Aplicació web pel restaurant Gat Blau

Laia Llunas Mestres

Enginyeria Tècnica d'informàtica de Sistemes

Antoni Oller Arcas

16 de Juny 2006

Quan vaig fer el meu primer TFC ("Underwater communication systems for divers"), a Gales i fa ja cinc anys, no vaig posar cap apartat de dedicatòries. Ara, és gairebé el primer que escric d'aquest document. Com canvien les coses...

Dedico aquest treball al meu company, el Jose, sense la paciència del qual hauria estat impossible tirar endavant. També vull agrair la col·laboració de companys de feina com el Toni, i el saber escoltar de la Maye, el David, el Fèlix i el Jesús.

A la meva germana pel seu sentit de l'estètica i a la Jo, la mestressa del Gat, per la seva col·laboració interessada. I als que em deixo i que han estat amb mi. Gràcies a tots.

Resum treball

El passat mes de març de 2005 el restaurant Gat Blau va obrir les seves portes al públic. Des de llavors l'evolució del negoci ha estat molt positiva, el nombre de clients no para d'augmentar i l'acceptació del restaurant entre els seus comensals és molt bona.

Aquest camí que està seguint el Gat Blau va fer pensar als seus gestors en el fet de posar en marxa una pàgina web que oferís informació i serveis als seus clients.

Arrel d'aquesta necessitat va néixer aquest projecte final de carrera, i és que es va aprofitat la realització d'aquest treball per implementar l'aplicació web que ha de satisfer la necessitat de publicar una pàgina d'informació i serveis pels clients del Gat Blau.

L'aplicació web s'ha pensat i dissenyat en base a tots els requisits inicials, i és per això que ja des dels moments inicials va quedar clar que no es podria implementar tota la funcionalitat en el temps previst per l'execució del TFC. Es va pensar en una aplicació web basada en J2EE, que utilitzés diferents eines com strus i hibernate per a la seva implementació.

Cal notar que els gestors del restaurant no tenen coneixements per mantenir l'aplicació des de les interfícies de la base de dades o d'implementació. És per això que l'aplicació té dos punts entrada, l'accés general dels clients del restaurant a les diferents informacions i als serveis, però també l'accés pels gestors des d'on se'ls permet mantenir els continguts i informacions de la web gràcies a una interfície amigable.

Així doncs, en aquest document, es presenta un treball final de carrera que recull tot aquest desenvolupament. Es detallen les diferents etapes del projecte, pla de treball, anàlisi, disseny i implementació, així com les conclusions del treball realitzat. També es proporcionen els manuals que han d'ajudar a instal·lar l'aplicació en un servidor Tomcat i han d'orientar els usuaris en la seva utilització

1. INTRODUCCIÓ.....1

2. PLA DE TREBALL.....3

2.1. Continguts i serveis pàgina web:.....3
2.2. Recursos.....5
2.3. Riscos.....5
2.4. Descomposició en tasques.....6
2.5. Temportizació i documentació.....8
2.6. Calendari Projecte.....9
2.6.1. Anàlisi d'activitats i agenda.....9
2.6.2. Temporització – Diagrames Gantt.....11

3. ESPECIFICACIÓ I ANÀLISI.....14

3.1. Documentació dels requisits.....14
3.1.1. Model de domini.....14
3.1.2. Model de negoci.....14
3.1.2.1. Glossari del model de negoci.....15
3.2. Guions.....16
3.3. Casos d'ús, plantejament i anàlisi.....16
3.3.1. Actors.....16
3.3.2. Descripció dels casos d'ús.....17
3.3.2.1. Consultar Gastronomia.....17
3.3.2.2. Fer reserves.....17
3.3.2.3. Registre usuari.....17
3.3.2.4. Modificació Subscripció.....18
3.3.2.5. Gestió de Reserves.....18
3.3.2.6. Configuració del restaurant.....19
3.3.2.7. Gestió de la oferta gastronòmica.....19

4. DISSENY.....22

4.1. Disseny arquitectònic.....22
4.1.1. Patró MVC.....22
4.1.2. Paquets.....23
4.1.3. Tecnologies utilitzades.....23
4.1.4. Diagrama de l'Arquitectura.....24
4.1.5. Revisió dels casos d'ús.....25
4.1.6. Validació i gestió usuaris.....25
4.1.7. Revisió Casos d'ús.....26
4.1.7.1. Consultar Gastronomia.....26
4.1.7.2. Fer reserves.....26
4.1.7.3. Registre usuari.....27
4.1.7.4. Gestió de Reserves.....27
4.1.7.5. Configuració del restaurant.....28
4.1.7.6. Gestió de la oferta gastronòmica.....28
4.2. Diagrama estàtic.....29
4.3. Disseny de la persistència.....30
4.3.1. Consideracions.....33
4.4. Internacionalització: I18n.....33

4.5. Disseny interfície **34**
4.5.1. Interfície del gestor 34
4.5.2. Interfície pública 35

5. IMPLEMENTACIÓ **36**

5.1. Eines i tècniques utilitzades **36**
5.1.1. Eines de recolzament 36
5.1.2. Eines investigades per a la implementació de funcionalitats futures 37
5.2. Estructura de l'aplicació **38**
5.2.1. Estructura paquet cat.lllunas 38
5.2.1.1. Interfície usuari: Utilització Struts per MVC 38
5.2.1.2. Internacionalització 41
5.2.1.3. Les vistes i l'aplicació web 42
5.2.1.4. Lògica Negoci: Gestors i interfícies de dades 43
5.2.1.5. Persistència: Model Simulat i hibernate 45
5.2.1.6. Control d'errors i excepcions 46
5.3. Proves **47**

6. CONCLUSIONS **48**

7. MANUALS **51**

7.1. Guia Instal·lació **51**
7.1.1. Requisits 51
7.1.2. Preparar la base de dades 51
7.1.3. Carregar l'aplicació en el servidor d'aplicacions 51
7.1.4. Instal·lació de les llibreries 52
7.2. Manual usuari **53**
7.2.1. Entrar a l'aplicació 53
7.2.2. Gestor del Restaurant 54
7.2.3. Gestor Gastronomia 55
7.2.3.1. Revisió de llistes i plats 55
7.2.3.2. L'edició de llistes (cartes i menús) 57
7.2.3.3. Edició d'elements (afegir, editar plats i begudes) 58
7.2.4. Gestor Reserves 59
7.2.5. Desconnexió 60
7.2.6. Gestor de clients 60

8. BIBLIOGRAFIA **61**

1. Introducció

Dins el pla d'estudis de l'enginyeria Tècnica d'informàtica de Sistemes es troba, ja en el tram final dels estudis, el Treball Final de Carrera. Aquest treball es pot emmarcar dins de diferents àrees, però el que es descriu en aquesta memòria és un treball associat a dins de l'àrea J2EE. La tecnologia J2EE inclou un ampli ventall de possibilitats, sent les aplicacions web un dels màxims exponents d'aquesta àrea.

En la meva vida professional puc diferenciar dues grans línies, una d'elles es caracteritza pel desenvolupament d'aplicacions de baix nivell (controladors d'impressores, escànners, ports, etc.), programari per equips molt concretes i de maquinari dedicat, treballant a vegades sense sistema operatiu i amb requeriments molt estrictes de memòria, tot implementant en C++. L'altre línia a la qual m'he dedicat professionalment engloba el manteniment de sistemes i equips. És per això que a l'hora d'escollir l'àrea del treball final de carrera vaig decidir obrir una nova línia de coneixement.

D'altra banda, aproximadament a principis del 2006, els meus tiets, propietaris del Restaurant Gat Blau, em van comentar la possibilitat de fer una petita pàgina web que mostrés els menús de cada dia, que permetés als clients fer reserves, etc. Sense saber encara, quina seria l'àrea del TFC vaig començar a investigar com es podria fer això que em demanaven, doncs el meu coneixement del tema era força escàs. Aquesta investigació em va portar al món del .Net i del J2EE.

Aquestes són les dues raons de pes que em van fer decantar per escollir aquesta àrea i aquest projecte: conèixer noves tecnologies i poder satisfer una necessitat de persones molt pròximes.

És per tot això que l'objectiu principal d'aquest TFC és adquirir un mínim coneixement del que es cou en el món d'Internet, de les aplicacions web i del J2EE, això si, tenint clar que el punt de partida era el desconeixement total del que s'havia de fer.

El segon objectiu era obtenir una aplicació web útil per al restaurant, que permetés als seus gestors mantenir el contingut des d'una interfície senzilla. L'aplicació havia de ser flexible i havia de seguir criteris de disseny que la fessin escalable i ens permetessin ampliar-la posteriorment

L'enfocament i els mètodes he volgut seguir han estat força acadèmics, doncs la meva experiència professional m'ha mostrat que no sempre se segueix un mètode tan clar.

La planificació i el procés de construcció han seguit força el patró marcat pel cicle de vida clàssic, i és que que per limitacions de temps hem seguit força les etapes de recollida de requisits, anàlisi, disseny i implementació. La posada en producció i el manteniment queden fora de l'abast d'aquest treball. Tot i així durant les fases d'anàlisi es van retocar els requeriments i esperem poder ajustar aquesta primera implementació amb l'experiència dels usuaris per poder arribar a una implementació final

Pel que fa a l'enfocament del disseny he volgut que aquest estigués al màxim centrat a l'usuari, oferint els serveis necessaris i pensant en un producte final amb un alt nivell d'usabilitat. Malgrat això, aquests criteris sovint s'apliquen més en el disseny i implementació de la interfície que, degut a les restriccions temporals, ha quedat molt reduït.

D'altres criteris seguits han estat la utilització de patrons de disseny, la utilització d'eines i solucions adients a l'aplicació a desenvolupar, el disseny orientat a objectes

Les diferents fites marcades en el pla de treball han generat diferents resultats, el pla de treball l'anàlisi i el disseny. Aquest resultats s'exposen com a capítols d'aquesta memòria. També s'ha implementat la major part de la base de dades (que es pot crear i carregar amb dades de prova gràcies als scripts que també s'adjunten), i s'ha obtingut una aplicació web que implementa part de la funcionalitat descrita inicialment, aquesta aplicació s'adjunta en un fitxer de desplegament tipus war.

De forma esquemàtica la documentació entregada inclou:

- ✔ La memòria, aquest mateix document on s'exposa el projecte i es desenvolupen les diferents etapes del mateix.
- ✔ La presentació, a mode de resum de la feina realitzada i que recull els punts més rellevants d'aquesta memòria.
- ✔ L'aplicació, el producte desenvolupat, que juntament amb els manuals inclosos en la memòria i els scripts per la base de dades, pot ser fàcilment posada en marxa.
- ✔ El prototipus de la interfície per al públic en general.

En els següents capítols es mostra el procés de producció d'aquest projecte. En primer lloc presentem els requisits que d'acord amb els gestors del restaurant es van definir, seguidament el treball d'anàlisi i el disseny de l'aplicació. Un cop revisats aquests punts es comenta la implementació de l'aplicació i la valoració del treball realitzat. Finalment es donaran les eines per a la instal·lació de l'aplicació.

2. Pla de Treball

2.1.Continguts i serveis pàgina web:

Tenint en compte les necessitats dels clients, així com les característiques d'altes pàgines de restaurants, s'han acordat els continguts següents:

a) Divulgació: la presentació restaurant, situació, com arribar-hi (consulta a tmb.net per mostrar el recorregut), fotografies local, etc. Horari, pàrkings propers.

    a. Les fotografies del local han de ser facilitades pels clients. Serà una funcionalitat a implementar en una segona fase, ja que depèn de la disponibilitat dels clients

b) Gestor de reserves (recollida reserves, resposta automàtica en funció de la disponibilitat, confirmació per part del gestor, estat de la reserva, enviament correu electrònic confirmació)

    a. Dades reserva: dia, hora, no persones, telf contacte, adreça correu

    b. Pre-confirmació: El sistema podrà tenir informat el conjunt de taules disponibles i així assignar les reserves de forma semi automàtica, la configuració de taules ha de poder ser modificada pels gestors.

c) Menús i cartes: El sistema també hauria de permetre mantenir una base de dades que permetés als gestors del restaurant mantenir de forma fàcil actualitzada la informació del menú del dia i la informació de la carta i menús per grup, així com una publicació periòdica de ressenyes de plats. Igualment hauria d'haver la informació de la carta de vins.

    a. Cadascun dels plats ha de tenir la següent informació associada (nom, descripció, foto, preu, classificació – primers, segons postres -, menú o carta).

    b. Tenint en compte que els gestors hauran d'introduir la informació dels menús i cartes en aquesta aplicació serà molt útil que es pugui generar una sortida en format document word, pdf, etc. D'aquesta manera els gestors introduiràn aquesta informació un sol cop i obtindràn diferents sortides, una pel contingut de la pàgina, l'altre per poder fer la impressió de les cartes..... evitant així haver d'introduir la informació dos cops.

d) Els actuals gestors del restaurant volen poder organitzar events de forma regular, lectura de contes o poesies, etc, de manera que el sistema hauria de permetre gestionar un tauler d'anuncis. (Funcionalitat futura, no cal que s'implementi ara).

e) Existeix la possibilitat que qui ho desitgi rebi butlletí notícies per correu, cal registre usuaris. També poden rebre menú diari si ho desitgen

a. Subscripcions de clients al gestor d'anuncis de manera que rebin al seu correu les noticies del restaurant

b. Subscripcions a la informació diària (composició del menú del dia)

f) Sol·licitud d'informació: A través de formulari s'enviarà petició de consulta del client a restaurant

g) Hauria de poder ampliar-se a una funcionalitat que gestiones les ressenyes en premsa, etc. (Futura)

h) Seria convenient un aeina que gestionés els serveis oferts pel restaurant(nom servei i descripció). Exemples de la informació serien sobre serveis de càtering o servei de ticket-regal.

i) Multiidioma: Tenint en compte els clients que fins ara han visitat les instal·lacions és necessària una aplicació multiidioma, que inicialment serà en català però que hauria de poder ser traduïda d'una forma àgil. La traducció no pot ser estàtica doncs la llista de plats pot variar en el temps. Ha d'estar implementada tot i que no ha de ser funcional en la primera etapa del projecte

Alguns d'aquests continguts han d'estar previstos, tot i que no és necessari que estiguin implementats en una primera fase. D'altres poden estar implementats però no ser funcionals. La decisió final dependrà de la discussió que es realitzi durant l'etapa de planificació i anàlisi.

L'empresa no preveu contractar serveis per mantenir el contingut de la pàgina, així que caldrà definir i permetre un interfície pels gestors que de forma fàcil i senzilla permeti gestionar les reserves, el contingut de la base de dades de menús i l'apartat multiidioma.

També cal tenir en compte que encara que no és crític per al públic general, si que ho és per als gestors del restaurant, doncs la plataforma en que treballen és MAC i per tant caldrà dissenyar la pàgina web en base un HTML compatible amb navegadors com SAFARI

Hem de tenir en compte dos perfils d'usuaris clars, el client del restaurant i el gestor del restaurant que haurà de mantenir els continguts de la web, la traducció de la mateixa així com les reserves i la informació dels plats.

2.2.Recursos

Dins del marc del projecte final de carrera es disposarà dels següents recursos.

Programari:

a. Desenvolupament: Basat en programari lliure, eines de desenvolupament com Eclipse i altres definides dins l'entorn de J2EE Us de programari com DreamWeaver per al disseny d'interfícies HTML

b. Disseny: Eines per al disseny POO com les utilitzades per la realització de diagrames UML.

c. Gestió: Eines de gestió de projectes i documentació tal com el paquet MS Office i Ms Project

d. Comunicació: Campus UOC per la comunicació estudiant, consultor i personal suport. Correu electrònic per la comunicació amb client.

b) Maquinari: ordinador personal de l'estudiant encarregat de desenvolupar el projecte

c) Personal: En aquest projecte hi haurà involucrades 2 persones de forma directa i una donant suport a la tecnologia.

a. Director del projecte: Supervisió i assessorament

b. Estudiant TFC: execució de les tasques i activitats del projecte

c. Consultor laboratori: Persona encarregada del suport per les tecnologies i eines utilitzades.

2.3.Riscos

Tenint en compte les característiques del projecte i de l'equip que l'ha de desenvolupar s'han identificat els següents riscos.

a) Llistat de requeriments ampli: Caldrà que tant en la descripció general com en l'anàlisi funcional es limitin quines seran les funcionalitats obligatòries, opcionals i opcions futures.

b) Temps: Tenint en compte que el projecte s'ha de realitzar en 17 setmanes, caldrà que es faci un seguiment estricte de la temporalització establerta, d'aquesta manera es podran reajustar tasques definides inicialment quan es detecti que existeix una desviació respecte a la planificació inicial

c) Desconeixement plataforma: Tenint en compte que l'estudiant és qui ha de desenvolupar totes les tasques del projecte i la seva formació prèvia, es detecta un desconeixement de la plataforma J2EE. Malgrat que té coneixements del llenguatge de programació, així com de l'entorn de desenvolupament a utilitzar, no ha treballat anteriorment en aquesta plataforma. És per això que des de l'inici

l'estudiant instal·larà i farà proves de l'entorn de desenvolupament, tant pel seu compte com seguin els suggeriments del director del projecte i del consultor del laboratori J2EE. A banda assistirà a un curs de 40h de llenguatge java orientat al desenvolupament d'aplicacions gràfiques i web.

2.4.Descomposició en tasques

Com la majoria de projectes caldrà tenir en compte que s'hauran de realitzar activitats comunes com la gestió, l'anàlisi, el disseny, el desenvolupament, les proves, la documentació, la presentació i la instal·lació. Tenint en compte les característiques del projecte dins del marc de l'assignatura Treball Final de Carrera, caldrà tenir en compte que la presentació serà virtual i que s'establirà una setmana de defensa del treball en base al diàleg estudiant - tribunal per correu electrònic.

a) Definició del projecte
    a. Entrevista client
    b. Recollida informació webs existents
    c. Síntesi final contingut web
b) Planificació del projecte
    a. Introducció al projecte (descripció, esquema modular)
    b. Planificació temporal (anàlisi activitats – descripció, seqüència i durada - agenda i control)
    c. Recursos
    d. Riscos
c) Gestió (control fites i temporalització, col·laboració amb clients a fi d'ajustar al màxim)
d) Anàlisi (recollida requeriments detallats, anàlisi funcional)
e) Disseny
    a. Arquitectura – patrons
    b. Diagrames seqüència, classes
    c. Disseny BBDD
    d. Disseny interfície
    e. Disseny proves
    f. Control errors
f) Implementació
    a. Gestió usuaris

b. BBDD

c. Gestió carta

d. Multiidioma

e. Gestor reserves

f. Gestió comunicació (tauler, comunicació diària, comunicació events)

g. Integració

g) Proves

a. De cada mòdul

b. De cada nivell d'integració

c. Del global del sistema

h) Lliuraments

a. Elaboració document PAC1-Pla de treball

b. Document anàlisi (intern)

c. Document disseny (PAC2)

d. Implementació (PAC3)

e. Memòria (a la fi del projecte, recollint les etapes anteriors i el conjunt del projecte)

f. Instal·lable (programari i documents suport)

g. Presentació virtual

h. Debat

2.5. Temportització i documentació

Dins del marc de l'assignatura Treball Final de Carrera, s'estableixen unes determinades dates i documents a lliurar. Aquestes dates recollides a continuació determinaran en part la temporalització del projecte.

A continuació ho detallem segons el que es defineix en el pla d'estudis del projecte.

4/3/2006	Trobada presencial	Presentació de l'assignatura i introducció del projecte amb el director
13/3/06	Pla de Treball	Entrega del pla de treball, o pac1 d'acord amb els continguts descrits pel director
4/4/06	Anàlisi	Fi etapa anàlisi, no ha de generar cap document a entregar, però si discussió amb director
21/4/06	Disseny	Fi etapa disseny, s'entregarà document al director del projecte
29/5/06	Implementació	Entrega de la implementació del projecte
10/6/06	Trobada de síntesi	Moment de concreció final del document memòria, la presentació i el debat
16/6/06	Entrega final	Es farà entrega del programari, la presentació, la memòria i la informació necessària per l'ús del programa (manual d'instal·lació, d'usuari...)
26/6/06	Inici debat	Debat entre el tribunal i l'estudiant, a fi que el segon resolgui dubtes o justifiqui elements de les diferents etapes del projecte sobre els quals el tribunal presenti qüestions
2/7/06	Fi del debat	

D'acord amb el llistat anterior s'estableixen lliuraments parcials o fites que han d'ajudar al seguiment del projecte. A la fi de les etapes d'especificació, disseny i implementació es lliuraran els documents resultants d'els direrents treballs.

S'estableix com a data de lliurament del projecte la data de la trobada de síntesi, deixant 6 dies de marge per a últims ajustos fins a la data final.

A banda dels lliuraments parcials, a l'entrega final es lliurarà l'aplicació desenvolupada i la memòria del projecte. En la memòria han de quedar reflectides totes les etapes del projecte, incloent així la informació dels lliuraments parcials, especificació i disseny, les notes del procés d'implementació, documentació de classes, funcions, etc. També la

informació necessària per a la gestió de l'aplicació, així com l'ajuda per al seu ús i les notes d'instal·lació.

Caldrà que el director i l'estudiant estableixin una via de comunicació doncs el director ha de poder revisar i si cal comentar i suggerir els documents a lliurar i per tant el treball realitzat durant procés de desenvolupament.

2.6. Calendari Projecte

2.6.1. Anàlisi d'activitats i agenda

El llistat següent mostra la relació de tasques de forma més detallada que en l'apartat 5, així com la càrrega assignada a cada tasca, les tasques predecessores en cas de ser-hi i les dates d'inici fi.

En negreta les fases i en vermell els lliuraments.

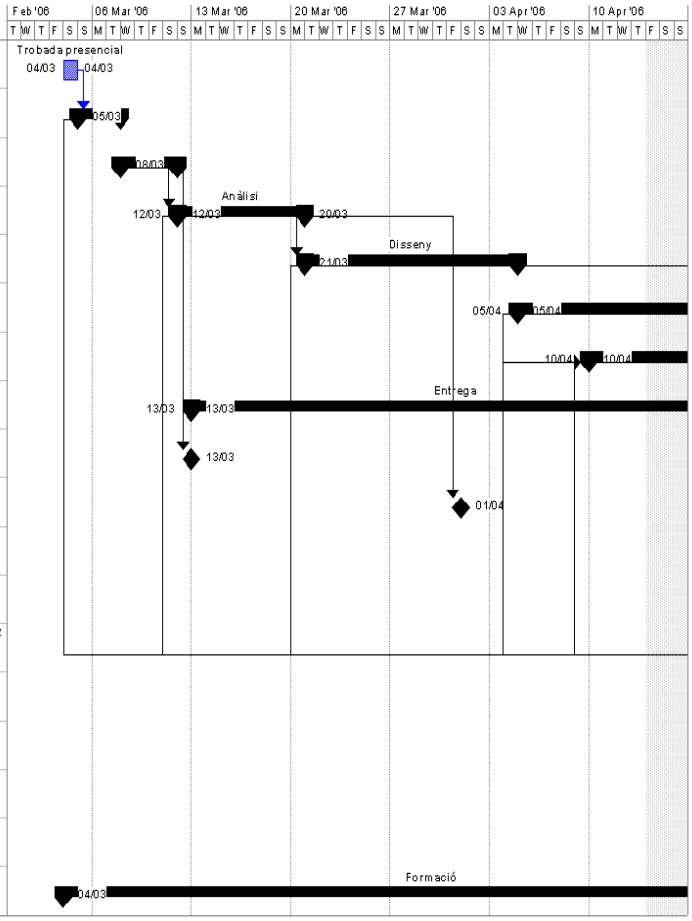
	Activitat	Durada	Predecessora	Inici	Fi
1	**Trobada presencial**	**1 day**		**4-mar-06**	**4-mar-06**
2	**Definició producte**	**3 days**	**1**	**5-mar-06**	**7-mar-06**
3	Entrevista usuaris finals	1 day	1	5-mar-06	5-mar-06
4	Recollida informació altres productes	1 day	3	6-mar-06	6-mar-06
5	Proposta producte	1 day	4	7-mar-06	7-mar-06
6	**Planificació projecte**	**5 days**	**5**	**8-mar-06**	**13-mar-06**
7	Descripció	1 day	5	8-mar-06	8-mar-06
8	Planificació temporal	1 day	7	9-mar-06	9-mar-06
9	Anàlisi activitats	0,5 days		9-mar-06	9-mar-06
10	Agenda I control	0,5 days	9	9-mar-06	9-mar-06
11	Recursos	1 day	10	10-mar-06	10-mar-06
12	Riscos	1 day	11	11-mar-06	11-mar-06
13	Document PAC1	0 days	12	13-mar-06	13-mar-06
14	**Anàlisi**	**19 days**	**6**	**13-mar-06**	**1-abr-06**
15	Revisió casos d'ús	1 day		13-mar-06	13-mar-06
16	Definició paquets	1 day	15	14-mar-06	14-mar-06
17	Classes d'anàlisi (frontera-contol-entitat)	1 day	16	15-mar-06	15-mar-06
18	Definició dades (atributs classes entitats)	2 days	17	16-mar-06	17-mar-06
19	Definició relacions de classes	2 days	18	18-mar-06	19-mar-06
20	Casos d'ús	1 day	19	20-mar-06	20-mar-06
21	Interfície usuari	1 day	20	21-mar-06	21-mar-06
22	Document anàlisi - Model anàlisi	0 days	21	1-abr-06	1-abr-06
23	**Disseny**	**16 days**	**14**	**1-abr-06**	**21-abr-06**
24	Arquitectura general - patrons	2 days		1-abr-06	2-abr-06
25	Disseny Casos d'ús	2 days	24	3-abr-06	4-abr-06
26	Diagrames estàtics	4 days	25	5-abr-06	8-abr-06
27	Disseny persistència (BBDD)	3 days	26	9-abr-06	11-abr-06
28	Disseny interfície usuari	2 days	27	12-abr-06	13-abr-06
29	Presentació dades	0,5 days		12-abr-06	12-abr-06

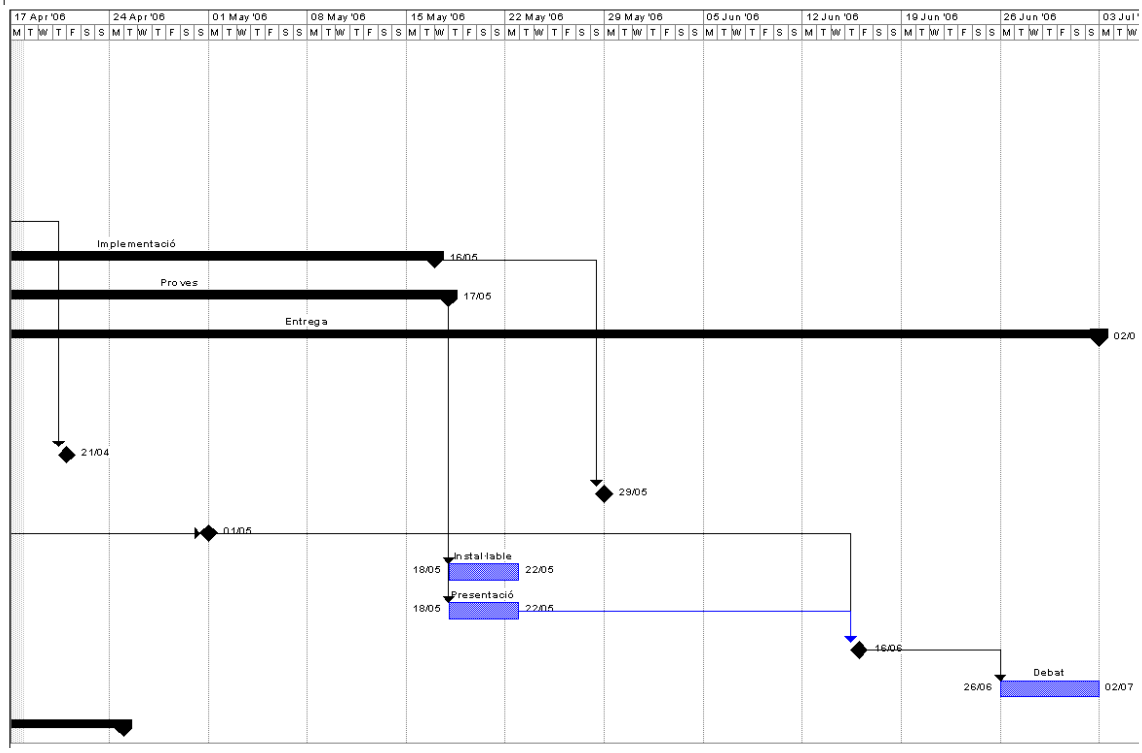
30	Diàlegs	1 day	29	12-abr-06	13-abr-06
31	Format finestres	0,5 days	30	13-abr-06	13-abr-06
32	Disseny proves	1 day	31	18-abr-06	18-abr-06
33	Control errors	1 day	32	19-abr-06	19-abr-06
34	Document Disseny	0 days	33	21-abr-06	21-abr-06
35	**Implementació**	**39 days**	**33**	**20-abr-06**	**29-may-06**
36	BBDD	5 days		20-abr-06	24-abr-06
37	Interfície	10 days	36	25-abr-06	4-may-06
38	Mòduls gestió	16 days	37	5-may-06	20-may-06
39	reserves	5 days		5-may-06	9-may-06
40	comunicació	2 days	39	10-may-06	11-may-06
41	carta	5 days	40	12-may-06	16-may-06
42	usuaris	4 days	41	17-may-06	20-may-06
43	Multiidioma	4 days	38	21-may-06	24-may-06
44	Integració	3 days	43	25-may-06	27-may-06
45	Document Implementació	0 days	44	29-may-06	29-may-06
46	**Proves**	**34 days**	**35SS+2 days**	**25-abr-06**	**28-may-06**
47	Proves BBDD	1 day	36	25-abr-06	25-abr-06
48	Proves interfície	1 day	43	25-may-06	25-may-06
49	Proves gestió	5 days	38	21-may-06	25-may-06
50	Sistema - integració	1 day	44	28-may-06	28-may-06
51	**Entrega**	**63 days**		**1-may-06**	**2-jul-06**
52	Memòria	40 days	2SS;14SS;23SS;35SS;46SS	1-may-06	9-jun-06
53	Instal·lable	5 days	46	29-may-06	2-jun-06
54	Presentació	5 days	46	29-may-06	2-jun-06
55	Entrega Documentació	0 days	54	16-jun-06	16-jun-06
56	Debat	7 days	55	26-jun-06	2-jul-06
57	**Formació**	**48 days**		**4-mar-06**	**24-abr-06**
58	Instal·lació plataforma	3 days		4-mar-06	6-mar-06
59	Proves I exercicis	5 days	58	7-mar-06	11-mar-06
60	Curs Java	30 days	58SS+5 days	22-mar-06	24-abr-06

2.6.2.Temporització – Diagrames Gantt

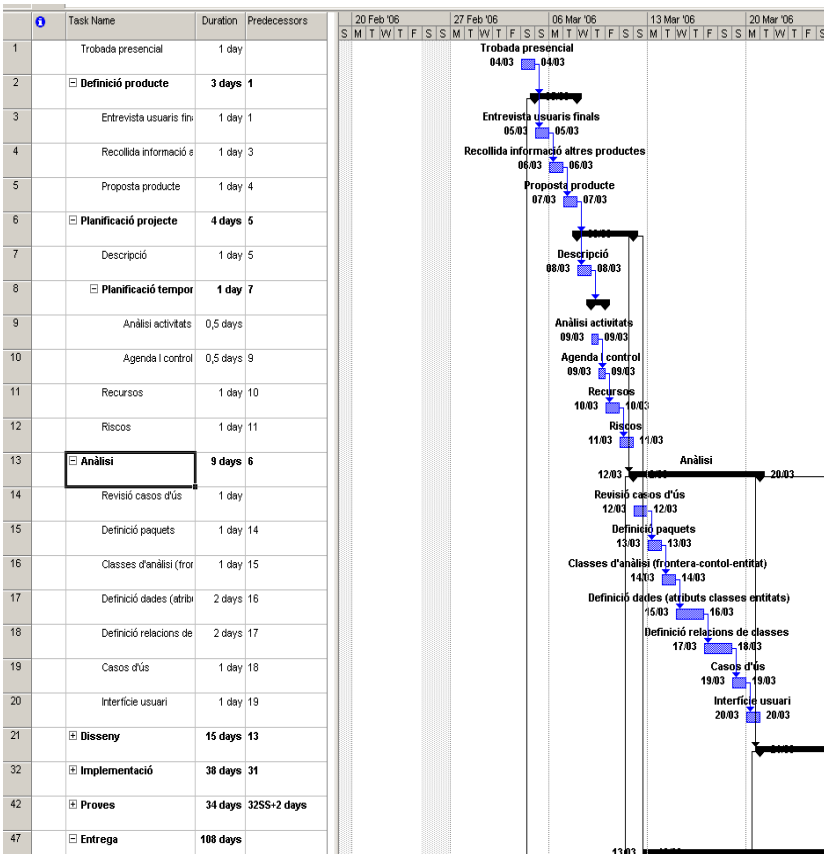
El següent diagrama mostra de forma gràfica la temporalització de les tasques.

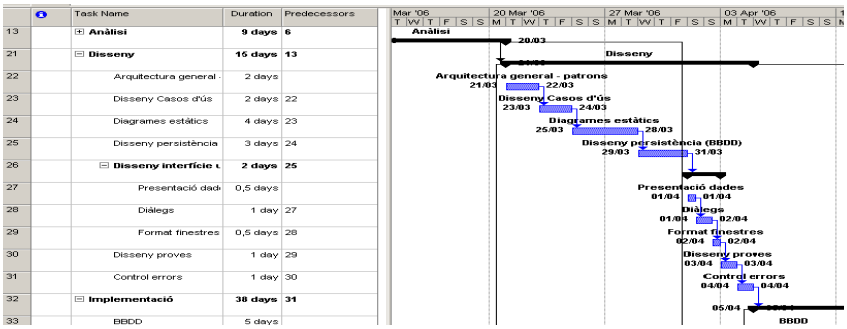
ID		Task Name	Duration	Predecessors
1		Trobada presencial	1 day	
2		Definició producte	3 days	1
6		Planificació projecte	4 days	5
13		Anàlisi	9 days	6
21		Disseny	15 days	13
32		Implementació	38 days	31
42		Proves	34 days	32SS+2 days
47		Entrega	108 days	
48		Document PAC1	0 days	6
49		Document anàlisi- Model a	0 days	13
50		Document Disseny	0 days	21
51		Document Implementació	0 days	32
52		Memòria	40 days	2SS;13SS;21SS;32
53		Instal·lable	5 days	42
54		Presentació	5 days	42
55		Entrega Documentació	0 days	54;52
56		Debat	7 days	55
57		Formació	48 days	

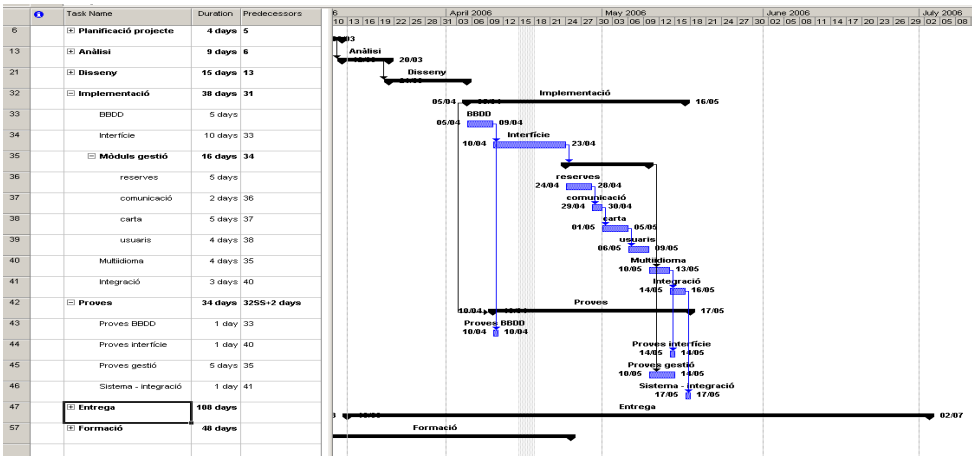


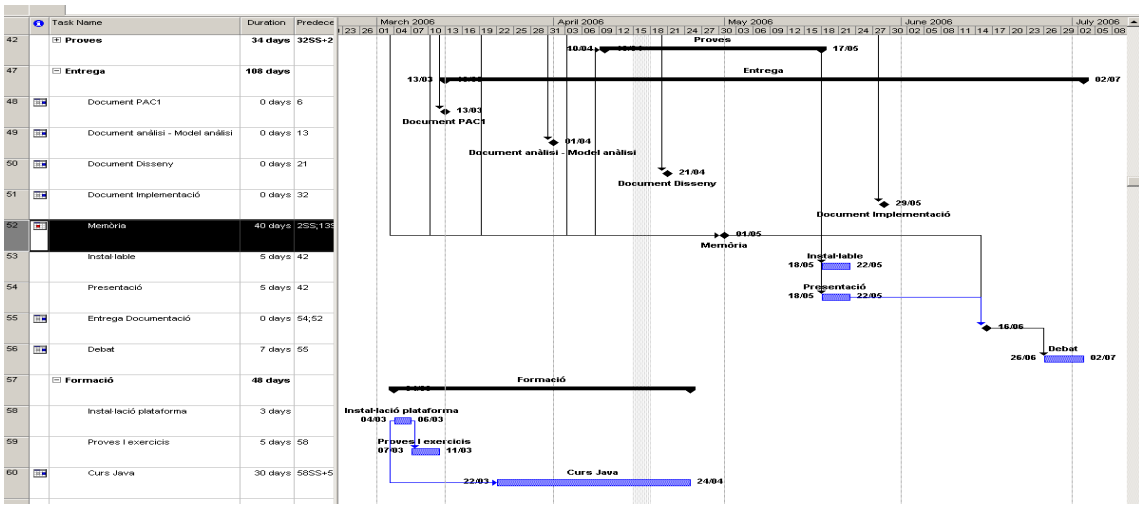


I els següents diagrames mostren el detall de cadascuna de les diferents fases






Task Name
Duration
Predecessors
Planificació projecte 4 days 5
Anàlisi 9 days 6
Disseny 15 days 13
Implementació 38 days 31
BBDD 5 days
Interfície 10 days 33
Mòduls gestió 16 days 34
reserves 5 days
comunicació 2 days 36
carta 5 days 37
usuaris 4 days 38
Multidioma 4 days 35
Integració 3 days 40
Proves 34 days 32SS+2 days
Proves BBDD 1 day 33
Proves interfície 1 day 40
Proves gestió 5 days 35
Sistema - integració 1 day 41
Entrega 108 days
Formació 48 days
April 2006
May 2006
June 2006
July 2006


Task Name
Duration
Predece
Proves 34 days 32SS+2
Entrega 108 days
Document PAC1 0 days 6
Document anàlisi - Model anàlisi 0 days 13
Document Disseny 0 days 21
Document Implementació 0 days 32
Memòria 40 days 2SS;13
Instal·lable 5 days 42
Presentació 5 days 42
Entrega Documentació 0 days 54;52
Debat 7 days 55
Formació 48 days
Instal·lació plataforma 3 days
Proves i exercicis 5 days 58
Curs Java 30 days 58SS+5
March 2006
April 2006
May 2006
June 2006
July 2006

3. *Especificació i anàlisi*

En aquest apartat es revisen els requisits i s'analitzen els actors implicats, així com els casos d'ús i una primera aproximació al diagrama estàtic, en base a la informació recollida a l'apartat anterior.

Cal remarcar que tal s'ha explicat en l'apartat del pla de treball, totes les funcionalitats inicials no seran implementades, així que no tindrem present les funcionalitats de tauler d'anuncis i serveis.

En la primera part del document es revisen els requisits, en una segona part s'analitzen els actors i els casos d'ús relacionats amb cadascun d'ells i el seu detall. Seguidament es veurà el diagrama estàtic i en l'última part es recull un anàlisi de la interfície de l'usuari.

3.1.Documentació dels requisits

3.1.1.Model de domini.

En base al plantejament inicial s'identifiquen els objectes reserva, places, carta, menú, subscripció, local, anuncis, serveis com a objectes del món exterior.

Pel que fa als objectes del domini identifiquem: petició de reserva, petició d'informació, publicació d'anuncis, petició de subscripció, canvi de carta, canvi del menú del dia, impressió de menús i cartes, publicació anuncis, publicació de serveis.

Com a esdeveniments cal destacar la publicació automàtica del butlletí, l'enviament del menú del dia, i la pre confirmació de la reserva per al tipus de servei menú.

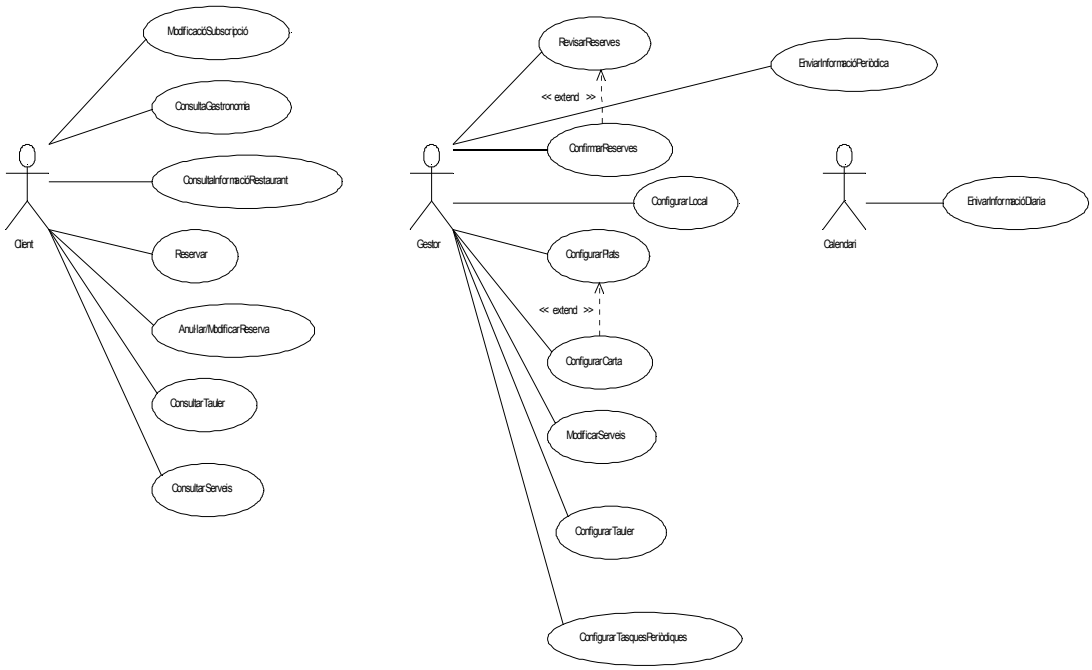
3.1.2.Model de negoci.

De la descripció inicial del projecte identifiquem dos rols, el Client i el Gestor, on el client és la persona client del restaurant que visita la pàgina per informar-se o fer reserves, el gestor és una persona relacionada amb el restaurant que s'encarregaran d'administrar la informació que es publiqui a la pàgina.

En el següent diagrama veiem com cada actor es relaciona amb determinades tasques

La relació d'aquests casos d'ús amb els objectes identificats és la que segueix:

1. ModificacióSubscripció (subscripció, client)
2. ConsultarGastronmia (menú, carta, plats, begudes)
3. Reservar (dia, franja_horaria, taula, dades_reserva)
4. ConsultarTauler i ModificarTauler (tauler, anuncis)
5. ConsultarServeis i ModificarServeis (llistat_serveis, servei)
6. Revisar i ConfirmarReserves (dia, franja_horaria, taula, dades_reserva, client)
7. ConfigurarLocal (local, aforament, taules, agrupació taules)



3.1.2.1.Glossari del model de negoci

Cal tenir en compte que en aquest tipus d'establiment s'utilitza un llenguatge molt específic, a continuació relacionem el glossari del model de negoci.

1. **menú** Llista dels diferents plats que componen un àpat.
2. **carta** Llista dels plats i de les begudes que se serveixen en un restaurant amb els preus corresponents. Menjar a la carta.
3. **restaurant** Establiment en què se serveixen àpats, menjars.
4. **reserva**:Destinar exclusivament un lloc, una cosa, per a una persona determinada. Els hem reservat aquesta taula.
5. **Reservar**:Excloure una cosa del nombre de les coses de què hom pot disposar lliurement. Seients reservats a les autoritats.
6. **Client**:Persona que habitualment fa les compres o se serveix en un establiment obert al públic.
7. aforament: capacitat del local, configuració taules
8. **plat**: unitat d'alimentació, es poden classificar per tipus aliment(carns, peix, ...) o per l'ordre dins l'àpat (primers, segons...)
9. **subscripció**: Acte pel qual una persona s'adhereix a una societat en canvi de l'adquisició d'una o més accions. En aquest cas a la recepció de notícies relacionades amb el restaurant o bé del menú diari.
10. **anunci**: Acte pel qual una persona s'adhereix a una societat en canvi de l'adquisició d'una o més accions. S'avisarà d'actes que es facin, canvis en la carta
11. **gestor**: persona que portarà els afers relacionats amb la pàgina, actualització de dades, reserves....

3.2.Guions

Identifiquem diversos guions relacionats amb els diferents actors, només tindrem en compte aquells que són més rellevants, ja que no són una mera cerca de petició.

Client que visita la pàgina per fer reserva: El client accedeix a la pàgina i cerca en l'oferta gastronòmica els plats i els tipus de servei. Accedeix al formulari de reserves i l'omple. A la fi sol·licita la reserva. Com que ha demanat carta rebrà un correu de pre confirmació però la reserva no serà confirmada fins que el Gestor no l'accepti.

Gestor que vol modificar l'oferta gastronòmica: El gestor s'identifica en el sistema el que li permet accedir a l'àrea reservada. Accedeix a l'aplicació per modificar l'oferta gastronòmica. Selecciona la carta de divendres i dissabtes a la nit. Treu plats que no són de temporada i n'afegeix alguns que ja estan a la llista de plats. Li falta un plat així que el dóna d'alta i s'afegeix al mateix temps a la configuració de la carta. Desa els canvis.

El sistema li permet generar una impressió de la carta modificada que servirà per omplir les cartes que s'entreguen als clients quan arriben al restaurant.

Gestor que vol revisar les reserves: El gestor s'identifica en el sistema el que li permet accedir a l'àrea reservada. Accedeix a l'aplicació per revisar les reserves. Visualitza les reserves pendents de confirmar. Selecciona i n'accepta algunes, mentre que d'altres les descarta, amb aquesta acció s'envia un correu al client. El gestor imprimeix la llista de les reserves confirmades a fi de trucar personalment i parlar amb el client. S'acosta l'hora de dinar i cal veure les reserves que hi ha. Indica al sistema que vol el llistat de les reserves pel mateix dia en la franja horària del dinar. Confirma la selecció i el sistema li mostra la llista que pot imprimir.

3.3.Casos d'ús, plantejament i anàlisi.

En els següents apartats s'identificaran i plantejaran les informacions dels diferents casos d'ús corresponents a la situació plantejada.

3.3.1.Actors

Els actors implicats en aquesta situació són dos, el client i el gestor. Tots són usuaris finals del sistema, ambdós interactuaran amb ell.

El client té diversos papers entre els que s'inclouen la cerca d'informació, així com la petició de subscripció a serveis, o la reserva de taula. El Gestor s'encarregarà d'alimentar la informació de la pàgina, gestionant la informació de plats i menús i cartes, confirmant o modificant les reserves i configurant la disposició del local.

3.3.2.Descripció dels casos d'ús

3.3.2.1.Consultar Gastronomia

Nom: ConsultarGastronmia

Funcionalitat: L'usuari accedeix a l'apartat per revisar el contingut del menú o carta, pot seleccionar el tipus d'informació a veure, el tipus de servei, és a dir visualitzar els diferents menús o la carta i accedir a la informació de cada plat.

Paper dins el treball de l'usuari: És una tasca habitual en els clients fidelitzats, i més puntual en la resta.

Actors: Client

CdU Relacionats: Fer reserva, ModificarGastronomia

No té pre condicions o post condicions

3.3.2.2.Fer reserves

Nom: FerReserva

Funcionalitat: L'usuari fa una petició de reserva indicant dia, hora i les seves dades de contacte si no és un usuari registrat. En cas que sigui reserva per menú rebrà confirmació automàtica mentre que si és per carta caldrà que el restaurant confirmi la pre confirmació que li ha fet el sistema

Paper dins el treball de l'usuari: S'espera que sigui una tasca habitual quan puguin enviar o accedir a la reserva des del correu electrònic, i més tenint en compte el volum de reserves que ja reben per telèfon

Actors: Client

CdU Relacionats: GestionarReserves, ConsultarGastronmia

No té pre condicions o post condicions

3.3.2.3.Registre usuari

Nom: registre usuari

Funcionalitat: l'usuari accedeix a l'apartat de subscripcions via la introducció del correu electrònic per identificar-se. Si no es coneix el correu electrònic el sistema mostrarà pàgina alta, altrament permetrà modificar les opcions de subscripció (que es mostraran) o fer una baixa. En cas de fer una alta o una modificació els clients informaran del nom i cognoms, del telèfon de contacte i del correu electrònic, també dels serveis que desitgin rebre per correu electrònic, com per exemple el menú del dia i o els anuncis i notícies. Finalment caldrà acceptar o cancel·lar els canvis

Actors: Client

CdU Relacionats:

Pre condicions: alta: l'usuari no està en el sistema

modificació o baixa: l'usuari està donat d'alta en el sistema

Post condicions: alta: l'usuari apareix en el sistema

modificació: l'usuari està en el sistema i s'han modificat les seves dades

baixa: l'usuari desapareix del sistema.

3.3.2.4.Modificació Subscripció

Nom: modificaciósubscripció

Funcionalitat: l'usuari accedeix a l'apartat de subscripcions via la introducció del correu electrònic per identificar-se. Si no es coneix el correu electrònic el sistema mostrarà pàgina alta, altrament permetrà modificar les opcions de subscripció (que es mostraran) o fer una baixa. En cas de fer una alta o una modificació els clients informaràn del nom i cognoms, del telèfon de contacte i del correu electrònic, també dels serveis que desitgin rebre per correu electrònic, com per exemple el menú del dia i o els anuncis i notícies. Finalment caldrà acceptar o cancel·lar els canvis

Actors: Client

CdU Relacionats:

Pre condicions: alta: l'usuari no està en el sistema modificació o

baixa: l'usuari està donat d'alta en el sistema

Post condicions: alta: l'usuari apareix en el sistema modificació: l'usuari està en el sistema i s'han modificat les seves dades

baixa: l'usuari desapareix del sistema.

3.3.2.5.Gestió de Reserves

Nom: GestioReserves

Funcionalitat: Gestionar les reserves fetes pels clients tant de menús com de carta, Permet llistar les reserves existents segons franges horàries i/o per l'estat de la reserva. El gestor accedeix a l'apartat de gestió de reserves i selecciona criteris de visualització (segons dies, hores, categoria llistat plats, segons estat reserves...). Les reserves que necessitin confirmació es podran seleccionar del llistat i activar procés de confirmació o denegació

Actors: Gestor

CdU Relacionats: FerReserva, GestióLocal

Pre condicions: existència de reserves

Post condicions:reserves de menú: cap reserves de carta: S'ha enviat correu d'acceptació o denegació de la reserva al client i s'ha actualitzat l'estat de la reserva en el sistema

3.3.2.6.Configuració del restaurant

Nom: ConfigurarRestaurant

Funcionalitat: Ha de permetre informar de les característiques del local que hagin de ser utilitzades per a la gestió de les reserves, dels llistats de plats, etc.

Un cop l'usuari accedeixi a l'opció del menú, actualitzarà:

dies de la setmana d'obertura

dies festius de l'any

franges de la cuina (hora inici i hora fi, tipus servei),

torns dins de cada franja

total de places

total de taules de 1 (barra), 2 (taules)

També pot accedir des d'aquí a la gestió de les llistes de plats i begudes (llistatsElementsApats), i a la gestió dels elements servits (begudes, tapes, plats....)

Actors: Gestor

CdU Relacionats: FerReserva, GestionarReserves

Pre condicions:

Post condicions: les modificacions han de quedar registrades en el sistema

3.3.2.7.Gestió de la oferta gastronòmica

Nom: GestioOfertaGastronomica

Funcionalitat: Inclou els casos d'ús GestioCategoriesOfertaGastronomica, GestioLlistaElementsApat, GestioElementApat

Actors: Gestor

CdU Relacionats: GestioLlistaElementsApat, GestioElementApat

Pre condicions: Les modificacions han de quedar registrades en el sistema.

Post condicions:

Nom: GestioCategoriesOfertaGastronomica

Funcionalitat: Permetre gestionar la classificació de l'oferta dels elements que el restaurant serveix, ja sigui de gastronomia o del celler.

Els elements que es serveixen en el restaurant s'agrupen LlistesElementsApats, és a dir plats que s'ofereixen en conjunt dins d'una carta o d'un menú.

Aquestes LlistesElementsApats s'agrupen en categories, com per exemple:

menú unipersonal (menú del dia, menú de dissabte, menú de degustació....)

menú de grup (menú_1, menú_2...)

carta (carta gastronòmica, carta del celler, carta de postres)

Dins d'aquesta aplicació l'usuari pot modificar tant les categories de les llistes com la classificació de cada llista en una categoria o altre, podent pertànyer a diversos elements.

Actors: Gestor

CdU Relacionats: GestioLlistaElementsApat, GestioElementApat

Pre condicions: Les modificacions han de quedar registrades en el sistema.

Post condicions:

Nom: GestioLlistaElementsApat

Funcionalitat: Permetre gestionar la els elements que componen una llista d'àpat així com les característiques de la mateixa.

L'usuari seleccionarà si s'escau una llista i indicarà l'operació modificar o esborrar del sistema. També podran donar d'alta una nova llista. Un cop hagi seleccionat l'operació podrà modificar les característiques de la llista (tipus de servei, descripció, nom ...) així com editar (afegir, eliminar o modificar) els elements que componen la llista.

Actors: Gestor

CdU Relacionats: GestioCategoriesOfertaGastronomica, GestioElementApat, GestioOfertaGastronomica

Pre condicions: Les modificacions han de quedar registrades en el sistema.

Post condicions:

Nom: GestioElementApat

Funcionalitat: Permetre gestionar els elements que componen una llista d'àpat, és a dir els plats, begudes, postres, etc que se serveixen en el restaurant.

L'usuari seleccionarà si s'escau un element de la llista de tots els possibles elements que se serveixen. Aquesta llista podrà ser filtrada per categories d'elements, llistesElementsApats, etc i indicarà l'operació modificar o esborrar del sistema. També podran donar d'alta un nou element. Un cop hagi seleccionat l'operació podrà modificar les característiques de l'element (categoria, descripció, nom, preu...) així com indicar a quina llista/es pertany (afegir, eliminar o modificar).

Actors: Gestor

CdU Relacionats: GestioCategoriesOfertaGastronomica, GestioLlistaElementsApat, GestioOfertaGastronomica

Pre condicions: Les modificacions han de quedar registrades en el sistema.

Post condicions:

4. Disseny

En aquest apartat es recull el procés de disseny de l'aplicació per la pàgina web del restaurant Gat Blau. És la continuació de l' apartat anterior, especificació i anàlisi. En base a l'evolució de l'anàlisi feta es detalla el disseny del diferents casos d'ús, del diagrama estàtic, així com de la interfície. També es revisa l'arquitectura utilitzada, els patrons i les tecnologies implementades.

4.1.Disseny arquitectònic

4.1.1.Patró MVC

Tal com ja s'ha discutit anteriorment, la implementació d'una aplicació web basada en J2EE és un dels objectius principals d'aquest projecte. El disseny d'aquest tipus d'aplicacions és una problemàtica comú que s'ha resolt en la descripció del patró model-vista-controlador. És per aquest motiu que s'ha decidit utilitzar el patró model-vista-controlador (MVC) de Smalltalk com a base del disseny de l'aplicació web.

En disseny d'una aplicació web i en general de les aplicacions amb interfície gràfica d'usuari, les interfícies tendeixen a canviar sovint, i es requereixen una certa flexibilitat, en el moment d'implementar els nus canvis. Si les interfícies d'usuari incorporen la funcionalitat de l'aplicació és molt difícil implementar i mantenir aquestes aplicacions i alhora i complir alhora amb els requeriments de flexibilitat exigits.

El patró MVC soluciona aquest problema, ja que proposa una estructura que permet diferenciar la lògica de negoci de la interfície d'usuari. El MVC divideix en tres àrees la funcionalitat d'una aplicació interactiva, de manera que la lògica queda dins l'àrea del model, la sortida dins l'àrea de la vista i el controlador gestiona les entrades, o accions d'usuari.

Dins l'àmbit del model trobem les dades i la funcionalitat, que ha de ser independent de les representacions o interfícies d'usuari, i del comportament de les entrades. L'àmbit de les vistes s'encarregarà de mostrar les informacions que rebi del model a l'usuari.

Cada una de les vistes que es descriguin tindrà un controlador associat que recollirà les entrades (events tals com moviments de ratolí, teclat, etc) i les traduirà en peticions tant a les vistes com al model, d'aquesta manera el controlador permet que l'usuari interactui amb el sistema. Aquest patró també descriu els mecanismes de propagació de canvis en l'estat del model que han de ser reflectits en les vistes

El controlador rep les peticions que a través del navegador que utilitza l'usuari. Al seu temps el Controlador fa la petició adient al Model que és qui implementa la lògica de negoci de l'aplicació i l'accés a la base de dades. El controlador inicialitza la vista que s'ha de mostrar a l'usuari i aquesta interactua amb el model a fi de mostrar les dades adequades.

També tindrem en compte que d'acord amb el framework que utilitzarem per implementar el MVC, podrem separar la lògica de negoci i la base de dades del model

de l'aplicació web, fent que la classe que ha d'implementar el model, actuï de pont entre l'aplicació web i l'aplicació amb la lògica de negoci. Això facilitaria entre altres provar els dos blocs per separat.

Particularitzant aquest punt en l'àmbit del framework utilitzat (struts), direm que la classe que implementa el model (Action), només ha de controlar el flux de l'aplicació, deixant la lògica d'aquesta en un altre paquet, aquesta estructura permetrà més flexibilitat i reusabilitat.

4.1.2.Paquets

Aquesta aplicació es dividirà en dos grans blocs, d'una banda la interfície d'usuari que s'implementarà en base el famework Struts d'Apache, i de l'altre la lògica de negoci, integrada per diferents mòduls, com:

a) Gestió Restaurant (gestió local, gestió serveis, gestió tauler, gestió clients)

b) Gestió Reserves (gestió reserves)

c) Gestió Gastronomia (gestió llistes elements, gestió elements àpats)

4.1.3.Tecnologies utilitzades

Dins del món J2EE s'inclou moltes tecnologies que proporcionen diferents eines per a la implementació d'aplicacions dins del seu àmbit. En aquest apartat veurem de forma general les eines escollides per a la implementació del projecte.

1. Implementació MVC-Interfície usuari: En base a Struts d'Apatche, que és un dels frameworks que es poden fer servir per implementar aquest patró.

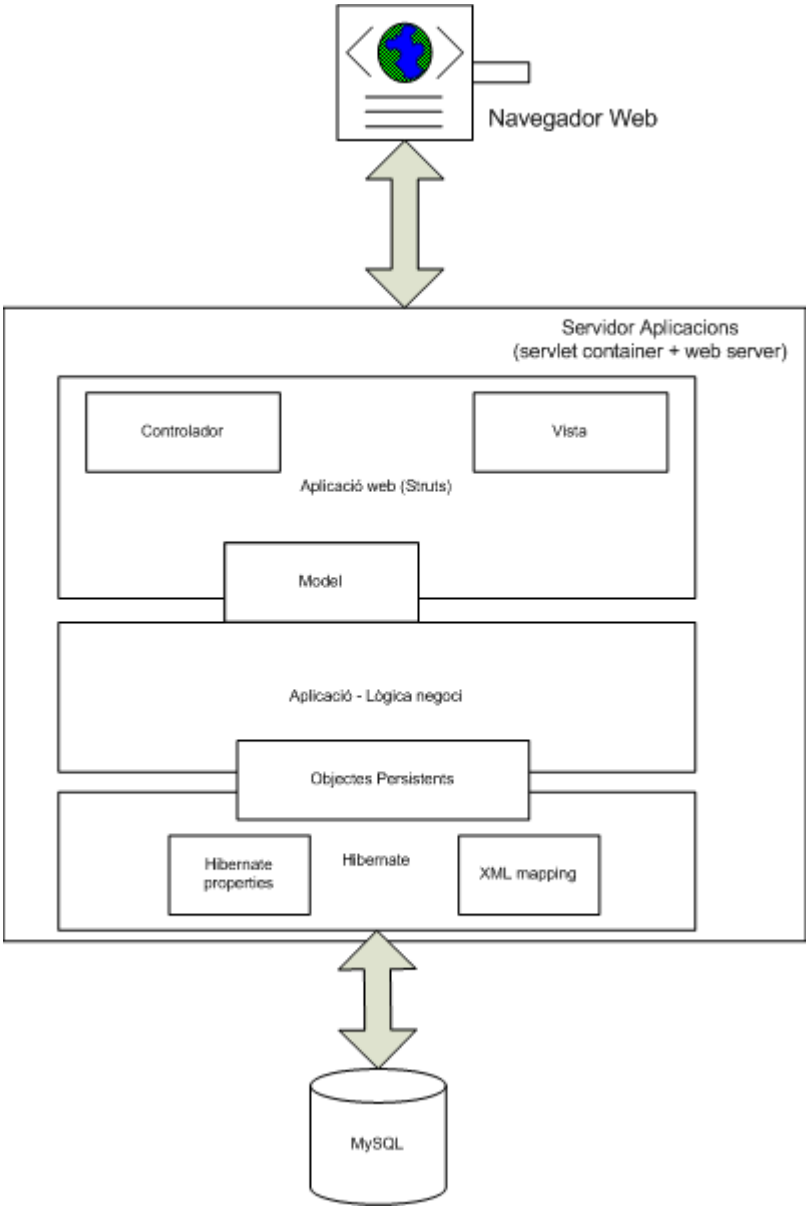
2. Separació continguts i layout: En base al framework Struts Tiles també d'Apache, permet diferenciar el contingut del continent en les vistes

3. Implementació de les vistes: Struts-layout és una llibreria d'etiquetes que ens ajudarà a la construcció de les vistes

4. Implementació gestors: en base a POJO's, és a dir les tradicionals classes en java

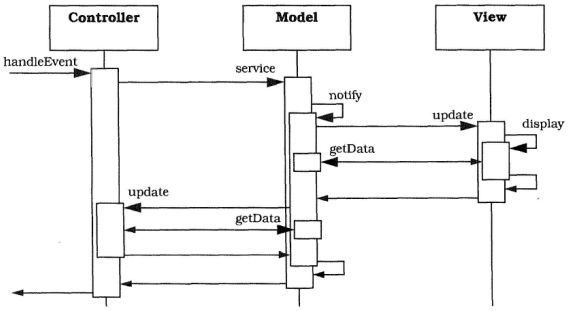
5. Model relacional: Hibernate, eina que ens permet fer l'ORM (object/relational mapping) entre els objectes i les entitats de la base de dades.

4.1.4.Diagrama de l'Arquitectura



4.1.5.Revisió dels casos d'ús

Tenint en compte que per implementar el MVC hem escollit Struts, d'acord amb aquest patró les classes frontera que hem descrit en l'anàlisi tindran la seva versió vista i la seva versió controlador.



D'acord amb la situació que planteja el patró i el framework utilitzat per a la seva implementació, diagrama de seqüència de la interacció entre el model, la vista i el controlador és el que es mostra a continuació, malgrat tot per no complicar en excés els diagrames de seqüència dels casos d'ús hem considerat encara les classes frontera mantenint plegats la vista i el controlador.

4.1.6.Validació i gestió usuaris

En el plantejament del disseny s'ha relacionat una nova funcionalitat, i aquesta és la identificació de l'usuari gestor en el sistema. Per aquesta etapa del desenvolupament considerarem un sol usuari gestor, amb una única identificació. Malgrat això deixarem el disseny preparat per tal que en el futur es puguin afegir més usuaris i també es pugin delimitar els drets d'aquests usuaris.

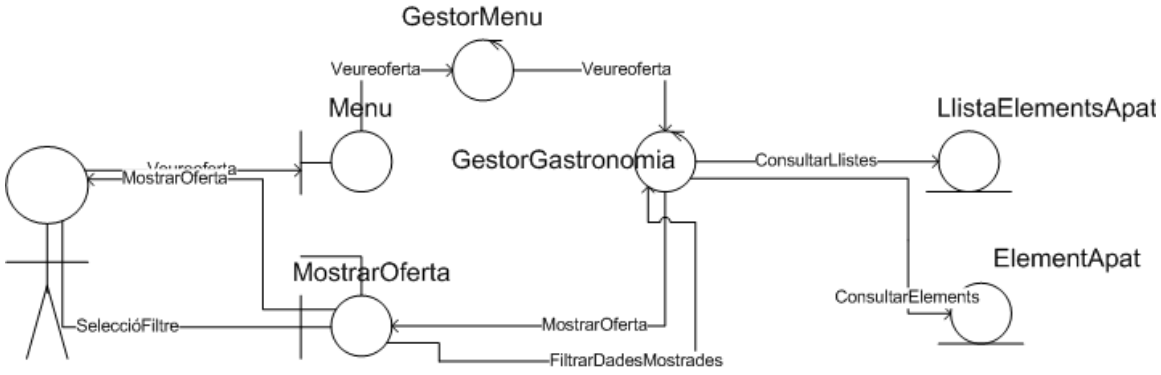
Cal incloure doncs una classe Usuari, una classe FuncionalitatsDisponibles

Caldria doncs veure el cas d'ús GestioUsuaris així com IdentificacioUsuari, però queden fora de l'abast del desenvolupament actual i per això no es consideren.

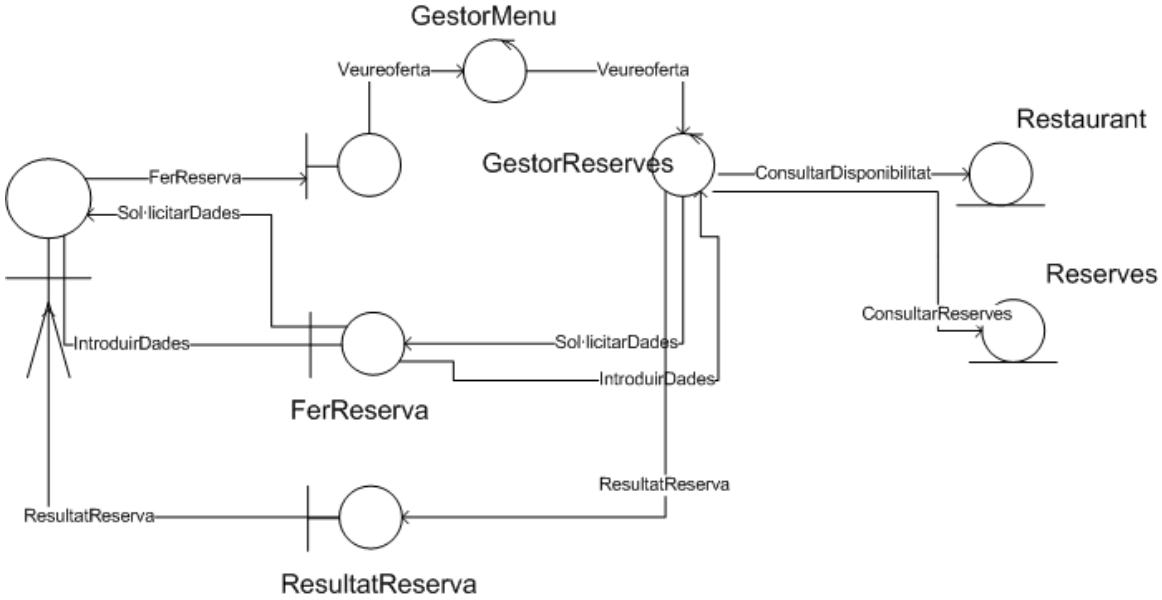
Cal però implementar la identificació de l'usuari gestor per tal que pugui entrar en l'aplicació de gestió del restaurant, i també per tal d'evitar que usuaris que no s'hagin validat a l'aplicació puguin entrar a pàgines de gestió mitjançant enllaços intermedis

4.1.7.Revisió Casos d'ús

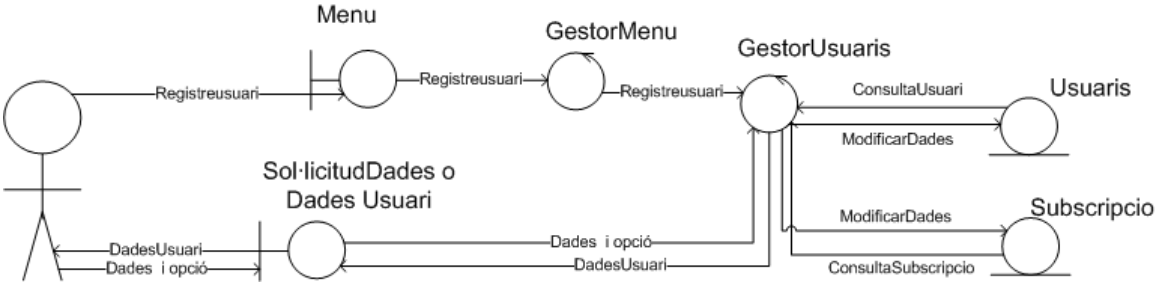
4.1.7.1.Consultar Gastronomia



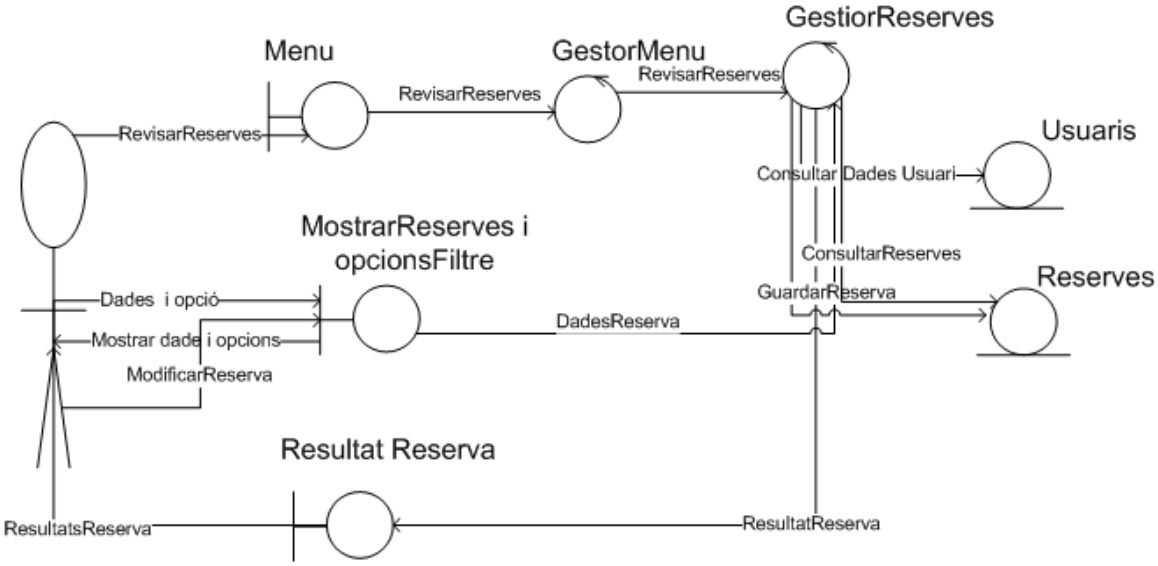
4.1.7.2.Fer reserves



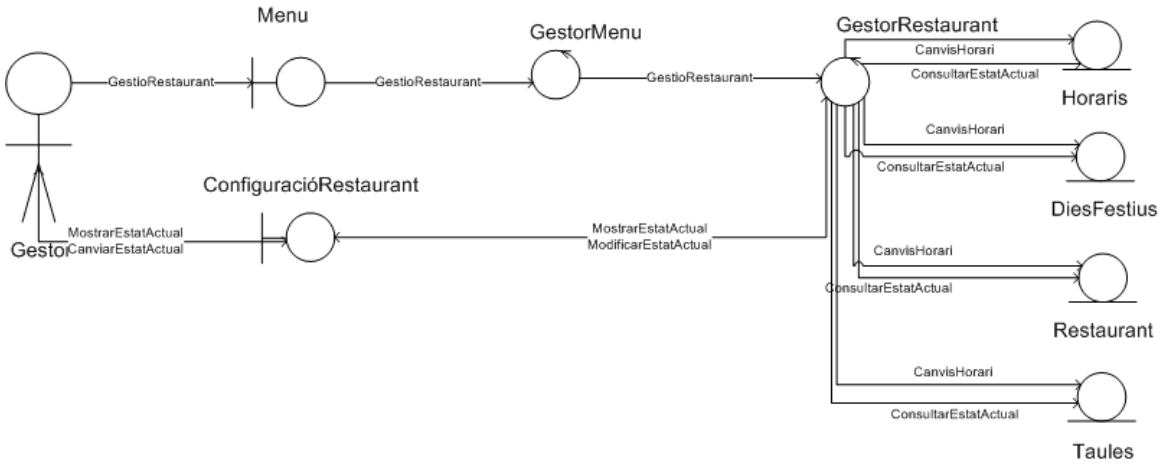
4.1.7.3.Registre usuari



4.1.7.4.Gestió de Reserves



4.1.7.5.Configuració del restaurant



4.1.7.6.Gestió de la oferta gastronòmica

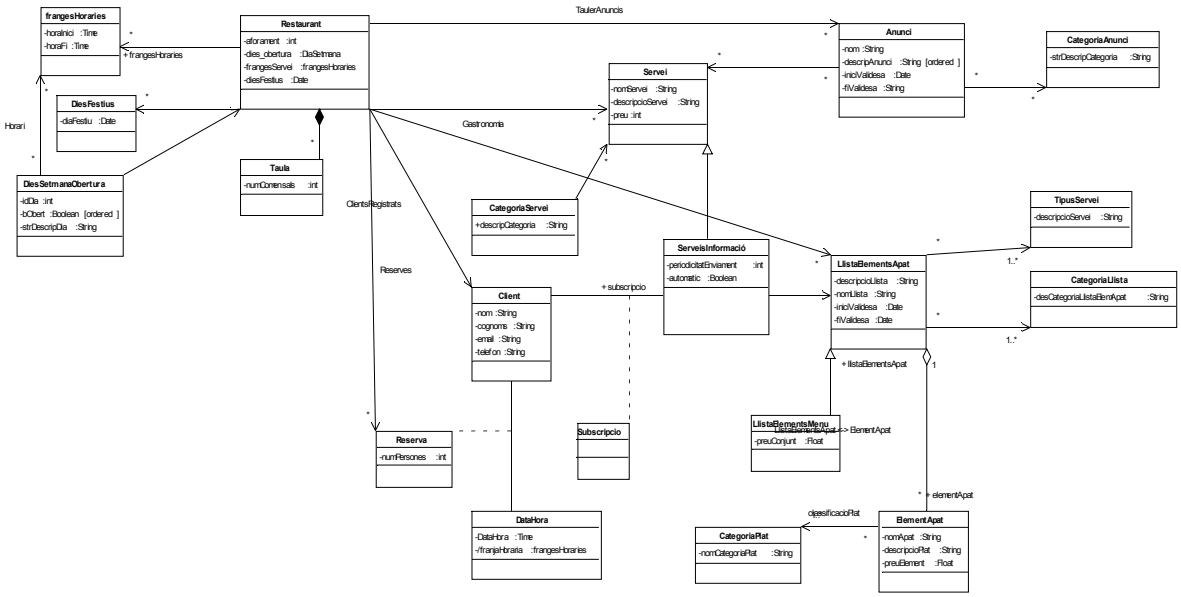
En aquest cas no es mostra el diagrama de seqüències, de forma general els passos a implementar seran:

1. l'usuari selecciona en el Menú l'opció GestionarOfertaGastronomica,
2. el GestorMenu processarà aquest event enviant la petició al GestorGastronomia,
3. el GestorGastronomia consultarà l'estat de les LlistesElementsApats
4. el GestorGastronomia enviarà a la classe frontera OfertaGastronomica les dades corresponents
5. l'usuari modificarà les dades que cregui oportunes
6. OfertaGastronomica recull els canvis i els transmet al GestorGastronomia
7. que enviarà les dades corresponents a LlistesElementsApats per tal que guardin les dades.

De forma similar funcionaran els casos d'ús de GestioElementApat, GestioCategoriesOfertaGastronomica, però enlloc d'accedir a les LlistesElementsApats s'accedirà a altres classes entitat.

4.2.Diagrama estàtic

Respecte el diagrama d'anàlisi s'han descrit noves entitats que han d'ajudar a la flexibilitat de l'aplicació.



4.3.Disseny de la persistència

En base al diagrama estàtic de disseny l'estructura de les taules de la base de dades quedarà com segueix.

La convenció per definir els noms dels camps de la base de dades és:

XXX_TYYYYY

on XXX són tres lletres que identifiquen la taula, T el tipus de dades i YYYY el nom descriptiu del camp.

S: cadena caràcters

N: enter

F: float

T: hora

D: date

RESTAURANT (RES_NID, RES_NAFORAMENT)

TAULES (TAU_NID, TAU_NNUMMINPERSONES, TAU_NUMMAXPERSONES)

FRANGES_HORARIES (FRH_NID, FRH_THORAINICI, FRH_THORAFI)

DIES_FESTIUS(DFE_DDATA_FESTIU)

DIES_SETMANA_OBETURA(DSO_NID, DSO_SDESCRIPDIA, DSO_BOBERT)

HORARI(HOR_NFRH_NID, HOR_NDSO_NID)

CLIENT(CLI_NID, CLI_SNOM, CLI_SCOGNOM1, CLI_SCOGNOM2, CLI_SEMAIL, CLI_STELEFON**)**

RESERVA(RES_NCLI_NID, RES_TDATAHORA)

SERVEI(SER_NID, SER_SNOM, SER_SDESCRIPCIO, SER_FPREU)

SERVEI_INFO(SEI_NSER_NID, SEI_NPERIODICITAT, SEI_BAUTOMATIC)

CATEGORIA_SERVEI(CSR_NID, CSR_SDESCRIPCATSERVEI)

RELACIO_SERVEI_CATEGORIA(RSC_NSER_NID , RSC_N CSR_NID)

SUBSCRIPCIO (SUB_NCLI_NID , SUB_NSER_NID)

LLISTES_ELEMENTS_APATS(LEA_NID, LEA_SDESCIPLLISTA, LEA_SNOMLLISTA, LEA_DDATAINICI, LEA_DDATAFI, LEA_NCLS_NID)

CLASSIFICACIO (CLS_NID, CLS_SDESCRIPCLASSIFICACIO)

LLISTES_MENU(LME_NLEA_NID, LME_NPREUMENU)

ELEMENTS_APATS(ELA_NDI, ELA_SNOMELEMENT, ELA_SDESCRIPELEMENT, ELA_FPREUELEMENT)

CATEGORIA_LLISTES_ELEMENT(CLE_NID, CLE_SDESCRIPCATEGORIA)

CATEGORIA_ELEMENT(CTE_NID, CTE_SDESCRIPCATEGORIA)

TIPUS_SERVEI(TSE_NID, TSE_DESCRIPSERVEI)

RELACIO_LLISTES_ELEMENTS(RLE_NLEA_NID , RLE_NELA_NDI)

RELACIO_ELEMENTS_CATEGORIES(REC_NCTE_NID, REC_NELA_NDI)

RELACIO_LLISTES_CATEGORIES(RLC_NCLE_NID, RLC_NLEA_NID)

RELACIO_LLISTES_TIPUS(RLC_NTSE_NID , RLC_NLEA_NID)

ANUNCIS(ANU_NID, ANU_STITOL, ANU_SDESCRIPANUNCI, ANU_DINICIVALIDESA, ANU_DFIVALIDESA)

CATEGORIA_ANUNCIS(CAN_NID, CAN_SDESCRIPCATEGORIA)

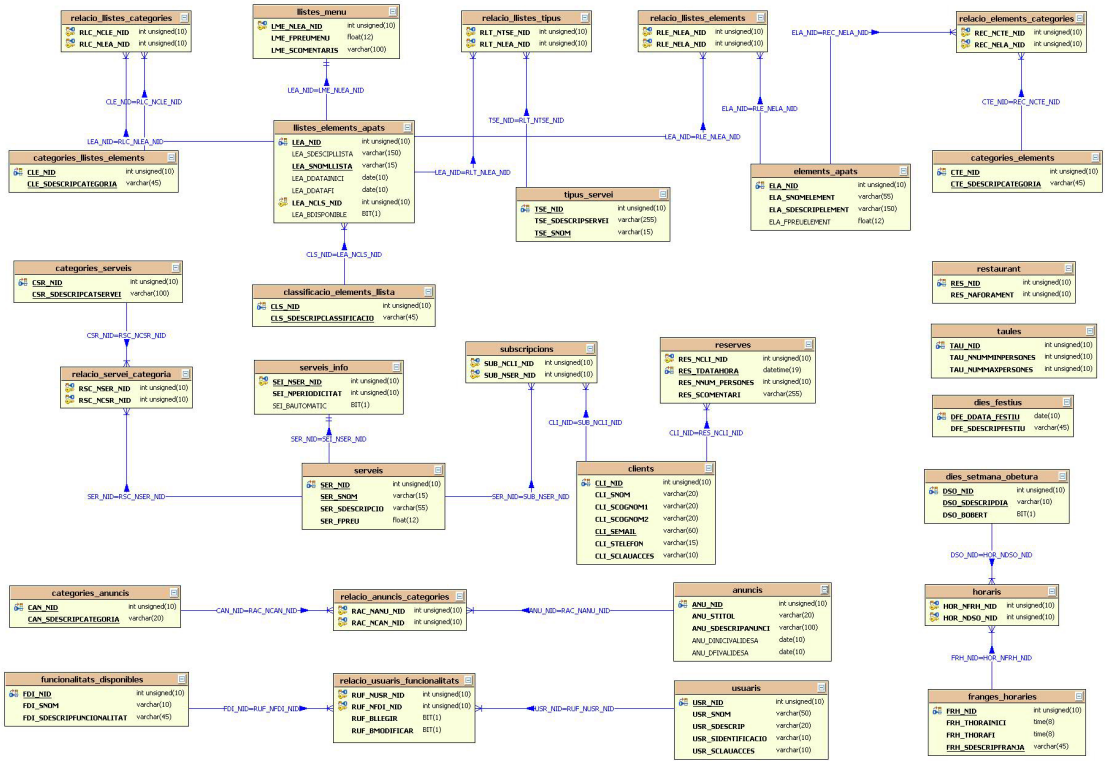
RELACIO_ANUNCIS_CATEGORIES(RAC_NANU_NID , RAC_NCAN_NID)

USUARIS(<u>USR_NID</u>, USR_SNOM, USR_SDESCRIP, USR_SIDENTIFICACIO, USR_SCLAUACCES)

FUNCIONALITATSDISPONIBLES(<u>FDI_NID</u>, FDI_SNOM, FDI_SDESCRIPFUNCIONALITAT)

RELACIO_USUARIS_FUNCIONALITATS (<u>RUF_NUSR_NID, RUF_NFDI_NID</u>, RUF_BLLEGIR, RUF_BMODIFICAR)

El següent diagrama mostra el model relacional de la base de dades de l'aplicació GatBlau



L'accés i el mapeig de la base de dades dins la nostra aplicació es farà utilitzant l'eina Hibernate. Aquesta eina fa l'ORM (object/relational mapping) entre els objectes de l'aplicació, que viuen en el món orientat a objectes, i les base de dades del món entitat relació dins del paradigma de java.

Hibernate proporciona la transacció necessària per passar de la representació de dades d'un model relacional a un model orientat a objectes i viceversa.

Un cop solucionat la problemàtica de la representació del model relacional caldrà dissenyar una serie de classes de control que gestionin les dades, és a dir els objectes que representen el model relacional. En l'apartat de revisió dels casos d'ús i en els diferents diagrames de seqüència simplificats, ja s'hi relacionen els diferents gestors.

Els gestors de dades han de permetre:

1. Recuperar llistats
2. Recuperar objectes segons un identificador
3. Guardar objectes, sigui actualitzant dades ja existents a la base de dades o afegint-ne de noves
4. Esborrar elements de la base de dades
5. Recuperar llistats segons criteris

4.3.1.Consideracions

D'acord amb la Ley de Protección de Datos Personales (Ley Orgánica 15/1999) de 13 de desembre de 1999 les dades relacionades amb els clients queden subjectes dins l'àmbit d'aquesta llei, és per això que en el moment de posada en marxa d'aquesta aplicació caldrà tenir en compte les implicacions d'aquesta llei.

4.4.Internacionalització: I18n

Un dels requisits del programa era que suportés la internacionalització (I18n). En aquest punt de la implementació s'ha preparat per tal que un cop fetes les traduccions, l'usuari vegi l'aplicació en el llenguatge que concordi amb la configuració local del seu ordinador.

Per tal que l'usuari decideixi en quin idioma vol veure l'aplicació, caldrà modificar, en la sessió de l'usuari, el paràmetre Locale, de manera que coincideixi amb l'opció escollida per l'usuari i no amb l'opció que marca el sistema. Aquesta funcionalitat ha quedat fora de l'abast del projecte, ja que només s'ha d'implementar per la zona pública de l'aplicació web.

Struts permet construir aplicacions multilingües tot preservant el model MVC i la mantenibilitat del codi. Struts permet separar la lògica del contingut a mostrar, és a dir dels missatges.

Quan un client, un navegador, fa una petició al servidor d'aplicacions, el servidor d'aplicacions reconeix quina és la configuració local de l'ordinador des d'on es fa la petició. Amb aquesta informació, sempre i quan disposi de la traducció adequada, mostrarà els missatges en un idioma o altre, si no es disposa de la traducció en l'idioma seleccionat el contingut es mostra en la llengua per defecte.

Aquestes eines permeten la internacionalització dels textos estàtics, el contingut dinàmic es gestionarà de forma diferent. La proposta, que queda fora de l'abast d'aquest projecte, per a la internacionalització dels continguts dinàmics, és utilitzar fitxers XML que contindran la traducció dels diferents textos dinàmics.

En el moment de la càrrega de les dades i el seu pas a les classes pont (classes en el paquet InterficiesDades) es produirà la traducció. Per a fer aquesta traducció un gestor d'idioma recuperarà els textos traduïts dels fitxers XML (segons l'identificador de taula, de camp i de registre), i els proporcionarà a les classes pont utilitzades a les vistes.

Dins el desenvolupament de la traducció dels textos dinàmics, caldrà implementar una eina per tal que els gestors puguin facilitar els textos traduïts i es puguin generar els corresponents fitxers XML.

Queda doncs, com a tasca futura el disseny d'aquesta funcionalitat, el disseny de l'estructura del fitxer XML amb els continguts dinàmics traduïts, i la implementació de les eines de recuperació i d'edició de les traduccions.

4.5.Disseny interfície

La interfície tindrà dos front end molt diferenciats entre si, un d'ells el que els clients accedeixin des de Internet, i que tindrà la fisonomia d'una pàgina web normal, això si incorporarà totes les avantatges descrites.

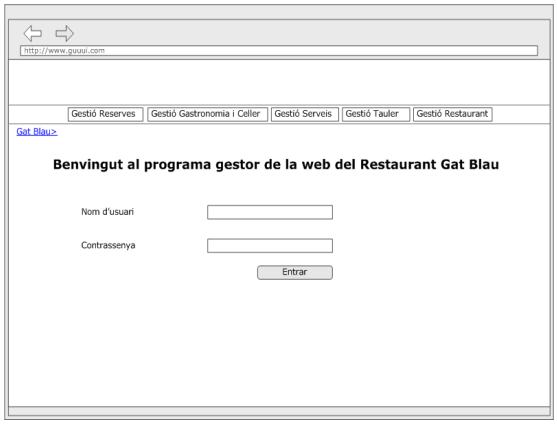
L'altre front end serà el que accedeixin els gestors del restaurant. Accediran a una pàgina web diferent, a on s'hauran d'identificar

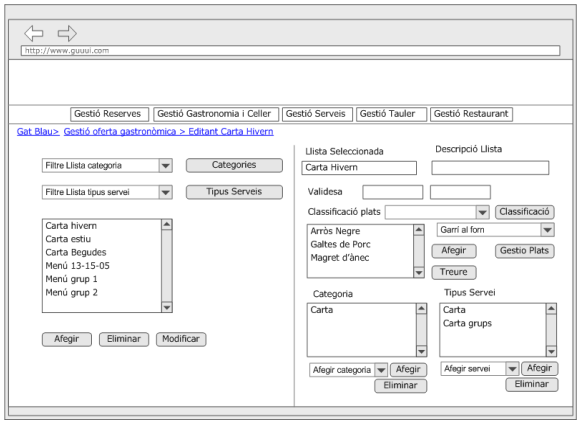
Per al disseny de la interfície es tindran en compte els següents criteris d'usabilitat:

1. Agrupament: organització en blocs de l'àrea visible, amb títol per cada bloc
2. Visibilitat i utilitat: Es posaran en un lloc molt visible les eines que s'utilitzen més freqüentment, mentre que amagarem en la mesura del possible les eines poc utilitzades.
3. Consistència: distribuir les informacions de forma similar en funcions similars
4. Economia: no sobrecarregar amb elements de disseny que no aportin informació
5. Color: Només s'utilitzarà en casos en que es vulguin emfasitzar algunes informacions. La pàgina ha de funcionar en blanc i negre, s'ha de tendir al MINIMALISME
6. Reduir el desordre: apliquem els 5 anteriors, de forma recursiva.

4.5.1.Interfície del gestor

Les dues figures que es mostren a continuació formen part del prototip implementat amb Visio i utilitzant l'eina de prototipatge "Web Prototyping" disponible en una llibreria d'eines i una plantilla per a documents d'aquesta aplicació de Microsoft. Aquestes eines es troben disponibles a "Visio - the interaction designer's nail gun" (http://www.guuui.com/issues/02_03_02.php)





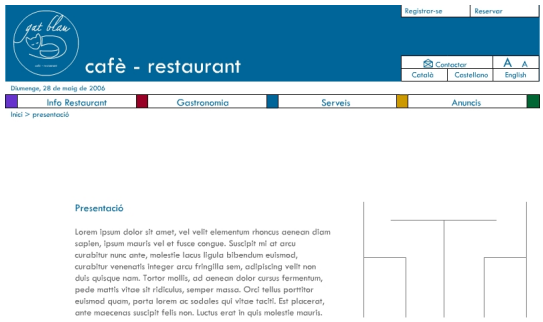
El disseny visual d'aquesta interfície no cal que sigui molt espectacular doncs és una interfície de treball per als gestors, s'ha reduït l'us del color a la mínima expressió, així com d'altres elements gràfics. També s'han deixat visibles les eines necessàries i s'ha disposat una una eina "on sóc" (tècnica breadcrumb), que permet que l'usuari sàpiga en tot moment en quin punt de la web es troba.

4.5.2.Interfície pública

Per a fer el prototipatge d'aquesta part de la interfície s'ha utilitzat l'eina Flash de Macromedia, doncs era més realista ensenyar un prototipus fet amb aquesta eina que no pas amb l'eina de Visio. I és que aquest prototip ja s'ha fet valorar a persones implicades en el projecte, com els gestors del restaurant i les persones que han creat la imatge corporativa d'aquest.

De cara a l'objectiu de fer conèixer el restaurant a través d'Internet, la qualitat d'aquesta interfície era molt important.

Les figures que es mostren a continuació són exemples de les pantalles que es poden veure en el prototipus "gatblauPrototipPublic.exe" que adjunt en la documentació entregada.



Com es veu a les figures s'ha prioritzat la definició d'àrees clares i l'agrupació d'elements, la mínima utilització del color i altres elements de disseny, s'han deixat visibles les eines més habituals. Es pot veure que totes les pàgines tenen una consistència pel que fa als elements comuns, i a la distribució dels elements en la pàgina.

Cal destacar que torna a aparèixer l'eina "on sóc" i que el menú i les eines d'idioma i tamany de lletra són sempre visibles.

5. Implementació

5.1.Eines i tècniques utilitzades

En l'apartat de disseny s'han exposat les tecnologies utilitzades, és necessari, en el moment de veure la implementació del projecte, revisar el perquè de la utilització de cadascuna d'aquestes eines

Java Platform Enterprise Edition (J2EE): L'ús d'aquesta tecnologia és bàsic per aquest projecte, doncs és la tecnologia dins la qual s'emmarca aquest projecte. J2EE és una plataforma de desenvolupament de programari existent dins la plataforma Java. El seu objectiu és desenvolupar i desplegar mòduls de software i aplicacions que siguin executades en un servidor d'aplicacions.

Jakarta Struts: Aquesta eina és un framework que proporciona l'estructura de l'arquitectura d'aplicacions basades en el model MVC, més concretament en la variant Model 2 del patró. Sota aquest model un sevlet gestiona la lògica de negoci, la presentació o vistes s'implmenten en pàgines jsp o similars. És un framework opensource i proporciona una serie d'eines que ajuden al desenvolupament.

Hibernate: És l'eina utilitzada per fer el la relació entre objectes i entitats de la persistència i que proporciona alhora un servei de consultes. L'eina permet desenvolupar classes de persistència dins el paradigma de la programació orientada a objectes (associació, herència, polimorfisme, composició i col·l3cions). Proporciona un llenguatge de consultes propi, alhora que suporta el llenguatge SQL, i la creació de consultes basades en criteris, entre d'altres.

Aquestes tres eines són possiblement les més importants i les claus en el desenvolupament d'aquesta aplicació. Ara bé, per al desenvolupament de la vista, s'han utilitzat altres eines que donen més potència a Struts.

5.1.1.Eines de recolzament

Struts-Tiles

D'acord amb criteris de reutilització i modularitat s'ha trobat una manera d'independitzar el contingut de la forma. En base a la idea del patró Composite View s'utilitzarà una eina d'Struts Tiles que ajuda a fer aquesta implementació de les vistes. Aquest tipus de dissenys faciliten la separació entre layout i contingut, de manera que si es vol fer algun canvi en el disseny no és necessari modificar pàgines de contingut.

Struts-Tiles és un framework que pertany a Struts. La seva finalitat és facilitar les eines necessaries per treballar amb plantilles per les vistes, de manera que tota l'aplicació pugui tenir un aspecte homogeni, permet alhora que els components de la vista siguin reutilitzables

Així doncs Tiles permet definir (ja sigui en un fitxer xml, directament en pàgines JSP o en les Action) una plantilla comú per a les diferents pàgines de l'aplicació. Una plantilla permet definir les regions comunes de totes les pàgines indicant en quin fitxer està definida cadascuna de les regions en concret (definim la capçalera, el menú, el contingut, etc). Aquesta plantilla es pot utilitzar com a base per a totes les pàgines de l'aplicació, però una aplicació pot suportar diferents plantilles.

Aquestes són les principals avantatges de Tiles, tot i que ofereixen una llista força àmplia de possibilitats.

Struts-Layout

D'altra banda, i tenint en compte que el coneixement dels llenguatges de marques de JSTL i html utilitzats per la construcció de vistes en jsp és força reduït, es planteja el fet d'utilitzar per a la implementació de les vistes Struts-layout.

Struts-Layout és una llibreria d'etiquetes(tagS) per a Struts. Aquesta llibreria proporciona una manera fàcil i ràpida per crear interfícies. Proporciona etiquetes molt potents per mostrar panells, camps per entrar dades, taules, treeviews, llistes, datagrids, popups i calendaris, entre d'altres.

Aquestes etiquetes poden proporcionar independència total respecte a l'HTML. Un dels objectius d'aquesta llibreria és reduir el temps de desenvolupament de vistes, i en aquest projecte, el factor decisiu per utilitzar aquesta llibreria és la seva simplicitat, potència i independència de l'HTML.

5.1.2.Eines investigades per a la implementació de funcionalitats futures

Struts-Menu

Com ja s'ha discutit en l'apartat de disseny d'interfícies d'usuari, cal implementar un tipus de menú horitzontal amb desplegament vertical. Malgrat ha quedat fora de l'abast del projecte s'han estudiat quines serien les eines necessàries per a la seva implementació. Hi ha moltes solucions basades en javascript i css, però s'ha considerat una eina com Struts-Menu que facilita la creació dels menús de l'aplicació. Malgrat tot, en la part desenvolupada de l'aplicació, el menú és poc profund i de moment no s'ha plantejat la seva implementació utilitzant aquestes eines.

BIRT (Business Intelligence and Reporting Tools)

Finalment cal considerar la implementació de la funcionalitat que recull la generació de documents. Aquests documents hauran de ser en format html, per mostrar menús i cartes d'una manera amigable a la web, o per generar pdf i poder-los imprimir per fer les cartes del restaurant.

L'eina escollida per fer aquestes tasques és BIRT (Business Intelligence and Reporting Tools), aquesta eina proporciona un motor que es pot integrar en qualsevol tipus d'aplicació (incloent aplicacions web) i que permet generar informes que poden ser exportats en pdf o en html.

La implementació d'aquesta funcionalitat queda, de moment, fora de l'abast del projecte, i és per aquest motiu que no s'ha aprofundit més en el seu us.

5.2.Estructura de l'aplicació

Les aplicació web J2EE tenen una estructura determinada característica. Per tant, l'estructura del projecte estarà d'acord amb el patró J2EE, que permet el seu deployment en un arxiu web (WAR). Així dins del nostre projecte trobarem dues carpetes principals "scr" on s'organitzen totes les classes java i "WebRoot" que és la base de l'estructura del l'arxiu WAR.

Dins del WebRoot, el directori Web-inf es troben els continguts no públics de l'aplicació, llibreries, classes compilades de java, etc, i també el web.xml, que és el fitxer amb la descripció del desplegament de l'aplicació on es descriuen els tags, els servlets i les llibreries que s'utilitzaran. Dins del webroot també trobarem els elements públics de l'aplicació, com les pàgines jsp, o les imatges.

L'estructura i continguts de la carpeta SRC són ja més lliures, aquest directori contindrà el conjunt de classes java i recursos de l'aplicació. Generalment en el desplegament les classes compilades es desen en la carpeta privada web-inf/classes seguint l'estructura de directoris que hi ha a "src"

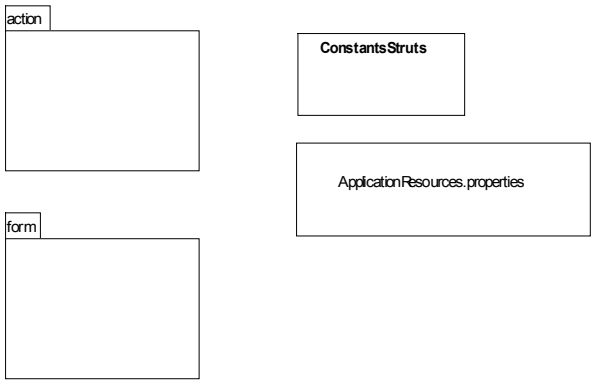
5.2.1.Estructura paquet cat.lllunas

5.2.1.1.Interfície usuari: Utilització Struts per MVC

El paquet cat.lllunas.struts té l'estructura que es mostra a la figura. D'acord amb l'esquema, dins d'aquest paquet hi ha dos paquets més Action i Form, així com una classe de constants i els fitxers de propietats.

En el subpaquet Action s'agrupen totes les classes de tipus "Action", és a dir les classes que encapsulen el model de l'aplicació.

Seguint amb les classes que s'utilitzen en Struts per implementar el MVC, el subpaquet Forms recull totes aquelles classes Forms, és a dir les que suporten els formularis de les vistes, aquestes classes mantenen l'estat de la sessió de l'aplicació web.



El controlador

El controlador del model MVC s'implementa en forma d'un servlet que fa la correspondència entre els events (generalment en forma d'un HTTP post) i classes corresponents. El controlador utilitza els fitxers de configuració (struts-config.xml) per establir aquesta relació.

En aquest projecte s'ha utilitzat com a servlet, la classe controlador que Struts estableix per defecte i que és ActionServlet.

Abans de veure la implementació particular de les classes Action, seguidament es discuteix breument com es configura el controlador de l'aplicació. Aquesta configuració es controla en el fitxer struts-config.xml.

De cara a la configuració de l'aplicació per utilitzar conjuntament amb Struts, el framework Struts-tiles i la llibreria struts-layout, és necessari canviar el controlador que es fa servir. En aquesta aplicació el controlador utilitzat és el que proporciona la llibreria Struts-Layout que al seu temps hereta del controlador proporcionat per Struts-Tiles.

Per tal que aquesta configuració tingui efecte, s'han afegit les següents línies al fitxer struts-config.xml

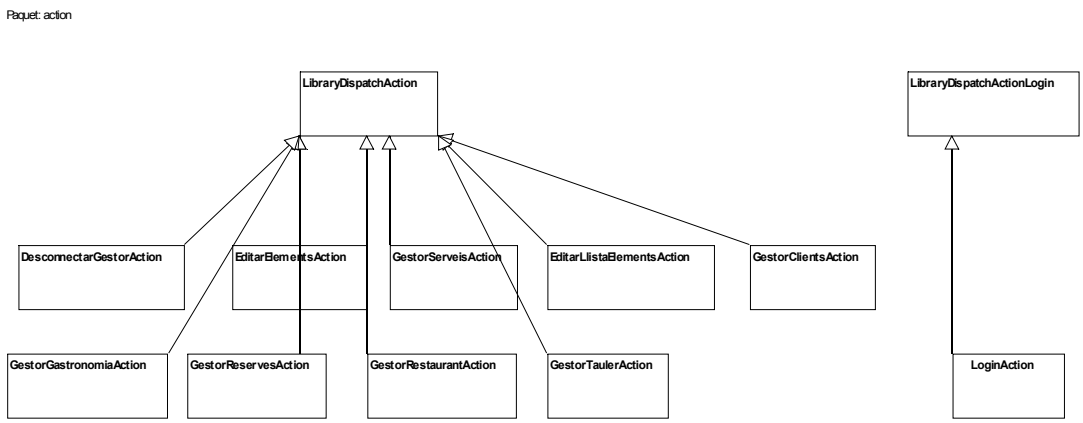
```
<!-- ========== Controller Configuration ========================== -->

<!-- extends org.apache.struts.tiles.TilesRequestProcessor: necessari
segons "Tiles Configuration"-->
<controller processorClass =
"fr.improve.struts.taglib.layout.workflow.LayoutRequestProcessor"/>
```

aquesta configuració, juntament amb la declaració del plugin de Tiles permet utilitzar la configuració Struts-Tiles-Layout

Encapsulat del model, les classes Action

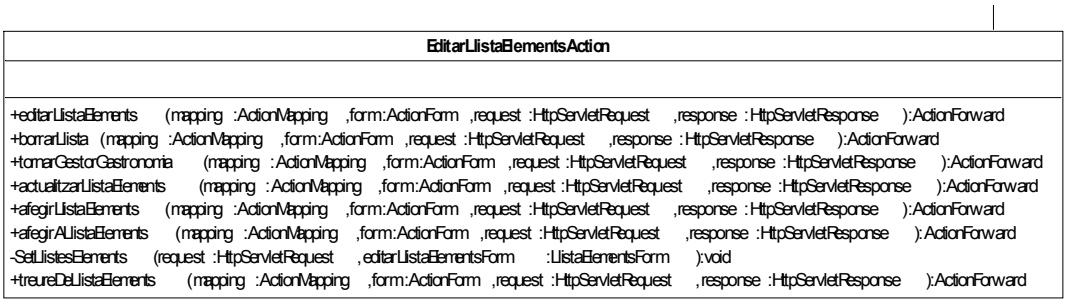
Tornant a la implementació de les diferents classes java del paquet Action, és interessant veure la jerarquia de les classes Action. Totes les classes que realitzaran les accions de Struts (Action) hereten de cat.lllunasm.struts.action.LibraryDispatchAction, o la seva homòloga cat.lllunasm.struts.action.LibraryDispatchActionLogin.



En aquest diagrama veiem quines són les relacions que s'estableixen entre les classes que implementen les diferents accions de l'aplicació i les superclasses que s'han escollit per facilitar la implementació.

Al seu temps, malgrat que no queda reflectit en el diagrama, les classes LibraryDispatchAction i LibraryDispatchActionLogin, hereten de DispatchAction. El fet que estiguin dins d'aquesta jerarquia no és casualitat. La classe DispatchAction ens permet agrupar en una sola classe un conjunt de funcions relacionades, funcions que representen accions que es poden trobar en una mateixa pàgina. D'aqusta manera es poden agrupar conceptualment sense haver d'implementar cadascuna en una Action independent.

En el moment de rebre una petició, LibraryDispatchAction decideix quin mètode ha d'executar en funció del paràmetre que rebi en la petició. En la figura següent veiem una de les DispatchAction de l'aplicació i els seus diferents mètodes.



Dins de la classe EditarLlistaElements s'inclouen els mètodes per esborrar la llista, afegir elements a la llista o treure'n, actualitzar els canvis de la llista etc. D'aquesta manera queden agrupats de forma lògica.

Comprovació de la sessió d'usuari

Tal com es veu en el diagrama de relacions entre classes Action i superclasses, LoginAction hereta de LibraryDispatchActionLogin, mentre que la resta d'accions ho fan de LibraryDispatchAction.

LibraryDispatchActionLogin no comprova que hi hàgi cap usuari validat en la sessió, mentre que LibraryDispatchAction, només fa la crida dels mètodes acció si hi ha un usuari validat, en cas contrari redirigeix el flux de l'aplicació cap al punt d'entrada. Aquesta dualitat ens permet assegurar que no es podrà accedir a l'aplicació sense que l'usuari s'hagi validat primer.

Al executar-se el mètode "execute" de LibraryDispatchAction, es comprova que la sessió té un atribut USERKEY, si és així continua normalment el procés i executa l'acció indicada en el paràmetre de la petició, altrament redirecciona el flux cap a la pantalla de login.

5.2.1.2.Internacionalització

Reprenent el que s'ha comentat en apartats anteriors, Struts proporciona un conjunt d'eines que ajuden a desenvolupar aplicacions I18n sense necessitat de fer múltiples pàgines traduïdes: traduir una aplicació Struts significa traduir un fitxer de propietats.

Amb Struts només cal utilitzar les etiquetes de la llibreria estàndard que permeten definir claus de cerca de textos, és a dir etiquetes que permetin el paràmetre "key", aquest paràmetre indica a Struts que ha de cercar el text del missatge.

Tags com bean:message, bean:write, layout:message, html:cancel, entre molts d'altres permeten parametritzar el paràmetre "key"

Finalment haurem de declarar els textos en els fitxers ApplicationResources.properties. Aquest fitxer tindrà la traducció de l'aplicació per defecte, quan es vulgui traduir, haurem de proporcionar la traducció en un fitxer ApplicationResources_xx_XX.properties, on xx_XX és l'estàndard per l'especificació local, per exemple en_US.

La especificació del programari indicava que l'usuari podia escollir l'idioma, per tal de que això sigui així, quan s'implementi aquesta funcionalitat, únicament serà necessari modificar la informació que aporta l'atribut *Session Locale.* Aquest atribut és per defecte la configuració d'idioma del sistema des d'on s'executa el navegador que ha establert la sessió.

Els fitxers de textos ApplicationResources_xx_XX.properties, es troben dins del paquet cat.lllunas.struts.

Cal considerar que la informació per a la internacionalització que aporten els fitxers de propietats és insuficient, i és que aquests fitxers només s'utilitzen per textos estàtics de l'aplicació.

Per tal de tenir l'aplicació totalment internacionalitzable caldria traduir també els textos dinàmics, és a dir els que es guarden a la base de dades. Aquest desenvolupament ha quedat fora de l'abast del projecte, i ja ha estat discutit en apartats anteriors.

5.2.1.3.Les vistes i l'aplicació web

D'acord amb el que s'ha comentat en apartats anteriors, les vistes i la configuració de l'aplicació web resideixen al directori WebRoot

En aquest cas no cal entrar en excessiu detall dels continguts, només explicar que en el directori ·config" hi ha tots els elements necessaris a nivell gràfic (imatges, css, js, entre altres). El directori "lib" conté els jars de les llibreries a utilitzar. A WEB-INF trobem els fitxers de definició de l'aplicació: el fitxer de definició de Struts, el de Tiles, el propi de l'aplicació web i el de les diferents llibreries d'etiquetes.

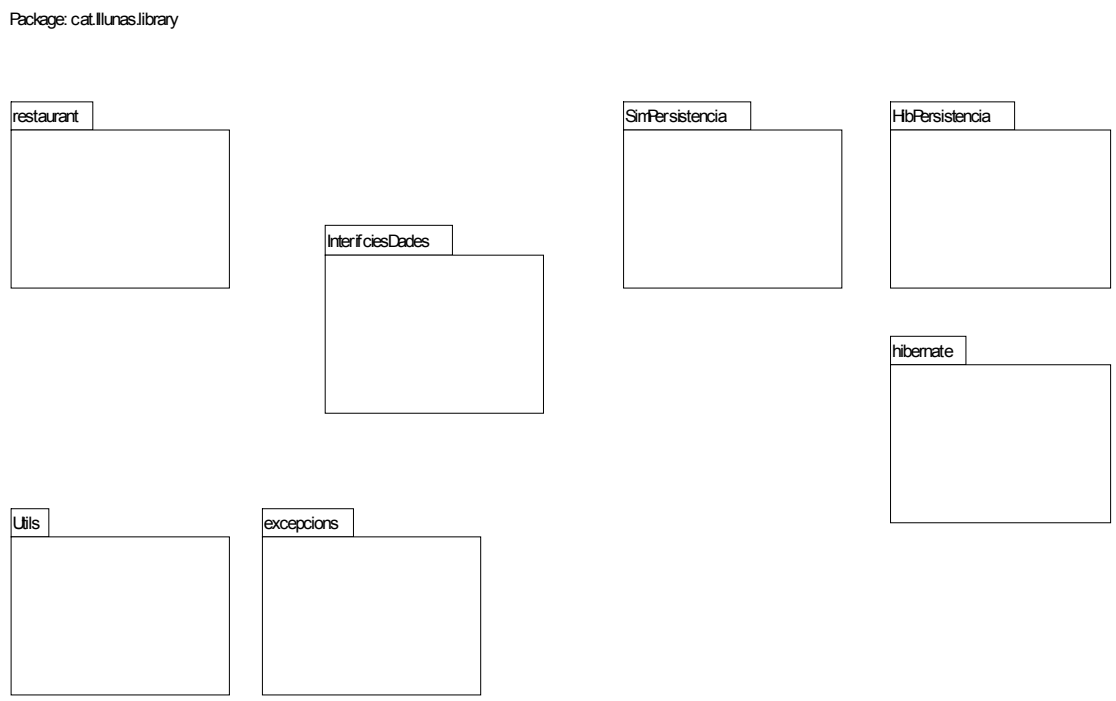
Una estructura d'aplicació basada en Tiles, pot definir les seves pàgines de diverses maneres, generalment utilitzant el fitxer xml de definició o directament jsp. L'opció basada en jsp ha estat l'escollida, en aquest cas per una qüestió de facilitat.

Així doncs en el l'arrel del directori WebRoot trobem els fitxers jsp que defineixen l'estructura de cada una de les diferents pàgines. El contingut de cada pàgina així com la definició de les diferents regions estàtiques de les vistes es troba dins del directori templates en fitxers jsp.

En aquestes vistes falta acabar d'unificar el criteri de disseny, i sobretot adaptar els continguts mostrats a una resolució de pantalla adequada. En el cas de l'aplicació pública la restricció serà la de pàgines visibles en una resolució de 800x600, mentre que la part de gestió, on es necessita més espai, es podrà fer a una resolució superior (1024x768) ja que es coneixen els equips on s'executarà.

5.2.1.4.Lògica Negoci: Gestors i interfícies de dades

La lògica de negoci es troba en el paquet cat.lllunas.library, i el paquet té la següent estructura:



La lògica de negoci dóna servei al model del MVC, en concret les crides a aquest a aquest paquet queden encapsulades dins les classes Action. Com ja s'ha discutit el model implementat en l'estructura MVC no és res més que la crida a una serie de gestors que queden fora del paquet MVC-Struts.

Aquesta estructura permet separar la interfície d'usuari de la lògica de negoci i les dades, tal com es pretenia en el disseny de l'arquitectura de l'aplicació.

Veurem a continuació quina és la funcionalitat i els continguts de cada paquet inclòs en cat.lllunas.library

cat.lllunas.library.restaurant

Dins del paquet restaurant trobem tots els gestors implementats (GestorRestaurant, GestorHoraris, GestorUsuaris GestorReserves, GestorClients, GestorGastronomia). Aquests gestors s'utilitzen d'interfície entre l'estructura MVC i la lògica de negoci, en concret, les crides a aquests gestors queden encapsulades dins les classe Action (el model de Struts). En cas que es decidís canviar l'interfície web en una interfície tipus swing tota la lògica de negoci es podria re utilitzar.

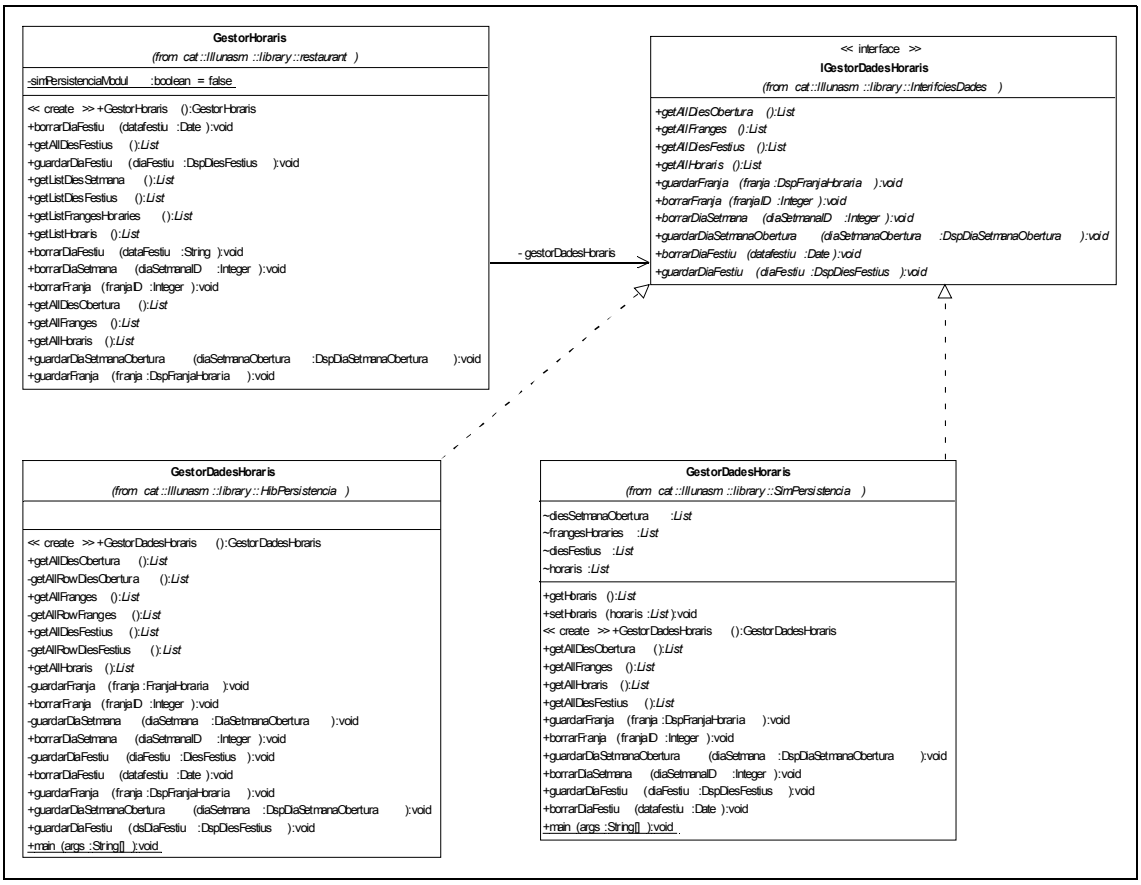
Cada gestor està associat a un gestor de dades, aquests gestors de dades s'encarreguen d'accedir a les dades de l'aplicació. Els gestors implementen mètodes delegats dels gestors de dades que tenen associats.

Aquesta estructura permet tenir implementacions diferents d'un gestor de dades, podent proporcionar gestors de dades simulades o gestors de dades que accedeixen a la base de dades.

En cas que l'aplicació requereixi algun tipus d'operació no vinculat directament a les dades, serà a aquest nivell de gestors els encarregats d'executar les operacions.

En aquest punt de la implementació no se n'ha trobat cap cas, tot hi així l'enviament de correus electrònics de confirmació de reserves seria un exemple d'aquestes operacions que es realitzen a nivell de gestor, en aquest exemple seria el GestorReserves qui implementaria aquesta funció.

La figura mostra l'estructura per al gestor d'horaris (GestorHoraris) que implementa els mètodes delegats de la interfície del gestor de dades d'horaris (IGestorDadesHoraris). Aquesta interfície és implementada per dos classes diferents, una dins del paquet de simulació de persistència i l'altre dins del paquet de persistència en base a l'hibernate.



La utilització d'una classe o altre per part del GestorDadesHoraris es decideix en temps d'execució quan cadascun dels gestors instancien les interfícies dels gestors de dades que utilitzaran, aquesta operació es fa utilitzant la següent tècnica:

En la classe Gestor, es declara un atribut com segueix:

```
IGestorDadesHoraris gestorDadesHoraris;
```

en el constructor s'instancia aquest atribut de la següent manera:

```
gestorDadesHoraris = (IgestorDadesHoraris)(Class.forName(classname).newInstance());
```

on classname és la ruta completa de la classe GestorDadesElQueSigui. En aquesta ruta s'indica si es tracta de la implementació en el paquet de simulació o en el paquet d'accés a la base de dades.

És probable que pel tipus i l'envergadura d'aquesta aplicació, no hauria estat estrictament necessari fer aquesta estructura dual, però per aquest projecte ha estat imprescindible.

Cap de les dues tecnologies utilitzades per implementar la interfície d'usuari i per la persistència eren conegudes. L'ús d'aquesta estructura dual ha permès separar la implementació d'una i altre part, facilitant així la comprensió i desenvolupament dels diferents elements.

cat.lllunas.library.InterficiesDades

El paquet InterficiesDades conté un conjunt de classes, molt similars a les que es van definir com a classes entitat en el disseny, i que actuen com a pont entre les classes que s'utilitzen a nivell d'hibernate per mapejar la base de dades i les vistes de l'aplicació.

D'aquesta manera el conjunt de classes d'aquest paquet adapta les classes del paquet hibernate per tal que es puguin utilitzar en les vistes, d'acord amb els formats de les dades que les vistes permeten. Són també el conjunt de classes que s'utilitzen com a entitats en els mòduls de simulació de la persistència.

5.2.1.5.Persistència: Model Simulat i hibernate

Tal com s'ha discutit en l'apartat anterior, existeix una dualitat en la implementació de la font de dades de l'aplicació. Una branca que simula la persistència i una segona que gestiona les dades de l'aplicació i la persistència d'aquestes a través d'una base de dades.

La gestió de la persistència es fa utilitzant hibernate, i el mapeig ORM de les entitats s'ha fet utilitzant les eines que ofereix MyEclipse i que permeten generar classes i fitxers de mapeig automàticament.

La implementació del mapeig entre els objectes i el model entitat-relació no és trivial, la utilització d'eines de mapeig automàtic proporciona una solució ràpida, però que ha de ser revisada.

També cal tenir en compte la complexitat de la base de dades inicial, els ORM de claus compostes són complicats, així com les relacions many to many; aquestes relacions han de ser persistides explícitament, ja que no es guarden al persistir els objectes implicats en la relació. Per tant, és necessari guardar les relacions entre objectes després d'haver guardat els objectes en si.

A fi de fer menys feixuc el mapeig i la utilització dels objectes de la persistència s'ha eliminat l'herència de les llistes d'elements i les llistes de menús, doncs al ser només dos camps en la subentitat és més econòmic i senzill utilitzar una sola taula.

Els paquets relacionats amb la persistència també es troben dins de cat.lllunas.library. Aquests són tres, cadascun amb un objectiu molt clar tal com es detalla a continuació.

Cat.lllunas.SimPersistencia: Paquet amb un conjunt de gestors de dades que s'encarreguen de gestionar dades de simulació, és a dir, permeten gestionar dades sense accedir a cap base de dades. És un mòdul pensat per fer proves amb les interfícies de dades.

Cat.lllunas.HibPersistencia: Paquet que conté un conjunt de gestors de dades que actuen sobre les dades de la base de dades. Són els encarregats de recuperar la informació persistida, cercar-la, actualitzar-la o eliminar-la. Treballen amb el conjunt de classes que hibernate ha generat per tal de mapejar el model entitat-relacio en un model d'objectes.

Cat..lllunas.Hibernate: Aquest és el paquet resultant de l'ORM, en ell s'agrupen totes les classes que representen el model relacional de la base de dades, així com els fitxers.hbm.xml que contenen les relacions entre les classes i la base de dades, és a dir la definició del mapeig objecte-entitat.

5.2.1.6.Control d'errors i excepcions

A aquest punt de desenvolupament les excepcions es tracten a nivell local, i en cas d'error o de no finalització d'una instrucció, l'usuari no rep cap avís. És per tant necessari solucionar aquest punt a fi de millorar la usabilitat del programa.

La gestió d'errors locals no té per que ser incorrecta, però en aquest cas no s'adiu gents amb els criteris d'usabilitat. Un error al guardar a la base de dades s'ha de propagar per tal que l'usuari sigui conscient i pugui efectuar accions correctives, i com aquests d'altres exemples que caldria reportar a l'usuari.

També cal notar que no es fa cap validació a nivell de formularis, per tant seria interessant aprofitar aquesta potència i detectar possibles errors en un primer nivell. D'aquesta manera els errors no es propagarien en la jerarquia de l'aplicació obligant a que es detectessin en el punt crític i fossin retornats de nou cap a l'usuari a través de tota la jerarquia de l'aplicació.

Finalment, i de cara a la revisió de la interfície gràfica, serà molt útil utilitzar les eines que dels camps de text i de les text àrees que permeten posar màscares d'entrada per camps numèrics o limitar la longitud del text introduït, entre d'altres.

5.3.Proves

En el pla de treball s'especifica una tasca que implica el disseny de proves. No s'ha generat cap document amb el resultat d'aquest procés, però tot i així el conjunt de probes realitzat a seguit un ordre lògic. També s'han desenvolupat mòduls, com la simulació de les dades, que han permès provar etapes d'integració de forma poc dramàtica. A continuació es descriuen els tipus de proves realitzats.

1. A nivell de mòduls de gestió i gestió de dades: s'han fet proves a nivell d'objectes, utilitzant petites aplicacions per validar cadascuna de les funcionalitats de les diferents classes gestores. Aquestes proves han afectat bàsicament als gestors, als gestors de dades i a les classes pont del paquet d'interfícies de dades.

2. A nivell de mòduls d'interfície, efectuant proves sobre el funcionament de les diferents eines a utilitzar (struts-layout, struts-tiles....), i de les diferents vistes implementades utilitzant mòduls de gestió amb dades simulades

3. Les proves d'integració, on s'ha provat el conjunt de la persistència i de la interfície d'usuaris alhora, és a dir habilitant les dades reals de l'aplicació.

4. S'han fet proves de caixa negra (sense accés al codi) i proves en mode debug de les etapes de proves per objectes, proves d'interfície i proves d'integració

No hi ha hagut proves amb els usuaris directes, ja que les funcionalitats implementades eren poques i no valia la pena crear un estat de desil·lusió. Si que es plantegen per més endavant un cop hi hagi una interfície més depurada, i les funcionalitats pel públic general implementades.

Finalment, a nivell de proves i test de la interfície faltarà passar els test a la interfície d'usuari, tests que mesuren, per exemple: el compliment del nivell AA d'usabilitat, el funcionament de l'aplicació en diferents navegadors per veure la compatibilitat a diferents plataformes, eines que fan la revisió dels ccs i que comproven la navegació en mode text.

6. Conclusions

Finalment, després de revisar el plantejament inicial del projecte, el seu disseny i la seva implementació, resta només fer la valoració del conjunt. Aquesta valoració es farà des del diferents punts de vista, així doncs es revisarà quina ha sigut l'evolució respecte la temporalització, quina ha estat l'abast del projecte, així com l'avaluació dels riscos. Per acabar es revisaran els objectius i la línia de treball a seguir per tal que aquest projecte acabi publicant-se a Internet.

Partint de la temporització inicial, cal considerar que fins a l'etapa d'anàlisi es van seguir els terminis fixats, després durant el disseny va haver un petit decalat per motius personals. Ara bé els terminis disposats per a la implementació han resultat ser del tot optimistes, i és que, malgrat haver assistit al curs de Java que em va refrescar la memòria sobre el llenguatge, el temps que he hagut d'esmerçar en qualsevol de les tasques d'implementació descrites ha estat molt superior al inicialment previst.

És aquest el motiu que la temporització no s'hagi seguit al peu de la lletra, i que l'etapa d'implementació s'hagi allargat més del previst. D'altra banda considero que sabent que l'embargadura del projecte era important, essent conscient dels coneixements previs i considerant com a risc el curt període de temps en que s'havia de desenvolupar l'aplicació, s'hauria d'haver distribuït la temporalització d'una altre manera.

Hagués estat més adient reduir els períodes destinats a la recollida de requeriments, a l'anàlisi i al disseny, avançant les dates de finalització de cada etapa respecte a les dates proposades en el pla d'estudi, d'aquesta manera s'hagués pogut incrementant el període destinat a tasques d'implementació.

Malgrat que durant les primeres etapes vaig dur a terme diverses accions a fi de familiaritzar-me amb la tecnologia i les noves eines, no vaig avançar prou en aquest sentit. Per tant el període destinat a la implementació m'ha queda curt. En canvi la realització de les tasques d'anàlisi i disseny, a les quals hi estava més avesada per motius laborals, es podria haver fet en un temps menor.

Pel que fa a l'abast del projecte queda palès que totes les funcionalitats inicialment descrites no s'han implementat, de totes maneres aquesta era una possibilitat, sinó una realitat, contemplada des del primer moment.

Creiem que haver començat per la part de gestió de l'aplicació ha estat una decisió encertada, i és que a nivell de complexitat és superior a la part d'usuari, cal fer modificacions, eliminació i afegit de dades, mentre que en la part pública es fan majoritàriament cerques.

D'altra banda la dificultat de la part pública de l'aplicació recau més en el desenvolupament de les vistes, ja que té una component de disseny visual superior. Per això es va considerar més adequat desenvolupar parts que em permetessin la familiarització amb les eines de disseny visual (tiles i layout) abans d'atacar la part visible pel públic en general.

En la definició de qualsevol projecte, cal tenir en compte els riscos associats a aquest. Si ho recordem, en el pla de treball d'aquest projecte se'n van identificar tres: l'ampli llistat de requeriments, el temps i el desconeixement de la tecnologia.

Pel que fa al llistat de requeriments ja hem vist que era un risc real, amb els gestors del restaurant es van definir les funcionalitats de serveis i tauler com a secundaries, així com la generació d'informes.

En vistes del nivell d'implementació al que s'ha arribat entenem que s'hauria d'haver ajustat més la llista de requeriments i de les seves prioritats. Aquest risc anava molt lligat amb la temporització, i és que si el període total per fer el projecte hagués estat de 8 mesos, possiblement cap dels dos elements de risc hagués estat considerable.

D'altra banda el desconeixement de la plataforma J2EE, també era un risc real, i és que quan es desconeix una tecnologia, cal fer un determinat aprenentatge de la mateixa a fi de poder-la utilitzar en un desenvolupament. La corba d'aprenentatge marca la velocitat en que es pot desenvolupar. En el meu cas la pendent d'aquesta corba va ser molt petit en les primeres etapes d'implementació, mentre que es va incrementar notablement a la part final del projecte.

Durant el mes d'abril vaig assistir a un curs de 40h que va servir per recordar conceptes i llibreries una mica rovellades de Java. També es va assistir a un seminari de 6h d'utilització d'Eclipse (l'entorn de desenvolupament) que va ser molt profitós.

Tot i aquestes accions i el fet de revisar i provar diversos tutorials de les diferents tecnologies utilitzades, el moment de posar en marxa el desenvolupament de GatBlau va ser una mica dramàtic. Una de les dificultats que vaig trobar va ser amb el mapeig ORM, i és que el model relacional és força extens i té relacions complexes, i per tant vaig haver de lluitar per entendre i ajustar el mapeig automàtic fet per l'assistent de MyEclipse.

La falta de documentació d'eines com Struts-layout, bàsiques en aquest desenvolupament, tampoc va ajudar massa. I és que sovint he descobert el funcionament de les eines a base de proves. Aquesta eina no va ser una excepció en la llista de coses a aprendre, i de nou la corba d'aprenentatge va tenir una pendent molt suau al principi fins que, arribats a un cert punt, la forma de la corba va canviar substancialmen.

Tanmateix, malgrat que el risc de fer un projecte desconeixent la tecnologia en la qual s'havia d'implementar, considerem un èxit el punt assolit. Ara mateix, els coneixements assolits són forces, tant a nivell d'eines, com de la mateixa plataforma J2EE, com de les llibreries i els frameworks utilitzats, de les tècniques i dels patrons utilitzats. Això si, també he descobert la immensitat d'aquest món del programari lliure, Java i J2EE.

A nivell d'objectius es van marcar dues grans línies: introduir-nos en les tecnologies J2EE i crear una eina útil pel Gat Blau. Entenc que tots dos s'han complert en gran part. Pel que fa el primer, tant les tecnologies utilitzades, la proposta de noves eines i el nivell d'implementació assolit, mostren que, fins a cert punt, ens hem introduït en aquest món. I no només introduït, sinó que hem estat capaços de crear una aplicació pensada per funcionar en un entorn real.

Pel que fa a l'objectiu de desenvolupar una eina pel restaurant Gat Blau, entenem que encara que no és una eina finalitzada, va per bon camí. La interfície i el plantejament de

l'aplicació ja s'han mostrat als gestors del restaurant, els quals han mostrat la seva satisfacció.

Ara queda una part important per acabar d'oferir les funcionalitats de gestió, i una altre part, també important, de desenvolupament de la interfície pública de la web. Totes dues línies plantegen grans reptes, ara sobretot pel que fa a la construcció de vistes.

Finalment direm que aquest era un projecte ambiciós pel que fa a requeriments i en vista del temps disponible i que en l'aprenentatge de la tecnologia s'ha fet un pas important. Per tant, malgrat no assolir el cent per cent dels objectius plantejats, aquest projecte es pot considerar un petit èxit i més si recordem quin era el punt de sortida..

7. Manuals

7.1.Guia Instal·lació

7.1.1.Requisits

Per a poder executar aquesta aplicació com a servidor cal tenir instal·lats el servidor d'aplicacions Tomcat 5.5 i el sistema gestor de bases de dades MySql5.0. Entenem que la instal·lació d'aquestes aplicacions està fora de l'abast d'aquest manual, en qualsevol cas la configuració d'aquestes eines és l'estàndard en el cas de l'aplicació GatBlau.

7.1.2.Preparar la base de dades

La distribució actual de GatBlau treballa ja amb la base de dades, és per tant imprescindible executar l'sql que ens generarà l'estructura de la base de dades necessària (gatblaubbdd.sql) i l'sql que ens carregarà un joc de dades de proves (gatblaudades.sql).

Aquest procediment es pot fer directament des de la consola de mysql, però per comoditat es pot fer amb una eina del tipus "MySql Query Browser"

També serà necessari crear i donar privilegis sobre la base de dades GatBlau a un usuari amb

nom d'usuari: GatBlauAdmin

clau d'accés: 5502leitz

Aquesta configuració del nom d'usuari coincideix amb la que està definida en el fitxer d'hibernate de configuració d'accés a la base de dades. Per fer aquesta gestió, malgrat es pot des de la consola, és més amigable utilitzar una aplicació tipus "MySQL Administrator"

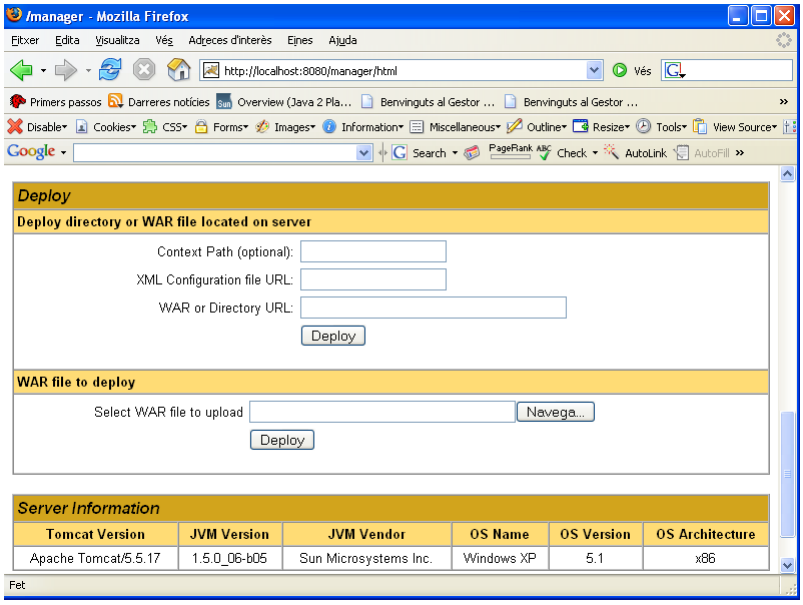
7.1.3.Carregar l'aplicació en el servidor d'aplicacions

La versió 5.5 del Tomcat que s'ha utilitzat hi ha una eina molt útil que permet fer el desplegament d'una aplicació web des d'un arxiu WAR d'una forma molt senzilla.

Un cop arrencat el Tomcat, des d'un navegador accedirem a http://localhost:8080/manager/html. En la part inferior d'aquesta pàgina trobem les següents eines per fer el deploy:

A la part inferior d'aquest apartat hi ha l'eina que ens facilita el desplegament d'un fitxer war.

Ens cal doncs, clicar a "Navegar..." per cercar i seleccionar el fitxer war de l'aplicació Gat Blau. Un cop seleccionat el fitxer resta clicar al botó "Deploy" per finalitzar la càrrega de l'aplicació.

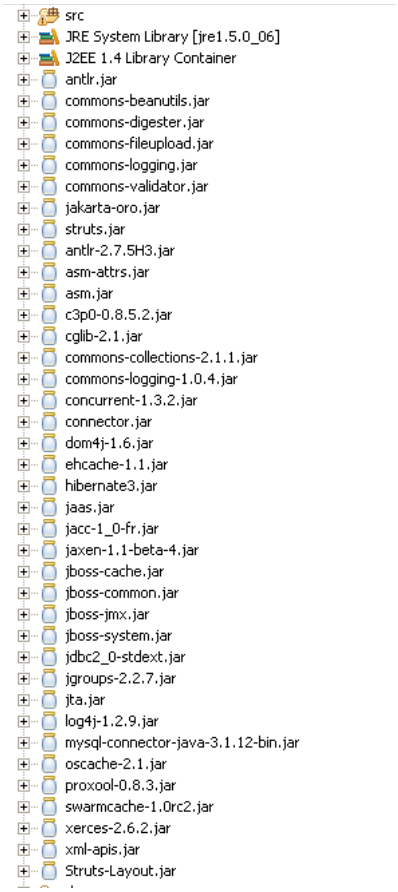


A partir d'aquest moment l'aplicació Gat Blau estarà disponible des del link: http://localhost:8080/GatBlau/

7.1.4.Instal·lació de les llibreries

Si la distribució utilitzada no inclou les llibreries jar necessàries, caldrà incorporar-les al directori lib en el directori de desplegament. Aquestes llibreries són necessàries per al funcionament de l'aplicació. La llista de llibreries a incloure és la que mostra la figura.

Degut a que la llibreria de Struts-Library no és estàndard, aquesta s'inclou en totes les distribucions.



7.2.Manual usuari

La distribució entregada en aquesta versió té la funcionalitat limitada, és a dir només es pot gestionar part de les dades del restaurant referents a l'horari, les llistes de clients i les reserves. També es poden gestionar les llistes d'àpats i els elements dels àpats.

Totes aquestes funcionalitats no estan disponibles al 100% ja que per exemple els filtres de les dades que han d'ajudar a les cerques no estan implementats.

Un cop instal·lada l'aplicació i arrencat el servidor d'aplicacions, en el nostre cas accedim a l'adreça http://localhost:8080/GatBlau/. Aquesta adreça pot variar en funció de com estigui configurat el servidor apache.

7.2.1.Entrar a l'aplicació

Un cop s'hagi accedit accedit al link de l'aplicació, caldrà que ens identifiquem. En aquesta versió només un usuari gestor es pot identificar, el nom d'usuari i la clau d'accés a utilitzar són:

nom usuari: GatBlauAdmin

contrasenya: 20maig2005

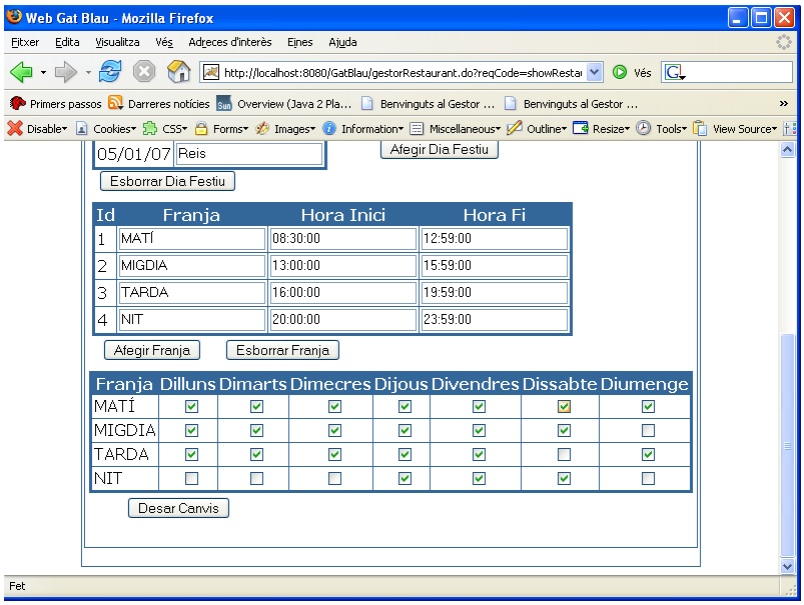
Si la identificació és correcta la mateixa aplicació ens conduirà a la pàgina de gestió del restaurant, on es pot configurar l'horari.

El menú de l'aplicació està disposat a la part superior de cada pàgina, i ens permet accedir a qualsevol funcionalitat des de qualsevol pàgina.

Els apartats Gastronomia, Restaurant, Reserves i Clients estan parcialment implementats. La resta d'opcions estan implementades però només accedeixen a una pàgina sense cap tipus de funcionalitat.

7.2.2.Gestor del Restaurant

Les dues figures següents mostren la pantalla per a la gestió d'horaris. La interfície és força explicativa en si mateix, i permet afegir dies festius, eliminar-los o modificar la descripció. Així mateix permet gestionar les franges horàries i l'horari del restaurant.



Cal considerar que les modificacions, només es desaran quan es polsi el botó “Desar Canvis”

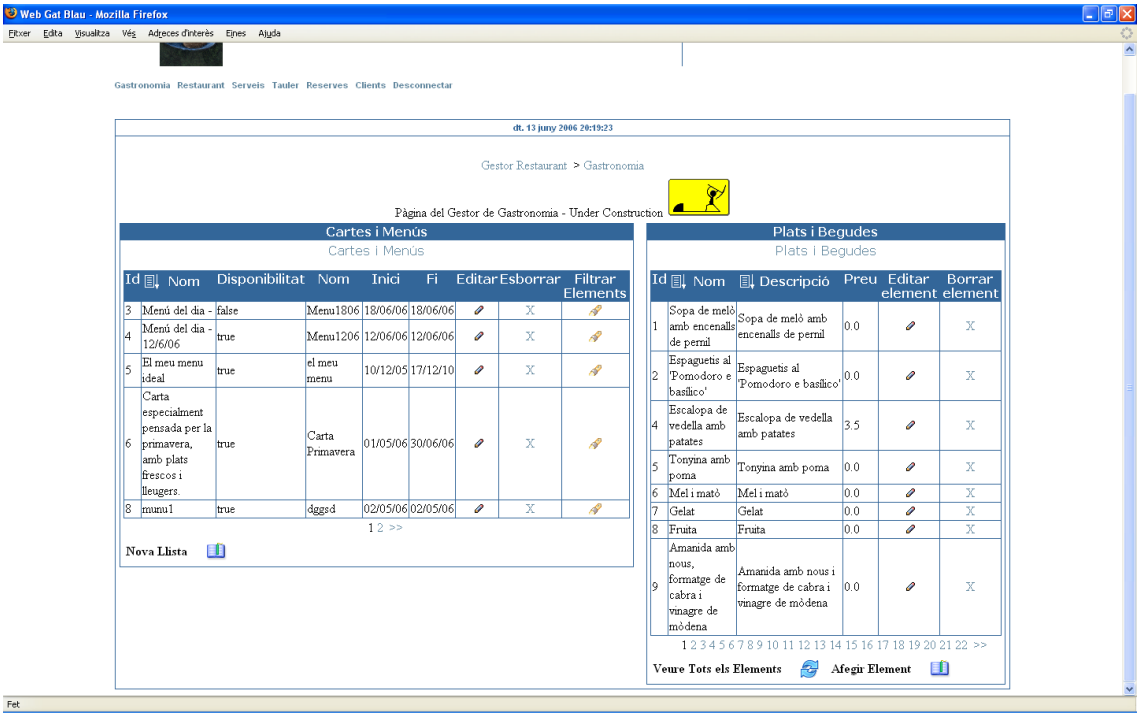
7.2.3.Gestor Gastronomia

El gestor de gastronomia té tres eines bàsiques:

1. La revisió de llistes i plats
2. L'edició de llistes
3. L'edició de plats

7.2.3.1.Revisió de llistes i plats

La figura següent mostra la pantalla inicial del gestor de gastronomia. En aquesta pàgina es pot revisar tant les llistes disponibles com els plats i begudes introduïts en el sistema.



Aquestes llistes es poden ordenar polsant l'eina , en versions posteriors es podran filtrar a fi de fer cerques més acurades.

Les següents eines ens permetran treballar amb les llistes i els seus elements.

EDICIÓ: Permet editar l'element de la llista al qual està associat. (és un signe que s'utilitzarà al llarg de l'aplicació)

X ELIMINACIÓ: Elimina l'element de la llista al qual està associat, en aquesta versió cal anar en compte per que no es demana confirmació abans d'efectuar l'operació. (és un signe que s'utilitzarà al llarg de l'aplicació)

FILTRAT: Filtra els plats, elements, de la llista d'elements que estan associats a la llista relacionada amb l'eina.

1 2 >> PAGINAT: Permet navegar pels elements no visibles de les llistes. (és un signe que s'utilitzarà al llarg de l'aplicació)

Nova Llista AFEGIR NOVA LLISTA: Afegeix una nova llista d'elements, és a dir un nou menú o carta. Condueix a la pàgina d'edició de llistes

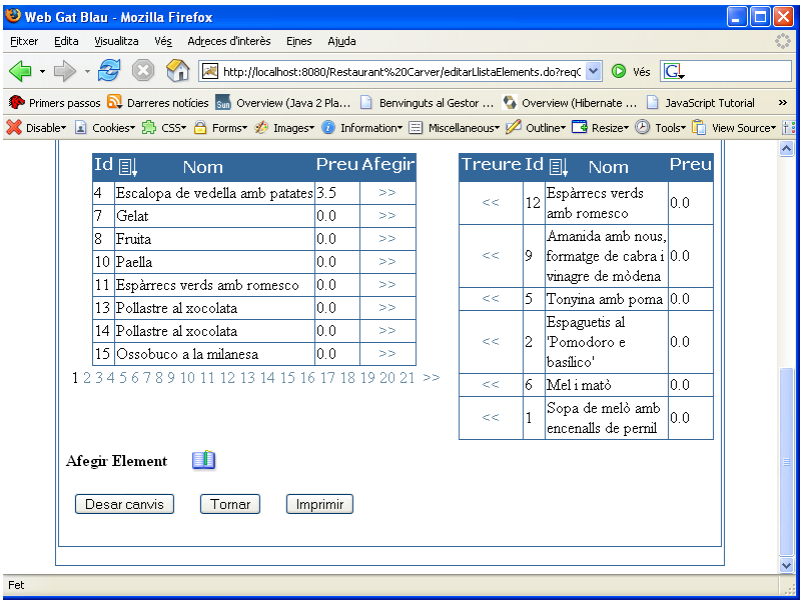
Veure Tots els Elements TREURE FILTRE: En cas que la llista de plats correspongui als que estan associats a una carta o menú, aquesta eina ens permetrà veure la llista complerta d'elements.

Afegir Element AFEGEIX ELEMENT: Mostra la pàgina que ens permet introduir un nou element, és a dir un plat, una beguda, un postre.

7.2.3.2.L'edició de llistes (cartes i menús)

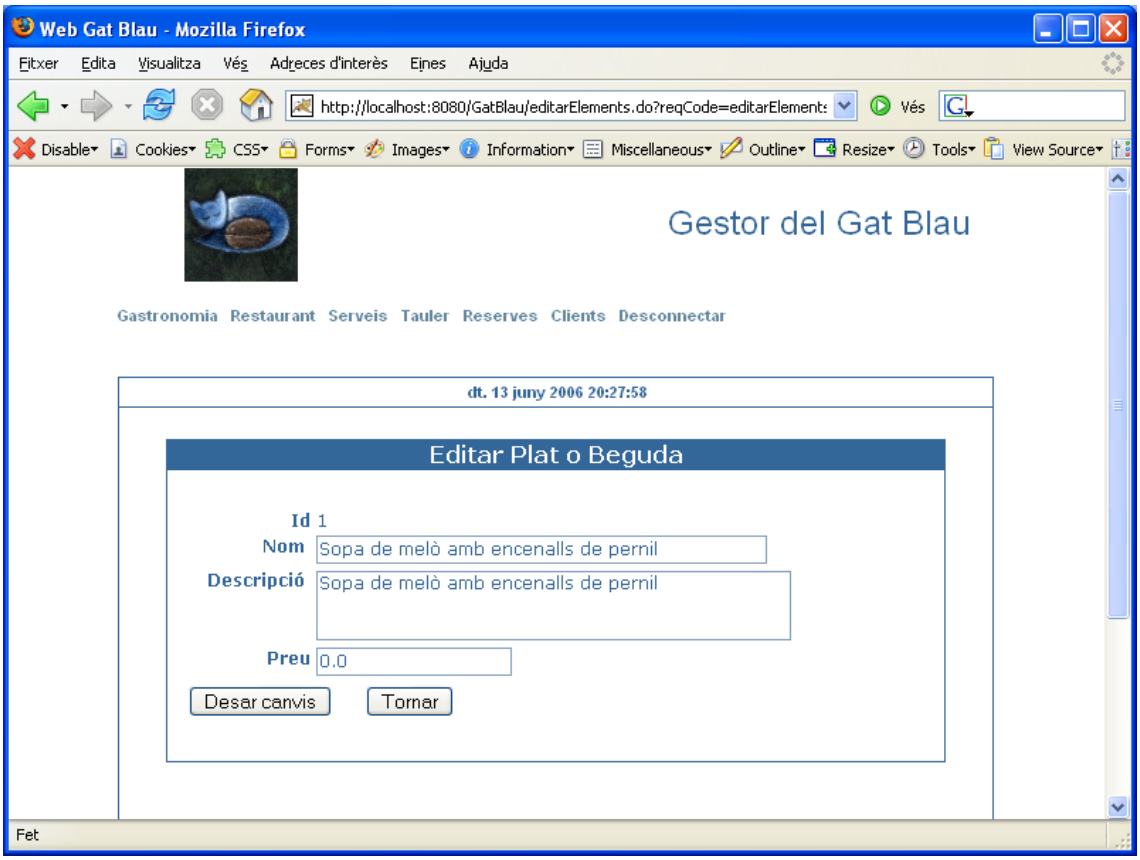
Un cop accedim a la configuració d'una llista d'elements podem editar tots els seus camps, nom, descripció dates de disponibilitat. Per tal d'afegir els elements a la llista caldrà que utilitzem l'eina afegir disponible al costat de cada element de la llista de la dreta. Per treure un element de la utilitzarem l'eina treure de forma similar.

En cas que l'element que vulguem afegir no estigui disponible, podem utilitzar l'eina "Afegir element", ara bé un cop aquest estigui donat d'alta tornarem a la pantalla de gestió de gastronomia enlloc de la d'edició d'una llista.



7.2.3.3.Edició d'elements (afegir, editar plats i begudes)

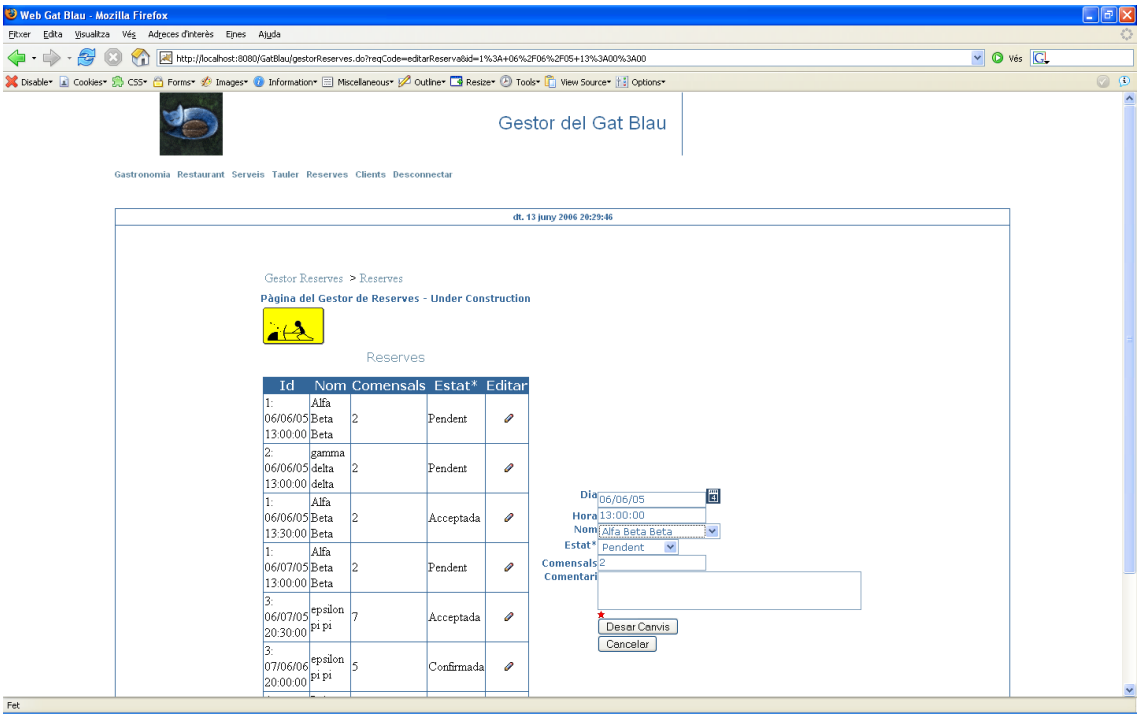
Finalment el gestor de gastronomia ens permet editar o afegir un plat o beguda, la figura següent mostra el detall de la pàgina que permet fer-ho.



7.2.4.Gestor Reserves

El gestor de reserves ens permet visualitzar les reserves existents, així com editar-les ( ) i afegir-ne de noves.

A l'esquerra queda la llista de reserves, mentre que a la dreta veiem els camps on es podran editar els valors de la reserva seleccionada o introduir una nova reserva.



7.2.5.Desconnexió

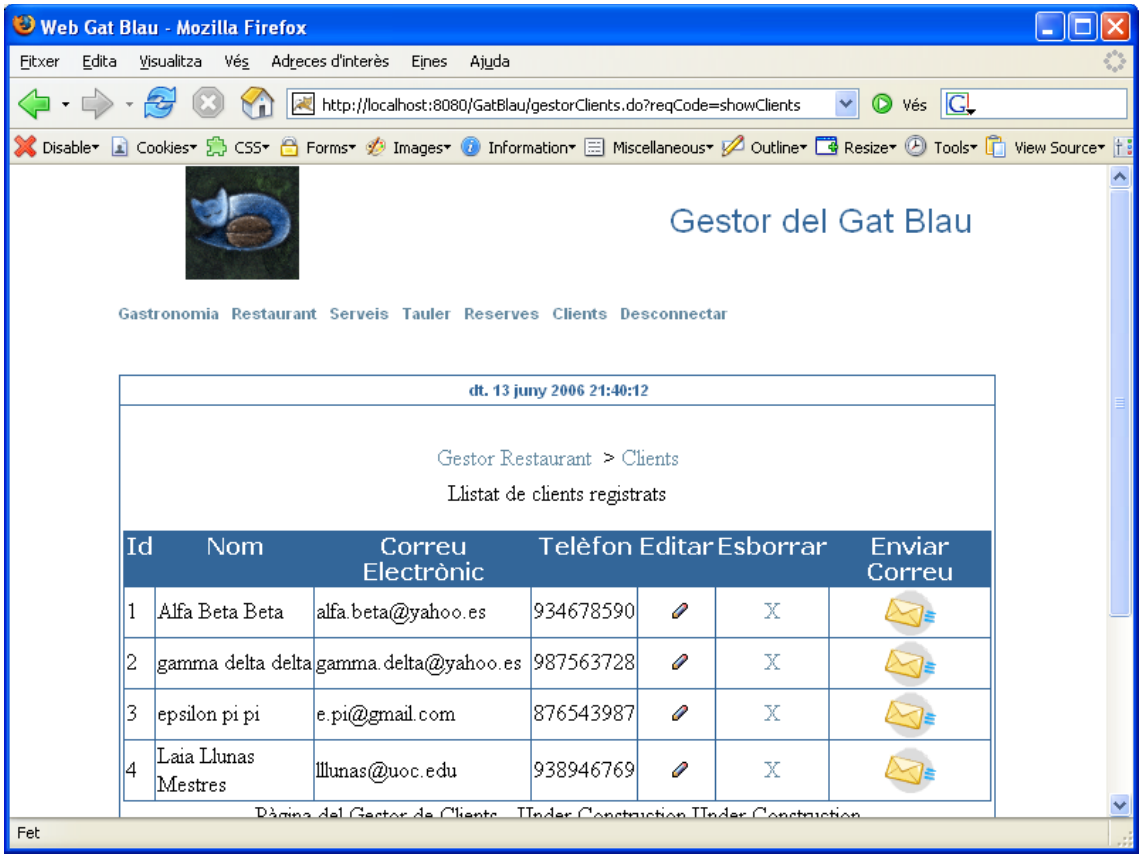
Tenint en compte que estem en una sessió d'un usuari, és recomenable desconnectar-se de la sessió. Aquest procediment elimina la informació d'usuari de la sessió, que si es tanqués el navegador no s'eliminaria fins que la sessió caduqués.

Per efectuar l'operació només cal cridar el link disponible en el menú,

Desconnectar

7.2.6.Gestor de clients

En aquesta pàgina només es mostra la llista de clients, i és que les eines que permetrien afegir-ne, editar-los, eliminar-los o enviar-los un correu electrònic encara no estan disponibles.



8. Bibliografia

Sommerville, Ian. *Ingeniería de software*. 6A edició. Naucalpan de Juárez, Mèxic. Pearson Educación. 2002

A system of patterns, F.Buschmann, R. Meunier, H. Rohnert, P. Sornmerlad, M. Stal, JOHN **WILEY** & SONS, 1996

Complete the MVC Puzzle with Struts, Sam Dalton, november 2003. http://www.sitepoint.com/article.php/1244

Struts, an open-source MVC implementation, Malcom Davis, febrer 2001, http://www-128.ibm.com/developerworks/library/j-struts/?dwzone=java

Husted, Ted, Cedric Dumoulin, George Franciscus i David Winterfeldt. *Struts in Action:Building web applications with the leading Java framework*. De 2003. Greenwich: Manning 2003

Cavaness, Chuck. *Programing Jakarta Struts: Building web applications with servlets & JSPs.* 1A ed. Sebastopol: O'Reilly & Associates, 2002

Design Patterns: Composite View. 2002 Sun Microsystems. Març 2006. http://java.sun.com/blueprints/patterns/CompositeView.html

Core J2EE Patterns – Composite View. 2001-2002. Sun Microxystems. Març 2006. http://java.sun.com/blueprints/corej2eepatterns/Patterns/CompositeView.html

Struts – Tiles. 2000-2006 Apache Software Foundation. Març 2006. http://struts.apache.org/struts-action/struts-tiles/

Struts- Layout. An extension libarary to improve intervaces creation with Struts. 06 June 2006. Improve Institute. http://struts.application-servers.com/

Bartkowiak, Dirik. "*Struts Applications and Multilanguage support*" informIT, 19 setembre 2003. http://www.informit.com/articles/printerfriendly.asp?p=101174

LaLiLuna - Tutorials for Struts, JavaServer Faces, JSF, EJB, Hibernate, Eclipse, JBoss, Tomcat, ... 2004-2006 by Sebastian Hennebrueder. Març 2006. http://www.lalilluna.de/index.html

BAUER, Christian i Gavin King. *Hibernate in Action: A guide to the concepts and pactice of object/relational mapping*. De 2005. Greenwich: Manning 2005

JBoss Inc. - Hibernate. *Hibernate Reference Documentation: HIBERNATE-Relational Persistence for Idiomatic* Java. Version: 3.1.1.

Hibernate -Relational Persistence for Java and .NET. 2002-2005 JBoss Inc . Març 2006. <www.hibernate.org>