

Index.....	a
1 Introducció	1
1.1 Context	1
1.2 Objectius del projecte.....	2
1.3 Estructura de la memòria	3
2 Gestió del projecte	5
2.1 Planificació.....	5
2.2 Costos del projecte	11
3 Context tecnològic.....	13
3.1 Desenvolupament d'aplicacions web sobre Java.....	13
3.1.1 Model 1	13
3.1.2 Model 2	14
3.2 Spring Framework.....	14
3.3 Hibernate	15
3.4 Servidors emprats.....	15
3.5 Eines de desenvolupament.....	16
3.5.1 Eclipse.....	17

3.5.2	Maven2.....	18
3.5.3	Subversion	19
3.6	Conceptes bàsics d' Spring Framework	20
3.6.1	Configuració d' Spring	20
3.6.2	Injecció de dependències	20
3.6.3	El dispatcher servlet	21
4	Arquitectura, disseny i implementació.....	23
4.1	Estructura de la pàgina	23
4.1.1	Pàgines d'administració.....	24
4.2	Configuració d' Spring.....	24
4.2.1	Capa de presentació	24
4.2.2	Capa d'accés a dades.....	27
4.3	Model del domini.....	30
4.4	Casos d'ús principals	33
4.4.1	Cas d'ús Mostra Calendari	35
4.4.2	Cas d'ús Mostra Entrega.....	43
4.4.3	Casos d'ús semblants.....	47
4.4.4	Cas d'ús Mostra Fitxer	52
4.5	Acegi Security.....	63

4.5.1	Fitxer web.xml	63
4.5.2	Fitxer de configuració	64
4.6	Casos d'ús d'administració	69
4.6.1	Cas d'ús Afegir	70
4.6.2	Cas d'ús Esborrar	72
4.6.3	Cas d'ús Editar	75
4.7	Cerca	78
4.7.1	Configuració	79
4.8	Proves	84
5	Conclusió	87
5.1	Objectius	87
5.2	Conclusions personals.....	88
5.3	Millores futures.....	88
Apèndix A	Instal·lació i configuració	91
Apèndix B	Manuais	95
B.1	Manual d'usuari	95
B.2	Manual de l'administrador	102
B.3	Crear un nou administrador.....	106
Apèndix C	Bibliografia.....	109

1 Introducció

Molts de pics, a l'hora de fer una assignatura de projecte, ens podem trobar molt perduts i no saber on anem ni el que hem de fer. L'assignatura de PXC¹ no n'és una excepció. Aquesta assignatura consta de dues parts ben diferenciades: la consecució d'unes practiques de laboratori guiades i la realització d'un projecte en grup. En la primera part no es sol tenir cap problema, ja que cada pràctica té uns guions molt detallats. És la segona part la que suposa un repte més gran per l'estudiant; sobretot si no han cursat anteriorment cap altra assignatura de projecte. La web resultant d'aquest projecte intentarà posar remei (o, almenys, ajudar una mica) a aquest problema.

La major part d'aquest projecte s'ha desenvolupat entre els mesos de Febrer i Maig del 2008. Tot i que prèviament hi ha hagut una feina important de documentació sobre les tecnologies a emprar.

1.1 Context

Quan veiem la plana web de l'assignatura de PXC² rebem molta informació sobre l'assignatura. Aquesta informació és sobretot de caràcter organitzatiu: com es desenvoluparà l'assignatura, quins són els professors,... Però, en canvi, no rebem cap tipus d'orientació sobre com s'hauria de dur a terme el projecte, que és la part més important de l'assignatura. A més, molta d'aquesta informació es perd per no estar estructurada i estar repartida en diferents pàgines sense connexió entre

¹ PXC: Projecte de Xarxes de Computadors

² <http://studies.ac.upc.edu/FIB/PXC/>

elles. Tot això pot provocar una sensació de frustració en l'estudiant, sobretot a l'hora de cercar informació sobre el projecte i no trobar-la.

Això és el que aquest PFC intenta posar remei. Intentarem posar a l'abast de l'estudiant de PXC tota la informació necessària per concloure el projecte amb èxit. A més, ho farem de tal manera que la informació estigui organitzada i sigui fàcil de trobar.

1.2 Objectius del projecte

L'objectiu principal del projecte és crear una web on puguem accedir a la informació necessària per orientar-nos a l'hora de fer el projecte de l'assignatura de PXC. A més, la web haurà d'acomplir els següents requisits funcionals:

- La informació haurà de ser accessible i fàcil de trobar. Cercarem una manera d'organitzar-la que en faciliti la cerca.
- Trobarem referències a projectes d'anys anteriors, així l'estudiant podrà agafar idees i veure què és el que ha funcionat bé i que és el que no ha funcionat tan bé.
- Hi haurà alguna manera de cercar la informació: en principi serà un formulari tipus "Google" cercarem el que ens faci falta.

També haurà d'acomplir una sèrie de requisits no funcionals. Aquests requisits defineixen aspectes de l'aplicació externs al funcionament d'aquesta. Com a requisits no funcionals tindrem els següents:

- La web haurà d'acomplir els estàndards del W3C³. Això afavorirà que la pàgina es pugui fer servir des de qualsevol plataforma i navegador.
- La web haurà de ser independent de la BD. Això ho aconseguirem emprant una capa d'abstracció de dades. En el cas de la nostra aplicació aquesta capa és Hibernate: un ORM⁴ del que en parlarem més endavant.
- Serà ampliable. En podrem ampliar la funcionalitat fàcilment. Per aconseguir això s'han emprat patrons de disseny ben coneguts, aprofitant les característiques d'Spring MVC.

Tots aquests punts seran explicats i expandits en capítols posteriors d'aquest document. A més, es veurà per que són importants tenir alguns d'aquests punts clars des de el principi.

1.3 Estructura de la memòria

Aquesta memòria està estructurada en diferents capítols per facilitar-ne la lectura. Els capítols dels que consta son els següents:

- 1. Introducció:** El primer capítol és la introducció. Ens posem en context sobre el contingut del projecte i de la memòria.
- 2. Gestió del projecte:** Explicarem quina ha estat la planificació del projecte i com ha variat aquesta planificació.

³ W3C: World Wide Web Consortium: És l'organisme de regular els estàndards d'Internet. <http://www.w3.org>.

⁴ Object Relational Mapping: Mapejador Objecte Relacional. Mapeja directament objectes del domini a la base de dades.

- 3. Context tecnològic:** En el tercer capítol explicaré perquè he triat la plataforma Java i Spring per fer el projecte. A més, farem una petita explicació de les altres eines que he emprat per realitzar el projecte.
- 4. Conceptes:** En aquest capítol explicaré els conceptes bàsics d' Spring Framework que facilitaran la comprensió dels capítols següents.
- 5. Disseny i implementació:** En aquest capítol farem una explicació més detallada del disseny emprat. També revisarem la implementació.
- 6. Conclusions:** Explicaré quines conclusions en trec d'aquest projecte i el resultat obtingut. A més, faré un petit incís sobre les millores que es poden dur a terme en un futur.

Als apèndixs veurem alguns temes annexos a la memòria:

- A. Instal·lació i manteniment:** Un manual d'instal·lació. Explicaré com es pot compilar el projecte i com s'ha d'instal·lar.
- B. Manual d'usuari:** Un petit manual d'utilització. Sobretot orientat als administradors del lloc.
- C. Bibliografia**

Seguint aquest guió començarem a entrar en matèria.

2 Gestió del projecte

En aquest capítol explicarem la planificació del projecte i com ha anat evolucionant a mesura que hem avançat. També farem una aproximació teòrica al cost que podria tenir. Tot i que aquest cost és molt orientatiu pot servir per estimar que podria costar un projecte d'aquestes característiques. A més, també hem de tenir en compte que a l'hora d'abordar un projecte, el punt de partida pot ser molt diferent depenent de molts factors.

2.1 Planificació

La planificació inicial que vaig proposar al director de projecte és la següent:

Data aproximada	Tasques a tenir fetes
Finals de Febrer	Enviar informe PFC als membres del tribunal, muntar la versió actual en algun servidor (prototipus 0.5)
Finals de Març	Versió bastant acabada en línies generals. Faltarà acabar alguns detalls.
Finals d'Abril	Versió final acabada. Poden restar alguns detalls puntuals.
15 d'Abril al 15 de Maig	Escriure Memòria

Taula 1 Planificació inicial

Aquesta planificació ha anat evolucionant a mesura que he avançat en el projecte. Veiem com ha quedat:

S'ha de tenir en compte que algunes tasques que aquí es mostren com a independents en realitat s'han solapat. Així i tot, es possible fer-se una idea general del desenvolupament del projecte des de aquesta planificació.

2.2 Costos del projecte

Els costos del projecte es poden estimar de moltes maneres. Aquí en presentaré una. Aquest mètode és una aproximació, ja que no se n'ha fet un seguiment exhaustiu a mesura que ha anat avançant el projecte i se n'ha fet l'estimació una vegada està acabat (o molt proper d'estar-hi). El que farem serà emprar Microsoft Project per aproximar el nombre d'hores que ha dut el projecte en total (emprant la planificació anterior) i multiplicar-ho per un cost per hora estàndard (he emprat 6€/hora, tot i que si el projecte s'encarregués a una consultoria externa possiblement es dispararia aquest cost).

Segons el Microsoft Project, el projecte ha dut 687 hores. Aquest valor d'hores pot ser un poc diferent a la realitat, ja que no s'ha fet feina tots els dies o no sempre s'ha fet feina les hores que s'han introduït al projecte (alguns dies ha estat més, alguns dies menys). Així i tot ho emprarem per els nostres càlculs. Emprant la simple fórmula $Cost = Hores \times CostHora$ ens surt un cost de 4122€. En cas d'estar en un grup de projecte on cada desenvolupador o component del grup tingués un preu diferent, el mateix Microsoft Project s'encarregaria de fer el càlcul. L'únic que hauríem de fer és crear un recurs de treball⁵ per cada rol i assignar-hi les tasques adientment. A més, a cada recurs li assignaríem un cost per hora diferent.

⁵ Project anomena recursos tant els recursos materials com les persones que treballaran en el projecte. Els recursos de treball són les persones que fan feina al projecte.

Per calcular el nombre d'hores hem hagut d'introduir en el Microsoft Project la nostra jornada laboral:

Dia	Hores
Dilluns-Divendres	19:00-22:00
Dissabte i Diumenge	10:00-12:00 i 14:00-19:00

Taula 2 Hores per data

Aquest horari és un horari teoric, ja que molts dies feia més hores. Lo que he intentat fer és una mitja per compensar altres dies que en fes menys.

Sapiguent els dies que s'ha fet feina i les hores per dia el Project ja pot calcular el nombre d'hores total.

3 Context tecnològic

L'elecció de la plataforma condicionarà tot el projecte. Una bona elecció ens pot facilitar molt la feina o pot dificultar-la. Per això és convenient prendre's el seu temps a l'hora de seleccionar la plataforma d'acord amb els requeriments del projecte.

La plataforma seleccionada inicialment per desenvolupar el projecte va ser Java. Les principals que vaig tenir per triar Java varen ser:

- **Orientació a objectes:** L'orientació a objectes ens permet crear programes més modulars i mantenibles.
- **Multi plataforma:** Al ser un llenguatge multi plataforma ens permet triar el servidor que vulguem i no ens fermem a un sistema operatiu en concret.
- **Eines disponibles:** La multitud d'eines disponibles per Java ens ofereix un ventall molt ampli de llibreries i aplicacions.

A més, hi ha molta documentació sobre el desenvolupament d'aplicacions web sobre Java. Això facilitarà trobar respostes i informació quan trobem algun problema.

3.1 Desenvolupament d'aplicacions web sobre Java

A l'hora de desenvolupar aplicacions web sobre Java hi ha dos models bàsics de desenvolupament. Tradicionalment se'ls anomena "Model 1" i "Model 2".

3.1.1 Model 1

En les aplicacions basades en el "Model 1" les peticions les tracta directament les pàgines JSP o els servlets programats. Això fa que hi hagi

un alt acoblament i no ajuda a evitar la duplicació de codi. El tema de la duplicació de codi és més delicat ja que fa que hi puguin haver més errors i duplica la feina inútilment.

3.1.2 Model 2

Les aplicacions basades en el “Model 2” segueixen el model MVC⁶. Això afavoreix la separació entre les vistes i el codi d’aplicació. També facilita reduir l’acoblament i la duplicació de codi.

3.2 Spring Framework

Spring Framework és un entorn de desenvolupament d’aplicacions web basat en Java. Spring es basa en el “Model 2” i segueix molt fermament el patró MVC. A més ofereix una sèrie de facilitats a l’hora de crear aplicacions web que permeten reduir el codi escrit i afavorir l’escriptura de codi reutilitzable i amb un baix acoblament. Entre les facilitats que ofereix Spring Framework hi ha:

- **Contenedor d’IoC⁷**: L’inversió de control permet ajuntar els diferents components que formen part de l’aplicació emprant fitxers de configuració XML. L’aplicació més comuna d’aquest concepte s’anomena Injecció de dependències.
- **Integració amb Hibernate**: Spring s’integra molt bé amb Hibernate. Això permet reduir el codi d’accés a dades i redueix el codi repetitiu (tot el codi per crear les connexions, tancar-les, etc...) ja que ara ho gestiona Spring directament.

⁶ MVC: Model-Vista-Controlador

⁷ IoC: Inversion of Control: Inversió de Control

- **Acegi security:** Spring ofereix també un framework de seguretat anomenat Acegi security. Aquest framework ens permet assegurar una pàgina web de manera senzilla mitjançant un mecanisme de filtres.
- **Controladors per gestionar peticions http:** A més, Spring MVC ofereix una ampla selecció de controladors per facilitar el tractament de les peticions http, ja sigui per el tractament de formularis, el pas de paràmetres,...

Spring té moltes altres funcionalitats, però no són rellevants per a la realització d'aquest projecte. Mes endavant veurem amb més detall tots aquests temes.

3.3 Hibernate

Hibernate és un ORM: mapeja objectes del domini a consultes SQL. Això facilita enormement la programació de la capa d'accés a dades de l'aplicació, ja que ofereix un conjunt d'operacions per fer les cerques i actualitzacions a la Base de Dades. La manera de funcionar és que anotem les classes del domini amb unes anotacions especials que defineixen com es mapejen les classes a les taules de la Base de Dades. Aquest tema també el veurem amb més detall en un capítol posterior.

3.4 Servidors emprats

El servidor d'aplicacions emprat és Apache Tomcat⁸. Hem escollit aquest servidor pel seu baix cost (és software lliure i gratuït) i per la seva facilitat de instal·lació i configuració. A més, és un dels servidors d'aplicacions Java més estesos, cosa que permet trobar ajuda més fàcilment.

⁸ <http://tomcat.apache.org>.

Com a servidor de Base de Dades hem escollit MySQL⁹. També l'hem escollit per el seu baix cost (per projectes no comercials també és software lliure i gratuït). Així i tot, gràcies a l'ús d'Hibernate no hauria de ser molt complicat canviar de servidor de Base de Dades. Això és degut a que Hibernate proporciona una abstracció sobre l'accés a la Base de Dades que ens permet canviar-ho sense haver de modificar res més que la configuració de Hibernate al fitxer de configuració d'Spring.

3.5 Eines de desenvolupament

Per facilitar el desenvolupament del projecte hem emprat les següents eines:

- **Eclipse:** Eclipse¹⁰ és un IDE¹¹ per Java basat en plugins. Mitjançant els plugins es pot integrar amb moltes altres eines per millorar-ne les característiques.
- **Maven2:** Maven2¹² és un gestor de projectes. Mitjançant un fitxer de configuració en XML definim les dependències del projecte i ell ens les descarrega i les gestiona. A més, empra un concepte anomenat "Convention over Configuration": mitjançant una estructura estàndard de directoris disminuïm al mínim els fitxers de configuració.

⁹ <http://www.mysql.org>.

¹⁰ <http://www.eclipse.org>.

¹¹ IDE: Integrated Development Environment. Entorn de desenvolupament integrat.

¹² <http://maven.apache.org>.

- **Subversion:** Hem emprat Subversion¹³ com a sistema de control de versions. Així tenim les versions del projecte centralitzades a un servidor (he emprat el servidor de la FIB). Per si un cas també he anat guardant versions intermèdies a diferents ordenadors, però no m'ha fet falta recuperar-les en cap moment.

Seguidament descriurem la configuració que hem fet servir per cada una d'aquestes eines.

3.5.1 Eclipse

Per eclipse hem emprat una sèrie de plugins per facilitar la feina amb Spring, Maven2 i Subversion. En concret he emprat els plugins següents:

- **SpringIDE:** <http://springide.org/blog/> - Aquest plugin facilita la gestió dels fitxers de configuració d'Spring. A més permet generar diagrames de dependència entre els beans d'Spring.
- **M4E:** <http://code.google.com/p/q4e/> - Aquest plugin ens ajuda a gestionar el fitxer de configuració de Maven2. A més ens permet executar els objectius (targets).
- **Subclipse:** <http://subclipse.tigris.org/> - Subclipse en permet gestionar un projecte mitjançant Subversion des de el propi IDE. També ens permet veure diferències entre versions, pujar versions al servidor (commit) i veure quins fitxers han canviat i quins no.

A part, hem emprat la versió d'Eclipse per desenvolupar aplicacions J2EE. Amb aquesta versió ja ens venen els plugins per editar XML, fitxers JSP, etc...

¹³ <http://subversion.tigris.org>.

3.5.2 Maven2

La configuració de Maven2 és fa des de un sol fitxer XML anomenat *pom.xml*. Les parts més importants d'aquest fitxer són les següents.

```
<modelVersion>4.0.0</modelVersion>
  <groupId>cat.company.projecte</groupId>
  <artifactId>projectePXC</artifactId>
  <packaging>war</packaging>
  <name>projectePXC Maven Webapp</name>
  <version>1.0-SNAPSHOT</version>
  <url>http://maven.apache.org</url>
```

En aquestes línies definim les característiques del projecte. La part més important és la que diu `<packaging>war</packaging>` ja que defineix el projecte com un projecte web. Això també ens definirà l'estructura de directoris que haurem de tenir per que Maven2 sàpiga empaquetar-ho tot com un war per desplegar al contenidor web (Tomcat).

Seguidament venen algunes línies de configuració de Maven per determinar com es "compilaran"¹⁴ els objectius i també alguns repositoris no estàndards. A Maven s'usen repositoris per descarregar les llibreries de les que depèn el projecte. Seguidament venen les línies que defineixen les dependències. En posaré una per que es vegi com funciona:

```
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate</artifactId>
  <version>3.2.6.ga</version>
</dependency>
```

Les dependències s'agrupen en grups i tenen un nom que a Maven es diu "artifactId". A més, hem d'especificar la versió de la que depèn el nostre projecte. Amb aquesta informació Maven2 es descarrega la llibreria i ja resta disponible per el nostre projecte. A més, Maven2 gestiona les dependències de manera transitiva. Això vol dir, per exemple, que si una

¹⁴ Ho poso entre cometes per que no tots el objectius impliquen una compilació en el sentit estricte de la paraula.

dependència del nostre projecte depèn d'una altra llibreria, Maven2 s'encarrega de descarregar-la sense que ho haguem d'especificar nosaltres.

3.5.2.1 Estructura de directoris

Maven2 utilitza un concepte anomenat "*Convention over configuration*", o lo que és lo mateix convenció sobre configuració. Aquest concepte ens indica que és millor seguir unes estructures estàndard que haver de configurar-ho tot. Així, si usem Maven2 sempre serà millor emprar la seva configuració estàndard que haver de configurar-ho tot. La part més important d'aquesta "convenció" és l'estructura de directoris. L'estructura de directoris de Maven2 que hem emprat és la següent:

Directori	Explicació
src/main/java	Fitxer font en Java
src/main/resources	Fitxers de recursos (xml, properties,...)
src/main/webapp	Fitxers de l'aplicació web.

Taula 3 Estructura de directoris de Maven2

Tenint aquesta estructura ja podem crear el fitxer war per desplegar al servidor simplement executant la comanda `mvn package` o, si emprem el plugin per Eclipse simplement executarem l'objectiu des de dins Eclipse.

3.5.3 Subversion

Subversion no necessita configuració. Només hem de pujar al servidor la primera versió del codi que tinguem. Això normalment és l'estructura del projecte. A més, també convé emprar una estructura de directoris estàndard. Això és per el propi funcionament de Subversion. L'estructura recomanada és tenir un directori "`trunk`" per guardar la branca de desenvolupament principal, un directori "`branches`" per guardar-hi les

diferents branques que anem creant i un directori "tags" per anar guardant les versions a les que vulguem accedir directament. Per més informació veieu (Collins-Sussman, Fitzpatrick, & Pilato, 2008).

3.6 Conceptes bàsics d'Spring Framework

Per comprendre l'arquitectura del projecte convé fer un repàs dels conceptes bàsics d'Spring Framework. Els conceptes més importants que repassarem són:

- La configuració d'Spring
- La injecció de dependències
- El "Dispatcher servlet"

3.6.1 Configuració d'Spring

Els programes d'Spring es basen en POJOs¹⁵. Aquests objectes s'ajunten amb altres objectes amb relacions de dependència. Per definir aquestes dependències empren uns fitxers de configuració en XML. Com que en programes mínimament complexos podem tenir molts beans aquests fitxers es poden dividir en fitxers diferents.

3.6.2 Injecció de dependències

La injecció de dependències és un concepte bastant innovador. La clau és que enlloc de definir les dependències d'un objecte al mateix objecte simplement les definim com si fos una propietat d'un Java Bean. Llavors podem definir la propietat en un fitxer XML, dient a quin bean fa referència. Ho veurem en un exemple:

En un bean definim que necessitem tenir accés al bean entregaService.

¹⁵ POJO: Plain Old Java Object: Normalment s'anomenen així els Java Beans normals (que no són EJBs).

```
private EntregaService entregaService;

public EntregaService getEntregaService() {
    return entregaService;
}

public void setEntregaService(EntregaService entregaService) {
    this.entregaService = entregaService;
}
```

Llavors, al fitxer de configuració d'Spring definim clarament el bean i la dependència:

```
//Definim el bean:
<bean name="entregaService"
      class="cat.company.projecte.serveis.EntregaServiceImpl">
  <property name="entregaDao" ref="entregaDao" />
</bean>
//Definim la dependència:
<bean name="/mostraentrega.htm"
      class="cat.company.projecte.controladors.EntregaController">
  <property name="entregaService" ref="entregaService" />
</bean>
```

Aquí hem definit que el bean amb nom “/mostraentrega.htm” (després explicarem com es mapegen les adreces de la web als controladors) depèn del bean “entregaService”.

La injecció de dependències està explicada molt clarament a un article de Martin Fowler: veure (Fowler, 2004).

3.6.3 El dispatcher servlet

El dispatcher servlet és l'encarregat de rebre totes les peticions http i redirigir-les als controladors adients. Ve a ser el que dirigeix tot.

