

Proposta de TFC

Donde Comer?

Consultor	Vicenç Font Sagrista
Alumne	Jose Antonio García Pérez
TFC	J2EE
Fases realitzades	Pla de projecte
Versió	1.0

Dedicatòria i agraïments

Ja fa temps que em vaig embarcar en aquesta aventura que ja toca a la seva fi, i ara em recordo de totes les hores dedicades a realitzar les PACs, pràctiques o estudiar pels exàmens i que gràcies a la família he pogut realitzar. A ells va dirigida aquesta dedicatòria, a la meva família i a la meva núvia, per totes les estones de comprensió i recolzament que m'han ofert al llarg d'aquesta etapa i que sense ells seria impossible haver-la realitzat.

Per últim, també vull agrair a tots els consultors, tutors, personal de la UOC i companys d'aula l'ajuda que m'han donat durant tots aquests anys.

INDEX

<u>Dedicatòria i agraïments</u>	<u>2</u>
1. <u>Descripció TFC</u>	<u>5</u>
1.1. <u>Resum</u>	<u>5</u>
<u>2. Objectius</u>	<u>6</u>
<u>3. Enfocament i mètode</u>	<u>10</u>
<u>4. Planificació temporal</u>	<u>11</u>
4.1. <u>Fites</u>	<u>11</u>
4.2. <u>Diagrama Gantt</u>	<u>12</u>
<u>5. Contingut del document</u>	<u>13</u>
<u>6. Descripció del projecte</u>	<u>13</u>
6.1. <u>Subsistema Client</u>	<u>15</u>
6.1.1. <u>Client invitat:</u>	<u>15</u>
6.1.2. <u>Client registrat:</u>	<u>15</u>
6.2. <u>Subsistema Propietari del restaurant</u>	<u>16</u>
6.3. <u>Subsistema Administrador</u>	<u>18</u>
6.4. <u>Anàlisi detallat dels casos d'ús</u>	<u>19</u>
6.4.1. <u>Descripció gràfica dels casos d'ús</u>	<u>19</u>
6.4.2. <u>Descripció textual dels casos d'ús</u>	<u>20</u>
6.4.2.1. <u>Enregistrar-se</u>	<u>20</u>
6.4.2.2. <u>Accedir</u>	<u>21</u>
6.4.2.3. <u>Manteniment de Clients</u>	<u>22</u>
6.4.2.4. <u>Alta Client</u>	<u>22</u>
6.4.2.5. <u>Baixa Client</u>	<u>23</u>
6.4.2.6. <u>Modificar Client</u>	<u>24</u>
6.4.2.7. <u>Manteniment restaurant</u>	<u>25</u>
6.4.2.8. <u>Alta restaurant</u>	<u>25</u>
6.4.2.9. <u>Baixa restaurant</u>	<u>26</u>
6.4.2.10. <u>Modificar restaurant</u>	<u>27</u>
6.4.2.11. <u>Cercar restaurants</u>	<u>28</u>
6.4.2.12. <u>Gestionar reserves</u>	<u>28</u>

6.4.2.13. Alta reserva	29
6.4.2.14. Baixa reserva	30
6.4.2.15. Modificació reserva	30
6.4.2.16. Consultar reserves	31
6.4.2.17. Puntuar restaurant	32
6.4.2.18. Comentar restaurant	32
6.4.2.19. Estadístiques	33
6.5. Disseny.	34
6.5.1. Arquitectura	34
6.5.2. Alternatives de disseny	35
6.5.3. Definició de l'arquitectura	36
6.5.4. Diagrama de classes del model de negoci	39
6.5.5. Persistència	40
6.5.6. Internacionalització	40
6.5.7. Control d'errors i excepcions	41
6.5.8. Diseny de la interfície gràfica	42
<u>7. Configuració e instal·lació</u>	<u>53</u>
<u>7.1. Requeriments del software</u>	<u>53</u>
<u>7.2. L'aplicació</u>	<u>54</u>
<u>7.2.1. Fitxers de configuració</u>	<u>55</u>
<u>7.2.2. Desplegament de l'aplicació</u>	<u>56</u>
<u>8. Conclusions</u>	<u>57</u>
<u>9. Annexos</u>	<u>58</u>
<u>9.1. Instruccions SQL</u>	<u>58</u>
<u>9.2. Interfície</u>	<u>67</u>

1. Descripció TFC

1.1. *Resum*

“Donde Comer?” és una pàgina especialitzada en la valoració de restaurants.

En els dies en els quals estem, les valoracions objectives sobre restaurants estan monopolitzades per les grans guies de restauració. Són guies fortament influenciades per interessos econòmics, comercials o fins i tot turístics.

La nostra proposta és diferent. En aquesta guia no volem sotmetre'ns a cap prejudici gastronòmic i esperem que sigui l'usuari, i solament ell, el que faci les valoracions. Tot de forma gratuïta.

El internauta anònim és el que posa els comentaris i valora els diferents punts forts de cada establiment. D'aquesta forma, considerem molta més rica la valoració de cadascun dels restaurants que presentem en la web.

A qui es dirigeix aquesta web

Als clients

La web es dirigeix sobretot als clients dels restaurants. D'una banda per obtenir informació sobre les dades del establiment (adreça, telèfon, situació...), i per un altre, per veure les valoracions desinteressades i sinceres d'altres clients. I fer reserves online dels restaurants.

Als establiments

També animem als propietaris dels establiments a col·locar i mantenir les dades del seu restaurant en la web. D'aquesta manera, a més de fer publicitat del seu restaurant, ho posarà en el punt de mira de la resta d'internautes que entrin a valorar-ho. I així podrà fer-se una idea de les impressions de la seva clientela després de pagar el compte, i millorar en aquest sentit, el seu servei tot el possible.

A part, millorar la qualitat del servei i l'atenció al client. I proposar un servei interactiu de reserves a través d'internet.

La nostra aplicació, sobre la plataforma JEE, segueix el disseny en capes, concretament en les capes de presentació, negoci i dades.

A la capa de presentació, s'han utilitzat les tecnologies de JSP, HTML i CSS intentant aconseguir una interfície gràfica que sigui entenedora i fàcil d'usar.

Farem servir el patró Model Vista Controlador, per tal de separar la interfície d'usuari de la lògica de negoci, a tal efecte s'ha utilitzat el *framework* STRUTS.

A la capa de negoci és on estarà el gruix de l'aplicació que s'executarà sobre un servidor d'aplicacions.

La capa de dades, fa servir BD Oracle. Per interactuar amb la base de dades, s'ha fet servir el *framework*, que ens permet tractar les taules de la base de dades com si fossin objectes Java ordinaris.

2. Objectius

Aquest treball fi de carrera (TFC) té com principal objectiu el familiaritzar-me amb la tecnologia JEE. Els objectius que em proposo per a aquest TFC són:

- Adquirir una mica més de domini en el llenguatge de programació Java.
- Fer un disseny de l'aplicació en tres capes: capa de presentació, capa de lògica de negoci, i capa dades, implementant cadascuna aquells aspectes de la funcionalitat de l'aplicació que li són propis, i utilitzant els patrons de disseny més adequats.
- Comprendre tot el cicle de desenvolupament de programari, des de la presa de requeriments fins al lliurament del producte final.
- Demostrar com amb eines Open Source es poden desenvolupar aplicacions totalment professionals.
- Estudiar els frameworks i estàndards disponibles per a les diferents capes de l'aplicació, tant els definits per les especificacions de JavaEE com els disponibles gràcies a la comunitat de software lliure per tal d'escollir els més adequats.

- Estructurar la lògica de negoci i la persistència de dades per garantir el correcte funcionament en situacions de concurrència d'usuaris.
- Establir un conjunt de rols i definir els mecanismes d'autenticació i autorització que donaran seguretat a l'aplicació.
- Intentar utilitzar en la mesura del possible la tecnologia GOOGLE per al desenvolupament de l'aplicació.

Bàsicament es tracta de crear una comunitat que puguin valorar restaurants, puntuar restaurants, així com poder fer reserves.

L'aplicació es pot dividir en dues grans parts, per una banda la part *clients* i per l'altra la part *restaurants*.

La primera part es divideix en dues vessants:

- Client invitat: La seva interfície estarà dedicada a la creació d'una reserva, sense excedir de tres passos o pantalles:
 1. Consulta de disponibilitat,
 2. Introducció de les dades del client,
 3. Confirmació de la reserva.
- Client registrat: amb una interfície dedicada a la creació, modificació i consulta de reserves i cerques. L'usuari disposarà d'un menú de selecció d'aquestes tres categories esmentades, sent els seus respectius passos o pantalles els següents:
 - Creació.
 1. Consulta de disponibilitat,
 2. Introducció de les dades del client,
 3. Confirmació de la reserva.
 - Modificació.
 1. Introducció de dades per a la recerca,
 2. Consulta de la reserva,
 3. Modificació de dades,

4. Confirmació de la reserva.

- Consulta.
 1. Introducció de dades per a la recerca,
 2. Consulta de la reserva.

- Cerca.
 1. Cerca de restaurants, veure les seves dades.
 2. Valoració o crítica de restaurants.
 3. Puntuació de restaurants.

A la segona part, els restaurants: amb una interfície dedicada a la creació, modificació de les seves dades i consulta de reserves dels clients.

- Creació.
 1. Alta del restaurant,
 2. Introducció al detall de les seves dades.
 - Nom, adreça, telèfon, situació.
 - Podrà incloure fotografies del restaurant.

- Modificació.
 1. Modificació de dades.

- Consulta.
 1. Consulta de les reserves dels clients.

A part d'aquestes vessants, també existeix una altre que es l'administrador, la seva interfície esta dedicada a l'administració del sistema. És a dir, que tot funcioni correctament, i que no existeixi cap problema entre el usuari client i els restaurants.

El 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):

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

Com a part innovadora la confirmació s'enviarà a través d'un SMS, aquesta part es farà en la segona fase.

Al detall dels restaurant hi haurà un mapa d'ubicació del restaurant fent una connexió al google. I una altre, que es farà en la segona fase, de com arribar-hi, on t'indicarà com pots arribar des d'un punt determinat que introdueixi el client.

3. Enfocament i mètode

La metodologia seguida per crear el projecte ha suposat un estudi de cadascuna de les tecnologies aplicables, aquesta és una part important del treball, per ser fonamental per a l'obtenció de resultats satisfactoris i tecnològicament correctes.

Java 2 Enterprise Edition (J2EE) és una arquitectura multicapa, que conté diverses extensions API que serveixen de complement a l'edició estàndard de Java (J2ES), de tal manera que es complementen les dues tecnologies per poder crear aplicacions empresarials i orientades a la web, i cobrir necessitats d'un servidor.

L'aplicació s'ha estructurat en capes, concretament en les capes de presentació, negoci i dades. Aquesta és una de les formes de disseny més utilitzada en l'actualitat, ja que proporciona diversos avantatges.

Es pot treballar independentment cada capa, tenint en compte l'API o tecnologia que les ha de interrelacionar, per altra part, quan es vol fer algun canvi, es pot atacar la capa afectada sense haver d'intervenir en les altres. Utilitzarem el patró de disseny MVC.

Aquest ens facilita una clara separació entre les capes entrevinents en una aplicació web, això és, la de model de negoci (Model), la de presentació o vista (View) i la del controlador (Controller).

Per a això, s'ha optat pel Jakarta Struts Framework que s'ha convertit en un entorn ideal per al desenvolupament d'aplicacions web que utilitzen el MVC.

A més, també s'ha vist convenient utilitzar el patró Data Access Object (DAO), patró que encapsula no només les fonts de dades sinó també la forma en la qual accedim a elles.

Finalment, s'ha fet servir la utilitat log4j per generar el fitxer de seguiment del funcionament de l'aplicació. A part dels missatges propis del sistema, se'n van generar de personalitzats, per indicar els punts més importants dins del funcionament normal de l'aplicació.

4. Planificació temporal

4.1. Fites

D'entrada es preveuen les següents tasques:

Fita: PAC2 – 15/04/2013

- Estudi de l'arquitectura JavaEE i frameworks disponibles.
- Anàlisi detallat dels casos d'ús.
- Anàlisi de la interfície gràfica.
- Definició general de l'arquitectura de l'aplicació.
- Disseny del model de dades.
- Disseny de la lògica de negoci.
- Realitzar l'informe de PAC2.
- Instal·lació i configuració del software.

Fita: PAC3 – 03/06/2013

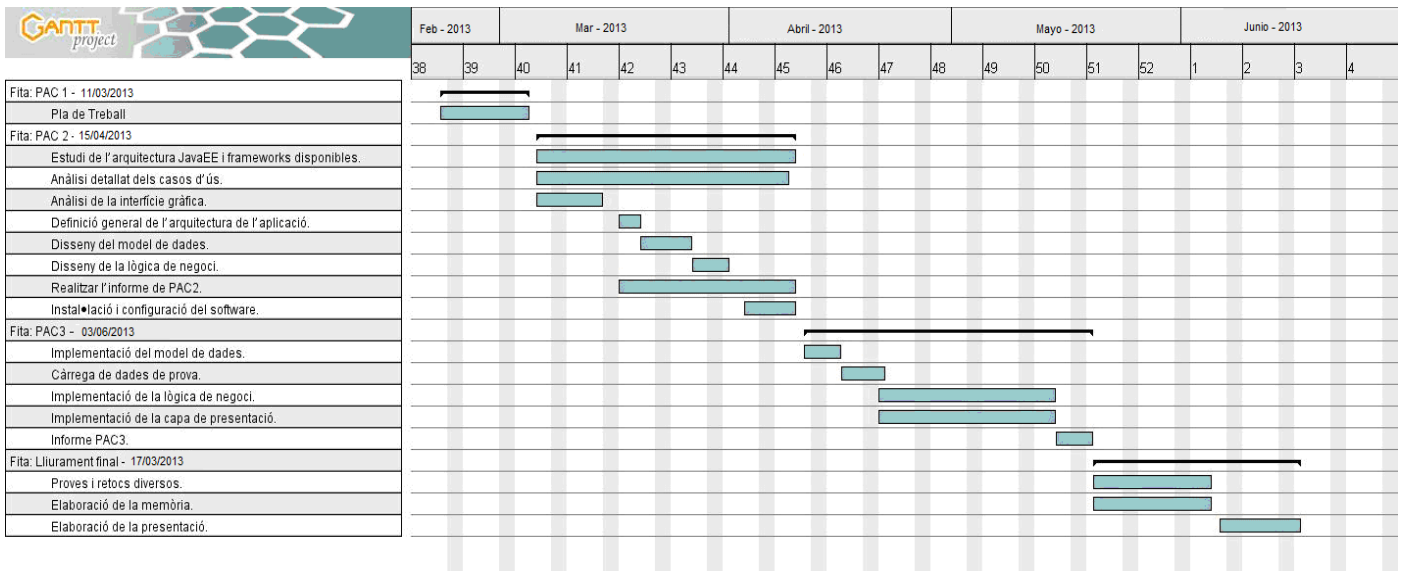
- Implementació del model de dades.
- Càrrega de dades de prova.
- Implementació de la lògica de negoci.
- Implementació de la capa de presentació.
- Informe PAC3.

Fita: Memòria - Lliurament final - 17/06/2013

- Proves i retocs diversos.
- Elaboració de la memòria.
- Elaboració de la presentació.

4.2. Diagrama Gantt

A continuació es mostra la planificació temporal de les tasques amb el diagrama de Gantt:



5. Contingut del document

Aquest document recull i documenta els requisits del sistema de programari a implementar. Per tant el seu objectiu defineix detalladament les necessitats d'informació que haurà de resoldre el programari. S'inclou l'anàlisi necessària per a poder traduir els requisits en especificacions més formals (diagrames d'UML) que facilitin posteriorment el desenvolupament de l'etapa de disseny, així com la identificació de les classes fonamentals i l'expressió dels casos d'ús.

Com es podrà desprendre de la seva lectura, es comença efectuant una aproximació general al sistema mitjançant una anàlisi global d'aquest, per després analitzar el sistema en subsistemes.

6. Descripció del projecte

Els objectius del projecte són implementar les funcions bàsiques per a la cerca de restaurants, creació, manteniment i consulta de reserves dels restaurants en una base de dades. El sistema està pensat per donar resposta a les necessitats pròpies d'una central de reserves de restaurants i constarà d'una sèrie de pantalles ben estructurades per facilitar la navegació. L'aplicació per una part oferirà informació sobre els restaurants i la seva disponibilitat i després permetrà al client fer la reserva del restaurant en cas de que aquest estigues lliure per a les dates indicades.

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

- Subsistema Client: Agruparà les funcionalitats del client que fa la cerca o la reserva de restaurants.
- Subsistema Propietari del restaurant: Agruparà les funcionalitats dels propietaris dels restaurants.
- Subsistema Administrador: 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 un servidor, i els clients que faran les cerques i reserves, els propietaris i els administratius de l'empresa accediran a la web 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 web de cerques i reserves virtuals de restaurants ha de ser en arquitectura de 3 capes, usant J2EE, on es separi clarament la presentació, la lògica de negoci i l'accés a les dades. La persistència de les dades serà en BD relacional, concretament Oracle. L'escalabilitat, fiabilitat i portabilitat seran les inherents a la tecnologia J2EE.

6.1. *Subsistema Client*

Com hem dit anteriorment aquest subsistema es divideix en dues parts:

6.1.1. *Client invitat:*

1. Cerca de restaurants: En aquesta opció es mostrarà una pantalla on el client podrà emplenar un formulari un filtre per simplificar la cerca, els camps claus seran: Tipus de restaurant, Nom, Província, Ciutat o CP, entre altres. Ens apareixerà un llistat de tots els restaurants que compleixen les condicions anteriors. En el llistat apareixerà el nom del restaurant, situació, preu aproximat i dos botons, un per obtenir o ampliar la informació sobre el restaurant seleccionat i l'altre per realitzar la reserva del restaurant.
2. Obtindre més informació: Ens mostrarà tota la informació disponible sobre el restaurant a la base de dades en una finestra apart. En aquesta finestra no es podrà fer cap modificació de la informació. A més tindrem una imatge amb el mapa de google on es situa aquest restaurant. També es podrà utilitzar altres opcions de google, com "anar des de...".

6.1.2. *Client registrat:*

1. Enregistrar client: Abans que el client pugui accedir a algunes de les funcionalitats de l'aplicació, com fer una reserva, 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 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 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 informará d'aquest fet al client i li donarà la possibilitat de triar-ne una altre.
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 informará 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.

3. Cerca de restaurants: En aquesta opció es mostrarà una pantalla on el client podrà emplenar un formulari un filtre per simplificar la cerca, els camps claus seran: Tipus de restaurant, Nom, Província, Ciutat o CP, entre altres. Ens apareixerà un llistat de tots els restaurants que compleixen les condicions anteriors. En el llistat apareixerà el nom del restaurant, situació, preu aproximat i dos botons, un per obtenir o ampliar la informació sobre el restaurant seleccionat i l'altre per realitzar la reserva del restaurant.
4. Obtenir més informació: Ens mostrarà tota la informació disponible sobre el restaurant a la base de dades en una finestra apart. En aquesta finestra no es podrà fer cap modificació de la informació. A més tindrem una imatge amb el mapa de google on es situa aquest restaurant. També es podrà utilitzar altres opcions de google, com "anar des de...".
5. Realitzar reserva: Al accedir a aquesta pantalla podrem seleccionar el dia i hora que volem fer la reserva.
6. Consultar reserva: Quan el client prem sobre l'opció consultar reserva se li mostra una llista de les reserves que té fetes. Se enviarà un mail al client amb les dades de la reserva.
7. Anul·lar reserva: En la pantalla anterior hi haurà una opció de anul·lar la reserva seleccionada, se'ns mostra un missatge com a que la reserva està anul·lada i també s'enviarà un mail.
8. Realitzar comentaris i puntuar. Un cop fet la reserva es podrà puntuar el restaurant, d'una a cinc estrelles. I també fer algun comentari.

6.2. Subsistema Propietari del restaurant

1. Enregistrar propietari del restaurant: 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 propietari del restaurant en cas contrari. En aquest segon cas, se'ns mostrarà una pantalla on el propietari 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.
2. Accedir com a propietari del restaurant al subsistema: Com he dit a dalt, si el propietari del restaurant 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.

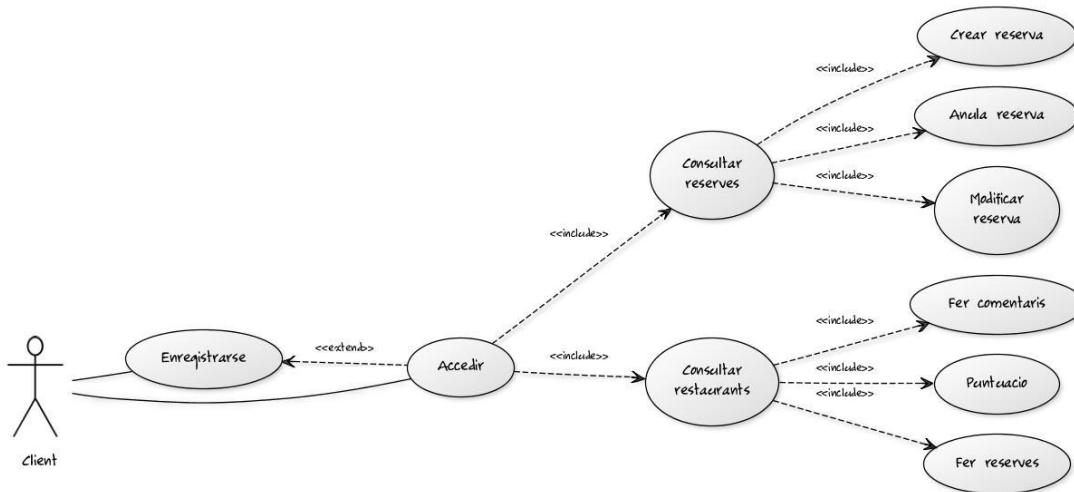
3. Manteniment un restaurant: El propietari del restaurant podrà fer la gestió de manteniment d'un o varis restaurants. Inclou l'alta, la baixa i la modificació d'un restaurant. A l'alta del restaurant inclourà totes les dades necessàries del restaurant, també la seva ubicació per tal de mostrar el mapa de google. A més del seu logotip i diferents imatges.
4. Consultar reserves: Els propietaris del restaurants podran consultar les reserves que tenen els seus restaurants per tal que saber quin dia vindran els clients.
5. Consultar estadístiques: També podran consultar les estadístiques que hi ha del seu restaurant.

6.3. *Subsistema Administrador*

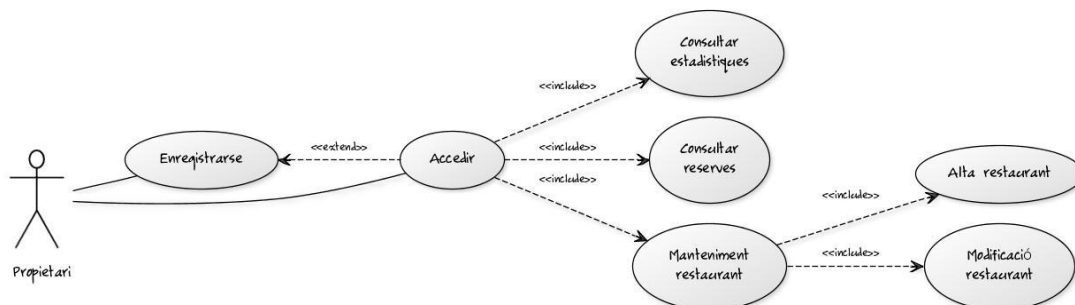
1. Accedir Administratiu a subsistema: L'administratiu tindrà una clau que el permetrà el màxims privilegis en quan a la gestió de clients, restaurants i reserves, és a dir, de tota l'aplicació. Accedirà a l'aplicació posant el seu nom d'usuari i contrasenya. Si s'equivoca en teclejar la informació el 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.
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. Manteniment de restaurants: Aquesta funcionalitat ens permetrà afegir, modificar, esborrar i consultar restaurants a la base de dades. Per afegir un restaurant l'aplicació ens mostrarà un formulari on s'hauran d'emplenar totes les dades d'aquest. Per modificar o esborrar abans s'haurà de fer una cerca dels restaurants, en el qual apareixeran les seves característiques. Des d'aquesta pantalla es podran modificar les característiques de qualsevol dels restaurants o esborrar-ho de la base de dades (deixar-l'hi en estat de "no disponible").
4. 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.
5. 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.
6. Manteniment dels comentaris: L'administrador podrà eliminar els comentaris que no siguin adequats o faltin el respecte.
7. Consultar estadístiques: També podran consultar les estadístiques que hi ha dels diferents restaurants.

6.4. Anàlisi detallat dels casos d'ús

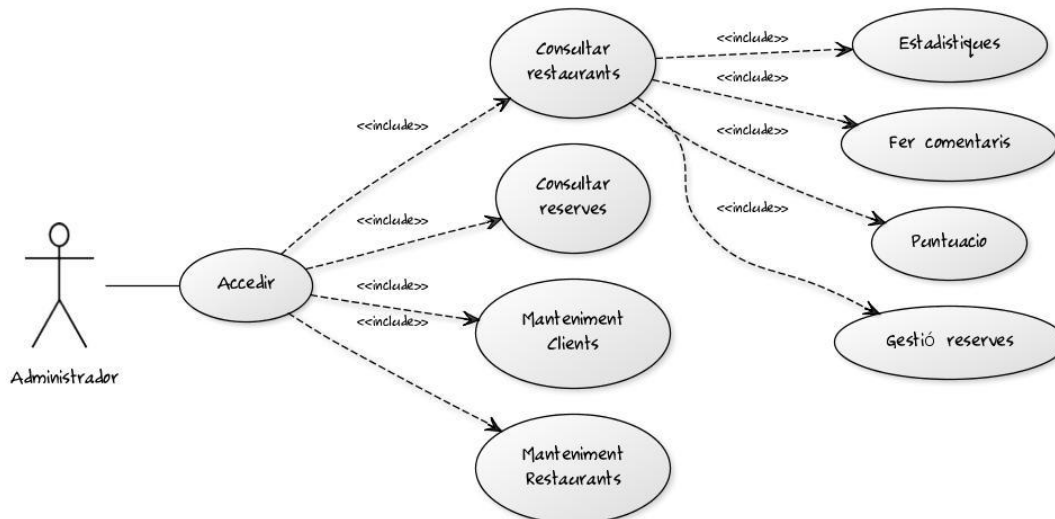
6.4.1. Descripció gràfica dels casos d'ús



Il·lustració 1: Cas d'ús Client



Il·lustració 2: Cas d'ús Propietari del Restaurant



Il·lustració 3: Cas d'ús Administrador

6.4.2. Descripció textual dels casos d'ús

6.4.2.1. Enregistrar-se

Permet al visitant obtenir un nom d'usuari únic que l'identifiqui com a Client.

Actors: Client i Propietari del restaurant.

Casos d'ús relacionats: "Accedir"

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

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

Procés normal principal:

1. L'aplicació ens mostrarà una pantalla on donarà la possibilitat de entrar al sistema, si el client ja està enregirat, o de donar-se d'alta com a client en cas contrari.
2. En aquest segon cas, se'ns mostrarà una pantalla on el client haurà de emplenar un formulari amb les seves dades, que inclouen, com a mínim, nom i cognoms, una adreça de correu electrònic, i la paraula clau per autenticar-se en el sistema.
3. L'usuari entra les dades
4. El sistema comprova les dades per garantir que l'adreça de correu electrònic sigui única en el sistema, ja que servirà com a nom d'usuari.

5. Si no hi ha conflicte, registra l'usuari a la base de dades i el redirigeix a la seva àrea privada o al destí original si aquest cas d'ús s'executa dintre d'un altre.

Alternatives de procés i excepcions:

- 3b. Si hi ha conflicte o algun error en les dades, es torna al pas 1 mostrant els missatges d'error necessaris.

6.4.2.2. *Accedir*

Permet l'accés a l'aplicació a un usuari registrat.

Actors: Client, propietari del restaurant i Administrador.

Casos d'ús relacionats: "Enregistrar-se"

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

Postcondició: L'usuari accedeix a la aplicació.

Procés normal principal:

1. Si el visitant no està autenticat, el sistema demana al visitant el seu nom d'usuari i la paraula clau. També li dona la possibilitat d'iniciar el cas d'ús "Registrar-se". Si ja està autenticat, s'ignoren els passos següents i es redirigeix l'usuari igual que en el pas 4.
2. El visitant/usuari entra les dades
3. El sistema comprova les dades.
4. Si la paraula clau correspon a l'usuari indicat, el visitant es converteix en usuari registrat i se li mostraran de forma clara totes les opcions que té disponibles.
5. Si s'equivoca en teclejar la informació el sistema informará d'aquest fet al usuari i li donará la possibilitat de tornar a provar.

Alternatives de procés i excepcions:

- 2b. Si s'inicia l'altre cas d'ús, es passa al pas 3 després de la seva execució.
- 4b. Si falla l'autenticació, es torna al pas 1 mostrant els missatges d'error necessaris. Si l'autenticació falla per tercera vegada dins d'una finestra temporal de 2 dies, es deshabilita l'usuari per precaució.

6.4.2.3. *Manteniment de Clients*

Presenta les opcions d'alta, baixa i modificació, per fer el manteniment dels clients registrats.

Actors: Administrador, Propietari del restaurant i Client.

Casos d'ús relacionats: "Alta client", "Baixa client", "Modificar client"

Precondició: L'usuari entra al sistema per procedir a una operació de manteniment dels clients que figuren a la base de dades del sistema o donar-ne d'alta un de nou.

Postcondició: L'usuari ha accedit a la part que li permet realitzar la opció de manteniment desitjada.

Procés normal principal:

1. El sistema presenta a l'usuari les opcions de manteniment de clients que són les corresponents a alta, baixa i modificació.
2. L'usuari tria l'opció desitjada.
3. El sistema presenta la pantalla corresponent a l'opció triada.

Alternatives de procés i excepcions:

- En el cas de que l'usuari que està fent servir el sistema sigui un client, només podrà actuar sobre les seves pròpies dades. Quan s'accedeixi als apartats de baixa i modificació, caldrà que l'usuari s'hagi autenticat davant del sistema.

6.4.2.4. *Alta Client*

Donar d'alta un client a la base de dades del sistema.

Actors: Administrador, Propietari del restaurant i Client.

Casos d'ús relacionats: "Manteniment client" , "Accedir".

Precondició: El client no consta a la base de dades com a registrat.

Postcondició: El client s'ha donat d'alta a la base de dades de clients registrats.

Procés normal principal:

1. El sistema demana les dades del client a inscriure.
2. L'usuari del sistema, que pot ser el propi client i propietari del restaurant o l'administrador, entra les dades del client a inscriure.
3. Les dades es guarden al sistema.

Alternatives de procés i excepcions:

- 2a. El sistema comprova que alguna de les dades entrades no es correspon amb el tipus esperat.
- 2a1. El sistema mostra un missatge d'error i demana que es corregeixin les dades errònies.
- 2a2. Es torna a mostrar la pantalla d'entrada de dades.
- 2b. El sistema comprova que el client que es vol inscriure ja consta a la base de dades.
- 2b1. El sistema mostra un missatge d'error indicant aquest fet.
- 2b2. Es torna a mostrar la pantalla d'entrada de dades.

6.4.2.5. *Baixa Client*

Donar de baixa un client a la base de dades del sistema dels clients registrats.

Actors: Administrador, Propietari del restaurant i Client.

Casos d'ús relacionats: "Manteniment client", "Accedir".

Precondició: El client consta a la base de dades com a registrat. L'usuari que vol efectuar la baixa s'ha autenticat davant del sistema.

Postcondició: El client s'ha donat de baixa a la base de dades de clients registrats.

Procés normal principal:

1. El sistema demana les dades del client a donar de baixa.
2. L'usuari del sistema, que pot ser el propi client, propietari del restaurant o l'administrador, entra les dades del client a donar de baixa.
3. S'efectua la baixa del client seleccionat.

Alternatives de procés i excepcions:

- 2a. El sistema comprova que alguna de les dades entrades no es correspon amb el tipus esperat.
- 2a1. El sistema mostra un missatge d'error i demana que es corregeixin les dades errònies.
- 2a2. Es torna a mostrar la pantalla d'entrada de dades.

- 2b. El sistema comprova que el client que es vol donar de baixa no consta a la base de dades.
 - 2b1. El sistema mostra un missatge d'error indicant aquest fet.
 - 2b2. Es torna a mostrar la pantalla d'entrada de dades.

En cas de que l'usuari sigui un client o un propietari del restaurant només podrà gestionar la seva pròpia baixa.

6.4.2.6. *Modificar Client*

Modificar un client a la base de dades del sistema dels clients registrats.

Actors: Administrador, Propietari del restaurant i Client.

Casos d'ús relacionats: "Manteniment client", "Accedir".

Precondició: El client consta a la base de dades com a registrat. L'usuari que vol efectuar la modificació s'ha autenticat davant del sistema.

Postcondició: El client s'ha modificat a la base de dades de clients registrats.

Procés normal principal:

1. El sistema demana la identificació del client del qual cal modificar les dades.
2. L'usuari del sistema, que pot ser el propi client, un propietari del restaurant o l'administrador, entra les dades del client a modificar.
3. El sistema presenta la pantalla que permet modificar les dades.
4. Les dades modificades es guarden al sistema.

Alternatives de procés i excepcions:

- 2a. El sistema comprova que alguna de les dades entrades no es correspon amb el tipus esperat.
 - 2a1. El sistema mostra un missatge d'error i demana que es corregeixin les dades errònies.
 - 2a2. Es torna a mostrar la pantalla d'entrada de dades.
- 2b. El sistema comprova que el client que es vol modificar ja consta a la base de dades.
 - 2b1. El sistema mostra un missatge d'error indicant aquest fet.
 - 2b2. Es torna a mostrar la pantalla d'entrada de dades.

En cas de que l'usuari sigui un client o un propietari del restaurant només podrà modificar les seves pròpies dades.

6.4.2.7. *Manteniment restaurant*

Presenta les opcions d'alta, baixa i modificació, per fer la gestió dels restaurants guardats a la base de dades del sistema.

Actors: Administrador i Propietari del restaurant.

Casos d'ús relacionats: “Alta restaurant”, “Baixa restaurant” i “Modificació restaurant”.

Precondició: L'usuari entra al sistema per procedir a una operació de gestió del restaurant que figuren a la base de dades del sistema o per introduir-ne un de nou.

Postcondició: L'usuari ha accedit a la part que li permet realitzar la opció de gestió desitjada.

Procés normal principal:

1. El sistema presenta a l'usuari les opcions de gestió de restaurant que són les corresponents a alta, baixa i modificació.
2. L'usuari tria l'opció desitjada.
3. El sistema presenta la pantalla corresponent a l'opció triada.

Alternatives de procés i excepcions:

Cap

6.4.2.8. *Alta restaurant*

Un client dona d'alta un restaurant.

Actors: Administrador i Propietari del restaurant.

Casos d'ús relacionats: “Manteniment restaurant”, “Accedir”.

Precondició: El restaurant no consta a la base de dades com a registrat.

Postcondició: El restaurant s'ha donat d'alta a la base de dades.

Procés normal principal:

1. El sistema demana les dades del restaurant a inscriure.

2. L'usuari del sistema, que pot ser el propi propietari del restaurant o l'administrador, entra les dades del restaurant a inscriure.
3. Les dades es guarden al sistema.

Alternatives de procés i excepcions:

- 2a. El sistema comprova que alguna de les dades entrades no es correspon amb el tipus esperat.
 - 2a1. El sistema mostra un missatge d'error i demana que es corregeixin les dades errònies.
 - 2a2. Es torna a mostrar la pantalla d'entrada de dades.
- 2b. El sistema comprova que el restaurant que es vol inscriure ja consta a la base de dades.
 - 2b1. El sistema mostra un missatge d'error indicant aquest fet.
 - 2b2. Es torna a mostrar la pantalla d'entrada de dades.

6.4.2.9. Baixa restaurant

Donar de baixa un restaurant a la base de dades del sistema.

Actors: Administrador i Propietari del restaurant.

Casos d'ús relacionats: "Manteniment restaurant", "Accedir".

Precondició: El restaurant consta a la base de dades com a registrat. L'usuari que vol efectuar la baixa s'ha autenticat davant del sistema.

Postcondició: El restaurant s'ha donat de baixa a la base de dades.

Procés normal principal:

1. El sistema demana les dades del restaurant a donar de baixa.
2. L'usuari del sistema, que pot ser el propi propietari del restaurant o l'administrador, entra les dades del restaurant a donar de baixa.
3. S'efectua la baixa del restaurant seleccionat.

Alternatives de procés i excepcions:

- 2a. El sistema comprova que alguna de les dades entrades no es correspon amb el tipus esperat.

- 2a1. El sistema mostra un missatge d'error i demana que es corregeixin les dades errònies.
- 2a2. Es torna a mostrar la pantalla d'entrada de dades.
- 2b. El sistema comprova que el restaurant que es vol donar de baixa no consta a la base de dades.
 - 2b1. El sistema mostra un missatge d'error indicant aquest fet.
 - 2b2. Es torna a mostrar la pantalla d'entrada de dades.

6.4.2.10. *Modificar restaurant*

Modificar un restaurant a la base de dades del sistema.

Actors: Administrador i Propietari del restaurant.

Casos d'ús relacionats: "Manteniment restaurant", "Accedir".

Precondició: El restaurant consta a la base de dades com a registrat. L'usuari que vol efectuar la modificació s'ha autenticat davant del sistema.

Postcondició: El restaurant s'ha modificat a la base de dades.

Procés normal principal:

1. El sistema demana la identificació del restaurant del qual cal modificar les dades.
2. L'usuari del sistema, que pot ser el propi propietari del restaurant o l'administrador, entra les dades del restaurant a modificar.
3. El sistema presenta la pantalla que permet modificar les dades.
4. Les dades modificades es guarden al sistema.

Alternatives de procés i excepcions:

- 2a. El sistema comprova que alguna de les dades entrades no es correspon amb el tipus esperat.
 - 2a1. El sistema mostra un missatge d'error i demana que es corregeixin les dades errònies.
 - 2a2. Es torna a mostrar la pantalla d'entrada de dades.
- 2b. El sistema comprova que el restaurant que es vol modificar ja consta a la base de dades.
 - 2b1. El sistema mostra un missatge d'error indicant aquest fet.
 - 2b2. Es torna a mostrar la pantalla d'entrada de dades.

En cas de que l'usuari sigui un client o un propietari del restaurant només podrà modificar les seves pròpies dades.

6.4.2.11. *Cercar restaurants*

Trobar un restaurant a la base de dades del sistema.

Actors: Administrador i Client.

Casos d'ús relacionats: “Fer reserva”. “Puntuar restaurant” i “Comentari restaurant”.

Precondició: El client consta a la base de dades com a registrat. L'usuari que vol efectuar la modificació s'ha autenticat davant del sistema.

Postcondició: Si el filtre es correcte es mostrarà els resultats de la cerca.

Procés normal principal:

1. Incloure les dades del filtre a cercar.
2. El sistema notificarà de les dades del resultat de la cerca.

Alternatives de procés i excepcions:

- 1a. El sistema comprova que alguna de les dades entrades del filtre no es correspon amb el tipus esperat.
 - 1a1. El sistema mostra un missatge d'error i demana que es corregeixin les dades errònies.
 - 1a2. Es torna a mostrar la pantalla d'entrada de dades del filtre.
- 2a. Si no hi ha cap resultat el sistema mostra un missatge d'informació.

6.4.2.12. *Gestionar reserves*

Presenta les opcions d'alta, baixa i modificació, per fer la gestió de les reserves guardades a la base de dades del sistema.

Actors: Administrador i Client.

Casos d'ús relacionats: “Alta reserva”, “Baixa reserva”, “Modificació reserva”.

Precondició: L'usuari entra al sistema per procedir a una operació de gestió de les reserves que figuren a la base de dades del sistema o per introduir-ne una de nova.

Postcondició: L'usuari ha accedit a la part que li permet realitzar la opció de gestió desitjada.

Procés normal principal:

4. El sistema presenta a l'usuari les opcions de gestió de reserves que són les corresponents a alta, baixa i modificació.
5. L'usuari tria l'opció desitjada.
6. El sistema presenta la pantalla corresponent a l'opció triada.

Alternatives de procés i excepcions:

Cap

6.4.2.13. *Alta reserva*

Un client fa una reserva.

Actors: Administrador i Client.

Casos d'ús relacionats: "Gestionar reserves".

Precondició: Hi ha disponibilitat per fer la reserva en les condicions desitjades.

Postcondició: S'enviarà un mail amb les dades de la reserva i es posarà en contacte amb el restaurant per realitzar la reserva.

Procés normal principal:

1. El sistema presenta la informació de la disponibilitat de reserves del restaurant.
2. L'usuari del sistema, que pot ser el propi client o l'administrador, selecciona la data i hora i proporciona la resta de dades necessàries per tal d'efectuar la reserva.
3. La reserva es registra al sistema.
4. El sistema genera avisos automàtics, si és el cas, tant al client com al propietari del restaurant.

Alternatives de procés i excepcions:

- 2a. El sistema comprova que alguna de les dades entrades no és vàlida.

2a1. El sistema mostra un missatge d'error i demana que es corregeixin les dades errònies.

2a2. Es torna a mostrar la pantalla d'entrada de dades.

En aquest cas es requereix que el client s'hagi autenticat davant del sistema, per evitar reserves malintencionades.

6.4.2.14. *Baixa reserva*

Un client elimina una reserva registrada a la base de dades.

Actors: Administrador i Client.

Casos d'ús relacionats: "Gestionar reserves".

Precondició: La reserva consta registrada a la base de dades.

Postcondició: S'ha efectuat la baixa de la reserva registrada a la base de dades.

Procés normal principal:

1. El sistema demana les dades del client a qui correspon la reserva.
2. L'usuari del sistema, que pot ser el propi client o l'administrador, entra les dades sol·licitades.
3. El sistema mostra les reserves que corresponen al client en qüestió.
4. L'usuari tria la reserva que vol eliminar.
5. La reserva s'elimina del sistema.

Alternatives de procés i excepcions:

Cap.

6.4.2.15. *Modificació reserva*

Un client modifica una reserva registrada a la base de dades.

Actors: Administrador i Client.

Casos d'ús relacionats: "Gestionar reserves".

Precondició: La reserva consta registrada a la base de dades.

Postcondició: S'ha efectuat la modificació de la reserva registrada a la base de dades.

Procés normal principal:

1. El sistema demana les dades del client a qui correspon la reserva.
2. L'usuari del sistema, que pot ser el propi client o l'administrador, entra les dades sol·licitades.
3. El sistema mostra les reserves que corresponen al client en qüestió.
4. L'usuari tria la reserva que vol modificar.
5. El sistema presenta la pantalla que permet modificar les dades.
6. Les dades modificades es guarden al sistema.

Alternatives de procés i excepcions:

- 2a. El sistema comprova que alguna de les dades entrades no és vàlida.
 - 2a1. El sistema mostra un missatge d'error i demana que es corregeixin les dades errònies.
 - 2a2. Es torna a mostrar la pantalla d'entrada de dades.
- 5a. El sistema comprova que alguna de les dades entrades no és vàlida.
 - 5a1. El sistema mostra un missatge d'error i demana que es corregeixin les dades errònies.
 - 5a2. Es torna a mostrar la pantalla d'entrada de dades.

6.4.2.16. *Consultar reserves*

Un client consultarà les reserves realitzades registrades a la base de dades.

Actors: Administrador, Propietari del restaurant i Client.

Casos d'ús relacionats: "Gestionar reserves".

Precondició: La reserva consta registrada a la base de dades.

Postcondició: Es mostrarà les dades de la reserva registrada a la base de dades.

Procés normal principal:

1. L'usuari del sistema, que pot ser el propi client o l'administrador, selecciona la reserva a consultar.
2. El sistema mostra les reserves que corresponen al client en qüestió.

Alternatives de procés i excepcions:

- 2a. El sistema comprova que hi hagin reserves, si no mostrarà un missatge informatiu.

6.4.2.17. *Puntuar restaurant*

Un client podrà puntuar un restaurant.

Actors: Administrador i Client.

Casos d'ús relacionats: "Cercar restaurants".

Precondició: Fer una cerca de restaurants i seleccionar-hi un.

Postcondició: La puntuació del restaurant quedarà registrada a la base de dades.

Procés normal principal:

1. El sistema presenta la pantalla que permet puntuar el restaurant.
2. L'usuari del sistema, que pot ser el propi client o l'administrador, entra les dades sol·licitades.
3. El sistema presenta la pantalla que permet modificar les dades.
4. Les dades modificades es guarden al sistema.

Alternatives de procés i excepcions:

- 2a. El sistema comprova que alguna de les dades entrades no és vàlida.
 - 2a1. El sistema mostra un missatge d'error i demana que es corregeixin les dades errònies.
 - 2a2. Es torna a mostrar la pantalla d'entrada de dades.

6.4.2.18. *Comentar restaurant*

Un client podrà incloure un comentari d'un restaurant.

Actors: Administrador i Client.

Casos d'ús relacionats: "Cercar restaurants".

Precondició: Fer una cerca de restaurants i seleccionar-hi un.

Postcondició: El comentari del restaurant quedarà registrada a la base de dades.

Procés normal principal:

1. El sistema presenta la pantalla que permet comentar el restaurant.

2. L'usuari del sistema, que pot ser el propi client o l'administrador, entra les dades sol·licitades.
3. El sistema presenta la pantalla que permet incloure un comentari.
4. Les dades del comentari es guarden al sistema.

Alternatives de procés i excepcions:

- 2a. El sistema comprova que alguna de les dades entrades no és vàlida.
 - 2a1. El sistema mostra un missatge d'error i demana que es corregeixin les dades errònies.
 - 2a2. Es torna a mostrar la pantalla d'entrada de dades.

6.4.2.19. Estadístiques

Un propietari del restaurant o l'administrador podrà veure les estadístiques d'un restaurant.

Actors: Administrador i Propietari del restaurant.

Casos d'ús relacionats: "Cercar restaurants".

Precondició: Fer una cerca de restaurants i seleccionar-hi un.

Postcondició: Es mostrarà una pantalla amb les estadístiques d'un restaurant registrat a la base de dades.

Procés normal principal:

1. El sistema presenta la pantalla les estadístiques del restaurant.

Alternatives de procés i excepcions:

- 1a. Si no hi ha dades estadístiques sobre el restaurant apareixerà un missatge informatiu.

6.5. Disseny.

En aquest apartat es recull el procés de disseny de l'aplicació per la pàgina web de cerca de restaurants.

6.5.1. Arquitectura

Tal com ja s'ha discutit anteriorment, la implementació d'una aplicació web basada en J2EE és un dels objectius principals d'aquest projecte.

Java 2 Enterprise Edition (J2EE, Sun Microsystems) que aporta tres valors clau:

- Independència de la plataforma.
- Portabilitat d'aplicacions.
- Interoperabilitat entre aplicacions.

Els objectius bàsics que s'intenten cobrir amb aquesta elecció són:

- Abstracció de tasques crítiques i repetitives mitjançant serveis amb una interfície uniforme.
- Preparació d'una infraestructura uniforme i d'una arquitectura programari basada en ella per a aplicacions empresarials.

A més amb aquesta arquitectura es cobreixen tots els requisits demanats per l'usuari de cara a la seva aplicació:

- A nivell de dades:
 - Emmagatzematge i accés a les dades d'una forma eficient mitjançant l'ocupació de sistemes de bases de dades (DBMS).
 - Mapping de dades i persistència. Amb la representació de les dades als programes (classes) i la seva correspondència amb la seva representació en la base de dades.
 - Consistència. Control d'accés concurrent a les dades.
 - Accés a dades comunes. Aïllament dels diferents accessos.
- A nivell d'usuari:

- Interacció amb l'usuari.
- Autenticació.
- Control d'accessos.

- A nivell d'aplicació:
 - Performance. Adequat temps de resposta.
 - Escalabilitat. Sistema fàcilment ampliable amb la incorporació de nous servidors. Distribució i balanceig de càrrega.
 - Disponibilitat. Seguretat enfront de caigudes, tolerància a fallades, clustering de servidors i dades.

- Disseny de programari.
 - Mantenibilitat i portabilitat.
 - Modularitat.
 - Disseny en nivells.
 - Reducció de dependències externes.

6.5.2. Alternatives de disseny

El sistema que es vol dissenyar es perfila com un sistema web en el qual es permet que l'usuari invoqui la lògica del negoci i per tant, canviï l'estat de negoci del servidor. Aquesta definició implica que almenys existeixen tres elements arquitectònics en un sistema web:

- Navegat client.
- Servidor Web.
- Servidor del Sistema.

I probablement:

- Servidor de Base de dades.

Existeixen tres patrons comuns que expressen un esquema d'organització estructural d'un sistema web:

- Client web lleuger: Es requereix que el client tingui un navegador web estàndard capaç de mostrar formularis. Tota la lògica del negoci s'executa en el servidor.
- Client web pesat: Una quantitat significativa de la lògica de negoci s'executa en la màquina del client.
- Lliura via web: A més d'usar http com a mitjà de comunicació entre el client i el servidor, s'empren altres protocols com IIOP i DCOM, per donar suport al sistema d'objectes distribuïts. El navegador web actua principalment com a lliurament i dispositiu contenidor del sistema d'objectes distribuïts.

6.5.3. Definició de l'arquitectura

Tal i com especifica el document original de SUN: http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/web-tier/web-tier5.html, la millor opció per dissenyar una aplicació empresarial que sigui mantenible i que contingui parts reusables, és la d'utilitzar el patró arquitectònic Model-View-Controller (MVC). Amb aquest patró clarament triem utilitzar un esquema de Client Web Lleuger.

Un patró arquitectònic és un patró d'alt nivell que fixa l'arquitectura global d'una aplicació. Posteriorment, el disseny farà ús de patrons de disseny per resoldre problemes específics.

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 nous 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.

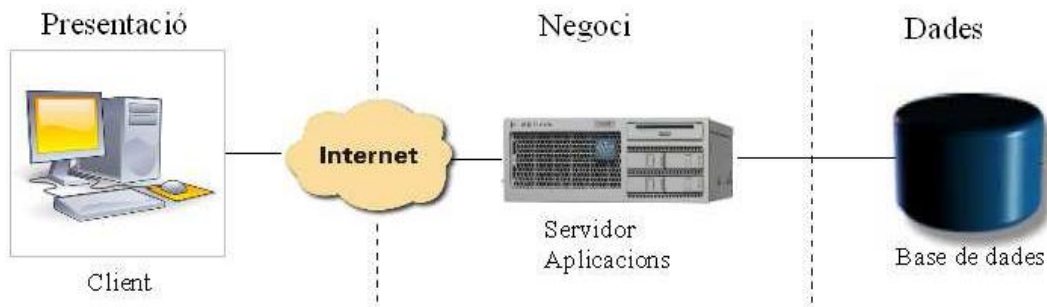
Dins l'àmbit del model trobem les dades i la funcionalitat, que ha de ser independent de les representacions o interfícies d'usuari, i del comportament de les entrades. L'àmbit de les vistes s'encarregarà de mostrar les informacions que rebí del model a l'usuari.

Cada una de les vistes que es descriuin tindrà un controlador associat que recollirà les entrades i les traduirà en peticions tant a les vistes com al model, d'aquesta manera el controlador permet que l'usuari interactuï 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, actui 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.



El patró disposa de tres components clau:

- **Model:** Responsable de la lògica de negoci.
Format per diferents nivells Model i DAO. Aquest últim, interactuarà amb la base de dades.
- **Vista:** Responsable d'una vista de presentació de la lògica de negoci.
Formada generalment per pàgines jsp i html. Html servirà contingut estàtic mentre que jsp en servirà tant d'estàtic com de dinàmic. Aplicaré pel projecte aquestes últimes per ser més complertes.
- **Controlador:** Responsable de controlar el fluxe i estat de l'entrada de dades per part de l'usuari. Intercepta peticions HTTP des d'un client, tradueix cada petició en una operació específica de negoci, invoca a l'operació de negoci en qüestió, ajuda a seleccionar la següent vista a mostrar al client i, finalment, retorna la vista al client.
El fitxer principal d'aquesta capa és struts-config.xml, fitxer que fa d'enrutador d'actions. La base de dades també mirarà de configurar-la en aquest mateix fitxer utilitzant les seves pertinents etiquetes.

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.

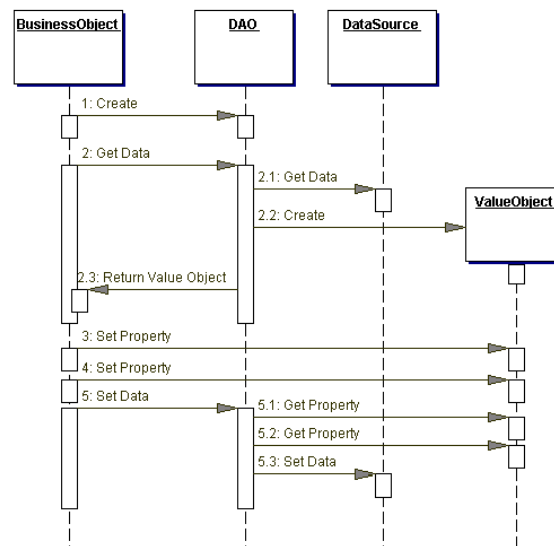
Patró DAO

En l'aplicació utilitzaré el patró DAO o Patró Objecte d'Accés a Dades.

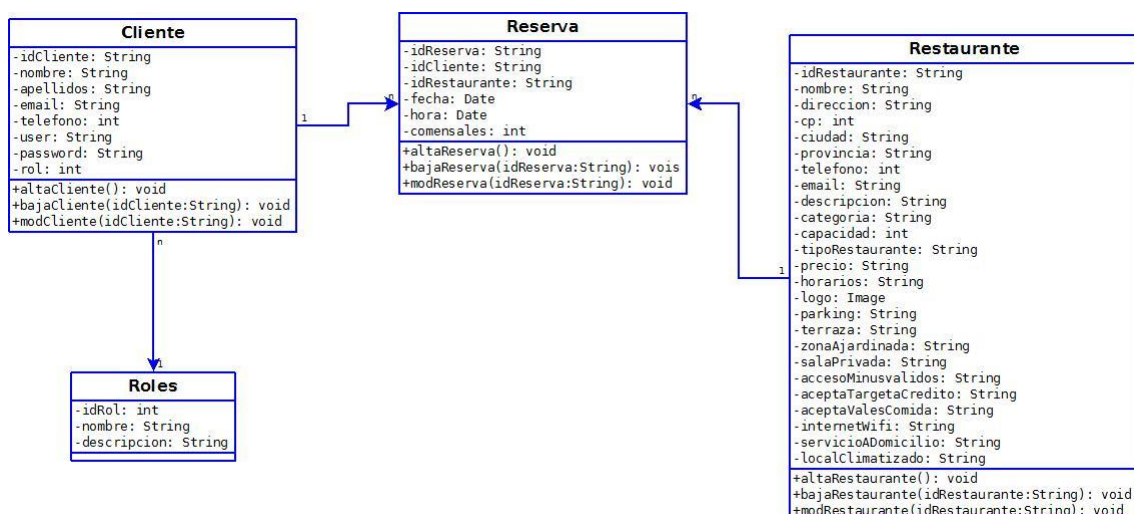
La finalitat és separar la lògica de negoci de l'aplicació de la lògica d'accés a dades.

Els participants i responsabilitats del patró queden representats segons aquest diagrama de seqüència:

- BusinessObject: en l'aplicació correspondrà a les classes de tipus Model.
- DAO: en l'aplicació correspondrà a les classes de tipus DAO.
- DataSource: en l'aplicació correspondrà a la base de dades a utilitzar.
- ValueObject: equivaldrà a tot objecte que es crearà i s'emmagatzemarà en la base de dades per a la seva posterior consulta, modificació, etc.

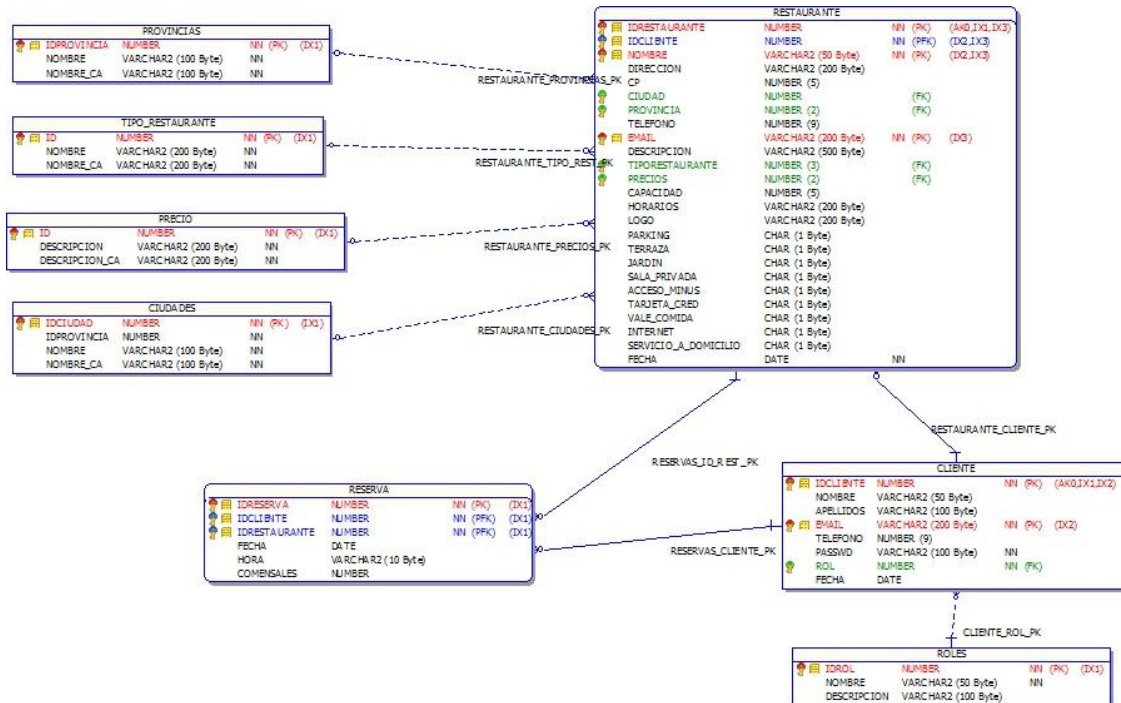


6.5.4. Diagrama de classes del model de negoci



6.5.5. Persistència

Diagrama entitat relació de la base de dades



Tenint en compte el diagrama anterior, el pas al model relacional, i les dades que realment ens interessa conservar, s'ha decidit guardar a la base de dades les següents taules:

CIUDADES(IDCIUDAD, IDPROVINCIA, NOMBRE, NOMBRE_CA)

CLIENTE (IDCLIENTE , NOMBRE, APELLIDOS , EMAIL, TELEFONO, PASSWD, ROL, FECHA)

PRECIO (ID , DESCRIPCION, DESCRIPCION_CA)

PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA)

RESERVA (IDRESERVA , IDCLIENTE , IDRESTAURANTE, FECHA , HORA, COMENSALES)

RESTAURANTE (IDRESTAURANTE, IDCLIENTE, NOMBRE, DIRECCION, CP, CIUDAD, PROVINCIA, TELEFONO, EMAIL , DESCRIPCION, TIPORESTAURANTE, PRECIOS , CAPACIDAD, HORARIOS, LOGO, PARKING , TERRAZA , JARDIN, SALA_PRIVADA , ACCESO_MINUS , TARJETA_CRED , VALE_COMIDA, INTERNET, SERVICIO_A_DOMICILIO, FECHA)

ROLES (IDROL , NOMBRE, DESCRIPCION)

TIPO_RESTAURANTE (ID , NOMBRE, NOMBRE_CA)

6.5.6. Internacionalització

Reprement 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 `es_ES`.

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.

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.

6.5.7. 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.

6.5.8. Diseny de la interfície gràfica

En definitiva la interface gràfica es dissenyarà seguint les següents pautes:

1. Dissenyar converses: Un interfície és simplement un punt de diàleg i enteniment entre un usuari i un sistema. Ens assegurarem de conèixer bé als interlocutors i ens recolzarem en normes de cortesia.
2. Feedback: La interacció és un diàleg. Informarem a l'usuari de què fer a cada moment i què succeeix amb cadascuna de les seves accions.
3. Senzillesa: no utilitzarem un estil barroc en dissenyar l'aspecte d'un interfície. Un interfície senzill serà més ràpid i fàcil d'usar.
4. Consistència i reutilització: No reinventarem la roda. L'usuari ja arriba après pel seu ús d'altres llocs, aplicacions i sistemes operatius, no li farem "estudiar" una altra vegada. Seguirem les convencions establertes per les plataformes i les bones pràctiques del disseny d'interfície.
5. Naturalitat: Intentem que imperi la senzillesa. No utilitzarem terminologies tècniques ni pedants.

6. Errors: Els evitarem i en cas que es produeixin permetrem una gestió senzilla dels mateixos mitjançant explicacions clares indicant què fer. Evitarem costis el que costi culpar a l'usuari de qualsevol error o utilitzar tecnicismes en descriure'ls.
7. Oculta la tecnologia: l'usuari ni sap ni té perquè saber tecnologia per usar un lloc. No coneix què és flaix o un applet. Farem que les coses funcionin de manera neta i fluida.
8. No facis treballar a l'usuari: Li donarem les accions més habituals en cada pantalla, tant si és novençà com a expert.
9. Interfície auto explicatiu: una pantalla ha d'explicar per si mateixa a l'usuari on es troba, què pot fer, què no ha de fer..
10. Quan tot falla, ofereix suport: Fins que l'usuari no es bloqueja no acudeix a sistemes d'ajuda o de suport. Desenvoluparem sistemes d'ajuda fàcilment accessibles i contextualitzats a la tasca que estigui desenvolupant l'usuari.
En tasques crítiques, oferirem vies de contacte.

Aspectes generals que tenim en compte en el disseny de l'aplicació web.

- **L'hipertext i la possibilitat de "navegar" per la informació**

En una pàgina web l'hipertext ens permet anar-nos desplaçant d'un document a un altre amb el simple acte de prémer sobre un enllaç. Aquesta peculiar forma de navegar per un conjunt d'informació entrelaçada pot provocar certa desorientació per part de l'usuari, ja que amb un sol pas pot haver-se desplaçat tant a una secció diferent del mateix web, com a un web totalment diferent. A més, rarament podrem incloure tota nostra informació en un sol document; haurem de fragmentar-la en diversos fitxers.

Aquestes dues circumstàncies ens obliguen a intentar estructurar tan bé com sigui possible la informació, de manera que el nostre usuari estigui sempre ben orientat sobre en quina secció es troba i entengui la relació entre la pàgina que està veient amb les altres del nostre web. Això podem aconseguir-ho amb diverses ajudes:

- Realitzant pàgines d'índex el més clares possible.

- Usant els botons de navegació que permetin a l'usuari tornar a les pàgines d'índex, a la pàgina principal o desplaçar-se a les pàgines relacionades.
 - Incloent gràfics o colors diferents segons la secció del nostre web en què ens trobem.
- **La lentitud de les xarxes de comunicacions.**

Sempre hem de ser conscients que la pàgina web que estem dissenyant es va a transmetre per una xarxa de comunicacions que no és tan ràpida com seria desitjable, que molts dels nostres usuaris han de pagar per fer aquesta transmissió i que desconeixem la potència de l'equip informàtic que posseeixen. Així, hem de cuidar que les nostres pàgines no tinguin una grandària massa gran, per facilitar la seva càrrega ràpida per la xarxa. Sol considerar-se com el màxim tolerable una grandària d'uns 40 o 50 Kb (fitxer + imatges).

En un document web el que ocupa més espai són els gràfics, per la qual cosa haurem de valorar acuradament la necessitat, quantitat i qualitat dels gràfics que posem.

També hem de tenir en compte la lentitud de la xarxa a l'hora d'estructurar la nostra informació: Cal intentar que l'usuari arribi a la informació desitjada amb el menor nombre de passos intermedis que sigui possible, per evitar que hagi de carregar fitxer després de fitxer. Per això, si bé les pàgines amb índexs són imprescindibles, no hem d'abusar d'elles posant índexs que remetin a més índexs, com de vegades podem trobar en Internet.

- **L'existència de diferents programes i versions dels programes navegadors**

Els usuaris necessiten una eina para interactua amb nosaltres, els navegadors o browsers. De moment, no existeix un programa estàndard, sinó que, per contra, els diferents fabricadors es fan la competència incloent nous avanços tècnics en successives versions dels seus productes que normalment no funcionen bé en el de la competència.

Això fa que una pàgina que nosaltres hem dissenyat veient-la amb el programa Netscape, ens doni alguna sorpresa desagradable en intentar visualitzar-la amb Internet Explorer. I viceversa. El mateix pot ocórrer-nos si veiem aquesta pàgina amb el mateix programa emprat, però en una versió inferior. Aquesta és la raó per

la qual hem de valorar acuradament la necessitat d'emprar la "última" tecnologia de disseny de pàgines www que estigui de moda en aquest moment i ser conscients que com més simple sigui nostra pàgina més possibilitats hi ha que sigui correctament visualitzada per un major nombre de lectors.

- **El monitor de l'ordinador**

Els documents web estan pensats per ser visualitzats en la pantalla d'un ordinador. El problema és que desconeixem el tipus de monitor des del qual es van a llegir les pàgines: petit o gran, de cristall líquid, en color o blanc i negre, amb resolució bona o dolenta, etc. Així, un document que hem dissenyat acuradament en el nostre monitor de 15 polzades amb una resolució 600 X 800, pot veure's "encongint" en una pantalla gran, o bé parteix del contingut "sortir-se" de la pantalla en un monitor de baixa resolució. Per aquesta raó es comprovarà l'aspecte que pren la nostra pàgina en diferents tipus de pantalles.

El fet de visualitzar el document en una pantalla també va a influir en la longitud que haurem de donar a les nostres pàgines, ja que no és convenient obligar als lectors a desplegar pàgines excessivament llargues. En tot cas és important que el màxim d'informació significativa aparegui en pantalla gens més accedir a la nostra pàgina.

- **L'actualització de la informació**

És necessari considerar prèviament la possible caducitat de la informació que anem a posar a la xarxa. Si pensem que no anem a ser capaços de mantenir actualitzada determinada pàgina, tal vegada hàgim de replantejar-nos l'oportunitat de la seva publicació. El mateix ocorre amb els comuns llistats d'enllaços recomanats: queden obsolets amb gran rapidesa. A més, és important indicar a la pròpia pàgina la data de l'última modificació, de manera que l'usuari estigui informat de la posada al dia de la mateixa.

- **Propietat intel·lectual**

Només haurem de publicar en Internet material que sigui propi o de lliure ús. En cas contrari, haurem de disposar del permís de l'autor. Tot el material publicat en Internet està protegit per la legislació sobre Propietat Intel·lectual. No obstant això, hem de tenir en compte que és molt senzill copiar qualsevol informació que posem

a la xarxa, per la qual cosa hem de valorar acuradament l'oportunitat o no de publicar algun tipus d'informació sensible o confidencial.

- **Difusió i publicitat de les pàgines**

Posar la informació en un servidor web no és garantia suficient d'aconseguir a tots els usuaris potencials desitjats. Necessitem donar difusió a l'existència de les nostres pàgines si desitgem que compleixin la seva funció. El més normal és que els nostres usuaris aconseguixin el nostre web després d'una recerca en algun dels nombrosos motors de recerca existents. Perquè apareguin les nostres pàgines reflectides en aquests cercadors, hem de tenir en compte els elements que aquests usen per localitzar informació:

- Posar clarament el títol del document.
- Introduir algunes paraules clau en l'etiqueta <meta> del llenguatge html.
- Incloure el màxim d'informació significativa possible en les primeres 25 línies de la pàgina, ja que alguns motors de recerca les usen per indexar la seva base de dades.
- Donar-nos expressament d'alta en els motors de recerca. Gairebé tots els cercadors inclouen un formulari per donar-se d'alta. Existeixen a més serveis que ens donen d'alta simultàniament en molts cercadors, encara que la majoria d'ells són de pagament.

[Aspectes concrets que tenim en compte sobre els diferents elements de les pàgines web:](#)

- **Longitud de les pàgines**

Les pàgines es realitzaran el més curtes i concises possible. Està demostrat que a la gent no li agrada llegir en la pantalla d'un ordinador: la vista es cansa i la gent es "avorreix". A més, el fet d'haver de desplegar molta quantitat de text (i/o imatges) en la pantalla, pot produir desorientació en el lector, ja que perd les referències de la capçalera i les ajudes a la navegació.

Com més important sigui una pàgina i més atenció requereixi per part dels lectors, més curta ha de ser. Si el contingut és especialment interessant, això s'ha de demostrar dins del que es veu en la primera pantalla. La longitud de la pàgina és a més una de les moltes raons que desaconsellen l'ús de tipus de lletra molt grans.

- **Els noms dels fitxers**

Una de les principals dificultats a l'hora de realitzar un document www és que nosaltres estem treballant en un PC amb l'entorn Windows, però les nostres pàgines van a residir en una màquina amb el sistema operatiu UNIX.

En UNIX són especialment importants les majúscules/minúscules, la qual cosa hem de tenir en compte a l'hora de posar el nom a un fitxer i fer un enllaç al mateix: si vam posar el nom en minúscules, haurem de fer igual els enllaços. Si no ho fem així, pot ser que els enllaços i imatges es vegin bé en el nostre PC, però no serà així en el servidor. El mateix succeeix amb les extensions dels fitxers: podem emprar ".html" o ".htm", ".jpg" o ".jpeg" però haurem de respectar això a l'hora de fer els enllaços.

Altres elements que hem d'evitar en posar noms als fitxers són:

- Caràcters especials com ñ, ç, ¿, ^, ", etc.
- Espais en blanc
- Lletres amb accents

El millor és que des del principi establim un criteri i ho seguim per a totes les pàgines relacionades: Per exemple, optar per posar tot en minúscules i amb les extensions html i jpg.

Sempre s'ha d'evitar canviar el nom als fitxers, encara que hàgim actualitzat la informació; no cal oblidar que la pàgina pot estar referenciada en diversos llocs. Si no quedés més remei que canviar-li el nom, es pot mantenir la pàgina antiga, amb una nota en la qual es que avisi que "aquesta pàgina ha canviat de lloc", donant a continuació la nova URL.

- **Pàgines amb frames o marcs**

Les pàgines amb frames o marcs permeten dividir la pantalla en diferents finestres, amb un document html diferent en cadascuna d'elles. Se solen utilitzar bastant ja que ens permeten exercir un gran control sobre la disposició general i l'aparença de la pàgina. No obstant això, han d'utilitzar-se amb cautela, ja que tenen alguns inconvenients:

- No es podran fer bookmarks o marcadors a les parts.

- Hi haurà dificultats per imprimir les pàgines individuals.
- No es poden "copiar" les URLs.
- El botó d'anterior o back dels navegadors pot no funcionar correctament.
- Redueixen l'espai útil on visualitzar la informació.
- No tots els navegadors suporten marcs.
- Podem tenir dificultats si algú vol fer un enllaç a una de les nostres pàgines: en aïllar una part de la resta dels marcs, aquesta pot perdre el sentit.

Algunes recomanacions si utilitzem pàgines amb frames:

- Eludir la fragmentació excessiva. Si es van a utilitzar més de dos frames, cal evitar la impressió de parcel·lació en múltiples finestres. Almenys una d'elles ha de ser major que les altres i actuar com a pàgina principal.
- Evitar la rigidesa de les frames (ús de les etiquetes html noresize o scrolling="no"). No hem d'impedir que els usuaris puguin "moure" les frames, ja que ho poden necessitar per tenir una resolució de pantalla diferent de la nostra.
- Enllaços exteriors al nostre web. No és convenient que quedin presoners dins de la nostra estructura de frames, ja que sol ser molt molest a causa que la nova pàgina quedarà en un espai reduït. A més, segurament tindrà un estil diferent a la nostra.
- Encara pot ser pitjor si la pàgina a la qual enllacem té al seu torn frames. Per evitar això podem utilitzar l'etiqueta html target="_top" amb el que la nova pàgina es carregarà en una pantalla completa.

- **Tipografia**

En escollir la tipografia que anem a emprar a la nostra pàgina, hem de tenir en compte que estem dissenyant un document perquè ser llegit en la pantalla d'un ordinador. Per tant, hem d'escollir tipus de lletres no gaire grans, per no fer massa deixa anar la pàgina, però tampoc excessivament petits, que puguin causar dificultats de lectura a les persones que no tinguin una bona visió.

En general és molt important una bona estructuració del text al llarg de la pàgina, emprant paràgrafs curts que facilitin la lectura i posant títols destacats en les diferents seccions del text. A més, és millor no apurar molt les vores de la pantalla de l'ordinador: les línies curtes es llegeixen amb major facilitat que les llargues.

Podem forçar això situant el text en una taula d'una sola columna i sense vores, definint que ocupi solament el 85-90 % de la pantalla.

Usarem tipus de lletres que siguin gairebé universals, com Arial o Times, ja que l'usuari solament podrà veure els tipus de lletres que té instal·lats en el seu ordinador. De gens serveix que s'utilitzin lletres rares que solament veurem nosaltres. A més, els navegadors tenen moltes opcions que poden ser configurades pel propi usuari: una d'elles és l'elecció personalitzada d'un tipus de lletra, amb el que el navegador no farà cas del tipus usat pel qual va dissenyar la pàgina. Si desitgem usar alguna tipografia especial per a un titular o logotip, haurem de convertir-ho en una imatge, amb el que garantirem la seva correcta visualització.

L'excessiu ús de majúscules dificulta la lectura. No s'han d'usar per a titulars llargs i encara menys per a blocs de text. El mateix pot dir-se de l'ús de les negretes, cursives o de l'ocupació del color: són recursos que usarem només per ressaltar paraules o parts del text. Si desitgem destacar tot un paràgraf és millor fer-ho amb un sagnat o introduint-ho centrat dins d'una taula sense vores de menor grandària que el paràgraf precedent. Podem destacar-ho encara més si ho desitgem, posant un color de fons diferent a aquesta taula.

No s'ha d'usar el subratllat per destacar un text: a les pàgines web estem acostumats al fet que les parts subratllades siguin enllaços i la gent sol prémer sobre ells esperant accedir a una altra pàgina. També hem d'evitar l'ús del "blink". És molt molest i pertorba la lectura del text. Podem combinar el text amb algunes imatges per evitar la monotonia, però hauran de ser imatges petites (que es carreguin ràpid) i trobar un bon equilibri visual entre les figures i el text.

- **Redacció d'enllaços**

La frase en la qual anem a situar l'enllaç ha de tenir significat. Fins i tot, si és possible, ha de contenir la mateixa frase que el títol del document al que es va a accedir des de l'enllaç. Per exemple:

Millor: Per a més informació, consulti el nostre Manual d'Estil.

En lloc de: Per veure el Manual d'Estil punxi aquí.

Una sola paraula és difícil de punxar i pot no tenir sentit.

Usar tota una frase per posar l'enllaç pot ser difícil d'entendre, especialment si canvia de línia.

Els llistats d'enllaços externs han de ser revisats periòdicament, ja que solen quedar-se obsolets amb gran rapidesa.

No s'han de canviar els colors estàndard dels enllaços (blau per als enllaços, violeta per als enllaços visitats), pot confondre als lectors. De la mateixa manera, no s'han d'utilitzar aquests dos colors al llarg del text, ja que la gent té tendència a punxar sobre ells.

- **Imatges**

La inclusió d'imatges a les nostres pàgines, ha de valorar-se amb detall a fi que la càrrega de la pàgina l'hi més ràpida possible. Solen considerar-se pàgines ràpides, aquelles d'una grandària no superior a uns 40 o 50 Kb (fitxer + imatges).

Atès que quanta major qualitat té una imatge, més ocupa, haurem de trobar un compromís entre la qualitat de la mateixa i la informació que es vol mostrar. Són molt rares les ocasions en què és necessari posar una imatge d'alta qualitat a les nostres pàgines. Existeixen programes de tractament de gràfics que permeten "baixar" la qualitat d'una imatge de forma raonable.

També podem jugar amb la grandària de les imatges a l'hora de llevar pes a la nostra pàgina, tractant que aquestes siguin el més petites possible. Si necessitem incloure alguna imatge gran, és millor posar a la pàgina una petita mostra de la mateixa, indicant que es pot punxar sobre ella per veure-la en grandària gran. Així, solament hauran de suportar una espera llarga aquells lectors que realment tinguin interès en veure-la. No s'ha de reduir la grandària de la imatge amb el programa amb el qual estem editant la pàgina (Netscape Composer, etc.): ens dóna la falsa impressió de que la imatge és més petita, però l'única cosa que fa és reduir la forma en què veiem la imatge, que en realitat és la mateixa. Per reduir de debò la grandària de la imatge haurem d'usar un programa de tractament de gràfics.

Es pot referènciar la mateixa imatge totes les vegades que es vulgui. No hem de sobrecarregar el servidor posant una vegada i una altra la mateixa imatge en diferents directoris: n'hi ha prou amb posar-la una vegada i referènciar la

mateixa. Això té a més l'avantatge que si un usuari ja ha carregat aquesta icona en alguna ocasió, ho conservarà en la "caché" del seu ordinador i no necessitarà carregar-la de nou, amb el que s'accelera la transmissió.

Els formats d'imatges més estesos són: GIF i JPG (o JPEG):

- El format GIF utilitza fins a un màxim de 256 colors o 64 tons de grisos (es poden usar menys, amb el que ocupa menys la imatge) i permet la possibilitat de definir fons transparents i animació de gràfics. Aquest format usa un sistema de compressió (per reduir el temps de transmissió per la xarxa) amb el qual no es perd qualitat. Per això, aquest format és apropiat per a imatges petites i amb bona resolució i per a dibuixos amb vores ben definides.
- El format JPG permet qualitats de més de 256 colors, de fet permet fins a 16 milions. El problema és que molts usuaris tenen una resolució de pantalla de sol 256 colors, amb el que pot ser que es vegin les imatges de forma defectuosa. Aquest format té un sistema de compressió que fa que la seva transmissió per la xarxa sigui més ràpida, pels quals és el format més apropiat per a imatges grans. Sense embrago, aquest sistema de compressió pot baixar la qualitat de la imatge.

Les imatges animades han d'utilitzar-se amb molta cura:

- Ocupen bastant més espai que les imatges normals.
 - Distreuen l'atenció del lector de la informació útil i acaben cansant.
 - Dificulten el saber quan ha acabat de carregar-se una pàgina.
 - Si es té oberta la pàgina web i es canvia a una finestra diferent amb una altra aplicació, el PC segueix processant la repetició de la imatge, amb el que ralenteix la velocitat de treball de l'ordinador.
 - Dificulten una impressió "neta" de la pàgina.
- **Colors i fons**

A les nostres pàgines web haurem d'anar amb compte a emprar una harmonia de colors que no pertorbi la lectura de les pàgines, procurant no emprar colors estridents o combinacions estranyes. No s'han de canviar els colors estàndard dels enllaços (blau per als enllaços, violeta per als enllaços

visitats), pot confondre als lectors. De la mateixa manera, no s'han d'utilitzar aquests dos colors al llarg del text, ja que la gent té tendència a punxar sobre ells.

El millor és utilitzar fons de colors clars i text de color fosc, ja que són tons que se solen llegir amb més comoditat, per la qual cosa sempre s'hauria de fer així en pàgines en les quals predomina el text. En el cas d'usar un color de fons molt fosc haurem d'emprar una tipografia en blanc o un altre color molt clar, amb el que s'impedeix la impressió correcta de la pàgina (poques persones tenen configurat el seu navegador perquè imprimeixi els fons).

En el cas que s'empri una imatge com a fons, han de seguir-se els mateixos consells que acabem de donar i usar textures simples i tènues, procurant no usar textures molt rugoses que es veuen malament en pantalles de baixa resolució.

7. Configuració e instal·lació

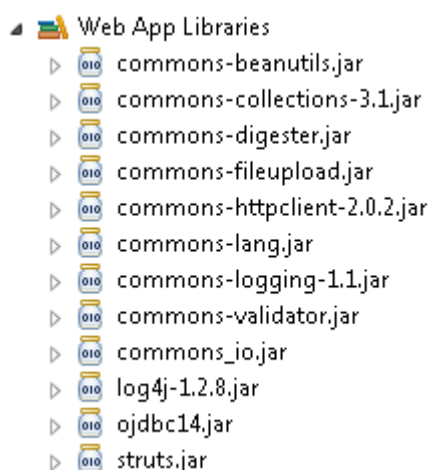
Aquest document recull i documenta els requisits del software e instal·lació per a l'aplicació.

7.1. Requeriments del software

L'aplicació necessita les següents aplicacions instal·lades previament al ordinador:

- Sistema operatiu Windows 7, es possible treballar amb altres sistemes operatius (altres versions de windows o linux), pero l'aplicació ha sigut desenvolupada amb aquest sistema operatiu.
- Motor de la base de dades: *Oracle Database Express Edition 11g Release 2 (Oracle Database XE)*. Es necessari executar el script de la creació de la BD del cercador de restaurants.
- Java Enterprise Edition, es necessari descarregar e instal·lar la JDK 5.0 i la JRE 1.6 com a mínim. El procés d'instal·lació es l'estandard.
- Rational Application Developer versió 7.5.4, amb el servidor WebSphere Application Server 6.1.

Llibreries instal·lades a l'aplicació:



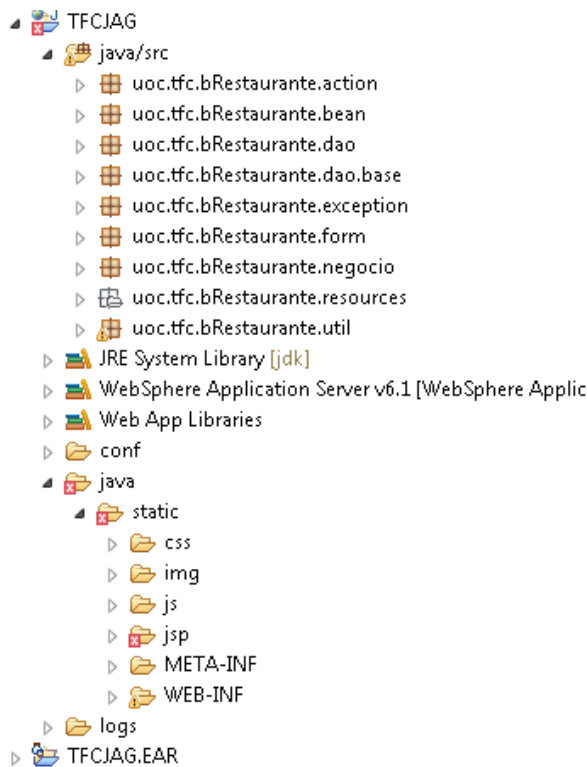
- Navegador web, aquesta aplicació ha sigut provada amb IE 8, Chrome, Firefox v.21.

7.2. L'aplicació

Aquesta es la URL per accedir a l'aplicació:

<http://localhost:9080/TFCJAG/entrada.do>

I la seva estructura queda així:



Tenim usuaris amb diferents rols:

- Usuari: ramon@ole.es amb el password 1234 té permisos del rol Restaurant amb el qual podràs crear més registres de restaurants.
- Usuari: kurtis@uoc.edu amb el password 1234 té permisos del rol Admin, no te restriccions.
- Usuari: marta@gmail.com amb el password 1234 té permisos del rol Client, podrà veure el llistat de restaurants i reservar.
- Usuari invitat: No s'ha registrat, por fer cerques o registrar-se.

Tots el restaurants estan creats a la primera provincia del combo, per fer les proves més rapides.

7.2.1. Fitxers de configuració

En aquest apartat veurem els fitxers de configuració més importants que s'han utilitzat en el desenvolupament del projecte.

El primer fitxer que es tractarà serà **web.xml**, que serà el fitxer de configuració de l'aplicació i on s'indicarà quin és el filtre que ha d'executar per a totes les peticions de l'aplicació web, l'escollidor de Tiles i la pàgina de benvinguda.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <display-name>TFCJ2EE</display-name>
  <context-param>
    <param-name>properties</param-name>
    <param-value>C:/workspace/J2EE/TFCJ2EE/conf/tfcj2ee.properties</param-value>
  </context-param>
  <context-param>
    <param-name>log4j</param-name>
    <param-value>C:/workspace/J2EE/TFCJ2EE/conf/log4j.properties</param-value>
  </context-param>
  <servlet>
    <servlet-name>action</servlet-name>
    <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
    <init-param>
      <param-name>config</param-name>
      <param-value>/WEB-INF/struts-config.xml</param-value>
    </init-param>
    <init-param>
      <param-name>debug</param-name>
      <param-value>2</param-value>
    </init-param>
    <load-on-startup>0</load-on-startup>
  </servlet>
  <!-- Principal Servlet Mapping -->
  <servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.do</url-pattern>
  </servlet-mapping>
  <!-- Taglibs para las JSP -->
  <jsp-config>
    <taglib>
      <taglib-uri>/WEB-INF/struts-bean.tld</taglib-uri>
      <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>
    </taglib>
    <taglib>
      <taglib-uri>/WEB-INF/struts-html.tld</taglib-uri>
      <taglib-location>/WEB-INF/struts-html.tld</taglib-location>
    </taglib>
    <taglib>
      <taglib-uri>/WEB-INF/struts-logic.tld</taglib-uri>
      <taglib-location>/WEB-INF/struts-logic.tld</taglib-location>
    </taglib>
  </jsp-config>
</web-app>
```

Figura . Fitxer de configuració: web.xml

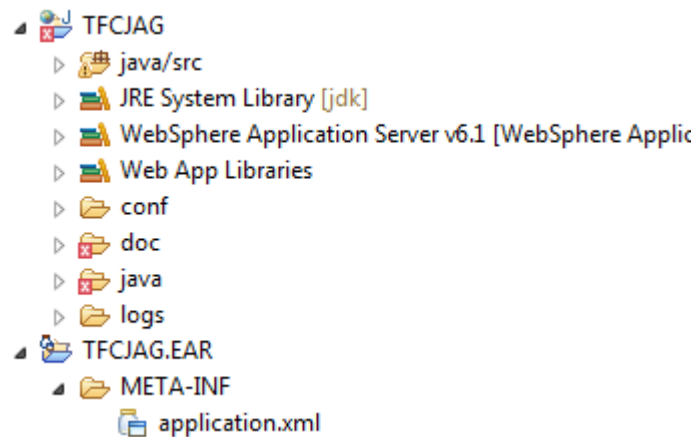
El fitxer de configuració de Struts serà **struts.xml** i s'utilitzarà per iniciar els recursos del projecte, concretament els interceptors, accions, resultats, etc.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts-config PUBLIC "-//Apache Software Foundation//DTD Struts Configuration 1.1/EN" "http://jakarta.apache.org/struts/dtds/struts-config.dtd">
<struts-config>
  <!-- ===== Data Source Configuration ===== -->
  <data-sources></data-sources>
  <!-- ===== Form Bean Definitions ===== -->
  <form-beans>
    <form-bean name="clienteForm" type="uoc.tfc.bRestaurantes.form.ClienteForm" />
    <form-bean name="restaurantesForm" type="uoc.tfc.bRestaurantes.form.RestaurantesForm" />
    <form-bean name="consultaRestaurantesForm" type="uoc.tfc.bRestaurantes.form.ConsultaRestaurantesForm" />
    <form-bean name="reservaForm" type="uoc.tfc.bRestaurantes.form.ReservaForm" />
    <form-bean name="contactoForm" type="uoc.tfc.bRestaurantes.form.ContactoForm" />
  </form-beans>
  <!-- ===== Global Exception Definitions ===== -->
  <global-exceptions>
    <exception type="java.lang.RuntimeException" scope="request" path="/jsp/errores/errorGeneral.jsp" key="error-generic" />
  </global-exceptions>
  <!-- ===== Global Forward Definitions ===== -->
  <global-forwards>
    <!-- Páginas para los errores -->
    <forward name="errorGeneral" path="/jsp/errores/errorGeneral.jsp"/>
    <forward name="errorBD" path="/jsp/errores/errorBD.jsp"/>
    <forward name="errorSesion" path="/jsp/errores/errorSesion.jsp"/>
  </global-forwards>
  <!-- ===== Action Mapping Definitions ===== -->
  <action-mappings>
    <action path="/entrada">
      <type>uoc.tfc.bRestaurantes.action.AccionInicio</type>
      <validate>"false"</validate>
      <scope>"request"</scope>
      <forward name="sucesso" path="consultaRestaurantes.do?method=inicio" redirect="false"/>
    </action>
    <action input="/login.jsp" name="clienteForm" path="/login" scope="session">
      <type>uoc.tfc.bRestaurantes.action.AccionLogin</type>
      <forward name="login" path="/jsp/login/login.jsp" redirect="false"/>
    </action>
    <action name="consultaRestaurantesForm">
      <parameter>"method"</parameter>
    </action>
  </action-mappings>
</struts-config>
```

Figura . Fitxer de configuració: struts.xml

7.2.2. Desplegament de l'aplicació

Al utilitzar el Rational per programar la web ja te un servidor incorporat, només tenim que importar el dos projectes:



Adjunto el fitxer WAR per si ho instales amb un Tomcat o Jboss.

A la carpeta "TFCJAG/doc" hi ha el JavaDoc generat automaticament amb el Rational.

8. Conclusions

L'elecció del treball fi de carrera relacionat amb J2EE va ser motivat pel meu interès en aquest tipus de desenvolupaments. Durant la carrera hem fet una sèrie de projectes en el llenguatge Java que em van motivar a escollir aquesta àrea pel meu projecte.

Al llarg de la carrera he adquirit una sèrie de coneixements que m'han ajudat especialment a la consecució d'aquest projecte. Especialment cal destacar les assignatures d'Enginyeria del Programari i Estructura de la Informació. La primera m'ha ajudat especialment per comprendre i realitzar correctament el pla de treball i l'anàlisi i disseny de l'aplicació, en canvi, la segona m'ha ajudat a realitzar la fase d'implementació del projecte.

Per realitzar el projecte de final de carrera es necessita temps, una dedicació constant i voluntat per continuar i no abandonar-ho.

Per últim, crec que he pogut aconseguir els objectius que havia plantejat en el pla de treball i que apareixen en el pla docent relatiu al treball fi de carrera. Els productes obtinguts durant la realització del projecte, és a dir, la memòria, la presentació i l'aplicació web, mostren l'assoliment d'aquestes metes.

En conclusió, el resultat final del projecte ha estat òptim. S'ha desenvolupat més d'un 90% de les pretensions inicials i s'ha complert amb totes les fites.

9. Annexos

9.1. Instruccions SQL

Les següents sentències SQL serveixen per crear la base de dades i les taules utilitzades en aquest projecte:

--Creación de un tablespace para datos y otro para índices

```
CREATE TABLESPACE "RESTDAT" LOGGING
DATAFILE 'C:\APP\ANTGARCI\ORADATA\REST\RESTDAT.dbf' SIZE 1024M
EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO
```

```
CREATE TABLESPACE "RESTIDX" LOGGING
DATAFILE 'C:\APP\ANTGARCI\ORADATA\REST\RESTIDX.dbf' SIZE 512M
EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO
```

--Creación del usuario que va a trabajar sobre estos tablespaces, y que será el propietario de los objetos que se se creen en ellos

```
CREATE USER "UOC" PROFILE "DEFAULT" IDENTIFIED BY "RESTPWD"
DEFAULT TABLESPACE "RESTDAT" TEMPORARY TABLESPACE "TEMP" ACCOUNT UNLOCK;
```

--Sólo falta asignarle los permisos necesarios para trabajar.

```
GRANT "CONNECT" TO "UOC";
GRANT "RESOURCE" TO "UOC";
```

```
GRANT ALTER ANY INDEX TO "UOC";
GRANT ALTER ANY SEQUENCE TO "UOC";
GRANT ALTER ANY TABLE TO "UOC";
GRANT ALTER ANY TRIGGER TO "UOC";
GRANT CREATE ANY INDEX TO "UOC";
GRANT CREATE ANY SEQUENCE TO "UOC";
GRANT CREATE ANY SYNONYM TO "UOC";
GRANT CREATE ANY TABLE TO "UOC";
GRANT CREATE ANY TRIGGER TO "UOC";
GRANT CREATE ANY VIEW TO "UOC";
GRANT CREATE PROCEDURE TO "UOC";
GRANT CREATE PUBLIC SYNONYM TO "UOC";
GRANT CREATE TRIGGER TO "UOC";
GRANT CREATE VIEW TO "UOC";
GRANT DELETE ANY TABLE TO "UOC";
GRANT DROP ANY INDEX TO "UOC";
GRANT DROP ANY SEQUENCE TO "UOC";
GRANT DROP ANY TABLE TO "UOC";
GRANT DROP ANY TRIGGER TO "UOC";
GRANT DROP ANY VIEW TO "UOC";
GRANT INSERT ANY TABLE TO "UOC";
GRANT QUERY REWRITE TO "UOC";
GRANT SELECT ANY TABLE TO "UOC";
GRANT UNLIMITED TABLESPACE TO "UOC";
```

-- TABLA CIUDADES:

```
ALTER TABLE UOC.CIUDADES
DROP PRIMARY KEY CASCADE;
```

```
DROP TABLE UOC.CIUDADES CASCADE CONSTRAINTS;
```

```
CREATE TABLE UOC.CIUDADES
(
  IDCIUDAD NUMBER NOT NULL,
  IDPROVINCIA NUMBER NOT NULL,
  NOMBRE VARCHAR2(100 BYTE) NOT NULL,
  NOMBRE_CA VARCHAR2(100 BYTE) NOT NULL
)
TABLESPACE RESTDAT;
```

```
CREATE UNIQUE INDEX UOC.CIUDADES_PK ON UOC.CIUDADES
(IDCIUDAD)
TABLESPACE RESTIDX;
```

```

ALTER TABLE UOC.CIUDADES ADD (
  CONSTRAINT CIUDADES_PK
  PRIMARY KEY
  (IDCIUDAD)
  USING INDEX
  TABLESPACE RESTIDX);

--SEQ CIUDADES
DROP SEQUENCE UOC.SEQ_CIUDADES;

CREATE SEQUENCE SEQ_CIUDADES
START WITH 0
INCREMENT BY 1
MINVALUE 0
MAXVALUE 999999999
NOCACHE
NOCYCLE
NOORDER;

-- Tabla CLIENTE
ALTER TABLE UOC.CLIENTE
DROP PRIMARY KEY CASCADE;

DROP TABLE UOC.CLIENTE CASCADE CONSTRAINTS;

CREATE TABLE UOC.CLIENTE
(
  IDCLIENTE NUMBER          NOT NULL,
  NOMBRE     VARCHAR2(50 BYTE),
  APELLIDOS  VARCHAR2(100 BYTE),
  EMAIL      VARCHAR2(200 BYTE) NOT NULL,
  TELEFONO   NUMBER(9),
  PASSWD     VARCHAR2(100 BYTE) NOT NULL,
  ROL        NUMBER          NOT NULL,
  FECHA      DATE
)
TABLESPACE RESTDAT;

CREATE UNIQUE INDEX UOC.CLIENTE_ID_PK ON UOC.CLIENTE
(IDCLIENTE)
TABLESPACE RESTIDX;

ALTER TABLE UOC.CLIENTE ADD (
  PRIMARY KEY
  (IDCLIENTE, EMAIL)
  USING INDEX
  TABLESPACE RESTDAT,
  CONSTRAINT CLIENTE_ID_PK
  UNIQUE (IDCLIENTE)
  USING INDEX
  TABLESPACE RESTIDX);

ALTER TABLE UOC.CLIENTE ADD (
  CONSTRAINT CLIENTE_ROL_PK
  FOREIGN KEY (ROL)
  REFERENCES UOC.ROLES (IDROL));

-- SEQ CLIENTE
DROP SEQUENCE UOC.SEQ_CLIENTE;

CREATE SEQUENCE SEQ_CLIENTE
START WITH 0
INCREMENT BY 1
MINVALUE 0
MAXVALUE 999999999
NOCACHE
NOCYCLE
NOORDER;

-- Tabla PRECIO
ALTER TABLE UOC.PRECIO
DROP PRIMARY KEY CASCADE;

DROP TABLE UOC.PRECIO CASCADE CONSTRAINTS;

```

```

CREATE TABLE UOC.PRECIO
(
  ID          NUMBER          NOT NULL,
  DESCRIPCION VARCHAR2(200 BYTE) NOT NULL,
  DESCRIPCION_CA VARCHAR2(200 BYTE) NOT NULL
)
TABLESPACE RESTDAT;

CREATE UNIQUE INDEX UOC.PRECIO_PK ON UOC.PRECIO
(ID)
TABLESPACE RESTIDX;

ALTER TABLE UOC.PRECIO ADD (
  CONSTRAINT PRECIO_PK
  PRIMARY KEY
  (ID)
  USING INDEX
  TABLESPACE RESTIDX);

-- Tabla PROVINCIAS
ALTER TABLE UOC.PROVINCIAS
DROP PRIMARY KEY CASCADE;

DROP TABLE UOC.PROVINCIAS CASCADE CONSTRAINTS;

CREATE TABLE UOC.PROVINCIAS
(
  IDPROVINCIA NUMBER          NOT NULL,
  NOMBRE       VARCHAR2(100 BYTE) NOT NULL,
  NOMBRE_CA    VARCHAR2(100 BYTE) NOT NULL
)
TABLESPACE RESTDAT;

CREATE UNIQUE INDEX UOC.PROVINCIAS_PK ON UOC.PROVINCIAS
(IDPROVINCIA)
TABLESPACE RESTIDX;

ALTER TABLE UOC.PROVINCIAS ADD (
  CONSTRAINT PROVINCIAS_PK
  PRIMARY KEY
  (IDPROVINCIA)
  USING INDEX
  TABLESPACE RESTIDX);

-- Tabla RESERVA
ALTER TABLE UOC.RESERVA
DROP PRIMARY KEY CASCADE;

DROP TABLE UOC.RESERVA CASCADE CONSTRAINTS;

CREATE TABLE UOC.RESERVA
(
  IDRESERVA  NUMBER          NOT NULL,
  IDCLIENTE  NUMBER          NOT NULL,
  IDRESTAURANTE NUMBER       NOT NULL,
  FECHA       DATE,
  HORA        VARCHAR2(10 BYTE),
  COMENSALES  NUMBER
)
TABLESPACE RESTDAT;

CREATE UNIQUE INDEX UOC.RESERVA_IDX_PK ON UOC.RESERVA
(IDRESERVA, IDCLIENTE, IDRESTAURANTE)
TABLESPACE RESTIDX;

ALTER TABLE UOC.RESERVA ADD (
  CONSTRAINT RESERVA_PK
  PRIMARY KEY
  (IDRESERVA, IDCLIENTE, IDRESTAURANTE));

ALTER TABLE UOC.RESERVA ADD (
  CONSTRAINT RESERVAS_CLIENTE_PK
  FOREIGN KEY (IDCLIENTE)

```

```
REFERENCES UOC.CLIENTE (IDCLIENTE),
CONSTRAINT RESERVAS_ID_REST_PK
FOREIGN KEY (IDRESTAURANTE)
REFERENCES UOC.RESTAURANTE (IDRESTAURANTE));
```

```
-- SEQ RESERVA
DROP SEQUENCE UOC.SEQ_RESERVA;
```

```
CREATE SEQUENCE SEQ_RESERVA
START WITH 0
INCREMENT BY 1
MINVALUE 0
MAXVALUE 999999999
NOCACHE
NOCYCLE
NOORDER;
```

```
-- Tabla RESTAURANTE
ALTER TABLE UOC.RESTAURANTE
DROP PRIMARY KEY CASCADE;
```

```
DROP TABLE UOC.RESTAURANTE CASCADE CONSTRAINTS;
```

```
CREATE TABLE UOC.RESTAURANTE
(
  IDRESTAURANTE      NUMBER          NOT NULL,
  IDCLIENTE          NUMBER          NOT NULL,
  NOMBRE              VARCHAR2(50 BYTE) NOT NULL,
  DIRECCION           VARCHAR2(200 BYTE),
  CP                  NUMBER(5),
  CIUDAD              NUMBER,
  PROVINCIA           NUMBER(2),
  TELEFONO            NUMBER(9),
  EMAIL              VARCHAR2(200 BYTE),
  DESCRIPCION         VARCHAR2(500 BYTE),
  TIPORESTAURANTE    NUMBER(3),
  PRECIOS             NUMBER(2),
  CAPACIDAD           NUMBER(5),
  HORARIOS            VARCHAR2(200 BYTE),
  LOGO                VARCHAR2(200 BYTE),
  PARKING             CHAR(1 BYTE),
  TERRAZA             CHAR(1 BYTE),
  JARDIN              CHAR(1 BYTE),
  SALA_PRIVADA        CHAR(1 BYTE),
  ACCESO_MINUS        CHAR(1 BYTE),
  TARJETA_CRED        CHAR(1 BYTE),
  VALE_COMIDA         CHAR(1 BYTE),
  INTERNET            CHAR(1 BYTE),
  SERVICIO_A_DOMICILIO CHAR(1 BYTE),
  FECHA               DATE            NOT NULL
)
TABLESPACE RESTDAT;
```

```
CREATE UNIQUE INDEX UOC.IDX_RESTAURANTE ON UOC.RESTAURANTE
(IDRESTAURANTE)
TABLESPACE RESTIDX;
```

```
CREATE UNIQUE INDEX UOC.IDX_RESTAURANTE_CLI_NOM ON UOC.RESTAURANTE
(IDCLIENTE, NOMBRE)
TABLESPACE RESTIDX;
```

```
ALTER TABLE UOC.RESTAURANTE ADD (
  PRIMARY KEY
  (IDRESTAURANTE, IDCLIENTE, NOMBRE, EMAIL)
  USING INDEX
  TABLESPACE RESTDAT,
  CONSTRAINT RESTAURANTE_ID_REST_PK
  UNIQUE (IDRESTAURANTE));
```

```
ALTER TABLE UOC.RESTAURANTE ADD (
  CONSTRAINT RESTAURANTE_CIUDADES_PK
  FOREIGN KEY (CIUDAD)
  REFERENCES UOC.CIUDADES (IDCIUDAD),
  CONSTRAINT RESTAURANTE_CLIENTE_PK
  FOREIGN KEY (IDCLIENTE)
  REFERENCES UOC.CLIENTE (IDCLIENTE),
  CONSTRAINT RESTAURANTE_PRECIOS_PK
```

```

FOREIGN KEY (PRECIOS)
REFERENCES UOC.PRECIO (ID),
CONSTRAINT RESTAURANTE_PROVINCIAS_PK
FOREIGN KEY (PROVINCIA)
REFERENCES UOC.PROVINCIAS (IDPROVINCIA),
CONSTRAINT RESTAURANTE_TIPO_REST_PK
FOREIGN KEY (TIPORESTAURANTE)
REFERENCES UOC.TIPO_RESTAURANTE (ID));

-- SEQ RESTAURANTE
DROP SEQUENCE UOC.SEQ_RESERVA;

CREATE SEQUENCE SEQ_RESERVA
START WITH 0
INCREMENT BY 1
MINVALUE 0
MAXVALUE 999999999
NOCACHE
NOCYCLE
NOORDER;

-- Tabla ROLES
ALTER TABLE UOC.ROLES
DROP PRIMARY KEY CASCADE;

DROP TABLE UOC.ROLES CASCADE CONSTRAINTS;

CREATE TABLE UOC.ROLES
(
  IDROL      NUMBER          NOT NULL,
  NOMBRE     VARCHAR2(50 BYTE) NOT NULL,
  DESCRIPCION VARCHAR2(100 BYTE)
)
TABLESPACE RESTDAT;

CREATE UNIQUE INDEX UOC.ROLES_PK ON UOC.ROLES
(IDROL)
TABLESPACE RESTIDX;

ALTER TABLE UOC.ROLES ADD (
  CONSTRAINT ROLES_PK
  PRIMARY KEY
  (IDROL)
  USING INDEX
  TABLESPACE RESTIDX);

-- Tabla TIPO_RESTAURANTE
ALTER TABLE UOC.TIPO_RESTAURANTE
DROP PRIMARY KEY CASCADE;

DROP TABLE UOC.TIPO_RESTAURANTE CASCADE CONSTRAINTS;

CREATE TABLE UOC.TIPO_RESTAURANTE
(
  ID      NUMBER          NOT NULL,
  NOMBRE  VARCHAR2(200 BYTE) NOT NULL,
  NOMBRE_CA VARCHAR2(200 BYTE) NOT NULL
)
TABLESPACE RESTDAT;

CREATE UNIQUE INDEX UOC.TIPO_RESTAURANTE_PK ON UOC.TIPO_RESTAURANTE
(ID)
TABLESPACE RESTIDX;

ALTER TABLE UOC.TIPO_RESTAURANTE ADD (
  CONSTRAINT TIPO_RESTAURANTE_PK
  PRIMARY KEY
  (ID)
  USING INDEX
  TABLESPACE RESTIDX);

-- insertamos provincias:
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (1, 'ALAVA', 'ÀLABA');

```

```

INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (2, 'ALBACETE', 'ALBACETE');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (3, 'ALICANTE', 'ALACANT');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (4, 'ALMERIA', 'ALMERIA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (5, 'ASTURIAS', 'ASTÚRIES');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (6, 'AVILA', 'ÀVILA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (7, 'BADAJOZ', 'BADAJOZ');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (8, 'BARCELONA', 'BARCELONA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (9, 'BURGOS', 'BURGOS');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (10, 'CACERES', 'CÁCERES');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (11, 'CADIZ', 'CADIS');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (12, 'CANTABRIA', 'CANTÀBRIA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (13, 'CASTELLON', 'CASTELLÓ');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (14, 'CIUDAD REAL', 'CIUDAD REAL');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (15, 'CORDOBA', 'CÓRDOVA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (16, 'CORUÑA, A', 'CORUÑA, A');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (17, 'CUENCA', 'CUENCA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (18, 'GERONA', 'GIRONA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (19, 'GRANADA', 'GRANADA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (20, 'GUADALAJARA', 'GUADALAJARA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (21, 'GUIPUZCOA', 'GUIPÚSCOA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (22, 'HUELVA', 'HUELVA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (23, 'HUESCA', 'HUESCA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (24, 'ISLAS BALEARES', 'ILLES BALEARS');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (25, 'JAEN', 'JAÉN');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (26, 'LEON', 'LEÓN');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (27, 'LERIDA', 'LLEIDA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (28, 'LUGO', 'LUGO');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (29, 'MADRID', 'MADRID');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (30, 'MALAGA', 'MÀLAGA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (31, 'MURCIA', 'MURCIA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (32, 'NAVARRA', 'NAVARRA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (33, 'ORENSE', 'ORENSE');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (34, 'PALENCIA', 'PALENCIA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (35, 'LAS PALMAS', 'LAS PALMAS');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (36, 'PONTEVEDRA', 'PONTEVEDRA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (37, 'LA RIOJA', 'LA RIOJA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (38, 'SALAMANCA', 'SALAMANCA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (39, 'SEGOVIA', 'SEGÒVIA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (40, 'SEVILLA', 'SEVILLA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (41, 'SORIA', 'SORIA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (42, 'TARRAGONA', 'TARRAGONA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (43, 'SANTA CRUZ DE TENERIFE', 'SANTA CRUZ DE TENERIFE');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (44, 'TERUEL', 'TERUEL');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (45, 'TOLEFO', 'TOLEDO');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (46, 'VALENCIA', 'VALÈNCIA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (47, 'VALLADOLID', 'VALLADOLID');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (48, 'VIZCAYA', 'BISCAIA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (49, 'ZAMORA', 'ZAMORA');
INSERT INTO PROVINCIAS (IDPROVINCIA, NOMBRE, NOMBRE_CA) VALUES (50, 'ZARAGOZA', 'SARAGOSSA');

```

-- Insertamos tipo de restaurantes:

```

INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (1, 'Brasería', 'Restaurant braseria');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (2, 'Restaurante Alternativo', 'Restaurant Alternatiu');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (3, 'Arrocería', 'Arrocería');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (4, 'Bar restaurante', 'Bar restaurant');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (5, 'Bodas y Banquetes', 'Noces i Banquets');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (6, 'Cafetería', 'Cafeteria');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (7, 'Cervecería', 'Cerveseria');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (8, 'De Mercado', 'De Mercat');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (9, 'Cocina casera', 'Cuina casolana');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (10, 'Fast Food', 'Fast food');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (11, 'Cocina tradicional', 'Cuina tradicional');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (12, 'Marisquería', 'Marisquería');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (13, 'Mesón', 'Fonda');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (14, 'Pizzería', 'Pizzeria');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (15, 'Taberna Vinos', 'Taverna Vins');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (16, 'Vegetariano', 'Vegetarià');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (17, 'Regional', 'Regional');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (18, 'Regional-Andaluz', 'Regional-Andalus');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (19, 'Regional-Asturiano', 'Regional-Asturià');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (20, 'Regional-Castellano', 'Regional-Castellà');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (21, 'Regional-Catalán', 'Regional-Català');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (22, 'Español', 'Español');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (23, 'Extremeño', 'Extremeny');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (24, 'Regional-Gallego', 'Regional-Gallego');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (25, 'Regional-Mediterráneo', 'Regional-Mediterrani');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (26, 'Regional-Navarro', 'Regional-Navarro');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (27, 'Regional-Vasco', 'Regional-Basc');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (28, 'Internacional', 'Internacional');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (29, 'Internacional-Árabe', 'Internacional-Àrab');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (30, 'Internacional-Alemán', 'Internacional-Alemany');

```

```

INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (31, 'Internacional-Argentino', 'Internacional-Argentí');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (32, 'Internacional-Brasil', 'Internacional-Brasil');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (33, 'Internacional-Caribe', 'Internacional-Carib');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (34, 'Internacional-Chino', 'Internacional-Xinès');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (35, 'Internacional-Francés', 'Internacional-Francès');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (36, 'Internacional-Griego', 'Internacional-Grec');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (37, 'Internacional-Internacional', 'Internacional-Internacional');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (38, 'Internacional-Indio', 'Internacional-Indi');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (39, 'Internacional-Italiano', 'Internacional-Italià');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (40, 'Internacional-Japonés', 'Internacional-Japonès');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (41, 'Internacional-Mexicano', 'Internacional-Mexicà');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (42, 'Internacional-Americano', 'Internacional-Americà');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (43, 'Internacional-Oriental', 'Internacional-Oriental');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (44, 'Internacional-Peruano', 'Internacional-Peruà');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (45, 'Internacional-Ruso', 'Internacional-Rus');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (46, 'Internacional-Tailandés', 'Internacional-Tailandès');
INSERT INTO TIPO_RESTAURANTE (ID, NOMBRE, NOMBRE_CA) VALUES (47, 'Internacional-Turco', 'Internacional-Turc');

```

-- Insertamos tipo de PRECIOS:

```

INSERT INTO PRECIO (ID, DESCRIPCION, DESCRIPCION_CA) VALUES (1, 'Entre 6 y 12 €', 'Entre 6 i 12 €');
INSERT INTO PRECIO (ID, DESCRIPCION, DESCRIPCION_CA) VALUES (2, 'Entre 12 y 20 €', 'Entre 12 i 20 €');
INSERT INTO PRECIO (ID, DESCRIPCION, DESCRIPCION_CA) VALUES (3, 'Entre 20 y 30 €', 'Entre 20 i 30 €');
INSERT INTO PRECIO (ID, DESCRIPCION, DESCRIPCION_CA) VALUES (4, 'Entre 30 y 50 €', 'Entre 30 i 50 €');
INSERT INTO PRECIO (ID, DESCRIPCION, DESCRIPCION_CA) VALUES (5, 'Más de 50 €', 'Més de 50 €');
INSERT INTO PRECIO (ID, DESCRIPCION, DESCRIPCION_CA) VALUES (6, 'No tiene', 'No te');
INSERT INTO PRECIO (ID, DESCRIPCION, DESCRIPCION_CA) VALUES (7, 'A consultar', 'A consultar');

```

-- Insertamos ROLES

```

INSERT INTO ROLES (IDROL, NOMBRE, DESCRIPCION) VALUES (1, 'ADMIN', 'Cuenta Administrador');
INSERT INTO ROLES (IDROL, NOMBRE, DESCRIPCION) VALUES (2, 'CLIENTE', 'Cuenta Cliente');
INSERT INTO ROLES (IDROL, NOMBRE, DESCRIPCION) VALUES (3, 'PROPIETARIO', 'Cuenta Propietario del restaurante');
INSERT INTO ROLES (IDROL, NOMBRE, DESCRIPCION) VALUES (4, 'INVITADO', 'Cuenta Invitado');

```

-- Insertamos CLIENTES

```

Insert into UOC.CLIENTE (IDCLIENTE, NOMBRE, APELLIDOS, EMAIL, TELEFONO, PASSWD, ROL, FECHA) Values (SEQ_CLIENTE.nextval, 'Jose', 'García', 'kurtis@uoc.edu', 620001122, '1234', 1, TO_DATE('06/02/2013 12:41:27', 'MM/DD/YYYY HH24:MI:SS'));
Insert into UOC.CLIENTE (IDCLIENTE, NOMBRE, APELLIDOS, EMAIL, TELEFONO, PASSWD, ROL, FECHA) Values (SEQ_CLIENTE.nextval, 'Marta', 'Pérez', 'marta@gmail.com', 650650650, '1234', 2, TO_DATE('06/02/2013 12:41:27', 'MM/DD/YYYY HH24:MI:SS'));
Insert into UOC.CLIENTE (IDCLIENTE, NOMBRE, APELLIDOS, EMAIL, TELEFONO, PASSWD, ROL, FECHA) Values (SEQ_CLIENTE.nextval, 'Pepe', 'Molina', 'k@ole.es', 650650650, '1234', 2, TO_DATE('05/29/2013 19:29:38', 'MM/DD/YYYY HH24:MI:SS'));
Insert into UOC.CLIENTE (IDCLIENTE, NOMBRE, APELLIDOS, EMAIL, TELEFONO, PASSWD, ROL, FECHA) Values (SEQ_CLIENTE.nextval, 'Jaime', 'Llull', 'jaime@gmail.com', 654654654, '1234', 2, TO_DATE('06/02/2013 12:41:27', 'MM/DD/YYYY HH24:MI:SS'));
Insert into UOC.CLIENTE (IDCLIENTE, NOMBRE, APELLIDOS, EMAIL, TELEFONO, PASSWD, ROL, FECHA) Values (SEQ_CLIENTE.nextval, 'Ramon', 'Hidalgo', 'ramon@ole.es', 654321654, '1234', 3, TO_DATE('06/02/2013 13:01:04', 'MM/DD/YYYY HH24:MI:SS'));
Insert into UOC.CLIENTE (IDCLIENTE, NOMBRE, APELLIDOS, EMAIL, TELEFONO, PASSWD, ROL, FECHA) Values (SEQ_CLIENTE.nextval, 'Gillermo', 'Lobato', 'kurtis@uoc.edu', 620001122, '1234', 1, TO_DATE('06/02/2013 12:41:27', 'MM/DD/YYYY HH24:MI:SS'));

```

--Insertar restaurantes

```

INSERT INTO RESTAURANTE (IDRESTAURANTE, IDCLIENTE, NOMBRE, DIRECCION, CP, CIUDAD, PROVINCIA, TELEFONO, EMAIL, DESCRIPCION, TIPORESTAURANTE, PRECIOS, CAPACIDAD, HORARIOS, PARKING, TERRAZA, JARDIN, SALA_PRIVADA, ACCESO_MINUS, TARJETA_CRED, VALE_COMIDA, INTERNET, SERVICIO_A_DOMICILIO, FECHA)
VALUES (SEQ_RESTAURANTE.nextVal, 1, 'restaurante ejemplo', 'restaurante ejemplo', 12121, 319, 1, 96969696, '1@ole.es', 'restaurante ejemplo1', 1, 1, 100, '10', 'S', 'N', 'N', 'N', 'N', 'N', 'S', 'N', 'N', sysdate);

```

```

INSERT INTO RESTAURANTE (IDRESTAURANTE, IDCLIENTE, NOMBRE, DIRECCION, CP, CIUDAD, PROVINCIA, TELEFONO, EMAIL, DESCRIPCION, TIPORESTAURANTE, PRECIOS, CAPACIDAD, HORARIOS, PARKING, TERRAZA, JARDIN, SALA_PRIVADA, ACCESO_MINUS, TARJETA_CRED, VALE_COMIDA, INTERNET, SERVICIO_A_DOMICILIO, FECHA)
VALUES (SEQ_RESTAURANTE.nextVal, 1, 'restaurante ejemplo2', 'restaurante ejemplo2', 12121, 319, 1, 96969696, '2@ole.es', 'restaurante ejemplo2', 1, 1, 100, '10', 'S', 'N', 'N', 'N', 'N', 'N', 'S', 'N', 'N', sysdate);

```

```

INSERT INTO RESTAURANTE (IDRESTAURANTE, IDCLIENTE, NOMBRE, DIRECCION, CP, CIUDAD, PROVINCIA, TELEFONO, EMAIL, DESCRIPCION, TIPORESTAURANTE, PRECIOS, CAPACIDAD, HORARIOS, PARKING, TERRAZA, JARDIN, SALA_PRIVADA, ACCESO_MINUS, TARJETA_CRED, VALE_COMIDA, INTERNET, SERVICIO_A_DOMICILIO, FECHA)
VALUES (SEQ_RESTAURANTE.nextVal, 1, 'restaurante ejemplo3', 'restaurante ejemplo3', 12121, 319, 1, 96969696, '3@ole.es', 'restaurante ejemplo3', 1, 1, 100, '10', 'S', 'N', 'N', 'N', 'N', 'N', 'S', 'N', 'N', sysdate);

```

```

INSERT INTO RESTAURANTE (IDRESTAURANTE, IDCLIENTE, NOMBRE, DIRECCION, CP, CIUDAD, PROVINCIA, TELEFONO, EMAIL, DESCRIPCION, TIPORESTAURANTE, PRECIOS, CAPACIDAD, HORARIOS, PARKING, TERRAZA, JARDIN, SALA_PRIVADA, ACCESO_MINUS, TARJETA_CRED, VALE_COMIDA, INTERNET, SERVICIO_A_DOMICILIO, FECHA)
VALUES (SEQ_RESTAURANTE.nextVal, 1, 'restaurante ejemplo4', 'restaurante ejemplo4', 12121, 319, 1, 96969696, '4@ole.es', 'restaurante ejemplo4', 1, 1, 100, '10', 'S', 'N', 'N', 'N', 'N', 'N', 'S', 'N', 'N', sysdate);

```

```

INSERT INTO RESTAURANTE (IDRESTAURANTE, IDCLIENTE, NOMBRE, DIRECCION, CP, CIUDAD, PROVINCIA, TELEFONO, EMAIL, DESCRIPCION, TIPORESTAURANTE, PRECIOS, CAPACIDAD, HORARIOS, PARKING, TERRAZA, JARDIN, SALA_PRIVADA, ACCESO_MINUS, TARJETA_CRED, VALE_COMIDA, INTERNET, SERVICIO_A_DOMICILIO, FECHA)
VALUES (SEQ_RESTAURANTE.nextVal, 1, 'restaurante ejemplo5', 'restaurante ejemplo5', 12121, 319, 1, 96969696, '5@ole.es', 'restaurante ejemplo5', 1, 1, 100, '10', 'S', 'N', 'N', 'N', 'N', 'N', 'S', 'N', 'N', sysdate);

```

```

INSERT INTO RESTAURANTE (IDRESTAURANTE, IDCLIENTE, NOMBRE, DIRECCION, CP, CIUDAD, PROVINCIA, TELEFONO, EMAIL, DESCRIPCION, TIPORESTAURANTE, PRECIOS, CAPACIDAD, HORARIOS, PARKING, TERRAZA, JARDIN, SALA_PRIVADA, ACCESO_MINUS, TARJETA_CRED, VALE_COMIDA, INTERNET, SERVICIO_A_DOMICILIO, FECHA)

```



```
INSERT INTO RESTAURANTE (IDRESTAURANTE, IDCLIENTE, NOMBRE, DIRECCION, CP, CIUDAD, PROVINCIA, TELEFONO, EMAIL,
DESCRIPCION, TIPORESTAURANTE, PRECIOS, CAPACIDAD, HORARIOS, PARKING, TERRAZA, JARDIN, SALA_PRIVADA, ACCESO_MINUS,
TARJETA_CRED, VALE_COMIDA, INTERNET, SERVICIO_A_DOMICILIO, FECHA)
VALUES (SEQ_RESTAURANTE.nextVal, 1, 'restaurante ejemplo20', 'restaurante ejemplo20', 12121, 319, 1, 96969696, '17@ole.es',
'restaurante ejemplo20', 1, 1, 150, '10', 'N', 'N', 'N', 'N', 'N', 'N', 'S', 'N', 'N', sysdate);
```

```
-- RESERVAS
```

```
Insert into UOC.RESERVA (IDRESERVA, IDCLIENTE, IDRESTAURANTE, FECHA, HORA, COMENSALES) Values (SEQ_RESERVA.nextVal,
12, 43, TO_DATE('06/12/2013', 'MM/DD/YYYY'), '23:00', 1);
Insert into UOC.RESERVA (IDRESERVA, IDCLIENTE, IDRESTAURANTE, FECHA, HORA, COMENSALES) Values (SEQ_RESERVA.nextVal,
12, 43, TO_DATE('06/12/2013', 'MM/DD/YYYY'), '12:00', 1);
Insert into UOC.RESERVA (IDRESERVA, IDCLIENTE, IDRESTAURANTE, FECHA, HORA, COMENSALES) Values (SEQ_RESERVA.nextVal,
12, 43, TO_DATE('06/12/2013', 'MM/DD/YYYY'), '13:00', 2);
Insert into UOC.RESERVA (IDRESERVA, IDCLIENTE, IDRESTAURANTE, FECHA, HORA, COMENSALES) Values (SEQ_RESERVA.nextVal,
12, 43, TO_DATE('06/12/2013', 'MM/DD/YYYY'), '14:00', 2);
Insert into UOC.RESERVA (IDRESERVA, IDCLIENTE, IDRESTAURANTE, FECHA, HORA, COMENSALES) Values (SEQ_RESERVA.nextVal,
12, 43, TO_DATE('06/12/2013', 'MM/DD/YYYY'), '15:00', 2);
Insert into UOC.RESERVA (IDRESERVA, IDCLIENTE, IDRESTAURANTE, FECHA, HORA, COMENSALES) Values (SEQ_RESERVA.nextVal,
12, 43, TO_DATE('06/12/2013', 'MM/DD/YYYY'), '16:00', 2);
Insert into UOC.RESERVA (IDRESERVA, IDCLIENTE, IDRESTAURANTE, FECHA, HORA, COMENSALES) Values (SEQ_RESERVA.nextVal,
12, 43, TO_DATE('06/12/2013', 'MM/DD/YYYY'), '13:00', 5);
Insert into UOC.RESERVA (IDRESERVA, IDCLIENTE, IDRESTAURANTE, FECHA, HORA, COMENSALES) Values (SEQ_RESERVA.nextVal,
12, 43, TO_DATE('06/18/2013', 'MM/DD/YYYY'), '16:00', 4);
Insert into UOC.RESERVA (IDRESERVA, IDCLIENTE, IDRESTAURANTE, FECHA, HORA, COMENSALES) Values (SEQ_RESERVA.nextVal,
12, 43, TO_DATE('06/11/2013', 'MM/DD/YYYY'), '21:00', 2);
Insert into UOC.RESERVA (IDRESERVA, IDCLIENTE, IDRESTAURANTE, FECHA, HORA, COMENSALES) Values (SEQ_RESERVA.nextVal,
11, 43, TO_DATE('06/11/2013', 'MM/DD/YYYY'), '16:00', 1);
Insert into UOC.RESERVA (IDRESERVA, IDCLIENTE, IDRESTAURANTE, FECHA, HORA, COMENSALES) Values (SEQ_RESERVA.nextVal,
12, 43, TO_DATE('06/04/2013', 'MM/DD/YYYY'), '13:00', 1);
```

Adjunt un sql de les ciutats i el de creació:



ciudades.sql

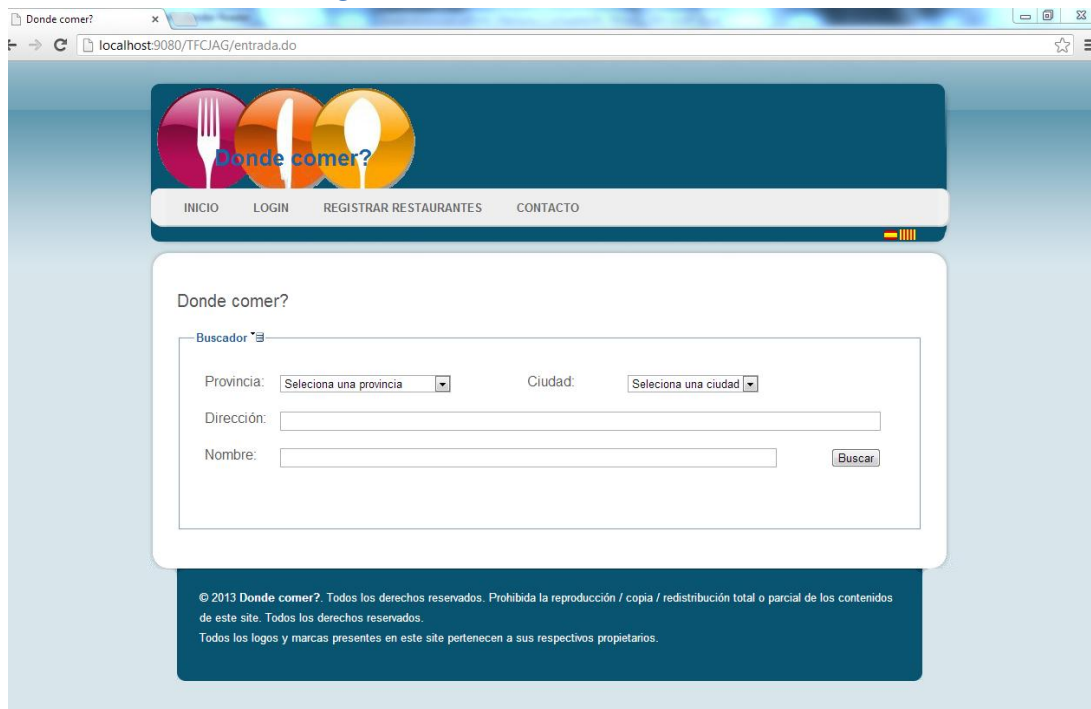


Script_Creacion_BD_REST.sql

9.2. Interfície

En aquest apartat podem veure algunes de les interfícies implementades al projecte:

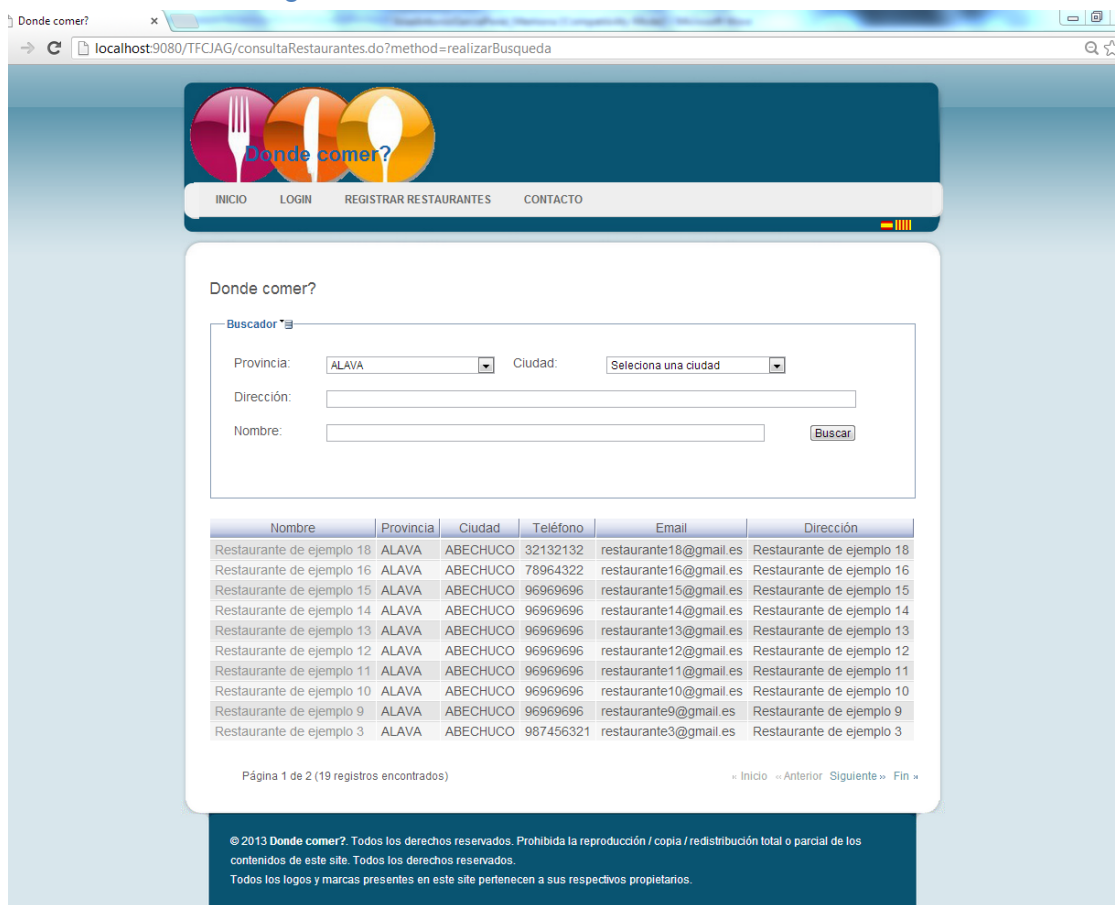
Figura . Interfície – Pantalla de inicial



Aquesta es pantalla inicial quan entrem a l'aplicació.

- Tenim el menú amb les pantalles disponibles.
- Podem realitzar una cerca simple o avançada fent un click a "Buscador".
- Veure el detall d'un restaurant.
- Sel·leccionar l'idioma.

Figura . Interfície – Pantalla de inicial amb resultats



Donde comer? x

localhost:9080/TFCJAG/consultaRestaurantes.do?method=realizarBusqueda

Donde comer?

INICIO LOGIN REGISTRAR RESTAURANTES CONTACTO

Donde comer?

Buscador

Provincia: ALAVA Ciudad: Selecciona una ciudad

Dirección:

Nombre:

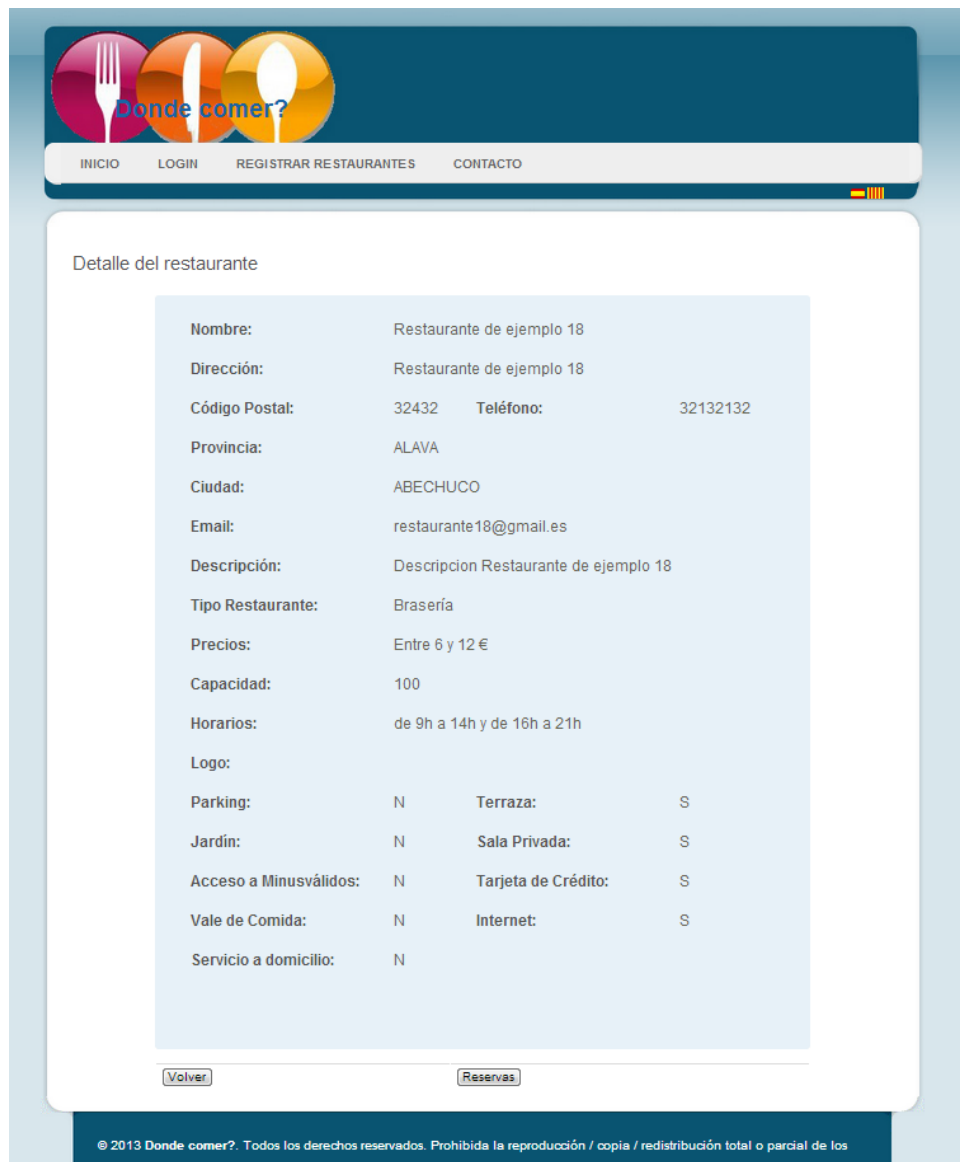
Nombre	Provincia	Ciudad	Teléfono	Email	Dirección
Restaurante de ejemplo 18	ALAVA	ABECHUCO	32132132	restaurante18@gmail.es	Restaurante de ejemplo 18
Restaurante de ejemplo 16	ALAVA	ABECHUCO	78964322	restaurante16@gmail.es	Restaurante de ejemplo 16
Restaurante de ejemplo 15	ALAVA	ABECHUCO	96969696	restaurante15@gmail.es	Restaurante de ejemplo 15
Restaurante de ejemplo 14	ALAVA	ABECHUCO	96969696	restaurante14@gmail.es	Restaurante de ejemplo 14
Restaurante de ejemplo 13	ALAVA	ABECHUCO	96969696	restaurante13@gmail.es	Restaurante de ejemplo 13
Restaurante de ejemplo 12	ALAVA	ABECHUCO	96969696	restaurante12@gmail.es	Restaurante de ejemplo 12
Restaurante de ejemplo 11	ALAVA	ABECHUCO	96969696	restaurante11@gmail.es	Restaurante de ejemplo 11
Restaurante de ejemplo 10	ALAVA	ABECHUCO	96969696	restaurante10@gmail.es	Restaurante de ejemplo 10
Restaurante de ejemplo 9	ALAVA	ABECHUCO	96969696	restaurante9@gmail.es	Restaurante de ejemplo 9
Restaurante de ejemplo 3	ALAVA	ABECHUCO	987456321	restaurante3@gmail.es	Restaurante de ejemplo 3

Página 1 de 2 (19 registros encontrados) [Inicio](#) [Anterior](#) [Siguiente](#) [Fin](#)

© 2013 Donde comer?. Todos los derechos reservados. Prohibida la reproducción / copia / redistribución total o parcial de los contenidos de este site. Todos los derechos reservados.
 Todos los logos y marcas presentes en este site pertenecen a sus respectivos propietarios.

Si fem un click al nom del restaurant accedim al detall del restaurant.

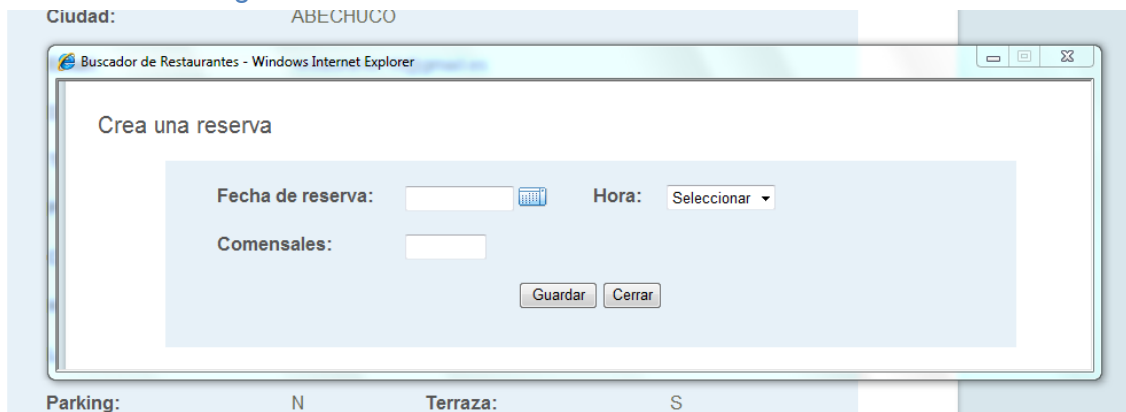
Figura . Interfície – Pantalla de detall del restaurant



Hi han dos botons:

- Tornar: Torna a la plana anterior.
- Reservas: Accedim a un popup on podem crear una reserva.

Figura . Interfície – Pantalla de creació de reserves



Ciudad: ABECHUCO

Buscador de Restaurantes - Windows Internet Explorer

Crea una reserva

Fecha de reserva: Hora: Seleccionar

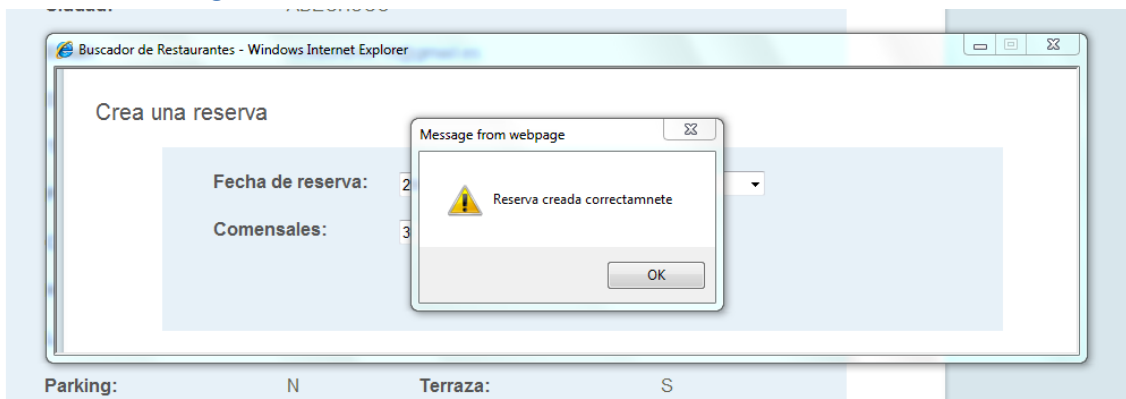
Comensales:

Guardar Cerrar

Parking: N Terraza: S

Després d'omplir les dades la reserva creada correctament:

Figura . Interfície – Pantalla de creació de reserves OK



Ciudad: ABECHUCO

Buscador de Restaurantes - Windows Internet Explorer

Crea una reserva

Fecha de reserva: 2 Hora: Seleccionar

Comensales: 3

Guardar Cerrar

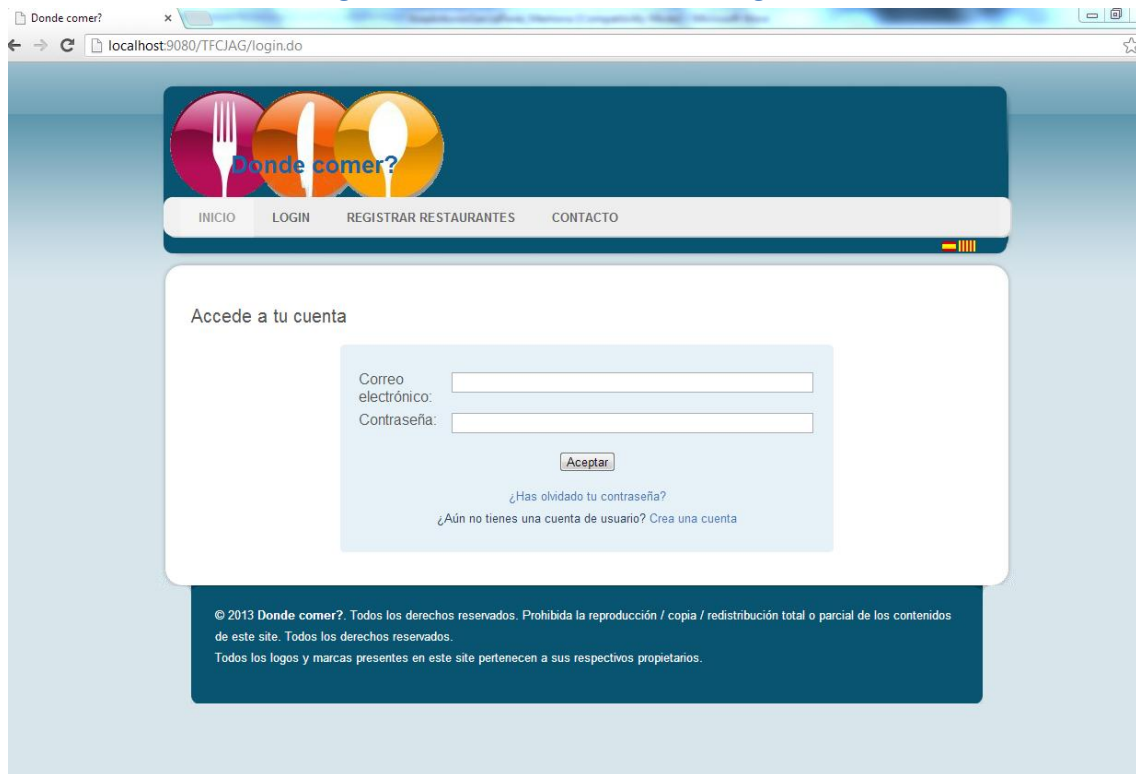
Parking: N Terraza: S

Message from webpage

Reserva creada correctamnete

OK

Figura . Interfície – Pantalla de Login



Omplir les dades accedim a la nostre compte creada, si les dades son correctes et redirecciona directament a la plana principal.

Podem recuperar el nostre password si l'hem oblidat.

Figura . Interfície – Pantalla de Login



O podem crear un compte nou si encara no estem registrats.

Figura . Interfície – Pantalla de Login



Donde comer?

INICIO LOGIN REGISTRAR RESTAURANTES CONTACTO

Accede a tu cuenta

Nombre:

Apellidos:

Correo electrónico:

Teléfono:

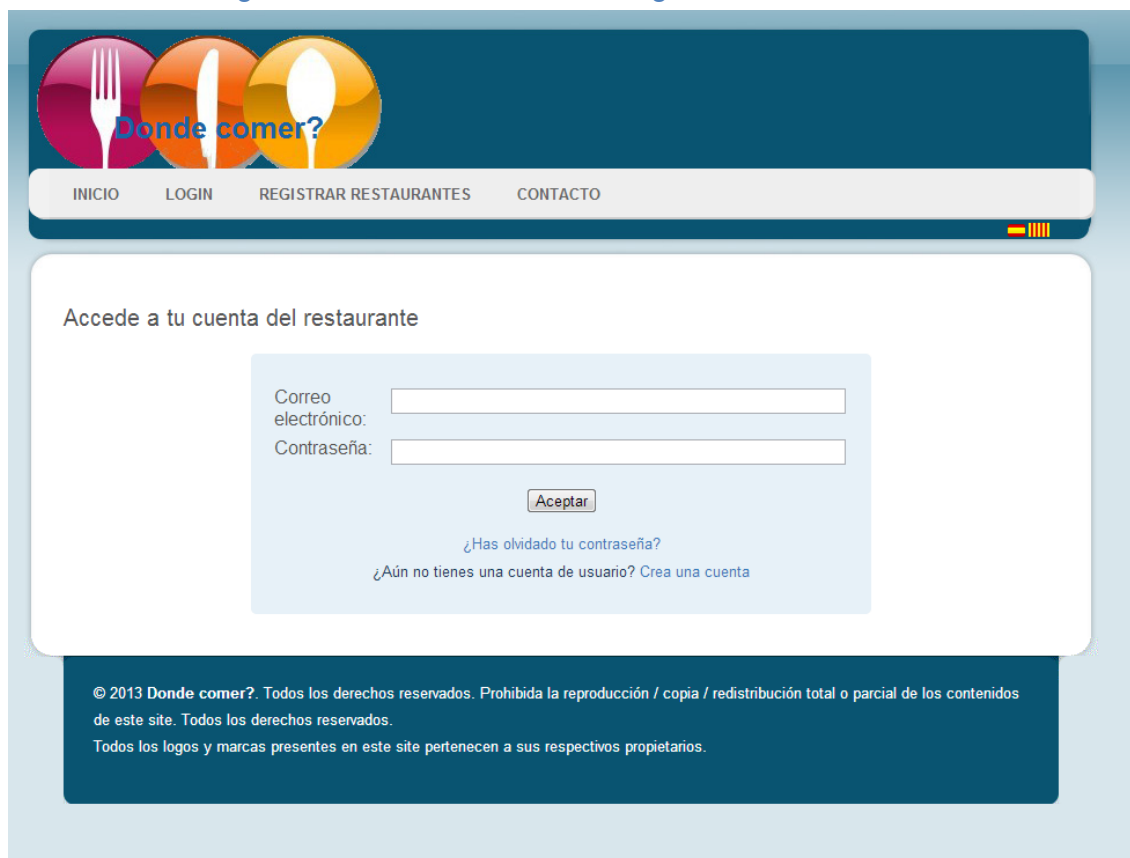
Contraseña:

Ya estás registrado, accede a tu cuenta: [login](#)

© 2013 Donde comer?. Todos los derechos reservados. Prohibida la reproducción / copia / redistribución total o parcial de los contenidos de este site. Todos los derechos reservados.
Todos los logos y marcas presentes en este site pertenecen a sus respectivos propietarios.

Si polsem el menú “Registrar restaurants” accedir a la pantalla de registrar-se si encara no ho hem fet.

Figura . Interfície – Pantalla de Login del Restaurant



Donde comer?

INICIO LOGIN REGISTRAR RESTAURANTES CONTACTO

Accede a tu cuenta del restaurante

Correo electrónico:

Contraseña:

Aceptar

[¿Has olvidado tu contraseña?](#)

[¿Aún no tienes una cuenta de usuario? Crea una cuenta](#)

© 2013 Donde comer?. Todos los derechos reservados. Prohibida la reproducción / copia / redistribución total o parcial de los contenidos de este site. Todos los derechos reservados.
Todos los logos y marcas presentes en este site pertenecen a sus respectivos propietarios.

Si ja estem registrats accedim al nostre llistat de restaurants donats d’alta.

Si no tenim cap restaurant en surt la següent pantalla:

Figura . Interfície – Pantalla de Llistat de Restaurants sense resultats



Donde comer?

INICIO LOGIN REGISTRAR RESTAURANTES CONTACTO

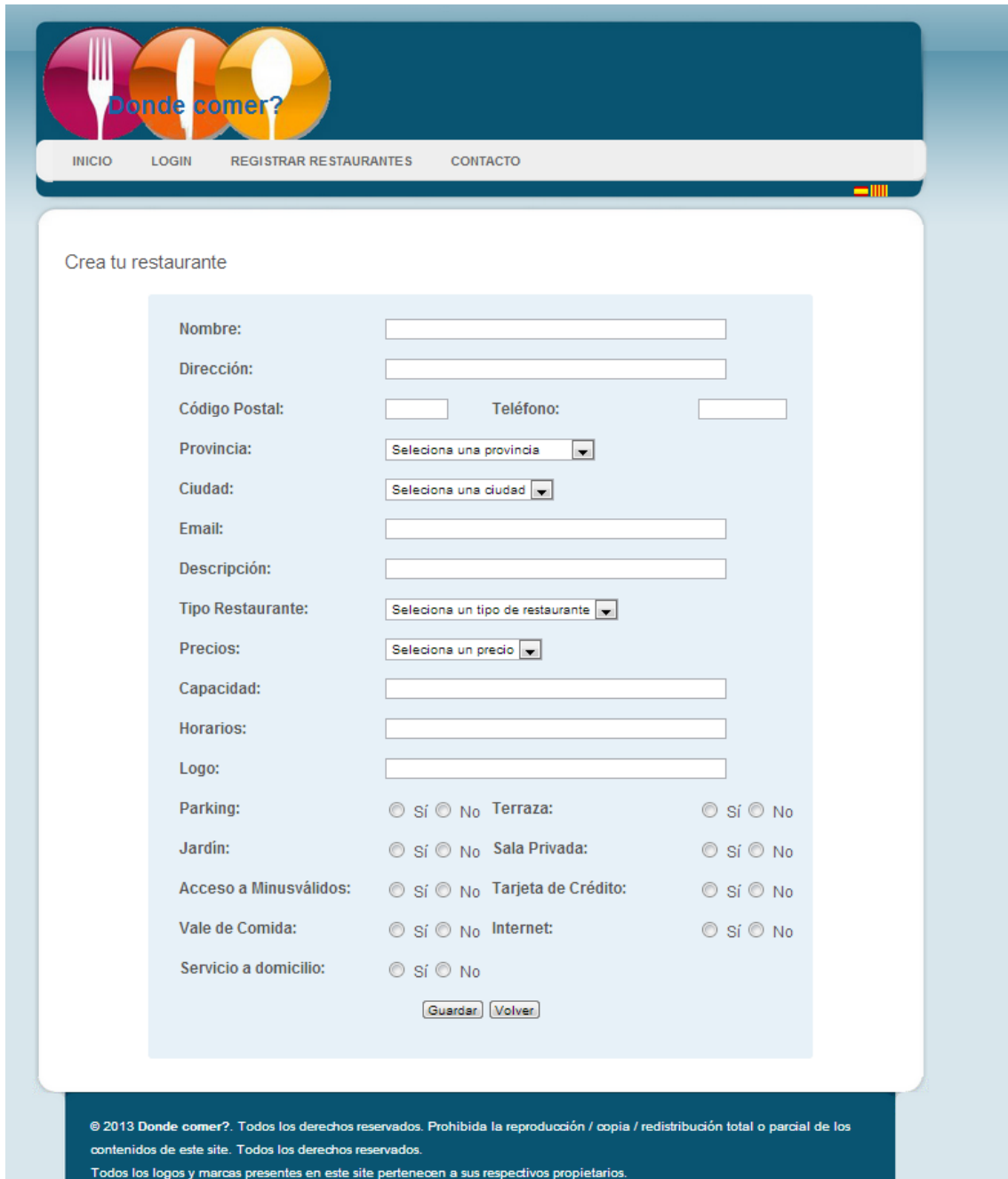
Crea tu restaurante

[¿Aún no tienes un restaurante dado de alta? Crea un restaurante](#)

© 2013 Donde comer?. Todos los derechos reservados. Prohibida la reproducción / copia / redistribución total o parcial de los contenidos de este site. Todos los derechos reservados.
Todos los logos y marcas presentes en este site pertenecen a sus respectivos propietarios.

Creem el nostre restaurant:

Figura . Interfície – Pantalla de creació de Restaurants



The screenshot shows the 'Donde comer?' website interface for creating a restaurant. The header includes navigation links: INICIO, LOGIN, REGISTRAR RESTAURANTES, and CONTACTO. The main content area is titled 'Crea tu restaurante' and contains a form with the following fields and options:

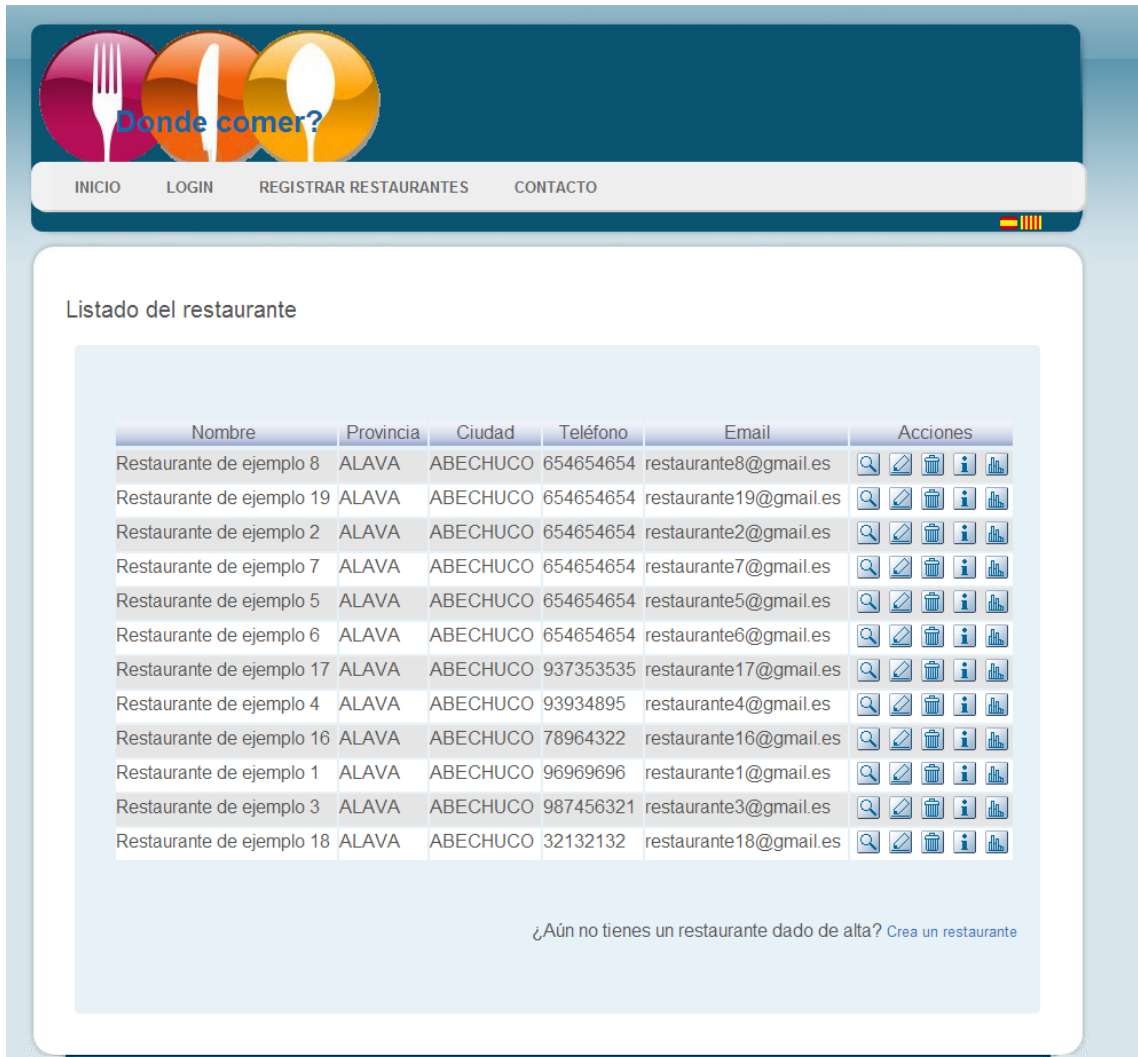
- Nombre:
- Dirección:
- Código Postal: Teléfono:
- Provincia:
- Ciudad:
- Email:
- Descripción:
- Tipo Restaurante:
- Precios:
- Capacidad:
- Horarios:
- Logo:
- Parking: Sí No
- Terraza: Sí No
- Jardín: Sí No
- Sala Privada: Sí No
- Acceso a Minusválidos: Sí No
- Tarjeta de Crédito: Sí No
- Vale de Comida: Sí No
- Internet: Sí No
- Servicio a domicilio: Sí No

At the bottom of the form are two buttons: and .

© 2013 Donde comer?. Todos los derechos reservados. Prohibida la reproducción / copia / redistribución total o parcial de los contenidos de este site. Todos los derechos reservados.
 Todos los logos y marcas presentes en este site pertenecen a sus respectivos propietarios.

Un cop donat d'alta els restaurants es mostra la llista:





























































Figura . Interfície – Pantalla de Llistat de Restaurants amb resultats



Donde comer?

INICIO LOGIN REGISTRAR RESTAURANTES CONTACTO

Listado del restaurante

Nombre	Provincia	Ciudad	Teléfono	Email	Acciones
Restaurante de ejemplo 8	ALAVA	ABECHUCO	654654654	restaurante8@gmail.es	    
Restaurante de ejemplo 19	ALAVA	ABECHUCO	654654654	restaurante19@gmail.es	    
Restaurante de ejemplo 2	ALAVA	ABECHUCO	654654654	restaurante2@gmail.es	    
Restaurante de ejemplo 7	ALAVA	ABECHUCO	654654654	restaurante7@gmail.es	    
Restaurante de ejemplo 5	ALAVA	ABECHUCO	654654654	restaurante5@gmail.es	    
Restaurante de ejemplo 6	ALAVA	ABECHUCO	654654654	restaurante6@gmail.es	    
Restaurante de ejemplo 17	ALAVA	ABECHUCO	937353535	restaurante17@gmail.es	    
Restaurante de ejemplo 4	ALAVA	ABECHUCO	93934895	restaurante4@gmail.es	    
Restaurante de ejemplo 16	ALAVA	ABECHUCO	78964322	restaurante16@gmail.es	    
Restaurante de ejemplo 1	ALAVA	ABECHUCO	96969696	restaurante1@gmail.es	    
Restaurante de ejemplo 3	ALAVA	ABECHUCO	987456321	restaurante3@gmail.es	    
Restaurante de ejemplo 18	ALAVA	ABECHUCO	32132132	restaurante18@gmail.es	    

¿Aún no tienes un restaurante dado de alta? [Crea un restaurante](#)

Als botons de les accions podem:

- Veure el detall del restaurant.
- Modificar el restaurant.
- Esborrar el restaurant.
- Veure les reserves del restaurant.
- Veure el comentaris (En construcció).


Figura . Interfície – Pantalla de detall de Restaurants

Detalle del restaurante

Nombre:	Restaurante de ejemplo 8		
Dirección:	Restaurante de ejemplo 8		
Código Postal:	89854	Teléfono:	654654654
Provincia:	ALAVA		
Ciudad:	ABECHUCO		
Email:	restaurante8@gmail.es		
Descripción:	Descripcion Restaurante de ejemplo 8		
Tipo Restaurante:	Internacional-Chino		
Precios:	Entre 12 y 20 €		
Capacidad:	200		
Horarios:	de 9h a 14h y de 16h a 21h		
Logo:			
Parking:	S	Terraza:	S
Jardin:	S	Sala Privada:	S
Acceso a Minusválidos:	S	Tarjeta de Crédito:	S
Vale de Comida:	S	Internet:	S
Servicio a domicilio:	S		

[Cerrar](#)

Figura . Interfície – Pantalla de modificació de Restaurants


Buscador de Restaurantes

[INICIO](#)
[LOGIN](#)
[REGISTRAR RESTAURANTES](#)
[CONTACTO](#)

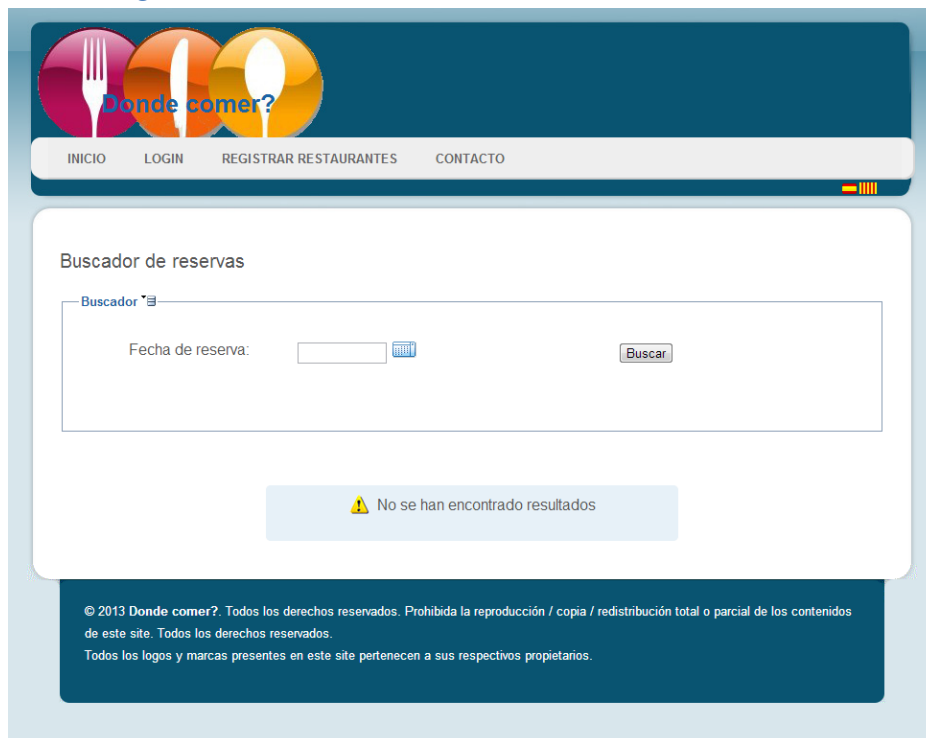
Crea tu restaurante

Nombre:	<input type="text" value="Restaurante de ejemplo 8"/>		
Dirección:	<input type="text" value="Restaurante de ejemplo 8"/>		
Código Postal:	<input type="text" value="89854"/>	Teléfono:	<input type="text" value="654654654"/>
Provincia:	<input type="text" value="ALAVA"/>		
Ciudad:	<input type="text" value="ABECHUCO"/>		
Email:	<input type="text" value="restaurante8@gmail.es"/>		
Descripción:	<input type="text" value="Descripcion Restaurante de ejemplo 8"/>		
Tipo Restaurante:	<input type="text" value="Internacional-Chino"/>		
Precios:	<input type="text" value="Entre 12 y 20 €"/>		
Capacidad:	<input type="text" value="200"/>		
Horarios:	<input type="text" value="de 9h a 14h y de 16h a 21h"/>		
Logo:	<input type="text"/>		
Parking:	<input checked="" type="radio"/> Sí <input type="radio"/> No	Terraza:	<input checked="" type="radio"/> Sí <input type="radio"/> No
Jardin:	<input checked="" type="radio"/> Sí <input type="radio"/> No	Sala Privada:	<input checked="" type="radio"/> Sí <input type="radio"/> No
Acceso a Minusválidos:	<input checked="" type="radio"/> Sí <input type="radio"/> No	Tarjeta de Crédito:	<input checked="" type="radio"/> Sí <input type="radio"/> No
Vale de Comida:	<input checked="" type="radio"/> Sí <input type="radio"/> No	Internet:	<input checked="" type="radio"/> Sí <input type="radio"/> No
Servicio a domicilio:	<input checked="" type="radio"/> Sí <input type="radio"/> No		

[Guardar](#) [Volver](#)

© 2013 Buscador de Restaurantes. Todos los derechos reservados. Prohibida la reproducción / copia / redistribución total o parcial de los contenidos de este site. Todos los derechos reservados.
 Todos los logos y marcas presentes en este site pertenecen a sus respectivos propietarios.

Figura . Interfície – Pantalla de reservas de Restaurants





Donde comer?

INICIO LOGIN REGISTRAR RESTAURANTES CONTACTO

Buscador de reservas

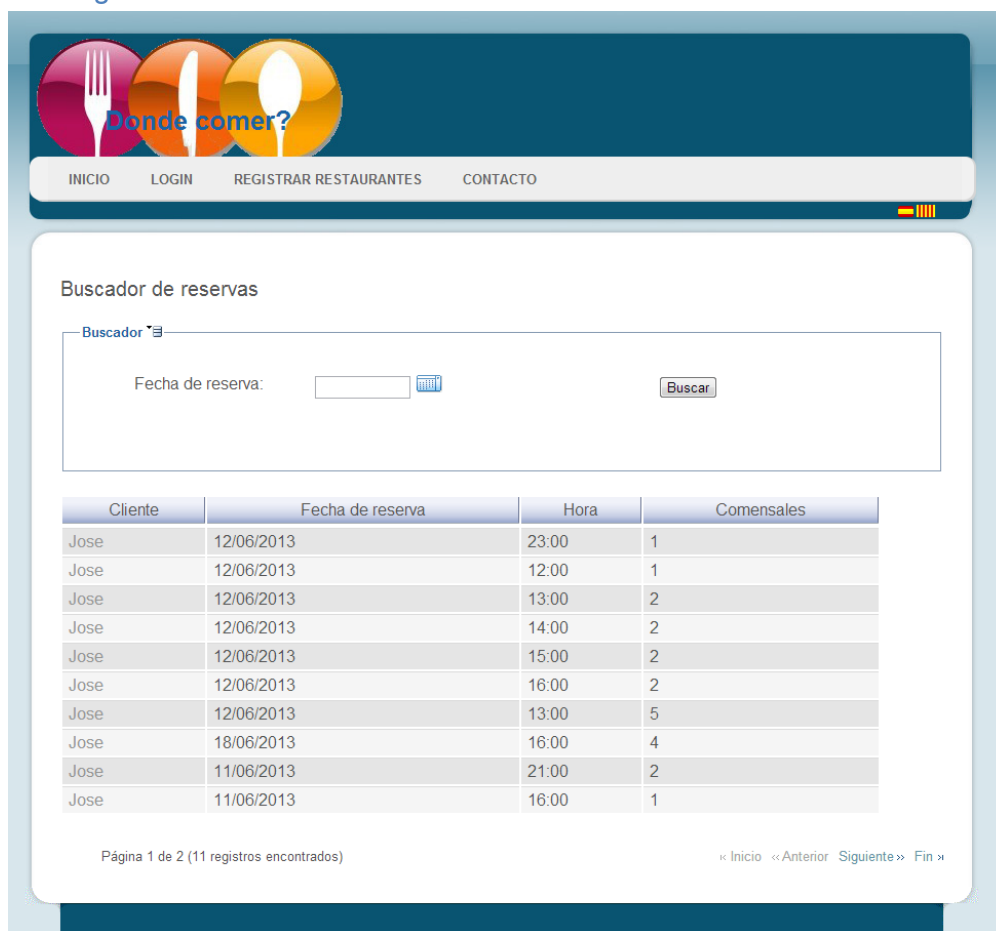
Buscador

Fecha de reserva: 

 No se han encontrado resultados

© 2013 Donde comer?. Todos los derechos reservados. Prohibida la reproducción / copia / redistribución total o parcial de los contenidos de este site. Todos los derechos reservados.
Todos los logos y marcas presentes en este site pertenecen a sus respectivos propietarios.

Figura . Interfície – Pantalla de reservas de Restaurants amb dades




Donde comer?

INICIO LOGIN REGISTRAR RESTAURANTES CONTACTO

Buscador de reservas

Buscador

Fecha de reserva: 

Cliente	Fecha de reserva	Hora	Comensales
Jose	12/06/2013	23:00	1
Jose	12/06/2013	12:00	1
Jose	12/06/2013	13:00	2
Jose	12/06/2013	14:00	2
Jose	12/06/2013	15:00	2
Jose	12/06/2013	16:00	2
Jose	12/06/2013	13:00	5
Jose	18/06/2013	16:00	4
Jose	11/06/2013	21:00	2
Jose	11/06/2013	16:00	1

Página 1 de 2 (11 registros encontrados) « Inicio << Anterior Siguiente >> Fin »

© 2013 Donde comer?. Todos los derechos reservados. Prohibida la reproducción / copia / redistribución total o parcial de los contenidos de este site. Todos los derechos reservados. Todos los logos y marcas presentes en este site pertenecen a sus respectivos propietarios.

