

Disseny i implementació d'una aplicació web per a la reserva on-line de localitats d'espectacles

Autor: Anna Maria Corominas Espinal
Director: Josep Maria Ribó Balust

Universitat de Lleida
Escola Politècnica Superior
Enginyeria Tècnica en Informàtica de Sistemes

Setembre 2007

Vull donar les gràcies a tots els que d'alguna manera han fet possible que arribés fins aquí: als meus pares, als/les de Solsona, als/les de Lleida, i especialment als/les que han llegit el projecte i m'han donat l'opinió, a l'equip del SBD de Capont, a l'equip del CGI de l'EPS, als companys d'Endesa i als de l'aula de projectes.

Resum

Aquest treball presenta un Terminal de Reserva d'Espectacles. Es tracta d'una aplicació web que permet als seus usuaris registrats reservar localitats de diferents tipus d'espectacles a través d'Internet i amb total llibertat d'horaris. L'aplicació s'ha dissenyat amb tecnologia Java (servlets, JSP, beans...) i s'ha llicenciat com a aplicació de programari lliure amb la llicència GPL.

Índex

1	Introducció	5
1.1	Objectius	5
1.2	Estructura d'aquest document	6
2	Anàlisi de requeriments	7
2.1	Requeriments generals	7
2.1.1	Espectacles	8
2.1.2	Representacions	8
2.1.3	Sales	9
2.1.4	Reserves	9
2.2	Actors	10
2.3	Casos d'ús	12
2.3.1	Autenticar	12
2.3.2	Donar d'alta a un usuari	13
2.3.3	Modificar les dades d'un usuari	14
2.3.4	Eliminar un usuari del sistema	15
2.3.5	Llistar els usuaris	16
2.3.6	Afegir una sala	17
2.3.7	Afegir un espectacle	18
2.3.8	Afegir una representació	19
2.3.9	Modificar una sala	20
2.3.10	Modificar un espectacle	21
2.3.11	Modificar una representació	22
2.3.12	Eliminar una sala	23
2.3.13	Eliminar un espectacle	24
2.3.14	Eliminar una representació	25
2.3.15	Consultar una reserva	26
2.3.16	Modificar una reserva	27
2.3.17	Eliminar una reserva	28
2.3.18	Registrar	29
2.3.19	Modificar les dades personals	30
2.3.20	Consultar les reserves d'un client.	31
2.3.21	Reservar	32

2.4	Model de domini	35
3	Aspectes de disseny	37
3.1	Classes de disseny	38
3.1.1	Classes DTO (Data Transfer Object)	38
3.1.2	Classes DAO (Data Access Object)	48
3.2	Servlets	54
3.3	Pàgines JSP (JavaServer Pages)	55
3.4	Excepcions (errors)	58
3.5	Disseny de la base de dades	59
3.5.1	Model relacional	59
3.5.2	Anàlisi de les entitats i atributs	60
3.6	Llicència	65
4	Conclusions i treball futur	66
A	Taules SQL	68
B	Llicència GPL	71

Capítol 1

Introducció

En la societat actual, Internet ha esdevingut una eina fonamental per al treball i l'oci, i les aplicacions web s'han encarregat en gran part de promoure aquesta evolució, ja que cada vegada ofereixen més funcionalitats i es fan més imprescindibles a l'hora de realitzar qualsevol cosa.

D'altra banda, en aquesta societat cada vegada més estressada, l'individu necessita l'oci com un element fonamental dintre de la seva vida.

Aquestes dues idees ens porten a pensar en realitzar un projecte que ho posi tot en comú: aplicacions web i oci. Així sorgeix la idea de crear un Terminal de Reserva d'Espectacles (TRE) que aprofiti tots els avantatges de cada idea.

El TRE ha d'agrupar molts espectacles, de manera que l'usuari no s'hagi de preocupar de visitar diverses pàgines web per veure la cartellera de diferents tipus d'espectacles i, al mateix temps, al ser una aplicació web, li permetrà comprar les entrades des de casa seva, sense horaris i amb total llibertat: 24/7, les 24 hores del dia, els 7 dies de la setmana.

1.1 Objectius

L'objectiu fonamental d'aquest projecte és realitzar una aplicació web que permeti reservar tot tipus d'espectacles amb senzillesa, claredat i facilitat.

Ha de ser una eina potent, però alhora fàcil, per a que a l'usuari li resulti una aplicació amena i prefereixi efectuar les seves reserves per aquesta via que anar directament a taquilla.

Per realitzar aquesta aplicació, les tecnologies que s'utilitzaran són les següents: Java [LD06] [Eck02] [Man07] [Jav03], JSP [Han02] [LD06], Servlets [Han02], HTML i CSS [FdI07].

1.2 Estructura d'aquest document

Aquest document està estructurat en capítols. Tot seguit s'explica breument en què consisteix cadascun d'ells:

Capítol 2. Anàlisi dels requeriments En aquest capítol s'analitzen les funcionalitats que es volen aconseguir en aquesta aplicació web, de quins mitjans es disposa i com seran utilitzats per assolir els objectius desitjats.

És a dir, s'estudiaran els requeriments, s'analitzaran els casos d'ús i es presentarà un model conceptual.

Capítol 3. Aspectes de disseny En aquesta part es descriuen les tecnologies utilitzades i s'explica com s'utilitzen per implementar l'aplicació.

Capítol 4. Conclusions i treball futur Aquest capítol està dedicat a valorar el treball realitzat i determinar si s'han complert els objectius; i a presentar les conclusions obtingudes.

També es proposen algunes de les millores que es podrien dur a terme a l'aplicació.

Capítol 2

Anàlisi de requeriments

2.1 Requeriments generals

L'objectiu principal és realitzar una aplicació web per a un TRE (Terminal de Reserva d'Espectacles). Una eina que permetrà reservar espectacles via web.

La idea bàsica és que un usuari pugui accedir a l'aplicació i reservar espectacles. Aquest usuari serà un client de l'aplicació.

Per poder dur a terme això, dins la base de dades han d'existir-hi espectacles i, per aquesta raó, prèviament algú els ha hagut de donar d'alta. En aquest moment apareix la figura de l'administrador web. Ell s'encarregarà de realitzar totes les tasques de manteniment sobre els espectacles, les sales, les representacions i les reserves.

Degut a que el sistema permet (mitjançant un entorn fàcil, amb formularis) que l'administrador web no hagi de tenir gaires coneixements d'informàtica, sorgeix la necessitat de que hi hagi algú per sobre d'aquest, un administrador de l'aplicació, que s'encarregui de gestionar els usuaris i de solucionar els possibles problemes que puguin sorgir.

Ens trobem davant d'una aplicació que necessita:

- Gestionar usuaris
- Gestionar espectacles
- Gestionar representacions
- Gestionar sales
- Efectuar i gestionar reserves

I per dur-ho a terme, disposem de tres tipus d'usuaris:

- Usuari o client
- Administrador web
- Administrador

Cada usuari tindrà les seves funcions dins l'aplicació i no podrà realitzar funcions que no li siguin permeses.

Una vegada arribat a aquest punt ens adonem que, per realitzar tot el que hem comentat, necessitem emmagatzemar informació sobre espectacles, representacions, sales i reserves, que es detalla a l'apartat 2.1.1 i següents.

I pel que fa a la informació referent als usuaris, vegeu l'apartat 2.2.

2.1.1 Espectacles

Els espectacles són la idea principal d'aquesta aplicació web: un usuari es connectarà per reservar espectacles, així que s'ha d'oferir tota la informació que un client pugui desitjar. És imprescindible mostrar el títol de l'espectacle i també cal tenir una secció de comentaris, en la que s'hi inclogui qualsevol informació d'interès, o també s'hi pot inserir la sinopsi d'una pel·lícula o d'una obra de teatre, per exemple. També hi ha d'haver un lloc on posar l'adreça de la pàgina web de l'espectacle (si en disposa) per si l'usuari vol més informació de la que li proposem.

També cal organitzar tots els espectacles en tipus. Bàsicament seran teatre, cinema, esports i música, però se'n poden afegir de nous si l'administrador web ho considera oportú. La raó d'aquesta organització és la següent: un usuari pot tenir clar quin espectacle vol anar a veure, però també pot ser que un usuari tingui clar que vol anar al cinema i vulgui veure quines pel·lícules hi ha en cartellera.

2.1.2 Representacions

Les representacions fan referència a cada passi que s'emet d'una pel·lícula. cada representació que té una obra de teatre o cada concert o actuació que fa un cantant, banda o grup musical. Llavors es lògic que una representació contingui la data i la hora.

També s'ha considerat oportú incloure el preu dins les representacions, ja que, per exemple, una sessió de cinema té un preu diferent si correspon al dia de l'espectador o a una sessió golfa.

2.1.3 Sales

Les sales són els espais on tenen lloc els espectacles. Cal tenir en compte el nom de la sala, l'adreça, la ciutat i la província, així com el telèfon i l'adreça de la pàgina web (si en disposa), ja que l'usuari pot necessitar aquesta informació per accedir a la sala o per fer qualsevol consulta.

Anàlogament, es considera que una sala pot estar dividida en zones, les zones en files i finalment les files en butaques. D'aquesta manera tenim consciència de com està organitzada la sala, de quantes butaques o seients hi han per filera i quantes files hi han per zona, i així poder saber l'aforament general de la sala i l'aforament per seccions.

2.1.4 Reserves

Les reserves són l'objectiu del final de l'aplicació. Cal tenir informació del client que efectua la reserva, de la representació que ha reservat (així com de l'espectacle) i de la sala on es realitza, sabent en tot moment la zona, la fila i la butaca que estan associades a la reserva.

També cal tenir coneixement de si l'entrada ha estat recollida o no. La forma de recollir entrades consistiria en agafar el codi de la reserva introduir-lo en un caixer automàtic i aquest dispensaria l'entrada. De moment això no està implementat.

2.2 Actors

Tots els usuaris que interactuen en qualsevol aplicació s'anomenen actors. Es defineixen diferents tipus d'actors segons el rol que desenvolupen dins el sistema. El rol d'un actor marcarà les funcions que podrà realitzar dins l'aplicació. D'aquesta manera, cada usuari tindrà el seu entorn d'actuació, podent realitzar les operacions que li permeti el seu rol.

En aquest cas, hem definit tres actors:

- **Usuari**

És el client de l'aplicació, qui realitza les reserves. El podem veure com l'usuari final, el que realment gaudirà de l'aplicació. Es pot donar d'alta a ell mateix i es pot gestionar les seves dades personals.

- **Administrador web**

És qui s'encarrega de mantenir l'aplicació web. Té competència per realitzar qualsevol operació sobre les sales, els espectacles, les representacions i les reserves.

- **Administrador**

És qui té els privilegis suficients per realitzar operacions sobre els altres usuaris. El primer administrador és creat directament sobre la base de dades, a partir d'aquí, pot donar d'alta i de baixa a altres administradors, administradors web i, si és necessari, també als usuaris (clients).

A continuació s'enumeren les funcions que estan associades a cada actor segons el seu rol d'activitat.

Client

- Donar-se d'alta (registrar-se)
- Modificar les seves dades personals
- Consultar les seves reserves
- Reservar espectacles

Administrador Web

- Afegir sales
- Afegir espectacles

- Afegir representacions
- Modificar informació de les sales
- Modificar informació dels espectacles
- Modificar informació de les representacions
- Eliminar sales
- Eliminar espectacles
- Eliminar representacions
- Consultar reserves
- Modificar reserves
- Eliminar reserves

Administrador de l'aplicació

- Donar d'alta usuaris
- Modificar les dades personals dels usuaris
- Eliminar usuaris
- Llistar els usuaris de l'aplicació

2.3 Casos d'ús

En aquest apartat es detallen totes les funcions que pertanyen al sistema, com a interacció entre el sistema i els actors.

2.3.1 Autenticar

Actor iniciador: Usuari.

Precondició: -

Postcondició: S'ha autenticat un usuari en l'aplicació.

USUARI	SISTEMA
1. Introdueix el nom d'usuari i la clau de pas al formulari d'autenticació.	2. Comprova si l'usuari existeix. 3. Comprova si la clau de pas entrada per l'usuari és la mateixa que la que hi ha associada a l'usuari dins la base de dades. 4. Autentica l'usuari com a usuari del sistema. 5. Mostra la pàgina d'inici corresponent a l'usuari dins del seu rol.

Excepcions:

1. No s'ha omplert tota la informació necessària. El sistema mostra els missatges d'error corresponents.
2. L'usuari no pertany al sistema. El sistema mostra els missatges d'error corresponents.
3. La clau de pas no correspon a l'usuari. El sistema mostra els missatges d'error corresponents.

2.3.2 Donar d'alta a un usuari

Actor: Administrador.

Precondició: L'usuari s'ha autenticat com a administrador.

Postcondició: S'ha afegit un usuari al sistema.

USUARI	SISTEMA
1. Indica que vol donar d'alta un usuari. 3. Introdueix totes les dades de l'usuari al qual es vol donar d'alta al sistema.	2. Mostra un formulari amb els camps requerits. 4. Es dona d'alta un usuari dins el sistema. 5. Mostra la pàgina d'inici de l'administrador.

Excepcions:

3. No s'ha omplert tota la informació necessària. El sistema mostra els missatges d'error corresponents.

3. El tipus de dades que s'ha introduït no concorda amb el tipus de dades esperat. El sistema mostra els missatges d'error corresponents.

4. L'usuari que es vol donar d'alta ja existeix. El sistema mostra els missatges d'error corresponents.

2.3.3 Modificar les dades d'un usuari

Actor: Administrador.

Precondició: L'usuari s'ha autenticat com a administrador.

Postcondició: S'han modificat diverses dades d'un usuari.

USUARI	SISTEMA
1. Indica que vol modificar les dades d'un usuari.	
3. Selecciona un usuari.	2. Mostra un llistat de tots els usuaris de l'aplicació.
5. Modifica les dades de l'usuari.	4. Mostra les dades de l'usuari en mode editable.
	6. Emmagatzema els canvis realitzats.
	7. Mostra la pàgina principal de l'administrador.

Excepcions:

5. No s'ha omplert tota la informació necessària. El sistema mostra els missatges d'error corresponents.

5. El tipus de dades que s'ha introduït no concorda amb el tipus de dades esperat. El sistema mostra els missatges d'error corresponents.

2.3.4 Eliminar un usuari del sistema

Actor: Administrador.

Precondició: L'usuari s'ha autenticat com a administrador.

Postcondició: S'ha eliminat un usuari del sistema.

USUARI	SISTEMA
1. Indica que vol esborrar un usuari. 3. Selecciona un usuari. 5. Confirma que vol realitzar l'eliminació.	2. Mostra tots els usuaris de l'aplicació. 4. Mostra un missatge comunicant que espera la confirmació per realitzar l'operació. 6. Esborra l'usuari, així com les seves relacions amb tots els camps de la base de dades. 7. Mostra la pàgina principal de l'administrador.

Excepcions:

- No hi ha cap altre usuari a l'aplicació a part del mateix administrador. El sistema mostra els missatges d'error corresponents.

2.3.5 Llistar els usuaris

Actor: Administrador.

Precondició: L'usuari s'ha autenticat com a administrador.

Postcondició: L'usuari ha consultat un llistat dels usuaris del sistema.

USUARI	SISTEMA
1. Indica que vol consultar el llistat d'usuaris del sistema.	
	2. Mostra una llista dels rols que pot adquirir un usuari dins el sistema. Només es mostren els que tenen, almenys, un usuari associat.
3. Selecciona el rol al qual pertanyen els usuaris que vol llistar.	
	4. Mostra per pantalla el llistat d'usuaris.

Excepcions: -

2.3.6 Afegir una sala

Actor: Administrador web.

Precondició: L'usuari s'ha autenticat com a administrador web.

Postcondició: S'ha afegit una sala al sistema.

USUARI	SISTEMA
1. Indica que vol afegir una sala al sistema.	
3. Introdueix totes les dades de la sala que es vol afegir al sistema.	2. Mostra un formulari amb els camps requerits.
	4. Afegeix la sala dins el sistema.
	5. Mostra la pàgina principal de l'administrador web.

Excepcions:

3. No s'ha oplet tota la informació necessària. El sistema mostra els missatges d'error corresponents.

3. El tipus de dades que s'ha introduït no concorda amb el tipus de dades esperat. El sistema mostra els missatges d'error corresponents.

4. La sala que es vol donar d'alta ja existeix. El sistema mostra els missatges d'error corresponents.

2.3.7 Afegir un espectacle

Actor: Administrador web.

Precondició: L'usuari s'ha autenticat com a administrador web.

Postcondició: S'ha afegit un espectacle al sistema.

USUARI	SISTEMA
1. Indica que vol afegir un espectacle al sistema. 3. Introdueix totes les dades de l'espectacle.	2. Mostra un formulari amb els camps requerits. 4. Afegeix l'espectacle dins el sistema. 5. Mostra la pàgina principal de l'administrador web.

Excepcions:

3. No s'ha omplert tota la informació necessària. El sistema mostra els missatges d'error corresponents.
3. El tipus de dades que s'ha introduït no concorda amb el tipus de dades esperat. El sistema mostra els missatges d'error corresponents.
4. L'espectacle que es vol donar d'alta ja existeix. El sistema mostra els missatges d'error corresponents.

2.3.8 Afegir una representació

Actor: Administrador web.

Precondició: L'usuari s'ha autenticat com a administrador web.

Postcondició: S'ha afegit una representació al sistema.

USUARI	SISTEMA
1. Indica que vol afegir una representació al sistema. 3. Introdueix totes les dades de la representació.	2. Mostra un formulari amb els camps requerits. 4. Afegeix la representació dins els sistema. 5. Mostra la pàgina principal de l'administrador web.

Excepcions:

3. No s'ha omplert tota la informació necessària. El sistema mostra els missatges d'error corresponents.
3. El tipus de dades que s'han introduït no concorda amb el tipus de dades esperats. El sistema mostra els missatges d'error corresponents.
4. La representació que es vol donar d'alta ja existeix. El sistema mostra els missatges d'error corresponents.

2.3.9 Modificar una sala

Actor: Administrador web.

Precondició: L'usuari s'ha autenticat com a administrador web.

Postcondició: S'han modificat una o diverses dades d'una sala.

USUARI	SISTEMA
1. Indica que vol modificar les dades d'una sala.	
3. Selecciona una sala.	2. Mostra totes les sales que pertanyen a l'aplicació.
5. Modifica les dades de la sala.	4. Mostra les dades de la sala en mode editable.
	6. Emmagatzema els canvis realitzats.
	7. Mostra un llistat de tots els usuaris que tenien una reserva en la sala modificada.

Excepcions:

2. L'aplicació no té cap sala associada. El sistema mostra els missatges d'error corresponents.

5. No s'ha omplert tota la informació necessària. El sistema mostra els missatges d'error corresponents.

5. El tipus de dades que s'ha introduït no concorda amb el tipus de dades esperat. El sistema mostra els missatges d'error corresponents.

2.3.10 Modificar un espectacle

Actor: Administrador web.

Precondició: L'usuari s'ha autenticat com a administrador web.

Postcondició: S'han modificat una o diverses dades d'un espectacle.

USUARI	SISTEMA
1. Indica que vol modificar les dades d'un espectacle. 3. Selecciona l'espectacle que vol modificar. 5. Modifica les dades de l'espectacle.	2. Mostra tots els espectacles que pertanyen a l'aplicació. 4. Mostra les dades de l'espectacle en mode editable. 6. Emmagatzema els canvis relitzats. 7. Mostra un llistat de tots els usuaris que tenien alguna reserva en l'espectacle modificat.

Excepcions:

2. L'aplicació no té cap espectacle associat. El sistema mostra els missatges d'error corresponents.

5. No s'ha omplert tota la informació necessària. El sistema mostra els missatges d'error corresponents.

5. El tipus de dades que s'ha introduït no concorda amb el tipus de dades esperat. El sistema mostra els missatges d'error corresponents.

2.3.11 Modificar una representació

Actor: Administrador web.

Precondició: L'usuari s'ha autenticat com a administrador web.

Postcondició: S'han modificat una o diverses dades d'una representació.

USUARI	SISTEMA
1. Indica que vol modificar les dades d'una representació. 3. Selecciona una representació. 5. Modifica les dades de la representació.	2. Mostra totes les representacions que conté l'aplicació. 4. Mostra les dades de la representació en mode editable. 6. Emmagatzema els canvis realitzats. 7. Mostra un llistat de tots els usuaris que tenien alguna reserva en la representació modificada.

Excepcions:

2. L'aplicació no té cap representació associada. El sistema mostra els missatges d'error corresponents.

5. No s'ha omplert tota la informació necessària. El sistema mostra els missatges d'error corresponents.

5. El tipus de dades introduïdes no concorden amb el tipus de dades esperades. El sistema mostra els missatges d'error corresponents.

2.3.12 Eliminar una sala

Actor: Administrador web.

Precondició: L'usuari s'ha autenticat com a administrador web.

Postcondició: S'ha eliminat una sala del sistema.

USUARI	SISTEMA
1. Indica que vol eliminar una sala del sistema. 3. Selecciona una sala. 5. Confirma que vol realitzar l'eliminació.	2. Mostra totes les sales que pertanyen a l'aplicació. 4. Mostra un missatge comunicant que espera la confirmació per realitzar l'operació. 6. Elimina la sala del sistema. 7. Mostra un llistat de tots els usuaris que tenien alguna reserva en la sala eliminada.

Excepcions:

2. L'aplicació no té cap sala associada. El sistema mostra els missatges d'error corresponents.

2.3.13 Eliminar un espectacle

Actor: Administrador web.

Precondició: L'usuari s'ha autenticat com a administrador web.

Postcondició: S'ha eliminat un espectacle del sistema.

USUARI	SISTEMA
1. Indica que vol eliminar un espectacle. 3. Selecciona l'espectacle que vol eliminar. 5. Confirma que vol realitzar l'eliminació.	2. Mostra tots els espectacles que conté l'aplicació. 4. Mostra un missatge comunicant que espera la confirmació per realitzar l'operació. 6. Elimina l'espectacle del sistema. 7. Mostra un llistat de tots els usuaris que tenien alguna reserva en l'espectacle eliminat.

Excepcions:

2. L'aplicació no té cap espectacle associat. El sistema mostra els missatges d'error corresponents.

2.3.14 Eliminar una representació

Actor: Administrador web.

Precondició: L'usuari s'ha autenticat com a administrador web.

Postcondició: S'ha eliminat una representació.

USUARI	SISTEMA
1. Indica que vol eliminar una representació del sistema. 3. Selecciona una representació. 5. Confirma que vol realitzar l'eliminació.	2. Mostra totes les representacions que conté l'aplicació. 4. Mostra un missatge comunicant que espera la confirmació per realitzar l'operació. 6. Elimina la representació del sistema. 7. Mostra un llistat de tots els usuaris que tenien alguna reserva en la representació eliminada.

Excepcions:

2. L'aplicació no té cap representació associada. El sistema mostra els missatges d'error corresponents.

2.3.15 Consultar una reserva

Actor: Administrador web.

Precondició: L'usuari s'ha autenticat com a administrador web.

Postcondició: S'ha mostrat per pantalla tota la informació relacionada amb una reserva.

USUARI	SISTEMA
1. Indica que vol consultar una reserva. 3. Selecciona el tipus de consulta i introdueix el paràmetre de cerca. 5. Selecciona una reserva. 7. Modifica les dades de la reserva.	2. Mostra un formulari amb els tipus de consulta que es poden realitzar (data, sala, espectacle, usuari, identificador de reserva) i el paràmetre de cerca. 4. Mostra un llistat amb totes les reserves que compleixen la condició de cerca. 6. Mostra per pantalla totes les dades de la reserva en mode editable. 8. Emmagatzema els canvis realitzats. 9. Mostra les dades de l'usuari que té associada la reserva.

Excepcions:

2. L'aplicació no conté cap reserva. El sistema mostra els missatges d'error corresponents.
3. El tipus de dades que s'ha introduït no concorda amb el tipus de dades esperat. El sistema mostra els missatges d'error corresponents.
4. No existeix cap reserva que compleixi la condició de cerca. El sistema mostra els missatges d'error corresponents.
7. No s'ha omplert tota la informació necessària. El sistema mostra els missatges d'error corresponents.
7. El tipus de dades que s'ha introduït no concorda amb el tipus de dades esperat. El sistema mostra els missatges d'error corresponents.

2.3.16 Modificar una reserva

Actor: Administrador web.

Precondició: L'usuari s'ha autenticar com a administrador web i ha de conèixer l'identificador de la reserva.

Postcondició: S'han modificat una o diverses dades d'una reserva.

USUARI	SISTEMA
1. Indica que vol modificar una reserva. 3. Introdueix l'identificador de reserva. 6. Modifica les dades de la reserva.	2. Mostra el formulari per inserir l'identificador de reserva. 4. Comprova si la reserva existeix 5. Mostra les dades de la reserva en mode editable. 7. Emmagatzema els canvis realitzats. 8. Mostra les dades de l'usuari que té associada la reserva.

Excepcions:

3. No s'ha omplert tota la informació necessària. El sistema mostra els missatges d'error corresponents.

3. El tipus de dades que s'ha introduït no concorda amb el tipus de dades esperat. El sistema mostra els missatges d'error corresponents.

4. La reserva no existeix dins el sistema. El sistema mostra els missatges d'error corresponents.

6. No s'ha omplert tota la informació necessària. El sistema mostra els missatges d'error corresponents.

6. El tipus de dades que s'ha introduït no concorda amb el tipus de dades esperat. El sistema mostra els missatges d'error corresponents.

2.3.17 Eliminar una reserva

Actor: Administrador web.

Precondició: L'usuari s'ha autenticat com a administrador web i ha de conèixer l'identificador de la reserva.

Postcondició: S'ha eliminat una reserva.

USUARI	SISTEMA
1. Indica que vol eliminar una reserva del sistema. 3. Introdueix l'identificador de reserva. 6. Confirma que vol realitzar l'eliminació.	2. Mostra el formulari per inserir l'identificador de reserva. 4. Comprova si la reserva existeix. 5. Mostra un missatge esperant la confirmació per realitzar l'operació. 7. Elimina la reserva del sistema. 8. Mostra per pantalla les dades de l'usuari que té associada la reserva eliminada

Excepcions:

3. No s'ha omplert tota la informació necessària. El sistema mostra els missatges d'error corresponents.

3. El tipus de dades que s'ha introduït no concorda amb el tipus de dades esperat. El sistema mostra els missatges d'error corresponents.

4. La reserva no existeix dins el sistema. El sistema mostra els missatges d'error corresponents.

2.3.18 Registrar**Actor:** Client.**Precondició:** -**Postcondició:** S'ha afegit un usuari client al sistema.

USUARI	SISTEMA
1. Indica que vol registrar-se dins el sistema. 3. Introdueix totes les dades requerides.	2. Mostra un formulari per donar-se d'alta. 4. Es dona d'alta l'usuari dins el sistema. 5. Mostra la pàgina principal del client.

Excepcions:

3. No s'ha omplert tota la informació necessària. El sistema mostra els missatges d'error corresponents.
3. El tipus de dades que s'ha introduït no concorda amb el tipus de dades esperat. El sistema mostra els missatges d'error corresponents.
4. L'usuari que es vol donar d'alta ja existeix. El sistema mostra els missatges d'error corresponents.

2.3.19 Modificar les dades personals

Actor: Client.

Precondició: L'usuari s'ha autenticat com a client.

Postcondició: S'han modificat una o diverses dades personals de l'usuari.

USUARI	SISTEMA
1. Indica que vol modificar les dades personals. 3. Modifica les dades. 4. Confirma la modificació.	2. Mostra les dades de l'usuari en mode editable. 5. Emmagatzema els canvis realitzats. 6. Mostra la pàgina principal del client.

Excepcions:

- 4. No s'ha omplert tota la informació necessària. El sistema mostra els missatges d'error corresponents.
- 4. El tipus de dades que s'ha introduït no concorda amb el tipus de dades esperat. El sistema mostra els missatges d'error corresponents.

2.3.20 Consultar les reserves d'un client.

Actor: Client.

Precondició: L'usuari s'ha autenticat com a client.

Postcondició: Es mostren per pantalla les reserves efectuades per l'usuari a partir de la data actual i en ordre descendent.

USUARI	SISTEMA
1. Indica que vol consultar les seves reserves.	2. Mostra per pantalla un llistat de les reserves i el seu estat (recollides o no).

Excepcions:

2. No hi ha cap reserva pendent. El sistema mostra els missatges d'error corresponents.

2.3.21 Reservar

Actor: Client.

Precondició: L'usuari s'ha autenticat com a client.

Postcondició: L'usuari ha efectuat una reserva.

USUARI	SISTEMA
1. Indica que vol efectuar una reserva.	
3. Selecciona un tipus d'espectacle.	2. Mostra un llistat amb els tipus d'espectacles que hi ha al sistema.
5. Selecciona un espectacle.	4. Mostra un llistat dels espectacles del tipus seleccionat.
7. Selecciona una representació.	6. Mostra un llistat de les representacions que es realitzen a partir de la data actual.
9. Insereix el nombre d'entrades.	8. Mostra el formulari per introduir el nombre d'entrades.
12. Selecciona una zona.	10. Comprova si hi ha suficients localitats lliures per a la representació.
14. Selecciona una fila.	11. Mostra un llistat amb les zones disponibles.
16. Selecciona els seients.	13. Mostra un llistat amb les files disponibles.
18. Confirma la reserva.	15. Mostra un llistat amb les butaques disponibles.
20. Confirma la reserva.	17. Mostra el preu i demana la confirmació per efectuar la reserva.
22. Finalitza la reserva.	19. Mostra un resum de la reserva i torna a demanar confirmació.
	21. S'insereix la reserva al sistema i es llisten els identificadors.
	23. Es mostra la pàgina principal de l'usuari.

Excepcions:

2. L'aplicació no conté cap espectacle, aleshores, no es pot mostrar cap tipus. El sistema mostra els missatges d'error corresponents.
9. L'usuari no ha introduït almenys una entrada. El sistema mostra els missatges d'error corresponents.
9. El tipus de dades que s'ha introduït no concorda amb el tipus de dades esperat. El sistema mostra els missatges d'error corresponents.
10. No hi ha suficients localitats per a la representació seleccionada. El sistema mostra els missatges d'error corresponents.
11. La representació no disposa de suficients localitats disponibles en una mateixa zona. El sistema mostra els missatges d'error corresponents.
13. La representació no disposa de suficients localitats disponibles en una mateixa fila. El sistema mostra els missatges d'error corresponents.
16. No s'han seleccionat el mateix nombre de seients que el nombre d'entrades que s'han introduït al formulari de les entrades. El sistema mostra els missatges d'error corresponents.

2.4 Model de domini

A continuació es mostra el model de domini (figura 2.1).

El model de domini dóna una visió general i sintetitzada de com està estructurada l'aplicació. Aquest model es dissenya abstractant les idees bàsiques de tot el que s'ha anat concluint al llarg de l'anàlisi de requeriments i dels casos d'ús.

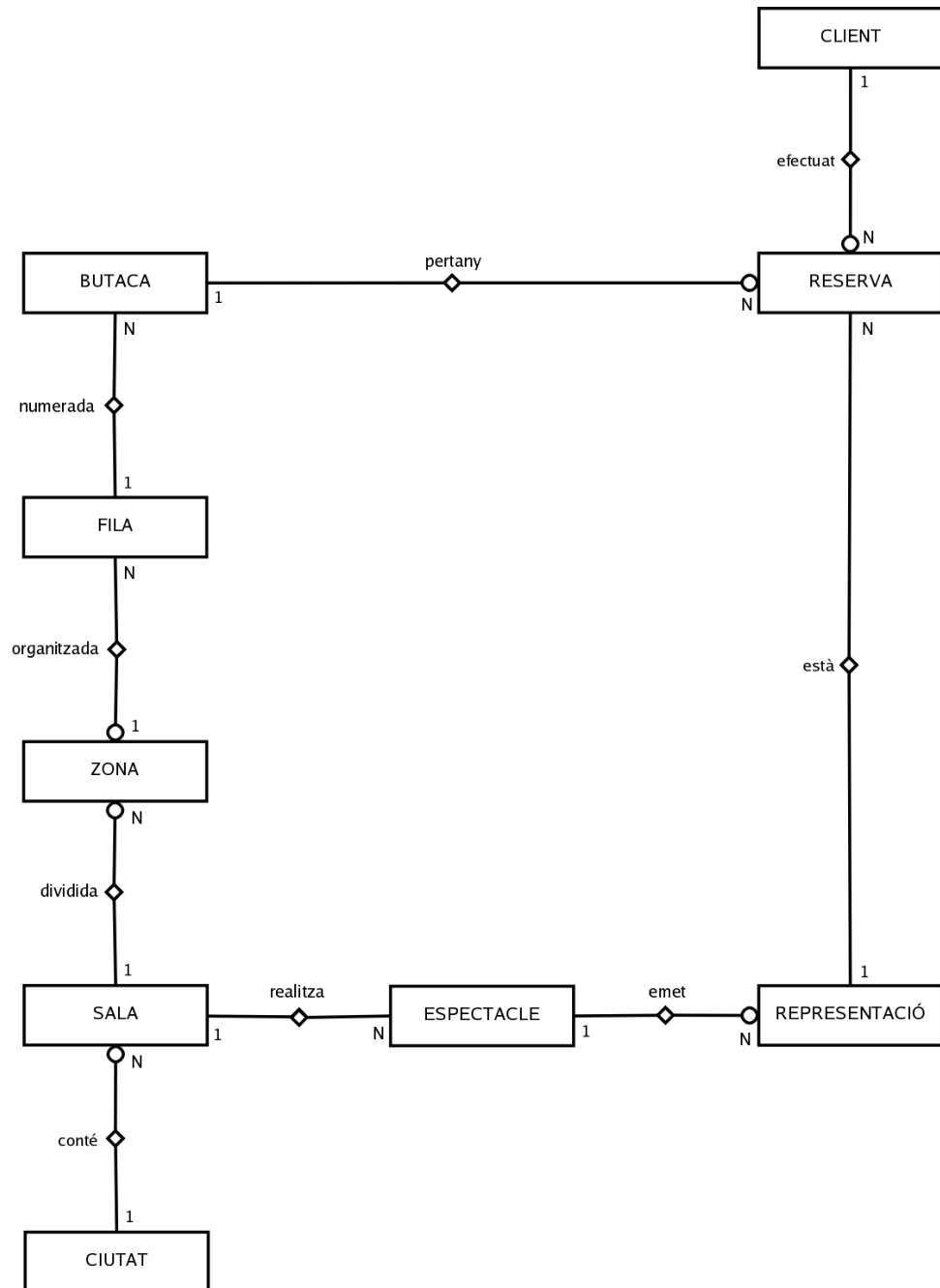


Figura 2.1: Model de domini

Capítol 3

Aspectes de disseny

A l'hora de dissenyar l'aplicació ens trobem en la necessitat de crear un sistema persistent, capaç de suportar canvis fàcilment i executable des de qualsevol lloc. La millor opció és a través d'Internet mitjançant un navegador.

Per realitzar-la, hem escollit Java, ja que les tecnologies lligades a aquest llenguatge de programació ofereixen un ampli ventall d'eines (Servlets, JSP, Struts, ...) molt utilitzades per la comunitat de l'Enginyeria del Software i que ens permetran assolir els objectius proposats. [LD06] [Eck02] [Jav03] [Man07]

3.1 Classes de disseny

3.1.1 Classes DTO (Data Transfer Object)

La funció principal d'aquestes classes és la d'encapsular els atributs que contenen la resposta d'una petició de la capa vista i servir de comunicació entre les diferents capes de l'aplicació.

A continuació prosseguirem amb l'especificació d'aquestes classes:

ButacaDTO

```
public class ButacaDTO implements Serializable
```

- `public String getIdButaca()`
Retorna un `String` amb l'identificador de butaca associat a l'objecte `ButacaDTO`.
- `public void setIdButaca(String IdButaca)`
Estableix l'identificador de butaca `IdButaca` a l'objecte `ButacaDTO`.
 - `IdButaca`: Identificador de la butaca que es vol inserir a l'objecte `ButacaDTO`.
- `public String getNumButaca()`
Retorna un `String` amb el número de butaca associat a l'objecte `ButacaDTO`.
- `public void setNumButaca(String NumButaca)`
Estableix el número de butaca `NumButaca` a l'objecte `ButacaDTO`.
 - `NumButaca`: Número de la butaca que es vol inserir a l'objecte `ButacaDTO`.

ClientDTO

```
public class ClientDTO implements Serializable
```

- `public int getIdClient()`
Retorna un `int` amb l'identificador del client associat a l'objecte `ClientDTO`.
- `public void setIdClient(int IdClient)`
Estableix l'identificador del client `IdClient` a l'objecte `ClientDTO`.
 - `IdClient`: Identificador del client que es vol inserir a l'objecte `ClientDTO`.

- `public String getNif()`
Retorna un `String` amb el NIF del client associat a l'objecte `ClientDTO`.
- `public void setDni(String dni, String lletra)`
Estableix el NIF del client, mitjançant el `dni` i la `lletra`, a l'objecte `ClientDTO`.
 - `dni`: Número del DNI del client que es vol inserir a l'objecte `ClientDTO`.
 - `lletra`: Lletre del DNI del client que es vol inserir a l'objecte `ClientDTO`.
- `public void setNif(String nif)`
Estableix el NIF del client, mitjançant el `nif`, a l'objecte `ClientDTO`.
 - `nif`: Codi del NIF del client que es vol inserir a l'objecte `ClientDTO`.
- `public String getClau()`
Retorna un `String` amb la clau de pas (password) del client associat a l'objecte `ClientDTO`.
- `public void setClau(String clau)`
Estableix la clau de pas `clau` del client a l'objecte `ClientDTO`.
 - `clau`: Cadena alfanumèrica corresponent a la clau de pass que es vol inserir a l'objecte `ClientDTO`.
- `public String getNom()`
Retorna un `String` amb el nom del client associat a l'objecte `ClientDTO`.
- `public String getCognom1()`
Retorna un `String` amb el primer cognom del client associat a l'objecte `ClientDTO`.
- `public String getCognom2()`
Retorna un `String` amb el segon cognom del client associat a l'objecte `ClientDTO`.
- `public void setNomComplet(String nom, String cognom1, String cognom2)`
Estableix el nom complet del client, mitjançant el `nom`, el `cognom1` i el `cognom2`, a l'objecte `ClientDTO`.
 - `nom`: Nom del client que es vol inserir a l'objecte `ClientDTO`.
 - `cognom1`: Primer cognom del client que es vol inserir a l'objecte `ClientDTO`.

- `cognom2`: Segon cognom del client que es vol inserir a l'objecte `ClientDTO`.
- `public String getEmail()`
Retorna un `String` amb l'adreça de correu electrònic del client associat a l'objecte `ClientDTO`.
- `public void setEmail(String email)`
Estableix l'adreça de correu electrònic `email` del client a l'objecte `ClientDTO`.
 - `email`: Adreça de correu electrònic del client que es vol inserir a l'objecte `ClientDTO`.
- `public String getNVisa()`
Retorna un `String` amb el número de la tarja de crèdit del client associat a l'objecte `ClientDTO`.
- `public void setNVisa(String nvisa)`
Estableix el número de la tarja de crèdit `nvisa` del client a l'objecte `ClientDTO`.
 - `nvisa`: Número de la tarja de crèdit del client que es vol inserir a l'objecte `ClientDTO`.
- `public String getTipus()`
Retorna un `String` amb el tipus (si és usuari, administrador web o administrador) del client associat a l'objecte `ClientDTO`.
- `public void setTipus(String tipus)`
Estableix el tipus `tipus` (si és usuari, administrador web o administrador) del client a l'objecte `ClientDTO`.
 - `tipus`: Tipus d'usuari que interactua amb l'aplicació del client que es vol inserir a l'objecte `ClientDTO`.
 - * 1=Administrador
 - * 2=Administrador web
 - * 3=Usuari

EspectacleDTO

```
public class EspectacleDTO implements Serializable
```

- `public String getIdEspectacle()`
Retorna un `String` amb l'identificador de l'espectacle associat a l'objecte `EspectacleDTO`.

- `public void setIdEspectacle(String IdEspectacle)`
Estableix l'identificador de l'espectacle `IdEspectacle` a l'objecte `EspectacleDTO`.
 - `IdEspectacle`: Identificador de l'espectacle que es vol inserir a l'objecte `EspectacleDTO`.
- `public String getTitol()`
Retorna un `String` amb el títol de l'espectacle associat a l'objecte `EspectacleDTO`.
- `public void setTitol(String titol)`
Estableix el títol de l'espectacle `titol` a l'objecte `EspectacleDTO`.
 - `titol`: Títol de l'espectacle que es vol inserir a l'objecte `EspectacleDTO`.
- `public String getTipus()`
Retorna un `String` amb el tipus de l'espectacle associat a l'objecte `EspectacleDTO`.
- `public void setTipus(String tipus)`
Estableix el tipus de l'espectacle `tipus` a l'objecte `EspectacleDTO`.
 - `tipus`: Tipus de l'espectacle que es vol inserir a l'objecte `EspectacleDTO`.
- `public String getComentaris()`
Retorna un `String` amb els comentaris de l'espectacle associat a l'objecte `EspectacleDTO`.
- `public void setComentaris(String comentaris)`
Estableix els comentaris de l'espectacle `comentaris` a l'objecte `EspectacleDTO`.
 - `comentaris`: Comentaris sobre l'espectacle que es volen inserir a l'objecte `EspectacleDTO`.
- `public String getWebEspectacle()`
Retorna un `String` amb l'adreça de la pàgina web de l'espectacle associat a l'objecte `EspectacleDTO`.
- `public void setWebEspectacle(String webEspectacle)`
Estableix l'adreça de la pàgina web de l'espectacle `webEspectacle` a l'objecte `EspectacleDTO`.
 - `webEspectacle`: Adreça de la pàgina web de l'espectacle que es vol inserir a l'objecte `EspectacleDTO`.

RepresentacioDTO

```
public class RepresentacioDTO implements Serializable
```

- `public String getIdRepresentacio()`
Retorna un `String` amb l'identificador de la representació associada a l'objecte `RepresentacioDTO`.
- `public void setIdRepresentacio(String IdRepresentacio)`
Estableix l'identificador de la representació `IdRepresentacio` a l'objecte `RepresentacioDTO`.
 - `IdRepresentacio`: Identificador de la representació que es vol inserir a l'objecte `RepresentacioDTO`.
- `public String getData()`
Retorna un `String` amb la data de la representació associada a l'objecte `RepresentacioDTO`.
- `public void setData(String data)`
Estableix la data de la representació `data` a l'objecte `RepresentacioDTO`.
 - `data`: Data de la representació que es vol inserir a l'objecte `RepresentacioDTO`.
- `public String getHora()`
Retorna un `String` amb l'hora de la representació associada a l'objecte `RepresentacioDTO`.
- `public void setHora(String hora)`
Estableix l'hora de la representació `hora` a l'objecte `RepresentacioDTO`.
 - `hora`: Hora de la representació que es vol inserir a l'objecte `RepresentacioDTO`.
- `public String getPreu()`
Retorna un `String` amb el preu d'una entrada per la representació associada a l'objecte `RepresentacioDTO`.
- `public void setPreu(String preu)`
Estableix el preu d'una entrada per la representació `preu` a l'objecte `RepresentacioDTO`.
 - `preu`: Preu en euros d'una entrada per la representació que es vol inserir a l'objecte `RepresentacioDTO`.

SalaDTO

```
public class SalaDTO implements Serializable
```

- `public String getIdSala()`
Retorna un `String` amb l'identificador de la sala associada a l'objecte `SalaDTO`.
- `public void setIdSala(String IdSala)`
Estableix l'identificador de la sala `IdSala` a l'objecte `SalaDTO`.
 - `IdSala`: Identificador de la sala que es vol inserir a l'objecte `SalaDTO`.
- `public String getNomSala()`
Retorna un `String` amb el nom de la sala associada a l'objecte `SalaDTO`.
- `public void setNomSala(String nomSala)`
Estableix el nom de la sala `nomSala` a l'objecte `SalaDTO`.
 - `nomSala`: Nom de la sala que es vol inserir a l'objecte `SalaDTO`.
- `public String getAdresa()`
Retorna un `String` amb l'adreça de la sala associada a l'objecte `SalaDTO`.
- `public void setAdresa(String adresa)`
Estableix l'adreça de la sala `adresa` a l'objecte `SalaDTO`.
 - `adresa`: Adreça de la sala que es vol inserir a l'objecte `SalaDTO`.
- `public String getTelefon()`
Retorna un `String` amb el telèfon de la sala associada a l'objecte `SalaDTO`.
- `public void setTelefon(String telefon)`
Estableix el telèfon de la sala `telefon` a l'objecte `SalaDTO`.
 - `telefon`: Número de telèfon de la sala que es vol inserir a l'objecte `SalaDTO`.
- `public String getWebSala()`
Retorna un `String` amb l'adreça de la pàgina web de la sala associada a l'objecte `SalaDTO`.
- `public void setWebSala(String webSala)`
Estableix l'adreça de la pàgina web de la sala `webSala` a l'objecte `SalaDTO`.

- `webSala`: Adreça de la pàgina web de la sala que es vol inserir a l'objecte `SalaDTO`.
- `public String getIdCiutat()`
Retorna un `String` amb l'identificador de la ciutat associada a l'objecte `SalaDTO`.
- `public void setIdCiutat(String IdCiutat)`
Estableix l'identificador de la ciutat `IdCiutat` a l'objecte `SalaDTO`.
 - `IdCiutat`: Identificador de la ciutat que es vol inserir a l'objecte `SalaDTO`.
- `public String getNomCiutat()`
Retorna un `String` amb el nom de la ciutat associada a l'objecte `SalaDTO`.
- `public void setNomCiutat(String nomCiutat)`
Estableix el nom de la ciutat `nomCiutat` a l'objecte `SalaDTO`.
 - `nomCiutat`: Nom de la ciutat que es vol inserir a l'objecte `SalaDTO`.
- `public String getProvincia()`
Retorna un `String` amb el nom de la província associada a l'objecte `SalaDTO`.
- `public void setProvincia(String provincia)`
Estableix el nom de la província `provincia` a l'objecte `SalaDTO`.
 - `provincia`: Nom de la província que es vol inserir a l'objecte `SalaDTO`.

ObjConsultaDTO

```
public class ObjConsultaDTO implements Serializable
```

- `public int getIdRsv()`
Retorna un `int` amb l'identificador de la reserva associat a l'objecte `ObjConsultaDTO`.
- `public void setIdRsv(int IdRsv)`
Estableix l'identificador de la reserva `IdRsv` a l'objecte `ObjConsultaDTO`.
 - `IdRsv`: Identificador de la reserva que es vol inserir a l'objecte `ObjConsultaDTO`.

- `public String getTitol()`
Retorna un `String` amb el títol de l'espectacle associat a l'objecte `ObjConsultaDTO`.
- `public void setTitol(String Titol)`
Estableix el títol de l'espectacle `Titol` a l'objecte `ObjConsultaDTO`.
 - `Titol`: Títol de l'espectacle que es vol inserir a l'objecte `ObjConsultaDTO`.
- `public String getData()`
Retorna un `String` amb la data de la representació associada a l'objecte `ObjConsultaDTO`.
- `public void setData(String Data)`
Estableix la data de la representació `Data` a l'objecte `ObjConsultaDTO`.
 - `Data`: Data de la representació que es vol inserir a l'objecte `ObjConsultaDTO`.
- `public String getHora()`
Retorna un `String` amb l'hora de la representació associada a l'objecte `ObjConsultaDTO`.
- `public void setHora(String Hora)`
Estableix l'hora de la representació `Hora` a l'objecte `ObjConsultaDTO`.
 - `Hora`: Hora de la representació que es vol inserir a l'objecte `ObjConsultaDTO`.
- `public String getCiutat()`
Retorna un `String` amb el nom de la ciutat associat a l'objecte `ObjConsultaDTO`.
- `public void setCiutat(String Ciutat)`
Estableix el nom de la ciutat `Ciutat` a l'objecte `ObjConsultaDTO`.
 - `Ciutat`: Nom de la ciutat que es vol inserir a l'objecte `ObjConsultaDTO`.
- `public String getRsvReco()`
Retorna un `String` amb informació sobre si la reserva associada a l'objecte `ObjConsultaDTO` està recollida o no.
- `public void setRsvReco(String RsvReco)`
Estableix informació `RsvReco` a l'objecte `ObjConsultaDTO` referent a si la reserva està recollida o no.
 - `RsvReco`: Informació sobre si la reserva que es vol inserir a l'objecte `ObjConsultaDTO` està recollida o no.

ObjFilaDTO

```
public class ObjFilaDTO implements Serializable
```

- `public String getIdFila()`
Retorna un `String` amb l'identificador de la fila associat a l'objecte `ObjFilaDTO`.
- `public void setIdFila(String IdFila)`
Estableix l'identificador de la fila `IdFila` a l'objecte `ObjFilaDTO`.
 - `IdFila`: Identificador de la fila que es vol inserir a l'objecte `ObjFilaDTO`.
- `public String getNumFila()`
Retorna un `String` amb el número de la fila associat a l'objecte `ObjFilaDTO`.
- `public void setNumFila(String NumFila)`
Estableix el número de la fila `NumFila` a l'objecte `ObjFilaDTO`.
 - `NumFila`: Número de la fila que es vol inserir a l'objecte `ObjFilaDTO`.

ObjListEspecDTO

```
public class ObjListEspecDTO implements Serializable
```

- `public String getIdEspec()`
Retorna un `String` amb l'identificador de l'espectacle associat a l'objecte `ObjListEspecDTO`.
- `public void setIdEspec(String IdEspec)`
Estableix l'identificador de l'espectacle `IdEspec` a l'objecte `ObjListEspecDTO`.
 - `IdEspec`: Identificador de l'espectacle que es vol inserir a l'objecte `ObjListEspecDTO`.
- `public String getTitoll()`
Retorna un `String` amb el títol de l'espectacle associat a l'objecte `ObjListEspecDTO`.
- `public void setTitoll(String Titoll)`
Estableix el títol de l'espectacle `Titoll` a l'objecte `ObjListEspecDTO`.
 - `Titoll`: Títol de l'espectacle que es vol inserir a l'objecte `ObjListEspecDTO`.

- `public String getComent()`
Retorna un `String` amb els comentaris sobre l'espectacle associat a l'objecte `ObjListEspecDTO`.
- `public void setComent(String Coment)`
Estableix els comentaris sobre l'espectacle `Coment` a l'objecte `ObjListEspecDTO`.
 - `Coment`: Comentaris sobre l'espectacle que es vol inserir a l'objecte `ObjListEspecDTO`.
- `public String getWeb()`
Retorna un `String` amb l'adreça de la pàgina web de l'espectacle associat a l'objecte `ObjListEspecDTO`.
- `public void setWeb(String Web)`
Estableix l'adreça de la pàgina web de l'espectacle `Web` a l'objecte `ObjListEspecDTO`.
 - `Web`: Adreça de la pàgina web de l'espectacle que es vol inserir a l'objecte `ObjListEspecDTO`.
- `public String getSala()`
Retorna un `String` amb el nom de la sala associada a l'objecte `ObjListEspecDTO`.
- `public void setSala(String Sala)`
Estableix el nom de la sala `Sala` a l'objecte `ObjListEspecDTO`.
 - `Sala`: Nom de la sala que es vol inserir a l'objecte `ObjListEspecDTO`.
- `public String getCiutat()`
Retorna un `String` amb el nom de la ciutat associada a l'objecte `ObjListEspecDTO`.
- `public void setCiutat(String Ciutat)`
Estableix el nom de la ciutat `Ciutat` a l'objecte `ObjListEspecDTO`.
 - `Ciutat`: Nom de la ciutat que es vol inserir a l'objecte `ObjListEspecDTO`.

ObjZonaDTO

```
public class ObjZonaDTO implements Serializable
```

- `public String getIdZona()`
Retorna un `String` amb l'identificador de la zona associat a l'objecte `ObjZonaDTO`.

- `public void setIdZona(String IdZona)`
Estableix l'identificador de la zona `IdZona` a l'objecte `ObjZonaDTO`.
 - `IdZona`: Identificador de la zona que es vol inserir a l'objecte `ObjZonaDTO`.
- `public String getNomZona()`
Retorna un `String` amb el nom de la zona associat a l'objecte `ObjZonaDTO`.
- `public void setNomZona(String NomZona)`
Estableix el nom de la zona `NomZona` a l'objecte `ObjZonaDTO`.
 - `NomZona`: Nom de la zona que es vol inserir a l'objecte `ObjZonaDTO`.

3.1.2 Classes DAO (Data Access Object)

La seva funció principal és la de realitzar els accessos a la base de dades de manera que la capa de la vista només hagi d'imprimir la informació de la forma adequada.

A continuació prosseguirem amb l'especificació d'aquestes classes:

AutenticarDAO

```
public class AutenticarDAO implements Serializable
```

- `public void autenticarDB (ClientDTO user)`
Estableix el tipus `user.tipus` d'un usuari si `user.nif` i `user.clau` existeixen i estan associats entre ells dins la base de dades.
 - `user.nif`: NIF de l'usuari del qual es vol obtenir el tipus.
 - `user.clau`: Clau de pas de l'usuari del qual es vol obtenir el tipus.

ConsultarDAO

```
public class ConsultarDAO implements Serializable
```

- `public List<ObjConsultaDTO> consultarDB (ClientDTO user, String data)`
Retorna una llista `List<ObjConsultaDTO>` amb totes les dades necessàries per consultar reserves.
 - `user.nif`: NIF de l'usuari sobre el qual es vol realitzar la consulta.

- **data**: Data actual del sistema a partir de la qual es realitza la consulta.

ModificarDAO

```
public class ModificarDAO implements Serializable
```

- **public void obtenirDadesDB (ClientDTO user)**
Obté totes les dades d'un usuari a partir de `user.nif` i les emmagatzema a `user`.
 - **user**: Objecte de tipus `ClientDTO` que permet emmagatzemar totes les dades referents a un usuari.
- **public void obtenirIdClientDB (ClientDTO user, String lastNif)**
Obté l'identificador de l'usuari a partir del NIF `lastNif` i l'emmagatzema a `user.IdClient`.
 - **user**: Objecte de tipus `ClientDTO` que permet emmagatzemar totes les dades referents a un usuari.
 - **lastNif**: Número inicial de NIF de l'usuari, previ a una possible modificació.
- **public void obtenirClauDB (ClientDTO user)**
Obté la clau de pas a partir de l'identificador del client `user.IdClient` i l'emmagatzema a `user.clau`.
 - **user.IdClient**: Identificador de l'usuari `user`.
- **public void modificaDB (ClientDTO user)**
Actualitza la base de dades amb tots els atributs de `user` mantenint intacte `user.IdClient`.
 - **user**: Objecte de tipus `ClientDTO` que conté totes les dades referents a un usuari.

PrincipalClientDAO

```
public class PrincipalClientDAO implements Serializable
```

- **public void principalClientDB (ClientDTO user)**
Estableix el nom complet de l'usuari (`user.nom`, `cognom1`, `cognom2`) a partir del seu NIF (`user.dni`).
 - **user**: Objecte de tipus `ClientDTO` que permet emmagatzemar totes les dades referents a un usuari.

RegistrarDAO

```
public class RegistrarDAO implements Serializable
```

- `public void registrarDB (ClientDTO user)`
Insereix a la base de dades un usuari `user` amb tipus d'usuari `user.tipus` com a usuari client.
 - `user`: Objecte de tipus `ClientDTO` que permet emmagatzemar totes les dades referents a un usuari.

ReservarDAO

```
public class ReservarDAO implements Serializable
```

- `public List<String> llistaTipusDB (String data)`
Retorna una llista `List<String>` amb tots els tipus d'espectacles que disposa l'aplicació a partir de la data `data`.
 - `data`: Data actual del sistema.
- `public List<ObjListEspecDTO> llistaEspectaclesDB (String data, String tipus)`
Retorna una llista `List<ObjListEspecDTO>` amb les dades d'interès dels espectacles d'un tipus concret `tipus` a partir d'una data `data`.
 - `data`: Data actual del sistema.
 - `tipus`: Tipus d'espectacle sobre el qual volem veure els espectacles.
- `public ObjListEspecDTO unEspectacleDB (String IdEspec, String data)`
Retorna un objecte `ObjListEspecDTO` amb les dades d'interès d'un espectacle a partir d'una data.
 - `IdEspec`: Identificador de l'espectacle del qual se'n vol obtenir les dades.
 - `data`: Data actual del sistema.
- `public List<RepresentacioDTO> llistaRepresentacionsDB (String IdEspec, String data)`
Retorna una llista `List<RepresentacioDTO>` amb les dades d'interès de les representacions d'un mateix espectacle a partir d'una data.
 - `IdEspec`: Identificador del l'espectacle del qual se'n vol llistar les representacions.

- **data**: Data actual del sistema.
- **public RepresentacioDTO unaRepresentacioDB (String IdRepre)**
Retorna un objecte **RepresentacioDTO** amb les dades d'interès d'una representació concreta.
 - **IdRepre**: Identificador de la representació de la qual en volem obtenir les dades.
- **public void checkEntradesDB (String IdRepre, String entrades)**
Revisa si hi ha suficients entrades disponibles per una representació concreta, si no hi són, llança un missatge d'error.
 - **IdRepre**: Identificador de la representació sobre la qual volem revisar si hi han suficients entrades disponibles.
 - **entrades**: És el nombre mínim d'entrades que han d'estar disponibles.
- **public List<ObjZonaDTO> checkByZonaDB (String IdRepre, String IdEspec, String entrades)**
Retorna una llista **List<ObjZonaDTO>** amb totes les zones que tenen suficients entrades i d'una representació concreta d'un espectacle.
 - **IdRepre**: Identificador de la representació sobre la qual volem revisar si hi han suficients entrades disponibles.
 - **IdEspec**: Identificador de l'espectacle al que pertany la representació sobre la qual volem revisar si hi han suficients entrades disponibles.
 - **entrades**: És el nombre mínim d'entrades que han d'estar disponibles a cada zona a llistar.
- **public String unaZonaDB(String IdZona)**
Retorna un **String** amb el nom de la zona.
 - **IdZona**: Identificador de la zona de la qual se'n vol obtenir el nom.
- **public List<ObjFilaDTO> checkByFilaDB (String IdRepre, String IdEspec, String IdZona, String entrades)**
Retorna una llista **List<ObjFilaDTO>** amb totes les files d'una zona concreta que tenen almenys **entrades** entrades d'una representació d'un espectacle.
 - **IdRepre**: Identificador de la representació sobre la qual volem revisar si hi han suficients entrades disponibles.

- **IdEspec**: Identificador de l'espectacle al que pertany la representació sobre la qual volem revisar si hi han suficients entrades disponibles.
 - **IdZona**: Identificador de la zona a la que pertanyen les files a llistar.
 - **entrades**: És el nombre mínim d'entrades que han d'estar disponibles a cada fila a llistar.
- **public String unaFilaDB(String IdFila)**
Retorna un **String** amb el número de la fila.
 - **IdFila**: Identificador de la fila de la qual se'n vol obtenir el número.
- **public List<ButacaDTO> llistaButacaDB (String IdZona, String IdFila, String IdEspec)**
Retorna una llista **List<ButacaDTO>** amb totes les butaques d'una fila i d'una zona que tenen tants seients disponibles com entrades es desitgen per un espectacle.
 - **IdZona**: Identificador de la zona a la que pertanyen les butaques a llistar.
 - **IdFila**: Identificador de la fila a la que pertanyen les butaques a llistar.
 - **IdEspec**: Identificador de l'espectacle al que pertany la representació sobre la qual volem revisar si hi han suficients entrades disponibles.
- **public List<ButacaDTO> selButacaDB(String[] selButaques, String entrades)**
Retorna una llista **List<ButacaDTO>** amb totes les butaques seleccionades.
 - **selButaques**: Array que conté els identificadors de les butaques seleccionades.
 - **entrades**: Número de les entrades que es volen reservar.
- **public String countPriceDB(String IdRepre, String entrades)**
Retorna un **String** amb el preu total (en euros) de totes les entrades que es volen reservar.
 - **IdRepre**: Identificador de la representació de la qual es volen reservar les entrades.
 - **entrades**: Nombre d'entrades que es volen reservar.

ResumDAO

```
public class ResumDAO implements Serializable
```

- `public SalaDTO obtenirSalaDB(String IdEspec)`
Retorna un objecte `SalaDTO` amb totes les dades associades a un espectacle.
 - `IdEspec`: Identificador de l'espectacle associat a la sala de la qual se'n volen obtenir les dades.

- `public List<String> insertReservesDB (ClientDTO user, String IdRepre, String IdEspec, String IdZona, String IdFila, List<ButacaDTO> llistaB)`
Insereix les reserves a la base de dades i retorna una llista `List<String>` amb els identificadors de les reserves inserides per un usuari en una representació d'un espectacle i amb unes butaques situades en una fila i en una zona concreta.
 - `user`: Objecte de tipus `ClientDTO` que conté totes les dades referents a un usuari.
 - `IdRepre`: Identificador de la representació de la qual es volen reservar entrades.
 - `IdEspec`: Identificador de l'espectacle del qual es volen reservar entrades.
 - `IdZona`: Identificador de la zona la qual conté les entrades que es volen reservar.
 - `IdFila`: Identificador de la fila la qual conté les entrades que es volen reservar.
 - `llistaB`: Llista de `ButacaDTO` de les butaques que es volen reservar.

3.2 Servlets

Els Servlets són programes que s'executen al servidor i permeten la generació de pàgines web dinàmicament. Avui en dia, ha quedat una mica en desús utilitzar únicament Servlets per generar contingut dinàmic, però donen bon resultat combinats amb pàgines JSP. [Han02]

A continuació prosseguirem amb l'especificació dels servlets usats a l'aplicació:

Autenticar

- `public class Autenticar extends HttpServlet`
Obté si l'usuari pertany al sistema i l'envia a la pàgina d'inici corresponent al tipus d'usuari al que correspon.
Si l'usuari no pertany al sistema, llança els missatges d'error corresponents.

Modificar

- `public class Registrar extends HttpServlet`
Efectua totes les comprovacions necessàries que determinen si les dades introduïdes al formulari són correctes.

Registrar

- `public class Modificar extends HttpServlet`
Efectua totes les comprovacions necessàries que determinen si les dades introduïdes al formulari són correctes.

3.3 Pàgines JSP (JavaServer Pages)

JSP és una tecnologia Java que permet generar pàgines web amb contingut dinàmic. Igual que els Servlets, s'executen en els servidors. Es programen dins la mateixa pàgina HTML incrustant-hi fragments de codi Java que s'anomenen scriptlets. [Han02]

A continuació prosseguirem amb l'explicació de les funcions de cada pàgina:

Autenticar

- `autenticar.jsp`
Vista del formulari d'autenticació.

Consultar

- `consultar.jsp`
Mostra una taula amb totes les reserves efectuades per l'usuari que ha iniciat la sessió.

Mostra el codi de reserva, el títol, la data l'hora, el lloc i si està recollida o no.

Error

- `error.jsp`
Es mostra quan es produeix un error, el que fa és imprimir els missatges d'error.

Modificar

- `modificar.jsp`
Mostra el formulari per la modificació de les dades personals d'un usuari. Mostra les dades actuals en mode editable. Permet canviar totes les dades referents a un usuari excepte el tipus, que l'ha de canviar directament un administrador.

PrincipalClient

- `principalclient.jsp`
Vista de la pàgina inicial de l'usuari client, amb totes les operacions que pot efectuar.

Registrar

- `registrar.jsp`

Mostra el formulari per a que un usuari client es pugui donar d'alta al sistema. Permet inserir totes les dades personals excepte el tipus d'usuari que ve prefixat. L'usuari client és l'únic usuari que es pot donar d'alta a ell mateix, als altres usuaris els ha de donar d'alta l'administrador.

Reservar

Per realitzar aquesta funció s'utilitzen les següents pàgines JSP:

- `reservar.jsp`
- `reservar2.jsp`
- `reservar3.jsp`
- `reservar4.jsp`
- `reservar5.jsp`
- `reservar6.jsp`
- `reservar7.jsp`
- `reservar8.jsp`

Cada pàgina mostra la informació escollida per l'usuari de les pàgines anteriors, en certa manera, es podria dir que es va acumulant la informació. A continuació, s'explica la funcionalitat general de l'apartat de Reservar i es va fent referència a cada pàgina que introdueix una nova opció.

- (1) Vista que mostra una llista desplegable amb els tipus d'espectacle (on només apareixen els tipus d'espectacle que hi ha al sistema) que permet escollir-ne un. [`reservar.jsp`]
- (2) Vista que mostra una llista amb els espectacles d'un tipus en concret, per escollir-ne un. Es mostra el títol, els comentaris, la pàgina web de l'espectacle i la ciutat. [`reservar2.jsp`]
- (3) Vista que mostra una llista amb les representacions d'un espectacle en concret, per escollir-ne una. Es mostra la data, l'hora i el preu. [`reservar3.jsp`]
- (4) Vista que mostra un formulari amb entrada de text, per inserir-hi el nombre d'entrades que es volen reservar. [`reservar4.jsp`]

- (5) Vista que mostra una llista desplegable amb les zones de la sala escollida¹ (només mostra les zones que tinguin suficients seients lliures) per seleccionar-ne una. [`reservar5.jsp`]
- (6) Vista que mostra una llista desplegable amb les files de la zona escollida (només mostra les files que tinguin suficients seients lliures) per seleccionar-ne una. [`reservar6.jsp`]
- (7) Vista que mostra una llista de selecció múltiple amb les butaques d'una fila (només amb les butaques que estiguin disponibles) per escollir-ne tantes com reserves es desitgin. [`reservar7.jsp`]
- (8) Vista que mostra el formulari amb el preu total de l'operació. [`reservar8.jsp`]

Resum

- `resum.jsp`
Mostra totes les dades que s'han anat seleccionant al efectuar la reserva i s'espera la confirmació de l'usuari si aquestes són correctes.

Concretament es mostra el nom, el tipus, els comentaris i la web de l'espectacle; també es mostra el nom, el telèfon, l'adreça, la ciutat, la província i la pàgina web de la sala; d'una banda es mostra la data i la hora de la representació i d'altra banda es mostra la zona, la fila i les butaques assignades; a més a més de mostrar la forma de pagament, el número de la tarja de crèdit emmagatzemat a la base de dades, el nombre d'entrades i el preu total.

- `resum2.jsp`
Mostra un missatge que comunica que la reserva s'ha efectuat amb èxit i mostra els identificadors de les reserves efectuades.

SortirSessio

- `sortirSessio.jsp`
Invalida la sessió així com totes les seves variables i mostra la pàgina d'inici `index.html`.

¹Com que cada espectacle està lligat únicament a una sala, llavors al escollir l'espectacle la sala s'assigna directament.

3.4 Excepcions (errors)

Són errors deguts a algun problema que ha tingut l'aplicació, però no són tant greus com per a que l'aplicació acabi dràsticament. Es poden detectar i manipular a la fi de corregir els problemes que les han ocasionat.

A continuació prosseguirem amb la seva especificació:

DuplicateUserDBException

- `public class DuplicateUserDBException extends Exception`
Aquesta excepció es llança quan, al voler inserir un usuari nou dins el sistema, aquest detecta que l'usuari ja existeix.

EmptyDBException

- `public class EmptyDBException extends Exception`
Aquesta excepció es llança quan s'efectua una consulta a la base de dades i aquesta no retorna res. Per tant, que la base de dades no conté cap dada de les que volem consultar.

FullDBException

- `public class FullDBException extends Exception`
Aquesta excepció es llança quan, al revisar si hi ha prou espai en files, zones o representacions per allotjar les entrades desitjades ens trobem que no.

BadLoginException

- `public class BadLoginException extends Exception`
Aquesta excepció es llança quan l'autenticació d'un usuari no s'efectua correctament, ja sigui perquè el nom d'usuari no existeix o perquè la clau de pas no està vinculada al nom d'usuari introduït, etc.

BadInformationException

- `public class BadInformationException extends Exception`
Aquesta excepció es llança quan la informació introduïda per l'usuari a qualsevol formulari de l'aplicació és incompleta o bé s'esperava un tipus d'informació diferent al que s'ha rebut.

3.5 Disseny de la base de dades

Per la base de dades utilitzarem PostgreSQL que és un SGBD (Sistema de Gestió de Bases de Dades). Té una alta protecció de la integritat de les dades.

3.5.1 Model relacional

La base de dades s'ha dissenyat seguint el model E-R (entitat-relació) que ofereix una gran riquesa semàntica. A trets generals, aquest model està format per:

- **Entitats**

Són objectes que es poden identificar fàcilment dels quals en volem emmagatzemar informació. Es poden representar mitjançant l'entitat tipus o classe i l'entitat ocurrència o instància. Però la primera és la més utilitzada, i és representada (en el model de domini) amb un rectangle amb el nom de l'entitat a l'interior.

- **Interrelacions**

Són associacions o vincles entre dos o més entitats. La seva funció és la d'unir les entitats entre elles i reben un nom que fa referència a la correspondència que efectuen entre les entitats.

- **Atributs**

Són característiques sobre les entitats i les interrelacions, és a dir, actuen com a propietats d'aquestes.

A continuació es mostra el model relacional, on el primer paràmetre de cada línia és el nom de l'entitat (o taula) i el que hi ha dins els parèntesis són els atributs.

Els atributs subratllats corresponen a la clau primària de la taula i els atributs en *cursiva* corresponen a les claus foranes.

Ciutat (IdCiutat, NomCiutat, Província)

Client (IdClient, DNI, Clau, Nom, Cognom1, Cognom2, Email, NVisa, TipusUsuari)

Sala (IdSala, NomSala, Adresa, Telefon, WebSala, *IdCiutat*)

Zona (IdZona, *IdSala*, NomZona)

Fila (IdFila, *IdZona*, *IdSala*, NumFila)

Butaca (IdButaca, IdFila, IdZona, IdSala, NumButaca)

Espectacle (IdEspectacle, Títol, Tipus, Comentaris, WebEspectacle,)

Representacio (IdRepresentacio, Data, Hora, Preu, *IdEspectacle*)

Reserva (IdReserva, Recollida, *IdRepresentacio*, *IdClient*, *IdSala*, *IdZona*, *IdButaca*)

3.5.2 Anàlisi de les entitats i atributs

En aquest apartat s'analitzen totes les entitats i s'expliquen detalladament els seus atributs.

Client

Fa referència a tots els usuaris que interactuen amb l'aplicació.

Es compon dels següents atributs:

- **IdClient:** Un identificador únic per cada usuari.
- **DNI:** Correspon al número del DNI amb la lletra de l'usuari, i al mateix temps realitza les funcions de login². Malgrat que cada NIF és diferent, no s'ha considerat adient col·locar-lo com a clau primària de la taula, ja que no és recomanable que una clau primària sigui cap element que l'usuari escrigui a mà, ja que en l'escriptura manual es poden cometre errors. [Mel07]
- **Clau:** És la clau de pas que permet a l'usuari accedir a l'aplicació.
- **Nom:** El nom de l'usuari.
- **Cognom1:** El primer cognom de l'usuari.
- **Cognom2:** El segon cognom de l'usuari.
- **Email:** Adreça de correu electrònic de l'usuari, en el cas dels clients, serà el medi amb el que se'ls notificaran les modificacions realitzades a l'aplicació que els puguin afectar.
- **NVisa:** És el número de la tarjeta de crèdit de l'usuari, a través de la qual efectuarà el pagament de les reserves.
- **TipusUsuari:** És l'identificador que distingeix cada usuari dins el seu rol.

²Identificador amb el qual l'usuari accedeix a l'aplicació.

Reserva

Emmagatzema tota la informació de les reserves realitzades.

Es compon dels següents atributs:

- **IdReserva:** Un identificador únic per cada reserva, mitjançant el qual podrà recollir l'entrada associada.
- **IdRepresentació:** Vincle amb l'identificador de la representació.
- **IdClient:** Vincle amb l'identificador del client que ha realitzat la reserva.
- **IdButaca:** Vincle amb l'identificador de la butaca.
- **IdFila:** Vincle amb l'identificador de la fila.
- **IdZona:** Vincle amb l'identificador de la zona.
- **IdSala:** Vincle amb l'identificador de la sala.
- **Recollida:** Ofereix informació de si l'entrada ha estat recollida o no, com que és un sistema fictici i no està vinculat a cap caixer automàtic, en un principi sempre és que no.

Representació

Conté la informació de les representacions.

Es compon dels següents atributs:

- **IdRepresentacio:** Un identificador únic per cada representació. Cal dir que cada reserva té un únic codi associat, no s'ha implementat el concepte d'entrada múltiple en que amb un sol codi s'englobin diverses entrades.
- **IdEspectacle:** Vincle amb l'identificador de l'espectacle.
- **Data:** Conté la data en que s'emetrà la representació.
- **Hora:** Conté l'hora en que s'emetrà la representació.
- **Preu:** És el preu, en euros, que valdrà una entrada per la representació.

Espectacle

Aquesta entitat guarda tota la informació relacionada amb un espectacle.

Es compon dels següents atributs:

- **IdEspectacle:** Un identificador únic per cada espectacle.
- **IdSala:** Vincle amb l'identificador de la sala. Per motius de senzillesa s'ha considerat el següent: que quan una sala compra un espectacle per a que sigui representat allà, també compra tots els drets d'aquell espectacle, obligant-lo així a que no pugui ser representat en cap altra sala.
- **Títol:** El títol de l'espectacle.
- **Tipus:** El tipus de l'espectacle. Generalment podrà ser cinema, esports, música o teatre. Però, si és necessari, se'n podran afegir de nous.
- **Comentaris:** Qualsevol tipus de comentaris que es vulguin realitzar sobre l'espectacle, qualsevol informació que es consideri important o una breu sinopsi.
- **WebEspectacle:** La pàgina web de l'espectacle (si en té).

Sala

Aquí només es guardarà la informació general de la sala. Els camps que es puguin compartir entre diverses sales seran emmagatzemats en altres entitats relacionades.

Es compon dels següents atributs:

- **IdSala:** Un identificador únic per cada sala.
- **NomSala:** El nom de la sala.
- **Adresa:** L'adreça de la sala.
- **Telefon:** El telèfon de la sala.
- **WebSala:** La pàgina web de la sala (si en té).
- **IdCiutat:** Vincle amb l'identificador de la ciutat on està la sala. Això es fa perquè pot haver-hi ciutats amb més d'una sala.

Zona

Emmagatzema tota la informació necessària d'una zona.

Es compon dels següents atributs:

- **IdZona:** Un identificador únic per cada zona.
- **NomZona:** El nom que reb la zona respecte la sala.
- **IdSala:** Vincle amb l'identificador de la sala.

Fila

Conté la informació necessària de cada fila.

Es compon dels següents atributs:

- **IdFila:** Un identificador únic per cada fila.
- **NumFila:** El número que té cada fila respecte la sala o la zona.
- **IdZona:** Vincle amb l'identificador de la zona.
- **IdSala:** Vincle amb l'identificador de la sala.

Butaca

En aquesta entitat es guardarà la informació relacionada amb la butaca.

Es compon dels següents atributs:

- **IdButaca:** Un identificador únic per cada butaca.
- **NumButaca:** El número que té cada butaca o seient.
- **IdFila:** Vincle amb l'identificador de la fila.
- **IdZona:** Vincle amb l'identificador de la zona.
- **IdSala:** Vincle amb l'identificador de la sala.

Ciutat

Aquí s'emmagatzema la informació de les ciutats que contenen alguna sala o que es preveu que en un futur pròxim tindran alguna sala.

Es compon dels següents atributs:

- **IdCiutat:** Un identificador únic per cada ciutat.
- **NomCiutat:** El nom de la ciutat.
- **Província:** El nom de la província a la que pertany.

Per a més informació sobre el model relacional, vegeu l'Apèndix A on hi ha l'estructura real de les taules SQL.

3.6 Llicència

Per tal d'afavorir la seva difusió i millora per part d'altres programadors, s'ha considerat oportú llicenciar aquest producte com a software lliure, seguint la llicència GPL (General Public License) de GNU. [FSF07]

Les llicències de software lliure permeten a l'usuari executar el programa, modificar-lo i/o redistribuir-lo. Per a que aquest concepte s'entengui cal fer menció de les 4 llibertats:

- Llibertat d'utilitzar el programa per qualsevol finalitat.
- Llibertat d'estudiar el programa i adaptar-lo a les necessitats de cada usuari.
- Llibertat de distribuir còpies (es poden distribuir gratuïtament o a canvi d'un preu).
- Llibertat de millorar el programa i publicar les millores per a que tothom se'n pugui beneficiar.

Per què s'ha de llicenciar?

La veritat es que la manera més fàcil de fer software lliure és posant el programa amb el seu codi a l'abast de tothom. Però no és massa recomanable degut a que qualsevol factoria de software podria adquirir-lo i privatitzar-lo.

La llicència GPL de GNU disposa de *copyleft*, que és un sistema legal que protegeix el software que en algun moment es va llicenciar com a software lliure per a que ningú l'agafi i el privatitzi. Ve a dir que el software que és llicenciat com a lliure, encara que es modifiqui, les noves versions han de continuar sent lliures.

Podeu trobar la llicència a l'apèndix B.

Capítol 4

Conclusions i treball futur

Una vegada arribat aquest moment, ens adonem que s'ha assolit el principal objectiu d'aquest projecte: realitzar una aplicació web orientada a la venda d'espectacles.

Cal esmentar que s'ha intentat tenir en compte el màxim de detalls possibles per permetre el futur creixement d'aquest projecte.

En la primera aproximació realitzada s'ha partit d'una base de dades existent, però a mesura que s'implementin les interfícies d'administrador i administrador web, la base de dades es podrà anar autorregenerant de manera natural.

Pel que fa al treball realitzat, òbviament, es poden millorar molts aspectes: personalment, distingiria entre aspectes físics i interns.

Pel que fa als aspectes físics, no s'ha d'oblidar que el TRE és una aplicació web dedicada a l'oci, i per aquesta raó, hauria de tenir una aparença agradable i atractiva, que captés l'atenció de l'usuari. Degut a això es proposa un treball acurat utilitzant fulles d'estil (CSS) i normes de disseny, per proporcionar-li una aparença elegant, ja que ara disposa d'un estil molt limitat.

En quant a aspectes interns, en primer lloc es proposa l'encriptació de la clau de pas (password) dels usuaris que interactuen amb aquesta, ja que totes les aplicacions haurien de tenir certes normes de seguretat, però encara més una aplicació que treballa amb les dades bancàries dels seus usuaris.

En segon lloc, es podria treballar en el tema del pagament bancari, posar-se en contacte amb entitats bancàries per poder efectuar el pagament real, ja que actualment és un projecte totalment fictici.

Com és del tot sabut, hi ha entitats bancàries que ja disposen del servei que es proposa en aquest projecte, però es podria negociar amb entitats bancàries que encara no en disposin per a que prepararessin i permetessin utilitzar els seus caixers automàtics per recollir entrades, però això ja és anar massa lluny...

Una altra possible millora (aquesta més real) seria la d'implementar el MVC (Model Vista Controlador) mitjançant Struts.

També seria necessari dotar aquesta aplicació de criteris d'usabilitat i accessibilitat.

I per acabar, un cop d'ull des del punt de vista personal: per una banda, he de fer un especial esment a la formació acadèmica adquirida durant aquests anys de carrera, que m'han permès assolir els coneixements de bases de dades i de programació necessaris per tal de poder dur a terme el present projecte.

D'altra banda, la realització del present treball, m'ha suposat que m'introduís en el món de Java, Servlets, JSP, CSS i aplicacions web a part de altres tecnologies sobre les que també m'he informat però que, malauradament, no he arribat a utilitzar.

Apèndix A

Taules SQL

```
CREATE TABLE Ciutat (  
    IdCiutat serial,  
    NomCiutat varchar(100),  
    Provincia varchar(100),  
    PRIMARY KEY (IdCiutat)  
);  
CREATE TABLE Client (  
    IdClient serial,  
    DNI varchar(9),  
    Clau varchar(200),  
    Nom varchar(30),  
    Cognom1 varchar(30),  
    Cognom2 varchar(30),  
    Email varchar(100),  
    NVisa varchar(16),  
    TipusUsuari varchar(1),  
    PRIMARY KEY (IdClient)  
);  
CREATE TABLE Sala (  
    IdSala serial,  
    NomSala varchar(100),  
    Adresa varchar(1000),  
    Telefon varchar(9),  
    WebSala varchar(100),  
    IdCiutat int,  
    FOREIGN KEY (IdCiutat) REFERENCES Ciutat (IdCiutat),  
    PRIMARY KEY (IdSala)  
);
```



```

CREATE TABLE Zona (
  IdZona serial,
  NomZona varchar(10),
  IdSala int,
  FOREIGN KEY (IdSala) REFERENCES Sala (IdSala),
  PRIMARY KEY (IdZona, IdSala)
);
CREATE TABLE Fila (
  IdFila serial,
  NumFila int,
  IdZona int,
  IdSala int,
  FOREIGN KEY (IdZona, IdSala) REFERENCES Zona (IdZona, IdSala),
  PRIMARY KEY (IdFila, IdZona, IdSala)
);
CREATE TABLE Butaca (
  IdButaca serial,
  NumButaca int,
  IdFila int,
  IdZona int,
  IdSala int,
  FOREIGN KEY (IdFila, IdZona, IdSala) REFERENCES Fila (IdFila,
  IdZona, IdSala),
  PRIMARY KEY (IdButaca, IdFila, IdZona, IdSala)
);
CREATE TABLE Espectacle (
  IdEspectacle serial,
  IdSala int,
  Titol varchar(200),
  Tipus varchar(15),
  Comentaris varchar(1000),
  WebEspectacle varchar(200),
  FOREIGN KEY (IdSala) REFERENCES Sala (IdSala),
  PRIMARY KEY (IdEspectacle)
);
CREATE TABLE Representacio (
  IdRepresentacio serial,
  IdEspectacle int,
  Data date,
  Hora time,
  Preu float,
  FOREIGN KEY (IdEspectacle) REFERENCES Espectacle (IdEspectacle),
  PRIMARY KEY (IdRepresentacio)
);

```

```
CREATE TABLE Reserva (  
    IdReserva serial,  
    IdRepresentacio int,  
    IdClient int,  
    IdButaca int,  
    IdFila int,  
    IdZona int,  
    IdSala int,  
    Recollida boolean,  
    FOREIGN KEY (IdRepresentacio) REFERENCES Representacio (IdRepresentacio),  
    FOREIGN KEY (IdClient) REFERENCES Client (IdClient),  
    FOREIGN KEY (IdButaca, IdFila, IdZona, IdSala) REFERENCES Butaca  
(IdButaca, IdFila, IdZona, IdSala),  
    PRIMARY KEY (IdReserva)  
);
```

Apèndix B

Llicència GPL

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you

received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities

but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect

to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- (a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- (b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- (c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- (d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an

“aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation’s users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- (a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- (b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- (c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- (d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

- (e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates

the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- (a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- (b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- (c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- (d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- (e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- (f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor’s “contributor version”.

A contributor’s “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work,

unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public

statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea
of what it does.>
```

```
Copyright (C) <year> <name of author>
```

```
This program is free software: you can redistribute
it and/or modify it under the terms of the GNU General
Public License as published by the Free Software
Foundation, either version 3 of the License, or (at
your option) any later version.
```

```
This program is distributed in the hope that it will
be useful, but WITHOUT ANY WARRANTY; without even the
implied warranty of MERCHANTABILITY or FITNESS FOR
A PARTICULAR PURPOSE. See the GNU General Public
License for more details.
```

```
You should have received a copy of the GNU General
Public License along with this program. If not, see
http://www.gnu.org/licenses/.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
```

```
This program comes with ABSOLUTELY NO WARRANTY; for
details type show w. This is free software, and you
are welcome to redistribute it under certain
conditions; type show c for details.
```

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, your program’s commands might be different; for a GUI interface, you would use an “about box”.

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

Bibliografía

- [Han02] Hanna, P. (2002). *JSP Manual de Referencia*. 1 ed. Aravaca. McGraw-Hill Osborne Media. ISBN: 84-481-3264-2002
- [LD06] Lima Díaz, F. (2006). *Java 2 v5.0*. 1 ed. Ediciones Anaya Multimedia. ISBN: 84-415-1946-3
- [Eck02] Eckel, B. *Piensa en Java*. 2 ed. Madrid. Prentice Hall, Pearson Educación. ISBN: 0-13-027363-5
- [Jav03] Java™ 2 Platform, Sun Microsystems, Inc. (2003). *API Specification*. Standard Edition, v1.4.2. [Consultat: 03/09/2007]. Disponible a Internet: <http://java.sun.com/j2se/1.4.2/docs/api/>
- [Man07] Manual de Java. [Consultat: 28/01/2007] Disponible a Internet: <http://manual-java.com/>
- [Mel07] MELLADO, J. (23/02/2007). *Cómo elegir la clave primaria de una tabla*. [Consultat: 14/03/2007]. Disponible a Internet: http://www.inmensia.com/blog/20070223/elegir_clave_primaria_tabla_base_datos_oracle.html
- [FSF07] Free Software Foundation, Inc. (07/09/2007). *GNU's Not Unix! - Free Software, Free Society*. Version 3. [Consultat: 20/09/2007]. Disponible a Internet: <http://www.gnu.org/>
- [FdI07] Factora de Internet S.L. (2007). *WebTaller.com* [Consultat: 21/09/2007]. Disponible a Internet: http://www.webtaller.com/manual-css/indice_manual_css.php