

TFC – Aplicació web per a la consulta i gestió de les benzineres estatals.

Autor: Oriol Bonany i Baguer.
Enginyeria Tècnica en Informàtica de Gestió.

Tutor: Albert Grau i Perisé.

Dilluns 20 de juny del 2011.

Als meus pares, **M^a Rosa Baguer** i **Manel Bonany**.

Aquest treball final de carrera us el dedico a vosaltres, també us vull donar les gràcies; moltes gràcies per estar sempre al costat meu, per la vostra paciència, per l'educació que m'heu donat i per ser com sou.

Agraïments.

Molt especialment vull donar les gràcies al meu xicot, **Angel Risco**, per ser tant optimista com pessimista soc jo, per ajudar-me a deixar el passat, per construir un futur al meu costat, per aguantar les meves dèries, per ser pacient amb els meus estudis i per tot el que no hagi dit explícitament.

També a la meva germana, **Berta Bonany**, que sempre ha cregut en mi. Gràcies Berta per estar al meu costat, tot el que pugui dir de tu serà poca cosa...

Agrair al meu tutor, **Albert Grau**, l'esforç per haver-me donat ànims en tot moment i incondicionalment, sense tu aquest treball no hagués estat possible.

A més a més, m'agradaria donar les gràcies a totes aquelles persones que durant aquests anys m'heu ajudat, suportat i heu tingut paciència amb mi. No en faré una llista ja que probablement seria molt llarga i hi faltaria alguna persona, estic segur però que cadascú sabrà entendre de qui parlo.

Resum.

El treball que a continuació es presenta consisteix en l'anàlisi, disseny i implementació d'una aplicació web que permeti, principalment, la consulta de preus dels combustibles de les gasolineres estatals. La plataforma tecnològica escollida ha estat J2EE ja que aquesta tecnologia, basada principalment en el llenguatge Java, permet programar i executar aplicacions en un servidor d'una forma econòmica i alhora professional. L'aplicació segueix principalment el patró arquitectònic MVC (Model Vista Controlador), utilitzant com a marc de treball Struts2 i per a la persistència de dades Hibernate juntament amb el patró DAO.

A part de la funcionalitat principal de l'aplicació, aquesta també permet fer més coses. Per una banda tindrem la gestió dels usuaris de l'aplicació, aquests s'han de poder gestionar de forma interna (un usuari del tipus administrador ha de poder fer les operacions típiques de cerca, alta, baixa i modificació) i de forma externa (el propi usuari ha de poder gestionar les seves dades, és a dir, poder donar-se d'alta i modificar-ne les preferències i dades emmagatzemades). A més a més, registrar-se ha de tenir algun al·licient més, per això, a l'usuari registrat se li permetrà fer una cerca ordenada segons els criteris escollits (preu, marca i localització en la prioritat que prefereixi) i que aquesta sigui del combustible preferit; això no impedirà que faci cerques d'altres combustibles, marques, localitzacions o establir preu màxims.

Per l'altra banda tindrem la gestió de les benzineres, que segons el tipus d'usuari l'aplicació permetrà unes o altres funcionalitats:

- En el cas dels usuaris anònims (aquells que o no estan donats d'alta o no han entrat a l'aplicació) se'ls permetrà fer cerques parametritzades, és a dir, cerques sobre el llistat de benzineres amb restriccions (per exemple, només mostrar les benzineres de certa marca o localització amb un preu inferior del combustible escollit inferior a el preu desitjat). El resultat d'aquestes cerques estarà sempre ordenat per preu (de menor a major).
- En el cas dels usuaris registrats, a part de fer cerques parametritzades se'ls permetrà guardar informació sobre les seves preferències d'ordenació i poder accedir al llistat ordenat segons aquestes. Si no s'indica el contrari, es buscarà les benzineres sobre el preu del combustible preferit (aquell que l'usuari estableixi al registrar-se). Si l'usuari registrat vol donar-se de baixa, ho haurà de comunicar a algun usuari administrador.
- En el cas dels usuaris administradors, a més de totes les funcionalitats esmentades, també se'ls hi ha de permetre donar d'alta, modificar-ne les dades i donar de baixa a benzineres i a usuaris (i sobre aquests, també fer-ne cerques). A un usuari administrador no se li permet canviar les claus dels altres usuaris (tot i que se li permet veure-la), només pot canviar la seva clau.

Índex.

RESUM.....	3
ÍNDEX.	4
ÍNDEX DE FIGURES.	5
1 – INTRODUCCIÓ.....	6
1.1 – JUSTIFICACIÓ.	6
1.2 – OBJECTIUS.	6
1.2.1 – GESTIÓ D’USUARIS.	7
1.2.2 – GESTIÓ DE BENZINERES.	8
1.3 – ENFOCAMENT I MÈTODE SEGUIT.	8
1.4 – PLANIFICACIÓ DEL PROJECTE.	9
1.5 – PRODUCTE OBTINGUT.	10
2 – ESPECIFICACIÓ I ANÀLISIS DELS REQUERIMENTS.	11
2.1 – DEFINICIÓ DELS ACTORS DEL SISTEMA.	11
2.2 – CASOS D’US.	12
2.2.1 – CASOS D’US DELS USUARIS ANÒNIMS.	13
2.2.2 – CASOS D’US DELS USUARIS REGISTRATS NO ADMINISTRADORS.	13
2.2.3 – CASOS D’US DELS USUARIS REGISTRATS ADMINISTRADORS.	14
2.2.4 – FITXES DELS CASOS D’US DEL SUBSISTEMA “GESTIÓ USUARIS”.	14
2.2.5 – FITXES DELS CASOS D’US DEL SUBSISTEMA “GESTIÓ BENZINERES”.	17
3 – DISSENY DE L’APLICACIÓ.....	20
3.1 – DISSENY ARQUITECTÒNIC.	20
3.2 – DIAGRAMA D’ENTITATS - RELACIÓ.	22
3.3 – DIAGRAMA ESTÀTIC DE CLASSES.	24
3.4 – DIAGRAMA DE PAQUETS.	25
3.5 – DIAGRAMES DE SEQÜÈNCIA.	26
3.5.1 – DIAGRAMES DE SEQÜÈNCIA DEL SUBSISTEMA “GESTIÓ USUARIS”.	26
3.5.2 – DIAGRAMES DE SEQÜÈNCIA DEL SUBSISTEMA “GESTIÓ BENZINERES”.	29
3.6 – DISSENY CAPA WEB.	32
4 – IMPLEMENTACIÓ.	32
5 – BREU ANÀLISIS ECONÒMIC.	33
6 – CONCLUSIONS.	33
7 – BIBLIOGRAFIA.....	34
7.1 – BIBLIOGRAFIA ESCRITA.	34
7.2 – CONSULTES A LA WEB.	34
8 – ANNEXES.	35
8.1 – INTERFÍCIE DE L’APLICACIÓ.	35
8.2 – <i>SCRIPT .SQL</i> DE CREACIÓ DE LA BASE DE DADES.	41

Índex de figures.

IL·LUSTRACIÓ 1: CASOS D'US DEL SISTEMA.....	12
IL·LUSTRACIÓ 2: CASOS D'US DELS USUARIS ANÒNIMS	13
IL·LUSTRACIÓ 3: CASOS D'US DELS USUARIS REGISTRATS NO ADMINISTRADORS	13
IL·LUSTRACIÓ 4: CASOS D'US DELS USUARIS REGISTRATS ADMINISTRADORS	14
IL·LUSTRACIÓ 5: FUNCIONAMENT ARQUITECTÒNIC	20
IL·LUSTRACIÓ 6: DIAGRAMA D'ENTITAT – RELACIÓ	22
IL·LUSTRACIÓ 7: DIAGRAMA ESTÀTIC DE CLASSES	24
IL·LUSTRACIÓ 8: DIAGRAMA DE PAQUETS	25
IL·LUSTRACIÓ 9: DIAGRAMA DE SEQÜÈNCIA DE “ALTA USUARI”	26
IL·LUSTRACIÓ 10: DIAGRAMA DE SEQÜÈNCIA DE “MODIFICAR USUARI”	26
IL·LUSTRACIÓ 11: DIAGRAMA DE SEQÜÈNCIA DE “BAIXA USUARI”	27
IL·LUSTRACIÓ 12: DIAGRAMA DE SEQÜÈNCIA DE “CERCA USUARI”	27
IL·LUSTRACIÓ 13: DIAGRAMA DE SEQÜÈNCIA DE “INICIAR SESSIÓ”	28
IL·LUSTRACIÓ 14: DIAGRAMA DE SEQÜÈNCIA DE “FINALITZAR SESSIÓ”	28
IL·LUSTRACIÓ 15: DIAGRAMA DE SEQÜÈNCIA DE “ALTA BENZINERA”	29
IL·LUSTRACIÓ 16: DIAGRAMA DE SEQÜÈNCIA DE “MODIFICAR BENZINERA”	29
IL·LUSTRACIÓ 17: DIAGRAMA DE SEQÜÈNCIA DE “CERCA BENZINERA ORDENACIÓ ESTÀNDAR”	30
IL·LUSTRACIÓ 18: DIAGRAMA DE SEQÜÈNCIA DE “BAIXA BENZINERA”	30
IL·LUSTRACIÓ 19: DIAGRAMA DE SEQÜÈNCIA DE “CERCA BENZINERA ORDENACIÓ USUARI”	31
IL·LUSTRACIÓ 20: DIAGRAMA DE SEQÜÈNCIA DE “VEURE DETALL BENZINERA”	31
IL·LUSTRACIÓ 21: PÀGINA INICIAL.....	35
IL·LUSTRACIÓ 22: PÀGINA DESPRÉS DE FER UNA CERCA ESSENT UN USUARI NO ADMINISTRADOR.	36
IL·LUSTRACIÓ 23: PÀGINA QUE MOSTRA EL DETALL D'UNA BENZINERA ESSENT UN USUARI NO ADMINISTRADOR.	36
IL·LUSTRACIÓ 24: PÀGINA DE CERCA ESSENT UN USUARI REGISTRAT ADMINISTRADOR.	37
IL·LUSTRACIÓ 25: PÀGINA MENTRE ES MODIFIQUEN LES DADES D'UNA BENZINERA.	37
IL·LUSTRACIÓ 26: PÀGINA DESPRÉS D'HAVER MODIFICAT UNA BENZINERA.	38
IL·LUSTRACIÓ 27: PÀGINA D'ALTA D'UN NOU USUARI ESSENT UN USUARI REGISTRAT ADMINISTRADOR.	38
IL·LUSTRACIÓ 28: PÀGINA DE REGISTRE DE NOU USUARI ESSENT UN USUARI ANÒNIM.....	39
IL·LUSTRACIÓ 29: PÀGINA DE CERCA D'USUARIS ESSENT ADMINISTRADOR.....	39
IL·LUSTRACIÓ 30: PÀGINA DE CONSULTA I ACTUALITZACIÓ DE LES DADES D'UN USUARI ESSENT ADMINISTRADOR.	40
IL·LUSTRACIÓ 31: PÀGINA DE CONSULTA I ACTUALITZACIÓ DE LES DADES PRÒPIES D'UN NO ADMINISTRADOR.....	40

1 – Introducció.

1.1 – Justificació.

L'elecció d'aquest treball ha estat causada sobre un sentiment de superació, de ganes d'aprendre i de conèixer tecnologies que no s'han donat durant la carrera o s'han donat molt per sobre; tecnologies que tenen un potencial enorme i penso que s'haurien d'aprofundir més en la carrera d'un informàtic.

L'escalada continua dels preus dels diferents combustibles i la creixent dependència que les societats estan desenvolupant sobre tot tipus d'energia fou la meua principal motivació alhora d'escollir un tema sobre el que aplicar la plataforma tecnològica J2EE.

L'elecció de la tecnologia fou relativament fàcil, J2EE és una plataforma que permet executar els programes sobre varis entorns i també afegir-hi diferents tecnologies que fan encara més independent l'aplicació (com Hibernate per a independitzar-la de la base de dades o Struts2 que permet independitzar-la del sistema informàtic client). Aquesta plataforma és a més gratuïta de lliure distribució.

Per una banda, l'elecció d'Hibernate i el patró DAO (de l'anglès *Data Access Object*) com a mètode d'accés a la base de dades fou senzilla, jo en coneixia una mica el funcionament, a més proporcionen a l'aplicació independència sobre la base de dades fent que el canvi d'una a altra sigui molt senzill. Per l'altra banda, l'elecció de Struts2 com a tecnologia que implementi el patró MVC (Model – Vista – Controlador) fou, tot i que arriscada, senzilla: Struts2 em proporcionava una forma senzilla d'implementar MVC, em donava un marc de treball (o *framework*) relativament senzill d'aprendre i poder treballar amb objectes.

Així doncs tenim que la part que conforma la vista del patró MVC son les planes webs que s'han realitzat amb JSP aplicant-hi un CSS molt senzill, també s'ha utilitzat JavaScript per solucionar algun problema puntual.

1.2 – Objectius.

L'objectiu principal del projecte és crear una eina informàtica que permeti de forma fàcil que, qui ho desitgi, pugui consultar el llistat de benzineres de tot l'estat (i els corresponents preus dels combustibles) de forma ordenada.

L'aplicació ha de poder donar suport a diferents usuaris i permetre la inserció, edició i eliminació tant dels usuaris com de les benzineres; així doncs es poden separar en dos grans funcionalitats:

- Gestió d'usuaris (implícitament inclou la gestió de les sessions).
- Gestió de benzineres.

1.2.1 – Gestió d'usuaris.

La gestió dels usuaris es farà implícitament utilitzant una taula dintre d'una base de dades relacional. Per a poder consultar els preus de les benzineres no es considera necessari estar registrat en el sistema, en aquests casos s'utilitzarà el llistat predeterminat (l'listat de les benzineres ordenades de menor a major preu de la benzina 95) com a plana de benvinguda. L'usuari anònim també podrà fer cerques parametritzades, però només podran ser ordenades de menor a major preu.

S'entén doncs que hi haurà una àrea pública i una àrea privada. Per accedir a l'àrea privada cal autenticar-se mitjançant el nom d'usuari (que correspondrà a l'adreça de correu electrònic) i una paraula clau o bé, si així ho desitja un usuari, registrar-se utilitzant la pàgina "Afegir usuari", la qual s'accedirà mitjançant un enllaç des de la pàgina inicial. Les dades que es sol·licitaran seran:

- Nom i cognoms.
- Adreça electrònica: s'utilitzarà com a "usuari" per accedir a l'àrea privada.
- Clau d'accés a l'aplicació web.
- Data de naixement.
- Tipus de carburant preferit: es deixarà escollir mitjançant un desplegable.
- Localització: nom de la població i província de residència, mentre es tecleja en sorgirà un desplegable amb les diferents possibilitats.
- Direcció.
- Tipus d'ordenació preferida: tres desplegables que indicaran els criteris pels qual l'usuari vol que les cerques surtin ordenades. No es poden repetir criteris.

La diferència entre l'àrea pública i l'àrea privada serà solament el fet d'utilitzar un usuari registrat o l'usuari anònim.

Amb l'usuari anònim: a la pantalla inicial es mostrarà el llistat de benzineres ordenat per preu de la benzina 95. Es permetrà fer cerques sobre tots els camps disponibles: nom, marca, combustible, preu del combustible, localització i/o direcció, però sempre sortiran ordenats pel camp preu.

L'usuari registrat no administrador, un cop autenticat, podrà accedir al les pàgines de modificació de dades "Modificació usuari" on podrà canviar les diferents opcions esmentades. Si es vol donar de baixa o consultar la paraula clau ho haurà de demanar a un usuari administrador.

Amb l'usuari registrat no administrador, a la pantalla inicial, es mostrarà el llistat de benzineres ordenades segons les prioritats dels criteris establerts al registrar-se (localitat, preu del carburant preferit i marca de la benzina); també se li permet fer cerques sobre tots els camps disponibles. L'ordenació serà sempre la preferida i només la pot canviar modificant el perfil. El combustible serà sempre per defecte el preferit, però el podrà canviar fàcilment al formulari de cerca .

L'usuari registrat administrador disposarà de totes les funcionalitats de l'aplicació. Això vol dir que pot fer cerques sobre el nom (o una part del nom) i/o els cognoms (o una part dels cognoms), afegir usuaris, esborrar usuaris i les funcions de la gestió de benzineres. Cal tenir en compte que l'administrador només pot consultar contrasenyes d'altres usuaris, mai canviar-les.

1.2.2 – Gestió de benzineres.

La gestió de les benzineres tant sols la podran realitzar els usuaris que formin part de l'administració de la pàgina web; aquests usuaris hauran de ser donats d'alta de forma manual a la taula d'usuaris de la base de dades pel propietari i administrador de la mateixa, o bé, des de l'aplicació web entrant com a usuari registrat administrador.

Per accedir a la gestió de les benzineres primer de tot caldrà autenticar l'usuari de la mateixa manera que es fa amb els usuaris normals i automàticament al seu apartat corresponent hi apareixeran les opcions per a gestionar-les:

“Afegir benzinera”: des d'aquest apartat es podrà donar d'alta una nova benzinera; en aquest apartat es demanaran les dades bàsiques imprescindibles per al bon funcionament de l'aplicació. No es permet la creació de noves marques, sinó que se'n haurà d'escollir una introduint les primeres lletres i tot seguit utilitzant el desplegable que apareixerà. De la mateixa manera, no es poden donar d'alta noves localitzacions (poblacions amb la corresponent província).

Per modificar i esborrar una benzinera, si la persona s'ha autenticat com a usuari administrador, l'aplicació web donarà la possibilitat de fer-ho de la següent manera:

“Modificació benzinera”: per accedir a aquesta funcionalitat caldrà primer fer una cerca a la base de dades de la benzinera que es vol modificar. Al llistat, el nom de la benzinera apareixerà com un enllaç, al qual fent-hi clic s'accedirà a l'apartat corresponent. Des d'aquí es podran modificar totes les dades relatives a la benzinera, aquesta opció serà especialment utilitzada alhora de corregir alguns preus de les benzineres, falles lèxiques, horaris, ...

“Baixa benzinera”: de la mateixa manera que amb la modificació, caldrà primer fer una cerca. Si l'usuari és administrador llavors apareixerà a la llista, al costat de cada benzinera, una última columna amb un enllaç amb la paraula “Eliminar” que donarà de baixa la benzinera.

1.3 – Enfocament i mètode seguit.

L'enfocament inicial per aquest treball era el de l'absoluta planificació, l'estudi exhaustiu de les tecnologies implicades i l'elaboració d'un producte informàtic rigorós, preparat per al món real i que fos alhora la presentació personal al món laboral. Per això, recordant les assignatures d'enginyeria del software, cal analitzar amb deteniment les especificacions de l'aplicació, dissenyar-ne les diferents classes amb cura i implementar la solució tecnològica tot utilitzant els patrons que millor donin resposta a les necessitats.

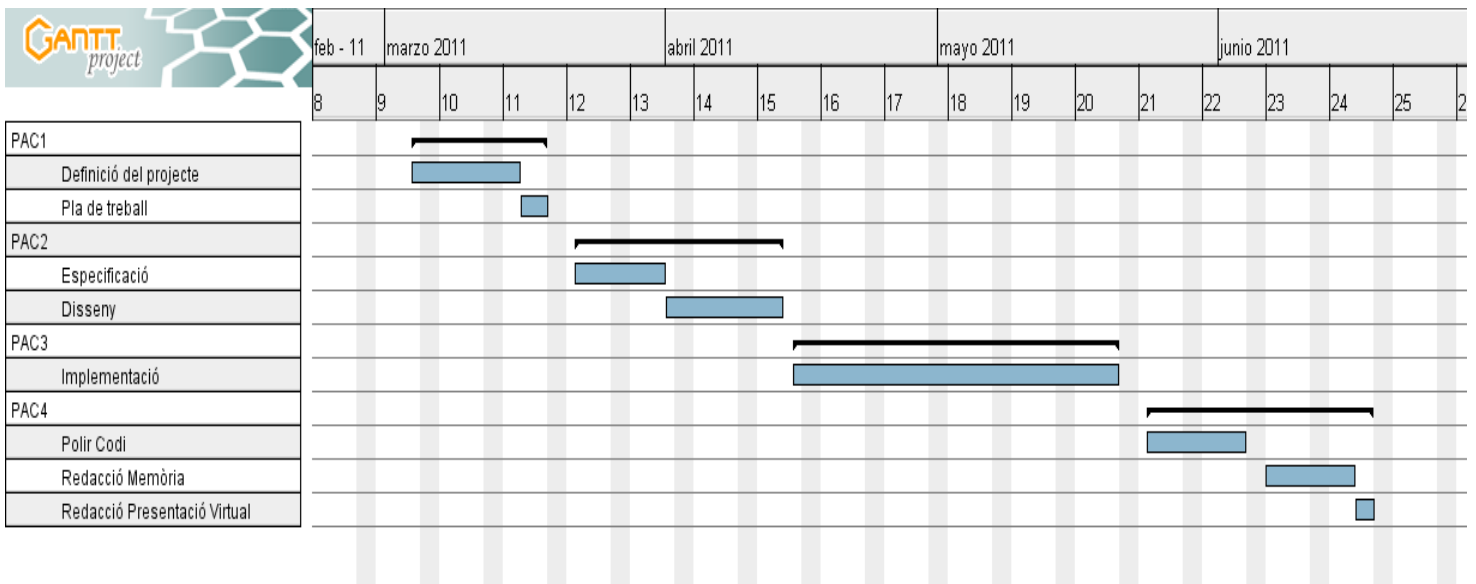
Finalment el mètode seguit ha variat, per causes diverses, de l'enfocament inicial; per aquest motiu alhora de fer la implementació s'ha hagut de retallar alguna petita funcionalitat inicialment planejada (com el fet de voler implementar una *newsletter* o de comprovar la correctesa del correu electrònic registrat). El mètode seguit ha estat essencialment el mateix, la principal variació ha estat en la planificació; aquesta ha provocat variacions en els dissenys degut a l'excessiva ambició de les funcionalitats inicials.

1.4 – Planificació del projecte.

Hi ha quatre dates que són clau per al seguiment i acompliment del TFC, aquestes corresponen amb les dates d'entrega de les diferents PAC i són:

Data	Document	Descripció
16/03/2011	PAC 1 / Pla de treball	Redacció del pla de treball: descripció, objectius, funcionalitats i planificació temporal.
14/04/2011	PAC 2	Redacció de l'especificació tècnica del treball: diagrames UML. Disseny de l'arquitectura tecnològica. Instal·lació i configuració de l'entorn de programació.
23/05/2011	PAC 3	Implementació del treball.
20/06/2011	PAC 4 (MEM+PRES)	Redacció dels documents "Memòria", "Presentació Virtual" i polir el codi.

Això mateix i separat per tasques es pot veure en el següent diagrama de Gantt:



Per a major claredat:

- **PAC1:** del 04/03/2011 al 21/03/2011. 17 dies naturals.
Definició del projecte: del 04/03/2011 al 16/03/2011. 12 dies naturals.
Redacció del pla de treball: del 17/03/2011 al 21/03/2011. 5 dies naturals.
- **PAC2:** del 22/03/2011 al 14/04/2011. 24 dies naturals.
Especificació: del 22/03/2011 al 01/04/2011. 12 dies naturals.
Disseny: del 02/04/2011 al 14/04/2011. 12 dies naturals.
- **PAC3:** del 15/04/2011 al 23/05/2011. 38 dies naturals.
- **PAC4:** del 24/05/2011 al 20/06/2011. 28 dies naturals.
Polir codi: del 24/05/2011 al 01/06/2011. 9 dies naturals
Memòria: del 02/06/2011 al 12/06/2011. 11 dies naturals.
Presentació Virtual: del 13/06/2011 al 20/06/2011. 8 dies naturals.

1.5 – Producte obtingut.

El principal producte obtingut és una aplicació web que permet la gestió de les benzineres a nivell estatal i la gestió dels usuaris de l'aplicació. En concret, s'ha obtingut:

- Un fitxer .war corresponent a l'aplicació web.
- Un fitxer .sql corresponent a les sentències de creació de la base de dades i de les taules, i de introducció de dades a les taules.
- El codi font empaquetat en un fitxer .zip en format de projecte per a l'IDE Eclipse.
- Una memòria del treball en format .pdf.
- Una presentació virtual que sintetitza la memòria en format .ppsx (presentació amb diapositives de l'Office 2007).

2 – Especificació i anàlisi dels requeriments.

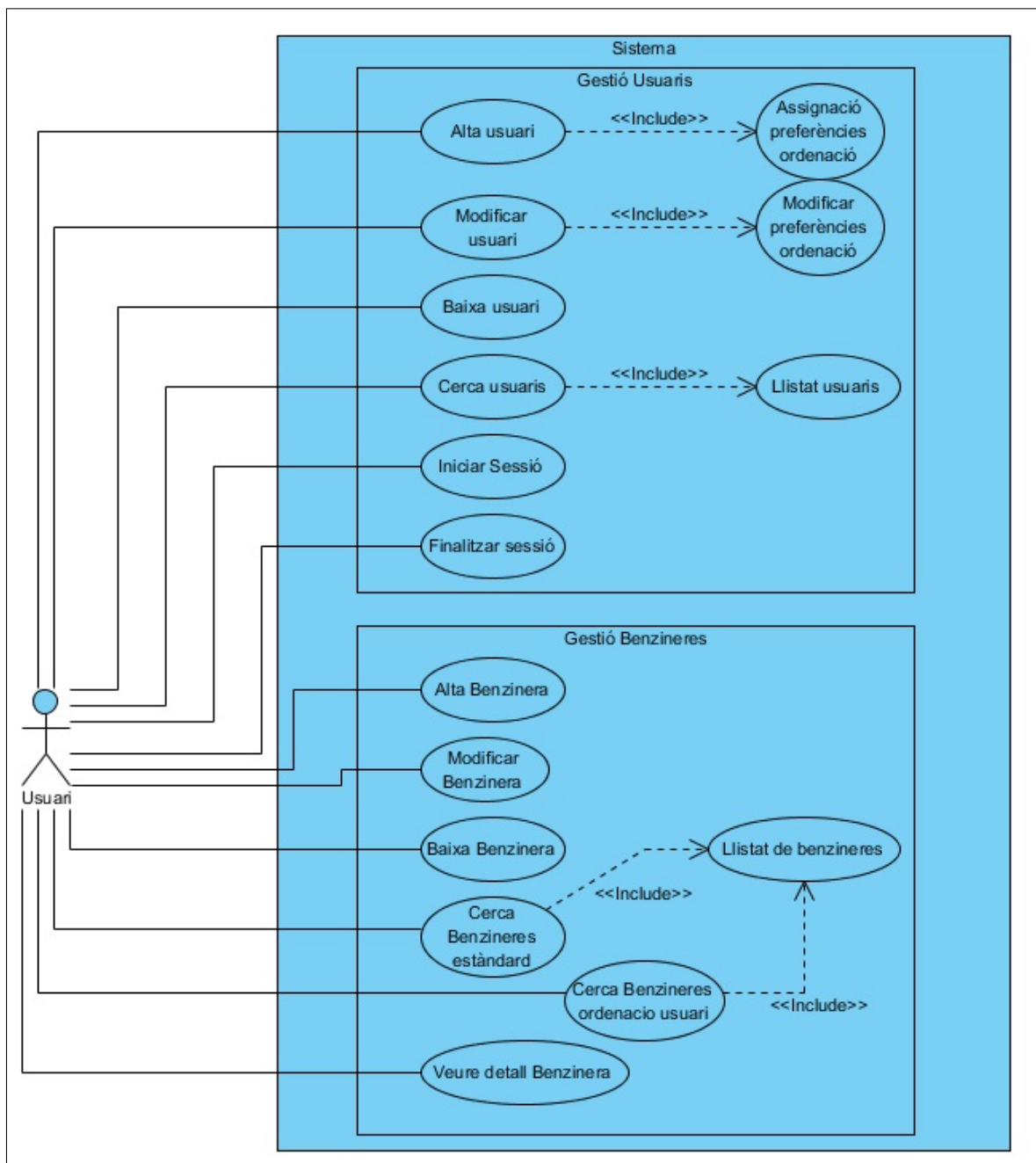
2.1 – Definició dels actors del sistema.

S'entén per actor un subjecte que interactua amb el sistema. Des d'aquesta petita definició entendrem que les actors del nostre sistema seran els diferents tipus d'usuaris, els quals tindran diferents rols i podran fer servir diferents funcionalitats del sistema. En el cas que ens ocupa es distingeixen tres **tipus d'usuaris**:

- **Usuari anònim:** aquell que accedeix a l'aplicació i no s'autentica amb un usuari i contrasenya correctament. Aquest tipus d'usuari únicament pot fer consultes de benzineres i veure els detalls de les mateixes (nom, marca, direcció, marge, localitat, província, horari d'obertura i preu dels diferents combustibles).
- **Usuari registrat no administrador:** aquell que s'ha autenticat i no disposa de permisos d'administració. Aquest tipus d'usuari pot, a part de realitzar les accions permeses pels usuaris anònims, modificar el seu perfil excepte el correu electrònic (nom, cognoms, paraula clau, data de naixement, direcció, població, província, marge, tipus de combustible preferit i motius d'ordenació). La principal diferència respecte dels usuaris anònims és que se li permet enregistrar els criteris d'ordenació preferits per a la cerca de benzineres, i aquests serveixen alhora per fer les cerques una mica més automatitzades.
- **Usuari registrat administrador:** aquell usuari que s'ha autenticat i disposa de permisos d'administració. Aquest tipus d'usuari té accés a totes les funcionalitats implementades a l'aplicació, és a dir, a part del que poden fer els usuaris registrats no administradors també se li permet gestionar els usuaris (fer cerques per nom i/o cognoms, afegir usuaris, esborrar usuaris i editar-ne el perfil) i gestionar les benzineres (afegir benzineres, esborrar benzineres i editar-ne el perfil).

2.2 – Casos d’us.

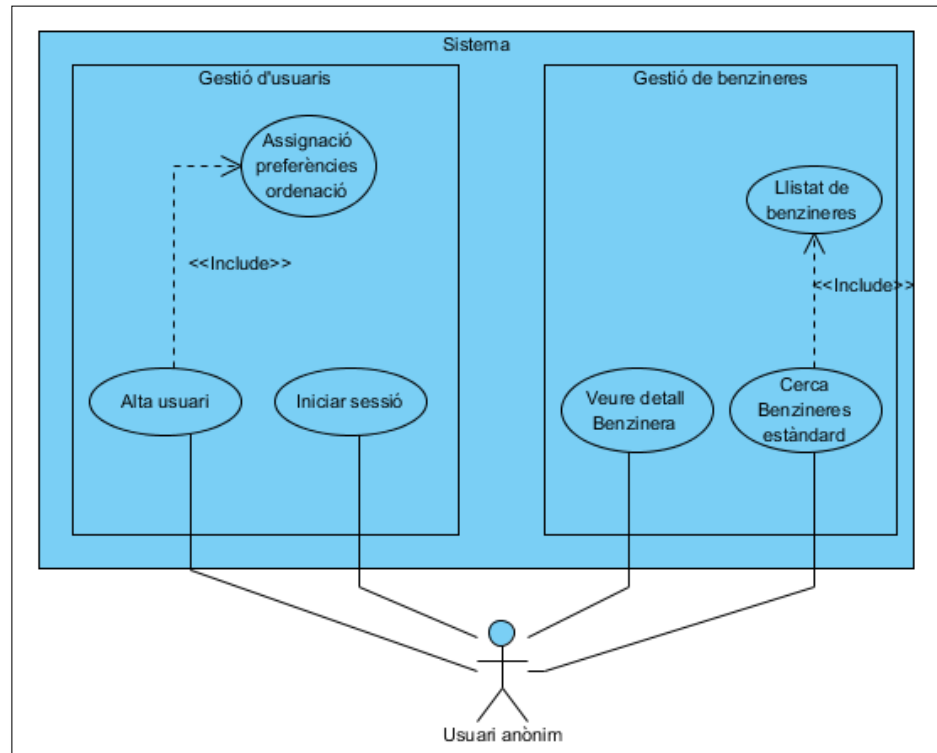
Tot seguit es mostraran els casos d’us implementats a l’aplicació final (II-lustració 1). El primer cas d’us presenta el cas d’us del sistema, agrupant tots els actors esmentats anteriorment en un de sol: Usuari. Tot seguit es posaran els casos d’us per a cada actor.



II-lustració 1: Casos d’us del sistema

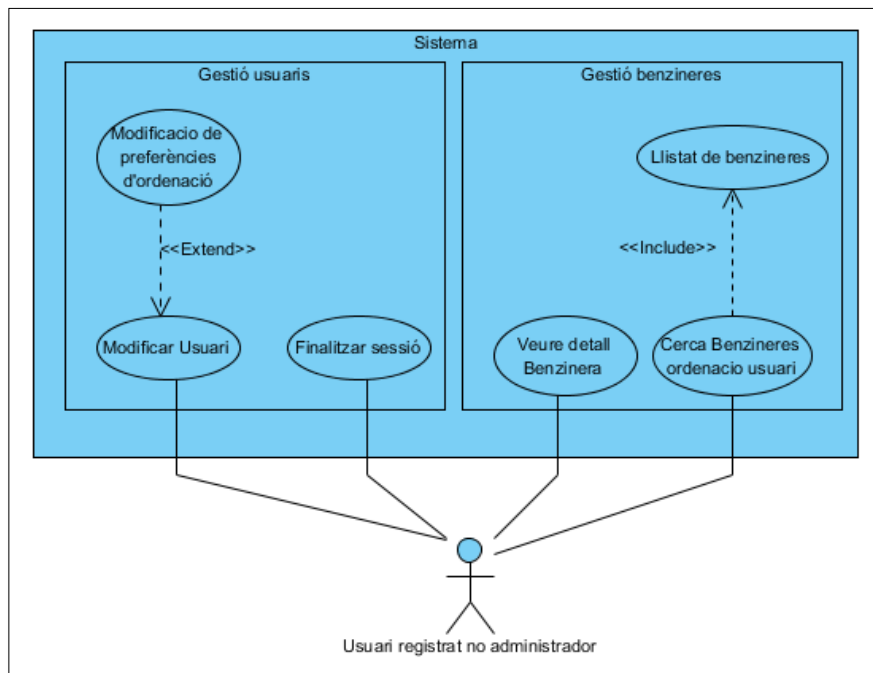
Així doncs s’observen dos grans subsistemes: el de gestió dels usuaris i el de gestió de benzineres. Veiem doncs per cada actor quines parts d’aquests subsistemes poden utilitzar i com.

2.2.1 – Casos d'us dels usuaris anònims.



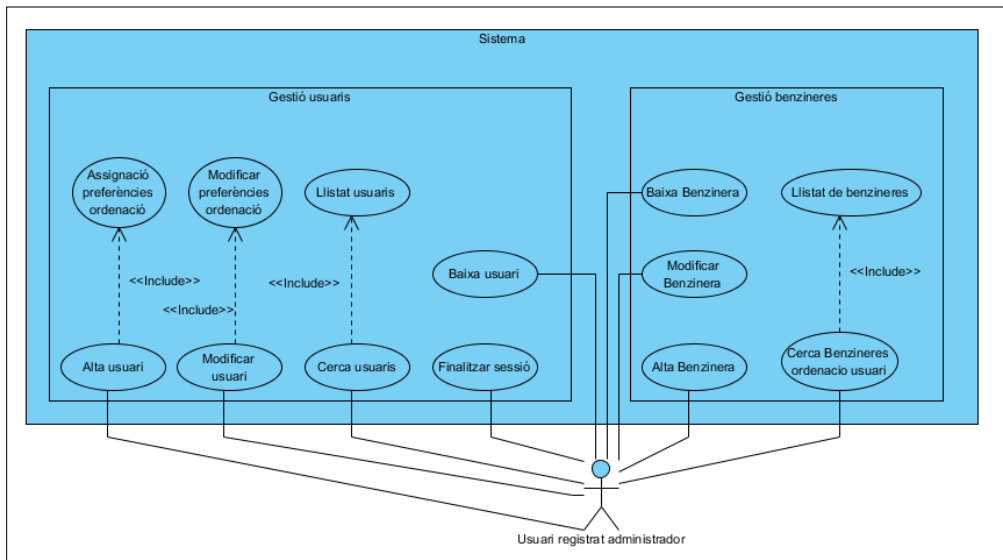
II-lustració 2: Casos d'us dels usuaris anònims

2.2.2 – Casos d'us dels usuaris registrats no administradors.



II-lustració 3: Casos d'us dels usuaris registrats no administradors

2.2.3 – Casos d’us dels usuaris registrats administradors.



II-lustració 4: Casos d’us dels usuaris registrats administradors

2.2.4 – Fitxes dels casos d’us del subsistema “Gestió usuaris”.

Alta usuari.

Descripció:	Dona d’alta un nou usuari al sistema i l’enregistra a la base de dades. No es permet la introducció de noves localitzacions.
Actors:	Usuari, Sistema.
Precondició:	El correu electrònic de l’usuari no ha estat donat d’alta.
Flux principal:	<ol style="list-style-type: none"> 1. L’usuari accedeix a la pantalla “Afegir usuari”. 2. El sistema mostra la pantalla. 3. L’usuari emplena les dades del formulari i prem el botó “Registra” 4. El sistema comprova “l’Assignació de preferències d’ordenació”. 5. El sistema enregistra l’usuari i l’informa de que així ha estat fet.
Flux alternatiu:	4.1 Si les preferències no son correctes s’avisa a l’usuari i es torna al punt 3
Postcondició:	L’usuari s’ha donat d’alta.

Baixa usuari.

Descripció:	Dona de baixa un usuari existent i l’esborra de la base de dades (només usuaris registrats administradors).
Actors:	Usuari, Sistema.
Precondició:	L’usuari es troba creat dintre de la base de dades.
Flux principal:	<ol style="list-style-type: none"> 1. L’usuari prem el botó “Eliminar” de l’usuari a esborrar de la llista d’usuaris cercats. 2. El sistema n’esborra totes les dades i l’informa de que així ha estat fet.
Flux alternatiu:	
Postcondició:	L’usuari ha estat esborrat del sistema i no en queden dades guardades.

Modificar usuari.

Descripció:	Modifica les dades de l'usuari, s'hi pot accedir via cerca de usuaris i polsant sobre el nom de l'usuari que es vol modificar (només usuaris administradors) o bé polsant sobre "Dades de..." a la pantalla principal del propi usuari. No es permet modificar el correu de l'usuari ni afegir noves localitzacions.
Actors:	Usuari, Sistema.
Precondició:	L'usuari existeix al sistema.
Flux principal:	<ol style="list-style-type: none">1. L'usuari accedeix a la pantalla "Detall de usuari".2. El sistema mostra la pantalla i n'emplena el formulari amb les dades actuals de l'usuari a modificar.3. L'usuari modifica les dades de l'usuari.4. El sistema comprova la correcta "Modificació de les dades d'ordenació".5. L'usuari prem el botó "Actualitzar".6. El sistema guarda les dades i confirma que s'han emmagatzemat.
Flux alternatiu:	Si no s'emplena el camp "Clau d'accés" el sistema mostra un avís conforme cal emplenar-lo i no actualitza les dades. Si l'usuari és administrador no se li permet modificar el camp "Clau d'accés" si no es tracta del seu propi perfil.
Postcondició:	Les dades de l'usuari han estat actualitzades.

Cerca usuaris.

Descripció:	Fa una cerca dels usuaris existents segons els camps "Nom" i "Cognoms" que s'indiquin (només usuaris registrats administradors).
Actors:	Usuari, Sistema.
Precondició:	Almenys hi ha un usuari (el propi).
Flux principal:	<ol style="list-style-type: none">1. L'usuari accedeix a la pantalla "Cerca de usuaris".2. El sistema mostra la pantalla i emplena el llistat complert d'usuaris.3. L'usuari emplena el "Nom" i/o els "Cognoms" i prem el botó "Buscar".4. El sistema en fa la cerca i mostra el llistat dels usuaris.
Flux alternatiu:	Si no hi ha usuaris que compleixin la cerca es mostra un missatge d'avís: "No s'han trobat resultats pels criteris establerts.". No es fa el llistat.
Postcondició:	S'ha mostrat la llista dels usuaris.

Iniciar sessió.

Descripció:	L'usuari anònim s'autentica i accedeix al sistema com a usuari registrat.
Actors:	Usuari, Sistema.
Precondició:	
Flux principal:	<ol style="list-style-type: none">1. L'usuari anònim accedeix a la pantalla principal i emplena els camps "Usuari" i "Password", tot seguit prem el botó "Entrar".2. El sistema comprova les dades introduïdes, si son correctes li dona la benvinguda i mostra la pantalla inicial segons quin sigui el rol de l'usuari.
Flux alternatiu:	Si l'usuari s'equivoca en algun dels camps a emplenar, el sistema l'informa de que no ha estat possible autenticar-lo i mostra la pantalla inicial.
Postcondició:	L'usuari ha iniciat sessió o ha estat rebutjat.

Finalitzar sessió.

Descripció:	L'usuari es desconnecta de l'aplicació, passant a ser un usuari anònim.
Actors:	Usuari, Sistema.
Precondició:	L'usuari ha inicialitzat la sessió.
Flux principal:	1. L'usuari prem el botó "Desconnectar-me". 2. El sistema enregistra la data i hora d'entrada i de desconnexió, mostra un missatge conforme s'ha desconnectat i mostra la pantalla principal.
Flux alternatiu:	
Postcondició:	L'usuari està desconnectat.

Llistat usuaris.

Descripció:	El sistema emplena el llistat d'usuaris segons quins siguin els paràmetres de cerca.
Actors:	Sistema.
Precondició:	
Flux principal:	1. El sistema emplena el llistat dels usuaris conforme la cerca que s'hagi realitzat i el mostra a la pantalla.
Flux alternatiu:	
Postcondició:	S'ha mostrat el llistat dels usuaris segons el criteri de cerca.

Assignació preferències ordenació.

Descripció:	L'usuari escull les preferències d'ordenació.
Actors:	Usuari, Sistema.
Precondició:	L'usuari és nou.
Flux principal:	1. L'usuari selecciona les preferències d'ordenació. 2. El sistema comprova que siguin escollides correctament.
Flux alternatiu:	Si les preferències han estat escollides correctament s'enregistren a la taula corresponent i es permet continuar. Si no ho estan, el sistema mostra un avís i es passa al punt 1.
Postcondició:	S'han assignat les preferències d'ordenació.

Modificació preferències ordenació.

Descripció:	L'usuari modifica les preferències d'ordenació.
Actors:	Usuari, Sistema.
Precondició:	L'usuari no és nou i, per tant, ja disposa de preferències enregistrades.
Flux principal:	1. L'usuari selecciona les noves preferències d'ordenació. 2. El sistema comprova que siguin escollides correctament.
Flux alternatiu:	Si les preferències han estat modificades correctament s'actualitza el registre a la taula corresponent i es permet continuar. Si no ho estan, el sistema mostra un avís i es passa al punt 1.
Postcondició:	S'han modificat les preferències d'ordenació.

2.2.5 – Fitxes dels casos d'ús del subsistema “Gestió benzineres”.

Alta benzinera.

Descripció:	Dona d'alta una nova benzinera al sistema i l'enregistra (només usuaris registrats administradors). No es permet la creació de noves marques ni de noves localitzacions.
Actors:	Usuari, Sistema.
Precondició:	La benzinera no ha estat donada d'alta.
Flux principal:	<ol style="list-style-type: none">1. L'usuari accedeix a la pantalla “Afegir benzinera”.2. El sistema mostra la pantalla.3. L'usuari n'emplena les dades i prem el botó “Registrar”.4. El sistema enregistra les dades de la nova benzinera.
Flux alternatiu:	
Postcondició:	La benzinera ha estat donada d'alta.

Baixa benzinera.

Descripció:	Dona de baixa una benzinera del sistema (només usuaris registrats administradors).
Actors:	Usuari, Sistema.
Precondició:	La benzinera existeix al sistema.
Flux principal:	<ol style="list-style-type: none">1. L'usuari prem el botó “Eliminar” corresponent a la benzinera a esborrar de la llista de benzineres cercades.2. El sistema esborra de la base de dades la benzinera i les dades relatives a ella.
Flux alternatiu:	
Postcondició:	La benzinera i les seves dades han estat eliminades de la base de dades.

Modificar benzinera.

Descripció:	Modifica les dades de la benzinera seleccionada (només usuaris registrats administradors). S'hi pot accedir polsant sobre el nom de la benzinera. No es permet afegir noves localitzacions ni noves marques.
Actors:	Usuari, Sistema.
Precondició:	La benzinera existeix al sistema.
Flux principal:	<ol style="list-style-type: none">1. L'usuari accedeix a la pàgina “Modificar benzinera”.2. El sistema mostra la pantalla i n'emplena el formulari amb les dades actuals de la benzinera a modificar.3. L'usuari modifica les dades de la benzinera i prem el botó “Actualitzar”.4. El sistema guarda les dades i confirma que s'han emmagatzemat.
Flux alternatiu:	
Postcondició:	Les dades de la benzinera han estat modificades correctament.

Veure detall benzinera.

Descripció:	Permet veure els detalls de la benzinera. S’hi pot accedir polsant sobre el nom de la benzinera. Aquesta funcionalitat només la poden utilitzar els usuaris anònims i els usuaris registrats no administradors. Els usuaris registrats administradors disposen de la funcionalitat “Modificar benzinera”, on poden veure els mateixos detalls i a més, modificar-los.
Actors:	Usuari, Sistema.
Precondició:	La benzinera existeix al sistema.
Flux principal:	1. L’usuari accedeix a la pàgina “Veure benzinera”. 2. El sistema mostra la pantalla amb les dades actuals de la benzinera.
Flux alternatiu:	
Postcondició:	Les dades de la benzinera han estat mostrades correctament.

Cerca benzineres estàndard.

Descripció:	Realitza la cerca parametrizada de les benzineres; l’ordenació estàndard és de menor a major preu del combustible escollit. S’utilitzen els paràmetres de cerca que contingui el formulari; per als usuaris anònims per defecte el tipus de combustible és Benzina 95.
Actors:	Usuari, Sistema.
Precondició:	L’usuari no ha iniciat sessió.
Flux principal:	1. L’usuari accedeix a la pàgina de cerca o emplena el formulari de cerca i prem el botó “Buscar”. 2. El sistema fa la cerca i en mostra el llistat de benzineres obtingut.
Flux alternatiu:	Si no hi ha benzineres que compleixin la cerca es mostra un missatge d’avís “No s’han trobat resultats pels criteris establerts.”
Postcondició:	S’ha mostrat per pantalla el llistat de benzineres estatals ordenades pel preu del combustible escollit menor a major.

Cerca benzineres ordenació usuari.

Descripció:	Realitza la cerca parametrizada de les benzineres; l’ordenació de la llista es fa utilitzant els criteris d’ordenació emmagatzemats en el perfil de l’usuari. S’utilitzen els paràmetres de cerca que contingui el formulari; per als usuaris registrats per defecte el tipus de combustible és el preferit.
Actors:	Usuari, Sistema
Precondició:	L’usuari ha iniciat sessió.
Flux principal:	1. L’usuari accedeix a la pàgina de cerca o emplena el formulari de cerca i prem el botó “Buscar”. 2. El sistema fa la cerca ordenant pels criteris establerts per l’usuari en el perfil i mostra el llistat de benzineres obtingut.
Flux alternatiu:	Si no hi ha benzineres que compleixin la cerca es mostra un missatge d’avís “No s’han trobat resultats pels criteris establerts.”
Postcondició:	S’ha mostrat per pantalla el llistat de benzineres segons els paràmetres introduïts i ordenada pels criteris que hagi establert l’usuari al perfil.

Llistat benzineres.

Descripció:	El sistema crea i omple el llistat de benzineres segons quins siguin els paràmetres de la cerca i l'ordenació indicada.
Actors:	Sistema.
Precondició:	Almenys hi ha una benzinera al llistat.
Flux principal:	1. El sistema omple el llistat de les benzineres conforme la cerca que s'hagi realitzat i el mostra a la pantalla.
Flux alternatiu:	
Postcondició:	S'ha mostrat el llistat de benzineres resultat de la cerca.

3 – Disseny de l'aplicació.

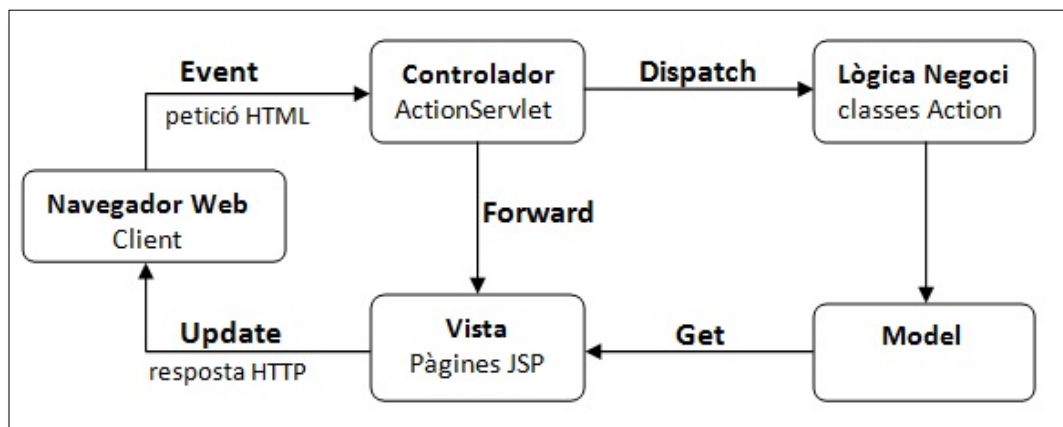
En el punt anterior hem vist què volem que faci l'aplicació, quins actors hi intervindran, què podran fer, etc. Ara toca doncs veure el com. Per això tot seguit presentaré el diagrama d'entitats derivat de l'anàlisi; el diagrama final depèn dels protocols i la tecnologia utilitzada, per això primer veiem com ens ajuden a dissenyar l'aplicació i tot seguit com quedarà.

3.1 – Disseny arquitectònic.

La plataforma tecnològica escollida, tema d'aquest treball final de carrera, és J2EE (tal com ja sabem). El disseny arquitectònic ha de respondre al fet de ser una aplicació distribuïda, s'ha de poder utilitzar en diferents ordinadors i això ens ho permet principalment J2EE.

Per tal de completar el disseny distribuït, es fa necessari l'ús d'un servidor d'aplicacions. En el meu cas s'ha utilitzat Apache Tomcat en la seva última versió. Aquest servidor s'encarrega doncs de compilar el codi de les pàgines JSP transformant-les en *servlets*, mitjançant l'ús del compilador JASPER que porta inclòs.

Amb la finalitat de fer una aplicació que sigui mantenible i ampliable fàcilment cal utilitzar el patró MVC que tot seguit explicaré amb més detall. Per a la implementació del mateix s'utilitzarà el marc de treball Struts2 que implementa de forma senzilla el patró. Struts2 em permet tenir clarament diferenciades la part de la presentació (vista), la part de controlador i la part de la lògica de negoci (model). D'aquesta manera, les peticions dels usuaris s'envien al controlador (al *servlet* *ActionServlet*, que consulta al fitxer *struts-config.xml* quin mètode i de quina acció ha de cridar); aquest s'encarrega doncs de cridar la classe *Action* corresponent que interactua amb les classes que siguin necessàries de la part de la lògica de negoci (model) per tal d'atendre la petició rebuda. Un cop executat el codi corresponent, el model retorna un string "ActionForward" que representa la pàgina que el controlador ha de retornar a la vista, al client. Aquest procediment quedaria resumit en el següent esquema:



Il·lustració 5: funcionament arquitectònic

Donat el fet que s'utilitza el bastiment Hibernate per a l'accés a les dades, és lògic que també s'utilitzi el patró DAO per a evitar que tinguem les classes d'entitat plenes de codi d'accés a les dades, codi que pot ser necessari canviar si se'n modifica l'origen en un futur.

Així doncs s'utilitzarà DAO per tal de delegar l'accés a les dades en una classe independent de la classe d'entitat, així aquesta no s'haurà de modificar en futures revisions. Per tant s'haurà de crear una classe DAO per a cada classe d'entitat. Per tenir accés a les diferents classes DAO, s'ha utilitzat el patró *Factory Method*, per tal d'obtenir una classe que em permet construir els DAO de forma dinàmica.

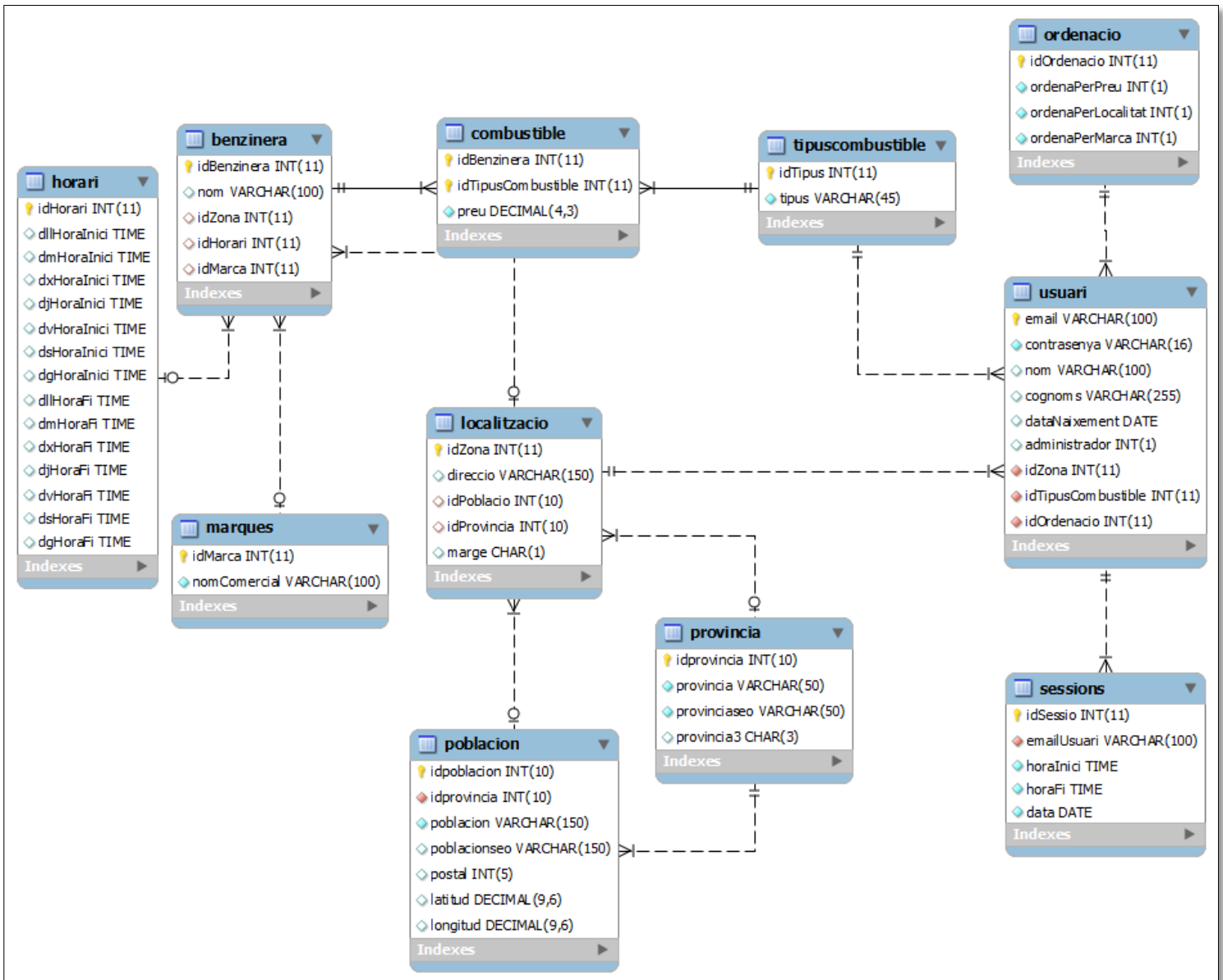
Aquesta classe, la *HibernateDaoFactory*, és una *Singleton* (una classe dissenyada per restringir la creació d'objectes o millor dit, dissenyada per a què no hi hagi més d'una instància a la vegada) que em dona accés als diferents DAO que es necessiten utilitzar des de les classes controladores. Bàsicament instancia (mitjançant *reflection*) el tipus de DAO que se li demana (utilitzant el *.class* que va especificat en els *getters*). Amb això el que obtenim és una abstracció de la instanciació dels *GenericDaoHibernateImpl* (classe pare de tots els DAO). Tot seguit es posa com a propietat el tipus d'objecte que gestionarà la DAO i la *session*.

La *GenericDAOHibernateImpl* és la classe, juntament amb la *HibernateDAOFactory*, més rellevant de la part servidora. Aquesta classe, bàsicament, proporciona els mètodes bàsics per a totes les classes DAO de l'aplicació; d'aquesta manera s'evita tenir el mateix codi a cada DAO (l'únic que canviaria seria el tipus de dada retornada o els paràmetres).

S'ha utilitzat a la part del client JSON, un format per al intercanvi de dades lleuger, que em permet treballar sense XML amb AJAX (permetent-me carregar de forma asíncrona les dades, sense obstaculitzar en cap moment la vista ni la lògica del negoci); és a dir, em permet crear una classe com si ho estigués fent des de una Action. Això ha estat necessari degut a la mancança de temps per a trobar una solució millor a la problemàtica de la càrrega de les localitats i de les marques, que calia comprovar que s'introduïen correctament.

3.2 – Diagrama d'entitats - relació.

En el següent diagrama es pot veure quines entitats estan representades a la base de dades, observant cada taula. També es pot veure com es relacionen mitjançant claus foranies. A l'annexa d'aquest treball s'hi pot trobar les sentències de creació de la base de dades i de les diferents taules; l'*script* amb els registres d'introducció de dades només (no es considera afegir-lo al treball com a annexa ja que les dades son relatives i ocuparia 422 planes) es pot trobar dintre de la carpeta del mateix nom en el fitxer *.war* entregat, el nom del mateix és *backup_tfc.sql*.



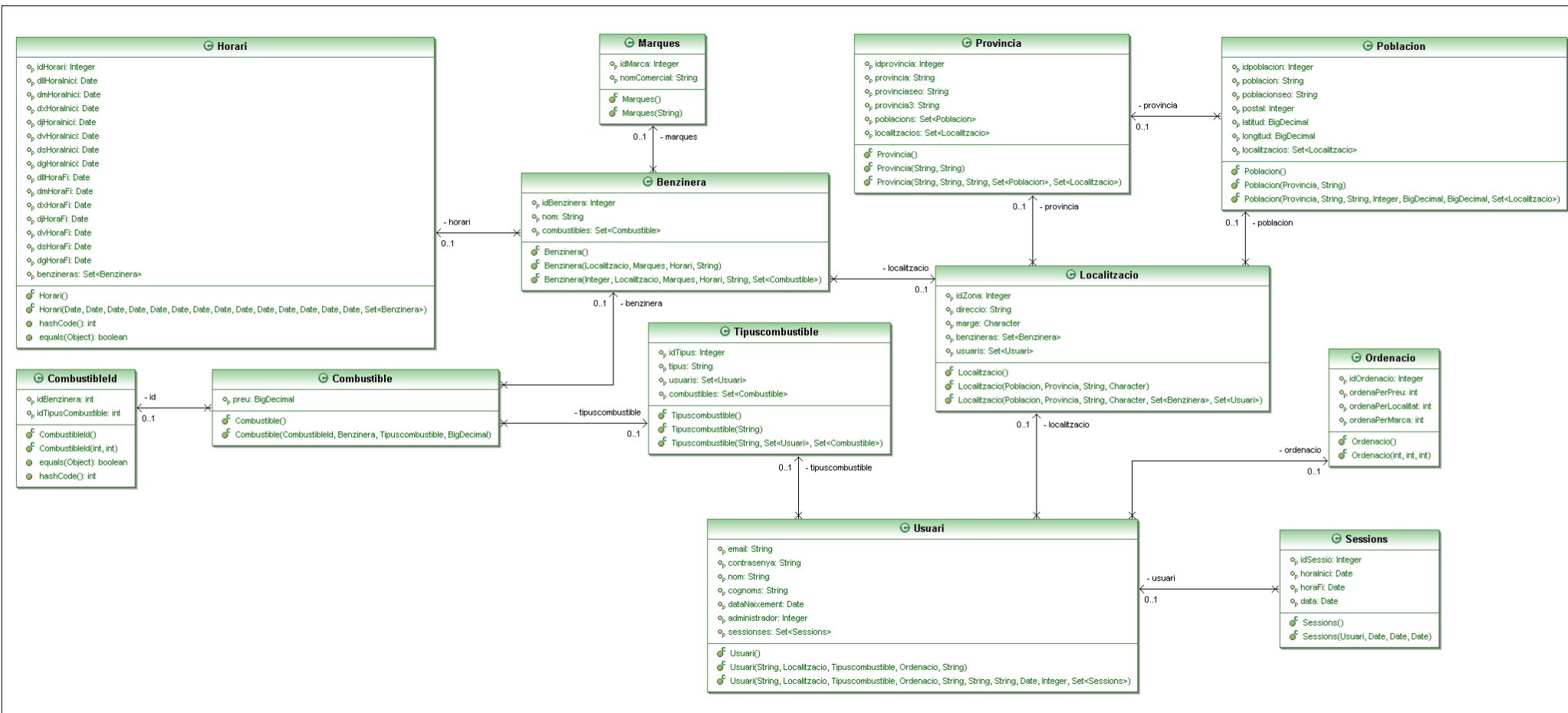
Il·lustració 6: Diagrama d'entitat – relació

Com es pot observar, hi ha hagut canvis respecte al diagrama d'entitat-relació de la segona entrega, tot seguit explicaré quins han estat i perquè s'han realitzat:

- **Taula Marques:** s'ha creat un "Unique Index" per tal de no permetre que es repeteixin els noms comercials de les marques. També s'ha augmentat la llargada del camp "nomComercial".
- **Taula Horari:** de la mateixa manera que amb marques, s'ha creat un índex per no permetre la repetició dels horaris, impeding tenir diferents claus primàries per als mateixos horaris.
- **Taula Ordenacio:** també ha estat necessari crear un índex per evitar repeticions de tipus d'ordenació.
- **Taula TipusCombustible:** s'ha creat un índex per evitar repeticions de tipus de combustibles.
- **Taula Localitzacio:** s'ha afegit una columna per indicar si la benzinera està situada al marge dret (D) o esquerra (E o I, segons el llenguatge del registre d'entrada). Es proposa en futures versions crear un *trigger* que controli que només es permet l'entrada de les lletres D, E i N (de no aplicable). S'han canviat els camps població i província i s'ha eliminat el camp codi postal, això ha estat així ja que s'ha aprofitat les taules "poblacion" i "provincia" del Sr. Javier Casares (javiercasares.com/blog/sql-poblacion-espana). Cal generar doncs les claus forànies *idProvíncia* i *idPoblacio* i adaptar-les al format d'origen de la taula (*unsigned type*). També s'ha hagut d'allargar el camp direcció per la falta d'espai en certes direccions introduïdes. Finalment, per assegurar l'integritat referencial de la taula s'ha afegit dues restriccions a cada camp *idPoblacio* i *idProvíncia* que assegurin que al actualitzar o al esborrar les poblacions i/o províncies a les taules d'origen, s'eviti si encara hi ha algun registre de la taula *Localitzacio* que els utilitzi (*ON DELETE RESTRICT, ON UPDATE CASCADE*).
- **Taula Benzinera:** al no disposar de la dada CIF, es canvia la clau primària per un codi de tipus *int* que s'autoincrementi en cada entrada. Per evitar problemes amb els noms, s'augmenta la grandària permesa del camp corresponent. De la mateixa manera que amb la taula *Localitzacio*, s'afegeixen regles de restricció per assegurar la integritat referencial als camps *idHorari*, *idMarca* i *idZona* (*ON DELETE RESTRICT, ON UPDATE CASCADE*).
- **Taula Combustible:** d'acord amb els canvis realitzats a la taula anterior cal que es modifiqui la clau primària (de *cifBenzinera* a *idBenzinera*). A més, s'estableix com a camp indispensable el camp *preu* (*NOT NULL*). També s'assegura la integritat referencial de les dades en els camps *idBenzinera* i *idTipusCombustible* (*ON DELETE RESTRICT, ON UPDATE CASCADE*).
- **Taula Poblacion:** S'assegura la integritat referencial del camp *idProvíncia* (*ON DELETE RESTRICT, ON UPDATE CASCADE*).
- **Taula Sessions:** S'assegura la integritat referencial del camp *emailUsuari* (*ON DELETE RESTRICT, ON UPDATE CASCADE*).
- **Taula Usuari:** de la mateixa manera, s'assegura la integritat referencial dels camps *idOrdenacio*, *idTipusCombustible* i *idZona* (*ON DELETE RESTRICT, ON UPDATE CASCADE*).

3.3 – Diagrama estàtic de classes.

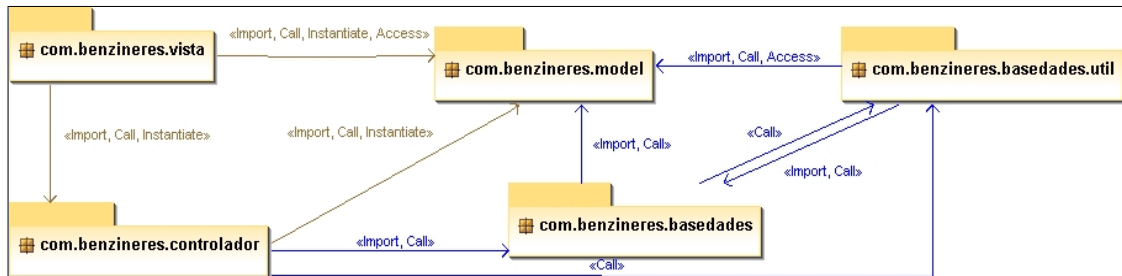
Aquest diagrama de classes està format per les classes que formen la part de la lògica del negoci, és a dir, el model dintre del patró MVC.



II-Il·lustració 7: Diagrama estàtic de classes

3.4 – Diagrama de paquets.

En la següent il·lustració es pot veure com l'aplicació està clarament separada per paquets de la forma que indica el patró MVC, interpretant una arquitectura per capes. Hi ha dos paquets que serveixen per guardar les classes responsables de tractar amb la base de dades i les que formen part dels patrons descrits anteriorment.



Il·lustració 8: Diagrama de paquets

Al paquet `com.benzineros.model` hi trobarem totes les classes que conformen la lògica de negoci, el model de l'aplicació, aquestes classes són les que majoritàriament s'emmagatzemen a la base de dades, tot i que hi podem trobar classes de sessió (és a dir, que només duren durant la sessió) que serveixen per controlar certs aspectes.

El paquet `com.benzineros.vista` conté totes les classes `Action` que són necessàries per a donar resposta al client via `ActionServlet`. Aquestes classes són cridades des del fitxer `struts-config.xml` que fa de controlador dintre del patró MVC que té Struts2 implementat. Les classes `Action` són les encarregades de cridar als controladors i a les DAO si fa falta.

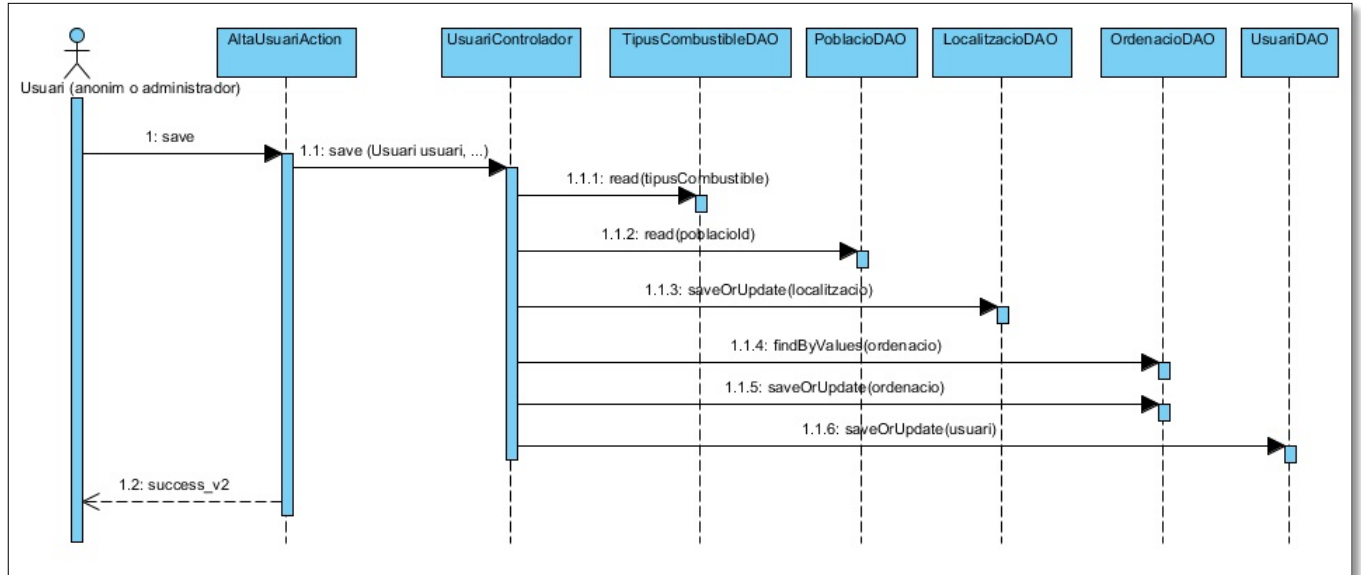
Al paquet `com.benzineros.controlador` hi trobarem les classes controladores de l'aplicació, és a dir, les classes que decideixen com donar resposta a les peticions fetes per les classes del paquet vista i que alhora interactuen amb les classes del paquet model, en el nostre cas de forma indirecte (via patró DAO).

Finalment, al paquet `com.benzineros.basedades` hi trobarem les classes que implementen l'accés a la base de dades, aquestes classes són les encarregades d'interactuar amb la base de dades i per tant se'n necessita una per a cada objecte que es vol fer persistir. El paquet `com.benzineros.basedades.util` conté classes que són necessàries per utilitzar Hibernate i els patrons esmentats de forma correcta.

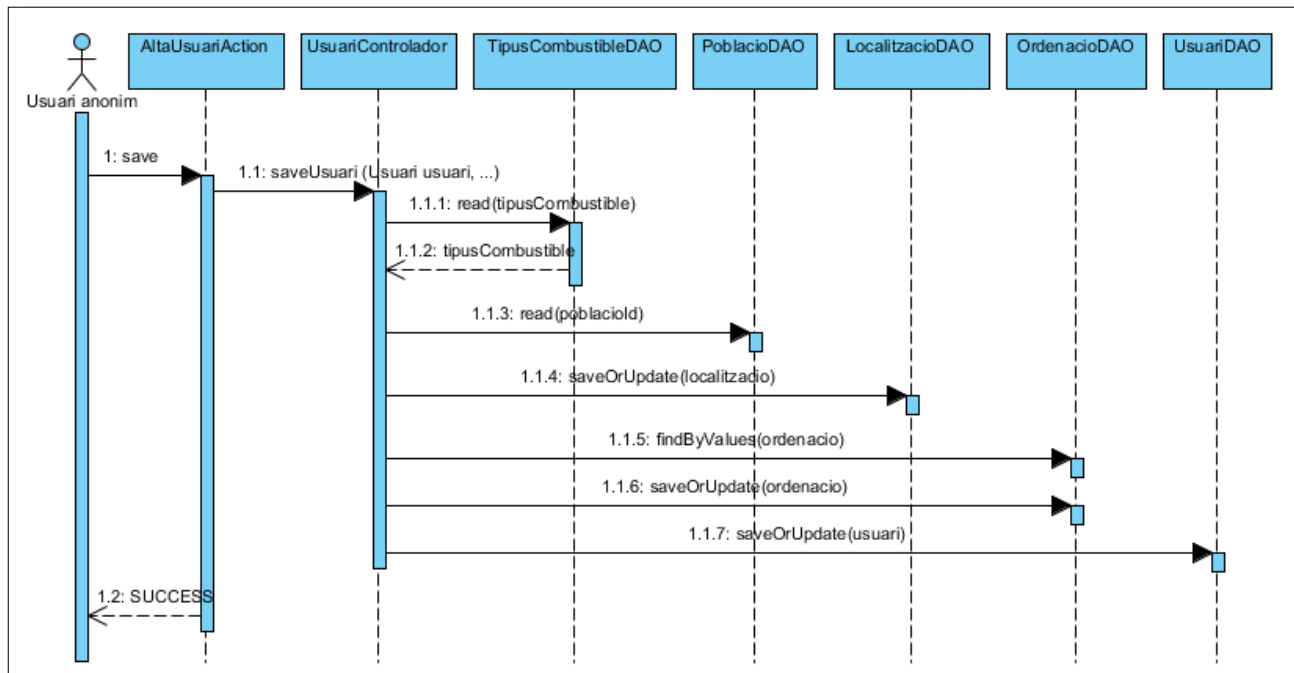
3.5 – Diagrames de seqüència.

En aquesta secció es mostraran els diferents diagrames de seqüència, com l'aplicació interactua amb les diferents classes i quins mètodes (entre d'altres) s'utilitzen en la execució dels casos d'us.

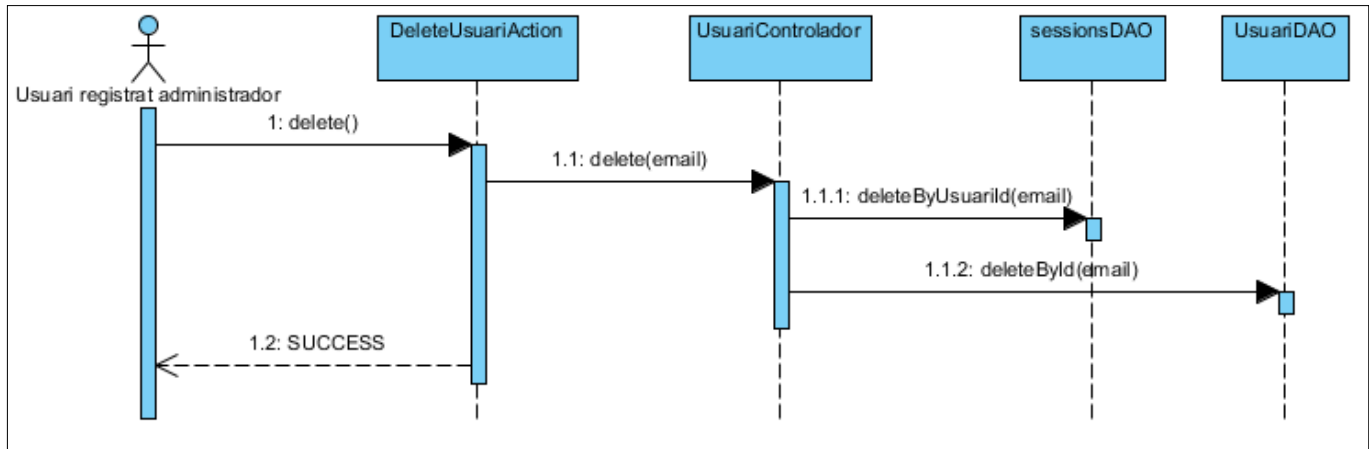
3.5.1 – Diagrames de seqüència del subsistema “Gestió Usuaris”.



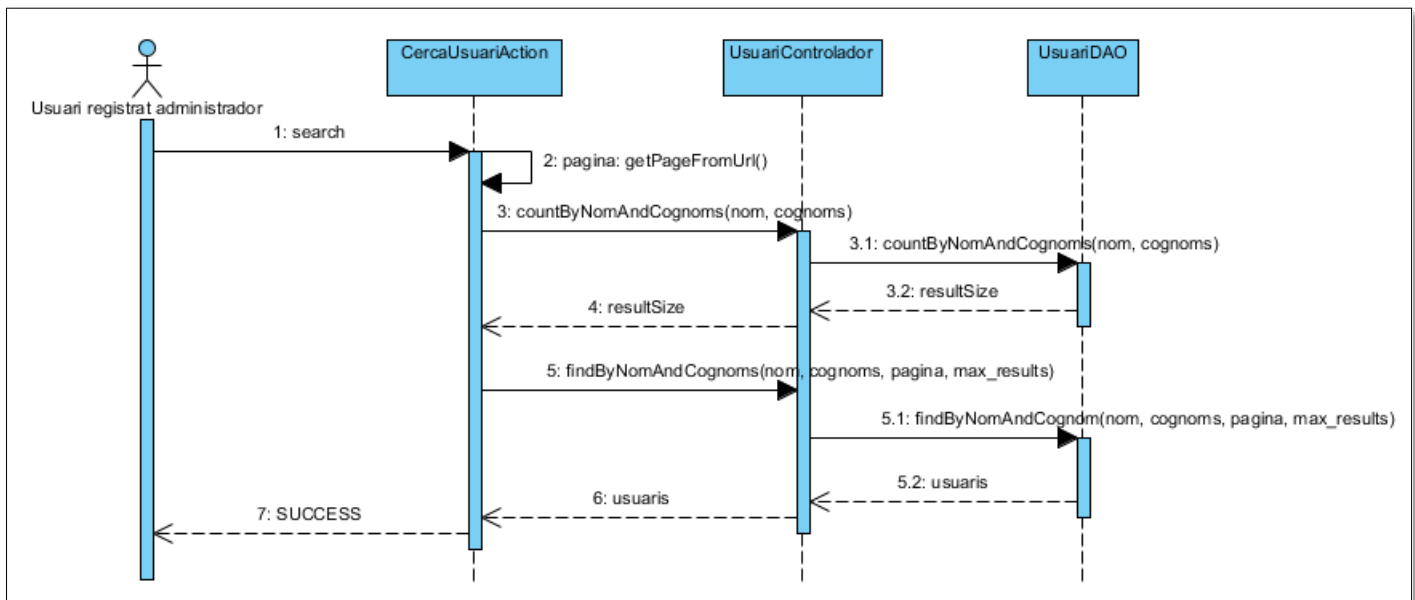
II-lustració 9: Diagrama de seqüència de “Alta Usuari”



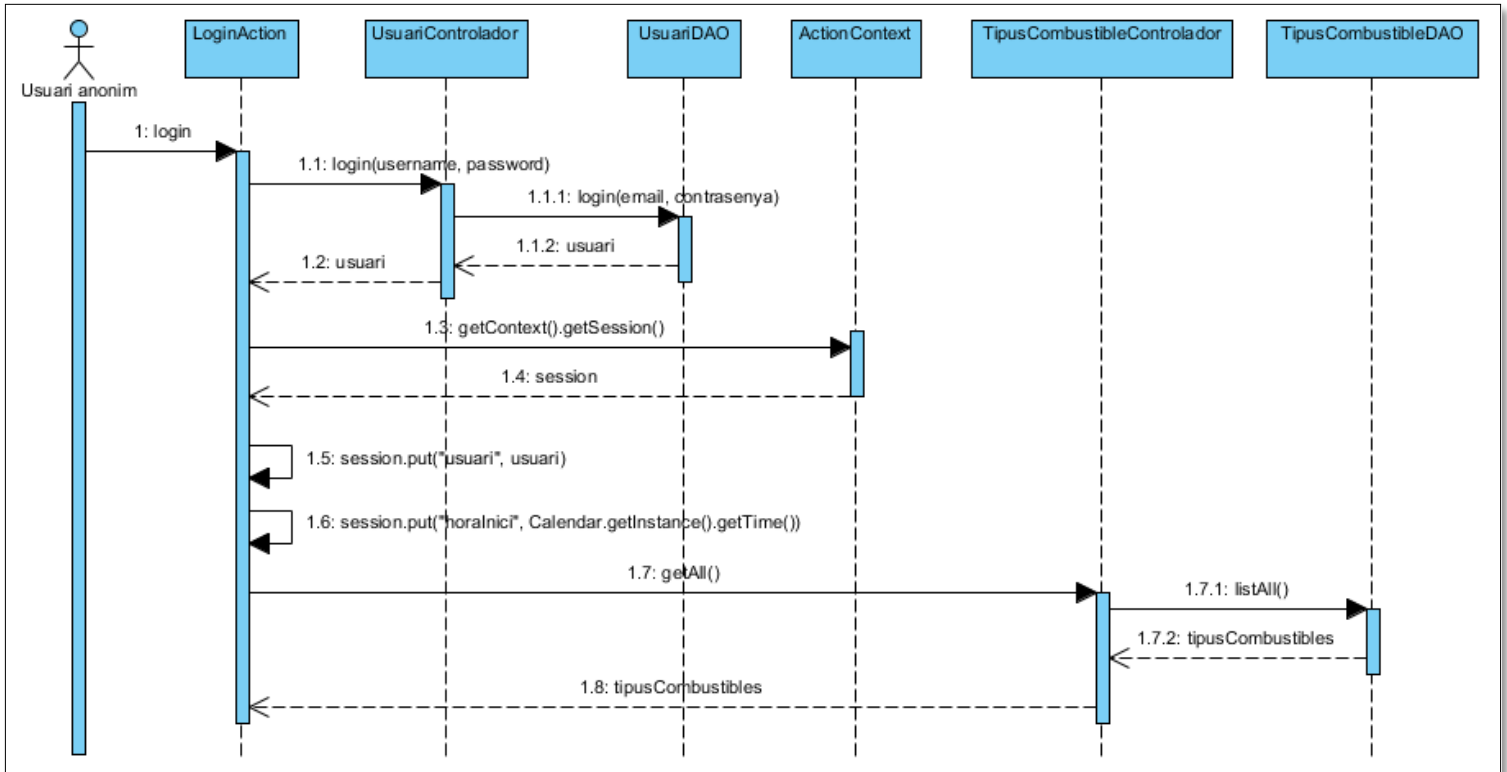
II-lustració 10: Diagrama de seqüència de “Modificar Usuari”



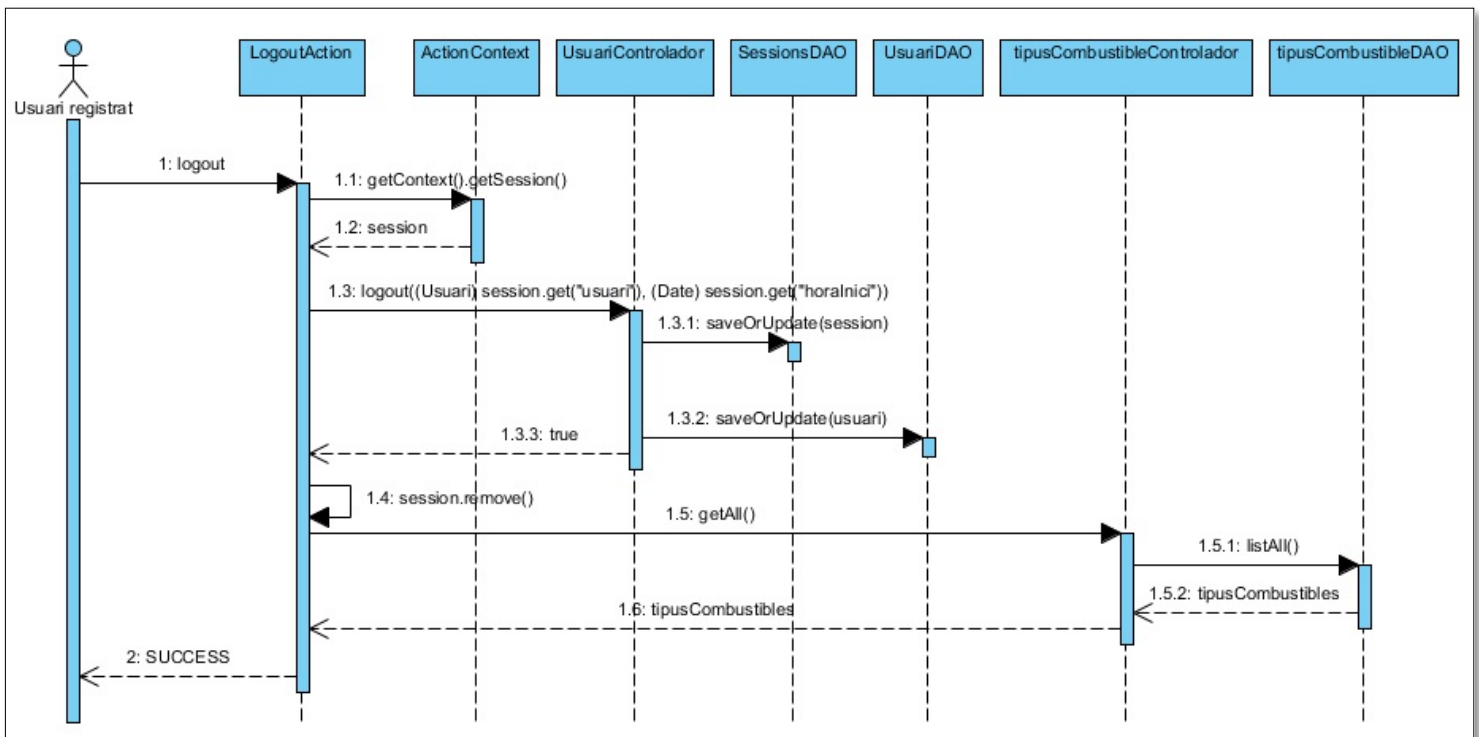
Il·lustració 11: Diagrama de seqüència de “Baixa Usuari”



Il·lustració 12: Diagrama de seqüència de “Cerca Usuari”

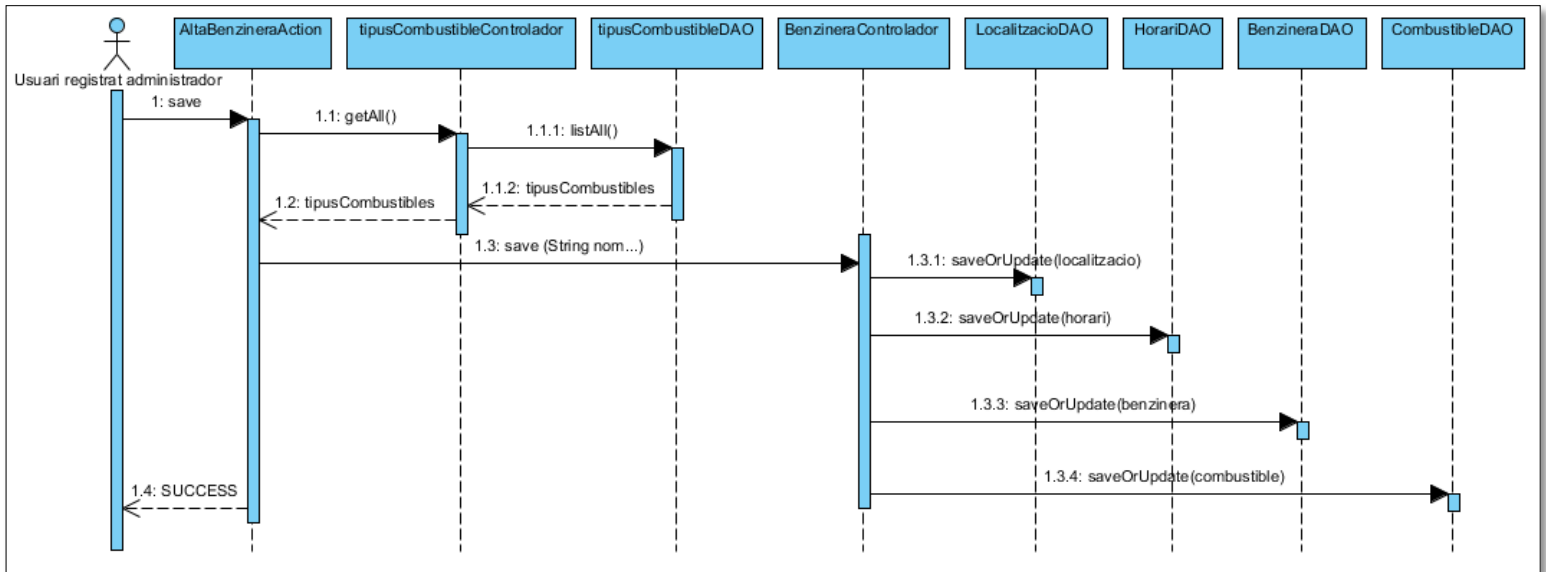


II-lustració 13: Diagrama de seqüència de "Iniciar Sessió"

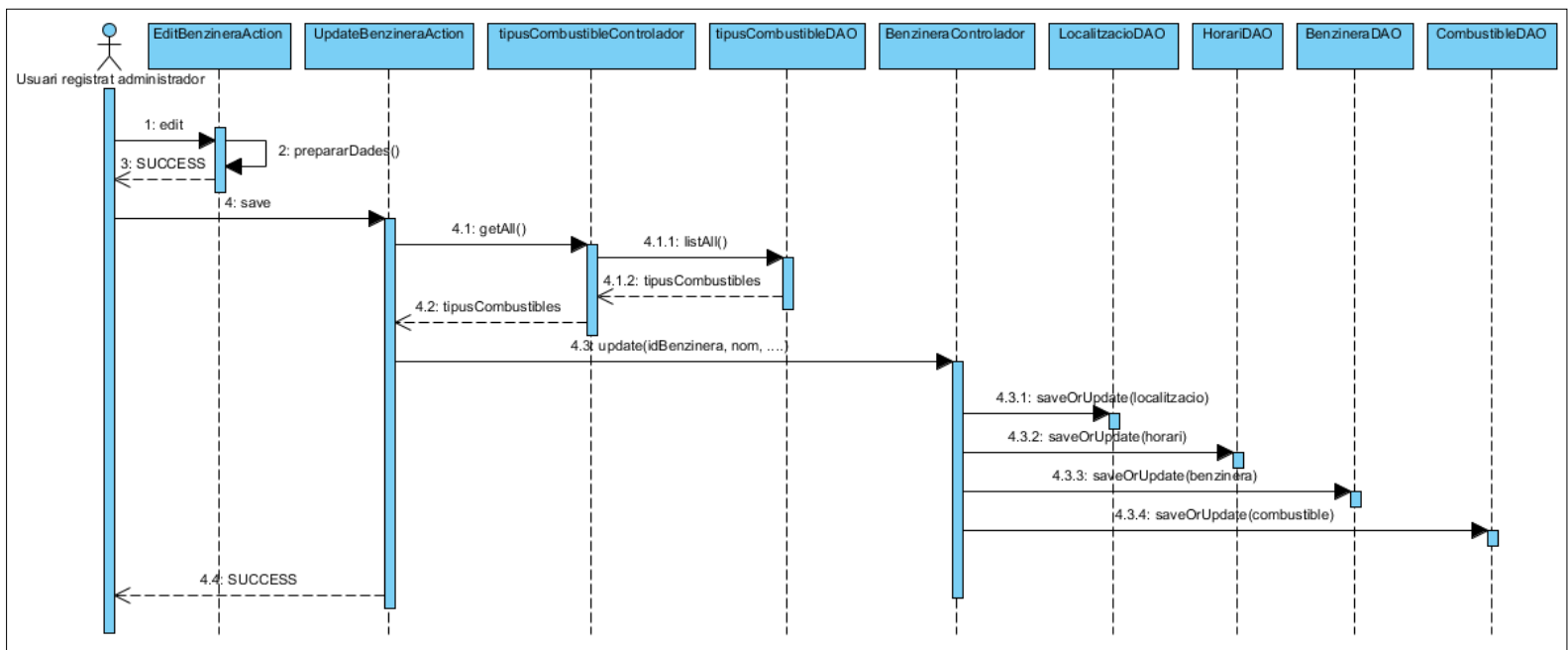


II-lustració 14: Diagrama de seqüència de "Finalitzar Sessió"

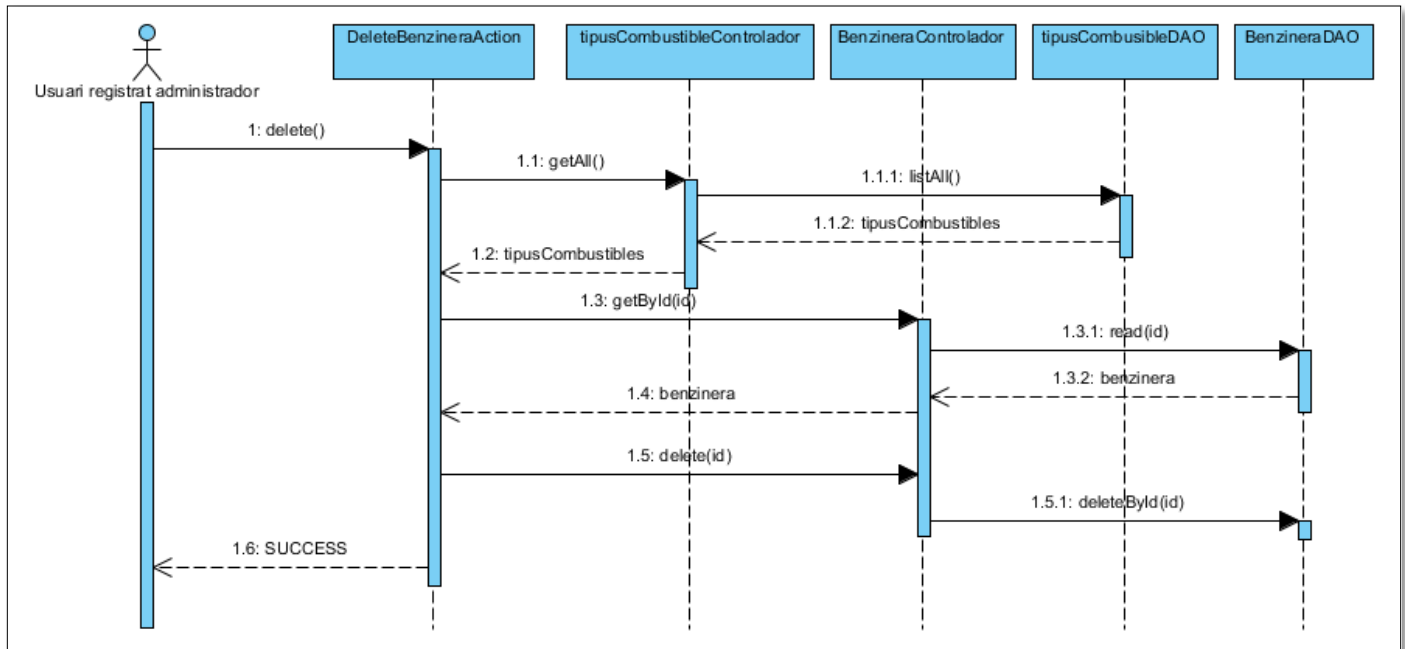
3.5.2 – Diagrames de seqüència del subsistema “Gestió Benzineres”.



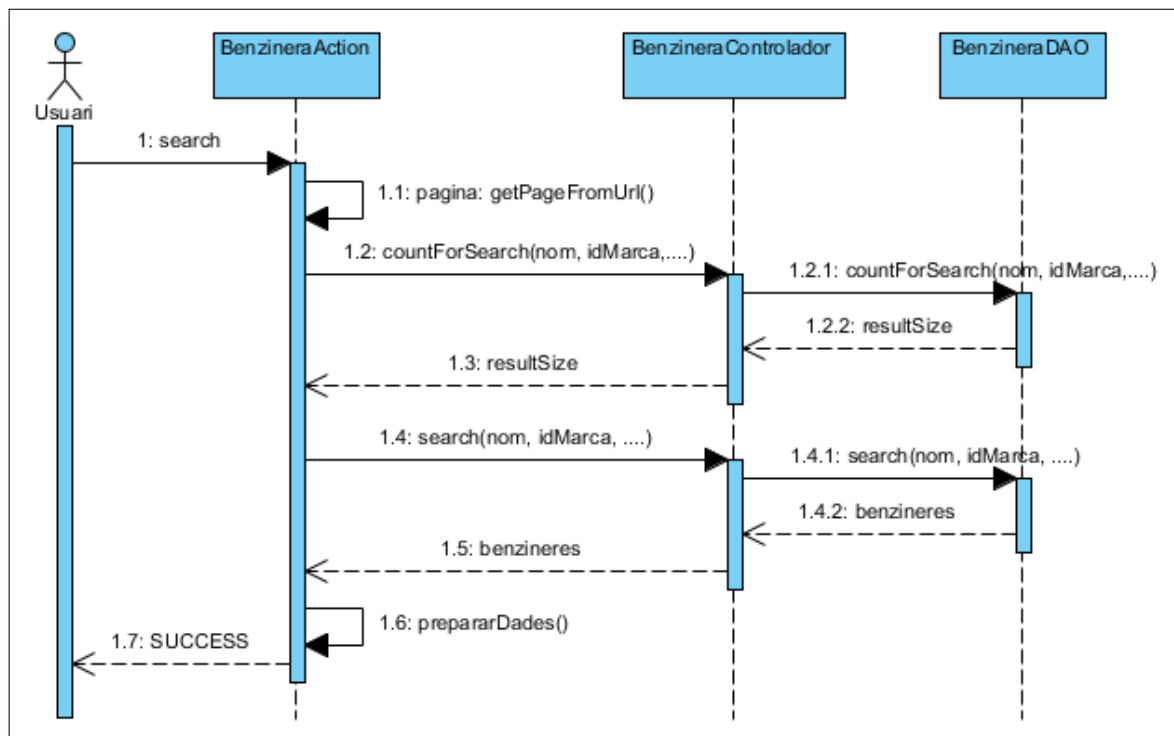
II-Il·lustració 15: Diagrama de seqüència de “Alta Benzinera”



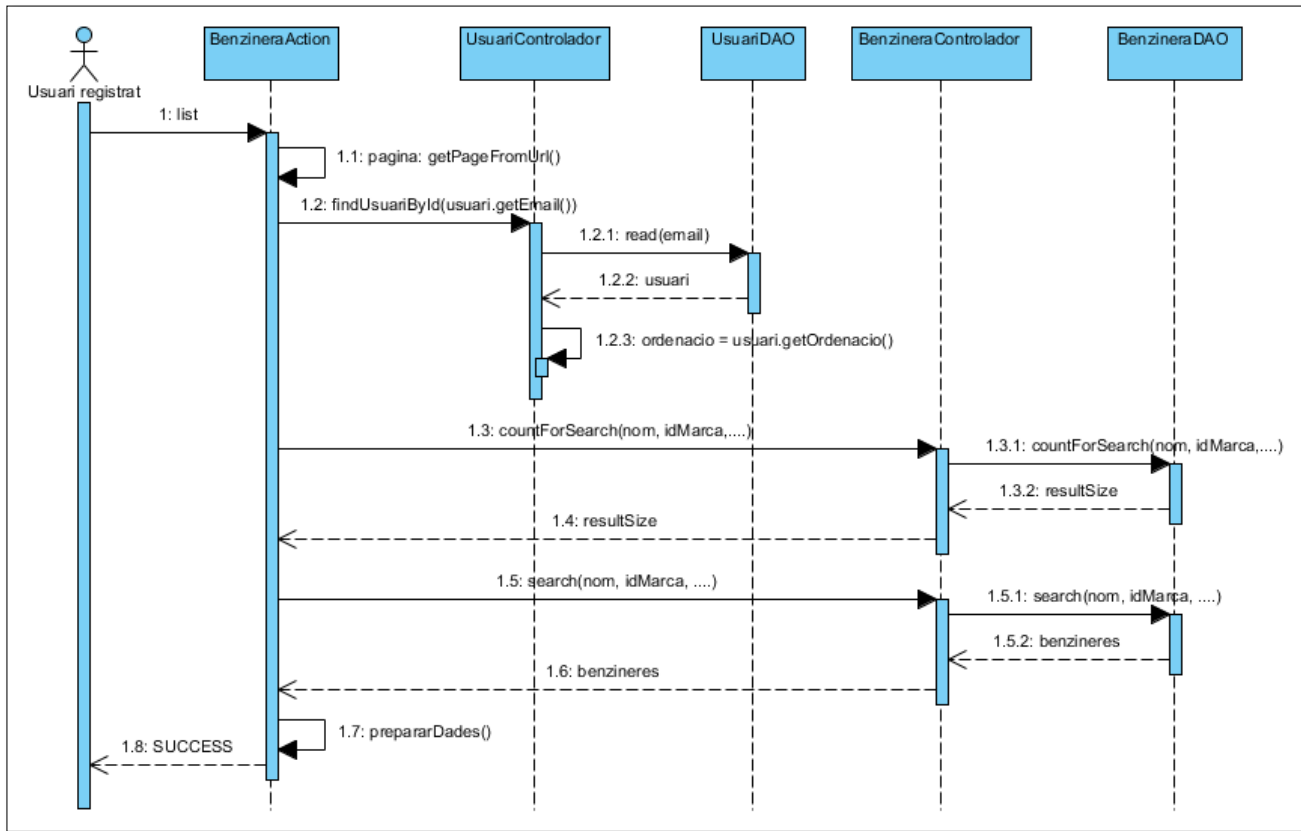
II-Il·lustració 16: Diagrama de seqüència de “Modificar Benzinera”



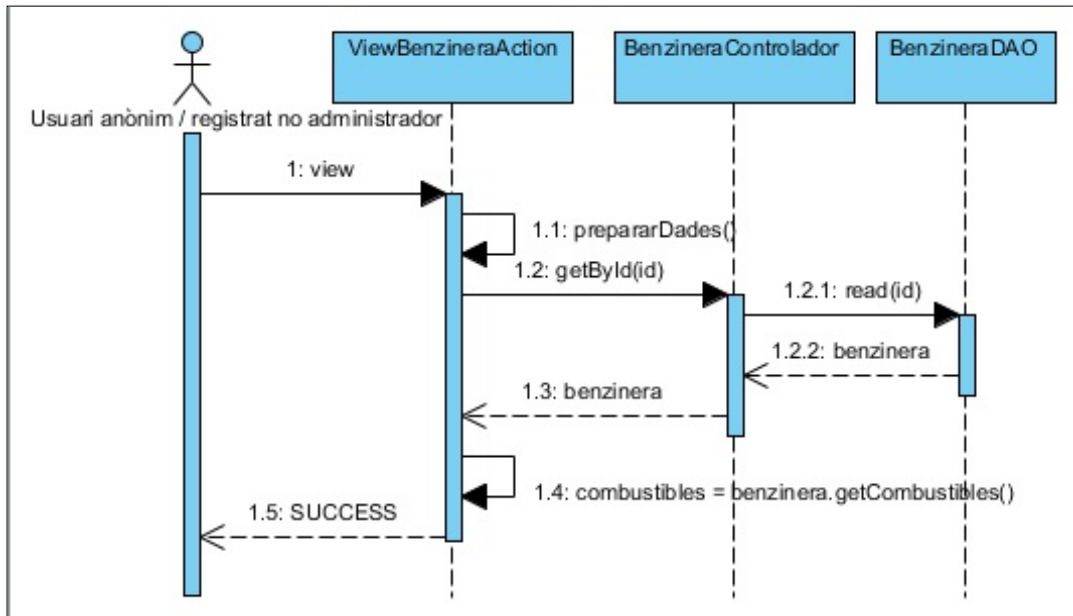
II-Il·lustració 18: Diagrama de seqüència de “Baixa Benzinera”



II-Il·lustració 17: Diagrama de seqüència de “Cerca Benzinera ordenació estàndard”



II-lustració 19: Diagrama de seqüència de “Cerca Benzinera ordenació usuari”



II-lustració 20: Diagrama de seqüència de “Veure detall benzinera”

3.6 – Disseny capa web.

Per a la capa web s'ha utilitzat Struts2 com a marc de treball, implementant així el patró MVC; així amb aquest patró s'aconsegueix enllaçar la capa de negoci amb la de presentació de forma independent. Amb aquest marc de treball tenim de controlador l'*ActionServlet*, les planes JSP fan la funció de vista i les classes Java de l'aplicació fan de model (el mapeig entre el model i la vista el realitza el fitxer *struts-config.xml*).

Les classes *Action* són les encarregades de gestionar la presentació, és a dir, són les que decideixen quines planes JSP es carreguen d'acord amb la petició que reben (d'acord amb el mètode executat). Aquestes classes són les que son cridades des del controlador (via el fitxer XML) i són les que s'encarreguen de cridar les classes que formen la lògica del negoci en el patró MVC implementat en el projecte.

Les planes JSP són necessàries per a generar les planes HTML que seran transportades mitjançant el protocol HTTP des del servidor web; aquestes planes HTML son la resposta que rep el contenidor web via protocol HTTP (del navegador web del client). Amb les planes JSP obtenim la possibilitat de introduir codi Java en una estructura HTML; aquesta possibilitat només serà utilitzada per a fer la presentació de les dades que ens arribin.

4 – Implementació.

Per al correcte funcionament de l'aplicació, els clients tant sols necessitaran un navegador (es recomana l'ús de Google Chrome o Mozilla Firefox). L'aplicació s'ha construït de tal manera que en un futur es pugui utilitzar altres mètodes d'accés com una aplicació per a *smartphones* (la intenció era fer una aplicació per a Android, però el temps i l'aprenentatge de la tecnologia necessària fou condicionant per no tirar-ho endavant). També es podrien fer aplicacions d'escriptoris per a tots els sistemes operatius que suportin la JVM (Màquina Virtual de Java), tot i que en aquest cas no tindria massa sentit.

Per la part del servidor, serà necessària la instal·lació de en primer lloc una plataforma Java EE SDK, ja que tot el projecte es basa en la tecnologia J2EE. A més, com a servidor d'aplicacions web es proposa la utilització d'un Apache Tomcat (un contenidor de lliure distribució i de codi obert), aprofitant que ha estat escrit en Java i, per tant, es pot executar en qualsevol màquina amb un sistema operatiu que disposi de la JVM. En la seva versió més actual (7.x) aquest fa tant de servidor d'aplicacions com de contenidor web i per tant no és necessari un servidor d'aplicacions específic.

Per a la persistència de les dades s'ha utilitzat MySQL (versió 5.5) com a sistema gestor de base de dades, per la potència del mateix i per tenir una llicència GNU GPL de lliure distribució. Per facilitar el treball s'ha utilitzat un programari d'ús gratuït que amb la seva interfície gràfica permet tractar amb més facilitat la gestió de la base de dades, estic parlant de MySQL Workbench CE (versió 5.2).

També és necessari l'ús del marc de treball Struts 2.x i Jakarta Ant (per a desplegar i generar els .war de l'aplicació). Com a IDE s'ha utilitzat l'Eclipse Helios, en la versió per a programadors Java EE.

Per a la generació del diagrama de classes i l'estàtic s'ha utilitzat un plugin d'Eclipse (eUML2) gratuït del Market d'Eclipse (i molt útil, per cert!). Per a la creació dels diagrames de cas d'us i de seqüència s'ha utilitzat Visual Paradigm, un programa de pagament que ha estat utilitzat amb una llicència d'avaluació.

5 – Breu anàlisi econòmic.

La valoració de l'actual aplicació pot diferir molt segons què és miri. Si el que es mira són les hores utilitzades pel desenvolupament total (des del disseny fins a la redacció de la memòria i la presentació virtual) aquest és un treball molt car: hi he dedicat 107 dies una mitjana de 4 hores diàries, és a dir, 428 hores. Tenint en compte que la persona que l'està fent es podria considerar un futur enginyer tècnic nivell Junior i que aquestes persones cobren uns 25€ l'hora estaríem parlant de 10.700€. Això sense tenir en compte que per a una versió comercial s'hi hauria d'afegir alguns costos de llicències (com per exemple de MySQL, del sistema operatiu, ...).

Evidentment, aquesta NO pot ser la valoració econòmica de l'aplicació realitzada: de les 428 hores dedicades moltíssimes han estat d'aprenentatge. També s'ha de tenir en compte que no totes les fases del treball les faria una mateixa persona, un mateix tècnic. Segurament en l'anàlisi de requisits, en el disseny i en la implementació les persones que hi treballarien serien diferents i tindrien costos diferents. A més, aquestes persones es trobarien dintre d'una empresa, tenint el procés de fabricació de solucions molt optimitzat. Si li hagués de posar un preu, penso que aproximadament en el món real seria de uns 5.000€ o 6.000€ (marge de beneficis inclòs).

6 – Conclusions.

L'aprenentatge d'una tecnologia sempre comporta molt esforç. Penso que aquest esforç ha estat recompensat amb escreix ja que superar un repte sempre dona satisfacció personal i si a sobre t'emportes un coneixement que pot ser útil en la vida real encara millor.

Tot i això, també vull quedar clar que no ha estat gens fàcil, m'he vist per primera vegada en TOTA la carrera desenvolupant una aplicació de cap a peus; penso que això és quelcom que cal que el futur grau en informàtica corregeixi, penso que ens hauríem de saber els passos amb els ulls tancats.

L'aprenentatge de J2EE i tota la tecnologia que ha estat envoltada en el projecte ha estat molt dura, en alguns moments desesperant tot i que jo en tenia algun coneixement gràcies a una assignatura optativa cursada a la Universitat de Girona.

Me les he vist especialment magres alhora de desenvolupar les primeres funcionalitats de l'aplicació. Struts2, tot i la fama de ser una tecnologia fàcil de comprendre, m'ha semblat força difícil. La creació dels *JSPs* ha estat una feina molt dura, de la qual he après molt poc, considero que resulta força poc entenedor la utilització dels *tags* i penso que aquí hi ha molt treball a fer. Potser la futura versió d'HTML corregirà aquestes mancances? La gran quantitat i varietat de tecnologia i a la vegada complexa ha estat més un impediment que un incentiu.

En general considero que ha estat molt interessant fer el treball, m'he vist involucrat en la creació des de zero d'una aplicació, incloent-hi els requisits que el suposat client vol. Penso que UML és quelcom que, personalment, m'ajuda poc i crec que hi hauria d'haver una forma més concreta de planificar el disseny i la implementació de les aplicacions. He observat de molt a prop la potència de J2EE i de totes les tecnologies que l'envolten; s'ha fet constant la necessitat de desenvolupar una aplicació d'aquest nivell utilitzant les millors pràctiques i sense deixar d'utilitzar cap dels patrons que hi puguin intervenir. Malgrat tot, estic content amb la realització d'aquesta aplicació però m'he quedat amb les ganes de seguir millorant-la!

7 – Bibliografia.

7.1 – Bibliografia escrita.

Hibernate in Action; Cristian Bauer i Gavin King; Manning Publications; ISBN 1932394-15-X

Manual de referència JSP; Phil Hanna ; McGraw-Hill Osborne Media ; ISBN 84-481-3264-5

Head First Servlets&JSP;Bryan Basham, Kathy Sierra i Bert Bates; O'Reilly; ISBN 978-0-596-511668-0

J2EE-Una plataforma de components distribuïda(Assignatura UOC EPCSD); Josep Maria Camps Riba

7.2 – Consultes a la web.

[Hibernate Reference Tools Guide](#)

[Hibernate Tools Reference Guide \(JBoss Developer Studio 3.0\)](#)

[Hibernate Tools Reference Guide \(JBoss Developer Studio 4.0\)](#)

[Mi primera web con Struts](#)

[Struts 2 Framework Tutorial - Session 1 - Introduction to Struts 2 Framework](#)

[Struts 2 MVC Framework - Session 2 - Struts 2 Simple Example and Configuration](#)

[Struts 2 MVC Framework - Session 3 - Struts 2 actions / ActionSupport / ModelDriven / wildcards](#)

[Struts 2 Interceptors - Struts 2 Framework Session 4 - Interceptors](#)

[Tutorial: Create Struts2 Hibernate Example in Eclipse](#)

[Apache Struts 2 Framework Course](#)

[JSP / Servlets Course](#)

[Aplicación que permite reservar a un profesor para una tutoría.](#)

[SQL de poblaciones españolas \(con código postal, latitud y longitud\)](#)

[Hibernate Tools y la generación de código](#)

[Ministerio de Industria, Turismo y Comercio - Geoportal Hidrocarburos](#)

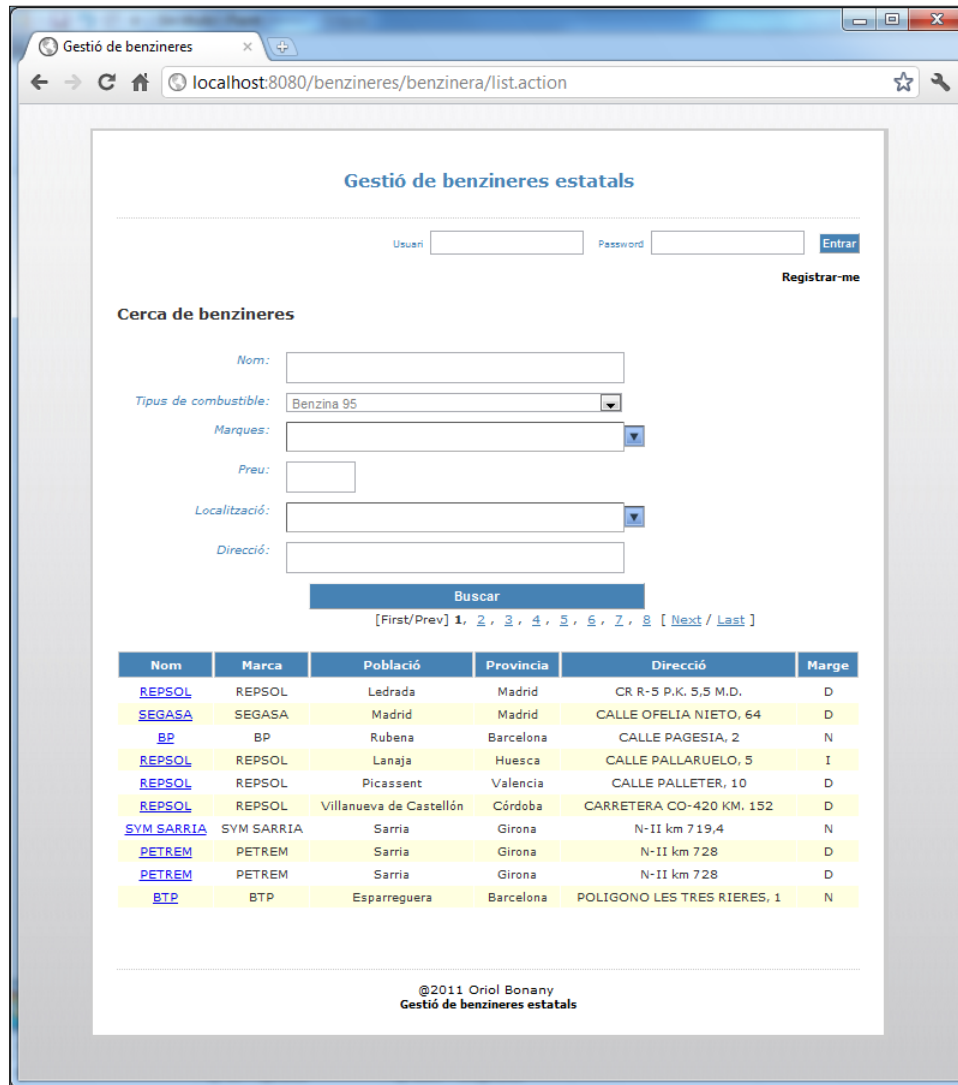
<http://www.proactiva-calidad.com/java/index.html>

[Ingeniería inversa con Eclipse](#)

[Hibernate - Parte 1: Persistiendo Objetos Simples usando Mapeos en XML](#)

8 – Annexes.

8.1 – Interfície de l’aplicació.

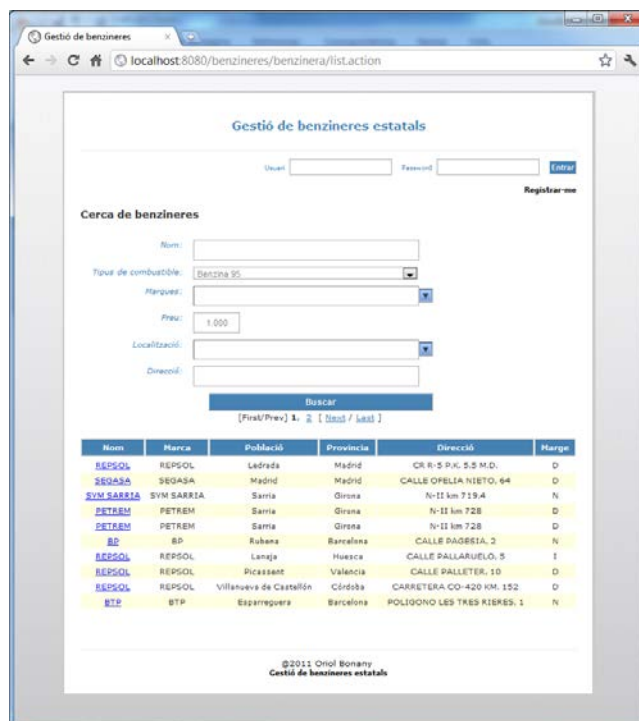


Il·lustració 21: Pàgina inicial.

A la il·lustració 21 s’aprecia la pàgina quan visites la pàgina sense que t’hagis autenticat.

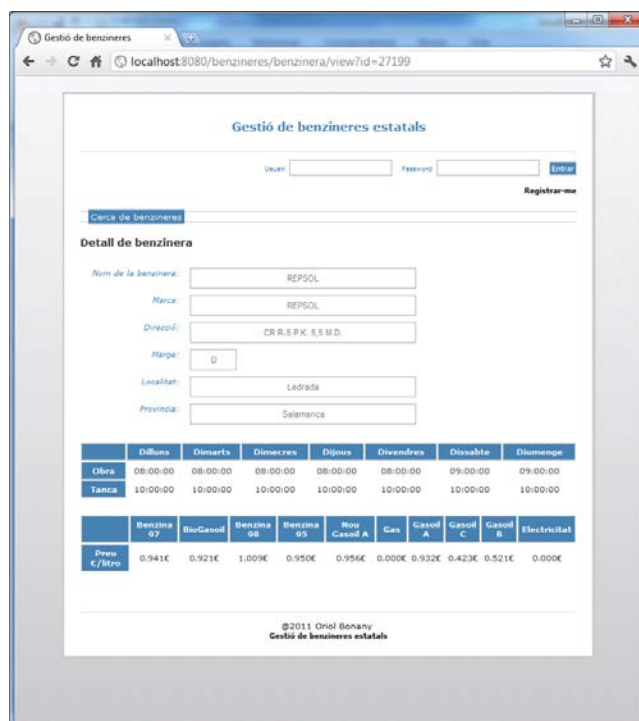
Tot seguit, a la il·lustració 22 podem apreciar la pàgina resultant d’una cerca per preu d’un usuari anònim; l’aplicació ens mostra aquelles benzineres (ordenades de menor a major preu) que tenen el preu de la benzina 95 inferior a 1€/l. La cerca d’una benzinera d’un usuari registrat o d’un usuari anònim només varia en la ordenació; a l’usuari anònim se li ordena per preu (de menor a major) i a l’usuari registrat se li ordena segons els criteris utilitzats i utilitzant la prioritat entre ells (per exemple, se li ordena primer per localitat, després de menys a més preu i finalment per marca de la benzinera).

Al ser usuari no administrador, no es poden esborrar benzineres, a la il·lustració 24 es pot veure com apareix una última columna amb enllaços per activar l’acció “eliminar”.

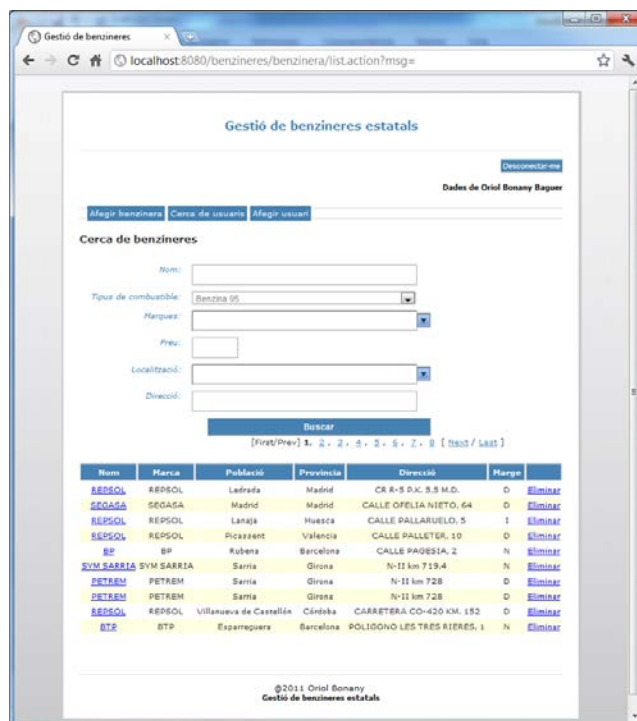


Il·lustració 22: Pàgina després de fer una cerca essent un usuari no administrador.

A la il·lustració, la 23, podem observar com es mostren les dades de la benzinera que ha sortit en primer lloc al llistat realitzat a la il·lustració 22, s'hi accedeix polsant el nom de la benzinera a la qual li volem veure els detalls. Aquesta vista només és accessible des dels usuaris anònims i des dels usuaris no administradors. Als usuaris administradors se'ls hi mostra la pàgina de la il·lustració 25.

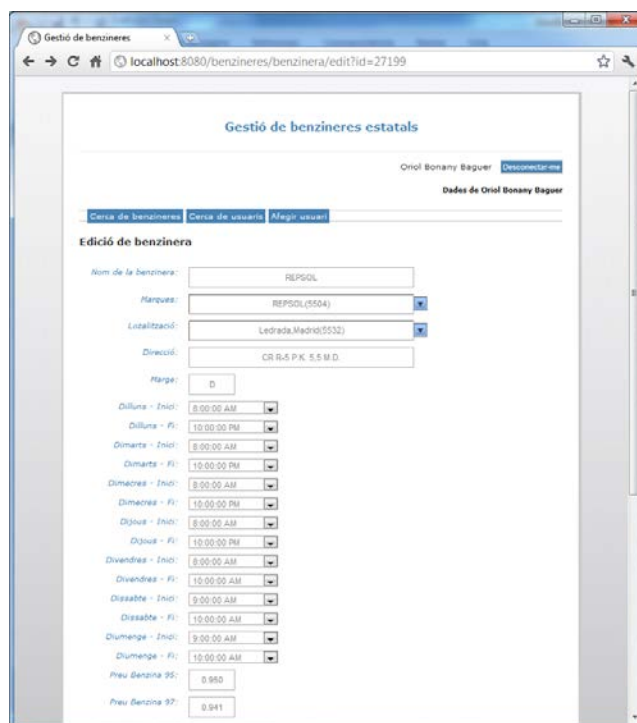


Il·lustració 23: Pàgina que mostra el detall d'una benzinera essent un usuari no administrador.

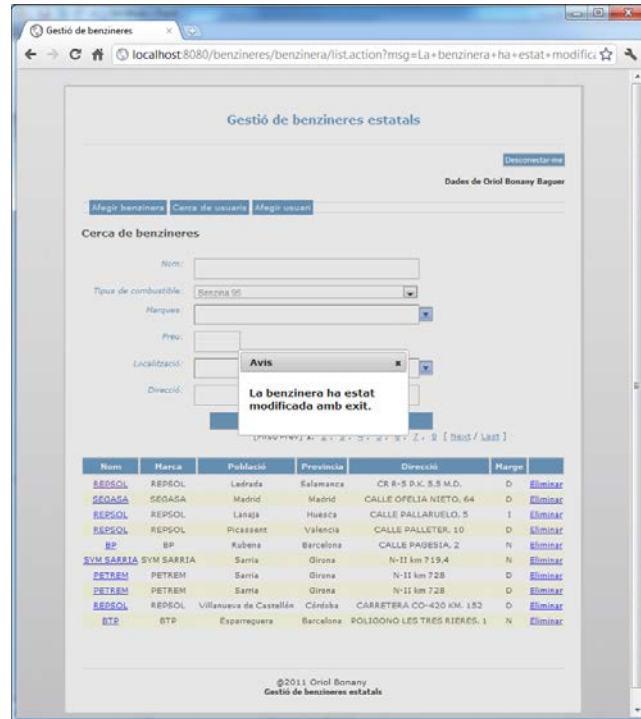


II-lustració 24: Pàgina de cerca essent un usuari registrat administrador.

A la il·lustració 24 cal fixar-se que han aparegut els botons que donen accés a les funcionalitats les quals la resta d'usuaris no poden accedir. La il·lustració 25 mostra com es donen d'alta (també modifiquen i consulten) les dades d'una benzinera des de la interfície de un usuari administrador; la il·lustració 26 mostra el missatge de confirmació que es rep al actualitzar una benzinera.

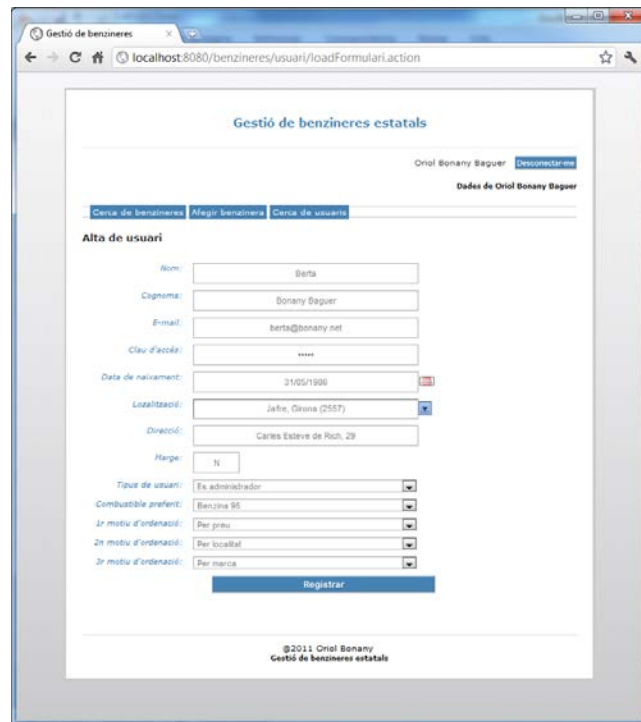


II-lustració 25: Pàgina mentre es modifiquen les dades d'una benzinera.

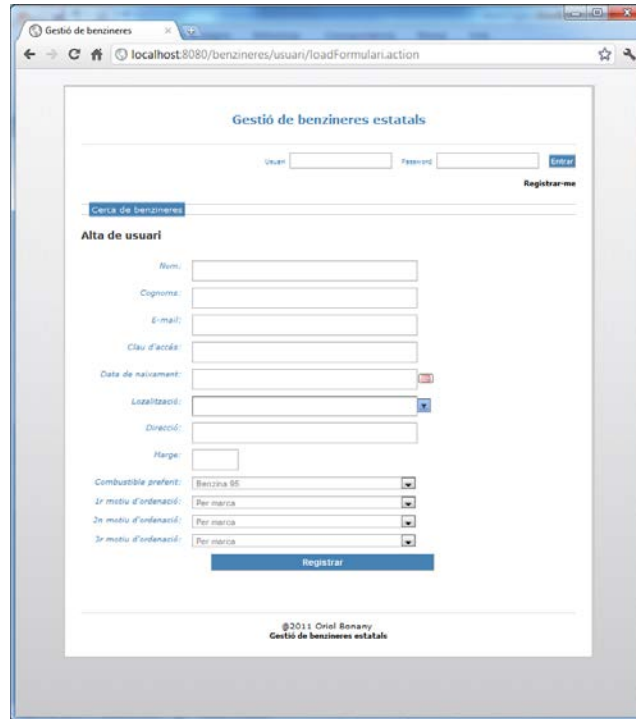


Il·lustració 26: Pàgina després d’haver modificat una benzinera.

A la il·lustració 27 es mostra el formulari que cal omplir per donar d’alta un usuari, en aquest cas la interfície és la de un usuari administrador; a la il·lustració 28 es pot observar quin aspecte té la pàgina quan és un usuari anònim que es registre (la característica més important és que no permet l’elecció “Administrador / No Administrador”)

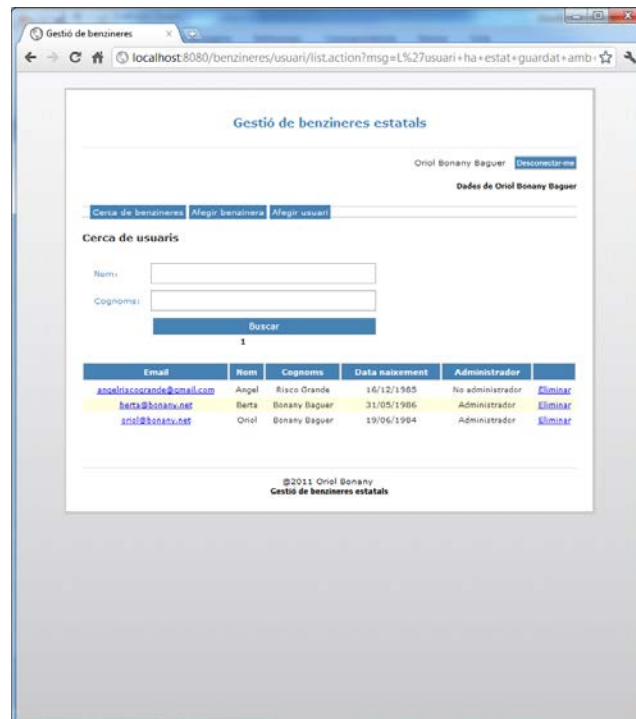


Il·lustració 27: Pàgina d’alta d’un nou usuari essent un usuari registrat administrador.

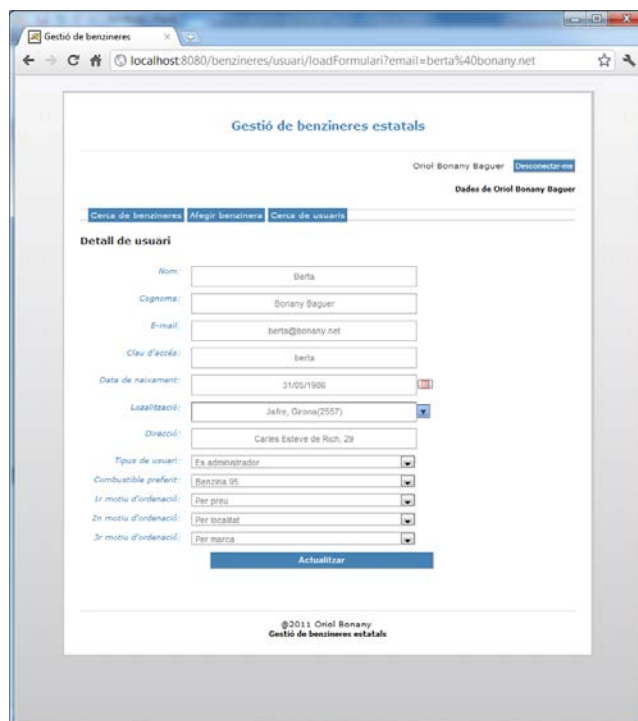


Il·lustració 28: Pàgina de registre de nou usuari essent un usuari anònim.

A la il·lustració 29 es pot veure la pàgina per a fer cerques d'usuaris, el funcionament és senzill: es pot posar el nom i/o un o dos cognoms, inclòs s'hi pot posar parts de un nom o de un cognom (per exemple, a la llista de la il·lustració es podria fer una cerca on al camp cognoms hi aparegués "bon" i el resultat seria els dos usuaris que tenen de cognom Bonany).

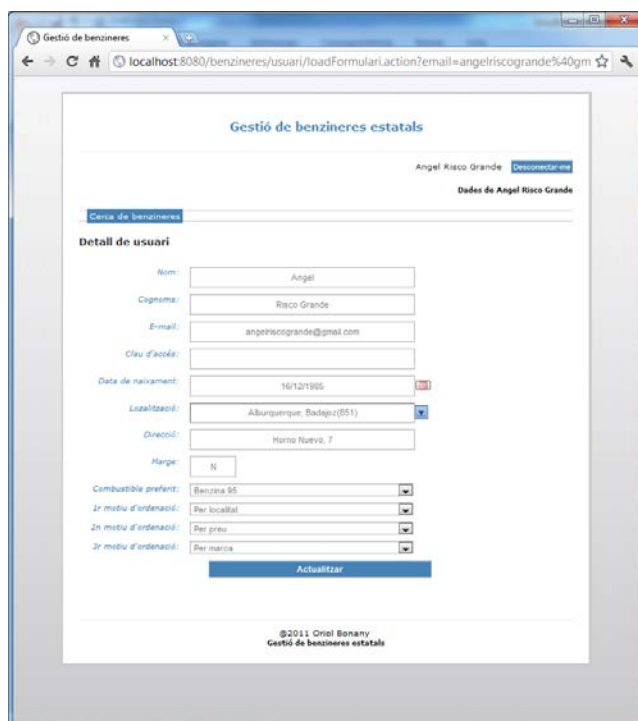


Il·lustració 29: Pàgina de cerca d'usuaris essent administrador.



Il·lustració 30: Pàgina de consulta i actualització de les dades d'un usuari essent administrador.

A la il·lustració 30 hi podem veure les dades requerides per a l'actualització de un usuari, aquí s'hi accedeix senzillament polsant sobre el nom de l'usuari que es vol modificar; cal ser administrador per accedir-hi. A la il·lustració 31 es pot veure com és essent un usuari no administrador; s'hi accedeix polsant a "Dades de...", a dalt a la dreta del formulari.



Il·lustració 31: Pàgina de consulta i actualització de les dades pròpies d'un no administrador.

8.2 – Script .sql de creació de la base de dades.

```

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

DROP SCHEMA IF EXISTS `tfc` ;
CREATE SCHEMA IF NOT EXISTS `tfc` DEFAULT CHARACTER SET utf8 ;
USE `tfc` ;

-----
-- Table `horari`
-----

DROP TABLE IF EXISTS `horari` ;

CREATE TABLE IF NOT EXISTS `horari` (
  `idHorari` INT(11) NOT NULL AUTO_INCREMENT ,
  `dllHoraInici` TIME NULL DEFAULT NULL ,
  `dmHoraInici` TIME NULL DEFAULT NULL ,
  `dxHoraInici` TIME NULL DEFAULT NULL ,
  `djHoraInici` TIME NULL DEFAULT NULL ,
  `dvHoraInici` TIME NULL DEFAULT NULL ,
  `dsHoraInici` TIME NULL DEFAULT NULL ,
  `dgHoraInici` TIME NULL DEFAULT NULL ,
  `dllHoraFi` TIME NULL DEFAULT NULL ,
  `dmHoraFi` TIME NULL DEFAULT NULL ,
  `dxHoraFi` TIME NULL DEFAULT NULL ,
  `djHoraFi` TIME NULL DEFAULT NULL ,
  `dvHoraFi` TIME NULL DEFAULT NULL ,
  `dsHoraFi` TIME NULL DEFAULT NULL ,
  `dgHoraFi` TIME NULL DEFAULT NULL ,
  PRIMARY KEY (`idHorari`) ,
  UNIQUE INDEX `UNIC` (`dllHoraInici` ASC, `dmHoraInici` ASC,
`dxHoraInici` ASC, `djHoraInici` ASC, `dvHoraInici` ASC,
`dsHoraInici` ASC, `dgHoraInici` ASC, `dllHoraFi` ASC, `dmHoraFi`
ASC, `dxHoraFi` ASC, `djHoraFi` ASC, `dvHoraFi` ASC, `dsHoraFi`
ASC, `dgHoraFi` ASC) )
ENGINE = InnoDB
AUTO_INCREMENT = 84
DEFAULT CHARACTER SET = latin1;

-----
-- Table `marques`
-----

DROP TABLE IF EXISTS `marques` ;

CREATE TABLE IF NOT EXISTS `marques` (
  `idMarca` INT(11) NOT NULL AUTO_INCREMENT ,
  `nomComercial` VARCHAR(100) NOT NULL ,
  PRIMARY KEY (`idMarca`) ,
  UNIQUE INDEX `nomComercial_UNIQUE` (`nomComercial` ASC) )
ENGINE = InnoDB
AUTO_INCREMENT = 7548
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `provincia`
-----
DROP TABLE IF EXISTS `provincia` ;

CREATE TABLE IF NOT EXISTS `provincia` (
  `idprovincia` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT ,
  `provincia` VARCHAR(50) NOT NULL ,
  `provinciaseo` VARCHAR(50) NOT NULL ,
  `provincia3` CHAR(3) NULL DEFAULT NULL ,
  PRIMARY KEY (`idprovincia`) ,
  UNIQUE INDEX `provinciaseo` (`provinciaseo` ASC) )
ENGINE = InnoDB
AUTO_INCREMENT = 53
DEFAULT CHARACTER SET = latin1;

-----
-- Table `poblacion`
-----
DROP TABLE IF EXISTS `poblacion` ;

CREATE TABLE IF NOT EXISTS `poblacion` (
  `idpoblacion` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT ,
  `idprovincia` INT(10) UNSIGNED NOT NULL ,
  `poblacion` VARCHAR(150) NOT NULL ,
  `poblacionseo` VARCHAR(150) NULL DEFAULT NULL ,
  `postal` INT(5) UNSIGNED ZEROFILL NULL DEFAULT NULL ,
  `latitud` DECIMAL(9,6) NULL DEFAULT NULL ,
  `longitud` DECIMAL(9,6) NULL DEFAULT NULL ,
  PRIMARY KEY (`idpoblacion`) ,
  UNIQUE INDEX `poblacionseo` (`poblacionseo` ASC) ,
  UNIQUE INDEX `lugar` (`latitud` ASC, `longitud` ASC) ,
  INDEX `idprovincia` (`idprovincia` ASC) ,
  INDEX `idprovincia_fk` (`idprovincia` ASC) ,
  CONSTRAINT `idprovincia_fk`
    FOREIGN KEY (`idprovincia`)
    REFERENCES `provincia` (`idprovincia`)
    ON UPDATE CASCADE)
ENGINE = InnoDB
AUTO_INCREMENT = 8175
DEFAULT CHARACTER SET = latin1;

-----
-- Table `localitzacio`
-----
DROP TABLE IF EXISTS `localitzacio` ;

CREATE TABLE IF NOT EXISTS `localitzacio` (
  `idZona` INT(11) NOT NULL AUTO_INCREMENT ,
  `direccio` VARCHAR(150) NULL DEFAULT NULL ,
  `idPoblacio` INT(10) UNSIGNED NULL DEFAULT NULL ,
  `idProvincia` INT(10) UNSIGNED NULL DEFAULT NULL ,
  `marge` CHAR(1) NULL DEFAULT NULL ,
  PRIMARY KEY (`idZona`) ,

```

```

INDEX `idProvincia_FK` (`idProvincia` ASC) ,
INDEX `idPoblacio_FK` (`idPoblacio` ASC) ,
CONSTRAINT `idPoblacio_FK`
  FOREIGN KEY (`idPoblacio` )
  REFERENCES `poblacion` (`idpoblacion` )
  ON UPDATE CASCADE,
CONSTRAINT `idProvincia_FK`
  FOREIGN KEY (`idProvincia` )
  REFERENCES `provincia` (`idprovincia` )
  ON UPDATE CASCADE)
ENGINE = InnoDB
AUTO_INCREMENT = 8867
DEFAULT CHARACTER SET = latin1;

-----
-- Table `benzinera`
-----

DROP TABLE IF EXISTS `benzinera` ;

CREATE TABLE IF NOT EXISTS `benzinera` (
  `idBenzinera` INT(11) NOT NULL AUTO_INCREMENT ,
  `nom` VARCHAR(100) NULL DEFAULT NULL ,
  `idZona` INT(11) NULL DEFAULT NULL ,
  `idHorari` INT(11) NULL DEFAULT NULL ,
  `idMarca` INT(11) NULL DEFAULT NULL ,
  PRIMARY KEY (`idBenzinera` ) ,
  INDEX `idHorariBenzinera_FK` (`idHorari` ASC) ,
  INDEX `idZonaBenzinera_FK` (`idZona` ASC) ,
  INDEX `idMarca_FK` (`idMarca` ASC) ,
  CONSTRAINT `idHorariBenzinera_FK`
    FOREIGN KEY (`idHorari` )
    REFERENCES `horari` (`idHorari` )
    ON UPDATE CASCADE,
  CONSTRAINT `idMarca_FK`
    FOREIGN KEY (`idMarca` )
    REFERENCES `marques` (`idMarca` )
    ON UPDATE CASCADE,
  CONSTRAINT `idZonaBenzinera_FK`
    FOREIGN KEY (`idZona` )
    REFERENCES `localitzacio` (`idZona` )
    ON UPDATE CASCADE)
ENGINE = InnoDB
AUTO_INCREMENT = 28715
DEFAULT CHARACTER SET = latin1;

-----
-- Table `tipuscombustible`
-----

DROP TABLE IF EXISTS `tipuscombustible` ;

CREATE TABLE IF NOT EXISTS `tipuscombustible` (
  `idTipus` INT(11) NOT NULL AUTO_INCREMENT ,
  `tipus` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`idTipus` ) ,
  UNIQUE INDEX `tipus_UNIQUE` (`tipus` ASC) )

```

```

ENGINE = InnoDB
AUTO_INCREMENT = 12
DEFAULT CHARACTER SET = latin1;

-----
-- Table `combustible`
-----

DROP TABLE IF EXISTS `combustible` ;

CREATE TABLE IF NOT EXISTS `combustible` (
  `idBenzinera` INT(11) NOT NULL ,
  `idTipusCombustible` INT(11) NOT NULL ,
  `preu` DECIMAL(4,3) NOT NULL ,
  PRIMARY KEY (`idBenzinera`, `idTipusCombustible`),
  INDEX `idTipusCombustible_FK` (`idTipusCombustible` ASC) ,
  INDEX `idBenzinera_FK` (`idBenzinera` ASC) ,
  CONSTRAINT `idBenzinera_FK`
    FOREIGN KEY (`idBenzinera` )
    REFERENCES `benzinera` (`idBenzinera` )
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `idTipusCombustible_FK`
    FOREIGN KEY (`idTipusCombustible` )
    REFERENCES `tipuscombustible` (`idTipus` )
    ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;

-----
-- Table `ordenacio`
-----

DROP TABLE IF EXISTS `ordenacio` ;

CREATE TABLE IF NOT EXISTS `ordenacio` (
  `idOrdenacio` INT(11) NOT NULL AUTO_INCREMENT ,
  `ordenaPerPreu` INT(1) NOT NULL ,
  `ordenaPerLocalitat` INT(1) NOT NULL ,
  `ordenaPerMarca` INT(1) NOT NULL ,
  PRIMARY KEY (`idOrdenacio`),
  UNIQUE INDEX `UNIC` (`ordenaPerPreu` ASC, `ordenaPerLocalitat`
ASC, `ordenaPerMarca` ASC) )
ENGINE = InnoDB
AUTO_INCREMENT = 158
DEFAULT CHARACTER SET = latin1
COMMENT = 'cada columna ha ede contenir un nombre de 1\`1 al 4 que
indic' ;

-----
-- Table `usuari`
-----

DROP TABLE IF EXISTS `usuari` ;

CREATE TABLE IF NOT EXISTS `usuari` (
  `email` VARCHAR(100) NOT NULL ,

```

```

`contrasenya` VARCHAR(16) NOT NULL ,
`nom` VARCHAR(100) NULL DEFAULT NULL ,
`cognoms` VARCHAR(255) NULL DEFAULT NULL ,
`dataNaixement` DATE NULL DEFAULT NULL ,
`administrador` INT(1) NULL DEFAULT NULL ,
`idZona` INT(11) NOT NULL ,
`idTipusCombustible` INT(11) NOT NULL ,
`idOrdenacio` INT(11) NOT NULL ,
PRIMARY KEY (`email`) ,
INDEX `idZonaUsuari_FK` (`idZona` ASC) ,
INDEX `idTipusUsuari_FK` (`idTipusCombustible` ASC) ,
INDEX `idOrdenacio_FK` (`idOrdenacio` ASC) ,
CONSTRAINT `idOrdenacio_FK`
  FOREIGN KEY (`idOrdenacio`)
  REFERENCES `ordenacio` (`idOrdenacio`)
  ON UPDATE CASCADE,
CONSTRAINT `idTipusUsuari_FK`
  FOREIGN KEY (`idTipusCombustible`)
  REFERENCES `tipuscombustible` (`idTipus`)
  ON UPDATE CASCADE,
CONSTRAINT `idZonaUsuari_FK`
  FOREIGN KEY (`idZona`)
  REFERENCES `localitzacio` (`idZona`)
  ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;

-----
-- Table `sessions`
-----

DROP TABLE IF EXISTS `sessions` ;

CREATE TABLE IF NOT EXISTS `sessions` (
  `idSessio` INT(11) NOT NULL AUTO_INCREMENT ,
  `emailUsuari` VARCHAR(100) NOT NULL ,
  `horaInici` TIME NOT NULL ,
  `horaFi` TIME NOT NULL ,
  `data` DATE NOT NULL ,
  PRIMARY KEY (`idSessio`) ,
  INDEX `emailUsuari_FK` (`emailUsuari` ASC) ,
  CONSTRAINT `emailUsuari_FK`
    FOREIGN KEY (`emailUsuari`)
    REFERENCES `usuari` (`email`)
    ON UPDATE CASCADE)
ENGINE = InnoDB
AUTO_INCREMENT = 9
DEFAULT CHARACTER SET = latin1;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```