

MEMÒRIA DEL PROJECTE FINAL DE CARRERA

Prestíssimo! 2.0

Plataforma oberta multiservei
per a la gestió de Societats
Musicals



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ENGINYERIA INFORMÀTICA

Àngel Bernat Ferre - 48294163M

Setembre 2011

Director: Juan Vicente Capella Hernández



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA





Índex

Capítol 1. Introducció.

- 1.1 - Justificació del Projecte.
- 1.2 - Objectius.
- 1.3 - Estructura de la Memòria.

Capítol 2. Tecnologies emprades.

- 2.1 - El llenguatge PHP.
- 2.2 - Javascript: El codi que s'executa al Navegador.
- 2.3 - La capa de persistència: el llenguatge SQL.
- 2.4 - La capa de persistència: El SGBD MySQL.
- 2.5 - Apache, el Servidor Web.
- 2.6 - El sistema base: Joomla!
- 2.7 - La llibreria de creació de PDF: TCPDF.
- 2.8 - L'aplicació de lectura de codis de barres: ZBarCam.

Capítol 3. Anàlisi i Disseny de l'aplicació.

- 3.1 - Especificació de Requisits Software.
- 3.2 - Diagrama Entitat - Relació.

Capítol 4. Implementació.

- 4.1 - L'API de *Joomla!*
- 4.2 - Estructura dels Components *Joomla!*
- 4.3 - Multillenguatge i *Joomla!*
- 4.4 - Capa de Presentació. Plantilles en *Joomla!*
- 4.5 - Implementació de la Base de Dades.

Capítol 5. Conclusions i futur de l'aplicació.

- 5.1 - Resum del treball realitzat.
- 5.2 - Futur de l'aplicació i futures ampliacions.
- 5.3 - Conclusions.

Annexos

- A.1 - Bibliografia.
- A.2 - Manual d'usuari.
- A.3 - Manual d'instal·lació



Capítol 1. Introducció.

1.1 Justificació del Projecte

Prestíssimo! És un aplicació web dissenyada especialment per a la gestió de Societats Musicals o Bandes de Música, que són un tipus de formació o agrupació musical molt pròpia de la cultura valenciana. Es tracta d'un Projecte Final de Carrera de Tipus B proposat per Àngel Bernat Ferre, alumne de l'*Escola Tècnica Superior d'Informàtica* i estudiant de la titulació *Enginyeria Informàtica*.

La motivació d'aquesta proposta de PFC de tipus B ve explicada en gran part pel fet que s'ha pogut observar que la majoria de les tasques administratives i de gestió que es duen a terme en una societat musical estan completament desinformatitzades, i les poques que sí ho estan, es troben molt descentralitzades. Podriem afirmar que avui en dia la majoria de les Societats Musicals s'han resistit a l'evolució dels Sistemes d'Informació i a la informatització de les múltiples dades que gestionen, de forma que segueixen emprant metodologies tradicionals (parlant, és clar, de la tupla bolígraf i paper). Amb la intenció de proporcionar una major comoditat en les tasques de gestió, agilitzar certes transaccions, millorar la seguretat i integritat de les dades que es manipulen i centralitzar totes aquestes en una única aplicació naix *Prestíssimo!*, l'aplicació que permetrà gestionar de forma centralitzada una societat musical.

Aquest projecte ja va estar iniciat com a PFC de Joan Albert Silvestre i Àngel Bernat per a la finalització dels estudis d'Enginyeria Tècnica en Informàtica de Gestió, i va donar com a resultat una aplicació de gestió, encara que de funcionalitats reduïdes. És tanta la motivació que provoca en l'alumne, com la necessitat que tenen les societats musicals d'informatitzar els seus processos de gestió, que l'aplicació ha estat objecte de millora i ha donat peu a la realització d'aquest PFC, que té com a objectiu l'ampliació profunda de l'aplicació en termes de funcionalitat i seguretat.



1.2 Objectius

Prestíssimo és una aplicació web per a la gestió de dades d'una societat musical, amb la qual cosa el principal objectiu que persegueix aquesta aplicació és suplir la desinformatització en la que es troben grans quantitats de dades generades per les Bandes de Música. Cal evolucionar cap a les noves tecnologies dels Sistemes d'Informació, no sols per la comoditat que això suposa, sinó també per la seguretat i integritat addicional que proporcionen, així com per la centralització de les dades amb les que treballa una Societat Musical en una única aplicació. A més a més, al tractar-se d'una aplicació Web que serà accedida a través d'una intranet o xarxa local, s'incrementa la disponibilitat i l'accessibilitat de les eines de gestió que Prestíssimo! proporciona.

A banda de les necessitats d'una Societat Musical que es pretenen informatitzar, amb *Prestíssimo!* també es pretén demostrar que el software lliure i els llenguatges de programació open-source són una alternativa potent i vàlida per al desenvolupament d'aplicacions web. La tupla *PHP* (llenguatge de Programació) - *MySQL* (Sistema Gestor de Bases de Dades) - *Apache* (Servidor Web) conforma el pilar sobre el que es sosté aquesta aplicació, representant les principals tecnologies lliures emprades per al desenvolupament d'aquesta, que posteriorment seran detallades. També s'analitzarà el CMS (*Content Management System*) lliure anomenat *Joomla!*, el qual constitueix l'aplicació base sobre la qual s'han desenvolupat les eines de gestió de *Prestíssimo!*, així com les eines utilitzades per a la creació de documents PDF i la lectura i creació de codis de barres en 1 i 2 dimensions (aplicació *ZBar*).



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA





1.3 Estructura de la Memòria

En aquesta memòria tractarem d'explicar al detall com s'han emprat les diferents tecnologies que han fet possible el desenvolupament de *Prestíssimo!* (Capítol 2), així com les decisions preses en les etapes d'anàlisi, disseny (Cap. 3) i implementació (Cap. 4). També explicarem com s'ha treballat amb el sistema base, el CMS Joomla! juntament amb les eines utilitzades per a la creació de PDF i la lectura i creació de codis de barres (Cap. 4), i per últim treurem les conclusions d'aquest Projecte Final de Carrera (Cap.5).



Capítol 2. Tecnologies emprades.

2.1 El llenguatge PHP

PHP (*PHP Hypertext Pre-processor*) és el llenguatge de programació que implementa la lògica de l'aplicació, estableix una API o framework per a treballar amb la capa de persistència, i genera el codi HTML que representa la capa de presentació. Podem dir que PHP és la tecnologia més fonamental sobre la qual s'asseu aquesta aplicació web, ja que en certa manera integra la major part de les tecnologies emprades. Tant el sistema base (*Joomla!*) com els components desenvolupats que donen la funcionalitat requerida a l'aplicació es troben desenvolupats en aquest llenguatge.

En concret, PHP és un llenguatge de programació interpretat de propòsit general, molt popular, i que està dissenyat especialment per al desenvolupament web (pàgines web dinàmiques) i pot ser embegut dins de codi HTML. Generalment aquest s'executa en un servidor web, prenent el codi PHP com a entrada i generant pàgines web (codi HTML) com a eixida. Pot ser desplegat en la majoria dels servidors web i en quasi tots els sistemes operatius i plataformes sense suposar un cost elevat.

Aquest llenguatge va veure la llum a l'any 1995 de la mà del seu progenitor, Rasmus Lerdof, però la seva implementació principal corre a càrrec actualment de *The PHP Group*. Està publicat baix la llicència *PHP License*, la qual la *Free Software Foundation* considera lliure.

PHP té un gran paregut als llenguatges estructurats com *C* o *Perl*, recollint les millors característiques d'aquests llenguatges, i inclou suport a l'Orientació a Objectes. A més a més, com a llenguatge de programació dinàmic ofereix múltiples biblioteques per a la connexió amb diferents SGBD com *MySQL*, *Postgres*, *Oracle*, etc. I com a llenguatge nascut de i per internet i per a la programació web, inclou també biblioteques per a treballar amb diferents protocols com HTTP, FTP, POP3, SMTP, IMAP, XML, LDAP, etc.

PHP és sense lloc a dubte un gran llenguatge de programació, senzill, fàcil i ràpid d'aprendre, a la vegada que sòlid i potent, que constitueix una alternativa vàlida i lliure a les



tecnologies de Microsoft *ASP* i *ASP .NET*, *ColdFusion* de Adobe, *JSP/Java* de Sun Microsystems i *CGI/Perl*.

Tot seguit podem veure algun exemple d'ús d'aquest llenguatge, incloent el mític “Hola Món”.

```
<?php

echo “Hola Món”;

//comentari suma

function suma ( $a , $b )
{
    $c=$a+$b;
    return $c;
}

/*
comentari resta
*/

function resta ( $a , $b )
{
    $c=$a-$b;
return $c;
}

$a=1;
$b=-5;

suma($a,$b);
echo $c;

resta($a,$b);
echo $c;

$variable=23;
$variable=“hola”;

$vector = array();
$vector[0] = “hola”;
$vector[“fruita”] = “poma”;
$variable = $vector;
```

?>



Com es pot apreciar, les variables no tenen tipat (de forma similar a Perl), cosa que confereix una gran flexibilitat i una gran facilitat d'ús i d'aprenentatge a aquest llenguatge. Ara bé, per la característica de ser un llenguatge de programació dinàmic aquest no es pot compilar, de forma que els errors de compilació tant sols poden ser detectats en temps d'execució, fet que dificulta en certa manera la depuració i optimització del codi.

La instrucció *echo* s'encarrega de treure per l'eixida estàndard una cadena de caràcters (o *string*) que s'inclou en el contingut d'una pàgina HTML que es genera automàticament i que s'envia al procés client per a ser visualitzada en un navegador web a l'acabar de processar tot el codi PHP. D'aquesta forma podem generar codi HTML dinàmicament.

També cal destacar que per les característiques que ofereix aquest llenguatge és molt fàcil generar codi desordenat i molt complex de mantindre, sobretot perquè és molt fàcil cometre l'error de mesclar la lògica de l'aplicació amb la presentació o codi HTML que hem de generar en un mateix fitxer, fet que pot arribar a ser caòtic. Afortunadament, *Prestíssimo!* separa la capa de presentació de la capa de lògica, però això s'explicarà detalladament al capítol 4.



2.2 Javascript: El codi que s'executa al Navegador.

Javascript és un llenguatge de programació creat a l'any 1995 per Brendan Eich (membre de Netscape Communications), i està pensat, a l'igual que el llenguatge PHP, únicament per a internet. Aquest llenguatge ens permet principalment definir la interactivitat de l'aplicació web amb l'usuari (generar respostes a determinats events), crear efectes visuals especials, etc. En concret, per a *Prestíssimo!* Javascript ocupa un lloc preferent en quant a la validació dels camps dels diferents formularis existents a l'aplicació, i en quant a la interactivitat de la pàgina web amb l'usuari.

Aprofundint més amb les característiques d'aquest llenguatge, Javascript és un llenguatge interpretat (és a dir, no requereix compilació per a ser executat), que és executat en el procés client (més concretament en el navegador web del client). Això suposa un gran avantatge en quant a l'escalabilitat de l'aplicació web, ja que la validació de les dades que s'han d'enviar al procés servidor es realitza sempre al procés client, amb la qual cosa es redueix substancialment el tràfic de dades entre el dos processos. Amb una sintaxis molt similar als llenguatges Java i C, Javascript és un llenguatge que suporta també el paradigma de l'orientació a objectes.

Un dels problemes que sorgeixen al treballar amb Javascript és que al ser un codi executat en el navegador dels processos clients, i donat que actualment hi ha una gran diversificació en el mercat dels navegadors web (Mozilla Firefox, Safari, Opera, Konqueror, Internet Explorer 6, Internet Explorer 7, etc.), moltes vegades sorgeixen incompatibilitats, en el sentit que hi ha scripts que en alguns navegadors funcionen, però en altres no. Ja en el seu dia, el *World Wide Web Consortium* (WWWC) va dissenyar l'estàndar *Document Object Model* (DOM), que anava encaminat a evitar gran part d'aquestes incompatibilitats, originades per la diferencia d'implementació del model d'objectes en cada navegador.

Tot seguit anem a veure un exemple sobre com s'ha implementat la validació de camps a *Prestíssimo!*. En concret, aquesta validació de camps forma part del component de Gestió d'Instruments.



("components/com_instruments/instruments.html.php", línies 80 – 114)

```
function validarFormulari()
{
    var Formulari = document.form;
    if(comprovarEspecialitat() && comprovarCampsText() &&
comprovarData() && comprovarPropietari()) Formulari.submit();
}

function comprovarEspecialitat()
{
    var selectEspecialitat = document.getElementById("select_especialitat");
    var indexSelecc =
selectEspecialitat.options[selectEspecialitat.selectedIndex].value;

    if (indexSelecc == 0)
    {
        alert("<?php echo JText::_("Per favor, selecciona una especialitat
musical."); ?>");
        return false;
    }

    return true;
}

function comprovarCampsText()
{
    var marca = document.form.marca.value;
    var model = document.form.model.value;

    if(marca.length == 0)
    {
        alert("<?php echo JText::_("Per favor, introdueix la marca de
l'instrument."); ?>");
        return false;
    }
    if(model.length == 0)
    {
        alert("<?php echo JText::_("Per favor, introdueix el model de
l'instrument."); ?>");
        return false;
    }

    return true;
}
```



Com es pot comprovar, per tal de poder enviar el formulari de dades (instrucció `Formulari.submit()`) abans hem de comprovar tots els camps necessaris. En el cas de que totes les validacions siguin satisfactòries, el formulari s'enviarà per tal de ser processat. En cas contrari, apareixerà una finestra informant de l'error (instrucció `alert()`). Cal dir que la instrucció `alert()` inclou embeguda una instrucció php, la funció de la qual és habilitar el suport multilinguatge. Això s'explicarà en detall al capítol 4, apartat 4.3.

Ara es mostra un altre exemple, en aquest cas, de control de la interacció de l'usuari amb l'aplicació web. És un fragment de codi extret de la gestió / enviament d'avisos, en concret de l'enviament d'avisos manuals, en el qual, si en "Tipus enviament" es marca el checkbox "E-Mail", automàticament s'habilita el camp d'introducció de l'adreça e-mail a la qual es desitja enviar el correu electrònic. D'altra banda, si es desmarca el checkbox "E-mail", dit camp es deshabilita.

("components/com_avisos/avisos.html.php", línies 27 – 35)

```
function activarTextBoxEmail()
{
    var email= document.getElementById("emailManual"); (1)

    if(email.disabled) email.value=""; (2)
        else email.value="<?php echo JText::_("Adresa e-mail"); ?>"; (3)

    email.disabled = !email.disabled; (4)
}
```

El mètode `activarTextBoxEmail()` s'executa en resposta a l'event *onchange* del checkbox "E-Mail". Sempre es canvia l'estat del camp (textbox) email (4), modificant la seva propietat `disabled`.

Com hem pogut comprovar en aquests xicotets exemples, Javascript és un llenguatge de programació quasi fonamental per al desenvolupament d'aplicacions web, doncs tota la interactivitat de l'usuari i la validació de les dades introduïdes per aquest és controlat pel codi Javascript.



2.3 La capa de persistència: El llenguatge SQL

El llenguatge SQL (Structured Query Language o Llenguatge d'interrogació estructurat) és un llenguatge estàndar (des de l'any 1987) que permet treballar amb la majoria de bases de dades relacionals, i que generalment, gràcies a llibreries que hi donen suport, es pot incloure dins d'altres llenguatges de programació (com en el cas de *Prestíssimo!*, on conviuen PHP i SQL). És un llenguatge simple i intuïtiu, però que no és complet ni relacionalment ni computacionalment.

Aquest llenguatge es pot dividir en tres conjunts d'instruccions atenent a la seva funció:

- Sentències de definició de dades: permeten crear, modificar i esborrar estructures de dades.
- Sentències de manipulació de dades: permeten llegir, inserir, modificar i esborrar registres de la bd.
- Sentències de control: permeten finalitzar o rebutjar transaccions i definir permisos a usuaris.

Tot seguit anem a veure alguns exemples d'ús d'aquest llenguatge juntament amb codi PHP a l'aplicació *Prestíssimo!*

("components/com_instruments/instruments.php", línia 132)

```
$query="SELECT DNI, NOM, COGNOMS FROM educand ORDER BY COGNOMS  
ASC";
```

En aquest exemple, que pertany al component Gestió d'Instruments, la sentència SQL selecciona els atributs dni, nom i cognoms de tots els registres existents a la taula educand, i en acabant els ordena ascendentment pels cognoms.

("components/com_instruments/instruments.php", línia 124)

```
$query="SELECT * FROM instrument WHERE ID_INSTRUMENT='$id_instrument';
```



En aquest cas, la instrucció SQL selecciona totes aquelles tuples de la taula instrument que complisquen la condició de que el valor de l'atribut id_instrument siga igual a un valor emmagatzemat per una variable php, més concretament \$id_instrument.

("components/com_instruments/instruments.php", línia 159)

```
$query="UPDATE instrument SET MARCA='$marca', MODEL='$model',  
SERIE='$serie', DNI_MUSIC='$dni_music', DNI_EDUCAND='$dni_educand',  
ID_ESPECIALITAT='$id_especialitat', DATA_COMPRA='$data_compra'  
WHERE ID_INSTRUMENT='$id_instrument'";
```

Podem veure com aquesta sentència SQL realitza una modificació d'una tupla existent, en concret aquella l'atribut id_instrument de la qual siga igual al valor que conté la variable php \$id_instrument.

("components/com_instruments/instruments.php", línia 103)

```
$query="DELETE FROM instrument WHERE ID_INSTRUMENT='$id_instrument'";
```

Per acabar, es mostra una sentència d'eliminació de dades, on s'esborra aquella tupla de la taula instrument on l'atribut id_instrument siga igual al valor que conté la variable php \$id_instrument.

Com s'ha pogut comprovar, la comunió entre els llenguatges PHP i SQL és total. De fet, aquesta parella de tecnologies és una de les més comunment usades en el desenvolupament d'aplicacions web, juntament amb el SGBD MySQL i el servidor Apache, dels quals es parlarà tot seguit.



2.4 La capa de persistència: El SGBD MySQL.

El llenguatge SQL per sí sol no pot treballar amb cap base de dades. Per poder fer-ho, necessita comunicar-se amb un SGBD (Sistema Gestor de Base de Dades), que serveix d'interfície entre l'aplicació web (lògica de l'aplicació) i la capa de persistència. No s'aprofundirà més en la definició d'un SGBD, ja que no és la finalitat d'aquesta memòria.

MySQL és, concretament, un SGBD relacional, multifil i multiusuari, i que treballa com és obvi amb el llenguatge SQL. La principal característica que ha fet assolir l'èxit a aquest SGBD és la ràpida velocitat en executar consultes, cosa que beneficia enormement les aplicacions web, augmentant la seua escalabilitat i rapidesa; així com el suport de forma nativa del llenguatge PHP a aquest SGBD. De fet, MySQL és un dels components del que s'anomena l'arquitectura LAMP o WAMP, on A, M i P fan referència a Apache, MySQL i PHP respectivament, i la L o la W es refereixen als sistemes operatius Linux o Windows.



2.5 Apache, el servidor Web.

L'últim component de l'arquitectura LAMP o WAMP que falta per descriure és l'Apache, el servidor web o servidor HTTP, que s'encarrega de gestionar les peticions HTTP dels processos clients i servir les pàgines web, la majoria d'elles prèvia execució de codi PHP, com és el cas de *Prestíssimo!*. Apache és un servidor web multiplataforma desenvolupat desde l'any 1995, de codi lliure i obert, amb suport per a la gran majoria de llenguatges de programació de servidor i esquemes d'autenticació, SSL (Secure Sockets Layer) i TLS, entre d'altres.



2.6 El sistema base: Joomla!

Partint de la premisa de reutilització de codi (o de “no reinventar, o en aquest cas no reimplementar la roda”), *Prestíssimo!* és una aplicació web desenvolupada sobre un sistema base, el popular CMS (*Content Management System*) de codi lliure i anomenat *Joomla!*. Desenvolupat en PHP, i pensat per a una arquitectura LAMP o WAMP, *Joomla!* permet administrar continguts web dinàmics de forma senzilla i intuïtiva, però a més a més *Joomla!* és capaç de crear memòria caché de pàgines per a millorar el rendiment, indexament web, habilitar feed RSS, obtenir versions imprimibles de pàgines, exportar documents i notícies a PDF, crear blogs, fòrums, comunitats d'usuaris, enquestes, calendaris, búsquedes al lloc web, i té suport multilinguatge.

No són precisament aquestes característiques les que han fet que *Joomla!* fos elegit com el sistema base de *Prestíssimo!*, si més bé cap d'aquestes funcionalitats no apareixen a l'aplicació, per un motiu evident: *Prestíssimo!* no és una pàgina web de continguts, sino una aplicació web de gestió. Aleshores, per què *Joomla!*? Doncs perquè ofereix una gran flexibilitat.

Des de l'actualització de *Joomla* a la versió 1.5 (que és la que empra *Prestíssimo!*), tota funcionalitat oferida és realment una aplicació que s'executa dins de l'estructura del mateix *Joomla*, fet que implica que és factible implementar qualsevol tipus de projecte sobre aquest CMS. *Joomla* permet ampliar la seva funcionalitat gràcies al sistema d'extensions, de forma que és molt senzill afegir al sistema base mòduls, plug-ins, plantilles de presentació i/o components, tots ells totalment independents de l'aplicació *Joomla*. Són aquests últims els que han definit la funcionalitat requerida per l'aplicació *Prestíssimo!*, de forma que cada funcionalitat o eina de gestió ha estat implementada com a un component totalment independent de la resta de components i del sistema. D'això se'n parlarà al capítol 4, apartat 4.2.

L'elecció de Joomla com a sistema base ens ha estalviat moltes hores innecessàries de treball, sobretot en quant al sistema d'autenticació d'usuaris, i ha conferit, sense dedicar esforç algun, una major seguretat a l'aplicació. A més a més, Joomla! ofereix una àmplia i



potent API (Application Program Interface) per poder accedir a implementacions que el sistema base proporciona, distribuïdes a través de diverses classes agrupades en paquets. Podem destacar, entre d'altres, el paquet HTML, que inclou nombroses classes estàtiques com JHTML, que permeten generar dinàmicament codi HTML; o el paquet Mail, amb la classe JMail, que permet crear i enviar correus electrònics de forma senzilla. D'aquesta API es parlarà més detalladament al capítol 4, apartat 4.1.

2.7 La llibreria de creació de PDF: TCPDF.

TCPDF és una llibreria/Classe Open Source creada per al popular llenguatge de programació web PHP en les seues versions 4 i 5, que permet la creació de documents PDF dinàmicament.

Dues de les qualitats més apreciades d'aquesta classe són la simplicitat a l'hora de crear fitxers PDF i la capacitat d'interpretar codi XHTML. Actualment és troba en desenvolupament la seua versió 5, que va ser llançada en primícia el maig de 2010. A la web oficial es poden trobar infinitat d'exemples d'utilització de la llibreria, cosa que permet aprendre fàcilment la utilització d'aquesta funcionalitat.

Aquesta llibreria ha estat utilitzada en l'aplicació *prestissimo* per a la creació de documents PDF. La primera fita que s'ha aconseguit mitjançant aquesta llibreria és la creació d'un document on apareix un llistat de tots els músics convocats a un acte, fet que permetrà als administratius de la societat musical colgar al tauló d'anuncis del local social un llistat per a que siga visible a tots els membres de dita societat musical.

Tot seguit anema veure un exemple d'ús d'aquesta llibreria des del llenguatge PHP a l'aplicació *Prestíssimo!*

("components/com_actescarrer/actescarrer.html.php", línia 273)

```
// create new PDF document
$pdf = new TCPDF(PDF_PAGE_ORIENTATION, PDF_UNIT, PDF_PAGE_FORMAT,
true, 'UTF-8', false);

// set document information
$pdf->SetCreator(PDF_CREATOR);
$pdf->SetAuthor('Prestíssimo!');
$pdf->SetTitle('Acte al carrer 001');
$pdf->SetSubject('Prestissimo Acte al carrer');
... ..
$pdf->Output('citacions'.DS.$anyDeCreacioDeCitacio.DS.$acte->LLOC_ACTE.'-'. $acte-
>DATA_ACTE.'.pdf', 'F'); //guarda el pdf sense preguntar amb el nom especificat
```



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



En aquest exemple es pot veure com es crea un objecte \$pdf utilitzant les funcions de la llibreria TCPDF. Aquest objecte va essent modificat fins que es realitza un Output i es crea definitivament el document PDF.



2.8 L'aplicació de lectura de codis de barres: ZBarCam.

Zbar és una aplicació *Open Source* que permet llegir codis de barra desde diversos formats, com són els fitxers d'imatge, les imatges raw i les captures desde l'entrada de video. A més, soporta molts formats o tipus de codis de barres, incluint EAN-13/UPC-A, UPC-E, EAN-8, Code-128, Code-39 i Codis QR (2 dimensions).

Aquesta aplicació està realitzada baix diferents tecnologies. Així podem trobar l'aplicació en C, Perl, Python i està també disponible per als sistemes operatius Linux i Windows. No obstant, no s'ha pogut trobar cap aplicació en PHP, i és per això que la seua utilització en l'aplicació *Prestissimo!* s'ha realitzat mitjançant la crida per línia de comandaments a l'aplicació ZBarCam prèviament instal·lada al sistema.

La funcionalitat que ofereix aquesta aplicació s'ha utilitzat per a que *Prestissimo!* poguera identificar el pagament de les liquidacions dels músics mitjançant la lectura d'un codi de barres o QR insertat en la fulla de la liquidació corresponent. És per això que prèviament, en la creació del document de liquidació, s'ha afegit un codi QR i un codi de barres que identifica la liquidació inequívocament.



Capítol 3. Anàlisi i Disseny de l'aplicació.

3.1 Especificació de Requisits Software.

Una de les etapes més importants, si no la que més, en els cicles o processos de producció o desenvolupament de software, ja siguin lineals o iteratius, és l'etapa d'Anàlisi, i posteriorment, la de Disseny. Un error en alguna d'aquestes dues etapes penalitza greument les etapes posteriors, per no dir si aquestes dues etapes s'obvien.

Pel que fa a l'anàlisi de l'aplicació, s'han capturat i documentat tots els requisits funcionals i no funcionals a un document d'Especificació de Requisits (ERS), que compleix amb l'estàndar IEEE 830. Aquest document es mostra a continuació, incrustat a la pròpia memòria.

Especificació de Requeriments Software

Aplicació per a la Gestió de Societats Musicals

1. Introducció

1.1 Propòsit.

El propòsit és definir quins són els requeriments software que ha de reunir una aplicació que s'encarregue de gestionar gran part dels aspectes una societat musical. Aquesta aplicació, juntament amb tota la documentació associada conformen el Projecte de Fi de Carrera (PFC) que presenta l'alumne Àngel Bernat Ferre.

1.2 Àmbit.

L'aplicació està dissenyada per accedir de forma remota a través de la web, implementant la típica arquitectura Client/Servidor web, de forma que l'aplicació siga accessible des de qualsevol navegador, amb la qual cosa les diferents tasques de gestió es poden realitzar als distints ordinadors o llocs de treball de la intranet o xarxa local. D'aquesta manera, la capa de presentació i part de la lògica (Javascript) residirà al client, mentre que la major part de la lògica de l'aplicació i la persistència residiran al servidor. No es contempla l'accés a l'aplicació des d'Internet, en primer lloc per qüestions de seguretat, i en segon lloc, perquè la naturalesa d'una societat musical no ho fa necessari, ja



que generalment els locals i despatxos de les bandes de música estan localitzats en un únic punt geogràfic.

Prestíssimo!, que és el nom que rebrà l'aplicació, oferirà a l'usuari un conjunt d'eines de gestió, que permetran dur un control centralitzat de gran part de les activitats de gestió d'una banda de música, les quals seran nomenades i explicades posteriorment.

1.3 Definicions, acrònims i abreviatures.

- **Acte/eixida al carrer:** Qualsevol actuació de la banda remunerada econòmicament al carrer o en lloc tancat, que ve motivada per una prèvia contractació per part d'un client i a la qual participa un nombre concret de músics, generalment estipulat al contracte. Cada músic percep una quantitat de diners determinada per participar en un acte.
- **Llista:** Relació de músics que estan citats per participar en un acte al carrer. Aquesta relació és pot modificar en qualsevol moment, i ha de permetre anotar quins músics tenen porra, quins instruments ha de portar cada músic (especialment en el cas de la percussió), o altres incidències.
- **Porra:** Per a un acte al carrer determinat, quantitat de diners que un músic ha de abonar a la societat musical a causa d'estar citat per a dit acte i no haver assistit, o bé haver assistit amb retràs.
- **Liquidació/Paga:** Tota aquella quantitat de diners que un músic ha de rebre a partir dels diners acumulats al participar en diferents actes al carrer durant un període en concret. A aquests guanys se li han de restar les porres i les quotes de soci, si hi ha, entre d'altres deutes que puga tenir el músic amb la banda, per a determinar la remuneració econòmica que percep finalment el músic.
- **Assaig:** Esdeveniment en el qual tots els músics de la societat estan citats per assajar el repertori d'un concert. Aquesta és una activitat no remunerada.
- **Concert:** Esdeveniment en el qual la societat musical interpreta un repertori d'obres al públic. Aquesta és una activitat no remunerada.
- **Soci:** Tot membre de la societat que participa activament o passivament en la vida de l'associació, i que satisfà una quota determinada de diners per ser soci.
- **Músic:** Tot aquell soci que té en possessió almenys un instrument i que participa en actes al carrer i en els concerts de la societat musical.
- **Cessió d'instruments:** Fet pel qual un instrument és proporcionat a un músic de la banda o un educand.
- **Educand:** Persona que, no sent soci de la banda, té en possessió un instrument d'aquesta per a la seva formació i futur ingrés a la banda.
- **Especialitat musical:** Es refereix a un tipus d'instrument, p.e. trompa, trompeta, clarinet, etc.
- **Instrument musical:** Es refereix a l'instrument físic que té en possessió la societat musical, i que cedeix als seus músics i/o educands.

1.4 Referències.

- IEEE Std 830 - IEEE Guide to Software Requeriments Specifications. IEEEStandards Board.



2. Descripció General

2.1 Perspectiva del producte.

Prestissimo és una aplicació basada en el CMS (Content Management System) Joomla, que és un potent i flexible gestor de continguts web dinàmic open source, desenvolupat en PHP i que permet el desenvolupament de software basat en components. D'aquesta forma cadascun dels requeriments o funcionalitats de l'aplicació representaran un component al sistema.

Aquesta aplicació podrà ser accedida a través de qualsevol navegador web, amb la qual cosa el procés client és totalment independent de la plataforma, conferint universalitat a l'aplicació.

Amb Prestissimo el que es pretén és millorar l'eficàcia de la gestió d'una societat musical, informatitzant i centralitzant part de les gestions que gran part de les bandes encara realitzen de forma manual.

2.2 Funcions del producte.

Les funcions que haurà de realitzar el producte són:

- Gestió de socis:
 - Alta, modificació i eliminació de socis.
 - Cerca de socis.
- Gestió de músics:
 - Alta, modificació i eliminació de músics.
 - Cerca de membres.
 - Assignació d'instruments.
- Gestió d'instruments musicals:
 - Alta, modificació i eliminació d'instruments.
 - Cerca d'instruments.
- Gestió d'especialitats musicals:
 - Alta, modificació i eliminació d'especialitats musicals.
- Gestió d'assajos:
 - Creació, modificació i eliminació d'un programa d'assajos.
 - Introducció de dades d'assistència als assajos.
 - Generació d'estadístiques d'assistència.
 - Enviament automàtic d'avisos.
- Gestió d'actes al carrer:
 - Creació, modificació i esborrament d'actes al carrer.
 - Generació de llistes en document PDF.
 - Introducció de dades d'assistència, porres i incidències.
 - Enviament automàtic d'avisos.
 - Generació automàtica de la relació de músics assignats a l'acte.
 - Cerca d'actes al carrer.



- Gestió de liquidacions/pagaments:
 - Liquidacions de diversos actes, per a tots els músics o individualment.
 - Generació de liquidacions en document PDF, identificades amb un codi de barres i/o codi QR.
- Gestió de l'arxiu musical:
 - Creació, modificació i eliminació d'obres musicals.
 - Emmagatzemament de partitures en pdf.
 - Cerca de partitures mitjançant lectura de codis de barres pels seus atributs.
- Enviament d'avisos:
 - Enviament d'avisos manuals.
- Gestió d'educands:
 - Alta, modificació i eliminació d'educands.
 - Assignació d'instruments musicals.
 - Cerca d'educands.
- Gestió de músics de la banda juvenil:
 - Alta, modificació i eliminació de músics.

2.3 Característiques de l'usuari.

Aquesta aplicació va dirigida a la junta directiva d'una societat musical, de forma que a aquests se'ls sobreentén un coneixement més que suficient de la gestió i funcionament d'una banda musical per poder emprar l'aplicació. A banda d'això, se'ls requereixen uns coneixements mínims de navegació web, ja que l'usuari final haurà de treballar amb un navegador web.

2.4 Restriccions generals.

Aquesta aplicació no té cap restricció hardware destacable ni al procés servidor ni al procés client, ja que, per al primer, l'aplicació no efectua ninguna simulació a temps real ni realitza càlculs costosos, així com no es contempla un gran volum d'usuaris concurrents (l'escalabilitat no suposa cap problema per a Prestissimo), mentre que per al segon, únicament amb l'execució d'un navegador web és suficient.

Quant a la utilització de la webcam per a llegir codis de barres, es necessitarà aquest maquinari correctament instal·lat al sistema, juntament amb l'aplicació ZBar per a poder gaudir d'aquesta funcionalitat de l'aplicació. No obstant, si no es vol utilitzar la webcam, l'aplicació també aporta l'opció d'ingressar el codi de forma manual o amb lector de codis de barres, pel que tota societat musical podrà utilitzar aquest component siuen quins siuen els accessoris connectats a l'ordinador.

Pel que fa a la plataforma, el procés client és totalment independent a aquesta, mentre que el procés servidor necessita un servidor web que suporti el llenguatge de programació PHP i incloga el SGBD MySQL. Per qüestions de seguretat és recomanable l'ús del sistema operatiu Linux per al procés servidor, així com el servidor web Apache.

En termes de seguretat, l'aplicació ha de vetllar per la integritat de les dades



manipulades i realitzar lectures i escriptures segures a la base de dades. A més a més, s'ha de garantir una identificació fiable per part de l'usuari a l'aplicació.

2.5 Dependències.

No hi ha cap dependència destacable, tan sols les dependències per al procés servidor imposades per la plataforma en que està desenvolupat Joomla, que és el llenguatge PHP.

3. Requeriments específics

3.1 Requeriments funcionals.

3.1.1. Alta, modificació i eliminació de socis

Introducció

Es podran donar d'alta, modificar i eliminar socis a la base de dades.

Entrades

L'usuari introduirà les dades necessàries per a crear/modificar un soci, o bé seleccionarà un soci per eliminar-lo.

Procés

L'aplicació crea/modifica un registre de soci a la base de dades, o bé el borra.

Eixida

L'usuari rebrà un missatge de confirmació i tot seguit veurà la llista de socis actualitzada.

3.1.2. Cerca de socis

Introducció

Buscar un o més socis a partir d'un criteri establert: per nom i/o per antiguitat.

Entrada

L'usuari indicarà quin criteri vol emprar per a la búsqueda i quin text o valor a buscar segons el criteri seleccionat.

Procés

L'aplicació realitza una búsqueda a la base de dades i mostra un llistat de socis que satisfan els criteris de búsqueda seleccionats.

Eixida

Apareix un llistat de socis si hi ha resultats coincidents, o un missatge informant que no hi ha cap coincidència.



3.1.3. Alta, modificació i eliminació de músics

Introducció

Es podran donar d'alta, modificar i eliminar músics a la base de dades.

Entrades

L'usuari introduirà les dades necessàries per a crear/modificar un músic, o bé seleccionarà un músic per eliminar-lo.

Procés

L'aplicació crea/modifica un registre de músic a la base de dades, o bé el borra.

Eixida

L'usuari rebrà un missatge de confirmació i tot seguit veurà la llista de músics actualitzada.

3.1.4. Cerca de músics

Introducció

Buscar un o més músics a partir d'un criteri establert: per nom, per antiguitat o per instrument.

Entrada

L'usuari indicarà quin criteri vol emprar per a la búsqueda i quin text o valor a buscar segons el criteri seleccionat.

Procés

L'aplicació realitza una búsqueda a la base de dades i mostra un llistat de músics que satisfan els criteris de búsqueda seleccionats.

Eixida

Apareix un llistat de músics si hi ha resultats coincidents, o un missatge informant que no hi ha cap coincidència.

3.1.5. Assignació d'instruments a músics

Introducció

En la creació/modificació d'un músic s'especificarà quin(s) instrument(s) en possessió de la societat musical tindrà cedit(s) un músic, si és que en té algun.

Entrada

L'usuari seleccionarà un instrument d'una llista. Òbviament l'instrument ha d'existir prèviament a la base de dades.

Procés

L'aplicació realitza una inserció/modificació a la relació music-instrument de la base de dades.



Eixida

L'usuari rebrà un missatge de confirmació i tot seguit veurà la llista de músics actualitzada.

3.1.6. Alta, modificació i eliminació d'instruments musicals

Introducció

Es podran donar d'alta, modificar i eliminar instruments musicals a la base de dades.

Entrades

L'usuari introduirà les dades necessàries per a crear/modificar un instrument musical, o bé seleccionarà un instrument per eliminar-lo.

Procés

L'aplicació crea/modifica un registre d'instrument musical a la base de dades, o bé el borra.

Eixida

L'usuari rebrà un missatge de confirmació i tot seguit veurà la llista d'instruments actualitzada.

3.1.7. Cerca d'instruments

Introducció

Buscar un o més instruments a partir d'un criteri establert: per especialitat, per antiguitat i/o per marca/model.

Entrada

L'usuari indicarà quin criteri vol emprar per a la búsqueda i quin text o valor a buscar segons el criteri seleccionat.

Procés

L'aplicació realitza una búsqueda a la base de dades i mostra un llistat d'instruments que satisfan els criteris de búsqueda seleccionats.

Eixida

Apareix un llistat d'instruments si hi ha resultats coincidents, o un missatge informant que no hi ha cap coincidència.

3.1.8. Alta, modificació i eliminació d'especialitats musicals

Introducció

Es podran donar d'alta, modificar i eliminar especialitats musicals a la base de dades (p.e. Trompa, Clarinet, etc.).

Entrades

L'usuari introduirà les dades necessàries per a crear/modificar una especialitat



musical, o bé seleccionarà una especialitat per eliminar-la.

Procés

L'aplicació crea/modifica un registre d'especialitat musical a la base de dades, o bé la borra.

Eixida

L'usuari rebrà un missatge de confirmació i tot seguit veurà la llista d'especialitats actualitzada.

3.1.9. Creació, modificació i eliminació d'un programa d'assajos.

Introducció

Es podrà crear, modificar i eliminar un programa d'assajos per a un concert en concret.

Entrades

L'usuari introduirà en quines dates i llocs es realitzaran els assajos i la data del concert (si hi ha) per a la creació/modificació, o bé seleccionarà l'assaig o programa d'assajos a eliminar.

Procés

L'aplicació crea/modifica un programa d'assajos a la base de dades, o bé el borra.

Eixida

L'usuari rebrà un missatge de confirmació i tot seguit veurà la llista de programes d'assajos actualitzada.

3.1.10. Introducció de dades d'assistència als assajos

Introducció

Donat un assaig d'un programa d'assajos, l'usuari podrà introduir l'assistència dels músics a dit assaig.

Entrades

L'usuari veurà un llistat amb tots els músics (membres) de la banda i marcarà quins han assistit a l'assaig.

Procés

L'aplicació insereix a la base de dades les dades d'assistència a dit assaig.

Eixida

L'usuari rebrà un missatge de confirmació.

3.1.11. Generació d'estadístiques d'assistència

Introducció

Es podran consultar les estadístiques d'assistència a assajos a partir de diferents criteris: d'un assaig en concret, d'un programa d'assajos sencer, etc.



Entrades

L'usuari seleccionarà els l'assaig o programa d'assajos que desitja per veure les estadístiques d'assistència.

Procés

L'aplicació genera les estadístiques segons els criteris definits per l'usuari.

Eixida

L'usuari veurà en pantalla el resum estadístic desitjat.

3.1.12. Enviament automàtic d'avisos d'assajos

Introducció

Una vegada definit cada assaig d'un programa d'assajos, el sistema enviarà automàticament avisos a tots els músics de la banda informant-los de l'assaig creat.

Entrades

S'haurà d'habilitar aquesta característica a la configuració del component, indicant els dies d'antelació i el(s) tipus d'avís.

Procés

L'aplicació envia automàticament l'avís als músics de la banda.

Eixida

Cap.

3.1.13. Creació, modificació i esborrament d'actes al carrer

Introducció

Es podran crear, modificar i eliminar actes al carrer.

Entrades

L'usuari haurà d'introduir les dades necessàries per a la creació/modificació d'aquest event, així com la relació de músics que hi participaran a l'acte; o bé selecciona l'acte al carrer a esborrar.

Procés

L'aplicació crea/modifica l'acte al carrer a la base de dades; o bé l'elimina.

Eixida

L'usuari rebrà un missatge de confirmació i tot seguit veurà la llista d'actes al carrer actualitzada.

3.1.14. Generació de llistes per a imprimir

Introducció



Per a un acte al carrer, es podrà generar un document per a imprimir la llista de músics que participaran a l'acte.

Entrades

La llista de músics prèviament creada.

Procés

L'aplicació genera un document PDF per a ser imprès.

Eixida

El document per a imprimir.

3.1.15. Introducció de dades d'assistència, porres i incidències

Introducció

Donat un acte al carrer, es podran introduir quins músics han assistit a l'acte, determinar porres, si s'escau, i/o incidències.

Entrades

L'acte al carrer prèviament creat i la relació de músics assignats. D'aquesta relació l'usuari marcarà quins músics han assistit i especificarà, si cal, porres i incidències.

Procés

L'aplicació emmagatzema les dades introduïdes a la base de dades.

Eixida

L'usuari rebrà un missatge de confirmació.

3.1.16. Enviament automàtic d'avisos d'actes al carrer

Introducció

Una vegada definit l'acte al carrer i la relació de músics, el sistema enviarà automàticament avisos als músics seleccionats informant-los, amb un determinat nombre de dies d'antelació, amb la informació de l'acte al carrer de la qual estan citats. L'enviament d'avisos s'efectuarà també en el cas de ser modificat l'acte o la llista de músics associada.

Entrades

L'usuari crearà, modificarà o eliminarà un acte al carrer, juntament amb la llista de músics associada.

Procés

L'aplicació envia automàticament l'avís als músics citats a l'acte, i als que han deixat d'estar citats, si s'escau.

Eixida

L'avís enviat als músics.

3.1.17. Generació automàtica de la relació de músics citats a l'acte



Introducció

Al crear/modificar un acte al carrer, es pot donar l'opció de generar automàticament la llista de músics que aniran citats a dit acte. Aquesta llista serà creada a partir de criteris com assistència assajos, guanys acumulats durant la temporada, etc, que otorguen una baremació a cada músic, de forma que els músics que millor puntuació obtinguen seran els músics seleccionats.

Entrades

S'haurà d'introduir el nombre total de músics que es citaran a l'acte, així com el nombre de músics per especialitat.

Procés

L'aplicació calcula les baremacions de tots els músics i genera la llista amb aquells que tinguen major puntuació per cada especialitat..

Eixida

La relació de músics.

3.1.18. Cerca d'actes al carrer

Introducció

Buscar un o més actes al carrer a partir d'un criteri establert: per nom, per data o per localitat.

Entrada

L'usuari indicarà quin criteri vol emprar per a la búsqueda i quin text o valor a buscar segons el criteri seleccionat.

Procés

L'aplicació realitza una búsqueda a la base de dades i mostra un llistat d'actes que satisfan els criteris de búsqueda seleccionats.

Eixida

Apareix un llistat d'actes si hi ha resultats coincidents, o un missatge informant que no hi ha cap coincidència.

3.1.19. Liquidacions de diversos actes, per a tots els músics o individualment

Introducció

Es permetrà generar liquidacions per a diversos actes en conjunt, tant per a tots els músics a la vegada com per a un músic determinat.

Entrada

L'usuari indicarà la informació necessària per crear una liquidació (incloent els actes al carrer existents) i indicarà si la liquidació és individual o colectiva.

Procés

L'aplicació crea la liquidació a la base de dades.



Eixida

Apareix un missatge de confirmació.

3.1.20. Generació de liquidacions per a imprimir

Introducció

Per a cada músic, es generarà un document resum de la seua liquidació.

Entrades

El període de liquidació prèviament creat.

Procés

L'aplicació genera un document per a ser imprès.

Eixida

El document per a imprimir.

3.1.21. Creació, modificació i eliminació d'obres musicals

Introducció

Es permetrà donar d'alta i modificar l'arxiu d'obres musicals, així com esborrar-ne.

Entrada

L'usuari indicarà la informació necessària per crear/modificar un registre d'obra musical, o bé en seleccionarà un per a eliminar-lo.

Procés

L'aplicació crea/modifica el registre d'obra musical a la base de dades; o bé l'elimina.

Eixida

Apareix un missatge de confirmació.

3.1.22. Emmagatzemament de partitures en pdf

Introducció

Per a una obra musical, es podrà adjuntar-li documents PDF amb les partitures.

Entrades

El fitxer de partitures i l'obra musical a la que va destinat.

Procés

L'aplicació emmagatzema el fitxer al disc dur i l'associa la seua ruta en el registre de l'obra musical a la base de dades.

Eixida

L'usuari rebrà un missatge de confirmació.



2.1.23. Cerca de partitures

Introducció

Buscar una o més partitures a partir d'un criteri establert: per nom, per autor o per modalitat. O bé buscar-la pel seu codi de barres (mitjançant webcam o introduint-lo manualment).

Entrada

L'usuari indicarà quin criteri vol emprar per a la búsqueda i quin text o valor a buscar segons el criteri seleccionat.

Procés

L'aplicació realitza una búsqueda a la base de dades i mostra un llistat d'obres musicals que satisfan els criteris de búsqueda seleccionats.

Eixida

Apareix un llistat d'obres musicals si hi ha resultats coincidents, o un missatge informant que no hi ha cap coincidència.

3.1.24. Enviament d'avisos manuals

Introducció

Es permetrà l'enviament d'avisos al conjunt o subconjunts de membres de la societat musical.

Entrada

L'usuari seleccionarà els membres als quals se'ls farà arribar l'avís, així com el mitjà d'enviament.

Procés

L'aplicació envia l'avís a cadascun dels membres seleccionats.

Eixida

Cap.

3.1.25. Alta, modificació i eliminació d'educands

Introducció

Es podran donar d'alta, modificar i eliminar educands a la base de dades.

Entrada

L'usuari indicarà la informació necessària per crear/modificar un educand, o bé en seleccionarà un per a eliminar-lo.

Procés

L'aplicació crea/modifica l'educand a la base de dades; o bé l'elimina.

Eixida

Apareix un missatge de confirmació.



3.1.26. Assignació d'instruments musicals

Introducció

En la creació/modificació d'un educand s'especificarà quin(s) instrument(s) en possessió de la societat musical tindrà cedit(s) l'educand, si és que en té algun.

Entrada

L'usuari seleccionarà un instrument d'una llista. Òbviament l'instrument ha d'existir prèviament a la base de dades.

Procés

L'aplicació realitza una inserció/modificació a la relació educand-instrument de la base de dades.

Eixida

L'usuari rebrà un missatge de confirmació i tot seguit veurà la llista d'educands actualitzada.

3.1.27. Cerca d'educands

Introducció

Buscar un o més educands a partir d'un criteri establert: per nom, per antiguitat o per instrument.

Entrada

L'usuari indicarà quin criteri vol emprar per a la búsqueda i quin text o valor a buscar segons el criteri seleccionat.

Procés

L'aplicació realitza una búsqueda a la base de dades i mostra un llistat d'educands que satisfan els criteris de búsqueda seleccionats.

Eixida

Apareix un llistat d'educands si hi ha resultats coincidents, o un missatge informant que no hi ha cap coincidència.

3.1.28. Alta, modificació i eliminació de músics de la banda juvenil

Introducció

Es podran donar d'alta, modificar i eliminar músics juvenils a la base de dades.

Entrada

L'usuari indicarà quin o quins educands entraran a formar part de la banda juvenil, o bé en seleccionarà un per a eliminar-lo.

Procés

L'aplicació associa o desassocia els educands amb la banda juvenil a la base de



dades.

Eixida

Apareix un missatge de confirmació i apareix actualitzada la llista de membres de la banda juvenil.

3.2 Requeriments d'interfícies externes

3.2.1. Interfícies d'usuari

D'una banda, pel que fa a la interacció visual, l'usuari tindrà contacte amb l'aplicació a través d'un navegador web. D'altra banda, les entrades de l'usuari a l'aplicació es realitzaran mitjançant el ratolí i el teclat.

3.2.2 Interfícies hardware

El maquinari utilitzat per interactuar amb l'aplicació és qualsevol maquinari que siga capaç d'arrancar un sistema operatiu que suporti l'execució d'un navegador web gràfic. Actualment tots els sistemes operatius sí donen aquest suport, amb la qual cosa la universalitat de l'aplicació està assegurada. A més a més, és necessària la disponibilitat d'una interfície de xarxa per poder contactar amb el servidor de l'aplicació web, així com els dispositius d'Entrada i Eixida (I/O) com la pantalla, el teclat i el ratolí. Si es desitja utilitzar també la funcionalitat de lectura de codis de barres desde la webcam, es necessitarà una webcam prèviament instal·lada al sistema.

3.2.3. Interfícies software

L'única interfície software necessària és un navegador web, amb independència del sistema operatiu sobre el qual s'execute. D'altra banda, si es desitja utilitzar la funcionalitat de lectura de codis de barres, es necessita la instal·lació prèvia de l'aplicació *ZBar*.

3.3 Requeriments d'eficiència

No existeix cap requeriment d'eficiència.

3.4 Restriccions de disseny

3.4.1. Estàndards complits

- L'IEEE 830 per a la redacció d'aquest document.
- La notació, definicions i acrònims utilitzats son estàndards *de facto* utilitzats al món de les bandes de música.
- El llenguatge utilitzat per a la implementació és PHP, i el llenguatge natural de l'aplicació serà el valencià.

3.4.2. Limitacions hardware

No hi existeix cap limitació hardware.

3.5. Atributs



3.5.1. Seguretat

L'aplicació, per tal de minimitzar els efectes de les possibles pèrdues d'informació motivades per algun esdeveniment extern sobre el procés servidor (tall del subministre d'alimentació, error crític del Sistema Operatiu, etc.), realitzarà operacions d'escriptura a la base de dades de forma immediata després de cada transacció iniciada per l'usuari, de forma que s'evitarà mantenir un temps excessiu únicament en memòria els canvis de les dades de l'aplicació.

En quant a l'accés remot, s'haurà de garantir que l'accés a les dades que manipula l'aplicació manté un nivell de seguretat adequat per evitar usuaris malintencionats. També s'haurà de garantir un nivell mínim de seguretat a l'iniciar sessió, durant la sessió, i al finalitzar la sessió a l'aplicació.

3.5.2. Manteniment

L'aplicació es trobarà per defecte en valencià, tot i que s'habilitarà una plataforma multilinguatge.

Pel que fa a la instal·lació, aquesta serà molt senzilla i no requerirà de ningun personal especialitzat.

Per últim, en termes de manteniment del codi de l'aplicació, s'abordaran tècniques d'aplicació de patrons de disseny que incidisquen en la millora de la mantenibilitat del programari. A més, la codificació estarà nodrida de nombrosos comentaris per clarificar la lectura del codi font, i se li serà aplicada refactorització de codi.

3.6. Altres requeriments

3.6.1. Bases de dades

La base de dades contindrà informació sobre tot el domini del problema. S'emprarà el SGBD *MySQL* per tal de comunicar l'aplicació amb la capa de persistència.

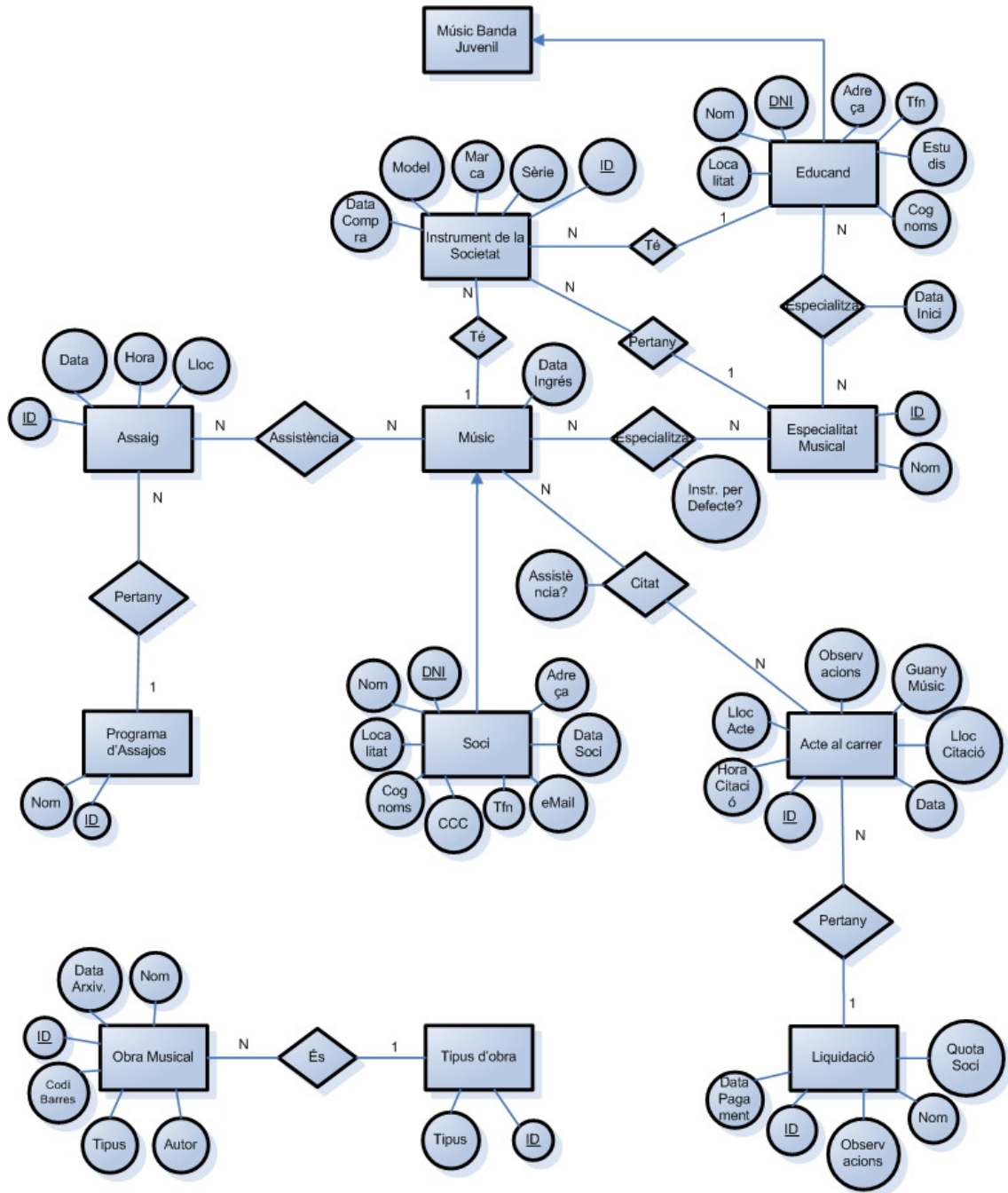
3.6.2. Operacions

A l'iniciar per primera vegada l'aplicació, l'usuari accedirà al sistema amb un usuari prèviament creat del qual se li proporcionaran les credencials.



3.2 Diagrama Entitat - Relació.

El disseny de l'aplicació s'ha centrat exclusivament en com s'estructurarà la informació que Prestíssimo! manipularà. Per tal de descriure aquesta estructura s'ha generat un diagrama entitat-relació, a partir del qual s'ha obtés posteriorment l'esquema relacional de la base de dades, i que es pot veure a continuació, en la figura 1.



R.1: Un instrument de la societat solament podrà estar en possessió d'un músic o d'un educand, però en ningun cas pot estar en possessió d'ambdues entitats.

Figura 1. Diagrama Entitat-Relació



Capítol 4. Implementació.

4.1 L'API de Joomla!

Com ja s'ha esmentat a l'apartat 2.6, capítol 2, Joomla ofereix una rica API amb multitud de mètodes i classes que proporcionen accés a diverses implementacions molt útils i fàcils d'accedir i utilitzar. Es pot consultar tot allò que ens ofereix l'API de *Joomla!* a la següent adreça web: <http://api.joomla.org> .

Tot seguit anem a demostrar, amb exemples extrets del codi de *Prestíssimo!*, com usar l'API de *Joomla!*, així com els avantatges que suposa la utilització d'aquesta.

4.1.1 El paquet HTML

El paquet HTML conté un total de 24 classes que ens permeten accedir als seus mètodes (estàticament o mitjançant una instància de la classe) per tal de generar dinàmicament codi HTML amb elements de presentació, com són llistes de selecció (comboBox), pestanyes (tabs), botons, etc. Seguidament veurem algun exemple d'ús d'alguna d'aquestes classes a *Prestíssimo!*.

4.1.1.1 La classe JHTML

La classe JHTML conté diversos mètodes estàtics que generen codi HTML de propòsit més general, com els mètodes `date()`[torna una data formatada], `image()`[genera el codi HTML d'una imatge], `link()`[genera el codi HTML d'un enllaç web], `script()` o `stylesheet()` [generen el codi necessari per incloure a la pàgina web generada arxius d'scripts en Javascript o de Fulles d'Estil en Cascada (css)].

Anem a mostrar l'exemple d'ús del mètode estàtic `calendar()`, que mostra un calendari quant es clica un control determinat d'un formulari (en aquest cas, un quadre de text o textbox).

(components/com_instruments/instruments.html.php, línies 440 i 469)

```
$calendari=JHTML::calendar("", "data_compra", "data_compra", "%d-%m-%Y");  
(...)  
<tr>  
<td class="titolCampsCreacioModificacio" width="10%">  
<?php echo JText::_("Data Compra"); ?></td>  
<td><input name="data_compra" id="data_compra" type="text"></td>  
</tr>
```

Segons l'API de Joomla, la definició del mètode `calendar()` és el següent:

```
void calendar (string $value, string $name, string $id, [string $format = '%Y-%m-%d'],  
[array $attribs = null])
```

on:

- \$value: el valor per defecte de la data.
- \$name: el nom del camp de text sobre el qual apareixerà el calendari.
- \$id: l'id del cap de text.
- \$format: el format de la data. (Opcional)
- \$attribs: atributs html addicionals. (Opcional)

El resultat és el que es mostra a la figura 2 següent:



Figura 2. JHTML::Calendar

4.1.1.2. La classe JPaneTabs

La classe `JPaneTabs` és una classe estàtica que implementa la classe abstracta `Jpane`. Aquesta permet representar un conjunt de pestanyes o "tabs", on cada pestanya conté el seu codi propi. Aquesta classe ha de ser instanciada, i ofereix mètodes per delimitar l'inici i el final del panell [`startPane()` i `endPane()`] que conté les pestanyes, així com mètodes per delimitar l'inici i fi d'una pestanya [`startPanel()` i `endPanel()`]. Podem comprovar el funcionament al següent exemple:

(components/com_avisos/avisos.html.php, línies 194 - 226)

```
$pane =& JPane::getInstance('tabs'); (1)  
  
echo $pane->startPane( 'paneAvisos' ); (2)  
echo $pane->startPanel( JText::_("Tots"), 'tots'); (3)  
  
mostrarPaneTots();  
  
echo $pane->endPanel(); (4)
```



```
echo $pane->startPanel(JText::_("Socis"), 'socis' );  
  
    mostrarPaneSocis($socis);  
  
echo $pane->endPanel();  
echo $pane->startPanel(JText::_("Musics"), 'musics' );  
  
    mostrarPaneMusics($musics);  
  
echo $pane->endPanel();  
echo $pane->startPanel(JText::_("Educands"), 'educands' );  
  
    mostrarPaneEducands($educands);  
  
echo $pane->endPanel();  
echo $pane->startPanel(JText::_("Membres BJ"), 'membresBJ' );  
  
    mostrarPaneMembresBJ($membresBJ);  
  
echo $pane->endPanel();  
echo $pane->startPanel(JText::_("Manualment"), 'manual' );  
  
    mostrarPaneManualment();  
  
echo $pane->endPanel();  
echo $pane->endPane(); (5)
```

A l'exemple podem comprovar com en (1) no s'instancia la classe JPaneTabs, sinó que s'obté una instància genèrica de pestanyes ja existent en el sistema *Joomla*, a través del mètode `getInstance()` de la classe superior (JPane). Una vegada tenim la instància, iniciem el panell de pestanyes assignant-li un codi o id, i anem definint pestanyes mitjançant (3) i (4). Entre la definició de cada pestanya s'inclou el codi que ha de mostrar-se a la pàgina web (en aquest cas són crides a funcions que retornen codi HTML). Cal destacar que la crida als mètodes d'inici i fi de panell i pestanyes requereixen d'una instrucció php `echo`, ja que aquests generen codi HTML que ha d'eixir i mostrar-se al generar-se la pàgina web.

El resultat és el que es mostra a la figura 3 següent:

Figura 3: JPaneTabs

4.1.1.3. La classe JHTMLSelect

JHTMLSelect és una classe estàtica que conté mètodes relacionats amb la generació de codi HTML de llistes de selecció (select's o combobox's) d'un formulari, entre d'altres. Trobem mètodes com `booleanlist()`, `integerlist()`, `radiolist()`, o `genericlist()`. Anem a veure un exemple d'ús a *Prestíssimo!* d'aquest darrer mètode, que genera un combobox o grup Option (<Option> en HTML)

Segons l'API de *Joomla!*, la definició del mètode `genericlist()` és la següent:

```
string genericlist (array $arr, string $name, [string $attrs = null], [string $key = 'value'],  
[string $text = 'text'], [mixed $selected = NULL])
```

on:

- `$arr`: Un array d'opcions.
- `$name`: El valor de l'atribut HTML name que se li assignarà a l'etiqueta Select.
- `$attrs`: Atributs HTML addicionals per a l'etiqueta Select. [Opcional]
- `$key`: El nom de l'objecte variable per al valor de l'etiqueta Select. [Opcional]
- `$text`: El nom de l'objecte variable per al text de l'etiqueta Select. [Opcional]
- `$selected`: La 'key' de l'opció seleccionada per defecte. [Opcional]

El vector `$arr` conté en cada posició el codi HTML d'una etiqueta <Option>, que es pot generar amb un mètode de la mateixa classe, `option()`. La definició d'aquest mètode és el següent:

```
object option (string $value, [string $text = "], [string $value_name = 'value'], [string $text_name = 'text'], [ $disable = false])
```

on:

- \$value: El valor de l'atribut HTML 'value' de l'etiqueta Option.
- \$text: El valor de l'atribut HTML 'text' de l'etiqueta Option. [Opcional]
- \$value_name: El nom de la propietat 'value' tornat a processar el formulari. [Opcional]
- \$text_name: El nom de la propietat 'text' tornat a processar el formulari. [Opcional]
- \$disable: Habilita o deshabilita per defecte l'opció. [Opcional]

D'aquesta forma, crear dinàmicament una llista de selecció desplegable és tasca senzilla:

(components/com_instruments/instruments.html.php, línies 430 - 437)

```
$opcions=array();  
$opcions[]=JHTMLSelect::option('0', JText::_("Selecciona una Especialitat"));  
    foreach($especialitats as $esp)  
    {  
        $opcions[]=JHTMLSelect::option($esp->ID_ESPECIALITAT, $esp->NOM);  
    }  
  
$selectEspecialitats=JHTMLSelect::genericlist($opcions, "select_especialitat");
```

Posteriorment, per mostrar la llista de selecció s'hauria d'executar la següent instrucció:

```
echo $selectEspecialitats;
```

L'eixida HTML generada a partir d'aquest codi PHP és el següent:

```
<select name="select_especialitat" id="select_especialitat">  
<option value="0">Selecciona una Especialitat</option>  
<option value="20">Clarinet</option>  
<option value="13">Trompa</option>  
</select>
```

I el resultat és el que es mostra a la figura 4 següent:



The screenshot shows a form with three fields. The first field, labeled 'Especialitat:', is a dropdown menu with a blue border and a downward arrow. The selected option is 'Selecciona una Especialitat'. Below the dropdown, the options 'Clarinet' and 'Trompa' are visible. The second field, labeled 'Model:', is empty. The third field, labeled 'Data Compra:', contains the text '00-00-2000'. There is an asterisk (*) to the right of the dropdown menu, indicating it is a required field.

Figura 4: JHTMLSelect

4.1.2 El paquet Database

El paquet Database ens facilita una sèrie de classes amb uns mètodes que serveixen de middleware entre la lògica de l'aplicació i la capa de persistència, permetent treballar amb la Base de Dades d'una forma molt més ràpida i senzilla.

Aquest paquet consta d'una classe abstracta, `JDatabase`, que defineix la interfície que han de tindre les classes descendents d'accés a bases de dades; a més de dues classes que implementen aquesta interfície o definició: `JDatabaseMySQL` i `JDatabaseMySQLi`. En el cas de *prestissimo* hem emprat la classe que implementa l'accés a bases de dades MySQL, `JDatabaseMySQL`.

Podem trobar una gran quantitat de mètodes que ens faciliten treballar amb la base de dades: desde control de transaccions (mètodes `BeginTrans()`, `EndTrans()` o `RollbackTrans()`), tots ells utilitzats a *Prestissimo!* i que proporcionen un segur de coherència en les dades, passant per múltiples mètodes d'obtenció de dades tornades per una sentència `SELECT`: `loadResult()`, `loadResultArray()`, `loadObject()`, `loadObjectArray()`, `loadAssoc()`, `loadAssocArray()`, `loadRow()`, `loadRowList()`, `getNumRows()`, etc.

Les possibilitats són infinites, entre les quals destaquem la possibilitat de fusionar els paradigmes objectual-relacional, de forma que es poden crear objectes que, sense implementar cap sentència SQL, poden ser extrets, modificats o eliminats de la base de dades de forma molt ràpida i senzilla. No obstant això, aquesta possibilitat no s'ha contemplat a *Prestissimo!* i per tant no és motiu d'aquesta explicació.



4.1.2.1 La classe JDatabaseMySQL

Seguidament es mostraran diversos exemples d'ús de la classe JDatabaseMySQL.

a) Consultes SELECT

(components/com_avisos/avisos.php, línies 74 - 82)

```
$db=& JFactory::getDBO(); (1)

$query="SELECT DNI, NOM, COGNOMS, EMAIL, MOBIL FROM soci ORDER BY
COGNOMS ASC"; (2)

$db->setQuery($query); (3)
$socis=$db->loadObjectList(); (4)

$query="SELECT e.DNI, e.NOM, e.COGNOMS, e.EMAIL, e.MOBIL FROM educand e,
music_juvenil mj WHERE e.DNI = mj.DNI ORDER BY COGNOMS ASC";

$db->setQuery($query);
$membresBJ=$db->loadAssocList(); (5)
```

Com podem comprovar, el treball amb Bases de Dades a *Joomla!* és senzill. En primer lloc, obtenim una instància ja existent de la connexió (o cursor) a la Base de Dades, o si no existeix se'n crea una automàticament, a partir del mètode estàtic de la classe `Jfactory getDBO()`, que implementa el patró de disseny Singleton (1). Aquesta simple instrucció ens garanteix un nivell de seguretat extra, ja que en cap cas s'ha d'establir manualment la connexió, evitant així emmagatzemar la contrassenya d'accés a la BD en el nostre component o fitxer php.

Una vegada obtesa la instància del cursor base de dades, ja es poden emprar tots els mètodes que la classe `JDatabaseMySQL` posa a la nostra disposició. S'ha de crear una cadena que continga la sentència SQL a executar-se, en aquest cas `$query` (2). Tot seguit, s'estableix quina és la sentència que s'executarà, mitjançant el mètode `setQuery()`, al que li passem la cadena de la instrucció SQL (3). Com que es tracta d'una consulta de sols lectura, no es necessari confirmar l'operació amb el mètode `query()`, per la qual cosa s'executa automàticament.



Per acabar, emmagatzemem les dades tornades per la sentència Select en una variable mitjançant algun dels múltiples mètodes que ho permeten, seleccionant el que més interesse, sempre atenent a si el nombre de tuples resultants de la consulta siguen una o més d'una. En (4), la instrucció loadObjectList() torna un array d'objectes, on cada element de l'array és un Objecte de tipus stdClass, que té els mateixos atributs (en quantitat i nom) que els atributs seleccionats en la sentència Select (p.e. per accedir al DNI de la primera tupla tornada: `$socis[0]-> DNI`). Per contra, en (5), la instrucció loadAssocList() torna una llista associativa o array multidimensional, on cada fila representa una tupla, i cada columna representa un atribut (p.e. per accedir al DNI de la primera tupla tornada: `$membresBJ[0][“DNI”]`).

b) Sentències INSERT, UPDATE, DELETE.

La diferència d'ús entre les sentències Select i les sentències Insert, Update i Delete és que aquestes últimes requereixen confirmar l'operació mitjançant una crida al mètode query() després del setQuery(), per la qual cosa no és necessari mostrar cap exemple addicional. Això sí, l'ús d'aquestes sentències pot requerir l'ús de mètodes relacionats amb el control de transaccions (mètodes BeginTrans(), EndTrans() o RollbackTrans()), com també realitzar un control d'errors (el mètode query() torna false si ha fallat la sentència).

4.1.3 El paquet Mail

Aquest paquet inclou les classes JMailHelper i JMail. Aquesta darrera és una classe instanciable que ofereix una interfície senzilla d'enviament d'e-mails.

4.1.3.1 La classe JMail

La classe JMail requereix ser instanciada, on una instància representa un correu electrònic a enviar. Aleshores s'ha de determinar quin o quins són els destinataris, qui és el remitent, l'assumpte i el cos o text del correu, etc. Tot això es pot realitzar mitjançant els mètodes que ens proporciona aquesta classe, com són addRecipient(), setSender(), setSubject() o setBody(). Finalment, per enviar el correu electrònic cal cridar al mètode send().

Al següent exemple, extret del component de gestió d'avisos, podem veure com utilitzar aquesta interfície d'enviament d'e-mails proporcionada per *Joomla!*.

(components/com_avisos/avisos.php, línies 323 - 362)

```
$email= new JMail();

//Incloem el prefix de l'assumpte del fitxer de configuració.
if(strlen($email_prefix_assumpte) > 0)
    $assumpte=$email_prefix_assumpte." ".JRequest::getVar('assumpte');
else $assumpte=JRequest::getVar('assumpte');

$email->setSubject($assumpte);

//Assignem com a remitent els paràmetres fixats al fitxer de configuració
$enviatPer=array();
$enviatPer[0]=$email_adr_remitent;
$enviatPer[1]=$email_nom_remitent;
$email->setSender($enviatPer);

//Incloem la capçalera Reply-To si s'ha especificat a la configuració
if($email_reply_to_habilitat)
{
    $reply_to=array();
    $reply_to[0]=$email_reply_to_adr;
    $reply_to[1]=$email_reply_to_nom;
    $email->addReplyTo($reply_to);
}

$email->setBody($text);
$email->addRecipient($destinatari);

//Enviem mitjançant SMTP si s'ha especificat a la configuració
```



```
if($email_smtp_habilitat) (1)
{
    if(strlen($email_smtp_autenticacio)==0) $autenticacio=NULL;

    if($email->useSMTP($autenticacio, $email_smtp_host, $email_smtp_usuari,
$email_smtp_contrassenya)) (2)

        $missatge=JText::_("Avis enviat correctament via e-mail.");

    else $missatge=JText::_("<b><font color=\"red\">Error a l'enviar l'avis via e-
mail (usant SMTP).</font></b>");
}
else
{
    if($email->Send()) $missatge=JText::_("Avis enviat correctament via e-mail."); (3)
    else $missatge=JText::_("<b><font color=\"red\">Error a l'enviar l'avis via
email</font></b>.");
}
}
```

En el component de gestió d'avisos s'ofereix un apartat de configuració en el qual l'usuari pot seleccionar quin és el remitent dels e-mails (nom i e-mail), o fins i tot configurar un servidor SMTP autenticat per a l'enviament de correus electrònics mitjançant aquest protocol, en lloc d'emprar la llibreria PHPMailer. Aquests paràmetres de configuració s'emmagatzemen a l'arxiu `config_avisos.php`, el qual és importat al component [instrucció `require('components/com_avisos/config_avisos.php');`], per tal de poder emprar els valors de configuració com simples variables php (com per exemple `$email_smtp_habilitat`, `$email_adr_remitent`, etc.).

Depenent de si a l'arxiu de configuració s'ha activat l'enviament d'emails via SMTP (1), l'enviament del correu electrònic es realitza mitjançant el mètode `send()` (3) o mitjançant el mètode `useSMTP()` (2). A més a més, l'ús d'aquests mètodes permet controlar si s'ha produït cap errada en l'enviament del correu electrònic.

Al llarg d'aquests exemples hem vist una xicoteta mostra de la potent i àmplia API que ofereix Joomla! als programadors que desitgen convertir aquest CMS en una veritable aplicació web professional. Sense lloc a dubte aquest és un dels punts forts d'aquesta plataforma, però hi ha molt més, com seguidament es comprovarà.



4.2 Estructura dels Components Joomla!

Els components en Joomla! estan formats per dos arxius php. Aquests serveixen per a dividir el codi en dues capes clarament diferenciades, l'una de treball amb la Base de Dades i l'altra de presentació i interacció amb l'usuari.

Pel que fa a l'arxiu d'interacció amb la base de dades, podem dir que es tracta d'un arxiu purament en php que està format per un "switch" o elecció de la tasca a realitzar i tants mètodes o "function" com tasques es puguen realitzar.

Al següent exemple extret del component de gestió de la banda juvenil es pot observar com dependent de la tasca a realitzar (que s'obté de la interacció amb l'usuari) s'executa una funció o una altra (en aquest cas la d'eliminar Músics Juvenils).

(components/com_bandajuvenil/bandajuvenil.php, línies 15 - 31)

```
switch( JRequest::getVar( 'tasca' )) {  
  
    case 'afegir': afegirMJuvenil();  
    break;  
  
    case 'crear': crearNouMJuvenil($mjuvenil=array());  
    break;  
  
    case 'eliminar' : eliminarMJuvenil();  
    break;  
  
    case 'nou' : crearNouMJuvenil();  
    break;  
  
    default: mostrarMJuvenils();  
    break;  
  
}
```

(components/com_bandajuvenil/bandajuvenil.php, línies 15 - 31)

```
function eliminarMJuvenil(){  
    $db =& JFactory::getDBO();//instanciem la BD  
  
    $countcb = JRequest::getVar('countcb');//nombre de checkbox seleccionats per l'usuari
```



```
for($i=0; $i<$countcb; $i++){//per a cada checkbox, és a dir, per a cada MJ
$nomCb="cb".$i;

    if($checkBox=JRequest::getVar($nomCb) != null){//si s'ha seleccionat el checkbox en
concret
    $dniABuscar = "dni".$i;
    $dni= JRequest::getVar($dniABuscar);
    $query="DELETE FROM music_juvenil WHERE DNI='$dni'";
    $db->setQuery($query);
    $db->Query();

    }//fi de l'if

} //fi del for
mostrarMJjuvenils();
}
```

D'altra Banda, pel que fa a la capa de presentació i interacció amb l'usuari, podem dir que aquesta consta d'un arxiu PHP en simbiosi amb: d'una banda, etiquetes HTML que donen format a la presentació i d'altra funcions en JavaScript que permeten interaccionar amb l'usuari de forma eficient, tals com la validació de dades.

Al següent exemple podem veure com és la funció que mostra el formulari de modificació d'un soci, que consta de: Primerament, etiquetes HTML que formen una taula per a mostrar els diferents contenidors dels atributs del soci i, en segon lloc, d'instruccions en PHP que mostren els valors dels atributs

(components/com_socis/socis.html.php, línies 130 - 158)

```
function mostrarFormulariModificacioSoci($soci){

CrearCapsaleraTaulaRedondejada();

?>
<form id="form1" name="form1" method="post"
action="index.php?option=com_socis&tasca=modificacio">
<table class="taulaCreacioModificacio">
```



```
<tr><td><?php echo JText::_("Nom");?></td><td><input name="nom" id="nom" type="text"
value="<?php echo $soci->NOM; ?>"></td></tr>

<tr><td><?php echo JText::_("Cognoms");?></td><td><input name="cognoms" id="cognoms"
type="text" value="<?php echo $soci->COGNOMS; ?>"></td></tr>

<tr><td>DNI:</td><td><input name="dni" id="dni" type="text" value="<?php echo $soci->DNI;
?>"></td></tr>

<tr><td><?php echo JText::_("Telefon");?></td><td><input name="telefon" type="text"
value="<?php echo $soci->TELEFON; ?>"></td></tr>

<tr><td><?php echo JText::_("Mobil");?></td><td><input name="mobil" id="mobil" type="text"
value="<?php echo $soci->MOBIL; ?>"></td></tr>

<tr><td><?php echo JText::_("Localitat");?></td><td><input name="localitat" type="text"
value="<?php echo $soci->LOCALITAT; ?>"></td></tr>

<tr><td><?php echo JText::_("Direccio");?></td><td><input name="direccio" type="text"
value="<?php echo $soci->DIRECCIO; ?>"></td></tr>

<tr><td>Email:</td><td><input name="email" id="email" type="text" value="<?php echo $soci-
>EMAIL; ?>"></td></tr>

<tr><td>CCC:</td><td><input name="ccc" type="text" value="<?php echo $soci->CCC;
?>"></td></tr>

<tr><td><?php echo JText::_("Data Ingres");?></td><td><input name="data_soci" id="data_soci"
type="text" value="<?php echo $soci->DATA_SOCI; ?>"> </td></tr>
<?php JHTML::calendar('data_soci', 'data_soci', '%d-%m-%Y');?>

<tr><td><input type="hidden" name="dni_vell" value="<?php echo $soci->DNI; ?>">

</td></tr>
</table>
</form>
<?php
CrearPeuTaulaRedondejada();
}
```



4.3 Multilinguatge i Joomla!

Com ja s'ha explicat anteriorment, Joomla! permet presentar aplicacions en format de multilinguatge, i d'aquesta forma donar un toc d'universalitat a qualsevol aplicació desenvolupada baix el seu entorn.

Per a aconseguir-ho es necessita un fitxer de traducció per a cada llenguatge i per a cada component. Aquests fitxers estan estructurats en diferents carpetes que separen els arxius de traducció dels diferents idiomes. Així, al directori on s'ubica prestíssimo, podem encontrar dues carpetes que corresponen als dos idiomes en que s'ofereix l'aplicació, és a dir, en valencià i en castellà. Dins de cada carpeta es podrà trobar, per tant, tants arxius de traducció de l'idioma en qüestió com components existisquen a l'aplicació i necessiten d'eixa traducció.

Al següent exemple podem veure com des d'un component, s'indica la voluntat de mostrar un text traduït a l'idioma seleccionat per l'usuari (mitjançant la funció `JText::_("text a traduir")`) i com es busca el text a l'arxiu desitjat per a mostrar la traducció.

(components/com_socis/socis.html.php, línies 106 - 111)

```
function mostrarErrorDNIDuplicat($dni) {  
    ?><script language="javascript">  
    alert("<?php echo JText::_("Ja existeix algun soci amb DNI ");?><?php echo($dni)?>");  
    </script>  
    <?php  
}
```

I buscant a l'arxiu de traducció trobem la traducció (p.e. al castellà).

(C:\AppServ\www\Proves15\language\es-ES\es-ES.com_socis, línies 8 - 14)

```
S'HA PRODUI'T UN ERROR AL REALITZAR UNA TRANSACCIO EN LA BASE DE DADES, PER  
FAVOR, COMUNICA-HO A L'ADMINISTRADOR DEL SISTEMA. =Se ha producido un error al
```

realizar la transacción en la Base de Datos, por favor, comunicalo al administrador del sistema para subsanarlo.

EL NOM I COGNOMS DEUEN DE TINDRE CONTINGUT =El nombre i los apellidos deben de tener contenido

LA DATA INTRODUIDA NO ES CORRECTA =La fecha introducida no es correcta

EL MOBIL INTRODUIT NO ES CORRECTE =El móvil introducido no es correcto

L'EMAIL INTRODUIT NO ES CORRECTE =El email introducido no es correcto

ACTUALMENT NO HI HA SOCIS =Actualmente no hay socios.

JA EXISTEIX ALGUN SOCI AMB DNI =Ya existe algún socio con DNI

Per tant, a l'aplicació, el resultat obtingut seria el que es mostra a la figura 5 següent:

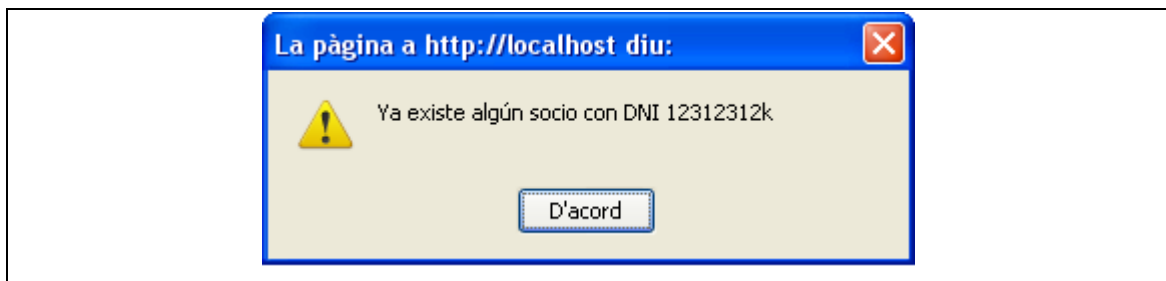


Figura 5: Missatge d'alerta traduït al castellà



4.4 Capa de Presentació. Plantilles en Joomla!

Un altre dels punts forts d'aquest CMS és la possibilitat de dissenyar atractives interfícies gràfiques totalment independents del sistema, les quals s'anomenen plantilles. Aquestes defineixen una estètica de presentació comú a tot el lloc web, i delimiten unes zones o llocs on es situarà el codi HTML generat pels diferents mòduls i components que es mostraran. Des de la interfície d'administració es poden afegir fàcilment plantilles i seleccionar quina plantilla de les disponibles serà la que es mostre per als visitants web.

L'avantatge que presenten les plantilles de *Joomla* és que a l'hora de dissenyar-les podem centrar-nos únicament en l'apariència o aspecte. No hem de preocupar-nos ni per la lògica de l'aplicació ni per cap altre aspecte que no siga purament de presentació. Únicament s'han d'especificar quines zones de la plana web (diguem-li contenidors) contindran els continguts dinàmics que s'han de mostrar-se.

Per entendre-ho, cal saber que una plana web en *Joomla!* defineix una sèrie de posicions amb noms predefinits pel sistema base, com per exemple "left", "right", "bottom", "banner", "component", etc. En aquestes posicions es poden situar els mòduls (o els components, però únicament i per defecte en la posició "component") disponibles. Per exemple, en el cas de *Prestíssimo*, el mòdul mod_login (que serveix per autenticar els usuaris) es troba situat a la posició "left" (i en efecte, apareix a l'esquerra de la pantalla, que és la posició "left" definida per la plantilla). Ara bé, la plantilla és l'encarregada de definir on es troba cada posició predifinida, és a dir, en quin lloc apareixeran els moduls que es configuren per a situar-se en una posició determinada. Generalment la posició "left" s'ubica en una columna a l'esquerra, la posició "component" és la zona central i la més gran (on apareix el codi html generat per un component), etc. Però insistim, la decisió d'ubicar aquestes posicions recala en el desenvolupament de la plantilla, podent trencar perfectament un posicionament raonable (per exemple, determinar que la posició "left" es mostrarà a la part inferior de la pantalla).

El desenvolupament de *Prestíssimo* no ha requerit desenvolupar una plantilla desde zero. En el seu lloc s'ha modificat i adaptat una plantilla existent, que s'inclou per defecte



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



en el sistema base, anomenada “ja_purity”. Bàsicament les modificacions realitzades s’han centrat en l’aspecte visual, sobretot en quant a les combinacions de colors (mantindre l’estètica elegida per a l’aplicació web, combinant les tonalitats negres, blanques i grises, que contrasten amb el color verd oliva del logotip de l’aplicació), a més de modificar els logotips i incloure una signatura en la part inferior de la plana web. No s’ha realitzat ninguna modificació de l’estructura de les posicions predefinides.

4.5 Creació i lectura de codis de barres.

Com s'ha comentat anteriorment, l'aplicació ofereix la possibilitat de la lectura de codis de barres o QR mitjançant una webcam. Aquesta funcionalitat resulta útil en el moment en que l'aplicació ha de conèixer la realització de pagaments als músics pel que fa a les seues liquidacions. És per això, que en el document PDF que crea l'aplicació com a liquidació s'inserten tant un codi de barres com un codi QR que identifiquen a quin músic correspon la liquidació i la pròpia liquidació, com es pot comprovar a la figura 6 següent.



Figura 6. Codi de barres i QR a la liquidació

Quant al format dels codis de barres o QR, donat que l'aplicació permet llegir infinitat de formats i observant les característiques de cadascun, s'ha optat per utilitzar els següents:

- Per a una liquidació individual: Format Code-128C, que permet formatar cadenes numèriques de longitud variable. En aquest cas l'aplicació utilitza el dígit 1 per a indicar que es tracta d'una liquidació individual (amb el número 1), els dígits 2-9 per a indicar el músic de la liquidació i els dígits 10-20 per a indicar la liquidació corresponent.

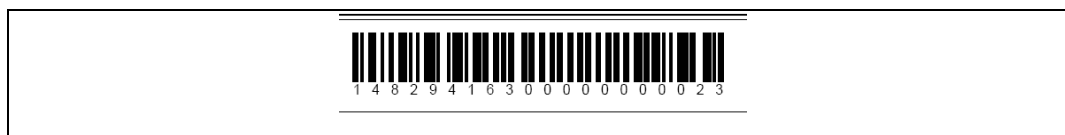


Figura 7. Codi de barres Code-128C

- Per a una liquidació col·lectiva: Format EAN-13, que permet formatar cadenes numèriques de 12 caràcters i insereix 1 caràcter extra com a dígit control. En aquest



cas l'aplicació utilitza el dígit 1 per a indicar que es tracta d'una liquidació col·lectiva (amb el número 0), i els dígit 1-12 per a indicar la liquidació corresponent.



Figura 8. Code de barres EAN-13

- Per a ampliar les possibilitats de lectura de codis s'utilitza també, en els dos casos anteriors, un codi bidimensional anomenat codi QR. Aquest format permet emmagatzemar cadenes alfanumèriques de longituds molt disperses. En aquest cas l'aplicació indica amb una cadena alfanumèrica quina liquidació correspon un identificador de a qui va dirigida la liquidació.



Figura 9. Codi QR

Quant a la lectura dels codis de barres, com s'ha indicat anteriorment, es realitza mitjançant l'aplicació ZBarCam. Açò és degut a que no s'ha trobat cap llibreria susceptible de ser utilitzada amb PHP o Joomla. No obstant, aquesta funció s'ha elaborat de forma que queda totalment incrustat en l'aplicació Prestússimol!. Açò s'ha aconseguit cridant a l'aplicació ZBarCam directament des de l'aplicació Prestússimol!. Ha estat possible gràcies a la funcionalitat del llenguatge PHP que permet la crida d'altres aplicacions.

La forma en que es crida a l'aplicació de lectura de codis de barres és mitjançant una línia de comandaments oculta. Aquesta funcionalitat de PHP s'aconsegueix utilitzant la funció COM, que permet instanciar aplicacions i entrar a les seues propietats i mètodes. Com es pot observar al codi següent, es realitza una crida per mitja de la línia de comandaments a l'aplicació ZBarCam de forma que es mostre en primer pla.

(components/com_pagaments/pagaments.php línies 96-98)

```
/*Executem el programa en primer pla "/R" i redirigim l'eixida al fitxer */  
$WshShell = new COM("WScript.Shell");  
$oExec = $WshShell->Run("cmd /R $path_zbarcam > c:\codi_barres\codi_barres.txt", 0,  
false);
```

Amb el codi anterior, s'aconsegueix cridar a l'aplicació ZBarCam i redirigir l'eixida del programa a un fitxer, on es guardarà el codi llegit. Pel que fa a l'apariència visual, la finestra de captura de codis de barres s'executa en primer pla i queda totalment integrat a l'aplicació prestíssimo!, com es pot observar a la següent figura.

The screenshot shows a web browser window with a sidebar menu on the left and a main content area. The sidebar menu includes items like 'Gestió de Músics', 'Pagaments', and 'Assajos'. The main content area displays a 'Pagaments' section with a table of pending payments. A small window titled 'ZBar' is overlaid on the top left, showing a barcode with the ISBN 978-84-672-3345-2 and the number 26930. The browser's address bar shows 'com_pagaments&Itemid=148&lang=ca'.

Setembre i Octubre			
Nom	Cognoms	Quantitat	
<input type="checkbox"/>	Àngel	Bernat Ferrer	330.00
<input type="checkbox"/>	Gemma	Molina Puerto	333.00

Novembre i Desembre			
Nom	Cognoms	Quantitat	
<input type="checkbox"/>	Gemma	Molina Puerto	-10870657.00

setembre 2011			
Nom	Cognoms	Quantitat	
<input type="checkbox"/>	Àngel	Bernat Ferrer	435.00
<input type="checkbox"/>	Gemma	Molina Puerto	115.55

Figura 10. Lectura de codis de barres



4.6 Implementació de la Base de Dades

Joomla! porta implementada una base de dades pròpia en el llenguatge SQL que utilitza tant per a emmagatzemar dades necessàries per al nucli que el fa funcionar com per a emmagatzemar les dades necessàries per a fer funcionar les extensions principals (mòduls, components, etc.).

Per tant, per a la persistència de les dades que contitueixen el domini d'informació de *Prestíssimo!* no és necessari crear una base de dades nova, simplement cal afegir les taules necessàries a la base de dades utilitzada pel Joomla!. Hi ha diverses formes d'afegir-ho, una seria pas a pas, i utilitzant l'administració de la base de dades que proporciona Joomla!, anar afegint cadascun dels atributs i taules, i l'altra, la utilitzada en aquest projecte, executar un script en llenguatge SQL per a afegir les noves taules necessàries per a cobrir tota la funcionalitat desitjada.

La implementació de la base de dades s'ha derivat directament de la fase de disseny, a partir del diagrama Entitat-Relació. L'esquema relacional de la base de dades obtés a partir d'aquest procés d'inferència s'ha introduït de forma gràfica en un programari de generació automàtica d'scripts SQL, per tal d'evitar escriure tot el codi "a mà".

En el següent exemple es pot observar part de l'script SQL utilitzat per a crear la base de dades de *Prestíssimo!*:

```
/* Table: SOCI */
create table SOCI
( DNI          varchar(20) not null,
  NOM          varchar(30),
  COGNOMS     varchar(40),
  LOCALITAT   varchar(30),
  DIRECCIO    varchar(40),
  EMAIL       varchar(40),
  CCC         varchar(30),
  TELEFON     varchar(20),
  MOBIL       varchar(15),
  DATA_SOCI  varchar(10),
  primary key (DNI)
```



```
);  
  
/* Table: MUSIC */  
  
create table MUSIC  
(  
  DNI          varchar(20) not null,  
  DATA_MUSIC  varchar(10),  
  primary key (DNI)  
);  
  
alter table MUSIC add constraint FK_REFERENCE_4 foreign key (DNI)  
  references SOCI (DNI) on delete restrict on update restrict;
```



Capítol 5. Conclusions i futur de l'aplicació.

5.1 Resum del treball realitzat.

Prestíssimo! és una aplicació que busca poder ser utilitzada de forma universal per qualsevol societat musical, és per això que s'ha intentat seguir un esquema propi de les bandes tal com l'aparició de socis capitalistes, les bandes juvenils que es formen per a entrar a la banda en quan un educand es trobe en condicions, etc.

És per això, que el primer treball realitzat va ser informar-se sobre l'estructura que forma una organització musical i, per tant, vam acudir a l' "Associació Unió Musical Bocairent", en la que ens van oferir tota la informació necessària.

La següent tasca a realitzar era dissenyar la Base de Dades, d'acord amb la informació recopilada sobre l'estructura de les societats musicals, i plasmada en el diagrama Entitat-Relació. Seguidament es va obtenir l'esquema relacional de la base de dades, el qual es va introduir visualment a un programa anomenat "Sybase Power Designer" per tal de generar un script SQL a partir de l'esquema visual de la base de dades.

Posteriorment, i pel que fa a la implementació, va ser necessari adaptar una plantilla a les nostres necessitats, i definir-nos una xicoteta API per generar taules i capçaleres amb la mateixa apariència visual i amb la mateixa funcionalitat (botons, etc). La intenció d'això és que un usuari qualsevol aprenga ràpidament el funcionament de l'aplicació i no tinga dificultat al realitzar operacions en distints components. Finalment, es va abordar la implementació, un a un, dels components que doten a l'aplicació de la funcionalitat requerida, amb les seves proves corresponents.

En resum, el que es busca a l'aplicació és poder ser emprada per un conjunt molt ampli de societats musicals. Amb aquesta finalitat s'ha creat una aplicació multilingatge que podrà ser traduïda a tants idiomes com siga necessari sense necessitat de canviar el codi. Una altra finalitat és que siga una aplicació moldejable per a cada societat musical, ja



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



que entenem que no totes les organitzacions segueixen un mateix patró de funcionament. És per això que s'ha creat una aplicació com a un conjunt de components relacionats entre ells però a la vegada independents. Exemplificant, una societat podria emprar Prestíssimo! si no disposen de banda juvenil simplement desinstal·lant dit component.



5.2 Futur de l'aplicació i futures ampliacions

Prestíssimo! és una aplicació que suporta perfectament els canvis i que ha estat dissenyada per a acceptar qualsevol ampliació o funcionalitat afegida, gràcies al suport que ofereix *Joomla!* al paradigma de l'enginyeria del software basat en components. Com que està format per diferents components, seria tant senzill com afegir-ne de nous per a donar cabuda a les necessitats que, en un moment donat, poguera tindre qualsevol societat musical.

En el document ERS està reflexat un estat final de l'aplicació si fóra destinat per a una societat musical. Com a PFC d'ampliació d'un anterior, s'han afegit components que doten a l'aplicació de més funcionalitat i seguretat. Per tant, l'aplicació *Prestíssimo!* consta ara d'una funcionalitat àmplia que podria substituir la gran part de tasques de gestió d'una societat musical. No obstant, com qualsevol altra aplicació, podria ser ampliada per a donar cabuda a nous processos de gestió requerits per alguna organització gràcies al disseny de l'aplicació mitjançant components.

5.3 Conclusions

Prestíssimo es tracta d'una aplicació web que té una clara visió de futur, ja que ha estat creada per a poder ser utilitzada a qualsevol societat musical i està dissenyada per a poder acoplar-se a qualsevol entitat i poder ampliar-se en qualsevol moment. A més de la facilitat de moldejar-se, la utilització del multilinguatge i l'empra del software Open-Source afavoreixen i incrementen el caràcter universal de l'aplicació.

Ens hem adonat de les facilitats que es donen per a treballar amb programari lliure i de la qualitat de software que ofereixen. Es pot realitzar una aplicació amb un acabat professional sense necessitat d'utilitzar programes de pagament.

He treballat en un llenguatge que no coneixia, com és PHP, i ha estat gratificant veure que no canvia molt dels altres llenguatges estudiats a la carrera i no ha resultat gens difícil afrontar el canvi. A més, a Internet he trobat moltes de les funcions o mètodes que necessitàvem (p.e. per connectar-se amb la base de dades) amb gran facilitat i que estaven perfectament explicats i exemplificats.

Joomla! ens ha facilitat el treball ja que, d'una banda, ens ha servit com a base per a desenvolupar la nostra aplicació web, facilitant-nos i evitant-nos feina que haguérem hagut de realitzar començant des de zero. D'altra banda, Joomla! ens ha facilitat la feina a l'hora d'implementar les funcions pròpies de *Prestíssimo!*, ja que al treballar amb components, hem pogut separar les funcionalitats amb gran facilitat.

Pel que fa a la funcionalitat, s'ha aconseguit crear una aplicació visualment atractiva i amb tota la funcionalitat que podria donar qualsevol altra aplicació d'escriptori, però amb l'avantatge de poder ser accedida des de diferents màquines. S'ha creat, per tant una aplicació que permet a qualsevol societat musical centralitzar i informatitzar la majoria de les seues gestions. Cal destacar, també, tant la funcionalitat de generació de documents PDF imprimibles, com la possibilitat de lectura de codis de barres o QR des de la webcam. Aquesta funcionalitat ha dotat a l'aplicació d'una alta professionalitat i facilita als usuaris de l'aplicació una utilitat que de segur que serà agraïda.



Quant a la realització de proves d'implantació i funcionament de l'aplicació, s'ha realitzat un "testeig" en viu, ja que l'aplicació ha estat instal·lada a un servidor local per tal de realitzar totes i cadascuna de les proves funcionals esmentades a l'especificació de requeriments.

Per últim, i pel que fa al projecte en general i a la universitat, estic content, ja que el PFC, al tractar-se d'un projecte sencer (des de l'anàlisi fins a la implementació) proporciona a l'alumne una visió del que pot encontrar-se al món laboral a l'eixir de la universitat. La valoració de la realització del PFC amb 15 crèdits és molt més justa que la valoració que es dona a la titulació d'Enginyeria Tècnica (6 crèdits). No obstant això, es continua devaluant el treball realitzat, ja que és necessiten moltes més hores per a abarcar la realització d'un programa des de l'inici i aconseguir un acabat funcional i professional.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA





Annexos

A.1 Bibliografia

- Funcionament de Joomla!

<http://api.joomla.org/index.html>

- Extensions de Joomla!

<http://extensions.joomla.org/>

- Servidor Apache

<http://httpd.apache.org/>

- SGBD MySQL

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

- Llenguatge JavaScript

<http://ca.wikipedia.org/wiki/JavaScript>

- Llenguatge de programació PHP

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

- Llibreria de creació de PDF

<http://www.tcpdf.org/>

- Lector de codis de barres: ZBar

<http://zbar.sourceforge.net/>

- Servidor de correu: ArGoSoft Mail Server

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

A.2 Manual d'usuari

L'usuari de Prestíssimo haurà d'autenticar-se a l'aplicació per a poder accedir al contingut de les gestions amb el nom d'usuari: 'prestissimo' i el password: 'prestissimo', ambdós sense cometes.

A l'aplicació s'ha buscat fer intuïtiva la gestió de qualsevol component de la societat, és així que simplement cal presionar sobre el botó verd amb símbol "+" per a afegir qualsevol cosa que es desitge. Una vegada presionat, s'hauran d'emplenar els camps necessaris per a donar d'alta el subjecte i presionar sobre l'opció de guardar (un disquet). Sempre que es vulga es podrà tornar enrere presionant sobre la fletxa cap a l'esquerra. Altres botons disponibles a l'aplicació serien el botó de configuració del component en qüestió, que s'indica amb una roda dentada i els botons de lectura desde la webcam o des del teclat, indicats cadascun amb la imatge corresponent. L'aplicació controlarà en tot moment que l'usuari plene correctament els camps necessaris per al bon funcionament de l'aplicació.

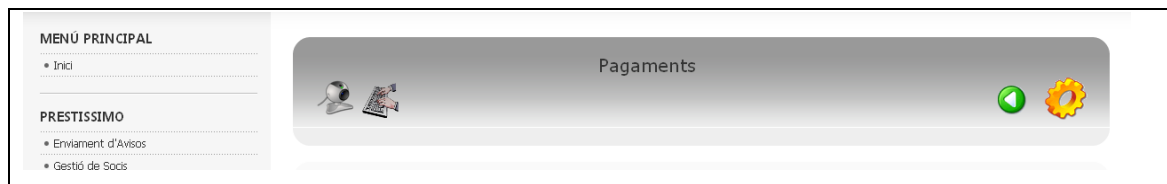


Figura 11. Capçalera de l'aplicació Prestíssimo!

Per accedir a l'aplicació des del navegador s'utilitzaria l'URL <http://localhost/prestissimo/>. *

D'altra banda, es pot accedir a l'administració del CMS Joomla amb la direcció <http://localhost/prestissimo/administrator> * cosa que permetrà administrar completament el sistema Joomla (gestió d'usuaris, gestió de components, administració de menus, etc.).

* Direcció vàlida suposant que s'està utilitzant un servidor local. Si s'utilitza un altre tipus de servidor, l'adreça hauria de ser modificada i canviar "localhost" per l'adreça al servidor.

Hi ha peculiaritats pròpies de l'estructura de les societats musicals, tals com que els membres de la Banda juvenil han de ser educands i que hi ha músics que poden ser a la



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



vegada socis de la societat. És per això, que per a donar d'alta un(s) membre(s) de la Banda juvenil, s'haurà de seleccionar-lo(s) d'un llistat d'educands que apareixerà per tal d'afegir-lo(s) a la Banda juvenil. D'altra banda, a l'hora de crear un músic, aquest podrà crear-se a partir d'un soci (es carregarà automàticament les seues dades) o desde zero.

A.3 Manual d'instal·lació

Aquests són els passos que hi ha que seguir per a poder instal·lar l'aplicació prestissimo:

- 1- **Descomprimir l'arxiu *prestissimo.zip*** (situat a la carpeta *Aplicacio* del cd) **en la carpeta arrel del servidor web**. És necessària la prèvia instal·lació d'un servidor (p.e. Apache).*

* Es recomana (si l'usuari no és expert) la instal·lació de l'aplicació *AppServ* (amb llicència LGPL), ja que aquesta instal·la i configura automàticament el servidor *Apache*, el SGBD *MySQL* i el *PHP*. L'aplicació ha estat creada utilitzant la versió v.2.5.9 de l'*AppServ*, encara funciona amb qualsevol altra versió.

- 2- **Crear una base de dades amb el nom *prestissimo* i importar l'arxiu SQL anomenat *prestissimo.sql*** mitjançant la interfície web del Sistema Gestor de la Base de Dades (SGBD) *MySQL*. En el cas del *PHPMyAdmin* (instal·lat automàticament en la instal·lació de l'*AppServ*) és realitza creant una base de dades nova desde la pantalla d'inici i posteriorment, des de la pestanya d'importar, seleccionar l'arxiu *prestissimo.sql* present a la carpeta BD del cd i executar-lo, tal i com es pot observar a la figura 12 següent. És necessària la prèvia instal·lació d'un sistema gestor de bases de dades *MySQL*.

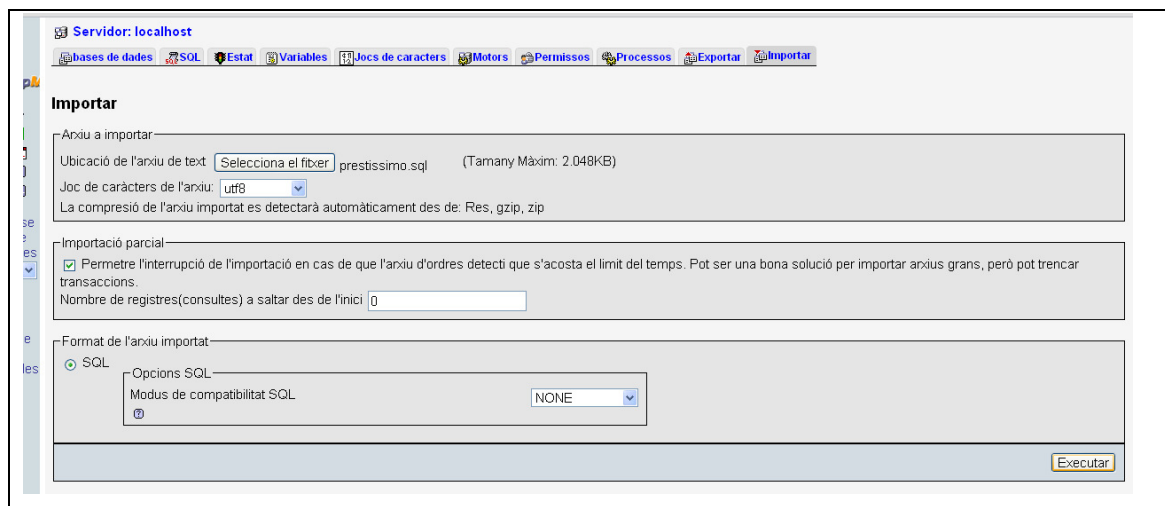


Figura 12. Importar una base de dades des d'un fitxer

- 3- **Editar l'arxiu *configuration.php*** situat al directori arrel de l'aplicació per a poder connectar-se a la base de dades creada anteriorment, utilitzant l'usuari i contrassenya del

SGBD. Els paràmetres a canviar estan correctament indicats al principi de l'arxiu, com es pot observar a la figura 13 següent.

```
1 <?php
2 class JConfig {
3
4   ### CONFIGURACIÓ DE LA CONNEIXIÓ A LA BASE DE DADES ###
5
6   var $dbtype = 'mysql'; // NO CANVIAR!!!
7   var $host = 'localhost'; // Nom del host del servidor, generalment serà localhost (en local).
8   var $user = 'root'; // VALOR A CANVIAR: nom d'usuari de la BD.
9   var $db = 'prestissimo'; // VALOR A CANVIAR: nom de la base de dades, si s'ha especificat un altre diferent a 'prestissimo'.
10  var $dbprefix = 'jos_'; // NO CANVIAR!!!
11  var $password = '80z5130'; // VALOR A CANVIAR: contrassenya de l'usuari de la BD.
12
13  ### FI CONFIGURACIÓ DE LA BD. NO CANVIAR RES!!! ###
14
15  var $max_execution_time = '50';
16  var $offline = '0';
17  var $editor = 'tinymce';
18  var $charset = 'utf-8';
```

Figura 13. Edició de l'arxiu configuration.php

4- Una vegada guardat l'arxiu ja **es pot obrir l'aplicació**. Si és a un servidor local la direcció URL seria <http://localhost/prestissimo/> i el resultat seria el mostrat a la figura 14 següent.



Figura 14. Pantalla d'inici de l'aplicació Prestíssimol!

5- Si es desitja utilitzar la funció de lectura de codis de barres mitjançant webcam s'hauria de fer el següent: **Instal·lar l'aplicació ZBar** (amb llicència LGPL) al sistema operatiu i, posteriorment, indicar-li a l'aplicació *prestissimo!*, en l'apartat de configuració

del component on s'utilitza la webcam, l'opció d'habilitar la lectura des de la webcam, tot indicant la direcció on s'ha instal·lat el programa ZBar.

En alguns casos, el servidor apache impedeix a l'aplicació Prestíssimo! executar comandaments des del shell. Si això ocorria, simplement hi hauria que activar l'opció "permetre als serveis d'aquesta aplicació que interactuen amb l'escriptori", a la configuració dels serveis del sistema. Per exemple, al sistema Windows es pot accedir clicant a inicia/executa i executar l'ordre "services.msc", com es mostra a la figura 15 següent.

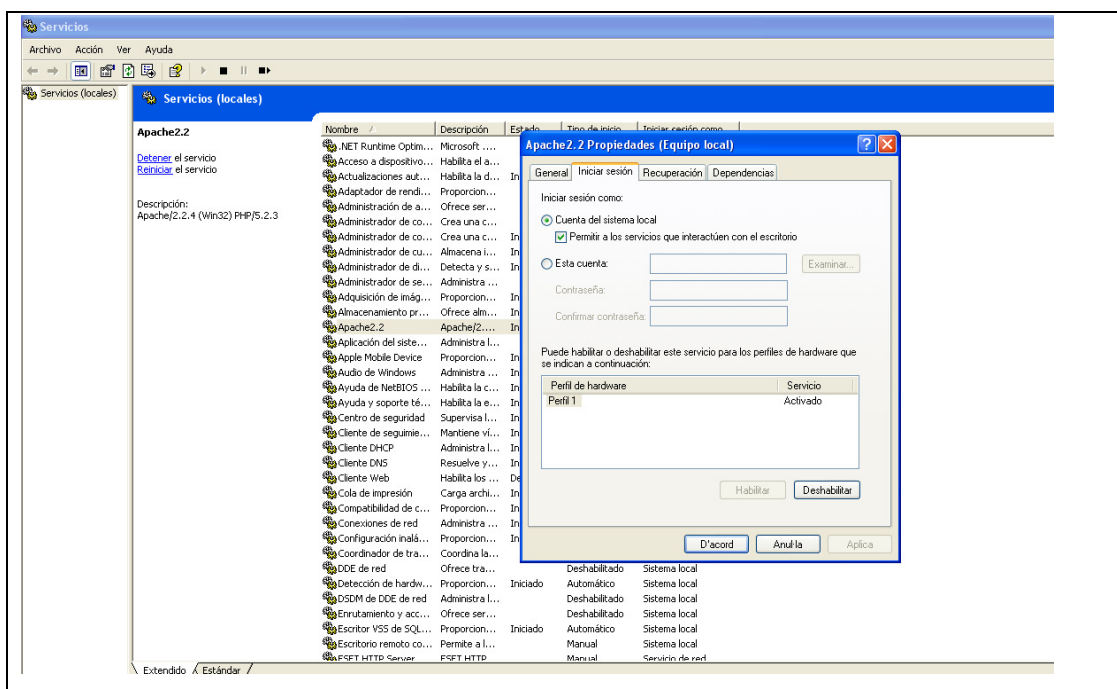


Figura 15. Configuració del servei Apache

- 6- Si es desitja utilitzar la funció d'**enviament d'avisos per email**, s'haurà de tindre en compte la ubicació del servidor. Si el servidor està situat a un hosting contractat a la xarxa, el proveïdor ja oferirà la funció de correu. No obstant, **si l'aplicació es troba a un servidor local, serà necessària la instal·lació d'un servidor de correu al sistema operatiu**. En aquest cas es recomana la instal·lació del programa *ArGoSoft Mail Server*, encara que l'aplicació funcionarà amb qualsevol servidor de correu.