



Projecte Espectaclia

Memòria

Treball Final de Carrera – Àrea J2EE
Enginyeria Tècnica en Informàtica de Gestió (ETIG)

Héctor Paredes Tomás
hectorparedes@uoc.edu

Consultor: Salvador Campo Mazarico

11 de Gener de 2016

Dedicatòria i agraïments

Vull dedicar aquest projecte a tots aquells que han cregut en mi i m'han recolzat durant tots aquests anys de carrera, especialment a la família i amics. Voldria fer alguns agraïments especials...

A **Sonia**, per haver estat al meu costat durant tota la carrera i ser una de les persones que més m'ha motivat per començar, continuar i acabar aquests estudis amb èxit. A més, va ser ella qui va inspirar la temàtica del meu projecte. Gràcies per haver-me ajudat i animat durant tots aquests anys, sobretot durant els moments difícils. Sense tu molt segurament aquest projecte no seria una realitat. Mai t'ho podré agrair prou.

Als meus pares, **Pepi i Teo**, que tot i els altibaixos en els meus estudis universitaris han tingut paciència i han sabut recolzar-me quan ho he necessitat, sobretot durant aquests últims mesos de projecte que s'han ajuntat amb alguns problemes personals. Moltíssimes gràcies pel vostre afecte, suport i comprensió.

A tots aquells membres de **la meva família** que m'han motivat i animat des de petit a estudiar i lluitar per tenir un bon futur. Una menció especial a la meva **àvia Carmen**, que no s'adonarà que ja té un altre nét graduat, i al meu **avi Ventura**, que ja no ho podrà veure.

A tots els meus **amics i amigues** que heu sabut animar-me i distreure'm de les complicacions dels meus estudis, sobretot els caps de setmana al Rodamon.

Als meus **companys de la feina**, per ajudar-me amb diferents dubtes durant la carrera (sobretot amb aquest projecte) i a distreure'm de la sovint monòtona feina de programador; i als meus caps, per facilitar-me els horaris de treball i animar-me a acabar l'enginyeria.

Per últim, però no per això menys important, voldria agrair al meu **consultor** del projecte, el **Salvador Campo Mazarico**, tot el suport i consells donats durant aquests darrers mesos. I aprofito també per donar les gràcies a tots els consultors i personal de la UOC que durant el transcurs dels meus estudis han estat a l'altre costat de la pantalla per ajudar-me a tirar endavant.

Tots sou part del meu èxit.

Resum del projecte

Aquesta memòria conté les diferents fases del desenvolupament de l'aplicació web Espectaclia, fonamentada en la tecnologia J2EE que alhora constitueix l'àrea d'aquest Treball Final de Carrera.

El document es divideix en diferents seccions per tal d'avançar poc a poc en la creació de l'aplicació. Es comença amb una introducció en la que s'expliquen la justificació i objectius del projecte, així com la metodologia i planificació seguides per a la seva confecció, a més d'una breu explicació dels productes obtinguts. A continuació es realitza una anàlisi exhaustiva dels actors i funcionalitats de l'aplicació, juntament amb un prototipus de les interfícies d'usuari que tindrà. Tot seguit s'explica el disseny intern del programari així com l'arquitectura i tecnologies que utilitzarà. Després, s'expliquen i raonen les diferents modificacions i decisions de disseny que s'han hagut de fer durant la implementació de l'aplicació, i se'n farà una valoració econòmica del desenvolupament de tot l'aplicatiu. Finalment, s'exposen les conclusions a les que s'han arribat durant tot el procés. Per complementar el document, trobem un glossari de termes, la bibliografia i 2 annexos amb un manual d'instal·lació de l'aplicació i les sentències DDL per crear i omplir amb dades bàsiques la base de dades.

L'aplicació Espectaclia té com a objectiu principal la gestió integral de compra-venda d'entrades d'espectacles de tot tipus. Es tracta d'aconseguir una eina capaç d'integrar d'una banda la gestió de recintes amb els seus propis espectacles, zones amb o sense localitats, les funcions que s'hi duin a terme, els taquillers i els descomptes aplicables per part dels seus administradors, i d'altra banda donar accés a tota aquesta oferta d'espectacles a tots aquells clients que s'hi vulguin registrar. Els clients podran fer reserves de localitats per a assistir a aquests espectacles o esdeveniments, que hauran de pagar als taquillers també a través de l'aplicació, i un cop hagin assistit a l'espectacle podran fer-ne una valoració per a que altres clients puguin triar més i els administradors rebin feedback. Per tant, l'aplicació tindrà 3 tipus d'usuaris, Administradors, Clients i Taquillers, que s'hauran d'identificar i interactuar entre ells per tal de vendre entrades d'espectacles.

Àrea del TFC: J2EE

Paraules clau: j2ee, tomcat, spring, jsf, primefaces, hibernate, mysql, mvc, dao, administrador, client, taquiller, recinte, zona, espectacle, funció, localitat, preu localitat, descompte, reserva, valoració

Índex de continguts

DEDICATÒRIA I AGRAÏMENTS	2
RESUM DEL PROJECTE	3
ÍNDEX DE CONTINGUTS	4
INTRODUCCIÓ	6
JUSTIFICACIÓ DEL PROJECTE	6
OBJECTIUS DEL PROJECTE	7
<i>Objectius tècnics</i>	7
<i>Objectius funcionals</i>	7
ENFOCAMENT I MÈTODE SEGUIT	8
PLANIFICACIÓ DEL PROJECTE	9
PRODUCTES OBTINGUTS	11
ANÀLISI	12
IDENTIFICACIÓ DELS ACTORS	12
MODEL DE CASOS D'ÚS	13
ESPECIFICACIÓ DELS CASOS D'ÚS.....	13
<i>Identificar usuari</i>	14
<i>Gestionar client</i>	14
<i>Gestionar reserves</i>	15
<i>Valorar espectacle</i>	16
<i>Vendre reserva</i>	17
<i>Calcular preu reserva</i>	17
<i>Gestionar taquillers</i>	18
<i>Gestionar recintes</i>	18
<i>Gestionar zones</i>	19
<i>Gestionar espectacles</i>	20
<i>Gestionar funcions</i>	21
<i>Bloquejar localitats</i>	22
<i>Gestionar preus localitats</i>	22
<i>Gestionar descomptes</i>	23
<i>Generar informe estadístic</i>	24
PROTOTIPUS DE PANTALLES	25
DISSENY	38
DIAGRAMA DE CLASSES	38
DIAGRAMES D'ESTATS.....	40
DIAGRAMES D'ACTIVITAT	41
DIAGRAMA ER DE LA BASE DE DADES.....	45
ARQUITECTURA DE L'APLICACIÓ	47
<i>Arquitectura client - servidor i patró MVC</i>	47
<i>Tecnologies i frameworks utilitzats</i>	49
MODIFICACIONS DE DISSENY I IMPLEMENTACIÓ	52
VALORACIÓ ECONÒMICA	53

CONCLUSIONS	54
OBJECTIUS ASSOLITS	54
CONCLUSIONS PERSONALS	54
GLOSSARI	56
BIBLIOGRAFIA CONSULTADA.....	57
ANNEX 1: MANUAL D'INSTAL·LACIÓ	58
REQUERIMENTS DE SOFTWARE	58
<i>Java JDK (v 8u45)</i>	58
<i>Spring Tool Suite (v 3.6.4)</i>	58
<i>Apache Tomcat (v 7.0.62)</i>	58
<i>MySQL Server (v 5.6.10)</i>	58
INSTAL·LACIÓ DE L'APLICACIÓ	59
ANNEX 2: SENTÈNCIES DDL DE LA BASE DE DADES	61
DDL DE CREACIÓ	61
DDL DE DADES INICIALS	68

Introducció

Justificació del projecte

Actualment existeixen múltiples aplicacions per a la compra i venda d'entrades a través d'internet, però això és una veritat a mitges. Aplicacions web com www.ticketmaster.es, www.atrapalo.com o www.telentrada.com, són grans aplicacions per a la compra d'entrades, és a dir, enfocades al client o consumidor final que compra les entrades des de casa seva, amb eines de cerca i classificació molt eficients i amigables.

Però que passa amb el venedor, és a dir, l'entitat que emet les entrades, organitza els espectacles i n'és la beneficiària directe? Resulten igualment útils aquestes eines per als administradors de recintes? Bé, d'una banda sí, doncs aquestes aplicacions tan conegudes permeten l'accés a una gran quantitat de clients potencials, però la majoria tenen un gran inconvenient: no permeten el control directe del venedor. Normalment aquestes aplicacions les implementen i gestionen empreses que fan d'intermediàries entre el client i el venedor, tot enduent-se una comisió per cada venda (lògicament, en això es fonamenta el benefici del seu negoci).

D'altra banda, també existeixen eines de compra-venda d'entrades que manegen directament els administradors de recintes, però o bé són d'implementació i ús propis (normalment només enfocades a la venda directa per guixeta al propi recinte), o bé tenen multitud d'opcions i són una mica obsoletes (com és el cas de l'aplicació local de Telentrada), o bé no arriben a un nombre de clients potencials massa elevat degut a la seva pobre difusió. En resum, les eines que ofereixen una gestió i control directe per al venedor normalment no són accessibles als clients, o ho són de forma molt reduïda i limitada. A més a més, segurament només actuaran dins l'àmbit d'un únic recinte, limitant-se així l'oferta d'espectacles pel client.

He tingut l'ocasió de veure el funcionament de l'aplicació local de Telentrada en un recinte cultural, i com a desenvolupador d'aplicacions amb experiència em sembla altament complicada d'utilitzar, amb un disseny molt desferrat ple de botons i menús amb codis i que depèn d'un hardware específic. A més, té dependència absoluta dels servidors externs de Telentrada per al seu funcionament, cosa que trobo incoherent amb la necessitat del hardware específic local.

Amb aquest rerefons es planteja crear Espectaclia: una aplicació web que incorpori tant la gestió total i completa dels recintes, aforaments, espectacles i funcions per als venedors, empreses o entitats culturals (tant grans com petits), així com l'accés massiu als clients a través d'internet que permeti la cerca i reserva de localitats per a tots els recintes i espectacles que s'hagin donat d'alta al sistema. L'objectiu és crear un gran sistema d'informació que prescindeixi d'intermediaris, i per tant que la venda es produeixi directament entre el client i el venedor.

Objectius del projecte

Objectius tècnics

- L'objectiu tècnic bàsic d'aquest projecte és obtenir una aplicació web funcional fonamentada en una arquitectura client - servidor construïda amb tecnologia J2EE, aplicant els coneixements adquirits a diferents matèries de la carrera d'Enginyeria Tècnica en Informàtica de Gestió (ETIG), com poden ser Tècniques de desenvolupament del programari, Enginyeria del programari, Estructura de la Informació, Programació Orientada a Objectes i Bases de Dades (I i II).
- D'altra banda, s'estableix com a objectiu derivat de l'anterior aprofundir en el coneixement, la integració i la utilització de frameworks basats en J2EE per a la implementació de les diferents parts de l'aplicació. Concretament, la utilització del framework transversal Spring que serà alhora contenidor i connector de la resta de frameworks: *Java Server Faces* (JSF), *PrimeFaces* i *Hibernate*.
- També es defineix com a objectiu tècnic la correcta implementació del patró de disseny MVC (Model - Vista - Controlador) per tal de modelar i separar les diferents capes funcionals de l'aplicatiu, així com el patró DAO (*Data Access Object*) per a la implementació de la capa Model que conformarà la manipulació i accés a les dades de la persistència.
- Finalment, es marca com a últim objectiu l'execució de l'aplicació en un servidor d'aplicacions Tomcat i la utilització d'un servidor de base de dades MySQL per a la persistència.

Objectius funcionals

- L'objectiu funcional principal de l'aplicació Espectaclia és poder comunicar administradors i gestors de recintes d'espectacles directament amb el client final per a que aquest últim pugui reservar i pagar localitats d'aquests espectacles.
- Els administradors hauran de poder gestionar (donar d'alta, modificar i eliminar) els seus recintes culturals, així com les diferents zones dels patis de butaques, bloquejar localitats, gestionar els espectacles que s'hi duuen a terme i les seves funcions, preus per localitat i descomptes. Els administradors també hauran de poder gestionar els seus propis taquillers, encarregats de generar reserves pels clients que ho sol·licitin i de cobrar les entrades a guixeta.
- D'altra banda, els clients hauran de ser capaços de cercar espectacles del seu interès ordenant i filtrant per una sèrie de dades, triar un espectacle per llistar les funcions disponibles i reservar localitats (numerades o no) d'alguna de les zones del recinte. També podran gestionar aquestes reserves, i valorar-ne els espectacles un cop venudes.

Enfocament i mètode seguit

L'aplicació Espectaclia s'ha plantejat com una aplicació web per tal d'evitar els problemes que acostumen a derivar d'instal·lacions en entorns locals, on la compatibilitat i la dependència del hardware i altre software instal·lat poden portar greus complicacions. Així, només cal un ordinador amb qualsevol navegador web per utilitzar l'aplicació. A més, l'arquitectura client - servidor deriva gran part del treball computacional a la banda del servidor, fent que els requeriments dels ordinadors a través dels quals s'utilitzi l'aplicació sigui encara menors.

Pel que fa a l'enfocament seguit per confeccionar aquest projecte, s'ha decidit dividir-lo en una sèrie de fases:

- **FASE 1: Pla de treball**

En aquesta fase es realitza una descripció general del projecte així com una especificació preliminar de les funcionalitats que haurà de cobrir. En base a aquests requeriments s'estableix una planificació cronològica orientativa amb les diferents fases del projecte.

- **FASE 2: Especificació i anàlisi de requeriments**

Es realitza l'especificació final de funcionalitats. Tot seguit s'elabora el model de casos d'ús, identificant els actors i elaborant el diagrama de casos d'ús, per després fer l'explicació textual dels casos d'ús més importants en forma de fitxes. També s'elabora un prototipus de pantalles d'usuari.

- **FASE 3: Disseny i inici d'implementació**

En aquesta fase es fa el disseny intern de l'aplicació, identificant en primera instància les entitats i les seves propietats per després elaborar el diagrama de classes, els diagrames d'estat i d'activitat de les entitats més importants i el diagrama relacional de la BD. Finalment, es presenta l'arquitectura definitiva de l'aplicació. Paral·lelament s'inicia la implementació del projecte, creant l'esquelet de l'aplicació, instal·lant tots els components necessaris per al seu desenvolupament i s'implementen les classes d'entitat.

- **FASE 4: Implementació I**

Es comencen a implementar un a un els diferents casos d'ús per ordre de prioritat. El codi de cada cas es comenta allà on sigui necessari, i es fan les proves necessàries per determinar si el cas d'ús s'ha implementat correctament.

- **FASE 5: Implemetació II, Memòria i Presentació**

Es realitzen les darreres implementacions i proves de codi. Es fan proves generals per comprovar el correcte funcionament combinat de tots els casos d'ús. Finalment, es confecciona la Memòria i la Presentació virtual.

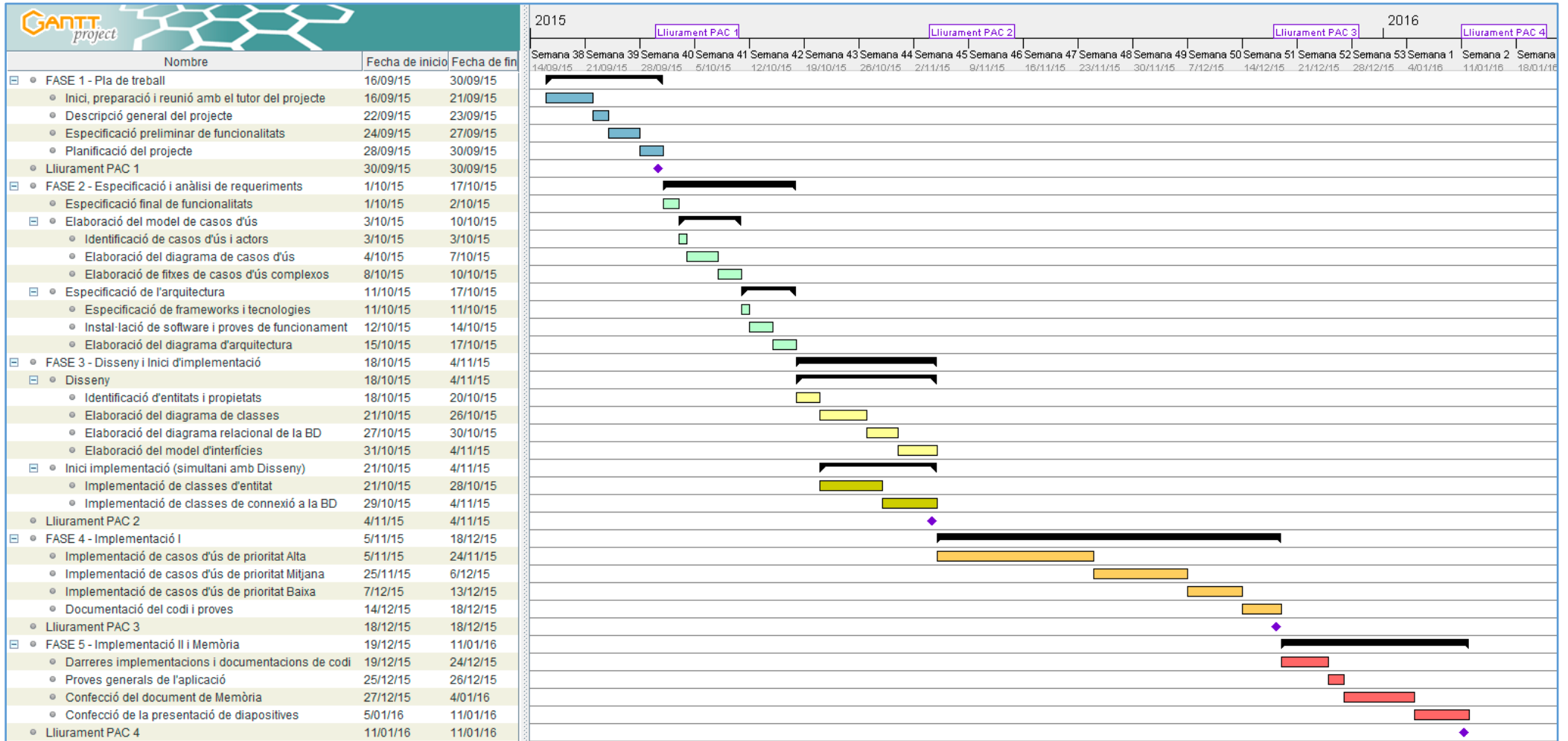
Pel que fa a la metodologia emprada en la implementació dels diferents casos d'ús, s'aprofitarà el caràcter reutilitzable, iteratiu i incremental que ofereix Java com a llenguatge de programació orientat a objectes que és per tal d'anar augmentant de forma repetida el grau de complexitat del codi a mesura que es vagin construint les diferents parts de cada cas d'ús. Així, s'aconsegueix implementar poc a poc l'aplicació, de forma compartimentada, començant per les estructures més senzilles i anar-les complicant poc a poc a mesura que s'hi va incorporant tota la lògica de negoci i les interfícies de comunicació amb l'usuari.

Planificació del projecte

El projecte s'ha planificat tenint en compte la divisió en fases explicada en l'apartat anterior. En la pàgina següent es mostra un diagrama de Gantt amb les dates d'inici i finalització de cadascuna d'aquestes fases, així com els diferents apartats que les componen. Com es pot veure, la majoria de les dates de finalització de les fases coincideixen amb les dates d'entrega de material del projecte, com és obvi.

Cal tenir en compte que aquest diagrama no contempla la corba d'aprenentatge de les diferents tecnologies que s'utilitzen al projecte, doncs aquest procés és duu a terme de forma continua durant tot el seu desenvolupament.

A continuació es detalla la planificació cronològica a mode orientatiu per a l'elaboració del projecte Espectaclia:



Productes obtinguts

Els productes obtinguts en finalitzar aquest projecte són els següents:

- Carpeta comprimida amb l'**aplicació web Espectaclia** (format .zip)
- Arxiu amb les **sentències DDL per crear la base de dades** des de consola (format .sql)
- **Memòria del projecte** (format .docx i .pdf)
- **Presentació virtual del projecte** (format .pptx)

Anàlisi

En aquest capítol s'analitzen els actors que prenen part en la utilització de les diferents funcionalitats de l'aplicació, així com els casos d'ús més destacables en què intervenen. A continuació, es presenta un prototipus de pantalles que modelitza la interfície de comunicació entre els actors i l'aplicació.

Identificació dels actors

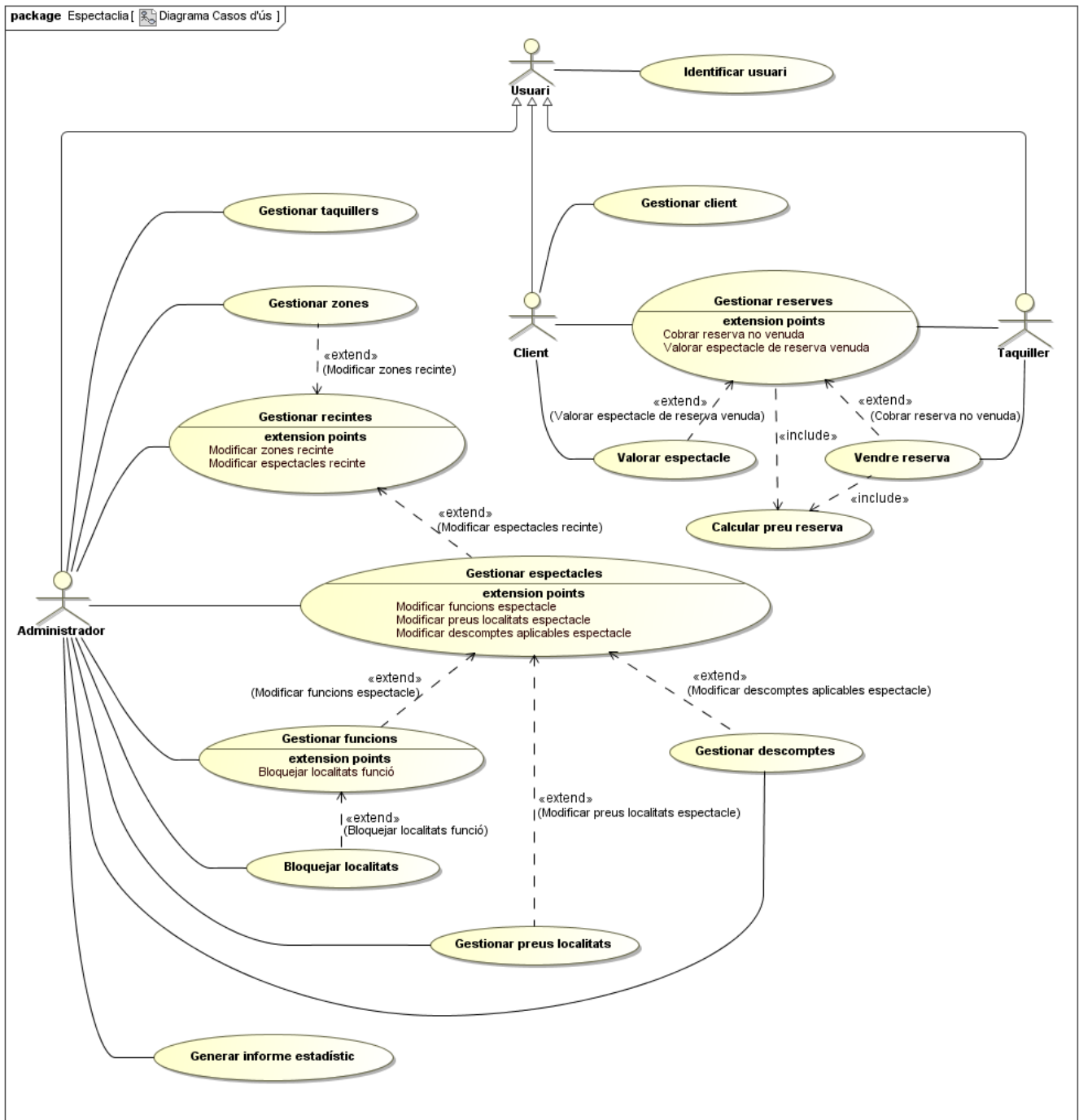
Cadascun dels tipus d'usuari que utilitza l'aplicació pot accedir a una sèrie de funcionalitats específiques, generalment no compartides amb els altres tipus d'usuari. És a dir, cadascun d'ells té un rol diferent. Aquesta caracterització s'estableix des del mateix moment en que accedeixen a l'aplicació quan s'identifiquen amb el seu usuari.

L'aplicació Espectaclia disposa de **3 rols d'usuari**:

- **Client:** És l'usuari més comú de l'aplicació. És l'únic tipus d'usuari que pot gestionar el seu propi usuari. D'altra banda, pot cercar espectacles i realitzar reserves de localitats per a funcions que tinguin localitats disponibles, podent fer una gestió completa de les seves reserves (creació, modificació i cancel·lació). A més, també pot valorar aquells espectacles dels que hagi fet alguna reserva totalment cursada, és a dir, que hagi arribat el dia de la funció i hagi pagat la reserva.
- **Taquiller:** Aquest tipus d'usuari s'encarrega principalment de tramitar el pagament de les reserves cursades pels clients dels espectacles del recinte cultural al que estigui assignat, encara que també pot gestionar una reserva de forma pràcticament idèntica a com ho faria un client. És a dir, també pot crear, modificar i cancel·lar una reserva, però sempre per petició expressa d'un client en particular. Els taquillers només els pot crear un usuari de tipus administrador.
- **Administrador:** És el rol que disposa d'accés a una major varietat de funcionalitats de l'aplicació. Pot gestionar íntegrament usuaris de tipus taquiller. Pot crear recintes culturals, modificar-ne les seves característiques i definir el seu aforament i pati de butaques si ho necessita, a més d'assignar-los taquillers. Per cada recinte, pot gestionar els seus espectacles de forma totalment transversal, des de les seves característiques més bàsiques a definir-ne funcions, bloquejar localitats, assignar preus a les diferents localitats i generar descomptes aplicables a les reserves que en facin els clients. En una futura ampliació de l'aplicació, també podrà generar informes estadístics amb informació sobre els seus recintes i espectacles. Els usuaris amb aquest rol, però, només poden ser generats manualment dins la base de dades de l'aplicació.

Model de casos d'ús

Analizant més a fons les tasques que poden desenvolupar els diferents actors dins l'aplicació, es presenta el següent diagrama de casos d'ús:



Especificació dels casos d'ús

A continuació, s'especifiquen en detall els diferents casos d'ús del model anterior en forma de fitxes:

Identificar usuari

- **Actors:** Client, Taquiller, Administrador.
- **Precondició:** L'usuari ha d'existir a l'aplicació.
- **Postcondició:** L'usuari està identificat a l'aplicació.
- **Casos d'ús relacionats:** TOTS (qualsevol usuari ha d'estar identificat abans de realitzar qualsevol dels altres casos d'ús).
- **Escenari principal:**
 1. L'usuari introdueix el seu identificador d'usuari i contrasenya.
 2. L'aplicació verifica que existeix un usuari amb aquestes credencials, l'autentifica i li concedeix accés a l'aplicació.
- **Fluxos alternatius:**
 - 2.a. L'aplicació comprova que no existeix cap usuari amb l'identificador proporcionat i mostra un missatge d'error.
 - 2.b. L'aplicació comprova que existeix l'usuari però que la contrasenya no es correcta, i mostra un missatge d'error.

Gestionar client

- **Actors:** Client.
- **Precondició:** Si el client vol modificar o eliminar el seu usuari, s'ha d'haver identificat prèviament.
- **Postcondició:** Cap.
- **Casos d'ús relacionats:** Identificar usuari.
- **Escenari principal:**
 1. Entre les opcions per gestionar el seu usuari, el client haurà de triar entre crear, modificar o eliminar.
 2. Si l'opció és crear o modificar, l'aplicació demana a l'usuari algunes dades personals per identificar el seu client.
 3. El client informa obligatòriament una serie de camps: usuari, contrasenya, nif, nom, cognoms i email de contacte. Opcionalment, i en el mateix pas, també pot indicar: telèfon, adreça i data de naixement.
 4. L'aplicació comprova que les dades introduïdes són correctes i demana confirmació al client per al seu emmagatzematge.
 5. El client confirma les dades i l'aplicació les emmagatzema, tot mostrant un missatge d'èxit.
- **Fluxos alternatius:**
 - 2.a. Si el client tria eliminar el seu client, i no té cap reserva pendent de pagar, l'aplicació demana confirmació, l'elimina i mostra un missatge d'èxit.
 - 2.b. Si el client tria eliminar el seu client, però té alguna reserva pendent de pagar, l'aplicació mostra un missatge d'error.
 - 4.a. Si l'aplicació comprova que falta alguna dada obligatòria, o que alguna de les dades és incorrecta, mostra un missatge d'error.
 - 5.a. Si el client ho creu convenient, pot triar cancel·lar l'emmagatzemament de les dades per tornar a modificar-les (torna a pas 3 de l'escenari principal).

Gestionar reserves

- **Actors:** Client, Taquiller.
- **Precondició:** L'usuari s'ha d'haver identificat prèviament.
- **Postcondició:** Cap.
- **Casos d'ús relacionats:** Identificar usuari, Valorar espectacle, Vendre reserva, Calcular preu reserva.
- **Escenari principal:**
 1. L'usuari accedeix a un llistat de les seves reserves: en el cas que sigui un usuari de tipus Client veurà les reserves que hagi cursat ell mateix, i en el cas d'un usuari de tipus Taquiller veurà totes les reserves no venudes realitzades sobre espectacles del recinte en el que treballa.
 2. En seleccionar una reserva, l'usuari haurà de triar una de les opcions per gestionar reserves: modificar o cancel·lar. També té l'opció de crear una nova reserva.
 3. Si l'opció és crear o modificar, l'usuari cerca l'espectacle sobre el que vol fer la reserva tot aplicant una serie de filtres opcionals.
 4. En triar un espectacle, es mostra a l'usuari una serie de dades identificatives i descriptives(nom, descripció, tipus, genere, autor, duració, ...), així com un llistat de les funcions disponibles indicant la data en que es duran a terme i el seu estat (activa o cancel·lada).
 5. En triar una funció, l'usuari té accés a un pati de butaques interactiu del recinte associat a aquell espectacle, on es mostren amb diferents colors la disponibilitat de les localitats per a aquesta funció, així com un llistat dels possibles descomptes als que podrà optar a taquilla.
 6. Si queden localitats disponibles, l'usuari tria les localitats de la zona que més li convenen fent clic sobre elles. Les localitats canvien de color segons estiguin seleccionades o no. Per deseleccionar una localitat, l'usuari torna a fer clic sobre ella. Amb cada canvi, l'usuari obté automàticament el preu total sense descomptes, resultat de l'execució del cas d'ús **Calcular preu reserva**.
 7. L'aplicació comprova que les dades introduïdes són correctes i demana confirmació al client per al seu emmagatzematge.
 8. El client confirma les dades de la reserva i l'aplicació les emmagatzema, tot mostrant un missatge d'èxit que adjunta un codiQR que identifica la reserva.
- **Fluxos alternatius:**
 - 3.a. Si l'usuari tria cancel·lar la reserva, i la reserva no ha estat venuda, l'aplicació demana confirmació, canvia el seu estat a "cancel·lada" i mostra un missatge d'èxit.
 - 3.b. Si l'usuari tria cancel·lar la reserva, però la reserva ha estat venuda, l'aplicació mostra un missatge d'error.
 - 3.c. Si l'usuari és de tipus Client i la reserva ha estat venuda, l'usuari té l'opció addicional valorar espectacle, que llença el cas d'ús **Valorar espectacle**.
 - 3.d. Si l'usuari és de tipus Taquiller, l'usuari té l'opció addicional vendre reserva, que llença el cas d'ús **Vendre reserva**.
 - 4.a. Si el client ho creu convenient, pot tornar enrere per triar una altra reserva (torna a pas 2 de l'escenari principal).
 - 5.a. Si el client ho creu convenient, pot tornar enrere per cercar un altre espectacle (torna a pas 3 de l'escenari principal).

- 6.a. Si no queden localitats disponibles en cap zona del recinte, es mostra un missatge informatiu.
- 6.b. Si el client ho creu convenient, pot tornar enrere per triar una altra funció de l'espectacle (torna a pas 4 de l'escenari principal).
- 7.a. Si l'aplicació comprova que falta alguna dada, o que alguna de les dades és incorrecta, mostra un missatge d'error.
- 8.a. Si el client ho creu convenient, pot triar cancel·lar l'emmagatzemament de les dades per tornar a modificar-les (torna a pas 6 de l'escenari principal).

Valorar espectacle

- **Actors:** **Client.**
- **Precondició:** El client s'ha d'haver identificat prèviament, la reserva de l'espectacle a valorar ha d'existir al sistema i ha d'estar venuda, i l'espectacle a valorar no ha d'haver estat valorat anteriorment pel mateix client.
- **Postcondició:** L'espectacle de la reserva està valorat pel client.
- **Casos d'ús relacionats:** **Identificar usuari, Gestionar reserves.**
- **Escenari principal:**
 1. El client valora l'espectacle triant una puntuació entre 5 opcions prefixades, de pitjor a millor: malament, regular, bé, molt bé o excel·lent. Opcionalment, també pot indicar un comentari associat a la seva puntuació.
 2. L'aplicació comprova que les dades introduïdes són correctes i demana confirmació al client per al seu emmagatzematge.
 3. El client confirma les dades de la valoració i l'aplicació les emmagatzema, tot mostrant un missatge d'èxit.
- **Fluxos alternatius:**
 - 3.a. Si el client ho creu convenient, pot triar cancel·lar l'emmagatzemament de les dades per tornar a modificar-les (torna a pas 1 de l'escenari principal).

Vendre reserva

- **Actors:** Taquiller.
- **Precondició:** El taquiller s'ha d'haver identificat prèviament, la reserva ha d'existir al sistema i no ha d'estar venuda.
- **Postcondició:** La reserva està venuda.
- **Casos d'ús relacionats:** Identificar usuari, Gestionar reserves, Calcular preu reserva.
- **Escenari principal:**
 1. El taquiller, opcionalment, pot triar un dels descomptes disponibles per a l'espectacle de la reserva, d'acord amb el client que ha cursat la reserva.
 2. El taquiller pot veure el preu final que tindrà la reserva, resultat de l'execució del cas d'ús Calcular preu reserva.
 3. L'aplicació comprova que les dades introduïdes són correctes i demana confirmació al taquiller per al seu emmagatzematge.
 4. El taquiller confirma les dades de la reserva, d'acord amb el client, i l'aplicació les emmagatzema, tot mostrant un missatge d'èxit. Se suposa que en aquest punt el client a abonat al taquiller el preu de la reserva.
- **Fluxos alternatius:**
 - 2.a. El taquiller, un cop informat el preu final de la reserva al client, pot tornar a triar un altre descompte si el client així o demana (torna a pas 1 de l'escenari principal).
 - 3.a. Si l'aplicació comprova que falta alguna dada, o que alguna de les dades és incorrecta, mostra un missatge d'error.
 - 4.a. Si el taquiller ho creu convenient, pot triar cancel·lar l'emmagatzemament de les dades per tornar a modificar-les (torna a pas 1 de l'escenari principal).

Calcular preu reserva

- **Actors:** Taquiller.
- **Precondició:** La reserva ha d'existir al sistema i no ha d'estar venuda.
- **Postcondició:** Preu de venda de la reserva.
- **Casos d'ús relacionats:** Identificar usuari, Gestionar reserves, Vendre reserva.
- **Escenari principal:**
 1. Es calcula el preu de la reserva a vendre en base a les seves dades, tals com el nombre de localitats, el preu de localitat per zona de l'espectacle i el descompte (opcional) triat pel taquiller.
 2. Es mostra per pantalla l'import final que el taquiller haurà de cobrar al client.
- **Fluxos alternatius:**

Cap.

Gestionar taquillers

- **Actors:** **Administrador.**
- **Precondició:** L'administrador s'ha d'haver identificat prèviament.
- **Postcondició:** Cap.
- **Casos d'ús relacionats:** **Identificar usuari.**
- **Escenari principal:**
 1. L'administrador accedeix a un llistat dels taquillers que ha creat prèviament.
 2. En seleccionar un taquiller, l'administrador haurà de triar una de les opcions per gestionar taquillers: modificar o eliminar. També té l'opció de crear un nou taquiller.
 3. Si l'opció és crear o modificar, l'aplicació demana a l'administrador algunes dades personals per identificar al taquiller.
 4. L'administrador informa obligatòriament una serie de camps sobre el taquiller: usuari, contrasenya, nif, nom i cognoms. Opcionalment, i en el mateix pas, també pot indicar a quin únic recinte el vol associar, tot seleccionant-lo en un llistat dels recintes creats per aquest administrador, o bé treure el que tinguéss assignat.
 5. L'aplicació comprova que les dades introduïdes són correctes i demana confirmació a l'administrador per al seu emmagatzematge.
 6. L'administrador confirma les dades del taquiller i l'aplicació les emmagatzema, tot mostrant un missatge d'èxit.
- **Fluxos alternatius:**
 - 3.a. Si l'administrador tria eliminar el taquiller, l'aplicació demana confirmació, l'elimina i mostra un missatge d'èxit.
 - 5.a. Si l'aplicació comprova que falta alguna dada obligatòria, o que alguna de les dades és incorrecta, mostra un missatge d'error.
 - 6.a. Si l'administrador ho creu convenient, pot triar cancel·lar l'emmagatzemament de les dades per tornar a modificar-les (torna a pas 4 de l'escenari principal).

Gestionar recintes

- **Actors:** **Administrador.**
- **Precondició:** L'administrador s'ha d'haver identificat prèviament.
- **Postcondició:** Cap.
- **Casos d'ús relacionats:** **Identificar usuari, Gestionar zones, Gestionar espectacles.**
- **Escenari principal:**
 1. L'administrador accedeix a un llistat dels recintes que ha creat prèviament.
 2. En seleccionar un recinte, l'administrador haurà de triar una de les opcions per gestionar recintes: modificar, gestionar zones, gestionar espectacles o eliminar. També té l'opció de crear un nou recinte.
 3. Si l'opció és crear o modificar l'aplicació demana a l'administrador algunes dades per identificar i descriure el recinte.
 4. L'administrador informa obligatòriament una serie de camps sobre el recinte: nom, tipus de recinte i aforament. Opcionalment, i en el mateix pas, també pot indicar: descripció, email i telèfon de contacte i adreça.

5. L'aplicació comprova que les dades introduïdes són correctes i demana confirmació a l'administrador per al seu emmagatzematge.
 6. L'administrador confirma les dades del recinte i l'aplicació les emmagatzema, tot mostrant un missatge d'èxit.
- **Fluxos alternatius:**
 - 3.a. Si l'administrador tria eliminar el recinte, i el recinte no té cap espectacle associat, l'aplicació demana confirmació, l'elimina i mostra un missatge d'èxit.
 - 3.b. Si l'administrador tria eliminar el recinte, però el recinte té algun espectacle associat, l'aplicació mostra un missatge d'error.
 - 3.c. Si l'administrador tria gestionar zones del recinte, i el recinte no té cap espectacle associat, es llença el cas d'ús **Gestionar zones**.
 - 3.d. Si l'administrador tria gestionar zones del recinte, però el recinte té algun espectacle associat, l'aplicació mostra un missatge d'error.
 - 3.e. Si l'administrador tria gestionar espectacles del recinte, es llença el cas d'ús **Gestionar espectacles**.
 - 5.a. Si l'aplicació comprova que falta alguna dada obligatòria, o que alguna de les dades és incorrecta, mostra un missatge d'error.
 - 6.a. Si l'administrador ho creu convenient, pot triar cancel·lar l'emmagatzemament de les dades per tornar a modificar-les (torna a pas 4 de l'escenari principal).

Gestionar zones

- **Actors:** **Administrador**.
- **Precondició:** L'administrador s'ha d'haver identificat prèviament i el recinte ha d'existir al sistema i no tenir cap espectacle associat.
- **Postcondició:** Cap.
- **Casos d'ús relacionats:** **Identificar usuari**, **Gestionar recintes**.
- **Escenari principal:**
 1. L'administrador accedeix a una representació gràfica de les zones que ha creat prèviament per al recinte seleccionat.
 2. En seleccionar una zona, l'administrador haurà de triar una de les opcions per gestionar zones: modificar o eliminar. També té l'opció de crear una nova zona.
 3. Si l'opció és crear o modificar l'aplicació demana a l'administrador algunes dades per tal de configurar la zona.
 4. L'administrador informa obligatòriament el nom, l'aforament i les coordenades inicial i final de la zona. Aquest parell de coordenades, cadascun format per una parella lletra-número de l'estil "A1" o "H8", s'utilitzaran per a determinar la posició de la zona dins el recinte en relació a altres zones, sense que aquestes se superposin. Opcionalment, en cas que l'administrador desitgi que la zona tingui localitats numerades, també pot indicar el nombre de files y el nombre de seients per fila de la zona (el seu producte haurà de coincidir amb el valor indicat al camp aforament). L'aplicació, doncs, només permet configurar zones de forma rectangular. En tot moment es mostra una representació gràfica de com quedaria configurada la zona en relació al recinte i les altres zones que s'hagin creat

anteriorment, que es va actualitzant a mesura que es modifiquen les dades.

5. L'aplicació comprova que les dades introduïdes són correctes i demana confirmació a l'administrador per al seu emmagatzematge.
 6. L'administrador confirma les dades de la zona i l'aplicació les emmagatzema, tot mostrant un missatge d'èxit.
- **Fluxos alternatius:**
 - 3.a. Si l'administrador tria eliminar la zona, l'aplicació demana confirmació, l'elimina i mostra un missatge d'èxit.
 - 5.a. Si l'aplicació comprova que falta alguna dada obligatòria, o que alguna de les dades és incorrecta, mostra un missatge d'error.
 - 6.a. Si l'administrador ho creu convenient, pot triar cancel·lar l'emmagatzemament de les dades per tornar a modificar-les (torna a pas 4 de l'escenari principal).

Gestionar espectacles

- **Actors:** **Administrador.**
- **Precondició:** L'administrador s'ha d'haver identificat prèviament i el recinte ha d'existir al sistema.
- **Postcondició:** Cap.
- **Casos d'ús relacionats:** **Identificar usuari, Gestionar recintes, Gestionar funcions, Gestionar preus localitats, Gestionar descomptes.**
- **Escenari principal:**
 1. L'administrador accedeix a un llistat dels espectacles que ha creat prèviament per al recinte seleccionat.
 2. En seleccionar un espectacle, l'administrador haurà de triar una de les opcions per gestionar recintes: modificar, gestionar funcions, gestionar preus de les localitats, gestionar descomptes o eliminar. També té l'opció de crear un nou espectacle.
 3. Si l'opció és crear o modificar l'aplicació demana a l'administrador algunes dades per identificar i descriure l'espectacle.
 4. L'administrador informa obligatòriament una serie de camps sobre l'espectacle: nom, descripció, tipus d'espectacle i duració. Opcionalment, i en el mateix pas, també pot indicar: gènere, autor i intèrprets.
 5. L'aplicació comprova que les dades introduïdes són correctes i demana confirmació a l'administrador per al seu emmagatzematge.
 6. L'administrador confirma les dades de l'espectacle i l'aplicació les emmagatzema, tot mostrant un missatge d'èxit.
- **Fluxos alternatius:**
 - 3.a. Si l'administrador tria eliminar l'espectacle, i l'espectacle no té cap funció activa, l'aplicació demana confirmació, l'elimina i mostra un missatge d'èxit.
 - 3.b. Si l'administrador tria eliminar l'espectacle, però l'espectacle té alguna funció activa, l'aplicació mostra un missatge d'error.
 - 3.c. Si l'administrador tria gestionar funcions de l'espectacle, es llença el cas d'ús **Gestionar funcions**.
 - 3.d. Si l'administrador tria gestionar preus de les localitats de l'espectacle, es llença el cas d'ús **Gestionar preus localitats**.

- 3.e. Si l'administrador tria gestionar descomptes de l'espectacle, es llença el cas d'ús **Gestionar descomptes**.
- 5.a. Si l'aplicació comprova que falta alguna dada obligatòria, o que alguna de les dades és incorrecta, mostra un missatge d'error.
- 6.a. Si l'administrador ho creu convenient, pot triar cancel·lar l'emmagatzemament de les dades per tornar a modificar-les (torna a pas 4 de l'escenari principal).

Gestionar funcions

- **Actors:** **Administrador**.
- **Precondició:** L'administrador s'ha d'haver identificat prèviament i l'espectacle ha d'existir al sistema.
- **Postcondició:** Cap.
- **Casos d'ús relacionats:** **Identificar usuari**, **Gestionar espectacles**, **Bloquejar localitats**.
- **Escenari principal:**
 1. L'administrador accedeix a un llistat de les funcions que ha creat prèviament per a l'espectacle seleccionat.
 2. En seleccionar una funció, l'administrador haurà de triar una de les opcions per gestionar funcions: modificar, bloquejar localitats o eliminar. També té l'opció de crear una nova funció.
 3. Si l'opció és crear o modificar l'aplicació demana a l'administrador algunes dades per configurar la funció.
 4. L'administrador informa obligatòriament la data i hora de la funció així com el seu estat (activa o cancel·lada).
 5. L'aplicació comprova que les dades introduïdes són correctes i demana confirmació a l'administrador per al seu emmagatzematge.
 6. L'administrador confirma les dades de la funció i l'aplicació les emmagatzema, tot mostrant un missatge d'èxit.
- **Fluxos alternatius:**
 - 3.a. Si l'administrador tria eliminar la funció, i la funció no té cap reserva no cancel·lada, l'aplicació demana confirmació, l'elimina i mostra un missatge d'èxit.
 - 3.b. Si l'administrador tria eliminar la funció, però la funció té alguna reserva no cancel·lada, l'aplicació mostra un missatge d'error.
 - 3.c. Si l'administrador tria bloquejar localitats de la funció, es llença el cas d'ús **Bloquejar localitats**.
 - 5.a. Si l'aplicació comprova que falta alguna dada obligatòria, o que alguna de les dades és incorrecta, mostra un missatge d'error.
 - 6.a. Si l'administrador ho creu convenient, pot triar cancel·lar l'emmagatzemament de les dades per tornar a modificar-les (torna a pas 4 de l'escenari principal).
 - 6.b. Si l'administrador indica l'estat de la funció com a "cancel·lada", quan confirma l'emmagatzemament de les dades s'envia un correu a tots aquells clients que tinguin una reserva no cancel·lada per a aquesta funció. A més, totes aquestes reserves passaran automàticament a l'estat "cancel·lada".

Bloquejar localitats

- **Actors:** **Administrador.**
- **Precondició:** L'administrador s'ha d'haver identificat prèviament i la funció ha d'existir al sistema i estar activa.
- **Postcondició:** Cap.
- **Casos d'ús relacionats:** **Identificar usuari, Gestionar funcions.**
- **Escenari principal:**
 1. L'administrador accedeix a un pati de butaques interactiu del recinte al que pertany l'espectacle de la funció, on es mostren amb diferents colors la disponibilitat de les localitats per a aquesta funció en relació a les reserves que n'hagin fet els clients i les localitats anteriorment bloquejades.
 2. En fer clic sobre una localitat lliure (no reservada) o bloquejada anteriorment, l'administració altera el seu estat de "lliure" a "bloquejada" i viceversa. Aquest canvi es veu reflectit amb un canvi de color.
 3. Quan l'administrador ha bloquejat / desbloquejat totes les localitats que desitja, pitja un botó per tal de guardar els canvis. L'aplicació demana confirmació a l'administrador per al seu emmagatzematge.
 4. L'administrador confirma els canvis i l'aplicació els emmagatzema, tot mostrant un missatge d'èxit.
- **Fluxos alternatius:**
 - 4.a. Si l'administrador ho creu convenient, pot triar cancel·lar l'emmagatzemament de les dades per tornar a modificar-les (torna a pas 2 de l'escenari principal).

Gestionar preus localitats

- **Actors:** **Administrador.**
- **Precondició:** L'administrador s'ha d'haver identificat prèviament i l'espectacle ha d'existir al sistema i no tenir reserves per cap de les seves funcions.
- **Postcondició:** Cap.
- **Casos d'ús relacionats:** **Identificar usuari, Gestionar espectacles.**
- **Escenari principal:**
 1. L'administrador accedeix a un pati de butaques interactiu del recinte al que pertany l'espectacle. Aquest pati de butaques només permet interactuar amb les zones, doncs l'aplicació només permet definir el preu de les localitats a nivell de zona i espectacle. Per tant, totes les localitats d'una mateixa zona tindran el mateix preu.
 2. En fer clic sobre una zona, l'administrador pot veure el seu preu assignat, si en té. Si ho desitja, pot modificar aquest preu o bé deixar-lo com està.
 3. Quan l'administrador ha realitzat tots els canvis de preus que desitja, pitja un botó per tal de guardar els canvis. L'aplicació comprova que les dades introduïdes són correctes i demana confirmació a l'administrador per al seu emmagatzematge.
 4. L'administrador confirma els canvis i l'aplicació els emmagatzema, tot

mostrant un missatge d'èxit.

- **Fluxos alternatius:**

3.a. Si l'aplicació comprova alguna de les dades és incorrecta, mostra un missatge d'error.

4.a. Si l'administrador ho creu convenient, pot triar cancel·lar l'emmagatzemament de les dades per tornar a modificar-les (torna a pas 2 de l'escenari principal).

Gestionar descomptes

- **Actors:** **Administrador.**

- **Precondició:** L'administrador s'ha d'haver identificat prèviament i l'espectacle ha d'existir al sistema.

- **Postcondició:** Cap.

- **Casos d'ús relacionats:** **Identificar usuari, Gestionar espectacles.**

- **Escenari principal:**

1. L'administrador accedeix a un llistat dels descomptes que ha creat prèviament per a l'espectacle seleccionat.

2. En seleccionar un descompte, l'administrador haurà de triar una de les opcions per gestionar descomptes: modificar o eliminar. També té l'opció de crear un nou descompte.

3. Si l'opció és crear o modificar, l'aplicació demana a l'administrador algunes dades per definir i descriure el descompte.

4. L'administrador informa obligatòriament una serie de camps sobre el descompte: nom, tipus i percentatge de descompte. Opcionalment, i en el mateix pas, també pot indicar una data d'inici i/o una data de fi del descompte. Aquestes dates es comprovaran quan un client o taquiller estiguin modificat una reserva, o bé quan un taquiller estigui venent una reserva, per tal de mostrar el descompte o no en funció de si la data en què s'estigui realitzant l'acció està dintre del rang definit per aquestes dues dates.

5. L'aplicació comprova que les dades introduïdes són correctes i demana confirmació a l'administrador per al seu emmagatzematge.

6. L'administrador confirma les dades del descompte i l'aplicació les emmagatzema, tot mostrant un missatge d'èxit.

- **Fluxos alternatius:**

3.a. Si l'administrador tria eliminar el descompte, l'aplicació demana confirmació, l'elimina i mostra un missatge d'èxit.

5.a. Si l'aplicació comprova que falta alguna dada obligatòria, o que alguna de les dades és incorrecta, mostra un missatge d'error.

6.a. Si l'administrador ho creu convenient, pot triar cancel·lar l'emmagatzemament de les dades per tornar a modificar-les (torna a pas 4 de l'escenari principal).

Generar informe estadístic

- **Actors:** **Administrador.**
- **Precondició:** L'administrador s'ha d'haver identificat prèviament.
- **Postcondició:** S'obté un informe estadístic per pantalla.
- **Casos d'ús relacionats:** **Identificar usuari.**
- **Escenari principal:**
 1. L'administrador tria un dels diferents tipus d'informe que pot generar l'aplicació: Informe de recaudació (anual, mensual o setmanal), Informe d'ús dels recintes, Informe de caracterització de clients.
 2. L'aplicació genera l'informe i el mostra per pantalla.
- **Fluxos alternatius:**

Cap.

Prototipus de pantalles

Tot seguit es presenta un prototipus de pantalles per a l'aplicació Espectaclia. Cal tenir en compte que aquest prototipus podrà variar en forma, estètica i continguts durant la fase d'implementació de l'aplicació, doncs només pretèn ser un disseny preliminar bàsic que serveixi per a entendre el funcionament de l'aplicació.

El model de pantalles que es proposa es fonamenta en un esquelet idèntic per a totes les pantalles format per una barra superior, un menú lateral a l'esquerra, i la zona de continguts a la dreta:

- A la **barra superior** podem trobar el nom de l'aplicació i el nom de l'usuari identificat, així com el nom del recinte que té assignat en cas d'un usuari de tipus Taquiller, o bé l'indicatiu "Administració" en cas d'un usuari de tipus Administrador.
- Al **menú lateral** es mostraran els accessos a les principals funcionalitats a les que tindrà accés l'usuari segons el seu rol.
- Finalment, a la **zona de continguts** s'aniran mostrant les dades i formularis d'edició de les diferents funcionalitats de l'aplicació que es vagin utilitzant. En aquesta zona també trobem una barra superior de "Breadcrumbs" (també anomenada guia-rastre) que permet ubicar a l'usuari en quin lloc de l'aplicació i en quina funcionalitat es troba.

Identificació d'usuari

Prototipus de pantalla d'identificació d'usuari per a Espectaclia.com. La pantalla mostra un navegador amb el títol "Navegador" i botons de control de finestra. El contingut principal és:

Benvingut a
Espectaclia.com

Usuari

Contrasenya

Pantalla d'inici

The screenshot shows a browser window titled "Navegador" with the address bar displaying "Espectaclia.com" and the user name "Héctor Paredes Tomás". The page has a sidebar with two menu items: "Inici" and "Gestionar reserves". The main content area is titled "Inici" and contains the text "Quin espectacle t'interessa?". Below this text is a search input field with the placeholder "Cercar espectacle..." and a "Cercar" button.

Gestió del client

The screenshot shows the "Modificar perfil" page in the browser window. The breadcrumb navigation is "Inici > Modificar perfil". A dropdown menu is open, showing options: "Modificar perfil", "Eliminar client", and "Sortir". The form contains the following fields:

- Nom *
- Cognoms *
- NIF *
- Data naixement (format: dd/mm/aaaa)
- Adreça
- Email *
- Telèfon
- Usuari *
- Contrasenya *

Below the fields, there is a note: "* Dades obligatòries". At the bottom of the form, there are two buttons: "Cancel·lar" and "Crear client / Guardar canvis".

Gestió de reserves

Espectaclia.com Héctor Paredes Tomás

Inici > Gestionar reserves

(C) Aquestes són les teves reserves / (T) Reserves del teu recinte 7 reserves

Data funció	Espectacle	Tipus	Recinte	# Locs.	Estat
10/08/2015	Los Minions	Pel·lícula	Cinesa Barnasud	3	Venuda
25/08/2015	La Traviata	Ópera	Teatre Liceu	2	Venuda
14/09/2015	Muse (Tour 2015)	Concert	Palau Sant Jordi	2	(C) Valorar espectacle
14/09/2015	Los Vengadores 2	Pel·lícula	Velmo Castelldefels	4	Venuda
07/10/2015	Mag Lari en acció	Actuació	Teatre Colisseum	2	Venuda
10/11/2015	Mar i Cel	Obra teatral	Teatre Victòria	4	Confirmada
18/12/2015	Star Wars: Episodi 7	Pel·lícula	Cinesa Barnasud	2	Confirmada

Modificar reserva
 Cancel·lar reserva
 (T) Vendre reserva

Crear reserva

(C) En el cas d'un usuari de tipus Client (T) En el cas d'un usuari de tipus Taquiller

Gestió de reserves - Cerca d'espectacle

Espectaclia.com Héctor Paredes Tomás

Inici > Gestionar reserves > Nova reserva / Modificar reserva > Cerca d'espectacle

mar i cel Cercar Tipus Tipus recinte Data funcions

Espectacle	Descripció	Tipus	Recinte	Gènere	Duració
Mar i Cel	Una obra sobre la intolerància i la incomprensió entre dos móns...	Obra teatral	Teatre Victòria	Musical	2h 30m

Tornar 1 espectacles

Gestió de reserves - Vista d'espectacle

Espectaclia.com Héctor Paredes Tomás

Inici > Gestionar reserves > Nova reserva / Modificar reserva > **Espectacle: Mar i Cel**

Nom: Mar i Cel

Descripció: Un espectacle sobre la intolerància i la incomprensió de dos móns, aparentment irreconciliables, Orient i Occident. Dues religions eternament enfrontades: la islàmica i la cristiana. Una bella història d'amor impossible entre una dona cristiana i un corsari morisc.

Tipus: Obra teatral

Gènere: Musical

Autor: Dagoll Dagom

Intèrprets: Carlos Gramaje, Roger Berruezo, Ana San Martín, Pep Cruz, Xavi Lite, Júlia Jové, Xavi Fernández, Toni Viñals, Mireia Dolg, Rubén Yuste, Elena Tarrats, ...

Duració: 2h 30m

Recinte: Teatre Victòria

31/10/2015	18:00		^
31/10/2015	21:00		—
01/11/2015	18:00	Cancel·lada	v

Tornar

Gestió de reserves - Vista de funció i sel·lecció de localitats

Espectaclia.com Héctor Paredes Tomás

Inici > Gestionar reserves > ... > Espectacle: Mar i Cel > **Funció: 31/10/2015 18:00**

Preu final [Veure descomptes aplicables](#)
2 x 29,95 € = 59,90 €

Selecciona les localitats que vols reservar

Platea A **Platea B**

15
14
13
12
11
10
9
8
7
6
5
4
3
2
1

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

ESCENARI

Disponible
 Seleccionada
 Ocupada
 Bloquejada

Tornar Confirmar reserva

Preu localitat: 29,95 €

- Família nombrosa (-10%)
- Carnet Jove (-8%)
- Carnet Estudiant (-8%)
- Jubilats (-8%)

Gestió de reserves - Reserva creada / actualitzada amb èxit

Navegador

Espectaclia.com Héctor Paredes Tomás

Inici > Gestionar reserves > Reserva creada / actualitzada

▶ Inici

▶ Gestionar reserves

La teva reserva de l'espectacle "Mar i Cel" s'ha creat / actualitzat amb èxit!

El dia de la funció hauràs d'abonar a taquilla **59,90 €**

Aquest és el codi QR que serveix d'identificador de la teva reserva. Presenta'l a taquilla el dia de la funció per facilitar la gestió del seu pagament.

Tornar al llistat de reserves

Valoració d'espectacle

Navegador

Espectaclia.com Héctor Paredes Tomás

Inici > Gestionar reserves > Valorar espectacle

▶ Inici

▶ Gestionar reserves

Espectacle	Mar i Cel
Tipus	Obra teatral
Gènere	Musical
Duració	2h 30m
Recinte	Teatre Victòria
Vist el dia	31/10/2015 18:00
Puntuació *	:::: TRIAR PUNTUACIÓ :::
Comentari (opcional)	Què t'ha semblat l'espectacle?...


* Dades obligatòries

Tornar Valorar espectacle

Venda de reserva

Espectaclia.com Teatre Victòria > Ramon Pérez Ramírez

Inici > Gestionar reserves > Vendre reserva

Identificador 

Client **Héctor Paredes Tomás**

Espectacle **Mar i Cel**

Tipus **Obra teatral**

Funció **31/10/2015 18:00**

Nº localitats **2**

Descompte a aplicar

	Nom	Tipus	Descompte
<input type="radio"/>	Família nombrosa	Família nombrosa	10%
<input checked="" type="radio"/>	Carnet Jove	Carnet descompte	8%
<input type="radio"/>	Carnet estudiant	Carnet descompte	8%

Preu final **59,9 € - 8% = 55,11 €**

[Tornar](#) [Vendre reserva](#)

Gestió de taquillers

Espectaclia.com Administració > Laura Millan Cercós

Inici > Gestionar taquillers

Aquests són els teus taquillers 7 taquillers

NIF	Nom	Cognoms	Recinte	Usuari
45211545O	Albert	Fernández Blasco	-	albertferblas
86786712H	Sonia	Caballero Cercós	Teatre Nou	soniacabcer
45864150A	Ramon	Pérez Ramírez	Teatre Nou	ramonperram
54545211E	Gerard	Varó Batlle	Teatre Balmes	gerardvarbat
45425445K	Nora	Gironés Baptista	Teatre Balmes	noragirbap
64577684B	Claudia	Bernal Capell	-	claudiabercap
67854255H	Amanda	Pastor García	Teatre El Cau	amandapasgar

[Modificar taquiller](#)
[Eliminar taquiller](#)

[Crear taquiller](#)

Gestió de taquillers - Creació / modificació

The screenshot shows a web browser window with the title 'Espectaclia.com' and the user 'Laura Millan Cercós'. The breadcrumb trail is 'Inici > Gestionar taquillers > Nou taquiller / Modificar taquiller: XXXXX'. The left sidebar contains menu items: 'Inici', 'Gestionar taquillers', 'Gestionar recintes', and 'Informes estadístics'. The main content area contains a form with the following fields:

- Nom *
- Cognoms *
- NIF *
- Recinte (dropdown menu)
- Usuari *
- Contrasenya *

Below the form, there is a note: '* Dades obligatòries'. At the bottom of the form, there are two buttons: 'Cancel·lar' and 'Crear taquiller / Guardar canvis'.

Gestió de recintes

The screenshot shows the 'Gestionar recintes' page. The breadcrumb trail is 'Inici > Gestionar recintes'. The left sidebar is the same as in the previous screenshot. The main content area displays the heading 'Aquests són els teus recintes' followed by '6 recintes'. Below this is a table with the following data:

Nom	Tipus	Aforament	# Espectacles	Adreça	
Teatre Nou	Teatre	600	5	C/Coloremes 54, Barcelona	⚙️
Teatre Balmes	Teatre	400	4	C/El colom 132, Barcelona	⚙️
Teatre El Cau	Teatre	300	6	Av. Roma 244, Barcelona	⚙️
Sala Cantinflas	Cinema	200	3	C/Rabal 78, Barcelona	⚙️
Sala l'Homenet	Sala polivalent	200	2	C/Indústria 32, Barcelona	⚙️
Sala Freddie Mercury	Sala concerts	500	5	C/Esperança 324, Barcelona	⚙️

A context menu is open over the last row, showing the following options: 'Modificar recinte', 'Gestionar zones', 'Gestionar espectacles', and 'Eliminar recinte'. At the bottom right of the page, there is a button labeled 'Crear recinte'.

Gestió de recintes - Creació / modificació

The screenshot shows a web browser window titled 'Navegador' with the URL 'Espectaclia.com'. The breadcrumb trail is 'Administració > Laura Millan Cercós > Inici > Gestionar recintes > Nou recinte / Modificar recinte: XXXXX'. The left sidebar contains menu items: 'Inici', 'Gestionar taquillers', 'Gestionar recintes', and 'Informes estadístics'. The main content area is a form for creating or editing a venue. The form fields are: 'Nom *' (text input), 'Descripció' (text area), 'Tipus recinte *' (dropdown menu with ':::: TRIAR TIPUS ::::'), 'Aforament *' (text input), 'Email contacte' (text input), 'Telèfon contacte' (text input), and 'Adreça' (text input). A note below the fields reads '* Dades obligatòries'. At the bottom, there are two buttons: 'Cancel·lar' and 'Crear recinte / Guardar canvis'.

Gestió de zones

The screenshot shows the 'Gestió de zones' interface for the 'Teatre Nou' venue. The breadcrumb trail is 'Administració > Laura Millan Cercós > Inici > Gestionar recintes > Recinte: Teatre Nou > Gestionar zones'. The left sidebar is the same as in the previous screenshot. The main content area displays the title 'Aquestes són les zones del recinte "Teatre Nou"' and '2 zones'. Below this, there are two grid-based seating charts labeled 'Platea A' and 'Platea B'. A context menu is open over the 'Platea A' grid, showing options 'Modificar zona' and 'Eliminar zona'. Below the grids, the word 'ESCENARI' is centered. At the bottom, there are two buttons: 'Tornar' and 'Nova zona'.

Gestió de zones - Creació / Modificació

Navegador

Espectaclia.com Administració > Laura Millan Cercós

Inici > Gestionar recintes > Recinte: Teatre Nou > Nova zona / Modificar zona: XXXXX

▶ Inicial
 ▶ Gestionar taquillers
 ▶ Gestionar recintes
 ▶ Informes estadístics

Nom * Anfiteatre
 Aforament *
 Nº files
 Nº seients per fila } Només si la zona té localitats numerades
 Coordenada inici * A1
 Coordenada final * C12 * Dades obligatòries
 Previsualització

Gestió d'espectacles

Navegador

Espectaclia.com Administració > Laura Millan Cercós

Inici > Gestionar recintes > Recinte: Teatre Nou > Gestionar espectacles

▶ Inicial
 ▶ Gestionar taquillers
 ▶ Gestionar recintes
 ▶ Informes estadístics

Aquests són els espectacles del recinte "Teatre Nou" 5 espectacles

Espectacle	Tipus	Gènere	Duració	# Funcions	
La vaca i el pollastre	Obra teatral	Infantil	1h 20m	10	⚙️
L'ovella Dolly	Obra teatral	Infantil	1h 30m	10	⚙️
Crim al soterrani	Obra teatral	Suspens	2h	15	⚙️
La mare que el va parir!	Obra teatral	Comèdia	2h 30m	20	⚙️
Tot per no res	Obra teatral	Tragicomèdia	2h 15m	15	⚙️

Modificar espectacle
 Gestionar funcions
 Gestionar preus localitats
 Gestionar descomptes
 Eliminar espectacle

Gestió d'espectacles - Creació / Modificació

Navegador

Espectaclia.com Administració > Laura Millan Cercós

Inici > Gestionar recintes > Recinte: Teatre Nou > Nou espectacle / Modificar espectacle: XXXXX

Tipus espectacle *

* Dades obligatòries

Gestió de funcions

Navegador

Espectaclia.com Administració > Laura Millan Cercós

Inici > Gestionar recintes > Recinte: Teatre Nou > Espectacle: Tot per no res > Gestionar funcions

Aquestes són les funcions de l'espectacle "Tot per no res" 15 funcions

Data	Hora	Estat	# Reserves	
01/11/2015	16:00	Activa	126	
01/11/2015	18:30	Activa	213	
01/11/2015	21:00	Activa	156	
02/11/2015	16:00	Cancel·lada	56	
02/11/2015	18:30	Activa	321	
02/11/2015	21:00	Activa	248	
03/11/2015	16:00	Cancel·lada	29	
03/11/2015	18:30	Activa	212	
03/11/2015	21:00	Activa	134	
04/11/2015	16:00	Cancel·lada	0	
04/11/2015	18:30	Activa	278	
04/11/2015	21:00	Activa	197	

Gestió de funcions - Creació / Modificació

The screenshot shows a web browser window titled 'Espectaclia.com' with the user 'Laura Millan Cercós' in the 'Administració' section. The breadcrumb trail is 'Inici > Gestionar recintes > Recinte: ... > Espectacle: ... > Nova funció / Modificar funció: XXXXX'. The left sidebar contains menu items: 'Inici', 'Gestionar taquillers', 'Gestionar recintes', and 'Informes estadístics'. The main content area contains a form with the following fields:

- Data ***: A text input field.
- Hora ***: A text input field.
- Estat ***: A dropdown menu currently set to 'ACTIVA'.

Below the form, there is a note: '* Dades obligatòries'. At the bottom of the form, there are two buttons: 'Cancel·lar' and 'Crear funció / Guardar canvis'.

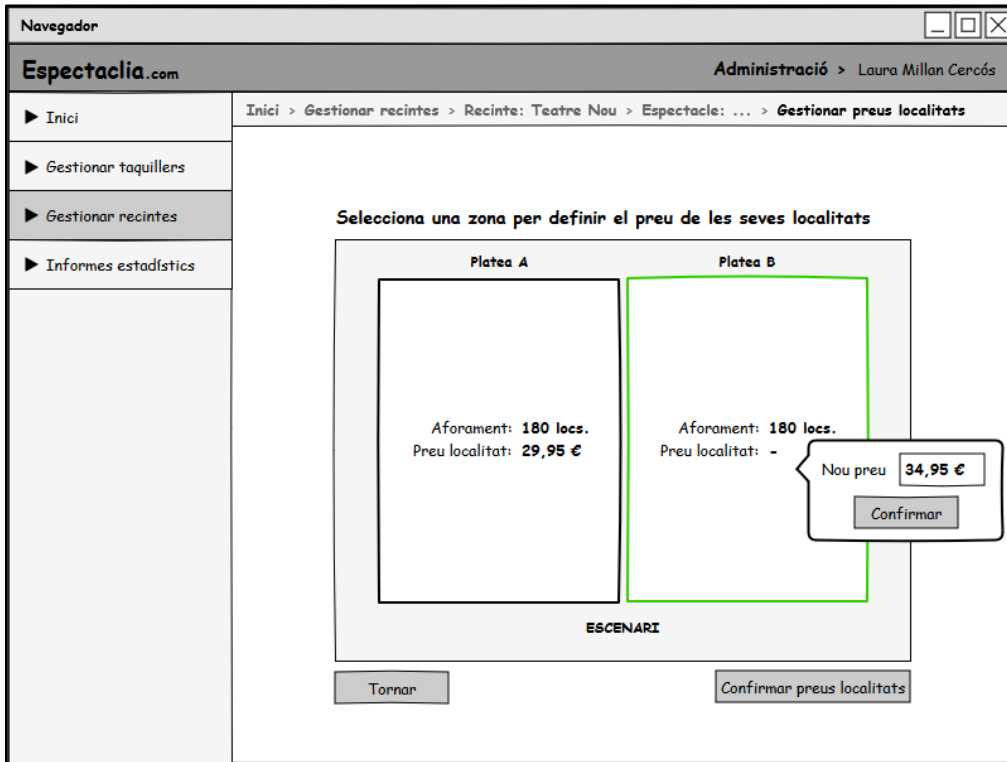
Bloqueig de localitats

The screenshot shows the 'Bloquejar localitats' (Block seats) interface. The breadcrumb trail is 'Inici > Gestionar recintes > Recinte: Teatre Nou > Espectacle: Tot per no res > Bloquejar localitats'. The left sidebar is the same as in the previous screenshot. The main content area is titled 'Selecciona les localitats que vols bloquejar' (Select the seats you want to block). It displays two seating charts, 'Platea A' and 'Platea B', each with 15 rows and 12 columns. The seats are represented by a grid of colored squares:

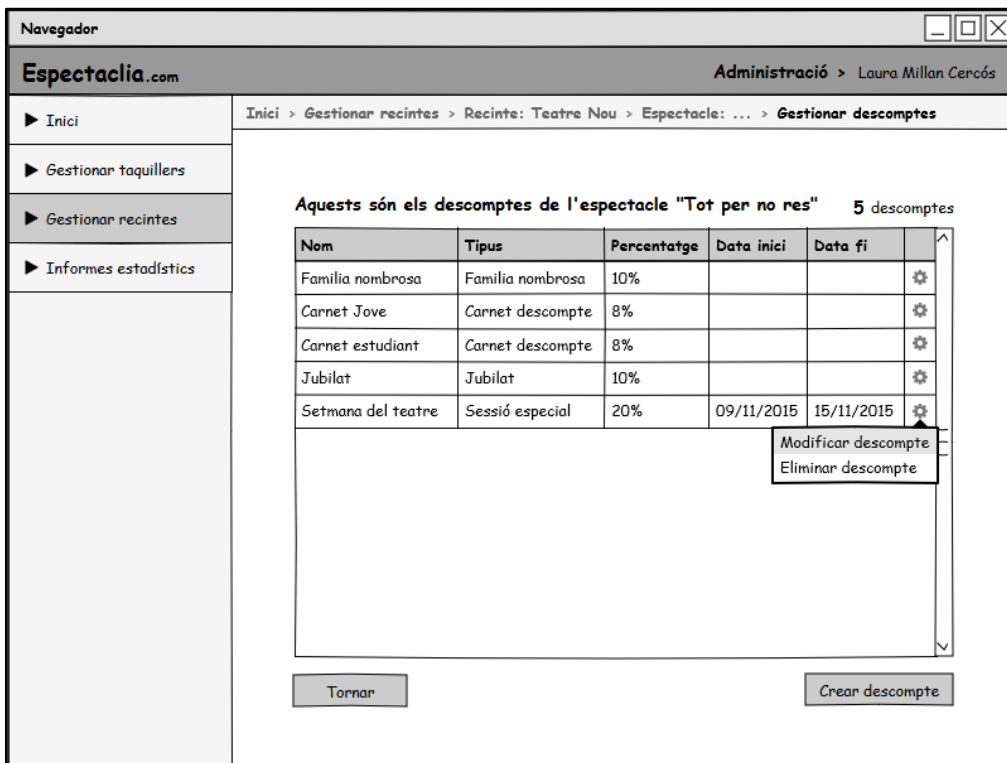
- White squares**: Available seats.
- Red squares**: Reserved seats.
- Blue squares**: Blocked seats.

The 'ESCENARI' (Stage) is labeled at the bottom of the grid. A legend below the grid identifies the colors: 'Disponibile' (white), 'Reservada' (red), and 'Bloquejada' (blue). At the bottom of the interface, there are two buttons: 'Tornar' (Return) and 'Confirmar localitats bloquejades' (Confirm blocked seats).

Gestió de preus de localitats



Gestió de descomptes



Gestió de descomptes - Creació / Modificació

The screenshot shows a web browser window titled 'Navegador' with the URL 'Espectaclia.com'. The page header indicates the user is 'Laura Millan Cercós' in the 'Administració' section. The breadcrumb trail is 'Inici > Gestionar recintes > ... > Gestionar descomptes > Nou desc. / Modificar desc.: XXXX'. The left sidebar contains navigation links: 'Inici', 'Gestionar taquillers', 'Gestionar recintes', and 'Informes estadístics'. The main content area is a form for creating or editing a discount. It includes the following fields and controls:

- Nom ***: A text input field.
- Tipus descompte ***: A dropdown menu with the text '::: TRIAR TIPUS :::'. A small downward arrow is visible on the right side of the dropdown.
- Percentatge descompte ***: A text input field.
- Data inici descompte**: A date input field.
- Data fi descompte**: A date input field.
- * Dades obligatòries**: A note indicating that the fields with an asterisk are required.
- Buttons**: 'Cancelar' and 'Crear descompte / Guardar canvis'.

Generació d'informes estadístics

The screenshot shows a web browser window titled 'Navegador' with the URL 'Espectaclia.com'. The page header indicates the user is 'Laura Millan Cercós' in the 'Administració' section. The breadcrumb trail is 'Inici > Informes estadístics'. The left sidebar contains navigation links: 'Inici', 'Gestionar taquillers', 'Gestionar recintes', and 'Informes estadístics'. The main content area displays the question 'Quin tipus d'informe vols generar?' and three buttons for selecting the report type:

- Informe de recaudació
- Informe d'ús dels recintes
- Informe de caracterització dels clients

Cal tenir en compte que aquest diagrama no reflexa algunes de les restriccions de negoci comentades al llarg de l'anàlisi dels casos d'ús. Aquestes restriccions s'aplicaran en la lògica de negoci durant el desenvolupament de l'aplicació:

- No es poden realitzar diferents reserves per a un mateix espectacle i funció sobre les mateixes localitats. Si una zona d'un recinte no té localitats numerades, un mateix usuari no podrà fer més d'una reserva per a un mateix espectacle i funció sobre aquesta zona.
- Un client només pot cancel·lar reserves en estat "Confirmada". Un cop cancel·lada, el client no podrà recuperar ni modificar aquella reserva, n'haurà de fer una de nova.
- Només es poden fer valoracions d'espectacles dels quals el client tingui reserves en estat "Venuda". Cada client només podrà fer una única valoració per espectacle.
- El taquiller només podrà vendre reserves en estat "Confirmada".
- L'administrador només pot gestionar zones d'un recinte sempre i quan no tingui espectacles associats.
- L'administrador no podrà eliminar cap tipus d'entitat relacionada amb una reserva (recintes, zones, espectacles, funcions, descomptes, ...) si hi ha alguna reserva no cancel·lada vigent relacionada amb alguna d'elles. Prèviament haurà de cancel·lar totes les funcions per a les que hi hagi reserves i anar eliminant entitats en ordre de dependència (per exemple, funcions > espectacles > zones > recintes).

Diagrames d'estats

Tant les reserves com les funcions d'espectacles poden passar per diversos estats dins la dinàmica de l'aplicació. A continuació es mostren els diagrames que modelitzen aquestes transicions d'estats.

Diagrama d'estats d'una reserva

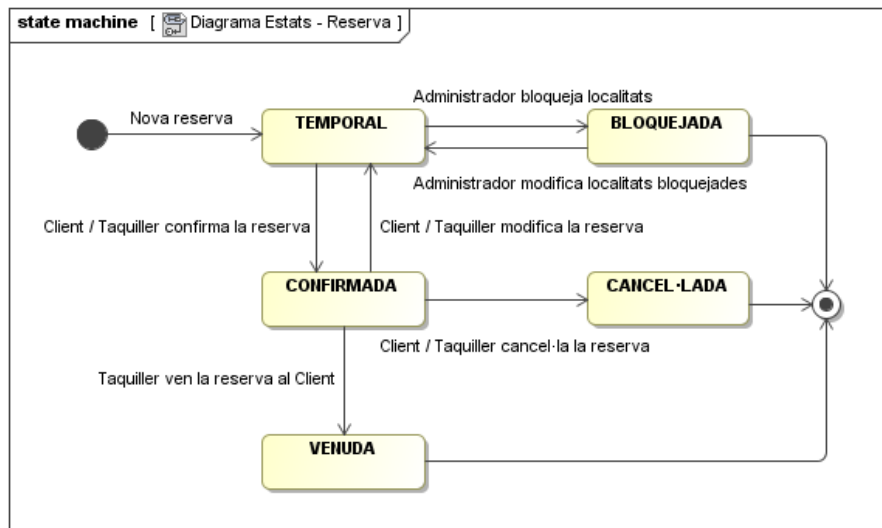
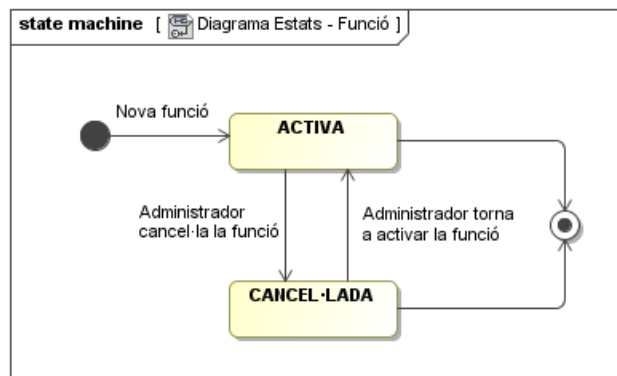


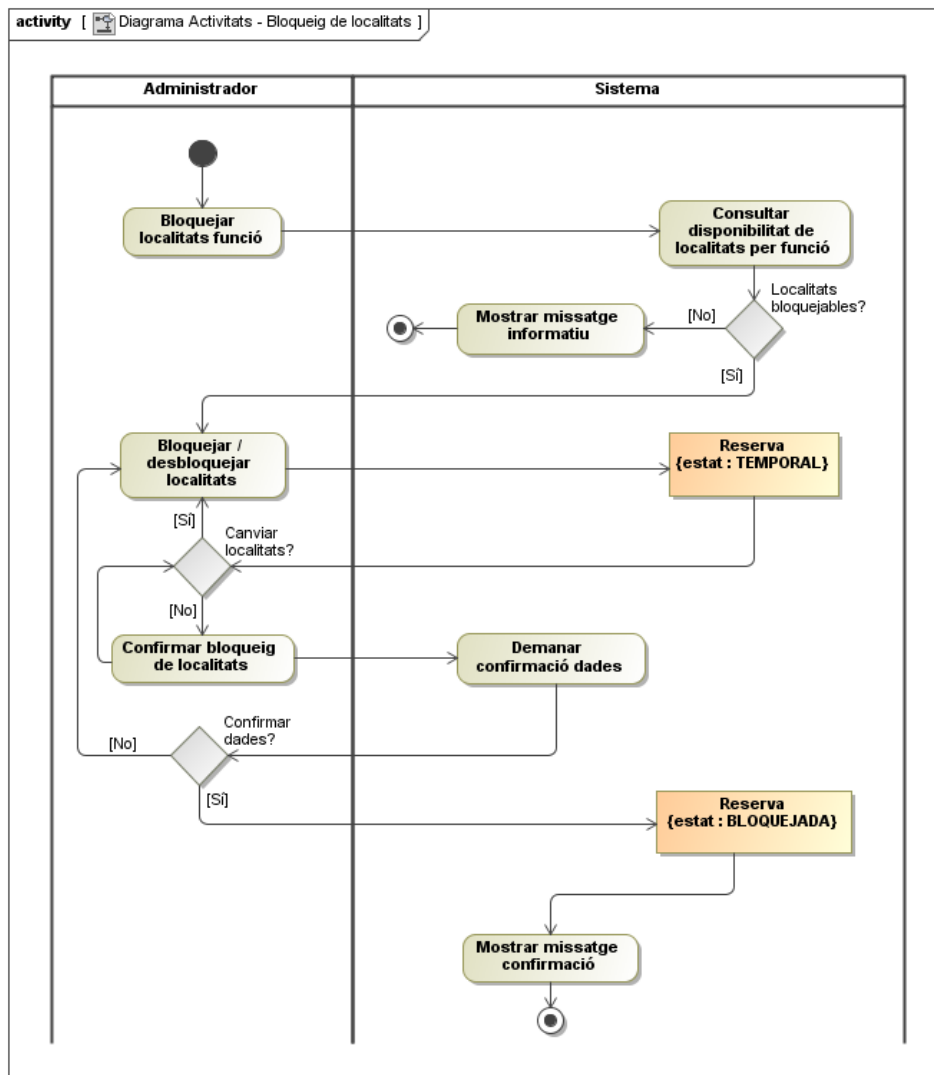
Diagrama d'estats d'una funció



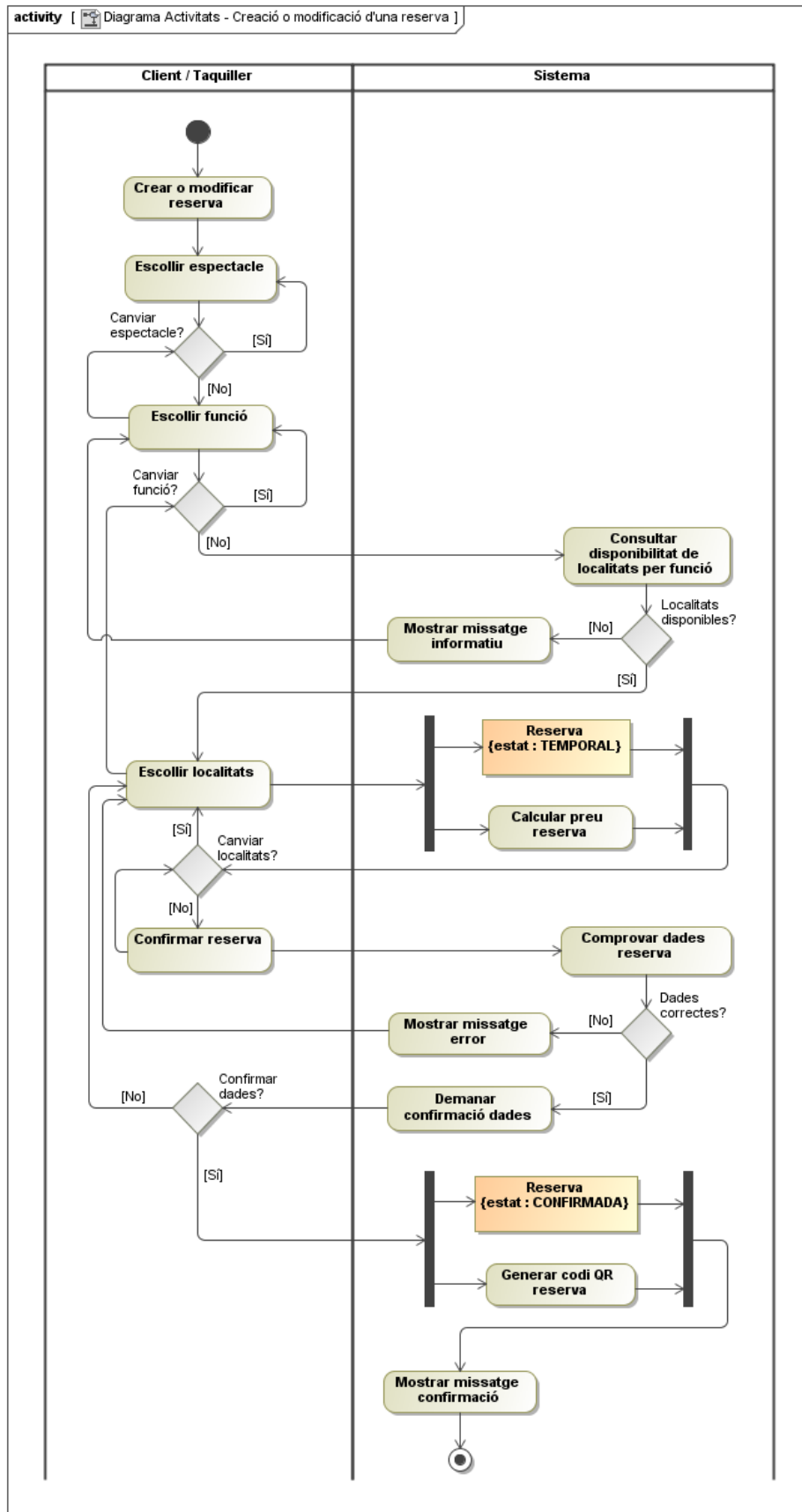
Diagrames d'activitat

El conjunt de funcionalitats més important de l'aplicació Espectaclia és la gestió de reserves de localitats. Aquestes reserves, tal i com hem vist en l'anterior apartat poden passar per diversos estats en funció de les activitats que realitzin els usuaris. A continuació es presenten els diagrames d'activitat de les 4 tasques que impliquen canvis d'estat en les reserves.

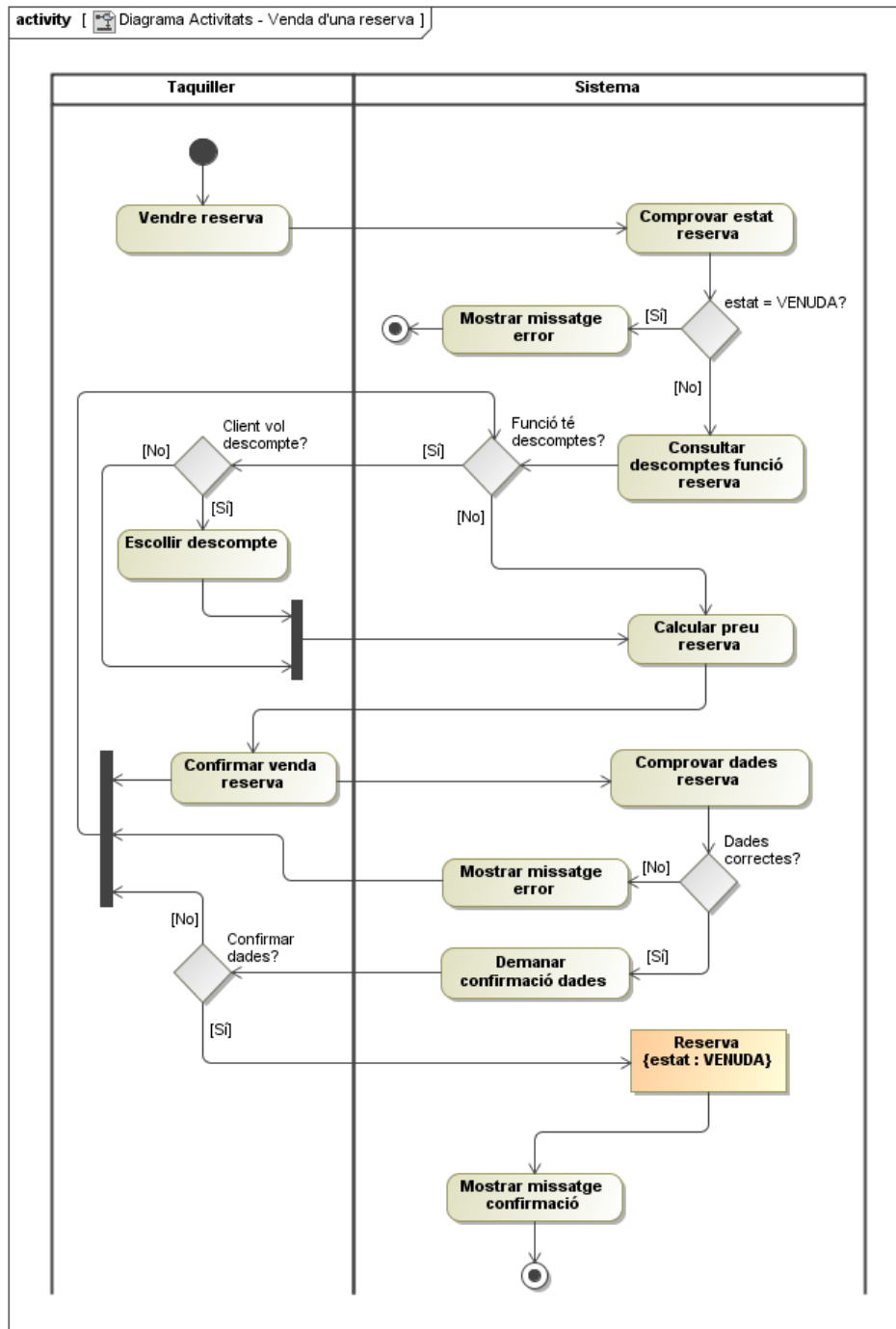
Bloqueig de localitats



Creació o modificació d'una reserva



Venda d'una reserva



Cancel·lació d'una reserva

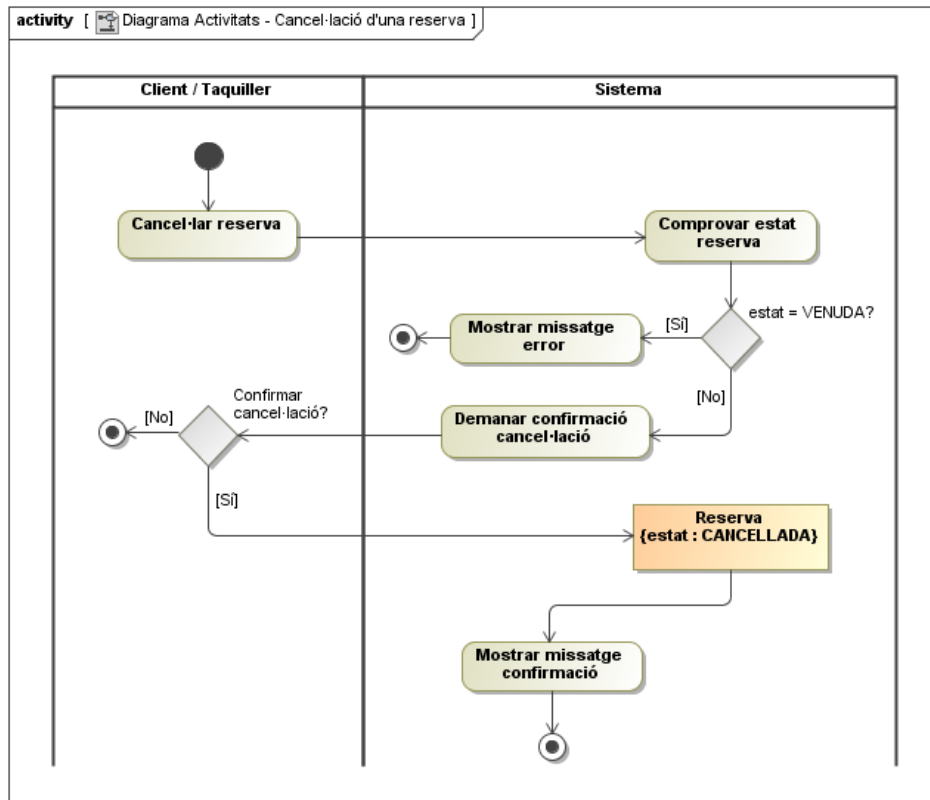
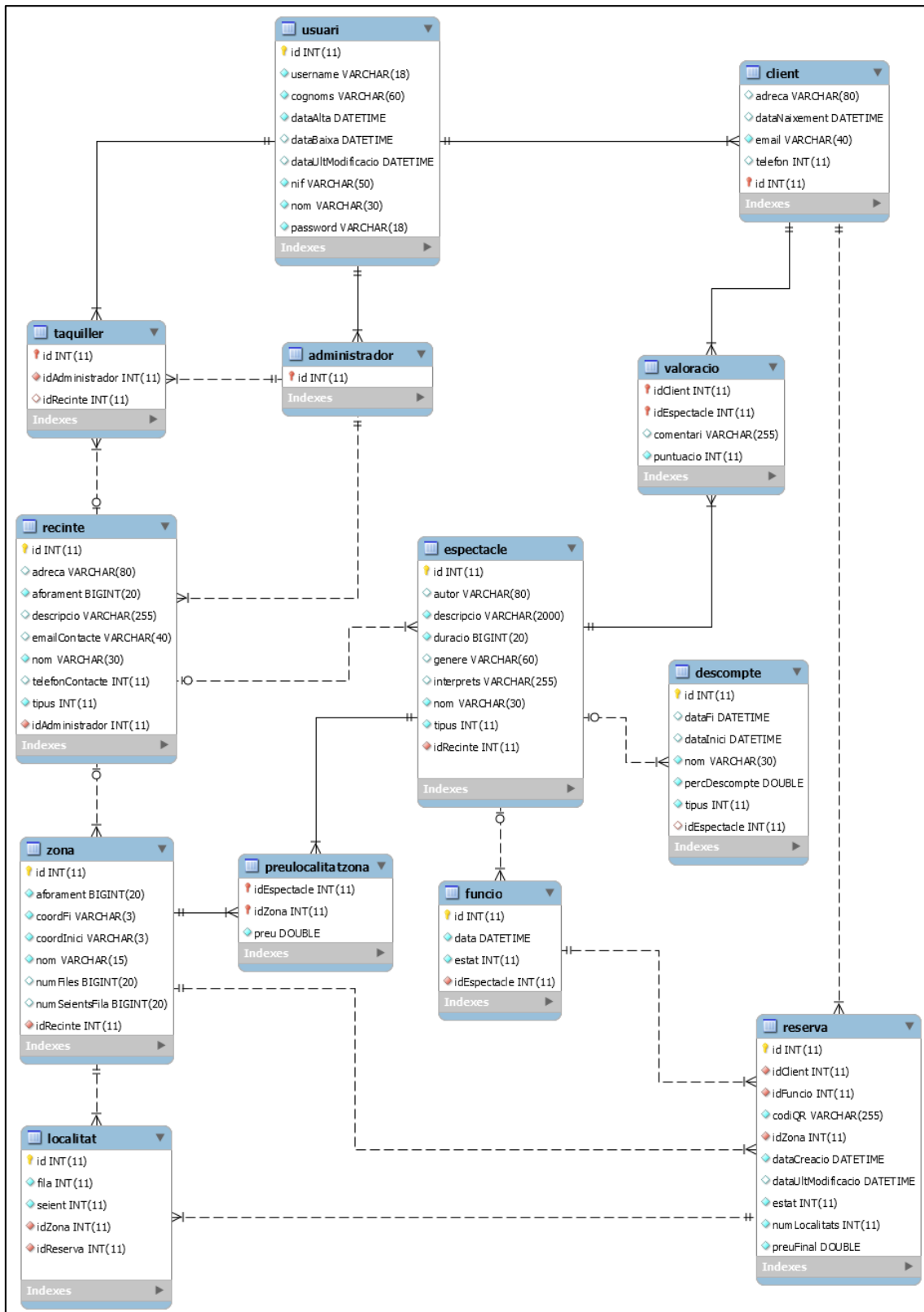


Diagrama ER de la base de dades

Tot seguit es mostra una proposta per a l'estructura de la base de dades de l'aplicació en forma de diagrama entitat-relació (o ER):



Com podem veure, les taules presents en aquest diagrama coincideixen amb les entitats que apareixen al diagrama de classes exposat anteriorment. En les taules només s'han afegit els camps pertinents per claus primàries i foranes.

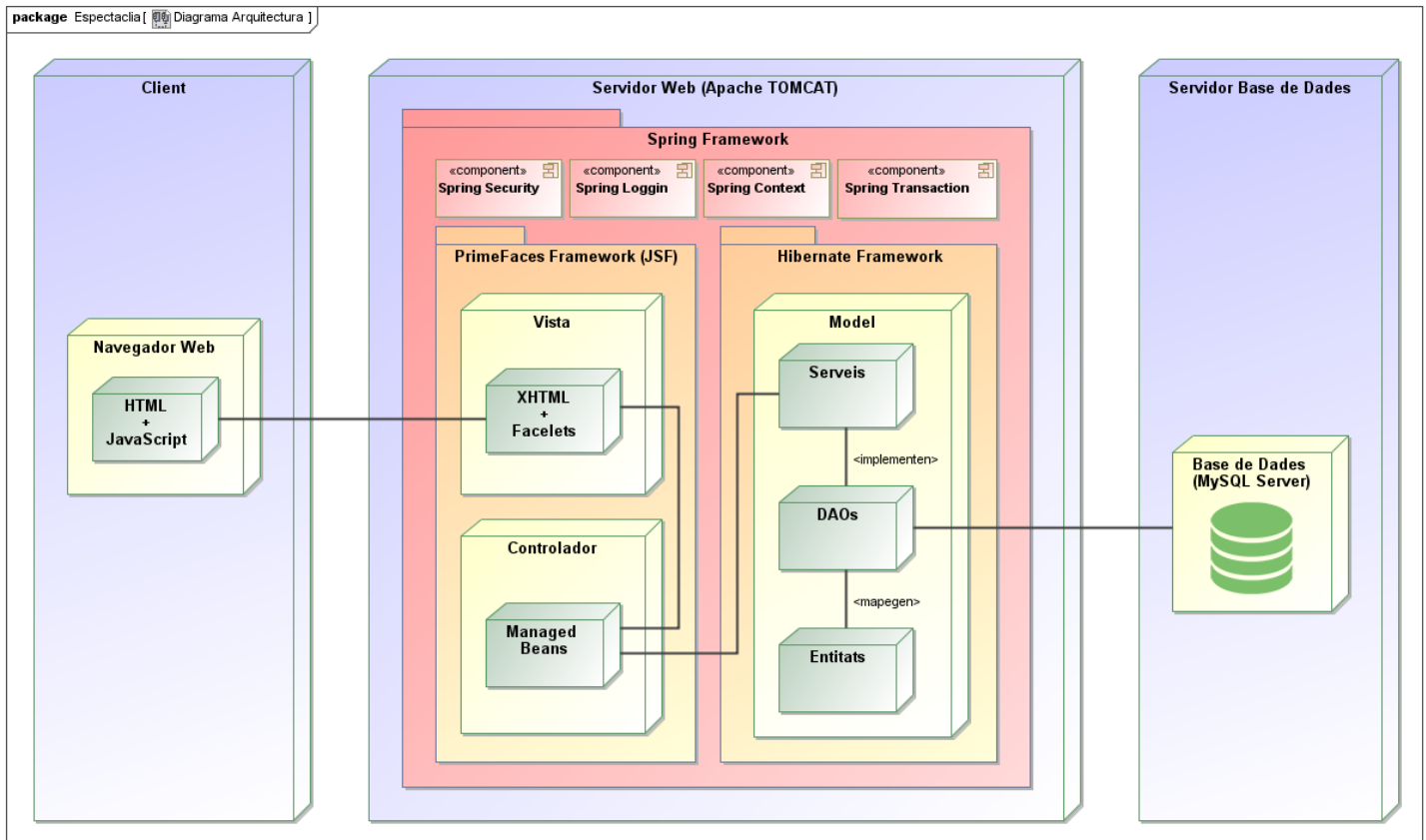
Els camps obligatoris, és a dir, els que no poden tenir mai valor "NULL" estan indicats amb un color blau més fort que els que no són obligatoris. A més, els camps que formen part de la clau primària estan indicats amb el símbol d'una clau, i els camps que són claus foranes en les seves o en altres taules estan indicats amb color vermell.

Es va considerar prescindir de la taula "Administrador", doncs no té altres atributs a banda de la seva clau primària, però degut a les importants relacions exclusives que manté aquest tipus d'usuari amb altres entitats es va decidir deixar-ho per a no complicar la lògica de la base de dades i al mateix temps la de l'aplicació.

Per últim, i de la mateixa manera que passa amb el diagrama de classes, les restriccions de negoci no es veuen reflexades en aquest model, i s'hauran de tenir en compte a l'hora d'implementar la lògica de negoci de l'aplicació (veure les indicacions sota el Diagrama de classes per a més informació sobre aquestes restriccions).

Arquitectura de l'aplicació

A continuació es mostra un diagrama complet amb l'arquitectura i tecnologies més importants que s'utilitzaran en el desenvolupament de l'aplicació:

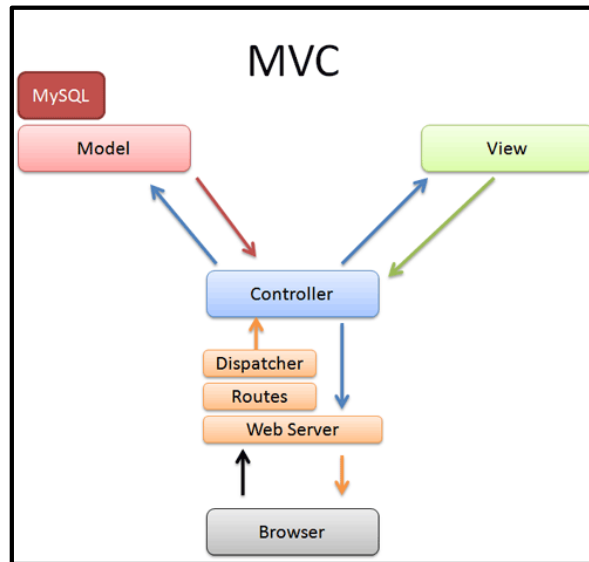


Arquitectura client - servidor i patró MVC

Tal i com es va definir al pla de treball, l'aplicació Espectaclia s'implementarà en forma d'**aplicació web** amb una arquitectura **client - servidor**. Aquest tipus d'arquitectura té una serie d'avantatges molt interessants:

- ✓ Permet als usuaris un accés ràpid, senzill i universal a través de la xarxa sense la necessitat d'instal·lar cap aplicació client, només fa falta un navegador web.
- ✓ Es disposa d'un client lleuger com és un navegador web, fet que rebaixa els requeriments computacionals per als futurs usuaris. No importa el hardware ni el sistema operatiu utilitzat, ni es depèn d'actualitzacions de software.
- ✓ Permet construir una aplicació distribuïda, en la que diversos clients es connecten a un mateix servidor que processa totes les peticions. Això permet tenir una aplicació centralitzada, amb una única base de dades global, fet que minimitza les possibles redundàncies o incongruències de dades en comparació a si es tinguessin diversos sistemes que s'haguessin de comunicar i sincronitzar de forma periòdica.

A més d'aquesta arquitectura més externa de tipus client - servidor, la part del **servidor** s'implementarà fent servir una **arquitectura de 3 capes** utilitzant el **patró de disseny MVC** (*Model - View - Controller*, o *Model - Vista - Controlador*). Aquest patró, molt utilitzat avui en dia en el món del desenvolupament de software, permet una clara separació de les classes i components d'una aplicació per tal de facilitar la seva ampliació i reusabilitat en 3 capes diferenciades segons la seva responsabilitat:



- **Model:** Conté tots aquells components que modelitzen les dades del sistema i la lògica de negoci. Rep peticions de dades del Controlador i respon amb les dades pertinents.
- **Vista:** S'encarrega de representar tota la interfície visual de comunicació i la presentació de les dades del sistema als usuaris. Rep peticions del Controlador i respon amb les interfícies d'usuari pertinents.
- **Controlador:** S'encarrega de rebre i respondre les peticions del client. Les interpreta per determinar quines crides ha de fer al Model i a la Vista per generar una nova interfície amb dades de les dues capes.

Aquests són alguns dels avantatges més importants de la utilització d'aquesta arquitectura de 3 capes mitjançant el patró MVC:

- ✓ Permet un manteniment separat de cadascuna de les capes, no tenint que modificar les altres capes de l'aplicació.
- ✓ Aquesta separació per capes dota l'aplicació d'una gran escalabilitat, permetent la divisió física o expansió dels seus components si així ho requereix.
- ✓ Com que les interfícies estan completament separades del model de dades i la lògica de negoci, es pot reutilitzar gran part de l'aplicació per part d'altres clients o terceres aplicacions.
- ✓ Permet que dissenyadors gràfics i programadors treballin de forma més independent, agilitzant la seva feina amb uns contractes de mínims.

Tecnologies i frameworks utilitzats

Pel que fa a tecnologies i frameworks que s'utilitzaran en la implementació d'aquesta arquitectura, a continuació es detalla quines es faran servir a cada part de l'aplicació:

- **Entorn**

Per desenvolupar l'aplicació es farà servir la **tecnologia J2EE** fonamentada en el llenguatge de programació orientada a objectes (POO) **Java**. Com a entorn de desenvolupament o **IDE** (*Integrated Development Environment*) s'utilitzarà **Sprint Tool Suite**, una versió modificada del conegut IDE Eclipse que integra el framework Spring (comentat més endavant).



- **Client**

A la banda del client es farà servir maquetació **HTML** juntament amb estils **CSS** per les interfícies. Els frameworks JSF i PrimeFaces, que es veuran tot seguit, s'encarregaran de generar tot aquest HTML i gran part del CSS de forma automàtica, a més de dotar les interfícies de dinamisme mitjançant codi **JavaScript** i **crides AJAX** autogenerats pels controladors de la capa Vista. Tot això en forma d'arxius **.XHTML**.

- **Servidor Web**

Al servidor s'utilitzarà un contenidor d'aplicacions **Apache Tomcat**, de codi obert i molt utilitzat, per executar l'aplicació i oferir-hi accés a través d'Internet. Pel que fa a l'aplicació en general, es farà servir el **framework Spring**, un framework transversal que ens permetrà gestionar de forma ràpida i centralitzada la implementació del patró MVC, amb aspectes tan importants com la gestió dels canals de comunicació entre els diferents components, el control de la seguretat (**Spring Security**), l'autenticació dels usuaris (**Spring Loggin**), la gestió del context de sessió (**Spring Context**) o la gestió de la transaccionabilitat de les connexions a la base de dades (**Spring Transaction**). A més, permet una integració fàcil d'altres frameworks que s'utilitzaran en l'aplicació i que s'expliquen més endavant.

Apache Tomcat



- **Capa Vista**

Pel que fa a la configuració de la capa de presentació finalment s'ha optat per aprofitar una eina relativament nova, molt atractiva i potent com es el **framework JSF** (*Java Server Faces*). Es tracta d'una tecnologia estàndar, i que forma part de l'especificació de J2EE, i que en la seva versió 2.0 utilitza **pàgines XHTML**, una versió millorada i extensible de la notació HTML, per a generar diferents interfícies anomenades **Facelets** de forma dinàmica en resposta a les peticions del client, i amb capacitat per a reaccionar a events d'usuari.

S'utilitzarà el **framework PrimeFaces** com a implementació del framework JSF, doncs resulta molt lleugera, fàcil d'implementar i molt vistosa a nivell visual. A més, es farà ús d'un tema basat en Bootstrap per a la seva presentació visual.



- **Capa Controlador**

En quant a la capa de controlador també es farà servir el **framework JSF** per implementar els anomenats controladors de vista en forma de **Managed Beans**, un tipus de entitat basat en POJOs (*Plain Old Java Object*) de Java anotats per tal d'injectar dependències amb altres components a través de JSF. Aquests controladors fan de connectors entre les peticions del client i els Serveis de la capa model per tal de modelar i modificar dinàmicament els continguts dels Facelets de la capa de presentació segons els continguts de les peticions que es reben.

- **Capa Model**

Per a la implementació del model es farà ús del **framework Hibernate**. Aquest framework permet el mapeig directe de les entitats de la nostra aplicació, així com els seus atributs i relacions, directament amb les taules de la base de dades. Això permet desvincular completament el codi de connexió a la base de dades de la lògica de negoci.

Per implementar la lògica de negoci pròpiament dita es farà ús del **patró DAO**. Aquest patró implica la creació d'un tipus de component específic, anomenat **DAO** (*Data Access Object*), que són interfícies on es declaren els mètodes de manipulació de les dades de la base de dades. Per tant, l'aplicació tindrà diferents **interfícies DAO**, una per a cada tipus d'entitat de l'aplicació, per a poder modificar de forma independent cadascuna d'elles. Aquestes interfícies tindran una o més implementacions en forma de **Serveis**, que seràn les classes encarregades d'atendre les peticions dels controladors, i d'aplicar la lògica de negoci que sigui necessària. En el cas de l'aplicació Espectaclia, aquests Serveis accedeixen a les dades de la base de dades a través de l'**Entity Manager d'Hibernate**, un tipus de servei proporcionat pel mateix framework.

Finalment, per implementar les **entitats** es faran servir **POJOs** (*Plain Old Java Objects*) de Java amb **annotacions** per tal de configurar els mapejos d'Hibernate a les diferents taules de la base de dades.



- **Persistència**: Per a implementar el servidor de la base de dades s'utilitzarà un servidor **MySQL Server**, doncs es tracta d'un motor de base de dades gratuït i ampliament utilitzat.



Modificacions de disseny i implementació

Aquestes són algunes de les decisions i canvis més importants que s'han fet durant la fase d'implementació de l'aplicació:

- Per estaviar temps i **centralitzar la gestió i instal·lació de totes les dependències i llibreries** s'ha optat per utilitzar **Maven**. Maven és una eina de gestió i construcció de projectes Java, a més integrada dins l'entorn Spring Tool Suite, que utilitza un arxiu amb format XML anomenat "*Project Object Model*" (POM) per a descriure el projecte a construir, les dependències d'altres mòduls i components externs, així com el seu ordre de construcció. També permet descarregar de forma dinàmica aquests components de diferents repositoris a través d'Internet.

En el cas d'Espectaclia, totes les llibreries dels diferents frameworks que s'utilitzen (Spring, JSF, PrimeFaces, Hibernate, etc.) les gestiona directament Maven.

maven

- En un principi, s'havia plantejat una primera pàgina d'inici comú per als 3 tipus d'usuaris de l'aplicació (Administrador, Taquiller i Client). Però durant la implementació de l'estructura del projecte i de la seguretat amb Spring Security no té gaire sentit que l'Administrador, que en principi no té accés a les funcionalitats per cursar reserves, pugui cercar espectacles com si iniciés una reserva. Ja pot llistar els espectacles dels seus recintes en altres vistes de l'aplicació i per tant no té sentit per a aquest tipus d'usuari. Per tant, s'ha decidit fer una **vista alternativa d'inici per a l'Administrador**, encara no implementada, que contingui un petit **quadre resum** de la quantitat de recintes, espectacles, funcions i reserves que té.
- En un principi, es va definir que quan un Administrador cancel·lava una funció d'un espectacle amb reserves, l'aplicació havia d'enviar un correu electrònic a tots aquells Clients que tinguessin un reserva pendent. Finalment **aquest comportament no s'ha implementat** per motius de temps i per no complicar excessivament la lògica de l'aplicació en aquesta primera fase. Obviament, sí que es cancel·laran totes aquestes reserves, i el Client ho podrà veure a la seva pantalla de gestió de reserves.
- Finalment s'ha decidit **prescindir definitivament de les funcionalitats per a la generació d'informes estadístics** per a l'Administrador, degut a la manca de temps per desenvolupar-les i a que ja des d'un principi es va plantejar descartar-les per aquest mateix motiu. Es preveu que en una futura fase d'ampliació de l'aplicatiu s'implementin aquestes funcionalitats.

Valoració econòmica

A continuació es mostra una taula amb el cost orientatiu que comportaria el desenvolupament d'aquest projecte en un entorn professional.

Com que tot el programari utilitzat és *Open Source*, no s'ha tingut en compte cap cost. només s'ha tingut en compte el cost del hardware que seria necessari per a tenir un servidor dedicat per a l'aplicació i la seva base de dades. S'ha considerat que el cost dels equips de desenvolupament seria menyspreable, suposant que l'empresa desenvolupadora ja els podria tenir anteriorment i no dedicar-los exclusivament a aquest projecte. El hardware valorat equivaldria al cost de tenir un servidor especialitzat dedicat a l'aplicació i a la seva base de dades.

Per últim, cal tenir en compte que tant les hores de dedicació de cadascun dels professionals com els imports indicats són aproximats:

Professional	Hores	Cost per hora	Cost total	Cost total (+ Impostos)
Analista funcional	70	25,00 €	1.750,00 €	2.117,50 €
Analista programador	100	22,00 €	2.200,00 €	2662,00 €
DBA	40	22,00 €	880,00 €	1.064,80 €
Dissenyador gràfic	40	20,00 €	800,00 €	968,00 €
Programador sènior	250	20,00 €	5.000,00 €	6.050,00 €
Servidor (hardware)	-	-	3.000,00 €	3.630,00 €
TOTAL	500	(promig) 21,80 €	13.630,00 €	16.492,30 €

Conclusions

Objectius assolits

Un cop finalitzades totes les fases del projecte i analitzant el producte obtingut es pot dir que s'han complert tant els objectius tècnics com els objectius funcionals que es van definir al començament d'aquest projecte.

S'ha aconseguit una aplicació funcional basada en tecnologia J2EE, en la que s'han integrat amb èxit diferents frameworks (Spring, JSF, PrimeFaces i Hibernate) i s'han implementat diferents patrons de programació (MCV i DAO). A més, l'aplicació s'executa sense problemes sobre un servidor Apache Tomcat 7 i accedeix correctament a una base de dades MySQL, tal i com es va projectar.

Pel que fa als objectius funcionals, s'han implementat tots els casos d'ús contemplats durant la fase d'anàlisi de requeriments, amb molt pocs canvis respecte a les especificacions originals i només obviant el cas d'ús d'Informes Estadístics, que prèviament ja s'havia estimat que es deixaria per a una posterior fase de millora del projecte. Per tant, l'aplicació realment permet que administradors de recintes i espectacles es comuniquin directament amb el client final, i que aquest pugui reservar i pagar a guixeta reserves de localitats mitjançant la figura del taquiller.

Conclusions personals

A nivell personal, l'elaboració d'aquest projecte de final de carrera m'ha permès aprofundir encara més els meus coneixements sobre J2EE que ja tenia de la carrera i del meu treball. A la feina sóc bàsicament programador i maquetador de *Front-End*, treballant sobretot amb frameworks de JavaScript, crides AJAX, llenguatge HTML i CSS, i per tant rarament faig implementacions de servidor. El fet d'implementar un projecte com aquest des de zero, i els frameworks i patrons que he utilitzat, m'han fet aprendre moltíssim sobre com s'estructura i configura un projecte. També m'ha fet aprendre la forma de programar amb un llenguatge orientat a objectes com Java i J2EE aprofitant conceptes com l'herència, el polimorfisme, l'encapsulament, la programació incremental i la reutilització de codi, aspectes que ja havia vist en matèries com Programació Orientada a l'Objecte, Enginyeria del programari, Estructura de la Informació i Tècniques de desenvolupament del programari però que adquireixen molta més rellevància desenvolupant un projecte real.

A nivell tècnic, he trobat especialment complicat aprendre a configurar i integrar els diferents frameworks. Per exemple, a Spring em va costar arribar a configurar correctament els contextes d'execució, l'escaneig de paquets i la seguretat a través d'Spring Security. D'altra banda, vaig tenir molts problemes per fer accessibles els Serveis de la capa Model als Managed Beans que controlen la capa Vista, doncs la injecció dels Serveis la controla Spring i la injecció dels Managed Beans la controla JSF. Al final, vaig trobar un codi que permetia injectar manualment els Serveis dins els Managed Beans.

També vaig trobar dificultats a l'hora de modificar o eliminar entitats mitjançant l'Entity Manager de Hibernate. Hibernate està alhora basat en un framework intern de

Java anotmenat *Java Persistence API* (JPA) que obliga a mantenir equiparat l'estat de la persistència amb l'estat de les instàncies del programa. Per tant, cal modificar o eliminar les instàncies d'un objecte a tota l'aplicació abans de persistir-ho amb Hibernate, i de vegades no resultava fàcil trobar totes aquestes instàncies.

Pel que fa a les fases d'anàlisi i disseny, al principi em va semblar molt difícil haver de documentar i fer diagrames d'una aplicació que encara no havia fet i que només tenia al cap, però conforme es va desenvolupant i implementant el projecte t'adones que tota aquesta feina prèvia no ha estat en va, i que realment ajuda a que la implementació de l'aplicació sigui molt més ràpida i aproximada a la funcionalitat real que s'espera que tingui l'aplicació. De fet, no he hagut de fer molts canvis en la implementació respecte a la documentació original.

En definitiva, aquest TFC en l'àrea de J2EE m'ha permès aprendre com crear una aplicació des de la primera especificació de funcionalitats fins al seu testeig emprant tecnologies actuals basades en un llenguatge tan estès al món del desenvolupament com és Java i l'entorn J2EE, que un cop s'aprén a configurar-les i utilitzar-les simplifiquen la feina i estalvien molt de temps al programador.

Glossari

Administrador: Tipus d'usuari de l'aplicació, l'únic amb permisos per gestionar taquillers, recintes, espectacles, zones, localitats, preus i descomptes. També és l'únic amb capacitat per bloquejar localitats. Aquest tipus d'usuari no es pot gestionar a través de l'aplicació. L'administrador de la base de dades l'ha de crear manualment.

Client: Tipus d'usuari de l'aplicació, amb capacitat per cercar espectacles i fer reserves de localitats sobre les seves funcions. També pot fer, si ho desitja, una valoració per cada espectacle sobre el que tingui com a mínim una reserva i estigui venuda. És l'únic usuari capaç d'autogestionar-se el seu propi usuari.

Descompte: Disminució percentual del preu nominatiu per localitat d'un espectacle, d'aplicació opcional requerida pel client, sempre i quan la funció sobre la que es vol aplicar o el client que el sol·licita compleixin les condicions definides pel tipus de descompte i/o per les seves dates d'aplicació.

Espectacle: Esdeveniment cultural caracteritzat per una serie de propietats que el defineixen, que es realitza dins un dels recintes creats per un administrador. L'administrador pot definir-ne funcions programades, preus per zona del recinte i descomptes aplicables.

Funció: Data i hora a la què es produeix un dels espectacles definits per un administrador. Un espectacle pot tenir diverses funcions, però no en la mateixa data i hora. Poden ser cancel·lades si l'administrador així ho desitja.

Localitat: Seient o plaça pertanyent a una de les zones d'un recinte destinada a ser reservada per a un espectacle i funció determinades per un client (o taquiller en nom d'un client), o bé bloquejada per un administrador. Les localitats poden ser numerades, amb una posició definida dins la seva zona, o bé no numerades.

Recinte: Espai gestionat per un administrador caracteritzat per una series de propietats on esdevenen els espectacles. Pot estar format per una o més zones amb localitats numerades o sense numerar.

Reserva: Conjunt de localitats que un client (o taquiller en nom d'un client) s'assigna a títol propi per a un espectacle, funció i zona del recinte determinades per tal que ningú més les pugui ocupar per a aquesta mateixa combinació. Per tal de fer-la efectiva, el dia de la funció el client haurà d'abonar el seu import al taquiller per guixeta.

Taquiller: Tipus d'usuari de l'aplicació, capaç de reserar localitats de la mateixa manera que un client, però sempre en relació a un client existent en l'aplicació. A més, és l'únic usuari amb capacitat per vendre entrades. Aquest tipus d'usuari és gestionat exclusivament per l'administrador que l'ha creat.

Usuari: Qualsevol tipus de persona que es registri i utilitzi l'aplicació. Pot ser de 3 tipus: Client, Taquiller o Administrador. Els clients

Valoració: Opinió opcional que pot enviar un client respect un espectacle d'una reserva seva una vegada que ja hagi estat venuda i hagi acabat la funció. Aquesta opinió consta d'una puntuació obligatoria entre 5 possibilitats i d'un comentari opcional.

Zona: Espai dins d'un recinte amb una posició determinada que delimita un conjunt de localitats que comparteixen unes mateixes característiques de localització i preu. Una zona pot tenir localitats numerades o bé no numerades, però mai dels dos tipus.

Bibliografia consultada

- PRADEL MIQUEL, Jordi; RAYA MARTOS, Jose. *“Enginyeria del programari”*. UOC. (Materials)
- BRÍNQUEZ JIMÉNEZ, Jordi; PIERA I EROLES, Miquel Àngel; RAMOS GONZÁLEZ, Juan José. *“Programació Orientada a l'Objecte”*. UOC. (Materials)
- XHAFÀ, Fatos. *“Tècniques de desenvolupament de programari”*. UOC. (Materials)
- LARMAN Craig. *“UML y Patrones”*. Prentice Hall.
- KAMAL, Az-addyne. *“PrimeFaces 5 & Sprint 4”*. (Tutorial):
<http://azaddyne.blogspot.com.es/2014/10/tutorial-how-to-integrate-primefaces-5.html>
- MARCO NG. *“PrimeFaces + JSF2 + Spring Security + Spring + Hibernate”*. (Tutorial):
<http://marco-ng.blogspot.com.es/2014/02/primefaces-jsf2-spring-security-spring.html>
- Java EE 7 Specification APIs:
<https://docs.oracle.com/javaee/7/api/>
- Spring Guides:
<https://spring.io/guides>
- Hibernate ORM Documentation (4.2):
<http://hibernate.org/orm/documentation/4.2/>
- PrimeFaces User Guide (5.3):
http://www.primefaces.org/docs/guide/primefaces_user_guide_5_3.pdf
- PrimeFaces Showcase:
<http://www.primefaces.org/showcase/>
- Bootstrap Components:
<http://getbootstrap.com/components/>

Annex 1: Manual d'instal·lació

Requeriments de software

A continuació es detallen els diferents components de software necessaris per a configurar l'entorn d'execució de l'aplicació. Tots aquests components són *Open Source*, és a dir, són de lliure disposició, utilització i ampliació per part de qualsevol.

Java JDK (v 8u45)

El **Java Development Kit** d'Oracle és el component de software transversal i indispensable de la nostra aplicació, doncs ens permetrà desenvolupar i executar el seu codi en el llenguatge de programació orientat a objectes Java en una màquina virtual de Java (JVM). És el primer component a instal·lar, doncs no només és necessari per a implementar el codi, sinò que també ho és per tal que l'entorn de desenvolupament (IDE) funcioni correctament.

En el nostre cas, al moment de la seva instal·lació vam triar la **versió 8u45**, que permet utilitzar l'última especificació de Java disponible: la 1.8. Tot i així, i per prevenir possibles incompatibilitats d'algunes llibreries amb l'última versió, l'aplicació Espectaclia s'ha configurat sota l'especificació de **Java 1.7**.

Spring Tool Suite (v 3.6.4)

Per al desenvolupament de l'aplicació s'ha utilitzat l'IDE **Spring Tool Suite** (STS), en la seva **versió 3.6.4**. Aquest IDE és una reimplementació del popular IDE Eclipse, ampliat amb diferents plugins i components del conegut framework Spring, el qual duu incorporat per defecte i sobre el què s'arquitectura gran part de l'aplicació. Cal instal·lar aquest entorn i crear un *Workspace* en el què poder importar i executar el codi font.

Apache Tomcat (v 7.0.62)

Per tal de poder executar l'aplicació dins l'entorn STS cal un **servidor d'aplicacions**. Tot i que Spring Tool Suite incorpora el seu propi servidor, hem optat per instal·lar el conegut i estès servidor d'Apache Tomcat, en la seva **versió 7.0.62**. Existeix una versió més actual, la 8, però s'ha optat per la versió 7 per motius de compatibilitat.

MySQL Server (v 5.6.10)

Paral·lelament als components anteriors, cal instal·lar el **servidor de bases de dades MySQL Server** que manegi la persistència de dades de l'aplicació. En el moment de la configuració de l'entorn, es va instal·lar l'última versió, la **5.6.10**.

En les darreres passes de la seva instal·lació sens demanaran algunes dades de configuració importants, com el port de connexió a la base de dades i la contrasenya de l'usuari *ROOT* de MySQL. Aquestes dades caldrà apuntar-les o enrecordar-les, doncs caldrà indicar-les a un arxiu de configuració de l'aplicació per tal que aquesta tingui accés a la base de dades.

Instal·lació de l'aplicació

Un cop instal·lats els diferents components de software esmentats en l'apartat anterior, ens dispoem a instal·lar l'aplicació a partir del paquet d'arxius de codi font subministrat. A continuació es detallen els passos a seguir per a la seva correcta implantació al nostre entorn. Aquest procediment s'ha realitzat en un sistema operatiu Windows 10, en la seva versió de 64 bits.

1. **Copiar el projecte al Workspace**: Sense obrir encara l'entorn, copiem el paquet de codi font directament a la carpeta del nostre ordinador que hem definit com a *Workspace* durant la instal·lació de l'Spring Tool Suite. Més concretament, cal copiar la carpeta amb el nom "*Espectaclia*" directament en aquest directori.
2. **Crear el servidor Tomcat a l'entorn**: Obrim l'Spring Tool Suite, i utilitzem els menús superiors per anar fins a "*File > New > Other...*". Busquem la carpeta "*Server*" i sel·leccionem "*Server*". En la següent finestra ens hauria de sortir una carpeta amb el nom "*Apache*", la obrim i d'aquí triem l'opció "*Tomcat v7.0 Server*". Si no ens aparegués aquesta opció, fem click al link "*Add...*" de més baix, al costat del selector "*Server runtime environment*". A la nova pantalla utilitzem "*Browse...*" per a cercar la carpeta de la instal·lació del servidor Apache Tomcat 7 al nostre ordinador, la sel·leccionem i al camp "*Name*" hauria de posar "*Apache Tomcat v7.0*". Finalitzem i ara ja podem triar aquest servidor com a "*Server runtime environment*". Finalitzem, i ja tindrem creat el servidor dins el nostre entorn.
3. **Importar el projecte de l'aplicació a l'entorn**: A l'Spring Tool Suite, utilitzem els menús superiors per anar fins a "*File > Import*". Aquí, busquem la carpeta "*Maven*" i sel·leccionem l'opció "*Existing Maven Projects*". A continuació, utilitzem el botó "*Browse...*" per triar la carpeta del nostre Workspace (si es que no surt ja presel·leccionada), i en la llista de sota ens hauria de sortir l'aplicació com a "*Espectaclia/pom.xml*". Marquem el check al costat i finalitzem l'assistent. Ara ja tindrem importat el projecte Espectaclia al nostre entorn.
4. **Descarregar les llibreries i dependències amb Maven**: A l'Spring Tool Suite, fem click dret sobre la carpeta del projecte Espectaclia que acabem d'importar, i al menú contextual anem fins a "*Maven > Update project...*". A la pantalla que apareix, marquem el check al costat d'Espectaclia, i comprovem que estiguin marcats els 3 últims checks, sobretot l'opció "*Update project configuration from pom.xml*". Fem click a OK, i l'entorn descarregarà i instal·larà automàticament totes les dependències indicades a l'arxiu "*pom.xml*" de la nostra aplicació.

5. **Configurar la connexió de l'aplicació amb la BD:** Cal revisar que els paràmetres de connexió amb la base de dades coincideixin amb els que hem indicat a l'hora d'instal·lar el MySQL Server. Per fer això, a l'Spring Tool Suite obrim el projecte de l'aplicació Espectaclia, i anem fins a l'arxiu "Espectaclia > src > main > resources > META-INF > persistence.xml". L'obrim, i comprovem el valor dels següents *properties*:
 - ***hibernate.connection.url***: ha de coincidir la url i el port amb els indicats al MySQL Server. El nom de la base de dades el deixem com està ("*espectaclia*").
 - ***hibernate.connection.password***: ha de coincidir amb la contrasenya indicada per l'usuari ROOT al MySQL Server. Obviament, al *property* immediatament superior ha de figurar el valor "*root*".

6. **Configurar la base de dades MySQL:** Cal crear la base de dades per a la nostra aplicació. Per a aconseguir-ho ho podem fer de dues formes:
 - a. Accedim al MySQL Server, ja sigui a través d'una consola amb el comand "*mysql*" o bé a través d'aplicacions com MySQL Workbench. Allà, executem l'arxiu "**Espectaclia DDL.sql**" adjunt a l'arrel del paquet de codi font subministrat, o bé copiem i executem manualment les comandes de l'annex "*Sentències DDL de la base de dades*" d'aquest mateix document. Un cop executades, ens hauran creat la base de dades "*espectaclia*" així com totes les taules de l'aplicació, amb tots els seus camps i claus primàries i forànies, a més d'algunes dades inicials necessàries per al funcionament de l'aplicació.
 - b. La més elegant: a l'arxiu indicat al pas anterior, modifiquem el *property* "***hibernate.hbm2ddl.auto***" amb el valor "*create*". Així, si hem configurat correctament la connexió a la base de dades, quan engeguem l'aplicació automàticament crearà tota la base de dades en base als mapejos d'entitats inclosos al codi font.

7. **Executar l'aplicació al servidor Tomcat:** A l'Spring Tool Suite, anem a la pestanya "*Servers*" situada a la part inferior del cos central de l'entorn. Allà hauria de figurar el servidor "*Tomcat v7.0 Server at localhost*" creat anteriorment (el nom pot variar si l'hem canviat a l'assistent). Fem click dret, i al menú contextual triem "*Add and Remove...*". A la pantalla que s'obre passem la nostra aplicació Espectaclia al costat dret del selector, i finalitzem. L'aplicació es publicarà automàticament al servidor Tomcat. Ara només caldrà iniciar el servidor fent click a la icona de play de color verd a la part dreta, i l'aplicació quedarà accessible.

Si tot ha anat bé, després de seguir tots aquests passos podrem accedir a l'aplicació web Espectaclia a través de qualsevol navegador web amb l'adreça <http://localhost:8080/Espectaclia/>.

Annex 2: Sentències DDL de la base de dades

DDL de creació

A continuació es mostren les sentències DDL necessàries per a generar la base de dades de l'aplicació Espectaclia:

```

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

-----
-- Schema mydb
-----
-- Schema espectaclia
-----

-----
-- Schema espectaclia
-----
CREATE SCHEMA IF NOT EXISTS `espectaclia` DEFAULT CHARACTER SET latin1 COLLATE
latin1_spanish_ci ;
USE `espectaclia` ;

-----
-- Table `espectaclia`.`usuari`
-----
CREATE TABLE IF NOT EXISTS `espectaclia`.`usuari` (
  `id` INT(11) NOT NULL AUTO_INCREMENT COMMENT '',
  `username` VARCHAR(18) CHARACTER SET 'latin1' COLLATE 'latin1_spanish_ci'
NOT NULL COMMENT '',
  `cognoms` VARCHAR(60) CHARACTER SET 'latin1' COLLATE 'latin1_spanish_ci' NOT
NULL COMMENT '',
  `dataAlta` DATETIME NOT NULL COMMENT '',
  `dataBaixa` DATETIME NULL DEFAULT NULL COMMENT '',
  `dataUltModificacio` DATETIME NULL DEFAULT NULL COMMENT '',
  `nif` VARCHAR(50) CHARACTER SET 'latin1' COLLATE 'latin1_spanish_ci' NOT
NULL COMMENT '',
  `nom` VARCHAR(30) CHARACTER SET 'latin1' COLLATE 'latin1_spanish_ci' NOT
NULL COMMENT '',
  `password` VARCHAR(18) CHARACTER SET 'latin1' COLLATE 'latin1_spanish_ci'
NOT NULL COMMENT '',
  PRIMARY KEY (`id`) COMMENT '',
  UNIQUE INDEX `username_UNIQUE` (`username` ASC) COMMENT '')
ENGINE = InnoDB
AUTO_INCREMENT = 10000
DEFAULT CHARACTER SET = latin1
COLLATE = latin1_spanish_ci;

-----
-- Table `espectaclia`.`administrador`
-----
CREATE TABLE IF NOT EXISTS `espectaclia`.`administrador` (
  `id` INT(11) NOT NULL COMMENT '',
  PRIMARY KEY (`id`) COMMENT '',
  INDEX `FK_bqmdyfc2fjonj08eip07pcn0e` (`id` ASC) COMMENT '',
  CONSTRAINT `FK_bqmdyfc2fjonj08eip07pcn0e`
FOREIGN KEY (`id`)
REFERENCES `espectaclia`.`usuari` (`id`))

```

```

ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1
COLLATE = latin1_spanish_ci;

-----
-- Table `espectaclia`.`client`
-----
CREATE TABLE IF NOT EXISTS `espectaclia`.`client` (
  `adreca` VARCHAR(80) CHARACTER SET 'latin1' COLLATE 'latin1_spanish_ci' NULL
  DEFAULT NULL COMMENT '',
  `dataNaixement` DATETIME NULL DEFAULT NULL COMMENT '',
  `email` VARCHAR(40) CHARACTER SET 'latin1' COLLATE 'latin1_spanish_ci' NOT
  NULL COMMENT '',
  `telefon` INT(11) NULL DEFAULT NULL COMMENT '',
  `id` INT(11) NOT NULL COMMENT '',
  PRIMARY KEY (`id`) COMMENT '',
  UNIQUE INDEX `email_UNIQUE` (`email` ASC) COMMENT '',
  INDEX `FK_ry5y27p5iwfa43hripgl2nly6` (`id` ASC) COMMENT '',
  CONSTRAINT `FK_ry5y27p5iwfa43hripgl2nly6`
  FOREIGN KEY (`id`)
  REFERENCES `espectaclia`.`usuari` (`id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1
COLLATE = latin1_spanish_ci;

-----
-- Table `espectaclia`.`recinte`
-----
CREATE TABLE IF NOT EXISTS `espectaclia`.`recinte` (
  `id` INT(11) NOT NULL AUTO_INCREMENT COMMENT '',
  `adreca` VARCHAR(80) CHARACTER SET 'latin1' COLLATE 'latin1_spanish_ci' NULL
  DEFAULT NULL COMMENT '',
  `aforament` BIGINT(20) NOT NULL COMMENT '',
  `descripcio` VARCHAR(255) CHARACTER SET 'latin1' COLLATE 'latin1_spanish_ci'
  NULL DEFAULT NULL COMMENT '',
  `emailContacte` VARCHAR(40) CHARACTER SET 'latin1' COLLATE
  'latin1_spanish_ci' NULL DEFAULT NULL COMMENT '',
  `nom` VARCHAR(30) CHARACTER SET 'latin1' COLLATE 'latin1_spanish_ci' NOT
  NULL COMMENT '',
  `telefonContacte` INT(11) NULL DEFAULT NULL COMMENT '',
  `tipus` INT(11) NOT NULL COMMENT '',
  `idAdministrador` INT(11) NOT NULL COMMENT '',
  PRIMARY KEY (`id`) COMMENT '',
  INDEX `FK_t69b5epap5ea7xjqdjttaa487b` (`idAdministrador` ASC) COMMENT '',
  CONSTRAINT `FK_t69b5epap5ea7xjqdjttaa487b`
  FOREIGN KEY (`idAdministrador`)
  REFERENCES `espectaclia`.`administrador` (`id`))
ENGINE = InnoDB
AUTO_INCREMENT = 4
DEFAULT CHARACTER SET = latin1
COLLATE = latin1_spanish_ci;

-----
-- Table `espectaclia`.`espectacle`
-----
CREATE TABLE IF NOT EXISTS `espectaclia`.`espectacle` (
  `id` INT(11) NOT NULL AUTO_INCREMENT COMMENT '',
  `autor` VARCHAR(80) CHARACTER SET 'latin1' COLLATE
  'latin1_spanish_ci' NULL DEFAULT NULL COMMENT '',

```

```

`descripcio` VARCHAR(2000) CHARACTER SET 'latin1' COLLATE
'latin1_spanish_ci' NOT NULL COMMENT '',
`duracio` BIGINT(20) NOT NULL COMMENT '',
`genere` VARCHAR(60) CHARACTER SET 'latin1' COLLATE
'latin1_spanish_ci' NULL DEFAULT NULL COMMENT '',
`interprets` VARCHAR(255) CHARACTER SET 'latin1' COLLATE
'latin1_spanish_ci' NULL DEFAULT NULL COMMENT '',
`nom` VARCHAR(30) CHARACTER SET 'latin1' COLLATE
'latin1_spanish_ci' NOT NULL COMMENT '',
`tipus` INT(11) NOT NULL COMMENT '',
`idRecinte` INT(11) NOT NULL COMMENT '',
PRIMARY KEY (`id`) COMMENT '',
INDEX `FK_hgckein2ees4sjuwoaeioycf` (`idRecinte` ASC) COMMENT
'',
CONSTRAINT `FK_hgckein2ees4sjuwoaeioycf`
FOREIGN KEY (`idRecinte`)
REFERENCES `espectaclia`.`recinte` (`id`))
ENGINE = InnoDB
AUTO_INCREMENT = 5
DEFAULT CHARACTER SET = latin1
COLLATE = latin1_spanish_ci;

```

```

-----
-- Table `espectaclia`.`descompte`
-----

```

```

CREATE TABLE IF NOT EXISTS `espectaclia`.`descompte` (
  `id` INT(11) NOT NULL AUTO_INCREMENT COMMENT '',
  `dataFi` DATETIME NULL DEFAULT NULL COMMENT '',
  `dataInici` DATETIME NULL DEFAULT NULL COMMENT '',
  `nom` VARCHAR(30) CHARACTER SET 'latin1' COLLATE
'latin1_spanish_ci' NOT NULL COMMENT '',
  `percDescompte` DOUBLE NOT NULL COMMENT '',
  `tipus` INT(11) NOT NULL COMMENT '',
  `idEspectacle` INT(11) NULL DEFAULT NULL COMMENT '',
  PRIMARY KEY (`id`) COMMENT '',
  INDEX `FK_eaxxuinmabarkvb0fx07pulbo` (`idEspectacle` ASC)
COMMENT '',
  CONSTRAINT `FK_eaxxuinmabarkvb0fx07pulbo`
FOREIGN KEY (`idEspectacle`)
REFERENCES `espectaclia`.`espectacle` (`id`))
ENGINE = InnoDB
AUTO_INCREMENT = 8
DEFAULT CHARACTER SET = latin1
COLLATE = latin1_spanish_ci;

```

```

-----
-- Table `espectaclia`.`funcio`
-----

```

```

CREATE TABLE IF NOT EXISTS `espectaclia`.`funcio` (
  `id` INT(11) NOT NULL AUTO_INCREMENT COMMENT '',
  `data` DATETIME NOT NULL COMMENT '',

```

```

`estat` INT(11) NOT NULL COMMENT '',
`idEspectacle` INT(11) NOT NULL COMMENT '',
PRIMARY KEY (`id`) COMMENT '',
INDEX `FK_178g6ept7px4l82wc7bvfp1nv` (`idEspectacle` ASC)
COMMENT '',
CONSTRAINT `FK_178g6ept7px4l82wc7bvfp1nv`
FOREIGN KEY (`idEspectacle`)
REFERENCES `espectaclia`.`espectacle` (`id`))
ENGINE = InnoDB
AUTO_INCREMENT = 8
DEFAULT CHARACTER SET = latin1
COLLATE = latin1_spanish_ci;

```

```

-----
-- Table `espectaclia`.`zona`
-----
CREATE TABLE IF NOT EXISTS `espectaclia`.`zona` (
  `id` INT(11) NOT NULL AUTO_INCREMENT COMMENT '',
  `aforament` BIGINT(20) NOT NULL COMMENT '',
  `coordFi` VARCHAR(3) CHARACTER SET 'latin1' COLLATE
'latin1_spanish_ci' NOT NULL COMMENT '',
  `coordInici` VARCHAR(3) CHARACTER SET 'latin1' COLLATE
'latin1_spanish_ci' NOT NULL COMMENT '',
  `nom` VARCHAR(15) CHARACTER SET 'latin1' COLLATE
'latin1_spanish_ci' NOT NULL COMMENT '',
  `numFiles` BIGINT(20) NULL DEFAULT NULL COMMENT '',
  `numSeientsFila` BIGINT(20) NULL DEFAULT NULL COMMENT '',
  `idRecinte` INT(11) NOT NULL COMMENT '',
  PRIMARY KEY (`id`) COMMENT '',
  INDEX `FK_9wfy3fm0dh2t72jnawgn3yfm9` (`idRecinte` ASC)
COMMENT '',
CONSTRAINT `FK_9wfy3fm0dh2t72jnawgn3yfm9`
FOREIGN KEY (`idRecinte`)
REFERENCES `espectaclia`.`recinte` (`id`))
ENGINE = InnoDB
AUTO_INCREMENT = 13
DEFAULT CHARACTER SET = latin1
COLLATE = latin1_spanish_ci;

```

```

-----
-- Table `espectaclia`.`reserva`
-----
CREATE TABLE IF NOT EXISTS `espectaclia`.`reserva` (
  `idClient` INT(11) NOT NULL COMMENT '',
  `idFuncio` INT(11) NOT NULL COMMENT '',
  `codiQR` VARCHAR(255) CHARACTER SET 'latin1' COLLATE
'latin1_spanish_ci' NOT NULL COMMENT '',
  `dataCreacio` DATETIME NOT NULL COMMENT '',
  `dataUltModificacio` DATETIME NULL DEFAULT NULL COMMENT '',
  `estat` INT(11) NOT NULL COMMENT '',
  `numLocalitats` INT(11) NOT NULL COMMENT '',

```



```

`preuFinal` DOUBLE NOT NULL COMMENT '',
`idZona` INT(11) NOT NULL COMMENT '',
`id` INT(11) NOT NULL AUTO_INCREMENT COMMENT '',
PRIMARY KEY (`id`) COMMENT '',
INDEX `FK_n61uy96f3ggv116okg0si5074` (`idClient` ASC) COMMENT
'',
INDEX `FK_c38710s9117lm2bojx49yr7q7` (`idFuncio` ASC) COMMENT
'',
INDEX `FK_ac5dd01ibtw6kn2ak5846bgdo` (`idZona` ASC) COMMENT
'',
CONSTRAINT `FK_ac5dd01ibtw6kn2ak5846bgdo`
  FOREIGN KEY (`idZona`)
  REFERENCES `espectaclia`.`zona` (`id`),
CONSTRAINT `FK_c38710s9117lm2bojx49yr7q7`
  FOREIGN KEY (`idFuncio`)
  REFERENCES `espectaclia`.`funcio` (`id`),
CONSTRAINT `FK_n61uy96f3ggv116okg0si5074`
  FOREIGN KEY (`idClient`)
  REFERENCES `espectaclia`.`client` (`id`))
ENGINE = InnoDB
AUTO_INCREMENT = 103
DEFAULT CHARACTER SET = latin1
COLLATE = latin1_spanish_ci;

```

```

-----
-- Table `espectaclia`.`localitat`
-----
CREATE TABLE IF NOT EXISTS `espectaclia`.`localitat` (
  `id` INT(11) NOT NULL AUTO_INCREMENT COMMENT '',
  `fila` INT(11) NOT NULL COMMENT '',
  `seient` INT(11) NOT NULL COMMENT '',
  `idZona` INT(11) NOT NULL COMMENT '',
  `idReserva` INT(11) NOT NULL COMMENT '',
  PRIMARY KEY (`id`) COMMENT '',
  INDEX `FK_dul0cp5bx8e7wet3ov47rpybi` (`idZona` ASC) COMMENT
'',
  INDEX `FK_reservaforeignkeyasdfsdf_idx` (`idReserva` ASC)
COMMENT '',
  CONSTRAINT `FK_dfsakljfdfs fjalsñdfjlk dads`
  FOREIGN KEY (`idReserva`)
  REFERENCES `espectaclia`.`reserva` (`id`),
  CONSTRAINT `FK_dul0cp5bx8e7wet3ov47rpybi`
  FOREIGN KEY (`idZona`)
  REFERENCES `espectaclia`.`zona` (`id`))
ENGINE = InnoDB
AUTO_INCREMENT = 420
DEFAULT CHARACTER SET = latin1
COLLATE = latin1_spanish_ci;

```

```

-----
-- Table `espectaclia`.`preulocalitatzona`

```

```

-----
CREATE TABLE IF NOT EXISTS `espectaclia`.`preulocalitzona` (
  `idEspectacle` INT(11) NOT NULL COMMENT '',
  `idZona` INT(11) NOT NULL COMMENT '',
  `preu` DOUBLE NOT NULL COMMENT '',
  PRIMARY KEY (`idEspectacle`, `idZona`) COMMENT '',
  INDEX `FK_du4cie3tekuqg6b3rkhafg43i` (`idEspectacle` ASC)
COMMENT '',
  INDEX `FK_wviroghp3anhij7mh6abvn4y` (`idZona` ASC) COMMENT
'',
  CONSTRAINT `FK_du4cie3tekuqg6b3rkhafg43i`
  FOREIGN KEY (`idEspectacle`)
  REFERENCES `espectaclia`.`espectacle` (`id`),
  CONSTRAINT `FK_wviroghp3anhij7mh6abvn4y`
  FOREIGN KEY (`idZona`)
  REFERENCES `espectaclia`.`zona` (`id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1
COLLATE = latin1_spanish_ci;

```

```

-----
-- Table `espectaclia`.`taquiller`
-----

```

```

CREATE TABLE IF NOT EXISTS `espectaclia`.`taquiller` (
  `id` INT(11) NOT NULL COMMENT '',
  `idAdministrador` INT(11) NOT NULL COMMENT '',
  `idRecinte` INT(11) NULL DEFAULT NULL COMMENT '',
  PRIMARY KEY (`id`) COMMENT '',
  INDEX `FK_ewrd72gb6qvg1pk1j014uanxq` (`idAdministrador` ASC)
COMMENT '',
  INDEX `FK_ndhlwv41k8ih2l8loktoiba95` (`idRecinte` ASC)
COMMENT '',
  INDEX `FK_ldfc2kbwkgj88et6h0eee224w` (`id` ASC) COMMENT '',
  CONSTRAINT `FK_ewrd72gb6qvg1pk1j014uanxq`
  FOREIGN KEY (`idAdministrador`)
  REFERENCES `espectaclia`.`administrador` (`id`),
  CONSTRAINT `FK_ldfc2kbwkgj88et6h0eee224w`
  FOREIGN KEY (`id`)
  REFERENCES `espectaclia`.`usuari` (`id`),
  CONSTRAINT `FK_ndhlwv41k8ih2l8loktoiba95`
  FOREIGN KEY (`idRecinte`)
  REFERENCES `espectaclia`.`recinte` (`id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1
COLLATE = latin1_spanish_ci;

```

```

-----
-- Table `espectaclia`.`valoracio`
-----

```

```

CREATE TABLE IF NOT EXISTS `espectaclia`.`valoracio` (
  `idClient` INT(11) NOT NULL COMMENT '',

```

```
`idEspectacle` INT(11) NOT NULL COMMENT '',
`comentari` VARCHAR(255) CHARACTER SET 'latin1' COLLATE
'latin1_spanish_ci' NULL DEFAULT NULL COMMENT '',
`puntuacio` INT(11) NOT NULL COMMENT '',
PRIMARY KEY (`idClient`, `idEspectacle`) COMMENT '',
INDEX `FK_9f8wd2xg7ixanlw69hp4xw848` (`idClient` ASC) COMMENT
'',
INDEX `FK_cfhpwsgfq5rmsvatmrg0aiif1` (`idEspectacle` ASC)
COMMENT '',
CONSTRAINT `FK_9f8wd2xg7ixanlw69hp4xw848`
  FOREIGN KEY (`idClient`)
  REFERENCES `espectaclia`.`client` (`id`),
CONSTRAINT `FK_cfhpwsgfq5rmsvatmrg0aiif1`
  FOREIGN KEY (`idEspectacle`)
  REFERENCES `espectaclia`.`espectacle` (`id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1
COLLATE = latin1_spanish_ci;

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

DDL de dades inicials

A continuació es mostren les sentències DDL necessàries per a generar uns usuaris de proves per poder entrar a l'aplicació. Es tracta de 3 usuaris, un de cada tipus. També es genera un usuari Client genèric que no es pot utilitzar per entrar a l'aplicació i que l'aplicació utilitza internament per tal que els Administradors puguin bloquejar localitats.

- **Administrador**
Usuari: **hector.paredes**
Contrasenya: **hectorpar**
- **Taquiller**
Usuari: **sonia.caballero**
Contrasenya: **soniacab**
- **Client**
Usuari: **gerard.varo**
Contrasenya: **gerardvar**

```
-----  
-- Usuari de tipus Administrador  
-----  
INSERT INTO `espectaclia`.`usuari` (`id`, `username`, `cognoms`,  
`dataAlta`, `nif`, `nom`, `password`) VALUES ('1',  
'hector.paredes', 'Paredes Tomás', '2015-11-15 00:00:00',  
'12345678A', 'Héctor', 'hectorpar');  
  
INSERT INTO `espectaclia`.`administrador` (`id`) VALUES ('1');  
  
-----  
-- Usuari de tipus Taquiller  
-----  
INSERT INTO `espectaclia`.`usuari` (`id`, `username`, `cognoms`,  
`dataAlta`, `nif`, `nom`, `password`) VALUES ('2',  
'sonia.caballero', 'Caballero Cercòs', '2015-12-15 21:41:30',  
'12345678H', 'Sonia', 'soniacab');  
  
INSERT INTO `espectaclia`.`taquiller` (`id`, `idAdministrador`)  
VALUES ('2', '1');  
  
-----  
-- Usuari de tipus Client  
-----  
INSERT INTO `espectaclia`.`usuari` (`id`, `username`, `cognoms`,  
`dataAlta`, `nif`, `nom`, `password`) VALUES ('3',  
'gerard.varo', 'Varó Mitjans', '2015-12-15 18:28:54',  
'15468244H', 'Gerard', 'gerardvar');
```

```
INSERT INTO `espectaclia`.`client` (`email`, `id`) VALUES
('gerard.varo@hotmail.com', '3');
```

```
-----
-- Client genèric per al bloqueig de localitats
-----
```

```
INSERT INTO `espectaclia`.`usuari` (`id`, `username`, `cognoms`,
`dataAlta`, `nif`, `nom`, `password`) VALUES ('9999',
'BLOQUEIG', 'BLOQUEIG', '1900-01-01 00:00:00', 'XXXXXXXXXX',
'BLOQUEIG', 'BLOQUEIG');
```

```
INSERT INTO `espectaclia`.`client` (`email`, `id`) VALUES
('BLOQUEIG', '9999');
```