>DESCRIPCIÓ</u>
id	INT(11)	Fins a 2147483647	AUTOINCREMENT PRIMARY KEY	Identificador
NOM	VARCHAR(45)	Cadena de 45 caràcters	UNIQUE	Municipi Sortida
CAMPANYA	VARCHAR(45)	Nadal, Setmana Sta, Estiu		Època sortida
lloc_id	VARCHAR(45)	Cadena de 45 caràcters	FOREIGN	De Taula LLOCS
GRUP	VARCHAR(45)	Veure <b>1.2.</b> <b>ACTUASERVEIS SCP</b>	NOT NULL	Tipus discapacitat dels usuaris
PLACES	INT(2)	Número de 2 xifres	NOT NULL	Número d'usuaris
ANADA	VARCHAR(9)	Qualsevol data vàlida	NOT NULL	Data d'anada
TORNADA	VARCHAR(9)	Qualsevol data vàlida	NOT NULL	Data de tornada
PRESSUPOST	DOUBLE	Número amb decimals	NULL	Pressupost sortida
OBSERVACIONS	TEXT	Fins 65.535 caràcters	NULL	Comentaris varis

### TAULA REQUUSUARIS (Requisits dels usuaris durant la sortida)

<u>NOM</u>	<u>TIPUS</u>	<u>RANG VALORS</u>	<u>PROPIETATS</u>	<u>DESCRIPCIÓ</u>
id	INT(11)	Fins a 2147483647	AUTOINCREMENT PRIMARY KEY	Identificador
sortide_id	INT(11)	Fins a 2147483647	FOREIGN, UNIQUE	De Taula SORTIDES
usuari_id	INT(11)	Fins a 2147483647		De Taula USUARIS
TRANSANADA	TINYINT(1)	Si / No	NOT NULL	Si l'usuari pot anar o tornar al punt de trobadra de la sortida
TRANSTORNADA	TINYINT(1)	Si / No	NOT NULL	
TELEFON	VARCHAR(9)	9 caràcters numèrics	NULL	Telèfon usuari
LLIT	TINYINT(1)	Si / No	NOT NULL	Pot anar llitera de dalt?
CADIRA_RODES	TINYINT(1)	Si / No	NOT NULL	Porta cadira de rodes?
SOL	TINYINT(1)	Si / No	NOT NULL	Pot sortir sol?
DIETA	TINYINT(1)	Si / No	NOT NULL	Fa dieta?
TIPUSDIETA	VARCHAR(45)	Cadena de 45 caràcters	NULL	Tipus de dieta
OBSERVACIONS	TEXT	Fins 65.535 caràcters	NULL	Comentaris varis

### TAULA ACOMPANYAMENTS

<u>NOM</u>	<u>TIPUS</u>	<u>RANG VALORS</u>	<u>PROPIETATS</u>	<u>DESCRIPCIÓ</u>
id	INT(11)	Fins a 2147483647	AUTOINCREMENT PRIMARY KEY	Identificador
DIA	DATE	Qualsevol data vàlida	UNIQUE	Data inici servei
usuari_id	INT(11)	Fins a 2147483647	(usuari_id FOREIGN)	De Taula USUARIS
monit_id	INT(11)	Fins a 2147483647	FOREIGN	De Taula MONITS

TIPUS	VARCHAR(45)	Oci, vetlla, urgència,...	NOT NULL	Tipus de servei
INICI	TIME	Format HH:MM	NULL	Hora inici servei
FINAL	TIME	Format HH:MM	NULL	Hora final servei
SOLICITANT	VARCHAR(45)	Cadena de 45 caràcters	NOT NULL	Responsable legal
RECULLADR	VARCHAR(45)	Cadena de 45 caràcters	NULL	Adreça recollida
VISITADR	VARCHAR(45)	Cadena de 45 caràcters	NULL	Adreça Visita
HORAVISITA	TIME	Format HH:MM	NULL	Hora de visita
DESPESES	TEXT	Fins 65.535 caràcters	NULL	Despeses servei
TRANSPORT	VARCHAR(45)	Taxi, a peu, cotxe,...	NULL	Tipus transport
KMSERVEI	FLOAT	Número amb decimals	NULL	Km's servei
KMTREBALLADOR	FLOAT	Número amb decimals	NULL	Km's monitor
PREUKMS	FLOAT	0.25 per defecte	NOT NULL	Preu Km/servei
PREUKMT	FLOAT	0.20 per defecte	NOT NULL	Preu Km/monitor
OBSERVACIONS	TEXT	Fins 65.535 caràcters	NULL	Comentaris varis

### SORTIDES\_MONITS

<u>NOM</u>	<u>TIPUS</u>	<u>RANG VALORS</u>	<u>PROPIETATS</u>	<u>DESCRIPCIÓ</u>
id	INT(11)	Fins a 2147483647	AUTOINCREMENT PRIMARY KEY	Identificador
sortide_id	INT(11)	Fins a 2147483647	FOREIGN,	De Taula SORTIDES
monit_id	INT(11)	Fins a 2147483647	UNIQUE	De Taula MONITS

### SORTIDES\_USUARI

<u>NOM</u>	<u>TIPUS</u>	<u>RANG VALORS</u>	<u>PROPIETATS</u>	<u>DESCRIPCIÓ</u>
id	INT(11)	Fins a 2147483647	AUTOINCREMENT PRIMARY KEY	Identificador
sortide_id	INT(11)	Fins a 2147483647	FOREIGN,	De Taula SORTIDES
usuari_id	INT(11)	Fins a 2147483647	UNIQUE	De Taula USUARI

## 5.3. Vistes de l'aplicació

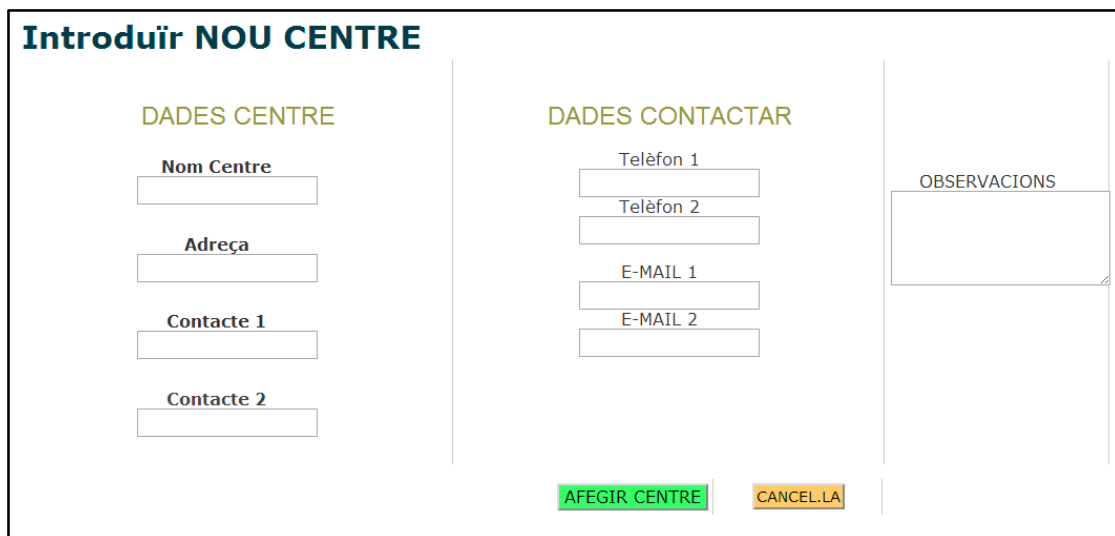
En aquest capítol, mostrarem, a mode d'exemple, algunes d'aquelles vistes que s'utilitzaran en la nostra aplicació i que permetran resoldre les necessitats descrites tan en punt **2.2. Requeriments Funcionals i no Funcionals** com, per extensió, en el punt **4.2. Diagrama Casos d'Ús**. Les vistes mostren formularis que el Framework CakePHP crea automàticament dependent

del número de camp i el tipus de variable que aquest necessita. Per exemple, si és un camp tipus TIME, ens ho mostra com un desplegable per seleccionar les hores i els minuts. En canvi, si es tracta d'un camp tipus TEXT o VARCHAR, ens apareix un camp de formulari on s'hi pot escriure.

*NOTA: TOTES LES DADES PERSONALS DELS EXEMPLES SÓN INVENTADES*

### VISTA DONAR D'ALTA UN ELEMENT (Centre, lloc, usuari o treballador)

A al **Figura 13**, prenem com a exemple el formulari que apareix en la creació d'un nou centre:



**Introduir NOU CENTRE**

**DADES CENTRE**

Nom Centre

Adreça

Contacte 1

Contacte 2

**DADES CONTACTAR**

Telèfon 1

Telèfon 2

E-MAIL 1

E-MAIL 2

OBSERVACIONS

**AFEGIR CENTRE** **CANCEL·LA**

Figura 13. Formulari per donar d'alta un centre

### VISTA MODIFICAR UN ELEMENT (Centre, lloc, usuari o treballador)

Semblant al que vèiem en el formulari de creació, en el cas de la modificació, ens apareix el formulari amb les dades ja introduïdes i la possibilitat de canviar-les. A la **Figura 14**, l'exemple del formulari que es mostra per modificar un usuari:



### Modificar Usuari

#### DADES GENERALS

**Nom**  
Marc

**Cognom 1**  
Pera

**Cognom 2**  
Sala

**DNI**  
7322224G

**Data Naixement**  
16 / 10 / 1971

**Centre**  
Santa Creu

**SEXE**  
 Home  
 Dona

#### DADES ACTUA

**GRUP**  
DISC. Limitat Extens

**Referent 1**  
Maria Sala

**Telf. Referent 1**  
938787873

**OBSERVACIONS**  
Bastant tranquil

Figura 14. Formulari per modificar les dades d'un usuari

### VISTA ELIMINAR UN ELEMENT (Centre, lloc, usuari o treballador)

Per eliminar un element, cal anar a la pantalla on es llisten tots els elements d'aquella entitat (tots els centres, tots els treballadors,...). Al costat de cada element, apareix tan l'enllaç per modificar l'element (veure l'anterior **Figura 14**) com per eliminar-lo. Quan cliquem a "BORRAR" ens apareix un quadre de diàleg que ens demana la confirmació de l'operació. Per exemple, si volem eliminar un usuari ens apareix el que mostrem a la **Figura 15**:

### Usuaris d'ACTUA SERVEIS(8)

MENU USUARIS
MENU PRINCIPAL
BUSCAR

NOM	CENTRE	CONDICIONS	ACCIONS	FOTO?
<a href="#">Bustos Riera, Marta</a>	Autònoma Solidària	d'ansietat	<a href="#">MODIFICAR</a> <a href="#">BORRAR</a>	si
<a href="#">Font Auseas, Antònia</a>	(PARTICULAR)		<a href="#">MODIFICAR</a> <a href="#">BORRAR</a>	si
<a href="#">López Dorn, Ignasi</a>	Centre Unió Bages		<a href="#">MODIFICAR</a> <a href="#">BORRAR</a>	si
<a href="#">Montilivi Suárez, Xavi</a>	Centre Gaudí Nord	DISC. Lim. Ext. Sènior M Problemes al caminar	<a href="#">MODIFICAR</a> <a href="#">BORRAR</a>	si
<a href="#">Pera Sala, Marc</a>	Santa Creu	DISC. Limitat Extens M Bastant tranquil	<a href="#">MODIFICAR</a> <a href="#">BORRAR</a>	si
<a href="#">Pérez Cruz, Pep</a>	Autònoma Solidària	DISC. Intermitent M Caracter tranquil	<a href="#">MODIFICAR</a> <a href="#">BORRAR</a>	si
<a href="#">Sacristán Valle, Aurelia</a>	Centre Unió Bages	MALALT MENTAL F Problemes auditiu	<a href="#">MODIFICAR</a> <a href="#">BORRAR</a>	si
<a href="#">Salgado Bermúdez, Josep Miquel</a>	Santa Creu	DISC. Generalitzat M Molt Disciplinat	<a href="#">MODIFICAR</a> <a href="#">BORRAR</a>	si

MENU USUARIS
MENU PRINCIPAL

Figura 15. Vista d'eliminació d'un usuari

## VISTA BUSCAR UN ELEMENT (Centre, lloc, usuari o treballador)

Per tal de buscar cadascun dels elements, tenim varies opcions. La més usual és escriure el nom de l'element que busquem i que l'aplicació ens retorni tots aquells elements que es diguin com la paraula clau escrita. En el cas dels usuaris però, també podem definir altres criteri de cerca a part del nom i cognoms. Podem buscar per centre i també per grup d'usuari (tipus de discapacitat) tal i com es mostra a continuació a la **Figura 16**:

**BUSCAR USUARIS**

Defineix el criteri per buscar (NOM COMPLET || NOM + COGNOM || COGNOMS):

**PER NOM**  **BUSCAR** NOTA: No calen accents ni majúscules

**PER CENTRE** Autònoma Solidària **BUSCAR**

**PER GRUP** 3a EDAT **BUSCAR**

CANCEL·LA **INCIPAL**

- 3a EDAT
- 3a EDAT
- DISC. Intermitents
- DISC. Limitats Extens
- DISC. Generalitzats
- DISC. Interm. Sènior
- DISC. Lim. Ext. Sènior
- MALALTS MENTALS

Figura 16. Vista de la cerca d'un usuari

## VISTA CREAR / MODIFICAR SORTIDA

A petició dels responsables de l'empresa, la gestió de les sortides comença amb un menú on es pot crear una nova sortida o consultar les ja creades. Les sortides ja creades es mostren primer triant l'any (donant la opció de consultar les sortides dels últims cinc anys) i llavors seleccionant la campanya a mostrar (Nadal, Setmana Santa o Estiu) o directament tot l'any. Hi ha la possibilitat de filtrar les sortides mostrades pel criteri del grup (tipus de discapacitat). Ho veiem a la **Figura 17**:



## Gestió de Sortides d'ACTUASERVEIS

### Lista de torns SETMANA SANTA de 2014

Filtrar per:

[3a EDAT](#) [Intermitents](#) [Lim. Extens](#) [Generalitzats](#) [Interm. Sènior](#) [Lim. Ext. Sènior](#) [Malalts Mentals](#) [TOTS els Grups](#)

PERIODE	TORN	GRUP	LLOC	PLACES	MONITS.	SORTIDA		ADMINISTRACIÓ	
06/04/14 - 13/04/14	TARRAGONA A	Malalts Mentals	Àrea 55	0	1	<a href="#">MODIFICAR</a>	<a href="#">BORRAR</a>	<a href="#">ASSIGNA MONITORS</a>	<a href="#">APUNTA USUARIS</a>
10/04/14 - 17/04/14	LLEIDA A	Disc. Limitats Extens	Casa Vidal	0	1	<a href="#">MODIFICAR</a>	<a href="#">BORRAR</a>	<a href="#">ASSIGNA MONITORS</a>	<a href="#">APUNTA USUARIS</a>
13/04/14 - 20/04/14	GRANADA A	Disc. Intermitents	Àrea 55	3	3	<a href="#">MODIFICAR</a>	<a href="#">BORRAR</a>	<a href="#">ASSIGNA MONITORS</a>	<a href="#">APUNTA USUARIS</a>
13/04/14 - 20/04/14	GRANADA B	3a. EDAT	(SENSE LLOC)	0	0	<a href="#">MODIFICAR</a>	<a href="#">BORRAR</a>	<a href="#">ASSIGNA MONITORS</a>	<a href="#">APUNTA USUARIS</a>
13/04/14 - 20/04/14	CUENCA	Disc. Generalitzats	Masia La Trobada	0	0	<a href="#">MODIFICAR</a>	<a href="#">BORRAR</a>	<a href="#">ASSIGNA MONITORS</a>	<a href="#">APUNTA USUARIS</a>
15/04/14 - 22/04/14	CERVERA A	Malalts Mentals	Masia La Trobada	1	2	<a href="#">MODIFICAR</a>	<a href="#">BORRAR</a>	<a href="#">ASSIGNA MONITORS</a>	<a href="#">APUNTA USUARIS</a>

[CANVIA D'ANY](#)

[MENU](#)

Figura 17. Vista principal de la gestió de sortides

De la mateixa manera que comentàvem en els casos anteriors, amb les sortides disposem d'un mateix formulari tan per donar d'alta la sortida com per modificar-la apareixent, en aquest últim cas, les dades ja introduïdes. Les sortides, primer es creen com si d'un esdeveniment amb característiques determinades es tractés. Un cop donada d'alta la sortida, es poden assignar treballadors i apuntar usuaris des d'un altre secció de l'aplicació que mostrarem més endavant. Veiem ara, a la **Figura 18**, el formulari de creació/modificació d'una sortida:

### Modificar SORTIDA

<b>DADES GENERALS</b>	<b>DIA DE SORTIDA</b>	<b>ALTRES DADES</b>
<b>NOM SORTIDA</b> <input type="text" value="TARRAGONA A"/>	<b>ANADA</b> 6 ▾ 4 ▾ 2014 ▾	<b>PRESSUPOST</b> <input type="text" value="250"/>
<b>LLOC</b> <input type="text" value="Àrea 55"/>	<b>TORNADA</b> 13 ▾ 4 ▾ 2014 ▾	<b>OBSERVACIONS</b> <input type="text"/>
<b>GRUP</b> <input type="text" value="Malalts Mentals"/>	<b>CAMPANYA</b> <input type="text" value="SETMANA STA"/>	
<a href="#">MODIFICAR SORTIDA</a>	<a href="#">CANCEL·LA</a>	<a href="#">MENU</a>

Figura 18. Formulari per modificar les dades d'una sortida

## VISTA ASSIGNAR TREBALLADOR A UNA SORTIDA

Un cop creada la sortida, s'ha d'accedir a l'apartat de consultar sortides que comentàvem abans. Triant l'any a consultar i l'època apareixen les sortides ordenades per data d'inici. Com hem vist, cada sortida té una opció anomenada "ASSIGNAR MONITORS". Dins d'aquesta secció, se'ns mostra la informació resumida de la sortida, els monitors ja assignats (amb la possibilitat de des assignar-los), i la llista de monitors disponibles en les dates de la sortida com mostrem a la **Figura 19**:

Informació sortida GRANADA A		Treballadors assignats a la sortida: 3		
Campanya	Setmana Sta	TREBALLADOR 1	Jauregi Izaskun, Joel	DESASSIGNAR
Anada:	13/04/14	TREBALLADOR 2	Álvarez Pinto, Joaquim	DESASSIGNAR
Tornada:	20/04/14	TREBALLADOR 3	Puig Rius, Marta	DESASSIGNAR
Places:	3			
<b>Treballadors disponibles:</b>				
MONITOR				
<input type="radio"/>	Estruch Plaza, Marta			
<input type="radio"/>	Giol Sánchez, Montse			
<input type="button" value="ASSIGNA"/>		<input type="button" value="ACABAT"/>		

Figura 19. Vista de l'assignació de treballadors a una sortida

## VISTA APUNTAR USUARI A UNA SORTIDA

Al costat de la opció "ASSIGNAR MONITORS" que veiem abans, hi ha l'opció anomenada "APUNTAR USUARIS". Com que a part dels usuaris, s'han de definir els requisits d'aquests, si s'escau, hem optat per seguir el següent ordre, tot oferint un petit buscador d'usuaris en aquesta secció, per facilitar el procés:

1) Buscar usuari: per nom, centre o grup (Figura 20)

**Usuaris apuntats a la sortida: 3**

Bustos Riera, Marta	MODIFICAR	DESAPUNTAR
Montilivi Suárez, Xavi	MODIFICAR	DESAPUNTAR
Pérez Cruz, Pep	MODIFICAR	DESAPUNTAR

**INFORMACIÓ SORTIDA:**

SORTIDA	GRANADA A
GRUP	DISC. Intermitents
CAMPANYA	Setmana Sta
ANY	2014
LLOC	Àrea 55
PLACES	3

**BÚSQUEDA RÀPIDA:**  
 Defineix el criteri per buscar (NOM COMPLET || NOM + COGNOM || COGNOMS):  
 NOTA: No calen accents ni majúscules

PER NOM

PER CENTRE Autònoma Solidària

PER GRUP 3a EDAT

Figura 20. Vista principal d'apuntar usuaris (amb el buscador)

2) Triar l'usuari i definir-ne els requisits si es vol (es pot fer més endavant). Veiem la Figura 21:

**INFORMACIÓ PARTICIPANT:**

FOTO	NOM COMPLET	Salgado Bermúdez ,Josep Miquel	OBSERVACIONS Molt Disciplinat  HISTORIAL PARTICIPANT Participant no apuntat a cap sortida
	DNI	89214124Q	
	DATA NAIXEMENT	18-05-1968	
	SEXE	Masculí	
	CENTRE	<u>Santa Creu</u>	
	GRUP	DISC. Generalitzat	

**INFORMACIÓ DE SORTIDA:**


SORTIDA	GRANADA A	TRANS. ANADA	Si	LLIT DE DALT	Si
GRUP	DISC. Intermitent	TRANS.TORNADA	No	CADIRA DE RODES	No
CAMPANYA	Setmana Sta	SORTIR SOL	- Triar -	TELÈFON	--
ANY	2014	DIETA	No	OBSERVACIONS:	--
LLOC	3	TIPUS DIETA	--		
PLACES	0				

Figura 21. Formulari per definir els requisits d'un usuari en una sortida

3) Apuntar-lo a la sortida

### VISTA ELIMINAR SORTIDA

Aquest cas, no difereix pràcticament del sistema per eliminar la resta d'elements. Podem veure a la **Figura 21**, l'enllaç per eliminar una sortida, ja vista amb anterioritat:



**Gestió de Sortides d'ACTUASERVEIS**

**Llista de torns SETMANA SANTA de 2014**

Filtrar per: [3a EDAT](#) | [Intermitents](#) | [Lim. Extens](#) | [Generalitzats](#) | [Interm. Sènior](#) | [Lim. Ext. Sènior](#) | [Malalts Mentals](#) | [TOTS els Grups](#)

PERIODE	TORN	GRUP	LLOC	PLACES	MONITS.	SORTIDA	ADMINISTRACIÓ
06/04/14 - 13/04/14	TARRAGONA A	Malalts Mentals	Àrea 55	0	1	<a href="#">MODIFICAR</a> <a href="#">BORRAR</a>	<a href="#">ASSIGNA MONITORS</a> <a href="#">APUNTA USUARIS</a>
10/04/14 - 17/04/14	LLEIDA A	Disc. Limitats Extens	Casa Vidal	0	1	<a href="#">MODIFICAR</a> <a href="#">BORRAR</a>	<a href="#">ASSIGNA MONITORS</a> <a href="#">APUNTA USUARIS</a>
13/04/14 - 20/04/14	GRANADA A	Disc. Intermitents	Àrea 55	3	3	<a href="#">MODIFICAR</a> <a href="#">BORRAR</a>	<a href="#">ASSIGNA MONITORS</a> <a href="#">APUNTA USUARIS</a>
13/04/14 - 20/04/14	GRANADA B	3a. EDAT	(SENSE LLOC)	0	0	<a href="#">MODIFICAR</a> <a href="#">BORRAR</a>	<a href="#">ASSIGNA MONITORS</a> <a href="#">APUNTA USUARIS</a>
13/04/14 - 20/04/14	CUENCA	Disc. Generalitzats	Masia La Trobada	0	0	<a href="#">MODIFICAR</a> <a href="#">BORRAR</a>	<a href="#">ASSIGNA MONITORS</a> <a href="#">APUNTA USUARIS</a>
15/04/14 - 22/04/14	CERVERA A	Malalts Mentals	Masia La Trobada	1	2	<a href="#">MODIFICAR</a> <a href="#">BORRAR</a>	<a href="#">ASSIGNA MONITORS</a> <a href="#">APUNTA USUARIS</a>

[CANVIA D'ANY](#) [MENU](#)

Figura 22. Vista principal de la gestió de sortides amb l'opció "BORRAR" remarcada

### VISTA CREAR ACOMPANYAMENT

En aquest cas, la secció principal dels acompanyaments també té la part de consultar els acompanyaments ja creats de cadascun dels últims cinc anys i especificar després el mes concret a consultar, quelcom semblant al que ja veiem en les sortides. També hi ha l'opció lògica de crear un acompanyament nou. A més, degut al càlcul de les facturacions que ampliarem més endavant, vàrem habilitar un buscador de treballadors per tal de poder consultar ràpidament els acompanyaments realitzats per tal treballador durant un any concret i per cadascun dels mesos d'aquell any.

El tractament dels acompanyaments però, no té massa similitud amb la creació de les sortides ja que, després de consultar amb l'empresa ACTUASERVEIS SCL quin era el mètode que utilitzaven per fer el seguiment d'un acompanyament, vàrem creure convenient seguir una sèrie de passos diferents que els utilitzats en la creació de la sortida. S'il·lustra amb la vista corresponent en aquells casos que hem cregut necessaris:

- 1) Buscar l'usuari que demana un servei d'acompanyament.
- 2) Buscar el treballador, si és que ja està concretat, que faci l'acompanyament.
- 3) Omplir el formulari que apareix a la **Figura 23**, per la creació de l'acompanyament

### CREAR ACOMPANYAMENT

<p style="text-align: center; margin: 0;"><b>DADES ACOMPANYAMENT</b></p> <p><b>DATA ACOMPANYAMENT</b>  Dia ▼ Mes ▼ Any ▼</p> <p><b>TIPUS SERVEI</b>  -- Tria -- ▼</p> <p><b>INICI</b>  0 ▼ :00 ▼</p> <p><b>FINAL</b>  0 ▼ :00 ▼</p> <p><b>HORA VISITA</b>  0 ▼ :00 ▼</p> <p><b>Adreça Recollida</b>  <input style="width: 100%;" type="text"/></p> <p><b>Adreça Visita</b>  <input style="width: 100%;" type="text"/></p> <p><b>TRANSPORT</b>  -- Tria -- ▼</p> <p style="text-align: center; margin-top: 10px;"> <span style="background-color: #28a745; color: white; padding: 2px 10px; border: 1px solid black;">CREAR ACOMPANYAMENT</span> <span style="background-color: #ffc107; color: black; padding: 2px 10px; border: 1px solid black; margin-left: 10px;">CANCEL·LA</span> </p>	<p style="text-align: center; margin: 0;"><b>DADES USUARI</b></p> <p><b>NOM USUARI</b>  <u>Pérez Cruz, Pep</u></p> <p><b>ENTITAT / REFERENT</b>  Autònoma Solidària</p> <p style="text-align: center; margin: 10px 0 0 0;"><b>TREBALLADOR</b></p> <p><b>Realitzat per</b>  Sense treballador assignat <span style="background-color: #28a745; color: white; padding: 2px 5px; border: 1px solid black;">ASSIGNA</span></p> <p><b>OBSERVACIONS</b>  <div style="border: 1px solid black; height: 40px; width: 100%;"></div> </p>	<p style="text-align: center; margin: 0;"><b>FACTURACIONS</b></p> <p><b>Kms. Servei</b>  <input style="width: 100%;" type="text"/></p> <p><b>Preu Kms. Servei</b>  0.25</p> <p><b>Kms. Treballador</b>  <input style="width: 100%;" type="text"/></p> <p><b>Preu Kms. Treballador</b>  0.20</p> <p><b>Despeses</b>  <div style="border: 1px solid black; height: 40px; width: 100%;"></div> </p>
---	---	--

Figura 23. Formulari per crear un acompanyament (sense treballador assignat)

### VISTA MODIFICAR ACOMPANYAMENT

La modificació d'un acompanyament, ens mostra de nou el mateix formulari de creació amb les dades ja introduïdes. A més, es pot assignar o reassignar el treballador encarregat d'oferir tal servei.

Per tal de modificar un acompanyament, s'ha de consultar des de l'any i mes que es realitza o bé, des del nom del treballador seguit de l'any i mes. Modificar un acompanyament significa que es pot canviar qualsevol cosa excepte l'usuari ja que, en cas que l'usuari cancel·li l'acompanyament, el més correcte serà eliminar-lo. Veiem la vista de modificació (**Figura 24**):

### MODIFICAR ACOMPANYAMENT

#### DADES ACOMPANYAMENT

**DATA ACOMPANYAMENT**

**TIPUS SERVEI**

**INICI**

**FINAL**

**HORA VISITA**

**Adreça Recollida**

**Adreça Visita**

**TRANSPORT**

#### DADES USUARI

**NOM USUARI**  
López Dorn, Ignasi

**ENTITAT / REFERENT**  
Centre Unió Bages

#### TREBALLADOR

**Realitzat per**  
Estruch Plaza, Marta CANVIAR

#### OBSERVACIONS

#### FACTURACIONS

**Kms. Servei**

**Preu Kms. Servei**

**Kms. Treballador**

**Preu Kms. Treballador**

**Despeses**

MODIFICAR ACOMPANYAMENT
CANCEL·LA

Figura 24. Formulari per modificar un acompanyament (amb treballador ja assignat)

### VISTA ELIMINAR ACOMPANYAMENT

Seguint el que ja hem vist per l'eliminació d'altres elements, l'eliminació d'un acompanyament tindria la següent vista mostrada a la **Figura 25**:

**Acompanyament borrrat correctament**

#### CONSULTAR ACOMPANYAMENTS MAIG 2014

DIA	INICI	FINAL	TIPUS	TRANSPORT	KM S. (preu/km)	ADREÇA RECOLLIDA	ADREÇA VISITA	USUARI	ENTITAT / REFERENT	TREBALLADOR	OBSERVACIONS	MODIFICAR	ELIMINAR
10/05	15:00:00	23:00:00	Urgència	Cotxe	10 kms (0.25)	C/ Eixample, 44		Font Auseas, Antònia	Pol Auseas	Jauregi Izaskun, Joel		<a href="#">MODIFICAR</a>	<a href="#">BORRAR</a>
17/05	10:30:00	12:30:00	Acomp. Mèdic	Taxi	15 kms (0.25)	C/ Joanot Martorell	C/ Hospital 23	Pérez Cruz, Pep	Autònoma Solidària	Estruch Plaza, Marta	Comprar aigua	<a href="#">MODIFICAR</a>	<a href="#">BORRAR</a>
28/05	16:00:00	19:00:00	Acomp. Oci	Taxi	15 kms (0.25)	c/ Flauta, 444		Pérez Cruz, Pep	Autònoma Solidària	[SENSE ASSIGNAR]		<a href="#">MODIFICAR</a>	<a href="#">BORRAR</a>
31/05	17:18:00	19:34:00	Acomp. Oci	Cotxe	0 kms (0.25)	C/ Prim, 45 (Olost)	--	Salgado Bermúdez, Josep Miquel	Santa Creu	Estruch Plaza, Marta		<a href="#">MODIFICAR</a>	<a href="#">BORRAR</a>
<b>TOTAL</b>					<b>10 €</b>								

ACOMPANYAMENTS
MENU

Figura 25. Vista principal dels acompanyaments del maig de 2014 amb l'opció "BORRAR" remarcada







## 6. Implementació

### 6.1. Programari instal·lat

La implementació de la nostra aplicació web va començar instal·lant el software necessari per a treballar en la seva construcció. Com hem dit en anteriors capítols (veure **3.4. El paquet XAMPP**), comptàvem amb l'avantatge que ens suposava tenir tot el programari encapsulat en el paquet de programes XAMPP. Aquest paquet ja disposa d'un executable que permet instal·lar-lo de manera ràpida i senzilla [16]. Concretament, nosaltres hem instal·lat la versió 1.7.3.

Un cop instal·lat, ens vam disposar a configurar algunes opcions de seguretat. Per a accedir a aquestes opcions, és necessari engegar el **XAMPP Control Panel**, que és l'aplicació que permet controlar quins elements del paquet vols utilitzar. En el nostre cas, tan sols cal posar en funcionament el servidor Apache i MySQL, clicant en els respectius botons Start que ens apareixen en el Control Panel. Un cop funcionant, ens apareix en estat *Running*, indicat de la forma que mostrem a la **Figura 27**:

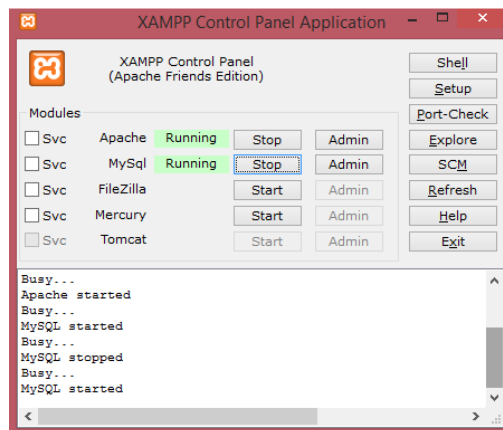


Figura 27. Control Panel amb Apache i MySQL funcionant

Seguidament, des del navegador escrivint <http://localhost/xampp>, i veiem aparèixer una pàgina web que ens felicita per la instal·lació correcte de XAMPP. Just a l'esquerra hi apareix un menú d'opcions on hi ha el *Chequeo de Seguridad*, al qual accedim i se'ns mostra l'estat de seguretat del sistema, depenent de la configuració actual de XAMPP.

Per a solucionar els problemes de seguretat que apareixen amb la configuració per defecte, hem definit un nom d'usuari i contrasenya per accedir a MySQL / phpMyAdmin i un altre per accedir al directori arrel de XAMPP. Seguidament tornant al *Chequeo de Seguridad*, ens apareix una taula-resum de l'estat del sistema. En els punts que afecten la nostra aplicació (directori XAMPP, MySQL i phpMyAdmin), es mostren ara amb un estat segur com veiem a la **Figura 28**:

Concerniente a	Estado
Estas paginas XAMPP NO se visualizan a través de la red	SEGURO
MySQL-root tiene clave de acceso	SEGURO
PhpMyAdmin password login is enabled.	SEGURO
A FTP server is not running or is blocked by a firewall!	DESCONOCIDO
A POP3 server like Mercury Mail is not running or is blocked by a firewall!	DESCONOCIDO
The Tomcat add-on is not installed.	DESCONOCIDO

Figura 28. Taula resum de seguretat. Es elements en Desconocido, son utilitats que no fem servir

## 6.2. Creació de la base de dades i taules a través de phpMyAdmin

Un cop finalitzats la instal·lació i els aspectes de seguretat, vam entrar a phpMyAdmin a través de la direcció <http://localhost/phpmyadmin>. Vam comprovar que, efectivament, ens demanava el nom d'usuari i contrasenya que vam definir en el pas anterior.

Primer de tot se'ns demanava el nom de la base de dades i quin tipus de codificació volíem. Vam seleccionar *UTF8 – Spanish* que ens permetia guardar registres que continguessin caràcters especials com els accents oberts, tancats, *c trencada* (ç) o la *enya* castellana (ñ), cosa que prevèiem que passaria amb tota seguretat amb, per exemple, els noms i/o cognoms dels actors que intervindrien en la nostra base de dades.

Amb la base de dades creada (i buida), ens apareix la opció de crear una taula amb el número de camps que vulguem. A partir d'aquí, per a evitar una repetició excessiva dels passos seguits, prendrem com a exemple l'entitat CENTRES per a explicar la creació de la resta de taules, explicant les excepcions pertinents en les entitats que calgui (sobretot, en SORTIDES i ACOMPANYAMENTS).

Cal dir que, a part de les convencions ja comentades, s'ha de seguir un cert ordre lògic alhora de crear les taules. Per exemple, no podem crear la taula USUARIS sense haver creat la taula CENTRES, degut a la relació que tenen aquestes entitats entre si: s'ha de definir una clau externa a la taula USUARIS que contingui el camp identificador de la taula CENTRE.

El fet d'utilitzar claus externes, ens obliga a utilitzar el motor d'emmagatzematge InnoDB, ja que l'altra opció (MyISAM) no permet utilitzar-ne. Observem la taula Centres creada a la **Figura 29**:

Camp	Tipus	Ordenació	Atributs	Nul	Defecte	Extra	Acció
<input type="checkbox"/> id	int(11)			No	Cap	auto_increment	[Icons]
<input type="checkbox"/> NOM	varchar(45)	utf8_spanish2_ci		No	Cap		[Icons]
<input type="checkbox"/> ADRESA	varchar(45)	utf8_spanish2_ci		No	Cap		[Icons]
<input type="checkbox"/> CONTACTE	varchar(45)	utf8_spanish2_ci		No	Cap		[Icons]
<input type="checkbox"/> CONTACTE2	varchar(45)	utf8_spanish2_ci		Si	NULL		[Icons]
<input type="checkbox"/> TELEFON1	varchar(9)	utf8_spanish2_ci		No	Cap		[Icons]
<input type="checkbox"/> TELEFON2	varchar(9)	utf8_spanish2_ci		Si	NULL		[Icons]
<input type="checkbox"/> EMAIL1	varchar(45)	utf8_spanish2_ci		Si	NULL		[Icons]
<input type="checkbox"/> EMAIL2	varchar(45)	utf8_spanish2_ci		Si	NULL		[Icons]
<input type="checkbox"/> OBSERVACIONS	text	utf8_spanish2_ci		Si	NULL		[Icons]

Figura 29. Taula centres creada.

En resum, primer varem crear les taules: **centres, llocs i monits** (treballadors). Llavors la taula **usuaris** (amb clau externa a centres). Seguidament, les taules **sortides** (amb clau externa a llocs) i **acompanyaments** (amb claus externes a usuaris i monits). Finalment, va arribar el torn de les taules **requsuaris** (claus externes a usuaris i sortides), **sortides\_usuaris** (claus externes a usuaris i sortides) i **monits\_sortides** (amb claus externes a monits i sortides)

### 6.3. Creació dels índexs de restricció de les taules

Amb les taules ja creades, el següent pas va ser definir els índexs que faltaven. Totes les taules ja tenien la seva clau primària que, com indicàvem en les convencions de CakePHP, està reservada al camp obligatori "id". Per mantenir la integritat de les dades, tal i com anunciàvem, hem definit com a **UNIQUE** els següents camps:

<u>TAULA</u>	<u>CAMP O CAMPS UNIQUE</u>
ACOMPANYAMENTS	usuari_id, DIA
CENTRES	NOM
LLOCS	NOM
MONITS	DNI
MONITS_SORTIDES	monit_id, sortide_id
REQUSUARIS	sortide_id, usuari_id
SORTIDES	NOM, CAMPANYA
SORTIDES_USUARIS	sortide_id, usuari_id

USUARIS

DNI

El phpMyAdmin, ens mostra les claus UNIQUE de la forma que veiem a la **Figura 30**:

Índexs: ⓘ





Acció	Nom Clau	Tipus	Única	Comprimit	Camp	Cardinalitat	Ordenació	Nul	Comentari
 	PRIMARY	BTREE	Si	No	id	2	A		
 	NOM	BTREE	Si	No	NOM	2	A		

Figura 30. Índexs de la taula centres

#### 6.4. Creació dels directoris necessaris

Aquest pas s'ha anat realitzant en paral·lel a la creació dels nous models i els seus corresponents vistes i controladors. Hem seguit l'estructura que demana per convencions el CakePHP (veure **3.4.2. Funcionament de CakePHP dins XAMPP**). Com a mostra, a la **Figura 31**, els directoris creats per a contenir els fitxers .ctp de les vistes de cada model marcats en blau:

TI31215400A (C:) > xampp > htdocs > actua2 > app > views > Cerca
















Nom	Data de modificació	Tipus	Mida
 <b>acompanyaments</b>	10/06/2014 18:27	Carpeta de fitxers	
 <b>centres</b>	15/04/2014 12:04	Carpeta de fitxers	
 elements	15/04/2014 12:04	Carpeta de fitxers	
 errors	15/04/2014 12:04	Carpeta de fitxers	
 helpers	15/04/2014 12:04	Carpeta de fitxers	
 layouts	15/04/2014 12:04	Carpeta de fitxers	
 <b>llocs</b>	15/04/2014 12:04	Carpeta de fitxers	
 <b>monits</b>	15/04/2014 12:04	Carpeta de fitxers	
 <b>monits_sortides</b>	15/04/2014 12:04	Carpeta de fitxers	
 pages	30/05/2014 12:41	Carpeta de fitxers	
 <b>requisuaris</b>	21/04/2014 13:16	Carpeta de fitxers	
 scaffolds	15/04/2014 12:04	Carpeta de fitxers	
 <b>sortides</b>	15/04/2014 12:04	Carpeta de fitxers	
 <b>sortides_usuaris</b>	15/04/2014 12:04	Carpeta de fitxers	
 <b>usuaris</b>	15/04/2014 12:05	Carpeta de fitxers	

Figura 31. Carpetes creades a partir de les taules de la Base de Dades

Paral·lelament, a dins de la carpeta “app”, trobem la carpeta per defecte “webroot”. En aquesta carpeta, s’hi ha de col·locar tots aquells arxius als que es vulguin accedir des de qualsevol Vista o Controlador. Per exemple, hi col·locarem la foto del logotip de l’empresa si la vull mostrar a la pàgina principal o les fotografies dels treballadors i dels usuaris.

## 6.5. Creació dels models en CakePHP

En aquest punt, començava la part ja dedicada a la codificació de l'aplicació web. Per crear una classe model, s'estén la classe per defecte anomenada AppModel. En la definició dels models, s'utilitza la declaració de la variable **\$name** per a definir el nom amb el que ens referirem al model a l'hora d'escriure el codi de les relacions de la resta de models. També depenent de la cardinalitat de la relació amb els altres models es defineixen variables de tipus String o Array de Strings amb el nom corresponent (**\$hasOne** per 1 a 1, **\$hasMany** per 1 a N, **\$belongsTo** per N a 1 o **\$hasAndBelongsToMany** per N a N) on s'hi anota el nom del model que manté aquesta relació amb el model que estem definint. Una altra variable necessària és **\$validate**, on definim les regles de validació dels formularis que utilitzi el model. Com a mostra un fragment de codi de la definició de la classe model Centre a la següent **Figura 32**:

```
<?php
class Centre extends AppModel { //nom de la classe (taula en singular i primera en majúscula)
    //totes les classes model k creem hereden de la classe AppModel
    var $name = 'Centre';

    var $hasMany = 'Usuari'; //relacio 1 a n amb usuaris

    var $validate = array(
        //regles de validacio del formulari per centres

        'NOM' => array ('rule' => 'notEmpty', 'message' => 'El NOM no pot estar buit'),
        'ADRESA' => array ('rule' => 'notEmpty', 'message' => 'El camp no pot estar buit'),
        'CONTACTE' => array ('rule' => 'notEmpty', 'message' => 'El CONTACTE no pot estar buit'),
        'TELEFON1' => array (
            'numeric' => array(
                'rule' => 'numeric',
                'allowEmpty' => false,
                'message' => 'TELEFON 1 escrit malament'),
            'length' => array('rule' => array('minLength', 9), 'message'=>'TELEFON 1 incomplet' )
        )
    );
}
```

Figura 32. Fragment de codi de la classe model Centre

En aquesta darrera figura veiem, per exemple que la regla de validació per a TELEFON1, exigeix que les dades entrades siguin numèriques, amb un mínim de longitud igual a 9 (el màxim ja ve controlat per el número d'espais definits en el camp de la taula, varchar(9)), no pot ser un camp buit i, a més, es defineixen els missatges que es mostraran a l'administrador de l'aplicació des de la vista corresponent.

## 6.6 Creació dels controladors i les vistes en CakePHP

Els controladors i les vistes, són elements que van fortament lligats ja que, les vistes mostren els resultats de les accions que els controladors realitzen sobre els models de l'aplicació. És molt habitual que intercanviïn informació, variables, missatges,...

Tal i com passava amb els models, els controladors es defineixen mitjançant l'extensió d'una classe creada per defecte pel propi Framework, la classe `AppController`. Dins aquesta classe, es defineixen els mètodes amb el mateix nom que crearem a continuació, l'arxiu `.ctp` de la vista. Per exemple, un mètode **function add()**, tindrà una vista associada dins de la carpeta del model corresponent, amb la creació del fitxer **add.ctp**.

En els mètodes es processen totes les peticions de l'administrador de l'aplicació: Buscar, crear, modificar, eliminar, redirigir cap a una pàgina diferent,... Per exemple, el mètode per afegir un nou centre seria com veiem a la **Figura 33**:

```
function add() {  
  
    if(!empty($this->data))  
    { //si shan enviat dades, els guarda a this->data  
        $this->Centre->create(); //i crida la funcio create del model Centre  
        if ($this->Centre->save($this->data) { //si podem guardar les dades a la BD  
            $this->Session->setFlash('Centre guardat correctament!');//mostrem missatge de OK  
            $this->redirect(array('action'=>'index'),null,true);  
        }  
        else { //si no es pot guardar...  
            $this->Session->setFlash('Centre no guardat. Torna-ho a provar.');//  
        }  
    }  
}
```

Figura 33. Fragment de codi del mètode `add()` del model Centre

Pel que fa a les vistes, hem vist ja alguns exemples (veure capítol **5.3. Vistes de l'aplicació**) de com es mostraven en pantalla, però si parlem de la seva codificació, aquesta és bàsicament una combinació de codi HTML amb codi PHP, de manera que és possible alternar fragments de codi d'un llenguatge amb fragments de codi de l'altre. Això ens permet gran flexibilitat i facilitat alhora de programar. Com a mostra, la **Figura 34**, on mostrem el codi per crear el formulari per donar d'alta un nou centre:



```

<table class="table1">
  <td width="400">
    <h3>DADES CENTRE</h3>
    <center>
      <br>
      <?php echo $form->create('Centre'); ?>

      <?php echo $form->input('Centre.NOM', array('label'=>'Nom Centre'));
      echo $form->input('Centre.ADRESA', array('label'=>'Adre&cedil;a'));
      echo $form->input('Centre.CONTACTE', array('label'=>'Contacte 1'));
      echo $form->input('Centre.CONTACTE2', array('label'=>'Contacte 2')); ?>
    </center>
  </td>
  <td width="400">
    <h3>DADES CONTACTAR</h3>
    <br>
    <?php echo $form->input('Centre.TELEFON1', array('label'=>'Tel&egrave;fon 1'));
    echo $form->input('Centre.TELEFON2', array('label'=>'Tel&egrave;fon 2')); ?>

    <br>
    <?php echo $form->input('Centre.EMAIL1', array('label'=>'E-MAIL 1'));
    echo $form->input('Centre.EMAIL2', array('label'=>'E-MAIL 2')); ?>
  </td>
  <td valign="top">
    <br><br><br><br><br><br>
    <?php echo $form->input('Centre.OBSERVACIONS', array('label'=>'OBSERVACIONS')); ?>
  </td>
</table>
<center>
  <table class="table1">
    <td>
      <?php echo $form->end('AFEGIR CENTRE'); ?>
    </td>
  </table>
</center>

```

Figura 34. Fragment de codi de la vista add.ctp del model Centre

## 6.7. Creació d'un sistema d'autenticació amb nom d'usuari i contrasenya

Degut a les dades que manipularà la base de dades, estàvem en l'obligació de crear un sistema per assegurar-nos que la nostra aplicació només pot ser utilitzada per l'administrador de la base de dades.

En CakePHP, la manera d'implementar aquest sistema és molt senzill i automàtic. Veurem que per a crear-lo, necessitem crear una taula, un model, una vista i un controlador nous. Hem considerat no mencionar aquests elements durant l'explicació feta fins ara, ja que creiem que no formen part del disseny de la base de dades pròpiament dita, sinó que són el resultat d'haver d'utilitzar un procés fixat de nou per les convencions de CakePHP. Anem a explicar els passos seguits [17]:

### 1) Afegir component "Auth" a la llista de components d'AppController

En el nostre cas, volem restringir l'accés a la totalitat de funcionalitats de l'aplicació web. Per tant, hem d'indicar a la classe AppController, que utilitzarem el component que gestiona el sistema d'autenticació. Aquest component s'anomena "Auth" i l'implementa la classe

AuthComponent que ve per defecte creada dins del Framework CakePHP. Per fer-ho hem definit la variable \$components de la manera que veiem a la **Figura 35**:

```
<?php
class AppController extends Controller {
    ...
    var $components = array('Auth');//utilitzarem component Auth
}
?>
```

Figura 35. Classe AppController utilitzant component "Auth"

## 2) Crear taula d'usuaris i contrasenyes

Podem definir varis usuaris amb la seva corresponent contrasenya. Els hem de tenir guardats com a registres d'una taula, dins la nostra base de dades. Per convenció, aquesta taula s'ha d'anomenar **users** i ha de tenir, com a mínim, 3 camps:

- id: Clau primària de cada usuari. Numèrica amb autoincrement.
- username: Nom de l'usuari. VARCHAR de 40.
- password: Contrasenya encriptada utilitzant la funció hash SHA1. VARCHAR de 40.

## 3) Creació del Controlador

Novament, es facilita molt la feina ja que, en la creació del controlador, simplement s'ha de seguir els passos descrits per les convencions. Tan sols cal crear un arxiu anomenat users\_controller.php i guardar-lo juntament amb la resta de controladors dins la carpeta "\app\controllers\". El codi que conté és el següent que mostrem a la **Figura 36**. Com es pot apreciar, el mètode login() està buit i es que, el Framework ja s'encarrega de tot el procés de manera automàtica:

```

<?php

class UsersController extends ApplicationController{

    var $name = 'Users';

    function login()
    {
        .....
    }

    function logout() {
        $this->redirect($this->Auth->logout());
    }

}
?>

```

Figura 36. Codi del controlador UsersController

#### 4) Creació de la vista

De la mateixa manera, per convenció, creem la carpeta “users” dins la carpeta “\app\views\” i hi col·loquem l’arxiu encarregat de la vista, que s’ha de dir “login.ctp”. És un arxiu que, com les altres vistes, conté una combinació de codi HTML amb PHP, i que s’encarrega de generar el formulari per demanar les credencials a l’administrador de l’aplicació. Una mostra a la **Figura 37**:

```

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es"> <!--idioma-->

    <body>
        <h1>Accés Restringit - Log in: </h1>

        <?php
            echo $session->flash('auth');
            echo $form->create('User', array('action' => 'login'));
            echo $form->input('username');
            echo $form->input('password');
            echo $form->end('Login');
        ?>

    </body>
</html>

```

Figura 37. Codi de la vista login.ctp

#### 5) Introduir els registres usuari/contrasenya a la taula

Aquest pas, s’ha dut a terme des de phpMyAdmin ja que, per agilitzar el procés, hem introduït manualment l’usuari i la contrasenya desitjades dins de la taula users. La contrasenya s’ha d’introduir codificada en SHA1. A la web, hi ha molts convertidors text-> SHA1. Hem utilitzat el

de la pàgina <http://online-code-generator.com/sha1-hash-with-optional-salt.php> [18] i a la taula users hem introduït l'únic (en aquests moments) usuari vàlid i la seva contrasenya en SHA1. La informació queda guardada tal i com la veiem a la **Figura 38**:



	id	username	password
<input type="checkbox"/>  	1	actua	27aa6311ff68e11e156d9c9b19082824f8adfd6a

Figura 38. Contingut actual de la taula users

## 6) Personalitzar opcions

Sense que hagi de ser un pas obligatori (en aquest punt el sistema d'autenticació ja funciona), hem cregut convenient modificar algunes opcions que venen definides per defecte.

- **Missatges de la vista login.ctp**: Simplement hem traduït els missatges que apareixien per defecte en la pantalla d'identificació. Això es fa dins el codi de l'arxiu de la classe AuthComponent anomenat "Auth.php". Per exemple, com veiem a la **Figura 39**:

**Accés Restringit - Log in:**

**Error d'identificació. Username o Password incorrecte**

Username

Password

Figura 39. Exemple de missatge en error d'autenticació

- **Definir "Session Time Out"**: És a dir, indicar quants minuts sense activitat en l'aplicació deixem de marge per a tornar a demanar credencials. Per defecte, hi ha definits 20 minuts, temps que hem cregut excessiu. Finalment, consensuant amb l'empresa, hem definit un "Session Time Out" de 10 minuts. Això es realitza modificant una línia de codi a l'arxiu "core.php" situat a la carpeta "\app\config\". Mostrem la **Figura 40** amb el fragment de codi en qüestió amb alguns exemples de funcionament en els comentaris:

```
/**
 * Session time out time (in seconds).
 * Actual value depends on 'Security.level' setting.
 * 120 son 2 minuts -> si session.timeout es high 2x10 => 20 minuts
 * 360 son 6 minuts -> si session.timeout es high 6x10 => 1 hora
 * 8640 son 144 minuts -> si session.timeout es high 24 hores
 */
Configure::write('Session.timeout', '60');//10 minuts
```

Figura 40. Session Time Out configurat a 10 minuts

## 7. Tests i Avaluació

En els tests i avaluació de l'aplicació, hem intentat seguir la cronologia de fases prevista i explicada en el capítol **2.4. Planificació de Tasques**. Cal dir que s'han superat finalment totes les proves de maneig de l'aplicació, tot i que se'n continuarà fent un seguiment exhaustiu del funcionament almenys durant la pròxima campanya de sortides de l'empresa prevista per l'estiu d'enguany. A continuació, anem a descriure el procés de tests i avaluació realitzat:

### **TEST 1: FUNCIONALITATS BÀSIQUES DE L'APLICACIÓ**

En aquest test, s'ha comprovat si la creació, modificació i eliminació dels quatre elements bàsics (Centres, Llocs, Treballadors i Usuaris) es realitzava correctament. S'han forçat situacions voluntàriament on l'aplicació havia de prohibir la creació de l'element degut a no respectar les regles de validació dels models (per exemple, deixar el número de telèfon incomplet o deixar el camp nom buit) i també s'ha provat de crear elements que tinguessin alguna dada en conflicte degut a les restriccions dels índexs UNIQUE (com intentar crear dos usuaris amb DNIs idèntics o dos llocs amb el mateix nom). Veiem un exemple de missatges d'error en la **Figura 41**:

Centre no guardat. Torna-ho a provar.

### Introduïr NOU CENTRE

DADES CENTRE	DADES CONTACTAR	OBSERVACIONS
<p>Nom Centre</p> <input type="text"/> <p><b>El NOM no pot estar buit</b></p>	<p>Telèfon 1</p> <input type="text"/> <p><b>TELEFON 1 incomplet</b></p> <p>Telèfon 2</p> <input type="text"/>	<input type="text"/>
<p>Adreça</p> <input type="text"/> <p><b>El camp no pot estar buit</b></p>	<p>E-MAIL 1</p> <input type="text"/>	
<p>Contacte 1</p> <input type="text"/> <p><b>El CONTACTE no pot estar buit</b></p>	<p>E-MAIL 2</p> <input type="text"/>	
<p>Contacte 2</p> <input type="text"/>		

Figura 41. Missatges d'error al intentar crear un Centre

### **TEST 2: CREACIÓ, MODIFICACIÓ I ELIMINACIÓ DE SORTIDES**

Seguint el sistema de l'anterior, en aquest segon test hem creat sortides correctament. En segon lloc, em intentat crear sortides al mateix lloc i dates, i hem comprovat que ens prohibia fer-ho degut a la restricció UNIQUE explicada en el capítol **6. Implementació**.

El següent pas ha set la d'assignar treballadors. Aquí el que s'havia de provar és que no apareguessin treballadors que, en les dates de la sortida, estiguessin de servei en una altra. I,

efectivament, aquells treballadors ocupats durant la sortida, no apareixen en la llista per ser assignats.

Igualment, en els usuaris, no es permet apuntar dos vegades a un usuari a la mateixa sortida, ni un usuari en dues sortides que passin alhora. Els requeriments dels usuaris es poden definir al moment d'apuntar-los, o bé, un cop ja apuntats.

Finalment, al eliminar una sortida, s'eliminen tots els registres de les taules amb alguna relació amb la sortida: els registres corresponents de monits\_sortides, els de sortides\_usuaris i tots els requisits dels usuaris per aquella sortida.

### ***TEST 3: CREACIÓ, MODIFICACIÓ I ELIMINACIÓ D'ACOMPANYAMENTS***

En aquest test, hem creat els acompanyaments de les dos maneres possibles que hi ha. Per una banda, assignant des d'un principi el treballador que el durà a terme. Per l'altra, sense assignar-lo. En els dos casos ens permet crear l'esdeveniment i tota la seva informació sense problemes.

Per a modificar-lo, també podem accedir-hi des de qualsevol dels dos mètodes per calcular les facturacions dels quilometratges. Primer, podem triar l'any a consultar i el mes. Des d'allà ens permet veure i modificar o eliminar els acompanyaments d'aquell mes. Si canviem el mes en el dia de l'acompanyament, aquest es mostra en la pantalla del mes corresponent.

Quelcom semblant trobem si accedim des del nom del treballador. Llavors triem l'any i el mes i ens apareixen correctament només els acompanyaments del treballador en aquell mes. Si canviem de treballador, aquell acompanyament passa a llistar-se en els acompanyaments del nou treballador assignat en el mes corresponent.

Els acompanyaments sense treballador assignat no es mostren, lògicament, si hem accedit a al consulta des del segon mètode (per nom de treballador).

En els dos casos, es poden eliminar sense problemes els acompanyaments i s'ha comprovat que qualsevol dels canvis definits en aquest TEST 3 es veuen reflectits correctament en la suma total de les facturacions (tan dels serveis d'un mes, com del resum mensual dels treballadors).

### ***TEST 4: CONSULTAR FACTURACIONS***

Com hem comentat en el TEST3, les facturacions tan dels serveis com dels treballadors, s'actualitzen correctament si fem canvis en el treballador o dia assignats.

A més, hem fet proves de modificar tan el número de quilòmetres del servei o del treballador com el preu de cadascun d'ells que, en un principi, surt per defecte ja definit. En tots els casos la suma total del preu dels quilometratges dóna el resultat correcte i esperat.

### ***TEST 5: PROVAR EL SISTEMA D'AUTENTIFICACIÓ***

En principi, nosaltres hem deixat un sol usuari i contrasenya correctes per a identificar-se com a l'administració de l'aplicació. Tot i això, hem creat varis usuaris per comprovar si ens deixava accedir al menú principal amb cadascuna de les tuples usuari/contrasenya guardades en la taula "users". En tots els casos, hem pogut identificar-nos amb èxit. De la mateixa manera, hem provat

amb usuaris i/o contrasenyes no vàlides i apareix correctament el missatge d'error d'identificació.

Per acabar aquest test, hem esperat després de 10 minuts d'inactivitat (el Session Time Out que hem definit en consens amb l'empresa com dèiem a l'apartat corresponent de **6. Implementació**) i hem provat de continuar utilitzant l'aplicació des del punt on ens trobàvem. Ens ha aparegut un missatge demanant-nos de nou les credencials com mostrem a la **Figura 42**:



The image shows a login form with the following elements:

- Title: **Accés Restringit - Log in:**
- Error message: **No estàs autoritzat a accedir en aquesta pàgina.**
- Username field: A text input box with the label "Username" above it.
- Password field: A text input box with the label "Password" above it.
- Login button: A green button with the text "Login".

Figura 42. Missatge que apareix en exhaurir el "Session Time Out"

#### **TEST 6: REPETICIÓ DELS TESTS PER PART DE L'ADMINISTRADOR DE L'APLICACIÓ**

A mode d'ensenyament i demostració de l'aplicació i, per tal de tornar a comprovar que els tests tenen un resultat satisfactori quan es realitzen a la màquina on s'utilitzarà definitivament l'aplicació, hem dirigit a l'administrador de l'aplicació mentre li fèiem repetir cadascun dels anteriors tests. Al tractar-se del primer cop que s'utilitzava l'aplicació al complet, amb més o menys rapidesa, s'han pogut comprovar tots els tests.

#### **TEST 7: SEGUIMENT**

Com comentàvem al començament d'aquest capítol, farem un seguiment intensiu del funcionament de l'aplicació durant aquest estiu, període de més volum de feina de l'empresa.





## 8. Conclusions

Primerament, ens agradaria comentar que l'objectiu principal que se'ns demanava per aquest projecte (crear una aplicació web que gestioni la base de dades de l'empresa) s'ha resolt satisfactòriament.

També cal dir, que ens ha suposat un gran avantatge el fet d'utilitzar XAMPP, degut a la seva fàcil instal·lació i configuració. Això ens va permetre guanyar força temps al començament. D'altra banda, l'ús del XAMPP no té cap mena de garantia per part dels seus creadors, quelcom sempre a tenir en compte si s'ha d'implementar algun altra tipus d'aplicació per a tercers.

La decisió d'utilitzar PHP com a llenguatge de programació creiem que ha estat l'adequada ja que, personalment, és un llenguatge del que ja en teníem força experiència en utilitzar-lo. A més, parlant més concretament, ens permetia combinar les seccions de codi de les vistes amb el llenguatge HTML, cosa que ens feia guanyar gran flexibilitat i comoditat en el disseny d'aquestes.

Un dels grans interrogants que se'ns va presentar a l'hora de començar, va ser si utilitzar algun software estructurat com a punt de partida, o bé, programar tot el sistema des de zero. En aquest punt, creiem que la utilització de CakePHP, per una banda, ens ha permès estalviar gran quantitat de codi ja que, moltes de les funcions implementades (com, per exemple, la d'editar els models) eren bàsicament una còpia les unes de les altres. Per contra, cal dir que, aquest codi, ens va resultar una mica complex al començament degut a les convencions a seguir. Degut a aquestes normes d'utilització, el codi sovint donava error si no es seguien al peu de la lletra.

Amb això, CakePHP ens permetria, en un cas hipotètic, la possibilitat de modificar, actualitzar o estendre les funcionalitats de l'aplicació només modificant petites parts. A més, per casos on es pugui preveure una similitud en el funcionament d'un altre sistema, es podria partir tranquil·lament del codi utilitzat en aquest projecte i adequar-lo a les noves necessitats.

Cal dir, per altra banda, que la programació en CakePHP està orientada a persones amb un coneixement mitjà-alt de programació. Això fa que, si no estàs molt avesat a programar, corris el risc d'agafar certes males costums degut a que "facilita" la feina en alguns casos de manera no massa ortodoxa. Un exemple, seria les sentències per a buscar un registre en concret en una taula. Enlloc d'utilitzar una sentència SQL, que seria el més natural, és suficient en fer:

```
$variable = $this->Model->FindAllbyid($id);
```

Un altre inconvenient al utilitzar CakePHP, va ser el complicat anàlisi de les classes que utilitzava per defecte el Framework per tal de oferir l'arquitectura MVC d'una manera tan pautada i estructurada. És una feina que en principi no era obligatòria de fer però, la vam realitzar per a un millor aprofundiment en l'entesa del seu funcionament.

Si entrem a comentar el disseny propi de l'aplicació, cal dir que algunes coses les podríem haver estructurat d'alguna manera diferent, però en la majoria d'aquests casos, depeníem directament de les peticions i les necessitats de l'administrador de l'aplicació, és a dir, del funcionament propi de l'empresa.

Per a fer aquest disseny, hem volgut acostar-nos el més possible al sistema de menús, botons i enllaços que utilitzen les aplicacions web centrades en l'usuari com serien les web 2.0. Cal dir que això ho trobem tot un encert ja que, en pocs minuts d'utilització, de seguida s'aprèn el funcionament i la navegació per les vistes de les pàgines de l'aplicació.

En resum, podríem dir que, tot i haver pogut utilitzar un sistema de en el que partíssim des de 0 que ens permetés la llibertat total en la programació i disseny de l'aplicació, l'ús del CakePHP dins del paquet XAMPP ha acabat resolent molts més problemes dels que ha plantejat, com també ens ha permès una major rapidesa a l'hora de programar i, inclús, de realitzar canvis, adaptacions, correccions o ampliacions de l'aplicació.

## Referències

- [1] Generalitat de Catalunya. (15 / Octubre / 2013). *Persones amb discapacitat reconeguda. Per tipus de discapacitat. Comarques, àmbits i províncies.*
  
- [2] ACTUASERVEIS SCP. (15 / Octubre / 2013). *ActuaServeis | Vetllem per la qualitat de vida.* Recollit de <http://www.actuaserveis.com/>
  
- [3] Creative Commons. (15 / Octubre / 2013). *Entorno cliente/servidor.* Recollit de <http://es.kioskea.net/contents/148-entorno-cliente-servidor>
  
- [4] Microsoft Development Network. (18 / Octubre / 2013). *Model-View-Controller.* Recollit de <http://msdn.microsoft.com/en-us/library/ff649643.aspx>
  
- [5] Wikipedia.org. (15 / Octubre / 2013). *Comparison of web server software.* Recollit de [http://en.wikipedia.org/wiki/Comparison\\_of\\_web\\_server\\_software](http://en.wikipedia.org/wiki/Comparison_of_web_server_software)
  
- [6] Dmitrij. (15 / Octubre / 2013). *NGINX VS CHEROKEE VS APACHE VS LIGHTTPD.* Recollit de <http://www.whisperdale.net/11-nginx-vs- Cherokee-vs-apache-vs-lighttpd.html>
  
- [7] Wikipedia.org. (15 / Octubre / 2013). *Servidor HTTP Apache.* Recollit de [http://es.wikipedia.org/wiki/Servidor\\_HTTP\\_Apache](http://es.wikipedia.org/wiki/Servidor_HTTP_Apache)
  
- [8] Wikipedia.org. (3 / Novembre / 2013). *Navegador web.* Recollit de [http://ca.wikipedia.org/wiki/Navegador\\_web](http://ca.wikipedia.org/wiki/Navegador_web)
  
- [9] Wikipedia.org. (12 / Octubre / 2013). *Ruby (programming language).* Recollit de [http://en.wikipedia.org/wiki/Ruby\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/Ruby_(programming_language))
  
- [10] Dicken, C. (11 / Octubre / 2013). *HTML Goodies. The Ultimate html resource.* Recollit de <http://www.htmlgoodies.com/primers/database/top-5-databases-for-web-developers.html>

- [11] SoftMexico. (12 / Octubre / 2013). *7 Herramientas alternativas a XAMPP*. Recollit de <http://www.sinapsystemx.net/internet/7-herramientas-alternativas-a-xampp/>
- [12] Wikipedia.org. (12 / Octubre / 2013). *CakePHP*. Recollit de <http://en.wikipedia.org/wiki/CakePHP>
- [13] Wikipedia.org. (15 / Octubre / 2013). *XAMPP*. Recollit de <http://es.wikipedia.org/wiki/XAMPP>
- [14] The Bakery. (11 / Novembre / 2013). *Blog Tutorial*. Recollit de <http://book.cakephp.org/2.0/en/tutorials-and-examples/blog/blog.html>
- [15] The Bakery. (11 / Octubre / 2013). *Cake Conventions*. Recollit de <http://bakery.cakephp.org/articles/sajt/2007/02/25/cake-conventions>
- [16] Apache Friends. (21 / Novembre / 2013). *XAMPP Installers and Downloads for Apache Friends*. Recollit de <https://www.apachefriends.org/index.html>
- [17] The Bakery. (29 / Abril / 2014). *Authentication*. Recollit de <http://book.cakephp.org/1.3/en/The-Manual/Core-Components/Authentication.html>
- [18] Cagintranet Networks . (29 / Abril / 2014). <http://online-code-generator.com/sha1-hash-with-optional-salt.php>.

## Índex de figures

Figura 1. Taula-Resum de l'Anàlisi de Riscos .....	12
Figura 2. Diagrama de Gantt .....	14
Figura 3. Esquema client-servidor amb fluxos de petició i resposta .....	18
Figura 4. Esquema de col·laboració entre els components MVC .....	19
Figura 5. Gràfic d'utilització de Servidors Web .....	21
Figura 6. Logo de CakePHP .....	26
Figura 7. Logo del projecte XAMPP .....	28
Figura 8. Vista de l'arbre de directoris principal de XAMPP .....	29
Figura 9. Diagrama casos d'ús .....	35
Figura 10. Diagrama de classes del Framework i l'aplicació Web .....	47
Figura 11. Diagrama ER .....	49
Figura 12. Model Relacional de Taules .....	55
Figura 13. Formulari per donar d'alta un centre .....	60
Figura 14. Formulari per modificar les dades d'un usuari .....	61
Figura 15. Vista d'eliminació d'un usuari .....	61
Figura 16. Vista de la cerca d'un usuari .....	62
Figura 17. Vista principal de la gestió de sortides .....	63
Figura 18. Formulari per modificar les dades d'una sortida .....	63
Figura 19. Vista de l'assignació de treballadors a una sortida .....	64
Figura 20. Vista principal d'apuntar usuaris (amb el buscador) .....	65
Figura 21. Formulari per definir els requisits d'un usuari en una sortida .....	65
Figura 22. Vista principal de la gestió de sortides amb l'opció "BORRAR" remarcada .....	66
Figura 23. Formulari per crear un acompanyament (sense treballador assignat) .....	67
Figura 24. Formulari per modificar un acompanyament (amb treballador ja assignat).....	68
Figura 25. Vista principal dels acompanyaments del maig de 2014 .....	68
Figura 26. Vista principal dels acompanyaments realitzat per un treballador.....	69
Figura 27. Control Panel amb Apache i MySQL funcionant .....	71
Figura 28. Taula resum de seguretat. ....	72

Figura 29. Taula centres creada .....	73
Figura 30. Índexs de la taula centres .....	74
Figura 31. Carpetes creades a partir de les taules de la Base de Dades .....	74
Figura 32. Fragment de codi de la classe model Centre .....	75
Figura 33. Fragment de codi del mètode add() del model Centre .....	76
Figura 34. Fragment de codi de la vista add.ctp del model Centre .....	77
Figura 35. Classe ApplicationController utilitzant component "Auth" .....	78
Figura 36. Codi del controlador UsersController .....	79
Figura 37. Codi de la vista login.ctp .....	79
Figura 38. Contingut actual de la taula users .....	80
Figura 39. Exemple de missatge en error d'autenticació .....	80
Figura 40. "Session Time Out" configurat a 10 minuts .....	80
Figura 41. Missatges d'error al intentar crear un Centre .....	81
Figura 42. Missatge que apareix en exhaurir el "Session Time Out" .....	83