



Implementació d'una Intranet pel Restaurant “Cal Fesu”

Lluís Manel Maldonado Rubio
ETIG

José Juan Rodríguez

18 de Juny de 2007



A la meva estimada dona Núria, pel seu encoratjament i suport en tot el que em proposo.

A Jose Juan Rodriguez per la seva professionalitat i inestimable ajuda.



Resum

Aquest projecte consisteix en el disseny i implementació d'una intranet amb zona privada pel restaurant "Cal Fesu", utilitzant una tecnologia puntera com és J2EE (struts).

En la nostra aplicació, l'administrador té accés a la zona privada i pot introduir el menú del dia, les novetats de la carta, les suggerències i les receptes. Els clients poden veure des de la zona pública el menú del dia, la carta, les suggerències i les receptes. També poden subscriure's, omplir un formulari, a un butlletí de receptes que rebran per e-mail.

És una aplicació útil i funcional, que es pot fer servir en el món real i que m'ha permès introduir-me en el món del J2EE.

L'aplicació implementa el model MVC (Model-Vista-Controlador), fent servir un dels framework més utilitzats, com és Jakarta Struts.

Tenim una aplicació escalable, ja que per accedir a la BBDD fa servir un pool de connexions basat en el framework Jakarta struts, i com a servidor de J2EE usa Apache-Tomcat.

Aquesta memòria presenta doncs, un treball final de carrera i recull totes les etapes del projecte, pla de treball, anàlisi, disseny i implementació. També conté un manual d'instal·lació.



INDEX

1	Introducció.....	pàg. 6
1.1	Justificació.....	pàg. 6
1.2	Objectius.....	pàg. 6
1.2.1	Objectius Generals.....	pàg. 6
1.2.2	Objectius específics.....	pàg. 7
1.3	Metodologia.....	pàg. 7
1.4	Planificació.....	pàg. 7
1.4.1	Estratègia.....	pàg. 7
1.4.2	Dates significatives	pàg. 8
1.4.3	Tasques i Temporalització	pàg. 8
1.4.3.1	PAC -1 12 / 03 / 07 – (Definició del TFC + Planificació) ..	pàg. 8
1.4.3.2	PAC2 – 16 / 04 / 07 – (Anàlisi + Disseny)	pàg. 8
1.4.3.3	PAC3 – 21 / 05 / 07 – (Implementació)	pàg. 8
1.4.3.4	Lliurament Final del TFC - 18 / 06 / 07	pàg. 9
1.4.4	Diagrama de Gannt.....	pàg. 9
1.5	Productes Obtinguts	pàg. 9
1.6	Descripció del capítols següents.....	pàg.10
2	Software Utilitzat.....	pàg.10
3	Arquitectura de l'aplicació	pàg.11
3.1	Capa M (Model)	pàg.12
3.2	Capa V (Vista)	pàg.12
3.3	Capa C (Controlador)	pàg.12
4	Especificacions i anàlisi	pàg.13
4.1.1	Casos d'ús del actor Client.....	pàg.13
4.1.2	Casos d'ús del actor Administrador.....	pàg.14
4.1.3	Diagrama de casos d'us.....	pàg.17
5	Disseny de la interfície gràfica.....	pàg.19
6	Disseny de la bbdd.....	pàg.37
7	Implementació.....	pàg.38
7.1	Model	pàg.38
7.1.1	Connexions a la BBDD – Pool de Connexions.	pàg.39
7.1.2	Java Mail.....	pàg.40
7.2	Vista	pàg.41
7.3	Controlador.....	pàg.44
8	Proves.....	pàg.47
9	Estructura de l'aplicació.....	pàg.47
10	Funcionalitats implementades de l'aplicació.....	pàg.52
10.1	Funcionalitats Implementades Zona Pública.....	pàg.52
10.2	Funcionalitats Implementades Zona Privada.....	pàg.52
11	Conclusions.....	pàg.53
12	Manual d'instal·lació.....	pàg.54
12.1	Per a poder accedir a la bbdd cal instal·lar el mysql	pàg.54
12.2	Instal·lació de mysqlcc.....	pàg.54
12.3	Creació de la BBDD.....	pàg.54
12.4	Instal·lació tomcat	pàg.54
13	Millores i manteniment futur	pàg.56
14	Bibliografia.....	pàg.57



INDEX DE FIGURES

Figura 1:	Diagrama de Gannt.....	pàg.9
Figura 2:	Diagrama funcionament struts.	pàg.11
Figura 3:	Cas d'ús client.	pàg.17
Figura 4:	Cas d'ús Administrador.	pàg.18
Figura 5:	Pantalla inici aplicació.	pàg.19
Figura 6:	Pantalla consultar receptes.	pàg.20
Figura 7:	Pantalla consultar receptes detall.	pàg.20
Figura 8:	Pantalla consultar suggerències del dia.	pàg.21
Figura 9:	Pantalla subscripció al receptari.	pàg.21
Figura 10:	Pantalla error subscripció al receptari.	pàg.22
Figura 11:	Pantalla inici carta.	pàg.22
Figura 12:	Pantalla primers plats de la carta.	pàg.23
Figura 13:	Pantalla segons plats de la carta.	pàg.23
Figura 14:	Pantalla postres de la carta.	pàg.24
Figura 15:	Pantalla inici zona privada.	pàg.25
Figura 16:	Pantalla inici gestió menús.	pàg.26
Figura 17:	Pantalla alta menú diari.	pàg.26
Figura 18:	Pantalla modificació menú diari.	pàg.27
Figura 19:	Pantalla consultar històric menús.	pàg.27
Figura 20:	Pantalla consultar històric menús detall.	pàg.28
Figura 21:	Pantalla inici gestió receptes.	pàg.28
Figura 22:	Pantalla alta receptes.	pàg.29
Figura 23:	Pantalla consultar històric receptes.	pàg.29
Figura 24:	Pantalla consultar històric receptes detall.	pàg.30
Figura 25:	Pantalla inici gestió suggerències.	pàg.30
Figura 26:	Pantalla alta suggerències.	pàg.31
Figura 27:	Pantalla modificació suggerències del dia.	pàg.31
Figura 28:	Pantalla consultar històric suggerències.	pàg.32
Figura 29:	Pantalla consultar històric suggerències detall.	pàg.32
Figura 30:	Pantalla inici gestió clients.	pàg.33
Figura 31:	Pantalla consultar històric clients.	pàg.33
Figura 32:	Pantalla consultar històric clients detall.	pàg.34
Figura 33:	Pantalla inici gestió carta.	pàg.34
Figura 34:	Pantalla modificació primers plats de la carta.	pàg.35
Figura 35:	Pantalla modificació segons plats de la carta.	pàg.35
Figura 36:	Pantalla modificació postres de la carta.	pàg.36
Figura 37:	Disseny BBDD de l'aplicació.	pàg.37
Figura 38:	Algunes de les accions utilitzades en el projecte.	pàg.39
Figura 39:	Datasource del arxiu struts-config.xml.	pàg.40
Figura 40:	Extracte mètode enviar e-mail.	pàg.41
Figura 41:	Declaració de Forms estàtics a l'arxiu struts-config.xml.	pàg.42
Figura 42:	Declaració de Forms dinàmics a l'arxiu struts-config.xml.	pàg.42
Figura 43:	Declaració del fitxer de recursos a l'arxiu struts-config.xml.	pàg.42
Figura 44:	Detall de etiquetes a la JSP.	pàg.43
Figura 45:	Value Objects utilitzats en el projecte.	pàg.43
Figura 46:	Classe ActionServlet.	pàg.44
Figura 47:	Funcionament de la classe ActionServlet.	pàg.45
Figura 48:	Declaració d'un plugin al struts-config.xml.	pàg.46
Figura 49:	Detall accions de l'arxiu struts-config.xml.	pàg.46
Figura 50:	Detall estructura de directoris de l'aplicació.	pàg.48
Figura 51:	Detall arxiu web.xml.	pàg.50
Figura 52:	Detall build path.	pàg.51
Figura 53:	Pantalla tomcat.	pàg.55
Figura 54:	Pantalla tomcat manager.	pàg.55



1 Introducció

1.1 Justificació

J2EE permet desenvolupar aplicacions, simplificant la complexitat i els costos. Té un cos reduït per les empreses, ja que es tracta de programari lliure i, cada cop, hi ha més empreses que fan servir tecnologia d'aquest tipus. Aquest TFC apareix com a necessitat de moltes empreses per a modernitzar la seva imatge i tenir presència a Internet.

1.2 Objectius

1.2.1 Objectius Generals

L'objectiu principal del projecte ha estat introduir-me en el desenvolupament d'aplicacions J2EE (investigar i posar-me al dia amb aquesta tecnologia): conèixer de prop les JSP, els Servlets; fer servir un framework (Struts); conèixer i utilitzar el patró MVC (Model-Vista-Controlador) i fer servir com a servidor J2EE Apache-Tomcat. Tot això, creant una aplicació útil i funcional, que es pugui fer servir en el món real.

- Utilització del IDE Eclipse.
- Creació de pàgines dinàmiques JSP.
- Utilització de servlets.
- Utilització d'un framework (Struts).
- Interacció amb un sistema gestor de BBDD mysql.
- Servidor de J2EE Apache-Tomcat.

En concret, volem desenvolupar una intranet amb zona privada per un restaurant, on el xef tingui accés a la zona privada i pugui introduir diàriament el menú diari, les novetats de la carta i les suggerències. Els clients podran veure el menú del dia i podran consultar la carta i els preus. També s'inclourà un butlletí de receptes publicades, amb possibilitat de subscripció per part dels clients.



Objectius específics

La intranet del restaurant tindrà una zona privada on, un cop validat l'usuari i contrasenya, el xef podrà:

- Accedir a la “Secció menú”:
El xef podrà publicar el menú diari.
- Accedir a la “Secció suggerències”:
El xef podrà publicar les suggerències que vulgui.
- Accedir a la “Secció receptes”:
En aquesta secció el xef publicarà receptes, que automàticament seran enviades via e-mail als clients que estiguin subscrits.
- Accedir a la “Secció de gestió de clients”:
Els clients subscrits al receptari podran consultar aquesta secció.

La zona pública de la intranet serà d'accés lliure, amb publicitat del restaurant, on es podrà consultar el menú diari, les suggerències del dia, la carta i les últimes receptes publicades. Els clients que ho desitgin, també podran omplir un formulari per a subscriure's al receptari, que rebran via e-mail, cada cop que es doni d'alta una nova recepta.

1.3 Metodologia

Per poder dur a terme el projecte amb èxit, hem dividit el treball en diverses fases, d'acord amb el cicle de vida clàssic. Es farà una planificació detallada fent servir el MS Project.

1.4 Planificació

1.4.1 Estratègia

Per tal de planificar el Treball, s'ha fet servir l'eina MS Project: s'ha decidit considerar una dedicació d'una hora diària al TFC, de dilluns a divendres, i 4 hores dissabtes i diumenges.

Donat que partim amb **zero** coneixements sobre aplicacions J2EE – Struts, és molt probable que aquesta planificació sigui modificada sobre la marxa, en funció de les necessitats.



1.4.2 Dates significatives

- 28 / 02 / 07 – Inici de l'assignatura.
- 12 / 03 / 07 – Lliurament de la PAC1 (Definició del TFC + Planificació).
- 16 / 04 / 07 – Lliurament de la PAC2 (Anàlisi + Disseny).
- 21 / 05 / 07 – Lliurament de la PAC3 (Implementació).
- 18 / 06 / 07 – Lliurament del TFC (Memòria + presentació virtual).

1.4.3 Tasques i Temporalització

1.4.3.1 PAC –1 12 / 03 / 07 – (Definició del TFC + Planificació)

Descripció del TFC:

- Justificació.
- Documentar-se en J2EE-Struts.
- Objectius generals i específics.
- Identificació del programari a utilitzar.
- Identificació de les funcionalitats de la intranet.
- Estudi etapes del projecte.
- Planificació amb fites i temporalització.
- Creació MS Project.
- Documentació.

1.4.3.2 PAC2 – 16 / 04 / 07 – (Anàlisi + Disseny)

- Instal·lació de software.
- Estudi del Model MVC.
- Documentar-se en J2EE-Struts.
- Identificació de classes.
- Diagrama estàtic.
- Casos d'ús.
- Anàlisi de la interfície.
- Disseny interfície.
- Documentació.

1.4.3.3 PAC3 – 21 / 05 / 07 – (Implementació)

- Creació BBDD amb mysql.
- Implementació.
- Testing.
- Documentació.

1.4.3.4 Lliurament Final del TFC - 18 / 06 / 07

- Creació de la Memòria.
- Presentació virtual.

1.4.4 Diagrama de Gannt

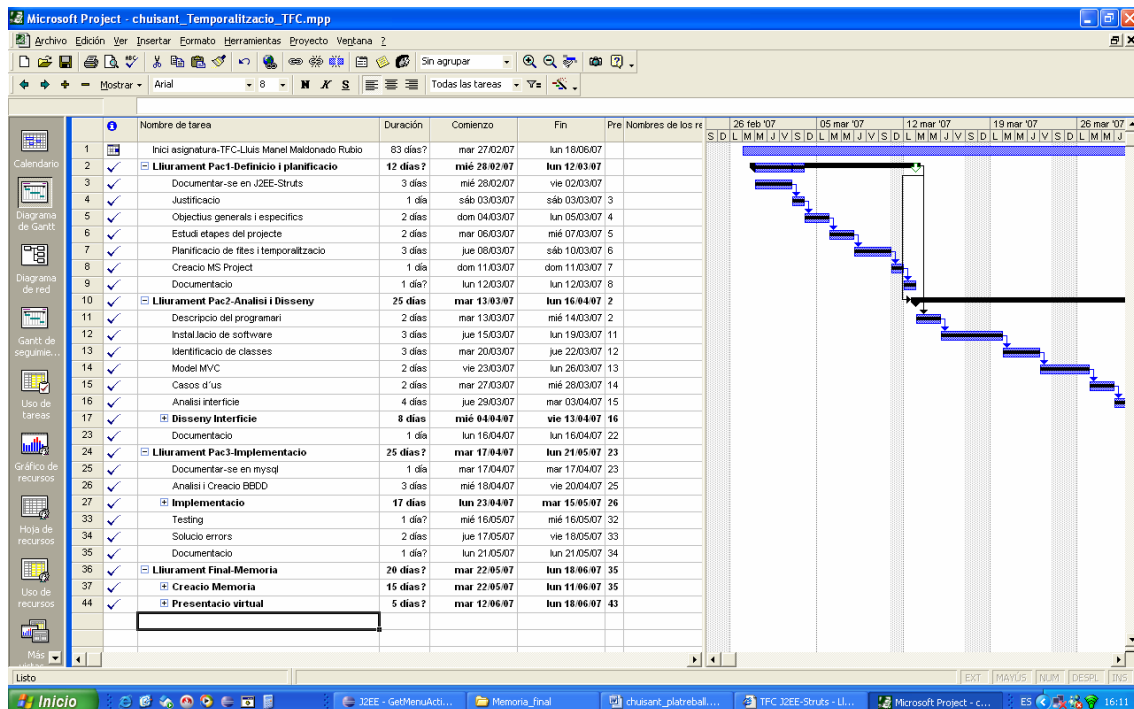


Figura 1: Diagrama de Gannt.

1.5 Productes Obtinguts

Durant la realització d'aquest projecte s'ha anat obtenint diferents productes, fruit del treball en cada etapa, aquest documents i programes són :

- Document del pla de treball.
- Document MS Project amb la planificació i temporalització del projecte.
- Document de anàlisi de requeriments.
- Document del disseny.
- La intranet amb zona privada d'un restaurant.
- Manual d'instal·lació de l'aplicació.
- Document de la memòria.
- Presentació en PowerPoint.



1.6 Descripció del capítols següents

- Software utilitzat.
- Arquitectura de l'aplicació.
- Especificacions i Anàlisi.
- Disseny de la interfície gràfica.
- Disseny de la bbdd.
- Implementació (Model-Vista-Controlador).
- Proves.
- Estructura de l'aplicació.
- Funcionalitats implementades de l'aplicació.
- Conclusions.
- Manual d'instal·lació.
- Millores i manteniment futur.
- Bibliografia.

2 Software Utilitzat

Software utilitzat en aquest projecte

- Instal·lació del IDE Eclipse 3.2.
- Instal·lació de la BBDD mysql 5.2.
- Instal·lació de mysql Control Center 0.9.4.
- Instal·lació del servidor de J2EE Apache-Tomcat 5.5.
- Instal·lació del driver JDBC per mysql.
(org.gjt.mm.mysql.Driver del mysql-connector-java-5.0.5-bin.jar)

3 Arquitectura de l'aplicació

Farem servir el patró de disseny MVC model 2, tindrem 3 capes en la nostra arquitectura totalment independents, amb totes les avantatges que això comporta de cara al manteniment. Tot seguit, explicarem com implementar cadascuna d'elles:

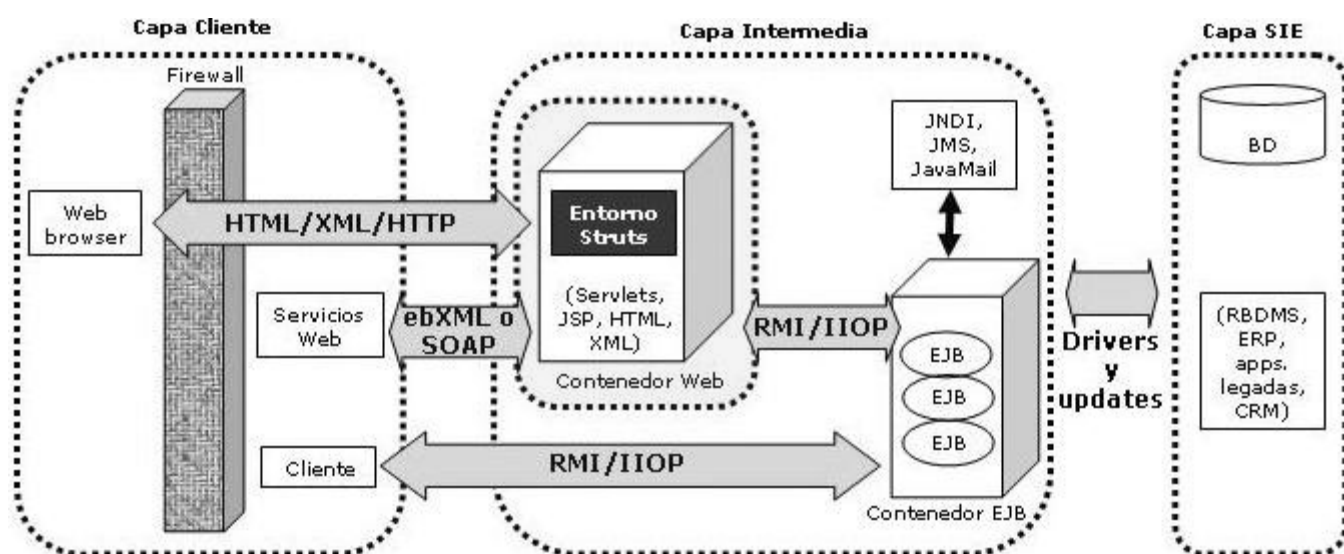


Figura 2: Diagrama funcionament struts.

Struts fa servir diferents àrees de recursos compartits per emmagatzemar objectes:

- Petició (*javax.servlet.http.HttpServletRequest*).
- Sessió (*javax.servlet.http.HttpSession*).
- Aplicació (*javax.servlet.ServletContext*).
- Pàgina (*javax.servlet.jsp.PageContext*).

Nosaltres farem servir la "request" per guardar objectes, que després necessitarem.



3.1 Capa M (Model)

És la capa de que conté la lògica del negoci i l'accés a les dades. Accedirem a la BBDD (mysql) amb el driver JDBC per mysql, i farem servir el patró de disseny "Value Objects" (VO) per retornar les dades. Utilitzarem un framework per a l'accés a bdd i implementem un pool de connexions amb la interface "DataSource".

3.2 Capa V (Vista)

La implementació la farem amb el framework Jakarta struts, amb les JSP utilitzarem: Struts Tags (struts-tiles, struts-html, struts-logic, struts-bean) etc.

3.3 Capa C (Controlador)

El controlador farà d'intermediari entre la capa Vista i la capa Model. Gestiona les entrades o accions de l'usuari. Estarà implementat amb un servlet propi del framework (struts) el "ActionServlet", que fa servir el fitxer de configuració struts-config.xml.



4 Especificacions i anàlisi

Els actors de la nostra aplicació i anàlisi són dos: els clients i els administradors del restaurant.

Els clients poden consultar la informació, i els administradors gestionen i mantenen aquesta informació. Els administradors tenen accés a les funcionalitats dels clients, a més de les seves específiques.

4.1.1 Casos d'ús del actor Client

Cas d'ús: Consulta menú

- Funcionalitat: el client pot consultar el menú del dia en curs.
- Actors: client.
- Precondicions: el menú del dia ha d'estar donat d'alta, sinó sortirà com a "no disponible".
- Postcondicions: cap.

Cas d'ús: Consulta recepta

- Funcionalitat: el client pot consultar les últimes cinc receptes publicades.
- Actors: client.
- Precondicions: les receptes han d'estar donades d'alta.
- Postcondicions: cap.

Cas d'ús: Consulta suggerències

- Funcionalitat: el client pot consultar les suggerències del dia en curs.
- Actors: client.
- Precondicions: les suggerències del dia han d'estar donades d'alta, sinó sortirà com a "no disponible".
- Postcondicions: cap.

Cas d'ús: Subscripció receptari

- Funcionalitat: els clients que vulguin rebre el butlletí de receptes podran omplir un formulari i subscriure's al receptari.
- Actors: client.
- Precondicions: el client ha de tenir un e-mail en el qual per poder-li enviar les receptes.
- Postcondicions: el client ha estat donat d'alta a la subscripció del receptari. El registre s'ha donat d'alta a la bbdd.

Cas d'ús: Consulta primers plats de la carta

- Funcionalitat: el client pot consultar els primers plats de la carta.
- Actors: client



- Precondicions: cap.
- Postcondicions: cap.

Cas d'ús: Consulta segons plats de la carta

- Funcionalitat: el client pot consultar els segons plats de la carta.
- Actors: client.
- Precondicions: cap.
- Postcondicions: cap.

Cas d'ús: Consulta postres de la carta

- Funcionalitat: el client pot consultar els postres de la carta.
- Actors: client.
- Precondicions: cap.
- Postcondicions: cap.

4.1.2 Casos d'ús de l'actor Administrador

Cas d'ús: Accés a la zona privada

- Funcionalitat: el administrador pot accedir a la zona privada.
- Actors: administrador.
- Precondicions: el administrador ha de conèixer el seu mail i el seu password i aquests han d'estar donats d'alta al sistema.
- Postcondicions: cap.

Cas d'ús: Alta Menú del dia

- Funcionalitat: el administrador pot donar d'alta el menú del dia en curs.
- Actors: administrador.
- Precondicions: el menú del dia no ha d'estar donat d'alta.
- Postcondicions: el menú del dia estar donat d'alta.

Cas d'ús: Modificació Menú del dia

- Funcionalitat: l'administrador pot modificar el menú del dia en curs.
- Actors: administrador.
- Precondicions: el menú del dia ha d'estar donat d'alta.
- Postcondicions: el menú ha estat modificat.

Cas d'ús: Consulta Històric de Menús

- Funcionalitat: el administrador pot consultar els cinc últims menús, o buscar el menú d'un dia en concret.
- Actors: administrador.
- Precondicions: els menús han d'existir.
- Postcondicions: cap.



Cas d'ús: Alta receptes

- Funcionalitat: l'administrador pot donar d'alta una recepta, omplin el formulari, al prémer el boto també s'enviarà als subscriptors un e-mail amb la recepta.
- Actors: administrador.
- Precondicions: cap.
- Postcondicions: la recepta ha estat donada d'alta i s'ha enviat per e-mail als subscriptors.

Cas d'ús: Consulta Històric de receptes

- Funcionalitat: l' administrador pot consultar les últimes cinc receptes publicades o pot buscar una en concret pel nom, i es mostraran totes les que coincideixin amb el nom fins a una màxim de cinc.
- Actors: administrador.
- Precondicions: les receptes han d'existir.
- Postcondicions: cap.

Cas d'ús: Alta suggerències del dia

- Funcionalitat: el administrador pot donar d'alta les sugerències del dia en curs.
- Actors: administrador.
- Precondicions: les suggerències del dia no han d'estar donades d'alta.
- Postcondicions: les suggerències del dia estan donades d'alta.

Cas d'ús: Modificació suggerències del dia

- Funcionalitat: l'administrador pot modificar les suggerències del dia en curs.
- Actors: administrador.
- Precondicions: les suggerències del dia han d'estar donades d'alta.
- Postcondicions: les suggerències han estat modificades.

Cas d'ús: Consulta Històric de Suggerències

- Funcionalitat: l'administrador pot consultar les suggerències dels cinc últims dies, o buscar les sugerències d'un dia en concret.
- Actors: administrador.
- Precondicions: les suggerències han d'existir.
- Postcondicions: cap.

Cas d'ús: Consulta Històric de Clients

- Funcionalitat: l'administrador pot consultar els cinc últims clients que s'han subscrit al receptari, o buscar un client en concret, fent la recerca a partir del seu e-mail.
- Actors: administrador.
- Precondicions: els clients han d'existir.
- Postcondicions: cap.



Cas d'ús: Modificació dels primers plats de la carta

- Funcionalitat: l'administrador pot modificar els noms i els preus dels primers plats de la carta.
- Actors: administrador.
- Precondicions: els primers plats han d'existir.
- Postcondicions: s'ha modificat els primers plats de la carta.

Cas d'ús: Modificació dels segons plats de la carta

- Funcionalitat: l'administrador pot modificar els noms i els preus dels segons plats de la carta.
- Actors: administrador.
- Precondicions: els segons plats han d'existir.
- Postcondicions: s'han modificat els segons plats de la carta.

Cas d'ús: Modificació dels postres de la carta

- Funcionalitat: l'administrador pot modificar els noms i els preus dels postres de la carta.
- Actors: administrador.
- Precondicions: els postres han d'existir.
- Postcondicions: s'han modificat els postres de la carta.

Cas d'ús: Sortir de la zona privada

- Funcionalitat: l'administrador pot sortir de la zona privada.
- Actors: administrador.
- Precondicions: l'administrador ha d'estar dins de la zona privada.
- Postcondicions: l'administrador ha sortit de la zona privada.

4.1.3 Diagrama de casos d'us

Diagrames de casos d'us realitzats amb l'eina: **UML Magic Draw 11.0**

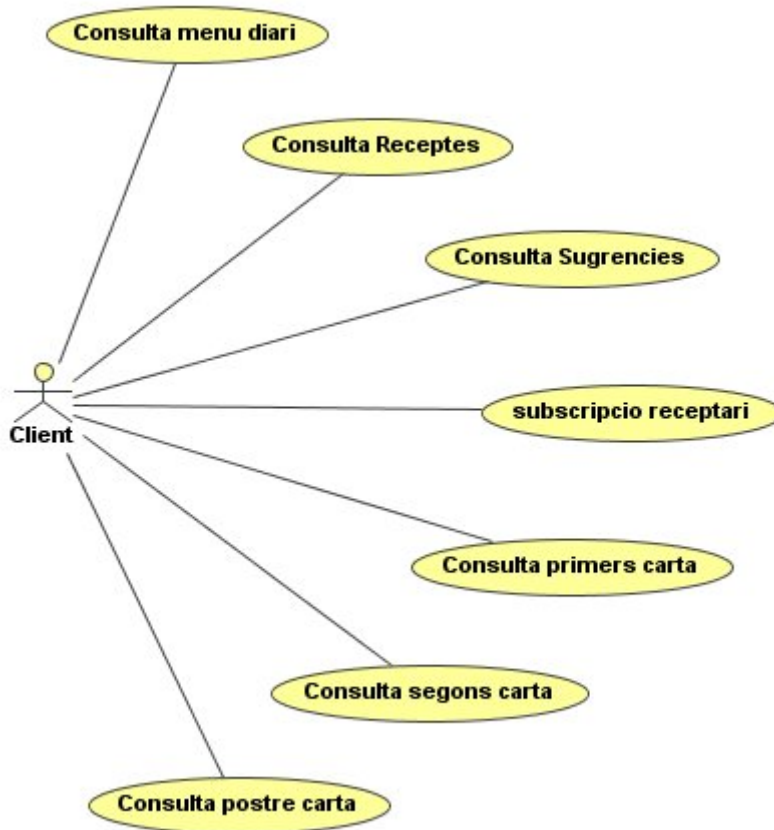


Figura 3: Cas d'us client.

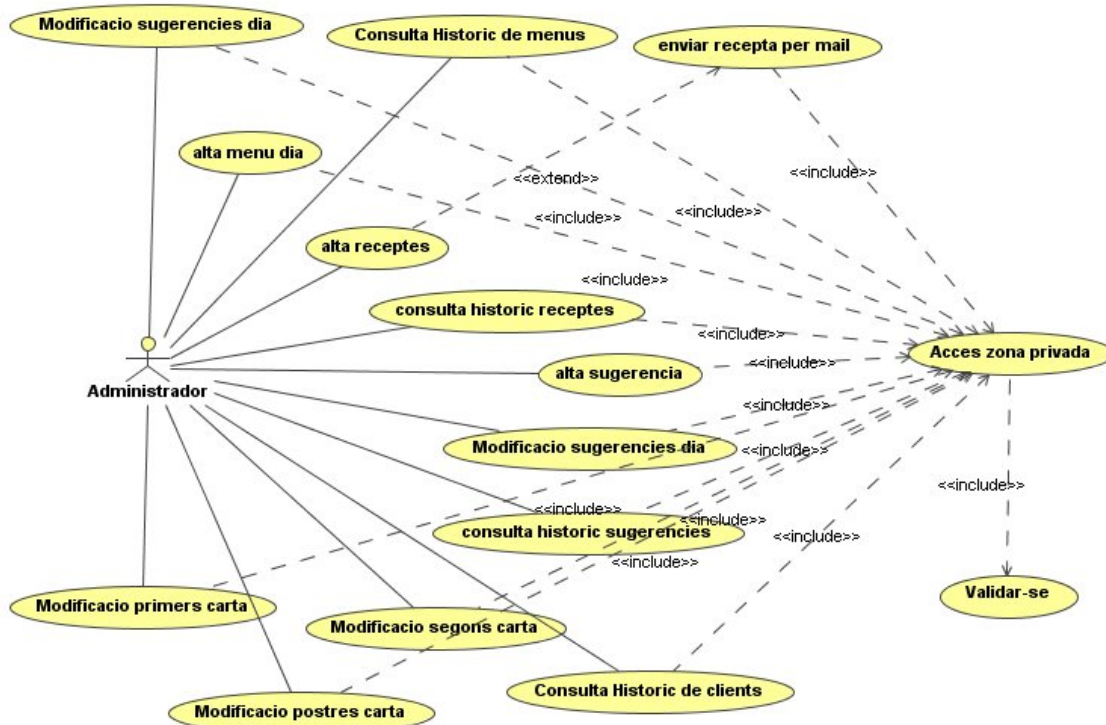


Figura 4: Cas d'ús Administrador.



5 Disseny de la interfície gràfica.

5.1 PANTALLES ZONA PÚBLICA

5.1.1 Pantalla inicial del Restaurant

A la pantalla inicial podem veure el menú del dia, i al menú de l'esquerra les opcions disponibles de la zona pública. També podem validar-nos i entrar a la zona privada.



Figura 5: Pantalla inici aplicació.

Opció menú del dia de la zona publica:

Mostra el menú del dia vigent (es carrega a la pàgina inicial). Si el menú no està donat d'alta surt "no disponible".



5.1.2 Pantalles Receptes

Opció receptes de la zona pública:
Mostra les últimes receptes elaborades.



Figura 6: Pantalla consulta receptes.

Si fem clic al link mostrarà la descripció:

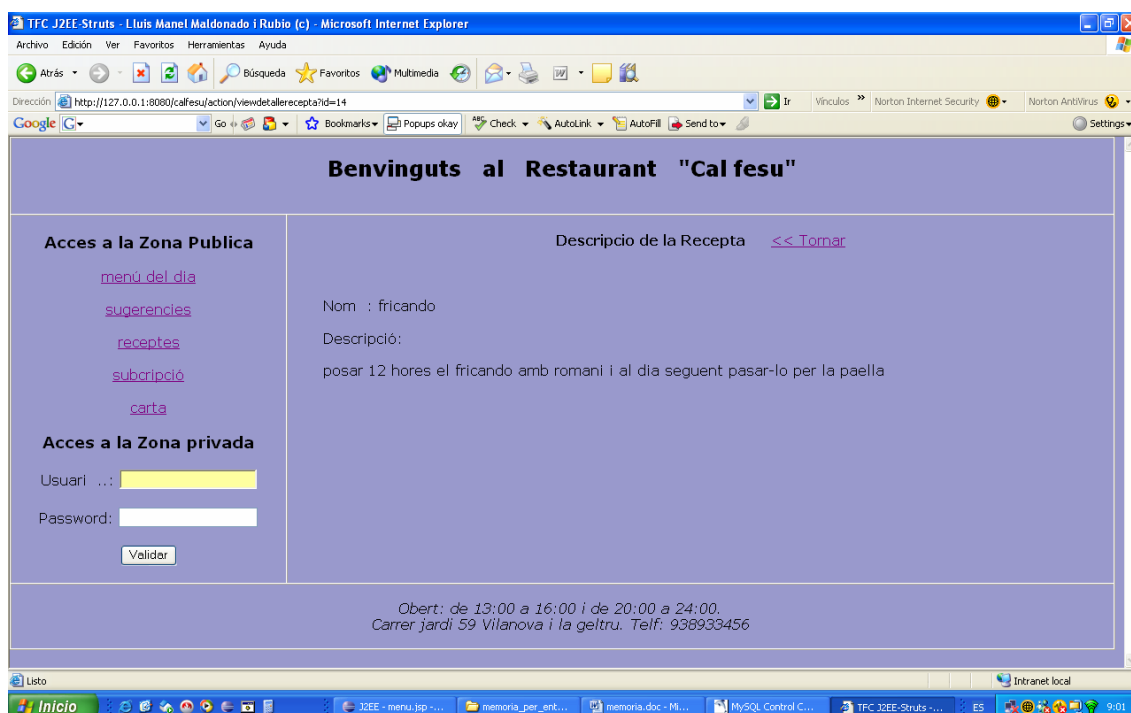


Figura 7: Pantalla consulta receptes detall.



5.1.3 Pantalla Suggerències

Opció suggerències de la zona pública: Mostra les suggerències del dia.



Figura 8: Pantalla consulta suggerències del dia.

5.1.4 Pantalla Subscripció

Opció subscripció de la zona pública :

Mostra un formulari, on el client es pot donar d'alta i subscriure's al butlletí de receptes que s'envia per mail.

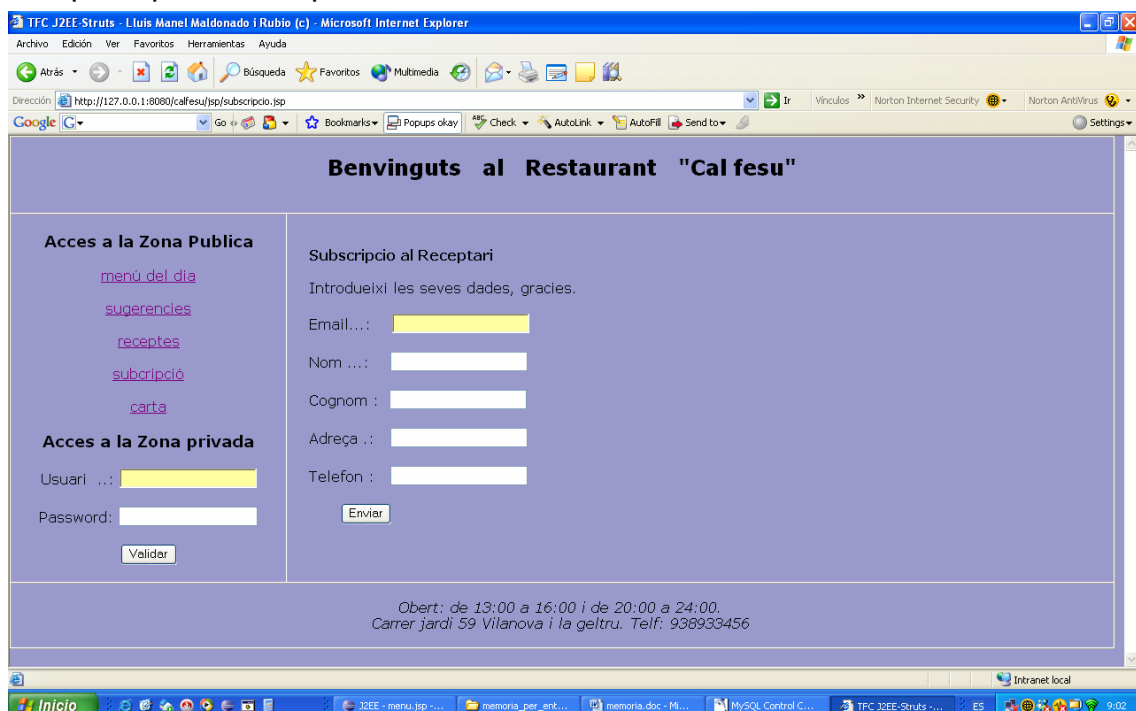




Figura 9: Pantalla subscripció al receptari.
Si les dades son errònies o el mail ja existeix a la BBDD donarà error:



Figura 10: Pantalla error subscripció al receptari.

5.1.5 Pantalles Carta

Opció carta de la zona publica:

Mostra la carta dels primers i segons plats i dels postres, i els preus.



Figura 11: Pantalla inici carta.



Podem escollir si volem consultar els primers, segons plats, o les postres. La carta constarà de 6 primers, 6 segons i 6 postres.

Consultar primers plats de la carta

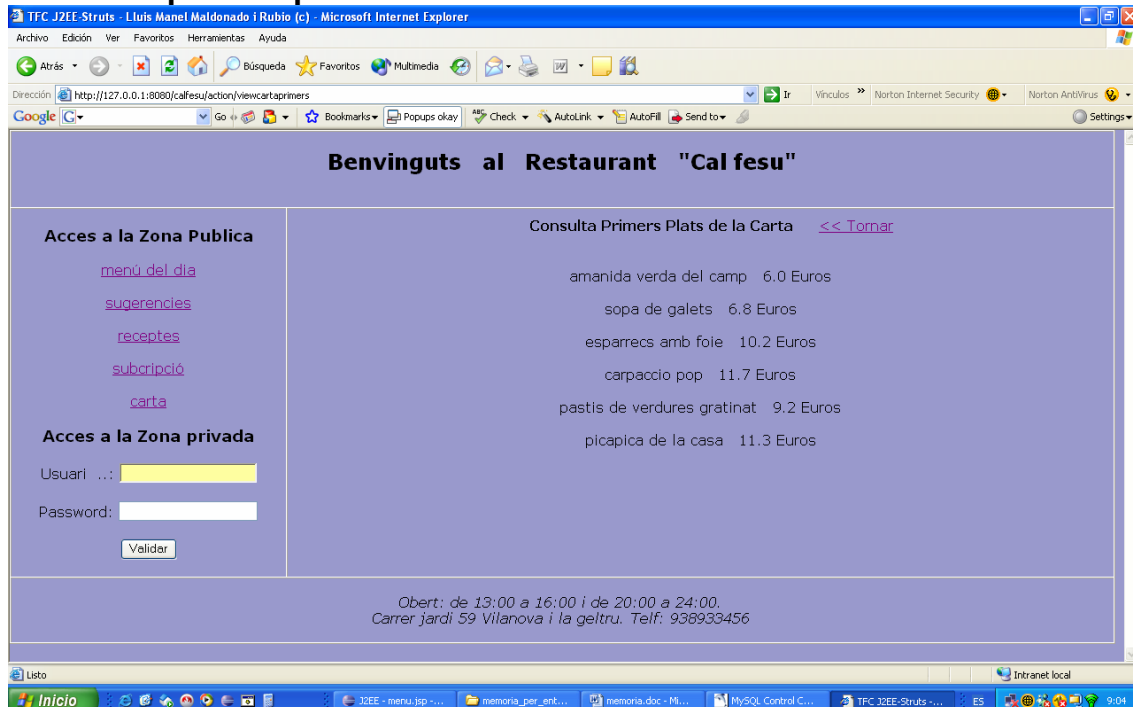


Figura 12: Pantalla primers plats de la carta.

Consultar segons plats de la carta

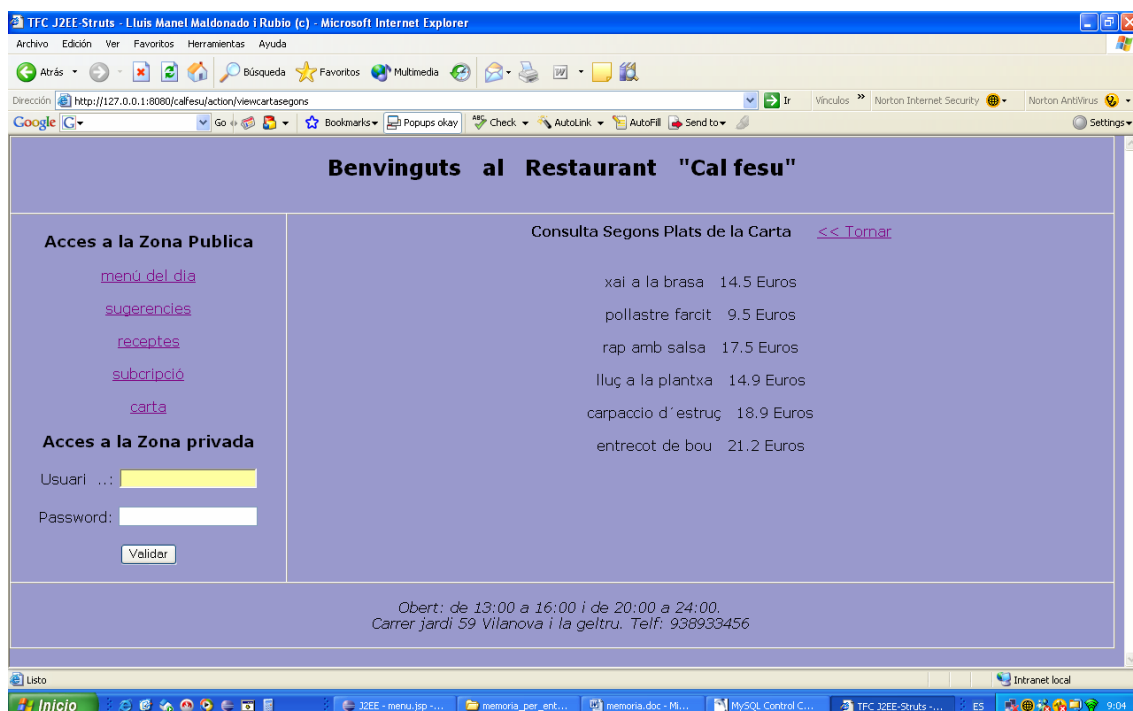


Figura 13: Pantalla segons plats de la carta.



Consultar les postres



Figura 14: Pantalla de les postres de la carta.

5.2 PANTALLES ZONA PRIVADA

Per accedir a la zona privada cal posar un usuari i password autoritzat, per exemple:

Usuari: lluis@lluis.com

Password: lluis

Quan s'accedeix a la zona privada surt la pantalla de benvinguda:



Figura 15: Pantalla inici zona privada.

A la part esquerra tenim el menú de la zona privada:

Opció sortir:

A sota de tot tenim aquesta opció per tornar a la pàgina inicial del restaurant i sortir de la zona privada.



5.2.1 Pantalles Menú

Gestió de menús



Figura 16 : pantalla inici gestió menús

Alta menú diari

El menú constarà de 3 primers i 3 segons plats, i 3 postres.

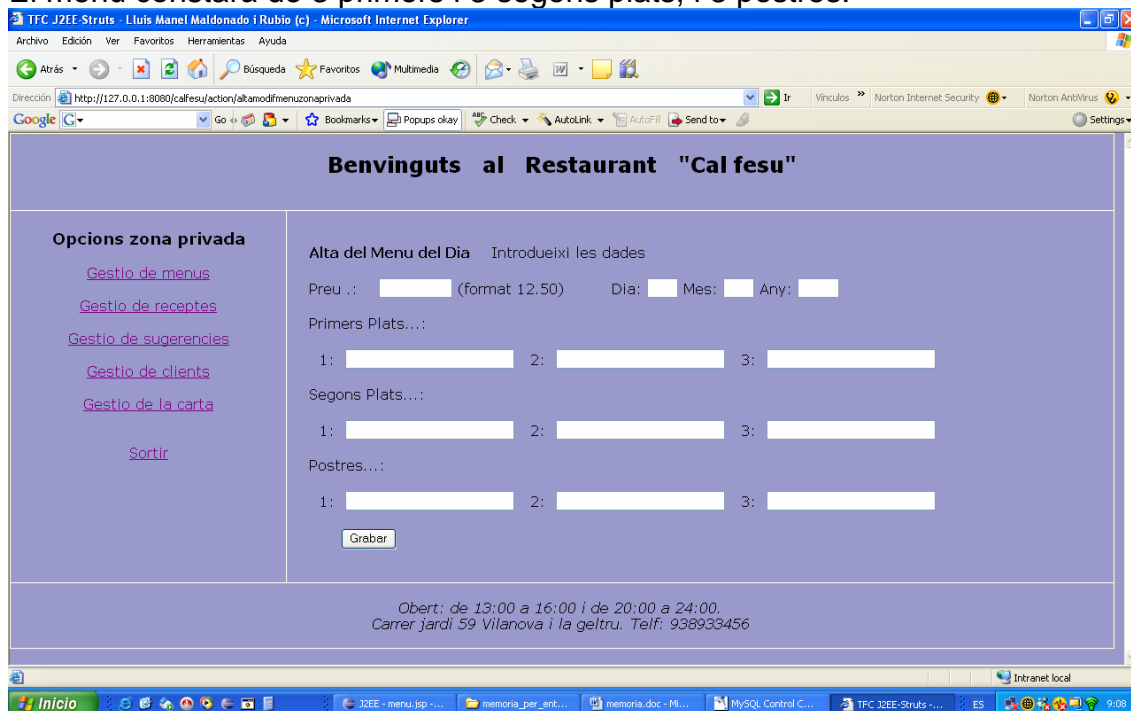


Figura 17: Pantalla alta menú diari.



Si el menú ja existeix, mostrarà les dades per poder modificar-les.

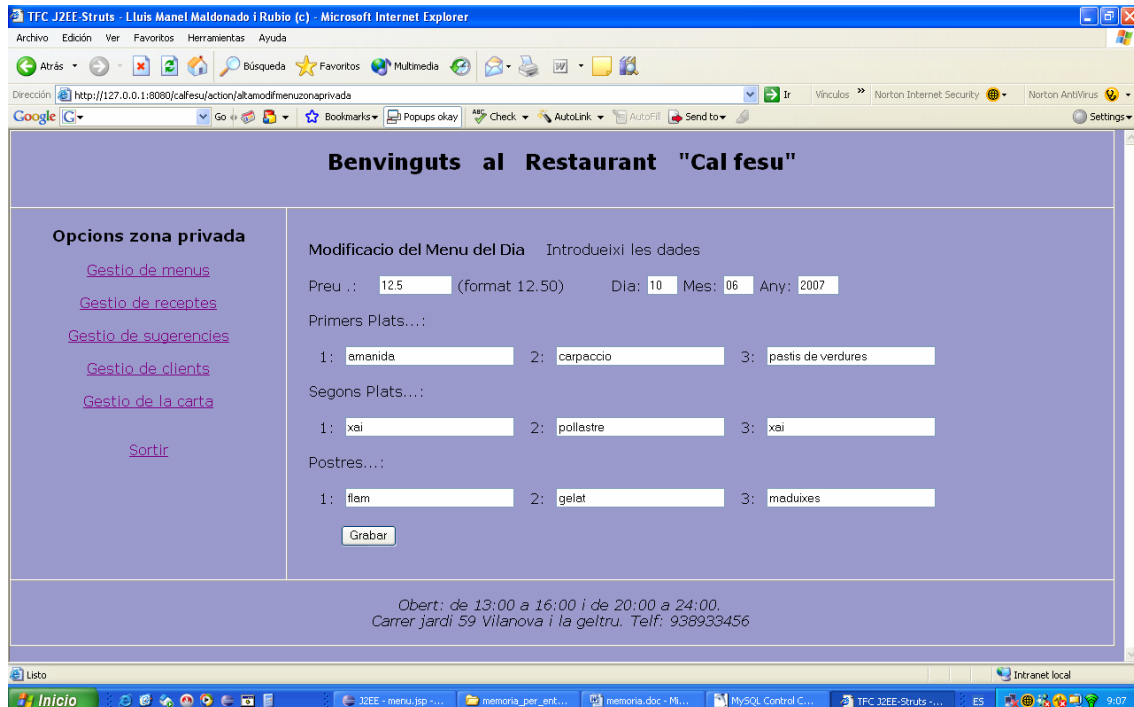


Figura 18: Pantalla modificació menú diari.

Consultar històric menús

Possibilitat de fer la búsqueda per dia.



Figura 19: Pantalla consulta històric menús.



Consultar un menú de l'històric



Figura 20: Pantalla consulta històric menús detall.

5.2.2 Pantalles Receptes

Gestió de receptes



Figura 21: Pantalla inici gestió receptes.



Alta de receptes

Quan es dona l'alta, s'enviarà un e-mail als subscriptors del receptari.



Figura 22: Pantalla alta receptes.

Consultar històric de receptes

Possibilitat de fer la búsqueda per nom.



Figura 23: Pantalla consulta històric receptes.



Consultar una recepta de l'històric



Figura 24: Pantalla consulta històric receptes detall.

5.2.3 Pantalles Sugerencies

Gestió de suggerències



Figura 25: Pantalla inici gestió suggerències.



Alta suggerències

Hi haurà 3 suggerències per dia.



Figura 26: Pantalla alta suggerències del dia.

En cas que les suggerències del dia ja existeixen, es poden modificar.



Figura 27: Pantalla modificació suggerències del dia.



Consultar històric de suggerències

Possibilitat de fer la búsqueda per dia.



Figura 28: Pantalla consulta històric suggerències.

Consultar una suggerència de l'històric



Figura 29: Pantalla consultar històric suggerències detall.



5.2.4 Pantalles Clients

Gestió de clients



Figura 30: Pantalla inici gestió clients.

Consultar històric de clients



Figura 31: Pantalla consultar històric clients.



Consultar un client de l'històric

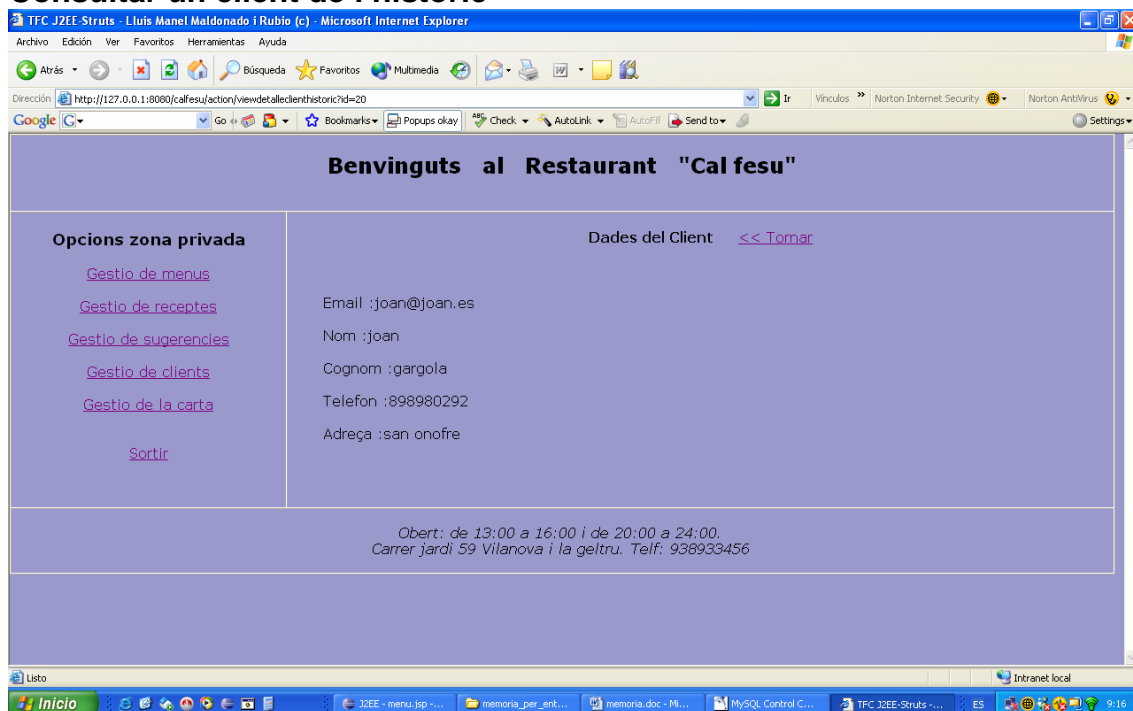


Figura 32:Pantalla consultar històric clients detall.

5.2.5 Pantalles Carta

Gestió de la carta

La carta constarà de 6 primers i 6 segons plats, i 6 postres.





Figura 33: Pantalla inici gestió carta.

Opció primers plats

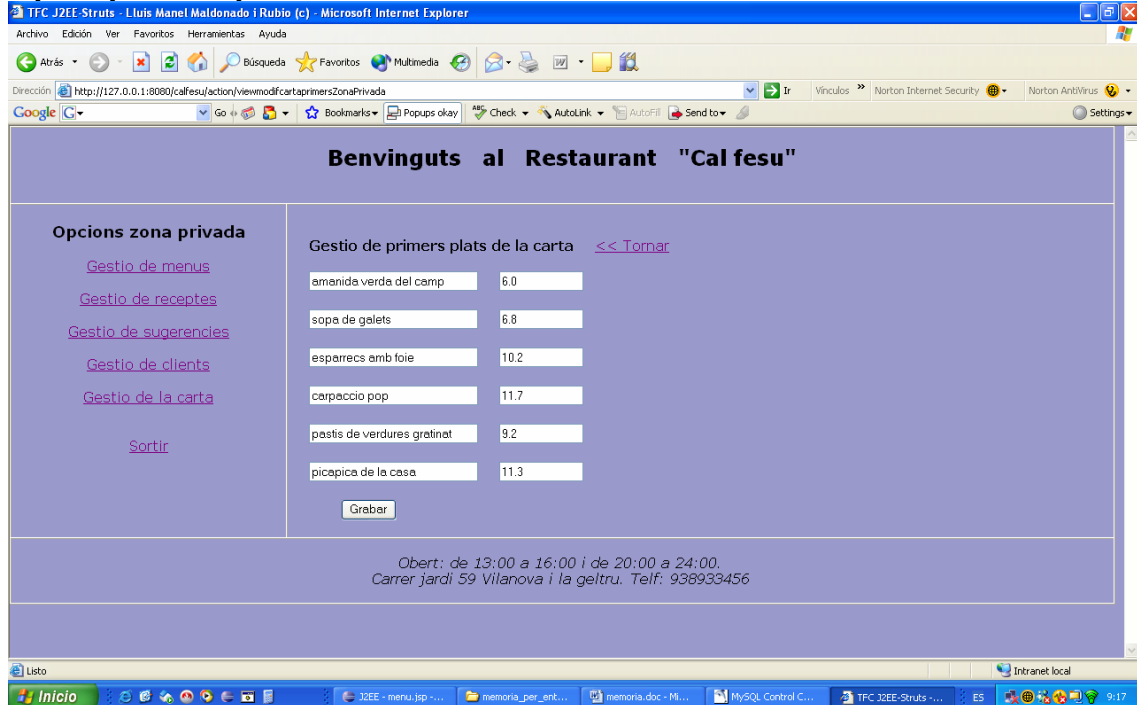


Figura 34: Pantalla modificació primers plats de la carta.

Opció segons plats de la carta



Figura 35 : Pantalla modificació segons plats de la carta.



Opció postres

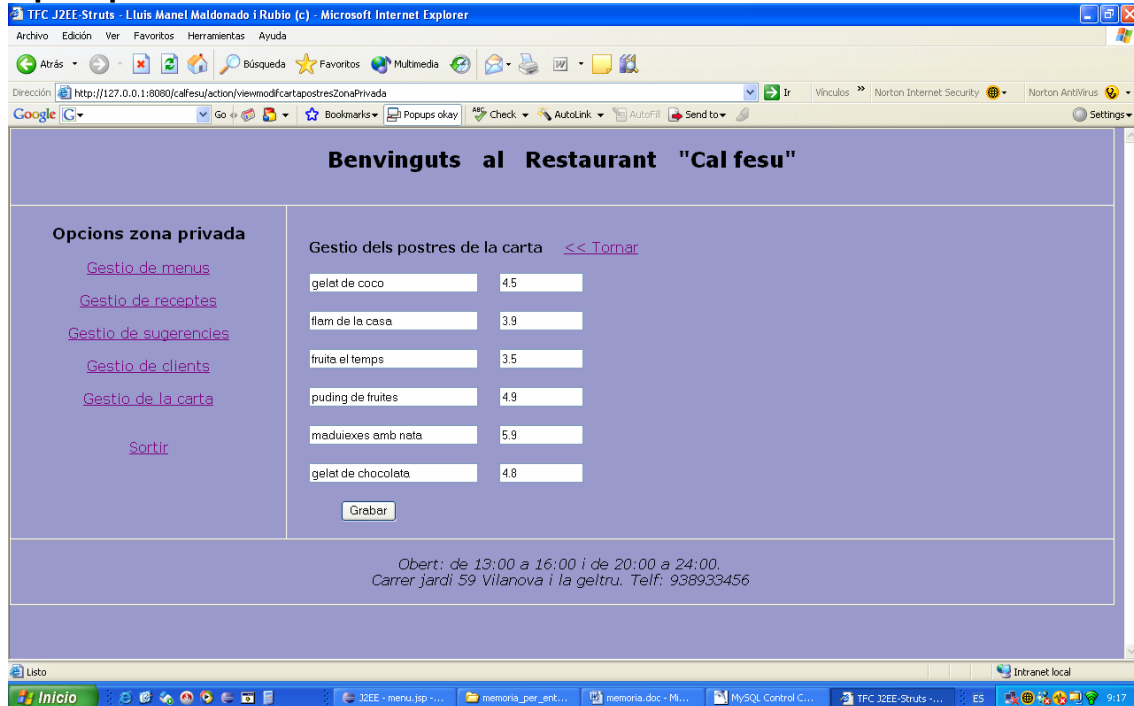


Figura 36: Pantalla modificació postres de la carta.

6 Disseny de la bbdd

Disseny del diagrama de bbdd realitzat amb l'eina: **Microsoft office visio 2003**.

En la nostra aplicació les taules de la bbdd són independents unes de les altres.

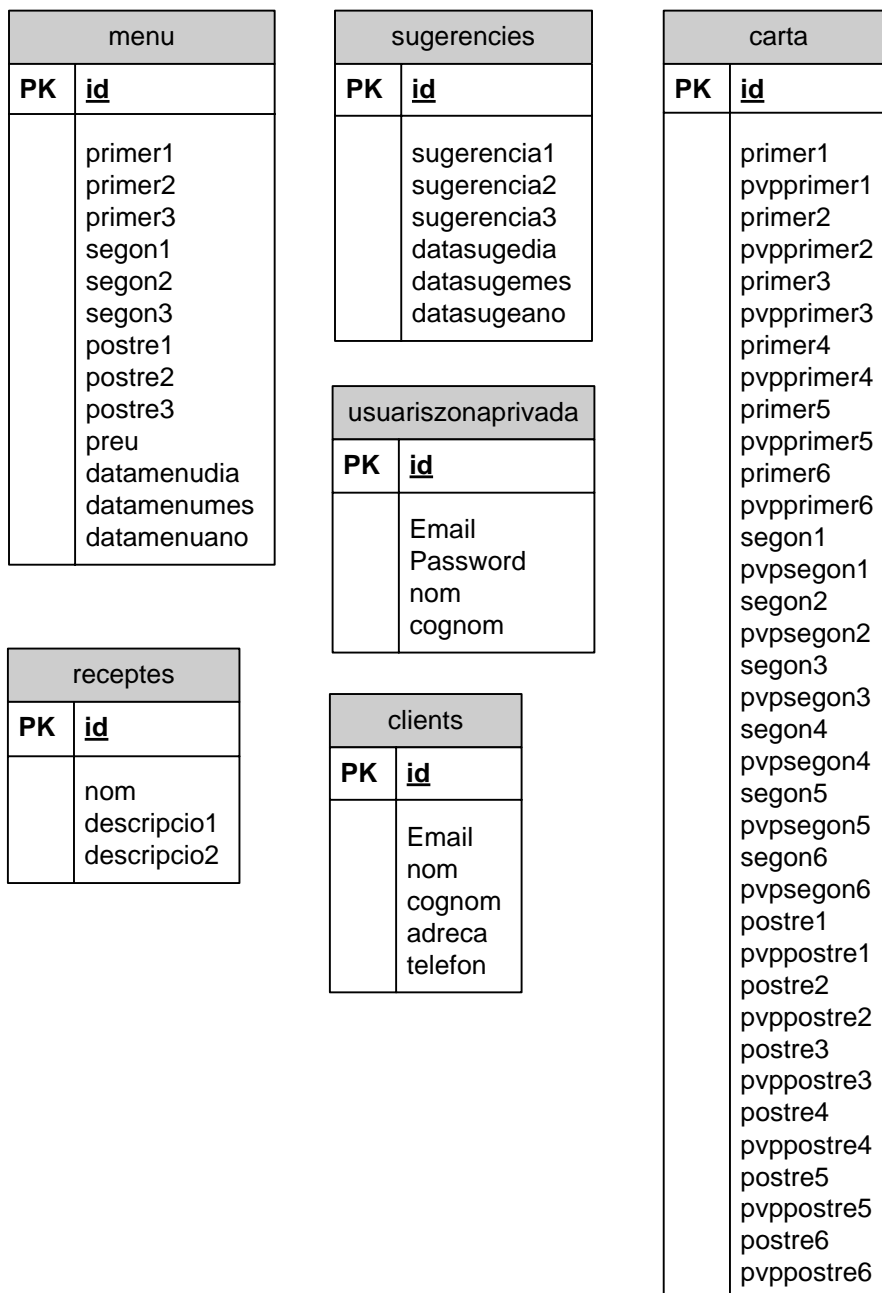


Figura 37: Disseny BBDD de l'aplicació.



7 Implementació

7.1 Model

La capa M(Model) conté la lògica del negoci i l'accés a les dades. Accedirem a la BBDD (mysql) amb el driver JDBC per mysql. Farem servir el patró de disseny "Value Objects" (VO) per retornar les dades. Utilitzarem un framework per a l'accés a bbdd i implementem un pool de Connexions amb la interface "DataSource".

Una particularitat del framework utilitzat (struts) és la classe que implementa el model (Action), que només ha de controlar el flux de l'aplicació deixant la lògica apart.

Un servlet gestiona la lògica del negoci.

En el nostre cas, totes les classes que realitzen les accions de Struts hereden de la classe CustomBaseAction. Aquesta classe hereta de Action i l'hem adaptat a les nostres necessitats per a que, cada cop que realitzem alguna acció Struts, tinguem disponible una instància del servei, tal i com ho hem implementat anteriorment. La idea és que sempre que executem una acció, puguem fer servir els mètodes del servei per a accedir a la bbdd.

Per altre banda, tenim dins de *com.restaurante.struts* una classe que NO hem fet servir, però que es força interessant: "CustomDispatchAction", que hereta "DispatchAction". Aquesta classe permet agrupar en una sola classe un conjunt de funcions relacionades, per exemple les accions d'una mateixa pàgina. Així, es poden agrupar conceptualment sense tenir que implementar una action independent per cadascuna.

En aquest cas, com es tracta de un grup d' accions (DispatchAction), necessitariem indicar `parameter="method"`. Així, poder cridar a la subacció dins del grup d'accions de la forma `/GetMenuAction?method=detalle`, per exemple.

Encara que totes les classes que realitzen les accions en la nostre aplicació hereden de la classe.

Per altra banda, la classe "CustomDispatchAction" no l'hem fet servir en la nostre aplicació, però és força interessant, com hem comentat abans.

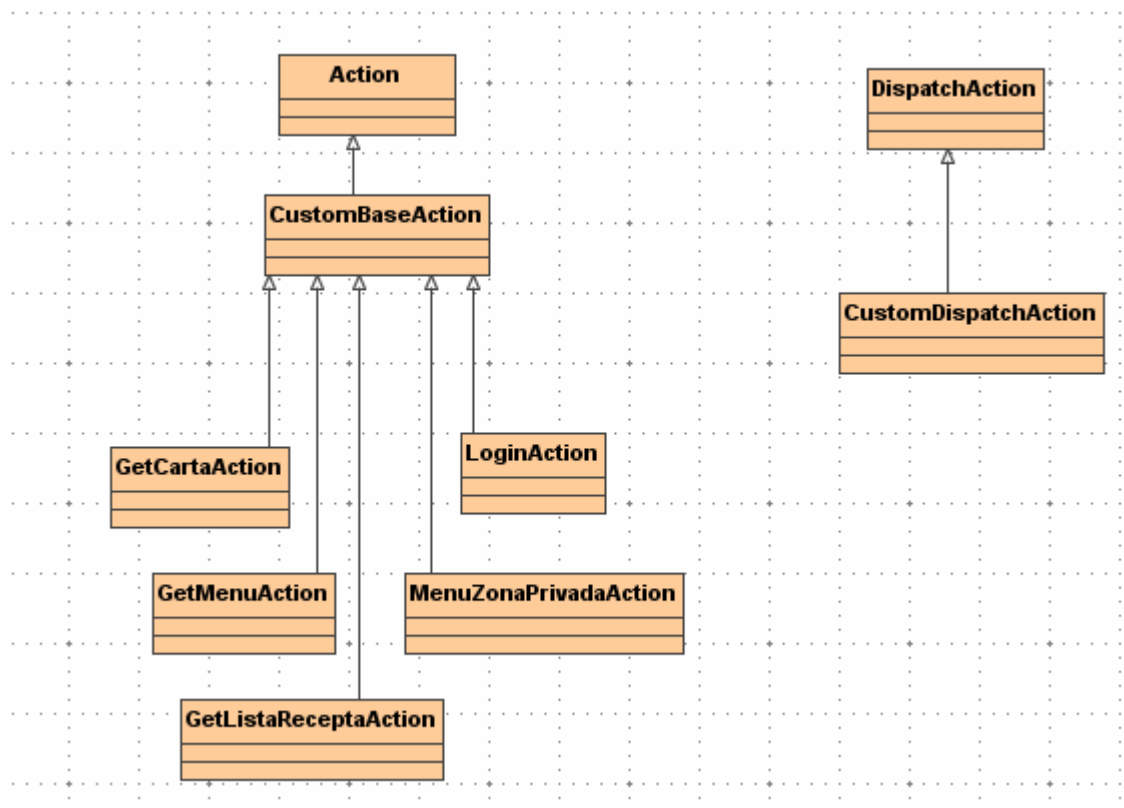


Figura 38: Algunes de les Actions utilitzades en el projecte.

7.1.1 Connexions a la BBDD – Pool de Connexions.

Un dels principals problemes de les aplicacions web, és la gestió de les connexions a base de dades. Una pràctica imprescindible és utilitzar un framework per a l'accés a bbdd, i així poder implementar un pool de connexions.

Struts permet definir un pool de connexions, i hem decidit utilitzar-lo en comptes del que ofereix el Tomcat, ja que tenir un pool de connexions al Tomcat requereix una parametrització al "web.xml" i una configuració addicional al fitxer "Server.xml". Cosa que pot ser un problema, si no tenim accés a aquests arxius. Això ho podem evitar amb struts, ja que a struts la configuració està dins de la pròpia aplicació al fitxer "struts-config.xml".

La interface "DataSource" és la millor manera de implementar un pool de connexions.

Un pool de connexions és una col·lecció de connexions gestionades i controlades. Amb l'ús d'un pool les connexions només les obre i les tanca el pool. Per al nostre codi, aquesta gestió és transparent. Així, podem controlar el



nombre màxim de connexions obertes i evitem que el sistema caigui, i ens assegurem que l'aplicació resulta escalable.

Struts ens proporciona el suport per poder gestionar un Pool de connexions a traves de l'arxiu struts-config.xml.

```
<struts-config>
<data-sources>
  <data-source type="org.apache.commons.dbcp.BasicDataSource">
    <set-property property="driverClassName" value="com.mysql.jdbc.Driver" />
    <set-property property="url" value="jdbc:mysql://localhost/calquesu" />
    <set-property property="username" value="root" />
    <set-property property="password" value="admin" />
    <set-property property="maxActive" value="10" />
    <set-property property="maxWait" value="5000" />
    <set-property property="defaultAutoCommit" value="false" />
    <set-property property="defaultReadOnly" value="false" />
    <set-property property="validationQuery" value="SELECT COUNT(*) FROM carta" />
  </data-source>
</data-sources>
<form-beans>
```

Figura 39: Datasource del arxiu struts-config.xml.

Per a poder utilitzar el pool hem afegit al nostre projecte els següents jar:

commons-dbcp-1.2.1.jar
commons-pool-1.3.jar
struts-legacy.jar

7.1.2 Java Mail

Cal dir que, per enviar els e-mails als subscriptors del receptor, hem fet servir L'API estàndard JavaMail (<http://java.sun.com/products/javamail>), és el que s'acostuma a usar per aquestes tasques.

Per a fer l'enviament, necessitem connectar amb un servidor de correu (servidor SMTP). En aquest punt tenim dues opcions:

- a) Instal·lar i configurar un servidor de correu al teu equip.
- b) Buscar servidor per internet que ho permeti (recomanable).

Hem fet servir la segona opció, que és la més recomanada, i a tal efecte hem creat una compta gratuïta a *terra.es* (javamailuoc@terra.es), que ens permetrà fer aquest enviament.



Extracte del codi del mètode que enviarà els mails:

```
/ Enviar Mail
public void EnviarMail(String mail, ReceptaVO receptaVO)
{
    final String user, password;
    user = "javamailuoc";
    password = "123456";

    class MiAutenticador extends Authenticator{
        public PasswordAuthentication getPasswordAuthentication(){
            return new PasswordAuthentication(user,password);
        }
    }

    // em creat un compte de correu gratuït a terra per aquest propòsit
    String from = "javamailuoc@terra.es";
    String to = mail;
    String recepta = receptaVO.getNom() +": "+receptaVO.getDescripcio2();
    String smtpHost = "smtp.terra.es" ;
    Properties props = System.getProperties();
    props.put("mail.smtp.host",smtpHost);
    props.put("mail.smtp.auth", "true");

    Authenticator auth = new MiAutenticador();
    Session sesion = Session.getDefaultInstance(props,auth);
    try{

        Message mensaje = new MimeMessage(sesion);
        mensaje.setSubject("restaurante Cal Fesu");
        mensaje.setFrom(new InternetAddress(from));
        mensaje.addRecipient(Message.RecipientType.TO, new InternetAddress(to));
```

Figura 40: Extracte mètode enviar mail.

7.2 Vista

La implementació la farem amb el framework Jakarta struts. Amb les JSP utilitzarem:

Struts Tags (struts-tiles, struts-html, struts-logic, struts-bean) etc.

Struts-tiles

Per a independitzar el contingut de la forma en base al patró composite view farem servir "struts-tiles": és un framework, que pertany a struts, permet treballar amb plantilles per a les vistes, perquè tota l'aplicació tingui un aspecte homogeni i permet que els components de les vistes siguin reutilitzables.

Struts-tiles permet definir una plantilla comuna per totes les jsp, i permet definir un àrea comuna de totes les pàgines, indicant en quin fitxer estan (capçalera, menú esquerra, contingut, peu, etc).

Per mostrar les dades, Struts proporciona dos tipus **ActionForms**, uns estàtics i els altres dinàmics (*DynaActionForm*).



Els estàtics es generen manualment i per el “*DynaActionForm*”, és un tipus de formulari que utilitza un Map per emmagatzemar valors internament, a partir d'una estructura definida als Vos, i que es genera automàticament.

Cal declarar-los al “Struts-config.xml” de la següent manera .

Els estàtics es declaren així:

```
<form-beans>

    <form-bean name="loginForm" type="com.restaurante.struts.forms.LoginForm" />

    <form-bean name="subscripcioForm" type="com.restaurante.struts.forms.SubscripcioForm" />

    <form-bean name="menuZonaPrivadaForm" type="com.restaurante.struts.forms.MenuZonaPrivadaForm" />
```

Figura 41: Declaració de Forms estàtics a l'arxiu struts-config.xml.

I els dinàmics així:

```
<form-bean name="detalleReceptaForm" dynamic="true" type="org.apache.struts.action.DynaActionForm">
    <form-property name="detalle" type="com.restaurante.struts.vo.ReceptaVO" />
</form-bean>

<form-bean name="detalleClientForm" dynamic="true" type="org.apache.struts.action.DynaActionForm">
    <form-property name="detalleclient" type="com.restaurante.struts.vo.UserClientVO" />
</form-bean>
```

Figura 42: Declaració de Forms dinàmics a l'arxiu struts-config.xml.

Fem servir un fitxer de recursos per a la nostra aplicació, amb extensió .properties, i que hem de declarar en el fitxer struts-config.xml així :

```
<message-resources parameter="resources.ApplicationResource" />
```

Figura 43: Declaració del fitxer de recursos a l'arxiu struts-config.xml.

Altres elements utilitzat per a la implementació de les vistes són las etiquetes lògiques com <logic:iterate>), etiquetes per a mostrar els elements dels formularis ActionForm i DynaActionForm definits (<bean:write>) i etiquetes d'HTML (<html:link>).



Veiem un exemple:

```

<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>

<br>
<center><p><b>Les ultimes receptes elaborades</b></p></center>
<p>
  <table border="0" cellpadding="5" cellspacing="5" width="100%">

  <logic:iterate id="recepta" scope="request" name="ListaReceptaKey">

    <tr>
      <td align="left">
        <html:link page="/action/viewdetallerecepta" paramId="id" paramName="recepta" paramPro:
          <bean:write name="recepta" property="nom"/>
        </html:link>
      </td>
      <td align="left">
        <bean:write name="recepta" property="descripcio1"/>
      </td>
    </tr>

  </logic:iterate>

</table>

```

Figura 44: Detall de etiquetes a la JSP.

Struts rep la informació en forma de VO's (value objects):

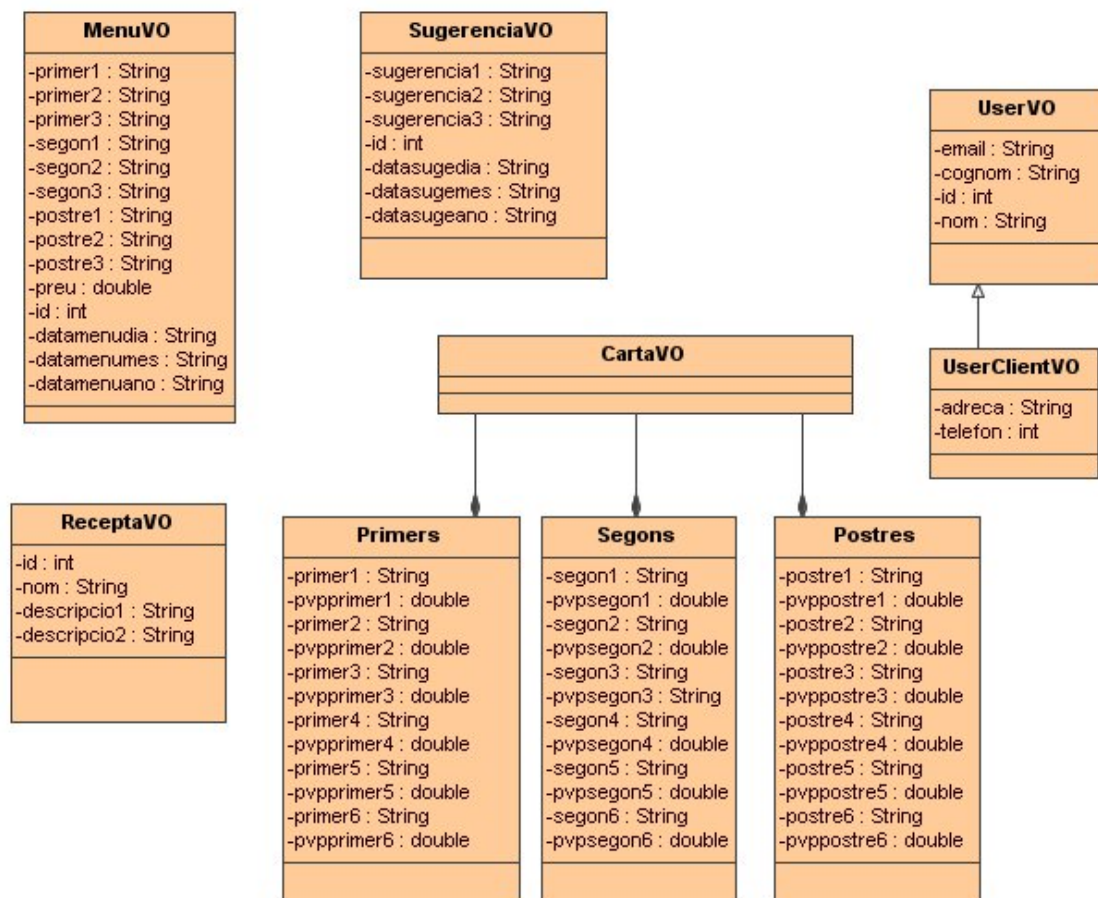


Figura 45: Value Objects utilitzats en el projecte.

7.3 Controlador

El controlador farà d'intermediari entre la capa Vista i la Model i gestiona les entrades o accions de l'usuari. Estarà implementat pel servlet propi del struts "ActionServlet", que fa servir el fitxer de configuració struts-config.xml.

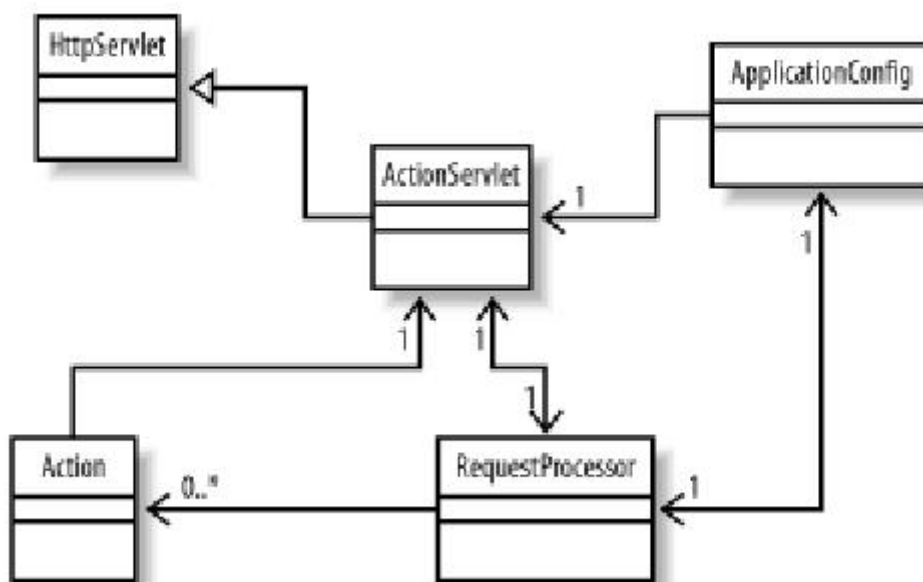


Figura 46: Classe ActionServlet.

ActionServlet, classe que hereda de `javax.servlet.http.HttpServlet`, i és la responsable del empaquetat i enrutament del tràfic HTTP cap al manejador apropiat dins del entorn (framework).

RequestProcessor, classe que permet desacoplar el procés de petició (request process) del ActionServlet, i així poder modificar com es processa la petició, fent una 'subclass' de RequestProcessor.

Action, classe que independitza la petició del client del model de negoci. És una extensió del component de control (capa C) i permet fer funcions como autorització, logging, o validació de sessions, abans d'invocar la lògica de negoci. El seu mètode més important és: `public ActionForward execute(ActionMapping mapping, HttpServletRequest request, HttpServletResponse response) throws Exception`.

ActionMapping, classe que representa una acció de mapeix que es defineix al fitxer de configuració de Struts(struts-config.xml). Indicar al controlador quina instància de Action cal invocar a cada petició.

ActionForward, classe que representa el destí, on el controlador ha d'enviar el control, una cop s'ha completat una Action.

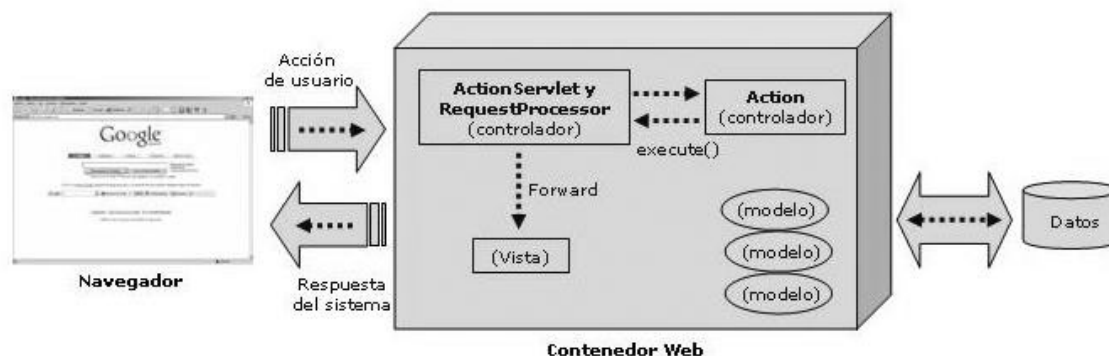


Figura 47: Funcionament de la classe ActionServlet.

El Controlador rep les peticions del navegador que fa servir l'usuari, llavors el controlador fa la petició que li cal al Model, que és qui implementa la lògica del negoci i l'accés a la bbdd. El controlador inicialitza la vista, i aquesta interactua amb el model, a fi de mostrar les dades adequades.

Struts rebrà informació de la capa M, encara que no sap com està implementada. Struts rep la informació en forma de vistes (VOs), però no sap com s'han creat. Per això, implementem un patró de disseny molt conegut, el Business Delegate. Amb aquest patró crearem un servei que pertany a la capa de negoci, i servirà de nexa d'unió amb la capa de control i serà a través del qual Struts demanarà i rebre els únics objectes que entén, els VOs.

Val a dir que, la implementació del servei connectarà amb JDBC amb la nostra bbdd i retornarà la vista adequada (patró Business Delegate). Per això, fem servir el patró de disseny "Business Interface", o sigui, primer crearem un interface com una caixa negra que cerra lo únic que conegui el client. Així doncs, podrem modificar el comportament del servei sense que ho noti Struts.

Un cop implementat el servei, necessitem crear una classe factoria de serveis perquè el client pugui crear el servei i fer-lo servir, això ho farà la classe "IRestauranteServiceFactory" i com abans tindrem la classe que implementa aquesta interface "RestauranteServiceFactory".

Per a implementar la factoria de serveis, primer s'hi implementa la classe PlugIn, que és exclusiva de Struts i permet que una classe externa a Struts es carregui com un mòdul addicional en l'entorn, o sigui, permet introduir en el context del controlador de Struts el servei per a que estigui disponible sempre public void init (ActionServlet servlet, ModuleConfig config) throws ServletException).

Així, també tenim el mètode amb el que el client podrà crear una instància del servei sempre que ho necessiti (public IRestauranteService createService throws ClassNotFoundException, IllegalAccessException, InstantiationException).

Per a que Struts sàpiga que es tracta d'un PlugIn i el carregui a l'inicialitzar l'aplicació, cal posar-ho al fitxer struts-config.xml:

```
</action-mappings>

<message-resources parameter="resources.ApplicationResource" />

<plug-in className="com.restaurante.struts.service.RestauranteServiceFactory" />
<plug-in className="org.apache.struts.validator.ValidatorPlugIn">
  <set-property property="pathnames" value="/WEB-INF/validator-rules.xml, /WEB-INF/va
</plug-in>

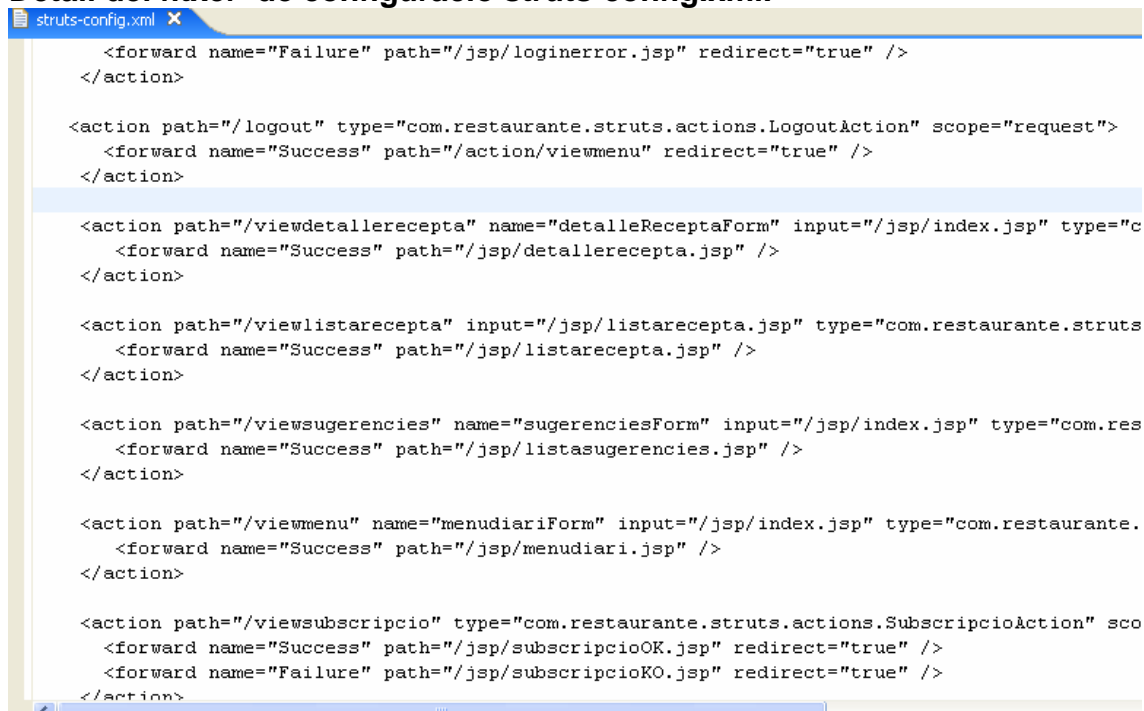
</struts-config>
```

Figura 48: Declaració d'un plugin al struts-config.xml.

Struts ens permet definir al "struts-config.xml" mapetjos d'accions genèriques. Como por exemple, un simple 'forward': indiquem que es tracta d'una acció predefinida posant : type="org.apache.struts.actions.ForwardAction".

Si ens fixem, quasi tots els mètodes retornen un objecte ActionForward. És un objecte exclusiu de Struts, i permet definir al fitxer de configuració (struts-config.xml), mitjançant mapeats, les característiques de les accions: nom, ruta, àmbit, validació, què fer en cas d'èxit i que fer en cas de fracàs (cap on redirigir la petició). Així, a codi posem "return mapping.findForward("Success")", perquè continuï el fluxe de l'aplicació cap a un altre acció o JSP. La lògica de les redireccions la controla Struts i ja no cal posar redireccions a les JSP.

Detall del fitxer de configuració struts-config.xml.



```
struts-config.xml x
<forward name="Failure" path="/jsp/loginerror.jsp" redirect="true" />
</action>

<action path="/logout" type="com.restaurante.struts.actions.LogoutAction" scope="request">
  <forward name="Success" path="/action/viewmenu" redirect="true" />
</action>

<action path="/viewdetallerecepta" name="detalleReceptaForm" input="/jsp/index.jsp" type="c
  <forward name="Success" path="/jsp/detallerecepta.jsp" />
</action>

<action path="/viewlistarecepta" input="/jsp/listarecepta.jsp" type="com.restaurante.struts
  <forward name="Success" path="/jsp/listarecepta.jsp" />
</action>

<action path="/viewsugerencies" name="sugerenciesForm" input="/jsp/index.jsp" type="com.res
  <forward name="Success" path="/jsp/listasugerencies.jsp" />
</action>

<action path="/viewmenu" name="menudiariForm" input="/jsp/index.jsp" type="com.restaurante.
  <forward name="Success" path="/jsp/menudiari.jsp" />
</action>

<action path="/viewsuscripcio" type="com.restaurante.struts.actions.SubscripcioAction" sco
  <forward name="Success" path="/jsp/subscripcioOK.jsp" redirect="true" />
  <forward name="Failure" path="/jsp/subscripcioKO.jsp" redirect="true" />
</action>
```

Figura 49: Detall actions de l'arxiu struts-config.xml.



8 Proves

Cal dir que hem fet un petit test de l'aplicació, on hem comprovat que les dades mostrades corresponguessin a les que hi ha a la bbdd i que els updates e inserts funcionessin correctament. També hem fet proves de navegabilitat, d'accés i de validació per entrar a la zona privada. En general, tot ha funcionat bé.

9 Estructura de l'aplicació

L'aplicació està estructurada en els següents directoris.

Dins de la carpeta "src" hi trobarem el codi java i els recursos de l'aplicació:

- ✓ "**com.restaurante.struts**": on es troben les classes genèriques.
- ✓ "**com.restaurante.struts.actions**": on es troben totes les classes "Action".
- ✓ "**com.restaurante.struts.forms**": on es troben tots els forms.
- ✓ "**com.restaurante.struts.service**": on es troben les classes que fan de factoria de serveis.
- ✓ "**com.restaurante.struts.vo**": on es troben les classes VO.

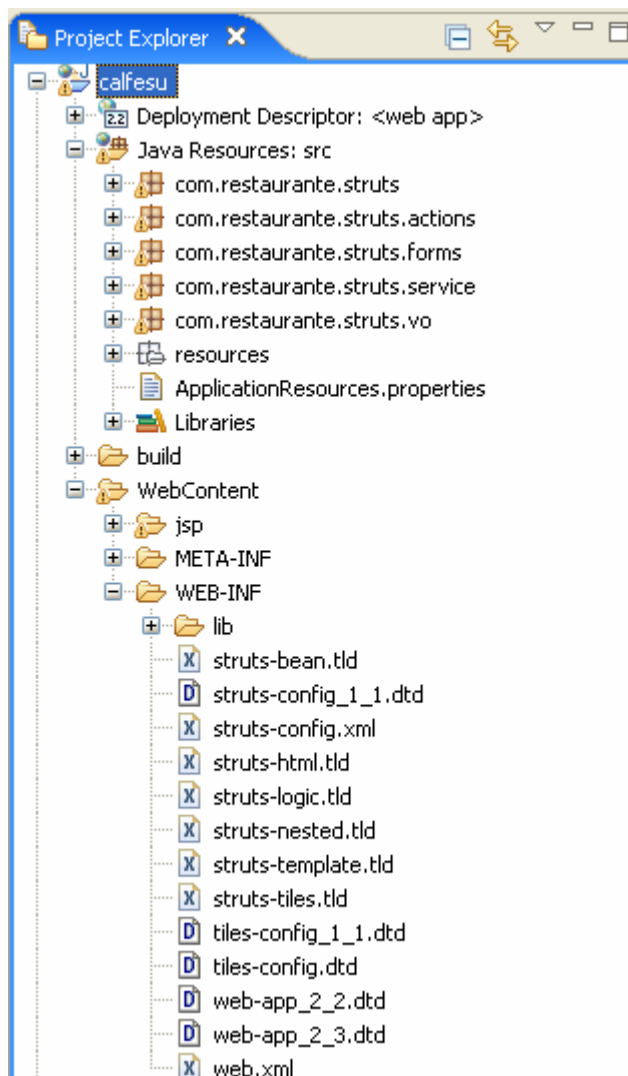


Figura 50: Detall estructura de directoris de l'aplicació.



Per altre banda, tenim la carpeta “WebContent”, que contindrà totes les JSP, i dins de la carpeta WEB-INF tindrem el fitxer “struts-config.xml”.

Amb Struts l'arxiu web.xml queda de la següent manera:

```
<web-app>

<servlet>
  <servlet-name>restaurant</servlet-name>
  <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
  <init-param>
    <param-name>config</param-name>
    <param-value>/WEB-INF/struts-config.xml</param-value>
  </init-param>
  <init-param>
    <param-name>debug</param-name>
    <param-value>3</param-value>
  </init-param>
  <init-param>
    <param-name>detail</param-name>
    <param-value>3</param-value>
  </init-param>
  <init-param>
    <param-name>application</param-name>
    <param-value>ApplicationResources</param-value>
  </init-param>
  <init-param>
    <param-name>validating</param-name>
    <param-value>true</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```



```
<servlet-mapping>
  <servlet-name>restaurante</servlet-name>
  <url-pattern>/action/*</url-pattern>
</servlet-mapping>

<!-- The Welcome File List -->
<welcome-file-list>
  <welcome-file>jsp/welcome.jsp</welcome-file>
</welcome-file-list>

<!-- Template Tag Library Descriptor -->
<taglib>
  <taglib-uri>/WEB-INF/struts-html.tld</taglib-uri>
  <taglib-location>/WEB-INF/struts-html.tld</taglib-location>
</taglib>

<taglib>
  <taglib-uri>/WEB-INF/struts-bean.tld</taglib-uri>
  <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>
</taglib>

<taglib>
  <taglib-uri>/WEB-INF/struts-logic.tld</taglib-uri>
  <taglib-location>/WEB-INF/struts-logic.tld</taglib-location>
</taglib>

<taglib>
  <taglib-uri>/WEB-INF/struts-template.tld</taglib-uri>
```

Figura 51: Detall arxiu web.xml.

Així queda el Java Build Path de l'aplicació:

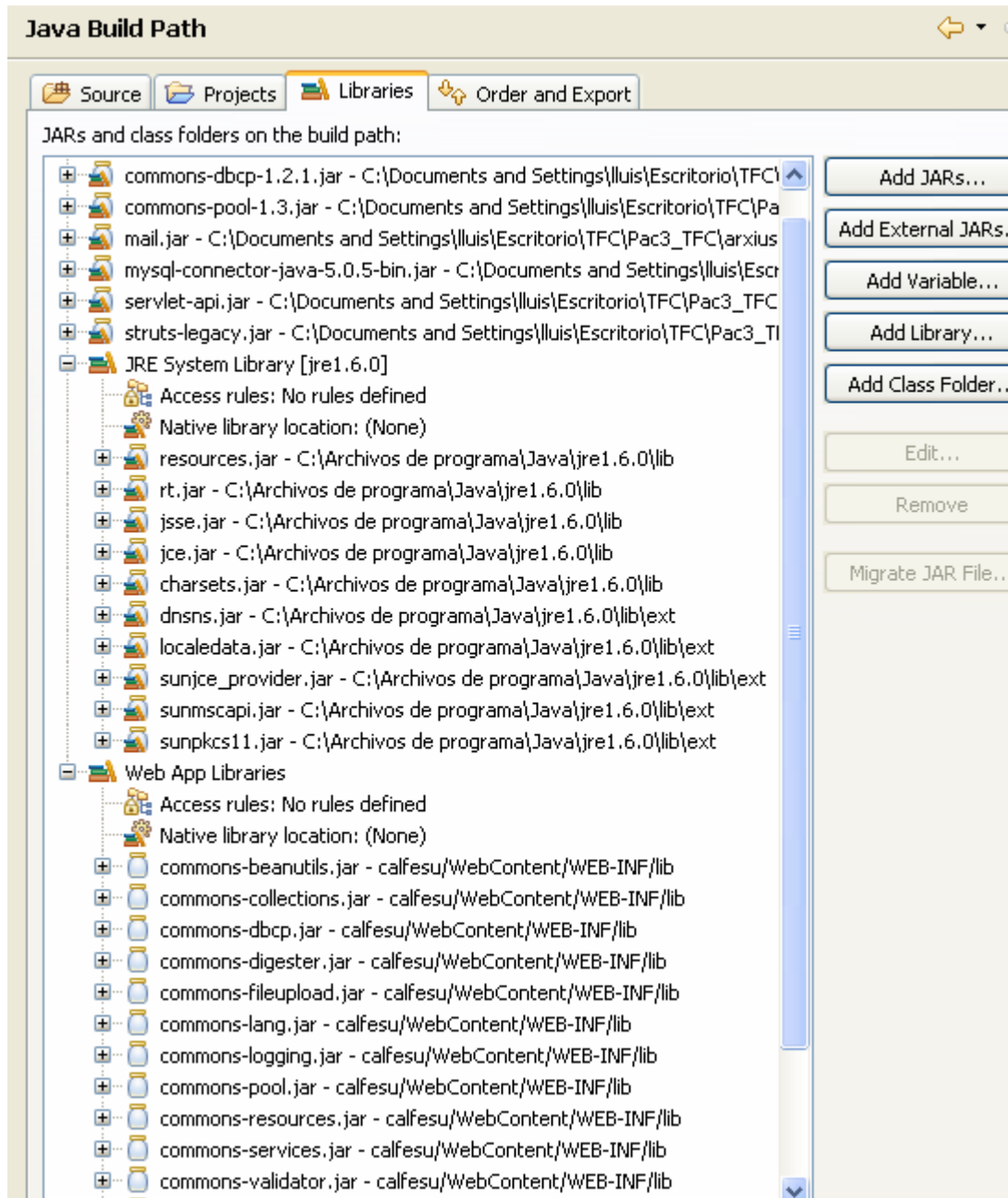


Figura 52: Detall build path.



10 Funcionalitats implementades de l'aplicació

10.1 Funcionalitats Implementades Zona Pública

- **Menú del dia:** carrega el menú del dia en curs de la bbdd. Si no hi és surt "no disponible".
- **Suggerències del dia:** mostra les tres suggerències del dia en curs. Si no estan donades d'alta a la bbdd, surt "no disponible".
- **Receptes:** mostra les cinc últimes receptes publicades. Fent clic podem veure la descripció llarga.
- **Subscripció al receptari:** mostra un formulari que els clients poden omplir per subscriure's al butlletí de receptes. Quan es prem el botó "Enviar", la subscripció es donarà d'alta automàticament en la bbdd del restaurant. Cada vegada que es creï una recepta nova, els clients subscriptes la rebran per e-mail.
- **Carta:** mostra els primers i els segons plats, i les postres de la carta, amb els preus corresponents.

10.2 Funcionalitats Implementades Zona Privada

Per accedir a la zona privada cal posar un usuari i password autoritzat, per exemple:

Usuari: lluis@lluis.com

Password: lluis

Un cop introduït l'usuari i el password, si existeix a la bbdd es pot accedir a la zona privada.

Gestió de menús: Té dues opcions :

- 1) **Alta/modificació menú d'avui:** Es pot donar d'alta el menú del dia en curs, però si aquest ja existeix llavors surten les dades i es poden modificar.
- 2) **Consultar menús antics:** Es veuen els menús dels últims cinc dies, però també es pot buscar el menú d'un dia en concret.

Gestió de receptes: Té dues opcions:

- 1) **Alta receptes:** Es poden donar d'alta receptes. Quan es dona d'alta la recepta s'envia un e-mail automàticament als subscriptors del receptari.
- 2) **Consultar menús antics:** Es veuen les cinc últimes receptes publicades, però també es pot buscar per nom la recepta que es vulgui.



Gestió de sugerències: Té dues opcions :

- 1) **Alta/modificació de les suggerències d'avui:** Es pot donar d'alta les tres suggerències del dia en curs, però si aquestes ja existeixen, llavors surten les dades i es poden modificar.
- 2) **Consultar les suggerències antigues:** Es veuen les suggerències dels últims cinc dies, però també es poden buscar les suggerències d'un dia en concret.

Gestió de clients: Mostra els cinc últims clients donats d'alta, i també es pot buscar en la bbdd de clients pel seu e-mail.

Gestió de la carta: Mostra els 6 primers i 6 segons plats, i 6 postres que componen la carta, i es poden modificar els plats i els preus.

11 Conclusions

Finalment, si revisem els objectius inicials del projecte, queda fer una valoració final.

Resta dir que els objectius inicials del projecte s'han satisfet plenament, tot i que queda molt per aprendre en aquest entorn, i petites coses que es poden millorar en aquest projecte.

Per una persona que ve del món del cobol i rpg, la corba d'aprenentatge ha estat molt gran, però ha valgut la pena. "He realitzat el meu primer projecte WEB".

Puc dir que, me posat al dia amb aquesta tecnologia(J2EE), fins ara totalment desconeguda i s' ha obtingut un producte final útil pel món real.

Es per això que, estic molt satisfet de la feina feta, especialment pel poc temps que he tingut, ja que aquest projecte ha coincidit amb un canvi de feina, un canvi de pis i una mudança.

Una eina com l'eclipse, ha estat fonamental per a poder implementar l'aplicació amb rapidesa. En algun cas la temporalització no s'ha seguit al peu de la lletra i les hores invertides, de bon troç, han estat superiors a les planificades inicialment, cosa que ja esperavem.



12 Manual d'instal·lació

12.1 Per a poder accedir a la bbdd cal instal·lar el mysql

Fem servir : **mysql-5.2.3-falcon-alpha-win32.zip**

Atenció:

al fer la instal·lació cal posar de password de root: "admin."

12.2 Instal·lació de mysqlcc

Per a poder manipular i accedir a la bbdd podem instal·lar el mysqlcc:

Fem servir: "mysqlcc-0.9.4-win32.zip".

12.3 Creació de la BBDD

Un cop tenim instal·lat i configurat el mysql, hem de copiar la nostra bbdd al directori on tinguem instal·lat el mysql. Copiem la nostra carpeta "calfesu" a dins del directori data. Així ja tenim la nostra bbdd amb les taules corresponents.

C:\Archivos de programa\MySQL\MySQL Server 5.2\data\calfesu

12.4 Instal·lació tomcat

Cal tenir instal·lat **apache-tomcat-5.5.20**, i afegir al directori d'instal·lació dins de common/lib els .jars següents: a "C:\Tomcat 5.5\common\lib".

- mail.jar (del javamail-1.4).
- activation.jar (del jaf-1.1).
- mysql-connector-java-5.0.5-bin.jar.

Ara, si anem a <http://127.0.0.1:8080> podem accedir al Tomcat i anar al Tomcat manager, on a "Archivo WAR a desplegar" escollirem el nostre WAR i desplegarem l'aplicació. Des de <http://127.0.0.1:8080/calfesu> ja podem accedir a la nostra aplicació J2EE.

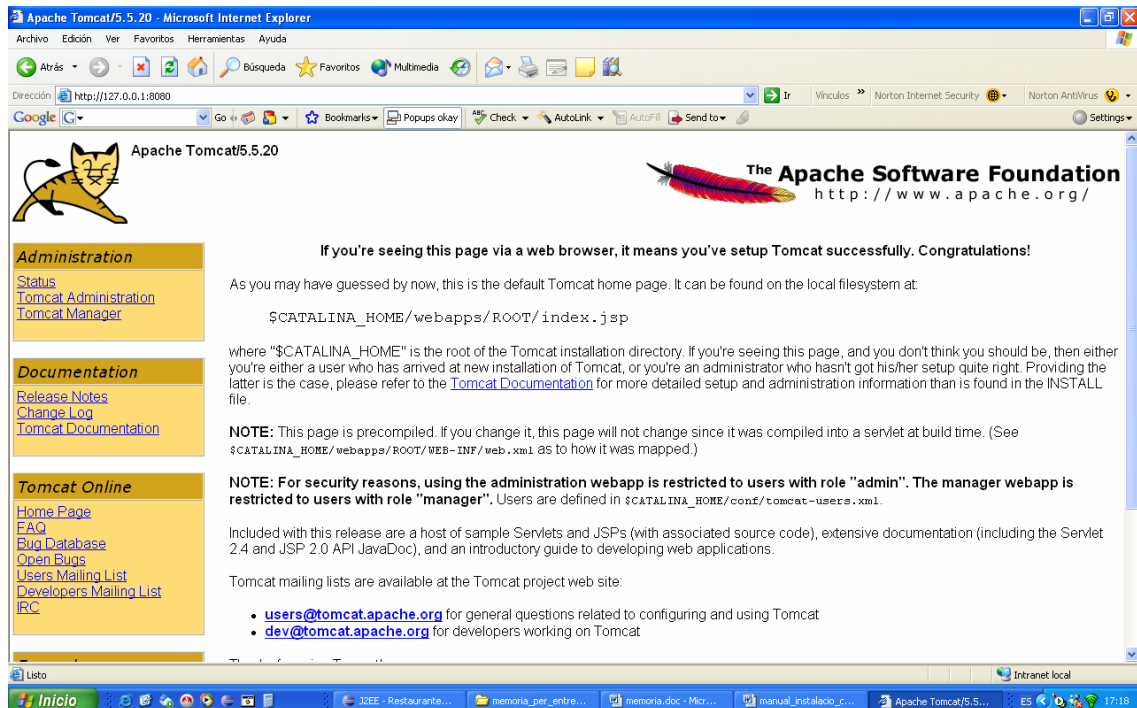


Figura 53: Pantalla tomcat.

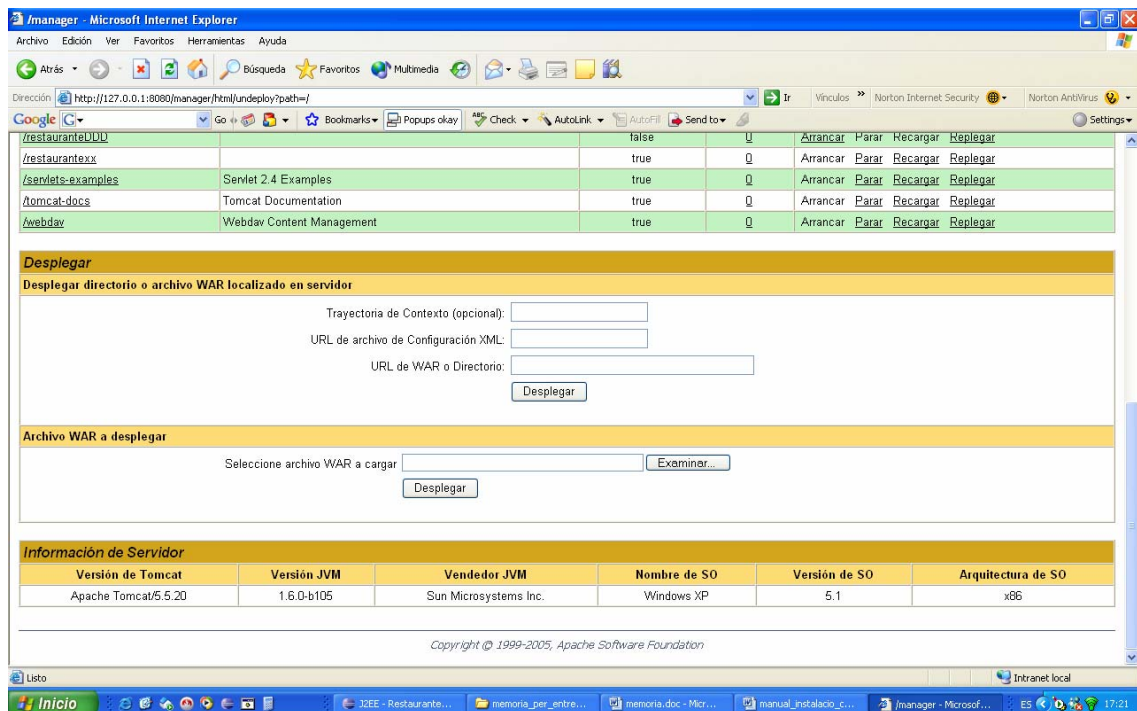


Figura 54: Pantalla tomcat manager.



13 Millors i manteniment futur

De cara al futur, alguna de les millors pendents que queden fora d'aquest projecte per falta de temps són:

- ✓ Millorar la gestió d'errors, ja que hi ha molts errors que no es monitoritzen.
- ✓ Afegir la opció de esborrar (clients, menús, suggerències etc..)de la bbdd, ara mateix les opcions d'esborrar no estan implementades, però poden ser necessàries en un futur.
- ✓ Millorar un xic la interfície gràfica.
- ✓ Potser la generació de llistats, o fins i tot la impressió del menú diari, les suggerències i la carta.

També queda fora de l'abast d'aquest projecte, l'ús d'algun framework de persistència com

- [Hibernate](#)
- [iBATIS](#)
- [Object Relational Bridge](#)
- [Torque / Peers](#)

Que farien la nostra aplicació persistent.



14 Bibliografia

- <http://struts.apache.org/>
- <http://www.theserverside.com/tt/books/sourcebeat/JakartaStrutsLive/index.tss>
- <http://struts.sourceforge.net/>
- http://www.programacion.com/java/tutorial/joa_struts/
- <http://www.programacion.net/java/tutorial/javamail/>
- <http://www.desarrolloweb.com/articulos/2243.php>
- <http://www.mysql.com>
- <http://dev.mysql.com/doc/refman/5.0/es/index.html>
- <http://www.laliluna.de/tutorial-struts-eclipse-espanol.html>
- <http://www.ibm.com/developerworks/library/j-struts/>
- <http://www.roseindia.net/struts/strutsguide.shtml>
- <http://jakarta.apache.org/>
- <http://courses.coreservlets.com/Course-Materials/struts.html>
- <http://struts.application-servers.com/>
- <http://java.sun.com/blueprints/corej2eepatterns/Patterns/index.html>
- http://www.programacion.net/bbdd/articulo/ale_poolstruts/