



**Universitat Autònoma
de Barcelona**

CONTROL D'INVENTARIS MITJANÇANT CODIS DE BARRES

Memòria del projecte
d'Enginyeria Tècnica en
Informàtica de Sistemes

realitzat per

Sergi González Pérez

i dirigit per

Jordi Pons Aróztegui

Escola d'Enginyeria

Sabadell, Setembre de 2010

El sotasignat, **Jordi Pons Aróztegui**,
professor de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el treball al que correspon la present memòria
ha estat realitzat sota la seva direcció
per en **Sergi González Pérez**
I per a que consti firma la present.
Sabadell, setembre de 2010.

Signat: **Jordi Pons Aróztegui**.

En aquest projecte es desenvolupa un software per a la gestió d'inventaris d'actius de la Universitat Autònoma de Barcelona. El hardware pel qual està dissenyada aquesta aplicació és un *notebook* i una pistola lectora de codis de barres.

Amb aquest software es poden crear inventaris per centres i espais important la informació mitjançant un fitxer extern. L'aplicació detecta automàticament canvis d'ubicació, elements perduts i permet inserir comentaris per a la identificació d'elements en mal estat.

L'aplicació està basada en el web i pel seu desenvolupament s'han fet utilitzat tecnologies com el PHP, SQL, XHTML, CSS, Javascript, servidors Apache i MySQL per a l'execució de l'aplicació.

Amb l'ús d'aquest software permetrem portar un control exhaustiu dels actius de la Universitat i millorar el servei optimitzant el control del material.

Índex

<u>CAPÍTOL 1: INTRODUCCIÓ</u>	<u>1</u>
1.1 Pròleg	1
1.2 Situació actual	1
1.3 Objectius del projecte	2
1.4 Motivacions personals	2
1.5 Contingut de la memòria	3
<u>CAPÍTOL 2: ESTUDI DE VIABILITAT</u>	<u>5</u>
2.1 Introducció	5
2.1.1 Tipologia i paraules clau	5
2.1.2 Descripció	5
2.2 Descripció i priorització d'objectius	5
2.3 Parts interessades	6
2.4 Equip de projecte	6
2.5 Normativa	7
2.6 Producte i documentació del projecte	7
2.7 Context de la situació actual	7
2.8 Alternatives	8
2.8.1 Descripció d'alternatives	8
2.8.1.1 Alternativa 1: ERP	8
2.8.1.2 Alternativa 2: GDS Bicicletaria	8
2.8.1.3 Alternativa 3: Aplicació pròpia	9
2.8.2 Comparació d'alternatives	9
2.8.3 Solució proposada i justificació	9
2.9 Planificació	9

2.9.1 Descripció	9
2.9.2 Recursos humans	10
2.9.3 Tasques del projecte	10
2.9.4 Planificació temporal.....	12
2.10 Riscos	13
2.10.1 Catalogació i priorització.....	13
2.10.2 Pla de contingència	14
2.11 Pressupost	15
2.11.1 Estimació cost personal.....	15
2.11.2 Estimació cost dels recursos	15
2.11.3 Resum i anàlisi cost - benefici	16
2.12 Conclusions.....	16
2.12.1 Beneficis	16
2.12.2 Inconvenients	16
 <u>CAPÍTOL 3: ANÀLISI</u>	 <u>17</u>
3.1 Perfils d'usuari	17
3.2 Requisits funcionals	17
3.3 Requisits no funcionals.....	19
3.3.1 Restriccions de disseny.....	19
3.3.2 Objectius de Disseny	19
3.3.3 Requisits sobre les interfícies externes.....	20
 <u>CAPÍTOL 4: DISSENY.....</u>	 <u>21</u>
4.1 Introducció	21
4.1.1 Selecció del llenguatge de programació i de la base de dades	21
4.1.2 Selecció entorn de desenvolupament.....	23

4.2 Base de dades	25
4.2.1 Estructura de la base de dades	26
4.2.2 Especificació de taules i explicació de camps.....	27
4.3 Diagrames de casos d'ús.....	31
4.4 Diagrames de seqüència.....	34
4.5 Disseny d'interfícies	37
4.5.1 Introducció	37
4.5.2 Esbossos d'interfícies	38
4.6 Especificació de mòduls.....	42
4.6.1 import.....	42
4.6.2 list	43
4.6.3 inventory	43
4.6.4 login	43
4.6.5 admincp.....	43
4.6.6 export	43
4.6.7 incidence	44
4.6.8 welcome	44
4.6.9 nav	44

CAPÍTOL 5: CODIFICACIÓ

5.1 Patró de desenvolupament: Model Vista Controlador	45
5.1.1 Introducció	45
5.1.2 Model	46
5.1.3 Vista.....	48
5.1.4 Controlador	48
5.2 Estructura de directoris	50

5.3 Tecnologies utilitzades	51
5.3.1 PHP	51
5.3.2 XHTML	51
5.3.3 CSS	51
5.3.4 Javascript	51
5.3.5 AJAX i jQuery	51
5.4 Aspectes interessants de la codificació	52
5.4.1 Variables de sessió	52
5.4.2 Autoload	53
5.4.3 Autoemplenat de formularis	53
5.5 Configuració de l'aplicació	54
5.6 Proves	54
5.6.1 Funcionals	54
5.6.2 Compatibilitat	54
5.6.3 Accessibilitat	55
5.6.4 Seguretat	55
 <u>CAPÍTOL 6: CONCLUSIONS</u>	 <u>57</u>
6.1 Objectius assolits	57
6.2 Desviacions de la planificació	57
6.3 Línies d'ampliació	59
6.4 Valoració personal	59
 <u>BIBLIOGRAFIA</u>	 <u>61</u>
 <u>AGRAÏMENTS</u>	 <u>63</u>

Índex de figures

Figura 2.1 Parts interessades	6
Figura 2.2 Equip de projecte	7
Figura 2.3 Comparació d'alternatives	9
Figura 2.4 Detall d'hores dedicades al projecte	10
Figura 2.5 Detall d'honoraris	10
Figura 2.6 Desglossament de les tasques del projecte	10
Figura 2.7 Diagrama de Gantt de la planificació del projecte	12
Figura 2.8 Especificació de riscos	13
Figura 2.9 Catalogació i priorització de riscos	14
Figura 2.10 Pla de contingència	14
Figura 2.11 Detall dels costos de personal	15
Figura 2.12 Cost dels recursos i amortització	16
Figura 2.13 Taula resum del cost total del projecte	16
Figura 4.1 Esquema comunicació Apache+PHP	22
Figura 4.2 Esquema HTML	22
Figura 4.3 Interfície gràfica del phpmyadmin	23
Figura 4.4 Entorn Zend Studio	24
Figura 4.5 Entorn de desenvolupament phpDesigner 7	24
Figura 4.6 IDE Eclipse	25
Figura 4.7 Esquema funcionament Apache + MySQL	26
Figura 4.8 Diagrama entitat relació de la nostra base de dades	26
Figura 4.9 Casos d'ús per a l'usuari	32
Figura 4.10 Casos d'ús per al perfil responsable	33
Figura 4.11 Casos d'ús per al perfil administrador	33
Figura 4.12 Diagrama de seqüència del cas d'ús inventari	34
Figura 4.13 Diagrama de seqüència del cas d'ús importar	35
Figura 4.14 Diagrama de seqüència del cas d'ús informe d'incidències	36
Figura 4.15 Diagrama de seqüència del cas d'ús exportar	36
Figura 4.16 Segell acreditatiu W3C	38
Figura 4.17 Estructura de la pàgina	39
Figura 4.18 Pantalla de benvinguda a l'aplicació	40
Figura 4.19 Pantalla d'inventari	41
Figura 4.20 Aparença de l'aplicació quan es realitza l'inventari	42
Figura 5.1 Funcionament de l'arquitectura MVC	45
Figura 5.2 Classe centre	46
Figura 5.3 Mètode existsInventari de la classe centre	47
Figura 5.4 Vista llistarUsuaris	48

Figura 5.5 Controlador espais	49
Figura 5.6 estructura de directoris	50
Figura 5.7 Assignació de les variables de sessió	52
Figura 5.8 Control d'inactivitat i accés a funcionalitats restrictives	52
Figura 5.9 Ús del mètode màgic autoload	53
Figura 5.10 Funció que marca checkboxs segons la seva classe	53
Figura 6.1 Desviacions desglossades per tasca	57

Capítol 1: Introducció

1.1 Pròleg

El procés de fer inventari en organitzacions i empreses esdevé un procés capital a l'hora de controlar els béns i el patrimoni de les mateixes. En algunes empreses la compra d'aquest material esdevé la seva principal inversió, suposant d'aquesta manera una important despesa. Per a poder gestionar aquesta inversió i treure'n profit cal que tot el material estigui localitzat i en bon estat. Per a fer-ho possible cal disposar d'informació periòdica, real, veraç i actualitzada per a controlar diferents ítems. Cal que es puguin localitzar els diferents elements i saber on són exactament, determinar la quantitat d'elements que existeix en una ubicació determinada i, fins i tot, saber en quin estat es troba un material determinat.

Amb tota aquesta informació a l'abast, l'equip de gestió podrà prendre les accions necessàries per a preservar i millorar la qualitat de l'equipament i, d'aquesta manera, millorar també el servei. Mitjançant aquest sistema de control i seguiment el material trencat o defectuós podrà ser reparat o substituït si cal amb més rapidesa i efectivitat. A més a més, els canvis de la localització física dels diferents elements seran notificats conseqüentment per a disposar del que es necessiti quan es necessiti, sense perdre més temps del necessari.

1.2 Situació actual

Ja fa anys que la Universitat Autònoma de Barcelona (UAB) du a terme un procés d'identificació del material de la propietat de la Universitat mitjançant un codi de barres identificatiu únic per a cadascun dels elements. La compra de taules, cadires o equips informàtics són per a la UAB un patrimoni molt important. Per aquesta raó, el control i la localització periòdica de tots aquests elements és de gran importància per a poder identificar defectes en la organització, gestió i estat del material i prendre les mesures pertinents per a preservar la qualitat de l'equipament.

Tot i la identificació i indexació dels elements no es porta un procés de control com a tal. Així doncs i per exemple, no es realitzen revisions periòdiques del material. Els canvis d'ubicació s'han de comunicar a la Gestió Econòmica per a que faci els canvis pertinents a l'aplicació informàtica corresponent. És fàcil doncs que es produeixin canvis d'ubicació sense informar degudament, fet que, normalment, acaba amb la pèrdua de localització d'un element i la seva conseqüent pèrdua econòmica i de servei per la Universitat i pels seus usuaris. Amb el sistema actual, un cop perdem la localització d'un element és molt complicat recuperar-la, el que comporta un dèficit en la qualitat de la gestió del patrimoni de la Universitat.



Tampoc es porta un control exhaustiu del material trencat i/o defectuós. Aquestes anomalies només es detecten de manera totalment manual i fortuïta quan un treballador de la UAB ho percep i ho notifica a la Gestió Econòmica. Només llavors, els òrgans pertinents intervenen per a poder restablir qualsevol anomalia en el mobiliari o equipament si es considera convenient.

Caldria, doncs, poder oferir alguna solució a aquestes mancances detectades pel que fa a la gestió dels elements actius de la Universitat

1.3 Objectius del projecte

El projecte té com a objectiu realitzar un software que es converteixi en una eina real i usable per part dels treballadors de la Universitat encarregats de l'inventari per a millorar el procés de gestió del material de la UAB.

El programa ha de ser modulable, perdurable i extensible a futures millores i/o ampliacions. A més, es valora sobretot la seguretat del sistema i la seva correcta resposta tan pel que fa a circumstàncies anòmales com a qualsevol tipus d'error. El software ha de ser robust i estable per tal de respondre a aquests possibles errors sense comprometre ni la informació emmagatzemada ni el funcionament global de l'aplicació. Només així podrem aconseguir un software de confiança i allargar el període d'explotació del mateix.

L'aplicació ha de contemplar i s'ha d'adaptar a l'estructura logística de la Universitat, el personal disponible per a realitzar l'inventari del material i adaptar-se a la distribució geogràfica dels diferents centres de la UAB.

Cal que les interfícies i el codi s'adaptin als estàndards de qualitat vigents per a maximitzar la seva seguretat, estabilitat, eficiència i flexibilitat.

1.4 Motivacions personals

Com ja he comentat a anteriorment, penso que el procés que realitza la Universitat Autònoma de Barcelona per a la gestió i control del material és molt millorable. Començo aquest projecte amb l'objectiu de posar a disposició de la UAB les eines i els coneixements que he après aquests anys per tal de facilitar i optimitzar aquest procés i, perquè no, millorar en la mesura del que sigui possible l'eficàcia de la gestió de la Universitat.

La Universitat Autònoma de Barcelona es financia principalment amb capital públic que surt de tots els contribuents. És la responsabilitat de la Universitat rendir comptes amb la societat pel que fa a l'ús d'aquests recursos. Calen doncs, iniciatives des de dins de la UAB per a millorar la gestió d'aquests fons i maximitzar el seu rendiment. Aquest projecte va encarat cap a aquest

concepte: intentar millorar la gestió dels recursos de la Universitat per a treure'n un rendiment que repercuteixi en la millora de l'aprofitament de recursos públics i en la qualitat de la Universitat. Aquesta és una de les motivacions personals que m'ha portat a realitzar aquest projecte.

Personalment porto dos anys col·laborant amb els Serveis Informàtics de la UAB en l'implementació de petits projectes de software per a la millora dels processos interns de la Universitat. A més, molts dels projectes realitzats estaven desenvolupats amb algunes de les tecnologies que també es faran servir en aquest projecte com és el cas del llenguatge PHP, CSS, Javascript, servidors Apache, sistema MySQL per la gestió de la base de dades, etc. Penso que el desig d'aprofundir en aquestes tecnologies és també una gran motivació.

Aquest projecte és un repte personal per a posar en pràctica tots els coneixements adquirits en aquets anys d'estudi a l'Escola d'Enginyeria de Sabadell i als Serveis Informàtics Distribuïts (SID) de la Universitat Autònoma de Barcelona. Posar en acció i amb una finalitat pública tots els coneixements acumulats a les diferents assignatures cursades en el transcurs d'aquets anys que van des de disciplines com l'algorítmica fins a enginyeria del software, entre d'altres. Aplicar noves tecnologies i metodologies que no he fet servir fins ara, com és el cas del patró de desenvolupament model-vista-controlador o l'AJAX mitjançant llibreries com jQuery, etc.

Un dels factors que també em motiva és l'exigència d'aquest projecte. És dels més grans i complexos que he realitzat al llarg d'aquests anys. És per mi una experiència nova poder dissenyar i implementar un projecte d'aquesta magnitud. Aquesta responsabilitat, amb un objectiu ambiciós, incentiva totes les meves motivacions per a poder realitzar un software funcional, usable i segur per al seu ús a gran escala dins de la Universitat.

1.5 Contingut de la memòria

En aquest document es tractaran els principals aspectes que incideixen en el desenvolupament de la nostra aplicació.

En aquest primer capítol es comenten tant problemes exclusivament referents al problema a resoldre com motivacions personals a l'hora de triar aquest projecte. No ens hem endinsat massa en aspectes tècnics ja que aquest capítol té l'objectiu d'introduir els capítols posteriors. L'estat de la situació actual referent a l'aplicació que volem implementar i els objectius que volem assolir desenvolupant aquest projecte són temes que es tractaran en aquest apartat.

Tot seguit, s'avaluarà la viabilitat del projecte i es definiran uns objectius específics, realistes i realitzables per al desenvolupament del projecte. També es tindran en compte diferents alternatives que ens ofereix el mercat per poder valorar si es poden ajustar als nostres objectius. S'estudiarà l'aspecte econòmic realitzant un pressupost per a l'aplicació i discutirem

si el projecte és econòmicament viable. Un cop establim que el projecte és viable proposarem una planificació per al projecte.

A continuació, al tercer capítol, començarem l'anàlisi en profunditat del problema a tractar. Definirem els requeriments i les limitacions imposades per diferents circumstàncies com el hardware on ha de utilitzar-se l'aplicació. També dissenyarem els diferents perfils d'usuari que han d'actuar en el procés d'inventari. En aquest capítol tractarem, sobretot, el disseny conceptual de l'aplicació.

Un cop hem especificat el disseny conceptual a l'apartat interior, passarem al capítol 4, que fa referència al disseny, i on definirem més tècnicament les diferents característiques del nostre software. Raonarem l'elecció de l'entorn de desenvolupament triat i el sistema gestor de base de dades. En aquest apartat també trobem els diagrames de casos d'ús que ens especifiquen les diferents funcionalitats que cada tipus d'usuari pot dur a terme i els diagrames de seqüència que determinen el comportament de l'aplicació per a les principals funcionalitats. Finalment, hem realitzat un disseny conceptual de les interfícies amb les que l'usuari interactuarà i una breu especificació dels mòduls que comprèn l'aplicació.

Començada la codificació de l'aplicació, al capítol 5 tractarem tots els aspectes relatius a la programació de l'aplicació pròpiament dita. Comentem les principals tècniques utilitzades per millorar l'eficiència i llegibilitat del codi així com l'explicació del patró model-vista-controlador que hem utilitzat per al desenvolupament de l'aplicació. També expliquem el procés de proves que hem realitzat a l'aplicació per a certificar que aconsegueixi els objectius i requeriments explicats en els capítols anteriors.

Finalment al capítol final farem el tancament del projecte, i exposarem les conclusions del mateix. Analitzarem si s'han assumit els objectius assolits i les dificultats que hem trobat per a desenvolupar l'aplicació. També realitzarem una breu anàlisi personal.

Juntament amb la memòria en format paper, s'adjunta un CD on s'inclou la memòria en format PDF, el manual d'usuari en format PDF i tot el codi del projecte.

Capítol 2: Estudi de viabilitat

2.1 Introducció

2.1.1 Tipologia i paraules clau

El projecte està enquadrat dintre de la categoria de desenvolupament de sistemes de software.

Les paraules clau triades per a definir el projecte són: inventari i codi de barres.

2.1.2 Descripció

El procés per a inventariar el material d'una organització o empresa és avui una necessitat per a portar un control exhaustiu dels béns propietat de l'entitat.

L'aplicació a desenvolupar està destinada a dur a terme l'inventari del material dels diferents despatxos, aules, laboratoris, etc. dels centres de la Universitat Autònoma de Barcelona. Disposarem de tota la informació centralitzada referent al material indexada amb un codi de barres identificatiu. L'aplicatiu, que actuarà de manera distribuïda, haurà de detectar totes les possibles anomalies quan el personal realitzi l'inventari.

2.2 Descripció i priorització d'objectius

A continuació detallem els objectius proposats per el projecte així com la seva classificació segons la seva importància en tres categories: crítics, prioritaris i secundaris

- O.1.** Importar la informació de treball necessària per al funcionament de l'aplicació. **(Crític)**
- O.2.** Identificar els diferents elements de l'inventari. **(Crític)**
- O.3.** Detectar anomalies entre l'inventari teòric i el real: canvis d'ubicació, referències perdudes o referències noves. **(Crític)**
- O.4.** Generar un informe d'incidències al finalitzar l'inventari. S'ha de recollir la informació referent als elements perduts, canvis d'ubicació i referències noves. **(Crític)**
- O.5.** Els usuaris s'han d'identificar per utilitzar l'aplicació. L'administrador podrà assignar-los-hi una o més localitzacions d'actuació. **(Prioritari)**
- O.6.** Agilitzar i millorar el procés d'actualització de la informació centralitzada. **(Secundari)**
- O.7.** Exportar un fitxer de les incidències per a facilitar la integració amb futures aplicacions. **(Secundari)**

2.3 Parts interessades

A continuació presentem les parts interessades en el desenvolupament del projecte així com les seves responsabilitats i càrrecs (figura 2.1).

<i>Nom</i>	<i>Descripció</i>	<i>Responsabilitat</i>
Araceli Martos	Cap gestió econòmica del Campus de Sabadell	Encarregada de compres, actius, material, etc. del Campus de Sabadell

Figura 2.1 Parts interessades.

2.4 Equip de projecte

A la taula següent (figura 2.2) es mostra com queda confeccionat l'equip de projecte així com les funcions de cada component de l'equip.

<i>Nom</i>	<i>Descripció</i>	<i>Responsabilitat</i>
Jordi Pons Aróztegui	Cap de projecte (CP)	Defineix, gestiona, planifica i controla el projecte.
Sergi González Pérez	Analista (A)	Col·labora amb el cap de projectes en l'estudi de viabilitat i la planificació. Analitza l'aplicació: arquitectura, metodologia, especificacions, estàndards... Participa en el disseny i la validació.
Sergi González Pérez	Programador (P)	Dissenya i desenvolupa l'aplicació d'acord amb l'anàlisi i planificació prevista. Participa en el procés de validació i implantació.
Sergi González Pérez	Dissenyador (D)	Dissenya i desenvolupa l'aspecte de l'aplicació d'acord amb l'anàlisi i normes d'usabilitat establertes.
Sergi González Pérez	Tècnic de proves (TP)	Participa en el disseny de proves internes i externes. Realitza les proves i participa en el procés de control de qualitat.
Jordi Pons	Director del	

Aróztegui	projecte (DP)	Supervisa la feina de l'alumne.
-----------	---------------	---------------------------------

Figura 2.2 Equip de projecte.

2.5 Normativa

Existeixen varies normatives a l'hora de desenvolupar el projecte.

- Normativa de projectes d'enginyeria informàtica aprovada per la Junta Permanent de l'EUI el 30 de juny de 2009 [1].
- Llei orgànica de protecció de dades (LOPD) que no afectarà al nostre projecte ja que no emmagatzemem dades privades dels usuaris.
- Drets d'autor. Tampoc afecten al projecte ja que estarà desenvolupat en la seva totalitat amb software lliure, per tant no caldrà estar subjecte a normatives de tecnologies propietàries.

2.6 Producte i documentació del projecte

- Es lliurarà una aplicació informàtica.
- Es lliurarà un manual d'usuari.

2.7 Context de la situació actual

Actualment, al Campus de Sabadell de la UAB tot el material està inventariat amb un codi de barres identificatiu. Tot i això no es porta un seguiment i control del material. Així doncs el procés de control d'inventari no s'ha dut a terme mai.

La informació està continguda en dues aplicacions de comptabilitat. SIDECA, que es ve utilitzant des de 1990 i actualment es comença a implantar SUMMA. L'objectiu d'aquestes aplicacions, més que fer un inventari físic, és la de portar una comptabilitat dels actius de la UAB.

El procés d'alta d'un element el du a terme el servei de gestió econòmica emplenant un formulari amb tota la informació rellevant de l'element: nom, descripció, preu, número de sèrie, ubicació, etc. Després aquest formulari s'introdueix a l'aplicació SUMMA i finalment s'obté el codi de barres que s'enganxa en un lloc visible a l'element actiu.

2.8 Alternatives

2.8.1 Descripció d'alternatives

2.8.1.1 Alternativa 1: ERP

Adquisició i adaptació d'un ERP (*Enterprise Resource Planning* o Planificació de Recursos Empresariums), per exemple Openbravo [2] o OpenERP [3].

- Característiques OpenERP:
 - Empresa: gestió de vendes, compres, facturació, etc.
 - Logística: gestió de magatzems, transports, etc.
 - Comptabilitat: gestió de pagaments, d'actius, pressupostos, etc.
 - Recursos humans: gestió de despeses, vacances, etc.
 - Projectes: gestió operativa, financera, etc.
- Costos:
 - OpenERP està publicat sota llicència GPL (General Public Licence [4]). Per tant el seu cost d'adquisició és de 0 €.

2.8.1.2 Alternativa 2: GDS Bicicletaria

Adquisició d'una aplicació per a la gestió d'inventaris en magatzems com GDS Bicicletaria 5.0 [5].

- Característiques Bicicletaria 5.0:
 - Gestió d'inventaris amb suport per a lector de codi de barres.
 - Sistema de permisos per usuaris delimitant funcionalitats accessibles per a cadascun d'ells.
 - Impressió d'etiquetes.
- Costos:
 - L'aplicació té un cost de 250€ per llicència (per cada un dels ordinadors).

2.8.1.3 Alternativa 3: Aplicació pròpia

Desenvolupament d'una aplicació de gestió d'inventaris.

- Característiques:
 - Total independència de l'aplicació
 - S'ajusta exactament als requisits del patrocinador.
 - Ajustable als recursos disponibles per a l'entitat.

2.8.2 Comparació d'alternatives

Mostrem a continuació (figura 2.3) una taula comparativa de totes les alternatives tenint en compte els principals factors que afectaran al desenvolupament i implantació de la solució.

	Costos d'adquisició.	Costos d'adaptació	Suport	Nivell d'integració.	Complexitat	Formació
Alternativa 1	0€	Molt Alts	Partners, contracte manteniment	Baix	Molt Alta	Partners
Alternativa 2	250€ per PC	Alts	Tres mesos gratuïts.	Baix	Alta	Es desconeix
Alternativa 3	0€	No cal	Inclosa en el projecte	Alt	Mitjana	Inclosa al projecte

Figura 2.3 Comparació d'alternatives.

2.8.3 Solució proposada i justificació

Creiem que la solució més senzilla d'implementar i que s'ajustarà més a les expectatives posades sobre el programa serà l'alternativa 3. Desenvoluparem la nostra pròpia aplicació basant-nos en els requisits funcionals que ens proporciona el promotor.

2.9 Planificació

2.9.1 Descripció

- El projecte es desenvoluparà entre novembre del 2009 i maig del 2010.
- Es dedicaran al projecte 15 hores setmanals (3 hores diàries).
- Data inici: 19 Novembre 2009.
- Data final prevista: 25 de Maig del 2010.

- Està previst que el projecte quedi paralitzat el mes de gener (veure calendari de la planificació)
- El total d'hores dedicades al projecte serà de 302 hores. Tot seguit la taula d'hores desglossades per etapes (figura 2.4).

Eta pa	Hores
Inici del projecte: assignació i matriculació del projecte.	2
Planificació.	31
Anàlisi de l'aplicació.	24
Disseny de l'aplicació.	27
Desenvolupament.	128
Proves.	39,85
Implantació.	14
Generació documents (memòria).	30
Tancament del projecte.	1,1
Defensa del projecte.	5

Figura 2.4 Detall d'hores dedicades al projecte

2.9.2 Recursos humans

A continuació presentem la taula amb el detall dels honoraris de cada rang de l'equip (figura 2.5).

Recursos humans	Valoració
Director de projecte	90€/h
Cap de projecte	65€/h
Analista	45€/h
Programador	25€/h
Dissenyador	30€/h
Tècnic de proves	20€/h

Figura 2.5 Detall d'honoraris.

2.9.3 Tasques del projecte

Tot seguit expliquem mitjançant aquesta taula les diferents tasques que comprenen el projecte així com la seva duració i els recursos emprats en la mateixa (figura 2.6).



Número	Tasca	Duració	Recursos	Predecessora
1	Inici del projecte: assignació i matriculació del projecte	2 hores	CP; DP[10%]	
	Planificació	10,33 dies		
2	Estudi de viabilitat	30 hores	CP	1
3	Aprovació EV (Checkpoint)	1 hora	CP; DP[10%]	3
	Anàlisi de l'aplicació	8 dies		
4	Anàlisi requisits (casos d'ús)	10 hores	A	4
5	Anàlisi de dades (base de dades)	5 hores	A	6
6	Anàlisi seguretat i legalitat	5 hores	A	7
7	Documentació de l'anàlisi	3 hores	A	8
8	Aprovació Anàlisi (Checkpoint)	1 hora	CP; A[50%]; DP[10%]	9
	Disseny de l'aplicació	9 dies		
9	Disseny de la base de dades	5 hores	A[80%]; P[20%]	10
10	Disseny modular de l'aplicació	3 hores	A[80%]; P[20%]	12
11	Disseny de la interfície	10 hores	A[57%]; P[30%]; D[40%]	13
12	Disseny de les proves	5 hores	A[60%]; P[20%]; TP[20%]	14
13	Documentació del disseny	3 hores	A	15
14	Aprovació del disseny (Checkpoint)	1 hora	CP; DP[10%]	16
	Desenvolupament	42,67 dies		
15	Preparació entorn de desenvolupament	2 hores	P	17
16	Configuració BD	6 hores	P	19
17	Funcionalitats de l'aplicació	100 hores	P	20
18	Interfície	20 hores	D[80%]; P[20%]	21
	Proves	13 dies		
19	Proves unitàries	15 hores	TP[50%]; P[50%]	22
20	Proves d'integració	15 hores	TP[90%]; P[50%]	24
21	Proves d'estrès (incidències, riscos)	5 hores	TP[90%]; P[10%]	25
22	Documentació de les proves	3 hores	P	26
23	Aprovació de les proves	1 hora	CP[50%]; A; P[25%]; DP[10%]	27

	Implantació	4,67 d		
24	Instal·lació i proves reals	14 hores	A[70%]; P[30%]	28
	Generació documents (memòria)	30 hores	CP	28
	Tancament del projecte	1 hora	CP; DP[10%]	33
	Defensa del projecte	5 hores	CP	34

Figura 2.6 Desglossament de les tasques del projecte.

2.9.4 Planificació temporal

A continuació es mostra el diagrama de Gantt de la planificació del projecte amb les tasques numerades per a facilitar la seva comprensió (figura 2.7).



Figura 2.7 Diagrama de Gantt de la planificació del projecte.

2.10 Riscos

2.10.1 Catalogació i priorització

Tot seguit es presenten els principals riscos que pot comportar el desenvolupament del projecte (figura 2.8). També s'expliquen breument les possibles conseqüències d'aquets riscos i en quina etapa del desenvolupament poden sorgir.

	Títol	Localització de l'error	Conseqüències
R.1	Planificació temporal optimista.	• Estudi de viabilitat.	No s'acaba el projecte en la data prevista. Augmenten els recursos.
R.2	Manca alguna tasca necessària.	• Estudi de viabilitat.	No es compleixen els objectius del projecte.
R.3	Pressupost poc ajustat.	• Estudi de viabilitat.	Menys qualitat del producte, pèrdues econòmiques.
R.4	Canvi de requisits.	• Estudi de viabilitat. • Anàlisi.	Endarreriment en desenvolupament i resultat.
R.5	Equip de projecte massa reduït.	• Estudi de viabilitat.	Endarreriment en la finalització del projecte. No es compleixen els objectius del projecte.
R.6	Eines de desenvolupament inadequades.	• Implementació.	Endarreriment en la finalització del projecte. Menys qualitat.
R.7	Dificultat per accedir als stakeholders.	• Estudi de viabilitat. • Anàlisi. • Formació.	Manquen requisits o són inadequats. Endarreriment del projecte, menys qualitat.
R.8	Realització incorrecta de la fase de proves.	• Desenvolupament. • Implantació.	Manca de qualitat, deficiència operativa, insatisfacció dels usuaris. No es compleixen els objectius.
R.9	Incompliment de normativa o legislació.	• En qualsevol fase.	No es compleixen els objectius. Repercussions legals.
R.10	Manca d'implantació de mesures de seguretat.	• Estudi de viabilitat. • Anàlisi. • Desenvolupament.	Pèrdua d'informació, incompliment legal, pèrdues econòmiques.
R.11	Abandonament del projecte abans de la seva finalització.	• En qualsevol fase.	Pèrdues econòmiques.

Figura 2.8 Especificació de riscos.

Els riscos anteriors queden prioritzats segons la seva probabilitat d'aparició i la gravetat del seu impacte (figura 2.9).

	Probabilitat	Impacte
R.1	Alta.	Crític.
R.2	Alta.	Crític.
R.3	Alta.	Crític.
R.4	Alta.	Marginal.
R.5	Alta.	Crític.
R.6	Baixa.	Crític.
R.7	Baixa.	Crític.
R.8	Alta.	Crític.
R.9	Mitjana.	Crític.
R.10	Alta.	Crític.
R.11	Mitjana.	Catastròfic.

Figura 2.9 Catalogació i priorització de riscos.

2.10.2 Pla de contingència

En cas de que alguns dels riscos descrits a l'apartat 2.10.1 tingui lloc durant la realització del projecte, cal idear una sèrie de mesures a priori per a minimitzar l'impacte del mateix. A continuació es descriu el pla pensat per a cada risc (figura 2.10).

	Mesures a prendre
R.1	Ajornar alguna funcionalitat, afrontar possibles pèrdues, fer una assegurança.
R.2	Revisar l'estudi de viabilitat, modificar la planificació.
R.3	Renegociar amb el client, afrontar possibles pèrdues, fer una assegurança.
R.4	Renegociar amb el client, ajornar funcionalitat, modificar planificació i pressupost.
R.5	Demandar un ajornament, negociar amb el client, afrontar pèrdues.
R.6	Millorar formació de l'equip, preveure eines alternatives, millorar la qualitat.
R.7	Fixar un calendari de reunions, millorar el contacte amb

	el client.
R.8	Dissenyar els tests amb antel·lació, realitzar tests automàtics, negociar contracte de manteniment, donar garanties, afrontar pèrdues econòmiques.
R.9	Revisar les normes i legislació, consultar un expert, afrontar possibles repercussions legals.
R.10	Revisar seguretat en cada fase, aplicar polítiques de seguretat actives.
R.11	No té solució.

Figura 2.10 Pla de contingència

2.11 Pressupost

2.11.1 Estimació cost personal

Una vegada contabilitzades totes les hores de la planificació i tenint en compte el preu de l'hora segons el càrrec, a continuació detallem el cost de personal total del projecte desglossat per categories (figura 2.11).

	Total hores	Cost
Director de projecte	0,63	56,7€
Cap de projecte	70,77	4.600 €
Analista	50,13	2.255,85 €
Programador	139,08	3477 €
Dissenyador	18,85	565 €
Tècnic de proves	22,5	450 €

Total	11.404,55 €
--------------	--------------------

Figura 2.11 Detall dels costos de personal.

2.11.2 Estimació cost dels recursos

Cal tenir en compte també tots els costos dels recursos utilitzats pel projecte. Ens hem decantat, en la majoria de casos, pel software lliure opensource per a reduir al màxim aquets costos. Hem utilitzat també algun software privatiu amb llicència de pagament però gràcies a l'acord de MSDNAA (Microsoft Developer Network Academic Alliance) amb l'Escola d'Enginyeria hem aconseguit software de manera gratuïta com Microsoft Office, Microsoft Windows 7 o Microsoft Visio. No hem incluit aquests elements en el detall del cost dels recursos ja que el seu cost és nul. A continuació detallem els recursos a amortitzar (figura 2.12)

	Cost unitari	Període amortització	Període utilització	Cost Amortització
Amortització PC	1.200€	4 anys	5,5 mesos	137,5 €

Figura 2.12 Cost dels recursos i amortització.

2.11.3 Resum i anàlisi cost - benefici

Cal analitzar el cost total del projecte detallat a la següent taula (figura 2.13).

Cost de desenvolupament del projecte.....	11.404,55 €
Cost Amortitzacions.....	137,5 €
Cost total del projecte	11.542,05 €

Figura 2.13 Taula resum del cost total del projecte.

Tot i que el cost del projecte és alt. Pensem que al no portar-se a terme fins a la data un control del material de la UAB, a la llarga el balanç serà positiu. Al no portar-se aquest control els actius de la UAB poden perdre's o fins i tot ser robats sense que el departament implicat en tingui coneixement, el que a la llarga comporta despeses no planificades i una disminució de la qualitat del servei.

2.12 Conclusions

2.12.1 Beneficis

- ✓ Millorarem substancialment el procés de control del material dels diferents centres. Realitzant-lo d'una manera ràpida i eficient i permetent portar un seguiment dels actius de la universitat.
- ✓ Reducció de despeses de personal entre fer l'inventari físicament i fer l'inventari amb la nova aplicació.
- ✓ Millora de la seguretat del centre i en el seguiment dels seus actius. Amb un control exhaustiu podem estudiar les pèrdues de material i millorar la seguretat.

2.12.2 Inconvenients

- ✗ Inversió alta.
- ✗ Necessitat d'un període de formació.

Capítol 3: Anàlisi

3.1 Perfils d'usuari

L'aplicació serà utilitzada per tres tipus diferents d'usuari:

PU.1. Administradors.

Gestionen i administren l'aplicació. Poden donar d'alta nous responsables de centre i associar-los a un o més centres.

PU.2. Responsables de centre.

Gestionen l'inventari d'un centre. Donen d'alta nous usuaris i els associen permisos per a inventariar espais determinats.

PU.3. Usuaris.

Realitzen l'inventari dels espais que els han assignat els responsables de centre. Podran afegir comentaris als elements i als espais.

3.2 Requisits funcionals

RF.1. Identificació d'usuaris.

Els diferents perfils d'usuari (administradors, responsables de centre i usuaris) s'han d'identificar correctament per a poder entrar a l'aplicació. Això ens aportarà un major control del procés. Els responsables de centre podran delimitar àrees d'inventariat consistents en una enumeració d'espais i així distribuir la zona a inventariar entre diferents usuaris agilitzant el procés i fent-lo més ràpid.

RF.2. Importació de dades.

Els responsables de centre han de ser capaços d'importar la base de dades amb la informació de treball referent a l'inventari teòric mitjançant un mòdul de l'aplicació. El format de la base de dades pot ser un arxiu csv (*comma-separated values*) amb els camps separats per “,” i “;” com a caràcter de final de línia. Aquestes especificacions es podran modificar al fitxer de configuració de l'aplicació.

L'estructura d'aquesta base de dades ha de ser la indicada a RIE.1 (apartat requisits no funcionals) i l'aplicació importarà únicament els camps:

- Codi: Codi de barres de l'element.
- Centre: Centre de l'element.
- Ubicació: Espai on es troba l'element.

En aquest procés també es generaran les taules “centres” i “espais” a partir dels centres i ubicacions de la base de dades anterior.

RF.3. Classificació d'inventari

Una vegada importades les dades i generades les taules “centres” i “espais” (RF.1), el software haurà de mostrar els elements filtrats per aquests camps per a millorar la seva visualització. La navegació pels diferents centres i espais es realitzarà mitjançant menús desplegable.

Aquesta visualització agilitzarà enormement el procés d'inventariat ja que permetrà als usuaris poder recórrer cada espai i trobar llistats de l'inventari teòric d'aquesta localització.

RF.4. Identificació d'elements.

El software identificarà correctament els elements mitjançant un codi de barres identificador que serà únic. Comprovarà que aquest codi està inclòs a l'inventari teòric i que la seva localització es correspon amb el seu espai teòric. Els elements trobats apareixeran en color verd a la visualització de l'inventari de l'espai, els no trobats en vermell. Si l'element no és identificat correctament el programa haurà de comprovar que aquest no estigui situat en una altra localització teòrica o que el codi no es trobi en tot l'inventari. En aquets casos el programa afegirà una incidència.

RF.5. Comentaris a elements o espais.

Els usuaris que realitzin l'inventari han de poder deixar petits comentaris explicant alguna circumstància anormal associats a cadascun dels elements o espais si així ho desitgen. El software afegirà una nova incidència en aquets casos.

RF.6. Informe d'incidències.

Una vegada els usuaris donin l'inventari per conclòs, l'aplicació generarà un informe d'incidències que contindrà tots els possibles canvis d'ubicació, elements perduts, elements nous i tots els comentaris associats a elements o espais que s'hagin introduït. L'aplicació donarà accés a aquest informe al responsable de centre que podrà exportar-lo en format pdf triant les incidències i enviar-lo al seu correu electrònic. El software també desarà aquestes incidències en un històric d'incidències (RF.8.).

RF.7. Generació fitxer de sortida.

Una vegada el responsable del centre aprovi l'inventari, el software generarà un arxiu codificat en XML per a la possible futura integració amb aplicacions que permetin tractar les modificacions detectades a l'inventari. El responsable podrà seleccionar les incidències a incloure en aquest arxiu.

RF.8. Arxiu històric d'inventaris.

L'aplicació desarà les incidències generades en un inventari aprovat pel responsable de centre en un arxiu històric d'incidències. Així doncs, l'aplicació contindrà totes les incidències que s'hagin generat en inventaris anteriors i els responsables de centre tindran accés a aquest llistat per possibles consultes.

RF.9. Control del procés d'inventari.

Els responsables de centre podran consultar en tot moment al procés d'inventari. Podent saber quins espais ja estan inventariats i quins falten per completar el procés.

3.3 Requisits no funcionals

3.3.1 Restriccions de disseny

RD.1. Acompliment dels estàndards.

L'aplicació web ha d'acomplir els estàndards aprovats per la *World Wide Web Consortium* (W3C) [6]. Referents als llenguatges XHTML 1.0 [7] i CSS2 [8].

RD.2. Limitacions de Hardware.

El software a implementar ha d'estar optimitzat per a la seva utilització en dispositius mòbils, principalment Netbooks. El dispositiu agafat com a model és l'Asus Eee PC 1000H [8]:

- Processador Intel Atom N270 a 1.6 GHz.
- 1 GB RAM DDR2.
- 80 GB de disc dur.
- Bateria de 7 hores d'autonomia i 1'45 Kg de pes.
- Resolució de 1024 x 600.

La pistola lectora de codis de barres que utilitzarem serà la WASP WW-S800 [9]

- Distància de transferència de 30 metres i fins a 50 metres sense obstacles.
- Compatible amb caràcters de delimitació de codi.

3.3.2 Objectius de Disseny

OD.1. El software ha de ser *user friendly*.

L'aplicació ha de tenir una interfície senzilla i intuïtiva. Ha de ser capaç de ser utilitzada per usuaris amb coneixements bàsics d'informàtica i amb poca experiència amb programes similars.

OD.2. El software ha de ser ampliable i fàcil de mantenir.

El software ha de tenir una arquitectura que permeti la fàcil integració amb futures ampliacions de l'aplicació. El sistema de manteniment de l'aplicació ha de poder ser utilitzat per usuaris amb coneixements bàsics d'informàtica.

3.3.3 Requisits sobre les interfícies externes

RII.1. Identificació d'usuaris amb LDAP.

La identificació dels diferents perfils d'usuari de l'aplicació es durà a terme mitjançant el protocol LDAP ja implementat pels Serveis Informàtics de la UAB.

RII.2. Estructura de la base de dades.

L'estructura de la base de dades que conté l'inventari ve donada per la UAB i l'haurem de respectar a l'hora de dissenyar el nostre software. Els camps de la base de dades referent al material són els següents:

- Fitxa: Número identificatiu de l'element.
- Data: Data de quan s'ha donat d'alta l'element a l'inventari.
- Centre: Centre on està ubicat l'element.
- Unitat: Unitat docent de l'element.
- Ubicació: Aula o Despatx on està ubicat l'element.
- Tipus: Categoria a la que pertany l'element inventariat.
- Descripció: Breu descripció de l'element.
- Marca: Marca o fabricant de l'element.
- Número de sèrie: Número de sèrie de l'element.
- Proveïdor: Codi identificatiu del proveïdor de l'element.
- Import: Import en euros de l'element.
- Estat.

RII.3. Codi de Barres.

Tot el material inventariat s'identifica mitjançant un codi de barres únic. El software identificarà cadascun dels elements mitjançant una pistola lectora de codis de barres sense fils.

Capítol 4: Disseny.

4.1 Introducció

A l'hora de desenvolupar una aplicació és molt important dissenyar correctament el software abans de programar-lo. Acomplint aquesta etapa correctament ens estalviarem molts maldecaps un cop comencem a codificar i facilitarem la comprensió de l'estructura i funcionament del projecte per part de persones alienes al desenvolupament del mateix. Es tracta d'una mesura molt important a prendre per a que, per exemple, facilitar que futurs desenvolupadors puguin ampliar la nostra aplicació si es dóna el cas. Sense una correcta definició del disseny, a aquets desenvolupadors els serà molt difícil entendre l'estructura del projecte i perdran molt temps i recursos en la comprensió del mateix abans de posar-se a desenvolupar.

Estructurarem el software en mòduls independents seguint l'estratègia de *divide & conquer* amb l'objectiu d'aconseguir un nivell d'abstracció lògic de fàcil comprensió per al desenvolupament. Cal maximitzar en la mesura del possible la independència funcional del software, és a dir, aconseguir un sistema modular format per una sèrie de components de manera que cadascun proporcioni un bon nivell d'abstracció i que una modificació en un d'aquests components tingui un impacte mínim sobre la resta.

4.1.1 Selecció del llenguatge de programació i de la base de dades

Per a desenvolupar aquesta aplicació ens trobem amb el clàssic esquema client-servidor ja que la informació de l'inventari ha d'estar centralitzada i es pot donar el cas de que els usuaris en facin un ús concurrent. Per a desenvolupar aquesta arquitectura podríem desenvolupar un client d'escriptori i un servidor que gestionés les peticions d'aquests clients, però considerant que la tendència actual de portar aplicacions cap al web, amb el que s'anomena *cloud computing*, creiem era incoherent fer una aplicació d'escriptori. Desenvolupant una aplicació web ens estalviem problemes de compatibilitats amb depèn quins ordinadors, a més funcionarà en tots els sistemes operatius i no caldrà realitzar una versió per a cadascun d'ells. També estalviem als usuaris tot procés d'instal·lador de software ja que s'accedirà a l'aplicació mitjançant un navegador web que incorporen la totalitat de sistemes operatius.

Així doncs decidits a fer una aplicació web ens trobem amb multitud de llenguatges, a més de l'XHTML indispensable per a mostrar pàgines a Internet, capaços de satisfer les nostres necessitats: ASP.NET, Python, etc. Tot i això finalment ens vam decidir per PHP (*PHP Hypertext Pre-processor*). PHP és un llenguatge sotmès a una llicència *opensource* (i per tant és lliure) utilitzat per a la generació de pàgines web dinàmiques. Els principals motius per

optar per PHP són la seva facilitat d'ús, la quantitat d'eines disponibles per a complementar-lo, la gran comunitat d'usuaris que programen en aquest llenguatge convertint-lo en un dels més utilitzats al web i finalment l'absència de llicències privatives. Farem servir la versió 5 del llenguatge, actualment la més utilitzada. Finalment cal dir que PHP s'executa en el servidor remot sota Apache. Apache és un servidor web HTTP de codi obert i gratuït líder mundial en el camp dels servidors i amb una quota de mercat de casi el 70%. Per tant la nostra aplicació serà portable fàcilment a la gran majoria d'aplicacions sense haver-la de modificar.

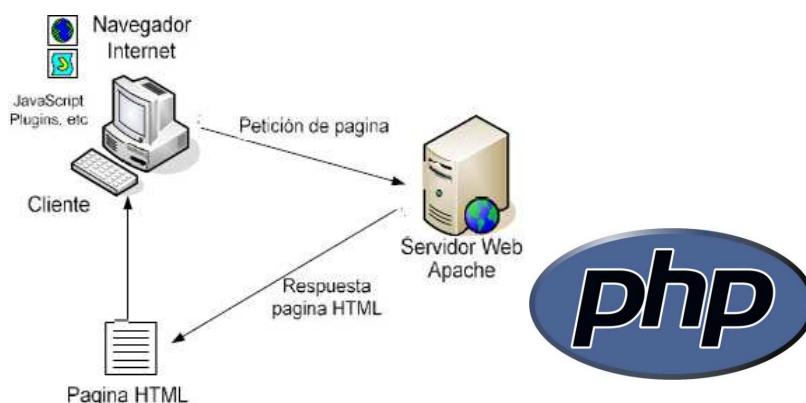


Figura 4.1 Esquema comunicació Apache+PHP

Per a complementar el PHP farem servir una sèrie de llenguatges auxiliars. Per a donar-li més dinamisme a la web utilitzarem Javascript. Javascript és un llenguatge d'*scripting* orientat a objectes i que s'executa al client (navegador). L'utilitzarem principalment per a la validació de formularis i per a millorar els efectes de presentació a l'aplicació. Parlarem d'aquests aspectes al capítol de codificació (capítol 5).

D'altra banda, per atorgar un aspecte més atractiu per a l'usuari farem servir el llenguatge CSS. El CSS defineix la presentació dels documents HTML. Amb la utilització d'aquest llenguatge perseguim l'objectiu de separar completament l'estructura del document de la seva presentació.

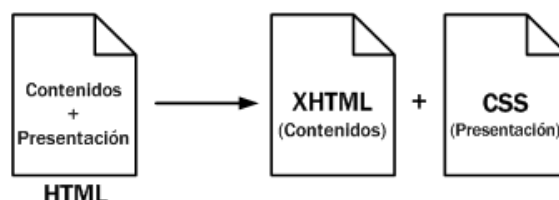


Figura 4.2 Esquema HTML.

Per a interactuar amb la base de dades hem optat pel llenguatge SQL (*Structured Query Language*) ja que té una molt bona integració amb PHP. SQL és el llenguatge dominant a la web pel que fa a interacció amb bases de dades i el farem servir juntament amb MySQL per a gestionar la nostra base de dades.

Un cop triats els llenguatges per a programar l'aplicació cal triar un gestor per a la base de dades que interaccioni amb el llenguatge SQL. Ens hem decantat pel Phpmyadmin. Es tracta d'una potent eina implementada en PHP per a administrar el MySQL i la nostra base de dades.

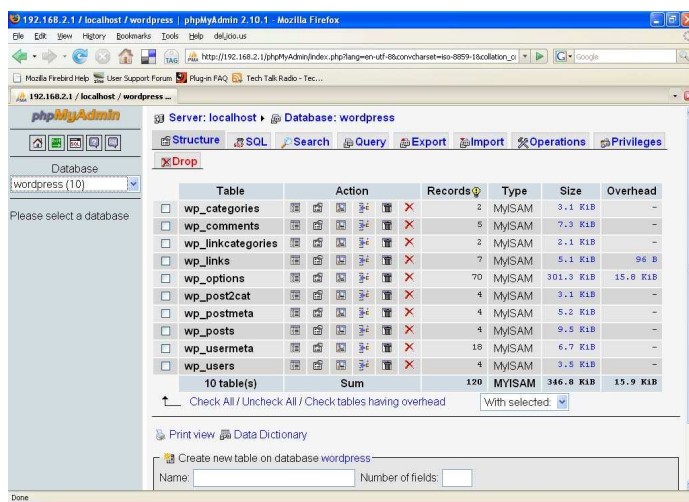


Figura 4.3 Interfície gràfica del phpmyadmin.

4.1.2 Selecció entorn de desenvolupament

Per a la implementació del projecte hem triat primerament un entorn de desenvolupament integrat (IDE). Ja que el programa estarà desenvolupat bàsicament en llenguatge PHP hem estudiat diferents IDEs per a programar en aquest llenguatge:

- Zend Studio 8: Es tracta de l'IDE de l'empresa Zend [11], el principal impulsor del llenguatge PHP. Té algunes característiques interessants per a l'ajuda de creació i gestió de projectes i eines per a la depuració i millora del codi. Suport nadiu per a HTML, Javascript, CSS, etc. Es tracta d'un software amb llicència propietària i de pagament (399 €). Per això queda descartat ja que incrementaria el cost del projecte i hi ha eines lliures que ens ofereixen el mateix que Zend Studio.

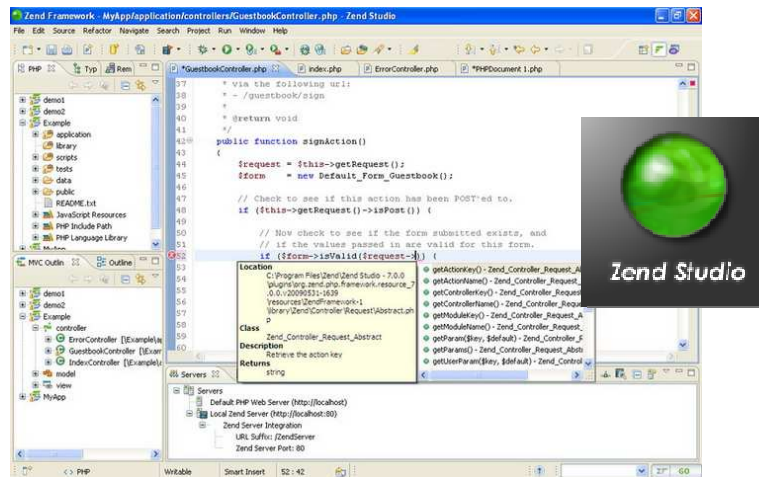


Figura 4.4 Entorn Zend Studio.

- PHP Designer 7: Aquest és l'entorn desenvolupat per l'empresa MPSoftware [12]. A l'igual que el Zend Studio, el PHP Designer també disposa d'eines per a la gestió de projectes, a més incorpora autocompletat de codi i un client FTP nadiu a dins de l'aplicació.

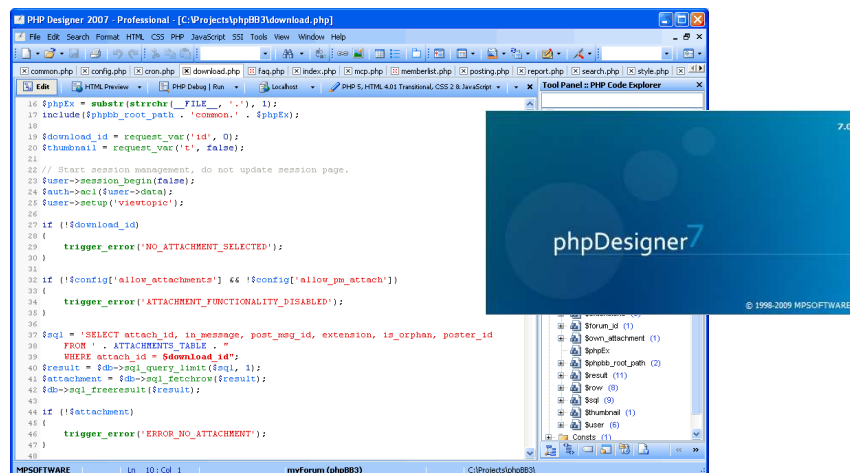


Figura 4.5 Entorn de desenvolupament phpDesigner 7

- Eclipse Helios 3.6 (PDT): Eclipse és un dels IDEs més utilitzats arreu del món [13]. Es tracta d'un software amb llicència opensource i és totalment gratuït. Consta d'infinitat d'eines per a fer més còmode el desenvolupament d'aplicacions com la gestió de projectes, el navegador d'arxius, els clients FTP, SFTP, SSH... per a gestionar els servidors remots des del mateix IDE i una gran comunitat d'usuaris al darrere que ajuda en el suport de l'aplicació i en la implementació de nous mòduls per a expandir les seves funcionalitats. Cal dir que Eclipse no és un IDE exclusivament per a PHP,

suporta molts més llenguatges com per exemple Java o C++ i a més a més és multiplataforma.

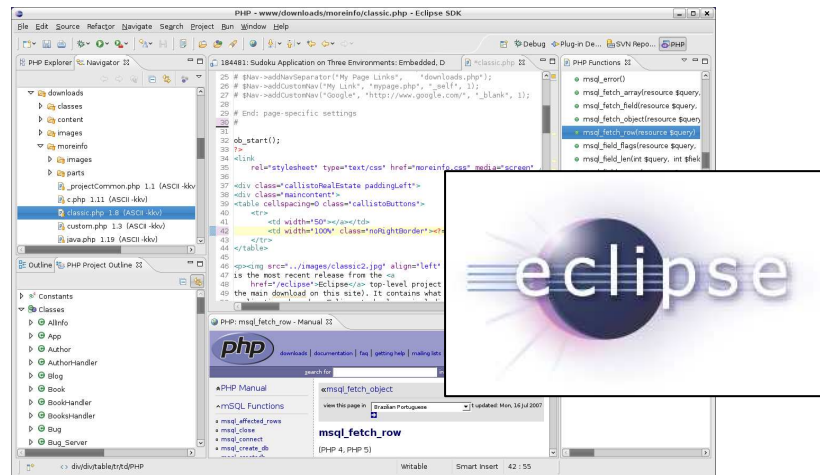


Figura 4.6 IDE Eclipse.

Finalment ens decidim per utilitzar Eclipse amb l'extensió PDT per a desenvolupar en llenguatge PHP. Els principals motius, com hem comentat són la gran col·lecció de funcionalitats i extensions de les que disposa, la gran comunitat d'usuaris disposats a donar suport, la seva llicència opensource i el seu cost nul.

4.2 Base de dades

La majoria d'aplicacions informàtiques han d'accedir a dades externes dipositades en algun tipus de memòria com al nostre cas. Hem de disposar de dades referents a usuaris, inventari teòric, procés d'inventariar, etc.

Com hem comentat en apartats anteriors, implementarem l'aplicació en llenguatge PHP que interactuarà amb la nostra base de dades mitjançant MySQL. El servidor Apache es comunicarà amb la base de dades mitjançant aquest motor per obtenir les dades que necessita la nostra aplicació en PHP per processar-la, fer les accions i procediments pertinents i mostrar-la a l'usuari. Aquest funcionament es pot veure a la següent figura (figura 4.7).

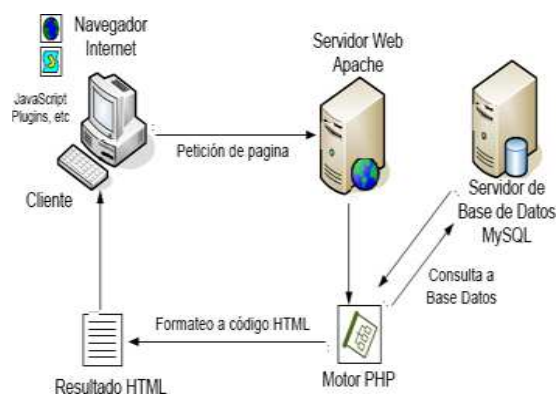


Figura 4.7 Esquema funcionament Apache + MySQL.

4.2.1 Estructura de la base de dades

A continuació presentem el diagrama entitat relació del model relacional de la base de dades dissenyada.

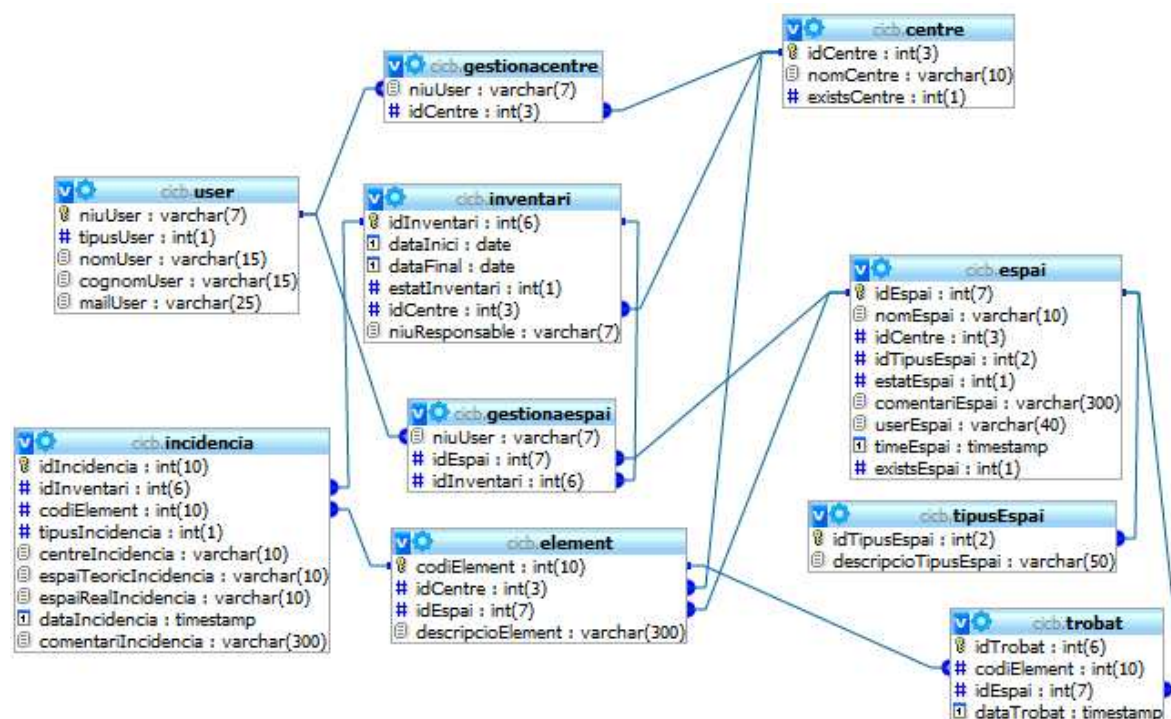


Figura 4.8 Diagrama entitat relació de la nostra base de dades.

4.2.2 Especificació de taules i explicació de camps

A continuació apareixen detallades totes les taules de l'esquema anterior amb els seus respectius camps. Les claus primàries apareixen subratllades amb una línia continua, les claus forànies estan subratllades amb una línia discontinua.

Pel que fa a l'escriptura dels atributs de les taules (així com els atributs de les classes i variables en PHP) hem utilitzat *lower camel case*, és a dir, tots els atributs comencen amb minúscula, en cas de que continguin més d'una paraula, la primera s'escriu en minúscula i les successives la primera lletra en majúscula i sense espais. També s'eviten accents i signes de puntuació. Un exemple de *lower camel case* seria: *aixoEsUnClarExempleDeLowerCamelCase*.

User:

La taula user fa referència a la informació relacionada amb els usuaris que fan servir l'aplicació. Per a cada usuari donat d'alta es crearà una nova entrada en aquesta taula que contindrà els camps:

- niuUser: Número d'identificació universitari de l'usuari.
- nomUser: Nom de l'usuari.
- cognomUser: Cognom de l'usuari.
- tipusUser: Fa referència al perfil al que pertany cada usuari. Com hem explicat en capítols anteriors els usuaris poden pertànyer als grups administradors, responsables o usuaris.
- mailUser: Correu electrònic de l'usuari. S'utilitzarà per realitzar notificacions als usuaris si s'escauen.

- Centre:

La taula centre conté tota la informació relativa als centres que consten a l'inventari teòric. Aquesta taula es genera automàticament quan importem un inventari mitjançant el mòdul d'importació. Per a cada centre es crearà una nova entrada que contindrà la següent informació:

- idCentre: Número identificatiu de cada centre. Val a dir que aquest número l'assigna l'aplicació arbitràriament i en ordre creixent pel que no té cap validesa fora del software. Cada cop que dintre de l'aplicació vulguem fer referència a un centre s'utilitzarà aquest nombre.

- nomCentre: Nom amb el que es coneix el centre. Sol ser un nombre de tres xifres.
- existsCentre: Aquest camp prendrà un valor enter 0 o 1 depenent de si el centre en qüestió existeix a l'inventari actual. Aquest camp ens serà de gran ajuda a l'hora de conservar les màximes dades possibles d'un inventari a un altre.

- Gestionacentre:

Aquesta taula fa de nexa entre les taules user i centre. Ja que la relació entre ambdues taules és de N a N, és a dir, que un usuari pot gestionar varis centres i un centre pot estar gestionat per varis usuaris ens cal una taula intermèdia a l'hora d'implementar aquesta condició d'una manera clara i eficient. Per a cada permís usuari-centre tindrem una entrada en aquesta taula.

- idCentre: Número identificatiu del centre (explicat anteriorment).
- niuUser: Nombre identificatiu universitari de l'usuari (explicat anteriorment).

- Espai:

La taula Espai emmagatzema tota la informació referent a les ubicacions on es troben els elements a inventariar. Per a cada espai trobarem una entrada en aquesta taula que contindrà la següent informació:

- idEspai: Nombre identificatiu de cada espai. Un cop es fa la importació de dades el software assigna automàticament un nombre a cada espai ordenats de manera ascendent. Cada cop que vulguem fer referència a un espai a dintre de l'aplicació, es farà servir aquest nombre.
- nomEspai: Nom de l'espai. Fa referència al nom en text de l'espai, es sol compondre d'una lletra que fa referència a l'edifici on es localitza i uns nombres que indiquen la planta i el número d'espai.
- idCentre: Nombre identificatiu del centre al que pertany l'espai (explicat anteriorment).
- idTipusEspai: Nombre identificatiu de la categoria a la que pertany l'espai (explicat amb més detall a la taula tipusEspai).
- estatEspai: Nombre enter que indica l'estat de l'inventari en aquest espai. Pot prendre tres valors diferents depenent si l'inventari a aquest espai ha acabat, està en curs o encara no s'ha començat.

- **userEspai**: Nombre identificatiu universitari de l'usuari que ha realitzat l'inventari de l'espai si s'escau.
 - **comentariEspai**: Camp de text on els usuaris poden deixar comentaris un cop finalitzat l'inventari d'un espai.
 - **existsEspai**: Aquest camp prendrà un valor enter 0 o 1 depenent de si l'espai en qüestió existeix a l'inventari actual. Aquest camp ens serà de gran ajuda a l'hora de conservar les màximes dades possibles d'un inventari a un altre.
- **tipusEspai**:

Aquesta taula conté totes les categories d'espais que hi ha com per exemple: despatxos, laboratoris, seminaris... Cada categoria tindrà una entrada en aquesta taula amb els següents camps:

 - **idTipusEspai**: Nombre identificatiu únic per a cada categoria.
 - **descripcioTipusEspai**: Cadena de text que conté el tipus de categoria que estem tractant com per exemple: despatx, laboratori, seminari...
 - **GetionaEspai**:

A l'igual que la taula GestionaCentre, aquesta taula té com a objectiu possibilitar la interrelació de les taules user i espai de N a N. Un usuari pot gestionar més d'un centre i un centre pot estar gestionat per més d'un usuari. Aquesta taula conté una entrada per a cada permís usuari-espai i els camps:

 - **idEspai**: Nombre enter identificatiu de cada espai.
 - **niuUser**: Nombre identificatiu universitari de l'usuari.
 - **Element**:

Aquesta taula conté tot l'inventari teòric importat, és a dir, la informació proporcionada que compren el codi de barres identificatiu de l'usuari així com la seva localització teòrica. Aquesta taula conté una entrada per a cada element de l'inventari i presenta els camps següents:

- codiElement: nombre enter del codi de barres de l'element. Aquest nombre és únic i identificatiu per a cadascun dels elements i apareix al codi de barres del mateix.
 - idCentre: Número identificatiu del centre on està teòricament localitzat l'element.
 - idEspai: Nombre identificatiu de l'espai on es troba teòricament situat l'element.
 - descripcioElement: Breu descripció en format text de l'element.
- trobat:

La taula trobat enumera els elements que es van inventariant durant el procés. Per a cada element es crea una entrada a la taula amb la següent informació:

 - idTrobat: Nombre identificatiu de cada registre de la taula trobat.
 - codiElement: Codi de barres de l'element que s'ha trobat.
 - idEspai: Nombre identificatiu de l'espai on s'ha trobat l'element.
 - dataTrobat: Data i hora de quan s'ha trobat l'element.
 - Incidència:

En cas que l'inventari d'un element no tingui èxit ja sigui per una pèrdua de l'element o per un canvi d'ubicació, cal que es creï una incidència. Això comporta una nova entrada en aquesta taula on es guarda la següent informació:

 - Idincidència: Nombre per a identificar cada incidència.
 - idInventari: Nombre identificatiu de l'inventari en que es dona la incidència.
 - codiElement: Codi de l'element en que es dona la incidència.
 - tipusIncidencia: Aquest camp és un nombre que identifica el tipus d'incidència que pot ser una pèrdua d'element o un canvi d'ubicació.
 - centreIncidencia: Descripció del centre on s'ha produït la incidència.
 - espaiTeoricIncidencia: Descripció de l'espai on havia d'estar l'element segons l'inventari teòric.

- **espaiRealIncidencia:** En cas de que la incidència sigui del tipus de canvi d'ubicació, en aquest camp s'emmagatzema la descripció de l'espai on realment es troba l'element.
 - **dataIncidencia:** Data i hora de la creació de la incidència.
 - **comentariIncidencia:** L'usuari té la possibilitat d'afegir un comentari per a cada incidència en format text. La incidència pot descriure qualsevol circumstància ressenyable com per exemple que un element es trobi en mal estat.
- **Inventari:**

Cal que portem un arxiu històric de tots els inventaris que es van realitzant als diferents centres. Així doncs la taula contindrà els següents camps:

 - **idInventari:** Nombre identificatiu per a cada inventari que es realitza.
 - **dataInici:** Data i hora en que s'ha iniciat l'inventari.
 - **dataFinal:** Data de finalització de l'inventari si s'escau.
 - **idCentre:** Número identificatiu del centre del que es realitza l'inventari.
 - **estatInventari:** Aquest camp ens indica l'estat actual de l'inventari, si està finalitzat o està en curs.
 - **niuResponsable:** Número identificatiu universitari de l'usuari responsable de l'inventari.

4.3 Diagrames de casos d'ús

A continuació és presenten els diagrames de casos d'ús per a cadascun dels perfils d'usuari. Hem agrupat algunes de les funcionalitats que si bé son independents entre si el procés que desenvolupen és similar.

Els tres perfils d'usuari: administrador, responsable i usuari comparteixen funcionalitats d'una manera jeràrquica. Les funcionalitats a les que té accés un usuari també són accessibles per a un responsable. El responsable però té algunes funcionalitats no accessibles per a un usuari. La mateixa relació es dona entre els perfils administrador i responsable, les funcionalitats del responsable també estan habilitades per a l'usuari administrador però l'usuari administrador disposa d'unes funcionalitats exclusives que només pot realitzar un usuari pertanyent a aquest perfil.

Als diagrames de casos d'ús hem tingut en compte això i no repetirem casos d'ús d'un perfil a l'altre.

El diagrama de casos d'ús per a l'usuari és el següent:

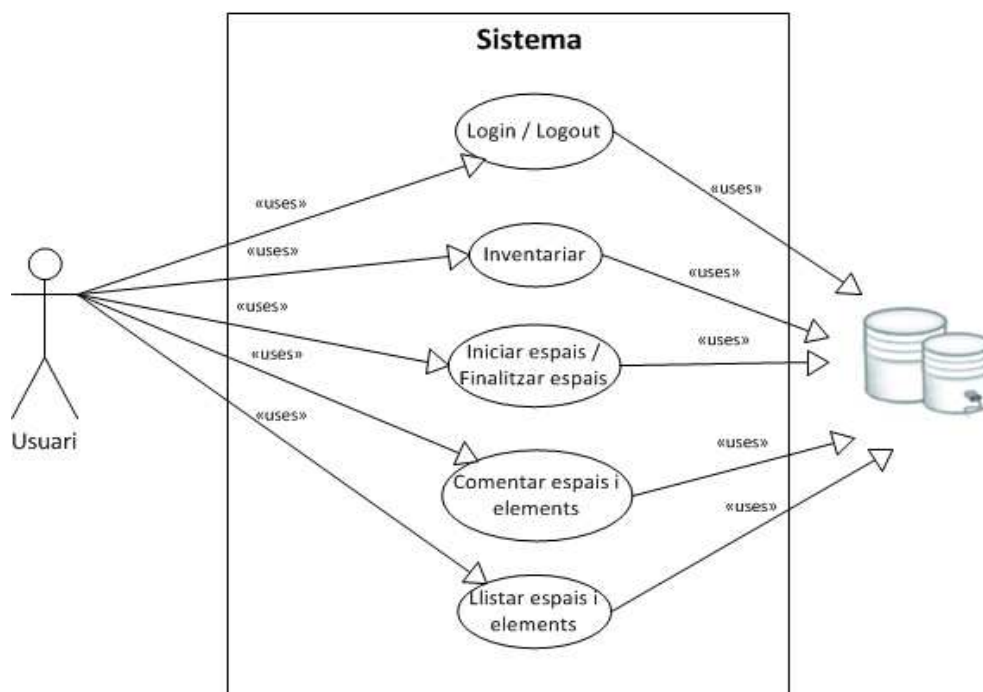


Figura 4.9 Casos d'ús per a l'usuari.

Com hem explicat, els casos d'ús comentats al diagrama anterior s'omiteixen al diagrama de casos d'ús del perfil d'usuari responsable.

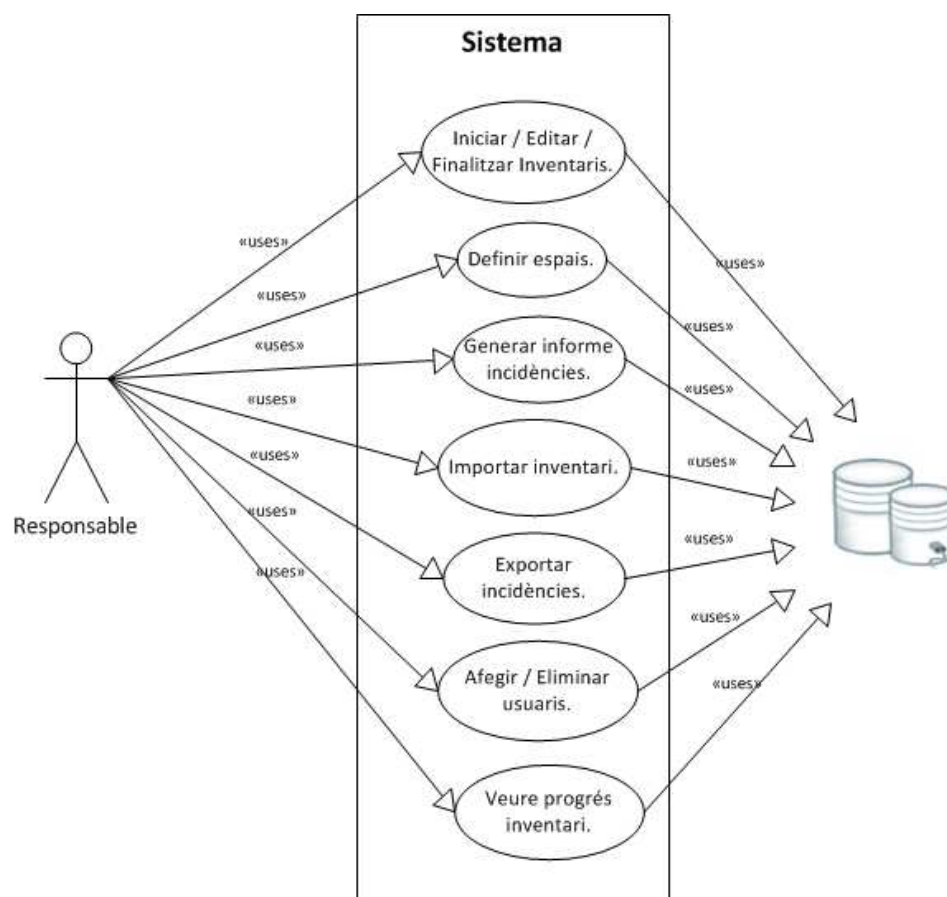


Figura 4.10 Casos d'ús per al perfil responsable.

De la mateixa manera s'ometen aquestes funcionalitats al diagrama de casos d'ús referent a l'usuari administrador que es presenta a continuació.

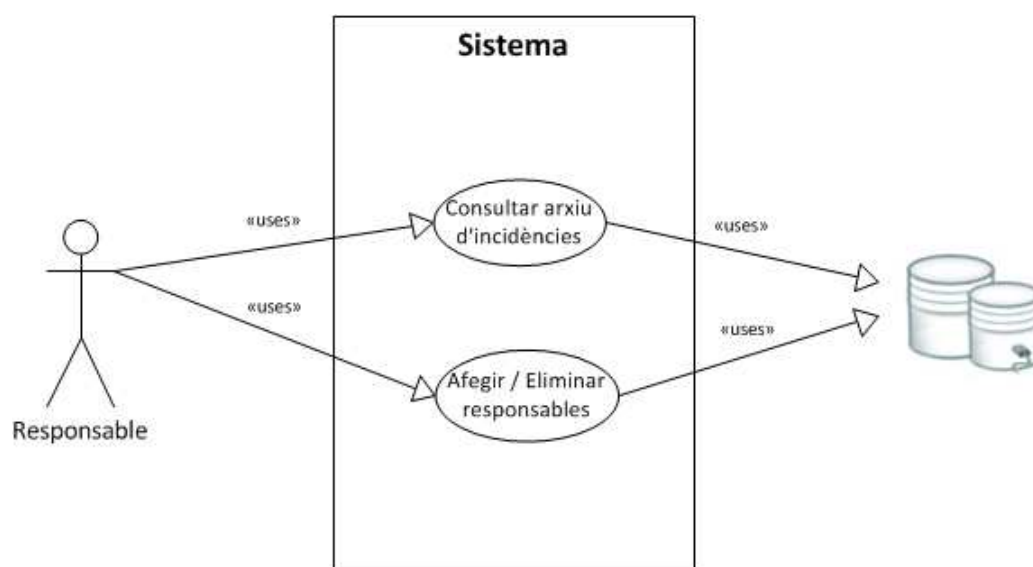


Figura 4.11 Casos d'ús per al perfil administrador.

4.4 Diagrames de seqüència

A continuació presentem els diagrames de seqüència per a les funcionalitats més importants de l'aplicació: inventari, importar, informe d'incidències i exportar.

Primerament el diagrama per al cas d'ús inventariar queda de la següent manera (figura 4.12):

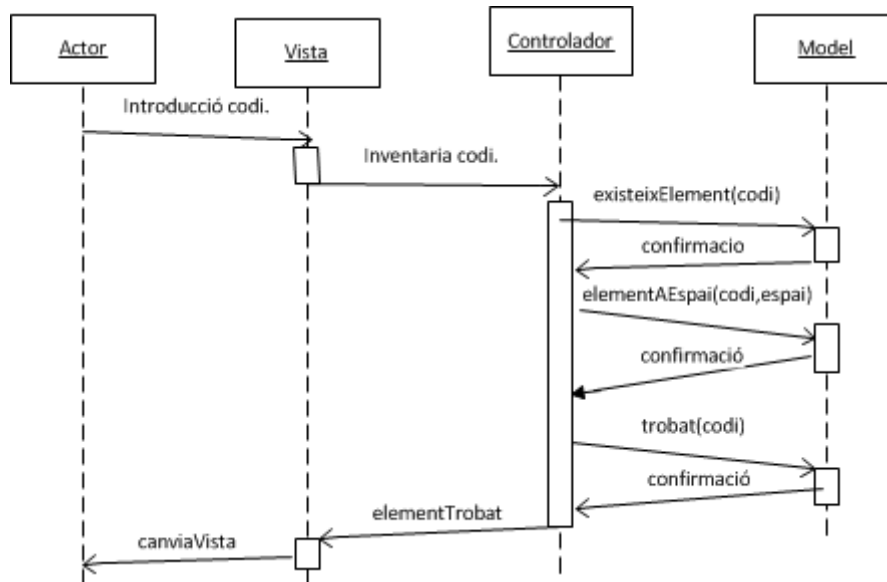


Figura 4.12 Diagrama de seqüència del cas d'ús inventari.

L'usuari introdueix el codi mitjançant la pistola lectora de codis de barres. Tot seguit el sistema s'encarrega de realitzar les comprovacions pertinents, comprovar que l'element existeix a l'inventari i que es troba a l'espai que s'estigui inventariant en aquell moment. En cas de que alguna d'aquestes comprovacions falli el sistema s'encarrega d'avisar a l'usuari i de realitzar els processos corresponents per a generar la incidència.

A continuació mostrem el diagrama de seqüència referent al cas d'ús importar (figura 4.13).

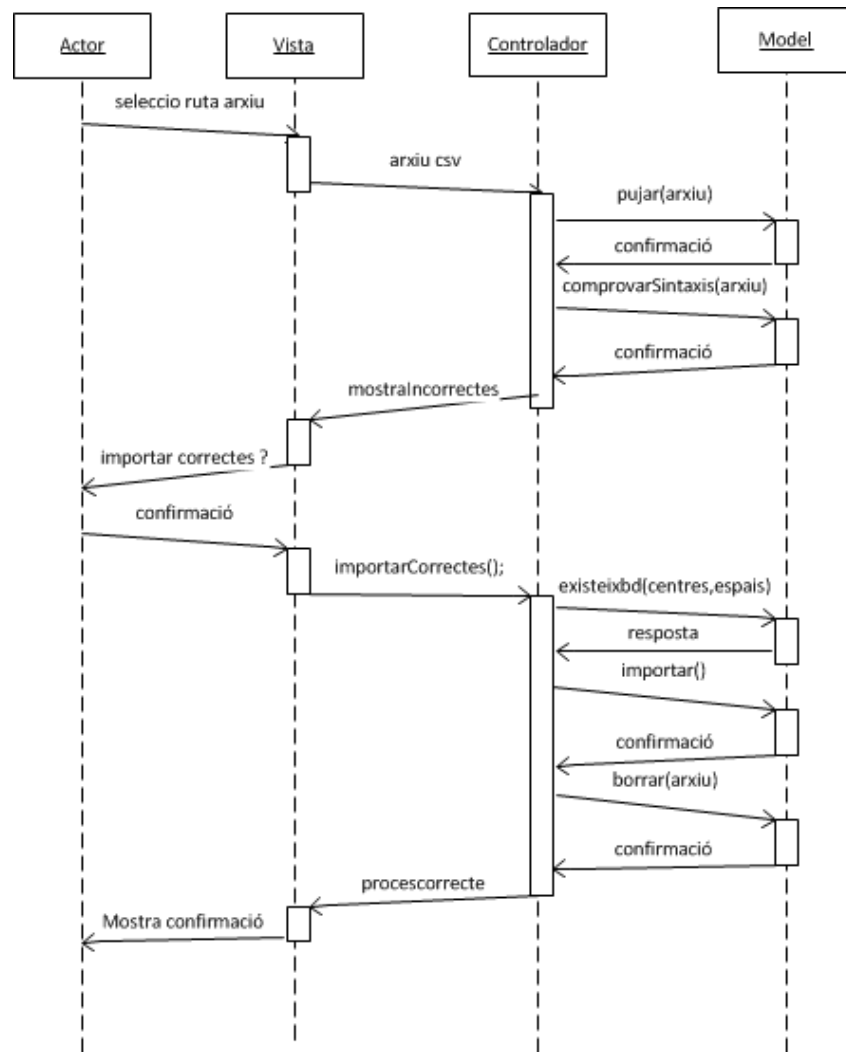


Figura 4.13 Diagrama de seqüència del cas d'ús importar.

Quan un administrador o responsable de centre vol importar informació per a realitzar l'inventari primerament ha d'introduir la ruta local on es troba el fitxer .csv amb l'estructura detallada en l'apartat 3.3.3. Tot seguit el sistema pujarà l'arxiu al servidor per a treballar més àgilment amb les dades i comprovarà la seva sintaxi. Si existeix alguna línia no vàlida el sistema ho notificarà a l'usuari amb el detall de la línia errònia i donarà l'opció de cancel·lar la importació per a la reparació manual del fitxer. Quan l'usuari prossegueixi amb el procés, el sistema importarà a la base de dades tota la informació referent als centres i espais nous i finalment la referent a tots els actius de l'inventari.

Mostrem tot seguit el diagrama de seqüència per al cas d'ús referent a la generació de l'informe d'incidències (figura 4.14).

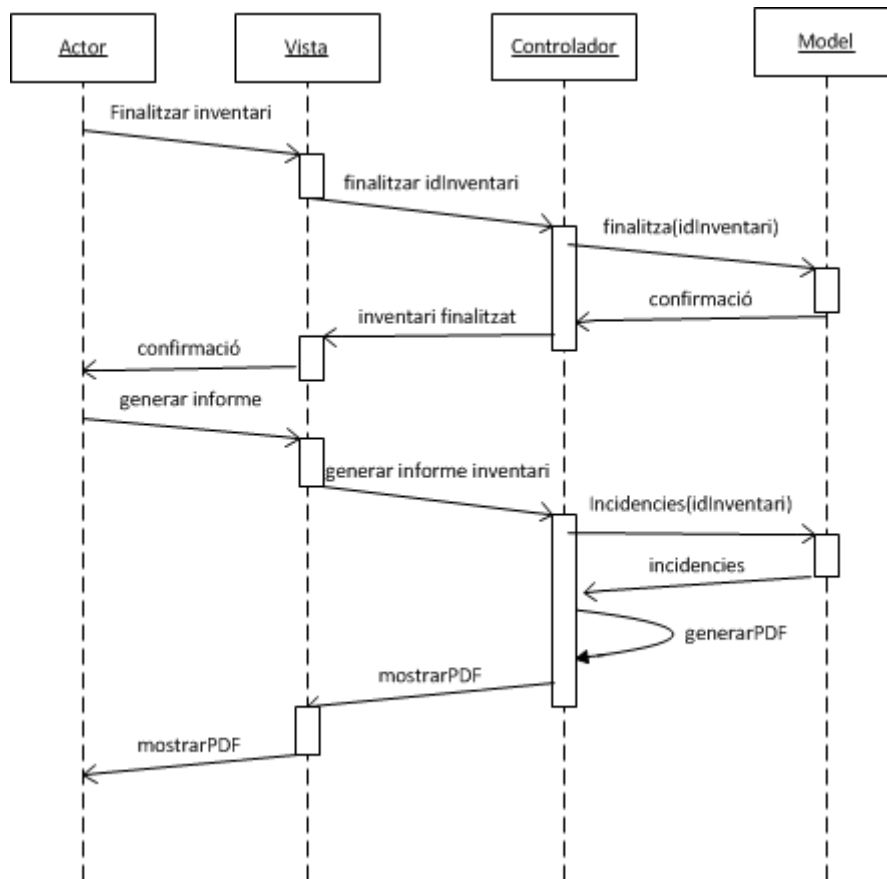


Figura 4.14 Diagrama de seqüència del cas d'ús informe d'incidències.

Un cop finalitzat l'inventari d'un centre el sistema dona l'opció de generar en format PDF un informe sobre les anomalies que s'han detectat durant el transcurs del mateix. En aquest informe hi consta tant informació relativa a l'inventari (data d'inici, data de final, nom del centre, responsable de l'inventari, etc.) com la informació referent a les incidències (canvis d'ubicació, elements perduts, comentaris, etc.).

Finalment presentem el diagrama de seqüència del cas d'ús exportar (figura 4.15).

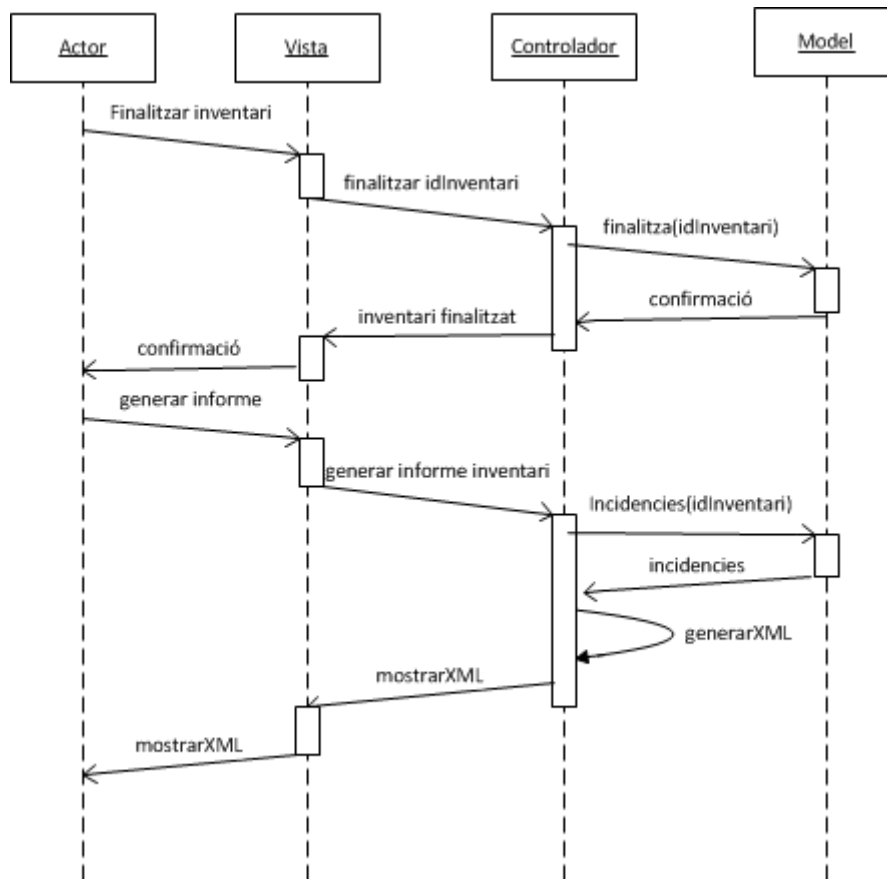


Figura 4.15 Diagrama de seqüència del cas d'ús exportar.

El procés que segueix l'aplicació per a generar el fitxer d'exportació és similar al del cas d'ús de la generació de l'informe d'incidències. En aquest cas però, no s'exporta tanta informació relativa a l'inventari sinó que només s'inclou el nom del centre i les incidències.

4.5 Disseny d'interfícies

4.5.1 Introducció

Dissenyar correctament les interfícies esdevé un tema importantíssim a l'hora de implementar una aplicació accessible i usable pels usuaris.

Entenem per accessibilitat web la capacitat de la web per a ser accessible a tots els usuaris, des d'usuaris amb discapacitats sensibles fins a fer una web accessible per a tot tipus de navegadors i terminals. Avui en dia el mercat ofereix diferents solucions per a fer accessibles les pàgines web a persones sordes o cegues. Tot i això cal que la semàntica de la web estigui correctament definida per a que aquestes solucions siguin efectives. Aconseguirem aquest objectiu codificant els llenguatges de l'aplicació segons els estàndards vigents definits per la

Word Wide Web Consortium (W3C). Validarem tot el codi de l'aplicació i hi afegirem el segell que acredita que tota la pàgina web està correctament implementada sota els estàndards (figura 4.16).



Figura 4.16 Segell acreditatiu W3C.

La majoria de navegadors s'adapten als estàndards en, pràcticament, tota la seva totalitat. Tot i això també existeixen navegadors que creen el seu propi estàndard de facto com és el cas de IE (*Internet Explorer*). Nosaltres implementarem l'aplicació basant-nos en els estàndards de la W3C i no en les normes pròpies instaurades per les diferents versions de IE. Per al desenvolupament i proves utilitzarem els navegadors Mozilla Firefox, Opera i Google Chrome que són els que millors resultats obtenen al test Acid. Concretament els dos últims són els únics navegadors comercialment estesos i en versió estable que superen el test obtenint una puntuació de 100/100 [14].

Com hem comentat, l'accessibilitat és un tema capital a l'hora d'implementar una aplicació al web. De l'accessibilitat depèn la correcta visualització de l'aplicació als diferents navegadors i dispositius. Tot i això s'han de tenir en compte altres factors quan implementem interfícies per a usuari, una de les més importants és la usabilitat. La usabilitat es defineix com la facilitat d'ús de la web per part dels usuaris, la facilitat d'accedir a l'eina d'una manera còmode. Aquest concepte incideix en la relació sistema-persona i la podríem definir en poques paraules com fer una web intuïtiva.

Com hem comentat a l'apartat 3.3.1, l'aplicació està dissenyada per a ser utilitzada en un netbook, concretament hem pres com a model un Asus Eee PC amb una resolució de 1024 x 600 que és la resolució més utilitzada en aquest tipus de dispositius.

Tenint en compte tots aquets factors hem realitzat el disseny de les principals "pantalles" que veurà l'usuari. Les més importants les descrivim a l'apartat 4.5.2.

4.5.2 Esbossos d'interfícies

Tenint en compte tot lo citat anteriorment hem fet l'esbós de les principals pantalles amb les que haurà d'interactuar l'usuari quan utilitzi l'aplicació.

E.1. Estructura de la pàgina.

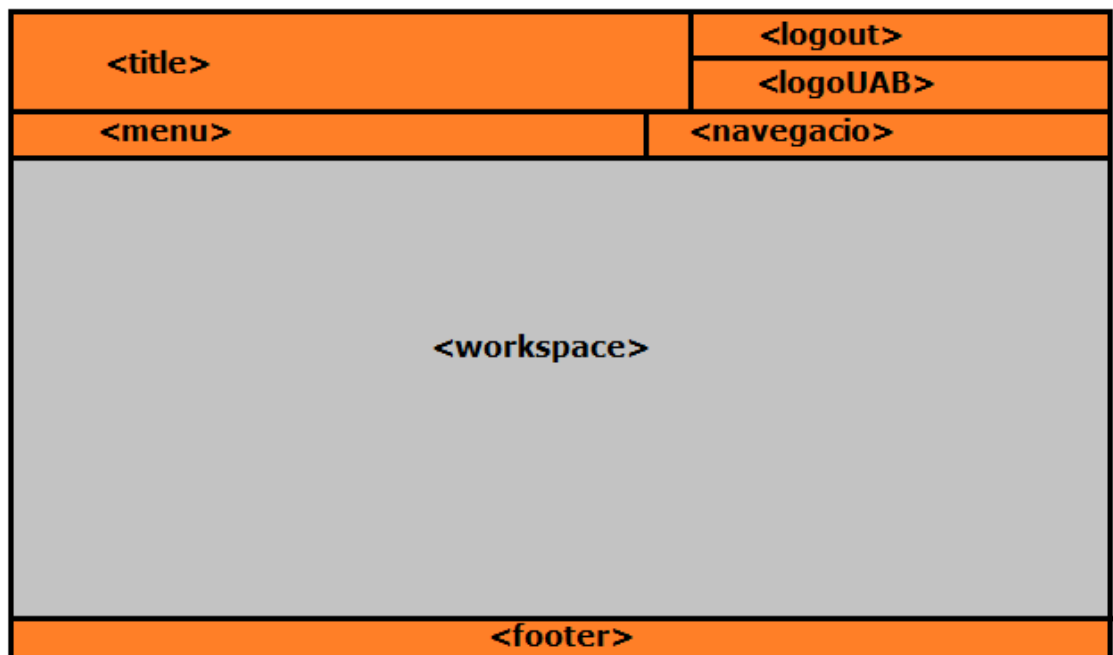


Figura 4.17 Estructura de la pàgina.

Podem observar l'estructura de la pàgina (figura 4.17). Hem optat per un disseny força clàssic a la vegada que accessible i intuïtiu. A la part superior de la pàgina es pot veure el títol o *title* així com informació de l'usuari loguejat i l'enllaç per a fer *logout*. Hem insertat també el logotip de la Universitat Autònoma de Barcelona en aquesta part superior de la pàgina.

Justament sota del títol hem insertat el menú de navegació per les diferents funcionalitats de l'aplicació. Aquest menú serà personalitzat per a cada tipus d'usuari, és a dir, només es mostraran les opcions a les que té accés cada perfil d'usuari. Un usuari del perfil administrador tindrà accés al submenú d'administració i apareixeran les opcions per a les que té privilegis. Tanmateix un usuari del tipus responsable tindrà accés a alguna d'aquestes opcions, per últim un usuari bàsic només tindrà accés a les opcions d'inventari.

A la dreta d'aquest menú hi apareix el menú de navegació pels diferents centres i espais als que té accés l'usuari. Ho implementarem mitjançant menús desplegable. Hem decidit posar aquest menú aquí per a que sigui accessible des de qualsevol funcionalitat de la pàgina ja que els apartats *title*, *logout*, logotip UAB, menú i menú de navegació es mostraran sigui quina sigui la localització de l'usuari dins l'aplicació.

A continuació hi tenim l'espai de treball pròpiament dit, és a dir, el lloc on es mostrarà tota la informació referent a inventaris i administració que sol·licitin els usuaris.

Finalment a la posició inferior del disseny s'hi troba el *footer*. Al *footer* hi trobarem informació referent a l'autoria del projecte, la llicència del software i els segells que certifiquen que la web compleix els estàndards de la W3C descrits a l'apartat 4.5.1.

E.2. Pantalla de benvinguda.

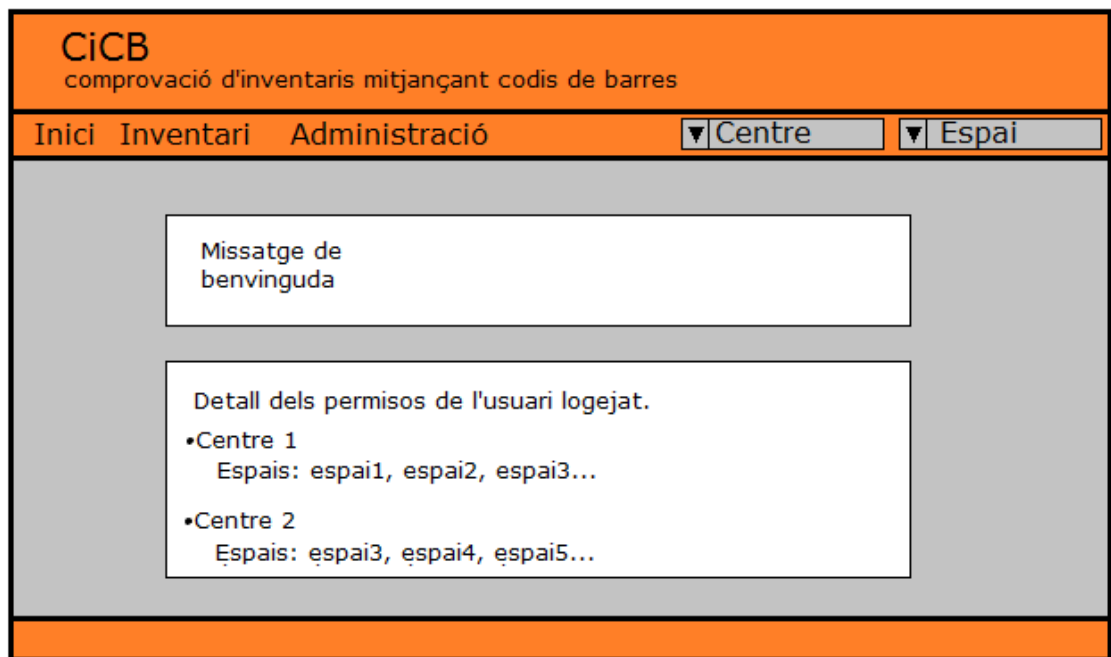


Figura 4.18 Pantalla de benvinguda a l'aplicació.

Aquesta pantalla ens apareixerà tot just ingressar el nostre NIU i contrasenya (figura 4.18). En ella podem veure un missatge de benvinguda així com un resum dels permisos que té l'usuari per inventariar els diferents centres i espais. També mostrarem el perfil al que pertany l'usuari. Cal dir que el disseny de la pàgina canviarà segons el perfil d'usuari.

Podem veure amb una mica més de detall els menús. Pel que fa al menú de navegació dintre de l'aplicació podem veure un exemple del que veuria un usuari pertanyent al perfil administrador o responsable de centre. Al menú de navegació pels centres i despatxos podem veure amb més detall el descrit a l'apartat anterior amb els dos menús desplegable.

E.3. Pantalla d'Inventari.

L'opció més utilitzada a l'aplicació serà sens dubte la d'inventariar el material. Hem ideat un sistema molt visual per a localitzar ràpidament els elements trobats i no trobats dins d'un espai com es pot veure a la imatge següent (figura 4.19).

CiCB		usuari [logout]	
comprovació d'inventaris mitjançant codis de barres		Logo UAB	
Inici	Inventari	Administració	
		▼ Centre	▼ Espai
Codi		Descripció	
codiElement 1		descripcióElement 1	
codiElement 2		descripcióElement 2	
codiElement 3		descripcióElement 3	
codiElement 4		descripcióElement 4	
codiElement 5		descripcióElement 5	
codiElement 6		descripcióElement 6	
codiElement 7		descripcióElement 7	

Figura 4.19 Pantalla d'inventari.

Els elements marcats en vermell representen els elements encara no inventariats, conforme es vagin inventariant s'aniran posant de color verd. Com hem indicat anteriorment aquest senzill mètode permetrà a l'usuari identificar ràpidament els elements que no trobats.

E.4. Realitzant l'inventari.

Un cop iniciat l'inventari d'un espai ens anirà apareixent una finestra *pop-up* on introduir el codi. Això comportarà un important avenç en usabilitat ja que l'usuari no haurà de tocar en cap moment el ratolí del *netbook* quan es realitzi l'inventari. Quan entri en un espai, podrà dipositar el portàtil en qualsevol lloc i anar inventariant els elements amb la pistola lectora de codis de barres inalhàmbrica sense estar pendent del *netbook*.

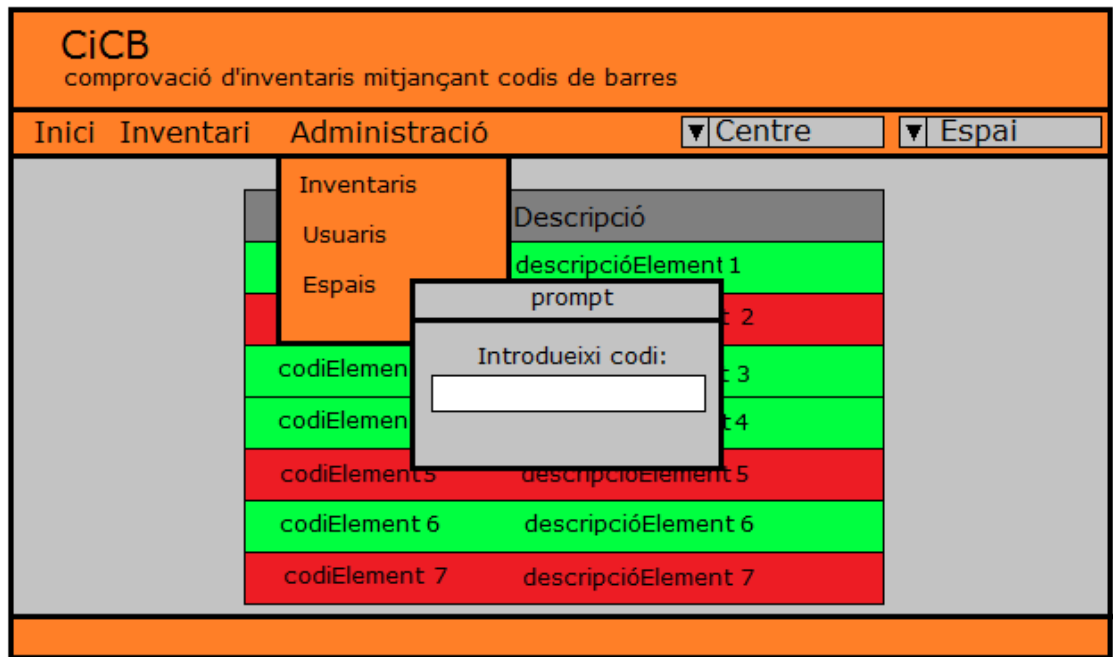


Figura 4.20 Aparència de l'aplicació quan es realitza l'inventari.

A la pantalla anterior (figura 4.20) podem observar també com estan estructurats els submenús d'administració mitjançant pestanyes desplegable al passar per sobre de "Administració".

4.6 Especificació de mòduls

A continuació comentem els diferents mòduls que conformen l'aplicació i els relacionem amb els requeriments funcionals especificats a l'apartat 3.2.

4.6.1 import

Aquest mòdul s'encarrega de la importació de dades relatives a l'inventari d'actius, centres i espais mitjançant un arxiu csv. Realitza la comprovació de sintaxis de l'arxiu i consulta a l'usuari si vol prosseguir. Si és així importa a la base de dades la informació continguda al fitxer. Aquest mòdul és totalment configurable mitjançant l'arxiu de variables globals per a modificar el format d'arxiu que accepta. Aquest mòdul està relacionat amb el requeriment funcional 2 (importació de dades).

4.6.2 list

Aquest mòdul s'encarrega de la visualització dels elements d'un espai quan s'inventarien. També és competència d'aquest mòdul que els elements trobats es marquin de color verd i els no trobats en vermell per a fer més intuïtiu el procés d'inventari. Aquest mòdul cobreix part dels requeriments funcionals 3 i 4 (classificació i identificació d'elements).

4.6.3 inventory

Aquest component s'encarrega de realitzar les comprovacions necessàries quan s'introdueix un codi d'un element. Comprova que l'element existeixi a la base de dades i posteriorment que es trobi al mateix espai teòric que el que estem inventariant. En cas de que no sigui així realitza les comprovacions necessàries per a identificar de quin tipus d'incidència es tracta i generar-la. També s'encarrega dels comentaris a espais i elements. Aquest mòdul es relaciona amb els requeriments funcionals 4 i 5 (identificació d'elements i comentaris).

4.6.4 login

El mòdul login s'encarrega d'identificar l'usuari mitjançant el mòdul CAS de la Universitat i de desloguejar-lo si en el transcurs d'un període de 5 minuts no hi ha activitat. Aquest interval de temps és totalment configurable a l'arxiu de variables globals de l'aplicació. Aquest mòdul cobreix el requeriment funcional 1 (identificació dels usuaris).

4.6.5 admincp

El mòdul admincp s'encarrega de tota la gestió per part dels responsables i administradors dels diferents centres, espais i usuaris. Afegir usuaris, definir pseudònims per a centres, definir categories d'espais, etc. Són competències d'aquest component. Aquest mòdul es relaciona amb el requeriment funcional 9 (visualització del procés d'inventari).

4.6.6 export

Aquest mòdul s'encarrega d'exportar les incidències en format XML per a possibles millores futures del sistema de inventariat. Mitjançant aquest sistema facilitarem enormement la interacció de la nostra aplicació amb altres incloses en el procés de inventariat. Aquest mòdul es relaciona amb el requeriment funcional 7 (generació del fitxer de sortida).

4.6.7 incidence

Aquest mòdul s'encarrega de tota la gestió de les incidències, de la visualització de l'arxiu històric i de la generació de l'informe d'incidències per a un inventari. Aquest mòdul està relacionat amb els requeriments funcionals 6 i 8 (informe i arxiu històric d'incidències).

4.6.8 welcome

Mòdul de benvinguda que s'encarrega de la informació que es mostra als usuaris quan es loguegen. Depenent del seu perfil d'usuari aquest mòdul mostrarà informació referent al procés dels inventaris iniciats, o als permisos que té un usuari en concret.

4.6.9 nav

Finalment aquest mòdul s'encarrega de generar correctament el menú de navegació dels espais i centres en forma de pestanyes desplegable. Aquest mòdul està relacionat amb el requeriment funcional 3 (classificació dels elements).

Capítol 5: Codificació

5.1 Patró de desenvolupament: Model Vista Controlador

5.1.1 Introducció

Per a desenvolupar un software de forma sistemàtica cal una metodologia correctament definida, és a dir, un patró de desenvolupament. Els patrons de desenvolupament s'engloben dintre de la disciplina de l'arquitectura de software i n'hi ha de diferents tipus com per exemple el RUP (*Rational Unified Process*), o el XP (*Extreme Programming*). Nosaltres ens hem decantat finalment pel patró de desenvolupament MVC (Model Vista Controlador). De les architectures de software que hem analitzat, creiem que és la que s'adapta millor a un projecte web, a més com que personalment ja he treballat amb aquesta arquitectura a alguns projectes anteriors, la conec força bé i no caldrà dedicar recursos ni temps addicional en l'aprenentatge d'una arquitectura nova.

L'arquitectura MVC es centra principalment en aconseguir independència entre el model o dades de l'aplicació i la presentació o vista. La nostra aplicació web, com pràcticament la totalitat de les mateixes, sustenta tota la seva informació en un SGBD (sistema gestor de bases de dades). A partir d'aquesta base de dades, on s'emmagatzema tota la informació referent a inventari, usuaris, etc.,. Muntarem la nostra aplicació basant-nos en un sistema de capes.

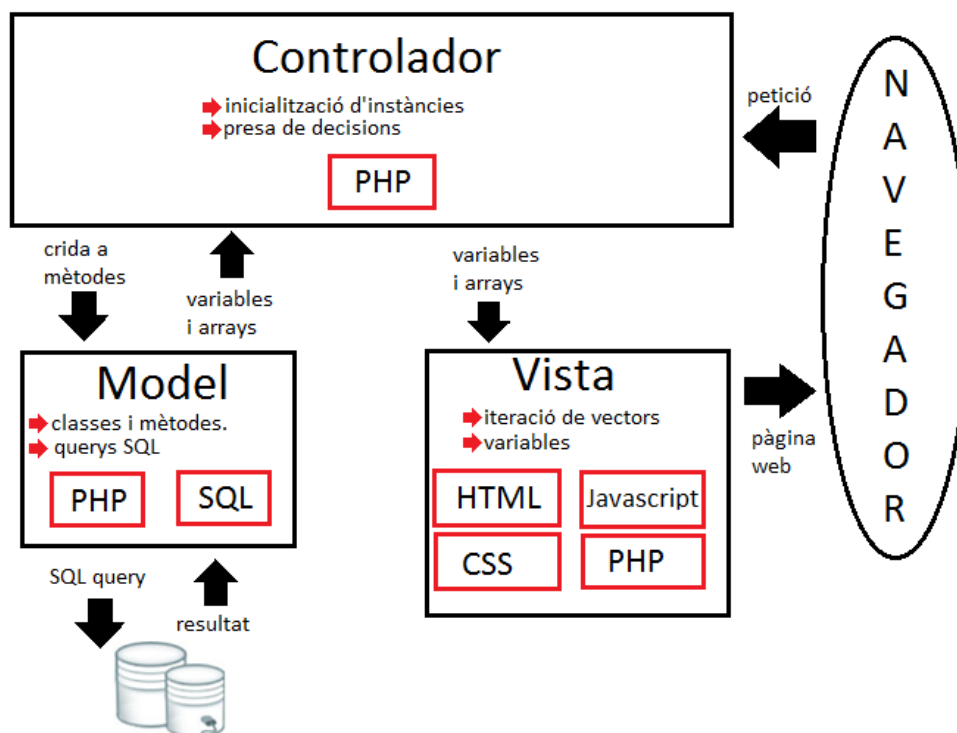


Figura 5.1 Funcionament de l'arquitectura MVC

A la figura 5.1 es pot veure el funcionament de l'arquitectura MVC. La petició HTTP (*Hypertext Transfer Protocol*) és atesa pel controlador, i és el mateix controlador el que s'encarrega de demanar la informació al model per a que, mitjançant instàncies de les nostres classes, interactuï amb la base de dades. Un cop obtenida aquesta informació relativa als usuaris, inventari, etc. El controlador la gestiona i la passa a la vista mitjançant vectors i variables. La vista només cal que s'encarregui de la presentació de la informació. També podem observar els diferents llenguatges que s'utilitzen en cada fase de l'arquitectura. Mentre que al model i al controlador s'utilitza PHP i SQL, a la vista s'utilitzen tots els llenguatges relatius a la presentació i a la interacció de l'usuari amb l'aplicació: XHTML, CSS i Javascript.

5.1.2 Model

El model és la capa de l'aplicació que interactua directament amb el SGDB. Ho fa mitjançant classes i POO (programació orientada a objectes). Definim les classes que ens calgui, normalment, i com a sistema orientatiu tindrem, com a mínim, una classe per a cada taula de la nostra base de dades relacional.

Un cop hem decidit quines classes implementarem, cal que dissenyem els mètodes d'aquestes classes. Els mètodes seran les accions a executar per un objecte o instància de la nostra classe. Aquests mètodes són els que interactuaran directament amb el nostre SGDB mitjançant MySQL. Per exemple, nosaltres hem implementat una classe anomenada centre. Com a la taula centre de la nostra base de dades (apartat 4.2.1 i 4.2.2) la classe té els mateixos atributs `idCentre`, `nomCentre` i `existsCentre`. La principal acció que ha de realitzar aquesta classe és la de possibilitar la navegació pels centres als menús, llistar centres, comprovar si hi ha un inventari iniciat per a un centre determinat... Per tant implementarem els mètodes necessaris per a dur a terme aquestes tasques.

```
1 <?php
2 class centre{
3     private $idCentre;
4     private $nomCentre;
5
6     function getIdCentre(){
7
8     }
9
10    function getNomCentre(){
11
12    }
13
14    function load($idCentre){
15
16    }
17
18    function listCentresByNiu($niu){
19
20    }
21
22    function listCentres(){
23
24    }
25
26    function existInventari(){
27
28    }
29 }
30 ?>
```

Figura 5.2 Classe centre.

La majoria de classes contenen una sèrie de mètodes genèrics per a facilitar la seva gestió. És el cas dels *setters*, els *getters*, l'*upload* i el *load*. Els mètodes *load* i *upload* s'encarreguen de carregar i descarregar la informació de la base de dades amb la instància de la nostra classe per a crear un objecte o guardar la informació amb la que hem treballat. Els *getters* i *setters*, en canvi, s'utilitzen per a modificar els atributs del nostre objecte. Així doncs per a modificar un element de la nostra base de dades primerament hauríem de crear una instància de la nostra classe, "carregar-hi" a dins la informació amb el mètode *load*, modificar-la amb el *setter* i tornar-la a carregar a la base de dades amb el mètode *upload*. En el cas de voler obtenir algun tipus d'informació el procés seria similar però no caldria cridar al mètode *upload*.

A la nostra classe d'exemple, la classe centre, mai cal modificar aquesta taula a la base de dades. L'únic moment on caldria és a la importació d'informació, però d'aquest procés se n'encarrega una classe específica per a importació.

A continuació podem veure el codi relatiu a un dels mètodes de la classe centre (figura 5.3).

```
function existInventari(){
    $mysql = new mysql(); //inicialització objecte mysql
    $sql="SELECT idInventari FROM inventari WHERE
        estatInventari=1 AND idCentre='$this->idCentre'"; // query SQL
    if ($mysql->num_rows($mysql->query($sql)) > 0) return TRUE; //consulta a la BD
    return FALSE;
}
```

Figura 5.3 Mètode existsInventari de la classe centre.

Com podem observar, i com hem comentat anteriorment, són els mètodes qui s'encarreguen de fer les peticions al nostre SGBD, obtenir-ne el resultat i retornar-lo al controlador.

5.1.3 Vista

La vista s'encarrega de rebre la informació ja preparada des del controlador per a mostrar-la a l'usuari i estructurar-la degudament mitjançant l'XHTML. Com a norma general, a la vista utilitzarem com menys llenguatge PHP millor. Així doncs utilitzarem principalment XHTML a més de CSS per a millorar estèticament l'estructura de la pàgina i el javascript per a automatitzar tasques monòtones per a l'usuari i donar-li dinamisme visual a la web.

Les funcions i estructures PHP més utilitzades a la vista seran els *foreach* per a recórrer vectors i la funció *echo* per a mostrar resultats a la pàgina. A continuació tenim una de les vistes de l'aplicació, concretament la vista que ens mostra els usuaris registrats (figura 5.4).

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" 'http://www.w3.org/
2 <html xmlns='http://www.w3.org/1999/xhtml'>
3 <head>
4     <meta http-equiv='Content-Type' content='text/html; charset=utf8' />
5     <link rel='stylesheet' type='text/css' href='css/style.css'/>
6     <title>CiCB - comprovaci&oacute; d'inventari amb codis de barra</title>
7 </head>
8 <body>
9 <p>Benvingut <?php echo $_SESSION['nomUser']. " ".$_SESSION['cognomUser'];?>.</p>
10
11 <p><a onclick="javascript:window.open('newUser/controladorNewUser.php','','scroll
12 <table>
13     <tr>
14         <td>Niu</td><td>Nom</td><td>Cognom</td><td>Tipus</td><td>Accions</td>
15     </tr>
16     <?php foreach ($niuArray as $niu):?>
17         <?php $user->load($niu);?>
18         <tr>
19             <td><?php echo $user->getNiuUser();?></td>
20         </tr>
21     <?php endforeach;?>
22 </table>
23 </body>
24 </html>
```

Figura 5.4 Vista *listarUsuarios*.

Podem veure a la captura de pantalla com utilitzem l'XHTML per a estructurar la pàgina. Utilitzem el mínim possible el PHP i sempre que podem no utilitzarem crides a mètodes, tot i això a vegades és inevitable quan utilitzem alguns bucles *foreach*.

5.1.4 Controlador

El controlador s'encarrega de rebre les peticions de l'usuari. La seva funció més important és la presa de decisions tenint en compte el que demana l'usuari. Un cop discernida la funció requerida, el controlador inicialitza les instàncies corresponents i mitjançant les classes i els seus mètodes del model aconsegueix la informació necessària. Després s'encarrega de que la informació estigui emmagatzemada en estructures de dades còmodes, principalment en

variables i vectors. Finalment les passa a la vista per a que aquesta generi la pàgina corresponent.

En la següent captura (figura 5.5) es mostra un controlador de l'aplicació:

```
9      $idCentre=$_REQUEST['idCentre'];
10     $defaultCentre = new centre();
11     $arrayCentres=$defaultCentre->listCentres();
12     if (isset($idCentre)){
13         $defaultCentre->load($idCentre);
14         $idEspai=$_REQUEST['idEspai'];
15         $defaultEspai = new espai();
16         $arrayEspais=$defaultEspai->getEspaisByCentre($idCentre);
17         if (isset($idEspai)){
18             $defaultEspai->load($idEspai);
19             $action=$_REQUEST['action'];
20             switch ($action){ //presa de decisions
21                 case "finalitzar":
22                     $defaultEspai->setEstatEspai(2); //finalitzat
23                     $defaultEspai->upload();
24                     $arrayEspais=$defaultEspai->getEspaisByCentre($idCentre);
25                     $defaultEspai->load($idEspai);
26                     $incidencia = new incidencia();
27                     $incidencia->finalitzarEspai($idEspai);
28                     break;
29                 case "iniciar":
30                     $defaultEspai->setEstatEspai(1); //iniciat
31                     $defaultEspai->upload();
32                     $arrayEspais=$defaultEspai->getEspaisByCentre($idCentre);
33                     $defaultEspai->load($idEspai);
34                     break;
35             }
36             $estat=$defaultEspai->getEstatEspai();
37             $element = new element();
38             $arrayElements=$element->getElementsByEspai($idEspai);
39         }
40     }
41     require 'vistaList.php'; //crida a la vista
42 }
43 ?>
```

Figura 5.5 Controlador espais.

Com es pot observar al controlador d'espais, la presa de decisions es fa en base a una variable anomenada *\$action*. Aquesta variable és la que utilitza el controlador per a saber que està demanant l'usuari. Mitjançant un *switch* es defineixen totes les accions a fer per a cadascuna de les accions. Finalment es fa la crida a la vista.

5.2 Estructura de directoris

A continuació descrivim l'estructura de directoris emprada a l'aplicació (figura 5.6).

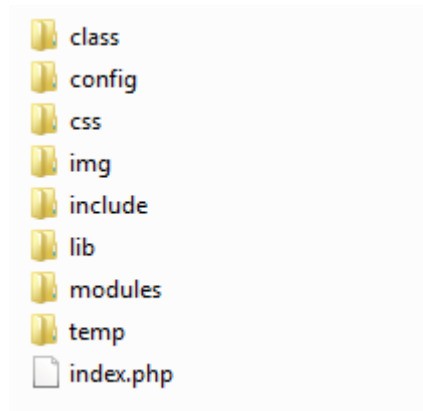


Figura 5.6 estructura de directoris.

- class: A dintre d'aquest directori s'hi troben totes les classes que hem fet servir a l'aplicació. El nom dels fitxers ve donat pel model nom.class.php.
- config: Dins d'aquest directori s'hi troben els arxius de configuració utilitzats per a la instal·lació de l'aplicació a un servidor Apache.
- css: Carpeta que conté tots els fitxers relatius a les fulles d'estil css.
- img: directori per a emmagatzemar imatges que fem servir a la vista per a fer més atractiva les interfícies.
- include: Totes les capçaleres XHTML i el codi que es repeteix a la majoria de pàgines de l'aplicació es troben en aquest directori.
- lib: Llibreries externes utilitzades a l'aplicació com el jQuery.
- modules: En aquest directori trobem tots els mòduls de l'aplicació. Cada mòdul està contingut en una o més subcarpetes. Cada mòdul disposa de com a mínim dels fitxers vistaModul.php i controladorModul.php.
- temp: carpeta per a fitxers temporals utilitzada a la importació de dades a través d'un fitxer extern.

5.3 Tecnologies utilitzades

A continuació citem tots els llenguatges emprats a l'aplicació. No és l'objectiu d'aquesta secció descriure'ls ja que ja s'ha fet en capítols anteriors (5.1.1). Explicarem per a què els hem utilitzat dins del desenvolupament del software.

5.3.1 PHP

Tota la lògica del programa està escrita en PHP. Els controladors i les classes estan totalment implementades en aquest llenguatge a excepció d'alguns components escrits en javascript per a la redirecció a pàgines.

5.3.2 XHTML

Hem utilitzat el llenguatge XHTML per a implementar les vistes. El llenguatge XHTML s'utilitza per a mostrar la informació de l'aplicació a l'usuari mitjançant una interfície gràfica.

5.3.3 CSS

Els fulls d'estil CSS els hem emprat per a fer més atractives les interfícies amb les que interactua l'usuari.

5.3.4 Javascript

El javascript ha estat utilitzat sobretot per a l'autoemplenat de formularis en algunes funcionalitats de l'administrador i responsable. Això facilita la tasca d'aquests usuaris ja que a l'aplicació hi ha formularis molt llargs i monòtons d'emplenar.

5.3.5 AJAX i jQuery

La utilització de l'AJAX i el jQuery s'ha fonamentat sobretot en millorar l'experiència de l'usuari pel que fa a estètica. Introduint en alguns casos efectes en l'aparició i desaparició d'alguns components.

5.4 Aspectes interessants de la codificació

A continuació citem alguns dels aspectes més interessants del procés de codificació. Tècniques utilitzades per a millorar l'eficiència del codi, la seva llegibilitat i fer més atractiva visualment la web.

5.4.1 Variables de sessió

Les variables de sessió en PHP són variables que no s'esborren en el pas d'una pàgina a una altra. Es tracta de variables superglobals i emmagatzemen informació durant tota l'estada d'un usuari a la nostra web.

Hem utilitzat les variables de sessió principalment per a la funcionalitat del login. Quan un usuari s'identifica correctament el sistema genera unes variables de sessió amb la informació més rellevant i utilitzada de l'usuari (figura 5.7). Amb aquest mètode aconseguim que per a cada pàgina no s'hagi de comprovar la validesa de la identificació de l'usuari.

```
if ($user->load($niuUser)) {  
    $_SESSION['niuUser']=$user->getNiuUser();  
    $_SESSION['tipusUser']=$user->getTipususer();  
    $_SESSION['nomUser']=$user->getNomUser();  
    $_SESSION['cognomUser']=$user->getCognomUser();  
    $_SESSION['caduca']=time()+($caduca*60);  
}
```

Figura 5.7 Assignació de les variables de sessió.

Per utilitzar les variables de sessió cal que la primera línia de cada script PHP inicialitzi aquesta funcionalitat mitjançant la funció `session_start()`. Un cop l'usuari fa logout o passa un temps d'inactivitat cal que el sistema destrueixi les variables de sessió amb l'ordre `session_destroy()`.

També utilitzem les variables de sessió juntament amb la funció `time()` per a controlar el temps d'inactivitat de l'usuari i restringir les funcionalitats segons el perfil d'usuari. A la figura 5.8 podem observar com es comprova el temps d'inactivitat i el tipus d'usuari per a realitzar unes accions restringides. En cas de que tot sigui correcte s'amplia la variable de sessió `caduca` per a reiniciar el comptador d'inactivitat.

```
if ($_SESSION['caduca']>time() && ($_SESSION['tipusUser']==admin || $_SESSION['tipusUser']==responsable)) {  
    $_SESSION['caduca']=time()+($caduca*60);  
}
```

Figura 5.8 Control d'inactivitat i accés a funcionalitats restrictives.

En cas de que la condició no es compleixi el sistema ens deslogueja automàticament i ens envia a la pantalla de login.

5.4.2 Autoload

Hem utilitzat el mètode màgic autoload per a la càrrega de fitxers que contenen les classes. D'aquesta manera el sistema inclou automàticament els fitxers que contenen les classes que utilitzem en un script determinat sense haver-ho de fer automàticament de la següent manera (figura 5.9).

```
function __autoload($class_name) { global $ruta; require_once $ruta.'/class/'.$class_name . '.class.php';}  
  
//include ('~/cicb/class/user.class.php'); AQUESTES LÍNIES NO CALEN AMB L'AUTOLOAD.  
//include ('~/cicb/class/espai.class.php');  
  
$user = new user();  
$espai = new espai();  
  
$user->load($_SESSION['niuUser']); //JA PODEM TREBALLAR AMB LES NOSTRES CLASSES.
```

Figura 5.9 Ús del mètode màgic autoload.

5.4.3 Autoemplenat de formularis

En alguns mòduls de l'aplicació s'ha utilitzat la autoemplenat per a formularis molt llargs i monòtons. Per exemple en el cas de la assignació de permisos per a espais hem implementat la possibilitat de seleccionar-los segons la seva categoria (despatxos, laboratoris...). Mitjançant un enllaç, s'executa una funció javascript que selecciona tots els elements pertanyents a la categoria seleccionada manipulant el DOM. Ho veiem més clarament a la figura 5.10 on podem veure la funció que realitza tot el procés.

```
function selecciona(niu,classe){  
    for (i=0;i<document["permisos"+niu].elements.length;i++){  
        if(document["permisos"+niu].elements[i].type == "checkbox"){  
            if (document["permisos"+niu].elements[i].className == classe){  
                document["permisos"+niu].elements[i].checked=1  
            }  
        }  
    }  
}
```

Figura 5.10 Funció que marca checkboxes segons la seva classe.

5.5 Configuració de l'aplicació

L'aplicació ofereix un gran potencial per a la seva personalització i adaptar-se així a les necessitats dels usuaris. Moltes de les variables es poden modificar a l'arxiu */class/globals.php* com per exemple el temps d'expiració de les sessions, el nombre de camps d'entrada del fitxer .csv o el nom i contrasenya de la base de dades, entre d'altres.

També es pot modificar l'arxiu */config/config.php* per a definir les rutes del servidor per a trobar els arxius. Amb aquest mecanisme aconseguim millorar la portabilitat de l'aplicació i facilitem la seva instal·lació en qualsevol servidor.

Finalment també adjuntem uns scripts sql que creen les taules de la base de dades i introdueixen nous administradors a l'aplicació.

5.6 Proves

A continuació detallem el procés de proves que hem realitzat a l'aplicació per a garantir el correcte funcionament de la mateixa.

5.6.1 Funcionals

S'han realitzat proves funcionals a tots els mòduls per a comprovar el seu correcte funcionament. Primerament s'han fet proves als mòduls per separat durant la seva implementació. Un cop finalitzat el desenvolupament s'han fet més proves amb tots els mòduls muntats i s'han completat satisfactòriament. Per a realitzar aquestes proves s'ha utilitzat l'aplicació Selenium [15] que automatitza tasques repetitives per a fer proves en multitud de circumstàncies

5.6.2 Compatibilitat

S'han realitzat proves de compatibilitat en les plataformes Windows 7, Windows XP i Ubuntu 10.04 satisfactòriament sense causar cap efecte no desitjat en el comportament de l'aplicació. Al ser una aplicació basada en la web és molt complicat que l'aplicació no funcioni en alguna plataforma concreta ja que totes utilitzen bàsicament els mateixos protocols d'Internet.

Pel que fa a la compatibilitat entre navegadors val a dir que l'aplicació respecta totalment els estàndards establerts per la W3C per tant no dona cap problema amb els navegadors que respecten aquestes normes. L'aplicació s'ha provat en els navegadors Opera, Chrome,



Firefox i Safari, passant satisfactòriament tots els tests. També s'ha provat l'aplicació en la família de navegadors Internet Explorer. Es pot observar algunes deficiències en la visualització de la pàgina per culpa del no acompliment dels estàndards per part d'aquesta família de navegadors. Cal remarcar que qualsevol deficiència que es pugui observar només afecta a la vista ja que la lògica de l'aplicació s'executa al servidor Apache.

Finalment també s'ha provat el software en varies resolucions de pantalla diferents. Recordem que el desenvolupament estava encarat a executar-lo en un notebook pel que la visualització de la web està optimitzada per a aquest tipus de resolucions. En resolucions majors que 800 x 600 la web es visualitza correctament però desaprofitem una part de la pantalla.

5.6.3 Accessibilitat

Tot el codi de la nostra aplicació ha passat uns controls rigorosos d'acompliment dels estàndards. No s'hi ha trobat cap error que dificultin la seva accessibilitat des de diferents dispositius o la interacció amb altres aplicacions.

5.6.4 Seguretat

Tot i que la seguretat no és un aspecte crític de l'aplicació donat que es tracta d'un software de gestió intern, s'han provat atacs principalment injecció SQL sense cap èxit. Això és degut a una sèrie de bones pràctiques durant el desenvolupament com per exemple delimitar totes les variables dintre de queries SQL amb cometes per a que es tractin com a cadenes de caràcters.



Capítol 6: Conclusions

6.1 Objectius assolits

Un cop finalitzat el desenvolupament del projecte cal analitzar de manera exhaustiva si s'han complert els objectius proposats en l'apartat 2.2 de l'estudi de viabilitat. En aquest apartat havíem prioritzat els objectius en tres categories: crítics, prioritaris i secundaris. Passem a analitzar-los a continuació.

Pel que fa els objectius crítics, és a dir: importar la informació de treball (O.1), identificar els elements mitjançant la pistola lectora de codis de barres(O.2), detectar les anomalies entre l'inventari teòric i el real (O.3) i generar un informe d'incidències un cop finalitzat l'inventari (O.4), han estat tots assolits satisfactòriament. Com hem pogut comprovar a les proves funcionals (apartat 5.6.1), aquestes funcionalitats es comporten correctament sota tota mena de circumstàncies possibles: recurrència en un mateix espai entre dos usuaris, diversos tipus d'incidències com elements perduts, canvis d'ubicació o referències noves i diferents navegadors, resolucions i hardwares.

En la segona classificació dels objectius, els prioritaris, només hi teníem un objectiu: identificació d'usuaris i la possibilitat de què els responsables i administradors assignéssim àrees d'actuació (O.5). Per a assolir aquest objectiu hem desenvolupat una interfície pels administradors i responsables per a poder assignar centres i espais a cadascun dels usuaris. També poden gestionar els usuaris i les categories dels espais. Aquesta interfície també ha estat testada a l'apartat 5.6.1. Així doncs, aquest objectiu també ha estat assolit correctament.

Finalment, teníem els dos objectius secundaris: agilitzar el procés d'actualització de la informació centralitzada (O.6) i la possibilitat d'exportar un arxiu amb les incidències d'un inventari per facilitar la integració amb futures aplicacions (O.7). La funcionalitat per exportar el contingut de l'informe d'incidències optimitzat en format XML funciona correctament. Pel que fa agilitzar el procés d'actualització de la informació creiem que amb la nostra aplicació aportem les eines pertinents per a assolir aquest objectiu. Tot i això no es pot garantir que s'acompleixi l'objectiu O.6, ja que resta encara la implantació de l'aplicació i el desenvolupament d'alguna funció que interactuï amb el nostre fitxer XML.

6.2 Desviacions de la planificació

S'han produït una sèrie de desviacions pel que fa a la planificació del projecte. Inicialment s'havia previst tancar el projecte el 25 de maig de 2010 però aquesta data no s'ha pogut assolir. Principalment atribuïm aquest retard a una planificació del projecte massa optimista ja que, per exemple, només havíem assignat 128 hores a l'etapa de codificació i finalment n'hem

necessitat moltes més. També ha influït considerablement el fet de desenvolupar el projecte durant l'any acadèmic; en ocasions no s'han pogut dedicar totes les hores planificades al projecte. Per a solucionar aquests entrebancs es va acordar una nova planificació del projecte durant l'estiu que, finalment, s'ha assolit satisfactòriament.

Detallem a continuació la desviació desglossada per les tasques (figura 6.1).

Número	Tasca	Durada teòrica	Durada real
1	Inici del projecte: assignació i matriculació del projecte	2 hores	2 hores
	Planificació	10,33 dies	10,33 dies
2	• Estudi de viabilitat	30 hores	30 hores
3	• Aprovació EV (Checkpoint)	1 hora	1 hora
	Anàlisi de l'aplicació	8 dies	11,3 dies
4	• Anàlisi requisits (casos d'ús)	10 hores	12 hores
5	• Anàlisi de dades (base de dades)	5 hores	6 hores
6	• Anàlisi seguretat i legalitat	5 hores	5 hores
7	• Documentació de l'anàlisi	3 hores	10 hores
8	• Aprovació Anàlisi (Checkpoint)	1 hora	1 hores
	Disseny de l'aplicació	9 dies	13,3 dies
9	• Disseny de la base de dades	5 hores	7 hores
10	• Disseny modular de l'aplicació	3 hores	7 hores
11	• Disseny de la interfície	10 hores	10 hores
12	• Disseny de les proves	5 hores	5 hores
13	• Documentació del disseny	3 hores	10 hores
14	• Aprovació del disseny (Checkpoint)	1 hora	1 hores
	Desenvolupament	42,67 dies	63,3 dies
15	• Preparació entorn de desenvolupament	2 hores	2 hores
16	• Configuració BD	6 hores	6 hores
17	• Funcionalitats de l'aplicació	100 hores	160 hores
18	• Interfície	20 hores	22 hores
	Proves	13 dies	13 dies
19	• Proves unitàries	15 hores	15 hores
20	• Proves d'integració	15 hores	15 hores
21	• Proves d'estrès (incidències, riscos)	5 hores	5 hores
22	• Documentació de les proves	3 hores	3 hores
23	• Aprovació de les proves	1 hora	1 hora
	Implantació	4,67 dies	4,67 dies
24	• Instal·lació i proves reals	14 hores	14 hores
	Generació documents (memòria)	30 hores	60 hores
	Tancament del projecte	1 hora	3 hores
	Defensa del projecte	5 hores	8 hores
	TOTAL	302 hores	422 hores

Figura 6.1 Desviacions desglossades per tasca.

Hi ha hagut una desviació important sobretot en les tasques de documentació i generació de documents i en l'etapa de codificació. En total hem tingut una desviació de 120 hores que ha estat el principal motiu a l'hora de decidir entregar el projecte al setembre.

6.3 Línies d'ampliació

Creiem que l'aplicació es pot ampliar amb noves funcionalitats. En citem algunes a continuació.

- Desenvolupar un sistema per a millorar l'exportació de dades substituint el sistema desenvolupat amb XML. Es podria realitzar l'exportació de dades de manera automàtica amb les aplicacions de gestió d'actius del servei de Gestió Econòmica.
- Introduir el tema de la traçabilitat dels elements per a poder veure un historial de cadascun dels actius de l'inventari. Amb aquesta funcionalitat seria possible veure, per exemple, les localitzacions on ha estat un element de l'inventari per a poder determinar on s'ha produït un desperfecte en algun element.
- La possibilitat de realitzar més d'un inventari per centre simultàniament per a poder fer inventaris per separat de les diferents categories d'espais (despatxos, laboratoris, etc.) i així tenir un informe d'incidències per a cada categoria. Actualment, aquest procés s'ha de realitzar seqüencialment per a cada categoria d'espais.

6.4 Valoració personal

El desenvolupament del projecte m'ha semblat una experiència enriquidora en molts sentits. He pogut utilitzar coneixements que he après durant el transcurs dels estudis universitaris i aplicar-los al desenvolupament d'un projecte real construït des de zero. A més, gràcies a alguns problemes que han sorgit durant el desenvolupament m'ha calgut consultar fonts d'informació externes per aprofundir més en algunes àrees de les que tenia coneixements molt superficials.

Crec que gràcies a aquest projecte he millorat considerablement els meus coneixements en tecnologies que ja coneixia: com el PHP, el CSS i l'XHTML i també en tecnologies que no havia utilitzat massa, com l'AJAX. M'ha permès utilitzar noves metodologies de treball com el patró de desenvolupament MVC, utilitzar un *framework* javascript com el jQuery o aprofundir en un IDE tan important i polifacètic com Eclipse.



He après a gestionar i documentar correctament un projecte gran, cosa que crec que em serà molt útil en la meva carrera professional. També he après a treballar en equip juntament amb el meu director de projecte, la gent del Serveis d'Informàtica Distribuïda (SID) i els responsables del servei de Gestió Econòmica.

Resumint, faig una valoració personal molt positiva del projecte i espero que em sigui molt útil.

Bibliografia

A continuació enumerem les referències bibliogràfiques que s'han fet a la memòria:

- [1] Escola d'Enginyeria, Campus de Sabadell. *Normativa de projectes*. [en línia]. Disponible a: <http://www.uab.cat/Document/639/153/normativaProjectesEEsabadell.pdf> [Última consulta: 2 de setembre de 2010].
- [2] Openbravo. *Web oficial del projecte Openbravo*. [en línia]. Disponible a: <http://www.openbravo.com> [Última consulta: 2 de setembre de 2010].
- [3] OpenERP. *Web oficial del projecte OpenERP*. [en línia]. Disponible a: <http://openerp.com/> [Última consulta: 2 de setembre de 2010].
- [4] Free Software Foundation. *General Public License*. [en línia]. Disponible a: <http://www.gnu.org/copyleft/gpl.html> [Última consulta: 2 de setembre de 2010].
- [5] GDS Sistemas. *Web oficial software Bicicletaria*. [en línia]. Disponible a: <http://www.gdssistemas.com.ar/> [Última consulta: 2 de setembre de 2010].
- [6] World Wide Web Consortium. *Web oficial* [en línia]. Disponible a: <http://www.w3.org> [Última consulta: 2 de setembre de 2010].
- [7] World Wide Web Consortium. *Especificació del llenguatge XHTML*. [en línia]. Disponible a: <http://www.w3.org/TR/2002/REC-xhtml1-20020801/> [Última consulta: 2 de setembre de 2010].
- [8] World Wide Web Consortium. *Especificació del llenguatge CSS*. [en línia]. Disponible a: <http://www.w3.org/TR/2008/REC-CSS2-20080411/> [Última consulta: 2 de setembre de 2010].
- [9] ASUSTeK. *Web oficial Asus Eee PC*. [en línia]. Disponible a: http://www.asus.com/product.aspx?P_ID=rvrDgP8iPxQFzF9i [Última consulta: 2 de setembre de 2010].
- [10] Wasp Barcode Technologies. *Web oficial de la pistola lectora de codis de barres Wasp*. [en línia]. Disponible a: http://www.waspbarcode.es/scanners/cordless_barcode_scanner.asp [Última consulta: 2 de setembre de 2010].
- [11] Zend. *Web oficial del projecte Zend Studio*. [en línia]. Disponible a: <http://www.zend.com/en/products/studio/> [Última consulta: 2 de setembre de 2010].
- [12] MP Software. *Web oficial del projecte PHP Designer* [en línia]. Disponible a: <http://www.mpsoftware.dk/phpdesigner.php> [Última consulta: 2 de setembre de 2010].



[13] Eclipse. *Web oficial del projecte Eclipse*. [en línia]. Disponible a: <http://www.eclipse.org/> [Última consulta: 2 de setembre de 2010].

[14] Wikipedia. *Navegadors que passen el test Acid*. [en línia]. Disponible a: http://es.wikipedia.org/wiki/Acid3#Navegadores_que_superan_la_prueba [Última consulta: 2 de setembre de 2010].

[15] SeleniumHQ. *Web oficial del projecte Selenium*. [en línia]. Disponible a: <http://seleniumhq.org/> [Última consulta: 2 de setembre de 2010].

A més s'han consultat les següents fonts d'informació.

Javier Eguíluz Pérez (2008) *Introducción al XHTML*. [en línia]. Disponible a: http://www.librosweb.es/xhtml/pdf/introduccion_xhtml.pdf

Javier Eguíluz Pérez (2008) *Introducción al CSS*. [en línia]. Disponible a: http://www.librosweb.es/css/pdf/introduccion_css.pdf

Javier Eguíluz Pérez (2009) *CSS Avanzado*. [en línia]. Disponible a: http://www.librosweb.es/css_avanzado/pdf/css_avanzado.pdf

Javier Eguíluz Pérez (2008) *Referencia de CSS 2.1*. [en línia]. Disponible a: <http://www.librosweb.es/referencia/css/index.html>

Jason E. Sweat. *PHP|Architect's Guide to PHP Design Patterns* (2005). Colecció PHP architect nanobooks.



Agraïments

- A la meva família, en especial a la meva mare Ester i a la meva germana Laura per tot el suport i efecte mostrat durant tots aquets anys, especialment aquets últims mesos de projecte.
- A la Sílvia, per la correcció de la memòria i sobretot per aguantar-me tots aquets mesos d'histèria informàtica.
- A la gent dels Serveis Informàtics Distribuïts de la UAB i en especial al Sr. Jordi Román per contribuir de manera determinant a la meva formació.
- Al Sr. Jordi Pons per guiar-me i dirigir-me durant aquets mesos de projecte i sobretot per la seva paciència.
- I finalment al meu pare Miquel Àngel per ser com és i perquè totes les coses bones que tinc son en gran part gràcies a ell. Per transmetre'm el seu esperit de superació i perquè a vegades no calen paraules per demostrar els sentiments.





Sergi González Pérez

a 14 de setembre de 2010.