

2011

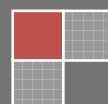
# GIP – Gestor d'Informació Parroquial

**Jaume Morey Riera**  
ETIG / ETIS

**Salvador Campo Mazarico**

20/06/2011

UOC  
01/01/2011





## CONTROL DOCUMENTACIÓ

### Històric de versions

VERSÍO	RESPONSABLE	DATA	DESCRIPCIÓ
1.0	Jaume Morey Riera	20/06/2011	Versió original

(Creative Commons)

Aquest treball està subjecte - excepte que s'indiqui el contrari- en una llicència de Reconeixement-NoComercial-SenseObraDerivada 2.5 Espanya de Creative Commons. Podeu copiar-lo, distribuir-lo i transmetre'ls públicament sempre que citeu l'autor i l'obra, no es faci un ús comercial i no es faci còpia derivada. La llicència completa es pot consultar en <http://creativecommons.org/licenses/by-nc-nd/2.5/es/deed.es>.



## ÍNDEX DE CONTINGUTS

<b>1. INTRODUCCIÓ</b>	<b>5</b>
1. RESUM DEL PROJECTE	5
2. JUSTIFICACIÓ DEL TFC I CONTEXT EN EL QUAL ES DESENVOLUPA: PUNT DE PARTIDA I APORTACIÓ DEL TFC.	5
3. OBJECTIUS.	6
4. ABAST DEL PROJECTE.	6
5. ENFOCAMENT I MÈTODE SEGUIT.	7
6. PLANIFICACIÓ DEL PROJECTE.	8
7. PRODUCTES OBTINGUTS.	10
<b>2. COMPOSICIÓ DEL PROGRAMARI</b>	<b>10</b>
<b>3. ESTRUCTURACIÓ DE PAQUETS</b>	<b>11</b>
<b>4. ESPECIFICACIONS DE LES FUNCIONALITATS PER MÒDUL</b>	<b>12</b>
<b>5. CASOS D'ÚS</b>	<b>14</b>
1. IDENTIFICACIÓ DELS ACTORS	14
2. CASOS D'ÚS PER MÒDULS	14
6. REQUISITS DEL MAQUINARI I PROGRAMARI	27
7. SEGURETAT EN EL SISTEMA.	28
8. FUNCIONALITATS DE LES PROPERES VERSIONS	30
<b>9. ARQUITECTURA</b>	<b>31</b>
<b>10. DIAGRAMA DE CLASSES ESTÀTIC</b>	<b>36</b>
<b>11. VALORACIÓ ECONÒMICA</b>	<b>37</b>



---

<b>12. CONCLUSIONS</b>	<b>38</b>
<b>13. BIBLIOGRAFIA CONSULTADA</b>	<b>39</b>
<b>ANNEX. MANUAL D'INSTAL·LACIÓ.</b>	<b>40</b>



# 1. Introducció

## 1. Resum del projecte

Les parròquies des de sempre, manegen molta quantitat d'informació, principalment sobre referent a les persones que han estat batejades, i a partir d'aquí pot ser que també hagin cursat catequesis de primera comunió, hagin rebut la primera comunió, confirmació, s'hagin casat, etcètera. Per cada una d'aquestes persones es conserva una fitxa amb tota la informació referent als actes esmentats, i aquesta informació pot ser requerida per les diferents parròquies en funció de si ho desitja cada persona.

La finalitat d'aquest projecte és la creació d'un nou programari per a la gestió d'aquestes fitxes personals, facilitant l'accés a aquestes dades per a qualsevol usuari i facilitant la seva actualització. Així mateix es pretén que aquesta informació deixi d'estar únicament conservada en papers per a passar a estar desada en una base de dades i tenint així l'opció de guardar còpies de seguretat.

El sistema a desenvolupar el farà amb tecnologia J2EE i l'accés al mateix es farà mitjançant un navegador d'Internet, evitant d'aquesta manera el problema derivats d'una instal·lació en entorn local.

## 2. Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC.

Actualment qualsevol administració, empresa o entitat, des de un ajuntaments fins a un petits comerç, conserva qualsevol informació mínimament apreciada en format electrònic.

Fins avui les parròquies sembla que encara no s'han llançat a informatitzar la informació que gestionen, ja sigui per falta de recursos o per manca de interès davant de la evolució informàtica, de forma que s'han quedat enrere.



Per aquests motius, aquest projecte pretén desenvolupar una aplicació informàtica per a informatitzar tota la documentació continguda a les fitxes personals de cada individu que es guarden a les parròquies.

Aquesta aplicació serà implantada inicialment a la Unitat Parroquial de Manacor, composta per les parròquies de Crist Rei, Sant Pau i Nostra Senyora dels Dolors. En primera instància només es crearan les fitxes personals dels nens que han cursat el primer curs de catequesis durant l'any escolar 2010/2012, per a continuació, al setembre poder afegir una nova funcionalitat, les inscripcions de nens a catequesis. Segons la satisfacció amb aquestes proves, es decidirà si implantar l'aplicació definitivament o si realitzar alguns canvis abans.

Després d'aquesta primera implantació, si el funcionament és satisfactori i eficient pels usuaris, es preveu que sigui implantat a altres municipis.

### **3. Objectius.**

L'objectiu d'aquest projecte és el desenvolupament d'una aplicació web per a gestionar les fitxes de informació personal que guarden les parròquies, de forma que es puguin trobar, consultar i afegir-hi informació de forma molt més àgil.

### **4. Abast del projecte.**

El projecte a realitzar ofereix les funcionalitats següents:

- Registrar persones, indicant nom, llinatges, adreça, telèfon, ...
- Esborrar les persones que així ho desitgin de la base de dades.
- Registrar els actes de cada persona (bateig, comunió, confirmació, casament, defunció) indicant la informació referent a cada un d'aquests.
- Crear horaris de catequesis, indicant el lloc, dia, hora i curs en els quals es podran inscriure els infants que s'inscriuran a catequesis.



- Crear nous usuaris per a que puguin accedir a la informació, modificar-los, desactivar-los o esborrar-los.

## **5. Enfocament i mètode seguit.**

La estratègia que s'ha plantejat per abordar aquest projecte consisteix en separar el projecte en diferents fases:

### Fase: Concepció del Sistema de Informació

L'objectiu d'aquest procés és l'obtenció d'una especificació detallada del SI, és a dir, una anàlisi funcional y la definició de les proves funcionals del sistema que serveixi de base pel posterior disseny o construcció del sistema.

### Fase: Prototipus del Sistema de Informació

L'objectiu del Prototipus és la elaboració d'una part del sistema (una part de anàlisi funcional, disseny d'aquesta part, construcció, proves i implantació o desplegament) per verificar la arquitectura que s'emprarà en la construcció del sistema.

Aquest prototipus permetrà conèixer el comportament del sistema construint una petita part i sobre aquesta part es construirà la resta del sistema.

També permet mostrar als usuaris potencials les característiques de l'aplicació i preveure mitjançant exemples quin serà el producte resultants, facilitant la tasca de validació dels usuaris i proporcionant un primer entorn de desenvolupament des d'on començar el desenvolupament.

### Fase: Elaboració del Sistema de Informació

L'objectiu de l'Elaboració del Sistema de Informació és el disseny tècnic del sistema i de l'entorn tecnològic que li donarà suport juntament amb l'especificació detallada dels diferents components del sistema d'informació.

A partir d'aquesta informació, es generen totes les especificacions de construcció relatives al propi sistema, la descripció tècnica del pla de proves, la definició dels requisits d'implantació i el disseny dels procediments de migració i càrrega inicial.

### Fase: Construcció del Sistema de Informació

L'objectiu d'aquesta activitat és desenvolupar el codi dels components del SI i assegurar el correcte funcionament del SI per a la seva posterior implantació.

Inclou les proves unitàries, de sistema i d'integració, de forma que es garanteixi que el producte resultant compleix amb la qualitat necessària per al seu lliurament.



#### Fase: Traducció

L'objectiu d'aquesta fase és traduir l'aplicació web, realitzada en primer lloc en català, al castellà, per tal de facilitar l'ús del sistema als potencials usuaris que es manegen millor en aquesta llengua.

#### Fase: Lliurament i Acceptació del Sistema de Informació

L'objectiu d'aquest procés és el lliurament i acceptació del sistema d'informació en la seva totalitat.

## **6. Planificació del projecte.**

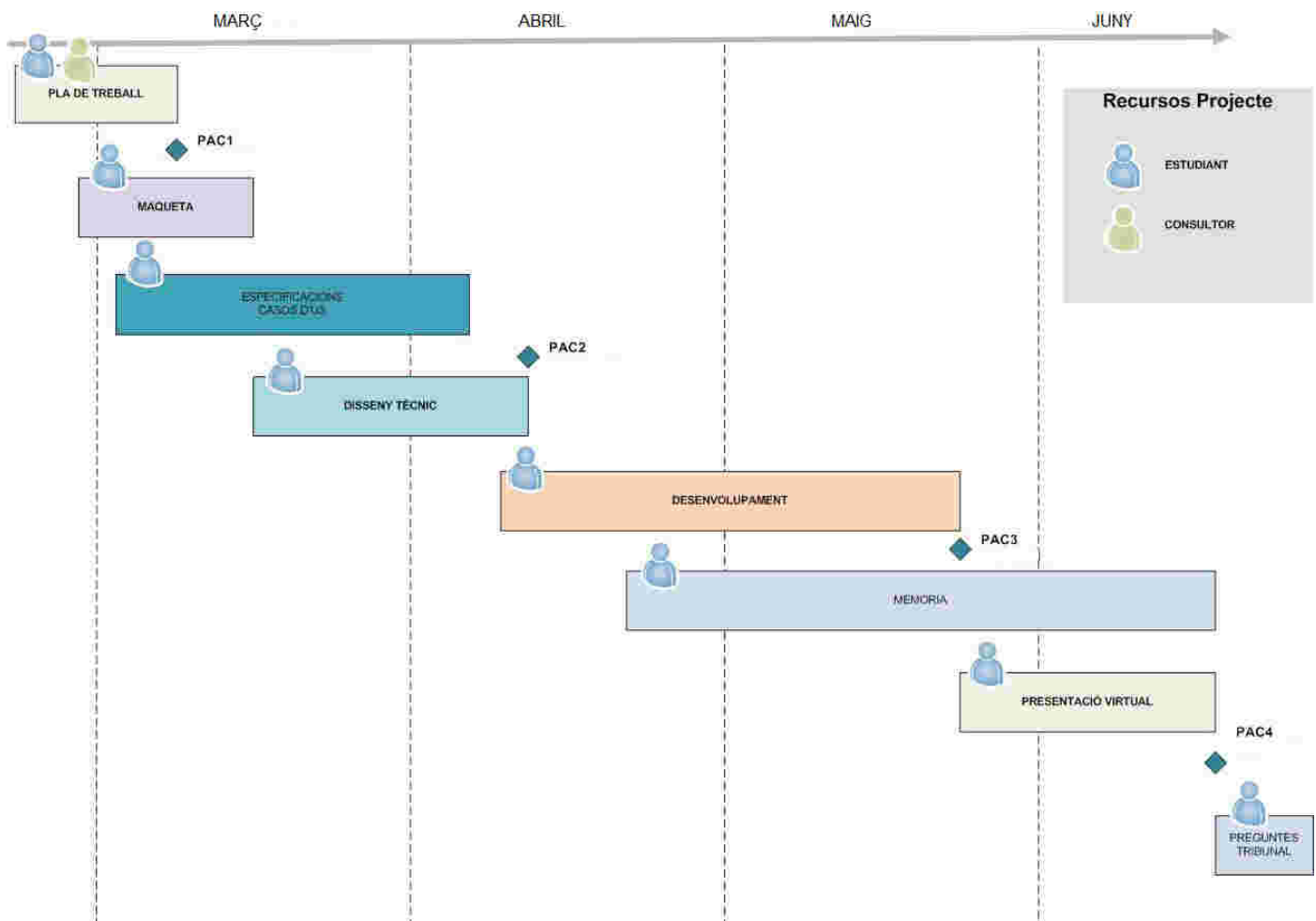
Per a la planificació, s'ha separat el projecte en diferents fases o activitats:

- **Pla de treball:** és primer document a lliurar. Inclou la descripció de treball fi de carrera, els objectius generals i específics del projecte i la planificació amb fites i temporització.
- **Maqueta:** disseny de la interfície d'usuari.
- **Especificacions i casos d'ús:** document amb diferents apartats amb la descripció del projecte, la divisió en mòduls o subsistemes, el diagrama de paquets, la descripció dels casos d'ús, l'especificació dels actors, els requeriments no funcionals de maquinari i programari i les especificacions de seguretat.
- **Disseny tècnic:** arquitectura del sistema, descripció dels frameworks i patrons de disseny utilitzats, diagrama de classes, diagrames d'estat i diagrames de seqüència agrupats per casos d'ús.
- **Desenvolupament:** codi font del programari desenvolupat, scripts de base de dades, fitxers de configuració, documentació amb instruccions d'instal·lació, etc.
- **Memòria:** inclourà, entre altres apartats, la justificació del TFC, els objectius i l'abast del mateix, la metodologia seguida, la planificació, els productes obtinguts, la valoració econòmica i les conclusions finals.



- **Presentació virtual:** document on es mostrarà de forma resumida i clara el treball realitzat i els resultats obtinguts, oferint una perspectiva general del TFC.
- **Preguntes tribunal:** una vegada presentada la memòria, el Tribunal d'avaluació del pot realitzar preguntes l'estudiant, per tant s'ha reservat a la planificació un temps per aquest tasca.

En el gràfic següent es mostren les diferents activitats que compondran el projecte, els perfils que intervindran a cada fase, així com les fites establertes i les dates.





## 7. Productes obtinguts.

El producte resultant és un programari per a la informatització i gestió de les fitxes personals que guarden les parròquies. El sistema està desenvolupat utilitzant tecnologia J2EE i l'accés al mateix es fa a través d'un navegador d'Internet.

# 2. Composició del programari

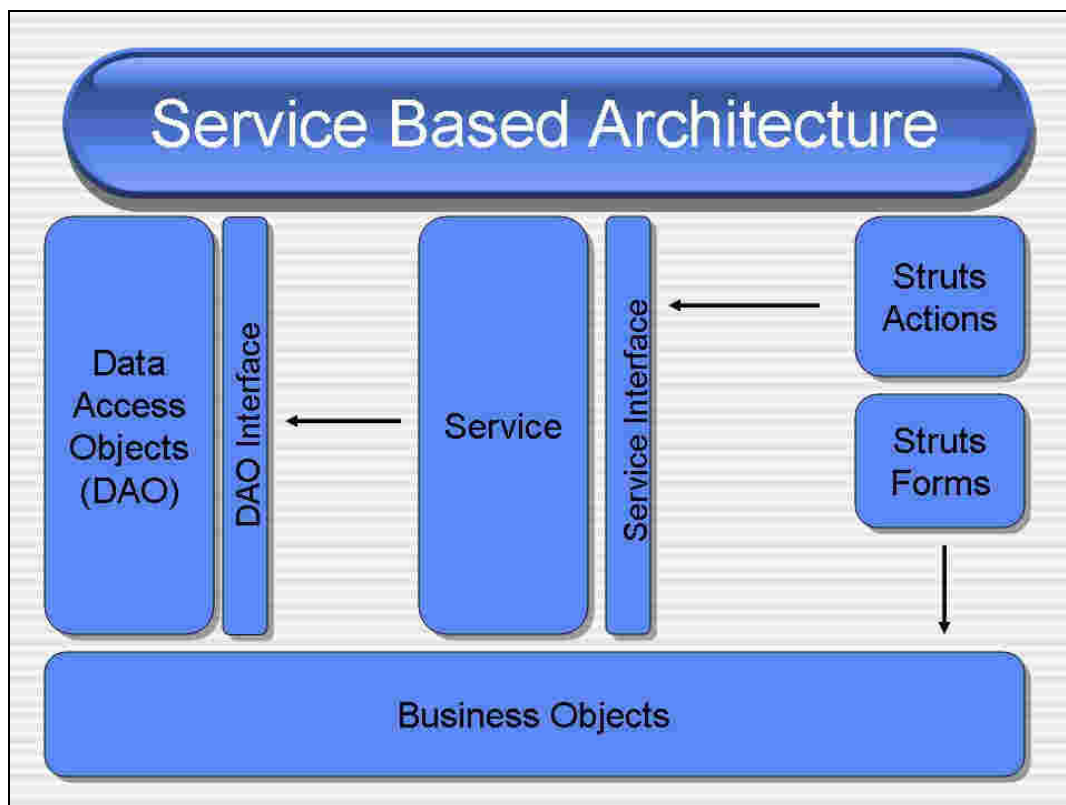
El sistema està compost per 5 mòduls segons el tipus de **funcionalitat**:

- **Mòdul de connexió i seguretat:** aquest mòdul és al que tenen accés els usuaris quan accedeixen al sistema. Realitza tasques l'autenticació, autorització i funcions de seguretat.
- **Mòdul d'administració:** Aquest mòdul inclou totes les funcionalitats que permeten gestionar l'alta, baixa i modificació d'usuaris. És competència dels usuaris administradors.
- **Mòdul de gestió d'informació:** Aquest mòdul inclou les funcionalitats que ofereixen la possibilitat de realitzar l'alta, baixa o modificació dels tercers, així com l'alta, baixa o modificació d'horaris de catequesis.

### 3. Estructuració de paquets

El sistema s'ha organitzat en paquets segons la funció de la classe java, per a facilitar la organització del projecte. Els paquets en que està estructurat el projecte són aquests:

- Model: Classes que defineixen usuaris, tercers, horaris, així com les taules i les respectives columnes que es crearan a la base de dades.
- Dao: Classes contenidores de les funcions d'accés a la base de dades.
- Service: Capa que ofereix les transaccions creades als DAO per a ser utilitzades per crear la vista.
- Controller: Classes encarregades de la generació de la vista que tindran els usuaris de taules, formularis, etcètera i que gestionarà les actuacions dels usuaris sobre els objectes.





## 4. Especificacions de les funcionalitats per mòdul

En aquest apartat es descriu de forma general les funcionalitats dels diferents mòduls que componen el sistema. A l'apartat següent (Casos d'ús) es descriuen les funcionalitats de forma més detallada i amb un enfocament més orientat a l'usuari.

### 1. Mòdul de connexió i de seguretat.

Aquest mòdul és l'encarregat de resoldre les operacions d'autenticació i autorització en el sistema. En aquest mòdul es gestionen els permisos per accedir a les diferents funcionalitats que componen el sistema per part dels usuaris. Dóna accés a les diverses funcionalitats que els hi correspon en funció del grup al que pertanyen i dels rols que tenen assignats en aquest grup. També s'encarrega de la gestió dels accessos (bloqueig d'usuaris, etc.).

Descripció de funcionalitats:

- i. Accés al sistema
- ii. Control de l'accés a les funcionalitats contingudes en cada mòdul en funció dels rols que té assignat l'usuari.

### 2. Mòdul de gestió de la informació.

Aquest mòdul és el que s'encarrega de la gestió de tota la informació. Gestiona els llistats de persones, els formularis d'alta o modificació de persones, així com de horaris de catequesis.

Les funcionalitats són:

- Llistat de tercers
- Alta de tercers



- Modificació de tercers
- Baixa de tercers
- Alta de horaris de catequesis
- Modificació d'horaris de catequesis
- Eliminació d'horaris de catequesis.

### **3. Mòdul d'administració.**

Des d'aquest mòdul es poden administrar diferents elements del programari: gestionar els usuaris, gestionar els grups d'usuaris, establir els paràmetres de configuració, etc. El perfil encarregat d'aquestes taques serà l'administrador.

El permisos del sistema estan organitzats en rols, de manera que un usuari tindrà permís per realitzar una determinada funcionalitat si té assignat el rol corresponent.

Descripció de funcionalitats:

- Consulta d'usuaris: permet la consulta dels usuaris registrats en el sistema i l'accés per a l'alta, baixa o modificació dels mateixos. Els perfils d'usuaris que accediran al sistema seran els gestors i els administradors.
- Alta d'usuari: L'administrador dóna d'alta un usuari del sistema.
- Baixa d'usuari. L'administrador dóna de baixa un usuari del sistema.
- Modificació d'un usuari: permet la consulta i/o modificació de les dades d'un usuari, el bloqueig i desbloqueig d'usuaris i l'assignació de rol.

# 5. Casos d'ús

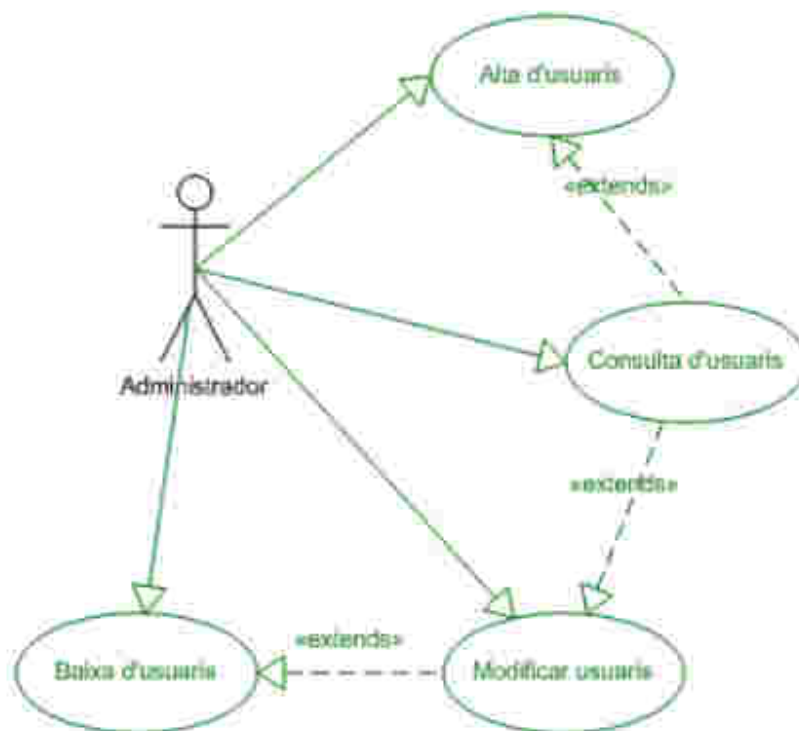
## 1. Identificació dels actors

Els actors que podran accedir al sistema i les seves característiques principals són els següents:

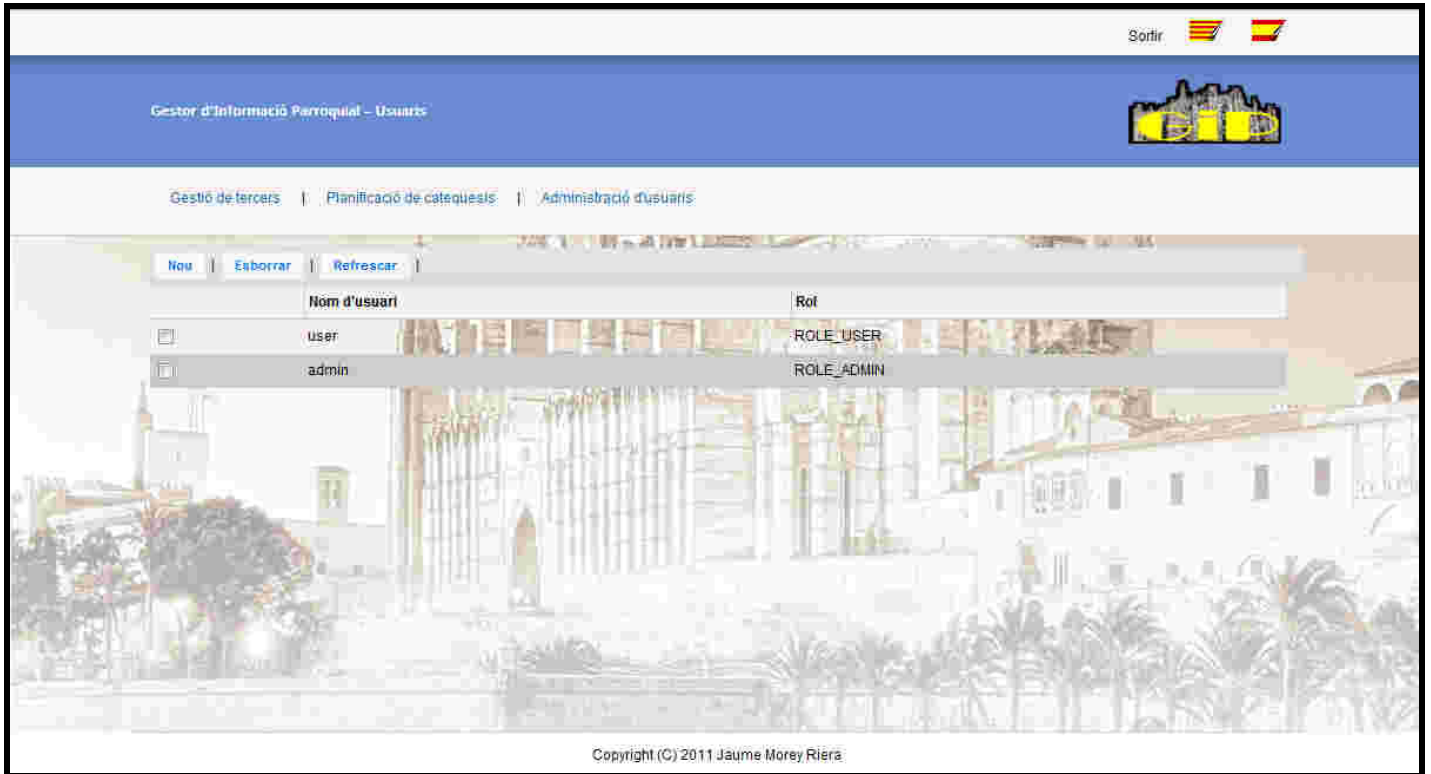
- Gestor: Encarregat de la gestió de la informació del sistema, tant de les fitxes personals, com dels horaris de catequesis.
- Administrador: A més de les funcions del gestor, s'encarregarà de la gestió dels usuaris.

## 2. Casos d'ús per mòduls

### 2.1. Mòdul d'administració



### 2.1.1. Cas d'ús "Consulta d'usuaris"



<b>Objectiu</b>	Consultar el llistat d'usuaris donats d'alta al sistema
<b>Actor</b>	Administrador
<b>Passos</b>	Accedir a la pantalla d'administració d'usuaris
<b>Postcondicions</b>	Es pot consultar una llista on apareixen els usuaris registrats al sistema i el seu rol

### 2.1.2. Cas d'ús "Alta d'usuari"

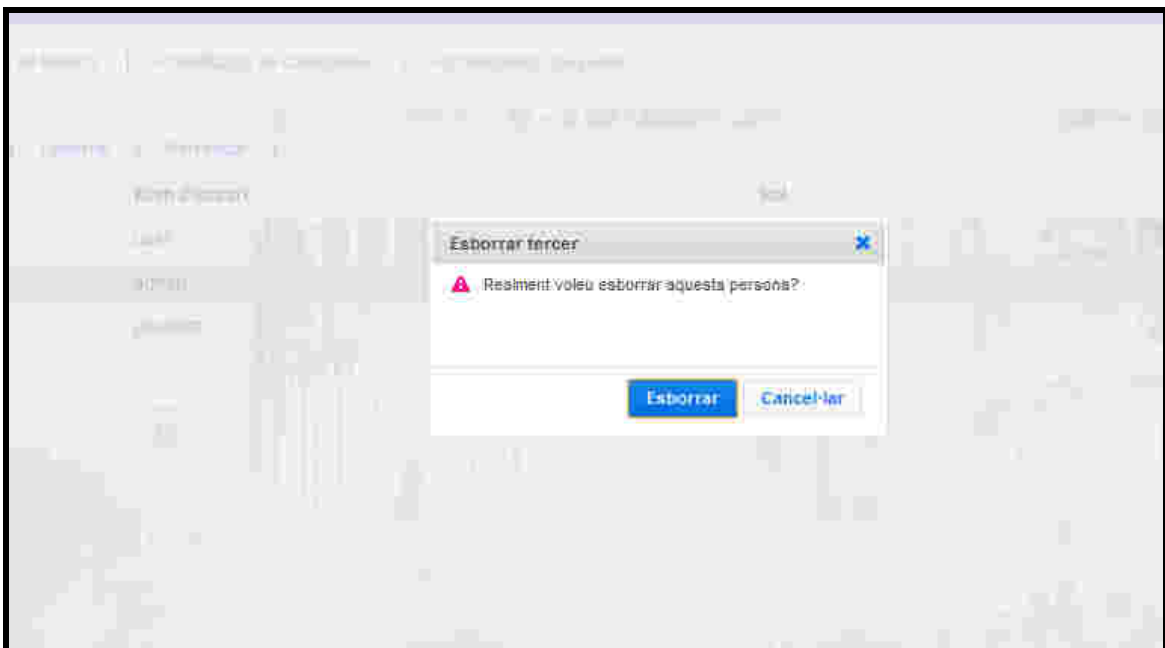
<b>Objectiu</b>	Donar d'alta un nou usuari
<b>Actor</b>	Administrador
<b>Passos</b>	Des de la pàgina d'administració d'usuaris fer click al botó "Nou" i inserir el nom d'usuari, contrassenya, rol i definir si està actiu el nou usuari
<b>Postcondicions</b>	L'usuari és donat d'alta al sistema, i si està actiu, pot accedir al sistema a aquelles pàgines a les que està autoritzat



### 2.1.3. Cas d'ús "Modificació d'usuaris"

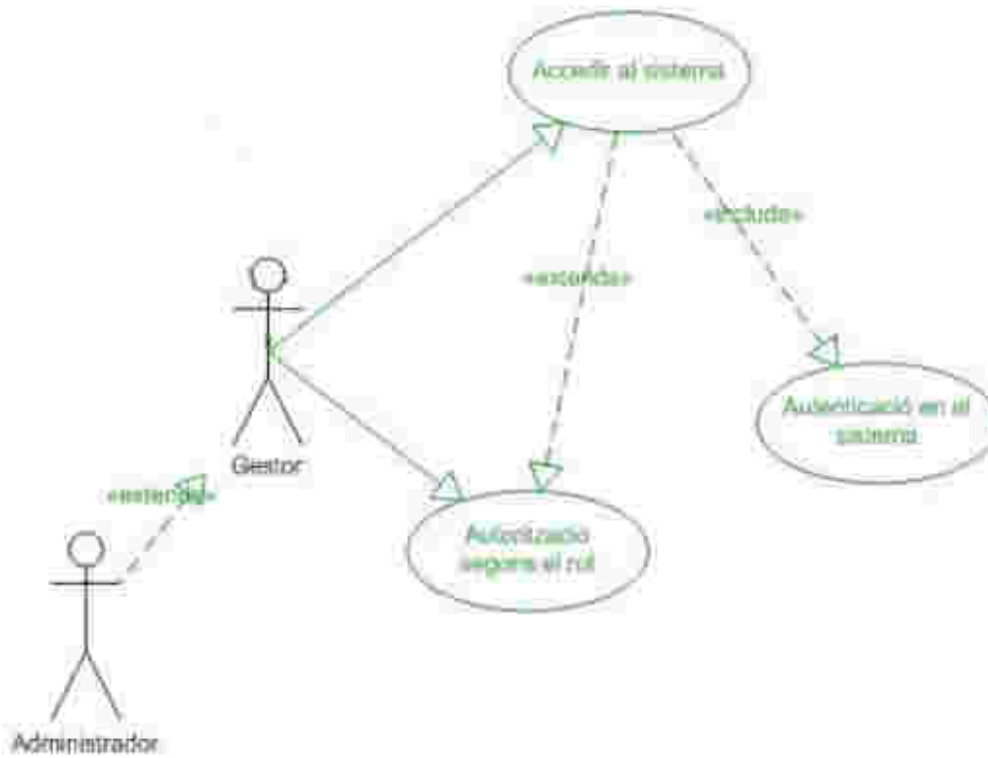
<b>Objectiu</b>	Modificar el nom d'usuari, contrasenya o rol d'un usuari o bé activar-lo o desactivar-lo
<b>Actor</b>	Administrador
<b>Passos</b>	Fer doble click sobre l'usuari que es desitja modificar, canviar les dades que es vulguin i acceptar
<b>Postcondicions</b>	Les dades de l'usuari s'han canviat

### 2.1.4. Cas d'ús "Baixa d'usuaris"



<b>Objectiu</b>	Eliminar del sistema un usuari
<b>Actor</b>	Administrador
<b>Passos</b>	Seleccionar l'usuari a eliminar i fer clic al botó Esborrar, confirmar que es vol esborrar
<b>Postcondicions</b>	L'usuari és eliminat definitivament del sistema

## 2.2. Mòdul de connexió



### 2.2.1. Cas d'ús "Accedir al sistema"

Gestor d'Informació Parroquial - Entrar

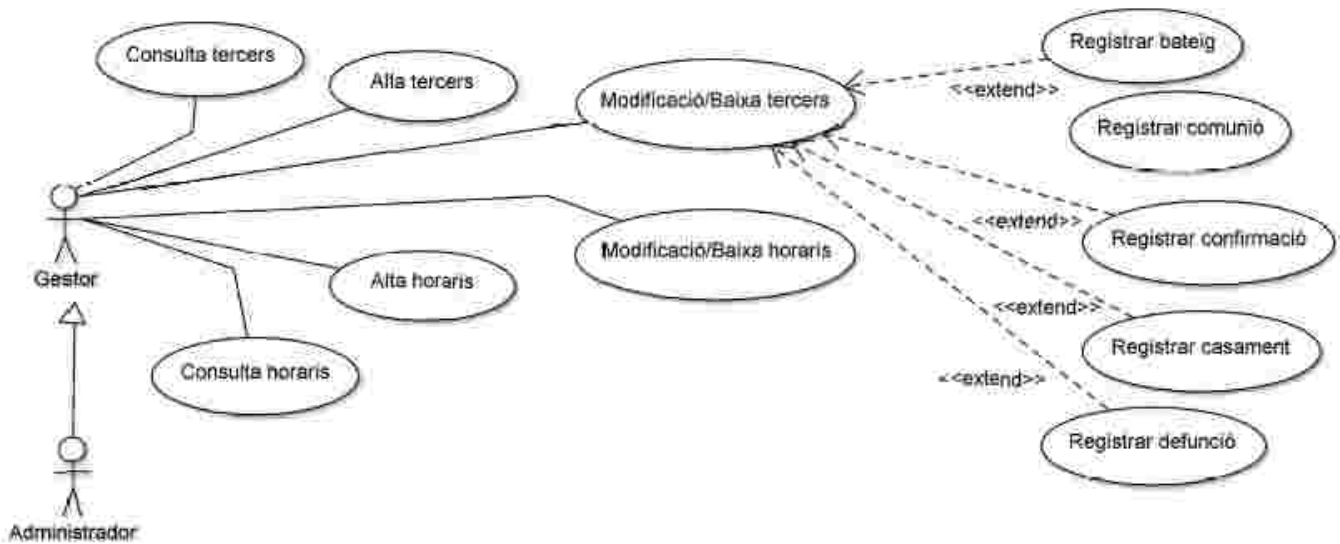
Nom d'usuari

Contrasenya

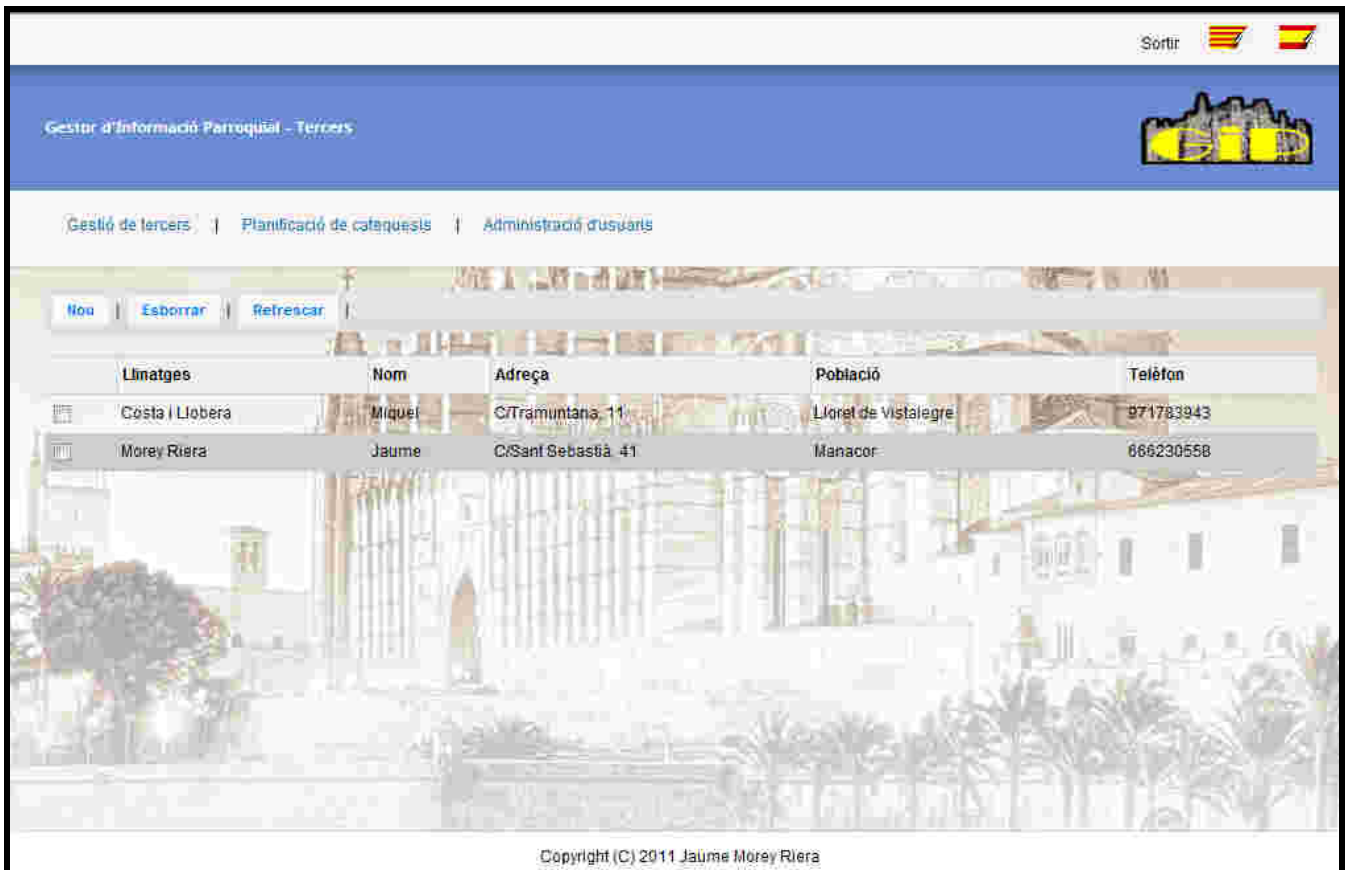
Recorda'm

<b>Objectiu</b>	Accedir al sistema
<b>Actor</b>	Gestor, Administrador
<b>Passos</b>	Introduir el nom d'usuari i la contrasenya i fe clic al botó "Entrar"
<b>Postcondicions</b>	L'usuari entra al sistema autenticat i amb els privilegis que li corresponen segons el seu rol.

### 2.3. *Mòdul de gestió de la informació*



### 2.3.1. Cas d'ús "Consulta de tercers"



<b>Objectiu</b>	Consultar el llistat de tercers donats d'alta al sistema
<b>Actor</b>	Gestor, Administrador
<b>Passos</b>	Simplement accedint al sistema la pantalla d'entrada serà el llistat de tercers
<b>Postcondicions</b>	L'usuari pot veure un llistat amb els tercers donats d'alta al sistema ordenats per ordre alfabètic segons els llinatges.

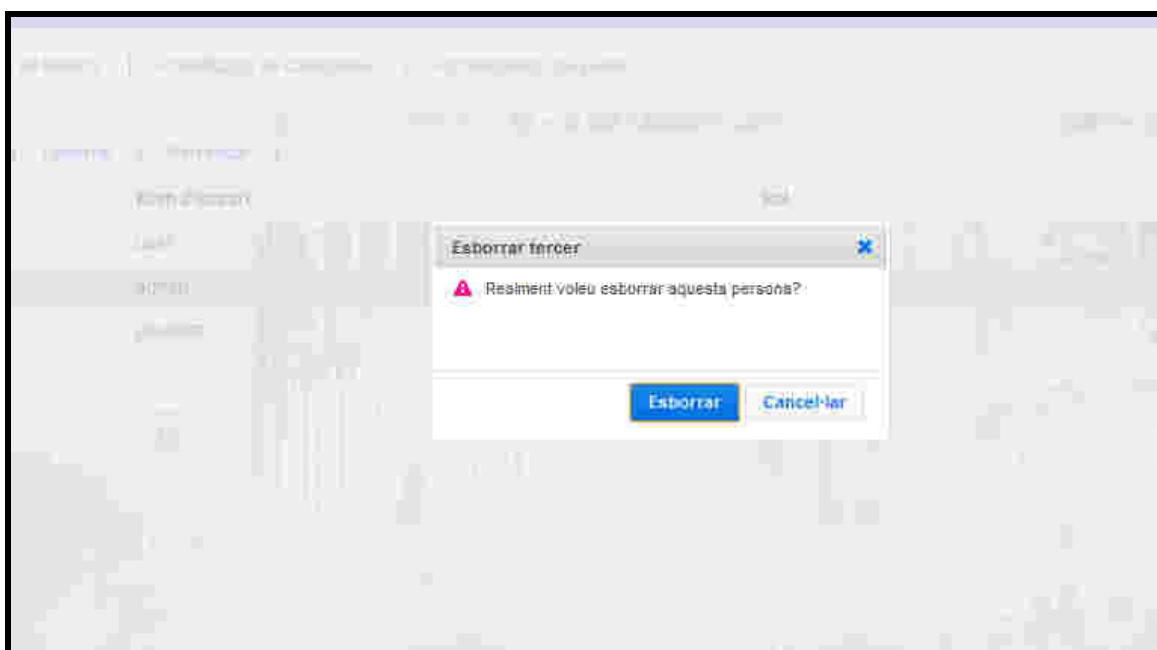
### 2.3.2. Cas d'ús "Alta de tercer"

<b>Objectiu</b>	Donar d'alta d'un tercer i introduir totes les dades referents a bateig, comunió, confirmació, casament
<b>Actor</b>	Gestor, Administrador
<b>Passos</b>	A la pantalla de gestió de tercers fer clic al botó "Nou" introduir totes les dades referents al tercer i fer clic a Guardar.
<b>Postcondicions</b>	El tercer és donat d'alta.

### 2.3.3. Cas d'ús "Modificació de tercer"

<b>Objectiu</b>	Modificar dades de la fitxa personal del tercer, registrar baptisme, comunió, confirmació, casament o defunció
<b>Actor</b>	Gestor, Administrador
<b>Passos</b>	Fer doble clic sobre l'usuari a modificar, fer els canvis i guardar
<b>Postcondicions</b>	El tercer és modificat segons els canvis introduïts.

### 2.3.4. Cas d'us "Baixa de tercer"



<b>Objectiu</b>	Esborrar un tercer
<b>Actor</b>	Gestor, Administrador
<b>Passos</b>	Seleccionar el tercer que es desitja esborrar, fer clic al botó "Esborrar" i a continuació acceptar.
<b>Postcondicions</b>	El tercer és esborrat definitivament del sistema

### 2.3.5. Cas d'ús "Consulta d'horaris"



<b>Objectiu</b>	Consultar el llistat d'horaris donats d'alta al sistema
<b>Actor</b>	Gestor, Administrador
<b>Passos</b>	Accedir a la pàgina de Planificació de catequesis
<b>Postcondicions</b>	L'usuari pot consultar el llistat d'horaris de catequesis ordenats alfabèticament segons el lloc.



### 2.3.6. Cas d'ús "Alta d'horari"

The screenshot shows a web form titled "Nou horari" (New class). The form is set against a light purple background. It contains the following fields:

- Lloc:** A text input field containing "Sant Pau".
- Dia:** A text input field containing "Dimarts".
- Hora:** A text input field containing "17:30".
- Curs:** A dropdown menu with "Segon" selected.

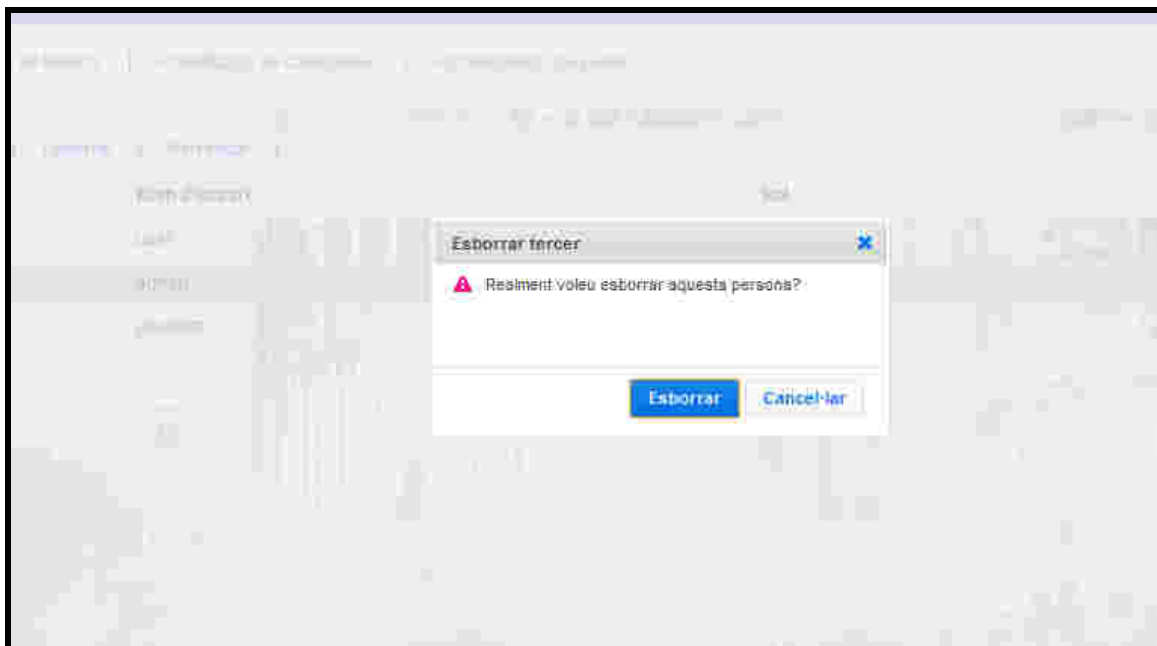
At the bottom right of the form, there are two buttons: "Guardar" (Save) and "Cancelar" (Cancel).

<b>Objectiu</b>	Donar d'alta un horari
<b>Actor</b>	Gestor, Administrador
<b>Passos</b>	A la pantalla de gestió de planificació de catequisis fer clic al botó "Nou" introduir totes les dades referents a l'horari i fer clic a Guardar.
<b>Postcondicions</b>	L'horari és donat d'alta.

### 2.3.7. Cas d'ús "Modificació d'horari"

<b>Objectiu</b>	Modificar dades de l'horari.
<b>Actor</b>	Gestor, Administrador
<b>Passos</b>	Fer doble clic sobre l'horari a modificar, fer els canvis i guardar
<b>Postcondicions</b>	L'horari és modificat segons els canvis introduïts.

### 2.3.8. Cas d'ús "Baixa d'horari"



<b>Objectiu</b>	Esborrar un horari
<b>Actor</b>	Gestor, Administrador
<b>Passos</b>	Seleccionar el horari que es desitja esborrar, fer clic al botó "Esborrar" i a continuació acceptar.
<b>Postcondicions</b>	L'horari és esborrat definitivament del sistema



## 6. Requisits del maquinari i programari

### 1. Requeriments del client.

Els usuaris accediran al sistema mitjançant un navegador web. El sistema serà compatible amb els navegadors utilitzats més habitualment en internet: Mozilla Firefox 3.x o 4, Internet Explorer 8.0 o Google Chrome.

### 2. Requeriments del servidor.

El sistema es construirà sobre una plataforma J2EE.

Com a servidor d'aplicacions J2EE utilitzarem TOMCAT, encara que el programari serà compatible amb altres, com per exemple JBoss, ja que el desenvolupament es farà respectant els estàndards.

A continuació es mostra de forma detallada el programari de base del projecte:

- Sistema operatiu del servidor: Linux, amb la distribució Ubuntu 11.04 o bé Windows XP, Vista o 7.
- Base de dades: MySQL
- Servidor HTTP: Apache Webserver
- Servidor d'aplicacions: Apache Tomcat versió 7.0.12
- Java J2EE 1.6 SDK
- IDE de desenvolupament Java: Eclipse (Helios)

## 7. Seguretat en el sistema.

És un subprojecte de Spring amb el qual podem gestionar l'autenticació en les aplicacions Java tant a la URL del projecte com a classes Java o accés a mètodes. La gestió d'usuari es pot realitzar a través de rols en bases de dades.

Aquest subprojecte utilitza filtres per a la seguretat de mètodes o classes o les peticions al servidor de l'aplicació i la configuració de la seguretat es troba dins del WAR. Alguns més dels avantatges que ofereix és que suporta diversos models d'autenticació com LDAP, HTTP Digest, etcètera i ho podem ampliar estenent les nostres classes del Spring Security.

Aquests filtres s'han de configurar en el web.xml.

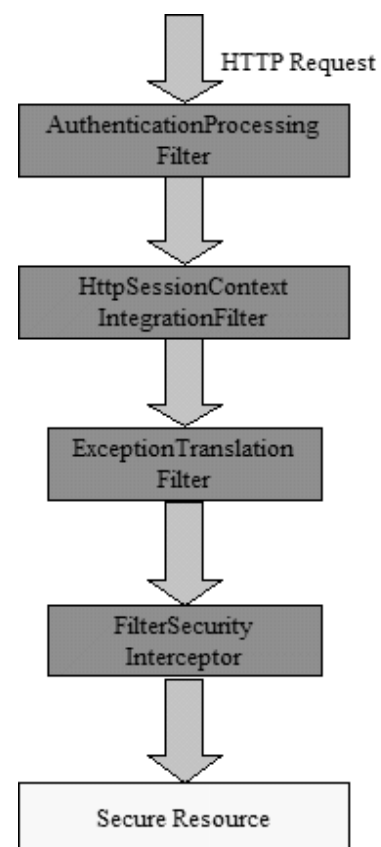
Amb Spring Security podem incloure seguretat a les nostres aplicacions fàcilment ja que tot just hem d'escriure codi per poder utilitzar-lo.

Spring Security, també anomenat Acegi, utilitza almenys una cadena d'almenys tres filtres per manejar la seguretat de l'aplicació:

1. El filtre `AuthenticationProcessingFilter` maneja la petició o requeriment (request) que revisa l'autenticació-`Authentication Request Check` ("el login de l'aplicació"). Per això fa servir el `AuthenticationManager`.

2. El filtre `HttpSessionContextIntegrationFilter` manté l'objecte `Authentication` entre diversos requests i l'hi passa al `AuthenticationManager` i al `AccessDecisionManager` quan sigui necessari;

3. El filtre `ExceptionTranslationFilter` verifica l'existència d'autenticació-`ExistingAuthentication Check`, maneja les excepcions de seguretat i executa





l'acció apropiada. Aquesta acció pot redirigir al diàleg d'autenticació (login form) o bé tornar el codid'error HTTP apropiat. ExceptonTranslationFilter depèn del filtre següent, FilterSecurityInterceptor, per funcionar.

4. El filtre FilterSecurityInterceptor controla l'accés restringit a recursos determinats - Restricted Acces Check -, i la revisió d'autorització. Coneix quins recursos són segurs i quins rols tenen accés. FilterSecurityInterceptor usa el AuthenticationManager i el AccessDecisionManager per funcionar.

En frameworks com Spring i seguint la moda actual de la injecció de dependència, les classes descrites anteriorment no funcionen soles sinó que serveixen a altres que actuen com proxys delegant el treball més difícil.



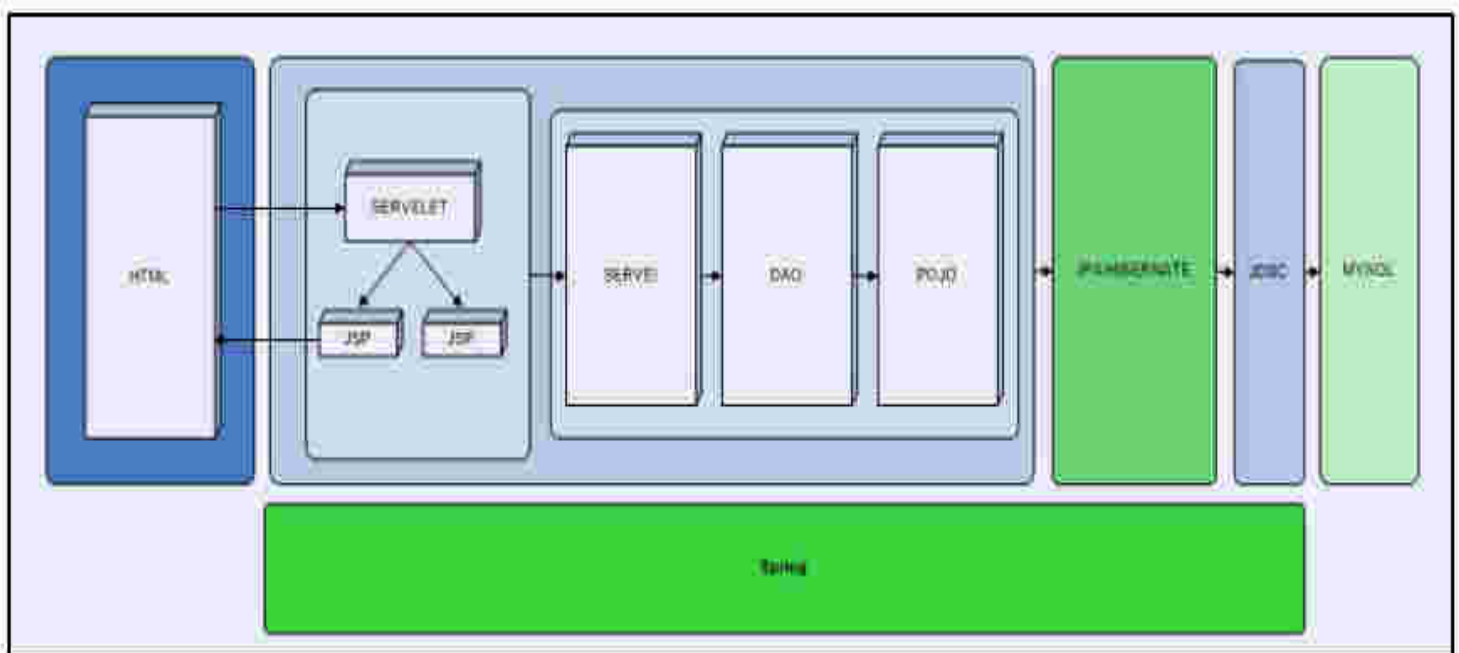
## 8. Funcionalitats de les properes versions

Les funcionalitats que està previst que inclogui la propera versió són:

- Inscripció d'infants als diferents horaris de catequesis.
- Generació automàtica de grups de catequesis.
- Assignació de catequistes als grups.
- Generar llistats de grups.
- Generar certificats de baptisme, realització de catequesis, comunió, confirmació o casament.

## 9. Arquitectura

L'arquitectura emprada per tal de implementar l'aplicació ha estat Spring + Hibernate + Apache Tomcat + MySQL a més s'ha aplicat el patró M-V-C.



L'arquitectura emprada dona molta flexibilitat al programador a l'hora de realitzar el manteniment donat que només s'ha de preocupar de quina capa desitja modificar i realitzar petites modificacions als fitxers de configuració si s'escau.

Un altra avantatge radica en la utilització de JPA per tal d'establir la comunicació amb el ORM, abans de l'aparició d'aquest API les migracions d'un ORM cap a un altre, per exemple de Hibernate cap a TopLink o a l'inrevés, resultaven molt costoses, amb la nova especificació són els ORM que s'han d'adaptar a l'aplicació i els canvis són molt més senzills.

A la figura es pot observar el funcionament del patró MVC. El model (M) està conformat per les capes POJO (Plain Old Java Object), DAO (Data Access Object) i Servei, el controlador (C) es correspon amb els servlets i finalment la vista (V) es correspon amb els JSP.

Quan un usuari desitja per exemple obtenir un llistat de persones, aquest envia una petició a través d'una pàgina HTML que és recollida per el servlet



aquest reenvia la petició cap a la capa de servei i la capa DAO i en aquest punt es produeix el mapeig entre la base de dades i DAO gracies a JPA i l'ORM, tot seguit s'envia una resposta , a través de les diferents capes que es recollida pel servlet el qual crida a la vegada als JSP necessaris per tal de genera la vista sol·licitada al començament del procés.

L'aplicació a desenvolupar es farà sobre una arquitectura basada en el framework Spring. Els frameworks, no només aporten un conjunt de peces de software reutilitzables, sinó que a més determinen les metodologies i paradigmes utilitzats en el desenvolupament de les aplicacions basades en ells. L'aplicació proposada seguirà lleialment les ideologies guiades pels frameworks i aplicarà les idees proposades als paradigmes MVC al desenvolupament de l'aplicació. A continuació es mostra un descripció dels diferents elements que ha de compondre l'aplicació, així com els principals serveis específics a utilitzar:

- **Capa de presentació:** fent ús de llibreries Javascript d'HTML dinàmic basades en AJAX i jQuery que permeten als usuaris una gestió molt àgil i eficient de la interfície d'usuari i Apache Tiles ens permet utilitzar sempre la mateixa capçalera, peu de pàgina, menú,... i només s'ha de recarregar el contingut. Així mateix s'implementa la traducció entre Castellà i Català.
- **Lògica de negoci:** fent ús de la tecnologia J2EE, especialment dels components i serveis proporcionats per Spring.
- **Persistència:** mitjançant l'ús de Hibernate, que permet l'abstracció d'una base de dades concreta, així com una configuració declarativa de la gestió de la transaccionalitat.
- **Autenticació i autorització:** utilitzant Spring Security com a tecnologia per resoldre l'autenticació i autorització en el sistema

### **Java 2 Enterprise Edition (J2EE).**

J2EE és una especificació oberta que proporciona un model complet per suportar arquitectures distribuïdes i representa un conjunt de tecnologies que es poden utilitzar conjuntament per implementar un model arquitectònic d'aplicacions distribuïdes de gran dimensions (orientat al món de les empreses). Bàsicament el model J2EE es basa en quatre nivells (arquitectura multicapa) per a una aplicació empresarial:





- La capa client proporciona la interfície d'usuari. Aquesta capa està formada per programes que demanen dades als usuaris i converteixen les seves respostes en peticions que s'envien a un component que processa la petició. Habitualment es comunica amb el nivell web mitjançant protocol HTTP, però en algunes aplicacions es pot interactuar directament amb el nivell Hibernate o amb el de la Base de dades.
- La capa web representada per un navegador o una aplicació client proporciona funcionalitat a l'usuari. El nivell web proporciona funcionalitat a partir un conjunt de continguts transportats al client a través de HTTP, rebent peticions i enviant respostes als clients que poden estar en qualsevol capa. Controla doncs la interacció entre l'usuari i la capa de negoci.
- La capa Hibernate o capa de la lògica de negoci ofereix un conjunt d'objectes relacionats amb aquesta i proporciona solucions a aquests problemes.
- La capa EIS (Enterprise Information System també anomenada capa d'integració) representa tots els recursos empresarials per a l'aplicació (bases de dades, sistemes empresarials de col·laboració, sistemes heretats, etc.)

Com podem observar cadascuna de les capes s'orienta a proporcionar una determinada funcionalitat a l'aplicació. És important distingir entre la ubicació física i la funcional. Dues o més capes poden existir en una mateixa JVM. L'aplicació J2EE que es dissenyi utilitzarà només aquelles capes que necessiti. Finalment indicar que J2EE es basa en components, contenidors i connectors. Els components són les unitats bàsiques de programació. Les tecnologies de components fonamentals són les JavaServer Pages (JSP) i *Servlets* que són les tecnologies a nivell de web. Els contenidors són programari que gestiona els components i que proporciona accés als serveis del sistema. La relació entre components i contenidor s'anomena freqüentment contracte.

### **1. Disseny capa client.**

El client utilitzat per accedir a l'aplicació serà una navegador d'internet. En aquest tipus de disseny la lògica de la presentació s'efectua a la capa web (o



capa de presentació). A més accedint amb un navegador web s'eviten els problemes derivats de la instal·lació d'un programari en l'entorn local.

## **2. Disseny capa web.**

La capa web és la encarregada de controlar la generació del contingut a l'usuari final. A més efectua la comunicació amb la lògica de negoci per obtenir les dades necessàries, control de flux de pantalles, perfils, autoritzacions, etc. La característica més important del disseny de la capa web que definim, és la utilització del patró MVC (Model View Controller). Com veurem més endavant, en projectes de determinada mida i estructura és recomanable la utilització d'un *framework* web que proporciona al desenvolupador una sèrie de serveis, facilitant la seva tasca. La major part dels *frameworks* implementen el patró MVC6,

## **3. Disseny capa de la lògica de negoci.**

Conté el conjunt de regles de negoci; entitats i regles que s'encarreguen de la implementació dels processos de negoci d'una empresa. Elegim la utilització de *POJO's* (Plain Old Java Objects), ja que la lògica de negoci és més senzilla d'implementar, aprofitem els avantatges que ens ofereixen la plataforma J2EE (transaccions, seguretat, etc.) per l'accés a dades.

### **Utilització de patrons.**

Fonamentalment els patró que utilitzarem és el MVC, que com hem vist és implementat per Spring.

### **Patró MVC (Model-View-Controller)**

Aquest patró (que té els seus fonaments en la tecnologia Smalltalk) defineix un element complex en termes de tres subunitats lògiques amb la finalitat de desacoblar el codi d'accés a les dades, el codi de la lògica de negoci i el codi de presentació.

- **Model.** L'estat de l'element; s'ocupa dels canvis d'estat. És el component que conté una o més classes i interfícies que són les responsables de mantenir les dades del model o aplicació. Mostra la funcionalitat de l'aplicació. Correspondrà a la lògica del negoci (capa de la lògica de negoci).

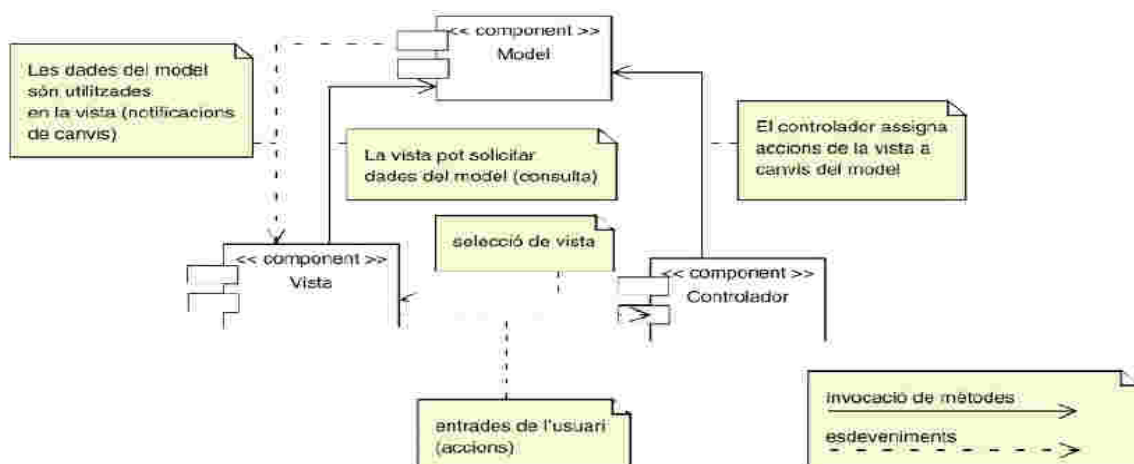
- Vista. La representació de l'element. Representen una representació de les dades en el component del model. El component vista pot recuperar informació del model però no pot modificar-la (aquestes peticions es dirigeixen al component controlador).
- Controlador. Aquest component gestiona els canvis en el model. Les peticions de canvi poden venir d'un component vista. Processa les entrades dels usuaris, inicia canvis en l'estat del model i en el cas de treballar en web, és el responsable del control de flux.

Els avantatges que s'obtenen de la utilització d'aquest patró són els següents:

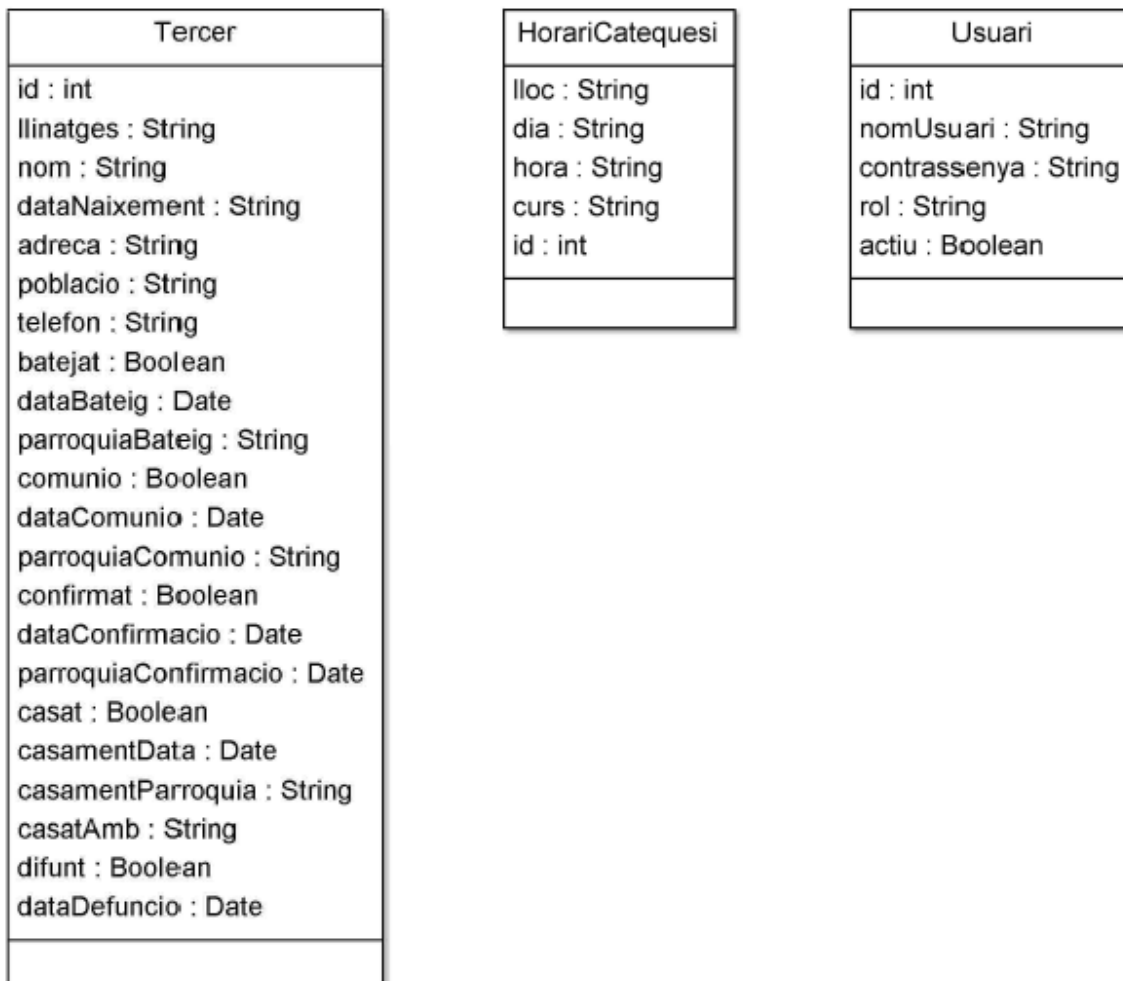
- Proporciona una manera excel·lent de fer que un element sigui flexible i adaptable a diverses situacions. Es pot modificar la presentació de la informació sense que això afecti, ni el model, ni la gestió de la interacció. També és possible canviar el model sense modificar el disseny.
- Permet un repartiment de tasques entre dissenyadors web i programadors J2EE. (separació de responsabilitats).
- Millora en la documentació.
- Independència de la lògica de negoci del protocol de comunicació emprat.
- Possibilitat de múltiples vistes.
- Possibilitat d'incorporar fàcilment tests de funcionalitats.

L'inconvenient és determinar una separació adient que permeti la reutilització en diversos projectes i no la faci excessivament dependent de la solució proposada.

El següent diagrama de components il·lustra el seu funcionament.



# 10. Diagrama de classes estàtic





# 11. Valoració econòmica

En el cas de que el projecte es realitzés en un entorn professional s'estima que tindria un preu de **6.074.41 Euros** (IVA inclòs).

El detall de les dedicacions i les tarifes aplicades es detallen a la taula següent:

PERFIL	Dedicació	€/hora	Import total	Import total + IVA
Analista funcional	20	43,10	862,00	1017,16
Analista programador	30	34,02	1020,60	1204,31
Dissenyador gràfic	20	27,21	544,20	642,16
Programador	100	27,21	2721,00	3210,78
<b>TOTAL</b>	170	31,56	5147,80	6074,41



---

## 12. Conclusions

Mitjançant aquest projecte he aprofundit en la tecnologia J2EE i he après a utilitzar moltes eines noves que no havia fet servir mai com son jQuery i AJAX, Spring Security.

A més, m'ha fet agilitzar molt en el maneig del servidor Tomcat i en l'ús de Hibernate, que només coneixia de forma superficial.

El resultat del projecte, tot i no ser definitiu, crec que està ben aconseguit, només que a causa de les pràctiques d'altres assignatures no he pogut avançar tant com hagués volgut.



## 13. Bibliografia consultada

[www.hibernate.org/docs](http://www.hibernate.org/docs)

[www.librosweb.es/ajax/](http://www.librosweb.es/ajax/)

[files.apache.org/tutorial/index.html](http://files.apache.org/tutorial/index.html)

[www.roseindia.net/spring/](http://www.roseindia.net/spring/)



# Annex. Manual d'instal·lació.

## Requeriments de software

L'aplicació es pot instal·lar sobre qualsevol servidor d'aplicacions que segueixi les especificacions J2EE. En aquest manual es mostra com realitzar la instal·lació partint d'aquest software:

- Tomcat 7.0 server

Es pot descarregar des d'aquesta adreça:

<http://tomcat.apache.org/download-70.cgi>

- MySQL server (XAMPP)

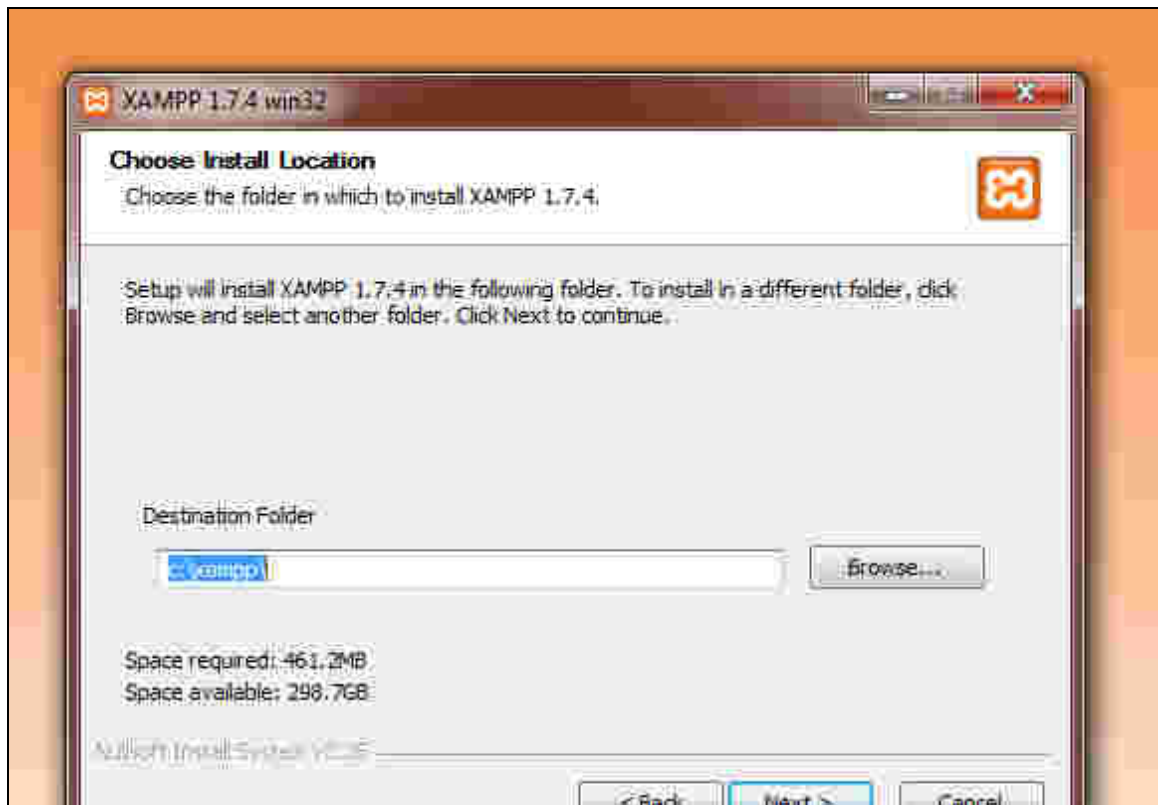
He optat per XAMPP, que inclou el servidor de base de dades de MySQL i a més ofereix un mòdul d'administració per aquest, des del qual podem gestionar intuïtivament la base de dades. Es pot descarregar a:

<http://www.apachefriends.org/en/xampp-windows.html>

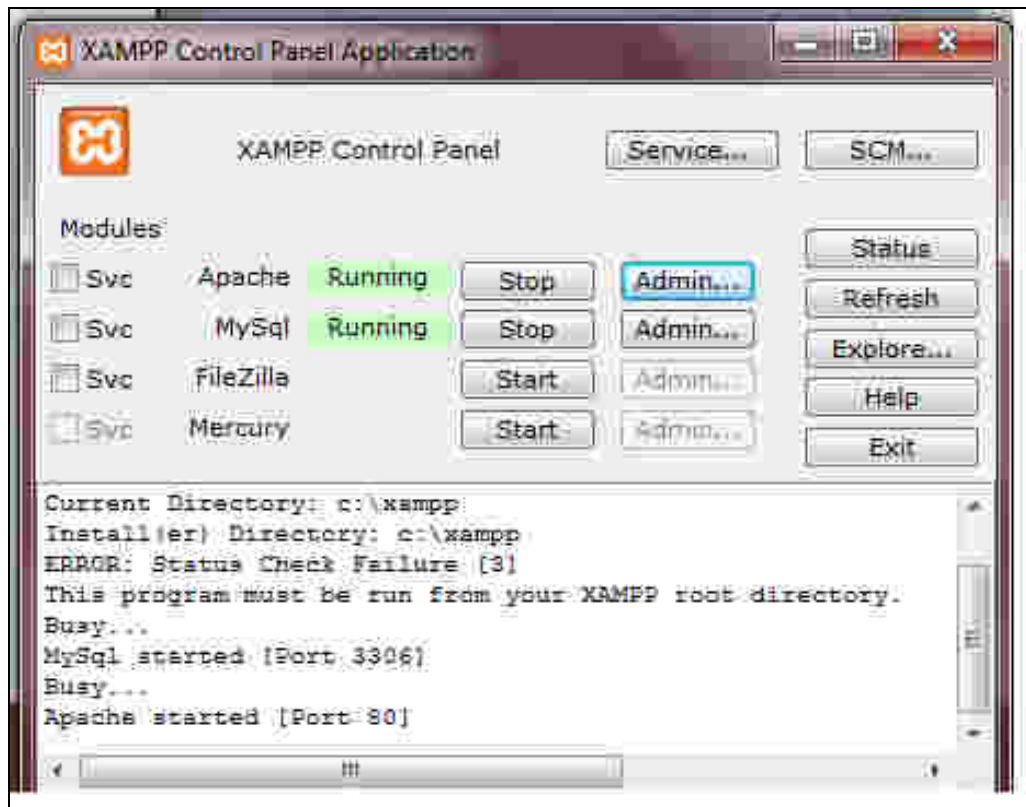
A més, XAMPP també inclou el servidor Tomcat, de forma que instal·lant simplement aquesta aplicació tindrem el servidor de base de dades i el servidor web.



1. Instal·lem XAMPP, en demanar el directori d'instal·lació, continuem amb el que indica per defecte l'instal·lador: C:/xampp



2. Un cop instal·lat, accedim al Panell de Control de XAMPP i arranquem Apache i MySQL. Tot seguit entrem al lloc d'administració.



- Una vegada a la pàgina d'administració de XAMPP, ens dirigim a PHPMyAdmin.

localhost/xampp/

# XAMPP for Windows

**XAMPP 1.7.4**  
(PHP: 5.3.0)

**Bienvenido a XAMPP para Windows!**

**Felicitades:**  
**XAMPP se instaló con éxito en su ordenador!**

Ahora se puede empezar a trabajar. :) Primero por favor pulse encima de «Estado» en la columna de la izquierda y las funciones estarán desactivadas. Es intencionado. Son funciones, que no funcionan en un sistema de producción.

Atención: XAMPP fue modificado a partir de la versión 1.4.x a una administración de paquetes de addons.

- XAMPP paquete básico
- XAMPP Perl addon
- XAMPP Tomcat addon
- XAMPP Cocoon addon
- XAMPP Python addon (developer version)

Y en un futuro:

- XAMPP Utility addon (Accesorio pero aún inactivo)
- XAMPP Server addon (otros servidores aún inactivos)
- XAMPP Other addon (otras cosas útiles aún inactivas)

Por favor "instalar" los paquetes adicionales, que aún necesiteis, simplemente a continuación de la instalación de XAMPP. A bueno, las versiones Instalador de los Addons individuales.

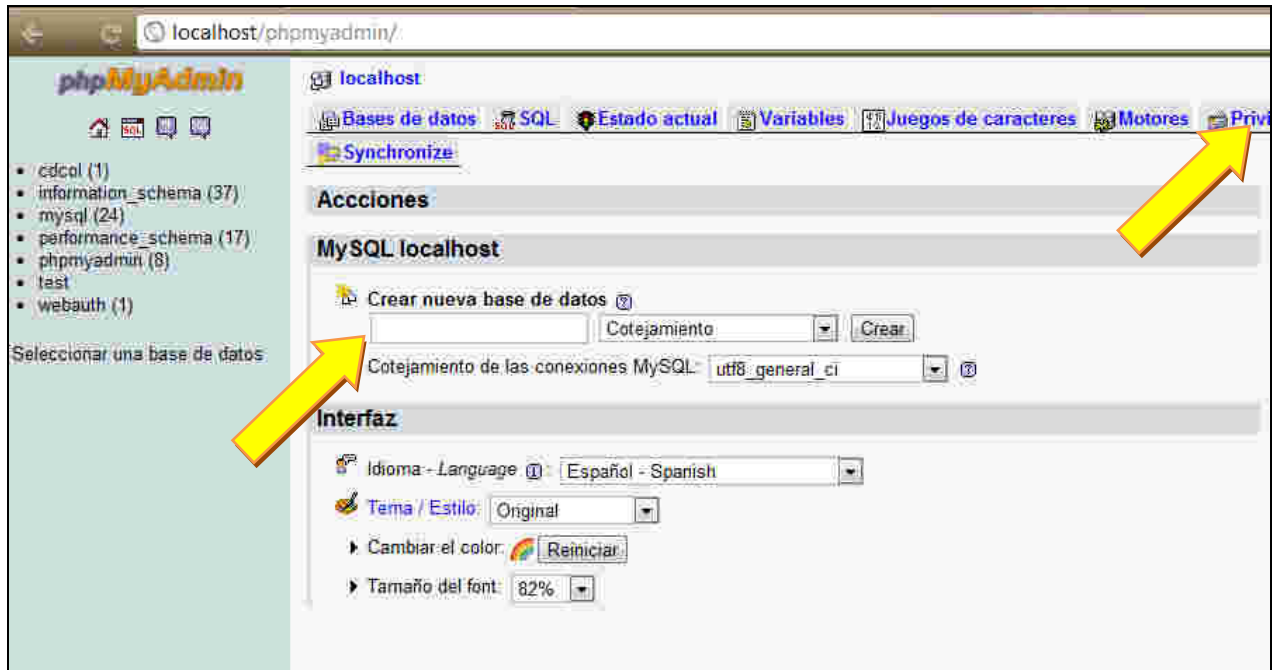
Para el soporte OpenSSL utilice por favor el certificado de chequeo con la URL <https://www.openssl.org/>

Os deseamos mucha diversión, Kay Vogelgesang + Kai 'Oswald' Seidler

**Herramientas**  
phpMyAdmin  
Webalizer  
Mercury Mail  
FileZilla FTP

©2002-2010  
...APACHE...  
FRIENDS...

4. Aquí crearem una nova base de dades, amb nom "gip". A continuació ens dirigim als privilegis del servidor de la base de dades.



5. Una vegada aquí, editarem l'usuari root del servidor localhost, posant-li la contrasenya "qwe.bnm".





6. Ara afegirem l'arxiu WAR corresponent al projecte a dins la carpeta des d'on l'ha de desplegar el servidor Tomcat: "C:/xampp/tomcat/webapps"
7. A continuació, ens disposarem a arrancar el servidor Tomcat. Per fer-ho, s'ha d'anar al directori C:/xampp/tomcat (ja sigui en l'explorador o en línia de comandes) i executar el fitxer catalina\_start.bat. Això desplegarà el projecte i crearà les taules a la base de dades.
8. Per acabar amb la instal·lació, anirem a l'editor SQL de la base de dades i executarem el següent script per a inserir dos nous usuaris a la aplicació web: un administrador (admin/12345) i un gestor (user/12345).

```
INSERT INTO `benutzer` (`id`, `username`,  
`password`, `authority`, `enabled`) VALUES  
(1, 'user', '12345', 'ROLE_USER', 1),  
(2, 'admin', '12345', 'ROLE_ADMIN', 1);
```

