

LLICÈNCIA CREATIVE COMMONS

Aquest treball es distribueix sota llicència Creative Commons CC BY-SA 3.0.

La llicència completa es pot consultar a <http://creativecommons.org/licenses/by-sa/3.0/deed.ca>.

PFC

DISSENY I IMPLEMENTACIÓ D'UN FRAMEWORK DE PERSISTÈNCIA



www.uoc.edu

Memòria | Llorenç Camps Vicente

Consultor: Òscar Escudero Sánchez

RESUM DEL PROJECTE

En aquest PFC - Disseny i implementació d'un framework de persistència - s'analitzarà els diversos frameworks de persistència que actualment existeixen i es facilitarà el desenvolupament d'un conjunt de components que permetran simplificar la capa de persistència en aplicacions multicapa desenvolupades amb JEE.

En aquesta memòria veiem que està formada per quatre apartats: descripció del PFC, anàlisi dels frameworks existents, disseny del framework i disseny de l'aplicació de proves.

En el primer apartat, que equival al primer punt de la memòria parlarem sobre els propòsits i els objectius generals del projecte a més del pla de treball amb el calendari i la planificació d'aquest.

El segon apartat estudiarem les diferents tecnologies que hi han actualment en el mercat i que utilitzarem de base alhora de realitzar el projecte. Primer parlarem de les diferents característiques de la persistència a més d'incloure un anàlisi dels models relacionals, ja que seran la base per implementar el Framework de persistència. I a continuació comentarem els diferents Frameworks de persistència: JPA, TopLink, Hibernate, MyBatis.

En el tercer apartat, el punt quatre de la memòria, realitzarem el disseny i l'arquitectura del framework de persistència tenint en compte l'estàndard JPA, comentat en l'apartat anterior. Parlarem de les principals característiques, amb un anàlisi de les seves funcionalitats, a més del diagrama de casos d'ús i de classes i per acabar alguns diagrames de seqüència de les funcionalitats més importants.

En el últim apartat, definirem el disseny de l'aplicació web que mostrarà el funcionament del framework realitzat. I al igual que en el tercer també hi haurà un diagrama de casos d'ús i de classes.

Per acabar, comentar que com a annexos hi hauran les guies d'instal·lació i ús tant del framework com de l'aplicació. A més del manual d'usuari de l'aplicació per veure tota la navegació de la web.

1 INTRODUCCIÓ	2
1.1 PROPÒSIT	2
1.2 OBJECTIUS GENERALS I ESPECÍFICS	2
1.3 PLA DE TREBALL	3
1.3.1 CALENDARI DE DESENVOLUPAMENT	3
1.3.2 PLANIFICACIÓ TEMPORAL	3
2 CAPA DE PERSISTÈNCIA	5
2.1 MECANISMES DE PERSISTÈNCIA	5
2.2 MAPEIG OBJECTE-RELACIONAT (ORM)	6
3 FRAMEWORKS DE PERSISTÈNCIA	7
3.1 CARACTERÍSTIQUES PRINCIPALS	7
3.2 JDBC	7
3.2 FRAMEWORKS	8
3.2.1 JAVA PERSISTENCE API (JPA)	8
3.2.2 TOPLINK ESSENTIALS	10
3.2.3 HIBERNATE	10
3.2.4 MYBATIS	12
3.2.5 JAVA DATA OBJECTS (JDO)	12
3.3 COMPARATIVA DE FRAMEWORKS	13
4 DISSENY DEL FRAMEWORK	14
4.1 CARACTERÍSTIQUES DEL FRAMEWORK	14
4.2 ANÀLISIS DE FUNCIONALITATS	15
4.3 CASOS D'ÚS	15
4.3.1 ESPECIFICACIÓ DELS CASOS D'ÚS	16
4.4 DIAGRAMA DE CLASSES	20
4.4 DIAGRAMES DE SEQÜÈNCIA	21
4.4.1 OBTENCIÓ D'UNA SESSIÓ	21
4.4.2 INSERCIÓ, MODIFICACIÓ I ELIMINACIÓ D'UN REGISTRE	22
4.4.3 CERCA DE REGISTRES	23
4.4.4 GENERACIÓ DELS BEANS DE BASE DE DADES	24
5 DISSENY DE L'APLICACIÓ D'EXEMPLE	25
5.1 INTRODUCCIÓ	25
5.2 CASOS D'ÚS	25
5.2.1 ESPECIFICACIÓ DELS CASOS D'ÚS	26

5.3 DIAGRAMA DE CLASSES	29
6 CONCLUSIONS	30
7 BIBLIOGRAFIA	31
8 ANNEX - INSTAL·LACIÓ I ÚS DEL FRAMEWORK	32
8.1 PREREQUISITS SOFTWARE	32
8.2 CONFIGURACIÓ ENTORN DE DESENVOLUPAMENT	32
8.3 ESTRUCTURA DEL PROJECTE	32
8.4 FITXERS DE CONFIGURACIÓ	33
8.5 COMPILACIÓ DEL FRAMEWORK	34
8.6 ÚS DEL FRAMEWORK	34
8.7 JAVADOC	34
9 ANNEX - INSTAL·LACIÓ I ÚS DE L'APLICACIÓ WEB	35
9.1 PREREQUISITS SOFTWARE	35
9.2 CONFIGURACIÓ ENTORN DE DESENVOLUPAMENT	35
9.3 ESTRUCTURA DEL PROJECTE	36
9.4 FITXERS DE CONFIGURACIÓ	36
9.5 COMPILACIÓ DE L'APLICACIÓ	36
9.6 ÚS DE L'APLICACIÓ WEB	36
9.6.1 PÀGINA D'INICI	37
9.6.2 REALITZAR RESERVA	37
9.6.3 CONSULTAR RESERVA	43
9.6.4 CONTINGUT ESTÀTIC	45
9.7 JAVADOC	47

1 INTRODUCCIÓ

1.1 Propòsit

El principal objectiu del projecte és desenvolupar un framework de persistència per aplicacions J2EE que facilitarà la comunicació entre l'aplicació i la bases de dades.

A més en el treball realitzarem un estudi dels diferents frameworks de persistència existents, com ara Hibernate, MyBatis, JPA, analitzant les seves principals característiques amb els seus avantatges i inconvenients que presenten cada un d'ells.

En el desenvolupament del framework a més de les funcionalitats bàsiques i imprescindibles necessàries per treballar amb la bases de dades, afegirem una funcionalitat per tal que a partir de la connexió a la bases de dades ens generi tots els objectes. Dit això, comentar que s'utilitzarà JPA com a referència alhora de desenvolupar el framework.

I un cop s'hagi desenvolupat el framework realitzaré una aplicació d'exemple per tal de demostrar les seves funcionalitats.

1.2 Objectius generals i específics

A partir del propòsit que he descrit l'objectiu general a assolir en aquest projecte d'investigació és la realització d'un estudi dels diferents frameworks existents i desenvolupar un framework per aplicacions J2EE, que tingui les funcionalitats bàsiques, i a més a més, que sigui lleuger i senzill.

Els principals objectius específics que es busquen assolir amb el projecte són:

- Avaluar la capa de persistència de JAVA: per estudiar la seva arquitectura i les parts més significatives.
- Analitzar les diferents implementacions de frameworks existents al mercat per aprofundir les possibilitats del framework que desenvoluparé durant la realització del projecte.
- Implementar i dissenyar un framework que faciliti el desenvolupament de les aplicacions multicapa.
- Avaluar les possibles millores del disseny per potenciar al màxim la funcionalitat de les aplicacions i reduir els inconvenients per tal de garantir les funcions bàsiques.
- Comprovar la utilitat del framework desenvolupat a través de la part pràctica i analitzar la seva efectivitat durant la realització de les proves de l'aplicació.

1.3 Pla de Treball

1.3.1 Calendari de desenvolupament

Per realitzar i planificar el pla de treball he pres com a base les dades de coordinació de l'assignatura que són les següents:

Fites de control		
Data	Fita	Observacions
02/10/2013	PAC 1	Lliurament del pla de treball
07/11/2013	PAC 2	Lliurament de l'estudi de les tecnologies existents, etc.
16/12/2013	PAC 3	Lliurament del disseny del framework desenvolupat.
13/01/2014	Memòria i presentació	Lliurament de la memòria, la presentació i el producte

La data inicial del projecte s'ha fixat en el dia d'inici del curs, el 18/09/2013 i el producte finalitzat s'ha de tenir el 13/01/2014, per tant disposem de 84 dies laborables per poder realitzar el projecte.

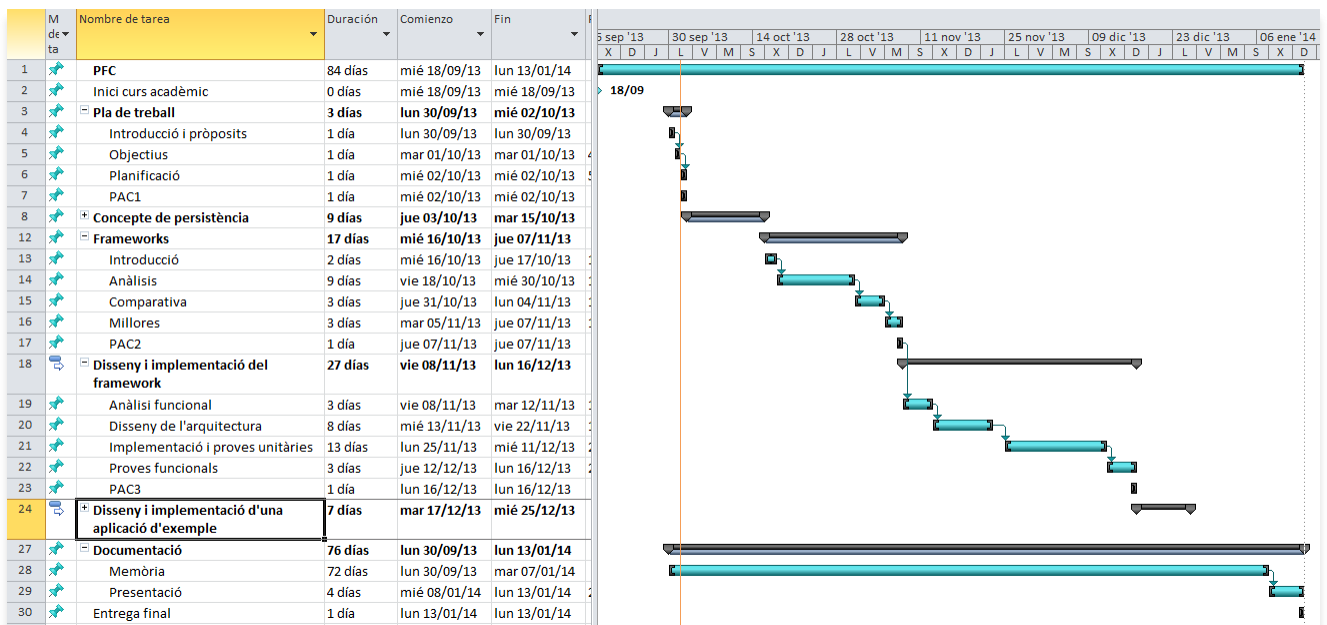
1.3.2 Planificació temporal

Per la planificació temporal mostrem les precedències de les activitats que formen el projecte:

Codi de l'activitat	Activitat	Descripció de l'activitat	Estimació (en dies)	Precedències
1	Inici curs acadèmic	inici curs	0	
2	Pla treball		3	
2.1	Introducció i propòsits	Definició del marc en el que es desenvolupa el framework i propòsits	2	1
2.2	Objectius	Definició dels objectius generals i específics	1	2.1
2.3	Planificació	Calendari del desenvolupament del projecte i planificació temp.	1	
3	Concepte de persistència		9	
3.1	Introducció	Introducció a la persistència de les dades	2	2
3.2	Anàlisi dels SGBD	Estudi dels diferents SGBD disponibles: models relacionals i altres.	5	3.1
3.3	Comparativa	Comparativa amb les avantatges i inconvenients de cadascun dels models.	2	3.2
4	Frameworks		17	
4.1	Introducció	Definició de frameworks i exemples d'us	2	3.3
4.2	Anàlisi	Estudi dels frameworks actuals: JPA, Hibernate, ...	9	4.1
4.3	Comparativa	Comparativa amb les avantatges i inconvenients de	3	4.2

		cadascun dels frameworks.		
4.4	Millores	Estudi de les millores de rendiment que es poden incorporar	3	4.3
5	Disseny i implementació del framework		27	
5.1	Anàlisi funcional	Anàlisi funcional amb els objectius i requeriments que ha d'implementar el framework	3	4
5.2	Disseny de l'arquitectura	Disseny de l'arquitectura del framework a desenvolupar	8	5.1
5.3	Implementació i proves	Implementació de la solució amb proves unitàries per a la validació del funcionament.	13	5.2
5.4	Proves funcionals	Validació funcional del framework desenvolupat	3	5.3
6	Disseny i implementació d'una aplicació d'exemple de l'ús del framework		7	
6.1	Objectius i requeriments	Objectius i requeriments de l'aplicació per a mostrar el correcte funcionament del framework de persistència	2	5
6.2	Implementació i proves	Implementació de l'aplicació web amb proves unitàries per a la validació del funcionament tant de la web com del framework.	5	6.1
7	Documentació		76	
7.1	Memòria	Memòria del projecte	72	
7.2	Presentació	Presentació del projecte	4	7.1

A continuació mostrem la planificació proposada amb la data d'inici del projecte el 18 de setembre del 2013.



Imatge 1. Diagrama de Gantt

2 CAPA DE PERSISTÈNCIA

Entenem com a persistència la capacitat de persistir, per tant que la informació quedi guardada de manera permanent i a posteriori poder recuperar i utilitzar aquesta informació. Dit això, el interès d'incorporar la persistència en l'aplicació recau en què les dades a memòria es perden, i que si es volen utilitzar a posteriori, és necessari molta memòria per poder gestionar aquesta informació.

Cal dir que la persistència és una de les parts més crítiques en el disseny de qualsevol aplicació, degut a que les crides d'entrada i sortida de dades afecten en el seu rendiment, i per tant, s'ha d'avaluar el seu correcte funcionament.

A continuació explicaré els principals mecanismes de persistència a més del mapeig de l'objecte relacionat.

2.1 Mecanismes de persistència

Els mecanismes de persistència s'utilitzen per resoldre l'accés a les dades i posteriorment poder-los manipular.

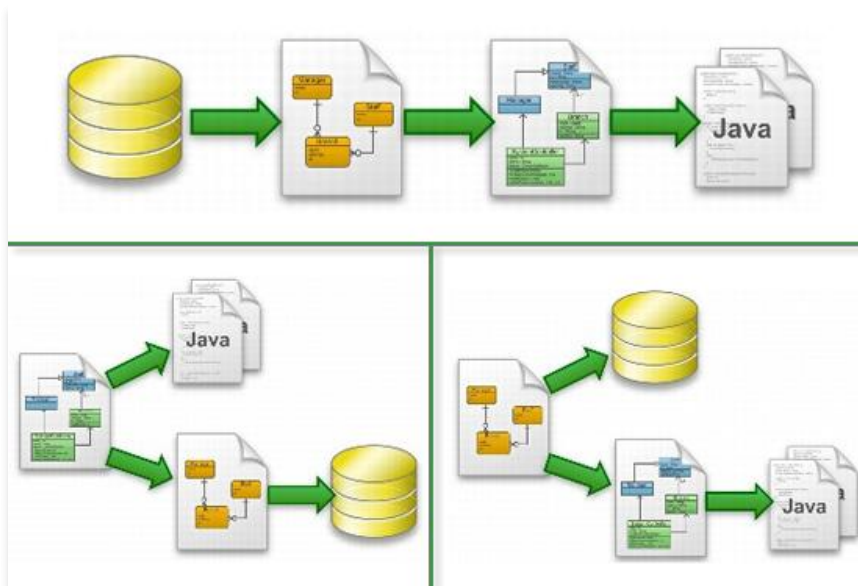
A continuació enumerarem algun dels diferents mecanismes més importants:

- **Orientació a aspectes:** La programació orientada a aspectes (POA) és un paradigma de programació que permet separar la part tècnica de les funcionalitats de l'aplicació. En aquest paradigma les classes es dissenyen i implementen de forma separada als aspectes i es fusionen a posteriori. Dins de cada mòdul es modularitza la persistència per a poder reutilitzar el codi generat.
- **Accés directe a base de dades:** amb aquest mecanisme s'aconsegueix accés directe a la base de dades per implementar la persistència, per tant, sense fer ús de cap tipus de middleware entre l'aplicació i el sistema de gestió de base de dades (SGBD).
- **Generadors de codi:** gràcies als generadors de codi s'aconsegueixen eines que generen un codi correcte, basat en patrons de disseny per resoldre la persistència del sistema, a partir de la metadata d'un projecte.
- **"Mappers":** són mecanismes que es basen en la traducció bidireccional entre les dades encapsulades en els objectes de la lògica d'un sistema orientat a objectes i una font de dades que va servir una paradigma diferent. S'encarrega de convertir les dades de dos paradigmes diferents; hi ha dos tipus de mappers: el primer serà el que s'utilitzarà per fer el framework i és el Mapping Objecte-Relacional) i el segon és el Mapping Objecte-XML.

- Esquemes de Prevalença: són frameworks que mantenen els objectes en una memòria volàtil i que periòdicament utilitzen mecanismes de serialització d'objectes per guardar una imatge (snapshot) dels objectes de l'aplicació. Encara que és una proposta simple i ràpida a la persistència presenta problemes d'escalabilitat.
- Llenguatges orientat a objectes: busca resoldre la persistència de dades utilitzant les funcionalitats del propi llenguatge de programació, evitant la inclusió de frameworks o altres eines alienes al llenguatge.

2.2 Mapeig objecte-relacionat (ORM)

El framework a desenvolupar es basarà en aquesta base de dades relacional, per tant, en aquest punt s'explicarà com persistir els objectes amb ORM. Un ORM, es tracta d'una capa que permet relacionar objectes amb el model de dades relacional, tal com veiem en la imatge:



Imatge 2. Esquema ORM

La dificultat recau en com donar solució a la correspondència entre el model objecte i el model relacional.

A continuació mostrem una taula amb la relació entre el model d'objectes i el model relacional

Model d'objectes	Model Relacional
Classe	Taula, Vista
Objecte	Fila
Identificador	Clau primària
Punter a un altre objecte	Clau forana
Herència	Relació afegida a tots els atributs de l'objecte.

3 FRAMEWORKS DE PERSISTÈNCIA

Quan parlem de Framework (marc de treball) ens referim a una arquitectura de software composta de components que poden ser personalitzables i que ens poden servir de base per la organització i desenvolupament de software.

Un framework defineix com els objectes col·laboren entre si per representar el domini que s'està dissenyant, i agafa tots els aspectes del domini que es consideren invariables.

En el desenvolupament del programari, un framework és un conjunt de classes que s'utilitzen per implementar una estructura estàndard d'una aplicació, o sigui un patró pel desenvolupament i implementació d'una aplicació completa.

Els principals avantatges són els següents:

- Reutilitzar components de programari.
- Reducció del manteniment.
- Agilitzar el desenvolupament d'aplicacions. Ja que permet en el programador centrar-se en el desenvolupament de l'aplicació.

Els principals inconvenients són els següents:

- Dificultats en l'aprenentatge, ja que l'usuari ha d'estudiar el que proporciona el framework i com utilitzar-lo.
- Limitació en la flexibilitat, ja que els programadors s'han d'adaptar a les funcionalitats que ofereix el framework.

A continuació comentarem els diferents frameworks més importants i només tindrem en compte els SGBD relacionals. També comentarem l'estàndard que utilitza JAVA per connectar-se a la base de dades.

3.1 Característiques principals

Les principals característiques dels frameworks són les següents:

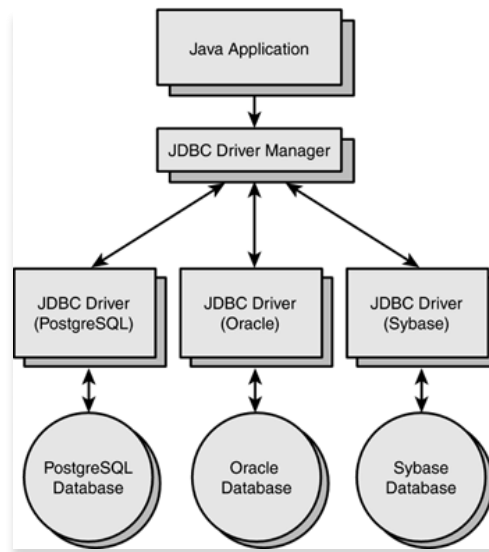
- Correspondència d'esquemes.
- Identitat d'objectes i registres.
- Conversor de la base de dades, o sigui transformació d'un registre en un objecte i viceversa.
- Suport a les transaccions: commit i rollback.

3.2 JDBC

JDBC (Java Data Base Connectivity) és l'estàndard que utilitza Java per a la connexió amb les bases de dades i permet accedir a les bases de dades independentment del SGBD.

Per utilitzar aquesta API amb un determinat SGBD només es necessitarà un adaptador (driver) del motor de base de dades que farà de intermediari entre el SGBD i la base de dades.

A continuació veiem la seva estructura:



Imatge 3. Arquitectura JDBC

Les principals classes que ofereix JDBC són les següents:

- **DriverManager**: Carrega un driver d'un SGBD determinat.
- **Connection**: Estableix connexions amb la base de dades.
- **Statement**: Executa sentències SQL i les envia a la base de dades.
- **ResultSet**: Emmagatzema el resultat d'una consulta.

Permet tres tipus de sentències:

- **Statement**: Les sentències s'envien a la base de dades una per una cada vegada.
- **PreparedStatement**: Compila la sentència abans d'executar-la i permet executar-la múltiples vegades de manera més eficient.
- **CallableStatement**: S'utilitza per executar store procedures a la base de dades.

També comentar que inclou suport per transaccions per tant es pot fer rollback si fos necessari.

3.2 Frameworks

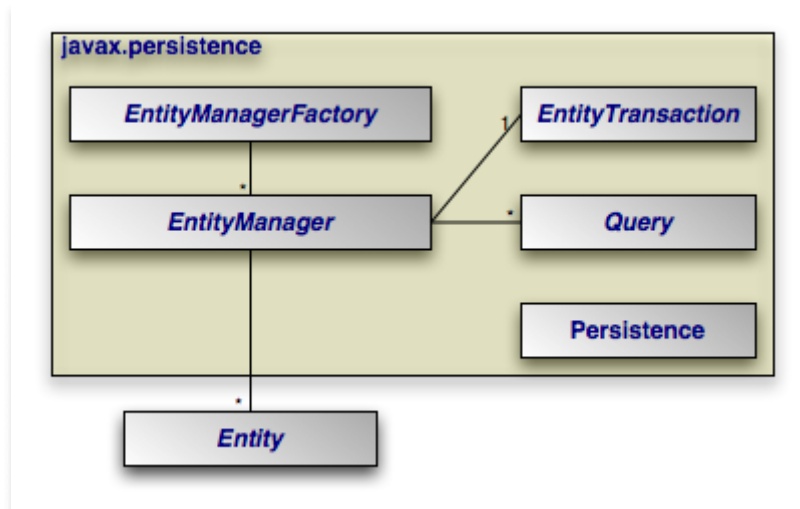
A continuació veurem les diferents alternatives de frameworks de persistència per Java que s'utilitzen que estan basats en JDBC amb les seves característiques.

3.2.1 Java Persistence API (JPA)

La Java Persistence API, és la API que apareix a partir de la utilització de Hibernate i TopLink. La JPA va ser definida com a part de l'especificació de l'estàndard EJB 3.0 desenvolupada per a la plataforma Java EE 5, i està basada en POJO's per a fer el mapping amb les bases de dades relacionals en Java.

La JPA defineix la metadada mínima pel mapping d'objectes Java a una base de dades relacional. Aquest mapping entre les entitats Java i les taules de base de dades es realitza mitjançant anotacions en les pròpies classes de l'entitat, així d'aquesta manera no es necessiten els fitxers descriptius XML.

La API de persistència està implementada a la classe anomenada EntityManager, que està representada per la interfície EntityManager. Aquests EntityManager son generats per a instancies de la EntityManagerFactory i una Persistence Unit es la que s'encarrega de definir les entitats que s'utilitzen o administren per un EntityManager, tal i com veiem en la següent imatge.



Imatge 4. Arquitectura JPA

Les principals característiques són les següents:

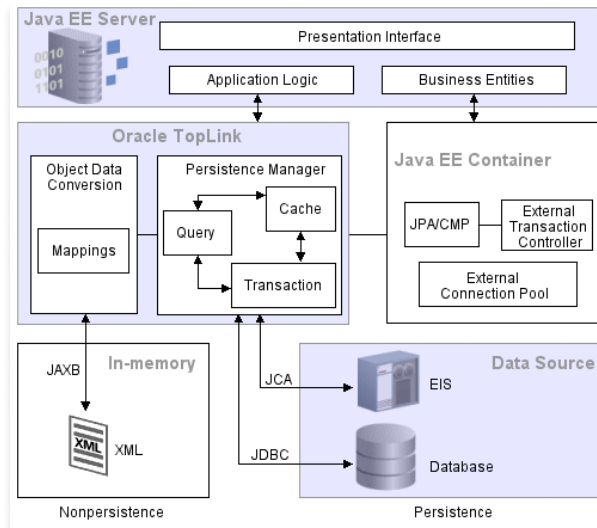
- Consultes orientades a objectes (JPQL). Les consultes permeten recórrer els objectes, les seves propietats i relacions sense necessitat de conèixer les columnes o claus foranes.
- Permet relacions ManyToOne, OneToOne, OneToMany i ManyToMany a través de les anotacions.
- Inclou control de la transaccionalitat de les consultes.
- Inclou una cache per optimitzar l'accés a les dades.

Les entitats JPA adopten un model de persistència lleuger que està dissenyat per treballar a la perfecció amb Oracle TopLink i Hibernate.

Per acabar, recordar que JPA és una especificació no una implementació, i per tant és necessari un framework que implementi aquesta especificació.

3.2.2 TopLink Essentials

TopLink Essentials és la versió gratuïta del producte Oracle TopLink i correspon a la implementació per a l'especificació JPA 1.0. Proporciona un mecanisme molt flexible i productiu per guardar els objectes i els EJB's de Java en bases de dades relacionals.



Imatge 5. Arquitectura Oracle TopLink

Les principals característiques són:

- XML no transaccional i no persistent, només en memòria.
- Conversió entre objectes Java i documents XML utilitzant JAXB.
- Configuració senzilla mitjançant anotacions alineades amb JPA.
- Control de concurrència.
- Permet configurar perquè emmagatzemi el resultat d'una consulta en una cache interna.
- Suporta persistència transaccional d'objectes relacionals utilitzant JDBC.

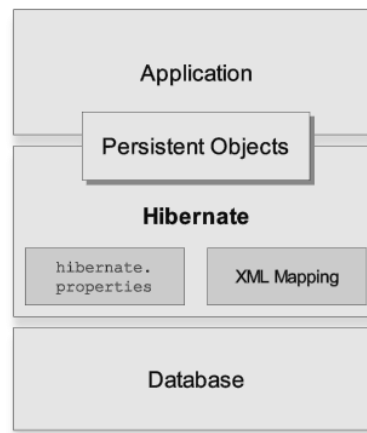
3.2.3 Hibernate

Hibernate és un framework de persistència que s'utilitza per guardar les dades de l'entorn Java a la base de dades relacionals. Serveix per mapejar objectes i relacions (ORM).

Hibernate utilitza les classes POJO juntament amb fitxers XML que permet relacionar aquests objectes amb la base de dades.

El seu objectiu principal és reduir al màxim en el programador totes les tasques de la persistència que es poden automatitzar. Disposa d'un Hibernate Tools que estan disponibles per Eclipse i ANT per tal de facilitar la seva utilització.

Amb Hibernate es crea una capa entre la base de dades i l'aplicació tal i com veiem en la següent imatge:



Imatge 6: Arquitectura Hibernate

Les principals característiques que ofereix són:

- Hibernate crea objectes persistents que sincronitzen les dades entre l'aplicació i la base de dades.
- Suporta l'ús d'Anotacions.
- Les consultes es poden realitzar a nivell d'objectes (HSQL) i a nivell de SQL.
- Implementa l'estàndard JPA.
- Permet herència i polimorfisme.
- Simplifica el codi i permet identificar els errors més fàcilment.
- No està restringit a cap tipus de dades Java ni objectes ni primitius.
- S'abstrau del SGBD utilitzat.
- Es distribueix sota la llicència LGPL.

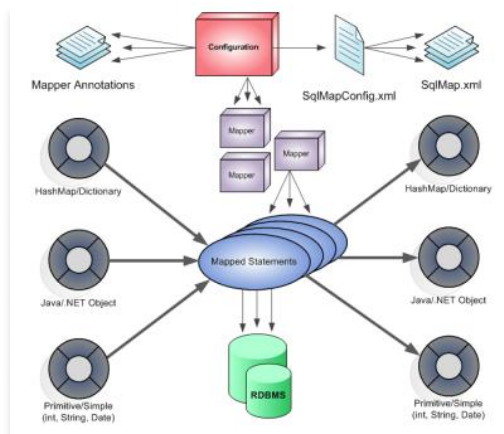
A continuació comentaré les classes i fitxers de configuració més importants que s'utilitzen:

- **SessionFactory:** És una instància singleton que implementa el patró de disseny Factory. Carrega el fitxer hibernate.cfg.
- **TransactionFactory:** Implementa totes les opcions de configuració en una base de dades.
- **ConnectionProvider:** Igual que el TransactionFactory.
- **Session:** És la instància de connexió amb la base de dades. I cada objecte de sessió es crea a partir dels objectes SessionFactory.
- **Transaction:** És la instància de transacció que representa una sola unitat atòmica de treball.
- **hibernate.cfg:** Fitxer xml a on tindrà la configuració d'hibernate.

3.2.4 MyBatis

MyBatis, conegut també com iBatis és un framework de persistència lliure que està disponible tant per la plataforma Java com .NET i està desenvolupat sota la llicència Apache 2.0. Elimina quasi tot el codi JDBC, l'establiment de paràmetres i la obtenció de resultats.

MyBatis no és un framework ORM, per tant no fa mapping d'objectes sinó de SQL i de procediments guardats a partir de fitxers XML o anotacions Java.



Imatge 7. Arquitectura de MyBatis

Les característiques més importants són:

- Simplifica la programació en front l'ús de JDBC.
- Proporciona un motor de mapeig de resultats SQL a arbres d'objectes.
- Suporta la composició de seqüències SQL dinàmiques mitjançant XML.
- Suporta l'ús de cache declaratives, com ara OSCache, EHCache.

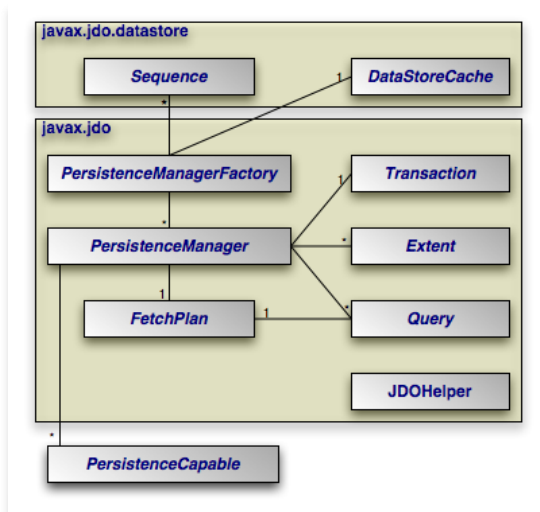
3.2.5 Java Data Objects (JDO)

JDO és un estàndard de persistència d'objectes Java dissenyat per Sun. Es tracta d'una alternativa a JPA.

Les característiques més importants són:

- Transparència en els serveis de persistència pel model del domini.
- Transparència alhora d'emmagatzemar, recuperar i modificar les instàncies.
- Suporta transaccions.
- Els objectes de persistència es defineix per fitxers XML.
- Fàcil d'usar: Els programadors poden centrar-se en el model i deixar els detalls de la persistència a la implementació JDO.
- Portabilitat: Les aplicacions que utilitzen l'API JDO es poden executar en diferents implementacions sense la necessitat de compilar el codi.
- Independència de la base de dades.

- Integració amb EJB: Les aplicacions poden aprofitar les característiques dels EJB, com el processament remot de missatges, etc.



Imatge 8. Arquitectura de JDO

3.3 Comparativa de frameworks

A continuació realitzem una comparativa dels principals conceptes dels frameworks analitzats.

	JPA	JDO	MyBatis
Tipus base de dades	Relacionals	Relacionals, XML, OO.	Relacionals
Llenguatge consultes	JPQL, HQL	JDOQL	SQL
Orientat a objecte	Si, i suporta conceptes avançats	Si, i suporta conceptes avançats	Si, i suporta conceptes avançats
Anotacions	Suporta	Suporta	No suporta
Simplicitat	Simple, es basa en POJO's	Simple, es basa en POJO's	Simple, fa servir sentències SQL
Configuració	Un únic fitxer XML o de properties	Un únic fitxer XML persistence.xml	Fitxers XML

Per acabar realitzem una comparativa de les dues implementacions comentades en els punts anteriors que son Oracle TopLink i Hibernate que implementen l'estàndard JPA.

	TopLink	Hibernate
Consultes dinàmiques	Suporta	Suporta
Creació automàtica d'esquema	Suporta	Suporta
Construcció manual SQL	No suporta	No suporta

DBMS suportats	JDBC	JDBC
Suport cascada	Suporta	Suporta
Generació de codi	No suporta	No Suporta
Relacions suportades	1 : 1	1 : 1
	1 : N	1 : N
	N : M	N : M
Llenguatge de consulta	JPQL	HQL

Veiem que les dos tenen les mateixes funcionalitats i limitacions.

4 DISSENY DEL FRAMEWORK

4.1 Característiques del framework

Després de revisar els diferents frameworks amb les seves característiques veiem que el nostre framework ha de:

- Ser independent de la base de dades i per tant s'ha de poder utilitzar en les diferents bases de dades existents.
- Ha de ser tenir una configuració senzilla, a on només li passarem la cadena de connexió amb el seu usuari i la paraula de pas.
- S'ha d'encarregar de generar automàticament les funcionalitats bàsiques CRUD (Create, Read, Update, Delete), a més d'alguna funcionalitat més.

Dit això, comentar que el disseny estarà basat en els principis SOLID, que ens permetran crear un codi més llegible, simple, reutilitzable, etc... Els cinc principis són els següents:

- SRP (Single responsibility principle): Cada classe només ha de tenir una única responsabilitat.
- OCP (Open close principle): Totes les classes, funcions, ... han d'estar obertes a estendre's però tancades a les modificacions. O sigui s'han de poder afegir noves funcionalitats sense tenir que modificar el seu comportament.
- LSP (Liskov substitution principle): Cada classe que hereta d'una altre pot utilitzar-se com el seu pare sense necessitat de conèixer les diferències entre elles.
- ISP (Interface segregation principle): Una classe que té dependència d'una altre classe no ha de dependre dels mètodes d'aquesta classe que no hagi d'utilitzar.
- DIP (Dependency inversion principle): S'ha de dependre de les abstraccions no de les concrecions. O sigui les classes d'alt nivell no haurien de dependre de les classes de baix nivell, ja que totes dos haurien de dependre de les abstraccions. I les abstraccions

no haurien de dependre dels detalls sinó que aquests haurien de dependre de les abstraccions.

4.2 Anàlisis de funcionalitats

Abans de començar a mostrar els diferents diagrames voldria comentar algunes de les decisions preses alhora de fer el disseny del framework.

He dividit les diferents funcionalitats en tres paquets descrits de la següent manera:

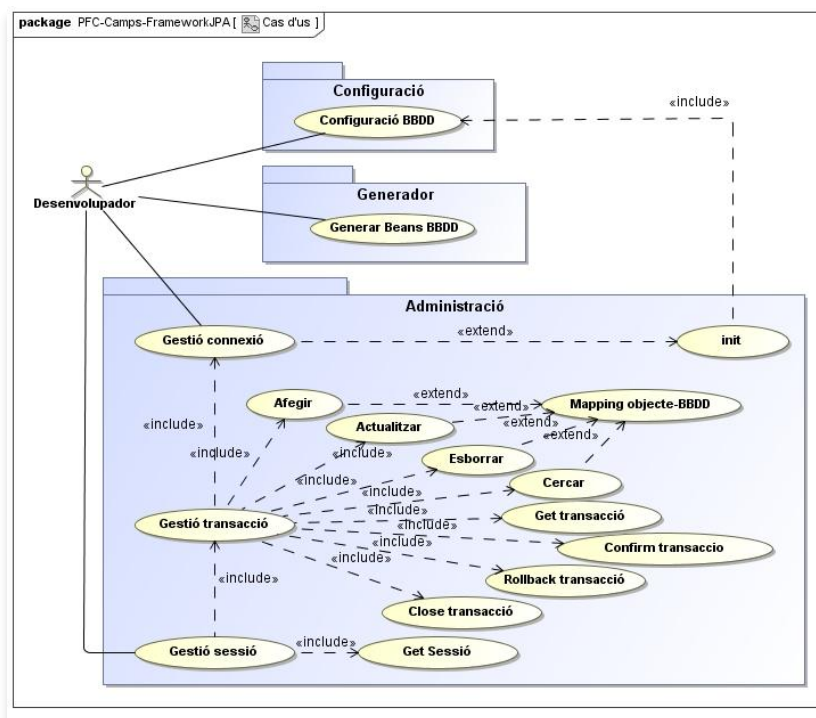
- **Configuració:** Aquest paquet gestionarà la funcionalitat de configuració de la base de dades.
- **Administració:** Aquest paquet gestionarà tota l'administració del framework, o sigui tindrà totes les operacions bàsiques, de creació, eliminació, actualització i consulta. També serà l'encarregat de gestionar les sessions i les transaccions i per tant aquestes han de complir les característiques **ACID** (Atomicitat, Consistència, Aïllament, Definitivitat), o sigui s'ha de garantir que totes les transaccions siguin fiables tant en el supòsit que ha anat bé i per tant s'han de reflectir tots els canvis en la base de dades com en el cas que hagi sorgit algun problema i per tant no s'hagi guardat cap canvi.
- **Generador:** Aquest paquet s'encarregarà de generar de manera automàtica tots els Beans de la base de dades.

Dit això, veiem que poden haver-hi dos actors alhora d'utilitzar el framework:

- **Desenvolupador:** Correspon al perfil de l'usuari encarregat d'utilitzar el framework.

4.3 Casos d'ús

Després de veure l'apartat anterior el diagrama de casos d'ús queda de la següent manera:



4.3.1 Especificació dels casos d'ús

Un cop vist el diagrama passarem a comentar cada un dels casos d'ús:

Nom	Configuració BBDD
Descripció	Realitza la configuració a la base de dades a partir de les dades del fitxer de configuració.
Flux principal	<ol style="list-style-type: none"> 1. El desenvolupador ha de crear el fitxer <i>uoc-persistenciaFW-config.properties</i>. 2. El desenvolupador introdueix les següents dades: <ol style="list-style-type: none"> 1. Introdueix la URL de la base de dades. 2. Introdueix l'usuari de la base de dades. 3. Introdueix el password de la base de dades. 4. Introdueix el directori base a on es generaran els beans de base de dades. 5. Introdueix el paquet dels beans a on es generaran.

Nom	Generar Beans BBDD
Descripció	Realitza la creació dels Beans relacionats amb les taules de la base de dades a partir de l'usuari/password.
Flux principal	<ol style="list-style-type: none"> 1. El desenvolupador ha d'introduir en el fitxer de configuració els punts 2.4 i 2.5. 2. El desenvolupador fa una crida al mètode de generateDBBeans(). 3. El sistema es connectarà amb la base de dades. 4. El sistema generarà els Beans relacionats amb les taules que continguin en l'usuari de la base de dades.
Flux alternatiu	<ol style="list-style-type: none"> 1. En el pas 3 i pas 4 en cas d'error, el sistema crearà una excepció i s'aturarà l'execució .

Nom	init
Descripció	A partir de les dades de configuració inicialitza el framework de persistència obtenint una connexió amb la base de dades.
Flux principal	<ol style="list-style-type: none"> 1. Amb les dades que ha introduït l'usuari en el cas d'ús de configuració, l'aplicació s'encarrega de carregar aquestes dades i realitzar la inicialització de la connexió. 2. Els sistema es connecta amb la base de dades. 3. El sistema retorna la connexió.
Flux alternatiu	<ol style="list-style-type: none"> 1. En el pas 2 en cas d'error, el sistema crearà una excepció i s'aturarà l'execució.

Nom	Gestió connexió
Descripció	Gestió de la connexió a la base de dades, inclou la connexió a la base de

	dades.
Flux principal	1. El desenvolupador obté una connexió.
Flux alternatiu	1. En el pas 1 en cas d'error, el sistema crearà una excepció i s'aturarà l'execució.

Nom	Gestió sessió
Descripció	Gestiona les connexions a la base de dades. Inclou tots els casos d'us de gestió de les transaccions.
Flux principal	<ol style="list-style-type: none"> 1. El desenvolupador obté una nova sessió. 2. Aquest realitza les operacions que vulgui. 3. Quan finalitza tanca la sessió.
Flux alternatiu	1. En qualsevol dels 3 passos si hi ha error, el sistema crearà una excepció i s'aturarà l'execució.

Nom	Get sessió
Descripció	Obté una sessió.
Flux principal	<ol style="list-style-type: none"> 1. El desenvolupador crea una nova sessió a partir del mètode <code>createSessionFW()</code>. 2. El sistema es connecta a la base de dades. 3. El sistema retorna una sessió.
Flux alternatiu	1. En el pas 2 i 3 en cas d'error, el sistema crearà una excepció.

Nom	Gestió transacció
Descripció	Gestiona les característiques ACID (Atomicitat, Consistència, Aïllament, Definitivitat).
Flux principal	<ol style="list-style-type: none"> 1. El desenvolupador demana una transacció a través de l'objecte de <code>SessioFW</code>. 2. Aquest realitza les operacions que desitgi, i un cop realitzades fa el <code>confirm()</code>. 3. El desenvolupador finalitza la transacció amb un <code>close</code>.
Flux alternatiu	1. En el pas 2, en cas d'error o d'un comportament no desitjat, es tiren enrere tots els canvis a través del <code>rollback()</code> .

Nom	Afegir
Descripció	És l'encarregat d'afegir un nou registre a la taula de base de dades segons l'objecte que ens passin.

Flux principal	<ol style="list-style-type: none"> 1. Es crea una instància del Bean que equival a la taula de base de dades amb les dades corresponents. 2. S'executa el mètode <i>insert()</i> amb el Bean que hem creat en el pas 1. 3. Obtenim l'id de la clau primària que ha creat en la taula.
Flux alternatiu	<ol style="list-style-type: none"> 1. En el pas 3, en cas d'error el sistema retornarà una excepció.

Nom	Actualitzar
Descripció	És l'encarregat d'actualitzar els valors d'un determinat registre d'una de les taules de la base de dades segons l'objecte que ens passin.
Flux principal	<ol style="list-style-type: none"> 1. Es crea una instància del Bean que equival a la taula de base de dades. 2. Obtenim la clau primària del registre que volem modificar. 3. Emplenem la instància que hem creat en el punt 1. 4. S'executa el mètode <i>update()</i> amb el Bean que hem emplenat en el pas 3. 5. Es realitza la modificació.
Flux alternatiu	<ol style="list-style-type: none"> 1. En el pas 2 i 4, en cas d'error el sistema retornarà una excepció. 2. En el pas 5, si no es realitza la modificació el sistema retornarà una excepció.

Nom	Esborrar
Descripció	És l'encarregat d'esborrar un registre a la taula de base de dades segons l'objecte que ens passin.
Flux principal	<ol style="list-style-type: none"> 1. Es crea una instància del Bean que equival a la taula de base de dades. 2. Obtenim la clau primària del registre que volem eliminar. 3. Emplenem la instància que hem creat en el punt 1. 4. S'executa el mètode <i>delete()</i> amb el Bean que hem emplenat en el pas 3. 5. Es realitza l'esborrat.
Flux alternatiu	<ol style="list-style-type: none"> 1. En el pas 2 i 4, en cas d'error el sistema retornarà una excepció. 2. En el pas 5, si no es realitza la modificació el sistema retornarà una excepció.

Nom	Cercar
Descripció	És l'encarregat d'obtenir un llistat amb els registres que compleixin amb les condicions de cerca.
Flux principal	<ol style="list-style-type: none"> 1. Es crea una instància del Bean que equival a la taula de base de dades. 2. S'emplena amb les dades que volem cercar. 3. S'executa el mètode <i>findListFrom..()</i> amb el Bean que hem emplenat en el pas 2. 4. Obtenim els resultats de la cerca.
Flux	<ol style="list-style-type: none"> 1. En el pas 3, en cas d'error el sistema retornarà una excepció.

alternatiu

Nom	Mapeig objecte-BBDD
Descripció	És l'encarregat de realitzar la traducció del codi JAVA al codi SQL .
Flux principal	<ol style="list-style-type: none"> 1. El desenvolupador realitza alguna de les operacions descrites en els 4 casos d'us anteriors. 2. El sistema obté la configuració de la base de dades per poder fer el mapping. 3. El sistema accedeix a la taula a través dels mappings dels objectes. 4. El sistema genera el codi SQL. 5. El sistema executa la query. 6. El sistema recupera les dades i les transforma en un Bean. 7. El sistema retorna les dades recuperades.
Flux alternatiu	<ol style="list-style-type: none"> 1. En els passos 2, 4, 5 i 6, en cas d'error, el sistema retorna una excepció.

Nom	Get transacció
Descripció	Obtenim una nova transacció.
Flux principal	<ol style="list-style-type: none"> 1. El desenvolupador demana una nova transacció a través de l'objecte <i>SessionFW</i>, passant-li el valor de l'autoCommit. 2. El sistema crea una nova sessió. 3. El sistema retorna la sessió creada.
Flux alternatiu	<ol style="list-style-type: none"> 1. En el pas 2, en cas d'error el sistema retorna una excepció.

Nom	Confirm transacció
Descripció	Realitza la confirmació de totes els canvis que s'hagin realitzat en la transacció.
Flux principal	<ol style="list-style-type: none"> 1. El desenvolupador obté la transacció a partir de l'objecte <i>SessionFW</i>. 2. El desenvolupador confirma les modificacions a través del mètode <i>confirm()</i>.
Flux alternatiu	<ol style="list-style-type: none"> 1. En el pas 2, en cas d'error el sistema retorna una excepció.

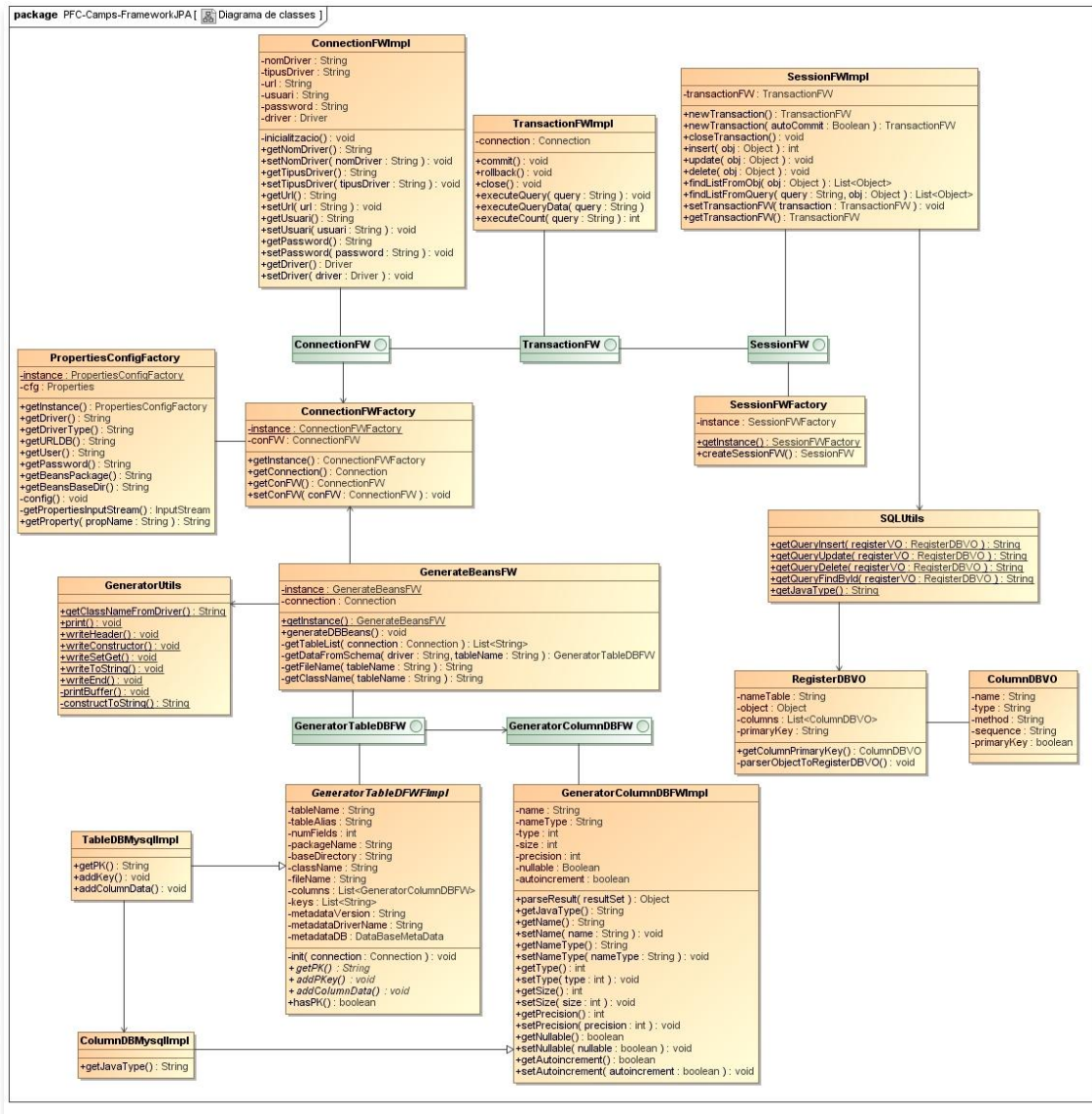
Nom	Rollback transacció
Descripció	Tira enrere tots els canvis que s'hagin realitzat en la transacció.
Flux principal	<ol style="list-style-type: none"> 1. El desenvolupador obté la transacció a partir de l'objecte <i>SessionFW</i>. 2. El desenvolupador tira enrere les modificacions a través del mètode <i>rollback()</i>.

Flux alternatiu	1. En el pas 2, en cas d'error el sistema retorna una excepció.
------------------------	-----------------------------------------------------------------

Nom	Close transacció
Descripció	Tanca la transacció.
Flux principal	<ol style="list-style-type: none"> 1. El desenvolupador obté la transacció a partir de l'objecte <i>SessionFW</i>. 2. El desenvolupador tanca la transacció a través del mètode <i>close()</i>.
Flux alternatiu	1. En el pas 2, en cas d'error el sistema retorna una excepció.

4.4 Diagrama de classes

A continuació podem veure el diagrama de classes amb les principals funcions



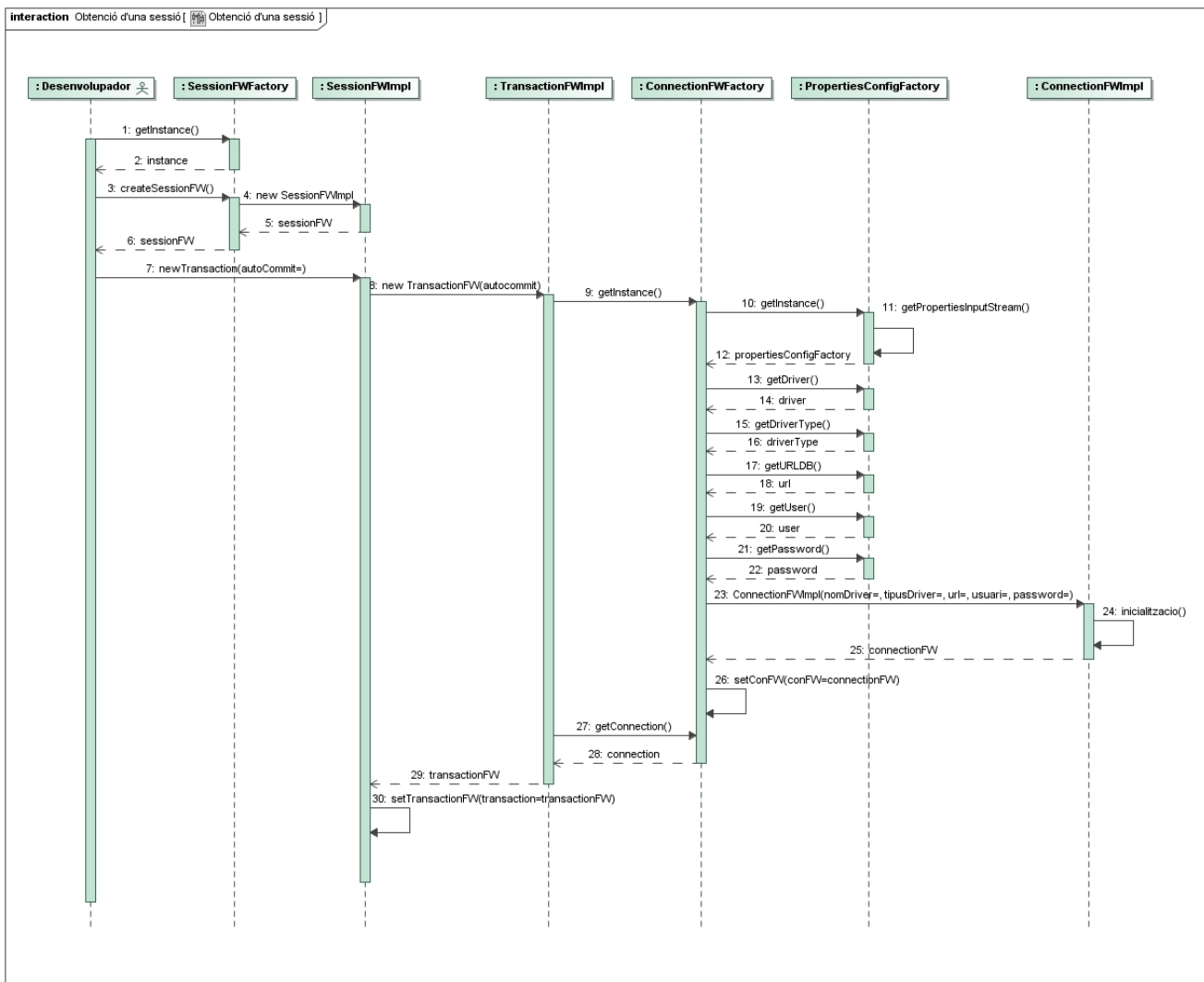
4.4 Diagrames de seqüència

En aquest apartat explicarem algun dels diferents diagrames de seqüència per veure alguna de les interaccions entre els diferents objectes del framework.

4.4.1 Obtenció d'una sessió

La creació de la sessió es realitza a través de la classe *SessionFWFactory*. A partir d'aquest objecte obtenim el *SessionFW* a on podem crear una transacció a partir dels dos mètodes que hi han per aquest propòsit, el *newTransaction()* i el *newTransaction(autoCommit)*. En el primer internament el *commit* serà true i per tant tots els canvis en la base de dades es realitzaran, i en el segon el programador informará del valor del commit; en aquest cas si passa un valor false ell haurà de gestionar si vol aplicar o no els canvis amb els mètodes *commit()* i *rollback()* que conté l'objecte *TransactionFW*.

Dit això, quan es fa el *newTransaction()*, internament crida a les classes *ConnectionFWFactory* i *PropertiesConfigFactory* per tal d'obtenir els valors del fitxer de configuració i crear una connexió amb la base de dades tal com veiem en el següent diagrama:

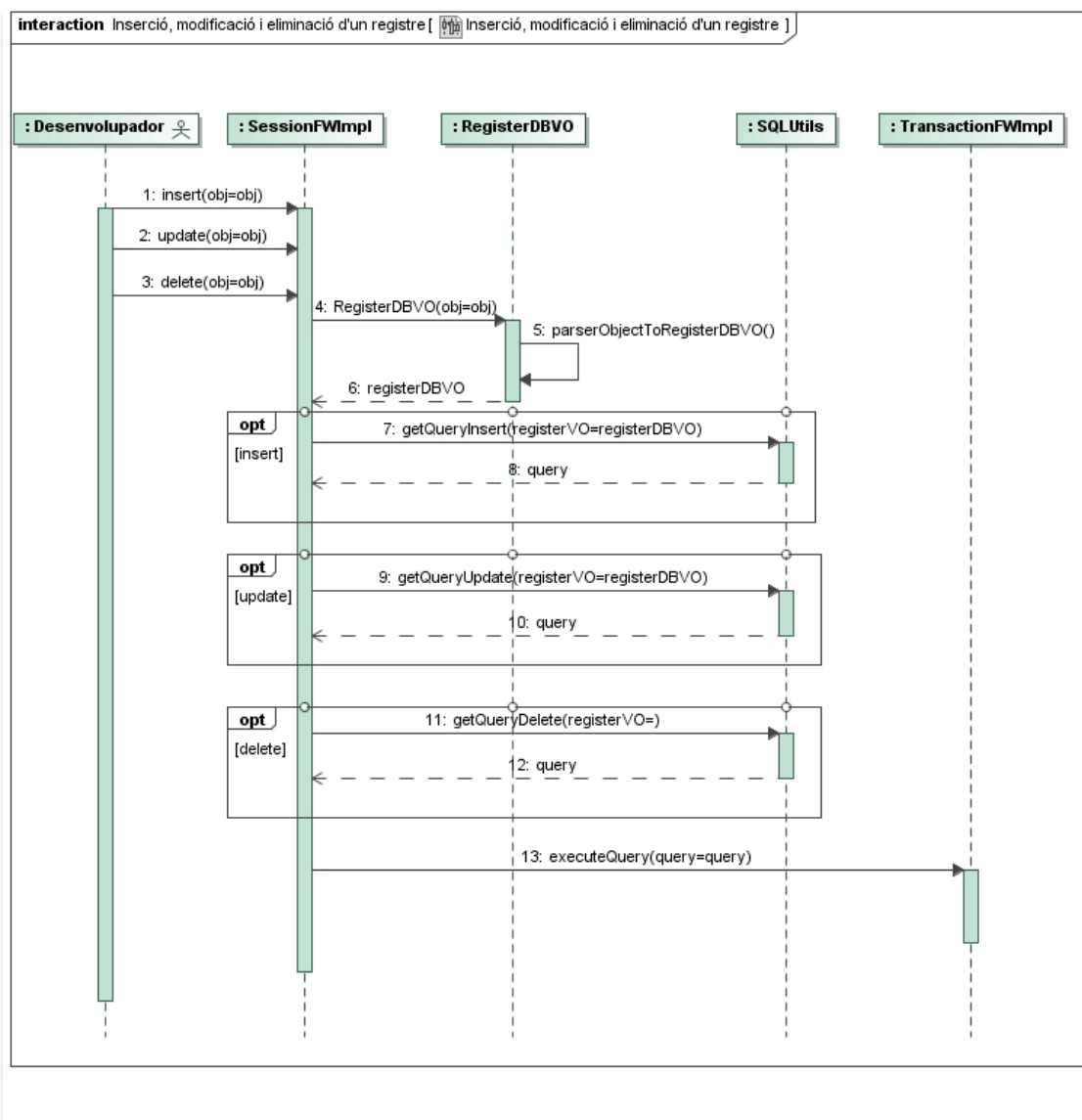


4.4.2 Inserció, modificació i eliminació d'un registre

Abans de fer la crida als mètodes de inserció, modificació i eliminació d'un registre, s'ha d'haver creat l'objecte *SessionFW* tal i com hem comentat en el diagrama anterior.

He realitzat els tres mètodes en el mateix diagrama ja que internament realitzen les mateixes funcionalitats i la única diferència rau en la crida a la classe *SQLUtils* a on es genera les diferents queries que s'executaran a la base de dades.

En els tres mètodes s'ha de passar l'objecte que equival en el bean de la taula de la base de dades que volem inserir, modificar o esborrar. A on a partir d'aquest objecte es crea la classe *RegisterDBVO* equivalent a un registre de la base de dades a on conté tota la informació.

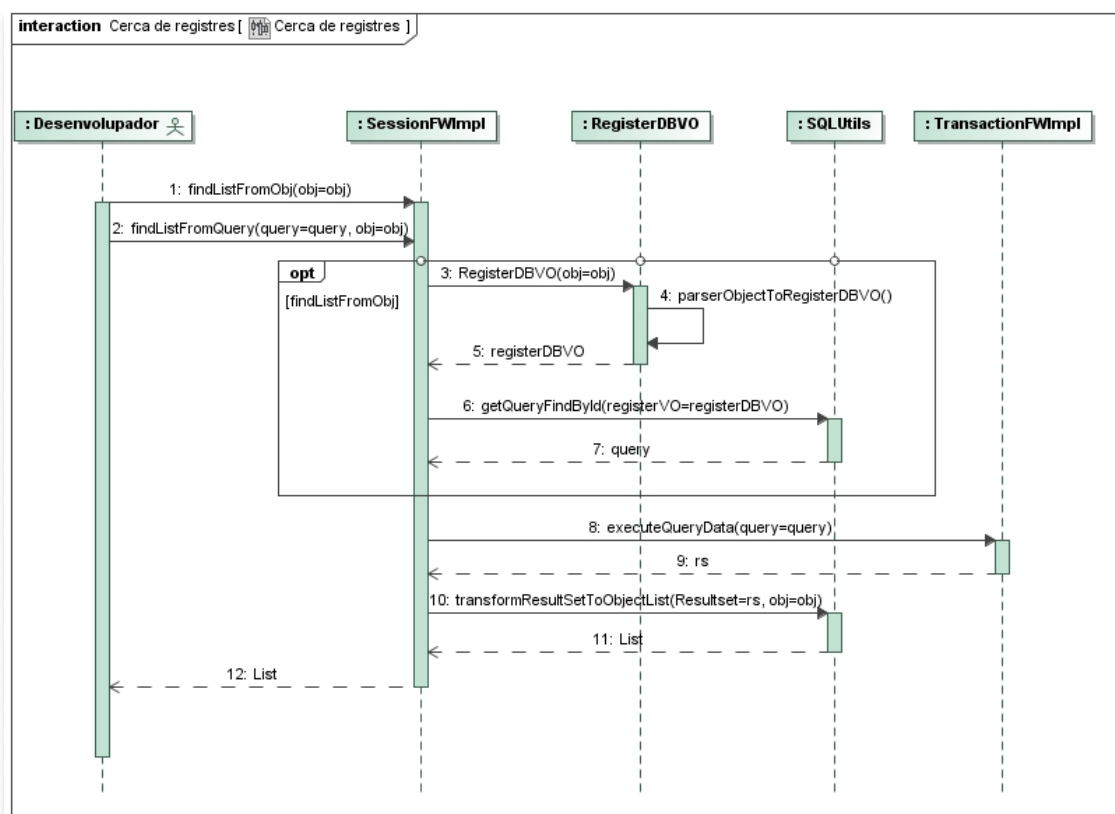


4.4.3 Cerca de registres

Igual que en el punt anterior abans de fer la cerca s'ha d'haver creat l'objecte *SessionFW*.

Hi han dos mètodes de cerca, el *findListFromObj* i el *findListFromQuery*. La diferència és que en el primer es fa el mateix que en el punt anterior, o sigui obtenir l'objecte *RegisterDBVO* i a partir d'aquest es crea la query per fer la consulta a la base de dades a través de la classe *SQLUtils*, i el segon mètode ja passem la query que es vol executar sobre la base de dades, a més de passar l'objecte que retornarà la consulta.

A partir de la query, aquesta s'executa amb el mètode *executeQueryData()* de la classe *TransactionFW* a on ens retornarà el *ResultSet* amb els resultats, i amb aquest farem el parser a través del mètode *transformResultSetToObjectList* de la classe *SQLUtils*.

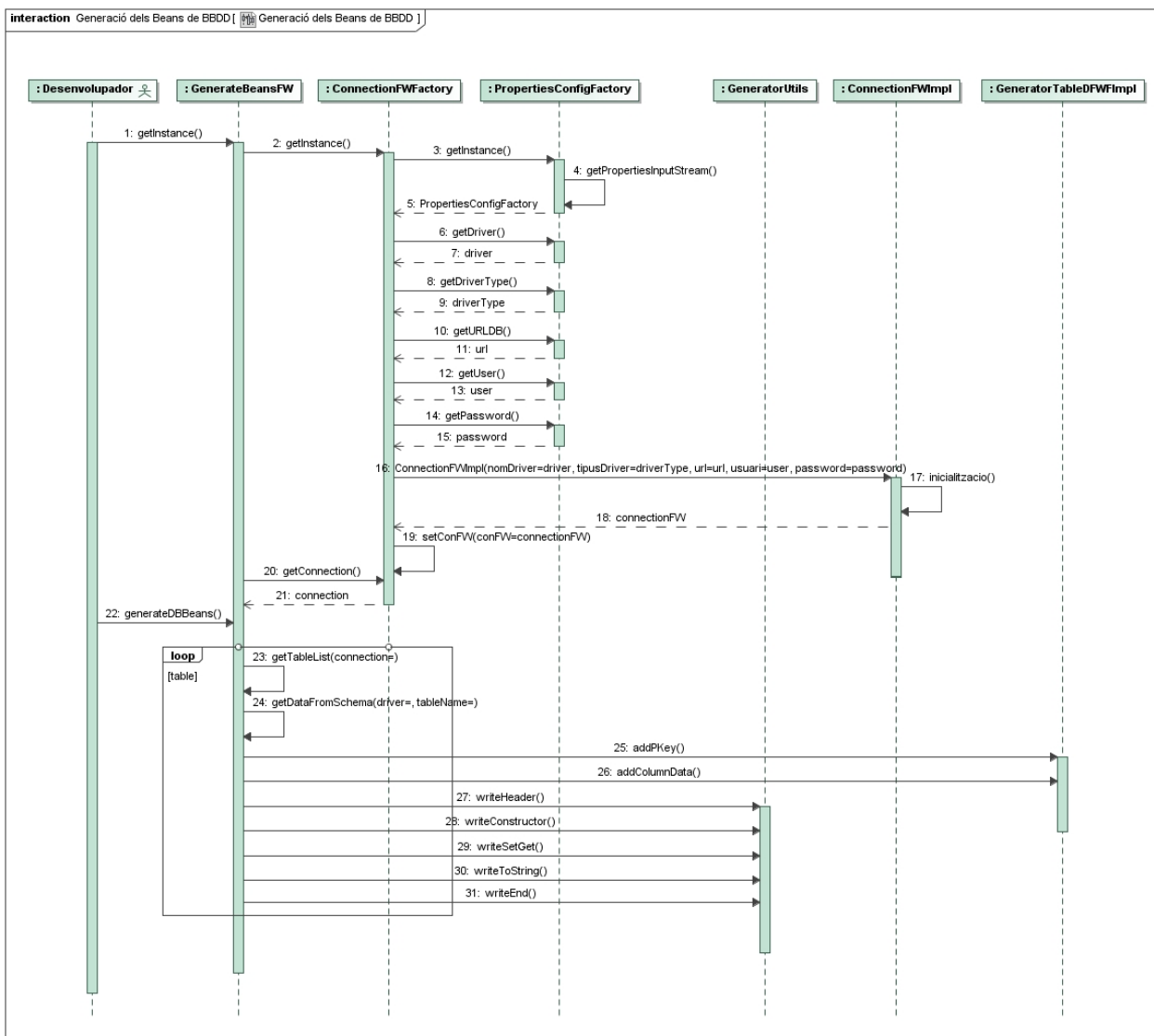


4.4.4 Generació dels beans de base de dades

La generació dels beans de manera automatitzada es realitza a través del mètode *generateDBBeans()* de la classe *GenerateBeansFW*. A partir d'aquesta classe s'obté la connexió a la base de dades amb la classe *ConnectionFWFactory*.

A partir de la connexió obtenim tot el llistat de les taules de base de dades. Per cada taula obtenim tota la informació d'aquesta amb el mètode *getDataFromSchema()*. Aquesta informació la tindrem a la classe *GeneratorTableDBFW* que ens haurà retornat el mètode anterior. Un cop tenim aquest objecte hem d'afegir la informació de la clau primària i de les columnes a través dels mètodes *addPK()* i *addColumnData()*.

Un cop ja tenim tota la informació només ens queda generar tot el fitxer que equivaldrà a una taula de la base de dades. Aquest fitxer tindrà el llistat d'atributs amb els seus *getters* i *setters*, el seu constructor i el mètode *toString* redefinit. Aquesta informació la crearà els mètodes *write* de la classe *GeneratorUtils*.



5 DISSENY DE L'APLICACIÓ D'EXEMPLE

5.1 Introducció

En aquest apartat explicarem el disseny de l'aplicació d'exemple que utilitzarà el framework de persistència realitzat.

L'aplicació d'exemple serà una aplicació web que es desplegarà en un servidor d'aplicacions a on realitzarà les operacions bàsiques de creació, consulta i actualització, d'aquesta manera es prova el framework realitzat.

L'aplicació tindrà els següents requisits:

- Un servidor d'aplicacions JAVA, que en el nostre cas serà un Tomcat 6.0.
- Un servidor de base de dades, que en el nostre cas serà un MySQL 5.1.
- Script SQL amb el model de base de dades.
- Script SQL amb la inserció de dades per defecte en el model de base de dades.
- WAR amb la distribució de l'aplicació.

L'aplicació d'exemple serà una nova companyia aèria de baix cost, que hem anomenat Cat-Wings, a on es permetrà consultar els vols, a més de realitzar, consultar, cancel·lar reserves. Al tractar-se d'una aplicació d'exemple senzilla, dir que no es farà una part d'administració per afegir, modificar i eliminar vols o aeroports.

L'aplicació web utilitzarà una arquitectura client-servidor distribuïda amb tres capes - model, vista, controlador - i per tant amb un manteniment fàcil i flexible.

Un cop comentat l'aplicació passarem a comentar el disseny amb els casos d'us, diagrama de classes, etc.

5.2 Casos d'ús

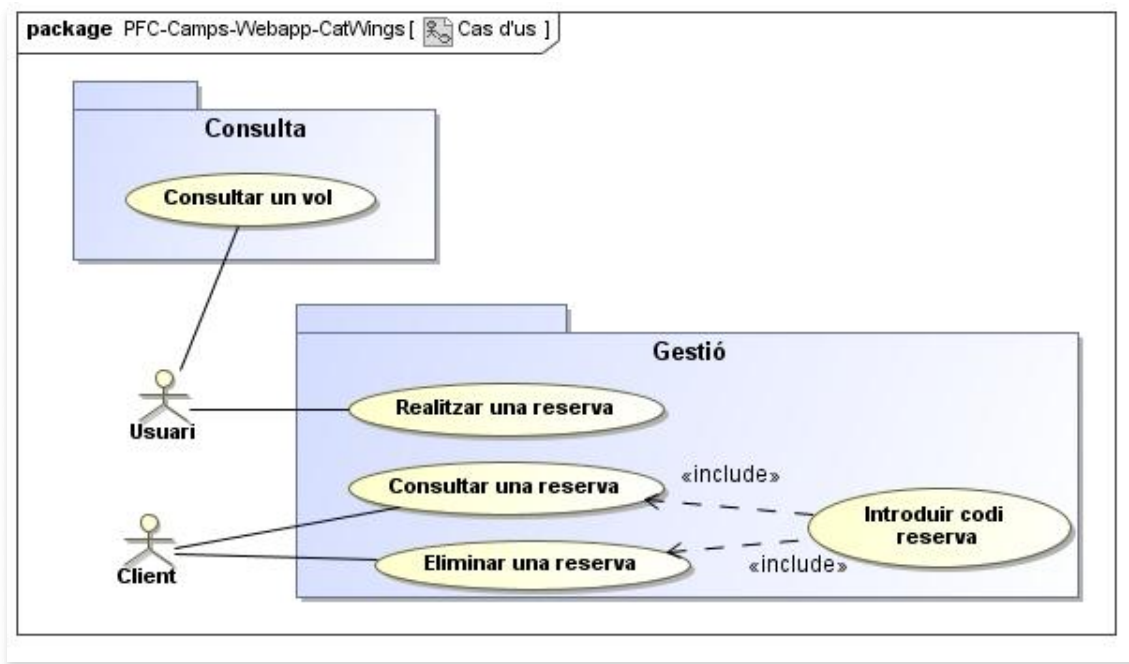
Aquesta aplicació web disposarà dels següents actors:

- **Usuari:** Correspon al perfil de l'usuari que entra a la web a fer una visita i a realitzar consultes de vols o una reserva.
- **Client:** Correspon al perfil de l'usuari que ha realitzat una reserva i la vol consultar, modificar o cancel·lar.

A més de comentar els actors, comentar a l'igual que he fet en el framework, he dividit les funcionalitats en dos paquets descrits de la següent manera:

- **Consulta:** Aquest paquet només contindrà el cas d'ús de realitzar la cerca d'un vol.
- **Gestió:** Aquest paquet gestionarà totes les reserves amb els casos d'ús de crear, consultar i eliminar una reserva.

El diagrama de casos d'ús queda de la següent manera:



5.2.1 Especificació dels casos d'ús

Un cop vist el diagrama passarem a comentar cada un dels casos d'ús:

Nom	Consultar un vol
Descripció	Realitza una recerca de vols disponibles segons l'origen i la destinació de l'usuari, la data de sortida i el número de persones que volen viatjar.
Flux principal	<ol style="list-style-type: none"> 1. El sistema demana a l'usuari que introdueixi les dades de vol que vol buscar, en aquest cas la informació de l'aeroport on vol sortir, i a on vol anar, també la data de sortida, i en el cas que vulgui tornar, la data de tornada, i finalment introdueix el número de passatgers que viatjaran. 2. L'usuari introdueix les dades. Un cop les ha introduït clica en el botó de 'Buscar Vols'. 3. El Sistema valida les dades. 4. El sistema ens mostra les dades de vols disponibles segons la informació que ens ha passat l'usuari.
Flux alternatiu	<ol style="list-style-type: none"> 1. En el pas 3, si les dades són incorrectes, el sistema demana a l'usuari que introdueixi correctament les dades. 2. En el pas 4, sinó hi han dades de vols disponibles el sistema retorna ens mostra un missatge dient-nos que hi han vols disponibles.

Nom	Realitzar una reserva
Descripció	Permet realitzar una reserva.
Flux principal	<ol style="list-style-type: none"> 1. El sistema demana a l'usuari que esculli entre la llista de vols un d'anada i un altre de tornada. 2. L'usuari escull els vols i clica al botó continuar. 3. El sistema mostra la informació detallada dels vols que l'usuari ha escollit. I li demana a l'usuari si està conforme amb el vol escollit. 4. L'usuari si està conforme marca a la casella que està conforme i clica en el botó de 'Confirmar Vol'. 5. El sistema li demana a l'usuari que introdueixi les dades dels passatgers que han de volar i la informació de la persona de contacte. 6. L'usuari introdueix la informació que se li demana. Un cop ha introduït les dades, clica al botó de continuar. 7. El sistema li mostra les dades del vol que l'usuari ha escollit, a més de la informació dels passatgers, i li demana que introdueixi les dades bancàries. 8. L'usuari introdueix les dades bancàries i clica al botó 'Comprar' per reservar el Vol. 9. El Sistema valida les dades bancàries. 10. El sistema crea una reserva del vol que l'usuari havia escollit i li mostra a l'usuari la informació de la reserva que ha realitzat amb el número de reserva que se li ha assignat.
Pre condicions	<ol style="list-style-type: none"> 1. L'usuari ha d'haver buscat un vol.
Flux alternatiu	<ol style="list-style-type: none"> 1. En el pas 3, si l'usuari no està conforme pot clicar al botó de 'Cancel·lar' i el sistema et retornarà a la pàgina de Buscar Vols. 2. En el pas 9, si les dades que ha introduït el client són incorrectes, el sistema torna a demanar a l'usuari que introdueixi correctament les dades. 3. En el pas 10, si quan el sistema valida si queden places disponibles, en el cas que no n'hi hagin, ens envia una pàgina informant-nos que no en queden de disponibles.
Post condicions	<ol style="list-style-type: none"> 1. En el pas 10, si la reserva s'ha creat correctament, doncs tota la informació es guardarà a la Base de dades

Nom	Introduir codi reserva
Descripció	Aquest cas engloba als casos d'ús que vénen a continuació i és el que dona accés a l'actor client a dins del sistema perquè tingui accés als altres casos d'ús.
Flux principal	<ol style="list-style-type: none"> 1. El sistema demana al client que introdueixi el codi de reserva i el mail. 2. L'usuari introdueix el codi i l'email i clica al botó de 'Consultar Reserva'. 3. El sistema valida la informació que el client ha introduït. 4. El sistema mostra la següent pantalla
Flux	<ol style="list-style-type: none"> 1. En el pas 3, si el client introdueix incorrectament les dades, el sistema

alternatiu	informa en el client que les dades introduïdes són incorrectes. 2. En el pas 4, si les dades introduïdes son correctes, però la reserva no existeix, el sistema ens redireccionarà a una pàgina informant-nos de l'error.
-------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

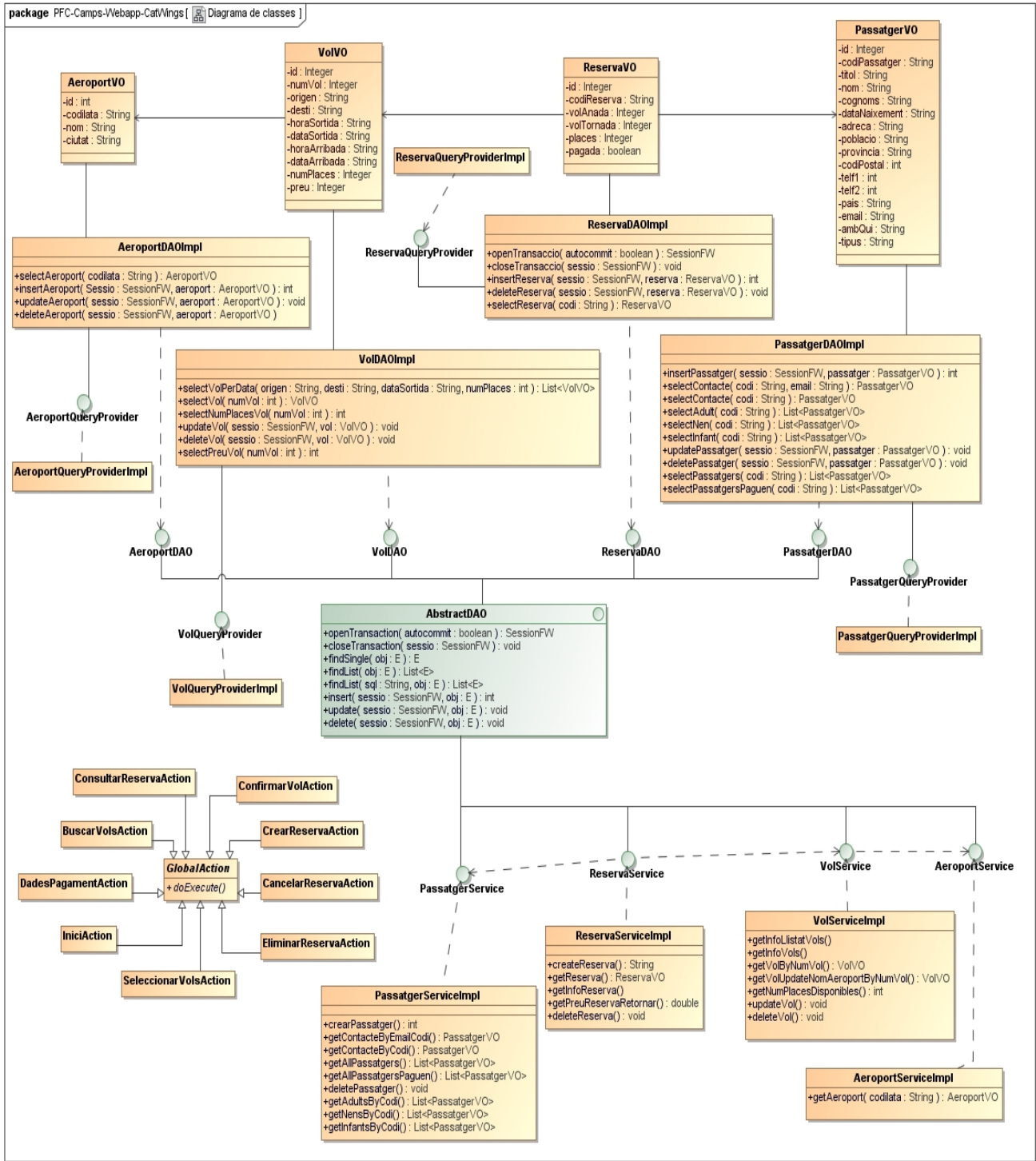
Nom	Consultar una reserva
Descripció	Permet veure en el client la informació de la seva reserva.
Flux principal	1. El sistema li mostra al client tota la informació de la seva reserva. 2. El Client ha d'haver introduït el codi de reserva correctament
Pre condicions	

Nom	Eliminar una reserva
Descripció	Permet en el client eliminar una reserva
Flux principal	1. El Sistema mostra a l'usuari la informació de la reserva. 2. El Client clica el botó de 'Cancel·lar Reserva'. 3. El sistema mostra per pantalla els diners que se li han de retornar en el client. I un camp perquè el client introdueixi el número de compte a on vol que se li retornin els diners. 4. El Client introdueix el seu número de compte. I li demana al sistema clicant el botó 'Eliminar la reserva'. 5. El sistema retorna els diners en el número de compte i elimina la reserva.
Flux alternatiu	1. En el pas 3, si el client no vol eliminar la reserva, ha de clicar en el botó de 'Tornar' per tornar en el pas 1. 2. En el pas 4, el sistema comprova que faltin més de dos hores perquè el vol de sortida surti. En el cas que faltin menys, se li mostra una pantalla al client notificant-li que no pot cancel·lar la reserva.
Pre condicions	1. El Client ha d'haver introduït el codi de reserva correctament.
Post condicions	1. En el pas 5, si tot ha anat correctament, la reserva s'esborra de la base de dades

5.3 Diagrama de classes

Igual que en el framework, l'aplicació d'exemple està basada en els principis SOLID. A més aplicarem el component DAO (Data Acces Object) per separar el negoci de la persistència i millorar la flexibilitat. També utilitzarem el framework d'Spring.

El diagrama de classes queda de la següent manera:



6 CONCLUSIONS

Un cop finalitzat el projecte i veient els resultats finals veig que els objectius que m'havia plantejat al començament de com volia que fos el framework s'han complert.

He realitzat un framework senzill i independent del motor de base de dades, a més permet realitzar les funcionalitats bàsiques CRUD.

El fet de realitzar i dissenyar un framework de persistència m'ha permès ampliar els meus coneixements, tant de la part de persistència, com la dels diferents frameworks que hi han actualment en el mercat.

A més cal remarcar, que he pogut conèixer a fons l'API Reflection de Java, ja que l'he hagut d'utilitzar en diverses parts durant el desenvolupament del framework, a més de l'ús de les Anotacions en Java. I, en la part de l'aplicació web m'ha servit per poder-hi incloure l'Spring, d'aquesta manera m'ha permès millorar els meus coneixements en aquest framework.

Per acabar, com a ampliació m'hauria agradat realitzar proves amb un model de dades més gran i complexa, que contingui uns tipus de dades, per exemple CLOB per veure el seu comportament. Aquest framework que he dissenyat és molt senzill, tot i que, ens pot servir en aplicacions on la complexitat és baixa, seria interessant veure com es comporta amb un entorn una mica més complexa i amb un gran volum de dades.

7 BIBLIOGRAFIA

- Persistència: <https://sites.google.com/site/estrategiasdepersistencia/material-teorico/db-persistencia-y-dbms-motor>
- Persistència: <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/14-Persistencia.pdf>
- Mecanismes de persistència: <http://www.fing.edu.uy/inco/grupos/coal/uploads/Investigaci%F3n/mps07.pdf>
- Persistència: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/818>
- Motors de persistència: http://www.programacion.com/articulo/motores_de_persistencia_231
- ORM: <http://www.visual-paradigm.com/VPGallery/orm/Overview.html>
- JPA: <http://ubuntulife.wordpress.com/2008/10/20/introduccion-a-jpa-java-persistence-api/>
- JPA: http://en.wikipedia.org/wiki/Java_Persistence_API
- JPA: http://openjpa.apache.org/builds/1.2.3/apache-openjpa/docs/jpa_overview_arch.html
- JDO: http://en.wikipedia.org/wiki/Java_Data_Objects
- JDO: http://docs.oracle.com/cd/E24329_01/apirefs.1211/e24396/jdo_overview_arch.html
- JDO: <http://www.oracle.com/technetwork/java/index-jsp-135919.html>
- JDBC: <http://etutorials.org/SQL/Postgresql/Part+II+Programming+with+PostgreSQL/Chapter+13.+Using+PostgreSQL+from+a+Java+Client+Application/JDBC+Architecture+Overview/>
- Oracle TopLink: http://docs.oracle.com/cd/E17904_01/web.1111/b32441/undtl.htm
- Hibernate: http://docs.jboss.org/hibernate/orm/4.2/quickstart/en-US/html_single/
- Hibernate: <http://ca.wikipedia.org/wiki/Hibernate>
- MyBatis: <http://mybatis.github.io/mybatis-3/>

8 ANNEX - INSTAL·LACIÓ I ÚS DEL FRAMEWORK

En els següent annex intentarem facilitar un manual per tal d'ajudar en la instal·lació i configuració de l'entorn de desenvolupament del Framework .

8.1 Prerequisits software

Els prerequisits per poder configurar l'entorn de desenvolupament són els següents:

- S'ha utilitzat Eclipse com a IDE de desenvolupament.
- Java JDK 1.5 o superior (Per compilar s'ha utilitzat la JDK 1.6.0_23).
- Ant 1.6 o superior (S'ha utilitzat la versió 1.8.1).
- Motor de base de dades instal·lat i funcionant (per les proves s'ha utilitzat el MySQL 5.1).
- Els scripts següents de BBDD executats en la base de dades:
 - createTableCatWings.sql
 - insertTaulesCatWings.sql

8.2 Configuració entorn de desenvolupament

Per compilar el projecte primer s'ha de configurar les propietats del fitxer executable (*setenv.bat*) que es troba situat a l'arrel del projecte:

```
${lcamps_uoc_fw}/setenv.bat
```

El fitxer té el següent format:

```
@echo off

set PRJ_HOME=C:\projectes\lcamps_uoc_fw
set ANT_HOME=C:\apache-ant-1.8.1
set JAVA_HOME=C:\Program Files\Java\jdk1.6.0_23

set PATH=%ANT_HOME%\bin;%JAVA_HOME\bin%;%PATH%

echo ----- [PFC - Llorenç Camps - FW] -----
echo PRJ_HOME=%PRJ_HOME%
echo ANT_HOME=%ANT_HOME%
echo JAVA_HOME=%JAVA_HOME%
echo ----- [PFC - Llorenç Camps - FW] -----
goto fin

:fin
cd %PRJ_HOME%
@echo on
```

Els tres paràmetres en negreta són els que s'han de definir, ja que són els que utilitzarà el fitxer *build.xml* utilitzant l'eina ANT per generar el fitxer JAR, que contindrà la llibreria del framework de persistència.

8.3 Estructura del projecte

El detall resumit de l'estructura del projecte es comenta a continuació:

```
${lcamps_uoc_fw}/
```

- build: Conté les classes compilades i el JAR del framework.
- config: Conté els fitxers de configuració del framework.
- dist: Conté el JAR del framework generat.
- lib: Conté les llibreries necessàries per realitzar el projecte.
- sql: Conté els scripts SQL per realitzar les proves del correcte funcionament del projecte.
- src: Conté tot el codi creat per la realització del framework.

8.4 Fitxers de configuració

El fitxer ***uoc-persistenciaFW-config.properties*** es troba situat a:

```

${lcamps_uoc_fw}/config/uoc-persistenciaFW-config.properties

```

El fitxer conté les següents propietats:

Propietat	Descripció
database.connection.driver	Classe del driver de connexió amb la BBDD.
database.connection.tipus.driver	Identificador del driver(mysql, oracle).
database.connection.url	URL d'accés a la BBDD.
database.connection.user	Usuari d'accés a la BBDD.
database.connection.password	Password d'accés a la BBDD.
database.beans.basedir	Directorio base a on es generaran els beans de la BBDD.
database.beans.package	Package java amb el que es crearan els beans.

Exemple de fitxer de configuració:

```

#Paràmetres de connexió a la BBDD
database.connection.driver=com.mysql.jdbc.Driver
database.connection.tipus.driver=mysql
database.connection.url=jdbc:mysql://127.0.0.1/catwings
database.connection.user=root
database.connection.password=root

# Generació dels Beans de BBDD
database.beans.basedir=C:/projectes/lcamps_uoc_fw/src
database.beans.package=test.exemple.beans.catwings

```

El fitxer ***log4j.xml*** es troba situat a:

```

${lcamps_uoc_fw}/config/log4j.xml

```

i correspon al fitxer de configuració del log4j per la generació de logs de l'aplicació.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

<appender name="persistenciaFWFileAppender"
  class="org.apache.log4j.RollingFileAppender">
  <param name="File" value="..//logs/lcamps_uoc_persistenciaFW.log"/>
  <param name="Append" value="true"/>
  <param name="MaxFileSize" value="1024KB"/>
  <param name="MaxBackupIndex" value="10"/>
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d{dd-MM-yyyy} HH:mm:ss}
[persistenciaFW] - %p - %m%n"/>
  </layout>

```

```
</appender>

<category name="edu.uoc.pfc.lcamps.persistenciaFW" additivity="false">
  <level value="debug"/>
  <appender-ref ref="persistenciaFWFileAppender"/>
</category>

<root>
  <level value="fatal"/>
  <appender-ref ref="persistenciaFWFileAppender"/>
</root>

</log4j:configuration>
```

8.5 Compilació del framework

Per compilar el codi s'ha d'executar la comanda d'Ant des de l'arrel del directori `${lcamps_uoc_fw}/`

```
ant clean compile
```

I per generar el fitxer JAR del framework s'ha d'executar la següent comanda:

```
ant clean dist
```

Aquesta última comanda internament també compila.

8.6 Ús del framework

Per provar el framework s'han creat dos test de proves. Un per provar la generació dels beans a partir de la base de dades, i l'altre per provar el correcte funcionament del framework realitzant les principals operacions d'aquest.

Els tests es troben al directori:

```
${lcamps_uoc_fw}/src/test/TestGenerateBeans.java
${lcamps_uoc_fw}/src/test/TestPersistenciaFW.java
```

Alhora d'executar-los és important que es posi el fitxer de configuració comentat anteriorment en el classpath, sinó donarà un error al executar-los.

8.7 JavaDoc

S'ha generat la documentació javadoc del codi font del framework, per tal de fer més comprensible el codi i el seu funcionament.

La documentació es troba a:

```
${lcamps_uoc_fw_javadoc}/index.html
```

9 ANNEX - INSTAL·LACIÓ I ÚS DE L'APLICACIÓ WEB

En els següent annex intentarem facilitar un manual per tal d'ajudar en la instal·lació i configuració de l'entorn de desenvolupament de l'aplicació web d'exemple. A més d'un manual d'usuari detallant el funcionament de l'aplicació web que utilitza el Framework creat pel projecte.

9.1 Prerequisits software

Els prerequisits per poder configurar l'entorn de desenvolupament són els següents:

- S'ha utilitzat Eclipse com a IDE de desenvolupament.
- Java JDK 1.5 o superior (Per compilar s'ha utilitzat la JDK 1.6.0_23).
- Ant 1.6 o superior (S'ha utilitzat la versió 1.8.1).
- Motor de base de dades instal·lat i funcionant (per les proves s'ha utilitzat el MySQL 5.1).
- Servidor d'aplicacions. (Per les proves s'ha utilitzat Apache Tomcat 6.0.37).
- Els scripts següents de BBDD executats en la base de dades:
 - createTableCatWings.sql
 - insertTaulesCatWings.sql

9.2 Configuració entorn de desenvolupament

Per compilar el projecte primer s'ha de configurar les propietats del fitxer executable (*setenv.bat*) que es troba situat a l'arrel del projecte:

```
${lcamps_uoc_webapp_catwings}/setenv.bat
```

El fitxer té el següent format:

```
@echo off

set PRJ_HOME=C:\projectes\lcamps_uoc_webapp_catwings
set ANT_HOME=C:\apache-ant-1.8.1
set JAVA_HOME=C:\Program Files\Java\jdk1.6.0_23
set TOMCAT_HOME=C:\apache-tomcat-6.0.37

set PATH=%ANT_HOME%\bin;%JAVA_HOME\bin%;%PATH%

echo ----- [PFC - Llorenç Camps - CATWINGS] -----
echo PRJ_HOME=%PRJ_HOME%
echo ANT_HOME=%ANT_HOME%
echo JAVA_HOME=%JAVA_HOME%
echo TOMCAT_HOME=%TOMCAT_HOME%
echo ----- [PFC - Llorenç Camps - CATWINGS] -----
goto fin

:fin
cd %PRJ_HOME%
@echo on
```

Els quatre paràmetres en negreta són els que s'han de definir, ja que són els que utilitzarà el fitxer *build.xml* utilitzant l'eina ANT per generar el fitxer WAR, que contindrà els arxius de l'aplicació web.

9.3 Estructura del projecte

El detall resumit de l'estructura del projecte es comenta a continuació:

```
/${lcamps_uoc_webapp_catwings}/
· build: Conté les classes compilades i el WAR de l'aplicació.
· config: Conté els fitxers de configuració de l'aplicació.
· dist: Conté el WAR del framework generat.
· lib: Conté les llibreries necessàries per realitzar el projecte.
· libExt: Conté les llibreries necessàries per la compilació de l'aplicació i
    que ja estan en el servidor d'aplicacions.
· sql: Conté els scripts SQL de l'aplicació d'exemple.
· src: Conté tot el codi creat per la realització de l'aplicació.
· webRoot
    · content
        · css: Conté els fitxers d'estils.
        · html: Conté les pàgines estàtiques.
        · img: Conté les imatges.
        · js: Conté els fitxers javascript.
        · jsp: Contés les JSP.
    · WEB-INF: Conté el fitxers de configuració(struts-config.xml i web.xml)
        · classes: Conté els arxius d'idiomes, de configuració dels logs
            i el de configuració del context d'Spring.
        · lib: Conté les llibreries necessàries per l'execució de
            l'aplicació.
        · tld: Conté els tld.
```

9.4 Fitxers de configuració

El fitxer de configuració ja l'hem comentat en l'apartat dels fitxers de configuració del framework.

9.5 Compilació de l'aplicació

Per compilar el codi i generar el fitxer WAR de l'aplicació s'ha d'executar la comanda d'Ant des de l'arrel del directori `/${lcamps_uoc_webapp_catwings}/`

```
ant clean war
```

Si es vol desplegar automàticament el fitxer WAR generat en el directori d'aplicacions del tomcat s'ha d'executar la següent comanda:

```
ant deploy
```

9.6 Ús de l'aplicació web

Abans de començar a comentar el funcionament de l'aplicació web, s'ha de tenir en compte que el WAR estigui desplegat en el servidor d'aplicacions i que aquest estigui funcionant. A més també validar que el motor de base de dades també estigui arrencat.

La URL d'accés a l'aplicació és la següent:

<http://localhost:8080/lcamps-pfc-webapp-CATWINGS/action/inici>

9.6.1 Pàgina d'inici

La pàgina d'inici ofereix dues opcions, la de reserva d'un vol o la consulta d'una reserva ja realitzada. Tots els altres enllaços accedeixen a contingut estàtic.

Cat-Wings

Inici **Reserva Vol** **Meva Reserva** Destinacions Preguntes sobre el viatge Sobre Cat-Wings Contacte

Busca el teu Vol

Anada i Tornada Anada

Selecció Aeroport:

Destinació:

Data Sortida:

Data Tornada:

NºPassatgers:

1 Adults

0 Nens (2-12 anys)

0 Infants (1-23 mesos)

Buscar Vols

CALL CENTER
De 8h a 22h de dill. a div., i de 10h a 18h de dis. a Diu.

RESERVA I COMPRA BITLLETS 987654321
ATENCIÓ AL CLIENT 987654321
INCIDÈNCIES WEB 987654321

OFERTES

Oferta ANDORRA
Des de 40€ trajecte
Tot inclòs

OFERTA MAUI
Des de 200€ trajecte
Tot inclòs

+ OFERTES

✦ Barcelona - Mallorca	Des de 20€
✦ Barcelona - Madrid	Des de 20€
✦ Barcelona - Girona	Des de 10€
✦ Barcelona - Eivissa	Des de 30€
✦ Barcelona - Amsterdam	Des de 20€
✦ Barcelona - París	Des de 20€
✦ Barcelona - Londres	Des de 20€
✦ Girona - Còrcega	Des de 25€
✦ Girona - Sardenya	Des de 35€
✦ Girona - Milà	Des de 40€
✦ Girona - Eivissa	Des de 25€
✦ Mallorca - Amsterdam	Des de 30€
✦ Mallorca - Londres	Des de 30€
✦ Mallorca - París	Des de 15€

*Preu per trajecte. Taxes i càrrecs inclosos.

CAT-WINGS Notícies

- ▶ Avis Oficial: Noves mesures de seguretat en els aeroports europeus.
- ▶ Cat-Wings augmenta el doble de la flota d'avions.
- ▶ Cat-Wings posa a la venda 10 rutes noves. 5 a Espanya i 5 a Europa.
- ▶ Cat-Wings augmenta la mitjana d'ocupació dels avions a un 85%.
- ▶ Llorenç Camps nomenat director general de Cat-Wings.

© 2014 Cat-Wings Airlines S.A. Política de Privacitat | Condicions de transport
Desenvolupada per Llorenç Camps

9.6.2 Realitzar reserva

Busca el teu Vol

Anada i Tornada Anada

Selecció Aeroport:

Destinació:

Data Sortida:

Data Tornada:

NºPassatgers:

1 Adults

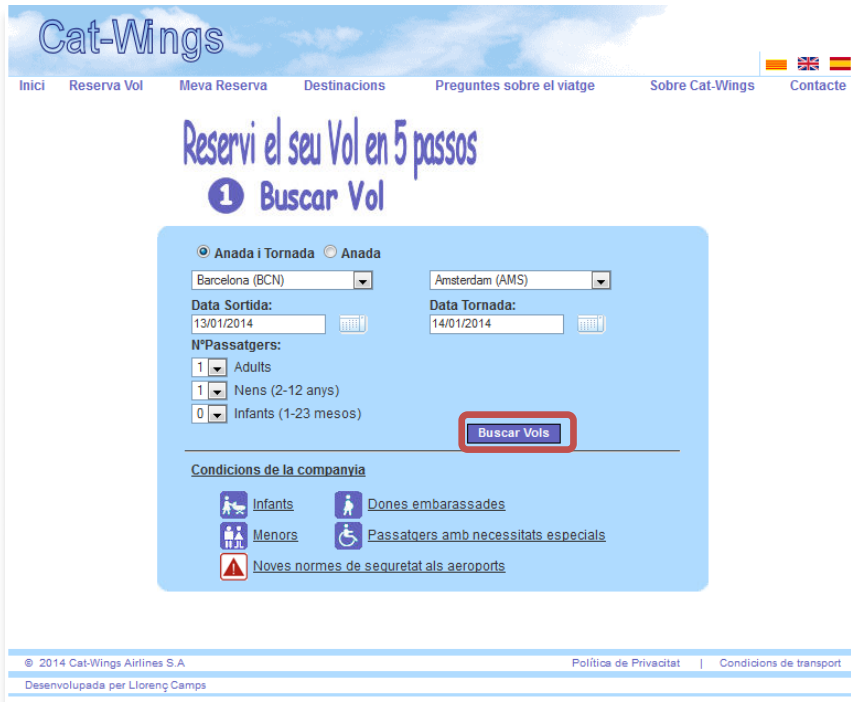
0 Nens (2-12 anys)

0 Infants (1-23 mesos)

Buscar Vols

La opció de reserva d'un vol es pot iniciar tant des de la pàgina d'inici a través del formulari a on es demana les dades de l'aeroport de sortida i de destinació, a més de les dates i el número de passatgers, tal com podem veure en la imatge de l'esquerra, com des de la pàgina de "Reserva vol", tal com veiem en la imatge de la següent pàgina, on conté exactament el mateix formulari.

Per iniciar la reserva l'usuari emplena el formulari i prem al botó de 'Buscar Vols' per realitzar la consulta de vols disponibles amb les dades introduïdes.



Si l'usuari no introdueix dades i prem a buscar vol, el sistema li mostrarà un missatge avisant-lo de quines dades li falten per realitzar la cerca.

Un cop s'han introduït les dades el sistema realitzarà la cerca retornant a la següent interfície amb la llista de vols disponibles segons les dades introduïdes, tal com veiem a continuació:



En cas que no hi hagin vols disponibles es mostra a l'usuari la següent pantalla:

En aquest punt, l'usuari pot tornar enrere prement al botó de 'Nova Cerca' o anar prement els enllaços de 'Dia Abans' i 'Dia Després' per anar realitzant consultes per si troba algun vol disponible en les dates properes que ha introduït en el formulari.

La informació que retorna depèn de si l'usuari vol bitllet de tornada o no, en cas que no el vulgui el sistema només mostrarà la informació del vol d'anada.

A continuació l'usuari ha de seleccionar els vols que l'interessen i prémer al botó de 'Seleccionar Vol'. En cas que premi el botó i no hagi seleccionat cap vol, el sistema mostrarà un missatge informatiu del que ha de fer.

En cas que tot sigui correcte el sistema ens retornarà la següent interfície amb les dades dels vols que hem seleccionat i els respectius imports desglossats amb el import final.

VOL D'ANADA		VOL DE TORNADA	
Anada: Barcelona (BCN) - Amsterdam (AMS)		Tornada: Amsterdam (AMS) - Barcelona (BCN)	
Nº Vol: 1401		Nº Vol: 520	
Sortida: 13/01/2014 a les 20:00		Sortida: 14/01/2014 a les 18:00	
Arribada: 13/01/2014 a les 20:50		Arribada: 14/01/2014 a les 18:50	

IMPORT			
<u>Anada:</u>			
2 Adults/Nens x 35 EUR	70 EUR	<u>Tornada:</u>	
Impostos i taxes incloses.	70 EUR	2 Adults/Nens x 55 EUR	110 EUR
		Impostos i taxes incloses.	110 EUR
Import Total del Vol:	180 EUR		

Important
 He llegit i accepto la reglamentació de les tarifes així com les condicions i restriccions associades al viatge.
 (Marqui amb una creu la casella per continuar, clicki [aquí](#) per llegir les condicions generals.)

En aquest punt, si hi ha alguna informació que no és correcte o es vol buscar un altre vol, es pot prémer al botó cancel·lar i es retornarà a la pàgina de 'Reserva de vol'.

I en cas que tot sigui correcte, l'usuari prem el botó 'Confirmar Vol' per passar a la següent interfície. Però abans d'accedir al següent pas, es valida que el client s'hagi llegit la reglamentació de les tarifes i hagi marcat confirmant que s'ho havia llegit i estava conforme. En aquest apartat hi ha un enllaç que obre la pantalla d'informació de les tarifes, tal i com es mostra a continuació:



Un cop l'usuari està conforme amb les condicions de les tarifes i ha premut el botó de 'Confirmar el Vol', el sistema ens retornarà la següent interfície amb la part d'introducció d'informació dels passatgers.

4 Introdueix informació dels passatgers

Important: Introdueix per cada passatger el primer nom i els cognoms tal i com estan escrits en el passaport. No utilitzi noms de pila ni inicials sino coincideixen amb la informació del passaport.

PASSATGERS

Adults:

Titul Nom Cognoms Data Naixement

1. - -

Nens:

Data Naixement Nom Cognoms

1. - -

DADES DEL CONTACTE

Titul Nom Cognoms Adreça

Població Província C.Postal País

Telèfon (*) Altre Telèfon (*) Email

(*) El número de telèfon ha d'incloure el prefix internacional (p.e: 0034)

He llegit i accepto les condicions de privacitat. (+info)

Continuar >>

En aquesta interfície el sistema ens demana que introduïm la informació que pertany als passatgers que viatgen, tant si son nens, com infants, encara que aquests últims no tinguin plaça en l'avió. També hem d'introduir la informació sobre la persona de contacte.

Falta informació sobre el passatger.
Siusplau, introduueixi les següent dades:

Títol per Adult 1
 Nom Adult 1
 Cognoms Adult 1
 Dia Adult 1
 Mes Adult 1
 Any Adult 1
 Dia Nen 1
 Mes Nen 1
 Any Nen 1
 Nom Nen 1
 Cognoms Nen 1

En cas que l'usuari premi a 'Continuar' sense haver introduït la informació, el sistema mostrarà un missatge informatiu detallant quins camps queden pendents d'emplenar-se.

Un cop s'ha introduït tota la informació dels passatgers i de les dades del contacte, s'han de llegir i acceptar les condicions de privacitat, si l'usuari està conforme ha de marcar el camp com a conforme. Per poder accedir a aquestes condicions hi ha un enllaç al costat, que si hi prems s'obre la finestra amb les condicions perquè es puguin llegir, tal com es veu en la següent

imatge.

Cat-Wings

Condicions generals d'ús del lloc web
Cat-Wings Airlines, S.A.

www.catwings.com

Aquestes condicions generals d'ús regulen l'accés i utilització del lloc web www.catwings.com (el "Lloc Web"), el contingut del qual té un caràcter exclusivament informatiu. El Lloc Web ha estat creat per Catwings Airlines, S.A. amb domicili social a Vidreres (Girona), C/Sant Jordi,35, inscrita al Registre Mercantil de Girona, Foli 196, Tom 36403, Full B-279224 i CIF A- 63422141 ("Catwings"). L'accés i navegació d'un usuari pel Lloc Web implica el coneixement i l'acceptació sense reserves dels següents termes i condicions (les "Condicions Generals"):

- 1. Termes i condicions

Un cop ha acceptat les condicions ja pot prémer 'Continuar' per seguir endavant amb la reserva. Fet això el sistema ens mostrarà la següent interfície que conté les dades dels vols, dels passatgers i del contacte. A més també hi ha un formulari perquè el client introdueixi les dades de cobrament per tal que se li sigui cobrat immediatament el import de la reserva.

5 Pagament

VOL D'ANADA	VOL DE TORNADA
Anada: Barcelona (BCN) - Amsterdam (AMS) Nº Vol: 1401 Sortida: 13/01/2014 a les 20:00 Arribada: 13/01/2014 a les 20:50	Tornada: Amsterdam (AMS) - Barcelona (BCN) Nº Vol: 520 Sortida: 14/01/2014 a les 18:00 Arribada: 14/01/2014 a les 18:50
INFORMACIÓ DELS PASSATGERS	INFORMACIÓ DEL CONTACTE
Nº Passatgers: 2 1 Llorenç Camps 1 Jordi Camps	Nom: Llorenç Camps Adreça: Ramon llull C.Postal: 17001 Població: Girona (Girona) País: Catalunya
PAGAMENT PER TARJA DE CRÈDIT	
Tipus de Tarjeta Número de tarjeta Nom del titular de la Tarjeta CVV (Què és CVC?) Data Expiració Import pagament de la tarjeta	Seleccioni el tipus de crèdit ▾ <input type="text"/> <input type="text"/> <input type="text"/> 01 ▾ 2014 ▾ 180 EUR

Un cop l'usuari ha introduït aquesta informació i està conforme prem al botó de 'Comprar', per tal de realitzar definitivament la reserva.

En cas que la reserva s'hagi creat correctament el sistema ens mostrarà la següent interfície.

The screenshot shows a confirmation page with a large blue '6' in a circle and the word 'Confirmació' in blue. Below this, the text 'La Reserva s'ha realitzat Correctament.' is displayed in red. A red-bordered button with a printer icon and the text 'imprimir' is located in the top right corner. The page is divided into four sections by light blue headers: 'VOL D'ANADA', 'VOL DE TORNADA', 'INFORMACIÓ DELS PASSATGERS', and 'INFORMACIÓ DEL CONTACTE'. The 'VOL D'ANADA' section lists: Anada: Barcelona (BCN) - Amsterdam (AMS), N° Vol: 1401, Sortida: 13/01/2014 a les 20:00, Arribada: 13/01/2014 a les 20:50. The 'VOL DE TORNADA' section lists: Tornada: Amsterdam (AMS) - Barcelona (BCN), N° Vol: 520, Sortida: 14/01/2014 a les 18:00, Arribada: 14/01/2014 a les 18:50. The 'INFORMACIÓ DELS PASSATGERS' section lists: N° Passatgers: 2, 1 Llorenç Camps, 1 Jordi Camps. The 'INFORMACIÓ DEL CONTACTE' section lists: Nom: Llorenç Camps, Adreça: Ramon Llull, C.Postal: 17001, Població: Girona (Girona), País: Catalunya. Below these sections, a light blue header reads 'NUMERO DE LA RESERVA', followed by the reservation number 'L1C69G' in large red text. A red-bordered button with a warning icon and the text 'Imprimeix aquesta pàgina i presenta-la en el mostrador d'embarcament el dia del vol.' is located below the reservation number. At the bottom center, there is a blue-bordered button with the text 'Continuar'.

VOL D'ANADA	VOL DE TORNADA
Anada: Barcelona (BCN) - Amsterdam (AMS)	Tornada: Amsterdam (AMS) - Barcelona (BCN)
N° Vol: 1401	N° Vol: 520
Sortida: 13/01/2014 a les 20:00	Sortida: 14/01/2014 a les 18:00
Arribada: 13/01/2014 a les 20:50	Arribada: 14/01/2014 a les 18:50

INFORMACIÓ DELS PASSATGERS	INFORMACIÓ DEL CONTACTE
N° Passatgers: 2	Nom: Llorenç Camps
1 Llorenç Camps	Adreça: Ramon Llull
1 Jordi Camps	C.Postal: 17001
	Població: Girona (Girona)
	País: Catalunya

NUMERO DE LA RESERVA

L1C69G

(!) Imprimeix aquesta pàgina i presenta-la en el mostrador d'embarcament el dia del vol.

Continuar

Tal i com podem veure en la imatge, el sistema ens retorna tota la informació del vol del qual hem fet una reserva, amb les dades dels passatgers i la informació del contacte. I també ens informa del número de la reserva. Amb aquest número podrem accedir a la part de la consulta de la reserva, per si l'hem de consultar o cancel·lar-la. En aquesta pàgina també hi ha la opció de poder imprimir la pàgina amb tota la informació.

El botó de 'Continuar' torna a la pàgina d'inici.

En cas que hi hagi un error alhora de realitzar el sistema ens mostrarà la següent interfície.

The screenshot shows an error page with a large blue circle and the word 'Error' in blue. Below this, the text 'Error en l'aplicació.' is displayed in blue. Underneath, a smaller blue text reads: 'S'ha produït un error a la hora de fer la reserva.Torni a intentar realitzar la reserva.' At the bottom center, there is a blue-bordered button with the text 'Continuar'.

Error

Error en l'aplicació.

S'ha produït un error a la hora de fer la reserva.Torni a intentar realitzar la reserva.

Continuar

9.6.3 Consultar reserva

La pàgina de consultar la reserva d'un vol és la següent:



The screenshot shows the 'Consultar Reserva' page on the Cat-Wings website. The page has a blue header with the 'Cat-Wings' logo and navigation links: 'Inici', 'Reserva Vol', 'Meva Reserva', 'Destinacions', 'Preguntes sobre el viatge', 'Sobre Cat-Wings', and 'Contacte'. There are also flags for Spain, the UK, and Spain. The main heading is 'Consultar Reserva'. Below it, a text block states: 'Aqui, pot durant les 24h del dia, consultar de manera ràpida i fàcil la informació de la seva reserva per poder-la consultar, i modificar la informació, si ho desitja. Introdueix les dades en els següents camps, per tal d'accedir a la seva reserva.' Below this is a light blue form with two input fields: 'Número de Reserva:' and 'Email:'. A red box highlights the 'Consultar Reserva' button. At the bottom, there is a footer with copyright information: '© 2014 Cat-Wings Airlines S.A.', 'Desenvolupada per Llorenç Camps', 'Política de Privacitat', and 'Condicions de transport'.

En aquesta pantalla el sistema només ens demana que introduïm el número de la reserva, i l'email de la persona de contacte. Un cop s'han introduït les dades premem el botó de 'Consultar reserva' per iniciar la consulta de la reserva.

En cas que les dades introduïdes siguin errònies el sistema ens mostrarà el següent missatge d'error:



The screenshot shows an error message on the Cat-Wings website. It features a blue circle icon with the word 'Error' next to it. The main heading is 'Error en l'aplicació.'. Below it, a text block states: 'No hi ha cap reserva amb les dades que ha introduït. Validi les dades i torna-ho a provar.' Below this is a blue button labeled 'Continuar'.

En cas d'introduir correctament les dades el sistema ens mostrarà la interfície amb tota la informació de la reserva tal i com veiem en la següent imatge.

Informació de la reserva

NºReserva: L1C69G



VOL D'ANADA	VOL DE TORNADA
Anada: Barcelona (BCN) - Amsterdam (AMS) Nº Vol: 1401 Sortida: 13/01/2014 a les 20:00 Arribada: 13/01/2014 a les 20:50	Tornada: Amsterdam (AMS) - Barcelona (BCN) Nº Vol: 520 Sortida: 14/01/2014 a les 18:00 Arribada: 14/01/2014 a les 18:50
INFORMACIÓ DELS PASSATGERS	INFORMACIÓ DEL CONTACTE
Nº Passatgers: 2 0 Llorenç Camps 1 Jordi Camps	Nom: Llorenç Camps Adreça: Ramon Llull C.Postal: 17001 Població: Girona (Girona) País: Catalunya

[Cancel·lar Reserva](#) [Sortir](#)

En aquesta interfície es pot tornar a imprimir la informació o cancel·lar la reserva.

El botó de sortir torna a la pàgina de consultar una reserva.

En cas que l'usuari premi al botó de 'Cancel·lar Reserva', el sistema ens mostra la següent interfície a on ens permetrà eliminar la reserva.

Cancel·lar Reserva

Introdueix el número de compte a on vol que li retornem els diners.
Recordeu que es descomptarà un 30% del cost de la reserva.

Diners a Retornar:	126.0
NºCompte:	<input type="text"/>

[Tornar](#) [Eliminar Reserva](#)

En aquesta pantalla el sistema ens mostra per pantalla la quantitat de diners que ens ha de retornar. La dada que mostra ja li ha fet el descompte del 30% de penalització per cancel·lar una reserva. I després també ens mostra un camp perquè el client introdueixi el número de compte a on vol que se li retornin els diners. Un cop ha introduït el número de compte ha de prémer el botó 'Eliminar Reserva' si vol eliminar-la, i en cas que no la vulgui eliminar ha de prémer en el botó 'Tornar', i d'aquesta manera tornarà a la pàgina de informació de la reserva.

Si l'usuari prem el botó 'Eliminar reserva', s'esborrarà la reserva del sistema i mostrarà a l'usuari la següent interfície:

Cancel·lar Reserva

S'ha eliminat correctament la Reserva.

Els diners han estat retornats al número de compte donat.

[Continuar](#)

En cas que hi hagi hagut algun error eliminant la reserva el sistema mostrarà el següent error:

Error


Error en l'aplicació.

No s'ha pogut eliminar la reserva. Comprovi les condicions per poder eliminar una reserva.


[Continuar](#)

9.6.4 Contingut estàtic

Les pàgines amb contingut estàtic són les següents:



- [Inici](#)
- [Reserva Vol](#)
- [Meva Reserva](#)
- [Destinacions](#)**
- [Preguntes sobre el viatge](#)
- [Sobre Cat-Wings](#)
- [Contacte](#)



Ciutats Cat-Wings

Amsterdam (AMS): Aeroport de Schiphol	Andorra (LEU): Aeroport de la Seu d'Urgell
Barcelona (BCN): Aeroport de Barcelona - El Prat	Bilbao (BIO): Aeroport de Bilbao
Còrcega (BIA): Aeroport de Bàstia	Eivissa (IBZ): Aeroport d'Eivissa
Ginebra (GVA): Aeroport internacional de Ginebra	Girona (GRO): Aeroport Costa Brava
Lisboa (LIS): Aeroport de Lisboa	Londres (STN): Aeroport de Stansted
Madrid (MAD): Aeroport Internacional de Barajas	Maui (MAU): Aeroport de Maui
Milà (MXP): Aeroport de Milà - Malpensa	Munic (MUC): Aeroport internacional de Munic - F.J. Strauss
Palma de Mallorca (PMI): Aeroport de Son Sant Joan	París (CDG): Aeroport de París, Charles de Gaulle
Praga (PRG): Aeroport internacional de Praga - Ruzyně	Roma (FCO): Aeroport de Roma - Fiumicino - Leonardo da Vinci
Sardenya (AHO): Aeroport de l'Alguer	València (VLC): Aeroport de València - Manises

© 2014 Cat-Wings Airlines S.A. [Política de Privacitat](#) | [Condicions de transport](#)
Desenvolupada per Llorenç Camps

Cat-Wings

Inici Reserva Vol Meva Reserva Destinacions **Preguntes sobre el viatge** Sobre Cat-Wings Contacte

PREGUNTES SOBRE EL VIATGE

Reserva de Vols

- Com puc reservar?
- Puc Canviar la meva reserva?
- Puc cancel·lar el meu bitllet?
- Quan s'ha de fer el pagament?
- Com puc fer el pagament?
- Puc canviar el titular del bitllet?
- Els nens han d portar documentació?
- Puc transportar Animals?
- Son segures les compres?

Com puc reservar?

Es poden fer reserves de vols:

- A la nostra pàgina web, a lapartat Reserva vol. Només has de seleccionar el trajecte/s desitjat/s , emplenar les dades i fer el pagament còmodament amb targeta de crèdit. En finalitzar la reserva, enviarem una confirmació de vol amb un número de localitzador que sha de presentar a laeroport amb el DNI o el passaport.
- Per telèfon, trucant al Centre de reserva i venda de bitllets. Aquesta via té un cost de tramitació de 7 euros per trajecte i passatger.

Telèfon de reserva i venda de bitllets: 987654321

Puc Canviar la meva reserva?

Per suposat, amb Cat-Wings no perdeu mai els diners dels bitllets, perquè la flexibilitat és total. Els nostres bitllets es poden cancel·lar fins a 2 hores abans de la sortida. Podeu canviar la informació dels passatgers.

El cost de la cancel·lació és d'un 30% del cost del trajecte i persona.

Puc cancel·lar el meu bitllet?

Els nostres bitllets es poden cancel·lar fins a 2 hores abans de la sortida, per tal que no perdeu mai els diners.

Quan s'ha de fer el pagament?

El pagament sha de fer en el mateix moment de la reserva. A causa de les condicions econòmiques ofertes per Cat-wings, els vols shan de pagar de forma immediata. No es faran reserves de places sense pagament previ.

Com puc fer el pagament?

El pagament del bitllets es pot fer exclusivament amb targetes de pagament acceptades per Cat-Wings (Visa, Mastercard).

Cat-Wings

Inici Reserva Vol Meva Reserva Destinacions Pregunta sobre el viatge **Sobre Cat-Wings** Contacte

SOBRE CAT-WINGS

Informació de l'empresa

- Història de l'empresa
- La nostra filosofia
- Els nostres avions
- Vol treballar amb nosaltres?

Història de l'empresa

Els orígens de Cat-Wings Airlines es remunten a mitjans de lany 2006, quan en Llorenç Camps va constituir una societat promotora que va finançar els estudis de mercat i la viabilitat econòmica necessàries per al projecte.

A partir daquell moment, tots els esforços es varen destinar a assolir IAOC -airline operating certificate- per tal desdevenir oficialment una companyia aèria.

Cat-Wings va començar a vendre el 20 de desembre de lany 2006, amb una campanya de vols a 20 euros; es van vendre més de 50.000 bitllets en 15 dies.

El nostre primer vol comercial (Girona - Bilbao) va enlairar-se l1 de gener del 2007. La flota original consistia en 6 A320 nous de trinca, i les rutes diàries oferien vols des de Barcelona a destinacions com ara París, Palma i Eivissa.

La nostra filosofia

Volem crear una companyia aèria on volar esdevingués veritablement un plaer, on pagar menys no suposés reduir l'estàndard de servei o comoditat; volem una companyia aèria que fos directa, honesta i fàcil d'utilitzar. Una companyia aèria que, si us diu que hi ha bitllets realment barats, és perquè els té. Volem una companyia aèria amb gent amable i un tracte de tu a tu, avions nous, aeroports principals, sense retards i sense overbooking (sobreserva). Així de senzill.

Us oferim els millors preus perquè som una companyia aèria amb una gestió diferent i disposem de la millor tecnologia.

Amb nosaltres la puntualitat està garantida: la nostra mitjana de puntualitat arriba al 95%, i supera la mitjana del sector.

Volem des dels principals aeroports de cada ciutat: París - Charles de Gaulle; Barcelona - El Prat; Madrid Barajas; Milà Malpensa; Roma Fiumicino; etc.

Els nostres avions

La flota original de Cat-Wings es composava de 6 Airbus A-320 nous de trinca. La companyia aèria de nova generació va començar a vendre bitllets el 20 de desembre de 2006.



9.7 JavaDoc

S'ha generat la documentació javadoc del codi font de l'aplicació web, per tal de fer més comprensible el codi i el seu funcionament.

La documentació es troba a:

```
/${lcamps_uoc_webapp_catwings_javadoc}/index.html
```