

Títol: Aplicació web de gestió d'un estudi d'arquitectura

Volum: 1

Alumne: Joaquim Hilari Horts

Director/Ponent: Joan Carles Gil Martín

Departament: Organització d'Empreses

Data:

DADES DEL PROJECTE

Títol del Projecte: Aplicació web de gestió d'un estudi d'arquitectura

Nom de l'estudiant: Joaquim Hilari Horts

Titulació: Enginyeria Tècnica en Informàtica de Sistemes

Crèdits: 22,5

Director/Ponent: Joan Carles Gil Martín

Departament: Organització d'Empreses

MEMBRES DEL TRIBUNAL *(nom i signatura)*

President: Francisco Javier Llinàs Audet

Vocal: Maria Teresa Abad Soriano

Secretari: Joan Carles Gil Martín

QUALIFICACIÓ

Qualificació numèrica:

Qualificació descriptiva:

Data:

Índex

1	Introducció.....	7
1.1	Introducció del funcionament de l'estudi d'arquitectura del nostre client	7
1.2	Motivació	8
1.3	Situació Inicial	9
1.4	Objectius	9
1.5	Tecnologia Web Utilitzada	10
1.5.1	Tecnologies del client.....	10
1.5.2	Servidors web	11
1.5.3	Servidor web Apache	13
1.5.4	Llenguatges de programació	13
1.5.5	PHP.....	17
1.5.6	Bases de dades.....	20
1.5.7	MySQL.....	21
2	Funcionalitats i requeriments del sistema	24
2.1	Requeriments funcionals.....	24
2.1.1	Els projectes	24
2.1.2	Els contactes	25
2.1.3	Els usuaris	25
2.2	Requeriments no funcionals.....	26
2.2.1	Interfície gràfica	26
2.2.2	Factors de qualitat	29
3	Planificació i estudi econòmic	31
3.1	Planificació temporal	32
3.2	Estudi econòmic.....	33
3.2.1	Costos dels recursos humans.....	33
3.2.2	Costos del material software	35
3.2.3	Cost total del projecte.....	35
3.3	Riscs del projecte	37
4	Especificació	39
4.1	Què és UML?	39
4.2	Model Conceptual.....	41
4.2.1	Diagrama de classes	41
4.3	Model de casos d'ús.....	42
4.3.1	Diagrama de casos d'ús.....	43
4.3.2	Especificació de casos d'ús	45
4.4	Model d'estats.....	65
4.4.1	Diagrama d'estats.....	65
5	Disseny	66
5.1	Arquitectura d'una aplicació web	67
5.2	Decisions de disseny	68

5.3	Disseny de la capa de presentació.....	68
5.3.1	Mòdul de Projectes.....	69
5.3.2	Mòdul de Contactes.....	72
5.3.3	Mòdul d'Usuaris.....	77
5.4	Disseny de la capa de negoci.....	79
5.4.1	Mòdul de Projectes.....	80
5.4.2	Mòdul de Contactes.....	80
5.4.3	Mòdul d'Usuaris.....	80
5.5	Disseny de la capa de dades.....	81
5.5.1	Disseny lògic de les dades.....	81
5.5.2	Disseny físic de les dades.....	83
6	Implementació.....	87
6.1	Esquema del funcionament.....	89
6.2	Seguretat.....	90
6.2.1	Autenticació.....	91
6.2.2	Encriptació de les contrasenyes.....	92
6.2.3	Seguretat a les pàgines.....	93
7	Proves.....	95
8	Manteniment.....	96
9	Conclusions.....	97
9.1	Anàlisi de desviacions.....	97
9.2	Cost del projecte.....	99
9.2.1	Cost de recursos humans i materials.....	100
9.2.2	Cost d'instal·lació i implantació del sistema.....	101
9.3	Objectius finals.....	102
9.4	Millores i ampliacions.....	103
9.5	Futur de l'aplicació web.....	104
9.6	Valoració personal.....	105
10	Bibliografia.....	107

1 Introducció

Aquest projecte va destinat a una empresa d'arquitectura d'unes 40 persones (arquitectes, aparelladors, enginyers, informàtics, delineants i modeladors 3D entre d'altres) i que no tenen una constància del que es va realitzant durant el dia a dia i, a més, faciliti els seus seguiments laborals.

Amb la incorporació d'aquesta aplicació podran fer les mateixes tasques que es realitzen durant el dia a dia, així com un seguiment més aproximat i més actualitzat de tots els treballs realitzats, però d'una manera més centralitzada.

Es va iniciar a l'octubre de 2006 amb un client real ubicat a Sevilla i la part pràctica va estar completament finalitzada i entregada a l'estudi d'arquitectura al maig de 2007. El fet de treballar a jornada complerta a una empresa del sector de les consultories, va impedir-me complir els terminis d'execució tal com es van establir. Després, la realització de la memòria va resultar ser especialment lenta, ja que coincidia amb pics de treball dins de la consultoria on treballava i amb un canvi de vivenda.

Per últim, la memòria va haver de ser parada completament després de crear una empresa personal dins del sector de l'entreteniment i les subhastes d'Internet. El nivell de treball i dedicació que em va exigir la creació d'aquesta empresa, em va fer impossible seguir avançant amb el projecte fi de carrera. Per això, després d'uns anys i vendre l'empresa personal, he pogut finalitzar-la.

1.1 Introducció del funcionament de l'estudi d'arquitectura del nostre client

L'estudi compte amb unes 40 persones. Entre elles podem trobar arquitectes, aparelladors, enginyers, informàtics, delineants i modeladors 3D entre d'altres.

Actualment els projectes d'obra són catalogats segons el seu any i el seu número d'entrada. Per exemple, existiria el projecte 57/06, projecte nº57 de l'any 2006, amb nom "84 vivendes a

Torremolinos". Amb un Excel i una taula Access realitzen el seguiment, associen un promotor, un arquitecte, una situació i poc més.

També tenen en un servidor on s'allotgen tots els projectes agrupats per anys, i dins de cada any tenen tota la informació que fa referència al projecte i seguiment d'obra, així com tota la documentació i fotografies.

Una part important és el seguiment d'obra, i malauradament és el tema més descontrolat. Un cop es tenen els projectes dibuixats i es dona la llicència, és quan cal passar-se per la obra per veure com s'està realitzant. Un cop per setmana més o menys es va a la obra, es fan fotografies, es plantegen dubtes i s'intenten solucionar en poc temps.

Un cop a l'oficina es realitza un informe amb el que s'ha vist, es guarden les fotografies en una carpeta i s'apunten les tasques a realitzar en *post-it's* o quaderns.

1.2 Motivació

El projecte va començar amb una conversa amb un amic arquitecte. M'explicava els seus entrebancs i comentàvem diferents possibilitats per donar solució als problemes.

Ens varem posar d'acord i varem acordar realitzar una aplicació que permetés agilitzar i simplificar el treball diari.

Realitzar una aplicació per un conegut és sempre molt més gratificant que realitzar-la per a gent desconeguda. La part compradora té la certesa que la part venedora no intentarà realitzar cap tipus d'engany. I la part venedora sap que podrà tenir un tracte molt més sincer i personal, a part de saber que els riscos econòmics haurien de ser mínims.

Per això vaig decidir realitzar el projecte sobre l'aplicació de l'estudi d'arquitectura. Tenia la oportunitat d'ajudar a millorar el treball diari d'un company a la vegada que realitzava el projecte final de carrera.

1.3 Situació Inicial

Qualsevol estudi d'arquitectura ha de gestionar nombrosos contactes, des d'aparelladors, fins a ajuntaments, passant per promotors i gestors, entre d'altres.

Inicialment, tots els contactes es gestionaven des de fulles d'Excel, des d'on s'intentava establir un cert ordre i una certa agrupació. Però les cerques es limitaven a taules dinàmiques i les relacions entre contactes es limitaven a agrupacions per tipus.

També s'ha de gestionar grans quantitats de documents, plànols i tasques dels projectes que porti l'estudi.

Tots aquests documents s'agrupaven en carpetes dins d'arxivadors i sempre calia tenir la cura d'ordenar degudament tot per tal de poder localitzar tot allò necessari ràpidament. Els documents digitals es solien desar en un servidor amb accés remot per a tota l'oficina, però no hi havia manera de controlar quan s'afegia nou material o quan es realitzaven modificacions en algun projecte en concret.

En un estudi d'arquitectura, es solen realitzar moltes fotografies que documenten i complementen totes les obres visitades i controlades pels arquitectes. Totes aquestes fotografies també es desen en format digital en el servidor que comentaven anteriorment. Per tal d'establir una cronologia a les fotografies, cal numerar-les o bé reanomenar-les de tal manera que ens indiquin la data i el lloc en que s'han realitzat.

1.4 Objectius

L'objectiu del nostre projecte és realitzar una aplicació que permeti millorar el treball diari de l'estudi que tractem. Però també, realitzar una aplicació suficientment genèrica com per que pugui ser utilitzada per a qualsevol altre empresa que realitzi activitats semblants.

Amb la creació d'una senzilla agenda relacional, solucionarem les dificultats a l'hora de buscar contactes. Permetrem crear relacions i així s'agruparan de manera automàtica. Per exemple, si permetem crear un ajuntament com un contacte, podrem establir relacions amb

l'alcalde, la oficina tècnica o fins i tot podrem relacionar-hi els arquitectes que hi estiguin empadronats. D'aquesta manera, sempre que consultem el contacte de tipus ajuntament, podrem consultar, també, tots els altres contactes relacionats.

Permetrem crear relacions entre els projectes i tots els seus contactes, així serà senzill saber quin arquitecte haurem de trucar que estigui assignat al projecte.

Realitzant mòduls de seguiment d'obra i de seguiment d'expedients, podrem organitzar i classificar tota la documentació, fotografies i plànols d'un projecte, així com definir els diferents estats en que es trobi.

Digitalitzant la documentació més important, serà més senzill de localitzar, realitzar cerques, seguiments de modificació o qualsevol altre comprovació que es vulgui realitzar.

Com que el nostre estudi, ara per ara, no vol modificar el sistema que utilitzen per desar la documentació, realitzarem el seguiment d'expedients i s'aplaçarà per a futures ampliacions tot el seguiment d'obra, que permetria desar i organitzar totes les documentacions i fotografies d'un projecte.

1.5 Tecnologia Web Utilitzada

Definirem les tecnologies necessàries per la nostra aplicació, les seves característiques i per a quina raó les utilitzarem.

1.5.1 Tecnologies del client

El nostre client ja disposa d'una màquina amb sortida a Internet ubicada al mateix estudi d'arquitectura. En aquesta màquina desen tots els plànols, fotografies, contactes i documents d'un projecte. En la mateixa màquina tenen un servidor web Apache, on està allotjada la pàgina web de l'estudi, i una base de dades en MySQL.

Les pàgines allotjades en el servidor del client estan realitzades en el llenguatge de programació PHP. I també és el llenguatge en que vol que estigui realitzada la nostra aplicació. La raó bàsica d'utilitzar el PHP, i no un altre llenguatge de programació, és la facilitat de no haver de modificar la configuració del seu servidor web Apache, i el fet de no tenir accés físic ni remot al servidor. Només se'ns permetrà accedir via FTP per tal de pujar noves versions de l'aplicació.

1.5.2 Servidors web

Un servidor web és un programa que implementa el protocol HTTP (*hypertext transfer protocol*). Aquest protocol està dissenyat per transferir el que anomenem hipertexts, pàgines web o pàgines HTML (*hypertext markup language*): texts complexos amb enllaços, formularis, botons i objectes incrustats com poden ser animacions o reproductores de música. A més d'implementar el protocol HTTP, a partir de mòduls, llibreries o nativament, poden interpretar llenguatges de programació o executar codis precompilats.

Avui en dia, podem dir que el servidor web Apache és el més utilitzat a Internet. El podem trobar en el 50% dels casos (sobre el 70% l'any 2005). Pel simple fet de ser el més utilitzat, moltes persones amb desconexió del mercat de servidors, escolliran l'Apache com a primera opció. Ens podríem atrevir, fins i tot, a assegurar que aquesta és la raó per la qual el nostre client haurà triat aquest servidor web.

La seva bona acceptació bé donada pels seus principis bàsics: velocitat, simplicitat, multiplataforma i facilitat en el desenvolupament distribuït. A més, la seva primera versió va ser al 1995 i mai ha parat de desenvolupar-se, ja que parlem d'un projecte de Software lliure. Això li aporta la suficient experiència i robustesa necessàries per un servidor web.

Un altre servidor molt reconegut és el Internet Information Services de Microsoft. De fet, es tracta del segon servidor web més utilitzat a Internet, encara que possiblement, el més utilitzat en entorns corporatius. Més que un servidor web, podem dir que és un conjunt d'aplicacions que ens permeten crear, configurar i administrar llocs web, a més d'incloure altres serveis d'Internet per la plataforma Windows. Una d'aquestes aplicacions és el Internet Information Server o més conegut com a IIS. A part d'aquest, ens permet tenir altres serveis, com el FTP

(servei de transferència de fitxers), el SMTP (servei de transferència de correus electrònics) o el NNTP (servei de transferència de notícies), entre d'altres.

El fet de no ser Software lliure i ser utilitzat per entorns corporatius ha ajudat a ser atacat constantment per persones que s'aprofitaven dels *exploits*¹ i aconseguien malmetre serveis web.

Si en algun moment hem de comparar els dos servidors web més utilitzats d'Internet, cal tenir en compte que ambdós són molt bones opcions. Tant l'IIS com l'Apache tenen multitud de documentació fàcil de trobar, per tant, ens serà més fàcil resoldre els problemes de forma ràpida. Això si, l'IIS ens semblarà més senzill de configurar i administrar, però l'Apache, tot i la seva complexitat, ens permetrà administrar d'una manera més precisa tots i cada un dels detalls del servidor.

Apart d'aquests dos servidors web, podem trobar altres servidors d'aplicacions, que es diferencien per fer un ús més extensiu del contingut dinàmic del servidor i per tenir certa persistència de la bases de dades. També solen estar limitats i optimitzats a un llenguatge de programació. Per exemple, podem trobar diferents servidors d'aplicacions especialitzats en Java: WebSphere, JBoss, Borland ES, Oracle AS, BEA WebLogic Server o Tomcat, com a mòdul pel servidor web Apache; especialitzat en .NET tenim el Framework .NET de Windows; en Ruby tenim Ruby on Rails.

Ja hem dit que els servidors d'aplicacions solen estar limitats a un sol llenguatge de programació, però els servidors webs permeten configurar mòduls externs per fer-los compatibles amb diferents llenguatges de programació. Així, encara que no tenim opció de triar en el cas del nostre client, ens decantaríem per Apache, un servidor web molt flexible i altament configurable, que ens permetria variar les seves característiques per tal d'adaptar-lo a futures noves necessitats.

¹ Exploit: forma d'utilitzar un error del sistema per accedir-hi i obtenir certs privilegis.

1.5.3 Servidor web Apache

El servidor web Apache es desenvolupa dins del projecte HTTP Server (httpd) de l'Apache Software Foundation, que és una organització creada per donar suport als projectes de software que portin la denominació d'Apache.



La seva avantatja més important és la gran popularitat que ha adquirits amb el temps, això ens permetrà trobar informació i ajuda molt fàcilment. Si no som experts en aquest servidor, sovint ens caldrà buscar informació per tal de resoldre els nostres problemes, però gràcies a la multitud d'ajuda existent, tinguem el problema que tinguem, tenim moltes possibilitats de no ser els únics amb aquest problema i poder trobar la solució ràpidament.

Per tal de tractar els errors del servidor, Apache ens proporciona certes característiques altament configurables, a més de permetre'ns visualitzar i administrar bases de dades d'autenticació i negoci del contingut. Però com ja comentàvem en l'apartat anterior, no és molt amigable a la hora de configurar-lo, ja que no té una interfície gràfica que faciliti les tasques de configuració i manteniment, tot ho haurem de realitzar a partir dels fitxers de configuració del servidor, encara que estan altament comentats i explicats dins dels propis fitxers i en tutorials i fitxers d'ajuda que no ens costarà trobar a través d'Internet.

Si volem utilitzar PHP amb aquest servidor web, que és el que vol fer el nostre client, ens caldrà incorporar un mòdul que permeti al servidor interpretar aquest llenguatge de programació. Però això pot ser tant senzill com afegir una línia al fitxer de configuració i assegurar-nos que tenim una llibreria dins la carpeta del servidor.

1.5.4 Llenguatges de programació

El llenguatge de programació és una part important a la hora de decidir com farem la nostra aplicació web. D'ell dependrà que el desenvolupament sigui més o menys senzill, ràpid o segur. Intentarem comparar els 3 llenguatges principalment utilitzats en el món web: Java, PHP i ASP.NET

PHP és l'únic llenguatge dels 3 que comentem que no és precompilat, és a dir, que es tracta d'un llenguatge d'*scripting*². Això ens obliga a executar el codi per detectar problemes de dependències de fitxer o errors de sintaxis. Els llenguatges que són *strong type*³ com Java i ASP.NET tenen la possibilitat de comprovar-ho abans. Aquest inconvenient no és massa important, ja que tenim eines de treball com les IDE's (Entorns de Desenvolupament Integrat), que permeten comprovar la sintaxis i es poden detectar molts errors abans de la primera execució.

Podem comparar els diferents llenguatges comprovant alguns dels punts importants a tenir en compte:

- **Sintaxis:** Java és el que potser té la millor sintaxis i permet escriure codi d'una forma més elegant. PHP fa servir sintaxis del tipus '::' i '->' i en alguns casos no ens permet blindar les funcions dels objectes per tal de no ser accedides.
- **Corba d'aprenentatge:** PHP és el més senzill d'aprendre, en part, gràcies a que és un llenguatge de programació *weak type*⁴ i no ens caldrà estar pendents dels tipus de variables que cal utilitzar en cada cas. ASP.NET també és força senzill, potser gràcies als IDE's que ens faciliten molt les tasques. J2EE, potser, és el més complicat d'aprendre, però no per la seva sintaxis, sinó per la multitud d'objectes que cal conèixer i saber quan utilitzar. També és important indicar que en tots 3 llenguatges de programació podrem trobar bons manuals on s'explica detalladament tots tipus de funcions. Per exemple, per a PHP podem consultar qualsevol funció de forma molt ràpida des de la pàgina oficial www.php.net i per a Java, tenim a disposició les APIs de Sun que ens explicaran de forma detallada el funcionament de tots els mètodes disponibles.

² Scripting: llenguatge de programació interpretat.

³ Strong type: s'utilitza per indicar els llenguatges de programació que no permeten assignar tipus de valors a les variables que no han estat inicialitzades amb el tipus del valor inicialment. Per exemple, una variable inicialitzada com a numèrica no pot rebre un valor considerat com a text.

⁴ Weak type: s'utilitza per definir el contrari de strong type, és a dir, els llenguatges de programació que permeten modificar el tipus de valor de les variables ja inicialitzades.

- **Velocitat de desenvolupament:** ASP.NET és clarament el més ràpid, perquè com hem dit abans, els IDE's utilitzats ens permeten crear multitud de components que ens faran tota la feina. PHP és molt ràpid si fem servir algun tipus de *framework*⁵ que ens faciliti i agilitzi la feina. I Java és el més lent, tot i que també podem utilitzar *frameworks* que ens aportaran millores substancials al nostre codi.
- **Plataforma:** PHP i Java es poden utilitzar en qualsevol tipus de plataforma, però ASP.NET només la podem utilitzar sota Windows.
- **Base de dades:** normalment sol utilitzar-se MySQL per PHP, Oracle per Java i MSSQL per ASP.NET. Encara que cap dels llenguatges està limitat a un tipus de base de dades en concret.
- **IDE:** ASP.NET es programa des de Visual Studio, una aplicació molt bona, però gens barata. Java té diferents eines de caràcter comercial, però el més utilitzat i possiblement un dels millors és la eina gratuïta Eclipse. Per PHP no existeix una aplicació que destaquï sobre les altres, encara que també es pot fer servir l'Eclipse mitjançant complements. Però si de totes maneres haguéssim de anomenar una, hauríem de destacar el Dreamweaver.
- **Suport orientat a objectes:** Java i ASP.NET són els millors, encara que el PHP ha millorat amb les últimes versions, especialment amb el PHP 5, que ja està pensat per a la programació orientada a objectes.
- **Seguretat:** Java sembla el més segur, però no és forçosament així, ja que encara que PHP tingui mala fama, és més degut als desenvolupadors que no pas al llenguatge. ASP.NET també té mala fama degut als errors de seguretat de Windows.
- **Rendiment:** PHP guanya en velocitat i en consum de recursos. Java i ASP.NET són força més pesats i necessitaran servidors més potents.
- **Servidor Web:** ASP.NET només funciona amb IIS. PHP i Java funcionen amb diferents servidors, tant comercials com gratuïts.

⁵ Frameworks: estructures de codi que ens permetran facilitar el desenvolupament de software, permetent als dissenyadors i programadors passar més temps identificant requeriments de software que intentant fer els detalls de més baix nivell per aconseguir un sistema funcional.

- **Llibreries i frameworks:** Tots tenen moltes llibreries i *frameworks* disponibles, però per PHP i Java la majoria són gratuïtes i *open source*⁶.
- **Suport i comunitat:** per ASP.NET la majoria dels fòrums, grups d'usuaris i comunitats de desenvolupadors estan controlats per Microsoft, mentre que per PHP i Java existeixen molts grups independents.
- **Cost:** PHP es l'alternativa totalment gratuïta, ASP.NET té llicències força cares, mentre que Java té la opció de desenvolupar-se tant amb eines gratuïtes com de pagament.
- **Cerca d'errors:** el php, al ser un llenguatge no compilat, pot portar a errors de sintaxis que facin que deixi de funcionar tota l'aplicació. El Java i el ASP.NET, al ser compilats, poder generar codi amb la sintaxis correcte gràcies als editors que van examinant el codi a mida que el generem. Però existeixen IDEs, com l'Eclipse, que són capaços d'analitzar la sintaxis dels llenguatges no compilats.
- **Desplegament de l'aplicació:** quan tenim una aplicació web funcionant a un servidor, per posar noves versions, cal tornar a desplegar l'aplicació si utilitzem Java o ASP.NET. Utilitzant PHP podem realitzar els canvis dels fitxers en calent, sense haver de reiniciar el servidor ni tornar a desplegar la nostra aplicació, encara que és una pràctica poc recomanable.
- **Inclusió de fitxers:** tant Java com ASP.NET ens permeten importar dins del codi només les classes i llibreries que necessitem. El PHP fa inclusions de fitxers, això fa que ràpidament es formi un caos en les inclusions de les pàgines força important, per això és molt recomanable fer la base de l'aplicació orientada a objectes.
- **Suport XML/XSL:** podem trobar llibreries de XML/XSL per a PHP a partir de la versió 5, però no per a la versió 4, que avui en dia, encara és de la més utilitzada. Java i ASP.NET tenen *parsers*⁷ millors de XML, com per exemple JDOM.
- **Proves:** amb PHP és molt senzill realitzar proves unitàries de tot el codi que anem generant, però tant en Java com amb ASP.NET, no tenim aquesta capacitat de prova unitària, sinó que ens serà més senzill provar blocs complets.

⁶ Open source: fa referència a la pràctica de desenvolupament de programari que promou l'accés lliure al codi font.

⁷ Parsers: algoritmes que permeten determinar l'estructura sintàctica d'una seqüència de caràcters en algun llenguatge.

- **Execució de comandes de sistema operatiu:** amb PHP podem executar i recuperar la sortida de qualsevol execució de forma trivial, en canvi, per a Java, haurem d'utilitzar llibreries especials que en alguns casos ens recuperaran la sortida de forma incorrecte.
- **MultiThreading:** amb Java es molt senzill crear múltiples *threads*⁸ d'execució per tal de realitzar tasques en paral·lel, però en PHP, encara que existeixen algunes llibreries, no és senzill realitzar-ho. Per això Java ens permetrà aguantar bones càrregues de treball.

Si pensem que la nostra aplicació requerirà una càrrega de treball no massa exigent, ens decidirem per el llenguatge de programació PHP, ja que ens permetrà realitzar un desenvolupament més ràpid i podrem fer proves unitàries de forma contínua per assegurar el bon funcionament de l'aplicació. Per intentar solucionar els possibles inconvenients que té el PHP respecte els altres llenguatges de programació, intentarem realitzar un bon disseny de la arquitectura inicial del projecte, per tal d'assentar unes bones bases.

1.5.5 PHP

Ja que utilitzarem PHP com a llenguatge de programació, és important conèixer-lo una mica més en profunditat.



PHP significa **PHP Hypertext Pre-processor**, i és un acrònim recursiu, però inicialment significava Personal Home Page . Però no només s'utilitza per a la creació de pàgines web dinàmiques, sinó que a més, el podem trobar, amb el nom de PHP CLI (*Command Line Interface*), utilitzat en programació per a consola tipus Perl o Shell Scripting, o fins i tot per a crear aplicacions gràfiques no web a partir de Qt/GTK+, que ens permet desenvolupar aplicacions d'escriptori per a certs sistemes operatius.

Anteriorment ja em comentat que el PHP és un llenguatge de programació senzill d'aprendre, a més, si tenim la sort de conèixer els llenguatges de programació C o Perl, ens serà més senzill, ja que té força similitud, fins i tot amb el nom de moltes funcions.

⁸ Thread: fil d'execució, permet a una aplicació realitzar varies tasques de forma simultània.

Com tots els llenguatges d'*scripting*, quan un explorador web envia una petició al servidor per demanar una pàgina, el servidor executa l'interpret de PHP per processar-la de forma dinàmica (obtenint la informació d'una base de dades o contingut dinàmic). Un cop finalitzat, s'envia al client, que ho entendreà com una pàgina web si hem generat codi HTML o com a qualsevol altre cosa depenent de si hem generat fitxers PDF, Flash o fins i tot imatges. Amb PHP podem generar multitud de formats de fitxers.

El desenvolupament del PHP és lliure, sota la llicència GNU, per això trobarem moltes llibreries que podrem utilitzar en la nostra aplicació i també trobarem diferents complements per fer-lo compatible amb la majoria de bases de dades, uns exemples completament compatibles són: MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite. Però encara que sigui lliure, no significa que no existeixin IDEs comercials, el més conegut es diu Zend Optimizer.

Actualment existeixen 3 versions de PHP. La més utilitzada i veterana és el PHP 4, que va sortir l'any 2000 i no va ser abandonada fins el 2007. Des de l'any 2004, podem trobar també PHP 5, que ja permet un ús del llenguatge orientat a objectes, un suport natiu millor per a MySQL, un suport millor per a *parsers* de XML, suport natiu per a la base de dades SQLite, suport integrat per a SOAP, variables iteradores de dades i, potser una de les millors actualitzacions seria, la possibilitat d'incorporar excepcions per tal de controlar els errors.

Per acabar de parlar sobre el PHP, pot ser interessant veure petites porcions de codi perquè així ens sigui més senzill reconèixer aquest llenguatge quan el veiem escrit. Com és habitual, mostrarem el codi que ens permet escriure la típica frase de "Hola món!" que tothom ha escrit amb les seves primeres pràctiques de programació:

```
<?php
echo "Hola món!";
?>
```

La funció **echo** ens mostrarà per pantalla o sortida estàndard *Hola món!* (sense les cometes).

També podem mesclar codi PHP amb codi HTML. A continuació es pot veure el codi d'una pàgina web ben formatada amb codi PHP que imprimirà *Hola món!* en el nostre navegador:

```
<html>
  <head>
    <title>Hola món HTML + PHP</title>
  </head>
  <body>
    <?php echo "<p>Hola món!</p>"; ?>
  </body>
</html>
```

Mostra de codi PHP amb funcions, variables, comentaris i operadors:

```
<?php
//Recull de variables del formulari
$usuari = $_POST["usuari"];
$pass = $_POST["password"];

//Validació d'usuari correcte
$id_usuari = get_id_usuari($usuari, $pass);
if ( $id_usuari != null ){
  //Comprovem si és administrador
  if ( es_administrador( $id_usuari ) {
    echo "Benvingut sigui, administrador!";
  } else {
    echo "Benvingut!";
  }
} else {
  echo "Vostè no és un usuari vàlid!";
}
?>
```

1.5.6 Bases de dades

Les bases de dades són sèries de dades organitzades i relacionades entre si. I si parlem d'un servidor de base de dades, estem parlant d'un conjunt de programes que permeten als usuaris accedir i modificar aquestes dades.

Podem trobar bases de dades de codi obert (per les que sovint no cal pagar llicències d'ús) i comercials. Quan parlem de Firebird, MySQL, PostgreSQL o SQLite, ho estem fent de bases de dades de codi obert. I quan parlem de Oracle, MSSQL, DB2 o Informix ho estem fent de bases de dades comercials.

La majoria d'aquests servidors, es poden utilitzar tant en plataformes Windows, com en sistemes basats en Unix. Per tant, no saber en quin sistema operatiu haurem de muntar el nostre servidor no serà un problema a no ser que utilitzem MSSQL, que només el podrem utilitzar sobre Microsoft Windows.

Si féssim una taula comparativa sobre les diferents bases de dades, veuríem que Oracle compleix satisfactòriament gairebé tots els punts comparatius. Però per desgracia, el seu alt cost econòmic, només el fa accessible a grans organitzacions. En canvi, per la seva gratuïtat, MySQL és la segona opció dels usuaris.

Entre Oracle i MySQL hi ha certes diferències que cal tenir en compte, però que sovint no ens faran decantar entre una o l'altra, com dèiem, el cost d'Oracle sol ser suficient per triar MySQL o altres servidors gratuïts.

Fins a la versió 4.1 de MySQL no es permetia realitzar subconsultes SQL. Per sort, això ja no és un inconvenient, però encara no es pot arribar a les 255 subconsultes niades que pot realitzar Oracle. També podem trobar la possibilitat de crear vistes a partir d'altres taules des de la versió 5 de MySQL. Tant Oracle com MySQL ens permetran utilitzar bases de dades de grans dimensions, però només d'Oracle es té la constància de poder administrar bases de dades de Terabytes de capacitat.

No sempre trobem que Oracle és superior. La base de dades MySQL és molt ràpida en l'accés a les taules per a realitzar lectures, això sempre i quan utilitzi el sistema de taules MyISAM, un accés no transaccional. Però aquest accés pot provocar problemes d'integritat en entorns d'alta concurrència en la modificació. Per sort, en les aplicacions web solem tenir força baixa concurrència en la modificació de dades. Per tant, MySQL és ideal per aquest tipus d'aplicacions.

Altres avantatges de MySQL, però no tant importants, són el cas dels auto-incrementals o el nombre màxim de columnes permeses a una taula: MySQL no precisa de l'existència de seqüenciadors que ens permetin augmentar un índex d'una taula, podem indicar una columna com a auto-incremental i anirà incrementant tot sol; el nombre màxim de columnes d'una taula és de 3398 per a MySQL i 1000 per a Oracle, però probablement mai farem servir tantes columnes.

Però l'element més diferenciador és el suport tècnic que rebrem de cada un, hi ha constància que el suport d'Oracle és excel·lent i adaptat a una gran varietat de llicències, per altra banda, per a MySQL podrem trobar multitud de llistes de correus i pàgines webs amb informació, però si volem atenció telefònica haurem de pagar-ho com a extra.

Com ja evidenciàvem, al nostre projecte utilitzarem MySQL com a servidor de bases de dades degut a la seva alta qualitat i baix cost econòmic (tenint en compte que, en cas que ho necessitéssim, haurem de pagar per un servei tècnic personalitzat).

1.5.7 MySQL

MySQL AB és l'empresa encarregada de desenvolupar aquest gestor de bases de dades i pertany a Sun Microsystems des de gener de 2008. Tot i tenir llicència GNU GPL (software lliure), tenim la possibilitat d'adquirir llicències que ens permetin incorporar-lo en productes no lliures.



Segons la pròpia empresa que desenvolupa l'aplicació, hi ha més de 6 milions de còpies de MySQL utilitzant-se actualment. Això superaria a qualsevol altra servidor de bases de dades.

L'aplicació s'encarrega de permetre gestionar i administrar bases de dades relacionals i multiusuari, i compleix tots els estàndards SQL sense sacrificar velocitat, fiabilitat o usabilitat.

Actualment existeix la versió estable 5.1 des de l'any 2008, però la anterior versió estable 4.1 encara està present a moltes aplicacions web, encara que ja no es dona suport oficial des del desembre de 2009. La darrera versió incorpora importants novetats com: els procediments emmagatzemats (*stored procedure*), que són petits programes desats dins la base de dades que s'executen a petició de l'usuari; els *triggers*, que són procediments que s'executen al complir-se certes condicions al realitzar insercions, modificacions o eliminacions de registres; els cursors, que ens permetran recórrer les consultes SQL dins un procediment emmagatzemat; les vistes, que són un tipus de taules virtuals que ens permeten alterar el seu contingut a través de consultes SQL de forma virtual, ja sigui restringint o ampliant l'accés a les taules originals; el suport a connexions segures tipus SSL; el suport de columnes de tipus *varchar*; la possibilitat de realitzar *caches*⁹ de consultes utilitzades sovint; la possibilitat de realitzar subconsultes; i la indexació i cerques de camps de text a través de MyISAM.

Hi ha molts llenguatges de programació que tenen APIs¹⁰ que permeten accedir a bases de dades de MySQL, per llistar-ne uns quants: C, C++, C#, Pascal, Delphi, Java, Lisp, Perl, PHP, Ruby, FreeBASIC, i Tcl. Però també existeix una interfície ODBC que permetrà a qualsevol llenguatge de programació compatible amb ODBC accedir a MySQL.

Un dels seus punts forts és la seva flexibilitat a la hora de triar entre diferents motors d'emmagatzematge per cada una de les taules que creem, de fet és l'únic servidor de bases de dades que ho permet. A part del motor MyISAM, que és el més ràpid per fer accessos de lectura, trobem: Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example.

⁹ Caches: Conjunts de dades duplicades per tal de no accedir les dades originals. S'utilitzen en els casos que l'accés a les dades originals és força costós en temps de lectura, comparant amb l'accés a les dades duplicades.

¹⁰ APIs: Interfície de Programació d'Aplicacions: conjunt de mètodes que permeten oferir accés a certs serveis d'ús general.

En aplicacions web trobem que MySQL està molt lligat al PHP i existeixen molts paquets d'aplicacions que ens presenten Apache, PHP i MySQL com un conjunt integrat. EasyPHP seria un dels més coneguts i sense fer res més que una petita instal·lació tindrem les tres aplicacions perfectament unides i preparades per ser explotades.

2 Funcionalitats i requeriments del sistema

Abans d'iniciar el projecte hem dut a terme un estudi per tal d'esbrinar les necessitats del nostre client. Aquest ha estat dividit en diferents actes no presencials on quedaven reflectits tots els funcionaments actuals del client i com orientar el projecte per adaptar-lo a les seves necessitats. Aquestes actes s'han anat desant per tal de crear un petit històric i evitar contradiccions futures amb el client.

A partir de les actes es van poder extreure certs requeriments funcionals, que seran realment les necessitats del client, i requeriments no funcionals, que seran el còm es duran a terme per tal de complir-les

2.1 Requeriments funcionals

Els requeriments funcionals són totes aquelles necessitats del nostre projecte, que si no hi foren, no permetrien utilitzar l'aplicació, ja que indiquen com s'ha de comportar.

A partir de les actes es poden detectar 3 classes de requeriments funcionals: els referents al mòdul de projectes, els del mòdul de contactes i els del petit mòdul d'administració d'usuaris.

2.1.1 Els projectes

A partir de les actes definim els següents requeriments funcionals dins del mòdul de projectes:

L'aplicació ha de permetre:

- realitzar cerques senzilles de projectes.
- crear projectes, el requisit mínim és informar el nom, el contacte i assignar-li una referència. Aquesta referència haurà de tenir el format: YY/XXX – On YY farà menció a l'any del projecte, 08 significaria 2008, i XXX es referirà al número de projecte dins de l'any. L'usuari pot assignar lliurement aquesta referència si no està assignada a cap altre projecte.
- visualitzar tota la informació de qualsevol projecte i poder realitzar un seguiment d'aquest.

- modificar l'estat i les dades dels projectes.
- associar contactes als projectes.

2.1.2 Els contactes

A partir de les actes definim els següents requeriments funcionals dins del mòdul de contactes:

L'aplicació ha de permetre:

- realitzar cerques senzilles de contactes i opcionalment, cerques més complexes.
- crear contactes, el requisit mínim és informar el nom i el tipus d'aquests.
- visualitzar totes les dades dels contactes incloent un llistat de contactes associats.
- la modificació de totes les dades corresponents al contacte.
- relacionar els contactes entre si.
- crear tipus de contactes i permetre realitzar-ho si no trobem el tipus desitjat durant la creació d'un contacte sense perdre el formulari de creació.
- modificar el nom dels tipus de contactes que no estiguin protegits contra escriptura.
- esborrar els tipus de contactes que no estiguin protegits contra escriptura. En cas que el tipus de contacte estigui assignat a un contacte, aquests seran esborrats en cascada. Cal deixar molt clar aquesta conseqüència a l'usuari de l'aplicació.

2.1.3 Els usuaris

A partir de les actes definim els següents requeriments funcionals dins el mòdul d'usuaris:

L'aplicació ha de permetre:

- realitzar altes d'usuaris nous.
- cercar usuaris.
- modificar les dades dels usuaris existents.
- modificar els permisos d'usuaris existents.
- visualitzar informes d'activitats.

2.2 Requeriments no funcionals

Els requeriments no funcionals ens especificaran com realitzar les funcionalitats de la nostra aplicació de la manera més òptima o millor adaptada a l'usuari.

Podem dir que l'estabilitat del projecte és un requeriment no funcional, ja que haurem de crear les funcionalitats de manera que no sigui fàcil crear situacions que deixin l'aplicació inestable, és a dir, haurà de ser tolerant a errors. D'aquesta manera, tindrem una aplicació robusta que independentment de l'ús que l'hi doni el client, no permetrà realitzar res pel que no estigui pensada l'aplicació.

També veiem que l'eficiència és un altre requeriment no funcional, ja que haurem d'adaptar les implementacions per tal de no realitzar els requeriments funcionals amb uns temps excessius.

A part de l'estabilitat i l'eficiència, detectem, de forma global, dos requeriments no funcionals més:

- com haurem de disposar la interfície gràfica per tal de facilitar-nos la implementació del projecte, fer-lo extensible i sobretot per tal de facilitar i fer intuïtiva l'aplicació al nostre client.
- elements qualitius (on podríem englobar l'estabilitat i l'eficiència entre d'altres) que ens permetran realitzar, a poc que creixi el projecte, noves millores i solucions d'incidències d'una manera més ràpida i còmode.

2.2.1 Interfície gràfica

La interfície gràfica de l'aplicació haurà de ser senzilla i intuïtiva per tal que l'usuari no trobi cap entrebanc a l'hora de treballar, ja que va dirigida a persones que no tenen per què estar familiaritzades en el món de l'informàtica en general.

També serà important realitzar una interfície gràfica atractiva a la vista i sobretot agradable en el tractament d'errors.

Es disposarà una capçalera simple amb el logotip de l'empresa, una salutació, tres accions bàsiques i informació sobre si té notificacions pendents o no.



Fig. 1: Capçalera de l'aplicació

Les tres accions bàsiques són:

- Anar a l'inici: ens portarà fins la pàgina per defecte a l'entrar a l'aplicació.
- Sortir: ens desconnectarà la sessió d'usuari i ens portarà fins la pantalla d'entrada d'usuari.
- Modificar contrasenya: ens permetrà accedir a la pàgina de modificació de la contrasenya de l'usuari en sessió.

Cada un dels mòduls de l'aplicació estaran disponibles a la part esquerra de la pantalla. Així serà molt senzill canviar de mòdul segons convingui a l'usuari:

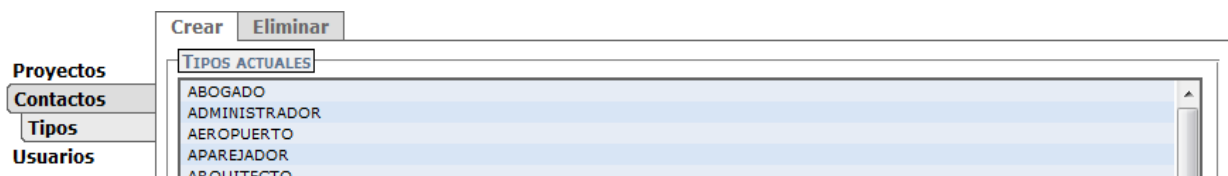


Fig. 2: Menú de l'aplicació

A la dreta del llistat de mòduls ens apareixerà un espai rectangular que ens permetrà dur a terme tots els requeriments funcionals. Cada una de les funcionalitats d'un mòdul estaran disponibles en un sistema de pestanyes a la part superior de la zona rectangular. Així, per exemple, en la imatge anterior es pot veure que estem situats al requeriment funcional de "Crear un nou tipus de contacte", on s'aprofita per "Consultar una llista dels tipus de contacte", i que també tenim accessible la funcionalitat "Esborrar tipus de contactes".

En qualsevol moment, l'usuari pot decidir canviar de mòdul i anar al de Projectes o al d'Usuaris, només haurà d'anar al menú de l'esquerra. Això actualitzarà la zona de la dreta per mostrar les funcionalitats del mòdul seleccionat.

Una de les parts més invasives per la navegació web dels usuaris són els missatges a través de pop-up's per JavaScript, altrament coneguts com *alerts*.

S'haurà d'intentar evitar l'ús d'aquests missatges a no ser que necessitem la intervenció directa de l'usuari per tal de prendre decisions, com per exemple, confirmar si desitja esborrar algun registre. Els errors que no necessitin d'aquesta confirmació, els mostrarem a l'espai en blanc entre la capçalera i el cos de la pàgina, i només apareixeran quan esdevinguin els errors. Per tal de fer-ho el més visible possible, mostrarem els missatges en una tonalitat àmbar. A continuació es mostra un missatge d'error al intentar donar d'alta un Tipus de Contacte sense introduir cap nom:



Fig. 3: Missatge d'error mostrat per l'aplicació

Sempre que calgui la intervenció de l'usuari, farem ús dels missatges JavaScript per fer decidir o corroborar a l'usuari la decisió a prendre. Així serà en el cas que un usuari vulgui esborrar un Tipus de Contacte i aquest estigui associat a algun Contacte:

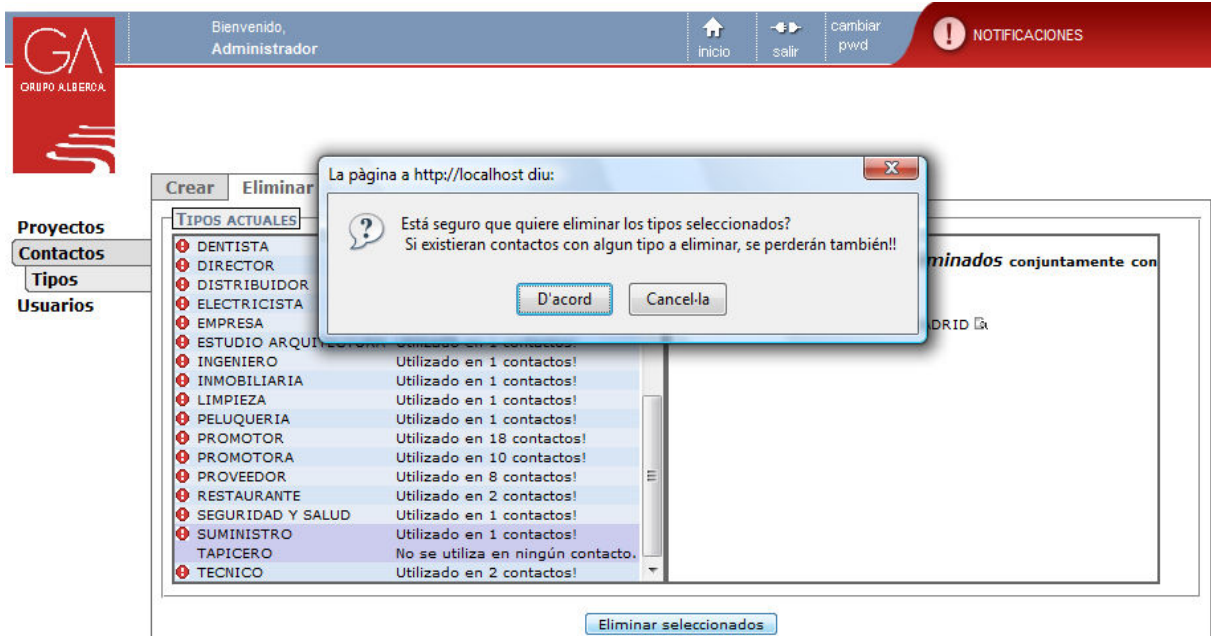


Fig. 4: Finestra emergent de l'aplicació

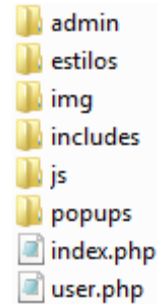
2.2.2 Factors de qualitat

Quan dissenyem una aplicació hem de tenir en compte un seguit d'elements que ens permetran donar estabilitat i eficiència, però sobretot ens permetran ampliar el projecte d'una manera senzilla i ràpida, i també podrem detectar i focalitzar els errors més còmodament.

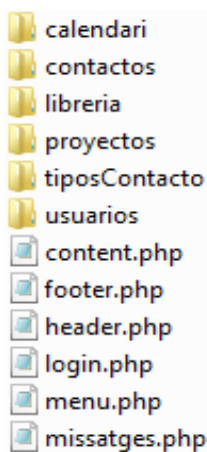
Haurem de realitzar una estructura inicial de fitxers que ens permeti separar els diferents mòduls de l'aplicació en diferents carpetes. Això ens permetrà localitzar més clarament on estan ubicats els fitxers que facin referència a una certa pantalla de l'aplicació.

Primer de tot separarem tots els fitxers referents a l'administració i seguretat de l'aplicació en una carpeta anomenada *admin*, així si volguéssim podríem treure els permisos d'escriptura d'aquests fitxers per tal que ningú els pugui modificar i així augmentar la seguretat.

Per altra banda separarem la resta de fitxers segons el seu objectiu dins l'aplicació: els estils que determinaran l'aspecte visual de l'aplicació aniran dins la carpeta *estilos*, totes les imatges que hem utilitzat les desarem a la carpeta *img*, els fitxers amb les funcions JavaScript necessàries pel correcte funcionament de l'aplicació les trobem dins la carpeta *js* i llavors tenim 2 carpetes, *popups* i *includes*, que contindran els fitxers de les capes de presentació i negoci de les pantalles de l'aplicació, referents a pop-up's i a la resta respectivament.



A l'arrel tenim dos fitxers, l'*index.php* encarregat de controlar si l'usuari té permisos per accedir a l'aplicació i si en té, realitzar les crides als diferents fitxers que permetran mostrar totes les funcionalitats de l'aplicació i l'*user.php* que serà l'encarregat de realitzar els logins, els logouts i els registres dels usuaris.



Dins la carpeta *includes* hem creat una estructura de fitxers que permeti separar els diferents blocs de l'aplicació. Hem creat una subcarpeta per cada un dels mòduls, com per exemple el de contactes i el de projectes. També desarem aquí els fitxers que constituïran l'estructura base de qualsevol pàgina de l'aplicació: els fitxers *footer.php* i *header.php* s'encarreguen de mostrar la part superior i inferior de totes les pàgines, ja que totes mostraran el mateix; el fitxer *content.php* s'encarregarà d'incloure els fitxers de negoci que demani l'usuari, però comprovant prèviament si aquest fitxer existeix, per així poder ser més tolerant als errors i mostrar un missatge d'error adequat si l'usuari demana una opció no implementada; el fitxer *login.php* es mostrarà en cas que l'usuari no estigui autènticat; i tant el fitxer *menu.php* com *missatges.php* formen part de l'estructura bàsica que apareixerà a totes les pàgines.

La carpeta *libreria* contindrà tots els fitxers amb els mètodes necessaris per realitzar tot el negoci de l'aplicació. Els mètodes estaran subdividits en fitxers, que amb el seu nom, ens permetran identificar fàcilment el seu contingut.

Amb aquesta estructura de fitxers ens serà més senzill localitzar possibles errors futurs a l'aplicació, un cop posada en producció, però sobretot aconseguirem que sigui més senzill realitzar millores, ja que tindrem una aplicació modulable i separable en blocs.

3 Planificació i estudi econòmic

Per tal de poder realitzar una estimació inicial del temps i cost del projecte, hem recollit tota la informació que ens ha estat possible facilitada per l'estudi d'arquitectura. Després de realitzar les actes de les reunions tingudes amb el client, hem arribat a la conclusió que es tracta d'un projecte fàcilment modulable, ja que consta de tres mòduls diferenciats, semi independents entre ells, els quals tenen una relació de col·laboració que interfereix poc en l'estructura bàsica dels altres mòduls.

Que el projecte fóra modulable en tres parts, ens ha permès planificar el projecte en dues etapes, una en la que realitzaríem els dos mòduls més grans: el de projectes i el de contactes; i un altre on realitzaríem el mòdul d'usuaris i ampliaríem el mòdul de projectes amb els seguiments d'expedients.

Dividint la planificació i la implementació en dues etapes aconseguirem que el client participi d'una forma més activa en la realització del projecte, el que ens permetrà solucionar errors de programació i de planificació en etapes més joves del desenvolupament de l'aplicació. Però sobretot, podrem detectar errors de comprensió en el recull de les dades facilitades pel client en etapes més senzilles de modificar.

3.1 Planificació temporal

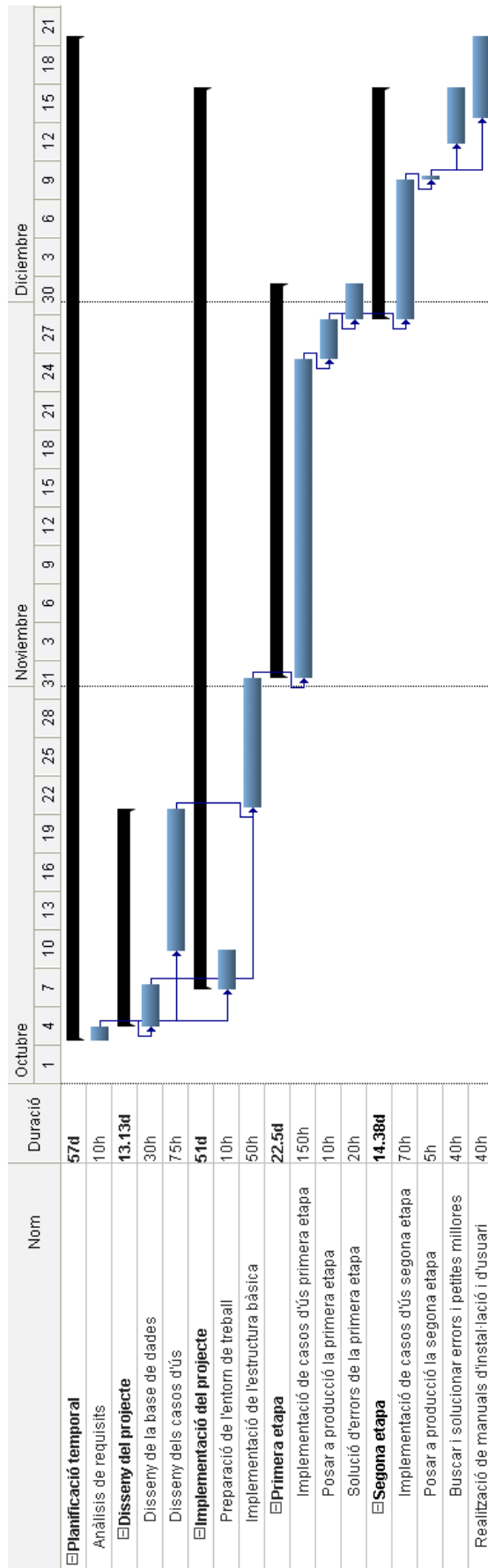


Fig. 5: Planificació temporal inicial

A la planificació s'ha tingut en compte que només disposem d'un recurs per a realitzar totes les tasques, així que s'han evitat solapar alguns dels treballs encara que podrien realitzar-se de forma concurrent, com per exemple la solució d'errors i la implementació de la segona fase, o els dissenys dels casos d'ús i la base de dades.

Segons el diagrama de Gantt de l'esquerra, podem deduir que necessitarem d'un total de 510 hores per poder realitzar el projecte. La part del disseny hauria de constar de 105 hores i la de la implementació de 355, de les quals, 180 hores es destinaran a la primera etapa del projecte, i 115 hores a la segona.

També ens mostra la planificació del projecte de l'equip de treball a excepció del cap de projecte, que haurà de realitzar diferents tasques (vistes en el punt 3.2.1) i que seran realitzades al llarg del projecte. Creiem convenient destinar unes 60 hores a aquesta tasca, que estaran concentrades pràcticament a l'inici del projecte i durant les entregues de les versions, a més del petit seguiment diari que s'hagi de realitzar.

Si sumem el total d'hores que necessitarem com a cap de projecte al total d'hores que ja teníem (510), veiem que caldrà un total de 570 hores per poder finalitzar el projecte.

3.2 Estudi econòmic

El projecte ha estat realitzat per una sola persona, però per tal de poder fer un estudi econòmic més acurat a la realitat, és necessari que dividim cada una de les tasques a realitzar entre els típics rols d'un projecte d'aquestes característiques, que assignem uns costos d'instal·lacions, d'equipaments i altres despeses d'una empresa petita.

3.2.1 Costos dels recursos humans

Com avançàvem, cada una de les parts del projecte hauria de ser realitzada per un rol de treballador diferent, ja que és necessari un perfil adequat a la tasca a realitzar:

- **Cap de projecte:** s'encarregarà d'organitzar i definir l'equip de treball, de mantenir la relació amb el client, de calcular el cost del projecte, i assegurar-se que el projecte avança de forma correcta a partir d'informes i seguiments.
- **Analista funcional:** important per la supervisió de les proves, la documentació del sistema i la realització dels dissenys de l'aplicació.
- **Arquitecte:** vetllarà per les bones pràctiques de programació, dissenyarà l'estructura primordial, liderarà l'equip de programadors i s'assegurarà que tots utilitzin la solució estructural ideada, també s'encarregarà de posar les versions a producció.
- **Analista-programador:** transformarà els dissenys funcionals en codi programació des d'una visió més global.
- **Programador:** amb l'ajuda de l'analista-programador crearà tot el codi programació per tal de complir amb tots els casos d'ús.

Per obtenir un cost força aproximat dels recursos humans utilitzats necessitem saber un cost per hora de treball per cada un dels rols anteriors.

Per obtenir cada un dels preus per hora, sumarem als sous bruts anuals el cost de la Seguretat Social que ha de costejar qualsevol empresa, normalment sobre el 30% adicional. Aquest valor el dividirem per 1800, que és un valor típic d'hores de treball anuals.

Si tenim en compte uns sous mitjans per a cada un dels rols descrits, obtindrem la següent taula de costos:

Càrrec (Sou brut anual)	Preu / hora	Hores	Cost
Cap de projecte (45.000 € bruts)	32,5 €/h	60	1.950 €
Analista funcional (35.000 € bruts)	25,0 €/h	155	3.875 €
Arquitecte (40.000 € bruts)	29,0 €/h	75	2.175 €
Analista-programador (26.000 € bruts)	18,5 €/h	90	1.665 €
Programador (20.000 € bruts)	14,5 €/h	190	2.755 €
TOTAL		570	12.420 €

3.2.2 Costos del material software

Donat que totes les eines de treball utilitzades seran de codi obert, no haurem de realitzar cap mena d'esforç econòmic. MySQL, tot i ser de codi obert, té una llicència GPL (General Public License) que ens permetrà utilitzar-la de forma gratuïta sempre que no se'n faci un ús comercial. Com que el projecte va destinat a un ús intern de l'empresa, no ens caldrà pagar la llicència ja que no se'n farà un ús comercial.

A més, encara que el servidor web no hagués estat de codi obert, com que el nostre client ja tenia un servidor preparat i configurat, no hauríem d'haver hagut de realitzar cap compra extra d'aplicacions. Ni tant sols en el nostre entorn de desenvolupament, ja que haguéssim utilitzat eines amb cost zero compatibles amb les suposades eines de pagament del client.

3.2.3 Cost total del projecte

Com que no tenim cap mena de cost per part del software utilitzat, només hem de tenir en compte els costos dels recursos humans, que són 12.420 €. Però si volem ser estrictes, hauríem de quantificar els costos de lloguer de local i altres despeses associades:

- Un local d'uns 110 metres quadrats té un cost mensual de 1.200 €.
- L'assegurança contractada per cobrir qualsevol robatori o desperfecte causat al local té un cost anual de 360 €, que si els repartim en 12 mesos, té un cost de 30 € mensuals.
- L'equip informàtic necessari per a un treballador té un cost mig de 1.200 €. Per a una empresa de 12 treballadors necessitarem un total de 14.400 €. Si volem amortitzar-ho en 4 anys, haurem d'amortitzar mensualment 300 €.
- Els equips informàtics funcionant durant més de 8 hores diàries i la il·luminació, té un cost de 150 € mensuals.
- El contracte amb una empresa subministradora de garrafes d'aigua té un cost de 25 € mensuals.
- El cost de les trucades telefòniques i la línia ADSL té un cost de 120 € mensuals.
- El servei de neteja setmanal té un cost de 80 € mensuals.

Per tal de repercutir aquests costos al projecte, hem de trobar el cost per hora de cada una de les despeses, és a dir, dividir-ho entre 1800 hores anuals de treball, i trobar la part proporcional dins el projecte, és a dir, multiplicar-ho pel número d'hores que durarà, 570.

Concepte	Cost mensual	Cost anual	Cost repercutit al projecte
Lloguer del local	1.200 €	14.400 €	4.560 €
Assegurança local	30 €	360 €	114 €
Amortització dels equips informàtics	300 €	3.600 €	1.140 €
Electricitat	150 €	1.800 €	570 €
Aigua	25 €	300 €	95 €
Telèfon + Internet	120 €	1.440 €	456 €
Neteja	80 €	960 €	304 €
TOTAL	1.905 €	22.860 €	7.239 €

El cost total de les despeses associades al local que hem repercutit al projecte és de 7.239 €. Però hem de sumar tots els costos per obtenir el cost econòmic total:

Concepte	Cost
Cost dels recursos humans	12.420 €
Cost del material software	0 €
Cost d'altres conceptes	7.239 €
TOTAL	19.659 €

Si estem valorant el projecte econòmicament, hem de dir que té un cost total de 19.659 €. Però hauríem de pretendre aconseguir un cert benefici per tal de realitzar un negoci rentable.

Segons l'experiència obtinguda dins del mercat de les consultories, un projecte de mida més aviat petita pot pagar-se entre 30.000 € i 50.000 €, preus que resulten ser molt poc competitius dins del mercat ofert per les empreses petites que busquen clients petits, ja que aquests preus només estan disposats a costejar-los les empreses que busquen garanties serioses en els seus projectes.

Basant-nos en experiències viscudes, en empreses petites, qualsevol projecte inferior a un mes, té un cost mínim de 6.000 €, però els projectes que duren entre dos i tres mesos, solen pagar-se entre els 10.000 i els 20.000 €.

Si incrementem el preu al voltant d'un 20% per tal d'obtenir un marge de benefici, obtenim un cost total aproximat de 23.600 €. Aconseguirem cobrir possibles entrebancs que puguin sorgir i tindrem un projecte econòmicament rentable. A més, tindríem cert marge per negociar el preu final si el client ens pressiona.

El preu final del projecte seria força competitiu enfront empreses mitjanes i grans, però hauríem d'esforçar-nos en mostrar als nostres clients que som una empresa que ofereix les mateixes garanties i qualitats que empreses mitjanes i grans.

3.3 Riscs del projecte

Tot projecte comporta uns riscos. En el nostre cas, el risc més gran seria no haver estimat de forma correcta el cost del projecte, el que resultaria amb una possible pèrdua econòmica. Però com veurem més endavant, no és un risc tant gran en funció del client amb el que estiguem tractant.

També es poden córrer riscos a nivell dels propis recursos humans, sobretot en empreses petites. Hem de preveure que qualsevol dels perfils utilitzats en el projecte, poden causar baixa per qualsevol motiu. Hauríem de ser capaços de preveure aquestes situacions i trobar substitutius amb certa brevetat.

Un altre risc, seria que el client volgués abandonar el projecte abans de finalitzar-se. A partir de contractes formals, minimitzarem aquest risc, podem incloure clàusules que obliguin a indemnitzacions per les dues parts o podem obligar al prepagament d'un percentatge del projecte.

En el cas que realitzéssim un projecte per a un client important i amb bones perspectives de futur, potser ens interessaria ajustar el cost fins el punt d'assumir certes pèrdues per tal d'aconseguir treballar amb el client sí o sí. Això ho podríem fer sempre que tinguem una alta

probabilitat que aquest mateix client ens contracti nous projectes o ampliacions del mateix, o fins i tot que ens proporcioni reputació dins del sector o altres clients relacionats. Diguem que estem invertint en el potencial del client per aconseguir un rendiment a més llarg termini.

4 Especificació

L'especificació ens permet formalitzar els requeriments funcionals definits anteriorment.

A continuació hem utilitzat UML per especificar i dissenyar l'aplicació, però abans hem realitzat una introducció teòrica al UML.

4.1 Què és UML?

UML significa Llenguatge Unificat de Modelatge (Unified Modeling Language) i es tracta d'un llenguatge d'especificació visual estandarditzada pel modelatge d'objectes i mètodes. S'utilitza per definir una aplicació, per detallar el funcionament del sistema, per documentar i construir.

En UML s'utilitzen diferents tipus de diagrames que ens permetran especificar i dissenyar qualsevol part de l'aplicació:

- **Diagrames d'Estructura:** indiquen quins elements han de formar part de l'arquitectura de l'aplicació:
 - **Diagrames de classes:** descriuen l'estructura d'un sistema indicant les seves classes¹¹, atributs i les relacions entre elles.
 - **Diagrames de components:** descriuen l'estructura dels components físics i indica les seves dependències relacionals.
 - **Diagrames d'objectes:** són un tipus de diagrames de classes que es centren en parts específiques de les classes.
 - **Diagrames d'estructura composta:** descriuen l'estructura interna de les classes, indicant-nos quins objectes estan compostos per altres objectes.
 - **Diagrames de desplegament:** mostren el hardware utilitzat i les seves relacions.

¹¹ Classe: representa un conjunt d'objectes amb una estructura, un comportament i unes relacions semblants. Aquest conjunt d'objectes comparteix els mateixos atributs, operacions, mètodes i relacions. En UML una classe és una implementació d'un tipus amb dos components: atributs, que representen les propietats, i operacions, que donen un servei i provoquen un comportament a la classe.

- **Diagrames de paquets:** ens indiquen com es divideix una aplicació en agrupacions lògiques i les seves dependències.
- **Diagrames de Comportament:** indiquen el funcionament de l'aplicació:
 - **Diagrames d'activitats:** descriuen el flux de treball i de control pas per pas.
 - **Diagrames de casos d'ús:** ens permet mostrar de forma gràfica les funcionalitats de l'aplicació des del punt de vista de l'actor¹², els seus objectius i la relació entre els casos d'ús.
 - **Diagrames d'estats:** descriuen els diferents estats que podem tenir i com podem passar d'uns als altres.
- **Diagrames d'Interacció:** subtipus de diagrames de comportament, però que ens indiquen el flux de control i de dades entre els elements de l'aplicació.
 - **Diagrames de seqüència:** ens mostra per cada cas d'ús com interactuen un conjunt d'objectes de l'aplicació a través del temps.
 - **Diagrames de col·laboració:** descriuen les relacions entre els diferents rols que tingui l'aplicació.
 - **Diagrames de temps:** descriuen amb gràfiques en forma d'ona la relació temporal per diferents senyals i com afecten unes a les altres.
 - **Diagrames de vista d'interacció:** descriuen la cooperació entre altres diagrames d'interacció per mostrar els fluxos de control de l'aplicació.

¹² Actor: rol que pot exercir un usuari dins l'aplicació. No fa referència a un usuari en particular, si no a un conjunt d'ells o a una tasca que poden realitzar.

4.2 Model Conceptual

El model conceptual representa els objectes i els conceptes més importants dins del l'entorn del sistema. A través del model conceptual es pot veure quins elements intervenen, les propietats que han de tenir i les seves relacions.

4.2.1 Diagrama de classes

Tots els elements estàtics, com les classes i les seves relacions, queden reflectits en el diagrama de classes.

Totes les classes poden tenir: atributs, que són els elements identificats per un nom i que corresponen a una propietat; i mètodes, que són les operacions que poden ser utilitzades per altres classes i que poden provocar o no canvis a les propietats internes de la classe.

Les relacions entre classes poden ser de diferents tipus, així podem tenir les relacions per associació, composició o especialització.

- **Associació:** ens indica la relació que hi ha entre dos o més classes a partir del comportament entre elles. Sovint se sol indicar la multiplicitat de la relació, que no és més que el nombre d'elements d'una mateixa classe que es poden relacionar amb una altra classe.
- **Composició:** ens indica la relació entre dos classes, les quals una d'elles forma part com atribut d'una altra, però que, independentment, pot funcionar ella mateixa com una classe en altres situacions.
- **Especialització:** ens indica la relació entre dos classes, les quals una d'elles hereta tots els atributs i mètodes de l'altra. La classe "pare" se la denomina SuperClasse i la classe "filla", la que hereta, se l'anomena SubClasse.

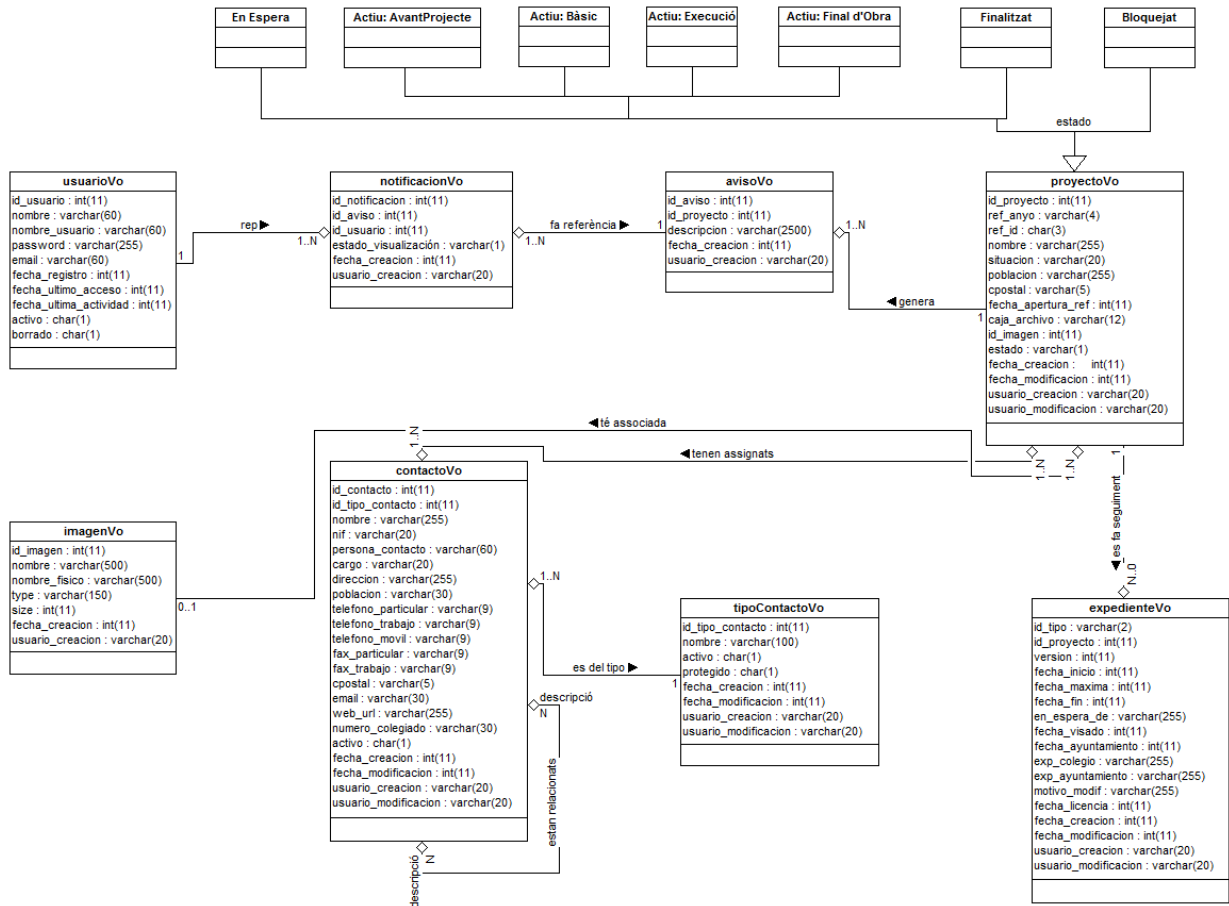


Fig. 6: Diagrama de classes de l'aplicació

4.3 Model de casos d'us

Els models de casos d'us mostren la interacció entre l'aplicació i tots els elements externs, anomenats actors, que poden alterar-ne l'estat. A més, ens permeten emmarcar i delimitar totes les funcionalitats de l'aplicació.

Per fer-ho necessitem saber quins són els actors i de quina manera interactuen amb l'aplicació.

4.3.1 Diagrama de casos d'ús

A través dels diagrames de casos d'ús podem definir la interacció entre els actors de l'aplicació i els seus casos d'ús.

En el cas de la nostra aplicació tenim 2 tipus d'actors: els usuaris autenticats i els usuaris autenticats amb rol d'administrador. Com veurem en els següents diagrames, els administradors són els únics usuaris autenticats que poden accedir als casos d'ús de creació i edició d'usuaris.

4.3.1.1 Mòdul de Projectes

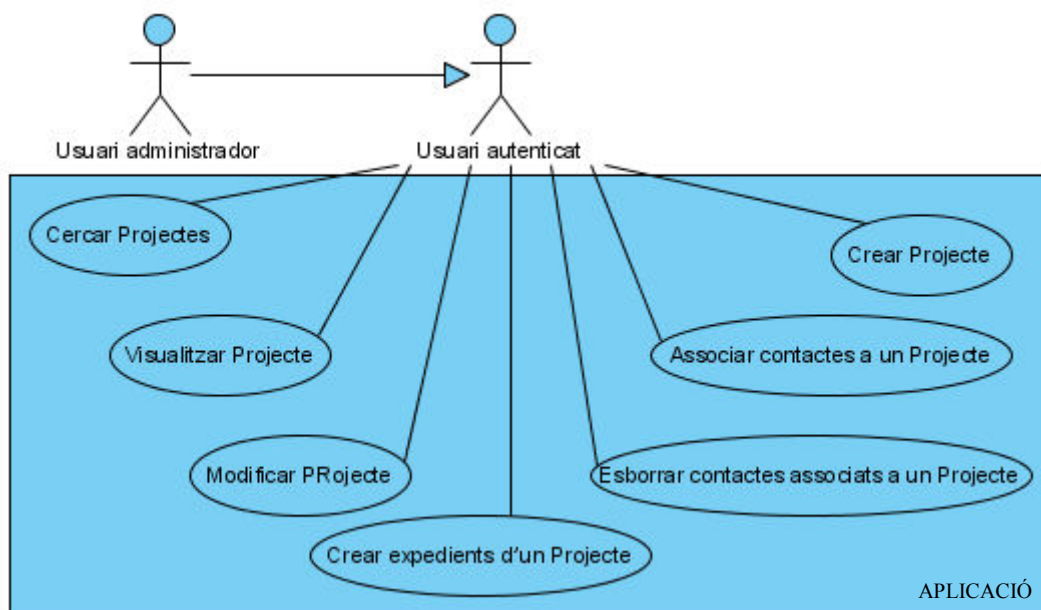


Fig. 7: Diagrama de casos d'ús del mòdul de projectes

4.3.1.2 Mòdul de Contactes

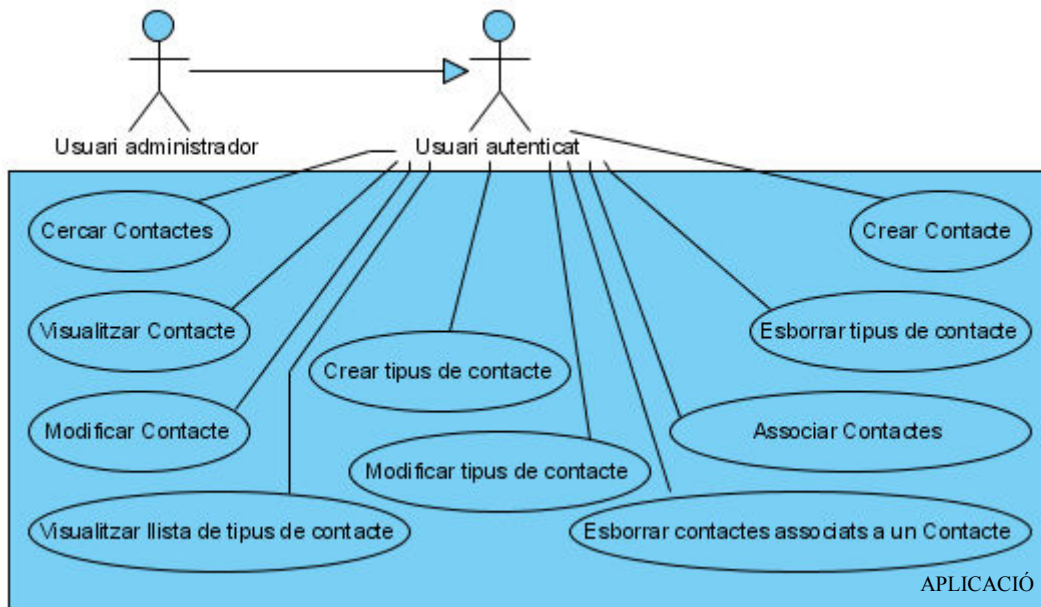


Fig. 8: Diagrama de casos d'ús del mòdul de contactes

4.3.1.3 Mòdul d'Usuaris

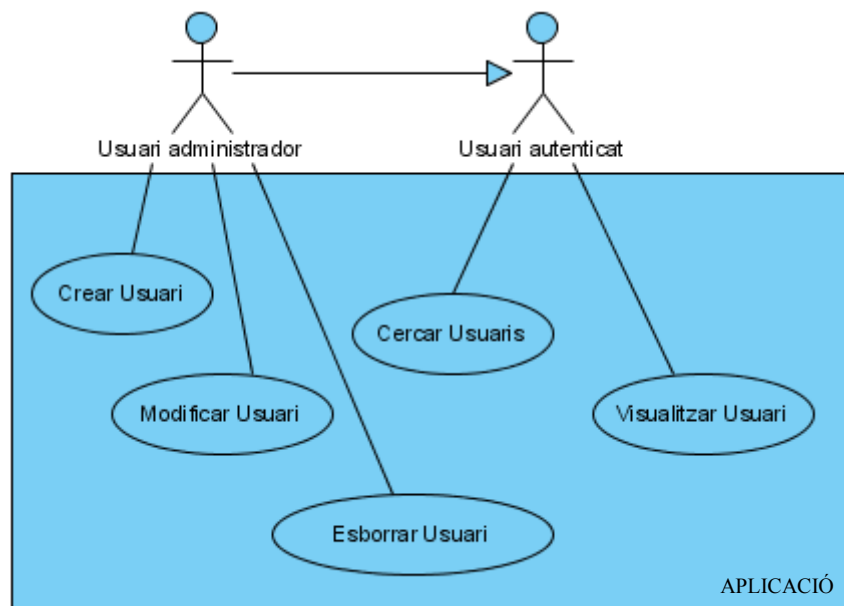


Fig. 9: Diagrama de casos d'ús del mòdul d'usuaris

4.3.2 Especificació de casos d'ús

L'especificació de casos d'ús ens permet explicar pas per pas quines són les accions entre els actors i l'aplicació. Les accions no tenen per què ser lineals, en molts casos, l'usuari podrà prendre diferents camins en funció de les seves necessitats i l'aplicació podrà decidir en tot moment si trencar l'ordre d'execució normal, generalment per obligar a l'usuari a rectificar les seves accions.

A continuació veurem de forma detallada tots els casos d'ús explicats pas per pas amb els corresponents passos alternatius.

4.3.2.1 Mòdul de Projectes

- **Cercar Projectes:**

Actors: Usuaris autènticats.

Descripció: Cerca de projectes mitjançant el formulari de la pantalla "Consultar".

Accions dels actors	Accions de l'aplicació
1. L'usuari vol realitzar una cerca de projectes i fa clic a la opció de menú "Proyectos".	
	2. L'aplicació mostra la pantalla "Consultar".
3. Introdueix en el formulari d'entrada les dades a cercar.	
4. Fa clic al botó "Buscar".	
	5. L'aplicació realitza la cerca dels projectes que compleixen tots els requisits que ha introduït l'usuari a través del formulari.
	6. Mostra per pantalla la llista de projectes trobats ordenats per referència. Depenent del camp cercat, l'aplicació mostrarà en vermell els camps que coincideixin amb els introduïts per l'usuari.
Pas alternatiu:	

7. L'usuari desitja esborrar tots els camps de cerca del formulari i fa clic al botó "Limpiar".	
	8. L'aplicació esborra tots els camps del formulari de cerca.

• **Crear Projecte:**

Actors: Usuaris autenticats.

Descripció: Permet crear un projecte mitjançant el formulari de la pantalla "Crear".

Accions dels actors	Accions de l'aplicació
1. L'usuari vol crear un projecte nou i fa clic a la pestanya "Crear".	
	2. L'aplicació comprova quina és la última referència assignada a un projecte i proposa una nova referència pel projecte a crear.
	3. Mostra la pantalla "Crear".
4. L'usuari introdueix en el formulari d'entrada les dades bàsiques del projecte.	
5. Indica una nova referència si la proposada no és satisfactòria.	
6. Fa clic al botó "Crear".	
	7. L'aplicació mostra un missatge d'error si l'usuari no ha indicat un nom pel projecte i continua amb el pas 2.
	8. Comprova si la referència proposada ja existeix i en cas d'existir mostra un missatge d'error i continua amb el pas 3.
	9. Realitza l'alta del projecte i mostra la pantalla de visualització: "Ficha de Proyecto".

• **Visualitzar Projecte:**

Actors: Usuaris autenticats.

Descripció: Permet visualitzar les dades d'un projecte mitjançant la pantalla "Ficha de Proyecto".

Accions dels actors	Accions de l'aplicació
1. L'usuari vol visualitzar les dades d'un projecte i fa doble clic a algun dels resultats de la cerca de projectes.	
	2. L'aplicació mostra la pantalla "Ficha de Proyecto".

• **Modificar Projecte:**

Actors: Usuaris autenticats.

Descripció: Permet modificar les dades d'un projecte mitjançant la pantalla "Modificar Proyecto".

Accions dels actors	Accions de l'aplicació
1. L'usuari vol modificar les dades d'un projecte i fa clic a l'enllaç "Modificar" de la pantalla "Ficha de Proyecto".	
	2. L'aplicació mostra la pantalla "Modificar Proyecto".
3. L'usuari modifica els camps a través del formulari.	
4. Fa clic al botó "Modificar".	
	5. L'aplicació mostra un missatge d'error si l'usuari no ha indicat un nom pel projecte i continua amb el pas 3.
	6. Realitza la modificació del projecte i mostra la pantalla de visualització: "Ficha de Proyecto".

• **Associar contactes a un Projecte:**

Actors: Usuaris autenticats.

Descripció: Permet associar qualsevol contacte a un projecte mitjançant la pantalla "Asociar Contactos".

Accions dels actors	Accions de l'aplicació
1. L'usuari vol associar contactes a un projecte i fa clic a l'enllaç "Añadir Contactos" de la pantalla "Ficha de Proyecto".	
	2. L'aplicació mostra la pantalla "Asociar Contactos".
3. L'usuari selecciona (fent clic) tots els contactes de la llista que vol associar al projecte.	
	4. L'aplicació mostra una llista dels contactes seleccionats a la part dreta de la pantalla.
5. L'usuari fa clic a "Asociar Seleccionados"	
	6. L'aplicació associa tots els contactes que ha seleccionat l'usuari.
	7. Mostra la pantalla de visualització "Ficha de Proyecto" amb la llista dels nous contactes associats.

- **Esborrar contactes associats a un Projecte:**

Actors: Usuaris autenticats.

Descripció: Permet esborrar qualsevol contacte associat a un projecte mitjançant la pantalla "Ficha de Proyecto".

Accions dels actors	Accions de l'aplicació
1. L'usuari vol esborrar un contacte associat a un projecte i fa clic a la "X" vermella, que podem trobar a la dreta de cada contacte associat de la pantalla "Ficha de Proyecto"	
	2. L'aplicació mostra un missatge per confirmar que l'usuari desitja esborrar el contacte associat.

3. L'usuari fa clic a "Aceptar".	
	4. L'aplicació esborra l'associació entre el contacte indicat i el projecte.
	5. Mostra la pantalla de visualització "Ficha de Proyecto" amb la llista de contactes associats actualitzada.
Pas alternatiu:	
3. L'usuari fa clic a "Cancelar".	
	4. L'aplicació no realitza cap acció i deixa la pantalla tal com estava.

- **Crear expedients d'un Projecte:**

Actors: Usuaris autenticats.

Descripció: Permet crear nous expedients per a un projecte mitjançant la pantalla "Seguimiento de Expediente".

Accions dels actors	Accions de l'aplicació
	1. L'aplicació comprova l'estat del projecte i quins tipus d'expedients han estat generats anteriorment.
	2. L'estat del projecte és "En Espera" i deixa accessible el formulari d'Avantprojecte.
3. L'usuari informa el formulari d'Avantprojecte sense indicar el camp "Fecha finalización".	
4. Fa clic a "Guardar".	
	5. L'aplicació comprova la validesa dels camps informats. Si algun camp no és correcte mostra un missatge d'error i passa al pas 3.
	6. Desa l'expedient Avantprojecte informat.
	7. Modifica l'estat del projecte a: "Activo: Anteproyecto" i passa a l'estat 1.
Pas alternatiu 1A:	
3. L'usuari informa el formulari d'Avantprojecte indicant el camp "Fecha finalización".	

4. Fa clic a "Guardar".	
	5. L'aplicació comprova la validesa dels camps informats. Si algun camp no és correcte mostra un missatge d'error i passa al pas 3.
	6. Desa l'expedient Avantprojecte informat.
	7. Modifica l'estat del projecte a: "Activo: Básico" i passa a l'estat 1.
Pas alternatiu 2:	
	2. L'estat del projecte és "Activo: Anteproyecto" i deixa accessible el formulari d'Avantprojecte i Bàsic.
3. L'usuari informa el formulari d'Avantprojecte sense indicar el camp "Fecha finalización".	
4. Fa clic a "Guardar".	
	5. L'aplicació comprova la validesa dels camps informats. Si algun camp no és correcte mostra un missatge d'error i passa al pas 3.
	6. Desa l'expedient Avantprojecte informat i passa a l'estat 1.
Pas alternatiu 2A:	
3. L'usuari informa el formulari d'Avantprojecte indicant el camp "Fecha finalización".	
4. Fa clic a "Guardar".	
	5. L'aplicació comprova la validesa dels camps informats. Si algun camp no és correcte mostra un missatge d'error i passa al pas 3.
	6. Desa l'expedient Avantprojecte informat.
	7. Modifica l'estat del projecte a: "Activo: Básico" i passa a l'estat 1.
Pas alternatiu 2B:	
3. L'usuari informa el formulari de Bàsic sense indicar el camp "Fecha finalización".	
4. Fa clic a "Guardar".	
	5. L'aplicació comprova la validesa dels camps informats. Si algun camp no és correcte mostra un missatge d'error i passa al pas 3.
	6. Informa automàticament la data de finalització

	i desa l'expedient Avantprojecte.
	7. Desa l'expedient Bàsic informat.
	8. Modifica l'estat del projecte a: "Activo: Básico" i passa a l'estat 1.
Pas alternatiu 2C:	
3. L'usuari informa el formulari de Bàsic indicant el camp "Fecha finalización".	
4. Fa clic a "Guardar".	
	5. L'aplicació comprova la validesa dels camps informats. Si algun camp no és correcte mostra un missatge d'error i passa al pas 3.
	6. Informa automàticament la data de finalització i desa l'expedient Avantprojecte.
	7. Desa l'expedient Bàsic informat.
	8. Modifica l'estat del projecte a: "Activo: Ejecución" i passa a l'estat 1.
Pas alternatiu 3:	
	2. L'estat del projecte és "Activo: Básico" i deixa accessible el formulari de Bàsic i Execució.
3. L'usuari informa el formulari de Bàsic sense indicar el camp "Fecha finalización".	
4. Fa clic a "Guardar".	
	5. L'aplicació comprova la validesa dels camps informats. Si algun camp no és correcte mostra un missatge d'error i passa al pas 3.
	6. Desa l'expedient Bàsic informat i passa a l'estat 1.
Pas alternatiu 3A:	
3. L'usuari informa el formulari de Bàsic indicant el camp "Fecha finalización".	
4. Fa clic a "Guardar".	
	5. L'aplicació comprova la validesa dels camps informats. Si algun camp no és correcte mostra un missatge d'error i passa al pas 3.
	6. Desa l'expedient Bàsic informat.
	7. Modifica l'estat del projecte a: "Activo: Ejecución" i passa a l'estat 1.
Pas alternatiu 3B:	

3. L'usuari informa el formulari d'Execució sense indicar el camp "Fecha finalización".	
4. Fa clic a "Guardar".	
	5. L'aplicació comprova la validesa dels camps informats. Si algun camp no és correcte mostra un missatge d'error i passa al pas 3.
	6. Informa automàticament la data de finalització i desa l'expedient Bàsic.
	7. Desa l'expedient d'Execució informat.
	8. Modifica l'estat del projecte a: "Activo: Ejecución" i passa a l'estat 1.
Pas alternatiu 3C:	
3. L'usuari informa el formulari d'Execució indicant el camp "Fecha finalización".	
4. Fa clic a "Guardar".	
	5. L'aplicació comprova la validesa dels camps informats. Si algun camp no és correcte mostra un missatge d'error i passa al pas 3.
	6. Informa automàticament la data de finalització i desa l'expedient Bàsic.
	9. Desa l'expedient Execució informat.
	10. Modifica l'estat del projecte a: "Activo: Final de Obra" i passa a l'estat 1.
Pas alternatiu 4:	
	2. L'estat del projecte és "Activo: Ejecución" i deixa accessible el formulari d'Execució i Final d'Obra.
3. L'usuari informa el formulari d'Execució sense indicar el camp "Fecha finalización".	
4. Fa clic a "Guardar".	
	5. L'aplicació comprova la validesa dels camps informats. Si algun camp no és correcte mostra un missatge d'error i passa al pas 3.
	6. Desa l'expedient Execució informat i passa a l'estat 1.
Pas alternatiu 4A:	
3. L'usuari informa el formulari d'Execució indicant el camp "Fecha finalización".	
4. Fa clic a "Guardar".	

	5. L'aplicació comprova la validesa dels camps informats. Si algun camp no és correcte mostra un missatge d'error i passa al pas 3.
	6. Desa l'expedient Execució informat.
	7. Modifica l'estat del projecte a: "Activo: Final de Obra" i passa a l'estat 1.
Pas alternatiu 4B:	
3. L'usuari informa el formulari de Final d'Obra sense indicar el camp "Fecha finalización".	
4. Fa clic a "Guardar".	
	5. L'aplicació comprova la validesa dels camps informats. Si algun camp no és correcte mostra un missatge d'error i passa al pas 3.
	6. Informa automàticament la data de finalització i desa l'expedient Execució.
	7. Desa l'expedient de Final d'Obra informat.
	8. Modifica l'estat del projecte a: "Activo: Final de Obra" i passa a l'estat 1.
Pas alternatiu 4C:	
3. L'usuari informa el formulari de Final d'Obra indicant el camp "Fecha finalización".	
4. Fa clic a "Guardar".	
	5. L'aplicació comprova la validesa dels camps informats. Si algun camp no és correcte mostra un missatge d'error i passa al pas 3.
	6. Informa automàticament la data de finalització i desa l'expedient Execució.
	7. Desa l'expedient Final d'Obra informat.
	8. Modifica l'estat del projecte a: "Concluido" i passa a l'estat 1.
Pas alternatiu 5:	
	2. L'estat del projecte és "Activo: Final de Obra" i deixa accessible el de Final d'Obra.
3. L'usuari informa el formulari de Final d'Obra sense indicar el camp "Fecha finalización".	
4. Fa clic a "Guardar".	

	5. L'aplicació comprova la validesa dels camps informats. Si algun camp no és correcte mostra un missatge d'error i passa al pas 3.
	6. Desa l'expedient Final d'Obra informat i passa a l'estat 1.
Pas alternatiu 5A:	
3. L'usuari informa el formulari de Final d'Obra indicant el camp "Fecha finalización".	
4. Fa clic a "Guardar".	
	5. L'aplicació comprova la validesa dels camps informats. Si algun camp no és correcte mostra un missatge d'error i passa al pas 3.
	6. Desa l'expedient Final d'Obra informat.
	7. Modifica l'estat del projecte a: "Concluido" i passa a l'estat 1.

4.3.2.2 Mòdul de Contactes

- **Cercar Contactes:**

Actors: Usuaris autènticats.

Descripció: Cerca de contactes mitjançant el formulari de la pantalla "Consultar".

Accions dels actors	Accions de l'aplicació
1. L'usuari vol realitzar una cerca de contactes i fa clic a la opció de menú "Contactos".	
	2. L'aplicació mostra la pantalla "Consultar".
3. L'usuari introdueix en el formulari d'entrada les dades a cercar.	
4. Fa clic al botó "Buscar".	
	5. L'aplicació realitza la cerca dels contactes que compleixen tots els requisits que ha introduït l'usuari a través del formulari.
	6. Mostra per pantalla la llista de contactes

	trobats ordenats primer per tipus i després per alfabètic. Depenent del camp cercat, l'aplicació mostrarà en vermell els camps que coincideixin amb els introduïts per l'usuari.
Pas alternatiu:	
7. L'usuari desitja esborrar tots els camps de cerca del formulari i fa clic al botó "Limpiar".	
	8. L'aplicació esborra tots els camps del formulari de cerca.

- **Crear Contacte:**

Actors: Usuaris autenticats.

Descripció: Permet crear un contacte mitjançant el formulari de la pantalla "Crear".

Accions dels actors	Accions de l'aplicació
1. L'usuari vol crear un contacte nou i fa clic a la pestanya "Crear".	
	2. L'aplicació mostra la pantalla "Crear".
3. L'usuari introdueix en el formulari d'entrada les dades bàsiques del contacte.	
4. Fa clic al botó "Crear".	
	5. L'aplicació mostra un missatge d'error si l'usuari no ha indicat un nom de contacte i/o un tipus i continua amb el pas 2.
	6. Comprova si el nom del contacte ja existeix i en cas afirmatiu mostra un missatge d'error i continua amb el pas 2.
	7. Realitza l'alta del contacte i mostra la pantalla de visualització: "Consultar Contacte".
Pas alternatiu:	
3. L'usuari desitja crear un nou tipus de contacte.	
4. Fa clic a l'enllaç "Añadir Tipo".	
	5. L'aplicació mostra una finestra emergent amb el formulari necessari per crear un nou tipus de contacte.

6. L'usuari indica el nom del nou tipus de contacte.	
	7. L'aplicació dona d'alta el nou tipus de contacte.
	8. Tanca la finestra emergent.
	9. Refresca la llista de tipus per introduir el nou tipus creat per l'usuari.

- **Visualitzar Contacte:**

Actors: Usuaris autenticats.

Descripció: Permet visualitzar les dades d'un contacte mitjançant la pantalla "Consultar Proyecto".

Accions dels actors	Accions de l'aplicació
1. L'usuari vol visualitzar les dades d'un contacte i fa doble clic a algun dels resultats de la cerca de contactes.	
	2. L'aplicació mostra la pantalla "Consultar Contacto".

- **Modificar Contacte:**

Actors: Usuaris autenticats.

Descripció: Permet modificar les dades d'un contacte mitjançant la pantalla "Modificar Contacto".

Accions dels actors	Accions de l'aplicació
1. L'usuari vol modificar les dades d'un contacte i fa clic a l'enllaç "Modificar" de la pantalla "Consultar Contacto".	
	2. L'aplicació mostra la pantalla "Modificar Contacto".
3. L'usuari modifica els camps a través del formulari.	
4. Fa clic al botó "Modificar".	

	5. L'aplicació mostra un missatge d'error si l'usuari no ha indicat un nom de contacte i/o un tipus i continua amb el pas 2.
	6. Realitza la modificació del contacte i mostra la pantalla de visualització: "Consultar Contacto".
Pas alternatiu:	
3. L'usuari desitja crear un nou tipus de contacte.	
4. Fa clic a l'enllaç "Añadir Tipo".	
	5. L'aplicació mostra una finestra emergent amb el formulari necessari per crear un nou tipus de contacte.
6. L'usuari indica el nom del nou tipus de contacte.	
	7. L'aplicació dona d'alta el nou tipus de contacte.
	8. Tanca la finestra emergent.
	9. Refresca la llista de tipus per introduir el nou tipus creat per l'usuari.

- **Associar Contactes:**

Actors: Usuaris autenticats.

Descripció: Permet associar qualsevol contacte a un altre contacte mitjançant la pantalla "Asociar".

Accions dels actors	Accions de l'aplicació
1. L'usuari vol associar contactes a un altre contacte i fa clic a l'enllaç "Añadir Relaciones" de la pantalla "Consultar Contacto".	
	2. L'aplicació mostra la pantalla "Asociar" amb dos llistes de contactes.
	3. El contacte actual es mostra seleccionat a la primera de les llistes.
4. L'usuari selecciona (fent clic) tots els contactes de la segona llista que vol associar al contacte de la primera llista.	
5. L'usuari informa els camps "Descripción de	

la relació”.	
6. Fa clic a “Asociar”.	
	7. L'aplicació mostra un missatge d'error en cas que no hagi indicat cap contacte de la primera o segona llista i en cas que no hagi indicat algun dels camps de la descripció de la relació i torna al pas 4.
	8. L'aplicació associa tots els contactes seleccionats a la segona llista amb el seleccionat a la primera llista.
	9. Mostra la pantalla de visualització “Consultar Contacte” del contacte seleccionat a la primera llista amb els contactes associats de la segona

• **Esborrar contactes associats a un Contacte:**

Actors: Usuaris autènticats.

Descripció: Permet esborrar qualsevol contacte associat a un altre contacte mitjançant la pantalla “Consultar Contacte”.

Accions dels actors	Accions de l'aplicació
1. L'usuari vol esborrar un contacte associat a un altre contacte i fa clic a la “X” vermella, que podem trobar a la dreta de cada contacte associat de la pantalla “Consultar Contacte”.	
	2. L'aplicació mostra un missatge per confirmar que l'usuari desitja esborrar el contacte associat.
3. L'usuari fa clic a “Acceptar”.	
	4. L'aplicació esborra l'associació entre els contactes.
	5. Mostra la pantalla de visualització “Consultar Contacte” amb la llista de contactes associats actualitzada.
Pas alternatiu:	
3. L'usuari fa clic a “Cancelar”.	
	4. L'aplicació no realitza cap acció i deixa la pantalla tal com estava.

- **Visualitzar llista de tipus de contacte:**

Actors: Usuaris autenticats.

Descripció: Permet visualitzar una llista de tipus de contacte mitjançant la pantalla "Crear".

Accions dels actors	Accions de l'aplicació
1. L'usuari vol visualitzar els tipus de contacte actuals i fa clic a l'opció de menú "Tipos".	
	2. L'aplicació mostra la pantalla "Crear" amb una llista de tots els tipus de contacte existents.

- **Crear tipus de contacte:**

Actors: Usuaris autenticats.

Descripció: Permet crear un nou tipus de contacte mitjançant el formulari de la pantalla "Crear".

Accions dels actors	Accions de l'aplicació
1. L'usuari vol crear un nou tipus de contacte i fa clic a l'opció de menú "Tipos".	
	2. L'aplicació mostra la pantalla "Crear" amb una llista de tots els tipus de contacte existents.
3. L'usuari introdueix en el formulari d'entrada el nom del nou tipus de contacte.	
4. Fa clic al botó "Crear".	
	5. L'aplicació mostra un missatge d'error si l'usuari no ha indicat cap nom de tipus de contacte i continua amb el pas 3.
	10. Realitza l'alta del nou tipus de contacte i mostra la pantalla de visualització: "Crear".

• **Modificar tipus de contacte:**

Actors: Usuaris autenticats.

Descripció: Permet modificar el nom d'un tipus de contacte existent mitjançant la pantalla "Modificar Tipo Contacto".

Accions dels actors	Accions de l'aplicació
1. L'usuari vol modificar el nom d'un tipus de contacte i fa doble clic a algun tipus de contacte de la llista de la pantalla "Crear".	
	2. L'aplicació mostra la pantalla "Modificar Tipo Contacto".
3. L'usuari modifica el nom del tipus de contacte.	
4. Fa clic al botó "Modificar".	
	5. L'aplicació mostra un missatge d'error si l'usuari no ha indicat un nom pel tipus de contacte i continua amb el pas 3.
	6. Realitza la modificació del tipus de contacte i mostra de nou la mateixa pantalla.

• **Esborrar tipus de contacte:**

Actors: Usuaris autenticats.

Descripció: Permet esborrar un tipus de contacte mitjançant la pantalla "Eliminar".

Accions dels actors	Accions de l'aplicació
1. L'usuari vol esborrar un tipus de contacte i fa clic a la pestanya "Eliminar".	
	2. L'aplicació mostra la pantalla "Eliminar" amb una llista de tots els tipus de contacte.
3. L'usuari llegeix la nota explicativa que alarma la possibilitat d'esborrar contactes que formin part del tipus a esborrar.	
4. Fa clic als tipus de contacte de la llista que desitja esborrar.	

	5. L'aplicació mostra a la dreta de la pantalla una llista de contactes associats als tipus marcats per l'usuari.
6. L'usuari fa clic a "Eliminar seleccionados".	
	7. L'aplicació mostra un missatge per confirmar que l'usuari desitja esborrar el tipus de contacte.
8. L'usuari fa clic a "Aceptar".	
	9. L'aplicació esborra els contactes associats als tipus seleccionats.
	10. Esborra els tipus de contactes associats.
	11. Mostra un missatge indicant que s'han esborrat correctament i mostra de nou la pàgina "Eliminar" amb la llista de contactes actualitzada.
Pas alternatiu:	
8. L'usuari fa clic a "Cancelar".	
	9. L'aplicació no realitza cap acció i deixa la pantalla tal com estava.

4.3.2.3 Mòdul d'Usuaris

- **Cercar Usuaris:**

Actors: Usuaris autenticats.

Descripció: Cerca de contactes mitjançant el formulari de la pantalla "Consultar".

Accions dels actors	Accions de l'aplicació
1. L'usuari vol realitzar una cerca d'usuaris i fa clic a la opció de menú "Usuarios".	
	2. L'aplicació mostra la pantalla "Consultar".
3. L'usuari introdueix en el formulari d'entrada les dades a cercar.	
4. Fa clic al botó "Buscar".	
	5. L'aplicació realitza la cerca dels usuaris que compleixen tots els requisits que ha introduït l'usuari a través del formulari.

	6. Mostra per pantalla la llista d'usuaris trobats ordenats per nom d'usuari. Depenent del camp cercat, l'aplicació mostrarà en vermell els camps que coincideixin amb els introduïts per l'usuari.
Pas alternatiu:	
7. L'usuari desitja esborrar tots els camps de cerca del formulari i fa clic al botó "Limpiar".	
	8. L'aplicació esborra tots els camps del formulari de cerca.

- **Crear Usuaris:**

Actors: Usuaris autenticats amb rol Administrador.

Descripció: Permet crear un usuari mitjançant el formulari de la pantalla "Crear".

Accions dels actors	Accions de l'aplicació
1. L'usuari administrador vol crear un usuari nou i fa clic a la pestanya "Crear".	
	2. L'aplicació mostra la pantalla "Crear".
3. L'usuari introdueix en el formulari d'entrada les dades bàsiques de l'usuari.	
4. Fa clic al botó "Crear".	
	5. L'aplicació mostra un missatge d'error si l'usuari no ha indicat tots els camps del formulari i continua amb el pas 3.
	6. Comprova si el nom d'usuari ja existeix i en cas afirmatiu mostra un missatge d'error i continua amb el pas 3.
	7. Realitza l'alta de l'usuari i mostra la pantalla de visualització: "Consultar Usuario".

• **Visualitzar Usuari:**

Actors: Usuaris autenticats.

Descripció: Permet visualitzar les dades d'un usuari mitjançant la pantalla "Consultar Usuario".

Accions dels actors	Accions de l'aplicació
1. L'usuari vol visualitzar les dades d'un altre usuari i fa doble clic a algun dels resultats de la cerca d'usuaris.	
	2. L'aplicació mostra la pantalla "Consultar Usuario".

• **Modificar Usuari:**

Actors: Usuaris autenticats amb rol Administrador.

Descripció: Permet modificar les dades d'un usuari mitjançant la pantalla "Modificar Usuario".

Accions dels actors	Accions de l'aplicació
1. L'usuari administrador vol modificar les dades d'un usuari i fa clic a l'enllaç "Modificar" de la pantalla "Consultar Usuario".	
	2. L'aplicació mostra la pantalla "Modificar Usuario".
3. L'usuari modifica els camps a través del formulari.	
4. Fa clic al botó "Modificar".	
	5. L'aplicació mostra un missatge d'error si l'usuari no ha indicat tots els camps del formulari i continua amb el pas 3.
	7. Realitza la modificació de l'usuari i mostra la pantalla de visualització: "Consultar Usuario".

- **Esborrar Usuarios:**

Actors: Usuaris autenticats amb rol Administrador.

Descripció: Permet esborrar lògicament un usuari mitjançant la pantalla “Eliminar Usuario”.

Accions dels actors	Accions de l'aplicació
1. L'usuari administrador vol esborrar un usuari i fa clic a la “X” vermella que apareix al costat de la llista d'usuaris de la pantalla “Consultar”.	
	2. L'aplicació mostra un missatge de confirmació.
3. L'usuari fa clic a “Aceptar”.	
	4. L'aplicació fa un esborrat lògic de l'usuari.
	5. Mostra la pantalla “Consultar”.

4.4 Model d'estats

Els models d'estats ens permeten visualitzar les diferents condicions que pot tenir un objecte o un cas d'ús a través del temps i com i què necessita per passar d'una condició a una altra.

4.4.1 Diagrama d'estats

L'objecte "Projectes" és l'element més canviant de l'aplicació, ja que segueix una seqüència en funció dels expedients que es van creant: des que és creat, en estat "En espera", fins que es declara com a finalitzat, en estat "Conclòs". En el següent diagrama d'estats podem veure les diferents etapes de la vida d'un Projecte:

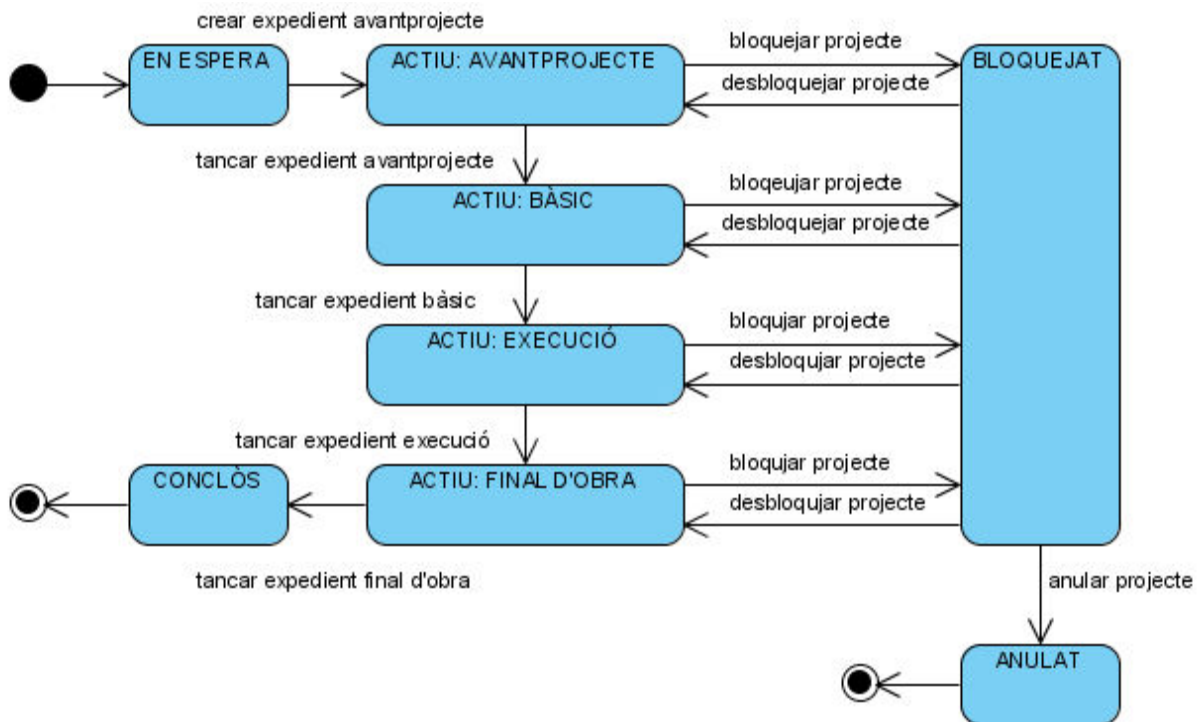


Fig. 10: Diagrama d'estats d'un projecte

5 Disseny

El disseny és l'etapa anterior a la implementació de qualsevol aplicació. En aquesta etapa es constitueix l'arquitectura que millor s'adapti a les necessitats de l'aplicació.

Del disseny depenen punts importants com la seguretat, l'escalabilitat, la disponibilitat i la reutilització.

- **Seguretat:** per tal de realitzar un disseny segur cal:
 - Confiar en sistemes de seguretat ja provats.
 - No confiar mai en les dades informades per l'usuari, verificar-les sempre.
 - Assumir que els sistemes externs no són segurs.
 - Aplicar el principi de privilegis mínims.
 - Reduir la disponibilitat de components i dades.
 - Amagar dades no és un sistema de seguretat.

A la nostra aplicació els elements de seguretat no seran massa estrictes, ja que l'aplicació només serà accessible pels usuaris de l'empresa, els quals, a excepció de crear i esborrar usuaris, podran accedir a la totalitat de l'aplicació. De totes maneres, l'aplicació s'ha d'implementar de manera que, si fes falta més endavant, es puguin introduir diferents rangs d'usuaris o més elements de seguretat.

- **Escalabilitat:** ens permet augmentar la capacitat de servei de forma lineal incrementant el nombre de recursos que disposa l'aplicació. Per aconseguir-ho hem d'intentar dissenyar processos que:
 - No esperin.
 - No es barallin per accedir als recursos.
 - Es puguin commutar.
 - Es puguin intercanviar.

Encara que el client de l'aplicació dissenyada no té masses treballadors, cal tenir en compte que aquesta mateixa aplicació podria ser utilitzada per altres empreses amb

quantitats relativament grans de treballadors. Llavors cal evitar temps d'esperes i sobretot, evitar sobreescrivre la feina d'altres treballadors. Per evitar-ho, s'han utilitzat camps d'auditoria que permetran a qualsevol modificació de les dades, saber si algú altre les ha modificat mentre ho estàvem fent nosaltres.

- **Disponibilitat:** una aplicació ha d'estar disponible pel seu ús cada cert temps. Decidir el nivell de disponibilitat depèn de les expectatives del client, de si aplicacions externes en depenen i sobretot, quin temps de “no disponibilitat” és acceptable. Per aconseguir-ho podem utilitzar:
 - Balanceig de càrrega.
 - Cues de missatges.
 - Components per separar les dades més utilitzades.

El volum de dades tractat pel nostre client no pot arribar mai a ser massa extens. Per a clients grans podria arribar a ser molt extens, però no pas accedit de forma massiva. Així doncs, podem assegurar que la disponibilitat de les dades a la nostra aplicació serà pràcticament sempre immediata i no ha fet falta realitzar cap disseny específic al respecte.

- **Reutilització:** és important dissenyar l'aplicació de manera que no haguem de repetir codi o realitzar diferents implementacions massa semblants. Per aconseguir-ho hem de separar el codi en mòduls i aconseguir que la dependència entre ells sigui baixa.

5.1 Arquitectura d'una aplicació web

L'arquitectura defineix l'estructura que ens permet unir totes les implementacions necessàries, és el marc de referència que ens marcarà el camí per construir l'aplicació. Realitzant una bona estructura inicial, aconseguirem simplificar aquest camí i podrem establir tots els objectius de seguretat, escalabilitat, disponibilitat i reutilització.

5.2 Decisions de disseny

Habitualment, podrem adaptar qualsevol arquitectura coneguda al nostre projecte. Per això hem de conèixer les més habituals.

- Monolítica: totes les implementacions estan íntimament lligades, tant a nivell físic com a nivell lògic. No és possible la distribució de cap dels seus components.
- Client – servidor: la càrrega de treball es divideix en dos parts independents, però no hi ha una distribució de funcions entre elles.
- De tres capes: és una expansió de l'arquitectura client – servidor. La càrrega es divideix en tres parts segons les següents funcionalitats: la presentació, els càlculs i la persistència. Les capes van en aquest ordre i només tenen relació amb la immediata.

Per al nostre projecte hem adaptat l'arquitectura de tres capes, altrament coneguda com el model vista – controlador.

La capa encarregada de la interfície d'usuari, se l'anomena capa de presentació; la de càlcul, capa de negoci; i la de persistència, capa de dades. A continuació explicarem el disseny de cada una d'aquestes capes.

5.3 Disseny de la capa de presentació

La capa de presentació és l'encarregada de mostrar la interfície a l'usuari de l'aplicació, comunicar la informació i capturar les accions de l'usuari. Només pot comunicar-se amb la capa de negoci, on es faran els càlculs necessaris per poder realitzar les funcionalitats. Tot i que aquesta capa no està dissenyada per a realitzar càlculs relatius a les funcionalitats, sí que hi podem realitzar filtres per evitar errors de format, d'aquesta manera aconseguirem una interfície més pràctica i amigable.

Aquesta capa també serà l'encarregada de distribuir els diferents fluxos de l'aplicació a la part corresponent de la capa de negoci. Aquesta distribució no es considera una realització de càlculs encara que se n'haguessin de realitzar per tal de poder distribuir correctament.

5.3.1 Mòdul de Projectes

- Cercar Projectes:

REF	Nombre	Fecha Creación	Fecha Modificación	Estado
08/023	Cinco fincas Fontferrol	16/06/2008	28/10/2008	En Espera
08/022	Prueba fecha apertura 2	10/06/2008	10/06/2008	En Espera
08/021	Prueba fecha apertura	10/06/2008	10/06/2008	En Espera
08/020	Proyecto prueba redireccion	10/06/2008	10/06/2008	En Espera
08/019	Proyecto de prueba con todos los datos	24/04/2008	10/06/2008	Activo: Ejecución
08/018	caja archivo	24/04/2008	10/06/2008	Activo: Básico
08/017	Prova sense fitxer	21/04/2008	10/06/2008	Concluido
08/016	Prova fitxer 3	21/04/2008	10/06/2008	Activo: Final de Obra
08/015	Prova fitxer 2	21/04/2008	21/04/2008	En Espera
08/014	Prova fitxer	21/04/2008	22/06/2008	Activo: Básico
08/013	Proyecto de prueba 3	07/04/2008	10/06/2008	Concluido
08/012	Proyecto de prueba 3	07/04/2008	07/04/2008	En Espera
08/011	Proyecto de prueba 3	07/04/2008	07/04/2008	En Espera

Fig. 11: Pantalla de cerca de projectes

- Crear Projecte:

Fig. 12: Pantalla de creació de projectes

- Visualitzar Projecte:

GA GRUPO ALBERCA

Bienvenido, **Joaquim Hilari Horts** Tu última visita fue 27/07/2008 13:52:27

Inicio salir cambiar pwd NOTIFICACIONES

Consultar Crear **Ficha de Proyecto** Expediente Seguimiento de Obra

CONSULTAR PROYECTO 2008/019

Nombre: Estación de Metro Mayor **REF.: 08/019**

Situación: Calle Mayor 5

Código postal: 08080 **Población:** Barcelona

Nº Caja en archivo: 2521

Imagen: P5210002.JPG

Fecha de apertura: 24/04/2008 **[Modificar]**

LISTA DE CONTACTOS ASOCIADOS **[Añadir Contactos]**

Nombre		
ABOGADO	AMALIA	X
	ISABEL BONILLA	X
ADMINISTRADOR	ADMINISTRADOR CABO DE GATA	X
	ADMINISTRADOR DE FINCAS ALMERIA	X
ARQUITECTO	FRANCISCO ALAMEDA MOLINA	X
	LUIS SANCHEZ GARCIA	X
	MARIANA CHIODI	X
	TITO	X
AYUNTAMIENTO	MANOLO BERMEJO	X
CONTACTO	COLEGIO DE DELINEANTES	X
EMPRESA	TECNISUR	X
INMOBILIARIA	SOL VIVIENDA	X
PROMOTORA	GRUPO INMOBILIARIO AGUAMAR S.L	X

Fig. 13: Pantalla de visualització de projectes

- Modificar Projecte:

GA GRUPO ALBERCA

Bienvenido, **Joaquim Hilari Horts** Tu última visita fue 28/10/2008 19:22:31

Inicio salir cambiar pwd NOTIFICACIONES

Consultar Crear **Ficha de Proyecto** **Modificar Proyecto** Expediente Seguimiento de Obra

MODIFICAR UN PROYECTO EXISTENTE

Nombre: Estación de Metro Mayor **REF.: 08/019**

Situación: Calle Mayor 5

Código postal: 08080 **Población:** Barcelona

Nº Caja en archivo: 2521

Imagen:

Fecha de apertura: 24/04/2008 **Modificar**

Fig. 14: Pantalla de modificació de projectes

- **Asociar contactes a un Projecte:**

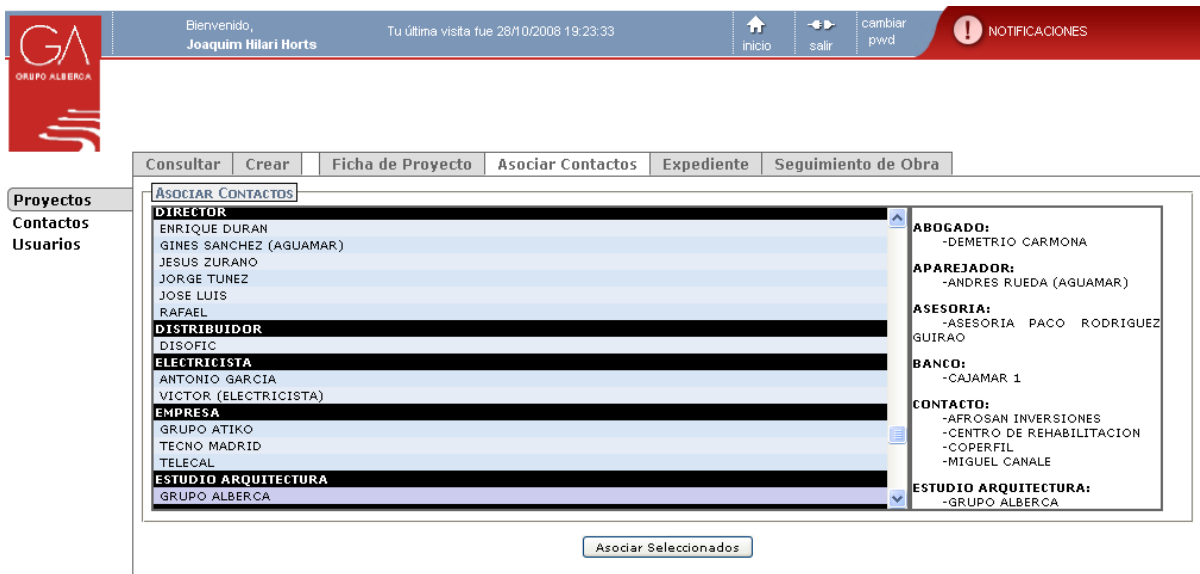


Fig. 15: Pantalla d'associació de contactes a projectes

- **Esborrar contactes associats a un Projecte:**

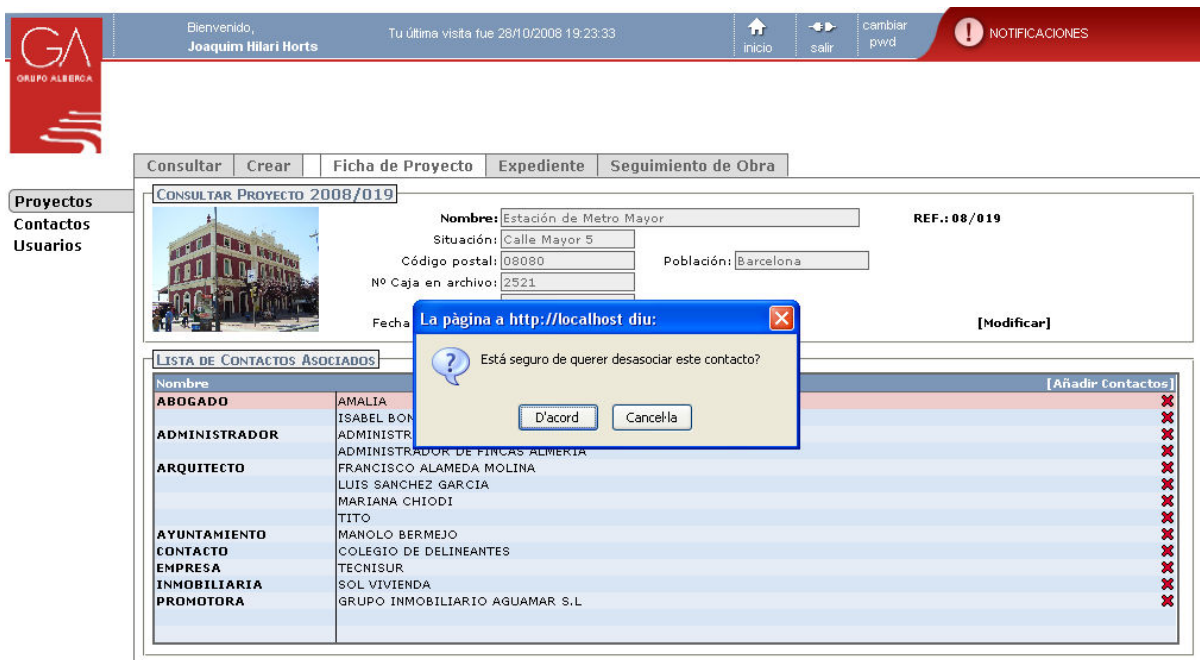


Fig. 16: Pantalla per esborrar contactes associats a projectes

- **Crear expedients d'un Projecte:**

Fig. 17: Pantalla de creació d'expedients de projectes

5.3.2 Mòdul de Contactes

- **Cercar Contactes:**

Nombre	Tipo	Móvil	Correo
IGNACIO EGEA	CONTACTO	639391104	
ARTURO EGEA	CONTACTO		
ECUS	CONTACTO	607921874	
RAFAEL DURBAN	CONTACTO	610226228	
JOHN DURAN	CONTACTO		
ENRIQUE DURAN	DIRECTOR	639118700	
INR OAL	PROMOTORA		
SERGIO DOMINGUEZ	CONTACTO	647672300	
DOMINGO SILVA	APAREJADOR	654513346	
DNI	CONTACTO		
DISOFIC	DISTRIBUIDOR	677576706	
DIPUTACION PROVINCIAL DE ALMERIA	CONTACTO		
DIMOBEL	CONTACTO		
DILOP	CONTACTO	670646424	
PACO DIAZ	CONTACTO	636990285	

Fig. 18: Pantalla de cerca de contactes

- **Crear Contacte:**

GA GRUPO ALBERCA

Bienvenido, **Joaquim Hilari Horts** Tu última visita fue 28/10/2008 19:30:58

Inicio salir cambiar pwd NOTIFICACIONES

Consultar Crear Asociar

CREAR UN NUEVO CONTACTO

Nombre: DNI/NIF:

Tipo: [Añadir Tipo]

Persona de contacto:

Cargo:

Dirección:

Código postal: Población:

Teléfono: Fax:

Tel. Trabajo: Fax Trabajo:

Móvil:

Correo electrónico:

Página web:

Crear

Fig. 19: Pantalla de creació de contactes

- **Visualitzar Contacte:**

GA GRUPO ALBERCA

Bienvenido, **Joaquim Hilari Horts** Tu última visita fue 28/10/2008 19:30:58

Inicio salir cambiar pwd NOTIFICACIONES

Consultar Crear Asociar Consultar Contacto

CONSULTAR CONTACTO

Nombre: **LUIS SANCHEZ GARCIA** DNI/NIF: **74716418k**

Tipo: **ARQUITECTO**

Persona de contacto:

Cargo:

Dirección: **C/Alcazar n4 Plt 1**

Código postal: **04006** Población: **Almería**

Teléfono: Fax:

Tel. Trabajo: **950281860** Fax Trabajo: **950281861**

Móvil: **657361592**

Correo electrónico: **luis@albercaproyectos.es**

Página web:

Colegiado: **1234567890XXXA** (solo para arquitectos) [Modificar]

LISTA DE PROYECTOS ASOCIADOS

REF	Nombre
08/019	Estación de Metro Mayor
08/013	Proyecto de prueba 3
08/001	Proyecto de prueba 1

LISTA DE CONTACTOS ASOCIADOS

Nombre	Tipo de relación	[Añadir Relaciones]

Fig. 20: Pantalla de visualització de contactes

- **Modificar Contacte:**

GA GRUPO ALBERCA

Bienvenido, **Joaquim Hilari Horts** Tu última visita fue 28/10/2008 19:30:58 inicio salir cambiar pwd NOTIFICACIONES

Consultar Crear Asociar Consultar Contacto **Modificar Contacto**

MODIFICAR UN CONTACTO EXISTENTE

Nombre: LUIS SANCHEZ GARCIA DNI/NIF: 74716418k

Tipo: ARQUITECTO [Añadir Tipo]

Persona de contacto:

Cargo:

Dirección: C/Alcazar n4 Plt 1

Código postal: 04006 Población: Almería

Teléfono: Fax:

Telf. Trabajo: 950281860 Fax Trabajo: 950281861

Móvil: 657361592

Correo electrónico: luis@albercaproyectos.es

Página web:

Colegiado: 1234567890XXXA (solo para arquitectos)

Proyectos
Contactos
Tipos
Usuarios

Fig. 21: Pantalla per modificar contactes

- **Asociar Contactes:**

GA GRUPO ALBERCA

Bienvenido, **Joaquim Hilari Horts** Tu última visita fue 28/10/2008 19:30:58 inicio salir cambiar pwd NOTIFICACIONES

Consultar Crear **Asociar**

ASOCIAR CONTACTOS

LUIS SANCHEZ GARCIA
MARIANA CHIODI
TITO
ASESOR
ANTONIO ESTEBAN
ASESORIA GADE
FRANCISCO
FRANCISCO MARTINEZ
MAIKE
ASESORIA
ABM
ASESORIA DELFOS
ASESORIA EDUARDO SEGURA
ASESORIA PACO RODRIGUEZ
GUIRAO
ASESORIA SALVADOR
AYUNTAMIENTO
ALCALDIA

Descripción de la Relación

Trabajador de

>-- Relación derecha -->

Arquitecto jefe

<-- Relación izquierda --<

ELECTRICISTA
ANTONIO GARCIA
VICTOR (ELECTRICISTA)
EMPRESA
GRUPO ATIKO
TECNISUR
TECNO MADRID
TELECAL
ESTUDIO ARQUITECTURA
GRUPO ALBERCA
INGENIERO
ENRIQUE ALCAZAR
INMOBILIARIA
SOL VIVIENDA
LIMPIEZA
TERESA (LIMPIEZA)
PELUQUERIA
PELLUQUERIA ANDREA

Proyectos
Contactos
Tipos
Usuarios

Fig. 22: Pantalla per associar contactes entre si

- **Esborrar contactes associats a un Contacte:**

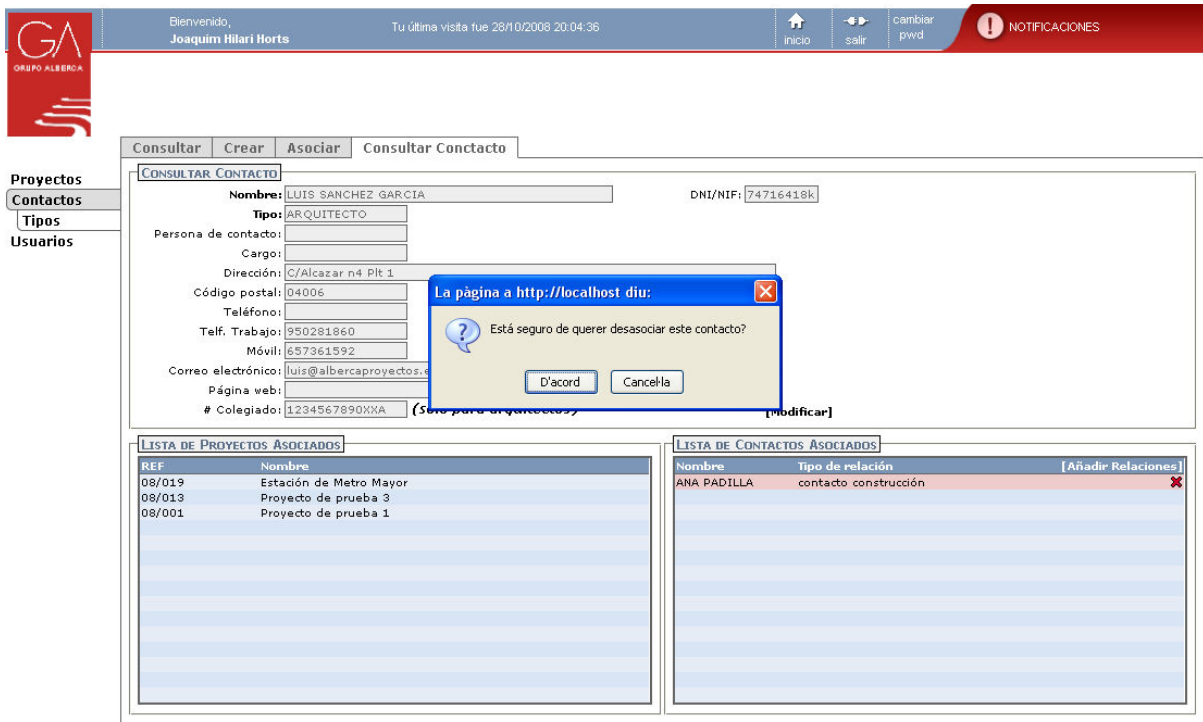


Fig. 23: Pantalla d'esborrar contactes associats entre si

- **Visualitzar llista de tipus de contacte:**

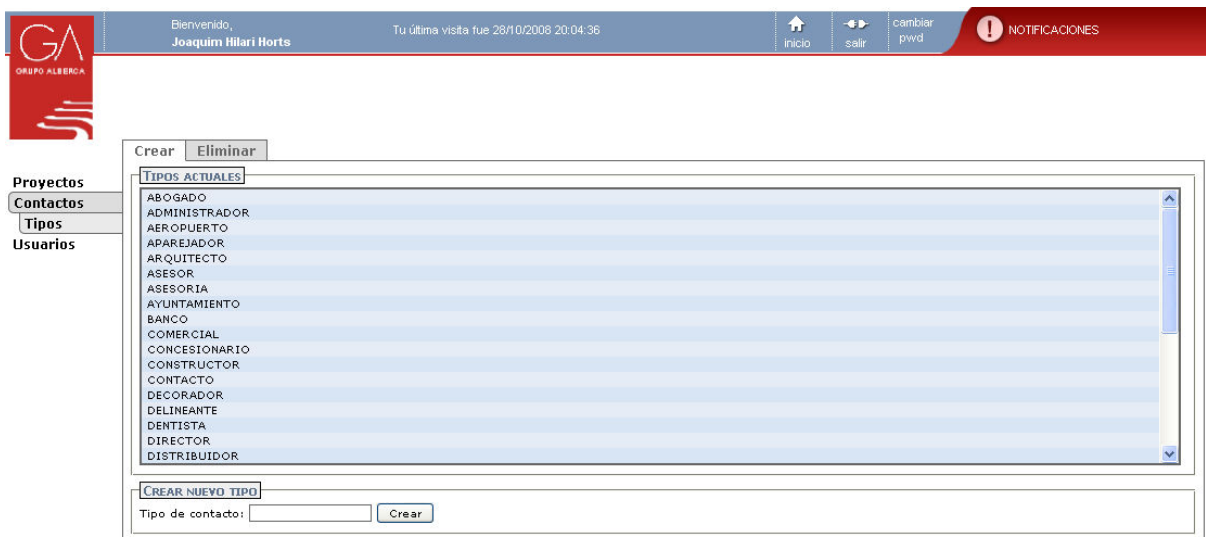


Fig. 24: Pantalla de visualització de tipus de contactes

- **Crear tipus de contacte:**

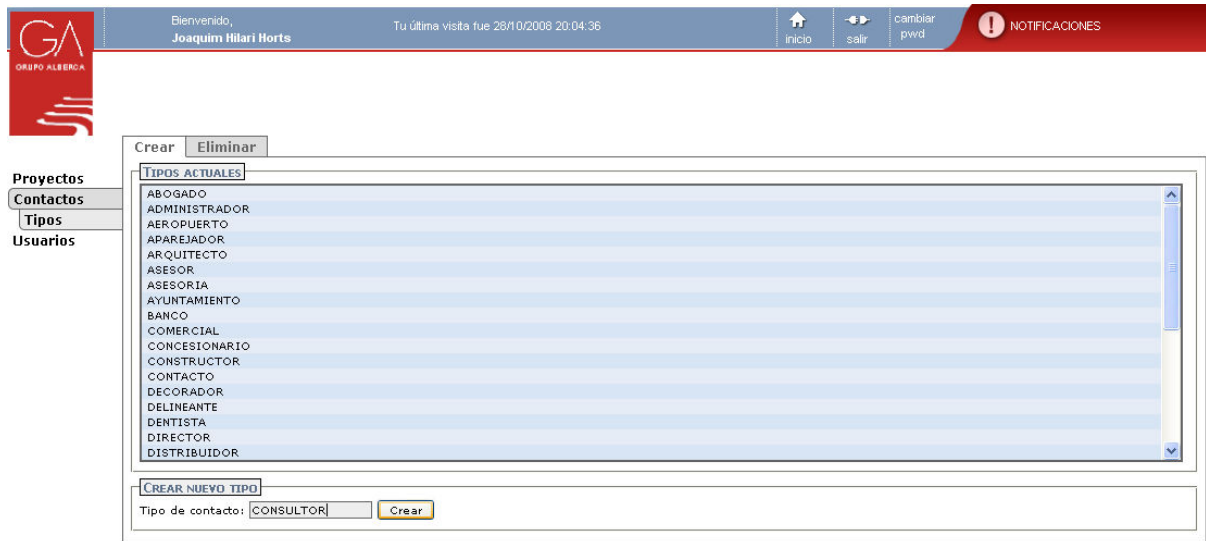


Fig. 25: Pantalla per crear tipus de contactes

- **Modificar tipus de contacte:**

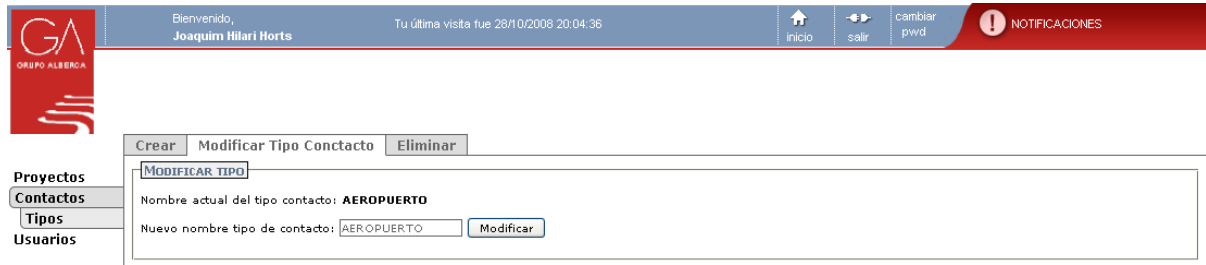


Fig. 26: Pantalla de modificació de tipus de contactes

- **Esborrar tipus de contacte:**

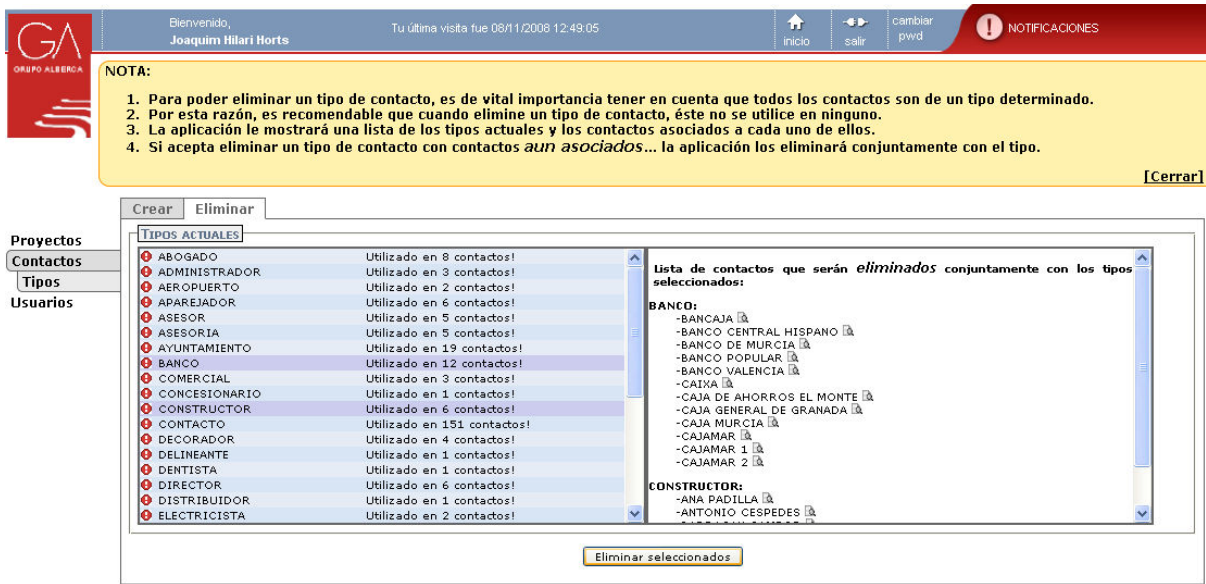


Fig. 27: Pantalla per esborrar tipus de contactes

5.3.3 Mòdul d'Usuaris

- **Cercar Usuaris:**

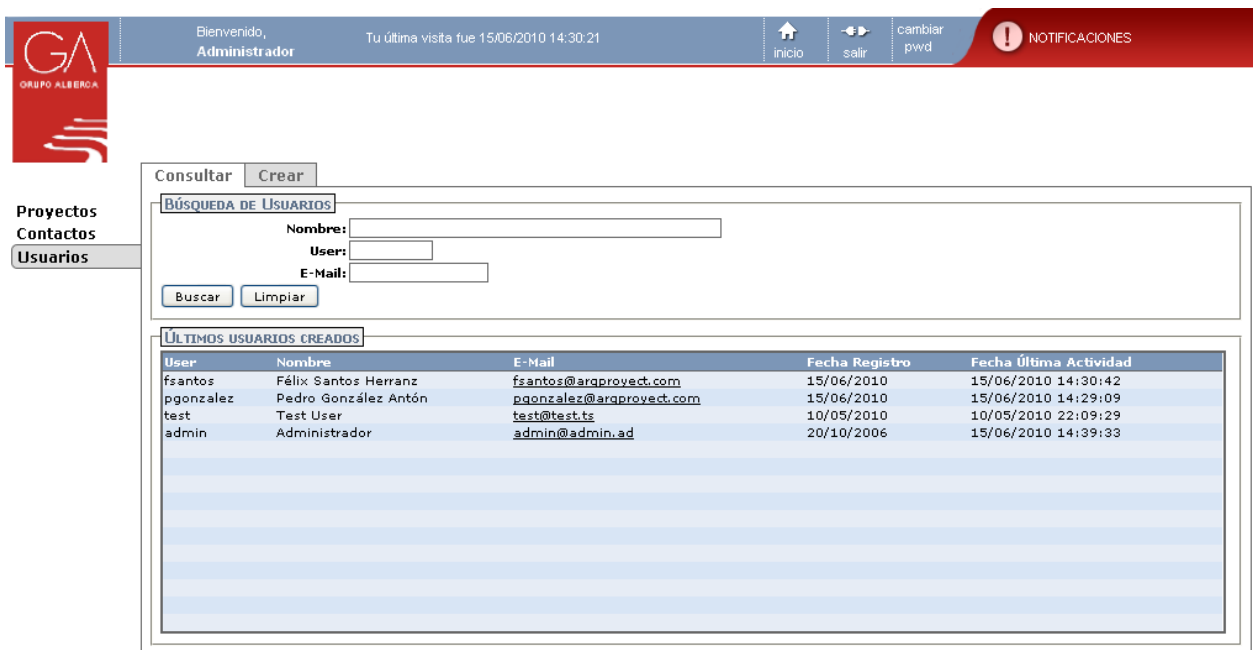


Fig. 28: Pantalla de cerca d'usuaris

- **Crear Usuaris:**

Bienvenido, Administrador Tu última visita fue 15/06/2010 14:30:21 inicio salir cambiar pwd NOTIFICACIONES

GA GRUPO ALBERCA

Proyectos Contactos Usuarios

Consultar Crear

CREAR UN NUEVO USUARIO

Nombre real: Félix Santos Herranz

Nombre de usuario: fsantos

Contraseña: *****

Repetir contraseña: *****

E-Mail: fsantos@arqproyect

Crear

Fig. 29: Pantalla per crear usuaris

- **Visualitzar Usuari:**

Bienvenido, Administrador Tu última visita fue 01/07/2010 16:58:22 inicio salir cambiar pwd NOTIFICACIONES

GA GRUPO ALBERCA

Proyectos Contactos Usuarios

Consultar Crear Ficha de Usuario

CONSULTAR USUARIO PGONZALEZ

Nombre real: Pedro González Antón

Nombre de usuario: pgonzalez

E-Mail: pgonzalez@arqproyect.com

Fecha de registro: 15/06/2010

Fecha último acceso: 15/06/2010 14:29:09

Fecha última actividad: 15/06/2010 14:29:09

[Modificar]

Fig. 30: Pantalla de visualització d'usuaris

- **Modificar Usuario:**

Bienvenido, Administrador Tu última visita fue 01/07/2010 16:58:22 inicio salir cambiar pwd NOTIFICACIONES

GA GRUPO ALBERCA

Proyectos Contactos Usuarios

Consultar Crear Ficha de Usuario Modificar Usuario

MODIFICAR UN USUARIO EXISTENTE

Nombre real: Pedro González Antón

Nombre de usuario: pgonzalez

Nueva contraseña: []

Repetir contraseña: []

E-Mail: pgonzalez@arqproy

Modificar

Fig. 31: Pantalla de modificació d'usuaris

- **Esborrar Usuarios:**

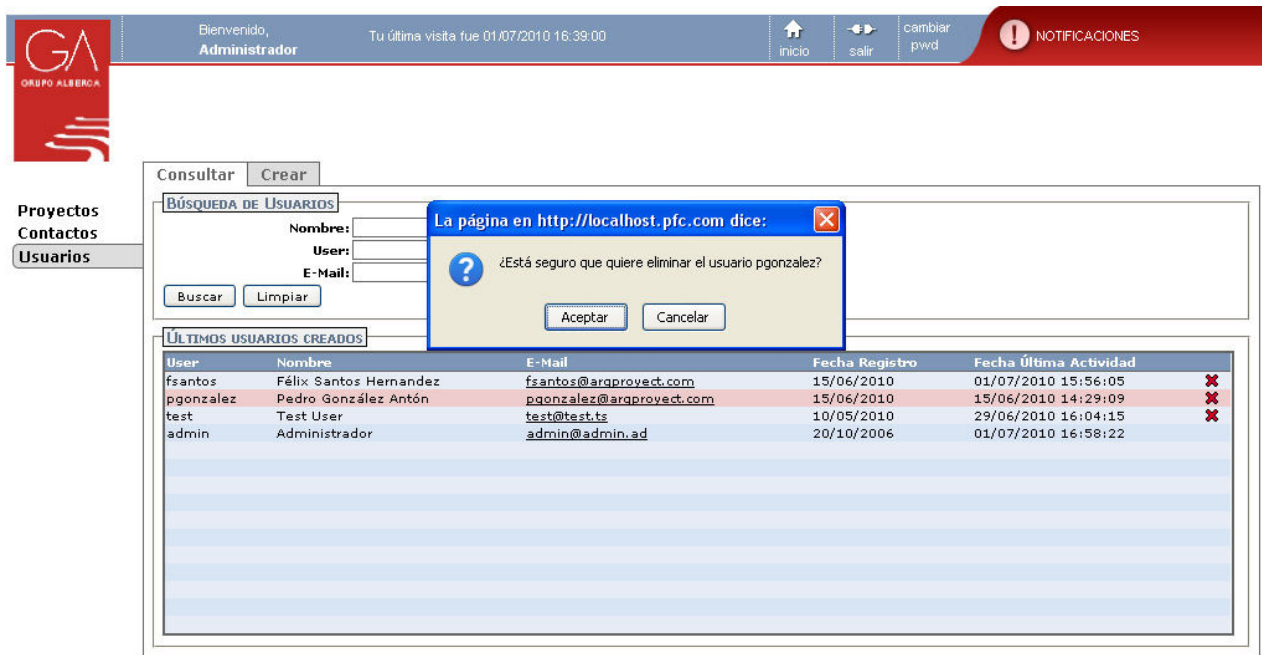


Fig. 32: Pantalla per esborrar usuaris

5.4 Disseny de la capa de negoci

La capa de negoci ens permet comunicar la presentació de les dades amb la persistència de les mateixes. S'hi realitzen tots els càlculs necessaris per tal de complir amb les funcionalitats de l'aplicació.

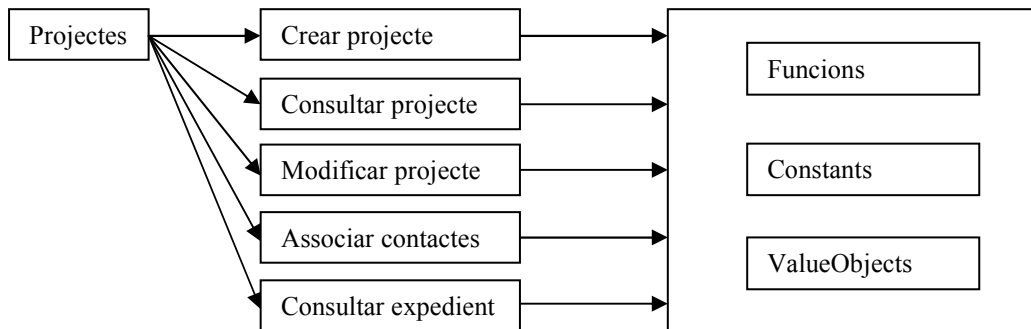
Totes les funcionalitats van agrupades en diferents fitxers i classes en funció de la temàtica a tractar. Per exemple, tots els càlculs de les funcionalitats referents als usuaris van agrupades dins del mateix conjunt de fitxers, i per evitar realitzar-ne d'una extensió innecessària, se n'han creat de diferents depenent del tipus d'operacions a realitzar.

També necessitarem uns contenidors que ens permetran transportar la informació desada a la capa de dades per tal de poder modificar-les i entregar-les a la capa de presentació. Això ho realitzarem mitjançant els *Value Objects* o altrament coneguts com *Data Transfer Objects* (objectes de transport de dades). Que no són res més que uns fitxers amb un conjunt de variables contenidores de cada un dels camps creats dins la capa de dades. D'aquesta manera,

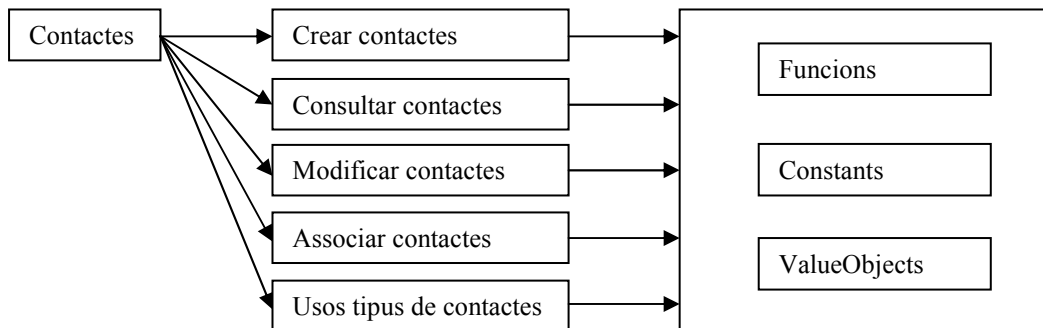
la capa de negoci, pot tractar, transformar i transportar aquestes dades abans de facilitar-les a una altra capa.

A continuació podrem veure com han estat agrupats els diferents mòduls de la capa de negoci.

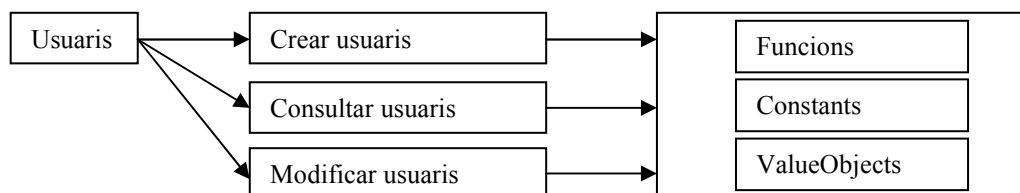
5.4.1 Mòdul de Projectes



5.4.2 Mòdul de Contactes



5.4.3 Mòdul d'Usuaris



5.5 Disseny de la capa de dades

A partir del model conceptual generat en el capítol 4.2 podem realitzar una transformació directa al seu model lògic. Un cop feta aquesta transformació, haurem de realitzar-la al model físic, que dependrà en cada cas del servidor de bases de dades utilitzat. En el nostre cas, veurem el pas al model físic per a MySQL.

5.5.1 Disseny lògic de les dades

Cada un dels objectes generats pel disseny de la base de dades tindrà un camp únic i numèric que ens permetrà identificar-lo dins del conjunt de dades. Per tal de seguir un estàndard rebrà el nom de “id_nom_de_l’objecte”.

Per tal de representar el model lògic farem servir la següent nomenclatura:

NOM DE LA TAULA: clau_primaria + (camps_que_poden_ser_NULL) + camps_que_no_poden_ser_NULL

CONTACTO: id_contacto + id_tipo_contacto + nombre + (nif) + (persona_contacto) + (cargo) + (cargo) + (direccion) + (poblacion) + (telefono_particular) + (telefono_trabajo) + (telefono_movil) + (fax_particular) + (fax_trabajo) + (cpostal) + (email) + (web_url) + (numero_colegiado) + activo + fecha_creacion + fecha_modificacion + usuario_creacion + usuario_modificacion

Per tal de poder expressar la relació n-ària entre els contactes i així poder fer el cas d'ús d'associació, hem de crear la següent taula:

CONTACTO_RELACION: id_contacto_1 + id_contacto_2 + descripcion_relacion

TIPO_CONTACTO: id_tipo_contacto + nombre + activo + protegido + fecha_creacion + fecha_modificacion + usuario_creacion + usuario_modificacion

PROYECTO: id_proyecto + ref_anyo + ref_id + (nombre) + (situacion) + (poblacion + (cpostal) + fecha_apertura_ref + caja_archivo + (id_imagen) + estado + motivo_bloqueo + fecha_creacion + fecha_modificacion + usuario_creacion + usuario_modificacion

EXPEDIENTES: id_tipo + id_proyecto + version + fecha_inicio + fecha_maxima + fecha_fin + en_espera_de + fecha_visado + fecha_ayuntamiento + exp_colegio + exp_ayuntamiento + motivo_modif + fecha_licencia + fecha_creacion + fecha_modificacion + usuario_creacion + usuario_modificacion

Per poder associar els contactes amb els projectes en una relació n-ària, hem de crear una taula apart que ho permeti:

PROYECTO_CONTACTO_RELACION: id_contacto + id_proyecto

Dins la taula usuari hem de crear un camp més que ens permeti indicar quin usuari és administrador i quins no, per això crearem un camp id_rol.

USUARIO: id_usuario + id_rol + (nombre) + nombre_usuario + password + email + fecha_registro + fecha_ultimo_acceso + fecha_ultima_actividad + activo + borrado

Quan decidim desar imatges associades a un projecte, aquestes s'emmagatzemen dins d'una carpeta del servidor i no dins de camps binaris de la base de dades. El que desarem serà un ruta relativa per tal de poder localitzar-les.

IMAGENES: id_imagen + nombre + nombre_fisico + type + size + fecha_creacion + usuario_creacion

Per tal de poder mantenir la sessió dels usuaris activa, hem de crear una taula de sessions que ens permetrà fer un seguiment dels usuaris dins de l'aplicació:

SESSION: id_session + id_usuario + ip_address + primer_acceso + ultimo_acceso

5.5.2 Disseny físic de les dades

El disseny físic de la base de dades depèn del tipus de servidor que utilitzem. Però existeix un estàndard anomenat SQL ANSI-92, encara que la revisió més recent és la del 2008 i tots els servidors el suporten.

Cada servidor de bases de dades, té la seva pròpia especialització en el disseny i estructura de les dades, però en termes generals, són molt semblants, així que la majoria de les vegades podrem portar els nostres dissenys físics entre servidors sense grans problemes.

A continuació mostrem el disseny físic per a un servidor MySQL:

Creació de la base de dades

```
CREATE DATABASE `arquitect_project` DEFAULT CHARACTER SET latin1 COLLATE
latin1_spanish_ci;
```

Creació de la taula CONTACTO

```
CREATE TABLE IF NOT EXISTS `contacto` (
  `id_contacto` int(11) NOT NULL AUTO_INCREMENT,
  `id_tipo_contacto` int(11) NOT NULL DEFAULT '0',
  `nombre` varchar(255) NOT NULL DEFAULT "",
  `nif` varchar(20) DEFAULT NULL,
  `persona_contacto` varchar(60) DEFAULT NULL,
  `cargo` varchar(20) DEFAULT NULL,
  `direccion` varchar(255) DEFAULT NULL,
  `poblacion` varchar(30) DEFAULT NULL,
  `telefono_particular` varchar(9) DEFAULT NULL,
  `telefono_trabajo` varchar(9) DEFAULT NULL,
  `telefono_movil` varchar(9) DEFAULT NULL,
  `fax_particular` varchar(9) DEFAULT NULL,
  `fax_trabajo` varchar(9) DEFAULT NULL,
  `cpostal` varchar(5) DEFAULT NULL,
  `email` varchar(30) DEFAULT NULL,
  `web_url` varchar(255) DEFAULT NULL,
  `numero_colegiado` varchar(30) DEFAULT NULL,
  `activo` char(1) NOT NULL DEFAULT 'A',
  `fecha_creacion` int(11) NOT NULL DEFAULT '0',
```

```

`fecha_modificacion` int(11) NOT NULL DEFAULT '0',
`usuario_creacion` varchar(20) NOT NULL DEFAULT "",
`usuario_modificacion` varchar(20) NOT NULL DEFAULT "",
PRIMARY KEY (`id_contacto`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

Creació de la taula CONTACTO_RELACION

```

CREATE TABLE IF NOT EXISTS `contacto_relacion` (
`id_contacto_1` int(11) NOT NULL DEFAULT '0',
`id_contacto_2` int(11) NOT NULL DEFAULT '0',
`descripcion_relacion` varchar(255) NOT NULL DEFAULT "",
PRIMARY KEY (`id_contacto_1`,`id_contacto_2`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

Creació de la taula TIPO_CONTACTO

```

CREATE TABLE IF NOT EXISTS `tipo_contacto` (
`id_tipo_contacto` int(11) NOT NULL AUTO_INCREMENT,
`nombre` varchar(100) NOT NULL DEFAULT "",
`activo` char(1) NOT NULL DEFAULT 'A',
`protegido` char(1) NOT NULL DEFAULT 'N',
`fecha_creacion` int(11) NOT NULL DEFAULT '0',
`fecha_modificacion` int(11) NOT NULL DEFAULT '0',
`usuario_creacion` varchar(20) NOT NULL DEFAULT "",
`usuario_modificacion` varchar(20) NOT NULL DEFAULT "",
PRIMARY KEY (`id_tipo_contacto`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

Creació de la taula PROYECTO

```

CREATE TABLE IF NOT EXISTS `proyecto` (
`id_proyecto` int(11) NOT NULL AUTO_INCREMENT,
`ref_anyo` varchar(4) NOT NULL DEFAULT '0',
`ref_id` char(3) NOT NULL DEFAULT '0',
`nombre` varchar(255) DEFAULT NULL,
`situacion` varchar(20) DEFAULT NULL,
`poblacion` varchar(255) DEFAULT NULL,
`cpostal` varchar(5) DEFAULT NULL,
`fecha_apertura_ref` int(11) NOT NULL DEFAULT '0',
`caja_archivo` varchar(50) NOT NULL,
`id_imagen` int(11) DEFAULT NULL,

```

```

`estado` char(2) NOT NULL DEFAULT 'EE',
`motivo_bloqueo` varchar(250) NOT NULL,
`fecha_creacion` int(11) NOT NULL DEFAULT '0',
`fecha_modificacion` int(11) NOT NULL DEFAULT '0',
`usuario_creacion` varchar(20) NOT NULL DEFAULT "",
`usuario_modificacion` varchar(20) NOT NULL DEFAULT "",
PRIMARY KEY (`id_proyecto`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

Creació de la taula EXPEDIENTES

```

CREATE TABLE IF NOT EXISTS `expedientes` (
  `id_tipo` varchar(2) NOT NULL,
  `id_proyecto` int(11) NOT NULL,
  `version` int(11) NOT NULL,
  `fecha_inicio` int(11) NOT NULL,
  `fecha_maxima` int(11) NOT NULL,
  `fecha_fin` int(11) NOT NULL,
  `en_espera_de` varchar(255) NOT NULL,
  `fecha_visado` int(11) NOT NULL,
  `fecha_ayuntamiento` int(11) NOT NULL,
  `exp_colegio` varchar(255) NOT NULL,
  `exp_ayuntamiento` varchar(255) NOT NULL,
  `motivo_modif` varchar(255) NOT NULL,
  `fecha_licencia` int(11) NOT NULL,
  `fecha_creacion` int(11) NOT NULL,
  `fecha_modificacion` int(11) NOT NULL,
  `usuario_creacion` varchar(20) NOT NULL,
  `usuario_modificacion` varchar(20) NOT NULL,
  PRIMARY KEY (`id_tipo`,`id_proyecto`,`version`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

Creació de la taula PROYECTO_CONTACTO_RELACION

```

CREATE TABLE IF NOT EXISTS `proyecto_contacto_relacion` (
  `id_contacto` int(11) NOT NULL DEFAULT '0',
  `id_proyecto` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id_contacto`,`id_proyecto`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

Creació de la taula USUARIO

```
CREATE TABLE IF NOT EXISTS `usuario` (
  `id_usuario` int(11) NOT NULL AUTO_INCREMENT,
  `id_rol` int(1) NOT NULL,
  `nombre` varchar(60) DEFAULT NULL,
  `nombre_usuario` varchar(255) NOT NULL DEFAULT "",
  `password` varchar(255) NOT NULL DEFAULT "",
  `email` varchar(60) NOT NULL DEFAULT "",
  `fecha_registro` int(11) NOT NULL DEFAULT '0',
  `fecha_ultimo_acceso` int(11) NOT NULL DEFAULT '0',
  `fecha_ultima_actividad` int(11) NOT NULL DEFAULT '0',
  `activo` char(1) NOT NULL DEFAULT 'A',
  `borrado` char(1) NOT NULL DEFAULT 'N',
  PRIMARY KEY (`id_usuario`),
  UNIQUE KEY `nombre_usuario` (`nombre_usuario`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Creació de la taula IMAGENES

```
CREATE TABLE IF NOT EXISTS `imagenes` (
  `id_imagen` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(500) NOT NULL,
  `nombre_fisico` varchar(500) NOT NULL,
  `type` varchar(150) NOT NULL,
  `size` int(11) NOT NULL,
  `fecha_creacion` int(11) NOT NULL,
  `usuario_creacion` varchar(20) NOT NULL,
  PRIMARY KEY (`id_imagen`),
  UNIQUE KEY `nombre_fisico` (`nombre_fisico`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Creació de la taula SESSION

```
CREATE TABLE IF NOT EXISTS `session` (
  `id_session` varchar(64) NOT NULL DEFAULT "",
  `id_usuario` int(11) NOT NULL DEFAULT '0',
  `ip_address` varchar(20) NOT NULL DEFAULT "",
  `primer_acceso` int(11) NOT NULL DEFAULT '0',
  `ultimo_acceso` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id_session`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

6 Implementació

En apartats anteriors hem explicat l'estructura de fitxers de l'aplicació, però encara no hem vist quina és l'estructura interna del comportament dels fluxos d'execució i perquè la hem escollida.

Abans de veure-la, mostrarem alguns aspectes importants dins del nostre codi i del codi PHP en general:

- Per escriure codi PHP, dins de fitxers de text, cal introduir-lo entre els següents símbols: `<? i ?>`
- No cal donar un valor inicial a les variables, ni tan sols inicialitzar-les amb un tipus concret (cadena de text, enter, decimal, objecte...).
- El codi cal comentar-lo de forma adequada mitjançant dos barres inclinades `//` a l'inici de cada comentari o entre els següents símbols: `/** i **/`
- A l'inici de tots els fitxers que no vulguem que siguin accedits directament, posarem un codi de seguretat que només funcionarà si és inclòs des del fitxer d'inici.

Fitxer d'inici:

```
define('_VALID_INCLUDE', TRUE);
```

Fitxer que no volem que sigui accedit directament:

```
/** SEGURIDAD **/
```

```
defined('_VALID_INCLUDE') or die('No està permès accedir directament a aquest fitxer.');
```

- Abans de realitzar qualsevol execució hem d'incloure i inicialitzar la classe mare per així tenir accés a les funcions principals, a les constants i l'accés a la base de dades.

```
/** INCLUDES **/
```

```
include_once("admin/includes/boffice.php");
```

```
/** SEGURETAT I INICIALITZACIÓ**/
```

```
$web = new Alberca();
```

```
$web->iniciarSession();
```

- La classe mare conté les variables que permetran posar l'aplicació en el mode de proves, que ens permetrà visualitzar els errors d'una forma més detallada directament per pantalla.

```

if($this->debug == true)
    error_reporting(E_ALL ^ E_NOTICE);
else
    error_reporting(E_ERROR);

```

- La classe d'accés a la base de dades disposa de la variable que ens indica l'entorn de treball i així poder tenir diferents entorns pre configurats.

```

// ##### VARIABLES D'ENTORN #####
if($this->entorn == "P") { //PRODUCCIÓ
    $this->servidorBD = "localhost";
    $this->usuariBD = "arquiprojectUser";
    $this->contrasenyaDB = "arquiprojectPass";
    $this->nomBD = "arquiproject";
}
elseif($this->entorn == "D") { //DESENVOLUPAMENT
    $this->servidorBD = "localhost";
    $this->usuariBD = "root";
    $this->contrasenyaDB = "";
    $this->nomBD = "arquitect_test";
}

```

- La classe d'accés a la base de dades disposa d'una variable que ens permetrà donar un alias a les diferents taules de la base de dades i així poder disposar de varies aplicacions per empreses diferents en un mateix servidor.

```

// ##### ALIES DE TAULES #####
var $taula = array( 'contacto' => "nombre_empresa_contacto",
    'contacto_relacion' => "nombre_empresa_contacto_relacion",
    'imagenes' => "nombre_empresa_imagenes",
    'proyecto' => "nombre_empresa_proyecto",
    'relacion_proyecto_contacto' => "nombre_empresa_relacion_proyecto_contacto",
    'session' => "nombre_empresa_session",
    'tipo_contacto' => "nombre_empresa_tipo_contacto",
    'usuario' => "nombre_empresa_usuario",
    'expedientes' => "nombre_empresa_expedientes"
);

```

6.1 Esquema del funcionament

Per tal de facilitar la comprensió del codi i conseqüentment facilitar els futurs manteniments, tots els casos d'ús de l'aplicació tenen un mateix punt d'entrada, el fitxer *index.php*, que s'encarregarà de realitzar les inicialitzacions bàsiques i, sobretot, de cridar el distribuïdor de casos als fitxers que els implementen.

Com a excepció, ens trobem amb els pop-up's, que segueixen una estructura web diferent a la pantalla d'inici per tal de fer-les cabre dins de finestres més petites; i els casos d'ús que requereixen un canvi a les cookies i que impliquen un canvi immediat per pantalla, com és l'inici de sessió d'usuari i el final de sessió.

A més de les inicialitzacions bàsiques, el fitxer *index.php* s'encarrega d'incloure tots els elements comuns de l'execució, com és l'estructura web de l'aplicació, com la capçalera, el menú i el peu de pàgina. Després d'incloure totes les parts comunes, només ens queda incloure el distribuïdor de fluxos, que és un fitxer encarregat de realitzar les crides als diferents fitxers que implementen cada un dels casos d'ús, d'aquesta manera, podem afegir o eliminar nous casos d'ús senzillament, fins i tot podrem controlar els casos d'ús no implementats i mostrar-ho per pantalla.

A la figura 33 podem veure un esquema de quin seria el funcionament descrit:

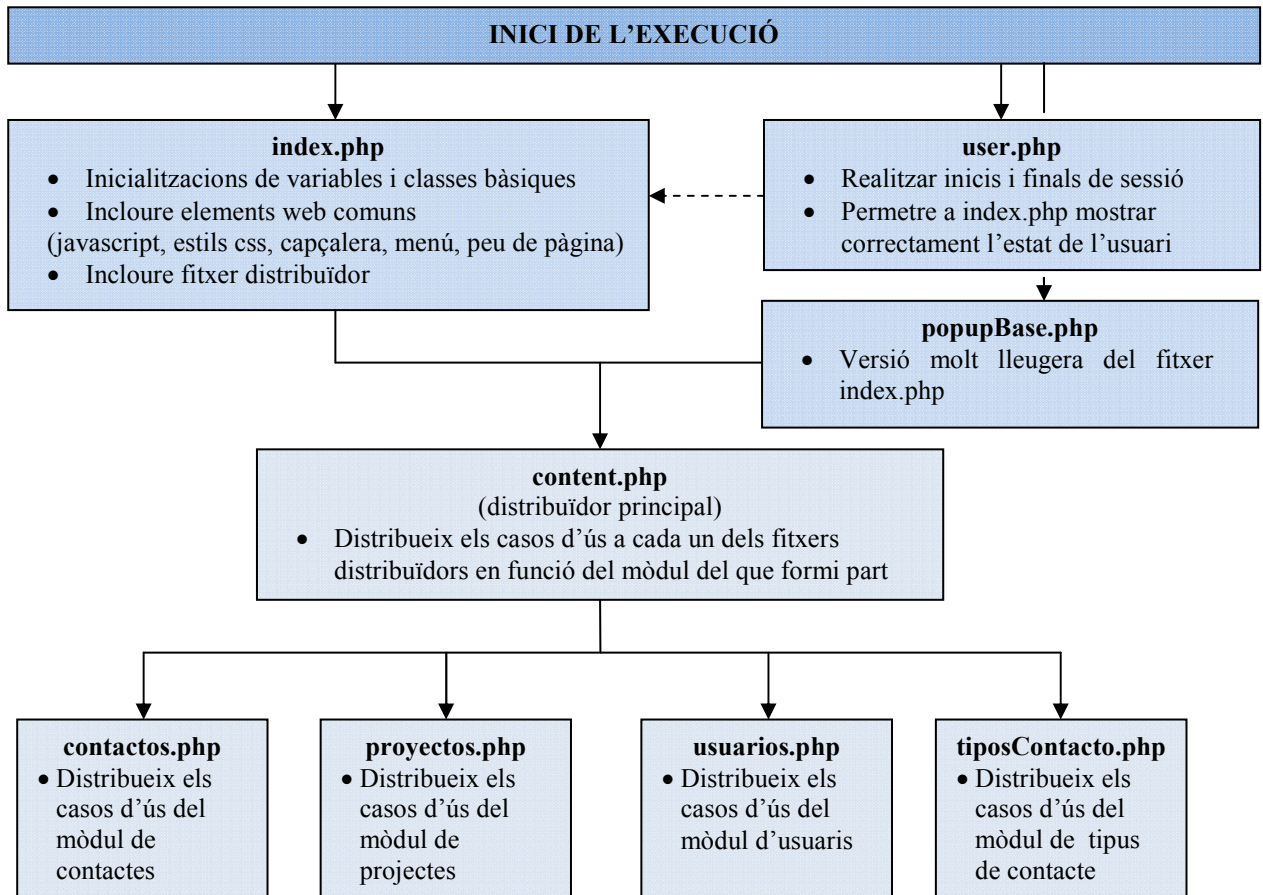


Fig. 33: Esquema del funcionament intern de l'aplicació

6.2 Seguretat

Tota aplicació que interactuï amb l'usuari ha de tenir dos sistemes de seguretat: el d'entrada errònia de dades i el de l'ús malintencionat que pugui donar-li l'usuari.

Per evitar l'entrada errònia de les dades, tots els nostres formularis tenen comprovacions perquè els tipus de dades demanats coincideixin amb els introduïts pels usuaris. També es comprova la coherència de les dades introduïdes, com podria ser que una data fi fóra menor a una data inici, i així evitar possibles errors d'introducció.

Per tal d'evitar l'ús malintencionat hem d'incloure elements de seguretat que, entre d'altres, no permetin l'ús de l'aplicació a cap usuari no identificat, no permetin accedir a cap dada d'ús

privat i no permeti realitzar cap acció dins l'aplicació per la qual no hagi estat creada, evitant, així, accedir a fitxers no desitjats o fer ús de *SQL Injection*¹³.

6.2.1 Autenticació

L'aplicació no pot ser utilitzada per usuaris no identificats de forma correcta pel sistema. Per tant, hem necessitat crear una pàgina per demanar l'accés a l'usuari mitjançant una contrasenya. Aquesta pàgina es mostrarà sempre que l'usuari no hagi realitzat l'accés correctament.

Inicialment, l'aplicació només tindrà un únic usuari, l'Administrador. A través d'aquest usuari, es podran anar donant d'alta d'altres usuaris que ja podran iniciar sessió a mida que vagin sent creats.

A continuació podem veure la pantalla d'inici de sessió:

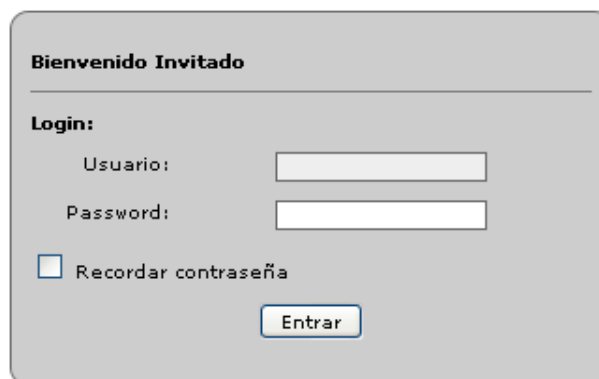
A screenshot of a web application's login page. The page has a light gray background and a rounded rectangular border. At the top left, the text "Bienvenido Invitado" is displayed in a bold, black font. Below this, the word "Login:" is written in a smaller, bold font. There are two input fields: the first is labeled "Usuario:" and the second is labeled "Password:". Below the password field, there is a checkbox labeled "Recordar contraseña". At the bottom center of the form, there is a button labeled "Entrar".

Fig. 34: Pantalla d'inici de sessió

¹³ SQL Injection: s'anomena així al procés d'introduir codi SQL dins d'un altre codi SQL per tal d'alterar el comportament d'aquest últim.

6.2.2 Encriptació de les contrasenyes

És important que les contrasenyes dels usuaris no puguin ser accedides per cap persona no autoritzada. De fet, ni tant sols els usuaris administradors haurien de tenir accés a aquesta informació. Si un administrador necessita modificar alguna dada d'un usuari de l'aplicació ho hauria de poder fer mitjançant alguna eina d'administració via web o accedint directament al contingut de la base de dades, però en cap cas necessitarà mai la contrasenya de l'usuari per realitzar modificacions. Molts usuaris utilitzen una mateixa paraula clau per a diferents aplicacions, així, si en sabéssim una, probablement sabríem la de la majoria d'aplicacions que utilitzés.

És per aquesta raó que hem necessitat encriptar les paraules clau d'una manera que no puguin ser accedides directament. Cal tenir en compte que no podem fer servir mètodes desencriptables com podria ser el sistema Base64, que fa una simple transformació del text introduït i que després es pot desfer.

Hem utilitzat l'encriptació anomenada MD5, que no pot ser desencriptada i, d'aquesta manera, encara que algun usuari malintencionat aconseguís fer-se amb les contrasenyes, no les podria utilitzar per accedir a l'aplicació.

L'encriptació MD5 és un algoritme de codificació de 128 bits, però és representat típicament per un número de 32 dígit hexadecimals (1 dígit hexadecimal equival a 4 bits). Veiem el valor MD5 d'uns exemples:

Contrasenya de l'usuari	Contrasenya encriptada amb MD5 (valor desat a la base de dades)
letmein	0d107d09f5bbe40cade3de5c71e9e9b7
contrasenya_muy_larga_no_afecta	dcfc1ea03356b5b993be1e38a099e406
(contrasenya buida)	d41d8cd98f00b204e9800998ecf8427e

6.2.3 Seguretat a les pàgines

Per mantenir la seguretat de l'aplicació, hem d'evitar que els usuaris malintencionats puguin accedir als fitxers que no han estat pensats per ser utilitzats directament i a l'accés o modificació del contingut de la base de dades.

En apartats anteriors hem vist que l'aplicació només té tres fitxers com a punt d'entrada a l'aplicació: *index.php*, *popupBase.php* i *user.php*. Llavors, si l'aplicació està pensada per tenir només aquests punts d'entrada, per què hauríem de permetre l'accés a la resta de fitxers PHP? Tots els altres fitxers estan pensats per ser inclosos des d'algun dels fitxers d'entrada, per això, mostrarem un error en el cas que no sigui així.

Ja ho hem vist resumit en forma de punt dins de l'apartat 6. Per fer-ho, els fitxers d'entrada hauran d'inicialitzar una variable global, que hem anomenat *_VALID_INCLUDE*, a certa. Així, quan incloem un dels fitxers que no volem que siguin accedits directament, només haurem de consultar si la variable global està inicialitzada a *CERT* i continuar amb l'execució. En cas que no estigui inicialitzada a *CERT* significarà que no s'està utilitzant cap fitxer definit com d'entrada i, per tant, hem de mostrar un error.

Definim en PHP la variable global del fitxer d'entrada:

```
define('_VALID_INCLUDE', TRUE);
```

Definim en PHP el codi que han d'incloure els fitxers que no volem que siguin accedits directament:

```
defined('_VALID_INCLUDE') or die('No està permès accedir directament a aquest fitxer.');
```

Per altra banda, també hem d'evitar que els usuaris malintencionats puguin accedir o modificar el contingut de la base de dades sense el nostre permís, així que haurem d'evitar la utilització de SQL Injection.

Per fer-ho, hem evitat la utilització de comodins * a la hora de realitzar consultes, així evitem que es puguin extreure camps que no vagin a ser utilitzats. Les consultes *INSERT* tampoc s'han utilitzat de forma genèrica, així només desarem els camps que nosaltres marquem.

També es tracten totes les dades introduïdes pels usuaris que han de ser incloses dins de consultes SQL amb la funció *SQLprepare*, que no és més que una funció creada per nosaltres per filtrar tots els textos de les consultes SQL que puguin ser susceptibles a ser atacades:

```
function SQLprepare ($string) {  
    if (!get_magic_quotes_gpc())  
        $string = addslashes($string);  
    else  
        $string = $string;  
  
    return $string;  
}
```

Com que hem creat una funció especialment per tractar l'entrada de dades destinades a consultes SQL, podem ampliar-la o modificar-la en qualsevol moment per que afecti a totes les consultes SQL a la vegada.

Per últim, totes les consultes *UPDATE* que s'espera que només modifiquin un sol registre, porten *LIMIT 1* al seu final. D'aquesta manera forcem que només es modifiqui un registre a la vegada encara que algú aconsegueixi realitzar SQL Injection i així aconseguim minimitzar els possibles danys.

7 Proves

En el capítol 3 hem explicat que la implementació dels casos d'ús ha estat dividida en dues parts per tal d'involucrar el client en el desenvolupament de l'aplicació. D'aquesta manera, mentre l'usuari ja disposava de la primera part per poder-la utilitzar i provar, nosaltres ja realitzàvem la segona part de la implementació.

Aquest mètode ha resultat ser força eficaç, ja que tal com esperàvem, els errors reportats pel client han permès no repetir-los en la segona etapa de la implementació i estalviar-nos canvis i modificacions en una etapa més complexa de l'aplicació.

Seguint aquesta estratègia de divisió per etapes, ens ha resultat més senzill solucionar els errors trobats, perquè interferien amb un volum de codi inferior. A més, el client ha pogut disposar d'una part de l'aplicació abans de la finalització d'aquesta i així hem evitat que tingués una sensació de *no avançar* en el desenvolupament.

Finalment, al realitzar l'entrega de la segona etapa de la implementació no hem hagut de solucionar tants errors, ja que l'aplicació ja havia estat provada i millorada en la primera etapa i només vam haver de solucionar errors senzills que no afectaven a l'estructura interna de l'aplicació.

En tot moment, el client ha demostrat estar satisfet amb aquest sistema i a nosaltres ens ha permès treballar d'una manera més dinàmica i interactiva amb ell, permetent un intercanvi d'opinions més constant.

8 Manteniment

Després d'haver realitzat la posada en marxa de la segona etapa del projecte, el client s'ha trobat amb la necessitat de realitzar canvis dins de l'aplicació que, per ara, no es poden realitzar a través de l'entorn web i que no havien estat pactats a l'inici del projecte.

Hem pactat entre nosaltres un servei de manteniment que ens permetrà solucionar els petits dubtes o canvis directes a la base de dades a canvi d'un petit esforç econòmic revisat mensualment, en funció del número d'hores destinades a la solució d'aquests dubtes i problemes.

A més, al nostre client, a mida que ha estat utilitzant l'aplicació, ha tingut la necessitat d'ampliar i millorar els mòduls existents i fins i tot de realitzar-ne de nous. En el futur, realitzarem aquestes modificacions seguint una estructura semblant a la seguida per aquest projecte, dividint-les en etapes.

9 Conclusions

A continuació veurem les desviacions dins la planificació inicial, quina repercussió ha tingut en el cost econòmic i temporal del projecte i si aquests canvis han perjudicat al projecte.

Després, analitzarem com hem integrat i implantat el nostre projecte amb l'estudi d'arquitectura per tal que puguin utilitzar-lo i veurem si s'han acomplert els objectius.

I per últim, veurem quin serà el futur de la nostra aplicació, quines millores i ampliacions es podran realitzar i valorarem, de forma personal, el projecte.

9.1 Anàlisi de desviacions

Finalment, la planificació horària realitzada a l'inici del projecte va ser lleugerament diferent i un mica més curta, és a dir, hem tardat més temps en finalitzar el projecte:

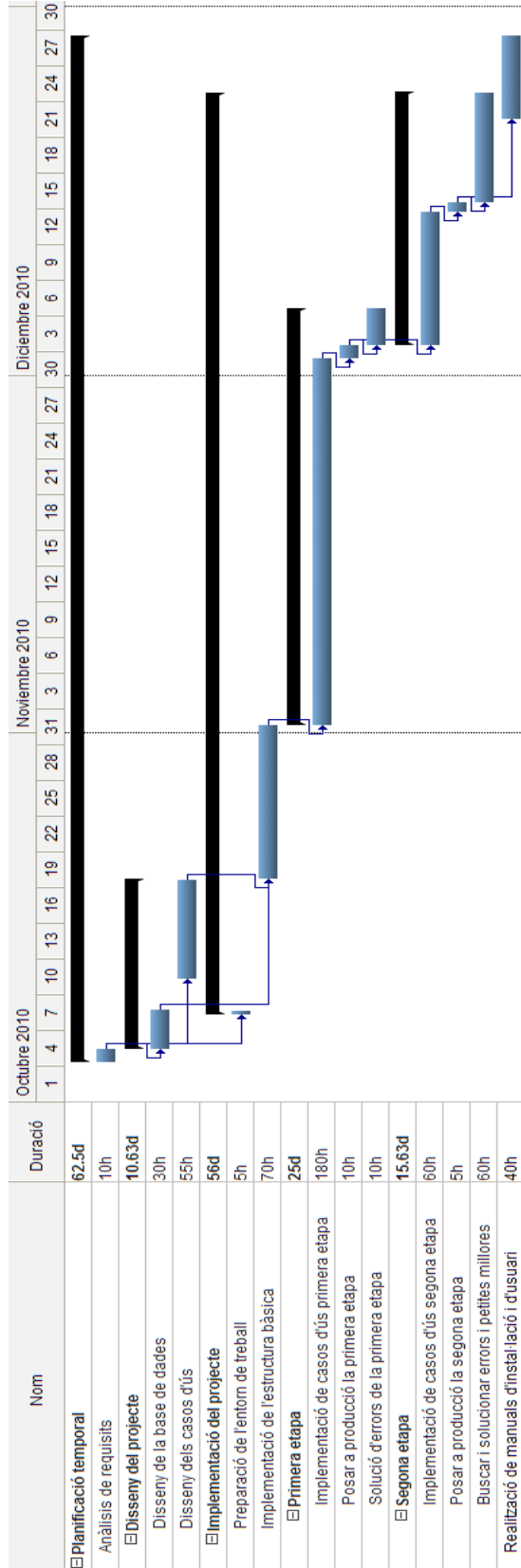
- El disseny del conjunt de casos d'ús ha resultat ser més senzill i ens ha ocupat un total de 55 hores i no pas 75 com havíem estimat. Això ha estat degut a que alguns casos d'ús tenien una estructura semblant i han pogut ser reaprofitats.
- L'entorn de treball ens ha ocupat 5 hores, la meitat del temps que havíem estimat. La utilització d'aplicacions que ja ens eren familiars ens ha facilitat les tasques i configuracions a preparar.
- Per tal d'aconseguir una estructura bàsica robusta, que ens facilités el manteniment i el desenvolupament de l'aplicació, hem dedicat més temps i esforços a la seva implementació. En total unes 70 hores.
- La implementació dels casos d'ús de la primera etapa ha estat més llarga del desitjable, un total de 180 hores, però ha estat compensat amb un codi de més qualitat, menys errors i de manteniment més senzill.
- Els errors trobats durant la primera etapa no han estat tants com podíem esperar i per això no hem dedicat tant de temps a resoldre'ls. N'hi hem dedicat al voltant de 10 hores.

- Per implementar els casos d'ús de la segona etapa no ens han fet falta el total de 70 hores planificades, amb poc més de 60 hem pogut finalitzar-los i així poder dedicar més temps a la cerca d'errors.
- Havíem de dedicar un total de 40 hores a la cerca i solució d'errors i al desenvolupament de noves, però petites, millores. Finalment, hi hem dedicat poc més de 60 hores, ja que les petites millores que ens ha demanat el client eren força necessàries i útils per l'aplicatiu, a més de millorar la satisfacció final del client.

La resta de tasques del projecte han tingut uns temps molt ajustats a la planificació.

9.2 Cost del projecte

A continuació veiem el diagrama de Gantt que a resultat finalment:



9.2.1 Cost de recursos humans i materials

Com ja comentàvem al capítol 3.2, el projecte ha estat realitzat per una sola persona, però per tal de poder-ne fer un estudi econòmic, l'hem dividit entre diferents rols típics dins dels sectors del desenvolupament informàtic.

Sabent les diferències de temps que hem experimentat, podem calcular el cost real que ha tingut el nostre projecte:

Càrrec (Sou brut anual)	Preu / hora	Hores	Cost
Cap de projecte (45.000 € bruts)	32,5 €/h	60	1.950 €
Analista funcional (35.000 € bruts)	25,0 €/h	135	3.375 €
Arquitecte (40.000 € bruts)	29,0 €/h	90	2.610 €
Analista-programador (26.000 € bruts)	18,5 €/h	110	2.035 €
Programador (20.000 € bruts)	14,5 €/h	200	2.900 €
TOTAL		595	12.870 €

Segons la planificació inicial, el cost estimat del projecte era de 12.420 € i finalment ens ha costat 12.870 €, uns 450 € per sobre del cost estimat.

Si fem referència al cost econòmic dels recursos humans, podem dir que el projecte ha estat un 3,6% més car del previst. I en cost temporal, podem dir que el projecte ha estat un 4,4% més llarg.

Com que el projecte ha durat 25 hores més del que havíem planificat, em de tornar a calcular els costos repercutits al projecte de totes les despeses associades al local:

Concepte	Cost mensual	Cost anual	Cost repercutit al projecte
Lloguer del local	1.200 €	14.400 €	4.760 €
Assegurança local	30 €	360 €	119 €
Amortització del equips informàtics	300 €	3.600 €	1.190 €
Electricitat	150 €	1.800 €	595 €
Aigua	25 €	300 €	99 €
Telèfon + Internet	120 €	1.440 €	476 €
Neteja	80 €	960 €	318 €
TOTAL	1.905 €	22.860 €	7.557 €

Els costos repercutits s'han vist incrementats en 318 €, un 4,4% més del que havíem previst.

Concepte	Cost estimat	Increment	Cost real
Cost dels recursos humans	12.420 €	+ 450 €	12.870 €
Cost del material software	0 €	+ 0 €	0 €
Cost d'altres conceptes	7.239 €	+ 318 €	7.557 €
TOTAL	19.659 €	+ 768 €	20.427 €

L'increment que vam fer del 20% al cost final del projecte ens ha permès absorbir aquest petit increment en el cost final i hem aconseguit realitzar un projecte rentable econòmicament.

9.2.2 Cost d'instal·lació i implantació del sistema

Per que el client pugui començar a utilitzar la nostra aplicació, haurà de realitzar una migració de tots els seus contactes cap a dins del mòdul de contactes. En el cas del nostre client, com que no té una base de dades extremadament extensa, hem realitzat nosaltres mateixos la migració de forma manual, dedicant-hi un total de 3 hores, que ja havíem quantificat dins les

10 hores necessàries per posar a producció la primera etapa. Per a clients amb bases de dades extenses, hauríem de quantificar el temps de migració a part o realitzar un sistema que facilités la importació d'aquests contactes.

En el cas dels projectes, el més senzill per tal de no alterar l'ordre del dia a dia de l'estudi d'arquitectura, és realitzar tots els nous projectes directament des de l'aplicació i finalitzar els ja iniciats amb el mètode tradicional. Encara que, si el client ho desitja, ens podrien facilitar un llistat de projectes i els seus estats per tal d'incorporar-los dins de l'aplicació. Com que per a nosaltres té un cost relativament baix, però és el client el que ens ha de recollir i facilitar les dades del projecte, i aquests canvien contínuament, no han decidit migrar els projectes antics dins de l'aplicació.

9.3 Objectius finals

Un cop finalitzat el projecte, podem dir que hem aconseguit complir pràcticament tots els objectius que ens proposava el nostre client i tots els objectius per aconseguir el correcte desenvolupament del projecte.

El client ens sol·licitava una aplicació que els permetés millorar el rendiment diari del seu estudi. Amb l'aplicació hem aconseguit centralitzar tota la informació que fins ara, s'havien de comunicar mitjançant post-it's o correus electrònics. A través del seguiment d'expedients i de projectes hem aconseguit que tots els treballadors de l'estudi puguin consultar l'estat concret de cada projecte d'obra, així, mentre els responsables de cada projecte vagin actualitzant la informació del seguiment d'expedients, la resta de companys poden saber exactament l'estat en que es troba en tot moment i si està pendent d'algun document o tràmit.

També volien una agenda relacional per tal d'evitar fer ús de taules d'Excel com a agenda, ja que no els permetia una relació entre els contactes i tenien greus problemes per disposar sempre de l'última versió. A través de l'agenda que hem dissenyat, tots els treballadors de l'oficina podran accedir a la versió més recent dels contactes i mantenir-la per tal que la resta de treballadors la puguin consultar.

Per altra banda, també s'han acomplert els passos a seguir per a posar en marxa el projecte:

- Hem recollit la informació proporcionada per l'estudi d'arquitectura en forma d'actes.
- Hem estimat el temps i cost del projecte per tal de fer una proposta econòmica a l'estudi.
- Hem preparat un entorn local de proves i desenvolupament per tal de realitzar el projecte.
- Hem dissenyat la base de dades i cada un dels casos d'ús extrets de les actes.
- Hem implementat una estructura bàsica de codi PHP que ens servirà com a base per a altres projectes.
- Hem implementat una primera etapa de casos d'ús.
- Hem posat a producció aquesta primera etapa.
- Hem solucionat els problemes reportats per l'estudi d'arquitectura.
- Hem implementat la resta de casos d'ús en una segona etapa.
- Hem posat a producció aquesta segona etapa més la solució dels problemes reportats.
- Hem solucionat els pocs problemes que ha trobat el client en aquesta segona entrega.
- Hem realitzat un petit manual d'instal·lació i d'usuari.

9.4 Millores i ampliacions

L'estudi d'arquitectura necessitava organitzar tota la documentació important de cada projecte, així com les fotografies realitzades, conegut per ells com "seguiment d'obra". Es va decidir deixar fora del projecte inicial la organització d'aquesta documentació a través de l'aplicatiu, perquè actualment els sembla correcte com ho organitzen, en carpetes compartides d'un servidor que reben el nom que rep el projecte dins de l'aplicatiu: YY/XXX – On YY fa menció a l'any del projecte, 08 significaria 2008, i XXX es referirà al número de projecte dins de l'any. Però ha estat proposat com una futura ampliació del projecte. De fet, l'aplicació ja permet pujar una fotografia per identificar el projecte d'obra, així que realment només caldria ampliar aquesta secció per tal d'aconseguir organitzar la documentació i les fotografies a través de l'aplicació. Hem creat una pestanya nova dins la secció de "Projectes" anomenada "Seguiment d'Obra" on anirà ubicada aquesta organització de documents en futures ampliacions.

Recentment, s'ha arribat a l'acord per a realitzar un mòdul de notificacions que permetrà als usuaris de l'aplicació associar-se projectes per anar rebent missatges sempre que tinguin canvis associats. D'aquesta manera, no hauran de consultar projecte per projecte per saber si han sigut modificats per algun company de l'estudi. De fet, aquesta millora ja s'està realitzant i, encara que quedi fóra del projecte, hem volgut mostrar com quedaria a les captures de pantalla que hem mostrat en aquest document.

Per tal de poder millorar el projecte i així poder convertir-lo en una eina comerciable entre d'altres estudis d'arquitectura, seria convenient afegir alguna eina que permetés discutir els aspectes de cada projecte, un mini fòrum o zona de xat permanent. Així, fins i tot, es podria donar accés a companys fóra de l'oficina que no haurien de planificar reunions per debatre coses senzilles que no requereixen d'una resposta urgent.

A més, si volem comerciar l'aplicació amb estudis més grans, o a grups d'estudis, cal afegir un sistema de permisos, perquè els usuaris de la pàgina no puguin consultar no més que els projectes als quals estiguin assignats. De fet, no seria molt complicat d'incloure, ja que l'aplicació ja disposa d'un sistema de rols per tal d'indicar quins usuaris són administradors, només caldria ampliar-lo.

Per altra banda, i per tal d'estalviar-nos treball en un futur, podem incloure un sistema de plantilles, perquè la capa de presentació sigui fàcilment modificable. En el cas que un estudi d'arquitectura ens faciliti uns dissenys a seguir, serien més fàcils d'incorporar dins de l'aplicatiu.

També es poden implementar mòduls de traducció de textos que ens permetran disposar de l'aplicació web en diferents idiomes. El que ens permetria incorporar la nostra aplicació en estudis d'arquitectura amb treballadors de diferents llengües.

9.5 Futur de l'aplicació web

Després d'utilitzar l'aplicació durant uns mesos, el nostre client ha mostrat la seva satisfacció amb el projecte i la seva eficàcia, no deixant-la en desús i traient-ne un rendiment diari. És per

això que es pot plantejar l'ús comercial de l'aplicatiu, sempre que es realitzin les millores suficients per adaptar-lo a les petites diferències entre estudis d'arquitectura.

L'aplicació es pot adaptar senzillament a l'entorn gràfic corporatiu, i fent els ajustaments pels requisits específics per cada estudi d'arquitectura, podrem aconseguir una plataforma adaptativa i si a més, incorporem un sistema de plantilles i de localització, podrem crear diferents entorns gràfics i en diferents idiomes, el que ens permetrà un sector comercial més ampli.

Com ja comentàvem en l'apartat anterior, si realment es vol comerciar l'aplicació a diferents estudis d'arquitectura, caldrà incorporar elements atractius, com el fòrum per debatre projectes, el sistema de permisos, evolució fotogràfica del projecte, notificació de permisos pendents d'un projecte, entre d'altres.

9.6 Valoració personal

La realització d'aquest projecte m'ha permès assentar i posar en pràctica tots els coneixements adquirits durant la universitat i l'entorn empresarial post universitari.

La part de la implementació no ha estat especialment complicada, ja que les pràctiques, el treball com a becari i la facilitat d'aprenentatge m'han permès desenvolupar-me correctament.

L'etapa de disseny sí que ha estat una mica més complicada, ja que no havia tingut oportunitat d'aplicar-ho de forma molt extensa dins del món laboral, encara que posteriorment m'ha permès demostrar tenir experiència en la realització de dissenys.

La part més complicada ha estat la realització de la memòria, ja que es necessita molt de temps per poder redactar-la i les circumstàncies personals no m'han permès dedicar-li tot el necessari, per aquesta raó, no he pogut evitar allargar-la en el temps.

El que més m'ha omplert i aportat d'aquest projecte, ha estat realitzar la seva estructura bàsica. Aquesta estructura inicial ha estat utilitzada en d'altres projectes personals i ha anat evolucionant fins convertir-se en una eina potent que permet ser la base de molts tipus

d'aplicacions diferents. Amb satisfacció, puc dir que va ser la base del portal web necessari per l'empresa que vaig crear i que m'ha absorbit tot el temps lliure d'aquests últims anys. Així doncs, puc dir que gràcies a aquest projecte he pogut reunir el valor necessari per constituir una empresa i tirar-la endavant durant el llarg de 2 anys.

10 Bibliografia

- PHP - <http://www.php.net/manual/es/>

Manual oficial del llenguatge de programació PHP. Consultat durant la programació per tal d'utilitzar el llenguatge de forma correcta.

- PHP - <http://www.desarrolloweb.com/php/>

Manuais i tècniques de programació per a PHP. Utilitzat com a manual de consulta sobre procediments i tècniques en PHP.

- MySQL - <http://dev.mysql.com/doc/refman/5.0/es/>

Manual oficial de referència de MySQL. Utilitzat com a manual de consulta per realitzar consultes SQL.

- JavaScript - <http://www.w3schools.com/js/>

Manuais per a JavaScript. Utilitzat com a referència per una programació estàndard i d'acord amb la normativa vigent.

- CSS - <http://www.w3schools.com/css/>

Manuais per a CSS. Utilitzat com a referència per una programació estàndard i d'acord amb la normativa vigent.

- Diagrames de Gantt - <http://gantter.com/>

Pàgina web que permet realitzar diagrames de Gantt de forma totalment en línia i sense cap tipus d'instal·lació. Utilitzat pels diagrames de les planificacions temporals.

- Varis - <http://es.wikipedia.org/>

Enciclopèdia en línia de contingut lliure. Utilitzada com a referència per la terminologia tècnica.

- Projectes fi de carrera:
 - José Luis Ojeda Martín. "Aplicació web per a la gestió d'un ajuntament". Barcelona: UPC. Facultat d'Informàtica de Barcelona. 2009.
 - Jose Miguel García Acosta i Pere Joan Ullate Valls. "Aplicació web per gestionar notes escolars". Barcelona: UPC. Facultat d'Informàtica de Barcelona. 2008.
 - Joaquín Val Ríos. "Aplicación web para grupo de escalada". Barcelona: UPC. Facultat d'Informàtica de Barcelona. 2008.
 - Neus Català Roig. "Desarrollo de una tienda electrónica". Barcelona: UPC. Facultat d'Informàtica de Barcelona. 2009.
- Llibres:
 - Juan Diego Gutierrez Gallardo. "Desarrollo web con PHP5 y MySQL (Guías Prácticas)". 2004.