

# Carteleria Web



**TFC – J2EE**

**Juan Ramón Segura Ponce**

## Introducció al J2EE

---

### Perquè la tecnologia J2EE

- Les aplicacions web són populars gràcies a la ubiqüitat del client
- Només es necessita un navegador i se'l classifica com client lleuger.
- La seva capacitat de mantenir i actualitzar aplicacions web sense instal·lar-les sobre una quantitat indeterminada d'ordinadors client.
- En accés a dades s'utilitza la potencia del servidor simplement desde un navegador Web
- Java, com a llenguatge de desenvolupament amb l'arquitectura J2EE s'ha convertit en un estàndard en el món de la indústria per la realització d'aplicacions empresarials a Internet.



## Justificació i context

---

### Context

- El treball fi de carrera és una assignatura pensada per realitzar un projecte de síntesi dels coneixements adquirits en altres assignatures i on han de ser posats en pràctica bona part d'aquests coneixements obtinguts.

### Justificació

- El TFC consisteix en realitzar l'anàlisi, el disseny i la implementació d'una aplicació en una interfície web mitjançant la tecnologia Java i J2EE.

### Motivació

- La principal motivació com a estudiant amb certs coneixements de Java, és dissenyar e implementar un exemple típic d'una aplicació web distribuïda, mitjançant la tecnologia J2EE i seguint els patrons de disseny MVC, sense tenir experiència prèvia en aquesta arquitectura.



## L'objectius general de l'àrea

---

- Estudiar els frameworks i estàndards disponibles per a les diferents capes de l'aplicació, tant els definits per l'estàndard J2EE com els desenvolupats per la comunitat amb llicència lliure (Struts 2, Hibernate, o d'Apache).
- 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.
- Perfilar els aspectes d'usabilitat i dissenyar una interfície web pràctica i intuïtiva tenint en compte el context web de comerç en línia en el que ens trobem.
- Mantenir separada la lògica de negoci de la persistència de dades.
- garantir les transaccions realitzades per diferents usuaris en concurrència.
- Afegir seguretat a l'aplicació mitjançant mecanismes d'autenticació establint els diferents rols d'usuari.



## Productes Obtinguts

---

- 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.

Els productes finals son:

- **jsegurapon\_producto.zip**: fitxers fonts, compilats, test..etc
- **jsegurapon\_memoria.pdf**: Fitxer amb la memoria
- **jsegurapon\_presentació.ppt**: Presentació power point.
- **Jsegurapon\_instalación.pdf**: Arxiu explicatiu de la instal·lacio del soft



## Resum 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

### Una aplicació de tres capes

- El navegador Web és la primera capa, des del punt de vista de l'usuari.
- La segona capa es el motor que usa una tecnologia de contingut JSP/Java
- La tercera capa és un servidor de bases de dades.

El navegador web envia peticions a la capa intermitja (controller), que les serveix fent queries i actualitzacions contra la base de dades i genera una resposta a l'usuari.

- S'ha fet servir el següents frameworks
  - Servidor Apache tomcat
  - Hibernate 3
  - Struts 2
  - Xwork Validator
  - iText



## Avantatges del patró MVC

---

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:

Avantatges que presenta el patró MVC:

### **Reutilització dels components :**

- Podem modificar el disseny d'una aplicació web, sense afectar al Model.
- Podem realitzar canvis al Model, sense afectar el disseny de la interfície.
- La lògica de negoci mai dependrà d'altre component que no sigui el seu.

### **Permet separació de responsabilitats:**

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

### **La separació de la lògica de negoci de la de presentació facilita:**

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



## Frameworks 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

### Una aplicació de tres capes

- El navegador Web és la primera capa, des del punt de vista de l'usuari.
- La segona capa es el motor que usa una tecnologia de contingut JSP/Java
- La tercera capa és un servidor de bases de dades.

El navegador web envia peticions a la capa intermitja (controller), que les serveix fent queries i actualitzacions contra la base de dades i genera una resposta a l'usuari.

- S'ha fet servir el següents frameworks
  - Servidor Apache tomcat
  - Hibernate 3
  - Struts 2
  - Xwork Validator
  - iText





## Frameworks del projecte

---

### **Servidor Apache tomcat**

Tomcat és un servidor web amb suport de servlets i JSPs. Inclou el compilador Jasper, Compila JSPs en servlets. Tomcat pot funcionar com servidor web per si mateix. Com Tomcat esta escrit en Java, funciona en qualsevol sistema que disposi de la JVM.

### **Hibernate 3**

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

### **Struts 2**

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.

### **Xwork Validator**

Es un marc de validació proporcionat per permetre l'aplicació de les regles de validació d'entrada de les seves accions abans de ser executats.

### **iText**

iText és una biblioteca Open Source per crear i manipular arxius PDF, RTF, i HTML en Java



## Actors del Sistema

---

Els actors són totes les entitats externes al sistema que guarden una relació amb. 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 la instal·lació i condicions inicials del sistema, a més a més donar d'alta almenys a un administrador del sistema.

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

**Responsable Carteleria:** És l'usuari que s'encarregarà de crear els treballs (cartells).

**Usuari final:** És l'usuari que accedirà al sistema per a imprimir la carteleria creada de forma centralitzada per el responsable de carteleria.



## Subsistemes del projecte

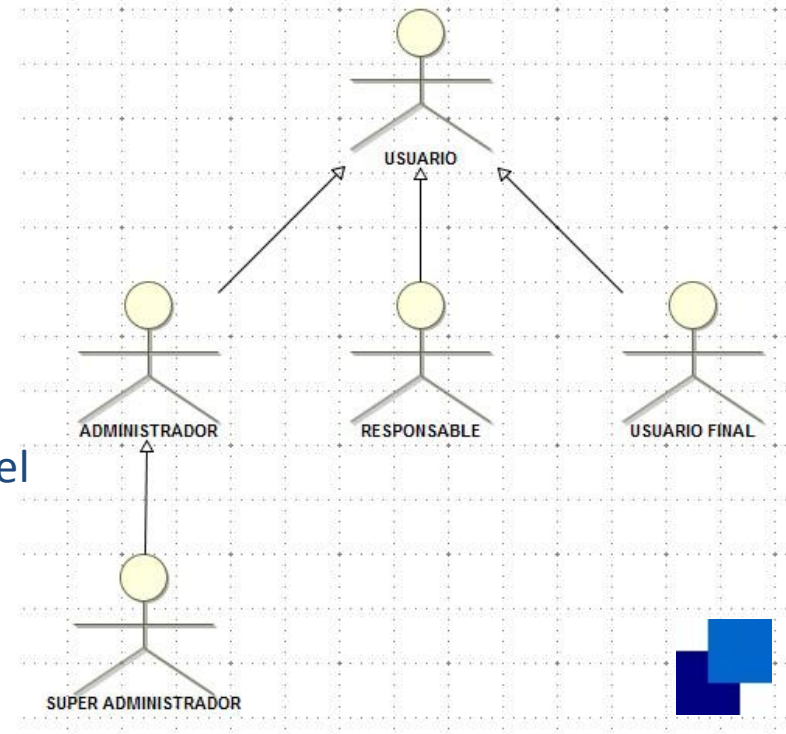
En l'àmbit del desenvolupament d'aplicacions web sota l'arquitectura J2EE.

Un aspecte especialment interessant de la tecnologia J2EE és que proporciona un model complert per suportar arquitectures distribuïdes, així s'aconsegueix que una cadena de establiments puguin fer us d'aquest programari.

**Aquest projecte te funcionalitats (subsistemes) clarament diferenciades, aquests son:**

- **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 pocs responsables de carteleria i un nombre indeterminat de usuaris finals.



## Subsistema de manteniment i conexio

---

**\*Tots els apartats tenen Inserció, modificació, eliminació i consulta.**

Les principals funcionalitats d'aquest subsistema són:

- Identificació i validació de contrasenya (Pantalla login)
  - Comprovació de contrasenya
  - Comprovació de caducitat de la contrasenya
- 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



## Subsistema de creacio de carteleria

---

El subsistema de creació de carteleria permetrà la creació de cartells basat en plantilles que després seran agrupats en llistes de impressió.

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 carteleria.



## Subsistema de Impresio de Carteleria

---

El subsistema de impressió de carteleria permetrà la impressió dels treballs creats i enviats per el responsable de carteleria.

Les principals funcionalitats d'aquest subsistema són:

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

Bàsicament aquest subsistema serà utilitzat per els usuaris finals.

El sistema consta de tres llistes d'impresio, treballs impresos , treballs no impresos i treballs rebutjats.

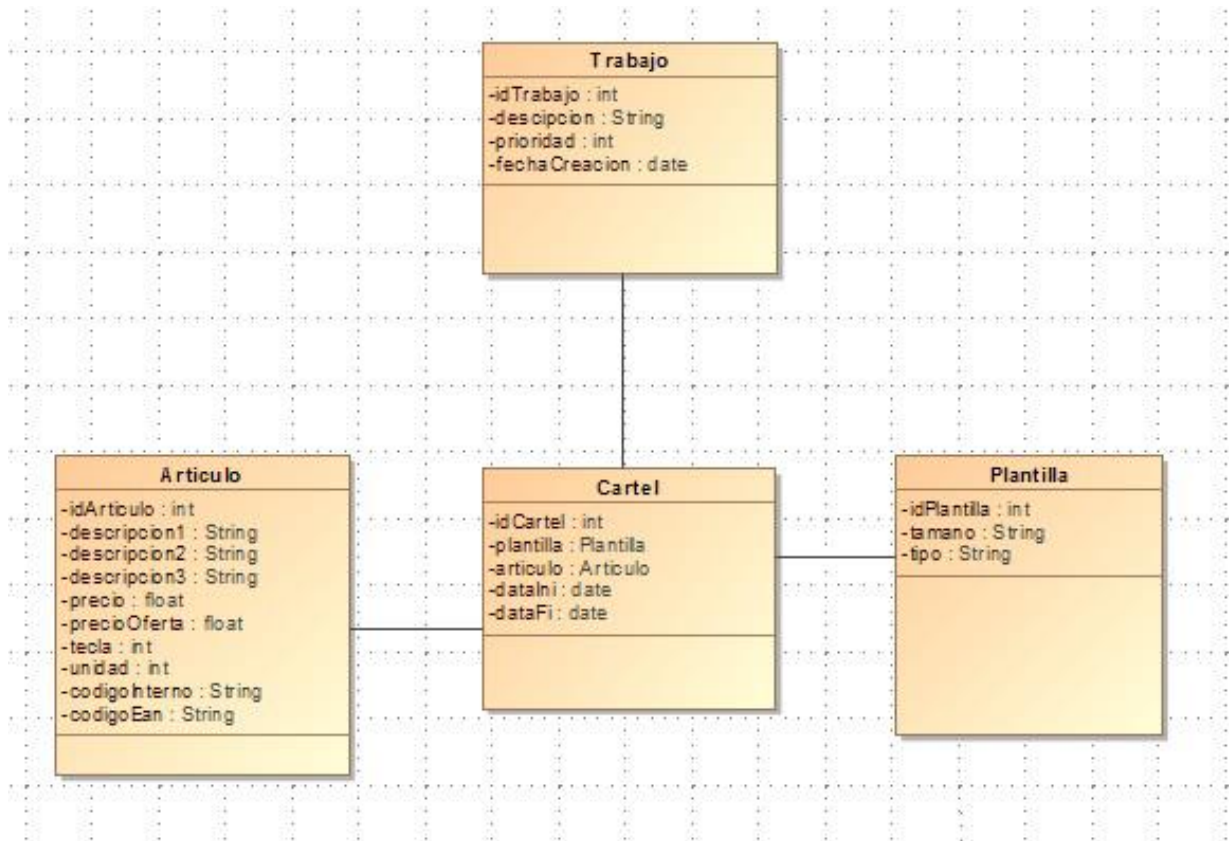
La llista de treballs no impresos serà on l'usuari trobarà els treballs nous que vagin creant els responsables de carteleria.



## Diagrama de classes Entities

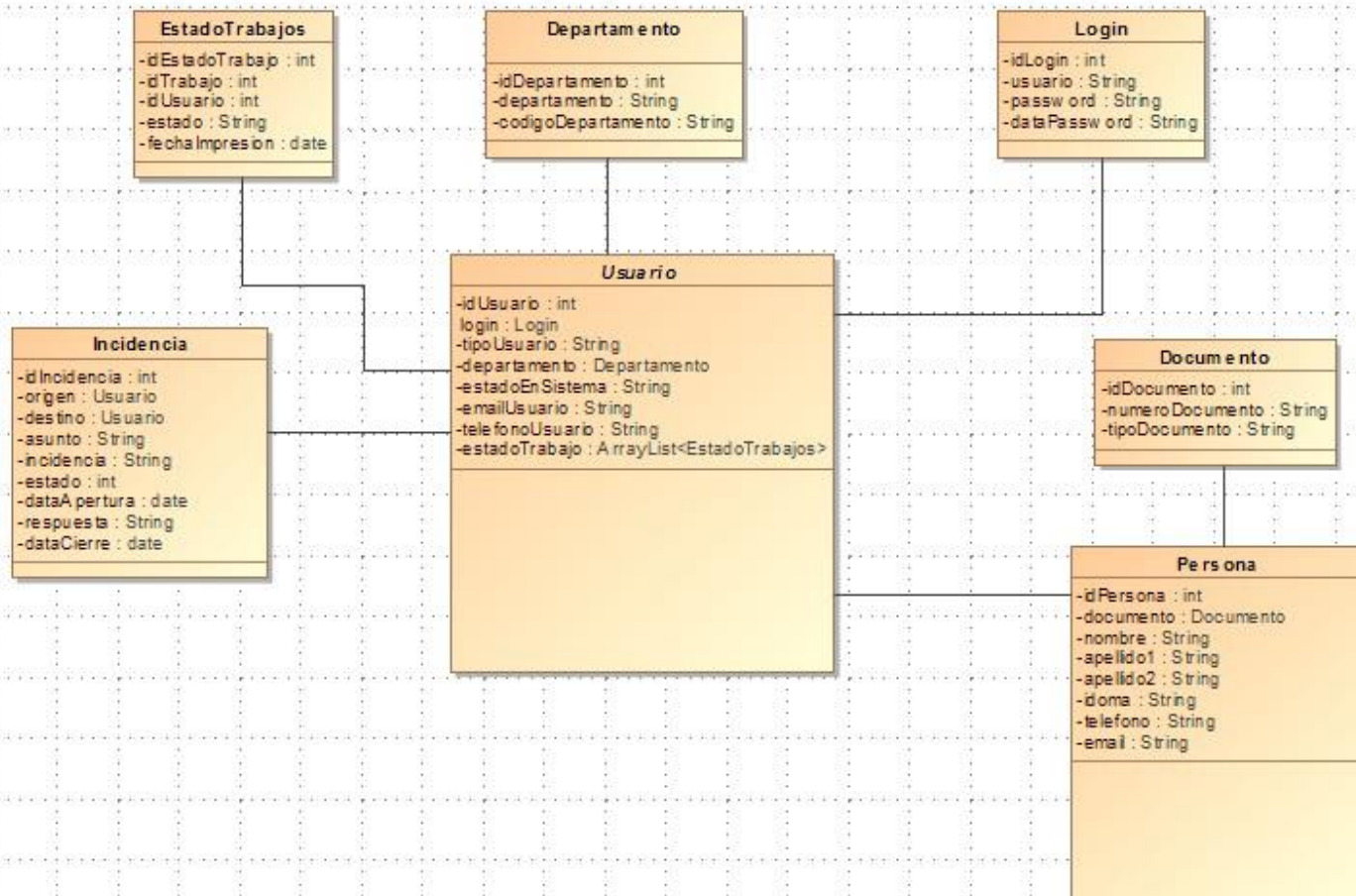
Les classes estan dividides en 2 grups, les classes que tenen relació amb els usuaris, i les classes que tenen relació amb els cartells.

Les classes relacionades amb Cartell son:



## Diagrama de classes Entities

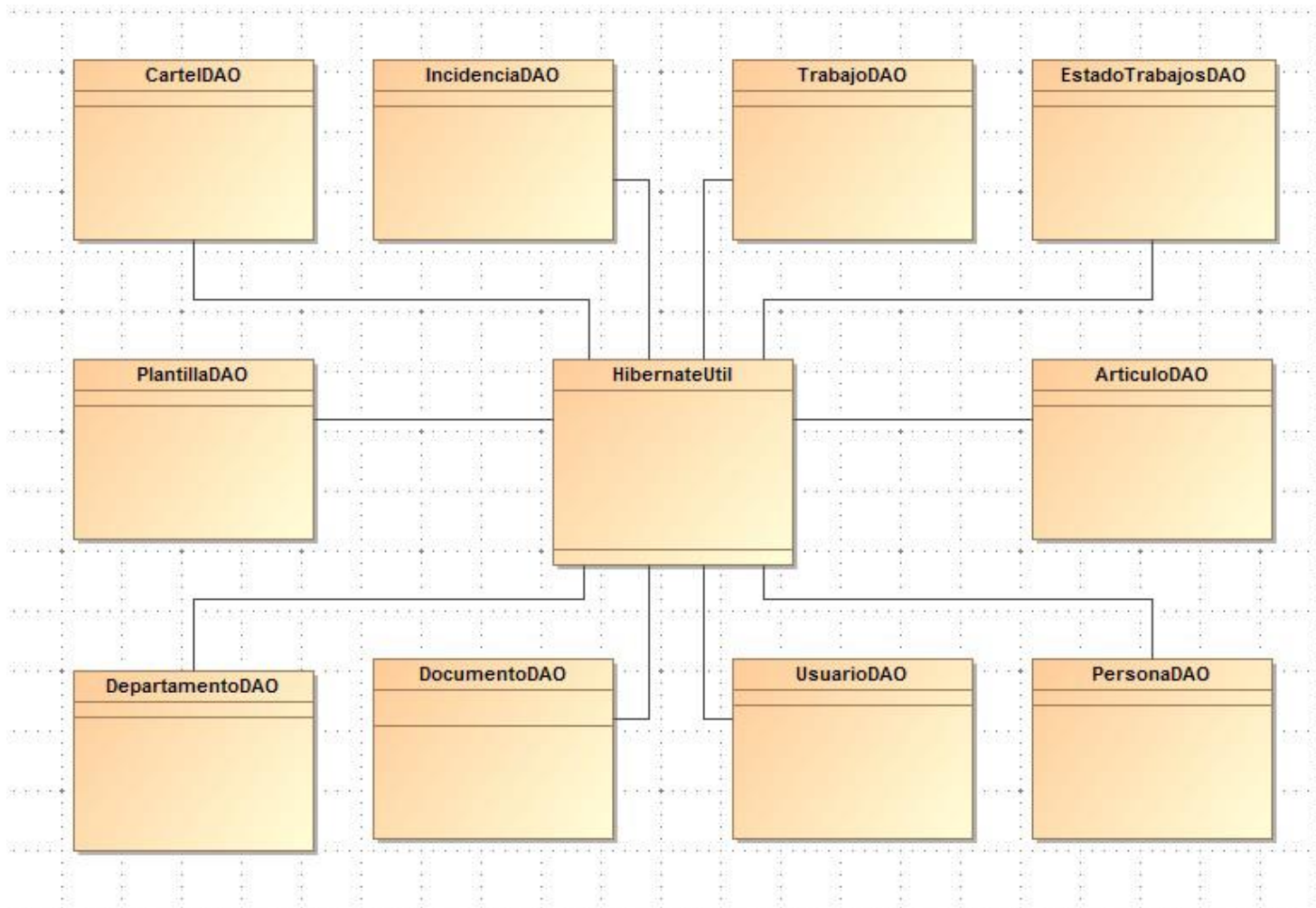
Les classes relacionades amb Usuari son:





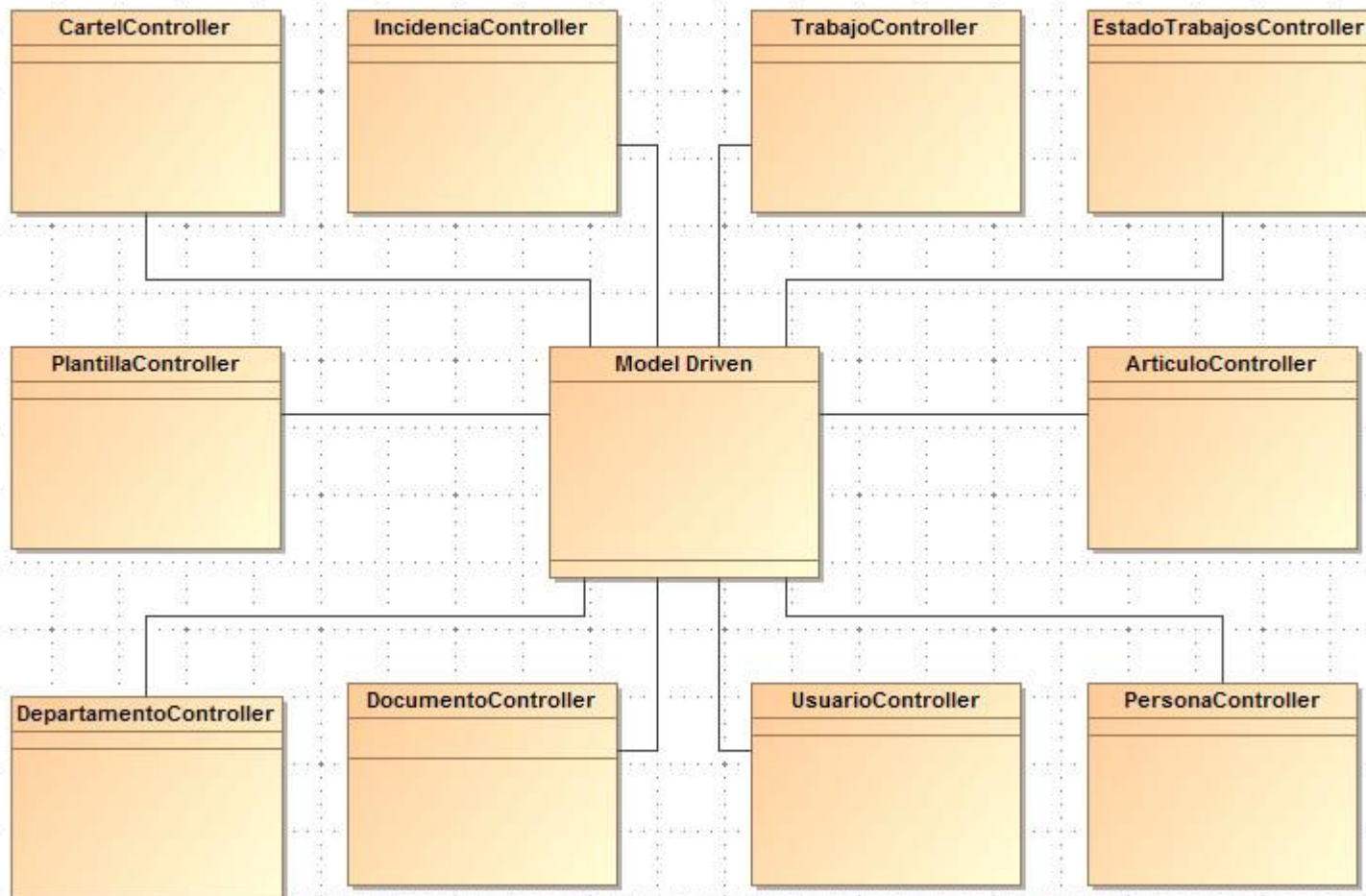
## Diagrama de classes Gestores del Sistema

El sistema tindrà un gestorDAO per a cada base de dades. En aquest cas la connexió es fa a través de hibernate.

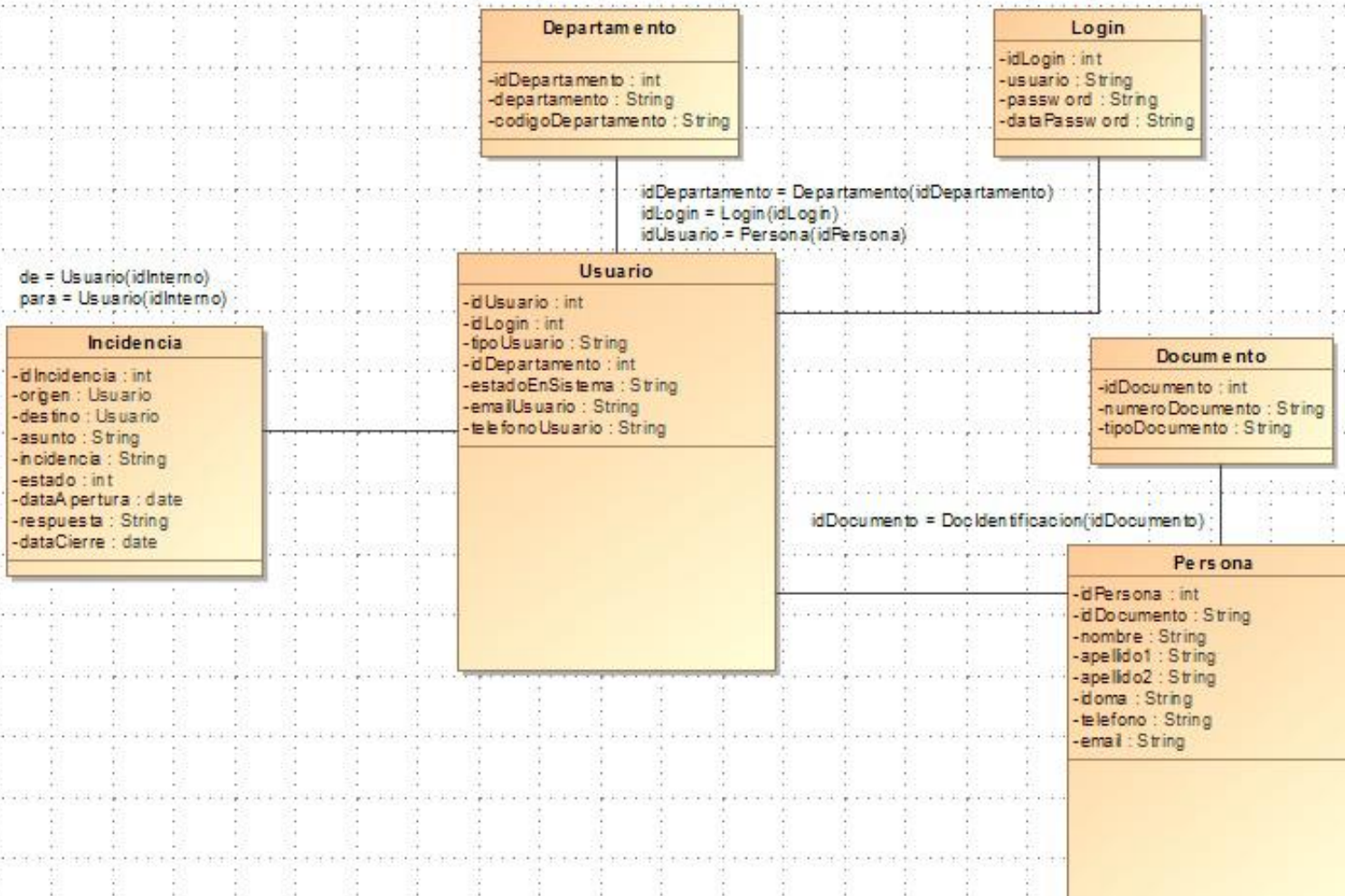


## Diagrama de classes Controller

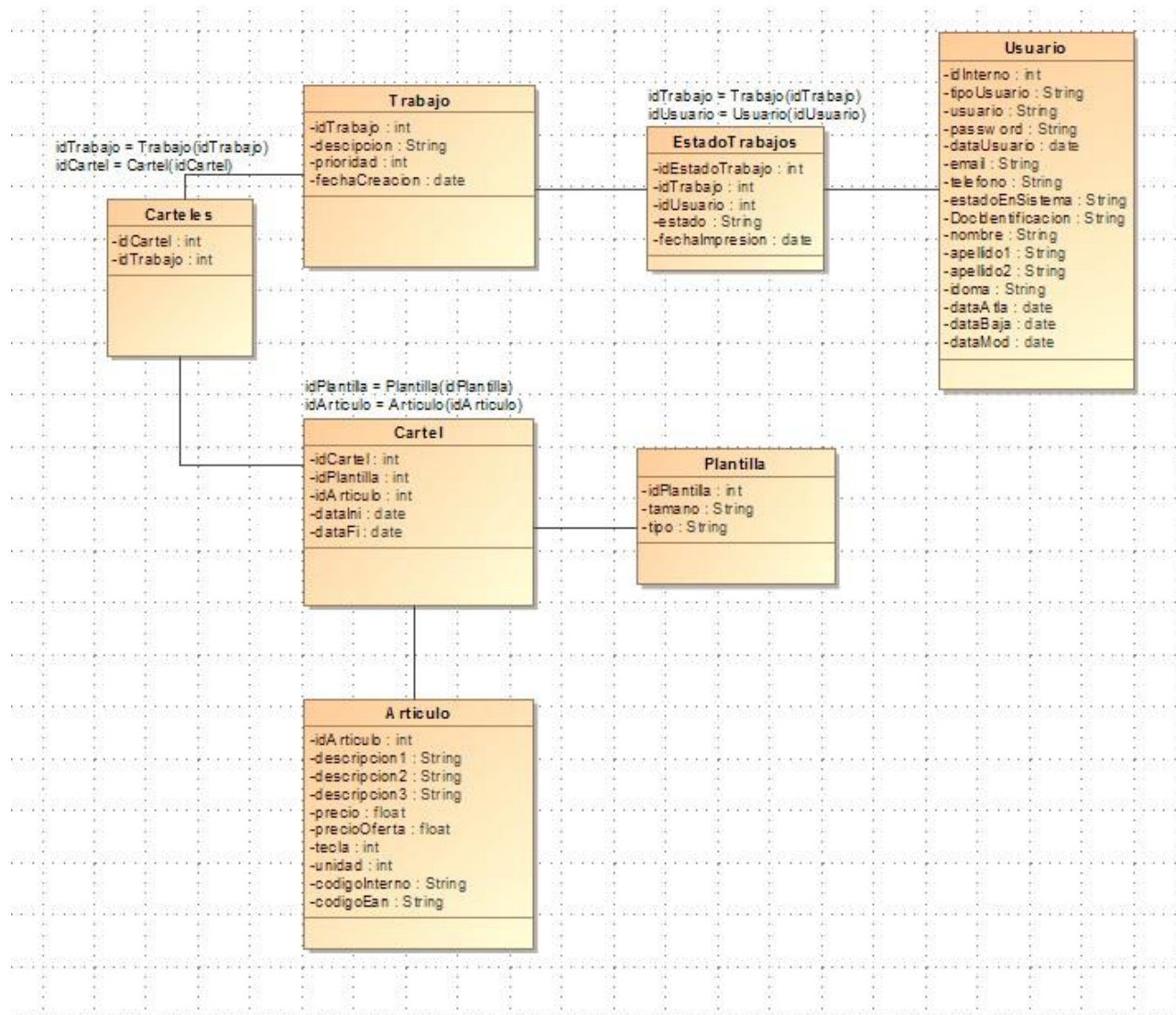
Dintre de la capa Model hi ha les classes controller, son les encarregades de gestionar i respondre als events accionats per l'usuari i invoca les peticions del al model.



## Disseny de la Base de dades



## Disseny de la Base de dades





## Resultat final

Al final de tot això... l'objectiu i idea principal de tot aquest projecte es augmentar la productivitat del personal que treballa en un comerç.

D'aquesta manera en comptes de tenir que fer la carteleria en plantilles (cada comerç) ve feta desde la central. A mes a mes l'arquitectura J2EE ens permet que crear la carteleria una única vegada, tota una cadena de comerços pugin connectarse i imprimir.

Utilitzant aquest metode, s'aconsegueix un augment de productivitat exponencial a nivell companyia.



## Resultat final

Es pot resumir encara mes... l'objectiu es que una vegada tens acces al sistema, clicant dos botons el resultat sigui el següent...

