

# **Web taller de servei ràpid**

Desenvolupament d'una aplicació web amb tecnologies J2EE

Autor: **Joan Plademunt Albó**  
ETIS

Consultor: **Xavier Escudero Sabadell**

14 de gener de 2009

Dedico aquest projecte a totes les persones que m'han donat suport i consells per portar a fi aquest treball.

**(Creative Commons)**

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

## RESUM

El projecte és una web per a un taller mecànic dels anomenats “de servei ràpid”, amb tecnologies J2EE. Els clients del taller poden demanar dia i hora per fer una reparació al seu cotxe. I el taller, en el seu cas el cap de taller o persona autoritzada, pot llistar les reparacions del dia.

La web, mostra informació dels serveis i ofertes que ofereix el taller mecànic. Amb un apartat per donar-se d'alta, i així fent un login es poden demanar serveis.

Per implementar la web s'ha escollit el framework SPRING que controla la lògica de negoci i que integra hibernate per la persistència i struts2 per la presentació

HIBERNATE utilitza DAOs, que juntament amb els POJOs accedim a la base de dades MySQL. STRUTS2 encamina les accions sol·licitades a partir dels JSPs, en que interacciona l'usuari.

APACHE TOMCAT conté tot el projecte web i s'encarregà d'executar els diferents JSPs segons l'opció sol·licitada per l'usuari.

La interfície és amigable i fàcil d'utilitzar per qualsevol persona. S'ha tractat de guiar l'usuari i informant-lo en tot moment, si per exemple ha introduït una dada malament surt un missatge indicant-ho.

---

# INDEX

---

<b>CAPÍTOL 1.INTRODUCCIÓ .....</b>	<b>1</b>
1.1 Justificació del TFC .....	1
1.2 Objectius del TFC .....	2
1.3 Enfocament i mètode seguit .....	3
1.4 Planificació del projecte .....	4
1.5 Productes obtinguts .....	5
1.6 Resum resta de capítols .....	6
<b>CAPÍTOL 2. ANÀLISI .....</b>	<b>8</b>
2.1 Estudi del funcionament .....	8
2.2 Actors i casos d'ús .....	9
2.3 Les pantalles .....	12
<b>CAPÍTOL 3. DISSENY .....</b>	<b>15</b>
3.1 Capa persistència .....	15
3.2 Capa Servei .....	16
3.3 Capa Presentació .....	17
3.4 Diagrama de classes .....	18
3.5 Base de dades .....	19
<b>CAPÍTOL 4. IMPLEMENTACIÓ .....</b>	<b>21</b>
4.1 Part de Struts2 .....	21
4.2 Part de Hibernate .....	23
4.3 Part de Spring .....	25
4.4 Fitxer WAR .....	26
4.5 Base de dades .....	27
<b>CAPÍTOL 5. INSTAL·LACIÓ .....</b>	<b>28</b>
5.1 Entorn necessari .....	28
5.3 Desplegar i executar l'aplicació .....	29
<b>CAPÍTOL 6.PROVES .....</b>	<b>30</b>
6.1 Rol client .....	30
6.2 Rol cap de taller .....	32
<b>CAPÍTOL 7. VALORACIÓ ECONÒMICA .....</b>	<b>34</b>
<b>CAPÍTOL 8. CONCLUSIONS .....</b>	<b>36</b>
<b>GLOSSARI .....</b>	<b>37</b>
<b>BIBLIOGRAFIA .....</b>	<b>39</b>
<b>ANNEXOS .....</b>	<b>40</b>
A. Base de dades .....	40
B. JARs del projecte .....	41
C. Manuals .....	42

## CAPÍTOL 1. Introducció

---

El present projecte consisteix en crear una web convencional descriptiva o informativa d'una empresa. L'empresa en qüestió és un taller mecànic de servei ràpid que ofereix serveis que requereixen poc temps de treball. Aquesta és una característica important perquè dona garanties de que els clients accedeixin via web. Al mateix temps que ofereix aquest servei a mode d'anunci, també ofereix un altre servei més particular: Una agenda virtual per gestionar les peticions de servei.

Aquest sistema d'agenda virtual, ha grans trets, gestiona les peticions que li arriben d'usuaris connectats a la web. Verificar que la petició no coincideixi a la mateixa hora del mateix dia amb una altra que s'ha demanat prèviament i estar enregistrada al sistema.

La restricció de no coincidir en el temps amb una altra és un requisit indispensable, ja que s'ha suposat en el nostre taller, que per la capacitat de treball que té, el nombre de treballadors i altres factors, s'enregistrarà com a màxim una petició per hora, nou per dia..

És una aplicació web, una aplicació de les anomenades on-line. Una pregunta podria ser: Perquè demanar un servei via web i no físicament o per altres mitjans tradicionals?

Doncs entre d'altres motius, és una nova cultura que té la seva lògica. En un portal web pots veure la imatge que té una empresa, la seva especialitat, on centra el seu negoci. Però no només una, sinó de desenes d'elles en poc temps i còmodament.

### 1.1 Justificació del TFC

---

Les empreses i comerços saben de la importància de tenir una web del seu negoci, com a vehicle transmissor de publicitat dels productes i serveis que ofereixen. Una web és equivalent ha tenir un logotip o un nom, els identifica i els dona conèixer.

Els usuaris internautes, realitzen cerques per diferents motius, segons

el producte, el moment o la circumstància en que es troben.

Així es poden fer cerques buscant una oferta o el millor preu d'un producte en concret: i triomfarà el més barat.

Un segon tipus de cerca pot ser per al lloc on es pugui aconseguir el producte. Els motius poden ser per la proximitat a la feina, lloc de residència, aprofitar un viatge o visita a una ciutat.

Un tercer tipus de cerca és el servei o producte oferit. No es mira tan el preu, que també és important, ja sigui perquè es té referències que una determinada marca ofereix bona qualitat ha un preu que acceptable. El cas és que si tant parlat del servei que ofereix una empresa pots confiar sense dubtar-ho.

Un cop l'usuari ha trobat el producte o servei que buscava, el que vol fer és comprar-ho o contractar-ho perquè ja està convençut i té tota la informació que desitjava. Un altre motiu afegit pot ser la comoditat de contractar-ho via web. Els formularis web són entenedors i s'omplen per teclat, més còmodament que si s'han de fer a mà o dictant les dades via telefònica. Es clar que l'empresa ha de donar la cobertura necessària i tenir la infraestructura de servidors i bases de dades apropiades per l'activitat que desenvolupa.

Cada vegada hi han més ordinadors personals, ja sigui de sobretaula o en el mòbil. Hi ha més hàbit en la cerca de serveis via web, tot això fa que el mercat per aquesta via sigui cada cop més gran, i si és més gran s'ha d'aprofitar, perquè el que es busca és tenir més quota de mercat.

---

## 1.2 Objectius del TFC

---

El primer objectiu és crear una web per informar sobre una determinada activitat, en aquest cas tot el referent a productes i serveis d'un taller mecànic. Per a que l'usuari sàpiga si el que busca s'apropa al que se l'hi està informant. Per suposat la ubicació del taller i com contactar-hi. Aquest seria un primer objectiu que per si sol ja donaria possibilitats de promoció del taller.

Un segon objectiu és el d'interaccionar amb l'usuari, i que l'usuari interaccioni amb la web i en el seu efecte amb el taller mecànic. No

solament es publica informació, sinó que també hi ha una comunicació personal de les dues parts: de l'usuari i del taller mecànic.

És una voluntat de servei, una voluntat de que el client demani un servei i que sàpiga que té un fil directe amb l'empresa. Té un canal obert de comunicació, no anònima, amb dades personal de registre i identificació.

Un tercer objectiu és el de tenir un registre de les dades que es gestionen i s'utilitzen en la web. Hi han dos grups de dades: les dades dels clients, i les dades de les sol·licituds de servei.

Dins les dades dels clients hi han les dades d'identificació per mitjà d'un nom i clau d'usuari. Aquesta identificació dona la possibilitat de tenir un espai d'usuari on guardar les seves dades.

Aquest registre és profitós per una segona via; el taller mecànic se serveix de les dades, que en forma de comanda realitza l'usuari. Per tant el taller rep dades útils en temps diari periòdic, que confinades amb unes llistes de taules són proveïdes al taller.

Tot això amb un darrer objectiu: la funcionalitat, tenir un conjunt de funcions que donin com menys problemes millor.

El disseny de les pantalles s'ha volgut que fos entenedor, intuïtiu, amb les funcions indispensables i treure les redundàncies per evitar confusions. La navegació per les pantalles és simple, encara que sigui el primer cop que s'hi accedeix. A partir de l'estructura inicial es poden ampliar les funcions amb un cost reduït.

---

### 1.3 Enfocament i mètode seguit

---

El primer de tot és crear una estructura a partir de la qual poder incorporar les funcions necessàries. Una estructura que permeti inserir-hi noves eines o noves accions en resposta a noves peticions. Inicialment s'ha dotat d'unes funcions bàsiques i concretes, però més endavant es poden ampliar si es volen donar altres serveis.

En aquesta estructura s'utilitzen unes eines de programari per implementar totes les accions, funcions, pantalles, resultats, etc....

S'han buscat unes eines de programari que simplifiquin al màxim el treball de fer codi i que donin bones prestacions. Una altre

característica és la facilitat d'acoblament del programari, de res serveix tenir bones eines si la integració de totes elles dona problemes, i perjudica el temps ha dedicar ha la implementació i més endavant en el seu funcionament.

Buscar un servei que el taller pugui oferir per mitjà de la web, donar un servei aprofitant que l'usuari es troba dins la web. El servei elegit és el de contactar amb el taller i fer una reserva demanant una hora per tal de fer una reparació en el cotxe.

És un servei que es molt àgil de realitzar i el client pot gaudir de la ventatja de fer reserves via web amb qualsevol PC que tingui a l'abast. Per al taller també és una comoditat, el sistema de forma autònoma pren nota de les sol·licituds i al mateix temps mirant de repartir-les en el temps.

El sistema de gestió de comandes té una complexitat i dona unes prestacions, sempre es poden incorporar més restriccions i funcions. S'ha considerat que en principi era suficient a l'espera de possibles ampliacions. L'Arquitectura està feta, sols falta codificar noves funcions si aquest fos el propòsit en el futur.

La sol·licitud de serveis vertebrava tota l'aplicació, de principi a fi. Començant per la primera pantalla de la web fins als formularis i taules que gestionen visualment les sol·licituds.

---

## 1.4 Planificació del projecte.

---

La planificació del projecte s'ha elaborat tenint en compte la naturalesa de la branca en que es fa el treball. El cicle de desenvolupament de programari està força estandarditzat, les etapes són molt semblants encara que es poden afegir o adaptar alguna d'elles.

En cada etapa es fan coses diferents, no es repeteix treball d'etapes anteriors. Per exemple, al fer codi, no ens preocupem d'analitzar nous casos d'ús. Aquesta és una manera, però hi han altres.

En la figura 1 es pot veure la planificació del projecte: les etapes estan a la part esquerra i cada franja de color representa el temps invertit en la fase. Tot hi ser un projecte relativament petit, al tenir que muntar el



programari, documentar-se i fer una estructura de classes com si es tractés d'un de més gran, la feina és considerable.

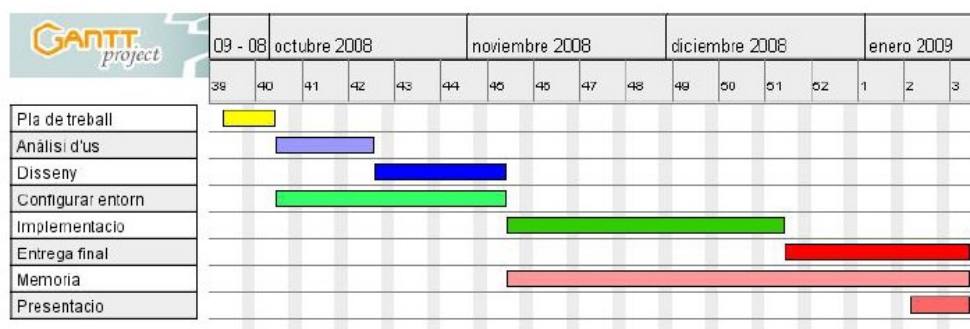


Figura 1. Diagrama de gant.

Ha diferència d'un projecte en el món laboral, on l'empresa treballa normalment en uns camps específics, cal configurar l'entorn de treball afegint el programari necessari. La memòria és una altra part ha tenir en compte, ja que és la part més important del projecte. Per això s'ha començat a redactar després del disseny; un cop sols falta implementar-lo.

Cal considerar que en la fase d'anàlisi i disseny, s'ha començat ha muntar el programari, així en el moment de la implementació es pot dedicar tot el temps a fer codi i es poden solucionar els problemes de compatibilitat de programari sense preses.

## 1.5 Productes obtinguts.

Tenim una web on el taller mecànic pot posar publicitat de marques de recanvis que utilitzi, i així promocionar la marca i ell mateix. També es poden penjar ofertes de productes, o preus de peces per captar clients. Si té contractes amb marques comercials, que millor que penjar-les a la web per atraure gent per a que compri i demani serveis, un reclam molt important que de ven segur agrairà.

Tot usuari que visiti la web pot tenir el seu espai d'usuari si es dona

d'alta al sistema. Llavors podrà fer sol·licituds, demanant hora per fer una reparació. Totes les dades es guarden a la base de dades, on cada client té les seves dades personals, i les dades de la seva sol·licitud si ha demanat un servei.

La suma d'espais dels clients formen una agenda gran, on per una part els clients, amb el control del sistema, anoten les cites a complir i per l'altra part, el taller, es mira l'agenda, per saber les reparacions a fer.

---

## 1.6 Resum resta de capítols

---

### -Capítol 2: Anàlisi.

Les persones físiques que faran ús de l'aplicació són: el client i el cap de taller.

Els rols de cadascú són diferents, tret de identificar-se al sistema.

Partint de la naturalesa d'un i de l'altre, s'han analitzat les accions que poden demanar, tenint en compte el tipus de web, i s'han escollit les més necessaris donada la grandària del projecte.

S'han detallat els casos d'ús, i què signifiquen. El diagrama de casos d'ús mostra la relació d'accions i actors.

### -Capítol 3: Disseny

El projecte està format per capes, i cadascuna fa una funció específica.

Cada capa té les seves classes i fitxers amb el que portar a terme la seva tasca. Això ajuda a tenir parts del codi totalment independents, amb el que podem fer tests més fàcilment, controlar els errors, etc...

Ha mode d'introducció, les capes són: Persistència, Servei, Presentació.

En el capítol s'expliquen els detalls de cada una.

La base de dades està formada per dues taules que cobreixen les necessitats de l'aplicació. Una és per enregistrar els clients i l'altre és per les sol·licituds.

### -Capítol 4. Implementació

Per implementar l'aplicació es segueix el model de capes de la fase de disseny de tal manera que per implementar la capa de presentació s'utilitzarà el framework Struts2, per la capa de persistència Hibernate

i per la capa de servei Spring.

Un cop implementada tota l'aplicació s'encapsularan tots els directoris i fitxers en un fitxer war.

Per últim cal crear la base de dades, en aquest projecte s'ha utilitzar el servidor de base de dades MySQL.

#### -Capítol 5. Instal·lació

Per executar l'aplicació cal tenir instal·lat una sèrie de programari com és: una eina de desenvolupament de java, un servidor web per desplegar l'aplicació, un servidor de base de dades per gestionar el tràfic de dades.

#### -Capítol 6. Proves

Es prova l'aplicació per tenir un contacte visual i mostrar la interfície que es trobarà l'usuari.

Distingim dos grups d'usuaris:

- ◆ Els clients que accediran a la web per fer una reserva per reparar el seu cotxe.
- ◆ Els responsables del taller, per llistar les reserves.

Ha partir dels dos rols es mostren les funcions per cada un d'ells.

#### -Capítol 7. Valoració econòmica

Es calcula el cost econòmic del projecte ha partir del nombre de treballadors amb el perfil i el cost per hora de cadascun d'ells.

Multiplicat per les hores emprades per tots ells en cada fase. Ha tot això el cost que ha suposat el programari i el maquinari per desenvolupar l'aplicació.

#### -Capítol 8. Conclusions

Conclusions i valoració final de la realització d'aquest projecte.

## Capítol 2. Anàlisi

---

S'ha fet un anàlisi de les funcions que necessitaran les persones que accedeixin a la web, des del seu punt de vista. Posar-se al seu lloc i veure les funcions que voldran demanar, i les que necessitaran en el context de l'aplicació. Els problemes que es poden trobar si insereixen dades errònies o no vàlides. I el que és més important, decidir quines accions es serveixen per a que les puguin utilitzar els actors.

Analitzar el comportament del sistema en el supòsit de la utilització de cada funció per als actors físics i veure si hi han funcions que pot fer un actor des de dins del sistema. Accions que es portaran ha terme sense que ningú les hagi demanat, ja que estaran programades per casos especials o cada cert temps.

Per a cada cas d'ús, s'ha d'implementar amb codi, la relació de tots ells és important per a que després l'aplicació tingui un funcionament àgil i coherent amb les necessitats en cada punt de l'aplicació.

L'usuari d'aplicacions informàtiques demana senzillesa en la interfície gràfica i en el nombre de funcions que se li ofereixen. El funcionament és més ràpid i l'usuari ho agraeix.

### 2.1 Estudi del funcionament

---

Primer de tot cal veure quin serà el funcionament de l'aplicació des d'un punt de vista conceptual. En base a la complexitat que es vulgui aconseguir es proposaran més o menys requeriments. En aquest cas no es vol gaire complexitat, per tant es tindran les eines necessàries per portar a terme les funcions principals.

Partint de que el client podrà fer una reserva per fer una reparació al seu cotxe, caldrà un cas d'ús per demanar un servei. Les dades ha introduir seran la descripció de la reparació i l'hora. Cal observar que si l'hora ja està reservada per un altre client, tindrà que demanar-ne una de diferent. En aquí es necessita un cas d'ús que llisti les hores lliures del dia demanat.

Per tal que el client vegi la seva reserva, cal un cas d'ús que la llisti. S'ha de llistar en el moment en que el sistema l'ha acceptat, per deixar constància. I també quant el client accedeixi al sistema s'ha de llistar per a que recordi quina sol·licitud ha fet. Una sol·licitud per client és suficient.

La reserva s'ha de poder esborrar en cas que no es pugui assistir o que el problema ja estigui solucionat. Cal doncs un cas d'ús per esborrar.

Per tal de tenir control de qui demana serveis, cal un cas d'ús per donar-se d'alta al sistema. Les dades personals es guardaran i es podran consultar per contactar-hi, per ser més rigorosos en la gestió dels clients. Aquestes dades es poden variar més endavant, però no s'ha considerat donar l'opció de modificació.

Si cada client té un espai d'usuari, caldrà que cada un tingui una clau per accedir al sistema. Al iniciar la sessió caldrà introduir un nom d'usuari i una clau per tenir accés a demanar serveis. En el formulari per donar-se d'alta hi haurà un camp per escollir la clau.

El taller ha de poder veure les sol·licituds rebudes. Per això cal un cas d'ús per llistar les reserves. El més normal és llistar les reserves seleccionant un dia determinat, llavors podrà veure la feina que té, i si fos el cas, ocupar els llocs lliures amb les sol·licituds que pugui rebre d'un altre medi.

---

## 2.2 Actors i casos d'ús

---

### -Actors:

Un actor és una entitat que està a fora del sistema, però té una relació amb aquest perquè fa ús de les funcions que aquest sistema presta. Per exemple una persona que des d'un processador de textos selecciona l'opció imprimir, l'actor és la persona que interacciona amb el programari i en fa ús.

En aquest projecte tenim els actors:

*Client:* Tota persona que accedeix a la web des d'un terminal i que al seleccionar les opcions disponibles desencadena accions de

programari tals com: donar-se d'alta o esborrar una sol·licitud, etc.

*Cap de taller:* És una persona que accedeix al sistema amb un nom d'usuari i clau especials. El sistema al detectar aquest nom i clau serveix una interfície diferent a la resta d'usuaris que accedeixen a la web, i que també fan un login. Els caps de taller no fan sol·licituds, sinó que les llisten en taules que el programari proporciona.

*Time:* És una entitat abstracta que desencadena accions depenent de les circumstàncies en que es trobi l'aplicació en algun punt del seu funcionament. Un exemple pot ser el de tenir actualitzada una taula d'una base de dades, eliminant les files a partir d'una determinada data d'antiguitat.

-Casos d'ús:

Un cas d'ús és un requisit potencial d'un sistema de programari que s'està actualitzant o creant. Consisteix en una seqüència d'interaccions en resposta a un event per part d'un actor. Per tal que es produeixi un event i la resposta d'aquest, es proporcionen varis escenaris o pantalles. Aquestes pantalles poden ser compartides per varis casos d'ús, segons les necessitats de cada cas.

En aquest projecte tenim els casos d'ús:

*Donar-se d'alta:* Serà utilitzat pel client.

El client tindrà aquesta opció en la interfície gràfica per tal de que es doni d'alta al sistema. Consisteix ha omplir un formulari amb les dades personals de la persona interessada. Es aquí on s'escull el nom d'usuari i la clau que posteriorment s'utilitzarà per accedir al espai reservat.

*Login:* Serà utilitzat per el client i el cap de taller.

Al accedir a l'aplicació, hi ha un formulari amb dos camps per introduir el nom i la clau. El sistema buscarà si estan a la base de dades, en cas afirmatiu es permet l'accés a la zona restringida.

*Demandar servei:* Serà utilitzat per el client.

En la zona restringida hi ha un espai per fer sol·licituds per demanar hora per fer una reparació. Consisteix en omplir un formulari on s'ha d'indicar un dia, una hora i la descripció de la reparació que es vol fer.

Només es pot tenir una sol·licitud en la base de dades, en cas que es vulgui modificar, s'ha d'esborrar la que es tingui feta.

*Esborrar servei:* Serà utilitzat per el client i time.

En el cas del client, en el seu espai d'usuari té un lloc on pot veure la sol·licitud que ha fet amb una opció per esborrar-la. Si no ha fet cap sol·licitud no fa res.

Quan un client s'identifica al iniciar la sessió, es carreguen les dades del client. Abans de carregar la sol·licitud es mira si la data ha caducat, en cas que fos així l'actor time l'esborra.

*Llistar serveis:* Serà utilitzat per el cap de taller.

La persona que s'identifiqui al sistema com a cap de taller disposarà d'un formulari on indicant un dia, es llistaran tots els serveis demanats per aquell dia. Cada fila hi haurà les dades de la comanda i les dades personals dels client.

*Llistar servei:* Serà utilitzat per time.

En el moment de identificar-se un client, si el troba a la base de dades, tot seguit comprova si té un servei sol·licitat, i al carregar la pantalla de la zona restringida surt llistat. Si per contra no hi ha cap servei demanat es mostra un missatge que ho indica.

*Llistar horaris:* Serà utilitzat per time.

Consisteix en llistar les hores lliures del dia que el client ha demanat una cita, però que l'hora que ha seleccionat està ocupada per un altre client.

En el diagrama següent es poden veure les relacions entre actors i casos d'ús. Destacar que per fer un login cal prèviament donar-se d'alta. El mateix passa amb llistar horaris, abans cal haver introduït una hora ocupada.

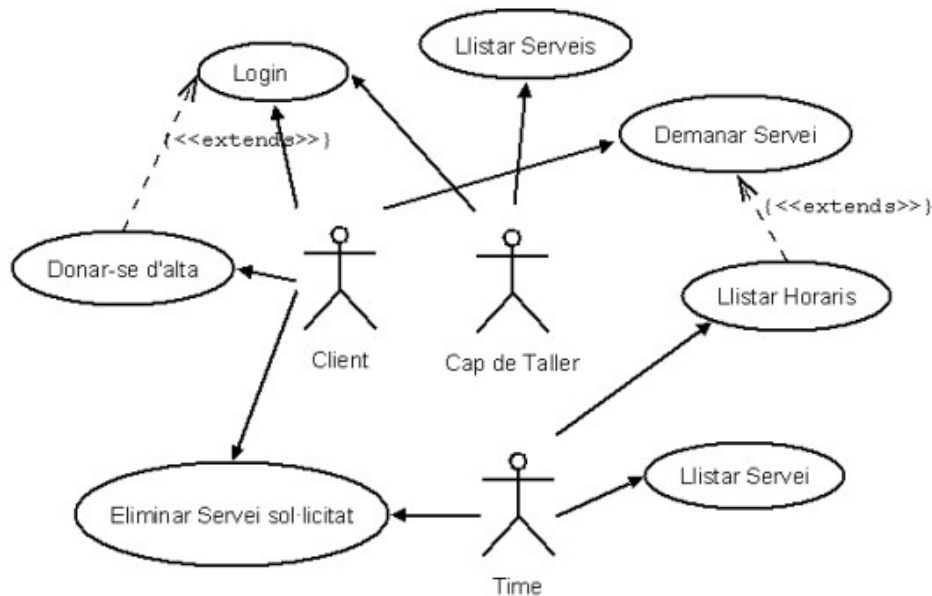
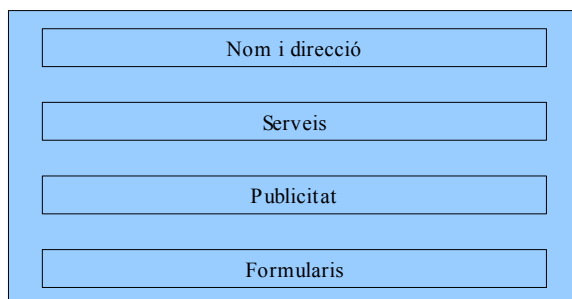


Figura 2. Diagrama de casos d'ús

## 2.3 Les pantalles

Els escenaris que es desenvolupa l'aplicació formen quatre pantalles, una d'elles per al cap de taller, dues per al client, i una compartida.

Excepte la pantalla d'alta que consta d'un formulari, les altres segueixen una estructura amb les següents parts: una part amb el nom i la localització del taller, una que mostra els serveis que dona, una per la publicitat, i una darrera part pels formularis.





Cada pantalla té un formulari diferent d'acord amb la seva funció.

Pantalles de l'aplicació:

*Pantalla login:* ha més de la part comuna, consta de dos camps: nom i clau. Té un enllaç per donar-se d'alta pels clients no registrats.

*Pantalla alta:* hi ha un formulari per omplir amb les dades personals del client i del cotxe.

*Pantalla reparacions:* ha més de la part comuna, com es pot veure a la figura 3 hi ha una part per introduir sol·licituds i una altra per llistar la que es tingui sol·licitada.

**Servei Sol·licitat:**

---

Reparació	Data	Hora
Cap Sol·licitud		
<input type="button" value="Esborrar"/>		

---

**Sol·licitar Servei:**

---

Només es pot reservar una cita per persona  
*No es poden fer reserves pel mateix dia*

*Exemple: Canvi d'oli i revisar tub d'escapament.*

Reparació ha fer\*:

*(any-mes-dia) Exemple: 2008-05-22*

Data\*:

Hora\*:

Figura 3. Pantalla per fer sol·licituds

*Pantalla llista:* ha més de la part comuna, com es pot veure en la figura 4 hi ha un camp per introduir la data per tal de llistar les

reparacions sol·licitades del dia assenyalat.

Els camps de la llista són les dades de la sol·licitud i les dades personals del client que ha demanat la cita. Així es té tota la informació ordenada i no cal consultar fitxes de dades en un altre lloc.

**Serveis sol·licitats**

(any-mes-dia) Exemple: 2008-05-22

Data:

<b>Id</b>	<b>Reparació</b>	<b>Data</b>	<b>Hora</b>	<b>Nom</b>	<b>Direcció</b>	<b>NIF</b>
Cap Sol·licitud						

Figura 4. *Pantalla per llistar sol·licituds*

## Capítol 3 Disseny

---

El disseny està format per capes, cadascuna d'elles realitza una funció i la formen una sèrie de classes. Hi han capes per controlar les pantalles que es mostren, altres a fer operacions sobre la base de dades. Totes unides fan funcionar l'aplicació, i facilita la tasca en futures ampliacions.

Una característica important és que així tenim un programari molt robust. Es poden controlar els errors que poden donar les instruccions de les diferents parts de l'aplicació.

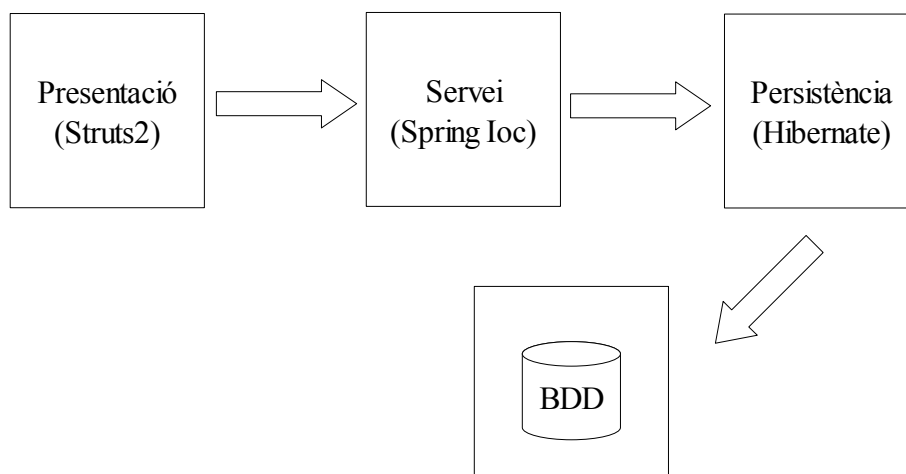


Figura 5. Capes de l'aplicació

### 3.1 Capa persistència

---

La capa persistència està formada per una interfície i una classe que la implementa (veure figura 6). Els mètodes utilitzen classes especialitzades per executar sentències de inserció, esborrat, cerca, etc..., dins la base de dades. Com s'explicarà en un altre capítol, mitjançant uns fitxers es crea una correspondència entre les taules de la base de dades i classes entitat. Amb aquests mapes es pot treballar mes còmodament perquè s'utilitza classes entitat java.

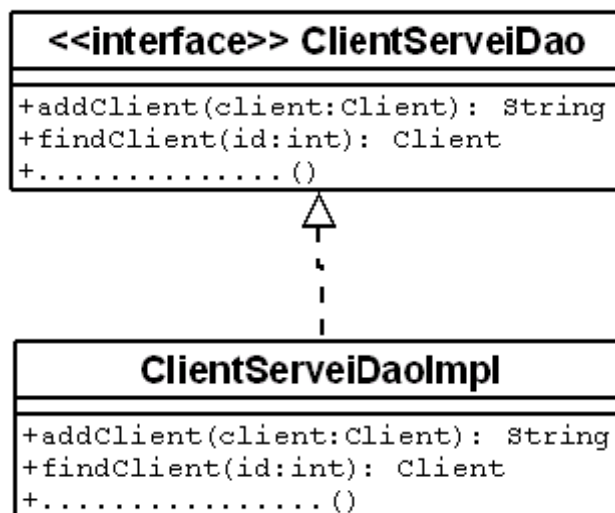


Figura 6. Diagrama de classes persistència

### 3.2 Capa servei

---

Com es pot veure a la figura 7, està formada per una interfície i una classe que la implementa. La classe `ClientServeiDao` té un atribut que correspon a la interfície `Dao`. I amb un `get` i un `set` per escriure i llegir l'atribut.

Aquesta capa serveix per unir la capa persistència, la capa de control i les pantalles. Amb aquest sistema d'integració s'aconsegueix tenir capes totalment independents.

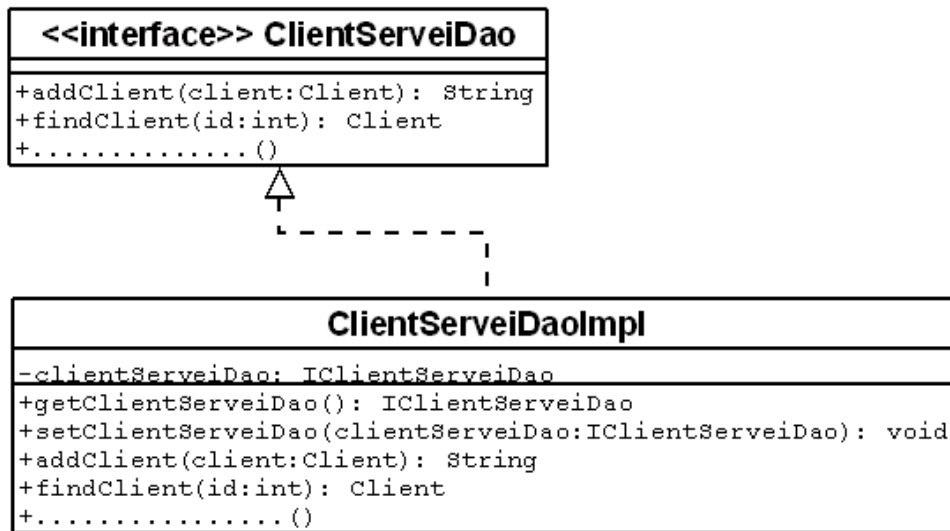


Figura 7. Diagrama de classes servei

---

### 3.3 Capa presentació

La capa de presentació està formada per una classe on hi ha el codi de les operacions a fer tenint en compte les opcions que ofereix la interfície gràfica. Ha mode de controlador, aquesta classe juntament amb un fitxer auxiliar s'encarreguen de mostrar una pantalla o canviar-la, i fer les accions que calguin a partir d'un mapa preestablert de treball.

Les pantalles també formen part d'aquesta capa, que són les encarregades de mostrar les opcions disponibles i posteriorment el resultat, tal com missatges, formularis, i tot el necessari depenent de l'aplicació.

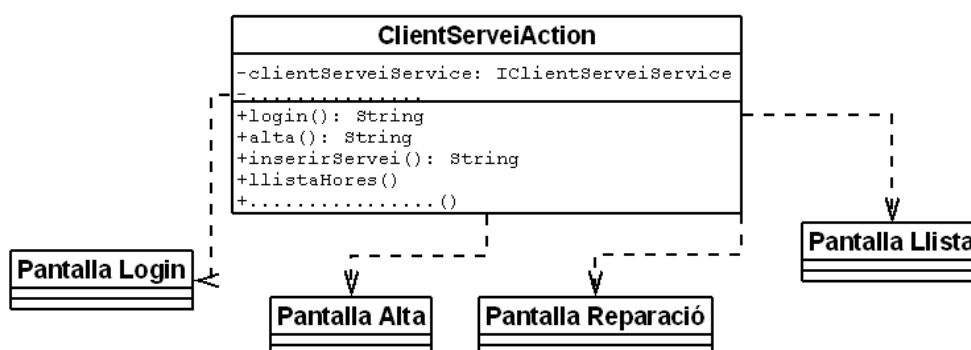


Figura 8. Diagrama de classes presentació

La classe ClientServeiAction incorpora uns atributs que es corresponen amb els camps dels formularis. I els corresponents mètodes de lectura i escriptura. El constructor rep com a paràmetre un objecte IclientServeiService i així pot executar mètodes de la capa de servei.

### 3.4 Diagrama de classes

En la figura 9 es pot veure el diagrama de classes de l'aplicació. Les capes persistència, servei i presentació formen una estructura molt fàcil de treballar degut a la independència de cada part.

Les classes entitat client i servei son utilitzades per totes les capes de l'aplicació. També es pot veure com van unides les diferents capes, d'aquesta manera es poden utilitzar diferents frameworks i es poden enllaçar fàcilment.

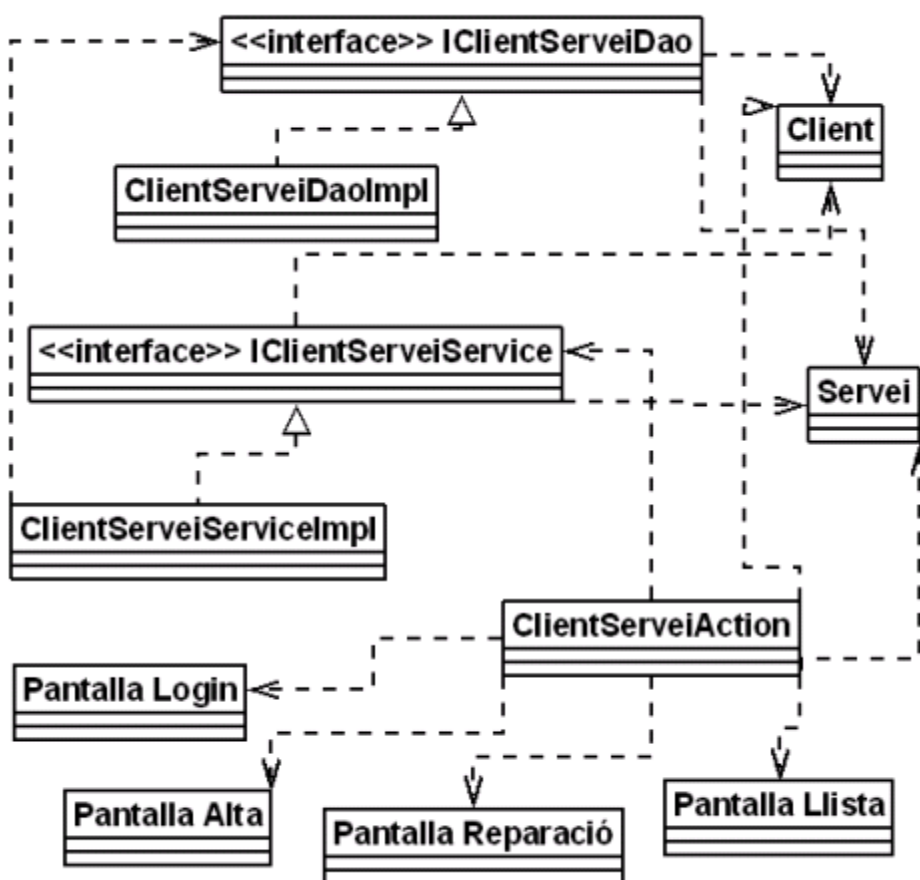


Figura 9. Diagrama de classes

### 3.5 Base de dades

La base de dades està formada per dues taules: client i servei (veure figura 10). Cada client de la taula client, té un o cap servei de la taula servei.

Els camps de la taula client es corresponen amb les dades personal i dades del cotxe. Això s'ha fet així perquè tant les operacions d'alta com de llistat, es necessiten totes les dades. I no s'ha fet més divisió de taules per la naturalesa de l'aplicació.

La taula servei té els camps necessaris per fer una sol·licitud, juntament amb un camp que és el identificador del servei i clau primària, que es correspon amb la clau primària del client . Amb el que tenim les taules enllaçades.

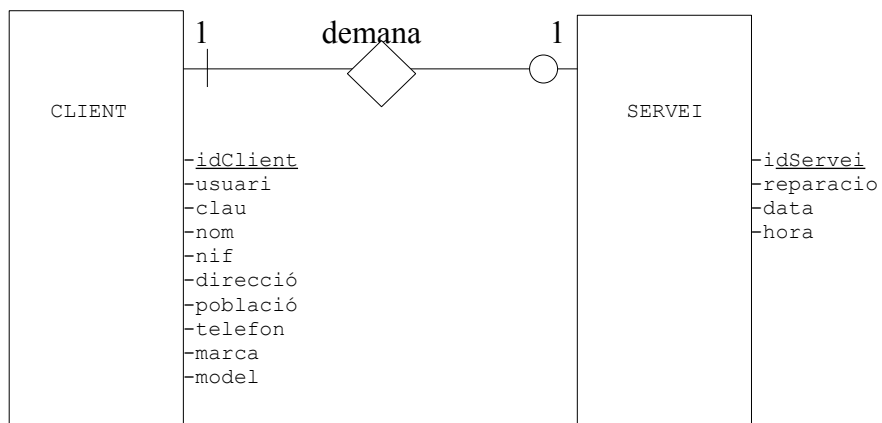


Figura 10. Diagrama ER de la base de dades



## Capítol 4. Implementació

Les diferents capes que formen l'aplicació tenen els seus respectius components, i que es poden ajuntar amb un fitxer war.

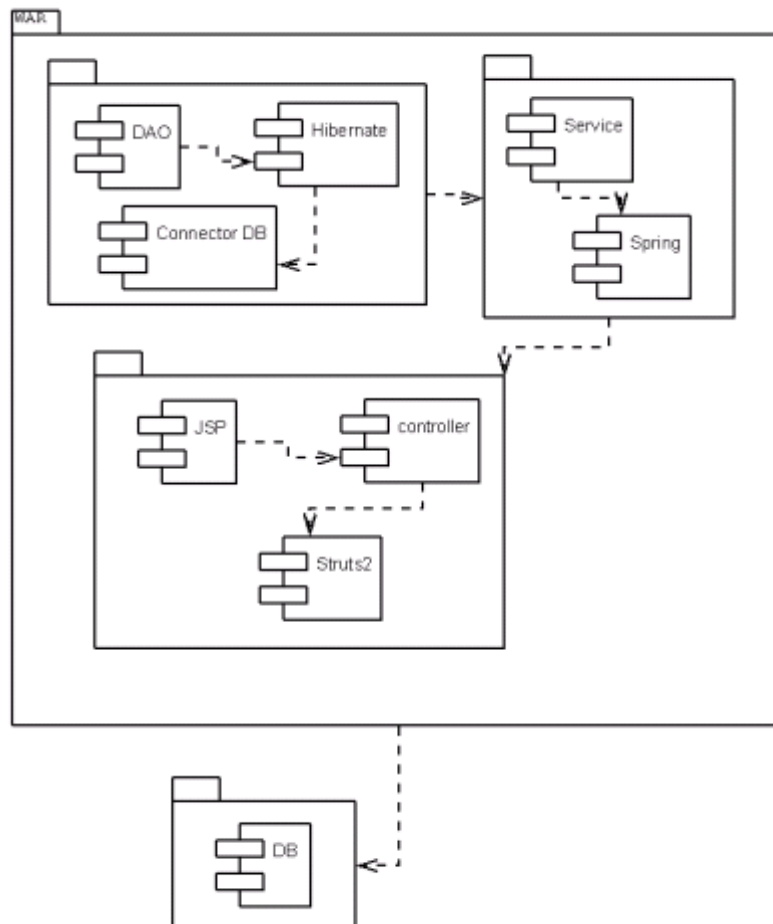


Figura 11. Diagrama de components

### 4.1 Part Struts2

Struts2 implementa la part del controlador i les vistes o pantalles.

Les principals característiques d'aquest framework són:

- ◆ S'integra molt bé amb Spring.
- ◆ Es pot especificar amb tot detall que farà cada acció.
- ◆ Facilitat per treballar amb JSPs.

El controlador consisteix en un fitxer on hi han implementats els mètodes que es cridaran per mitjà dels botons de les pantalles, i un altre fitxer per indicar-ho.

Una altra part, unida a les pantalles, és el validador. És l'encarregat de verificar les entrades dels camps dels formularis.

#### -Fitxers JSPs

Els jsp són uns fitxers amb codi html i java. El codi java es compila i s'executa en un servidor contenidor de servlets.

Cada jsp es correspon amb una pantalla, la qual es projectada per el controlador.

```
.....  
<s:form action="login" method="post">  
<s:textfield name="usuari" key="" label="Usuari" maxlength="20"/>  
<s:password name="clau" key="" label="Clau" maxlength="20"/>  
<s:submit value="Login" align="center"/>  
</s:form>  
.....
```

Figura 12. Codi del fitxer login.jsp

En la figura 12 es pot veure una part del fitxer login.jsp corresponent al formulari per identificar-se.

#### -Fitxer action

És un fitxer java on hi han implementats els mètodes que s'executaran segons l'acció que demani l'usuari.

```
.....  
public String registre() {  
    return Action.SUCCESS;  
}  
.....
```

Figura 13. Codi del fitxer ClientServeiAction.java

En la figura 13 es pot veure un mètode del fitxer ClientServeiAction.java. El tipus de retorn indica al fitxer Struts.xml quina pantalla s'ha de carregar.

### -Fitxer struts

La funció d'aquest fitxer és la d'indicar, per cada acció que sol·liciti l'usuari, el mètode que s'executara i la pantalla que es carregarà després.

```
.....  
<action name="login" method="login" class="clientServeiAction">  
  <result name="success">reparacio.jsp</result>  
  <result name="error">login.jsp</result>  
  <result name="login">llista.jsp</result>  
</action>  
.....
```

Figura 14. Codi del fitxer *struts.xml*

En la figura 14 es pot veure el codi que correspon a una acció. Tenim el nom de l'acció i el mètode, i tres retorns amb les seves pantalles.

### -Validar

Per tal de controlar que els diferents camps dels formularis s'insereixin el tipus de valor correcte o que no quedin buits, s'utilitzen uns fitxers que ho especifiquen.

Es pot especificar que un camp sigui un enter, una cadena, estar entre un mínim i un màxim, etc...

```
.....  
<field name="hora">  
  <field-validator type="required">  
    <message>Posa l'hora.</message>  
  </field-validator>  
</field>  
.....
```

Figura 15. Codi del fitxer *ClientServeiAction-alta-validation.xml*

En la figura 15 hi ha la validació del camp hora del formulari alta. En cas que aquest camp es deixi en blanc, es mostrarà un missatge.

---

## 4.2 Part de Hibernate

El framework Hibernate s'encarrega de la capa persistència. Utilitza

uns fitxers en forma de mapa, de tal manera que cada taula de la base de dades té la corresponent classe entitat.

D'aquesta manera les classes de la capa persistència, els seus mètodes crea'n sentències en llenguatge HQL. Hibernate proporciona classes pròpies que faciliten la creació de sentències per fer operacions amb la base de dades.

#### -Fitxer Dao

El fitxer ClientServeiDaoImpl implementa uns mètodes que executen operacions de selecció, esborrat, llistar, etc.. sobre les taules de la base de dades.

```
.....  
public List findHores(Date data) {  
    return getHibernateTemplate().find("select s.hora from  
    Servei s where s.dia = ?", data);  
}  
.....
```

Figura 16. Codi del fitxer ClientServeiDaoImpl.java

En la figura 16 es pot veure un mètode que executa una sentència HQL de cerca, i retorna una llista de dades. Aquesta llista passa a la capa de servei i aquesta l'ha dona a la capa presentació, i les dades es poden veure en una de les pantalles.

#### -Fitxers hbm

Cada taula de la base de dades té el seu fitxer hbm. El que fa es enllaçar cada camp d'una taula amb el corresponent atribut de la classe java. Així es poden fer operacions amb llenguatge HQL.

```
.....  
<property name="clau" column="clau" />  
.....
```

Figura 17. Codi del fitxer Client.hbm.xml

En la figura 17 es pot veure com s'enllaça l'atribut clau de la classe Client amb el camp clau de la taula Client.

## 4.3 Part de Spring

---

Spring implementa la capa de servei. La seva feina és la d'integrar la part de Struts2 i la part de Hibernate. També s'encarrega de controlar les transaccions, si la base de dades treballa amb transaccions. En el cas de MySQL s'han d'especificar taules de tipus InnoDB, per treballar amb transaccions..

### -Fitxers Serveis

El fitxer ClientServeiServiceImpl implementa els mètodes que criden mètodes de la capa de persistència.

```
.....  
public Client findClient(int id) {  
    return clientServeiDao.findClientById(id);  
}  
.....
```

Figura 18. Codi del fitxer ClientServeiServiceImpl.java

En la figura 18 es mostra el codi del mètode findClient que crida al mètode findClientById del fitxer DAO.

### -Fitxer web

El fitxer web.xml s'encarrega de gestionar les accions i seleccionar els interceptors a invocar. Aquesta tasca la porta a terme Struts2.

Una altra tasca és la d'inicialitzar el context amb la configuració que es troba al fitxer applicationContext.xml.

```
.....  
<listener>  
  <listener-  
class>org.springframework.web.context.ContextLoaderListener</listen  
er-class>  
</listener>  
.....
```

Figura 19. Codi del fitxer web.xml

En la figura 19 es pot veure el codi que inicialitza el context.

### -Fitxer applicationContext

El fitxer applicationContext.xml aporta la configuració de l'aplicació. Aquest fitxer és de lectura en temps d'execució.

Proporciona els mètodes necessaris per accedir a tots els components de l'aplicació. Es crea la sessió per accedir a la base de dades. S'uneixen les diferents capes escrivint el mètodes de les classes

```
.....  
<property name="transactionAttributes">  
  <props>  
    <prop key="add*">PROPAGATION_REQUIRED</prop>  
    <prop key="find*">PROPAGATION_REQUIRED,readOnly</prop>  
    <prop key="delete*">PROPAGATION_REQUIRED</prop>  
    <prop key="serveisl*">PROPAGATION_REQUIRED,readOnly</prop>  
  </props>  
</property>  
.....
```

Figura 20. Codi del fitxer applicationcontext.xml

En la figura 20 es pot veure el codi del fitxer applicationContext.xml on es configura el comportament del mètodes, en les transaccions a la base de dades, de la capa servei. D'aquesta manera s'optimitzen les operacions que es realitzen, i s'augmenta l'eficiència.

La base de dades del projecte treballa amb taules MyISAM perquè no hi ha cap clau forana i no cal realitzar les operacions de forma atòmica. En cas contrari s'escollirien taules InnoDB.

---

## 4.4 Fitxer WAR

---

El fitxer war conté les carpetes de l'aplicació, com són els jsp, els xml, les classes java, fitxers de configuració i tot el necessari per ha que funcioni l'aplicació web.

Tenir un projecte web empaquetat en un war facilita el poder manipula tota una sèrie se carpetes fàcilment. Les eines de desplegament reconeixen aquests fitxers comprimits, amb el que tenim un ràpid desplegament.

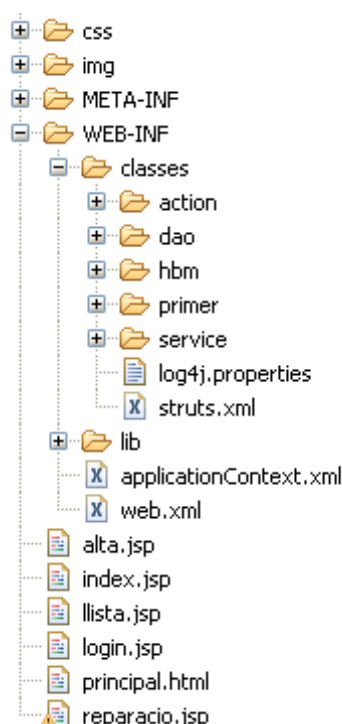


Figura 21. Estructura del fitxer WAR

Aquest projecte s'ha implementat des del ide eclipse, amb el que l'obtenció de fitxer war no dona cap complicació.

### 4.5 Base de dades

---

En aquest projecte s'ha utilitzat el servidor de base de dades MySQL. El fitxer jar connector per la connexió amb la base de dades s'ha dipositat en el directori lib del fitxer war, amb la resta de jars (veure annexe B) dels diferents frameworks.

En el annexe A es pot veure el codi de la base de dades. Cada Client té un o cap servei, i un servei té un client. Al inserir una fila a la taula client el identificador i clau primària s'incrementa en un de l'anterior..

## CAPÍTOL 5. Instal·lació

---

Realitzar la instal·lació no comporta cap problema al tenir el projecte empaquetat en un fitxer war, el servidor web el desplegarà automàticament en iniciar-se.

L'aplicació necessita un nom d'usuari i clau per accedir a la base de dades, per aquest motiu quant es crei la base de dades també es crearà un usuari. Aquest usuari tindrà el mateix nom i clau que els definits en el projecte.

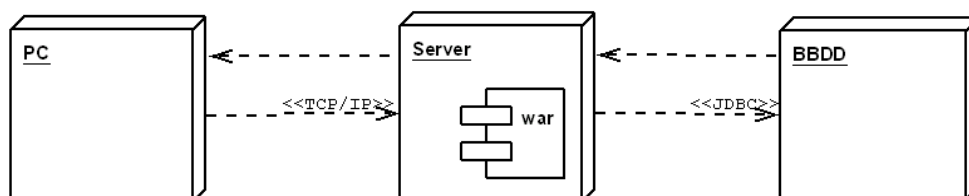


Figura 22. Diagrama de desplegament

### 5.1 Entorn necessari

---

Per configurar l'entorn, cal instal·lar el programari:

- ◆ **JDK**
- ◆ **Servidor**
- ◆ **MySQL**

En aquest projecte s'ha utilitzat el servidor web Apache Tomcat. Es pot utilitzar un servidor d'aplicacions o qualsevol altre servidor web per desplegar el projecte.

Cal mirar que les versions siguin compatibles, per ha que funcionin correctament.



## 5.2 Desplegar i executar l'aplicació

---

Per desplegar el projecte cal seguir els passos següents:

Pas 1: crear la base de dades, les taules i l'usuari amb el fitxer `taules.txt` (veure annex A).

Pas 2: posar el fitxer `war` en el directori `webapps` del servidor Apache Tomcat.

Per executar l'aplicació cal obrir un navegador i posar l'adreça web **`http://localhost:8080/tallerMecanic/`**

Al invocar el nom del `war`, `tallerMecanic`, es carregarà la pantalla de benvinguda especificada en el fitxer `web.xml`.

## Capítol 6. Proves

L'aplicació té dos rols o actors que poden ser interpretats per persones físiques i disposar de les funcions que s'ofereixen. En aquest sentit, provarem l'aplicació per veure el seu funcionament per cadascun dels rols.

### 6.1 Rol client

Un client que accedeix a la web i vol fer reserves, tindrà que donar-se d'alta, si no ho està.



Al introduir una clau repetida, surt un missatge d'error que ho indica. Si el client s'ha pogut enregistrar, tornarà a la pantalla login.

Taller de servei ràpid - Windows Internet Explorer

http://localhost:8080/tallerMecanic/login.action

Archivo Edición Ver Favoritos Herramientas Ayuda

Taller de servei ràpid

Página principal Fuentes (0) Imprimir Página Herramientas

*manteniment de vehicles amb la màxima qualitat.  
Ens trobarà a la carretera de Sant Climent nº54 de Barcelona.*

Aire Aco Aire Acon Amortidors Bateries Tubs d'escapament Frens Pneumàtics Canvi d'oli Pre-ITV

Al fer qualsevol reparació et regalem un clauer RENAULT F1  
Gaudeix dels preus més competitius!

Tenim les marques més fiables del mercat

**NOU SERVEI: -Enregistrat i demana hora per fer reparacions al teu cotxe.**  
Serveis

- **Usuari o Clau no vàlida!!**

Usuari:

Clau:

[Donar-se d'alta](#)

Listo Intranet local 100%

Inicio memoria ... Java EE ... 3 Inter... Adobe R... MySQL A... MySQL Q... ES 17:41

Sol·licitar Servei - Windows Internet Explorer

http://localhost:8080/tallerMecanic/addServei.action

Archivo Edición Ver Favoritos Herramientas Ayuda

Sol·licitar Servei

Página principal Fuentes (0) Imprimir Página Herramientas

**Sol·licitar Servei:**

Només es pot reservar una cita per persona  
*No es poden fer reserves pel mateix dia*

Exemple: Canvi d'oli i revisar tub d'escapament.  
Reparació ha fer\*:   
(any-mes-dia) Exemple: 2008-05-22  
Data\*:

**ERROR: Hora ocupada!!  
El dia 2008-12-23 té les  
següents hores lliures.**

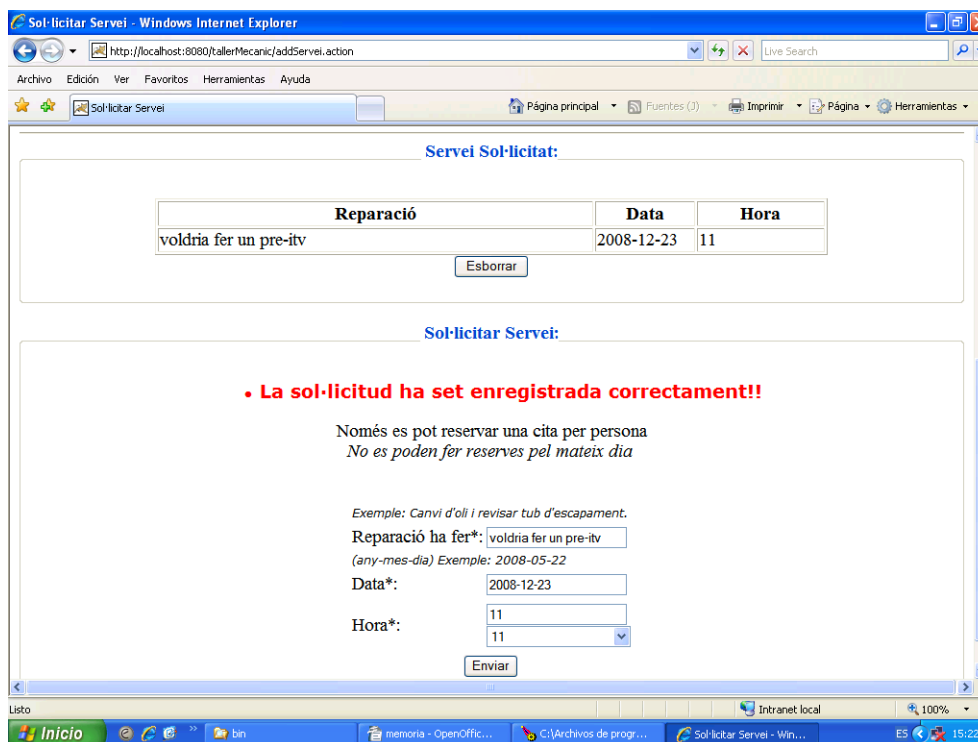
9
10
11
12
14
15

Hora\*:

Listo Intranet local Español (alfabetización internacional)

Inicio memoria - OpenOffic... C:\Archivos de progr... Sol·licitar Servei - Win... ES 14:59

Al identificar-se correctament el client pot fer reserves.  
Si demanar una hora reservada, es llisten les hores lliures del dia demanat.



Si el sistema ha admès la sol·licitud es llistarà en la pantalla, i es mostrarà un missatge de conformitat.

D'aquesta manera s'ha completat el procés de fer una sol·licitud amb que prèviament el client s'ha donat d'alta.

## 6.2 Rol cap de taller

---

El sistema reconeix un nom i clau que el cap de taller ha de donar al identificar-se en iniciar la sessió (nom: *cap*, clau: *890*).

El següent pas és posar una data al formulari, i es llisten les reserves del dia senyalat.

Ens trobarà a la carretera de Sant Clement nº54 de Barcelona.

Aire Aco  
Aire Acon

Amortidors

Bateries

Tubs d'escapament

Frens

Pneumàtics

Canvi d'oli

pre-ITV

Pre-itr

Al fer qualsevol reparació et regalem un clauer RENAULT F1  
Gaudeix dels preus més competitius!

Tenim les marques més fiables del mercat.

**Serveis sol·licitats**

(any-mes-dia) Exemple: 2008-05-22  
Data:

Id	Reparació	Data	Hora	Nom	Direcció	NIF	Pobl
27	voldria fer un pre-itv	2008-12-23	11	jonas	C/bna creu n?342	49761123E	hospitalet

Població	Telèfon	Marca	Model
hospitalet	874322212	volvo	740 vas

El resultat és una taula amb les dades de la reserva i les dades del client.

No es poden llistar les reserves d'un dia anterior al actual. S'ha considerat que no cal conservar les dades de les sol·licituds caducades. El sistema esborra la sol·licitud caducada del client al identificar-se en iniciar una sessió, d'aquesta manera només es guarda una sol·licitud per client i no es llisten les caducades.

## Capítol 7. Valoració econòmica

La valoració econòmica està feta en el supòsit que aquest projecte estès fet per una empresa que es dedica a fer webs. Es té una referència entre les fases, els perfils, les hores i el cost. Fer aquest projecte ha suposat moltes més hores de feina tenint en compte la inexperiència i la documentació.

La realització d'aquest projecte no ha comportat la compra de programari, s'ha utilitzat programari lliure. Per tant cost zero. El maquinari empleat no ha suposat cap cost, ja que no ha calgut comprar cap component extra que fes falta.

El cost de realitzar aquest projecte queda com es mostra a continuació:

	Cap de Projecte	Analista	Arquitecte	Programador Junior	Programador Senior	Total hores
Requeriments	8 hores					8 hores
Anàlisi i Disseny	1 hores	16 hores	2 hores			19 hores
Implementar	1 hores		2 hores	40 hores	40 hores	83 hores
Proves	1 hora		5 hores	5 hores	5 hores	16 hores
<b>Total hores</b>	<b>11 hores</b>	<b>16 hores</b>	<b>9 hores</b>	<b>45 hores</b>	<b>45 hores</b>	<b>126 hores</b>
€/ hora	35 € / hora	30 € / hora	30 € / hora	20 € / hora	30 € / hora	
€/ perfil	385,00 €	480,00 €	270,00 €	900,00 €	1.350,00 €	
<b>Total Projecte</b>						<b>3.385,00 €</b>

Tenim un resum de les hores empleades en cada fase, les hores que ha empleat cada perfil, el cost en euros de cada perfil. Que dona un total de 3385 € en concepte d'hores empleades en l'execució del projecte.

El nombre de persones que han participat en el projecte són cinc, cada una pertany a un perfil diferent. La fase que es dedica major nombre d'hores és la implementació. Tenint en compte que si els programadors tenen més o menys experiència en l'ús dels frameworks emprats,

necessitaran un nombre d'hores diferent.

-Fase de requeriments

En aquesta fase el cap de projecte crea un pla de treball per intentar conciliar la capacitat de l'empresa, els treballadors disponibles, els treballadors que es poden contractar, la durada del projecte, el temps que s'ha de tenir el projecte acabat, la grandària del projecte.

-Fase d'anàlisi i disseny

L'analista d'acord amb les especificacions del client, fa l'anàlisi del projecte amb els diagrames oportuns. I l'arquitecte comença a muntar el programari del projecte. Tot sota la supervisió del cap de projecte.

-fase d'implementació

El programador júnior i el programador sènior, formant equip, implementen l'aplicació. Per la grandària del projecte no fan falta més programadors. L'arquitecte prova i soluciona els problemes del programari. El cap de projecte es informa de l'evolució del projecte.

-Fase de proves

Els programadors proven l'aplicació conjuntament amb l'arquitecte. Es realitzen proves de rendiment i de funcionament. El cap de projecte amb els resultats decideix si donar-lo per finalitzat.

## Capítol 8. Conclusions

---

Treballar en aquest projecte ha set una experiència molt enriquidora per aprendre moltes coses. Proposar un projecte i fixar uns objectius és una primera fase, però després be la segona part, que es que s'han de complir tot hi passar per problemes i múltiples complicacions que van sorgint.

La part més complicada ha estat la d'aprendre ha programar amb nou programari, amb un temps per familiaritzar-s'hi una mica curt. Els objectius marcats en la fase d'anàlisi i disseny s'han complert en la fase d'implementació. S'havia d'acotar el treball ha proposar per després no tindre que retallar funcionalitats.

Després de treballar en aquest projecte els últims quatre mesos, i d'aprendre i utilitzar frameworks, i altres tecnologies d'aplicacions web, he pogut assolir uns coneixements que em permeten pensar en nous projectes i idees.

Vaig triar aquest camp per fer el treball final de carrera perquè em semblava molt interessant, i no m'ha decebut. Sinó tot el contrari, ara que al final estic escrivint les conclusions, em sembla encara més interessant per la diversitat de treballs diferents que es poden fer. Una mica de la culpa la tenen els molts projectes de programari que hi han i que van millorant versió darrera versió.

Desenvolupar aplicacions amb programari utilitzat per grans grups d'usuaris es un factor ha tenir en compte. Al estar més estès, hi ha molta documentació que et poden treure de més d'un problema. La documentació de cert programari és escassa, i es aquí on també molts usuaris creen manuals que són de gran ajuda.

La valoració professional és positiva, aprendre ha desenvolupar un projecte passant per totes les fases del cicle de vida del programari. I la valoració personal no menys positiva al aplicar molts coneixements apresos durant la carrera.



## GLOSSARI

---

**ANALISTA (Informàtic):** Persona que realitza anàlisis de sistemes informàtics.

**APACHE TOMCAT:** És un servidor web, i un contenidor de Java ServerPages (JSP) que compila i transforma en servlets.

**ARQUITECTE (informàtic):** Persona que s'encarrega de les necessitats de maquinari i programari per realitzar un projecte.

**CAP DE PROJECTE (informàtic):** Persona que dirigeix un projecte des del moment de la concepció fins a la comercialització. Les tasques que realitza són: Definir el projecte, redactar les especificacions, calcular el cost, contractar l'equip de treball, realitzar un seguiment periòdic.

**DAO:** Traduit de l'anglès és: Objecte per accedir a les dades  
Dades es refereix a la base de dades.

**ESPRING:** Framwork considerat substituït dels EJB.

**ESTRUTS2:** És un framework especialitzat en portar el control de les accions que es demanen en les pantalles per part de l'usuari, i invocar el mètodes oportuns.

**FRAMEWORK:** És una arquitectura de programari que estructura les relacions de les entitats del seu domini. La funció principal és evitar que el programador tingui que implementar els detalls de baix nivell.

**HIBERNATE:** És un framework per la persistència, que fa mapes entre els atributs d'una base de dades i els objectes d'una aplicació.

**HQL:** És el llenguatge orientat al objecte de hibernate per fer sentències sobre la base de dades.

**IOC (framework):** La inversió del control és una tècnica de

programació orientada a events. Donat un event el framework és el responsable de fer les accions oportunes d'acord amb el comportament atorgat. Ha tot això s'utilitzen arxius xml de configuració.

**InnoDB:** És un tipus de taules que utilitza MySQL. Treballen amb transaccions i tenen integritat referencial.

**J2EE:** (Java 2 Enterprise Edition) És un estàndard per desenvolupar i executar programari d'aplicacions en llenguatge java.

**JAR:** És un tipus d'arxiu que permet executar aplicacions en llenguatge java.

**JDK:** És un programari per desenvolupar programes java.

**JSP:** Tecnologia que permet la creació de documents html de forma dinàmica en un servidor.

**LOGIN:** Procés d'identificació per entrar en una aplicació.

**MyISAM:** És el tipus de taules que utilitza MySQL per defecte. Emmagatzemen dades ha gran velocitat perquè no tenen clau forana i les operacions no tenen atomicitat.

**MySQL:** És un sistema de gestió de base de dades relacional.

**POJO:** Classe senzilla que no depenen de cap framework en concret.

**PROGRAMADOR JUNIOR:** Persona que comença ha programar sense experiència laboral o professional.

**PROGRAMADOR SENIOR:** Persona amb anys d'experiència en el terreny de la programació.

## BIBLIOGRAFIA

---

- **API Java SE 6**  
< <http://java.sun.com/javase/6/docs/api/>>
- **JSP**  
< <http://java.sun.com/products/jsp/>>
- **Spring**  
< <http://www.springsource.org/>>
- **Struts2**  
< <http://struts.apache.org/2.x/>>
- **Hibernate**  
< <http://www.hibernate.org/>>
- **Eclipse**  
< <http://www.eclipse.org/>>
- **UML**  
< <http://www.agilemodeling.com/>>
- **MySQL**  
< <http://www.mysql.com/>>
- **Apache Tomcat**  
< <http://tomcat.apache.org/>>
- **Log4j**  
< <http://www.laliluna.de/log4j-tutorial.html>>

## ANNEXOS

---

### A. Base de dades

---

```
CREATE DATABASE taller;
```

```
USE taller;
```

```
CREATE TABLE client(  
    idClient int(20) not null AUTO_INCREMENT primary key,  
    usuari VARCHAR(20) not null,  
    clau VARCHAR(20) not null,  
    nom VARCHAR(40),  
    nif VARCHAR(20),  
    direccio VARCHAR(40),  
    poblacio VARCHAR(20),  
    telefon VARCHAR(20),  
    marca VARCHAR(20),  
    model VARCHAR(20)  
);
```



























```
CREATE TABLE servei(  
    idServei int(20) not null AUTO_INCREMENT primary key,  
    reparacio VARCHAR(200),  
    data DATE,  
    hora VARCHAR(20)  
);
```

```
GRANT ALL PRIVILEGES ON taller.* TO 'taller'@'localhost'  
IDENTIFIED BY '1234';
```

## B. JARs del projecte

---

JARs dels frameworks Struts2, hibernate i Spring, i de la base de dades MySQL.

 antlr.jar	443.432
 asm-attrs.jar	16.777
 asm.jar	26.360
 cglib.jar	282.338
 commons-collections.jar	559.366
 commons-logging-api-1.1.jar	44.598
 dom4j.jar	313.898
 ehcache-1.2.3.jar	208.048
 ejb3-persistence.jar	50.583
 freemarker-2.3.8.jar	802.494
 hibernate-annotations.jar	279.714
 hibernate-commons-annotations.jar	66.993
 hibernate-core.jar	2.266.769
 hibernate-entitymanager.jar	119.292
 hibernate3.jar	2.766.130
 javassist.jar	471.005
 jta.jar	13.236
 log4j.jar	367.444
 mysql-connector-java-5.1.6-bin.jar	703.265
 ognl-2.6.11.jar	168.081
 slf4j-api-1.5.2.jar	17.384
 slf4j-log4j12.jar	8.620
 spring.jar	1.669.573
 struts2-core-2.0.11.2.jar	2.323.059
 struts2-spring-plugin-2.0.11.2.jar	8.988
 xwork-2.0.5.jar	457.579

## **C Manuals**

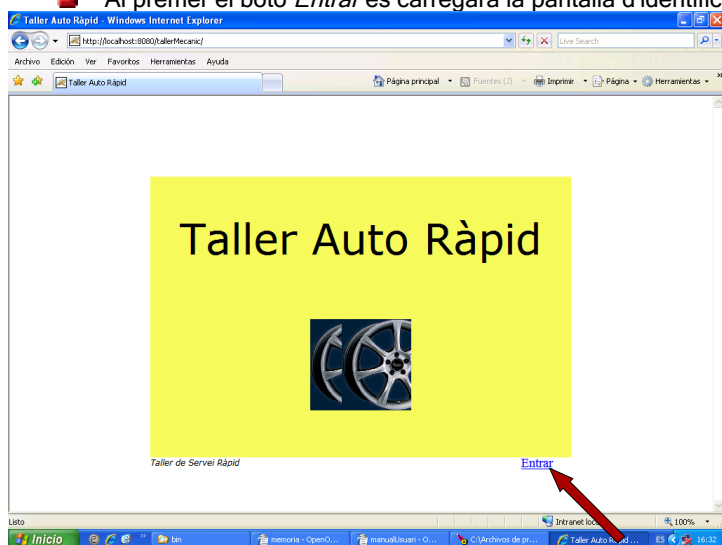
---

**Manual del client i  
Manual del cap de taller**

# MANUAL DEL CLIENT

## CONNECTAR-SE A LA WEB

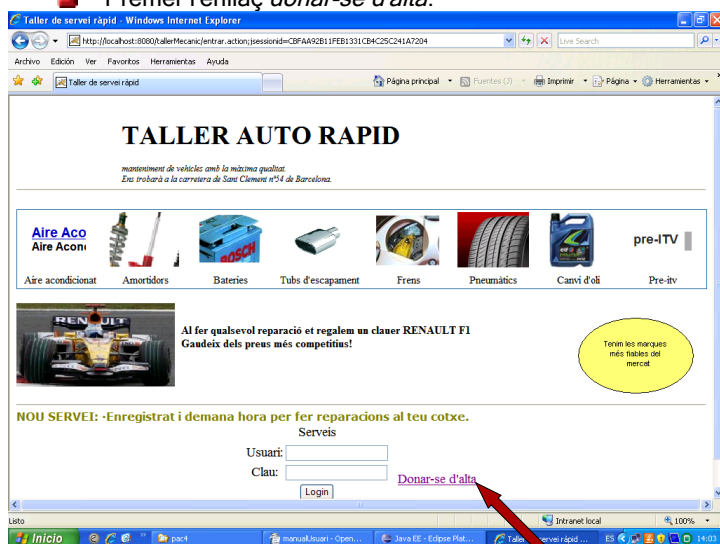
- Obrir un navegador web.
- Posar l'adreça: **http://local host:8080/tallerMecanic/**
- Seguidament es carregarà la pantalla de benvinguda.
- Al prémer el botó *Entrar* es carregarà la pantalla d'identificació.



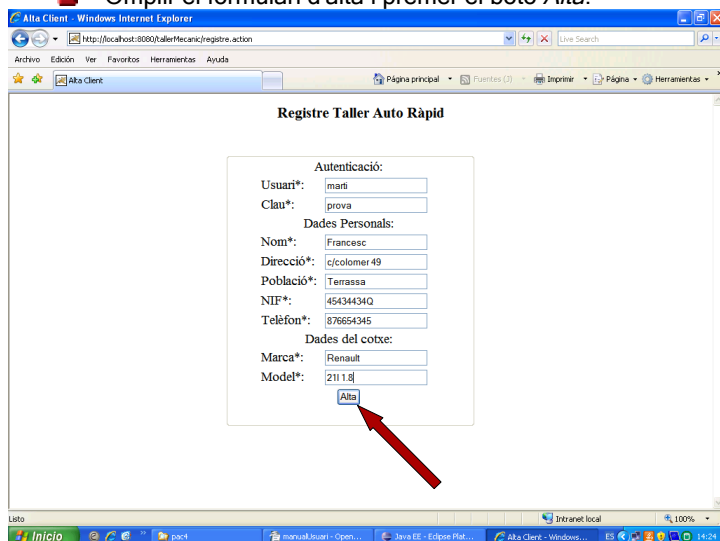
# MANUAL DEL CLIENT

## DONAR-SE D'ALTA

- Connectar-se i entrar a la web.
- Prémer l'enllaç *donar-se d'alta*.



- Omplir el formulari d'alta i prémer el botó *Alta*.



- Avis:** En el cas que es deixi un camp buit es mostrarà un missatge d'error.
- Avis:** Si el sistema no ha realitzat l'operació correctament es mostrarà un missatge d'error
- Avis:** Si s'escull una clau repetida es mostre un missatge d'error.

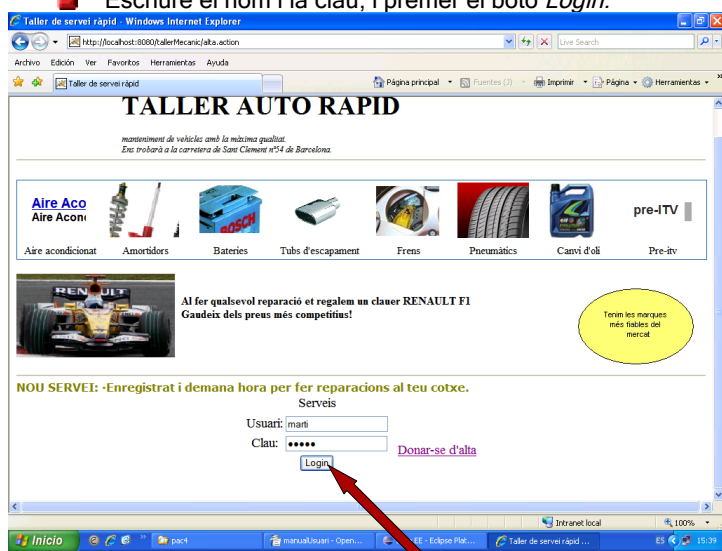


# MANUAL DEL CLIENT

## FER UNA RESERVA

No es poden fer reserves amb data actual.

- Connectar-se i entrar a la web.
- Escriure el nom i la clau, i prémer el botó *Login*.

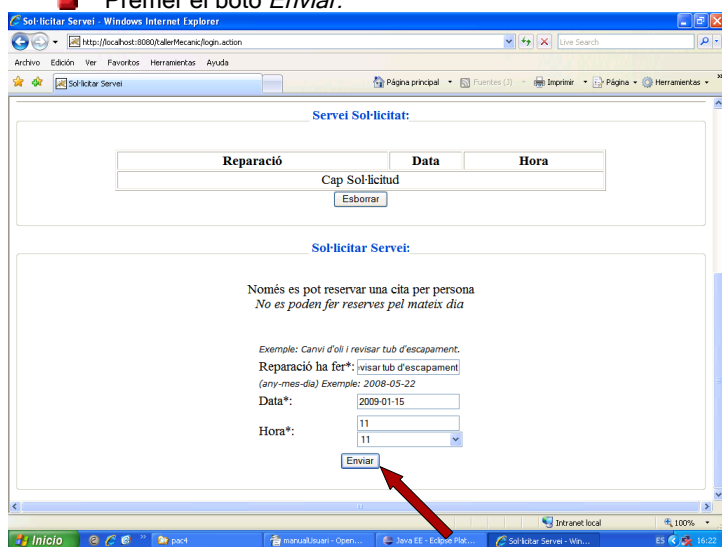


**Avis:** Si s'ha introduït un nom o clau incorrecte es mostrarà un missatge d'error.

**Avis:** En el cas que no es recordi el nom o la clau s'ha de donar-se d'alta de nou.

**Avis:** En el moment d'identificar-se, el sistema comprovarà si el client té una sol·licitud, en cas que estigui caducada l'esborrarà, en cas contrari es veurà al entrar en la pantalla de reserves.

- Omplir els camps: reparació, data, hora.
- Prémer el botó *Enviar*.



# MANUAL DEL CLIENT

**Avis:** Si es deixa un camp buit es mostrarà un missatge d'error.

**Avis:** La data ha de ser del format especificat sinó es mostrarà un missatge d'error.

**Avis:** Si en comptes de seleccionar una hora de la llista, es posa una altra dona error.

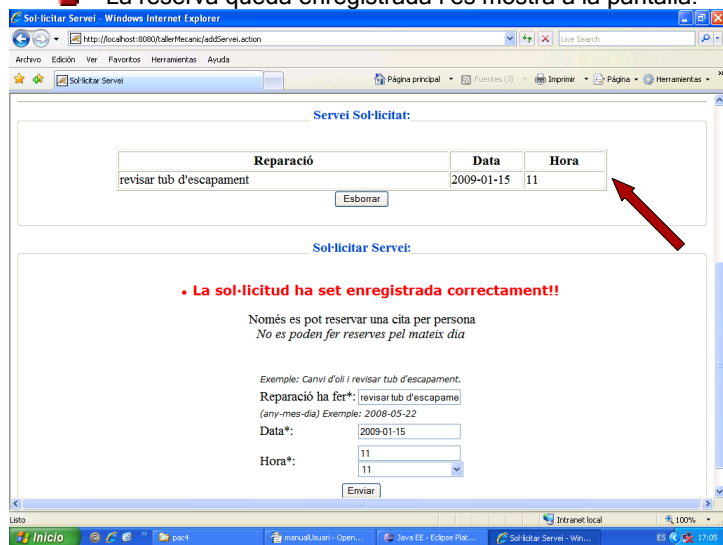
**Avis:** Si es posa una hora que ja està ocupada, es llistaran les hores lliures del dia demanat.

**Avis:** En el cas que s'intenti fer una sol·licitud i ja s'en tingui una, es mostrarà un missatge d'error.

**Avis:** Si l'operació ha anat correctament es mostrarà un missatge que ho indica.

**Avis:** Si la data és l'actual es mostrarà un missatge d'error.

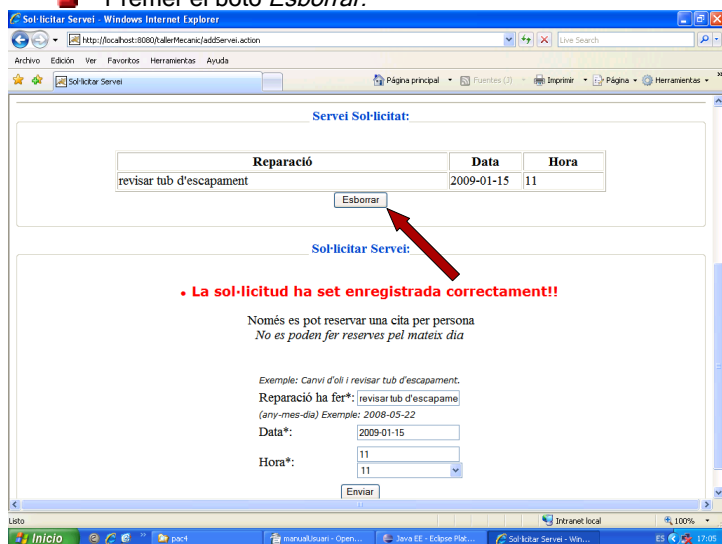
 La reserva queda enregistrada i es mostra a la pantalla.



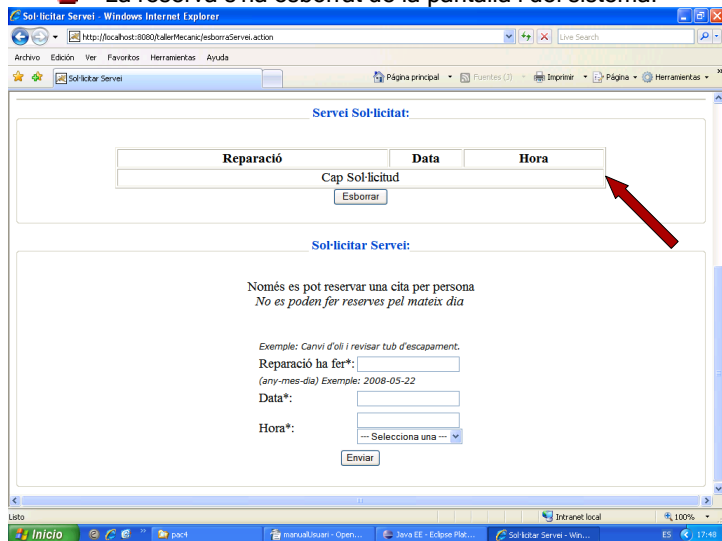
# MANUAL DEL CLIENT

## ESBORRAR RESERVA

- Connectar-se i entra a la web.
- Identificar-se al sistema per entrar a la pantalla de fer reserves.
- Prémer el botó *Esborrar*.



- La reserva s'ha esborrat de la pantalla i del sistema.



# MANUAL DEL CAP DE TALLER

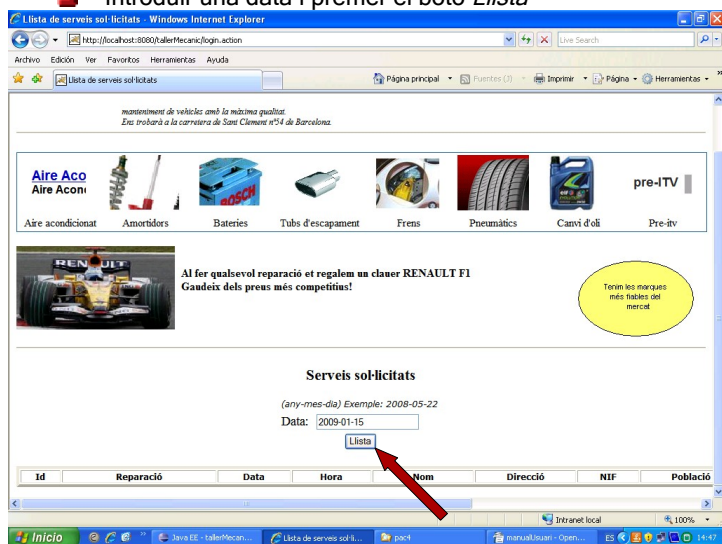
## LLISTAR RESERVES

Només es poden llistar les reserves amb data actual o posterior.

- ▣ Veure primer apartat del manual del client *connectar-se a la web*
- ▣ Identificar-se amb el nom d'usuari **cap** i amb la clau **890**
- ▣ Prémer el botó **Login**



- ▣ Introduir una data i prémer el botó **Llista**



**Avís:** En el cas que no hi hagi cap reserva pel dia seleccionat, es mostra un missatge que ho indica.

**Avís:** Si la data no té el format especificat, es mostra un missatge d'error.

# MANUAL DEL CAP DE TALLER

Es mostren les dades de les reserves.

Es mostren les dades de les reserves.

Llista de serveis sol·licitats - Windows Internet Explorer

http://localhost:8080/tallerMecanic/Lista.action

Aire Aco  
Aire Acon

Amortidors

Bateries

Tubs d'escapament

Frens

Pneumàtics

Canvi d'oli

Pre-ITV

Al fer qualsevol reparació et regalem un clauer RENAULT F1  
Gaudeix dels preus més competitius!

Torim les marques més fiables del mercat.

Serveis sol·licitats

(any-mes-dia) Exemple: 2008-05-22

Data: 2009-01-15

Llista

Id	Reparació	Data	Hora	Nom	Direcció	NIF	Pob
28	revisar tub d'escapament	2009-01-15	11	Francesc	c/colomer 49	45434434Q	Terrassa

**Avis:** Si s'intenta llistar reserves amb data anterior a l'actual, surt un missatge d'error.