

Creació de Carteleria via WEB



TFC – J2EE

Juan Ramón Segura Ponce

© JuanRamonSegura

Reservats tots els drets. Està prohibida la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel:lectual



Índex

Descripció	Pagina
1. Introducció	5
1.1 Introducció al TFC	5
1.2 Una mica d'Història	5
1.3 Justificació del Projecte	6
1.4 Objectiu del Treball	7
1.5 Enfocament Metodològic	7
1.6 Planificació del Projecte	8
1.7 Productes Obtinguts	9
1.8 Resum del projecte	10
2 Descripció general del projecte	11
2.1 Introducció	11
2.2 Actors del Sistema	13
2.3 Subsistemes del projecte	13
2.3.1 Subsistema de manteniment i connexió	14
2.3.1.1 Identificació i Validació de contrasenya	15
2.3.1.2 Menú Persones	15
2.3.1.3 Menú Departaments	19
2.3.1.4 Menú Usuaris	19
2.3.1.5 Menú Incidències	21
2.3.1.6 Menú Contrasenya	22
2.3.2 Subsistema de creació de cartelleria	23
2.3.2.1 Menú Articles	24
2.3.2.2 Menú Plantilles	24
2.3.2.3 Menú Cartells	25
2.3.2.4 Menú Treballs	26
2.3.2.5 Menú Incidències	26
2.3.2.6 Menú Contrasenya	26
2.3.3 Subsistema d'impressió de cartelleria	27
2.3.3.1 Menú Imprimir Cartells	28
2.3.3.2 Menú Incidències	29
2.3.3.3 Menú Contrasenya	29
3 Diseny	30
3.1 Diagrama de classes	30
3.1.1 Diagrama de classes Entities	30
3.1.2 Diagrama de classes Gestors del Sistema	32
3.1.3 Diagrama de classes Controller	33
3.2 Disseny de la Base de Dades	34
3.2.1 Disseny relacional	34
4 Diagrames	37
4.1 Diagrames de casos d'Ús i Actors	37
4.1.1 Casos d'Ús del Login	37
4.1.2 Casos d'Ús d'Incidències	38
4.1.3 Casos d'Ús de l'Administrador	38
4.1.4 Casos d'Ús del Responsable	39

4.1.5 Casos d'Ús de l'Usuari Final	39
4.2 Diagrames de Seqüència	40
4.2.1 Diagrama de seqüència de Login	40
4.2.2 Diagrama de seqüència d'Afegir Usuari	41
4.2.3 Diagrama de seqüència d'Eliminar Usuari	41
4.2.4 Diagrama de seqüència d'Imprimir Treball	42
4.3 Diagrames d'Estats	43
4.3.1 Diagrama d'Estats d'Usuari	43
4.3.2 Diagrama d'Estats d'Incidència	44
4.3.3 Diagrama d'Estats de Treball	44
5 Patró MVC	45
6 Frameworks del Projecte	46
6.1 Frameworks Struts 2	46
6.2 Frameworks Hibernate 3	47
6.2.1 Arxius de Mapeig	48
6.3 Apache Tomcat	49
6.4 Xwork Validator	49
6.5 iText	50
7 Annexos	50



Índex de Figures

Figures	Descripció	Pagina
Figura 1	Diagrama de Temps	8
Figura 2	Diagrama de Gantt	9
Figura 3	Pantalla Creació de la BD	11
Figura 4	Diagrama Actors 1	13
Figura 5	Pantalla Menú Administrador	14
Figura 6	Pantalla Login	15
Figura 7	Pantalla Menú Persones	15
Figura 8	Pantalla Alta persona	16
Figura 9	Pantalla Consultar Persona	16
Figura 10	Pantalla Persona Seleccionada	17
Figura 11	Pantalla Consulta Persona	17
Figura 12	Pantalla Consulta Persones amb Filtre	18
Figura 13	Pantalla Menú Departament	19
Figura 14	Pantalla Menú Usuaris	19
Figura 15	Pantalla Menú Incidència	21
Figura 16	Pantalla Menú Contrasenya	22
Figura 17	Pantalla Generar Password Manual	22
Figura 18	Pantalla Menú Responsable	23
Figura 19	Pantalla Menú Articles	24
Figura 20	Plantilla Menú Plantilles	24
Figura 21	Pantalla Menú Cartells	25
Figura 22	Pantalla Menú Treballs	26
Figura 23	Pantalla Menú Usuari	27
Figura 24	Pantalla Menú Imprimir Cartells	28
Figura 25	Pantalla Consulta Treballs no Impressos	28
Figura 26	Diagrama de classes relacionades amb Cartells	30
Figura 27	Diagrama de classes relacionades amb Usuari	31
Figura 28	Diagrama de classes Gestores DAO	32
Figura 29	Diagrama de classes Controller	33
Figura 30	Diagrama de Bases de Dades relacionades amb Usuari	35
Figura 31	Diagrama de Bases de Dades relacionades amb Cartells	36
Figura 32	Diagrama de Casos d'Ús de Login	37
Figura 33	Diagrama de Casos d'Ús d'Incidències	38
Figura 34	Diagrama de Casos d'Ús d'Administrador	38
Figura 35	Diagrama de Casos d'Ús del Responsable	39
Figura 36	Diagrama de Casos d'Ús de l'Usuari Final	39
Figura 37	Diagrama de Seqüència de Login	40
Figura 38	Diagrama de Seqüència d'Afegir Usuari	41
Figura 39	Diagrama de Seqüència d'Eliminar Usuari	41
Figura 40	Diagrama de Seqüència d'Imprimir Treball	42
Figura 41	Diagrama d'Estats d'Usuari	43
Figura 42	Diagrama d'Estats d'Incidència	44
Figura 43	Diagrama d'Estats de Treball	44
Figura 44	Esquema MVC	45

1-Introducció

1.1-Introduccio al TFC

En enginyeria de programari, una aplicació web és una aplicació que és accedida via web utilitzant el protocol TCP/IP, ja sigui a través d'internet o d'una intranet.

Les aplicacions web són populars gràcies a la ubiqüitat del client, que acostuma a ser un navegador i se'l classifica com client lleuger.

La principal característica de les aplicacions web és la seva capacitat de mantenir i actualitzar aplicacions web sense haver d'instal·lar-les sobre una quantitat indeterminada d'ordinadors client.

En una aplicació web les planes també poden ser dinàmiques (com és el cas) i l'usuari pot decidir quines dades envia, en forma de paràmetres, generalment amb l'ús de formularis. Si l'usuari demana dades que estan en una base de dades (o que el servidor pot demanar d'una altra manera), està disposant remotament de tota la potència d'accés a les dades que té el servidor, només complint el generalitzat requeriment d'un navegador web amb connexió a internet.

El primer avantatge dels clients web és que són multi plataforma i el seu comportament se suposa que és idèntic arreu dels diferents sistemes operatius.

1.2-Una mica d'Història

A les primeres tipologies d'arquitectures client-servidor cada aplicació tenia el seu propi programa client que feia la tasca d'interfície d'usuari i havia d'estar instal·lada separatament per cada ordinador personal. Una millora a la part del servidor podia implicar una reinstal·lació a cada estació de treball incrementant el cost de suport i decrementant la productivitat.

Un cop dissenyat el protocol HTTP, que va implicar el desenvolupament de navegadors web i que, ja des del principi, suportava enviament de paràmetres des del client cap al servidor, es va obrir la possibilitat de programar els primers CGIs, uns petits aplicatius en C que representaven els precursors de les aplicacions web.

En ser usats llenguatges que no disposaven de llibreries pensades en el món web, els CGIs no eren productius quant a desenvolupament ni quant a rendiment. A partir d'aquí Sun va desenvolupar la tecnologia de Servlets. Aquests tenien des del principi tres qualitats que van determinar la seva popularitat entre els programadors, per damunt de qualsevol de les ofertes del moment:

Eren multiplataforma. Això implicava que qualsevol servidor podia suportar un servlet (Escriu un cop, executa arreu). Això va seduir els programadors perquè seguia la línia del protocol TCP/IP, que és independent de la plataforma.

Corrien sobre un sol procés. Mentre cada invocació de cada client a un servidor implicava un nou procés, els servlets corrien tots sobre el mateix procés o conjunt de processos. El guany en rendiment era gran quan hi havia diverses peticions simultànies.

Incorporaven una complerta API pensada en l'HTTP, donant eines molt complertes als programadors. Paral·lelament d'altres companyies com Borland o Microsoft van començar a desenvolupar les seves tecnologies de la part servidora. Microsoft, líder en entorns de desenvolupament d'aplicacions locals,

amb la seva tecnologia .NET, uns pocs anys endarrerit va oferir eines que augmentaven notablement la productivitat d'aplicacions web vers les seves anteriors ofertes.

Posteriorment, l'aposta de Microsoft pels Serveis web ha donat una empenta a aquesta nova tecnologia d'intercanvi de dades entre servidors, alhora que marcava la diferència respecte els seus competidors.

Eines com AJAX han donat impuls a les aplicacions web, incrementant la usabilitat dels navegadors, requeriment imprescindible per oferir una eina al públic general.

Tot això ha portat al punt actual, el web 2.0, que trasllada la construcció dels webs als usuaris, que poden enviar els seus coneixements i les seves inquietuds i emmagatzemar-los a una base de dades des d'un entorn apte per la seva implicació.

***Wikipedia**

1.3-Justificació del Projecte

El present treball fi de carrera de l'Enginyeria Tècnica de Sistemes pretén mostrar les pautes seguides per desenvolupar una aplicació del món real basada en l'arquitectura J2EE.

L'objectiu principal del projecte és la realització de l'anàlisi, el disseny i la implementació d'una aplicació web de creació de carteleria.

Es tracta d'un projecte d'aplicació real, encara que el seu disseny s'ha fet de la manera més general possible la finalitat d'aquest es que sigui funcional.

L'aplicació està dividida en tres grans blocs: el bloc de l'administrador, el bloc del responsable de carteleria i el bloc de l'usuari, que serà qui imprimirà la carteleria.

Per al desenvolupament de l'aplicació es fa ús de l'arquitectura J2EE (Java 2 Enterprise Edition) seguint el patró de disseny MVC. Com a entorn de desenvolupament s'ha fet servir NetBeans.

El SGBD és MySQL i el servidor web emprat és Apache.

Molts dels conceptes estudiats al llarg dels semestres de l'Enginyeria, són aplicats ara en el projecte: programació orientada a objectes, enginyeria del programari, bases de dades relacionals; i d'altres desconeguts fins ara, són analitzats i estudiats, com són els diferents frameworks disponibles al mercat per desenvolupar aplicacions seguint l'especificació J2EE: com Struts 2, i Hibernate, per exemple.

La plataforma J2EE aporta avui dia una solució única, fiable i eficaç per desenvolupar i implementar aplicacions distribuïdes. Per aquest motiu em vaig sentir molt interessat per desenvolupar aquest projecte.



1.4-Objectius del treball

Passaré a determinar quins han estat els objectius per tal de portat a terme l'anàlisi, disseny i implementació de l'aplicació.

Els objectius principals d'aquest treball final de carrera han estat:

- Fer el disseny de l'aplicació en tres capes: la capa de presentació, la capa de negoci i la capa de dades, fent servir els patrons de disseny més adequats i implementant dins cada capa, els aspectes corresponents segons la seva distribució.
- Estudiar els frameworks i estàndards disponibles per a les diferents capes de l'aplicació, tant els definits per l'estàndard J2EE (JavaServer Pages, Servlets o Enterprise JavaBeans), com els desenvolupats per la comunitat amb llicència lliure (WebWork amb Struts 2, JBoss Community amb Hibernate, o Tiles d'Apache).
- Familiaritzar-se amb l'arquitectura MVC pel desenvolupament d'aplicacions
- Realitzar una Aplicació el suficientment robusta i funcional per tal de tenir una bona base per a aplicacions futures.

S'utilitzen diversos frameworks que seran explicats a posteriori.

1.5-Enfocament Metodològic

Com tot projecte de desenvolupament de programari, s'ha dividit el procés de construcció en diverses fases ben diferenciades amb objectius molt diferents.

- Especificació dels requeriments de l'aplicació.
- Anàlisi de l'aplicació.
- Disseny de l'arquitectura, model de dades i lògica de negoci.
- Disseny de l'estructura de l'aplicació.
- Implementació.
 - Desenvolupament del apartat DAO (Bases de dades)
 - Desenvolupament dels Controllers
 - Desenvolupament dels View (JSP)
 - Desenvolupament dels estils
- Proves i tests de l'aplicació.

Els beneficis d'un procés iteratiu controlat són: la reducció del risc de no treure el producte en el calendari previst, accelerar el ritme de desenvolupament i que s'adapti millor a les necessitats del client.

Ajudarà a determinar i diferenciar les funcionalitats més importants, que no poden faltar en el producte final, d'aquelles que poden quedar per a "una segona fase" pel seu baix impacte funcional.



1.6-Planificació del projecte

Per la realització del TFC hi ha delimitades unes dates d'entrega parcial de l'evolució del projecte, per tal de ser avaluat el desenvolupament del treball realitzat pel consultor de l'assignatura.

Descripció	Data Inici	Data Final
PAC 1	28/02/2013	11/03/2013
PAC 2	12/03/2013	15/04/2013
PAC 3	16/04/2013	03/06/2013
PAC 4	04/06/2013	17/06/2013

Tal i com il·lustren les figures que van a continuació, es planifica les activitats i es temporitzen.

Disposem de un diagrama Gantt

Diagrama de gantt	Fecha de inicio	Fecha de finalizació	Duración
<input type="checkbox"/> PAC 1	28/02/13	11/03/13	8
Lectura del material	28/02/13	01/03/13	2
Búsqueda de material complementari	02/03/13	04/03/13	2
Pla de treball	04/03/13	11/03/13	6
<input type="checkbox"/> PAC 2	12/03/13	12/04/13	24
Identificar classes i objectes	12/03/13	15/03/13	4
Diseny de diagrames UML	17/03/13	22/03/13	6
Creacio de classes principals	23/03/13	12/04/13	16
<input type="checkbox"/> PAC 3	15/04/13	03/06/13	36
Creacio de classes secundaries	15/04/13	19/04/13	5
Creacio de l'interfaç grafica	20/04/13	24/05/13	26
Creació de documentacio	24/05/13	03/06/13	7
<input type="checkbox"/> PAC 4	04/06/13	17/06/13	10
Finalitzar documentació	04/06/13	10/06/13	5
Creació de presentacio virtual	11/06/13	17/06/13	5

Figura 1



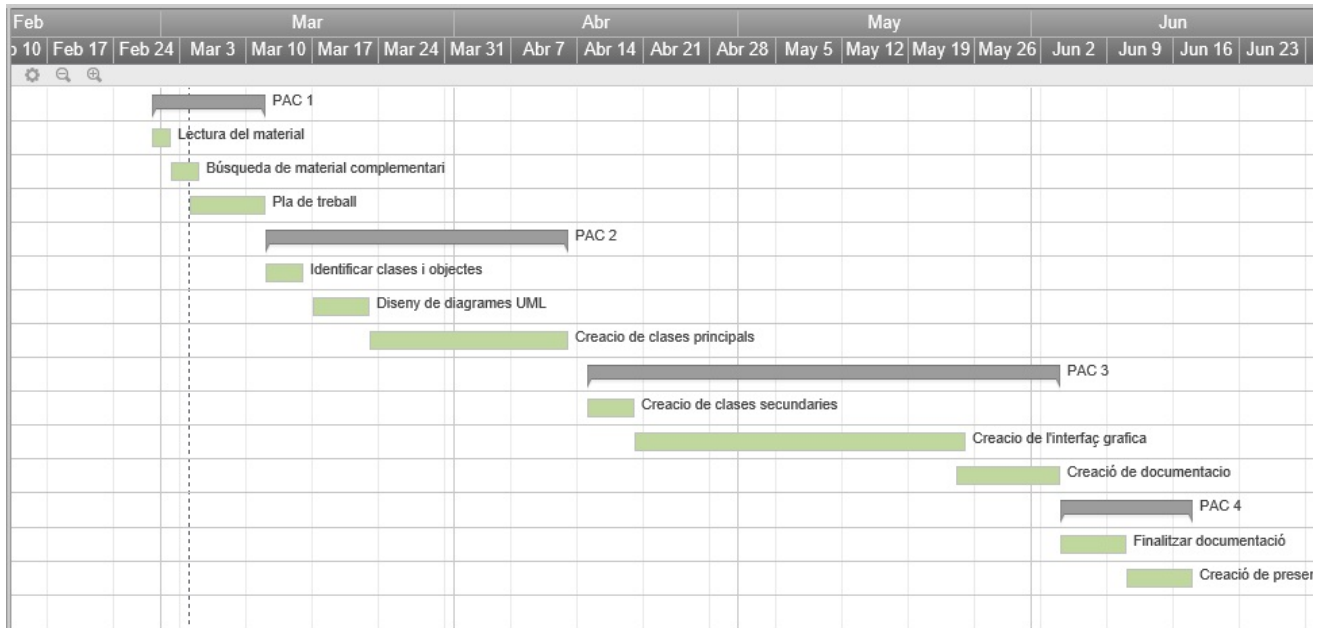


Figura 2

1.7-Productes Obtinguts

El productes obtinguts al finalitzar el TFC són els següents:

- Una aplicació web, totalment funcional de carteleria, desenvolupada seguint l'especificació J2EE i amb programari lliure.
- La memòria, que recull en forma escrita tots els aspectes tècnics del projecte, conté tota la informació rellevant per comprendre el problema plantejat en el TFC, així com la metodologia emprada per la resolució del problema plantejat.
- La presentació, que ofereix una perspectiva general del TFC, sintetitzant de forma clara i concisa el treball realitzat i els resultats obtinguts.

Carteleria, Depenent del rol dels usuaris, ofereix una interfície web tant "workbench" per a l'administrador i el responsable de carteleria com "service" per als usuaris que imprimiran els cartells ja realitzats.

El producte principal obtingut en aquest TFC ha estat l'aplicació web Carteleria destinada a la creació de carteleria i realitzada amb tecnologia J2EE–Apache Struts.

Els productes finals són:

- **jsegurapon_producte.zip**: fitxers fonts, compilats, test..etc
- **jsegurapon_memoria.pdf**: Fitxer amb la memòria
- **jsegurapon_presentación.ppt**: Presentació power point.

Tot enzipat en una arxíu anomenat **jsegurapon_Carteleria.zip**



1.8-Resum de l'estructura del projecte

L'aportació característica d'aquest TFC ha estat combinar tots els coneixements teòrics amb una implementació que utilitza fonamentalment el patró disseny MVC model 2 (patró adaptat a les característiques de les aplicacions web). Concretament s'ha fet servir el framework Struts Apache.

Els avantatges de tota aquesta arquitectura s'ha aniran exposant durant el desenvolupament de la memòria.

L'enfocament bàsic del projecte és desenvolupar un conjunt de serveis que donen resposta a les peticions provinents dels usuaris de l'aplicació web, com són els Clients i l'Administrador.

Tot i que hi ha moltes variacions possibles, una aplicació Web **s'estructura habitualment com una aplicació de tres capes**. En aquesta forma majoritària, un navegador Web és la primera capa, des del punt de vista de l'usuari. Un motor que usa una tecnologia de contingut JSP/Java que és la capa intermitja. La tercera capa és un servidor de bases de dades. El navegador web envia peticions a la capa intermitja, que les serveix fent queries i actualitzacions contra la base de dades i genera una resposta a l'usuari.

JavaServer Pages és una tecnologia que permet als desenvolupadors de pàgines web, generar respostes dinàmicament a peticions HTTP. La tecnologia permet que codi Java i certes accions predefinides siguin incrustades en un context estàtic.

La sintaxi de JSP incorpora tags XML addicionals, anomenats accions de JSP, per ser usats per invocar altres funcions. Addicionalment, la tecnologia permet la creació de llibreries d'etiquetes que actuen com extensions de l'estàndard d'etiquetes HTML o XML. Les llibreries d'etiquetes aporten una forma multi plataforma d'ampliar les capacitats d'un servidor web.

Els JSPs són compilats en Servlets per un compilador JSP. Aquest pot generar un servlet o generar bytecode directament.

El navegador web, que proporciona una interfície d'usuari, envia diverses peticions de serveis a la capa lògica i aquesta ofereix resposta tot utilitzant consultes i actualitzacions sobre la base de dades.

Depenent del rol dels usuaris (client o administrador) el sistema oferirà uns serveis o uns altres.

Com es podrà veure més endavant carteleria és una aplicació desenvolupada amb tecnologies actuals: J2EE, Apache Struts 2 i Hibernate 3.



2-Descripció general del projecte

2.1-Introducció

Cartelleria es una web dedicada a la creació i impressió de cartelleria. L'aplicació web, desenvolupada en Java en un entorn client-servidor amb suport d'un sistema de base de dades, disposarà d'una part d'accés pública – Usuaris Finals registrats (impressió de cartelleria)-, i una part d'accés privada, per a la gestió i l'administració de les dades de l'espai web – Administrador i responsable de cartelleria- .

Les principals funcionalitats i requeriments del projecte a l'**àrea pública** són les següents:

- Poder registrar-se com a usuari amb les seves dades personals i posteriorment identificar-se com a tal a la web
- Impressió de cartelleria (creada per el responsable) filtrada per 3 opcions, "No impresa", "Impresa" i "rebutjada"
- Obrir incidències i consultar-les
- Canvi de contrasenya
- Desconnectar-se com a client de la web

En l'**àrea privada** trobem tres àrees diferenciades, les funcionalitats i requeriments són les següents:

L'àrea del superAdministrador:

El sistema consta de un superAdministrador predefinit usuari=" ChopPrograms" i password="firstPrizy".



The screenshot shows a web browser window with the address bar displaying 'http://localhost:8084/'. The page title is 'Creacion de la BD - JSP Page'. The main content area has a header 'PANTALLA LOGIN' and 'Creacion de la BD'. The title 'Creacion de la BD' is centered in a large, bold font. Below the title, there are two labels: 'Usuario:' and 'Password:'. Under 'Usuario:' is a text input field containing the word 'Usuario'. Under 'Password:' is a text input field containing the word 'Password'. At the bottom, there are two buttons: 'Crear BD' and 'Valores Iniciales'. The browser's address bar and window controls are visible at the top.



Demana el usuari i contrasenya de la Base de Dades.

Aprofitant que tots els “Sistemes de Gestió de bases de dades” d’origen porten una BD anomenada “test”, fa una connexió a aquesta i des d’aquesta connexió crea la nostra BD i les taules.

A més a més es podran insertar els registres basics per al funcionament del programa.

L'àrea Administrador:

Tots els apartats permeten donar d’alta, modificar, eliminar i consultar els registres.

El sistema identifica a persona, les dades de la persona que pot tenir accés al sistema i usuari, que es la persona que s’ha registrat i te accés al sistema. Per temes de seguretat, un usuari no es podrà donar d’alta si no estan registrades les seves dades de “persona”.

- Persona: Són les persones que l’administrador doó la possibilitat de accés al sistema.
- Departaments: Són els departaments que donarà d’alta l’administrador, tots els usuaris tenen que estar assignats a un departament.
- Usuaris: L’administrador podrà donar d’alta usuaris de qualsevol tipus “Administrador”, “Responsable” i “Usuari”.
- Incidències: Les incidències són bi-direccionals, són les encarregades de transmetre informació entre usuaris.
- Canvi de contrasenya
- Desconnectar-se com a Administrador de la web.

L'àrea Responsable:

- Articles: El responsable serà l’encarregat de gestionar els articles, si es creen amb anterioritat, la tasca de creació de cartells es molt mes ràpida.
- Plantilles: Són les plantilles dels cartells
- Cartells: Els cartells unitat mínima d’impressió
- Treballs: Són un conjunt de cartells del mateix tipus (o que utilitzen el mateix paper). A l’hora d’imprimir, s’imprimeixen aquests.
- Incidències: Les incidències són bi-direccionals, són les encarregades de transmetre
- Canvi de contrasenya
- Desconnectar-se com a Responsable de la web.



2.2-Actors del sistema

Els actors són totes les entitats externes al sistema que guarden una relació amb aquest i que li demanden una funcionalitat. Poden ser operadors humans, o fins i tot entitats abstractes, com el temps. En el cas dels sers humans, es poden veure als actors com a definicions de rol, un mateix individu pot correspondre a un o més actors. Aquests són els actors que interactuen en el sistema:

Super administrador: Serà l'encarregat de Donar d'alta els administradors del sistema.

Administrador: Els administradors del sistema són aquells que s'encarreguen del manteniment del sistema així com de registrar als Usuaris.

Responsable Carteleria: Serà l'usuari qui s'encarregarà de crear els treballs d'impressió (cartells).

Usuari final: Serà l'usuari qui accedirà al sistema par a imprimir la carteleria creada de forma centralitzada.

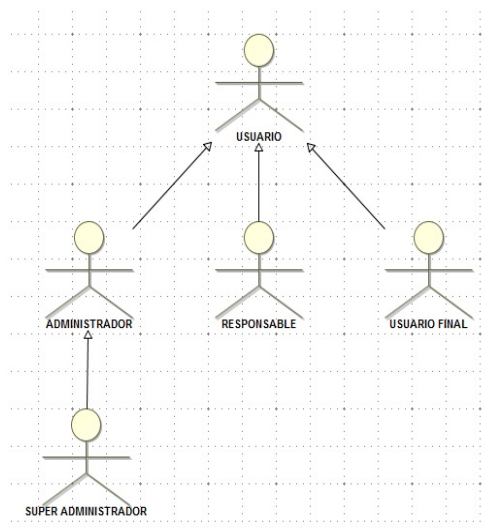


Figura 3

2.3-Subsistemes del projecte

En un marc dins de l'àmbit del desenvolupament d'aplicacions web sota l'arquitectura J2EE. Un aspecte especialment interessant de la tecnologia J2EE és que proporciona un model complet per suportar arquitectures distribuïdes, així s'aconsegueix que una cadena d'establiments puguin fer ús d'aquest programari.

Aquest projecte té varies funcionalitats (subsistemes) clarament diferenciades, aquestes són:

- Subsistema de manteniment i connexió (administrador del sistema)
- Subsistema de creació de carteleria (responsable carteleria)
- Subsistema de impressió de carteleria (usuari final)

Aquests usuaris podran ser un Administrador del sistema, uns quants responsables de cartelleria i un nombre indeterminat d'usuaris finals.



2.3.1-Subsistema de manteniment i connexió

El subsistema de manteniment i connexió permetrà la gestió i manteniment dels usuaris que tindran accés a la web, així com els elements bàsics per al bon funcionament del programari.

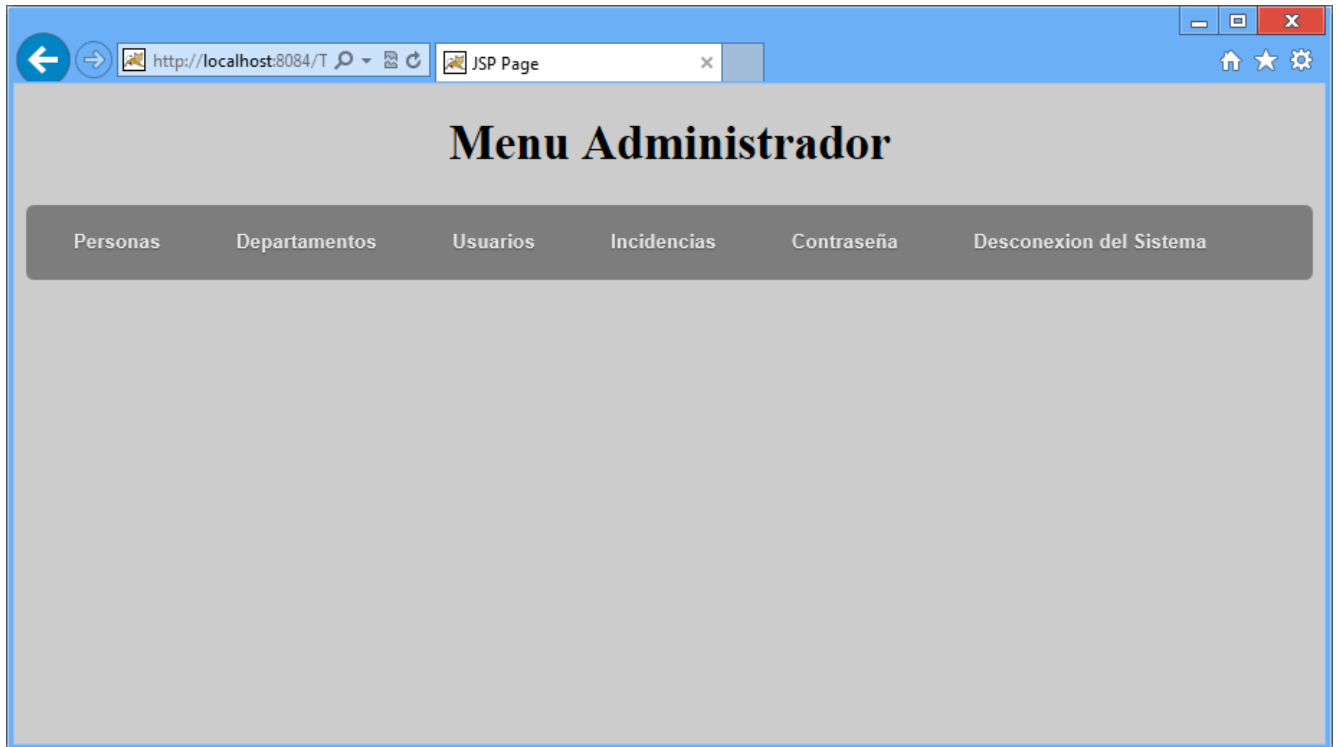


Figura 5

***Tots els apartats tenen la gestió de Inserció, modificació, eliminació i consulta.**

Les principals funcionalitats d'aquest subsistema són:

- Identificació i validació de contrasenya (Pantalla login)
 - Sistema de login
 - Comprovació de contrasenya
 - Comprovació de caducitat de la contrasenya
 - Comprovació de estat en sistema
 - Sistema de recuperació de contrasenya
 - Sistema de enregistrament automàtic
- Persona*
- Departament*
- Usuari
 - Alta manual d'usuaris
 - Consulta d'usuaris
 - Edició d'usuaris
- Incidències
 - Crear Incidència
 - Consultar Incidència
 - Tancar Incidència
- Canvi de contrasenya Usuari



2.3.1.1-Identificació i validació de contrasenya



Figura 6

En executar-se per el programa demanà un login amb un nom d'usuari i password que dóna accés al sistema.

En la mateixa pantalla, hi ha un link on l'usuari podrà recuperar la contrasenya i un altre on es podrà registrar.

2.3.1.2-Menú persones

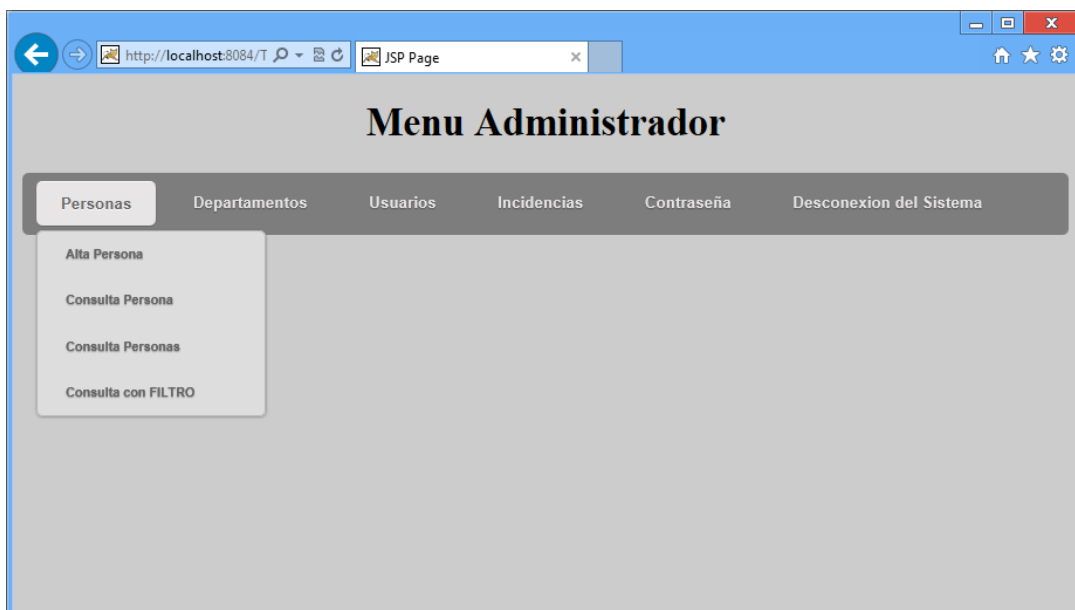
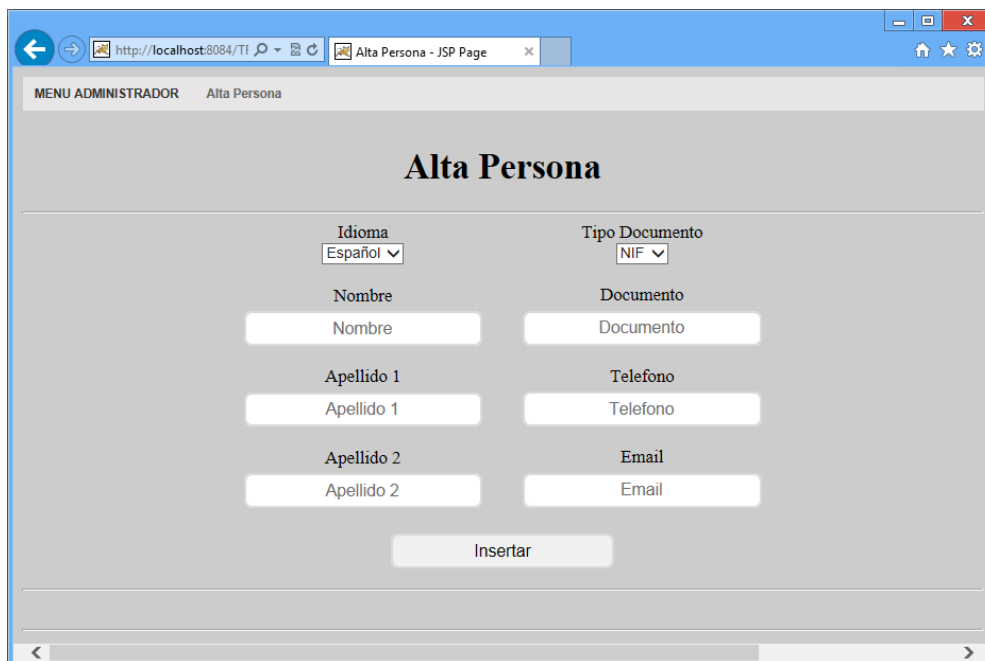


Figura 7

Tots els apartats tenen la gestió de Inserció, modificació, eliminació i consulta per tal de no repetir en excés s'explica només alguns dels casos.



Alta persona



The screenshot shows a web browser window with the URL `http://localhost:8084/TI` and the page title 'Alta Persona - JSP Page'. The page content includes a navigation menu with 'MENU ADMINISTRADOR' and 'Alta Persona'. The main heading is 'Alta Persona'. Below the heading, there are two dropdown menus: 'Idioma' (set to 'Español') and 'Tipo Documento' (set to 'NIF'). There are two columns of input fields: the left column contains 'Nombre', 'Apellido 1', and 'Apellido 2'; the right column contains 'Documento', 'Telefono', and 'Email'. Each field has a placeholder text matching its label. At the bottom center, there is an 'Insertar' button.

Figura 8

Un cop omplert les dades clicant al botó insertar es dona d'alta el registre. Tots els camps que tenen en el seu interior el nom són obligatoris. Per tant en aquest cas tots els camps són obligatoris.

Consulta persona

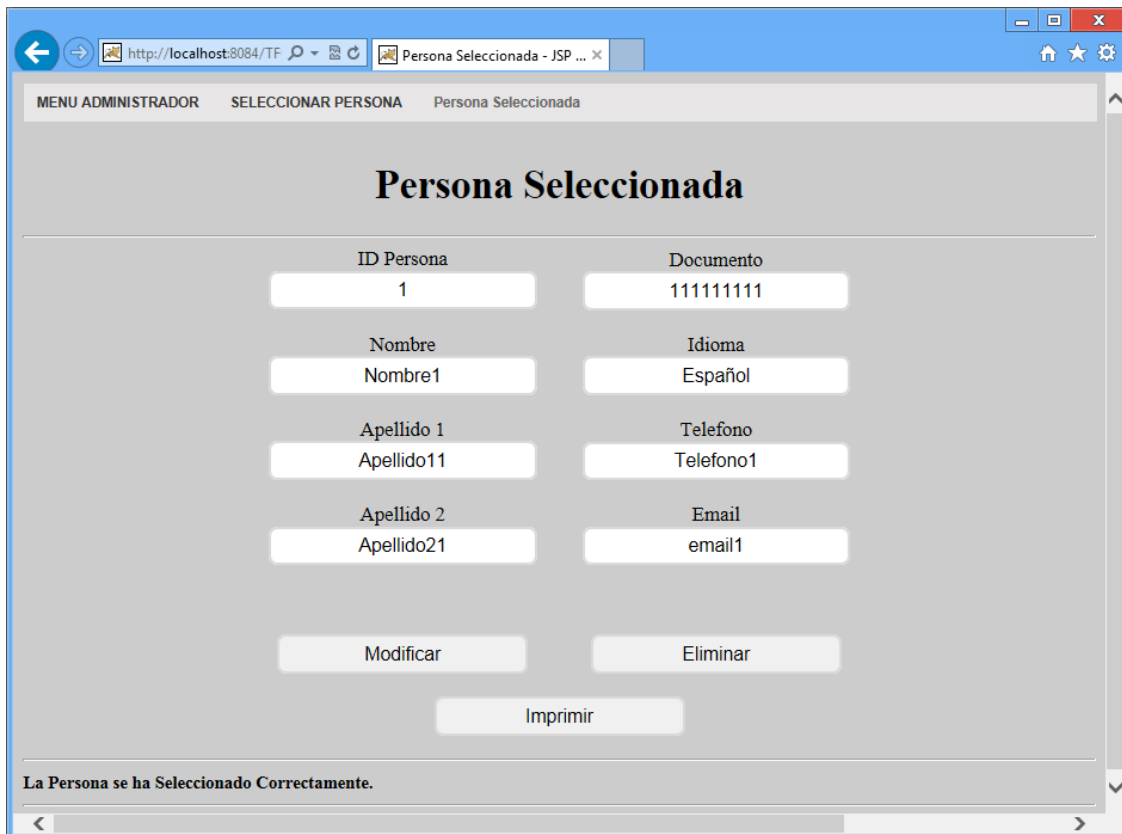


The screenshot shows a web browser window with the URL `http://localhost:8084/TI` and the page title 'Seleccionar Persona - JSP Pa...'. The page content includes a navigation menu with 'MENU ADMINSTRADOR' and 'Consultar Persona'. The main heading is 'Consultar Persona'. Below the heading, there is a dropdown menu labeled 'Buscar por ID'. There are two columns of input fields: the left column contains 'Codigo ID', 'Numero de Documento', 'Telefono', and 'Email'; the right column contains 'Documento', 'Telefono', and 'Email'. Each field has a placeholder text matching its label. At the bottom center, there is a 'Seleccionar' button.

Figura 9

La persona es selecciona a través dels seus camps únics, un cop seleccionada se'ns mostra la següent pantalla:





MENU ADMINISTRADOR SELECCIONAR PERSONA Persona Seleccionada

Persona Seleccionada

ID Persona	Documento
1	111111111
Nombre	Idioma
Nombre1	Español
Apellido 1	Telefono
Apellido11	Telefono1
Apellido 2	Email
Apellido21	email1

Modificar Eliminar

Imprimir

La Persona se ha Seleccionado Correctamente.

Figura 10

Des d'aquesta pantalla tenim accés a totes les funcions que podem aplicar al registre, en aquest cas es pot modificar, eliminar o Imprimir.

Consulta personas



MENU ADMINISTRADOR Consulta de Personas

Consulta de Personas

Lista de Personas

ID Persona	Documento	Nombre	Apellido 1	Apellido 2	Idioma	Telefono	Email
1	111111111	Nombre1	Apellido11	Apellido21	Español	Telefono1	email1
2	222222222	Nombre2	Apellido12	Apellido22	Ingles	Telefono2	email2
3	333333333	Nombre3	Apellido13	Apellido23	Frances	Telefono3	email3
4	1234562	joselete	asdasd	asdad	Español	973262819	qweqwe@asdsa.com
5	asd	asd	asd	asd	Español	asd	asd

Lista de Personas

Imprimir

Las Personas se ha Seleccionado Correctamente.

Figura 11



Des de l'opció Consulta persones accedim a una pantalla on se'ns mostra el llistat de tots els registres existents en el sistema. A més a més tenim l'opció d'imprimir aquest llistat clicant a imprimir.

El sistema genera un arxiu PDF amb les dades.

Consulta persones amb FILTRE

The screenshot shows a web browser window with the URL `http://localhost:8084/TFC/screenSe`. The page title is 'MENU ADMINISTRADOR Consulta de Personas con FILTRO'. The main heading is 'Consulta de Personas con FILTRO'. Below the heading, there are several input fields and dropdown menus for filtering:

- Nombre:
- Apellido 1:
- Apellido 2:
- TipoDocumento:
- Idioma:

Below the filters, there are two buttons: 'Listar' and 'Imprimir'. At the bottom, there is a table titled 'Lista de Personas' with the following columns: ID Persona, Documento, Nombre, Apellido 1, Apellido 2, Idioma, Telefono, and Email. The table content is currently empty.

Figura 12

Amb aquesta opció podem crear un llistat personalitzat amb les dades que volem amb un o varis camps com filtre.

Si no es selecciona cap filtre, el sistema generarà un llistat de tots els registres.



2.3.1.3-Menú Departaments

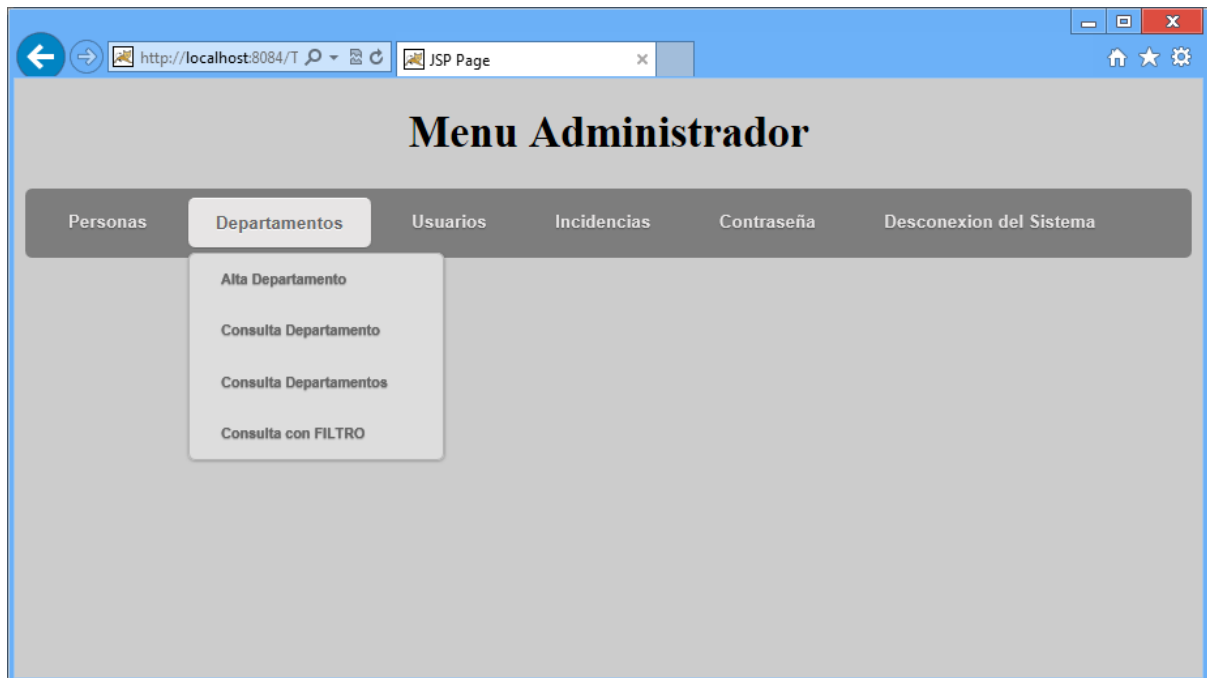


Figura 12

El menú Departaments consta de les mateixes funcions que persones. L'administrador és l'encarregat de donar d'alta i administrar els departaments existents.

2.3.1.4-Menú Usuarios

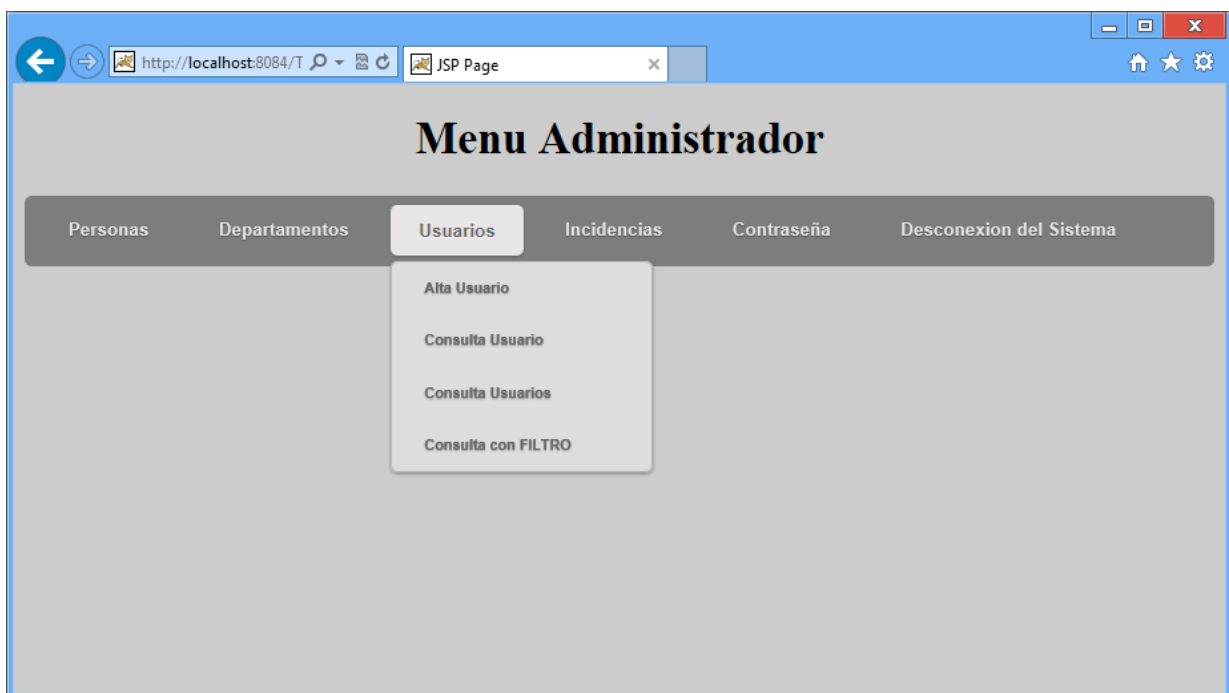


Figura 13



Aquest subsistema contindrà un sistema automàtic de enregistrament d'usuaris, per tant, sol·licitarà al usuari que ompli un formulari, si aquest és correcte enregistrarà les dades i donarà d'alta l'usuari.

Per a poder utilitzar aquesta opció, l'administrador primer ha de enregistrar-te com a persona, Un cop les teves dades estan en el sistema, es pots enregistrar com usuari.

Si l'administrador del sistema es l'encarregat d'enregistrar l'alta els nous usuaris, d'aquests haurà d'indicar:

- Usuari
- NIF
- Nom i cognoms
- Perfil d'usuari (ResponsableCarteleria o UsuariFinal)
- Numero identificatiu (numero de establiment o departament)
- Telèfon de contacte
- Correu electrònic

Un cop introduït les dades el sistema genera una contrasenya, la enregistra i l'envia per correu electrònic. També enregistra la data de l'alta del usuari.

El usuari i el NIF han de ser únics, de manera que no se'n acceptarà cap de duplicat. Per tant, si s'indica un repetit, el programa rebutjarà l'alta.

Consulta d'usuaris

Hi ha dos tipus de consulta, la consulta per valors únics, és a dir consulta que només donarà un registre en cas de ser satisfactori i la consulta per valors genèrics, en aquest cas el sistema pot donar un conjunt de registres.

Inicialment l'administrador del sistema des del menú principal entrarà en el apartat de Usuaris/Consulta d'usuaris i es mostrarà una pantalla on demana els valors de registre únics.

Si utilitzem l'opció de Usuaris/Consulta per FILTRE ens mostrarà una pantalla on podrem filtrar els registres.

Edició d'usuaris

Un cop feta la selecció de l'usuari, des de el menú Usuaris/Consulta d'usuaris, el sistema permet la edició d'aquest, les funcionalitats més importants son:

- Modificar Usuari
- Eliminar Usuari
- Bloquejar Usuari
- Desbloquejar Usuari
- Generar contrasenya Usuari
- Imprimir Usuari

Generar contrasenya Usuari



El sistema conta de dos sistemes de canvi de contrasenya, l'administrador podrà canviar la contrasenya del usuari generant una "aleatòria".

Cada Usuari utilitzant la barra superior, Contrasenya/Canvi contrasenya es mostra una pantalla on indicant la contrasenya en vigor es podrà fer un canvi de contrasenya.

2.3.1.5-Menu Incidències

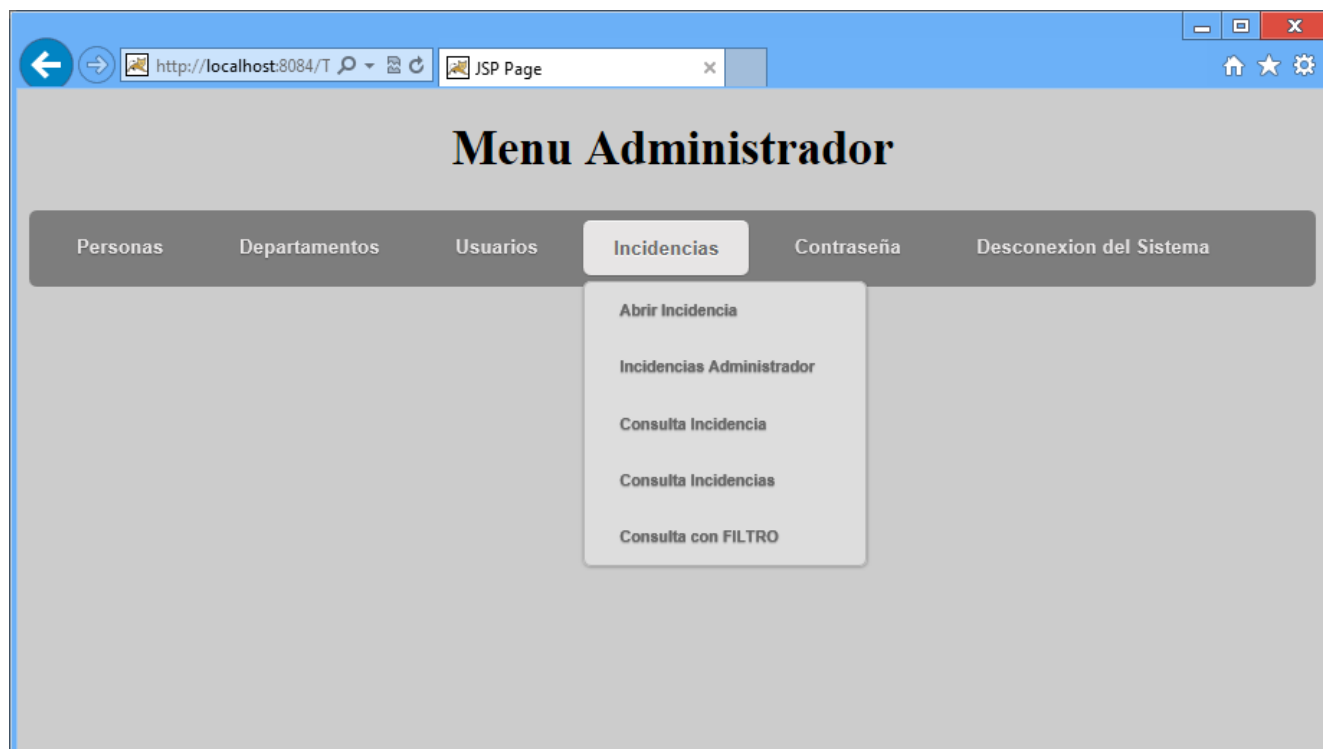


Figura 14

Aquest mecanisme que serveix com un mitjà de comunicació entre usuaris.

Sempre que sigui oportú els usuaris podran crear una incidència, serà com una mena de bústia de correu.

Entenem com a incidència un petit registre on constarà el l'usuari que la fa, la data, assumpte i un text.

L'usuari receptor un cop que te una incidència, podrà tancar-la afegint un petit comentari.

Per tant es podrà, obrir una incidència, consultar una incidència i tancar-la, consultar totes les incidències i consultar incidències que compleixen la condició d'un determinat paràmetre.

2.3.1.6-Menú Contrasenya



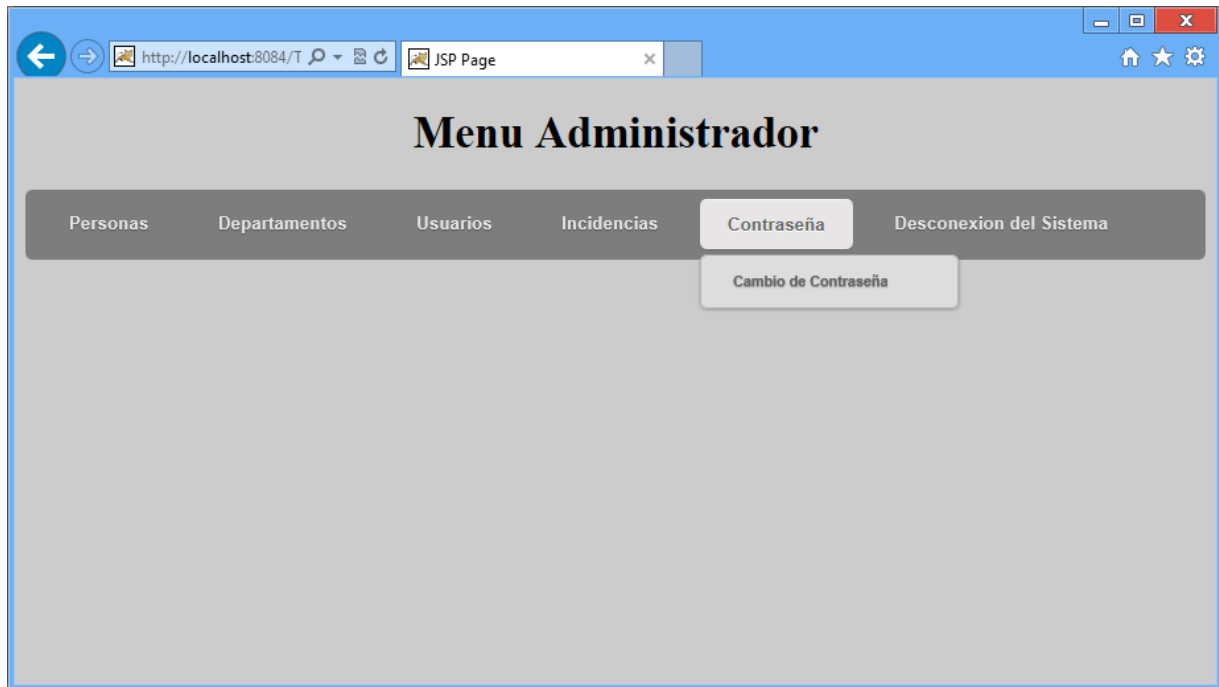


Figura 15

Aquest és el menú on cada usuari podrà canviar la seva contrasenya de forma manual, el sistema demana la contrasenya anterior i confirma la nova.

A screenshot of a web browser window displaying the 'Generar Password Manual' page. The browser's address bar shows 'http://localhost:8084' and the page title is 'Password Manual - JSP Page'. The page has a breadcrumb trail 'MENU ADMINISTRADOR > Generar Password Manual'. The main heading is 'Generar Password Manual'. Below the heading, there are four input fields for 'Usuario', 'Password Anterior', 'Password Nuevo', and another 'Password Nuevo' field. At the bottom, there is a 'Modificar' button.

Figura 16



2.3.2-Subsistema de creació de cartelleria



Figura 17

El subsistema de creació de cartelleria permetrà la creació de cartells basat en plantilles que després seran agrupats en llistes de impressió per a que l'usuari final pugui imprimir-les

Les principals funcionalitats d'aquest subsistema són:

- Creació Articles
- Creació de Plantilles
- Creació de cartells
- Creació de treballs (grup de cartells)
 - Visualitzar treballs
 - Creació de treballs en PDF
- Incidències
 - Crear Incidència
 - Consultar Incidència
 - Tancar Incidència
- Canvi de contrasenya

Bàsicament aquest subsistema l'utilitzarà el responsable de cartelleria per a crear la cartelleria.

Un cop creada la cartelleria, es generen llistes de impressió o treballs (separats per tipus de paper), un cop creats podran imprimir en PDF per a que l'usuari pugui imprimir-les.



2.3.2.1-Menú Articles



Figura 18

El sistema cada cop que crea un cartell o un treball sempre crea l'article, però per agilitzar aquestes tasques, el sistema consta del menú Articles

Si l'article ja esta en el sistema crear un cartell o un treball (grup de cartells amb mateixa plantilla) es molt mes ràpid, introduint alguns dels seus valors únics com el ID o el Codi EAN o CODI Intern (SMS) s'omplen automàticament tots el camps referents al l'article.

2.3.2.2-Menú plantilles

El responsable te la funcionalitat de crear les plantilles que s'utilitzaran per a crear els cartells.



Figura 19



2.2.2.3-Menú Cartells



Figura 20

Basats en les plantilles existents el responsable de carteleria omplirà un formulari amb les dades del cartell i clicarà afegir, si la operació s'ha realitzat correctament, el cartell s'afegirà a una llista.

Les dades a omplir (entre altres) son:

- Plantilla
 - Tampany (Ex:A4, A5, Display, 7x7)
 - Tipus (Ex: Acumula, Estalvia, Lot, Regal, Pgc, Pft, Preu anterior)
- Article
 - Descripció 1
 - Descripció 2
 - Preu
 - Preu oferta
 - Codi
- Data inici
- Data final



2.2.2.4-Menú Treballs



Figura 21

Quan el responsable de cartelleria ho cregui convenient, clicant al botó crear Treball, es crearan tants treballs com tipus de paper a utilitzar (hi ha cartells de diferents tipus que utilitzen el mateix paper) agrupant els cartells en elles.

2.2.2.5-Menú Incidències

Aquest menú és igual per a tots els usuaris, explicat en el apartat d'administrador.

2.2.2.6-Menú Contrasenya

Aquest menú és igual per a tots els usuaris, explicat en el apartat d'administrador.

2.3.3-Subsistema d'impressió de cartelleria





Figura 22

El subsistema d'impressió de cartelleria permetrà la ilmpressió dels treballs creats i enviats per el responsable de cartelleria.

Les principals funcionalitats d'aquest subsistema són:

- Visualització de cartelleria
- Impressió de cartelleria
- Rebuig de cartelleria
- Incidències
- Canvi de contrasenya

Bàsicament aquest subsistema serà utilitzat pels usuaris finals.

El sistema consta de tres llistes, treballs impresos , treballs no impresos i treballs rebutjats. La llista de treballs no impresos serà on l'usuari trobarà els treballs nous que vagin creant els responsables de cartelleria.

L'usuari final tindrà dos opcions, el que serà mes habitual, que serà imprimir la cartelleria el que farà que aquest treball sigui enviat a la llista de treballs impresos o rebutjar-lo que enviarà el treball a la "paperera de treballs" llista de treballs rebutjats.

2.3.3.1-Menú Imprimir Cartells





Figura 23

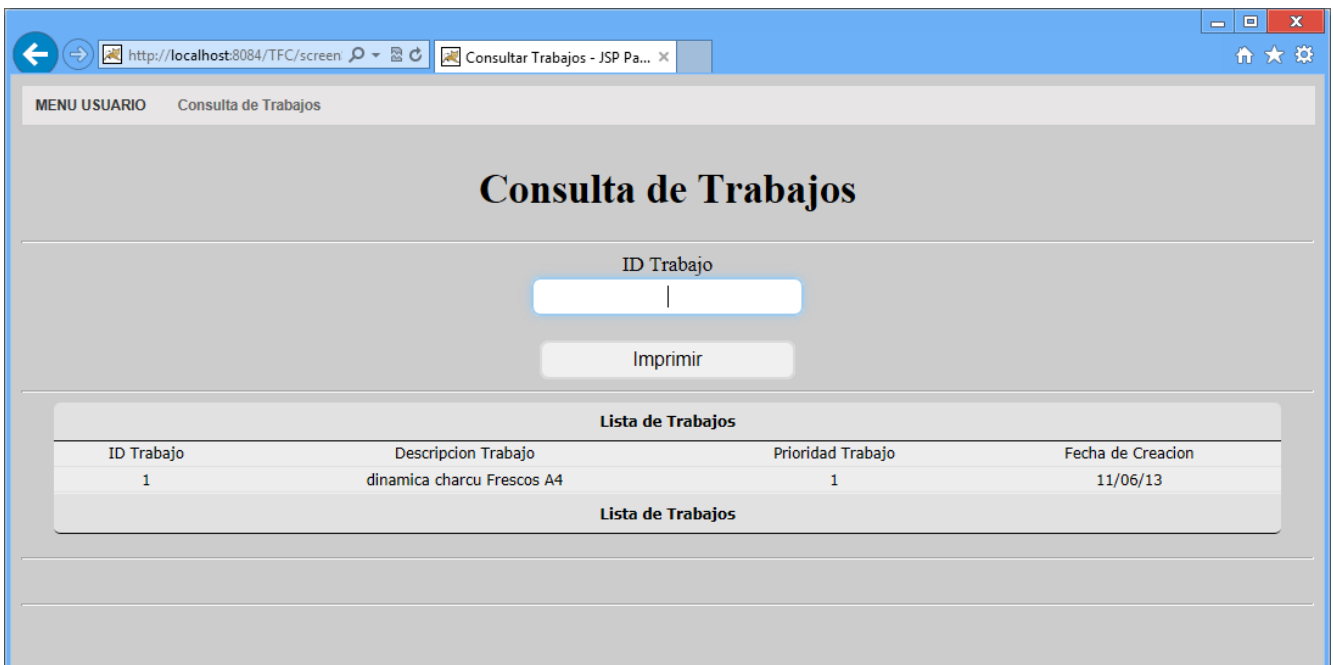


Figura 24



El sistema en proporciona 3 llistes per a la impressió de cartelleria, aquestes funcionen de igual manera. Ens mostra en tots els treballs que estan disponibles depenent de la llista (No impresos, impresos.)

Introduïm el ID del treball que volem imprimir i cliquem en imprimir, això generarà un arxiu PDF amb els cartells.

2.3.3.2-Menú Incidències

Aquest menú és igual per a tots els usuaris, explicat en el apartat d'administrador.

2.3.3.3-Menú Contrasenya

Aquest menú és igual per a tots els usuaris, explicat en el apartat d'administrador.



3-Disseny

Un diagrama de classes és un tipus de diagrama estàtic que descriu l'estructura d'un sistema mostrant les seves classes, atributs i les relacions entre ells.

Els diagrames de classes són utilitzats durant el procés d'anàlisi i disseny dels sistemes, on es crea el disseny conceptual de la informació que es manejarà en el sistema, i els components que s'encarregaran del funcionament i la relació entre tots dos.

3.1-Diagrama de Classes

3.1.1 Diagrama de classes Entities

Les classes estan dividides en 2 grups, les classes que tenen relació amb els usuaris, y les classes que tenen relació amb els cartells. Per tant tenim dos esquemes de classes.

Les classes relacionades amb Cartell són:

Artículo: És la classe encarregada de guardar tota la informació referent als articles

Plantilla: És la classe encarregada de guardar tota la informació de com i de què manera és el cartell.

Cartell: És la combinació entre una Plantilla i un Article.

Trabajo: És el conjunt de Cartells que utilitzen la mateixa plantilla.

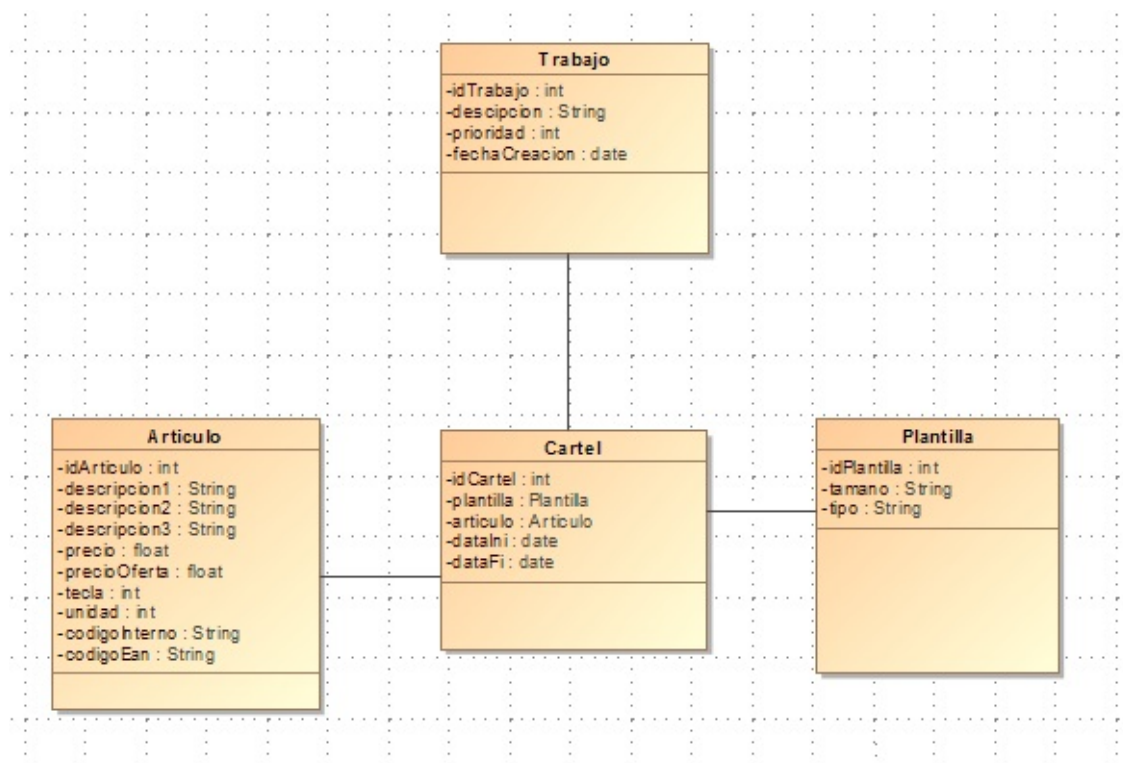


Figura 25



Les classes relacionades amb Usuari són:

Persona: És la classe que conte tota l'informació de la persona

Usuari: És la classe que conte l'informació de la persona com a Usuari

Login: És la classe encarregada de guardar el usuari i contrasenya del Usuari

Document: És la classe encarregada de guardar el DNI

Departament: És la classe encarregada de guardar l'informació referent al departament

Incidència: És la classe que estructura la incidència

EstatTreballs: És la relació entre el usuari i els treballs

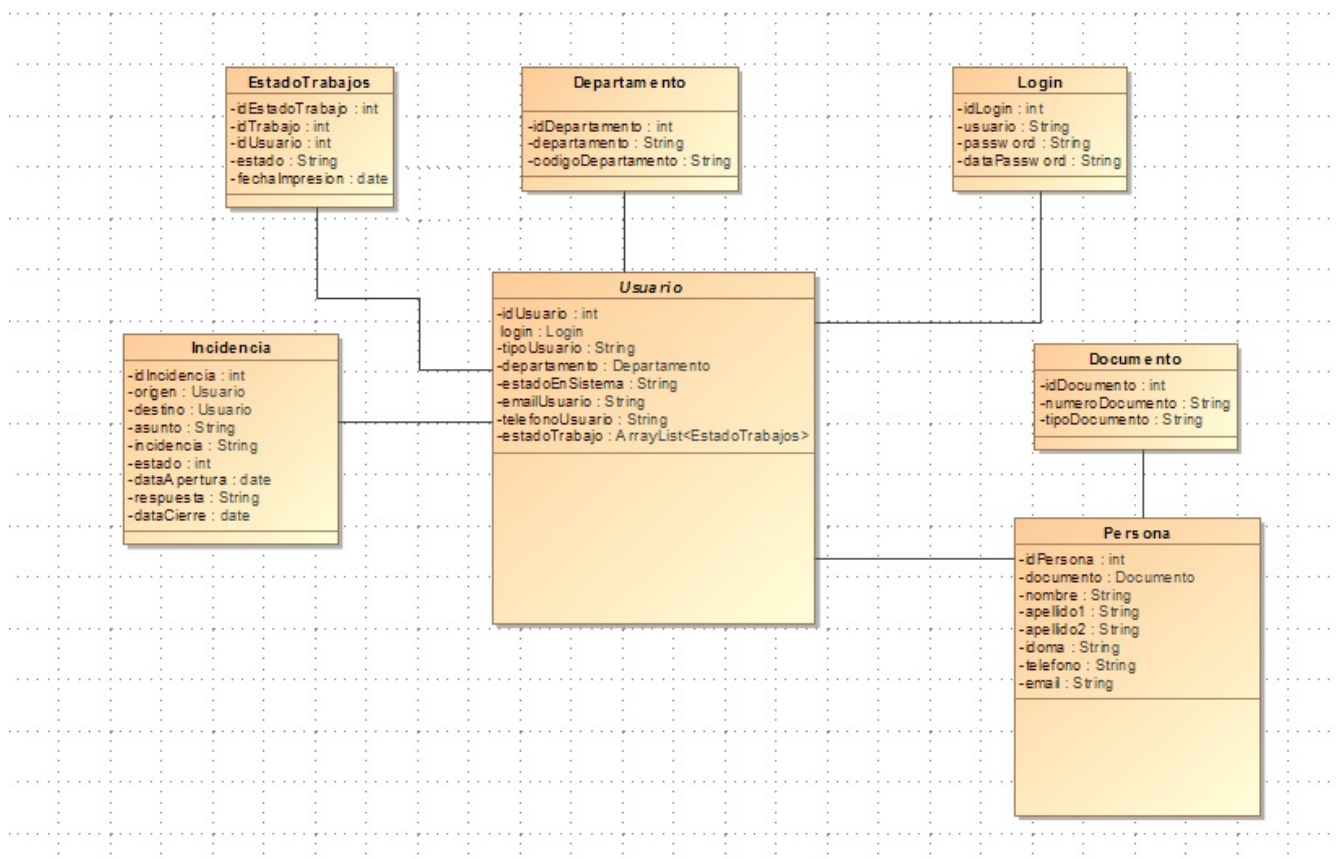


Figura 26



3.1.2 Diagrama de classes Gestors del sistema

El sistema tindrà un gestorDAO per a cada base de dades. En aquest cas la connexió es fa a través de hibernate. Totes les modificacions es realitzaran per persistència. Per tant els gestorsDAO no s'utilitzaran per a afegir elements a la BD i per a realitzar consultes.

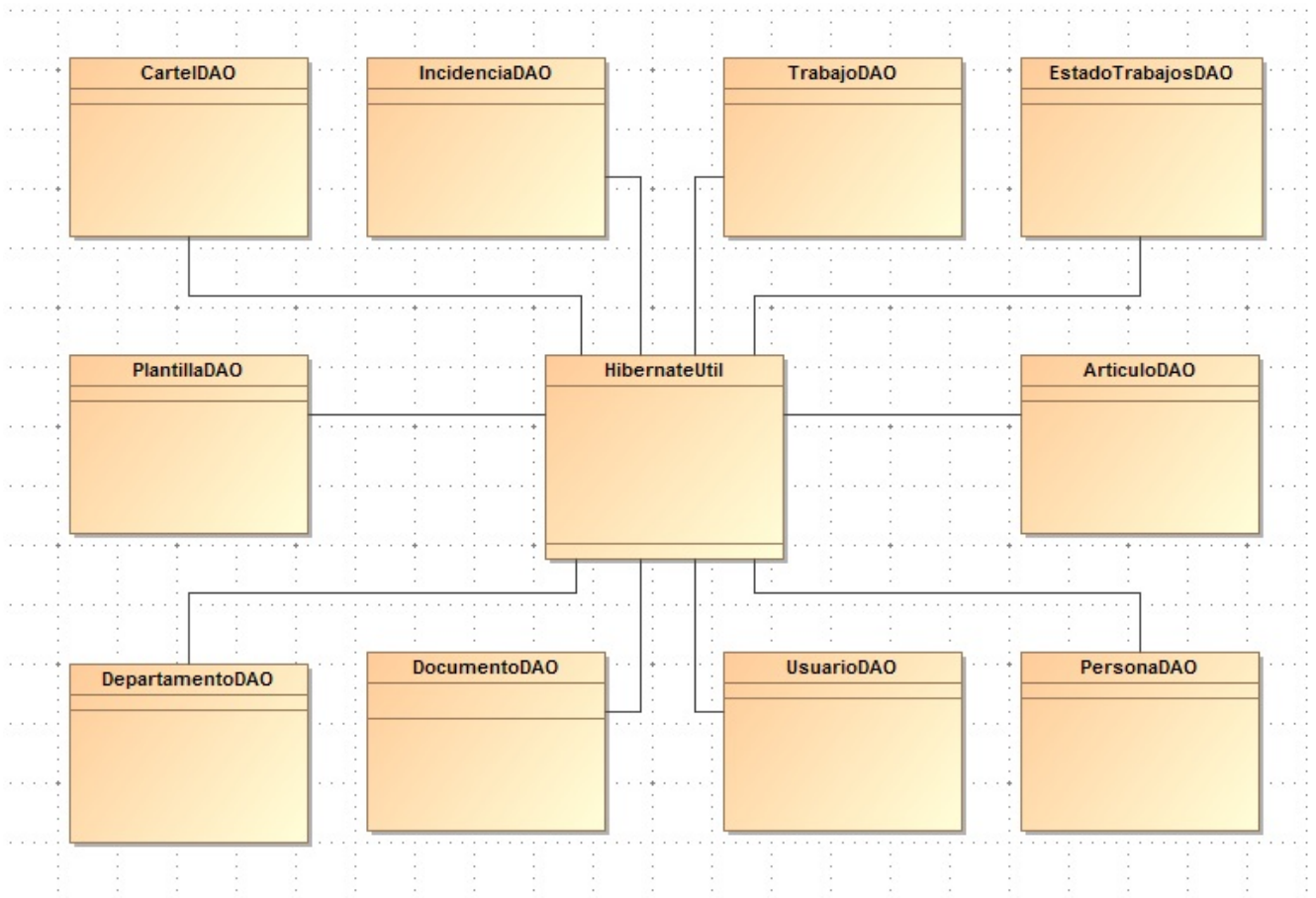


Figura 27



3.1.3 Diagrama de classes Controller

Dintre de la capa Model hi ha les classes controller, són les encarregades de gestionar i respondre als events, normalment seran les accions de l'usuari i invoca les peticions del al model, normalment seran les vistes.

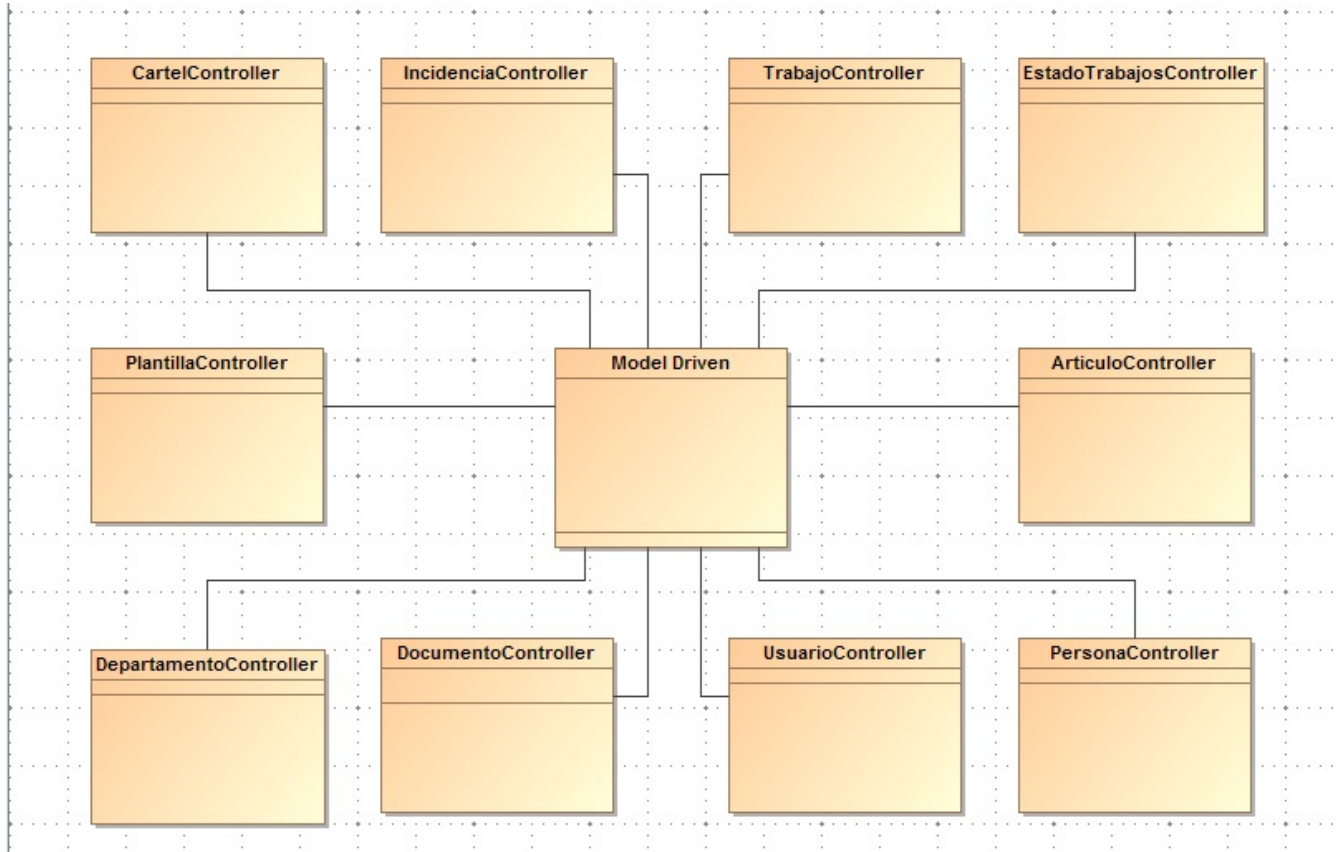


Figura 28



3.2 Disseny de la base de dades

Una base de dades relacional és una base de dades que compleix amb el model relacional, el qual és el model més utilitzat en l'actualitat per implementar bases de dades ja planificades. Permeten establir interconnexions (relacions) entre les dades (que estan guardats en taules), i a través d'aquestes connexions relacionar les dades de les dues taules, d'aquí prové el seu nom: "Model relacional".

En una base de dades relacional, totes les dades s'emmagatzemen i s'hi accedeix a través de relacions. Les relacions que emmagatzemen dades són anomenades "relacions base" i la seva implementació és anomenada "taula". Altres relacions no emmagatzemen dades, però són calculades en aplicar operacions relacionals. Aquestes relacions són anomenades "relacions derivades" i la seva implementació és anomenada "vista" o "consulta". Les relacions derivades són convenientes ja que expressen informació de diverses relacions actuant com si fos una sola.

Les restriccions són limitacions que obliguen el compliment de certes condicions en la base de dades. Algunes no són determinades pels usuaris, sinó que són inherentment definides pel simple fet que la base de dades sigui relacional.

Les restriccions no són part formal del model relacional, però són incloses perquè juguen el rol d'organitzar millor les dades. Les restriccions són molt discutides juntament amb els conceptes relacionals.

3.2.1 Disseny relacional

Aquest projecte constarà amb una forta presència de bases de dades.

DOCUMENTO (idDocumento integer, identificacion varchar(9), tipoDocumento varchar(3))

PERSONA(idPersona integer, idDocumento integer (FK), nombre varchar(16), apellido1 varchar (16),apellido2 varchar (16),idioma varchar 16),telefono varchar (10),email varchar (32))

DEPARTAMENTO(idDepartamento integer, departamento varchar (32), codigoDep varchar (4))

CREATE TABLE LOGIN (idLogin integer, usuario varchar (16),password varchar (16), dataPassWord date)

USUARIO (idInterno integer, idLogin integer(FK), tipoUsuario varchar (1), idDepartamento integer(FK), estadoEnSistema varchar (1), emailUsuario varchar (64), telefonoUsuario varchar (10))

INCIDENCIA (idIncidencia integer, origen integer(FK), destino integer(FK), asunto varchar (64), incidencia varchar (320), estado varchar (1), dataAbertura date, respuesta varchar (320), dataCierre date)

ARTICULO (idArticulo integer, descripcion1 varchar (64), descripcion2 varchar (64), descripcion3 varchar (64), precio float, precioOferta float, tecla integer, unidad varchar (1), codInterno varchar (6) , ean varchar (13))



PLANTILLA (idPlantilla integer, tipo varchar (16), tamaño varchar (16))

CARTEL (idCartel integer, idPlantilla integer(FK), idArticulo integer(FK), dataIni date, dataFi date)

TRABAJO (idTrabajo integer, descripción varchar (64), prioridad integer, fechaCreación date)

CARTELES (idTrabajo integer(FK), idCartel integer(FK))

ESTADOTRABAJOS (idEstadoTrabajo integer, idTrabajo integer(FK), idUsuario integer(FK), estado varchar(1), fechaEstado date)

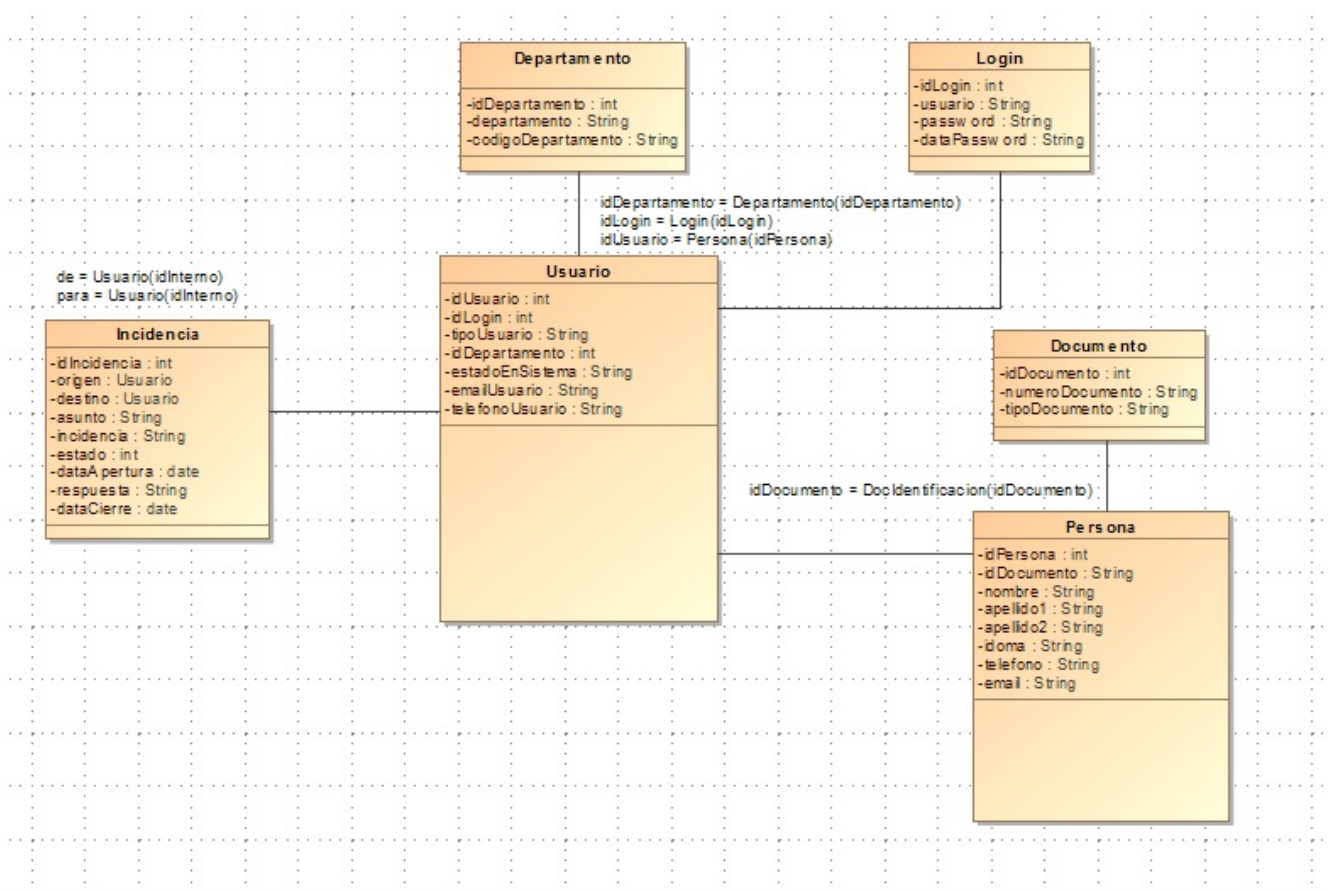


figura 29



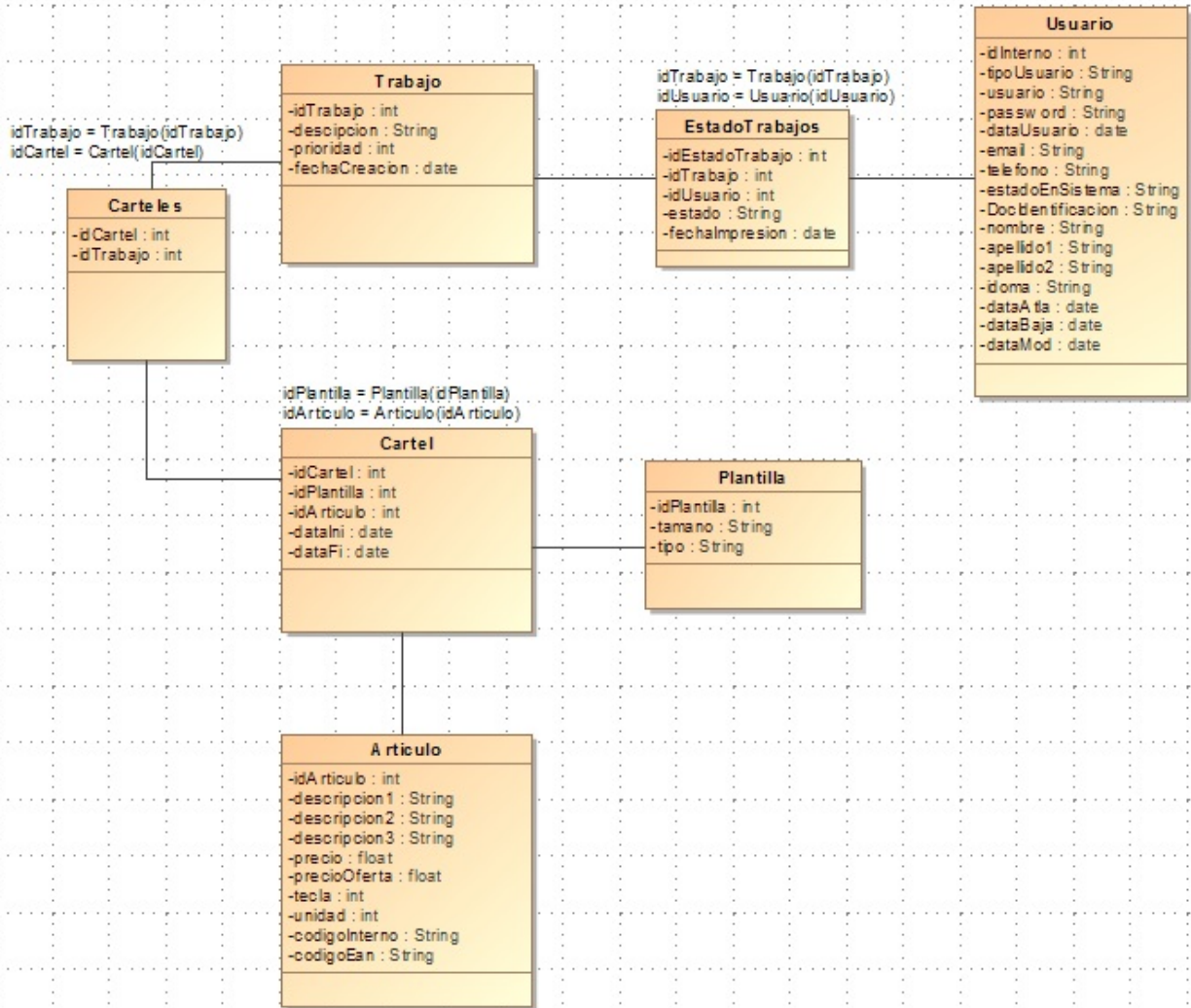


figura 30



4-Diagrames

En aquesta secció presentem alguns dels diagrames de seqüència, d'estats i de casos d'ús més rellevants.

4.1 Diagrames de casos d'ús i d'actors

Els diagrames de classes proporcionen una idea estàtica del sistema. Ara veurem als diagrames que estableixen una idea dinàmica, és a dir que canvien amb el temps.

Què són els casos d'ús?

És una estructura que ajuda els analistes a treballar amb els usuaris per determinar la forma en què es farà servir un sistema. Amb una col·lecció de casos d'ús es pot fer l'esbós d'un sistema en termes del que els usuaris intenten fer amb ell.

Aquest tipus d'anàlisi és crucial per a la fase d'anàlisi del desenvolupament d'un sistema. La manera com els usuaris utilitzin un sistema li dona la pauta per al que dissenyarà i crearà. Del que es tracta és d'obtenir un sistema que compleixi les nostres necessitats.

La importància dels casos d'ús és que es dissenya el sistema des del propi punt de vista de l'usuari. La idea és involucrar els usuaris en les etapes inicials de l'anàlisi i disseny del sistema.

4.1.1 Casos d'ús del Login

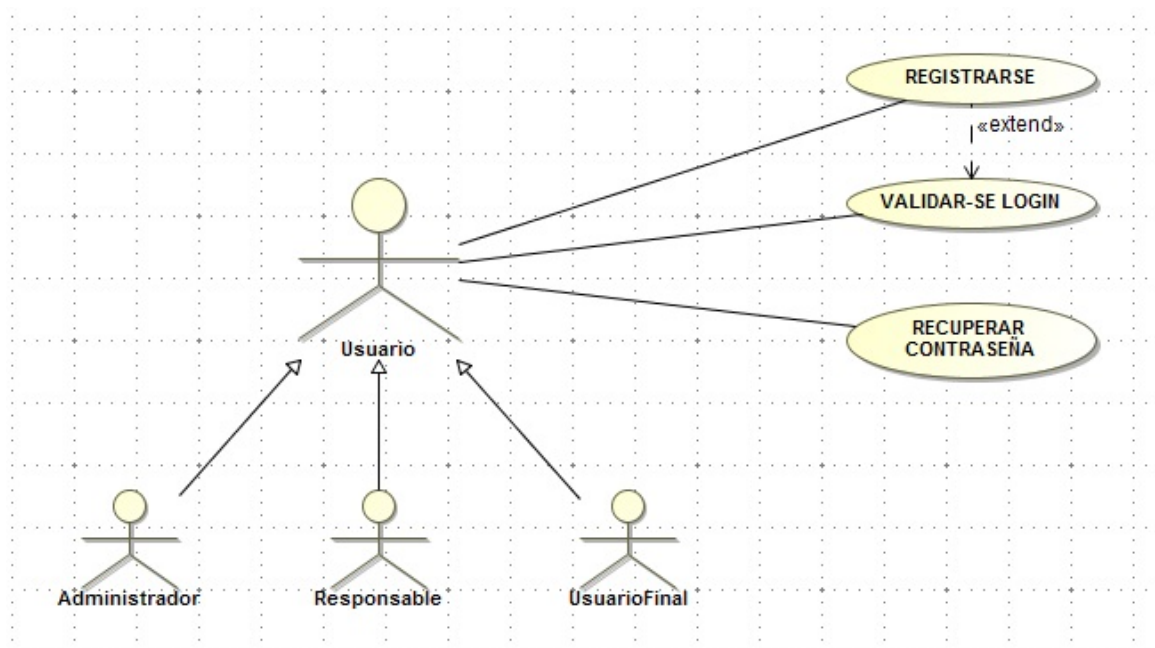


Figura 15



4.1.2 Casos d'ús de Incidències

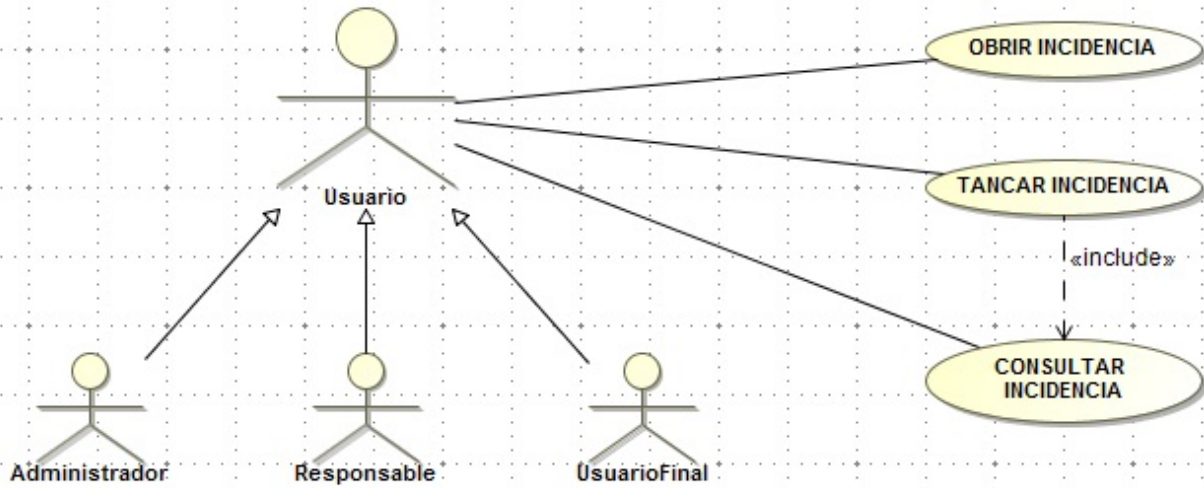


Figura 16

4.1.3 Casos d'ús de l'Administrador

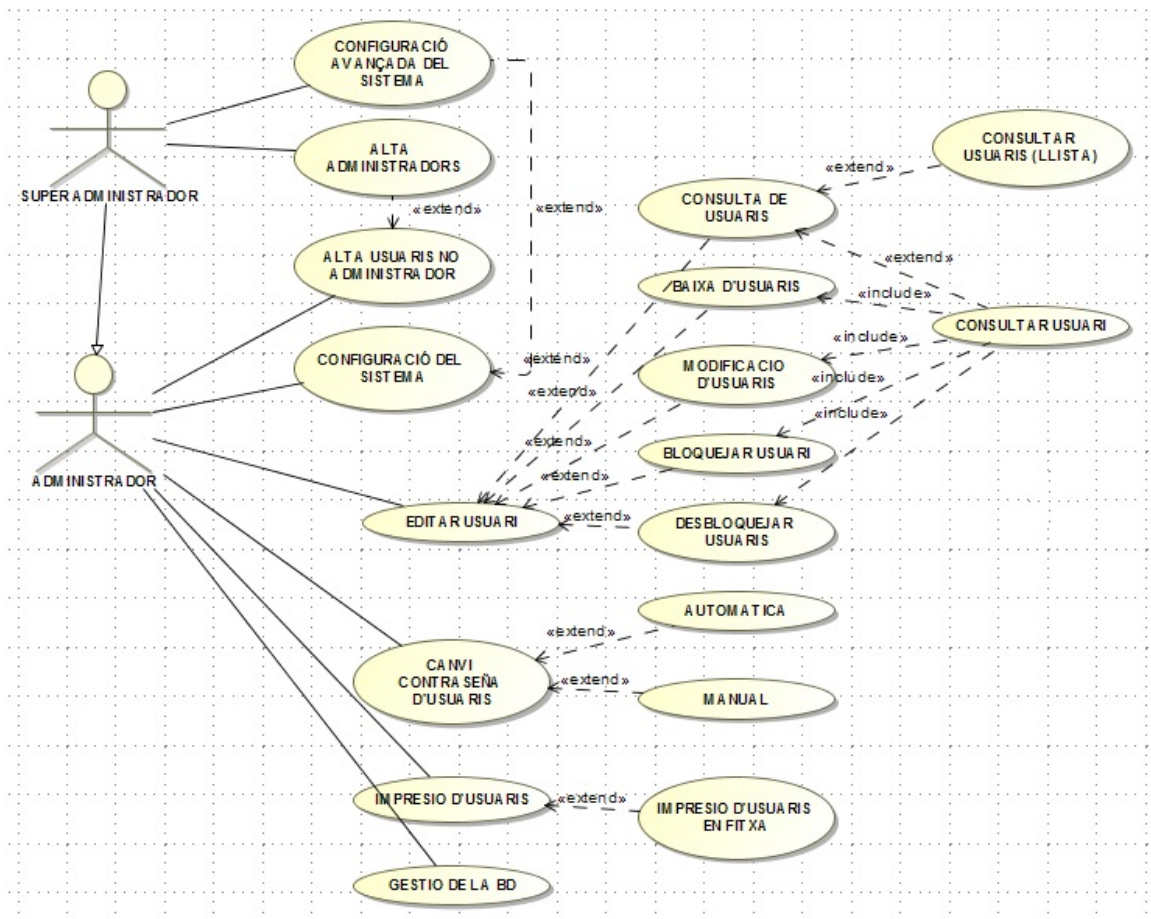


Figura 17

4.1.4 Casos d'ús del Responsable

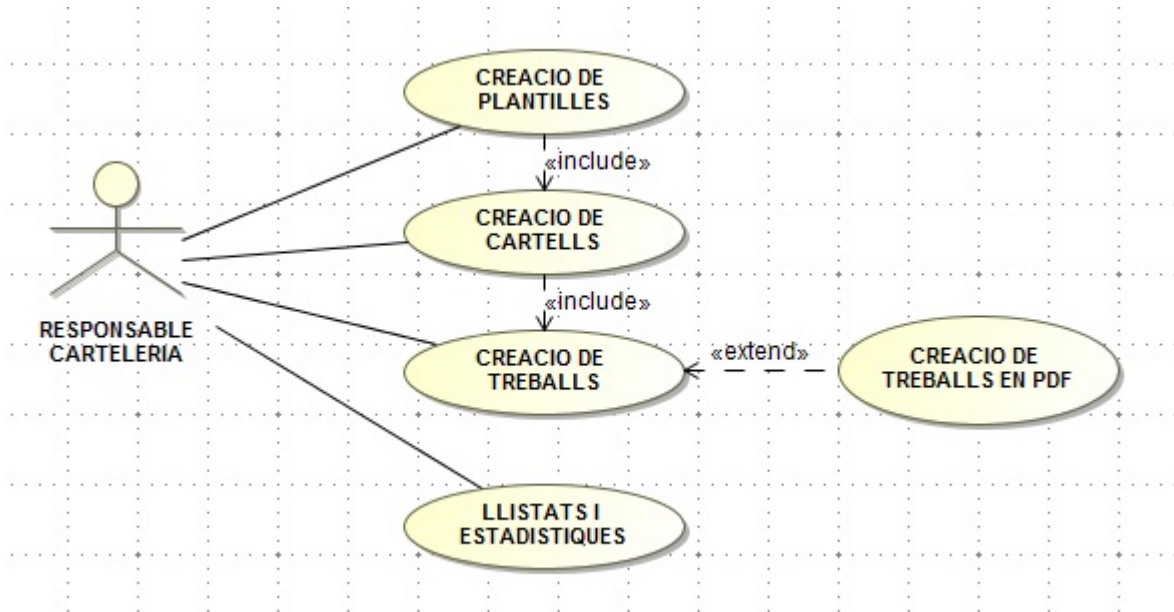


Figura 18

4.1.5 Casos d'ús de l'Usuari Final

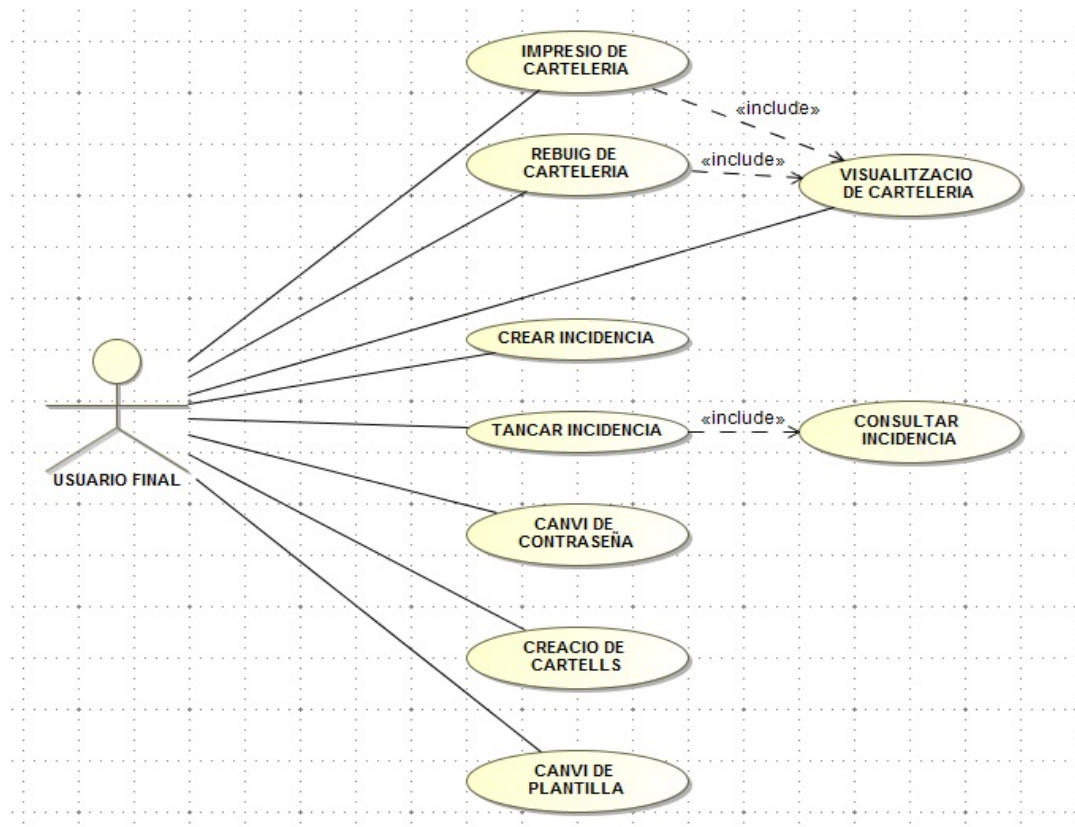


Figura 19



4.2 Diagrames de seqüència

Un diagrama de seqüència mostra la interacció d'un conjunt d'objectes en una aplicació a través del temps i es modela per a cada cas d'ús. Mentre que el diagrama de casos d'ús permet el modelatge d'una vista comercial de l'escenari, el diagrama de seqüència conté detalls d'implementació de l'escenari, incloent-hi els objectes i les classes que s'usen per implementar l'escenari, i missatges intercanviats entre els objectes.

Típicament s'examina la descripció d'un cas d'ús per determinar quins objectes són necessaris per a la implementació de l'escenari. Si es disposa de la descripció de cada cas d'ús com una seqüència de diversos passos, llavors es pot "caminar sobre" aquests passos per descobrir quins objectes són necessaris perquè es puguin seguir els passos. Un diagrama de seqüència mostra els objectes que intervenen en l'escenari amb línies discontinües verticals, i els missatges passats entre els objectes com a fletxes horitzontals.

Els missatges es dibuixen cronològicament des de la part superior del diagrama a la part inferior; la distribució horitzontal dels objectes és arbitrària. Durant l'anàlisi inicial, el modelador típicament col·loca el nom comercial d'un missatge en la línia del missatge. Més tard, durant el disseny, el nom comercial és reemplaçat amb el nom del mètode que està sent anomenat per un objecte en l'altre. El mètode anomenat, o invocat, pertany a la definició de la classe instanciada per l'objecte en la recepció final del missatge.

En Aquest cas, es creen els diagrames de seqüència que poden ser més significatius, com per exemple, els mètodes de login, accessos a la bd com afegir i eliminar i per últim la finalitat del sistema que es imprimir treballs

4.2.1 Seqüència de Login

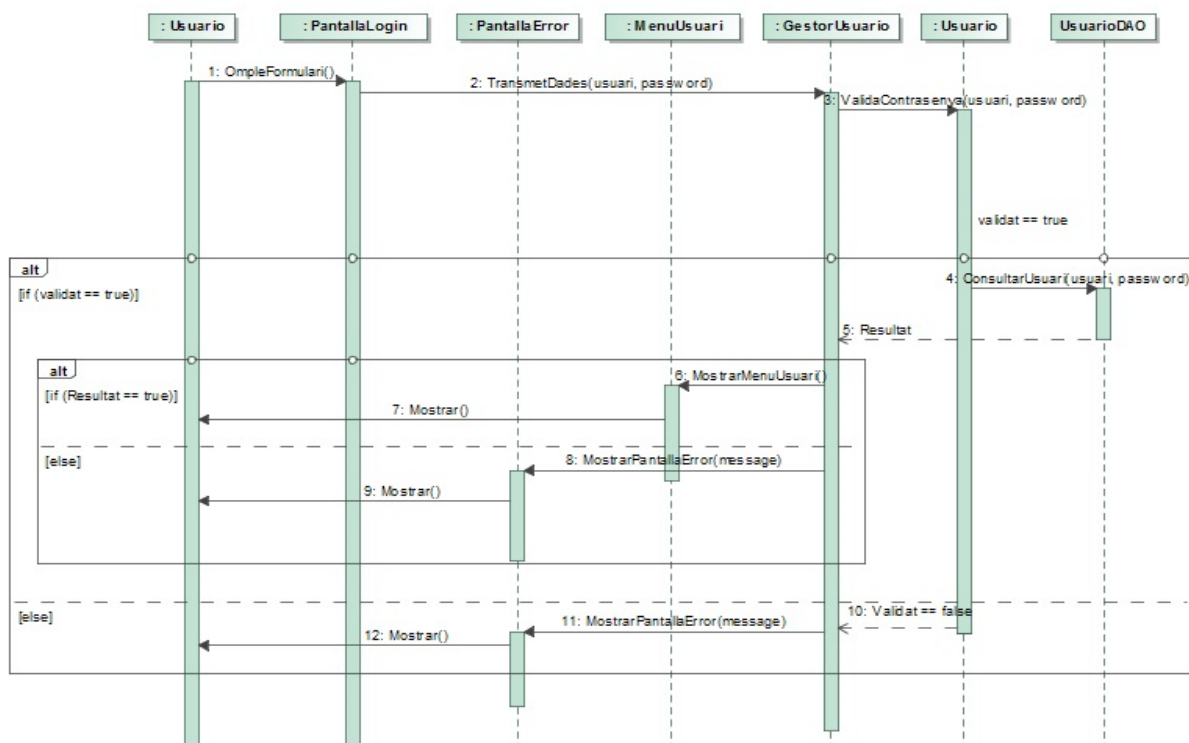


Figura 20

4.2.2 Seqüència de Afegir Usuari

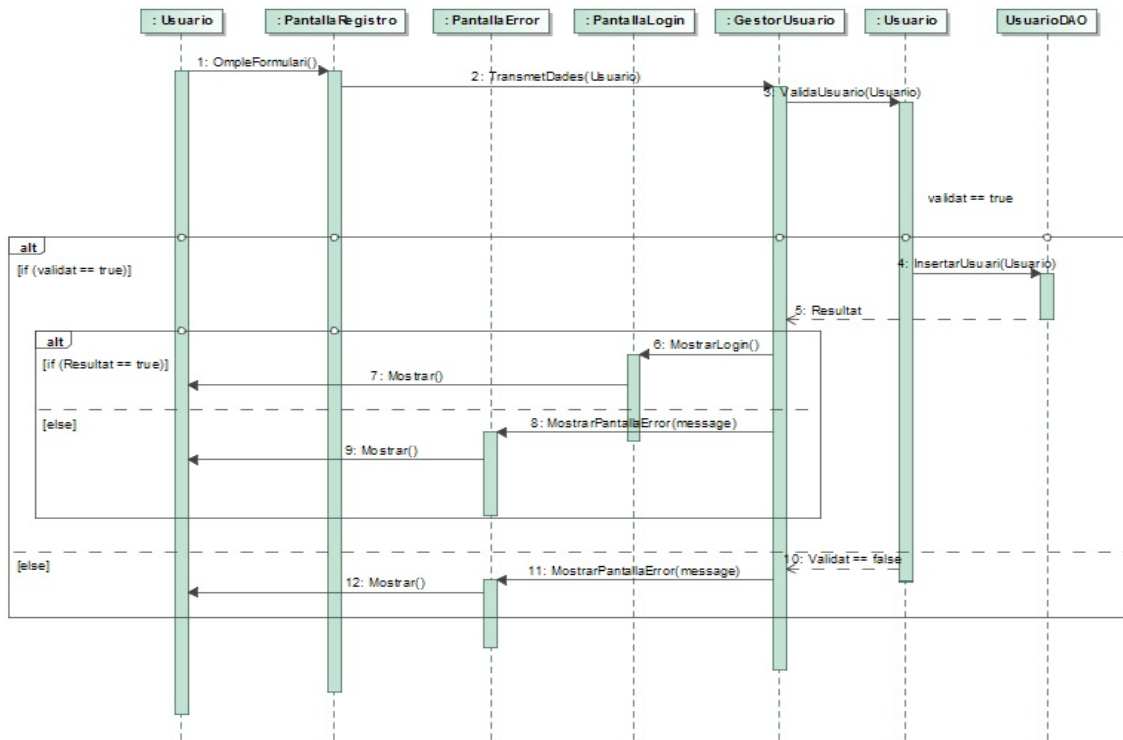


Figura 21

4.2.3 Seqüència de Eliminar Usuari

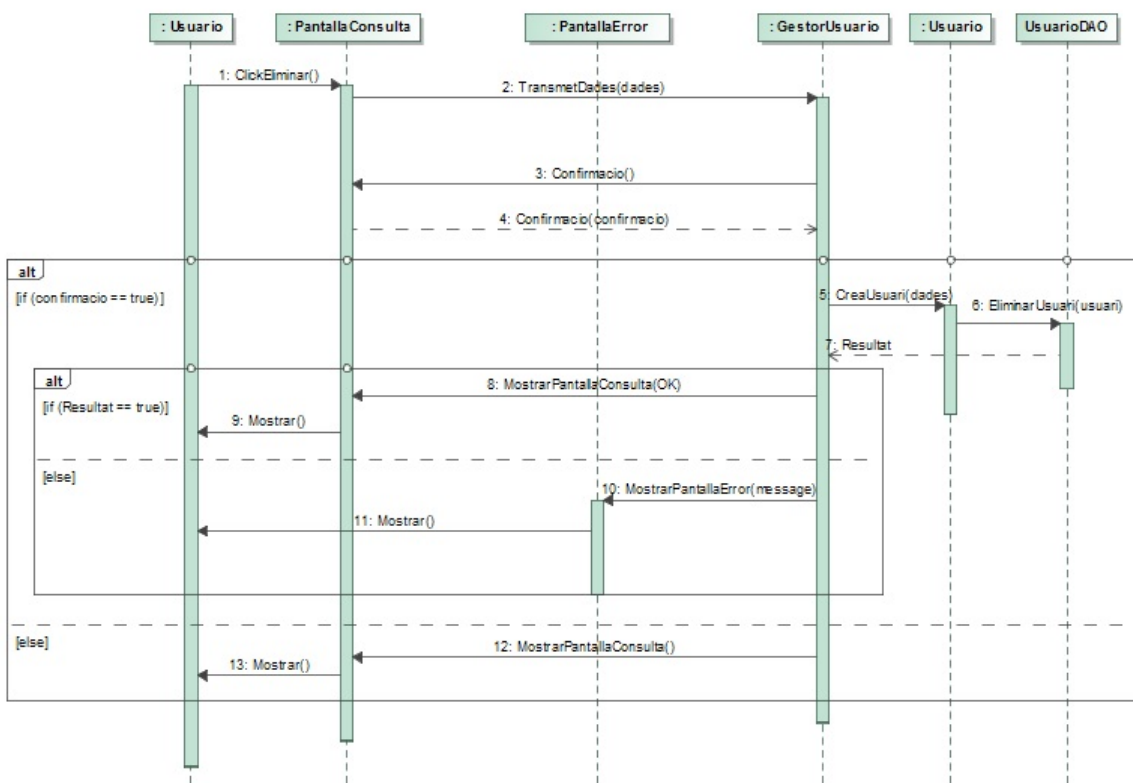


Figura 22



4.2.4 Seqüència Imprimir Treball

En aquest cas, amb la finalitat de simplificar el diagrama amb el que ens interessa, es té en compte que a la pantalla de treballs no impresos ja se fa la consulta de tots els treballs que no ha imprès l'usuari i la pantalla de error produïda per un error en el EstadoTrabajosDAO.

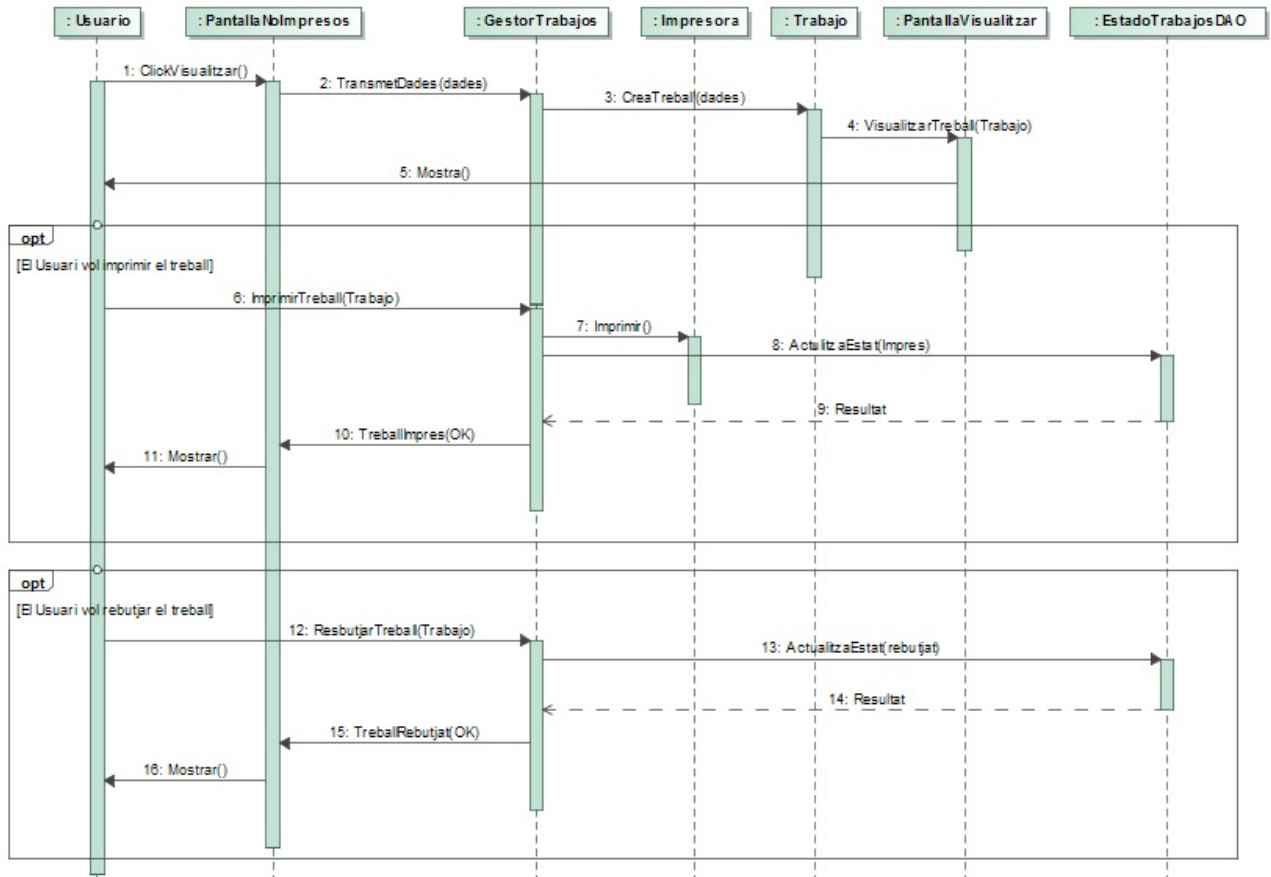


Figura 23



4.3 Diagrames d'estats

Un diagrama d'estat és un tipus de diagrama utilitzat en informàtica i àrees similars per descriure el comportament de sistemes. Els diagrames d'estat requereixen que el sistema descrit estigui compost d'un nombre finit d'estats.

En quan a la representació, un diagrama d'estats és un gràfic que els nodes del qual són estats i els arcs dirigits del qual són transicions etiquetades amb els noms dels esdeveniments.

Un estat es representa com una caixa arrodonida amb el nom de l'estat en el seu interior. Una transició es representa com una fletxa des de l'estat origen a l'estat destí

En UML, un diagrama d'estats es un diagrama utilitzat per identificar cadascuna de les rutes o camins que prendre un flux de informació després d'efectuar-se cada procés. Permet identificar sota que arguments s'executa cadascun dels processos i en que moment podran tindre una variació. El diagrama d'estats permet visualitzar d'una forma seqüencial l'execució de cadascun dels processos.

4.3.1 Estats de l'Usuari

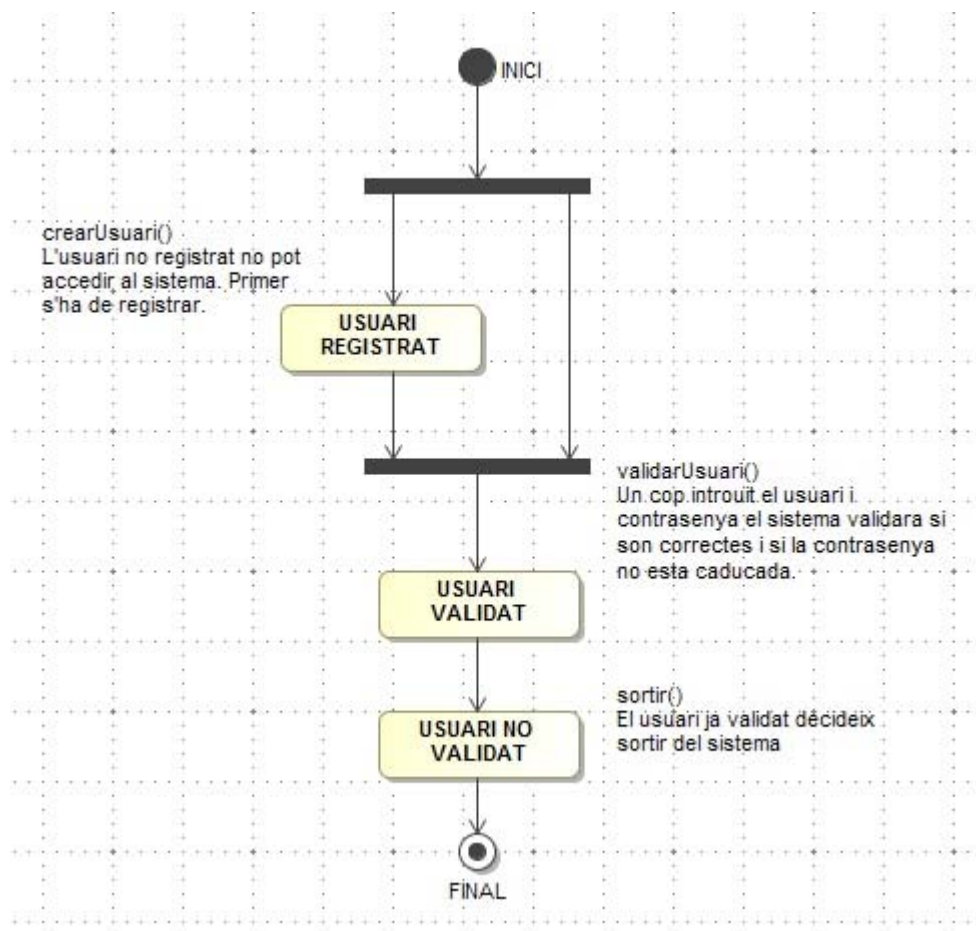


Figura 24



4.3.2 Estats d'Incidència

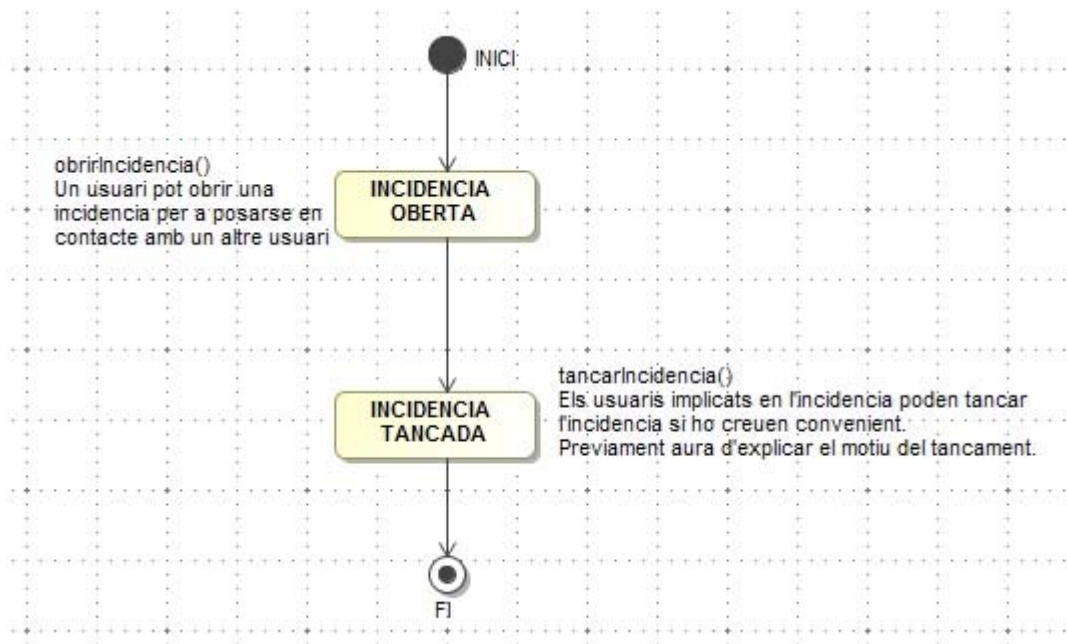


Figura 25

4.3.3 Estats de Treball

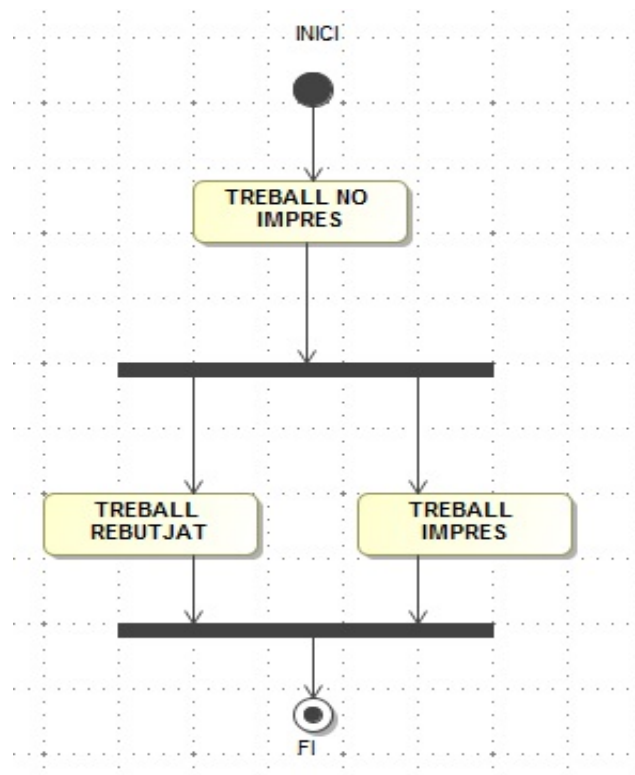


Figura 26



5-Patro MVC

Té els seus fonaments en la tecnologia Smalltalk. És un dels patrons més utilitzats per desenvolupar aplicacions web. Especifica com ha de ser estructurada una aplicació web, les capes que la formen i la funcionalitat de cadascuna. Permet separar la part lògica de la part de presentació d'una aplicació web.

L'arquitectura MVC, està dividida en tres tipus de components segons la seva responsabilitat en el sistema:

- **Component Model:** Representa l'estat intern de l'aplicació. És la representació específica de la informació amb la qual opera el sistema. Conté tot els objectes i regles de negoci per implementar la lògica de negoci de l'aplicació. Mostra la funcionalitat de l'aplicació.
- **Component Vista:** Presenta el model en un format adequat per interactuar. És el component de presentació de dades a l'usuari. Permet a l'usuari interactuar amb la lògica de negoci. Pot recuperar informació del model però no pot modificar-la (aquestes peticions es dirigeixen al component controlador). Rebrà les dades del model que mostrarà a l'usuari.
- **Component Controlador:** És el cervell de l'aplicació. És el responsable del control de flux i tractar la interacció amb l'usuari. Respon a les entrades de l'usuari, traduint events d'interfície d'usuari en crides al model i a la vista. Defineix com reacciona la interfície d'usuari davant d'aquests canvis.

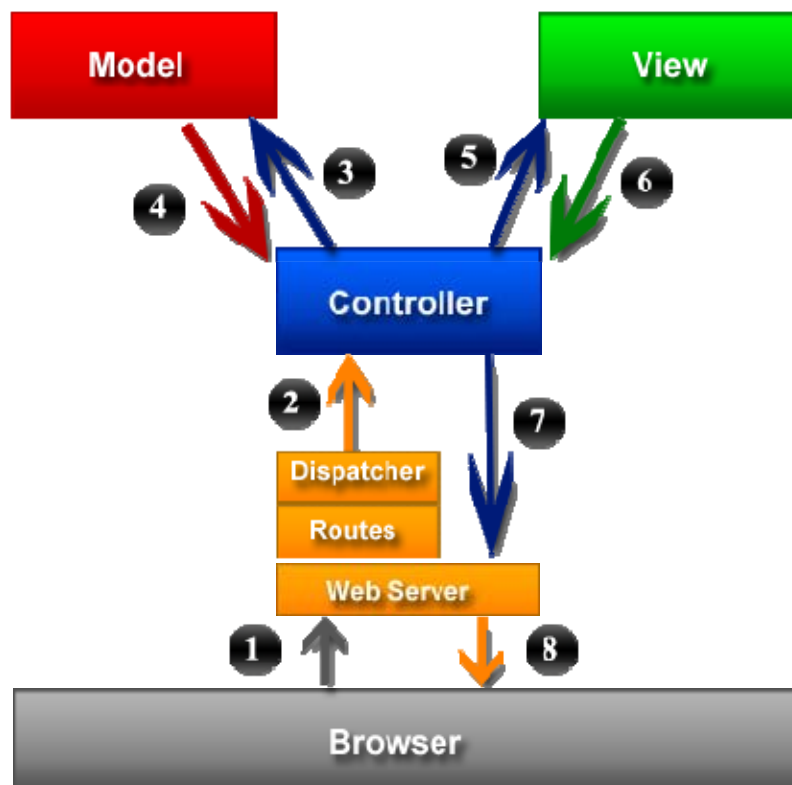


Figura 27



En la imatge anterior es veu clarament com es el funcionament dels projectes en MVC.

Avantatges que presenta el patró MVC:

1-Reutilització dels components :

- Podem modificar completament el disseny d'una aplicació web, sense afectar al Model.
- Podem realitzar canvis al Model, sense que sigui necessari modificar el disseny de la interfície d'usuari.
- La lògica de negoci mai dependrà d'altre component que no sigui el seu (component model), permetent la seva execució des de qualsevol entorn que implementi el model de Vista i Controlador.

2-Permet separació de responsabilitats:

- Els programadors i els dissenyadors gràfics, poden treballar independentment, tan sols respectant uns mínims requisits.

3-La separació de la lògica de negoci de la de presentació facilita en gran part:

- La documentació del codi.
- El manteniment de l'aplicació.

Per implementar la filosofia del patró de disseny MVC s'ha fet servir el framework Struts 2 Apache i el framework Hibernate 3 per a la connexió amb la base de dades.

5.1-Frameworks del Projecte

implementar la filosofia del patró de disseny MVC s'ha fet servir el framework Struts 2 Apache i el framework Hibernate.

Framework Struts 2

Apache Struts 2 és un marc elegant i extensible per a la creació d'empreses preparades per a aplicacions web Java. El marc està dissenyat per agilitzar el cicle de desenvolupament complet, des de la construcció, la implementació, el manteniment d'aplicacions a través del temps.

Apache Struts 2 va ser originalment conegut com Webwork 2. Després de treballar de forma independent durant diversos anys, les comunitats Webwork i Struts unit les seves forces per crear Struts2. Aquesta nova versió de Struts és més fàcil d'usar i més de com Struts sempre va haver de ser.

Struts és una eina suport per el desenvolupament d'aplicacions Web sota el patró MVC sota la plataforma Java EE (Java Enterprise Edition). Struts si desenvolupava com a instància de part del projecte Jakarta de l'Apache Software Foundation, però actualment es un projecte independent conegut com Apache Struts.

Struts permet reduir el temps de desenvolupament.

Amb la versió 2 del marc s'han introduït algunes millores sobre la primera versió, de cara a simplificar les tasques més comuns en el desenvolupament d'aplicacions Web, Així com millorar la Integració



amb AJAX, etc

Struts es basa en el patró d'arquitectura de programari Model-Vista-Controlador (MVC) el Com si s'utilitza àmpliament i és considerat de gran solidesa. D'acord amb el Framework, el processament en si separació en tres seccions diferenciades Trucades el Model, les vistes i el Controlador.

Framework Hibernate 3

Com totes les eines del seu tipus, Hibernate busca solucionar el problema de la diferència entre els dos models de dades coexistents en una aplicació: l'usat en la memòria de la computadora (orientació a objectes) i l'usat en les bases de dades (model relacional).

Per aconseguir això permet al desenvolupador detallar com és el seu model de dades, quines relacions existeixen i quina forma tenen. Amb aquesta informació Hibernate li permet a l'aplicació manipular les dades a la base de dades operant sobre objectes, amb totes les característiques de la POO.

Hibernate convertirà les dades entre els tipus utilitzats per Java i els definits per SQL. Hibernate genera les sentències SQL i allibera-la al desenvolupador del maneig manual de les dades que resulten de l'execució d'aquestes sentències, mantenint la portabilitat entre tots els motors de bases de dades amb un lleuger increment en el temps d'execució.

Està dissenyat per ser flexible quant a l'esquema de taules utilitzat, per poder adaptar-se al seu ús sobre una base de dades ja existent. També té la funcionalitat de crear la base de dades a partir de la informació disponible.

Hibernate ofereix també un llenguatge de consulta de dades anomenat HQL (Hibernate Query Language), al mateix temps que una API per construir les consultes programàticament (coneguda com "Criteria").

Hibernate necessita saber com llegir i emmagatzemar objectes d'una classe persistent. Aquí és on l'arxiu de mapeig de Hibernate entra en joc. L'arxiu de mapeig indica a Hibernate que taula a la base de dades ha de ser accedida, i que columnes en aquesta taula s'han d'usar.



Els arxius de mapeig són els arxius que utilitza Hibernate per a relacionar la memòria del pc amb la base de dades.

Cal configurar-los i estructurar-los per a aquest fi. Un exemple:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<!-- Generated 14-abr-2013 23:40:15 by Hibernate Tools 3.2.1.GA -->
<hibernate-mapping>
  <class catalog="tfc" name="model.entities.Tipo" table="tipo">
    <id name="idTipo" type="java.lang.Integer">
      <column name="idTipo"/>
      <generator class="identity"/>
    </id>
    <property name="tipo" type="string">
      <column length="16" name="tipo" not-null="true" unique="true"/>
    </property>
  </class>
</hibernate-mapping>
```

En aquest cas el que estem indicant a Hibernate és que la classe Tipo (ubicada en el Package model.entities.Tipo) està relacionada amb la taula "tipo" de la bd "tfc". Que aquesta taula té una columna idTipo auto incremental i que està relacionada amb el atribut idTipo de la classe Tipo.

S'expressa també, de igual manera la relació entre el camp tipo i l'atribut tipo.



Apache Tomcat és una implementació de codi obert del programari de Java Servlet i JavaServer Pages technologies. El Servlet Java i especificacions JavaServer Pages són desenvolupats sota el Java Community Process.

Apache Tomcat es desenvolupa en un entorn obert i participatiu i publicat sota la llicència Apache versió 2.

Tomcat és un servidor web amb suport de servlets i JSPs. Inclou el compilador Jasper, que compila JSPs convertint-los en servlets.

El motor de servlets de Tomcat sovint es presenta en combinació amb el servidor web Apache o altres servidors web.

Tomcat pot funcionar com servidor web per si mateix. En els seus inicis, va existir la percepció que l'ús de Tomcat de forma autònoma era només recomanable per a entorns de desenvolupament i entorns amb requisits mínims de velocitat i gestió de transaccions. Avui dia ja no existeix aquesta percepció, i Tomcat és usat com servidor web autònom en entorns amb alt nivell de tràfic i alta disponibilitat.

Com Tomcat va ser escrit en Java, això fa que funcioni en qualsevol sistema operatiu que disposi de la JVM.

XWORK Validator

Struts 2 es basa en un marc de validació proporcionada per XWork per permetre l'aplicació de les regles de validació d'entrada de les seves accions abans de ser executats.

Struts2 Validator Framework ens permet separar la lògica de validació de l'actual Java / JSP codi, on es pot revisar i modificar fàcilment més tard.

La validació Framework Struts2 alleuja gran part del mal de cap associat amb el maneig de la validació de dades, el que li permet centrar-se en el codi de validació i no en la mecànica de la captura de dades i nova visualització de dades incompletes o no vàlids .

El marc de validació ve amb un conjunt de rutines útils per manejar la validació de formularis de forma automàtica i es pot gestionar tant del costat del servidor, així com la validació de formularis del costat del client Si certa validació no està present, pot crear la seva pròpia lògica de validació mitjançant l'aplicació java interfície com.opensymphony.xwork2.Validator.

Validator utilitza fitxers de configuració XML per determinar què s'ha d'instal·lar les rutines de validació i com han de ser aplicats per a una aplicació donada

Si el fitxer validator.xml no està present en la ruta de classes, un arxiu de validació per defecte es carrega des de la ruta com/opensymphony/xwork2/validator/validators/default.xml.

iTEXT Framework



iText és una biblioteca Open Source per crear i manipular arxius PDF, RTF, i HTML en Java. Va ser escrita per Bruno Lowagie, Paulo Soares, i altres, i està distribuïda sota la llicència Affero General Public Llicència-se.

El mateix document pot ser exportat en múltiples formats, o múltiples instàncies del mateix format. Les dades poden ser escrits a un fitxer o, per exemple, des d'un servlet a un navegador web.

Més recentment, ha estat estesa a una biblioteca PDF de propòsit general, capaç d'omplir formularis, moure pàgines d'un PDF a un altre, i altres coses. Aquestes extensions són sovint mútuament excloents. Una classe et permet omplir en formularis, mentre una classe diferent i incompatible fa possible copiar pàgines d'un PDF a un altre.

El suport de PDF d'iText és, però, bastant extensiu. Això suporta signatures basades en PKI de PDF, xifrat de 40 bits i 128 bits, correcció de colors, PDF / X, gestió de colors per perfils ICC, i és amfitriona d'altres característiques.

8-Annexos

<http://struts.apache.org/development/2.x/>

http://ca.wikipedia.org/wiki/Apache_Struts

<http://es.wikipedia.org/wiki/Hibernate>

<http://www.onjava.com/pub/a/onjava/2007/02/08/an-introduction-to-hibernate-3-annotations.html>

