

Central de reserves de cases de turisme rural amb tecnologia J2EE

Jordi Cerdà Folch
ETIG

Verònica Peña Pastor

9 de gener de 2006

*A la meva dona, per la seva
paciència i pel seu suport en
el desenvolupament
d'aquest projecte*

Resum

La gran importància adquirida per Internet ha fet créixer la tendència cap a aplicacions distribuïdes que usen una interfície web entre l'usuari i l'aplicació. Es tracta d'aplicacions complexes degut al nombre de tecnologies que utilitzen, ja que han d'incloure des de la interacció amb l'usuari fins a la persistència de les dades, passant per la lògica del negoci i per tots els serveis de "middleware".

L'especificació J2EE proposa una arquitectura per aquest tipus d'aplicacions basada en components i en un model de tres capes: presentació, lògica del negoci i dades. Les aplicacions basades en aquesta tecnologia tenen una major capacitat de creixement i són robustes i més senzilles de mantenir donat el seu alt grau de modularitat.

En la present memòria hi trobareu un resum del meu Treball Final de Carrera dins de l'àrea de J2EE, amb reflexions del per què del projecte i del per què de J2EE com a eina de desenvolupament.

S'exposa el procés de desenvolupament d'una aplicació corresponent a una central de reserves de turisme rural, que s'implementa mitjançant J2EE, amb una clara separació de les capes de presentació, negoci i dades.

La capa de presentació s'ha implementat amb quatre servlets, que usen diversos JSPs per a crear les vistes dels diferents casos d'ús. La capa de negoci s'amaga darrera un únic Session Bean sense estat que fa de façana i dona accés a tots els serveis que els clients necessiten. Les dades s'encapsulen dins de Entity Beans que fan servir persistència gestionada pel contenidor (CMP) i accedeixen a la base de dades. Finalment també he fet servir dos Session Beans amb estat per implementar certes funcionalitats que necessitaven recordar l'estat entre crides HTTP.

Índex de continguts

Resum	3
Índex de continguts	4
Índex de figures	6
Capítol 1. Introducció.....	7
Justificació del TFC i context en el qual es desenvolupa	7
Objectius del TFC.....	8
Enfocament i mètode seguit.	8
Planificació del projecte.	10
Productes obtinguts.	11
Breu descripció dels altres capítols de la memòria.	11
Capítol 2. Especificació de requisits	13
Descripció dels requeriments.....	13
1. Subsistema Client.....	13
2. Subsistema Propietari	14
3. Subsistema Central.....	15
Capítol 3. Anàlisi	17
Casos d'ús	17
Descripció textual dels casos d'ús	18
Capítol 4. Disseny	21
Diagrama de classes	21
Persistència.	22
Disseny conceptual de la Base de Dades	22
Disseny lògic de la Base de Dades	24
Disseny de la Interfície d'Usuari.....	25
Diagrames de seqüència dels casos d'ús	39
Cas d'ús "Accedir com a client al subsistema"	39
Cas d'ús "Consultar Cases Disponibles"	40
Casos d'ús: "Afegir reserva Carret compra" , "Finalitzar compra i obtenir factura" i "Fer pagament bestreta i factura"	41

Cas d'ús "Consultar Reserves"	42
Alta de client.....	43
Baixa de client.....	44
Modificació de client	45
Arquitectura, patrons de disseny.....	46
Capítol 5. Implementació i proves.....	48
Procés d'implementació	48
Característiques de la implementació.....	49
Llista de classes	51
Proves	52
Capítol 6. Conclusions	54
Glossari	55
Bibliografia.	56
Annexos.....	57
Programari utilitzat i instal·lació.....	57
Script de creació de la base de dades en MySQL.....	58

Índex de figures

Il·lustració 1. Gràfic Planificació	10
Il·lustració 2. Casos d'ús del Client.....	17
Il·lustració 3. Casos d'ús del Propietari	17
Il·lustració 4. Casos d'ús de l'administratiu.....	18
Il·lustració 5. Diagrama de classes	21
Il·lustració 6. Menú Aplicació.....	25
Il·lustració 7. Pantalla de log del Client.....	26
Il·lustració 8. Pantalla per enregistrar-se.	26
Il·lustració 9. Error de login	27
Il·lustració 10. Contrasenya o nom d'usuari ocupats.	27
Il·lustració 11. Fer Reserva. Buscar cases disponibles entre dates.	29
Il·lustració 12. Llistat de cases disponibles per reservar entre dues dates.	29
Il·lustració 13. Detall de la casa.....	30
Il·lustració 14. Pantalla per afegir la reserva al carret.	30
Il·lustració 15. Pantalla de reserva afegida al carret.....	31
Il·lustració 16. Carret reserves, factura i pagament	31
Il·lustració 17. Llistat reserves d'un client.	32
Il·lustració 18. Enregistrar nova casa.	33
Il·lustració 19. Llistat de cases d'un propietari.	33
Il·lustració 20. Llistat de les reserves	34
Il·lustració 21. Pantalla de modificació d'una casa.	34
Il·lustració 22. Menú Administrador.....	35
Il·lustració 23. Alta nou propietari.....	36
Il·lustració 24. Llistat propietaris.....	36
Il·lustració 25. Modificació propietari.	37
Il·lustració 26. Ampliació de la informació sobre el propietari.	37
Il·lustració 27. Alta de nova reserva.....	38
Il·lustració 28. Esborrat d'una reserva.	38
Il·lustració 29. Arquitectura de l'aplicació	50

Capítol 1. Introducció.

Justificació del TFC i context en el qual es desenvolupa

Una de les activitats de l'empresa on treballa, com a economista, és la gestió d'una central de reserves de cases de turisme rural a la comarca de Els Ports de Morella (Castelló). Aquesta gestió és realitzada encara mitjançant el telèfon, és a dir, els clients que desitgen passar uns dies en alguna de les cases de la central es posen en contacte amb el personal de l'empresa a través del telèfon i si el servei ofert els agrada fan la reserva.

Sempre he tingut clar que el proper pas cap a una millora del servei cap als clients i una reducció de costos en la gestió passava per la realització de totes aquestes accions a través d'Internet. Aquest és, per tant, un dels motius pels quals em vaig sentir interessat en desenvolupar aquest projecte.

Un altre motiu, no menys important, és que considerava molt interessant per la meua formació no acabar la carrera sense haver tingut la oportunitat de desenvolupar un projecte de creació de software sencer, i donada la tendència actual cada vegada més clara cap a aplicacions distribuïdes amb interfície web, la tecnologia J2EE m'oferia tot allò que estava buscant.

També he de dir que vaig ser conscient des del principi que fer un projecte amb J2EE partint simplement dels coneixements de Java que havia adquirit durant la carrera era una tasca força complicada degut al nombre de tecnologies que és necessari dominar per desenvolupar una aplicació J2EE, i al poc temps disponible per fer el projecte. Aquesta ha estat, amb diferències la assignatura de tota la carrera a la que he hagut de dedicar més temps, però també penso que l'esforç m'ha portat a obtenir un aprenentatge molt important, que és el que en definitiva volia.

Dins de la tecnologia J2EE encara hem de prendre decisions sobre quines de les tecnologies que aquesta engloba triem per a fer el desenvolupament. Jo, seguint les recomanacions de la consultora i tenint el compte el poc temps per al desenvolupament del projecte i el meu punt de partida, he optat per fer servir Enterprise JavaBeans (EJB) que són components de negoci que s'executen en el servidor d'aplicacions (JBoss en el meu cas). Dins d'aquesta tecnologia dels EJB he optat per fer servir EJB d'entitat amb persistència gestionada pel contenidor (CMP), i EJB de sessió tant amb estat com sense estat. Per a les capes web i de presentació he optat per servir servlets i JSP i com a client qualsevol navegador web.

Pel que fa als patrons de disseny, en un principi vaig estar valorant la possibilitat d'utilitzar el bastiment Struts per implementar el patró Model View Controller 2 (MVC2), però ho vaig descartar pel poc temps disponible.

Finalment haig fet servir el patró MVC2 (però sense Struts) i el Session Façade, posant un EJB de sessió sense estat on s'adrecen totes les peticions dels clients.

En definitiva, en aquest TFC, l'estudiant, amb coneixements de Java però sense experiència prèvia en aplicacions distribuïdes, analitza, dissenya i implementa un exemple típic d'aplicació de comerç electrònic.

L'ús de la tecnologia J2EE m'ha permès no haver-me de preocupar de les funcionalitats del "middleware", obtenint d'aquesta forma guanys en temps, portabilitat i fiabilitat. El seguiment d'alguns patrons de disseny afavoreix l'ús de solucions eficaces i d'aquesta manera permet anticipar i evitar part dels errors i complicacions innecessàries que em podria trobar per falta d'experiència.

Objectius del TFC.

L'objectiu principal del TFC ha estat la introducció en l'ús de l'arquitectura J2EE i en les tècniques modernes de desenvolupament d'aplicacions empresarials per a Internet.

El següent objectiu del TFC ha estat desenvolupar una aplicació per a la gestió d'una central de reserves de cases de turisme rural i aquesta dividida en tres subsistemes: per una banda la part de l'aplicació on els clients interaccionen amb el programa per tal de fer les reserves, per un altra la part on els propietaris de cases de turisme rural fan l'alta de les seves cases i consulten les reserves fetes i finalment el subsistema administratiu on el empleats de la central de reserves tenen privilegis per fer qualsevol de les operacions disponibles a l'aplicació.

Enfocament i mètode seguit.

El desenvolupament del projecte segueix de manera simplificada el procés unificat del desenvolupament de programari. Es divideix, per tant, en les fases típiques del desenvolupament: especificació de requisits, anàlisi, disseny, implementació i proves, però fent iteracions en cadascuna d'aquestes fases.

En la primera fase d'especificació de requisits es presenta una descripció textual dels requisits que ha de complir l'aplicació.

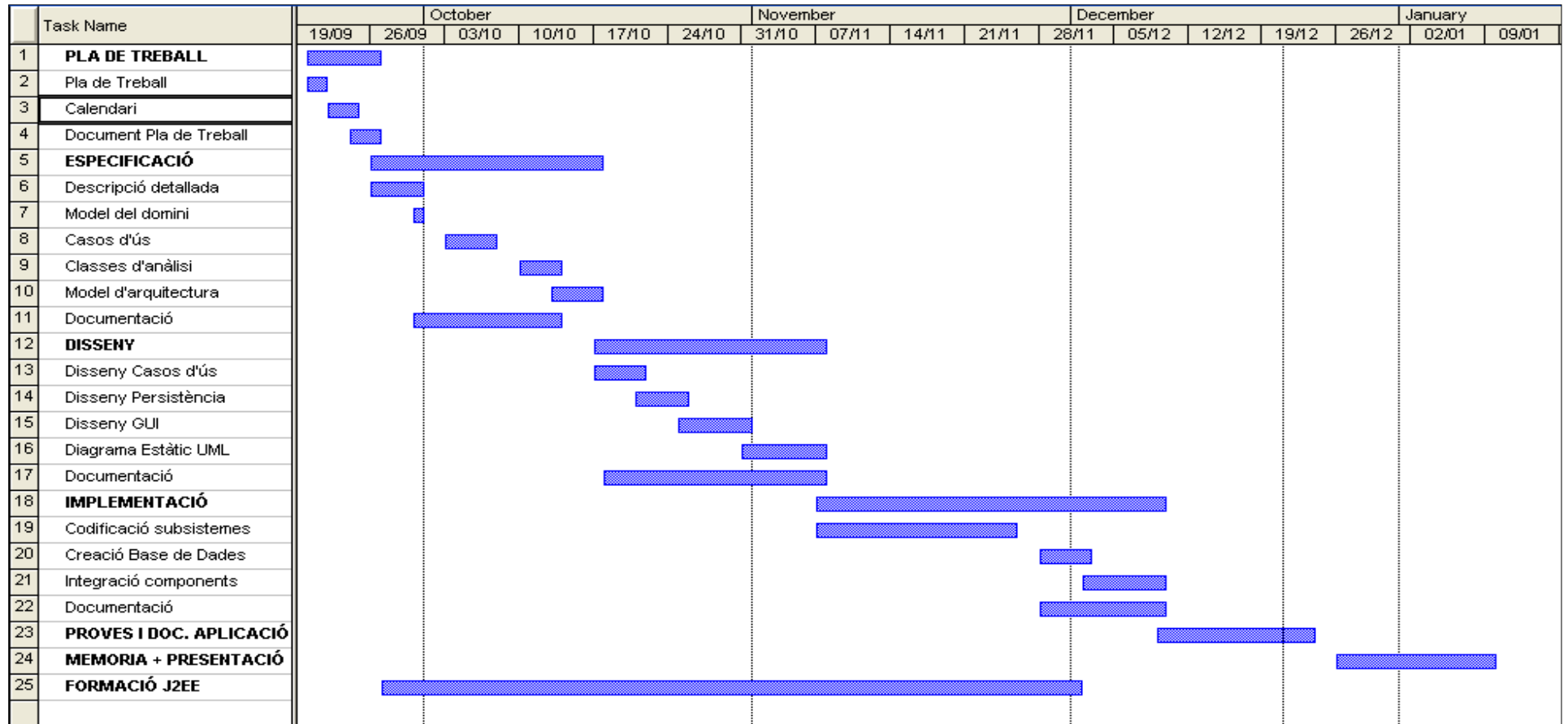
En la segona fase d'anàlisi s'obtenen els casos d'ús de cada subsistema i el diagrama d'entitats.

En la tercera fase de disseny es fa el disseny de la persistència, el disseny de la interfície d'usuari i els diagrames de seqüència dels casos d'ús.

En la quarta fase d'implementació es fa la codificació de cada subsistema, tant pel que fa a la codificació de les classes EJB, com a la dels servlets, JSP i les pàgines HTML. És conjuntament amb aquesta fase, que també es fan les proves de forma iterativa a mesura que es va codificant cadascuna de les funcionalitats. Aquestes proves les haig fetes tant amb classes Java que fan de

client com directament des de les pàgines JSP. Per poder fer les proves cal, cada vegada, fer el desplegament del codi en el servidor d'aplicacions.

Planificació del projecte.



II-Il·lustració 1. Gràfic Planificació

La planificació anirà marcada en tot moment per les dates dels lliuraments a realitzar, com es pot veure a l'anterior diagrama:

Lliurament	Data
PAC1 - Pla de Treball	26/09/2005
PAC2 - Disseny	07/11/2005
PAC3 - Implementació	09/12/2005
Memòria + Presentació	09/01/2005

Productes obtinguts.

Al final del projecte s'obté una aplicació funcional, on s'han implementat la pràctica totalitat de les funcionalitats que inicialment m'havia proposat.

Tota l'aplicació està continguda en un arxiu "casesrurals.ear" que conté tots els elements necessaris que requereix l'especificació J2EE per poder-se desplegar correctament en un servidor d'aplicacions (en aquest cas JBoss):

1. Arxiu "casesrurals.jar": Encapsula la lògica del negoci i està format bàsicament per:
 - Descriptor de Desplegament "ejb-jar.xml": Descriu el comportament i l'esquema abstracte de persistència dels EJB.
 - Directoris amb cadascuna de les classes dels EJB: Un directori de classes client per fer proves, un de Value Objects, un altre per les interfícies de cada EJB i finalment un per a les classes de component de cada EJB.
2. Arxiu "presentacio.war": Aquest encapsula a la presentació i està format bàsicament per:
 - Descriptor de Desplegament web.xml : Inclou diferents tipus d'informació sobre els components web.
 - Directoris amb els quatre servlets controladors i altres classes que es fan servir, amb les pàgines JSP, amb una fulla d'estil CSS i amb les imatges que fan servir les pàgines JSP.

Per un altra banda, des del punt de vista de l'usuari de l'aplicació, podríem considerar aquesta dividida en tres parts o subsistemes:

L'aplicació té dues parts que podríem anomenar públiques, una de elles és la que dona accés als serveis oferts als clients, on aquestos poden enregistrar-se, fer reserves de les cases disponibles, obtenir informació d'aquestes etc..

L'altra part pública de l'aplicació és la que ofereix al propietaris de cases de turisme rural la possibilitat d'enregistrar-se per tal de poder dona d'alta les seves cases i també per poder consultar les reserves fetes pels clients.

Finalment, tenim la part privada on els administratius de l'empresa que gestiona la central poden fer qualsevol de les operacions que permet l'aplicació.

Breu descripció dels altres capítols de la memòria.

Després d'aquesta breu presentació del projecte, algunes de les seves problemàtiques, la metodologia i algunes de les decisions que s'han pres sobre el camí, en els següents capítols es mostren els diferents resultats

obtinguts de les tres fases del projecte: especificació dels requisits, anàlisi, disseny i algunes decisions preses durant la implementació i les proves.

Segueixen les conclusions que s'han extret de la realització del projecte, el glossari que utilitzarem per comunicar-nos amb el client, la bibliografia consultada per a la realització del projecte i, finalment, els annexos a aquesta memòria, on entre altres tindrem el procés d'instal·lació de l'aplicació.

Capítol 2. Especificació de requisits

Descripció dels requeriments.

Els objectius del projecte són implementar les funcions bàsiques per a la creació, manteniment i consulta de reserves de cases de turisme rural en una base de dades. El sistema està pensat per donar resposta a les necessitats pròpies d'una central de reserves de turisme rural i constarà d'una sèrie de pantalles ben estructurades per facilitar la navegació. L'aplicació per una part oferirà informació sobre les cases i la seva disponibilitat i després permetrà al client fer la reserva de la casa en cas de que aquesta estigues lliure per a les dates indicades. També traurà una factura i permetrà fer el pagament d'una quantitat en concepte de bestreta per la reserva.

L'aplicació es pot dividir en tres subsistemes o parts:

Subsistema Client: Agruparà les funcionalitats del client que fa la reserva de l'allotjament.

Subsistema Propietari: Agruparà les funcionalitats dels propietaris de les cases de turisme rural.

Subsistema Central: Agruparà les funcionalitats dels administratius de l'empresa que gestionen la central de reserves.

El programari consistirà d'una part servidora implementada amb tecnologia J2EE, ubicada en una seu central, i tres clients "lleugers" corresponents als clients de faran les reserves, als propietaris i als administratius de l'empresa, que accediran tots tres a la central de reserves des d'Internet usant qualsevol navegador convencional i tindran accés a unes determinades funcionalitats de l'aplicació en funció de com s'identifiquen en accedir a aquesta.

El disseny i implementació de la central de reserves virtual ha de ser en arquitectura de 3 capes, usant J2EE, on se separi clarament la presentació, la lògica de negoci i l'accés a les dades. La persistència de les dades serà en BD relacional, concretament MySQL. L'escalabilitat, fiabilitat i portabilitat seran les inherents a la tecnologia J2EE.

1. Subsistema Client

1.1. Enregistrar client: Abans que el client pugui accedir a les funcionalitats de l'aplicació, cal que s'enregistri al sistema com a client. Per això, l'aplicació ens mostrarà una pantalla on donarà la possibilitat de entrar al sistema, si el client ja està enregidrat, o de donar-se d'alta com a client en cas contrari. En aquest segon cas, se'ns mostrarà una pantalla on el client haurà d'emplenar un formulari amb les seves dades, entre elles triar un nom d'usuari i una contrasenya. Si el nom d'usuari ja és a la base de dades el sistema informarà d'aquest fet al client i li donarà la possibilitat de triar-ne una altre.

1.2. Accedir com a client al subsistema: Com he dit a dalt, si el client ja està enregistrat al sistema, accedirà a aquest posant el seu nom d'usuari i contrasenya. Si s'equivoca en teclejar la informació el sistema informarà d'aquest fet al client i li donarà la possibilitat de tornar a provar. Una vegada el client hagi accedit al sistema se li mostraran de forma clara totes les opcions que té disponibles.

1.3. Consultar cases disponibles: En clicar sobre aquesta opció es mostrarà una pantalla on el client podrà emplenar en un formulari el període de temps per al qual vol reservar la casa, la zona, i el nombre de persones. Ens apareixerà un llistat de totes les cases que compleixen les condicions anteriors. En el llistat apareixerà el nom de la casa, situació, preu per nit i dos botons, un per obtenir o ampliar informació sobre la casa i l'altre per afegir la reserva de la casa al carret de compra.

1.4. Obtenir més informació: Ens mostrarà tota la informació disponible sobre la casa a la base de dades en una finestra apart. En aquesta finestra no es podrà fer cap modificació de la informació.

1.5. Afegir reserva al carret de compra: Ens permetrà afegir la reserva al carret de la compra. El sistema ens informarà que la reserva s'ha afegit i ens donarà dues possibilitats: afegir un altra reserva o finalitzar la compra i obtenir la factura.

1.6. Finalitzar compra i obtenir factura: Si triem finalitzar la compra el sistema ens mostrarà la factura de les reserves fetes i la quantitat a abonar com a bestreta.

1.7. Fer pagament bestreta i reservar: En prémer pagar, el sistema ens demanarà el número de la targeta i després d'acceptar introduirà la reserva al sistema. El pagament de la bestreta no es farà efectiu fins a les 24 h. de haver fet la reserva.

1.8. Anul·lar reserva: Quan el client prem sobre l'opció anul·lar reserva se li mostra una llista de les reserves que té fetes (cas d'ús "consultar reserves client"). Ara té l'opció de anul·lar la reserva o reserves que desitgi. Una vegada fa l'anul·lació el sistema pot presentar dues pantalles: si la reserva es va fer abans de 24 h. se'ns mostra un missatge com a que la reserva està anul·lada i la bestreta no es cobrarà, si la reserva es va fer fa més de 24 h. el sistema avisa abans d'anul·lar la reserva que la bestreta no serà tornada tot i que s'acull-li la reserva.

1.9. Consultar reserves client. Aquesta opció mostra un llistat de les reserves que el client té en aquest moment.

2. Subsistema Propietari

2.1. Donar d'alta propietari: Abans que el propietari pugui accedir a les funcionalitats de l'aplicació, cal que s'enregistri al sistema com a propietari.

Per això, l'aplicació ens mostrarà una pantalla on donarà la possibilitat de entrar al sistema, si el propietari ja està enregistrat, o de donar-se d'alta com a propietari en cas contrari. En aquest segon cas, se'ns mostrarà una pantalla on el propietari haurà de emplenar un formulari amb les seves dades, entre elles triar un nom d'usuari i una contrasenya. Si el nom d'usuari ja és a la base de dades el sistema informará d'aquest fet al client i li donarà la possibilitat de triar-ne una altre.

2.2. Accedir com a propietari al subsistema: Com he dit a dalt, si el propietari ja està enregistrat al sistema, accedirà a aquest posant el seu nom d'usuari i contrasenya. Si s'equivoca en teclejar la informació el sistema informará d'aquest fet al propietari i li donarà la possibilitat de tornar a provar. Una vegada el propietari hagi accedit al sistema se li mostraran de forma clara totes les opcions que té disponibles.

2.3. Fer alta provisional de casa rural: En prémer aquesta opció es mostra un formulari amb les característiques a emplenar corresponents a una casa de turisme rural. En fer acceptar les dades emplenades es graven a la base de dades corresponent, però amb estat "no disponible". És a dir, una vegada els responsables de la central se'n adonen que s'ha introduït una nova casa hauran d'anar en persona a revisar-la i si tot està bé la passaran al estat "disponible" on ja podrà ser utilitzada per fer reserves on-line.

2.4. Consultar reserves: Finalment els propietaris podran consultar les reserves que tenen les seves cases per tal que saber quin dia vindran els clients a la casa i quin dies sortiran.

3. Subsistema Central

3.1. Accedir Administratiu a subsistema: L'administratiu o administratius de l'empresa responsables de la gestió de la central tindran una clau que els permetrà el màxims privilegis en quan a la gestió de clients, cases, propietaris i reserves. Accedirà a l'aplicació posant el seu nom d'usuari i contrasenya. Si s'equivoca en teclejar la informació el sistema informará d'aquest fet al usuari i li donarà la possibilitat de tornar a provar. Una vegada l'usuari hagi accedit al sistema se li mostraran de forma clara totes les opcions que té disponibles.

3.2. Manteniment de clients: Aquesta funcionalitat permetrà afegir un nou client al sistema, esborrar-lo o modificar les seves dades. Per afegir un client l'aplicació ens mostrarà un formulari on s'hauran d'emplenar totes les dades del client, excepte el nom d'usuari i contrasenya. Per modificar o esborrar abans s'haurà de fer una cerca del client, en el qual apareixeran les seves característiques principals per poder identificar-los. Des d'aquesta pantalla es podran modificar les dades de qualsevol dels clients o esborrar-los de la base de dades.

3.3. Manteniment de cases: Aquesta funcionalitat ens permetrà afegir, modificar, esborrar i consultar cases a la base de dades. Per afegir una casa

l'aplicació ens mostrarà un formulari on s'hauran d'emplenar totes les dades de la casa. Per modificar o esborrar abans s'haurà de fer una cerca de les cases, en el qual apareixeran les seves característiques. Des d'aquesta pantalla es podran modificar les característiques de qualsevol de les cases o esborrar la casa de la base de dades (deixar-la en estat de "no disponible").

3.4. **Manteniment de Propietaris:** Aquesta funcionalitat permetrà afegir un nou propietari al sistema, esborrar-lo o modificar les seves dades. Per afegir un propietari l'aplicació ens mostrarà un formulari on s'hauran d'emplenar totes les dades del propietari, excepte el nom d'usuari i contrasenya. Per modificar o esborrar abans s'haurà de fer una cerca del propietari. Des d'aquesta pantalla es podran modificar les dades de qualsevol dels propietaris o esborrar-los de la base de dades.

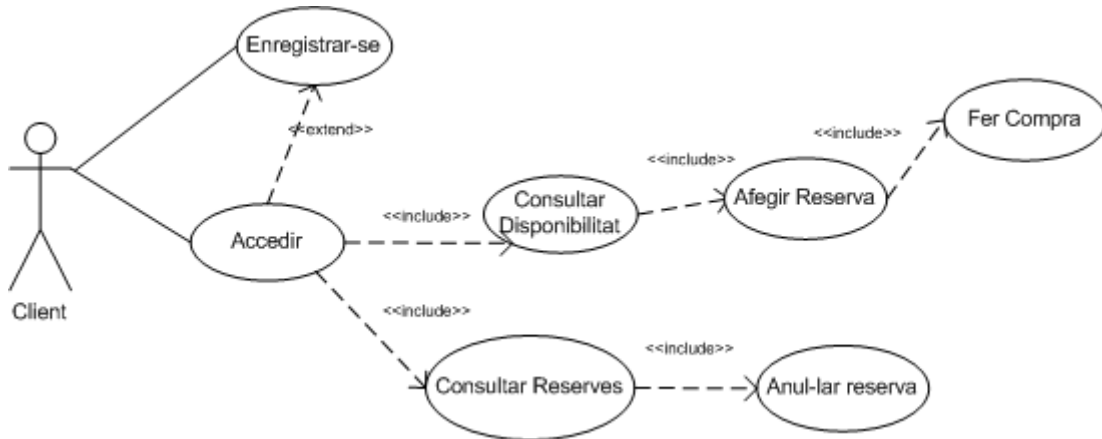
3.5. **Manteniment de reserves:** Aquesta funcionalitat ens permetrà fer una reserva per un client determinat per al cas de clients que no poden o no volen fer les reserves on-line. També ens permetrà modificar o esborrar reserves, per al cas de dades introduïdes incorrectes, etc.. Per poder modificar o esborrar primer s'haurà de fer una cerca de la reserva en concret i després esborrar-la o modificar-la.

3.6. **Consultar reserves:** Amb aquesta opció l'administratiu podrà treure llistats segons diferents criteris per tal de veure les reserves que estan fetes en el moment de fer la consulta.

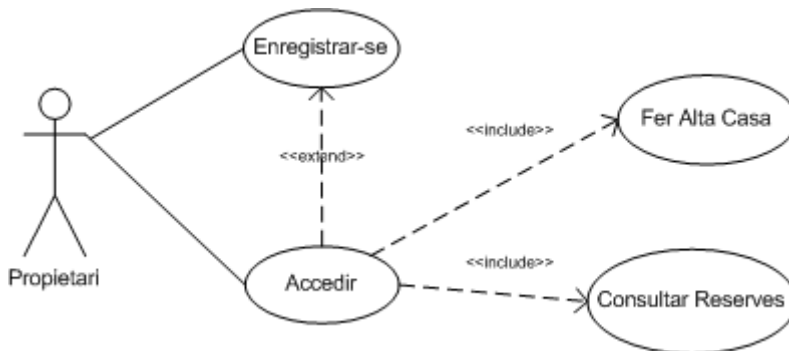
3.7. **Consultes i estadístiques diverses:** Aquesta funcionalitat ens permetrà obtenir diferents tipus d'estadístiques d'ocupació de les cases durant un període de temps. Per pantalla hauré de posar una data d'inici i una de fi i marcar la casella cases i el sistema ens traurà un llistat de totes les cases i el nombre de dies en que han estat ocupades cadascuna d'elles durant el període. Si marquem la casella clients el sistema ens traurà un llistat dels dies de pernoctació de cada client durant el període introduït.

Capítol 3. Anàlisi

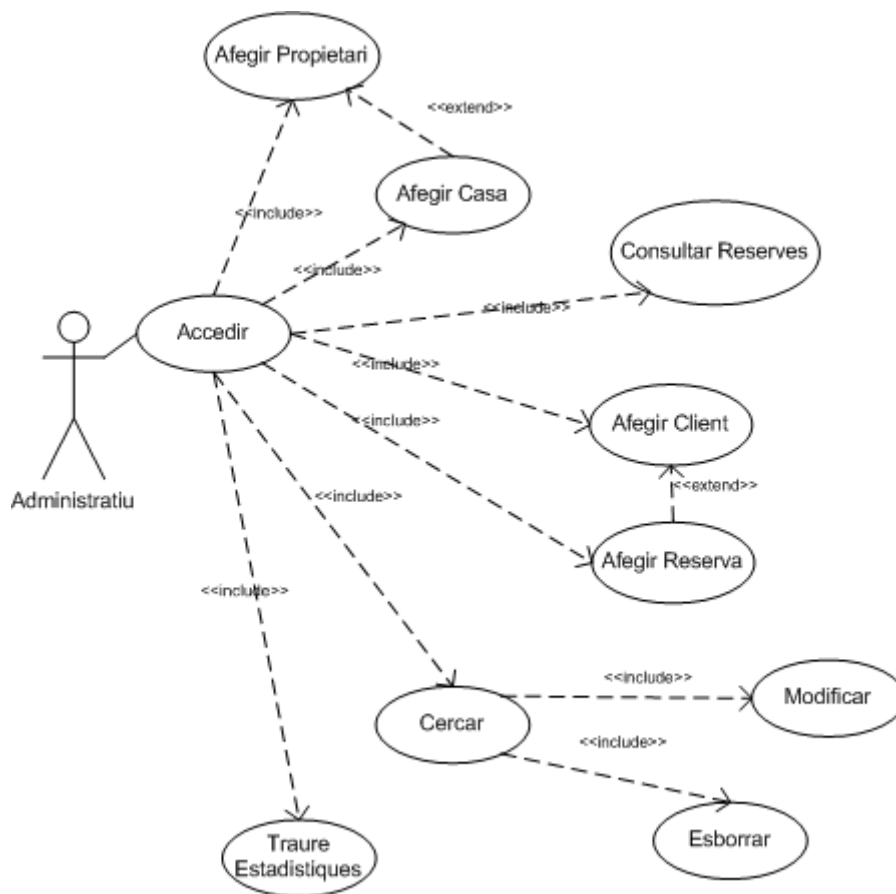
Casos d'ús



Il·lustració 2. Casos d'ús del Client.



Il·lustració 3. Casos d'ús del Propietari



II·lustració 4. Casos d'ús de l'administratiu

Descripció textual dels casos d'ús

Cas d'ús: Enregistrar-se.

Resum funcionalitat general: Enregistrar al sistema un usuari nou.

Paper dins el treball de l'usuari: No és un dels casos principals dels actors.

Actors: Client i Propietari.

Casos d'ús relacionats: Accedir

Precondició: L'usuari no està enregistrat a la base de dades.

Postcondició: Les dades del client o propietari s'han incorporat a la base de dades.

Descripció:

L'aplicació ens mostrarà una pantalla on donarà la possibilitat de entrar al sistema, si el client ja està enregistrat, o de donar-se d'alta com a client en cas contrari. En aquest segon cas, se'ns mostrarà una pantalla on el client haurà de emplenar un formulari amb les seves dades, entre elles triar un nom d'usuari i una contrasenya. Si el nom d'usuari ja és a la base de dades el sistema informarà d'aquest fet al client i li donarà la possibilitat de triar-ne una altre.

Cas d'ús: Accedir.

Resum funcionalitat general: Permet l'accés a l'aplicació a un usuari registrat.

Paper dins el treball de l'usuari: És un dels casos d'ús principals dels actors.

Actors: Client, Propietari, Administratiu.

Casos d'ús relacionats: Enregistrar-se.

Precondició: L'usuari està enregistrarat a la base de dades.

Postcondició: L'usuari accedeix a l'aplicació.

Descripció:

Si l'usuari ja està enregistrarat al sistema, accedirà a aquest posant el seu nom d'usuari i contrasenya. Si s'equivoca en teclejar la informació el sistema informarà d'aquest fet al usuari i li donarà la possibilitat de tornar a provar. Una vegada l'usuari hagi accedit al sistema se li mostraran de forma clara totes les opcions que té disponibles.

Cas d'ús: Consultar Disponibilitat

Resum funcionalitat general: Permet al client consultar la disponibilitat de les cases per a un període de temps.

Paper dins el treball de l'usuari: És un dels casos d'ús principal del client.

Actors: Client

Casos d'ús relacionats: Afegir reserva, Accedir.

Precondició: Cap

Postcondició: Cap, en ser una consulta.

Descripció:

El client podrà emplenar en un formulari el període de temps per al qual vol reservar la casa, la zona, i el nombre de persones. Ens apareixerà un llistat de totes les cases que compleixen les condicions anteriors. Des d'aquí podrà executar el cas d'ús "afegir reserva".

Cas d'ús: Fer compra.

Resum funcionalitat general: L'usuari executa la compra d'un servei consistent en la reserva d'una o més cases.

Paper dins el treball de l'usuari: És un dels casos d'ús principal del client.

Actors: Client

Casos d'ús relacionats: Accedir, consultar disponibilitat, afegir reserva.

Precondició: Cap.

Postcondició: La reserva o reserves queden enregistrades a la base de dades amb totes les dades.

Descripció:

Tot el procés comença amb l'accés del client a l'aplicació (Cas d'ús "accedir"). Després el client executa el cas d'ús "consultar disponibilitat" amb el qual obté un llistat de les possibles reserves que pot fer i que compleixen les condicions. Des d'aquí el programa ens permetrà afegir la reserva al carret de la compra. El sistema ens informarà que la reserva s'ha afegit i ens donarà dues possibilitats: afegir un altra reserva o finalitzar la compra. Si finalitzem la compra el programa ens mostrarà la factura i ens convidarà a que emplenem les dades de la targeta de crèdit per fer el

pagament de la bestreta. Una vegada emplenades les dades de forma correcta, en prémer acceptar la compra es farà efectiva i la reserva quedarà enregistrada a la base de dades.

Cas d'ús: Consultar Reserves

Resum funcionalitat general: Trau un llistat de les reserves actives d'un client o d'un propietari.

Paper dins el treball de l'usuari: És un dels casos d'ús principal del propietari i accessori del client.

Actors: Propietari i client.

Casos d'ús relacionats: Accedir, Anul·lar reserva

Precondició: Cap.

Postcondició: Cap, en ser una consulta

Descripció:

Aquesta opció traurà un llistat de les reserves que tenen actives tant un client com un propietari. Donarà informació sobre el dia d'entrada i de sortida, el nom del client, la casa etc..

Cas d'ús: Consultar reserves

Resum funcionalitat general: Consultar les reserves que estan actives en el moment actual.

Paper dins el treball de l'usuari: És un dels casos d'ús principals de l'administratiu.

Actors: Administratiu

Casos d'ús relacionats: Accedir

Precondició: Cap

Postcondició: Cap, en ser una consulta.

Descripció: En un formulari l'administratiu emplenarà camps com "dia d'entrada" i/o "dia de sortida", "client", "casa" etc. I segons aquestos criteris de cerca el programa traurà per pantalla un llistat de les reserves que es troben actives en el moment de fer la consulta.

Cas d'ús: Cercar.

Resum funcionalitat general: Trobar una reserva, casa, client o propietari a la BD.

Paper dins el treball de l'usuari: Cas d'ús principal, sobre tot la part que correspon a la reserva.

Actors: Administratiu

Casos d'ús relacionats: Modificar, esborrar.

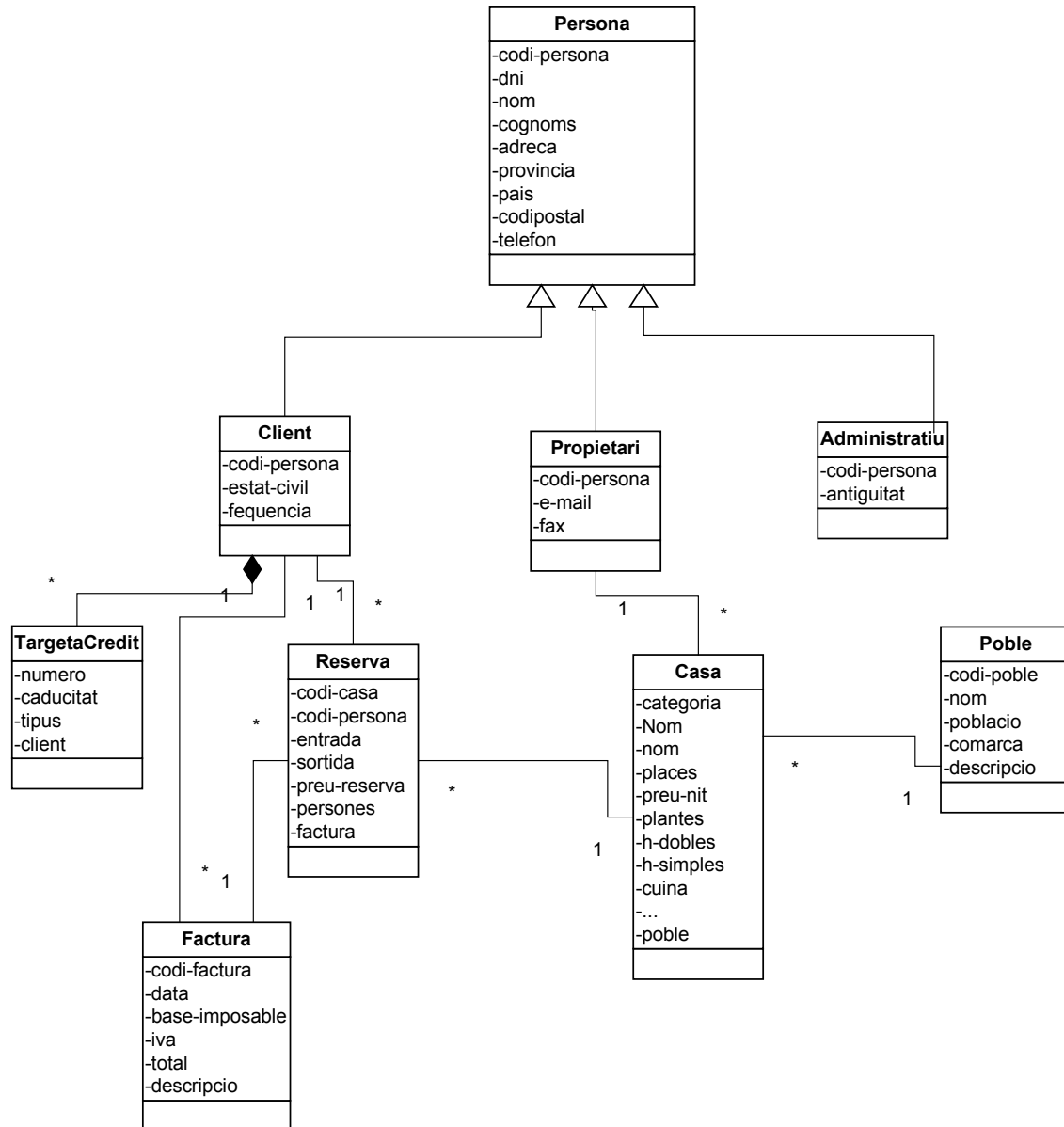
Precondició: El registre corresponent existeix a la base de dades.

Postcondició: Ens apareix per pantalla la reserva, client, propietari o casa buscada.

Descripció: Ens apareix per pantalla un formulari on l'administratiu pot introduir alguna de les dades principals de la reserva que està buscant i en fer acceptar l'aplicació obté de la base de dades la reserva buscada. El mateix per al cas de que l'administratiu estigui cercant una casa, un propietari o un client.

Capítol 4. Disseny

Diagrama de classes



Il·lustració 5. Diagrama de classes

Relació de classes:

- **Persona**: Classe que representa a una persona amb els atributs comuns de nom, cognoms, dni, etc.. Aquesta classe serà la superclasse de les entitats **Propietari**, **Client** i **Administratiu** que són els tres actors que intervenen en els casos d'ús de l'aplicació.
- **Propietari**, **Client** i **Administratiu**: Classes que representen els actors del sistema respectivament.

- Casa: Classe que representa una casa de turisme rural. Els seus atributs principals són el nom de la casa, el preu per nit i tota una sèrie de característiques de la casa que un client pot estar interessat en conèixer a l'hora de fer la reserva.
- Reserva: Classe que representa una reserva d'una casa de turisme rural. Els seus atributs principals seran la casa que es reserva, el client que fa la reserva, la data d'entrada i de sortida de la casa, el nombre de persones, etc..
- Factura: Classe que representa una factura de una o més reserves. Els seus atributs són bàsicament els camps típics de tota factura, data, base imposable, iva, total.
- Targeta: Classe que representa una targeta d'un client. Necessària a l'hora de fer el pagament de la reserva. Els seus atributs són número de la targeta, data de caducitat i tipus de targeta (American Express, Visa etc..).
- Poble: Classe que representa un poble on hi ha una casa de turisme rural. Els seus atributs són el nom del poble, la comarca on es troba, la població i una descripció.

Relacions entre classes:

- Casa – Propietari: Tenen una relació d'associació de ú a molts. És a dir, un propietari pot ser-ho de varies cases, però una casa sols pot ser d'un propietari.
- Casa – Reserva: És una relació d'associació de ú a molts. Una casa pot tenir varies reserves però una reserva sols pot correspondre a una casa.
- Client – Reserva: És una relació d'associació de ú a molts. Un client pot fer varies reserves però una reserva sols pot correspondre a un client.

Persistència.

Disseny conceptual de la Base de Dades

En aquesta primera etapa del disseny de la persistència obtindrè un esquema conceptual expressat en el model ER. Aquesta esquema és independent de la tecnologia que s'utilitzi per implementar la Base de Dades. No es té en compte, per tant, la base de dades que s'utilitzarà Això ens permet concentrar-nos únicament en la problemàtica de la estructuració de la informació.

Comentaris al disseny:

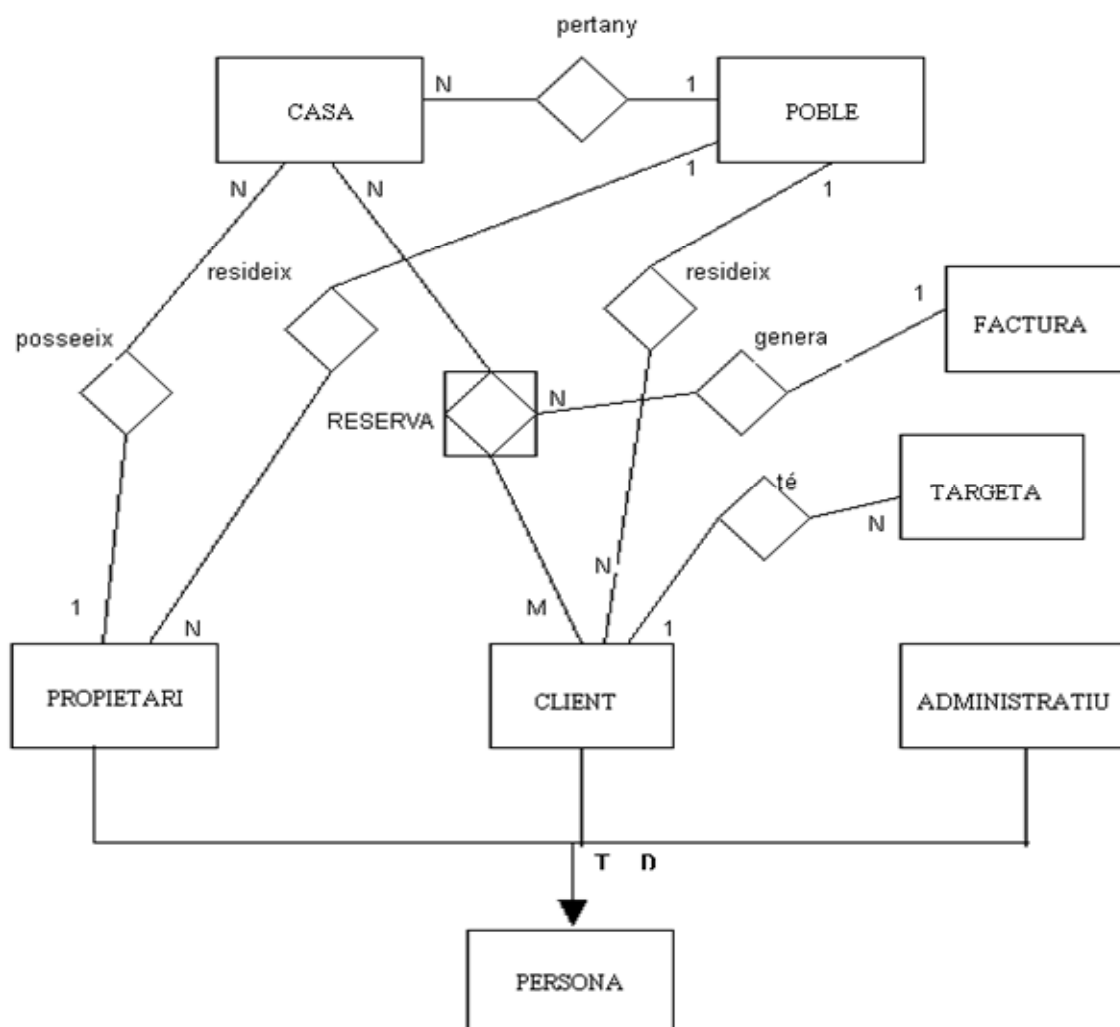
- Generalització/especialització: He derivat tres entitats subclasse de la superclasse Persona, de forma que les característiques de l'entitat Persona es propaguen cap a les entitats subclasse. En aquesta cas he considerat que la generalització/especialització és disjunta, ja que cap persona no pot ser de més d'un tipus, i total, ja que cada ocurrència

de l'entitat Persona ha de pertànyer a alguna de les entitats subclasse.

Interrelacions entre les entitats:

- Propietari – Casa: Un propietari pot tenir varies cases, però una casa sols pot pertànyer a un propietari.
- Casa – Poble: Una casa està dins del terme d'un únic poble, però dins d'un poble hi poden haver varies cases.
- Client – Targeta: Un client pot tenir varies targetes de crèdit, però una targeta sols pot pertànyer a un client.
- Client – Casa: Un client pot reservar varies cases i una casa pot estar reservada per varis clients. Aquesta relació MN ens genera una entitat associativa, que anomeno reserva. Aquesta entitat reserva tindrà per tant els seus propis atributs que després concretaré.
- Reserva – Factura: A una factura li poden correspondre una o varies reserves, en canvi una reserva sols pot tenir una factura.

L'estructura obtinguda és la següent:



Disseny lògic de la Base de Dades

Ara farem la transformació del model ER anterior al model relacional. Bàsicament es tracta de convertir cada entitat del model ER en una relació, segons la terminologia del model relacional.

La especialització/generalització dona lloc a quatre relacions (superclasse i les tres subclasses) on les subclasses tenen com a clau primària la clau de l'entitat superior.

Per a les relacions 1:N entre entitats del model ER he afegit a l'entitat del cantó N una clau forana que referencii la clau primària de la entitat del cantó 1.

I finalment la entitat associativa Reserva la transformo en una relació amb els seus atributs propis i que com a clau primària té les dues claus primàries de la relació casa i de la relació client.

Així es com queda cada relació amb els seus atributs i claus. La clau primària de cada relació està subratllada.

PERSONA (codi-persona, nom, cognoms, dni, adreca, ciutat, codi-postal, telefon)

PROPIETARI (codi-persona, e-mail, fax)

on (codi-persona) referencia PERSONA.

CLIENT (codi-persona, estat-civil, freqüència)

on (codi-persona) referencia PERSONA.

TARGETA (numero, caducitat, tipus, client)

on (client) referencia CLIENT(codi-persona).

ADMINISTRATIU(codi-persona, antiguitat)

on (codi-persona) referencia PERSONA.

CASA (codi-casa, nom, preu-nit, places, plantes, categoria, h-dobles, h-simples, cuina, rentadora, televisio, calefaccio, poble)

on (poble) referencia POBLE(codi-poble).

POBLE (codi-poble, nom, poblacio, comarca, descripcio)

RESERVA(codi-casa, codi-persona, entrada, sortida, preu-reserva, persones, factura)

on (codi-casa) referencia CASA (codi-casa)

on (codi-persona) referencia CLIENT (codi-persona)

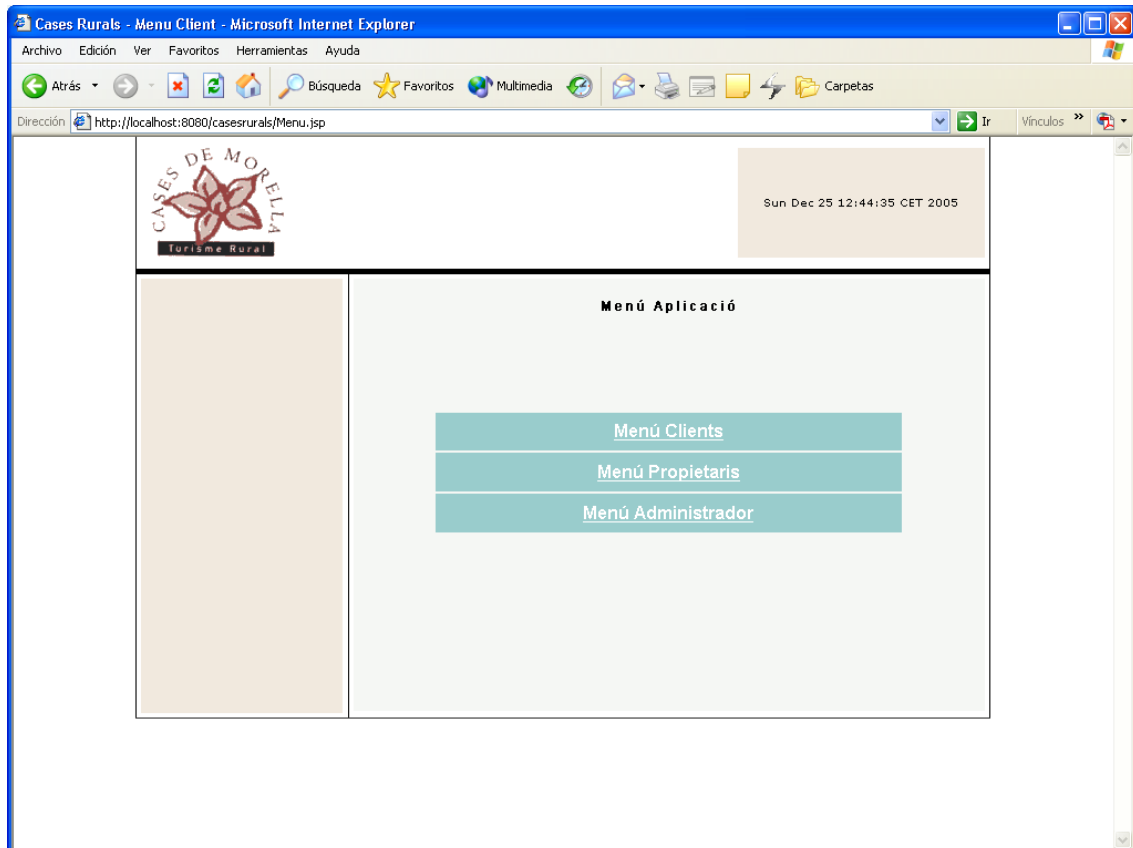
on (factura) referencia FACTURA (codi-factura).

FACTURA(codi-factura, data, base-imposable, iva, total, descripcio)

Disseny de la Interfície d'Usuari

Les captures de pantalla que es mostren a continuació corresponen ja al projecte real, per la qual cosa quasi es pot considerar com un manual d'usuari.

Quan entrem a l'adreça <http://localhost:8080/casesrurals/Menu.jsp> obtenim aquesta pantalla de Menú que ens permetrà navegar per tota l'aplicació. Aquí podem entrar en cadascun dels subsistemes: Client, Propietari, Administrador.



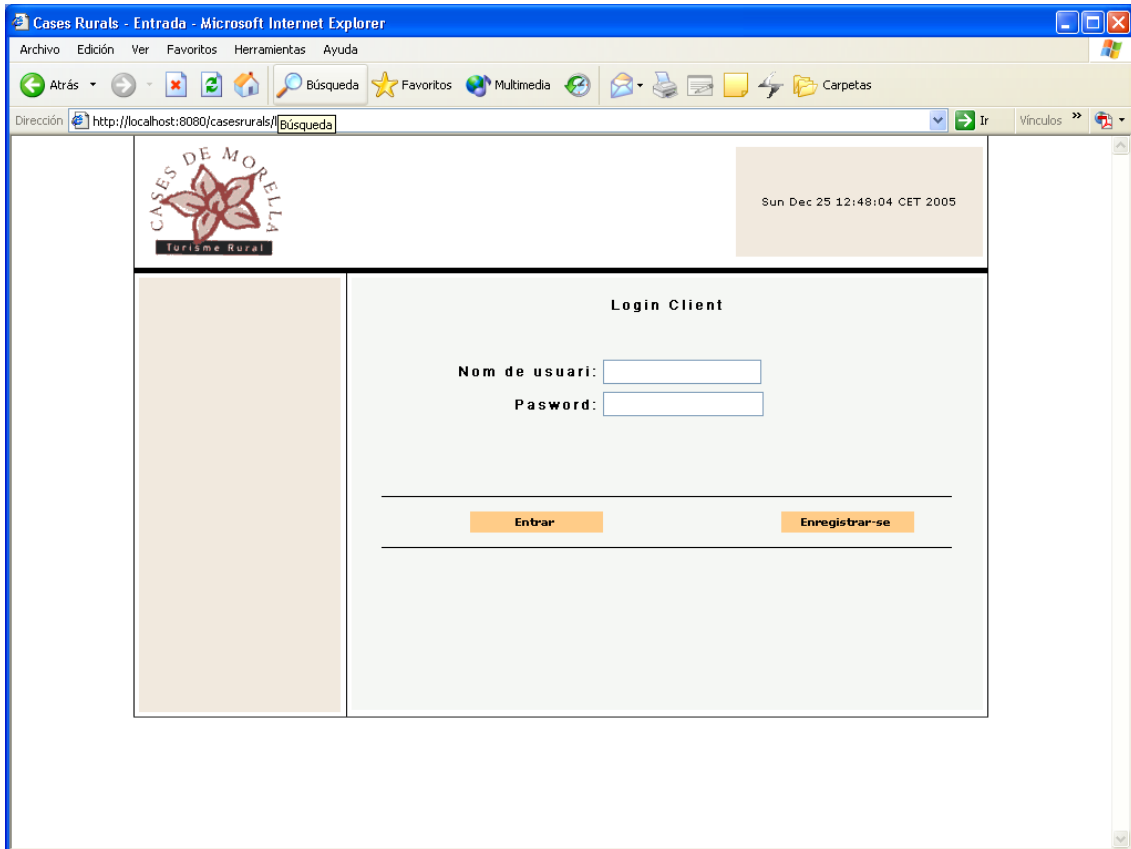
Il·lustració 6. Menú Aplicació

Quan entrem en qualsevol dels tres menús ens apareix una pantalla de log on hem de posar un nom d'usuari i una contrasenya (Il·lustració 7).

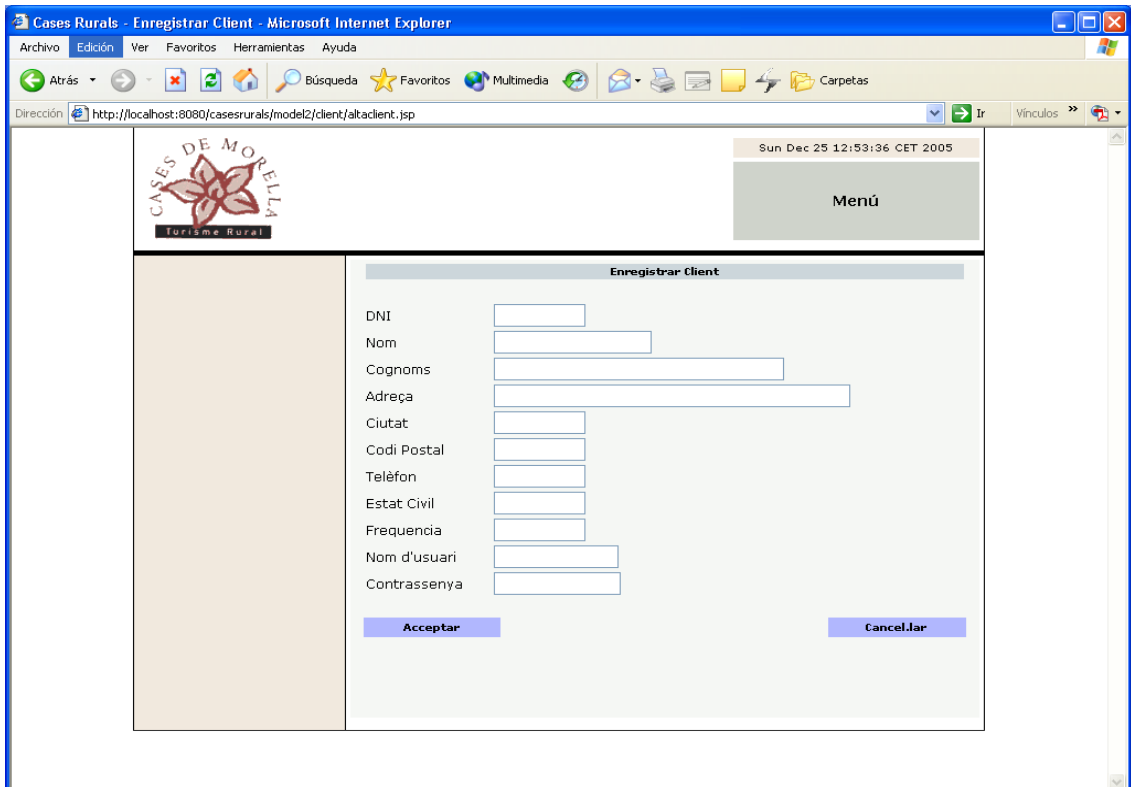
En cas de no tenir nom d'usuari i contrasenya les pantalles de "login client" i "login propietari" ens donen la possibilitat de registrar-nos per obtenir un nom d'usuari i contrasenya i poder accedir d'aquesta forma a l'aplicació. No passa el mateix amb la pantalla "login administrador", aquesta no té botó per registrar un nou usuari perquè per motius de seguretat tots els administratius entre amb el mateix nom d'usuari i contrasenya.

La pantalla per registrar-se ens demana una sèrie de dades, entre elles nom d'usuari i contrasenya (Il·lustració 8).

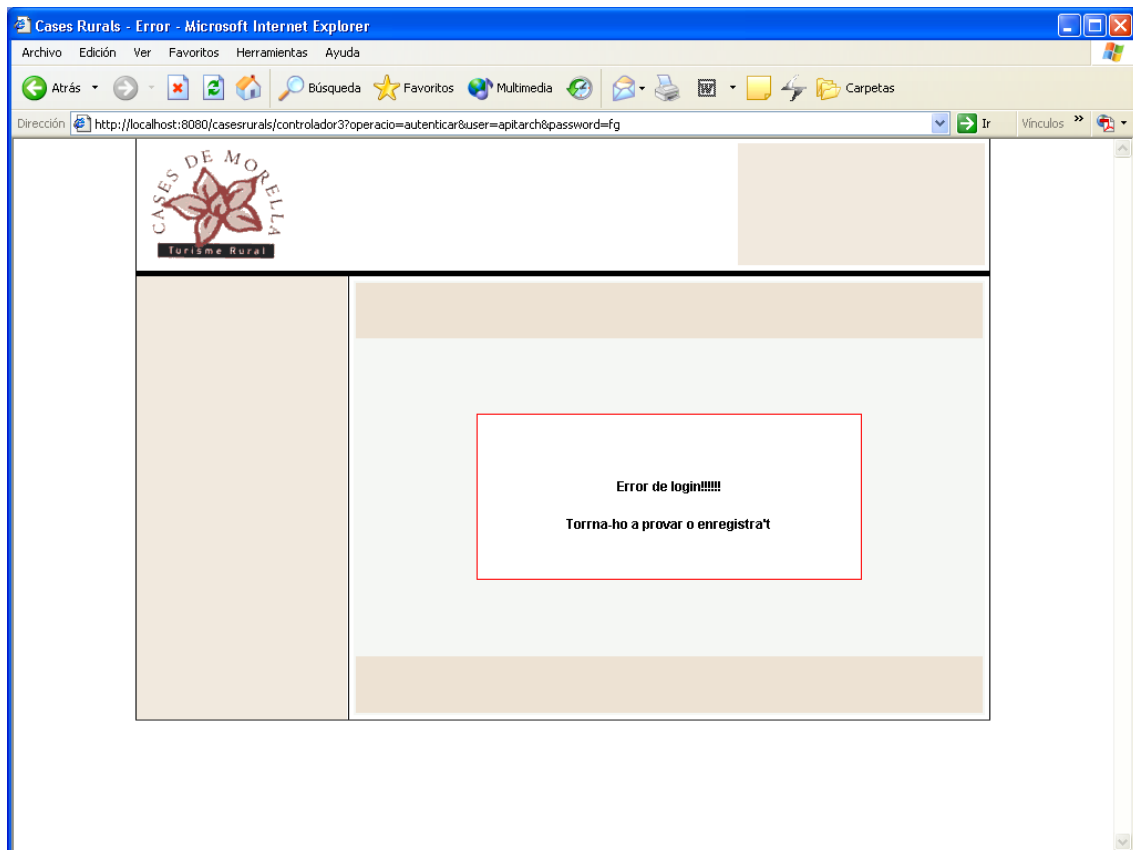
Si en fer el log el nom d'usuari - contrasenya no són correctes o si en registrar-nos el nom d'usuari - contrasenya ja estan ocupats ens apareixen pantalles d'error indicant cada situació (Il·lustració 9, 10).



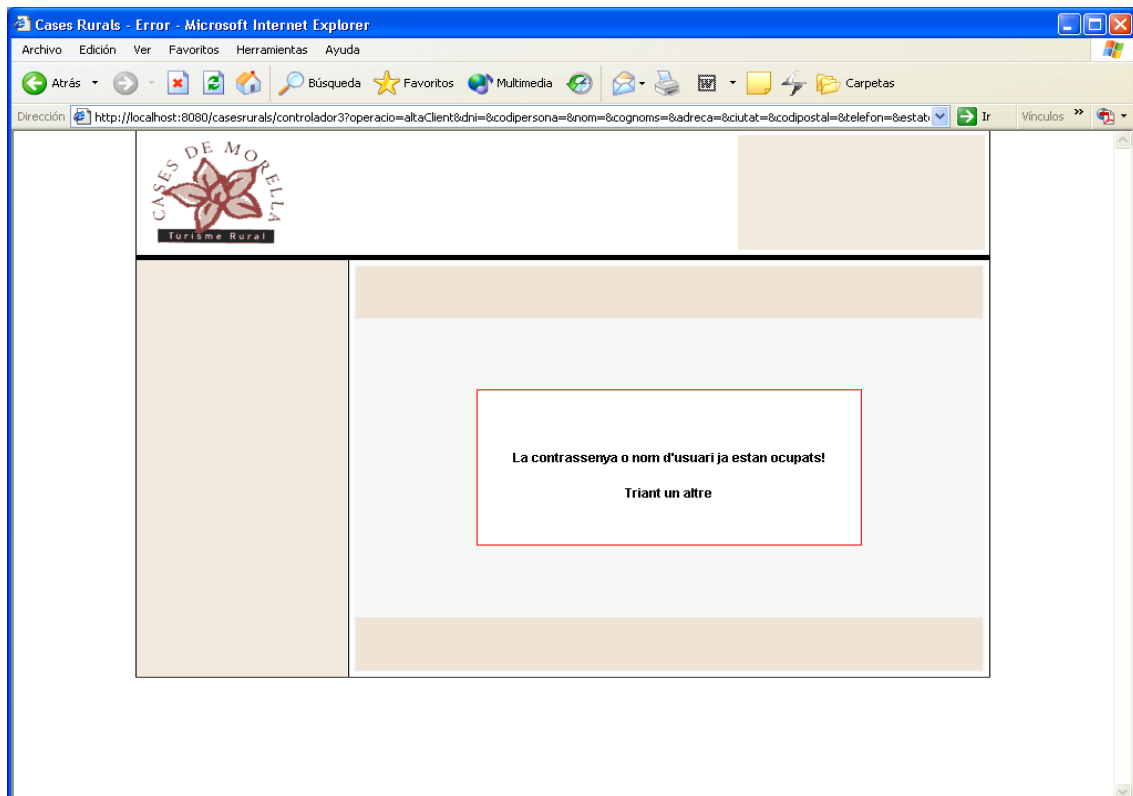
Il·lustració 7. Pantalla de log del Client



Il·lustració 8. Pantalla per enregistrar-se.



Il·lustració 9. Error de login



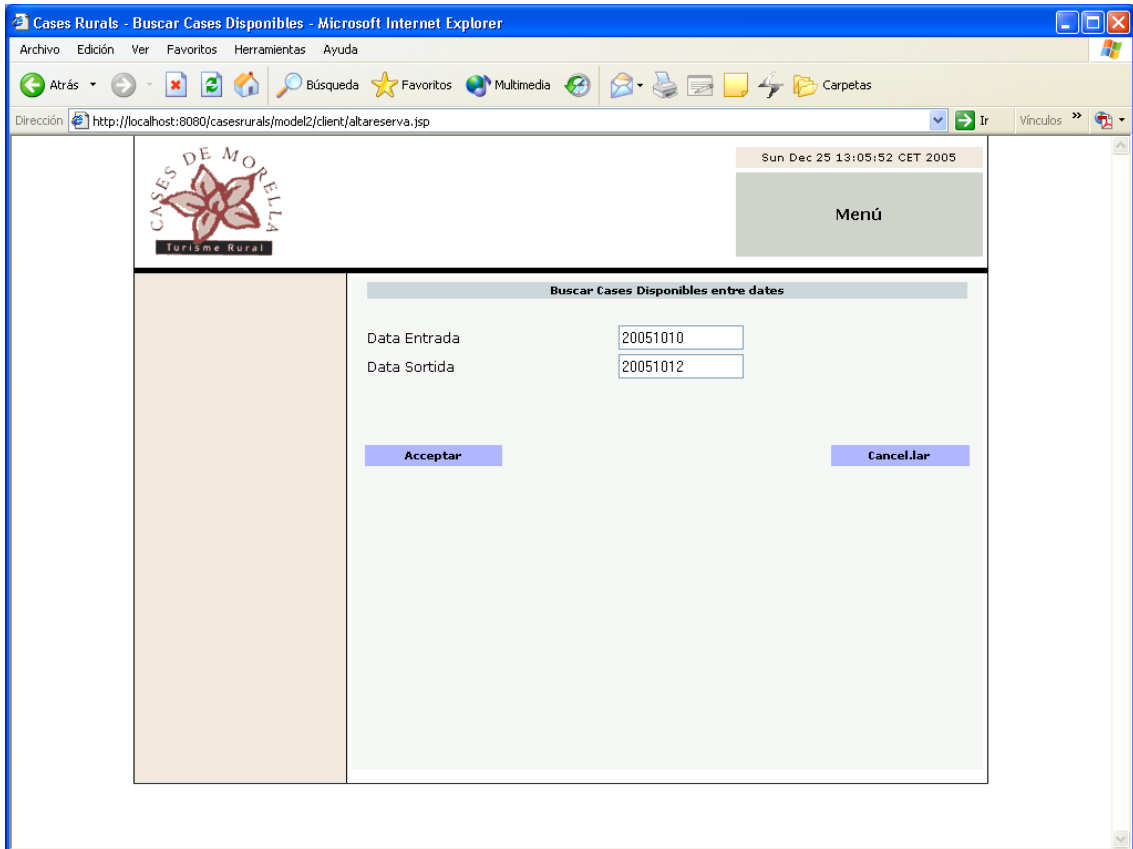
Il·lustració 10. Contrasenya o nom d'usuari ocupats.

Si ens validem correctament accedirem a un menú que inicialment ens permet Fer una reserva o Obtenir un llistat de les reserves que el client té fetes, en el cas del subsistema Client.

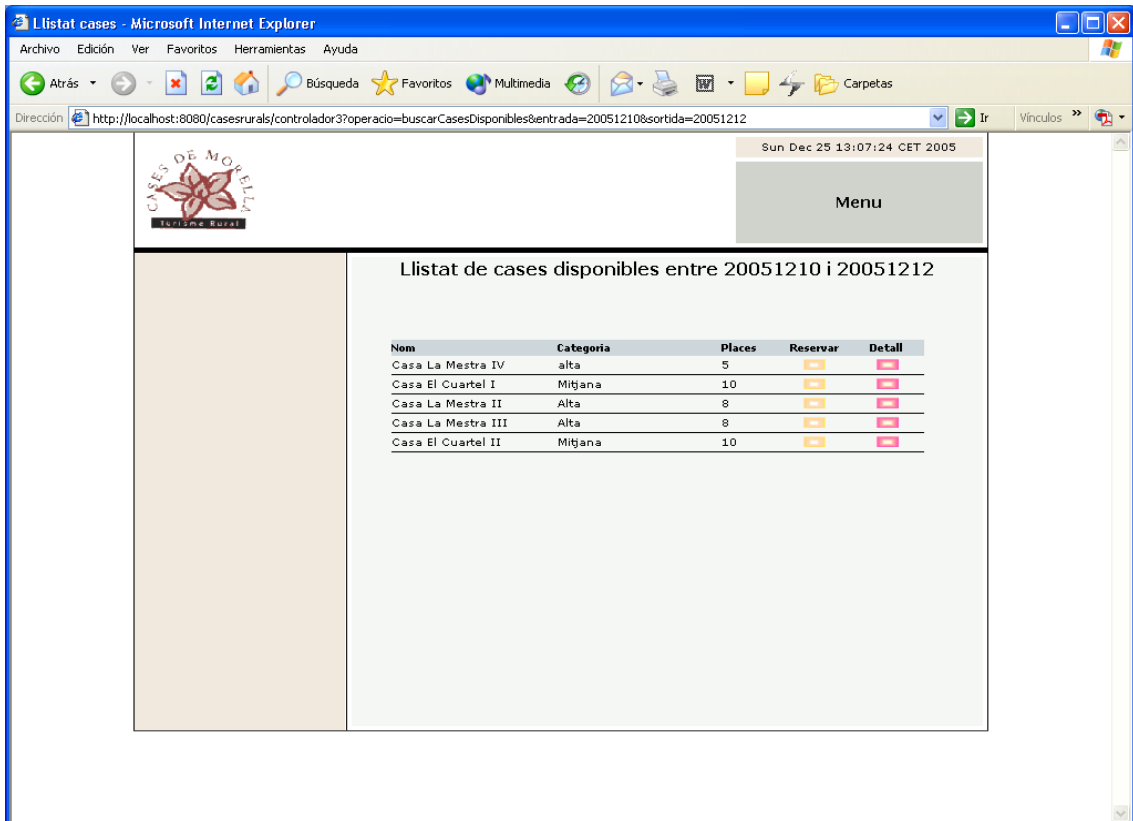
En el primer cas, **Fer Reserva**, accedim a una pantalla on triem una data d'entrada i una de sortida¹ per la reserva (Il·lustració 11), i en fer acceptar se'ns mostra un llistat de les cases que estan disponibles (Il·lustració 12), és a dir, que no estan reservades entre aquestes dues dates. Des d'aquest llistat de cases disponibles podem fer la reserva o obtenir més informació de la casa que triem (Il·lustració 13). Si triem fer la reserva se'ns mostrarà una nova pantalla on haurem d'emplenar la resta de camps de la reserva que volem fer i on podrem veure ja el preu de la reserva, i en fer clic sobre Acceptar ja tindrem introduïda la reserva al carret de la compra (Il·lustració 14). Ara tindrem la possibilitat de afegir més reserves al carret o de Finalitzar la compra de la reserva (Il·lustració 15). Si fem Finalitzar ens apareixerà una pantalla on se'ns mostrarà el llistat de totes les reserves que tenim afegides al carret de la compra en la sessió actual amb un botó que ens permetrà treure qualsevol de les reserves del carret abans de fer la compra definitiva. I a la part de baix de la pantalla ens apareixerà la factura amb la base imposable, IVA i total de totes les reserves del carret, i un altre apartat on haurem d'introduir les dades de la targeta de crèdit. En clicar sobre Fer Pagament les dades de la reserva s'introduiran a la base de dades de l'aplicació (Il·lustració 16).

En el segon cas, **Llistat Reserves**, podrem veure un llistat de les reserves que té fetes el client que entra. Des d'aquest llistat tindrem també la possibilitat d'esborrar qualsevol de les reserves (Il·lustració 17).

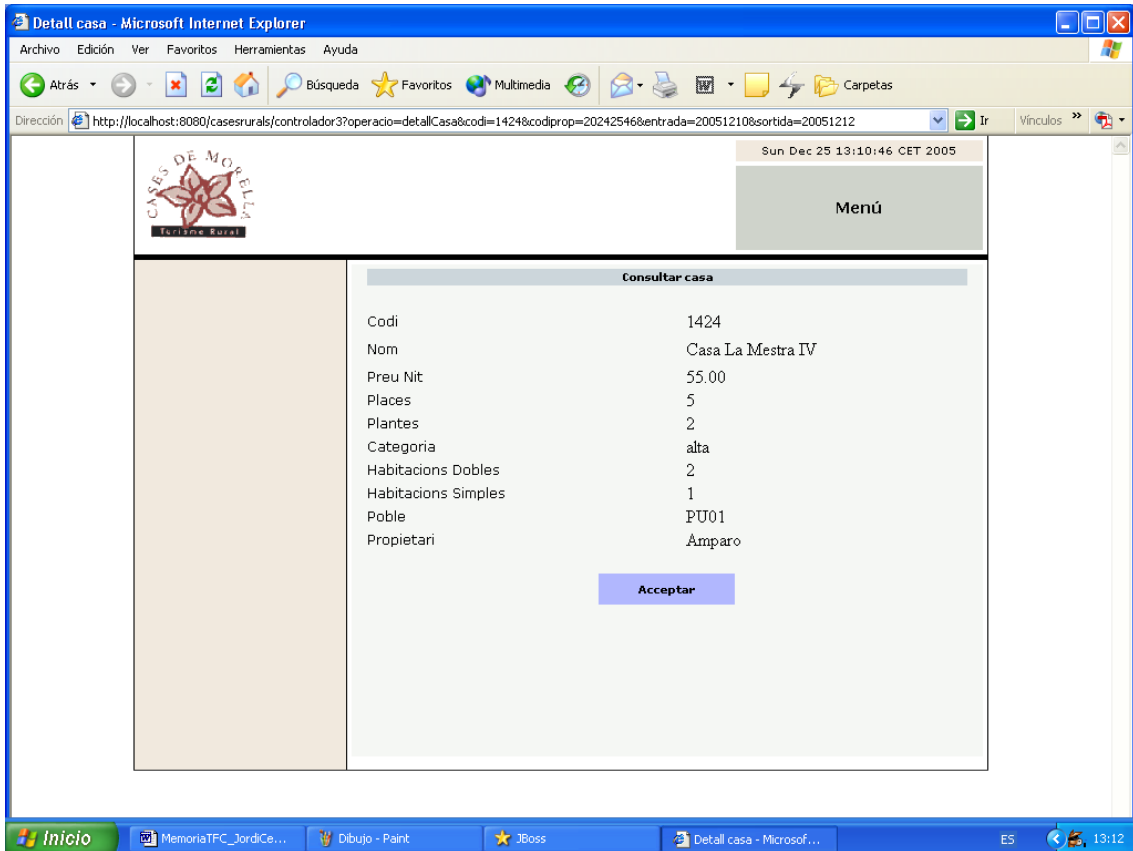
¹ La introducció de totes les dates de l'aplicació es fa amb el format 20051208, on 2005 es l'any, 12 és el mes de desembre i 08 és el dia 8.



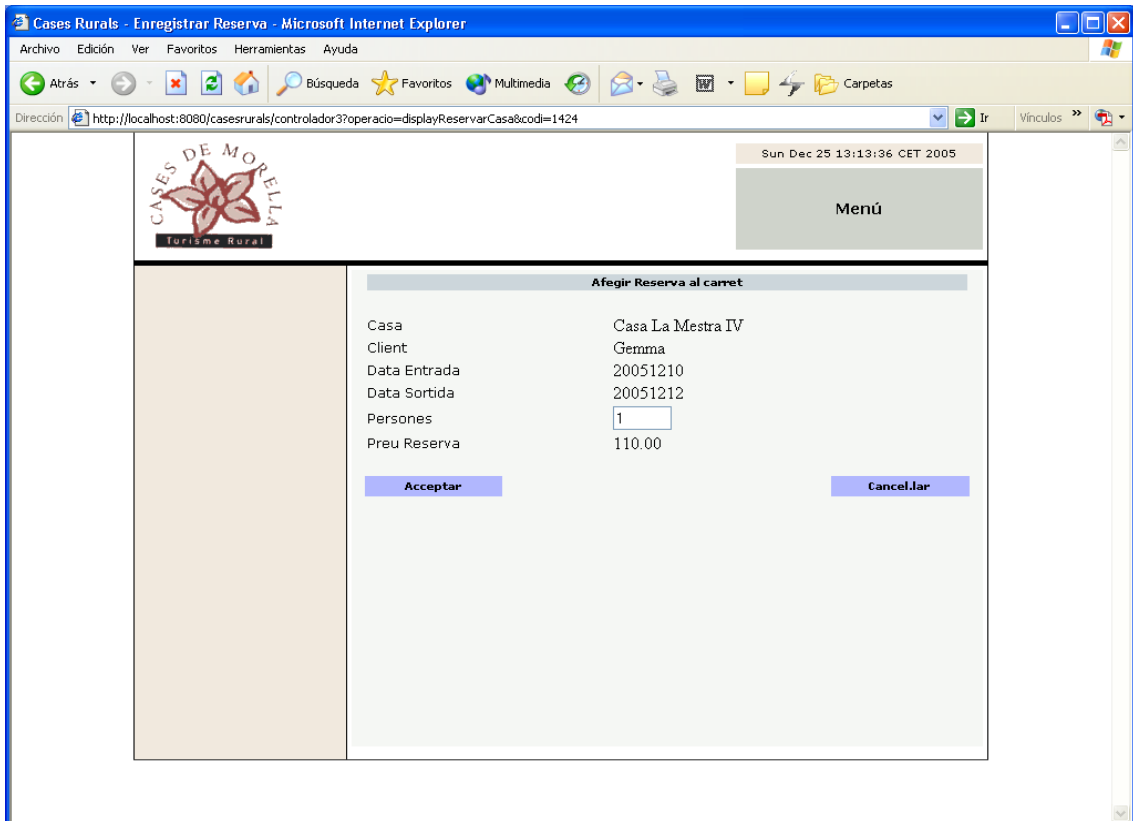
II-il·lustració 11. Fer Reserva. Buscar cases disponibles entre dates.



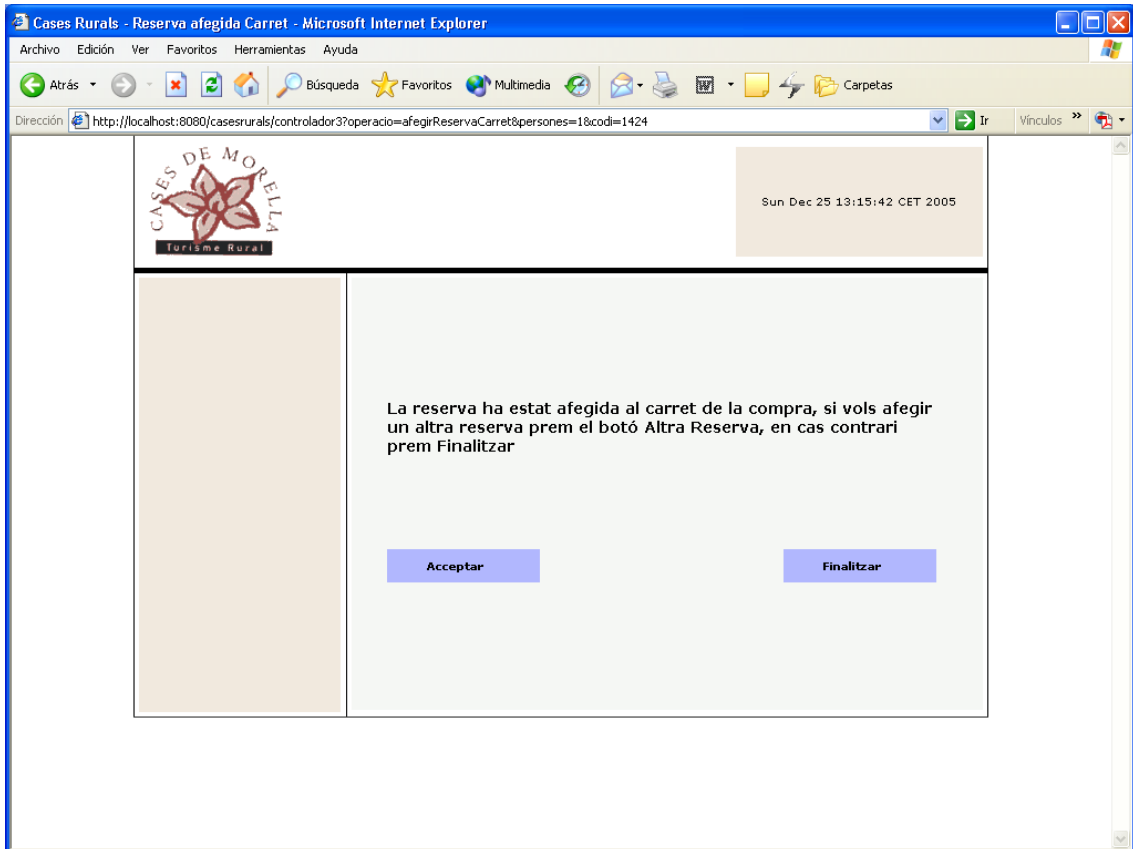
II-il·lustració 12. Llistat de cases disponibles per reservar entre dues dates.



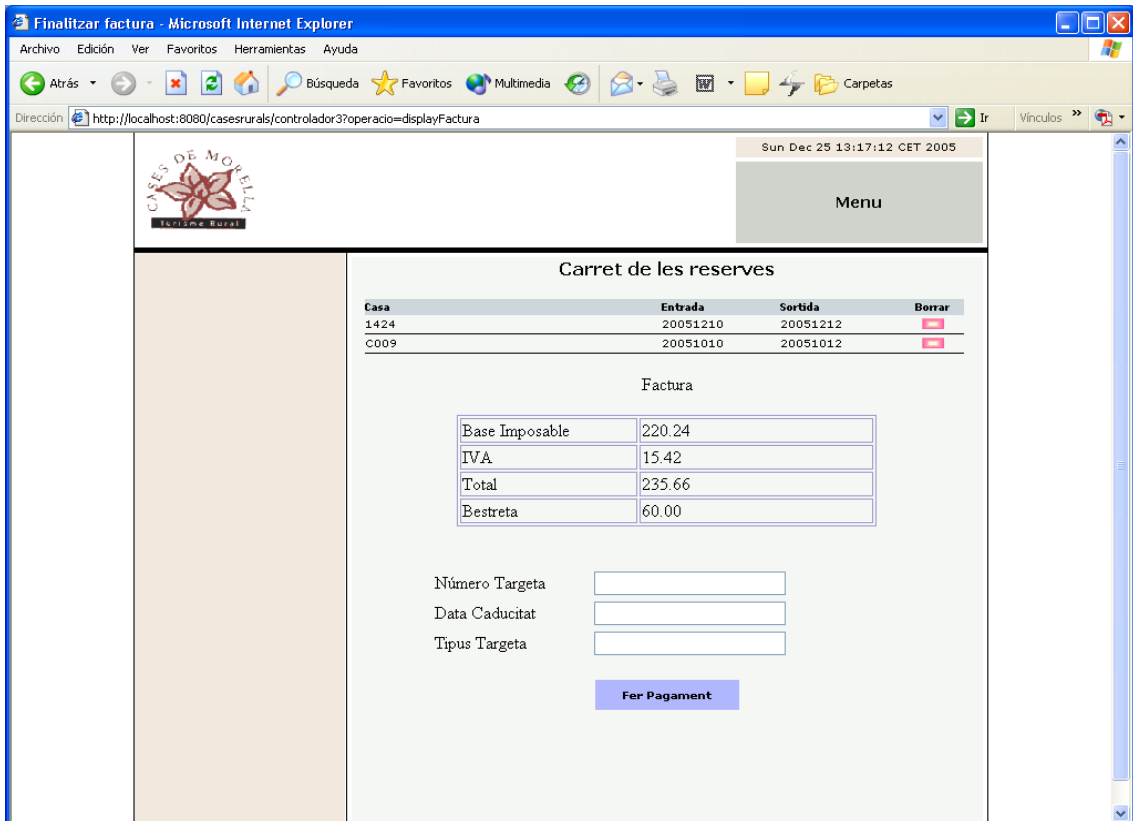
II-Il·lustració 13. Detall de la casa.



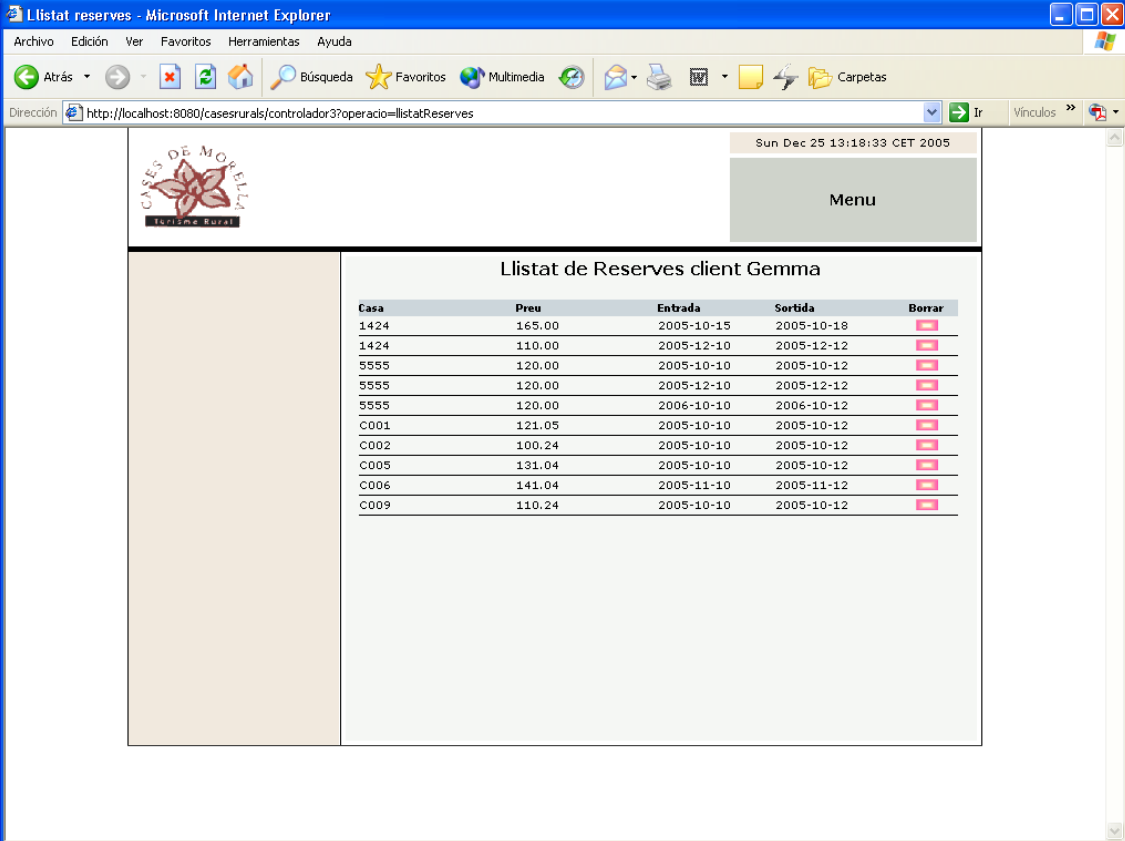
II-Il·lustració 14. Pantalla per afegir la reserva al carret.



Il·lustració 15. Pantalla de reserva afegida al carret.



Il·lustració 16. Carret reserves, factura i pagament



Casa	Preu	Entrada	Sortida	Borrar
1424	165.00	2005-10-15	2005-10-18	<input type="button" value="✖"/>
1424	110.00	2005-12-10	2005-12-12	<input type="button" value="✖"/>
5555	120.00	2005-10-10	2005-10-12	<input type="button" value="✖"/>
5555	120.00	2005-12-10	2005-12-12	<input type="button" value="✖"/>
5555	120.00	2006-10-10	2006-10-12	<input type="button" value="✖"/>
C001	121.05	2005-10-10	2005-10-12	<input type="button" value="✖"/>
C002	100.24	2005-10-10	2005-10-12	<input type="button" value="✖"/>
C005	131.04	2005-10-10	2005-10-12	<input type="button" value="✖"/>
C006	141.04	2005-11-10	2005-11-12	<input type="button" value="✖"/>
C009	110.24	2005-10-10	2005-10-12	<input type="button" value="✖"/>

Il·lustració 17. Llistat reserves d'un client.

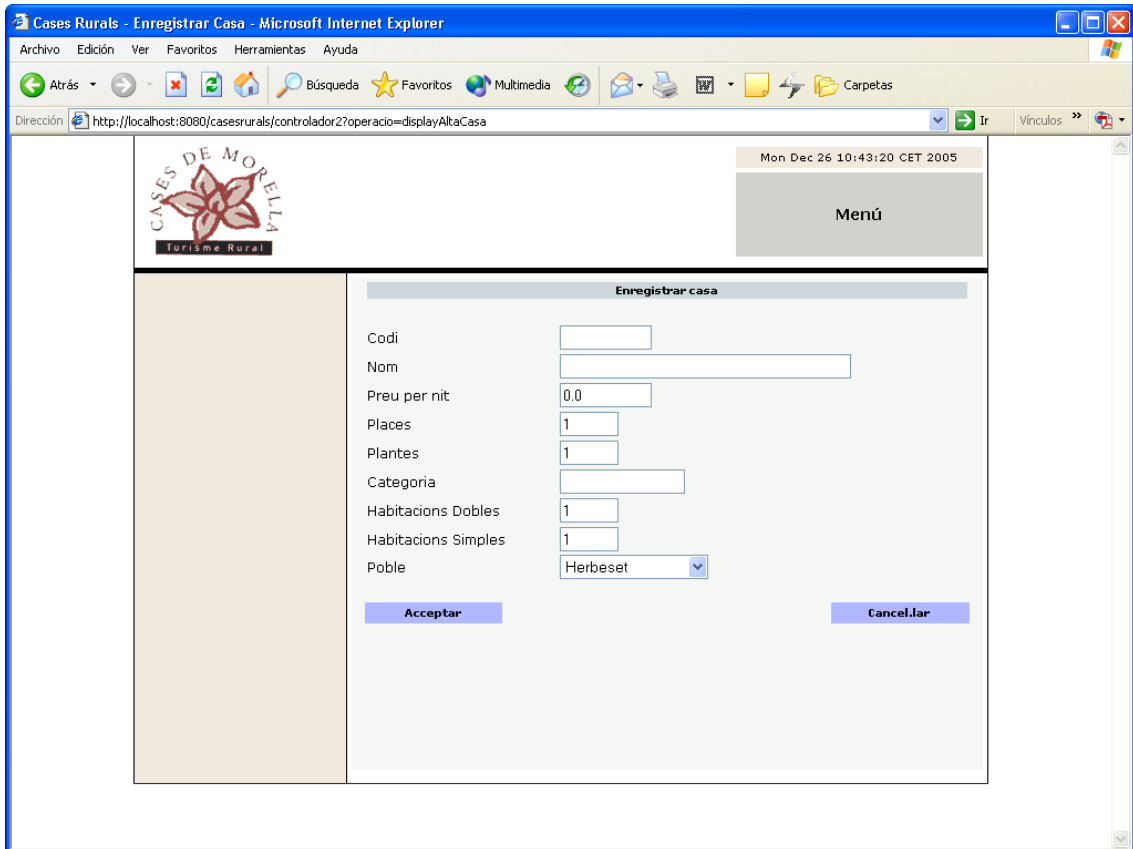
Per al subsistema Propietari, la validació i enregistrament de nous propietaris és exactament igual que com hem explicat al principi.

Quan el propietari es valida correctament podrem fer l'alta d'una casa, obtenir un llistat de cases o obtenir un llistat de les reserves.

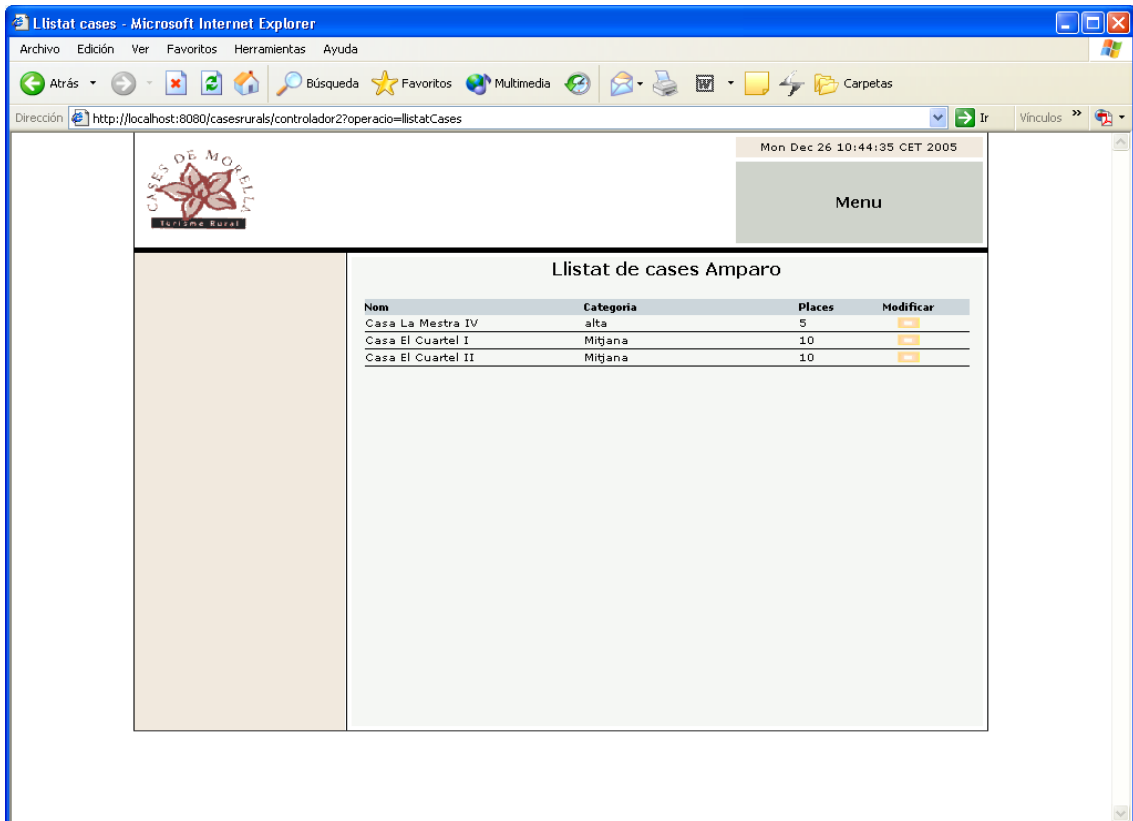
Si fem clic sobre Alta Casa, accedirem a una pantalla on haurem de introduir totes les dades de la casa que se'ns demanen i en fer Acceptar la casa serà enregistrada a la Base de dades (Il·lustració 18).

Si fem clic sobre Llistat Cases, a banda de mostrar una llista de les cases del propietari (Il·lustració 19), també podrem modificar qualsevol de les cases mostrades, accedint a la pantalla de modificació de la casa (Il·lustració 21).

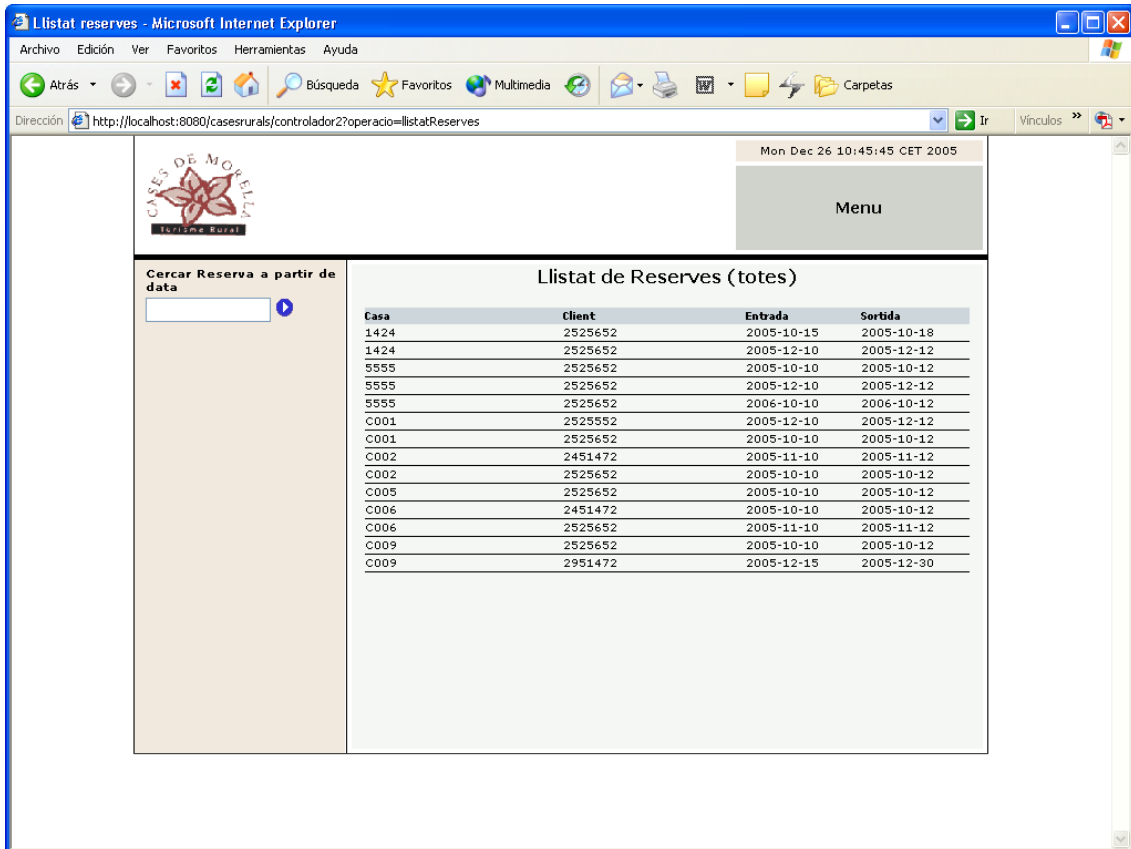
Si fem clic sobre Llistat Reserves obtindrem un llistat de totes les reserves i amb el camp Cercar Reserva a partir de data podrem obtenir un llistat de les reserves fetes a partir de la data introduïda (Il·lustració 20).



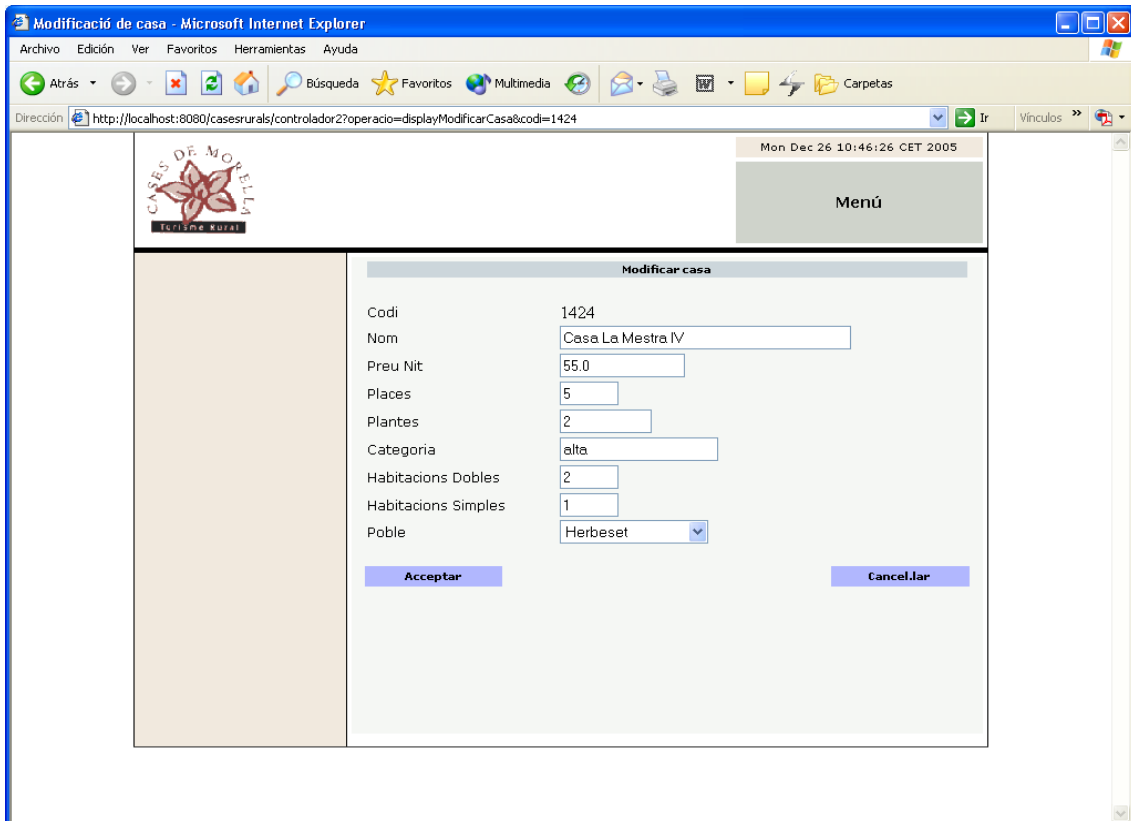
II-l·lustració 18. Enregistrar nova casa.



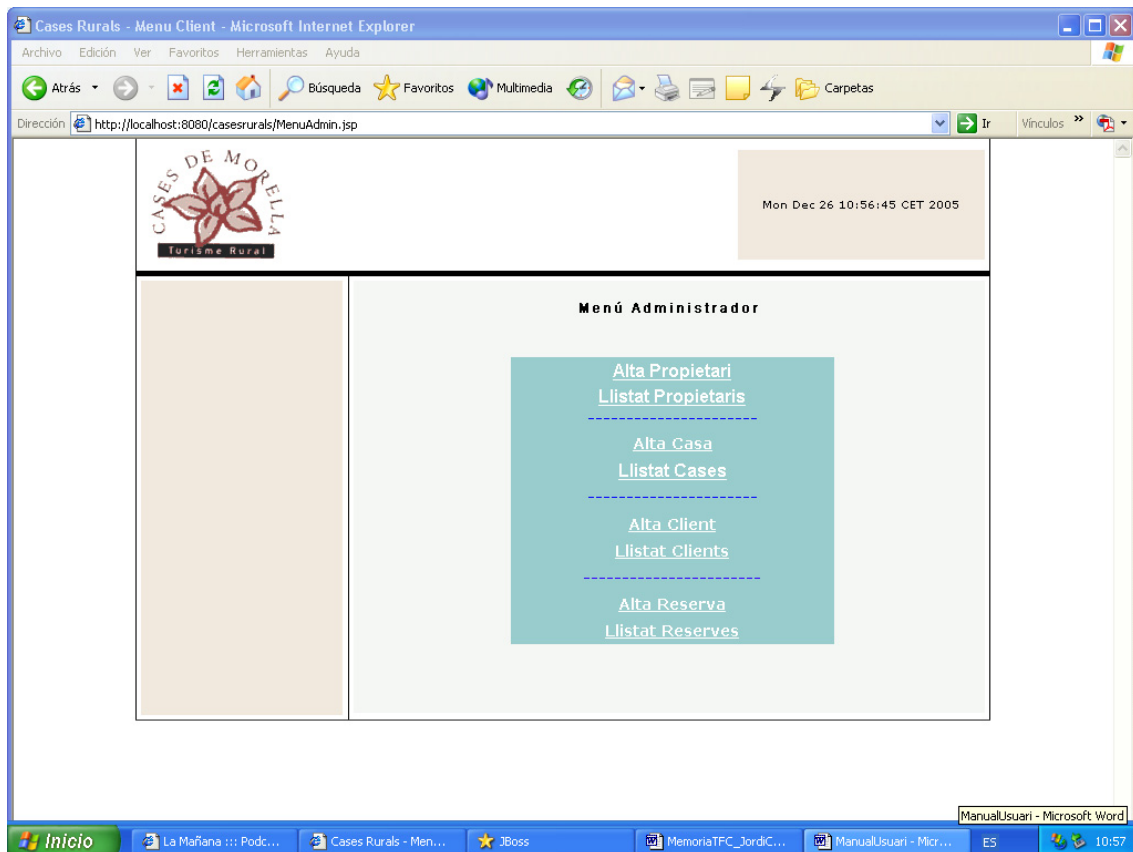
II-l·lustració 19. Llistat de cases d'un propietari.



II-il·lustració 20. Llistat de les reserves



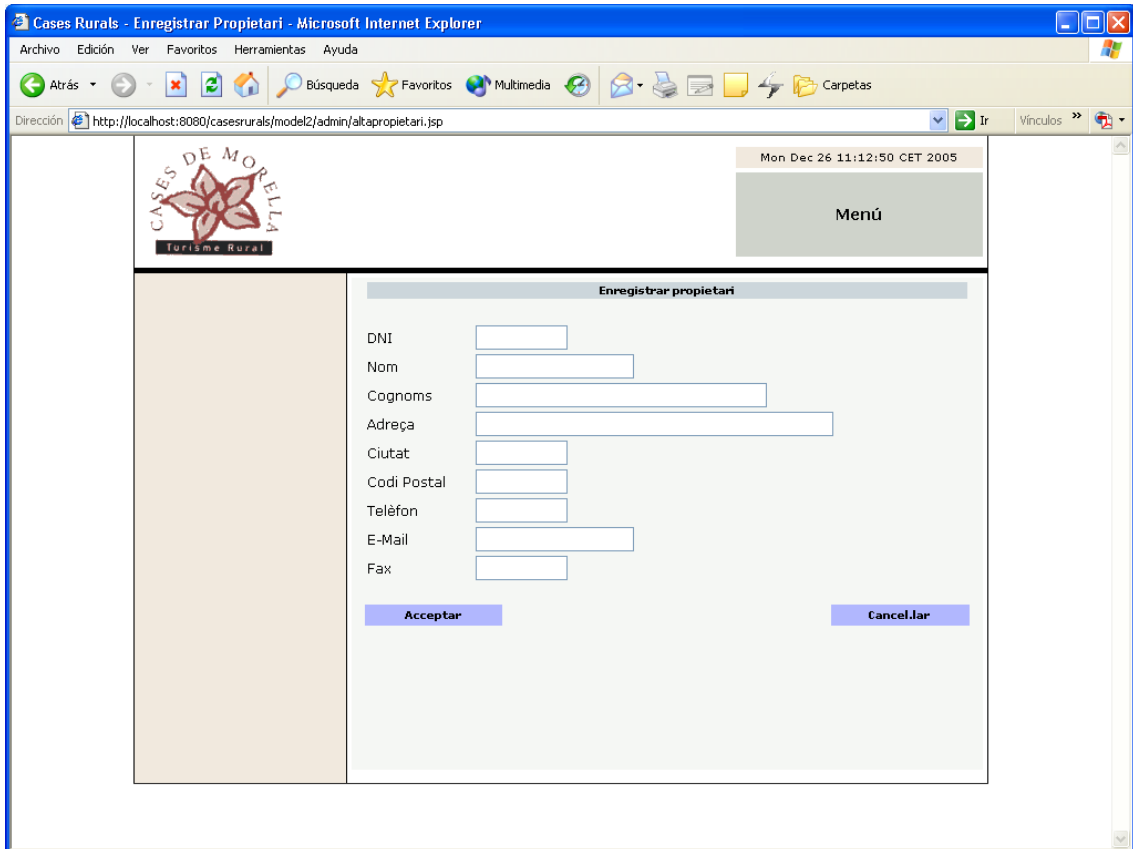
II-il·lustració 21. Pantalla de modificació d'una casa.



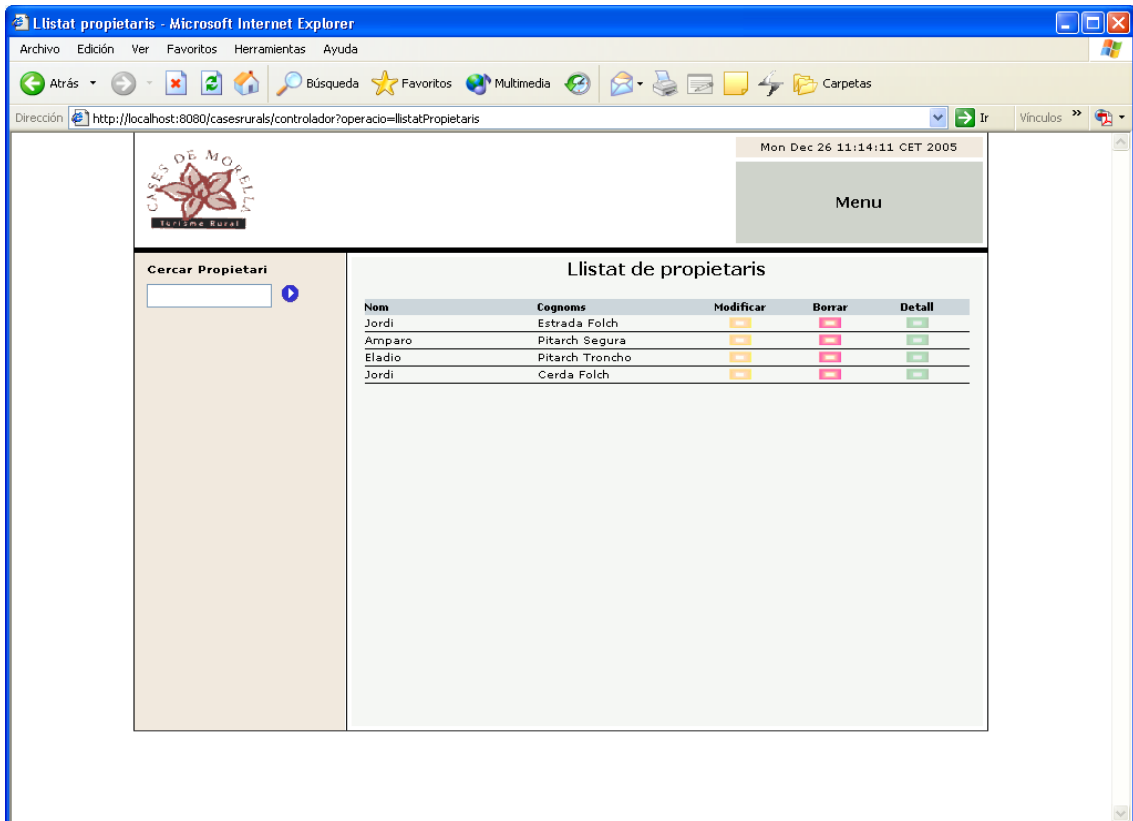
II·lustració 22. Menú Administrador.

Pel que fa al subsistema administratiu, per entrar ho haurem de fer amb **user: admin i password: admin**. Si intentem entrar directament a una altra pàgina d'aquest subsistema se'ns mostrarà la pàgina de validació i no ens deixarà passar fins que no posem les paraules correctes. Si posem un altre user i/o password se'ns mostrarà una pantalla indicant que no és correcte.

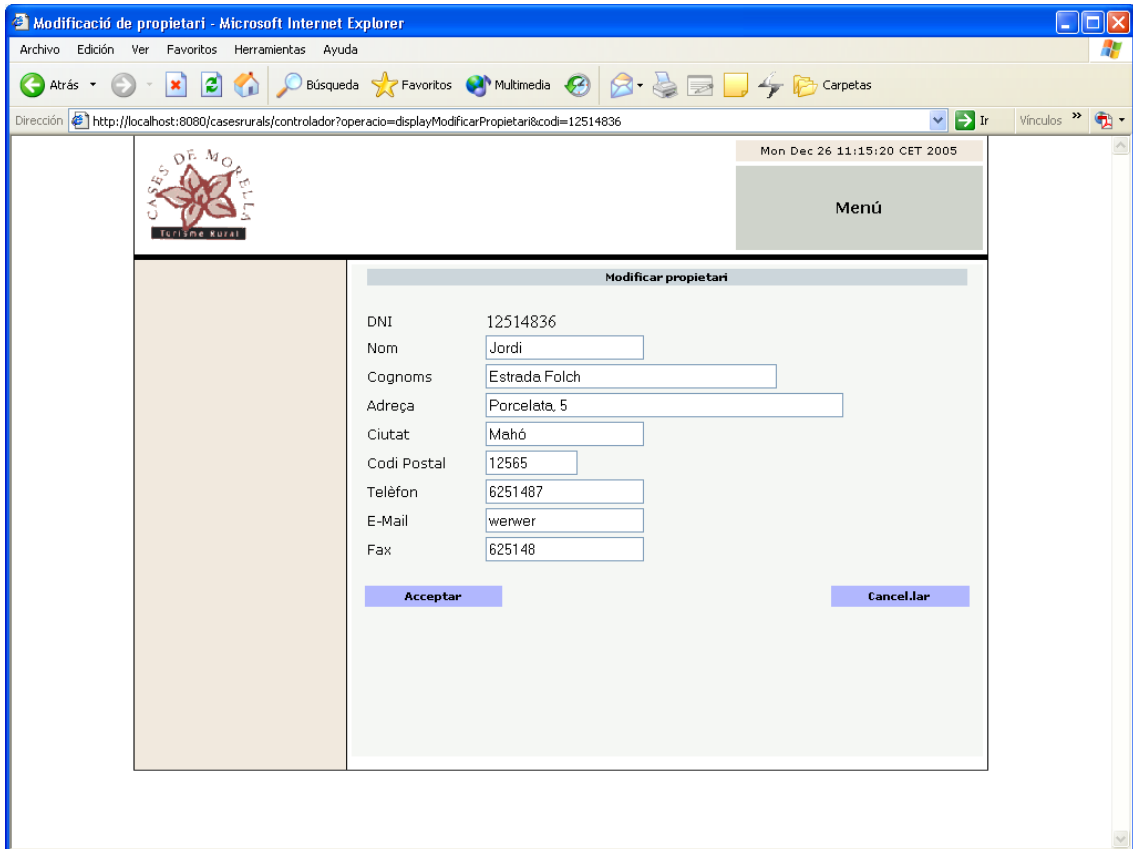
Si ens validem correctament, accedirem, a un menú (II·lustració 22) on podrem donar d'alta nous propietaris, clients, cases o reserves o obtenir un llistat de propietaris, clients, cases o reserves. Clicant sobre les opcions de "Llistat ..." a banda d'obtenir el llistat corresponent també se'ns donarà la possibilitat de modificar, esborrar o obtenir més informació de cadascun dels registres llistats en cada cas. En cadascuna d'aquestes pantalles de llistats també podrem fer cerques específiques per cada entitat. En el cas de les reserves podrem obtenir un llistat de les reserves a partir de la data introduïda.



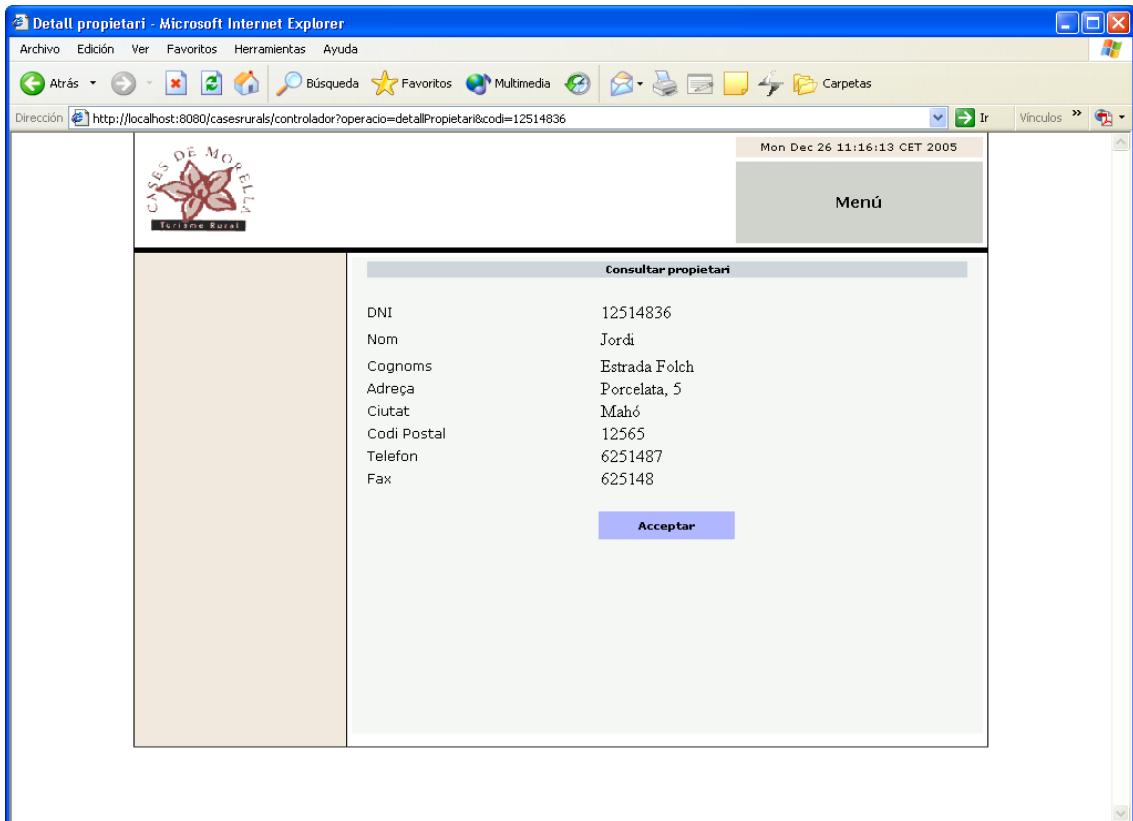
II-il·lustració 23. Alta nou propietari.



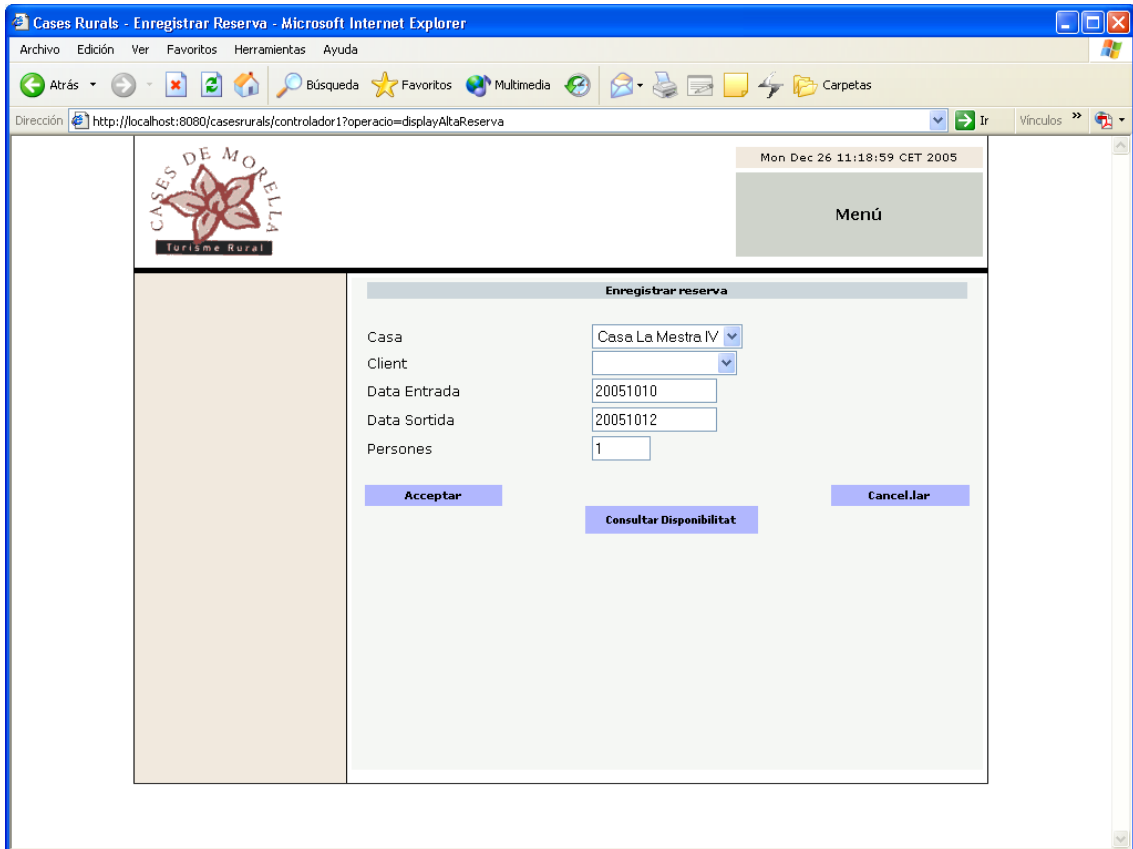
II-il·lustració 24. Llistat propietaris.



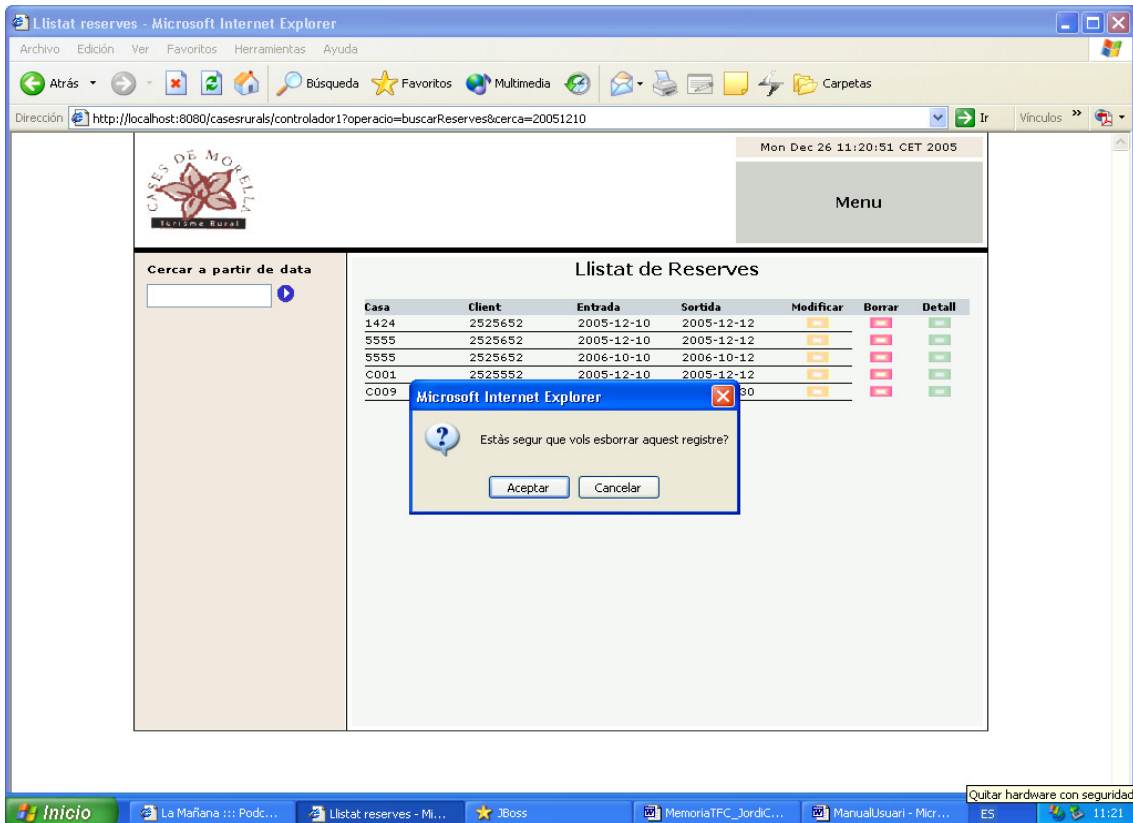
II-Il·lustració 25. Modificació propietari.



II-Il·lustració 26. Ampliació de la informació sobre el propietari.



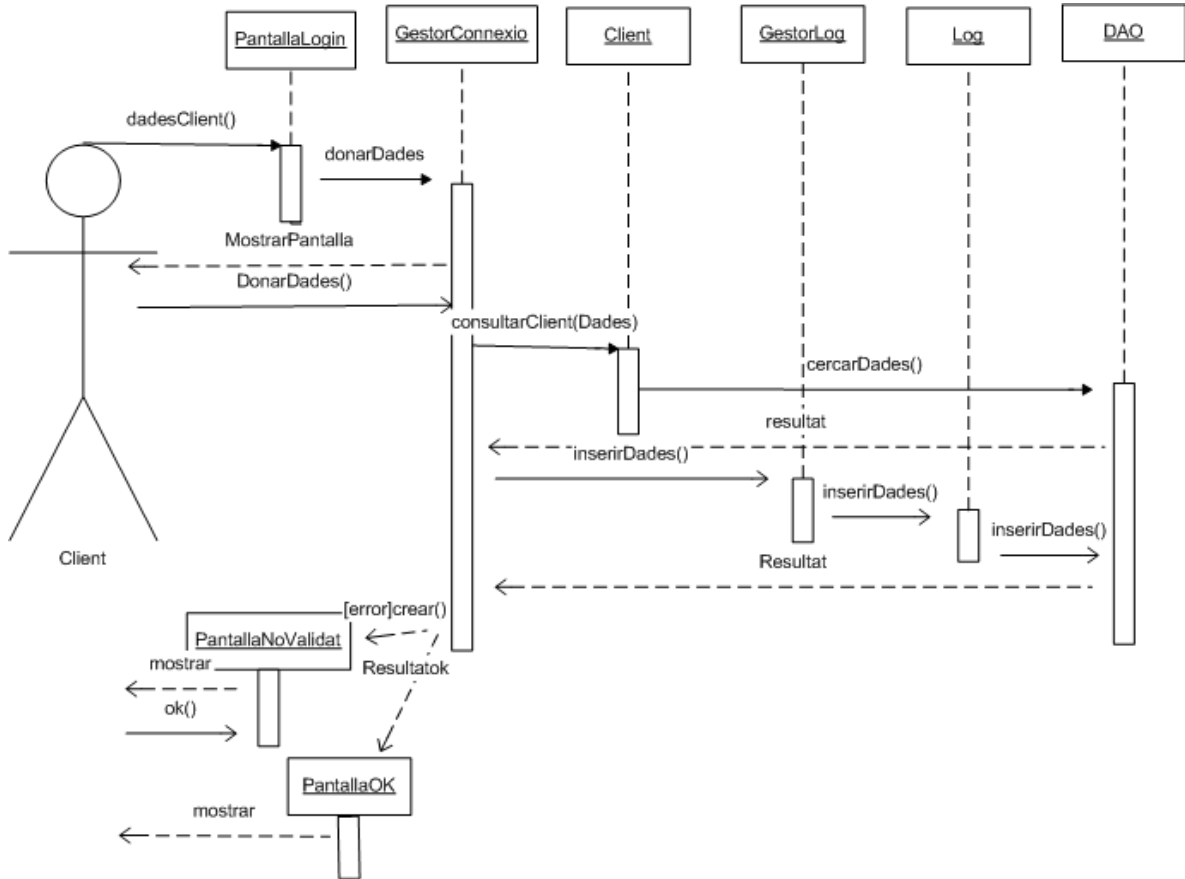
II-il·lustració 27. Alta de nova reserva.



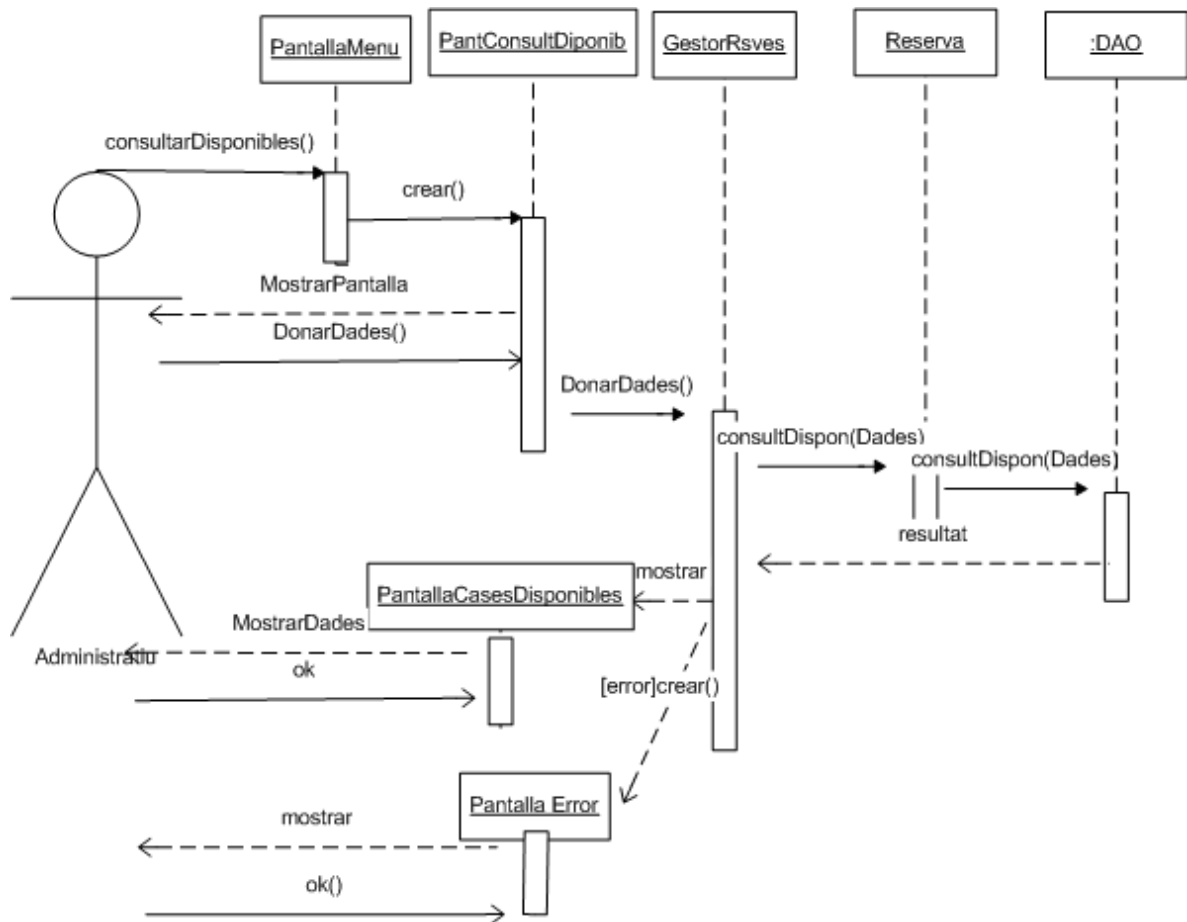
II-il·lustració 28. Esborrat d'una reserva.

Diagrames de seqüència dels casos d'ús

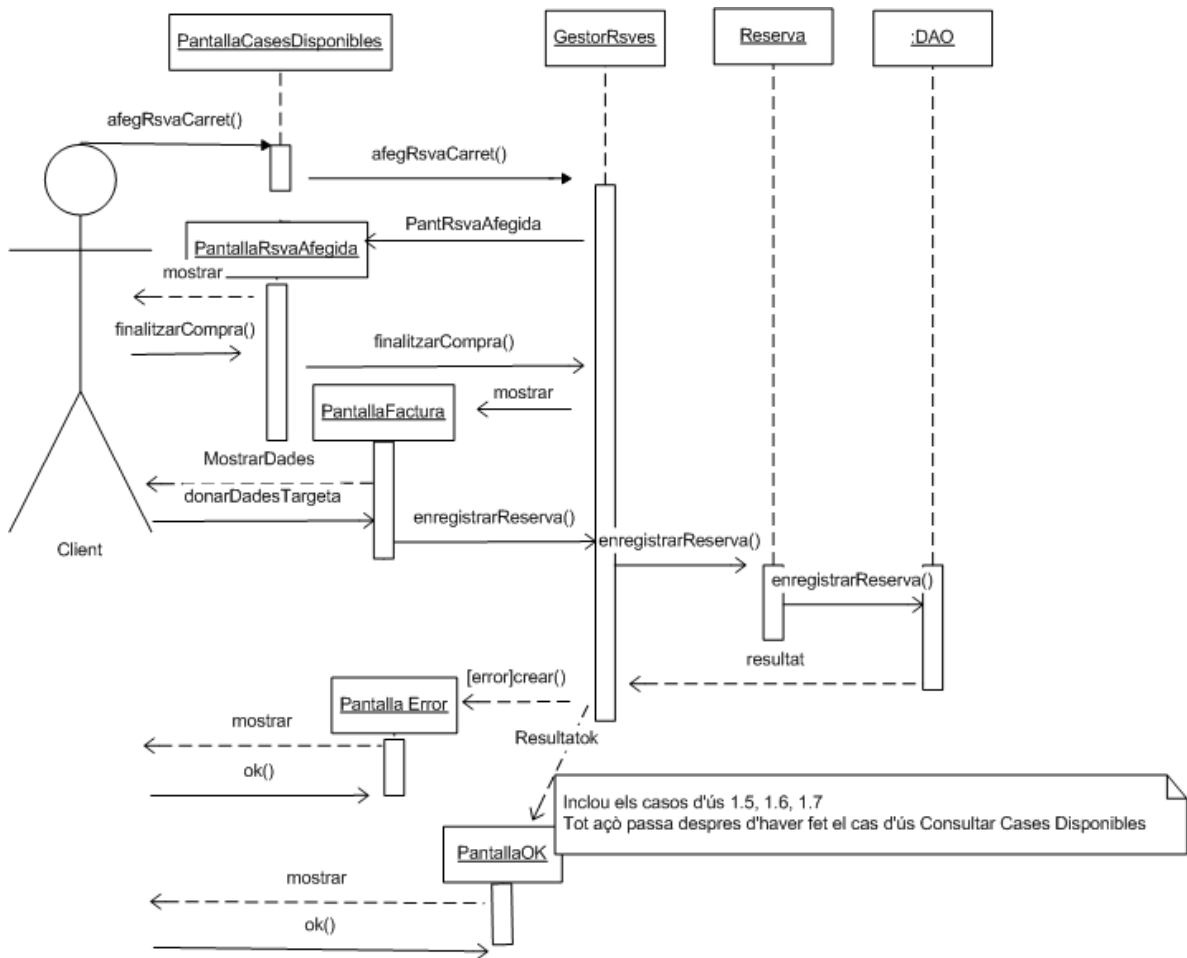
Cas d'ús "Accedir com a client al subsistema"



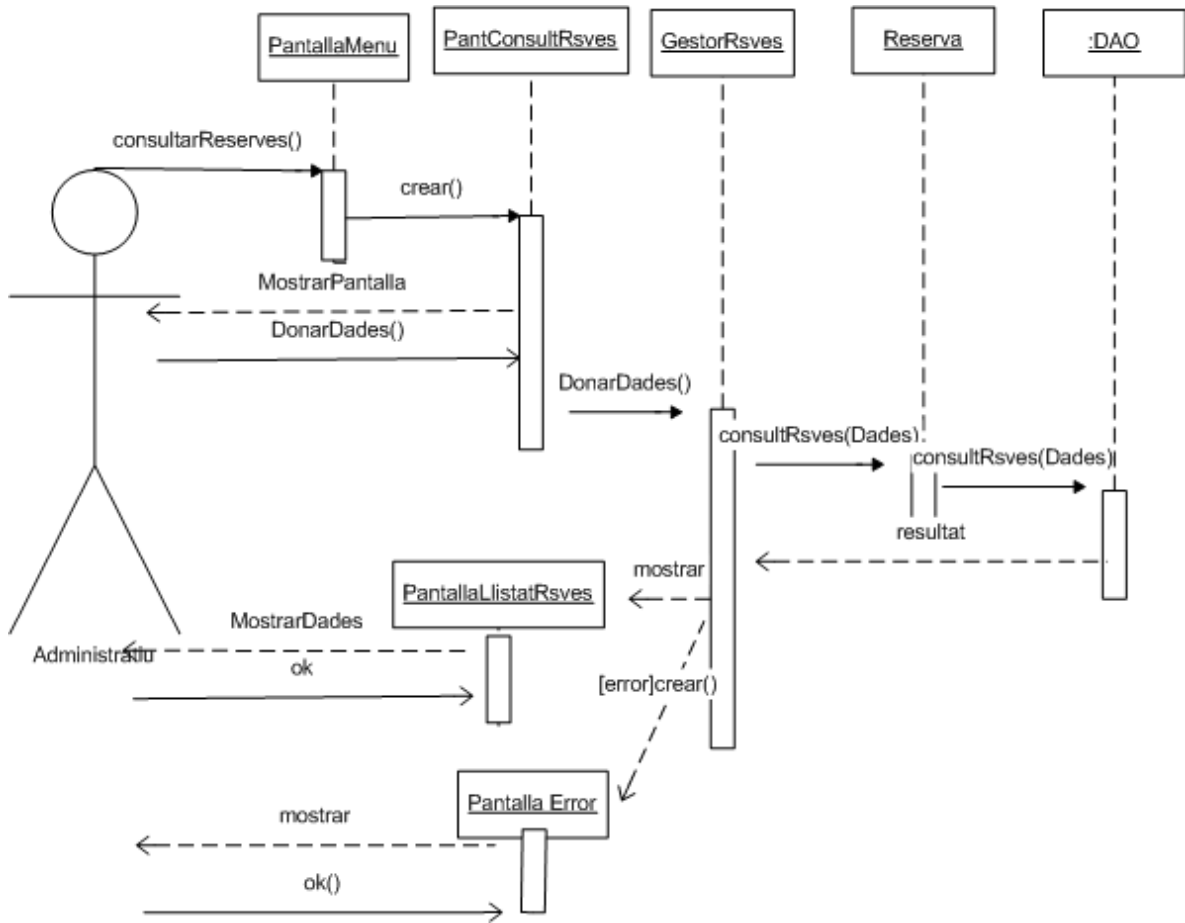
Cas d'ús "Consultar Cases Disponibles"



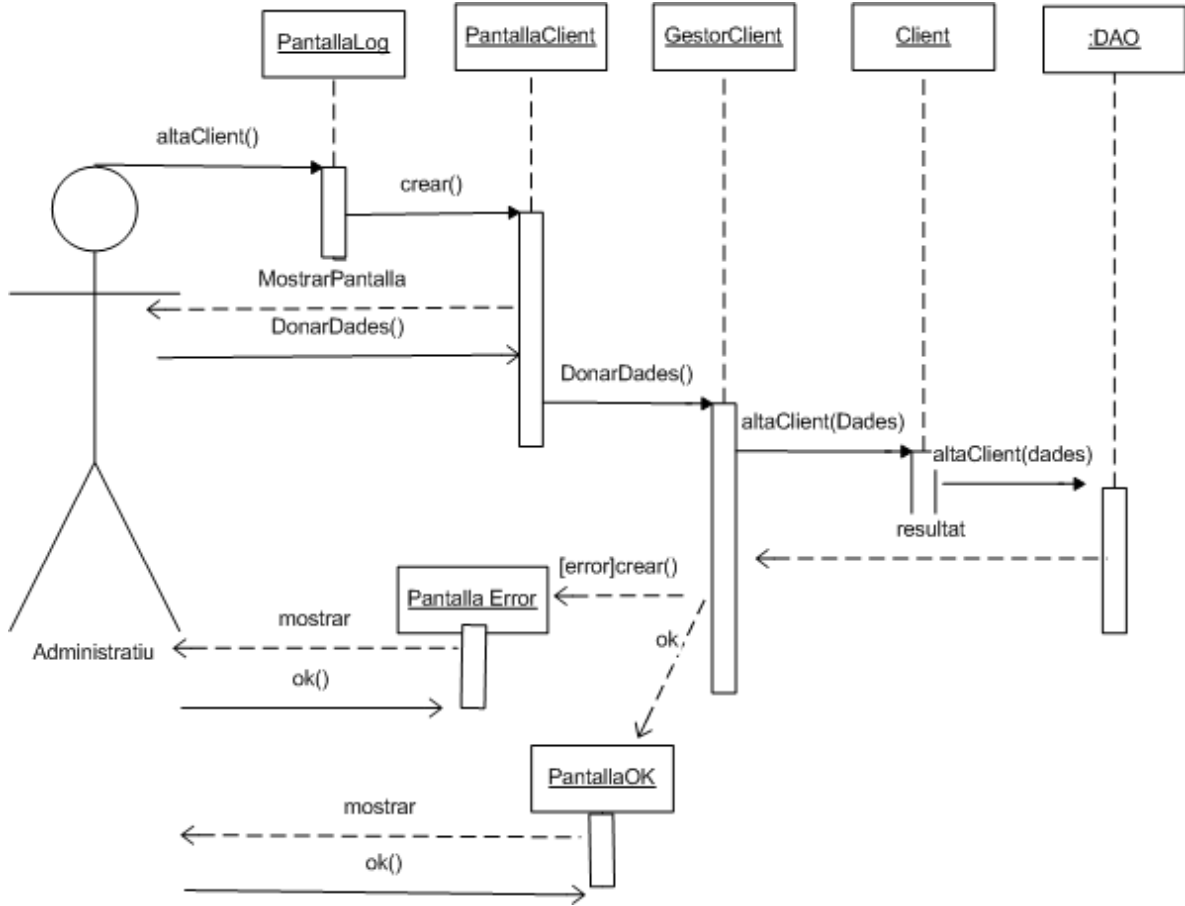
Casos d'ús: "Afegeir reserva Carret compra" , "Finalitzar compra i obtenir factura" i "Fer pagament bestreta i factura"



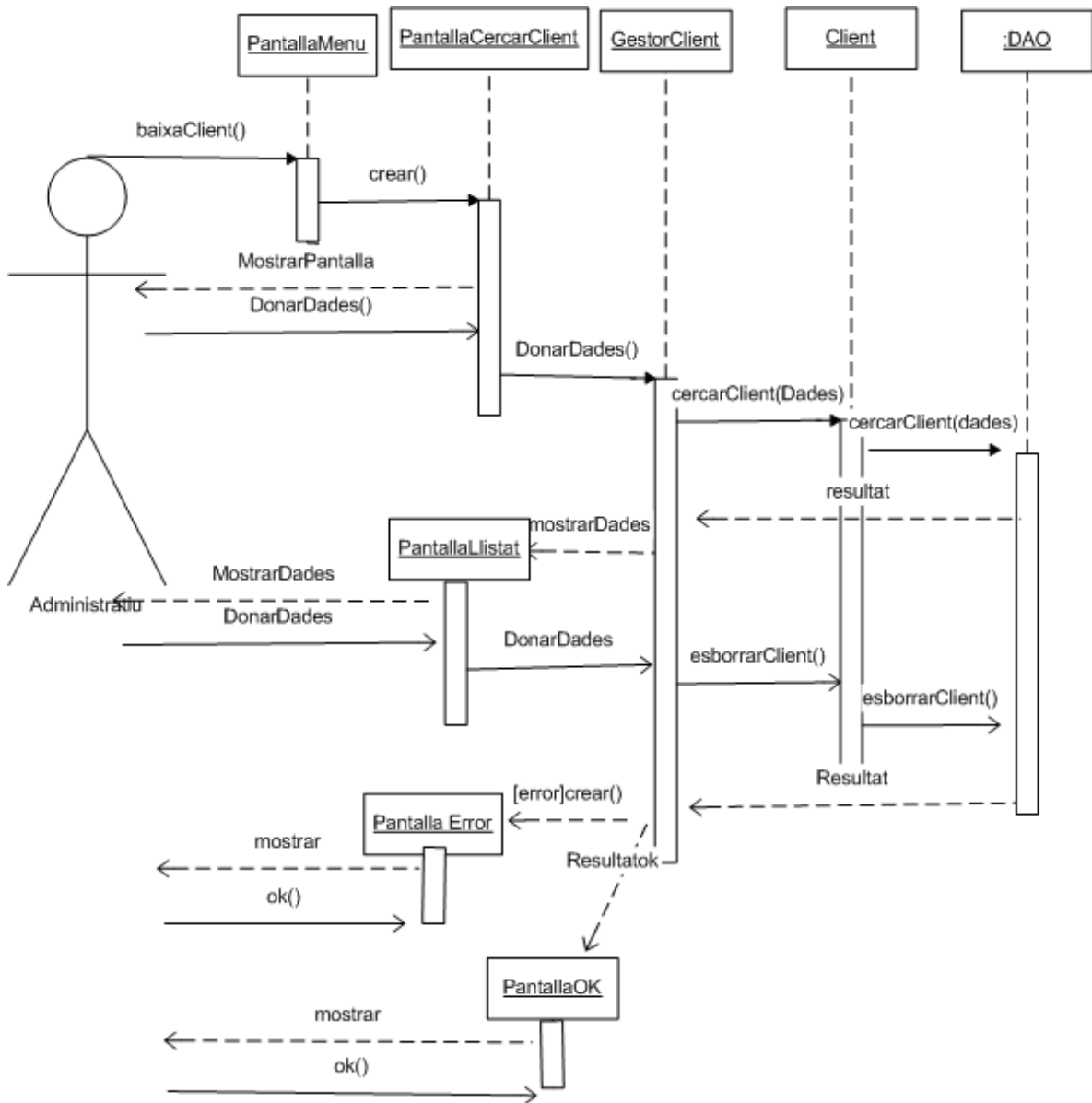
Cas d'ús "Consultar Reserves"



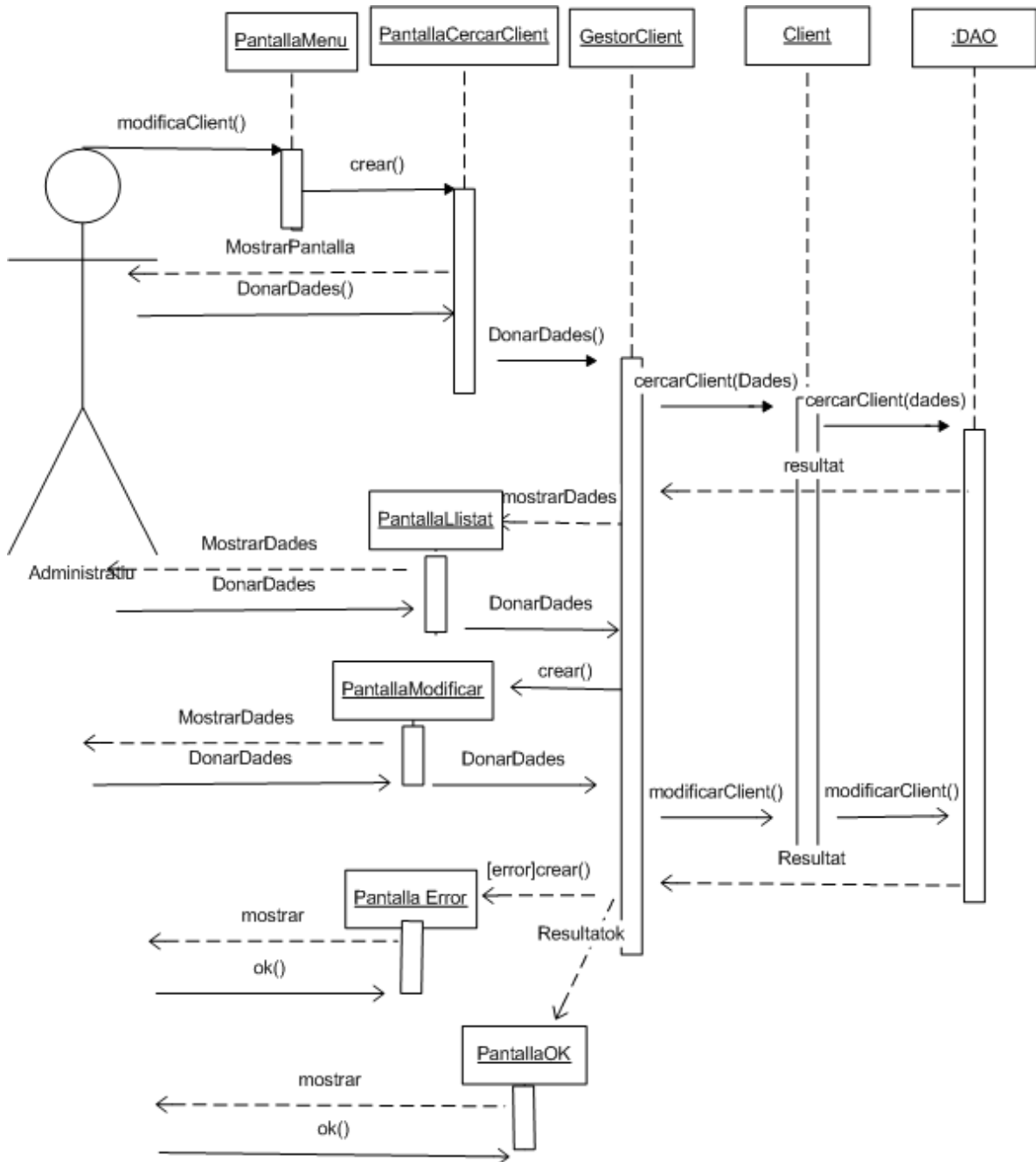
Casos d'ús per al "Manteniment de clients":
Alta de client



Baixa de client



Modificació de client



Arquitectura, patrons de disseny

La arquitectura que faré servir, com ja he comentat en diversos punts d'aquesta memòria, serà una arquitectura J2EE multicapa i distribuïda.

Les capes lògiques en que estarà dividida l'aplicació seran:

- Client
- Presentació
- Negoci
- Dades

Les tres màquines físiques que considerem són:

- Màquina client: Té un navegador i es pot connectar per HTTP al servidor d'aplicacions. Es correspon a la capa lògica Client.
- Servidor d'aplicacions: Té un servidor d'aplicacions instal·lat amb un contenidor web i un contenidor d'EJBs (al nostre cas el servidor d'aplicacions és un JBoss). En algunes arquitectures més complexes podríem tenir servidors dedicats a la lògica de negoci i servidors dedicats a la capa de presentació. Aquest servidor es comunica amb la base de dades relacional per JDBC. Es correspon a les capes lògiques de presentació i negoci.
- Base de dades relacional: Té un servidor de base de dades relacional (al nostre cas un MySQL) que emmagatzema les dades persistents de l'aplicació. Es correspon a la capa lògica de dades.

La capa de presentació s'ha dissenyat seguint un patró de Front Controller amb dos controladors per al subsistema d'Administració de la central, un controlador per al subsistema dels propietaris i un altre per als clients. Cadascun d'aquests tres subsistemes té les seves pròpies vistes. D'aquesta manera se centralitzen en un sol punt d'accés tots els processos d'autenticació, delegació del processament de la petició i creació de vistes. El seguiment d'aquest patró pot evitar la duplicació de codi, facilita mantenir la coherència entre les diverses vistes, i afavoreix el manteniment en tenir ben localitzats els punts a on es prenen decisions. Els controladors s'han implementat com a servlets i les vistes com a pàgines JSP.

La capa de negoci s'ha dissenyat seguint un patró de Session Façade, per tal d'obligar que totes les peticions a la part servidor passin per un sol punt d'accés, que amaga i encapsula la complexitat que hi pot haver al darrera. En aquest cas es tracta d'un EJB de sessió sense estat que fa aquesta funció.

Els EJB de entitat seran els encarregats d'accedir a les dades de la base de dades i fer els canvis necessaris. He triat que estiguin fets amb persistència gestionada pel contenidor (CMP), la qual cosa farà que l'aplicació sigui més independent de la plataforma i per tant, portable. Per un altra banda, la implementació amb CMP s'haurà de definir als arxius `ejb-jar.xml` corresponents i hauré d'usar el llenguatge EJBQL per als mètodes Find.

Aquesta arquitectura fomenta l'ús del patró de disseny Model View Controller 2, on el model està format pels components EJB que controlen les

dades que fa servir l'aplicació, la vista pels components que presenten les dades al client (pàgines JSP) i el controlador pels components responsables de la gestió i coordinació de les activitats del model i de la vista (servlets controladors). Un dels avantatges d'aquesta arquitectura és que no hi ha processament dins de la capa de presentació; aquesta simplement és responsable de recuperar qualsevol objecte que hagi estat creat prèviament per el controlador, extreure el contingut dinàmic i inserir-lo en la plantilla estàtica que enviarà al client per la seva presentació.

Per tal de poder transferir les dades entre les diferents capes de l'aplicació haig fet servir Value Objects. Es tracta de definir unes classes Serializables accessibles tant per la capa de negoci com per la capa de presentació, i omplir els atributs amb una còpia dels atributs de les entitats del model.

Capítol 5. Implementació i proves

Procés d'implementació

Per a la implementació del codi Java d'aquest projecte haig fet servir Eclipse 3.1. Les llibreries externes que s'han usat han estat servlet.jar, jbossall-client.jar i log4j.jar. La compilació i desplegament l'haig fet amb Apache Ant 1.6.4 des d'una finestra de sistema.

El servidor d'aplicacions sobre el que s'ha desplegat i provat l'aplicació ha estat JBoss 4.0.2, que ja porta Tomcat incorporat. Pel què fa al gestor de base de dades he usat MySQL 4.1 i el conector JDBC, MySQL Connector Java 3.1.10.

Finalment, per les pàgines JSP haig utilitzat el programa Macromedia Dreamweaver MX.

Hi ha un llistat detallat de tot el programari a l'[annex](#) d'aquesta memòria.

Donada la meua inexperiència en aplicacions distribuïdes i en J2EE en particular, era previsible que la fase d'implementació requeriria un esforç molt important, i així ha estat.

Per una banda he hagut de aprendre a fer servir tota una sèrie de tecnologies que no coneixia, com MySQL, EJBs, Ant, JSP, Servlets, CSS, JavaScript, EJBQL etc.. i posar-les en pràctica durant la implementació.

Per un altra banda, era necessari utilitzar alhora Eclipse i JBoss en tot moment durant la implementació i fins i tot en la fase de la presentació, havia de tenir també engegat el Dreamweaver. Tots tres programes, especialment Eclipse i JBoss, consumeixen una gran quantitat de recursos de l'ordinador, i com que la meua màquina té un processador a 700 Mhz. doncs el funcionament de tot ha estat lent i difícil i m'ha ocupat molt més temps del que m'hagués ocupat amb una màquina més potent.

Amb tot això, he hagut d'invertir una gran quantitat d'hores per portar a terme la implementació del projecte.

El meu punt de partida per portar a terme la implementació va ser la Pràctica tutoritzada del laboratori d'Enginyeria de Programari i Components Distribuïts. A partir d'aquí he hagut d'anar superant les dificultats que m'han anat sorgint. Algunes de les dificultats que m'he trobat en el procés d'implementació han estat les següents:

- La feina de fer totes les interfícies per cada EJB d'entitat és bastant lenta i repetitiva quan les taules de la Base de Dades tenen molts camps, com era el cas.
- Una de les taules que faig servir, Reserva, tenia una clau primària formada per tres camps i la implementació dels EJB en aquest cas és diferent. Vaig haver de crear un altra classe ReservaPK amb els tres camps que formen part de la clau primària.
- M'ha resultat també bastant difícil de fer servir el llenguatge EJBQL, que havia de fer servir per als EJBs d'entitat amb CMP. Concretament, a l'hora d'obtenir un llistat de les cases disponibles entres dues dates, en comptes d'obtenir aquest llistat directament de

les consultes EJBQL, ho vaig haver de fer amb codi Java amb un mètode.

- Per guardar l'estat de variables entre crides HTTP haig utilitzat Session Beans amb estat, la qual cosa m'ha permès que un client pugui fer la reserva de més d'una casa en una mateixa sessió i obtenir la factura de totes les reserves fetes en la sessió.
- Per millorar la eficiència del servlet controlador al qual s'adrezen totes les peticions del client, el vaig dividir en quatre servlets. Això, per una banda ha fet que tingui mètodes repetits en algun dels servlets però per un altra també ha fet que el mètode service() dels servlets sigui més petit i eficient.

Característiques de la implementació

S'ha aplicat un model MVC 2 que separa clarament la capa de presentació, la capa de negoci i la capa de dades.

La capa de presentació s'ha implementat amb quatre servlets i multitud de pàgines JSP. Els servlets fan de FrontController, dos per al subsistema Administrador, un per al subsistema client i un per al subsistema propietari. Cadascun d'aquests servlets centralitza tota la interacció amb l'usuari i usa com a Dispatcher diversos fitxers JSP que fan de vistes per als casos d'ús. En aquesta capa s'ha usat l'API de Servlets i Java Server Pages.

La capa de negoci s'ha amagat darrera d'un Session Bean sense estat que serveix a tots els subsistemes de l'aplicació i que actua com a intermediari entre la capa client i la capa de persistència. Aquest Session Bean gestiona els objectes de negoci i persistents donant una capa de servei d'accés als clients i alhora controla el cicle de vida dels participants, creant-los, localitzant-los, modificant-los i esborrant-los.

Un EJB de sessió actuant com a façana ens proporciona els següents avantatges:

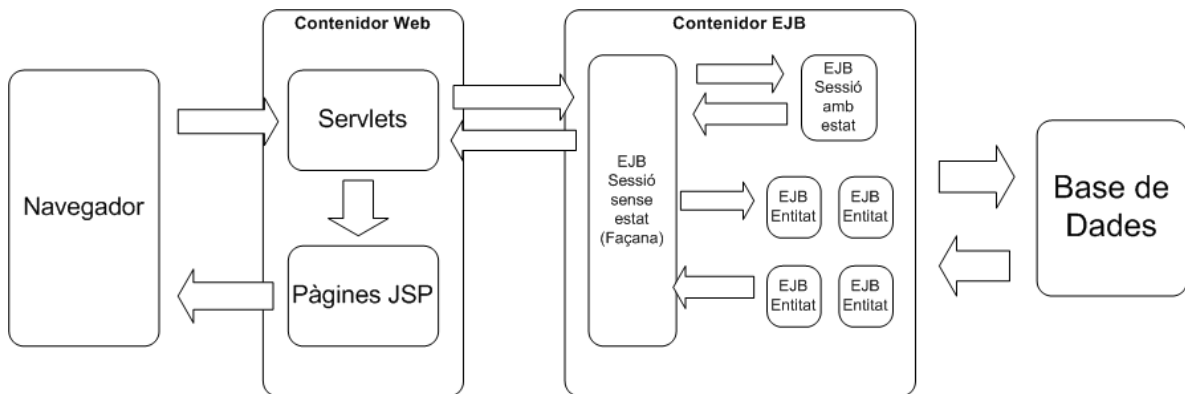
- Proporciona una interfície simple als clients tot ocultant-los les interaccions complexes dels components de negoci.
- Redueix el nombre d'objectes de negoci que exposem al client per mitjà de la capa de servei
- Ocultem al client les interaccions subjacents i les interdependències entre components de negoci
- Evita exposar els objectes de negoci subjacents directament al client mantenint l'acoblament entre capes al mínim

Per tal de mantenir l'estat entre crides HTTP haig fet servir Session Beans amb estat. Concretament per la implementació del procés de compra de varies reserves necessitava poder obtenir una factura amb informació de totes les reserves que el client feia en una sessió a l'aplicació. Els EJB amb estat es poden considerar com una extensió de l'aplicació client. Executen tasques per al client i mantenen un estat conversacional amb aquest client.

La capa de dades s'ha implementat amb cinc Entity Beans: Casa, Client, Propietari, Reserva i Factura que fan servir Persistència gestionada pel

contenedor (CMP). Els EJB d'entitat amb persistència gestionada pel contenidor deixen que el seu estat sigui gestionat automàticament pel contenidor. El programador tan sols ha de descriure els atributs i les relacions d'un EJB d'entitat en el descriptor de desplegament i crear les interfícies i components necessaris segons l'especificació J2EE, però les implementacions concretes són generades en temps de desplegament per part de les eines del contenidor. Quan es fa servir CMP també destaca la utilització del llenguatge EJB QL per declarar el comportament dels mètodes Find. L'EJB QL ens permet de descriure el comportament de mètodes de consulta d'una manera abstracta, fent aquestes consultes transportables a SGBDs de fabricants diferents.

Comentar també que faig servir Value Objects per a intercanviar informació entre les diferents capes de l'aplicació. Els clients per tant no reben EJBs d'entitat sinó que reben els Value Objects que representen aquesta informació.



Il·lustració 29. Arquitectura de l'aplicació

El processament típic d'una petició fent servir una arquitectura MVC 2 seria el següent:

- L'administrador de l'aplicació entra una petició al seu navegador per a demanar que el sistema li torni un llistat amb totes les cases que són a la central.
- La petició arriba al servidor web que s'adona que es tracta d'una petició dinàmica i per tant la delega al contenidor de servlets.
- El contenidor de servlets executa el codi del servlet que fa de controlador per aquesta petició (*ControladorServlet.java*).
- El servlet executa el codi que hi ha al mètode `service()` del servlet:
 - Mira quina acció ha d'executar
 - Crea una instància de l'EJB de sessió *Rural* i en crida el mètode corresponent per obtenir un llistat de cases en forma de Value Objects.
 - Posa aquesta llista de Value Objects a la request per tal que la vista que els ha de mostrar pugui trobar-los.
 - Redirecciona la petició cap a la vista encarregada de mostrar la llista de cases a l'administrador.

Pel que fa a la seguretat, haig fet servir seguretat declarativa a la capa web de l'aplicació, proporcionada per la mateixa infraestructura J2EE, que permet restringir per rols l'accés a les diferents parts del subsistema Administrador. Sols l'usuari amb rol Administrador podrà accedir a la part de l'aplicació del subsistema Administrador. Per fer-ho haig creat dos fitxers: users.properties i roles.properties, i després haig modificat el descriptor de desplegament del mòdul web (web.xml) per referenciar aquests rols. D'aquesta forma les URLs del subsistema administrador sols seran accessibles per usuaris que pertanyin al rol administrador.

Llista de classes

Classes de transferència de dades:

CasaVO.java
ClientVO.java
FacturaVO.java
FactVO.java
PropietariVO.java
ReservaVO.java

Capa de presentació:

Classes de control:

ControladorServlet.java (Subsistema Administrador)
ControladorServlet1.java (Subsistema Administrador)
ControladorServletCli.java (Subsistema Client)
ControladorServletProp.java (Subsistema Propietari)

Vistes Generals:

error.jsp, login.jsp, logincli.jsp, loginError.jsp, loginErrorOcupat.jsp,
loginprop.jsp, Menu.jsp, MenuAdmin.jsp, MenuCli.jsp, MenuProp.jsp

Vistes de la part Administrador:

altacasa.jsp, altaclient.jsp, altapropietari.jsp, altareserva.jsp,
buscarreserva.jsp, detallcasa.jsp, detallclient.jsp, detallpropietari.jsp,
detallreserva.jsp, llistatcases.jsp, llistatclients.jsp, llistatpropietaris.jsp,
llistatreserves.jsp, modificarcasa.jsp, modificarclient.jsp,
modificarpropietari.jsp, modificarreserva.jsp

Vistes de la part Client:

altaclient.jsp, altareserva.jsp, detallcasa.jsp, displayReserva.jsp,
finalitzarfactura.jsp, llistatcases.jsp, llistatcasesdisponibles.jsp,
llistatreserves.jsp, reservaExist.jsp

Vistes de la part Propietari:

altacasa.jsp, altapropietari.jsp, llistatcases.jsp, llistatreserves.jsp,
modificarcasa.jsp

Capa de negoci:

Session Beans sense estat:
RuralBean.java

Session Beans amb estat:
PropBean.java
CliBean.java

Classes clients per proves:
CasaClient.java
ClientClient.java
FacturaClient.java
PropietariClient.java
ReservaClient.java
RuralClient.java

Altres classes d'entitats:
ReservaPK.java

Capa de dades:

Entity Beans:
Casa
Client
Propietari
Reserva
Factura

Proves

La fase de test de l'aplicació ha consistit en omplir la BD amb algunes dades d'exemple i comprovar l'execució correcta dels casos d'ús especificats a l'anàlisi. Aquesta comprovació l'haig fet tant a nivell dels resultats que apareixien en pantalla com a nivell de comprovar en cada cas els efectes d'alguns dels casos d'ús en la base de dades.

Òbviament no haig esperat a tenir tota l'aplicació codificada per començar a fer les proves. Les proves s'han anat fent a mesura que la implementació anava avançant i s'anaven afegint noves funcionalitats en cada capa. Les classe Java que haig fet servir per les proves es troben al directori cat/client del lliurament fet a la fase d'implementació.

Per una banda haig fet proves per cada EJB d'entitat una vegada considerava que el tenia acabat, fent servir directament els mètodes des del client Java. Per exemple, per al EJB d'entitat Casa, vaig fer les proves amb el Client *CasaClient.java*, i així mateix per a tots els altres.

Després, quan vaig tenir fet el EJB de sessió sense estat, Rural, que servia de façana cap als EJB d'entitat, vaig fer proves per aquest nivell de capa, amb RuralClient.java.

Finalment, quan ja vaig tenir fets els servlets controladors i els JSP vaig començar a fer les proves definitives amb el navegador, comprovant també que la navegació entre les pàgines web funcionava correctament.

Capítol 6. Conclusions

Tot i que al començament ho veia tot molt complicat, degut a la gran quantitat d'informació nova a aprendre, finalment he aconseguit implementar quasi totes les funcionalitats que m'havia proposat inicialment i ara puc dir que penso que he aconseguit l'objectiu principal d'aquest projecte, que no era altre que introduir-me en el món de les aplicacions empresarials de comerç electrònic amb la tecnologia J2EE.

He après el funcionament de les parts més importants de la tecnologia J2EE: les pàgines JSP, Servlets, EJBs d'entitat i de sessió amb i sense estat. He fet servir CMP per als EJB d'entitat i penso que és una tecnologia de facilita molt la feina del programador però que encara ha de madurar molt més en els pròxims anys. També he descobert altres eines accessòries que es fan imprescindibles en un projecte d'aquestes característiques com Ant, que m'ha permès automatitzar el procés de compilar, empaquetar, copiar i desplegar l'aplicació. Sense aquesta eina fer totes les proves hagués estat pràcticament impossible. Una altra eina que ha estat imprescindible ha estat l'IDE Eclipse que m'ha permès detectar els errors de compilació de les classes i corregir-los ràpidament.

Des del punt de vista del projecte, com a projecte de construcció de programari, he après que portar la planificació dia a dia ajuda a controlar i gestionar millor el projecte. Per un altra banda, penso que haver fet totes les passes necessàries per al desenvolupament del projecte, des dels requeriments fins a la implementació i documentació és una tasca que aproxima el projecte al que hagués estat el desenvolupament d'un projecte real.

També haig pogut apreciar el benefici d'usar J2EE amb patrons de disseny. La complexitat d'una aplicació distribuïda pot desbordar fàcilment un programador que toca per primera vegada aquesta tecnologia. D'una banda l'ús de J2EE permet no haver-se de preocupar pel "middleware" en temps de desenvolupament. Per una altra banda, l'aplicació de patrons de disseny s'ha mostrat molt útil per a compensar la falta d'experiència. El seguiment de patrons de disseny reconeguts permet una corba d'aprenentatge molt més ràpida i fonamentada.

Pel que fa a les possibles millores del producte concret desenvolupat penso que podrien ser les següents:

- L'estètica de la interfície d'usuari és millorable.
- La definició dels mètodes "Find" amb EJB QL també penso que és millorable, ja que poden ser més complexos que els que jo haig posat i això milloraria l'eficiència d'aquestos mètodes.
- La utilització d'un patró de disseny EJB Locator per centralitzar en una única classe especialitzada els "lookups" a l'arbre JNDI del servidor d'aplicacions també milloraria la eficiència i posterior manteniment del codi.

- No haig fet tractament d'errors ni de dades d'entrada, tasca que seria imprescindible en un projecte real.
- No haig utilitzat "includes" en les pàgines JSP la qual cosa m'ha portat ha tenir duplicats trossos de codi.
- He implementat el mètode service() dels servlets controladors, que en funció de la petició del client tria els mètodes a invocar i la pàgina JSP a mostrar, amb una cadena if-then-else. En un projecte real s'hauria d'utilitzar algun patró de disseny ja que aquesta estratègia es fa inviable si tenim moltes operacions.
- En una aplicació real seria molt recomanable fer servir el bastiment Struts que ja ens donaria implementat el patró de disseny MVC 2.
- La seguretat també és un altre element que s'hauria de millorar. S'haurien de fer servir els mecanismes que proporciona el contenidor d'aplicacions per impossible l'accés a pàgines privades sense autenticar-se.

Línies futures de l'aplicació i parts que s'haurien d'incorporar per a poder-se utilitzar en una central de reserves real:

- Incorporació de mòduls reals per al pagament de les reserves fetes.
- Afegir al projecte l'enregistrament a la base de dades de les factures generades i fins i tot donar la possibilitat de generar aquestes factures per lliurar-les al client i al propietari, al qual se li cobraria pel servei de lloguer de les seves cases a la central.
- L'alta de cases per part del propietari no haurien d'estar disponibles per fer reserves fins que el personal administratiu de l'empresa donés el seu vist-i-plau.

Amb totes aquestes millores i ampliacions penso que el producte desenvolupat estaria preparat per ser utilitzat en una central de reserves real.

Glossari

Central de reserves de cases rurals: Estructura que fa d'intermediària entre els propietaris de cases de turisme rural i els clients que volen llogar aquestes cases. Normalment es tracta d'una base de dades amb informació sobre les cases i les seves característiques, per una banda, i per l'altra amb la possibilitat que els clients puguin fer reserves de les cases disponibles entre dues dates.

Subsistema: Part de l'aplicació amb una certa autonomia i característiques diferencials respecte a la resta de l'aplicació, especialment des del punt de vista de l'usuari de l'aplicació.

Client: Persona física o jurídica que es posa en contacte amb la central de reserves per llogar una de les cases de la central per un període de temps determinat.

Administratiu: Treballador de l'empresa que gestiona la central de reserves i que té privilegis per fer qualsevol operació a l'aplicació.

Propietari: Persona física o jurídica que té la propietat legal d'una o varies de les cases que formen part de la central.

Casa: Casa de turisme rural. Una unitat autònoma amb tots els serveis necessaris per ser llogada durant un període de temps, tant pot ser una casa sencera, com una part d'una casa, apartament etc..

Reservar: Llogar una casa de turisme rural concreta per un període de temps i un client determinat. La reserva es fa efectiva en el moment en que les seves dades són gravades a la base de dades.

Carret de la compra: Estructura de dades que emmagatzemarà per un client determinat les dades sobre totes les reserves que ha fet en la sessió actual. En acabar la sessió aquesta estructura de dades serà esborrada i es farà persistent a la base de dades.

Casa disponible: Una casa rural es diu que està disponibles per ser llogada entre dues dates quan en aquest període de temps l'esmentada casa no té feta cap reserva, o sigui, no està ocupada per cap client.

Data d'entrada / sortida: Data de començament i acabament, respectivament, del període de lloguer de la casa per una reserva concreta.

Bibliografia.

- **Practica Tutoritzada.** Enginyeria del Programari de Components i Sistemes distribuïts. Josep Maria Camps i Riba. UOC. 2005
- **Modul EJB:Components distribuïts.** Felip Miralles Barrachina.
- **Mastering Enterprise Javabeans.** Third Edition. Ed Roman.Wiley.
- **EJB Design Patterns.** Floyd Marinescu. Wiley
- **Introduction to JSP Technology.** ibm.com/developerWorks
- **JAVA 2. Interfaces gráficas y aplicaciones para Internet.** Fco. Javier Ceballos. RA-MA
- **Jakarta Struts for dummies.** Mike Robinson and Ellen Finkelstein
- **Java™ 2 Enterprise Edition 1.4 Bible.** James McGovern, Rahim Adatia, Yakov Fain, Jason Gordon, Ethan Henry, Walter Hurst, Ashish Jain, Mark Little, Vaidyanathan Nagarajan, Harshad Oak, Lee Anne Phillips
- **JBoss 3.2 Workbook for Enterprise Javabeans.** Third Edition. Bill Burke and Sancho Labourey.
- J2EE API Documentation
<http://java.sun.com/j2ee/1.4/docs/api/index.html>
- **Sam's Teach Yourself MySQL in 21 Days.** Mark Maslakowski
- MySQL Reference Manual
<http://dev.mysql.com/doc/>

Annexos.

Programari utilitzat i instal·lació

Necessari pel funcionament de l'aplicació

- JDK 1.5.03
- JBOSS 4.0.2
- Ant 1.6.4
- MySQL 4.1
- MySQL Connector Java 3.1.10

Per al desenvolupament

- IDE Eclipse 3.1
- Macromedia Dreamweaver MX
- MySQL Administrator
- MySQL Query Browser
- Microsoft Visio

Instal·lació

Primer de tot s'haurà de configurar el JBoss 4.0.2 per a treballar amb MySQL.

Després haurem de crear les taules a la Base de Dades MySQL.

La instal·lació de l'aplicació consisteix simplement a desplegar l'arxiu "casesrurals.ear" al directori C:\JBOSS_HOME\server\default\deploy, on JBOSS_HOME és el directori on tenim instal·lat el JBoss.

Per utilitzar l'aplicació després d'engegar el JBoss i MySQL haurem d'obrir un navegador i apuntar a <http://localhost:8080/casesrurals/Menu.jsp> i ens apareixerà el menú principal de l'aplicació.

Alternativament, per compilar i desplegar tota l'aplicació al JBoss cal situar-se dins del directori /src i des d'aquí, que és on tenim l'arxiu build.xml, fem "ant all". Per fer les proves amb els clients haig fet servir l'ordre ant "runClient...".

Script de creació de la base de dades en MySQL

```
CREATE TABLE Propietari (  
    codipersona VARCHAR(10) NOT NULL,  
    nom VARCHAR(45),  
    cognoms VARCHAR(45),  
    dni VARCHAR(9),  
    adreca VARCHAR(100),  
    ciutat VARCHAR(20),  
    codipostal VARCHAR(5),  
    telefon VARCHAR(10),  
    email VARCHAR(20),  
    fax VARCHAR(10),  
    user VARCHAR(45),  
    password VARCHAR(45),  
    PRIMARY KEY(codipersona)  
)TYPE=InnoDB;
```

```
CREATE TABLE Client (  
    codipersona VARCHAR(10) NOT NULL,  
    nom VARCHAR(45),  
    cognoms VARCHAR(45),  
    dni VARCHAR(9),  
    adreca VARCHAR(100),  
    ciutat VARCHAR(20),  
    codipostal VARCHAR(5),  
    telefon VARCHAR(10),  
    estatcivil VARCHAR(20),  
    frequencia VARCHAR(10),  
    user VARCHAR(45),  
    password VARCHAR(45),  
    PRIMARY KEY(codipersona)  
)TYPE=InnoDB;
```

```
CREATE TABLE Administratiu (  
    codipersona VARCHAR(4) NOT NULL,  
    nom VARCHAR(45),  
    cognoms VARCHAR(45),  
    dni VARCHAR(9),  
    adreca VARCHAR(100),  
    ciutat VARCHAR(20),  
    codipostal VARCHAR(5),  
    telefon VARCHAR(10),  
    antiguitat INTEGER,  
    PRIMARY KEY(codipersona)  
)TYPE=InnoDB;
```

```
CREATE TABLE Poble (  
    codipoble VARCHAR(4) NOT NULL,  
    nom VARCHAR(45),  
    poblacio INTEGER,  
    comarca VARCHAR(45),  
    descripcio VARCHAR(200),  
    PRIMARY KEY(codipoble)
```

```
)TYPE=InnoDB;
```

```
CREATE TABLE Casa (  
    codicasa VARCHAR(4) NOT NULL,  
    nom VARCHAR(45),  
    preunit FLOAT,  
    places INTEGER,  
    plantes INTEGER,  
    categoria VARCHAR(20),  
    hdobles INTEGER,  
    hsimples INTEGER,  
    poble VARCHAR(4),  
    propietari VARCHAR(10),  
    PRIMARY KEY(codicasa)  
)TYPE=InnoDB;
```

```
ALTER TABLE Casa add index (poble), add constraint FK_POBLE  
foreign key (poble) references POBLE (codipoble) ON DELETE SET NULL;  
ALTER TABLE Casa add index (propietari), add constraint FK_PROPIETARI  
foreign key (propietari) references PROPIETARI (codipersona) ON DELETE SET  
NULL;
```

```
CREATE TABLE Factura (  
    codifactura VARCHAR(3) NOT NULL,  
    data DATE,  
    baseimposable FLOAT,  
    iva FLOAT,  
    total FLOAT,  
    descripcio VARCHAR(200),  
    PRIMARY KEY(codifactura)  
)TYPE=InnoDB;
```

```
CREATE TABLE Reserva (  
    codicasa VARCHAR(4) NOT NULL,  
    codipersona VARCHAR(10) NOT NULL,  
    entrada DATE,  
    sortida DATE,  
    preureserva FLOAT,  
    persones INTEGER,  
    factura VARCHAR(3),  
    PRIMARY KEY(codicasa,codipersona,entrada)  
)TYPE=InnoDB;
```

```
ALTER TABLE Reserva add index (factura), add constraint FK_FACTURA  
foreign key (factura) references FACTURA (codifactura) ON DELETE SET NULL;  
ALTER TABLE Reserva add index (codicasa), add constraint FK_CASA  
foreign key (codicasa) references CASA (codicasa);  
ALTER TABLE Reserva add index (codipersona), add constraint FK_PERSONA  
foreign key (codipersona) references CLIENT (codipersona);  
--- on delete restrict, per defecte, això pot donar problemes, ho pots  
--- canviar des de MySQL Query Browser
```

```
CREATE TABLE Targeta (  
    numero VARCHAR(20) NOT NULL,
```

```
        caducitat DATE,  
        tipus VARCHAR(30),  
        client VARCHAR(4),  
        PRIMARY KEY(numero)  
    )TYPE=InnoDB;
```

```
ALTER TABLE Targeta add index (client), add constraint FK_CLIENT  
foreign key (client) references CLIENT (codipersona) ON DELETE SET NULL;
```

---INSERCIONS DE TOTES LES TAULES

```
INSERT INTO PROPIETARI VALUES ('73384794','Jordi','Cerdà Folch',null, 'Julian  
Prats 1', 'Morella','12300','964160780','jcerda@morella.net', '964160780', 'jcerda',  
'20graos');
```

```
INSERT INTO PROPIETARI VALUES ('20242546','Amparo','Pitarch Segura',null,  
'Julian Prats 1', 'Morella','12300','964160780','apitarch@morella.net', '964160780',  
'apitarch', 'hipo');
```

```
INSERT INTO PROPIETARI VALUES ('20247846','Eladio','Pitarch Troncho',null,  
'Julian Prats 29', 'Morella','12300','964160790','epitarch@morella.net',  
'964160780', 'epitarch', 'soro');
```

```
INSERT INTO POBLE VALUES ('PU01','Herbeset',40,'Els Ports', 'Poble situat a lloc  
tranquil i asolellat');
```

```
INSERT INTO POBLE VALUES ('PU02','Castell de Cabres',60,'El Maestrat', 'Casa  
situada a lloc tranquil i gelat');
```

```
INSERT INTO POBLE VALUES ('PU03','Xiva',83,'Els Ports', 'Casa situada a lloc  
tranquil i asolellat, paratges impressionants');
```

```
INSERT INTO POBLE VALUES ('PU04','Ortells',130,'Els Ports', 'Casa situada a lloc  
tranquil i asolellat, ben comunicada');
```

```
INSERT INTO CASA VALUES ('C001','Casa La Mestra I', 60.52,8,2,'Alta',3,2,  
'PU01','73384794');
```

```
INSERT INTO CASA VALUES ('C002','Casa El Cuartel I',  
50.12,10,2,'Mitjana',3,2,'PU02','20242546');
```

```
INSERT INTO CASA VALUES ('C005','Casa La Mestra II', 65.52,8,2,'Alta',3,2,  
'PU01','73384794');
```

```
INSERT INTO CASA VALUES ('C006','Casa La Mestra III', 70.52,8,2,'Alta',3,2,  
'PU01','73384794');
```

```
INSERT INTO CASA VALUES ('C009','Casa El Cuartel II',  
55.12,10,2,'Mitjana',3,2,'PU02','20242546');
```

```
INSERT INTO CLIENT VALUES ('2525652','Gemma','Fernandez Lopez',null,  
'Avinguda Lluís Prats 89', 'Barcelona','12378','928160780','casada', '928160780',  
'apitarch', 'hipo');
```

```
INSERT INTO CLIENT VALUES ('2451472','Robert','Aguilar Lopez',null, 'Avinguda  
Lluís Prats 100', 'Barcelona','12378','928199980','casat', '928160780','epitarch',  
'soro');
```

```
INSERT INTO CLIENT VALUES ('2525552','Genís','Lopez Lopez',null, 'Avinguda Lluís  
Prats 89', 'Barcelona','12378','928160780','casada', '928160780', 'glopez', 'gesr');
```

```
INSERT INTO CLIENT VALUES ('2951472','Francesc','Sorribes Lopez',null, 'Avinguda  
Lluís Prats 100', 'Barcelona','12378','928199980','casat', '928160780','epinto',  
'soron');
```

```
INSERT INTO CLIENT VALUES ('2951487','Francesca','Prats Lopez',null, 'Avinguda  
Lluís Prats 100', 'Barcelona','12378','928199980','casat', '928160780','fprats',  
'frte');
```