

TREBALL FI DE CARRERA J2EE  
BOTIGA VIRTUAL  
INT STOCKIST

Estudiant ETIS

**Joan Gibert Badia**

Consultor TFC J2EE

**Vicenç Font Sagrista**

13 de gener de 2013

1. Introducció.....	4
1.1 Descripció.....	4
1.2 Tecnologies.....	4
1.3 Funcionalitats.....	5
1.4 Rols.....	6
1.5 Tasques.....	6
1.6 Temporització.....	7
2. Anàlisi.....	8
2.1 Actors del sistema.....	8
2.2 Casos d'ús dels actors .....	9
2.2.1 Casos d'ús del visitant.....	9
2.2.2 Casos d'ús del client.....	9
2.2.3 Casos d'ús de l'administrador.....	10
2.2.4 Casos d'ús de l'administrador de comandes.....	10
2.3 Diagrama de l'aplicació.....	11
2.4 Descripció dels casos d'ús.....	12
2.4.1 Registrar-se.....	12
2.4.2 Identificar-se.....	12
2.4.3 Consultar productes.....	12
2.4.4 Veure detall.....	13
2.4.5 Afegir detall.....	13
2.4.6 Veure cistell.....	13
2.4.7 Veure detalls producte.....	14
2.4.8 Actualitzar detall.....	14
2.4.9 Eliminar detall.....	14
2.4.10 Confirmar cistell.....	15
2.4.11 Log Out.....	15
2.4.12 Afegir producte.....	15
2.4.13 Buscar producte.....	16
2.4.14 Actualitzar producte.....	16
2.4.15 Eliminar producte.....	16
2.4.16 Consultar comandes.....	17

3. Disseny.....	18
3.1 Diagrama de classes.....	18
3.2 Diagrama relacional.....	19
3.3 Creació base de dades.....	20
3.4 Diagrames d'activitats.....	23
3.4.1 Realitzar compra.....	23
3.4.2 Actualitzar/eliminar producte .....	24
3.5 Diagrama de seqüència "Alta de client".....	25
3.6 Diagrama de col·laboració "Login" .....	26
3.7 Interfícies d'usuari final.....	27
4. Implementació.....	30
4.1 Estructura del projecte.....	30
4.2 Proves.....	32
5. Conclusions.....	33
6. Glossari.....	34
7. Bibliografia.....	35
8. Annex - Manual d'instal·lació.....	36

## 1. Introducció

### 1.1 Descripció

L'aplicatiu a realitzar tracta d'una botiga on-line, on els clients podran realitzar la compra de diferents plaques base de la marca Intel. Un cop posats a disposició una sèrie de productes, l'administrador de la distribuïdora podrà portar a terme el manteniment de la web.

### 1.2 Tecnologies

L'aplicació serà de tipus J2EE.

El framework escollit és Java Server Faces, el qual permet definir la navegació entre classes, disposa d'una llibreria d'etiquetes Java Server Pages, components JavaBeans ...

S'utilitzarà el patró MVC (Model Vista Controlador), tot separant l'especificació, la implementació i l'aspecte visual aconseguint un manteniment simple

El servidor Apache Tomcat ens permetrà visualitzar el projecte al nostre navegador (HTTP).

MySQL serà el sistema d'administració de base de dades utilitzada, dins l'aplicació es realitzaran accions sobre una base de dades creada (consultes, insercions, modificacions).

El component DAO(Data Acces object) serà l'encarregat de realitzar les accions entre la web i la base de dades. Funció de pont (comunicació entre ells)

Components reutilitzables Beans, arxius .java.

Les pàgines .jsp, ajudaran a crear pàgines dinàmiques basades en html.

L'entorn de desenvolupament NetBeans, fet principalment per el llenguatge de programació Java

## 1.3 Funcionalitats

Es poden diferenciar tres tipus d'actors, els clients que compren a la tenda un cop registrats i visualitzen productes sense la necessitat de logerar-se, l'administrador, el qual es dedica al manteniment de la web i el visitant, persona que pot accedir a la web per tal de consultar els productes sense la obligació de registrar-se.

Els clients s'han de donar d'alta mitjançant un formulari, un cop estigui registrat el seu nom d'usuari i contrasenya podran realitzar compres

Un cop estigui funcionant la tenda, amb els productes i les gestions disponibles, l'administrador de la distribuïdora podrà realitzar tasques de manteniment. S'inclourà un cercador de productes per tal d'agilitzar les accions.

Més endavant es podria crear un quart actor, el qual es dedicaria a consultar les comandes que s'han realitzat. Seria el responsable d'iniciar els tràmits d'enviament i només podria accedir a aquesta secció.

L'administrador tindrà tots els privilegis de consultes i el responsable de comandes només un.

Sobre la gestió de connexió i seguretat, els clients i administradors necessitaran identificar-se amb una contrasenya que no serà visible, apareixeran "\*\*\*" al camp contrasenya. Depenent de quin actor es tracti accedirà a una part de la web o a una altre.

## 1.4 Rols

Visitant: Consultar Productes i registrar-se

Client: Login a la web, consulta i inserció de productes al carro de la compra, modificació del carro i confirmació del carro

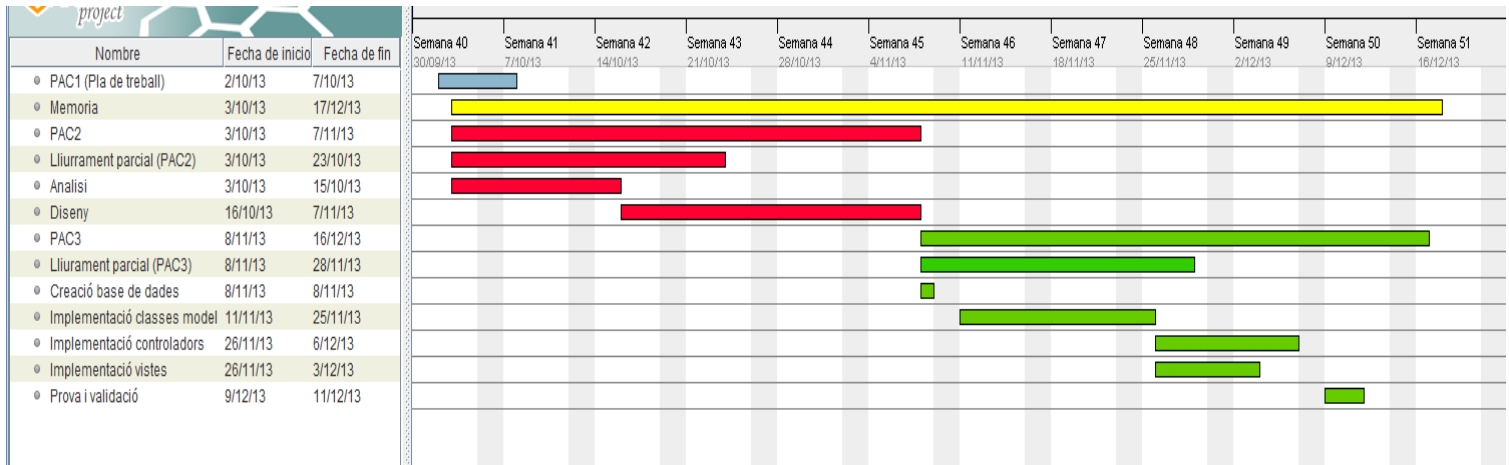
Administrador: Login a la web. Visualització, cerca, inserció, modificació i eliminació de productes, consulta de comandes

Responsable comandes: Login i consultar comandes

## 1.5 Tasques

- Anàlisi prèvia; Descriurem breument el projecte a realitzar i realitzarem el pla de treball on definirem les tecnologies, les funcionalitats i la metodologia que es duran a terme durant el projecte.
- Anàlisi de requisits; Especificació dels components que formaran part de la web mitjançant diagrames de casos d'ús. Podrem veure els diferents rols que tenen els actors (clients, visitants i administrador). Diferenciant els privilegis que tenen els clients registrats i els clients no registrats (visitants) i veient el control que disposarà l'administrador.
- Disseny; Adaptació de les especificacions a la tecnologia J2EE. Mitjançant diferents diagrames, obtindrem la base per implementar els diferents components
- Implementació; Creació de la base de dades, implementació de la persistència, implementació dels controladors, implementació de les vistes, validació de camps
- Prova; Correcció de possibles errors i verificació de la aplicació

# 1.6 Temporització



## 2. Anàlisis

### 2.1 Actors del sistema

Depenent del tipus d'usuari que accedeixi a l'aplicació podrà dur a terme unes funcions o unes altres. Cada usuari tindrà uns rols assignats, tots els usuaris excepte el visitant hauran d'estar identificats per tal de realitzar les seves gestions.

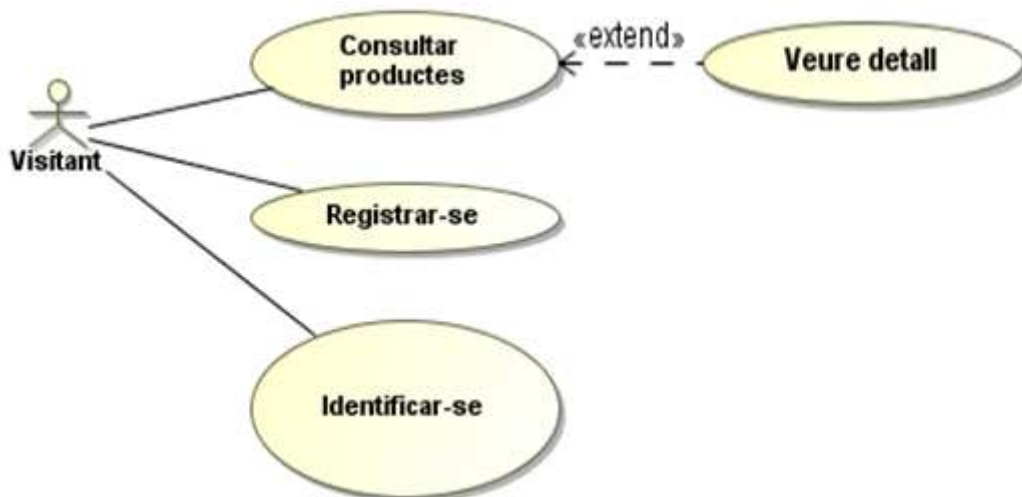
Es diferencien 4 tipus d'usuaris diferents:

- Visitant. Es tracta de la persona que entra a la pàgina, només podrà consultar els productes oferts i registrar-se. Un cop registrat podrà accedir i esdevenir client.
- Client. Es tracta del visitant que s'ha identificat. Un cop inicia amb un nom d'usuari i contrasenya, apart de consultar els productes, podrà afegir la quantitat que desitgi d'un determinat producte al seu cistell. Posteriorment podrà modificar-ne o eliminar-ne els detalls. I confirmar la compra.
- Administrador. Un cop identificat accedirà a l'apartat d'administració, on podrà modificar, afegir, consultar i eliminar productes. Disposarà d'un cercador per tal d'agilitzar el manteniment. També podrà consultar les compres que s'han realitzat.
- Administrador de comandes.Usuari que consulta les compres que s'han realitzat a la web, encarregat d'iniciar el procés d'enviament.

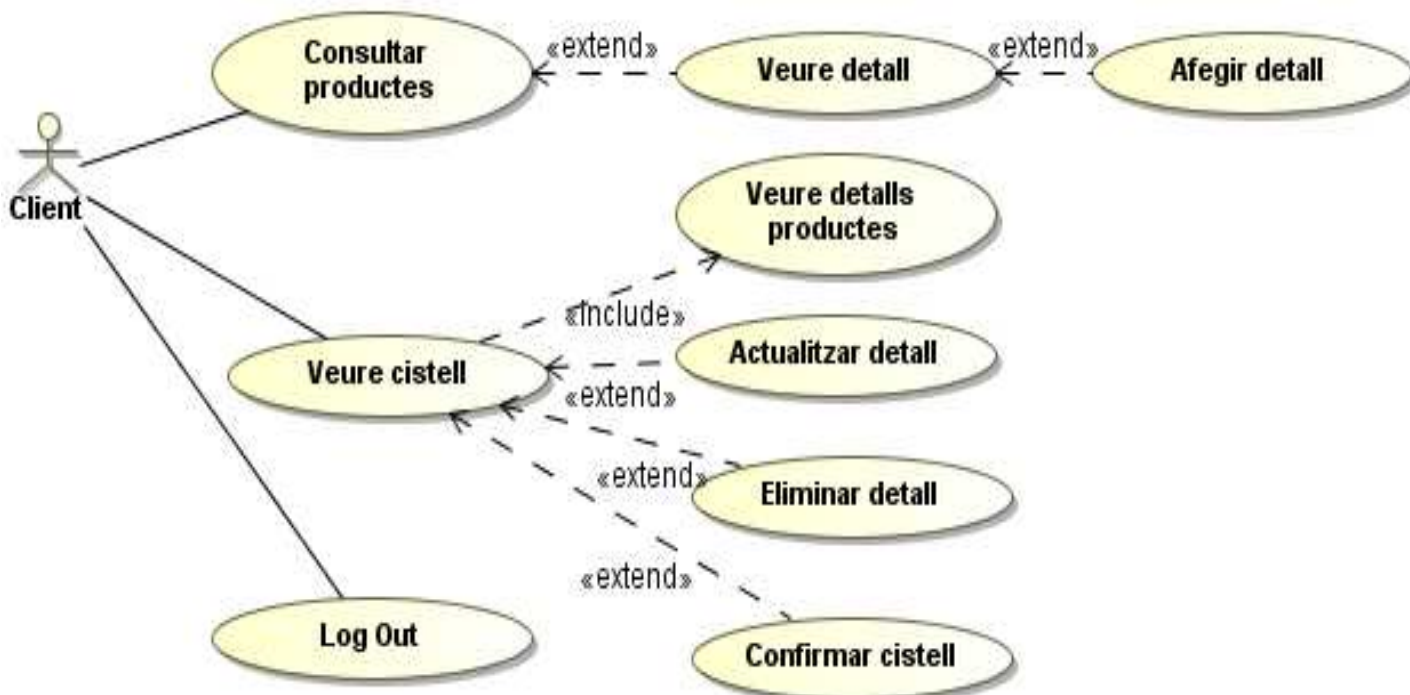


## 2.2 Casos d'ús dels actors

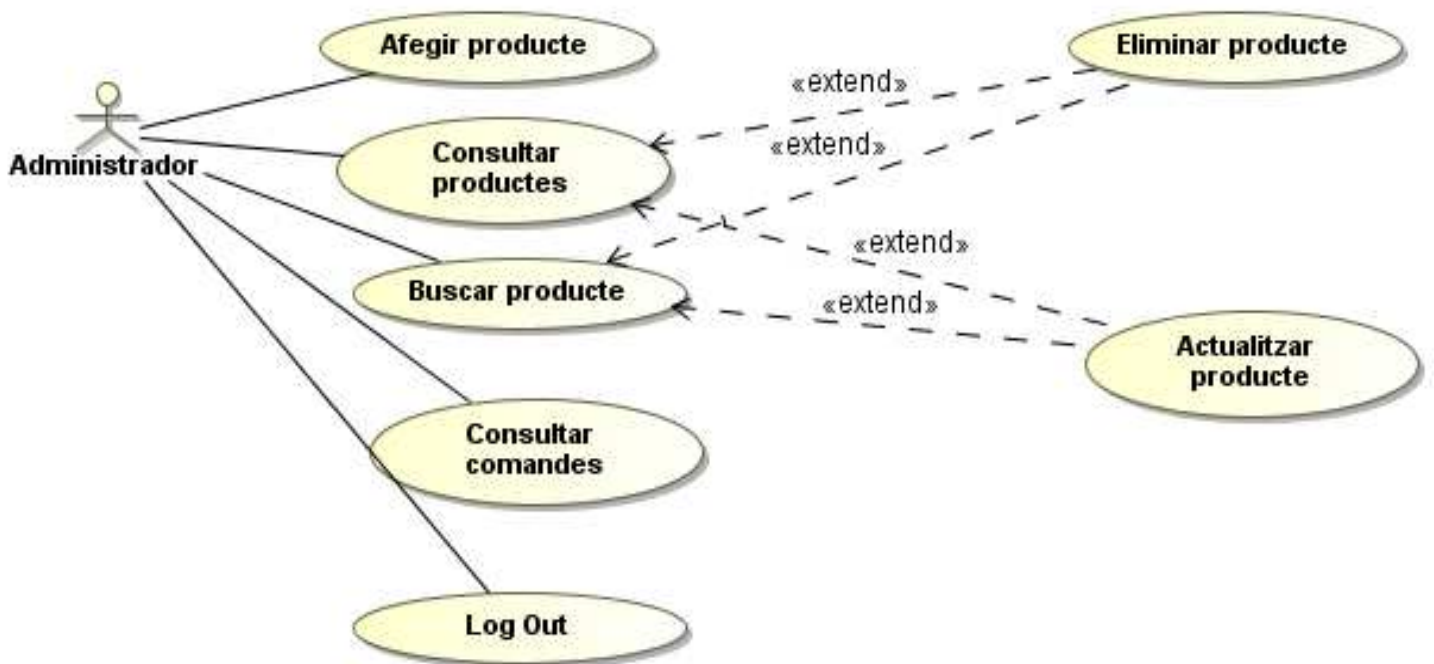
### 2.2.1 Cas d'us del Visitant



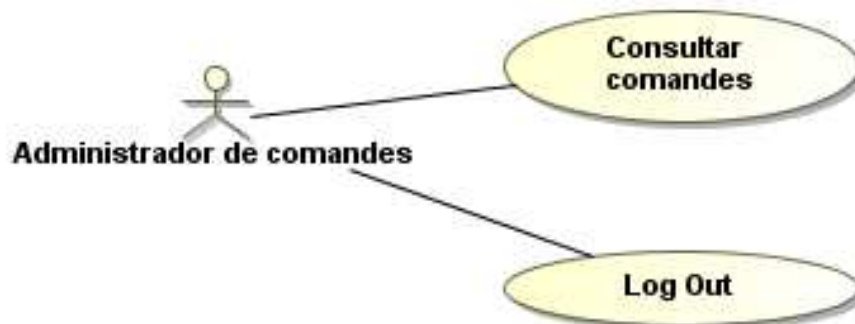
### 2.2.2 Cas d'us del Client



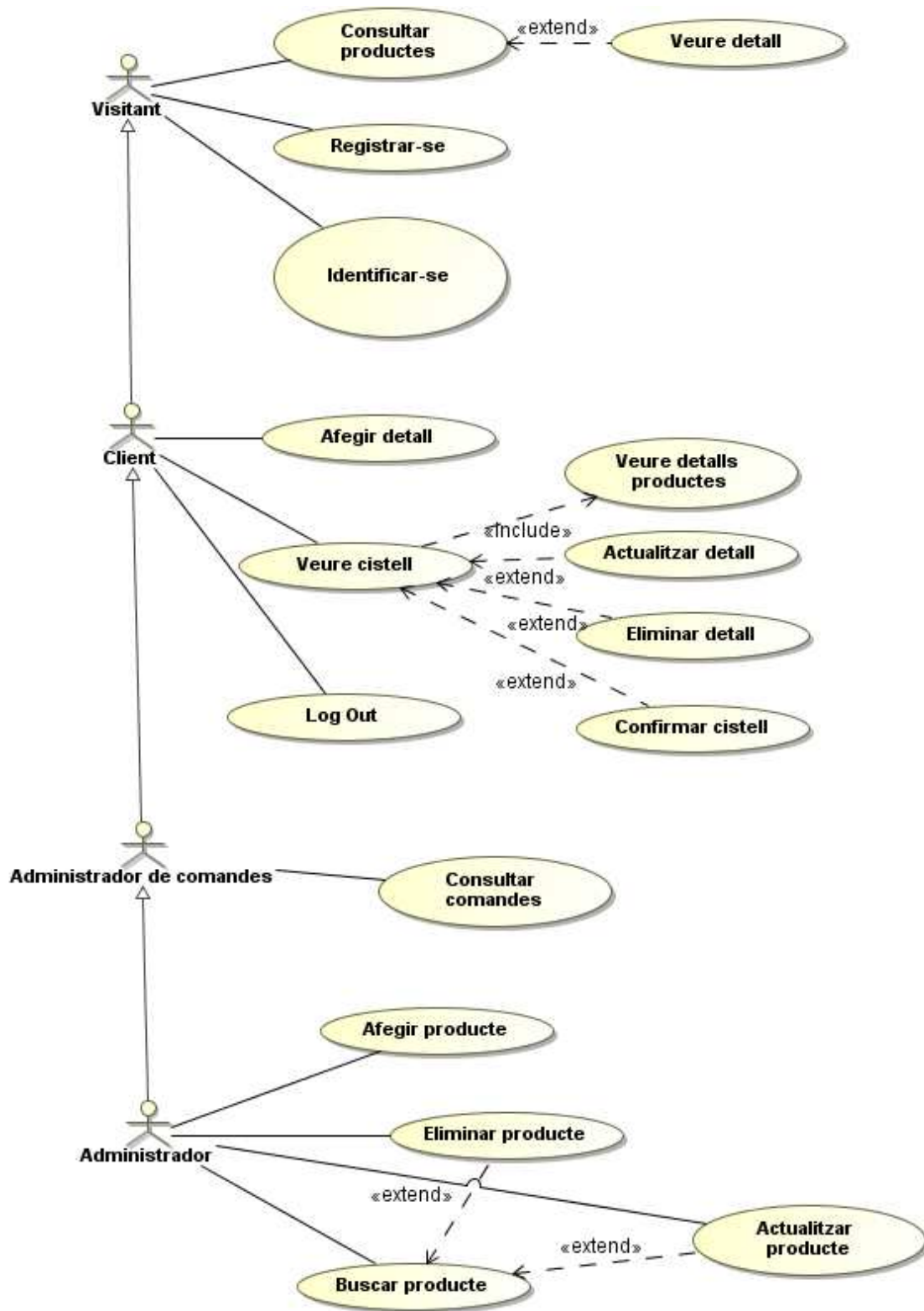
### 2.2.3 Cas d'us de l'Administrador



### 2.2.4 Cas d'us de l'Administrador de comandes



## 2.3 Diagrama de l'aplicació



## 2.4 Descripció dels casos d'ús

### 2.4.1 Registrar-se

Funcionalitat: Permetre als visitants registrar-se al sistema per tal de que hi puguin accedir.

Actors implicats: Visitant.

Casos d'ús relacionats: Cap.

Descripció detallada: Per tal de que els usuaris puguin realitzar compres han de haver iniciat amb un nom d'usuari i una contrasenya, el qual han d'escollir en el formulari del registre. Es demanaran les dades del visitant.

### 2.4.2 Identificar-se

Funcionalitat: Permetre a els usuaris accedir a la seva secció.

Actors implicats: Visitants, Administrador i Administrador de comandes.

Casos d'ús relacionats: Cap.

Descripció detallada: El clients accediran al sistema amb el nom d'usuari i contrasenya que han escollit, els usuaris i contrasenyes dels administradors estaran registrats a la base de dades, no es podran crear. Depenent de l'usuari que es tracti accedirà a una secció o a una altre.

### 2.4.3 Consultar productes

Funcionalitat: Permetre visualitzar els productes que estan actualment a la venda.

Actors implicats: Visitants, Clients i Administrador.

Casos d'ús relacionats: Veure detall, actualitzar producte i eliminar producte.

Descripció detallada: Els visitants només podran consultar productes, els clients durant la consulta podran visualitzar els detalls si estan interessats en comprar-los. L'administrador tindrà l'accés a l'actualització i modificació mitjançant uns enllaços.

### **2.4.4 Veure detall**

Funcionalitat: Permetre visualitzar el detall d'un producte.

Actors implicats: Visitant i Client.

Casos d'ús relacionats: Consultar productes i Afegir detall.

Descripció detallada: El client podrà veure el detall del producte, i escollir la quantitat desitjada. El visitant només podrà veure el detall sense opció d'escollir la quantitat

### **2.4.5 Afegir detall**

Funcionalitat: Permetre incorporar un producte al cistell.

Actors implicats: Client.

Casos d'ús relacionats: Veure detall.

Descripció detallada: Quan el client ha vist el detall i confirma que el vol afegir al seu cistell.

### **2.4.6 Veure cistell**

Funcionalitat: Permetre veure i gestionar els productes incorporats al cistell.

Actors implicats: Client.

Casos d'ús relacionats: Veure detalls producte, Actualitzar detall, Eliminar detall i Confirmar cistella.

Descripció detallada: En aquest apartat el client veurà l'estat actual de la cistella de la compra, els productes escollits amb el seu cost i el cost total de la compra, el qual calcula el IVA.

### **2.4.7 Veure detalls producte**

Funcionalitat: Permetre visualitzar els productes escollits

Actors implicats: Client.

Casos d'ús relacionats: Veure cistell

Descripció detallada: Quantitat, descripció i preu d'un producte que es mostra a la cistella de la compra.

### **2.4.8 Actualitzar detall**

Funcionalitat: Permetre actualitzar un producte al cistell.

Actors implicats: Client.

Casos d'ús relacionats: Veure cistell.

Descripció detallada: Modificació de la quantitat desitjada, serà possible recalculer el cost total de la compra abans de confirmar-la.

### **2.4.9 Eliminar detall**

Funcionalitat: Permetre eliminar un producte al cistell.

Actors implicats: Client.

Casos d'ús relacionats: Veure cistell.

Descripció detallada: Si el client no desitja comprar un producte un cop incorporat a la cistella tindrà la opció de eliminar-lo.

## 2.4.10 Confirmar cistell

Funcionalitat: Permetre validar el cistell.

Actors implicats: Client.

Casos d'ús relacionats: Veure cistell.

Descripció detallada: Acceptació per part del client que vol adquirir els productes que es mostraven al cistell, quantitats i preus confirmades.

## 2.4.11 Log Out

Funcionalitat: Sortir

Actors implicats: Client , Administrador i Administrador de comandes

Casos d'ús relacionats: Cap.

Descripció detallada: Permetrà la sortida immediata dels usuaris registrats al sistema.

## 2.4.12 Afegir producte

Funcionalitat: Permetre incorporar nous productes a la venda.

Actors implicats: Administrador.

Casos d'ús relacionats: Cap.

Descripció detallada: L'administrador selecciona l'opció registre de producte i introdueix les dades necessàries al formulari. Un cop validades les dades s'introdueixen a la base de dades i el producte es visible per als clients.

### **2.4.13 Buscar producte**

Funcionalitat: Buscar un producte a la tenda.

Actors implicats: Administrador.

Casos d'ús relacionats: Actualitzar producte i Eliminar producte.

Descripció detallada: L'administrador realitzarà la cerca quan vulgui actualitzar o eliminar un producte determinat. Trobarà el producte per la seva referència.

### **2.4.14 Actualitzar producte**

Funcionalitat: Actualitzar un producte existent.

Actors implicats: Administrador.

Casos d'ús relacionats: Buscar Producte i Consultar Productes.

Descripció detallada: Quan l'administrador hagi realitzat la cerca de producte tindrà l'opció d'actualitzar (botó). L'actualització també serà disponible quan consulti els productes mitjançant un botó(actualitzar), a diferència del client que tindrà únicament l'opció afegir(a la cistella)

### **2.4.15 Eliminar producte**

Funcionalitat: Eliminar un producte existent.

Actors implicats: Administrador.

Casos d'ús relacionats: Buscar Producte i Consultar Productes.

Descripció detallada: Quan l'administrador hagi realitzat la cerca de producte tindrà l'opció d'eliminar(botó). La eliminació també serà disponible quan consulti els productes mitjançant un botó(eliminar), a diferència del client que tindrà únicament l'opció afegir(a la cistella)



## 2.4.16 Consultar comandes

Funcionalitat: Permetre consultar les compres que han demanat els clients.

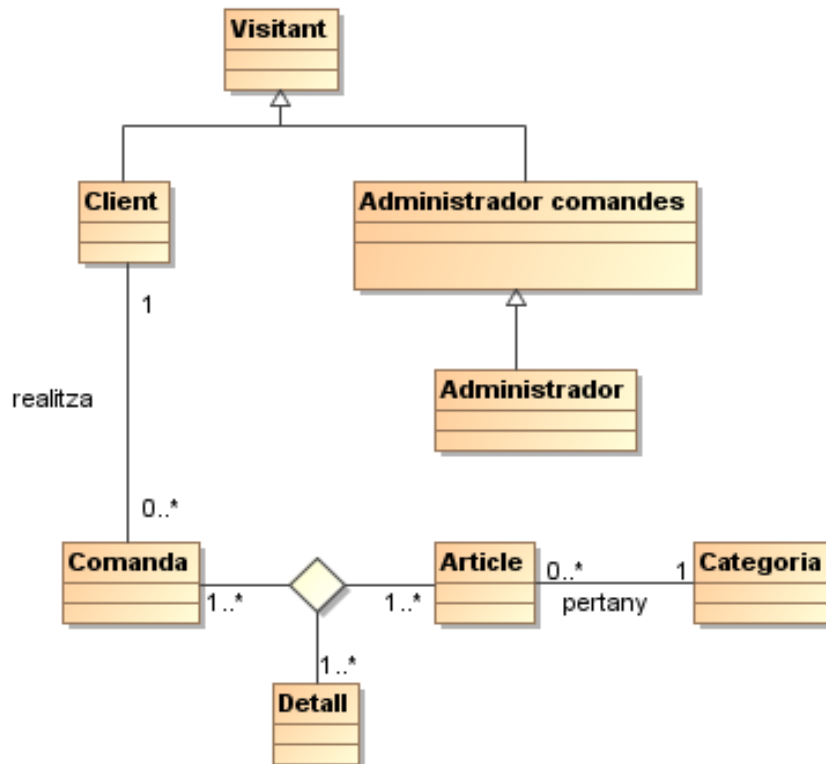
Actors implicats: Administrador i Administrador de comandes.

Casos d'ús relacionats: Cap.

Descripció detallada: Es visualitzaran les comandes que s'han realitzat a la tenda. Llistat de les comandes per tal de poder tramitar-les per part de l'administrador de comandes. L'administrador general podrà també podrà consultar-les.

## 3. Disseny

### 3.1 Diagrama de classes



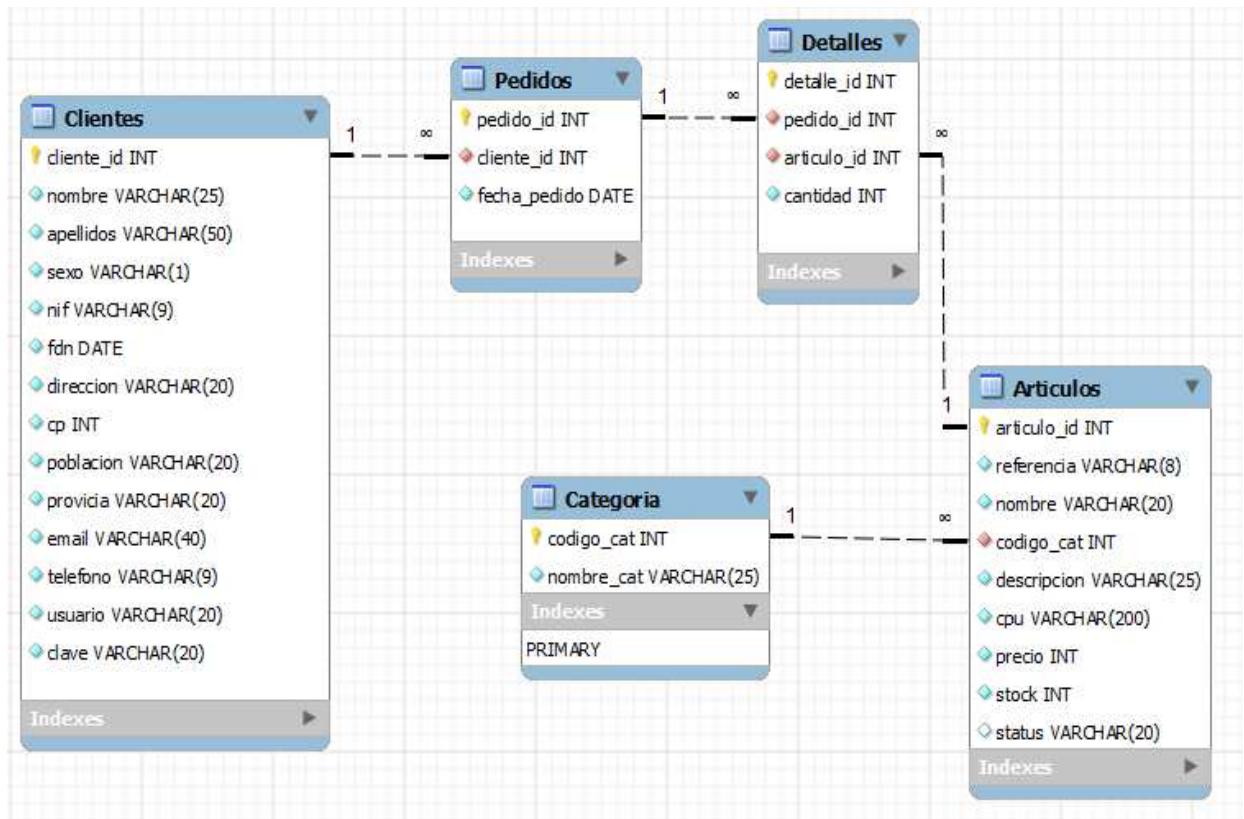
Les classes Client i Administrador de comandes hereten de la classe Visitant.

Un client pot realitzar cap o moltes comandes.

Un article pertany a una sola categoria, pot ser que d'una categoria no trobem cap article

Relació ternària entre Comanda, Detall i Article. Una comanda pot estar formada per un article o més, els quals formaran part dels detalls de la comanda.

## 3.2 Diagrama relacional



En aquest esquema es mostren les taules de la base de dades de l'aplicació. Cada una d'elles disposa d'una clau primària, formada per un camp numèric. També es pot observar la relació que mantenen les taules amb les claus forànies. Com a exemple, es pot dir que un client pot realitzar varies comandes o que una comanda està formada per diferents detalls (podria ser un o varis). A l'hora d'afegir un valor a un camp que correspon a una clau forània, posteriorment cal haver introduït el codi corresponent a la taula on fa referència.

### 3.3 Creació base de dades

```
DROP DATABASE IF EXISTS stockist;  
CREATE DATABASE IF NOT EXISTS stockist;  
USE stockist;
```

```
CREATE TABLE CLIENTES  
(  
  cliente_id int(8) NOT NULL AUTO_INCREMENT,  
  nombre varchar(25) NOT NULL,  
  apellidos varchar(50) NOT NULL,  
  sexo varchar(1) NOT NULL,  
  nif varchar(9) NOT NULL,  
  fdn date NOT NULL,  
  direccion varchar(20) NOT NULL,  
  cp int(5) NOT NULL,  
  poblacion varchar(20) NOT NULL,  
  provincia varchar(20) NOT NULL,  
  email varchar(40) NOT NULL,  
  telefono varchar(9) NOT NULL,  
  usuario varchar(20) NOT NULL,  
  clave varchar(20) NOT NULL,  
  PRIMARY KEY (cliente_id)  
);
```

```
CREATE TABLE CATEGORIA  
(  
  codigo_cat int(8) NOT NULL AUTO_INCREMENT,  
  nombre_cat varchar(25) NOT NULL,  
  PRIMARY KEY (codigo_cat )  
);
```

```
CREATE TABLE PEDIDOS  
(  
  pedido_id int(8) NOT NULL AUTO_INCREMENT,  
  cliente_id int(8) NOT NULL,  
  fecha_pedido date NOT NULL,  
  PRIMARY KEY (pedido_id),  
  FOREIGN KEY (cliente_id) REFERENCES clientes(cliente_id)
```

);

```
CREATE TABLE ARTICULOS
(
articulo_id int(8) NOT NULL AUTO_INCREMENT,
referencia varchar(8) NOT NULL,
nombre varchar(30) NOT NULL,
codigo_cat int(8) NOT NULL,
descripcion varchar(25) NOT NULL,
cpu varchar(200) NOT NULL,
precio int(4) NOT NULL,
stock int(20) NOT NULL,
status varchar(20) NOT NULL,
PRIMARY KEY (articulo_id),
FOREIGN KEY (codigo_cat) REFERENCES categoria(codigo_cat)
);
```

```
CREATE TABLE DETALLES
(
detall_id int(8) NOT NULL AUTO_INCREMENT,
pedido_id int(8) NOT NULL,
articulo_id int(8) NOT NULL,
cantidad int(8) NOT NULL,
PRIMARY KEY (detall_id),
FOREIGN KEY (pedido_id) REFERENCES pedidos(pedido_id)
);
```

```
insert into categoria values ('1', 'Socket 1150');
insert into categoria values ('2', 'Socket 1155');
insert into categoria values ('3', 'Socket 2011');
insert into categoria values ('4', 'Socket 478');
insert into categoria values ('5', 'Socket 775');
```

```
insert into clientes values ('1', 'Joan', 'Gibert Badia', 'H', '12345678A', '1983-01-11', 'Prat', '08232', 'Vila', 'BCN', 'a@gmail.com', '123456789', 'clientuoc', 'uocclient');
```

```
insert into articulos values ('1', 'z87pro', 'Asus Z87 Pro', '1', 'socket 1150', 'i7/Core i5/Core i3/Pentium/Celeron Processors', '156', '23', 'En stock');
insert into articulos values ('2', 'gah87hd3', 'GigaByte GA-H87-HD3', '1', 'socket 1150', 'Intel Core i7 / Intel i5 / Core Intel Core i3 procesadores / Intel Pentium procesadores / Intel Celeron procesadores en el paquete LGA1150', '75', '4', 'En stock');
```

insert into articulos values ('3', 'gab85m', 'Gigabyte GA-B85M-D3', '1', 'socket 1150', 'Soporte para procesadores Intel Core i7 / Intel i5 / Core Intel Core i3 procesadores / Intel Pentium procesadores / Intel Celeron procesadores en el paquete LGA1150', '56', '44', 'En stock');

insert into articulos values ('4', 'h87fatal', 'Asrock H87 Fatal1ty', '1', 'socket 1150', 'Soporta la 4 generacion de procesador Intel Core i7 / i5 / i3 / Xeon / Pentium / Celeron en el paquete LGA1150', '73', '3', 'En stock');

insert into articulos values ('5', 'h87iplus', 'Asus H87I-Plus C', '1', 'socket 1150', 'Intel Socket 1150 para la 4 generacion de procesadores Core i7/Core i5/Core i3/Pentium/Celeron', '85', '12', 'En stock');

insert into articulos values ('6', 'b75pro3', 'Asrock B75 Pro3', '2', 'socket 1155', 'Soporta la tercera y segunda generacion de procesador Intel Core i7 / i5 / i3 en el paquete LGA1155', '53', '4', 'En stock');

insert into articulos values ('7', 'gaz77', 'GigaByte GA-Z77', '2', 'socket 1155', 'Soporte para procesadores Intel Core i7 / Intel Core i5 / Intel Core i3 procesadores / Intel Pentium / Intel Celeron en LGA1155', '77', '12', 'En stock'); ....

## 3.4 Diagrames d'activitats

### 3.4.1 Diagrama d'activitats "Realitzar compra"

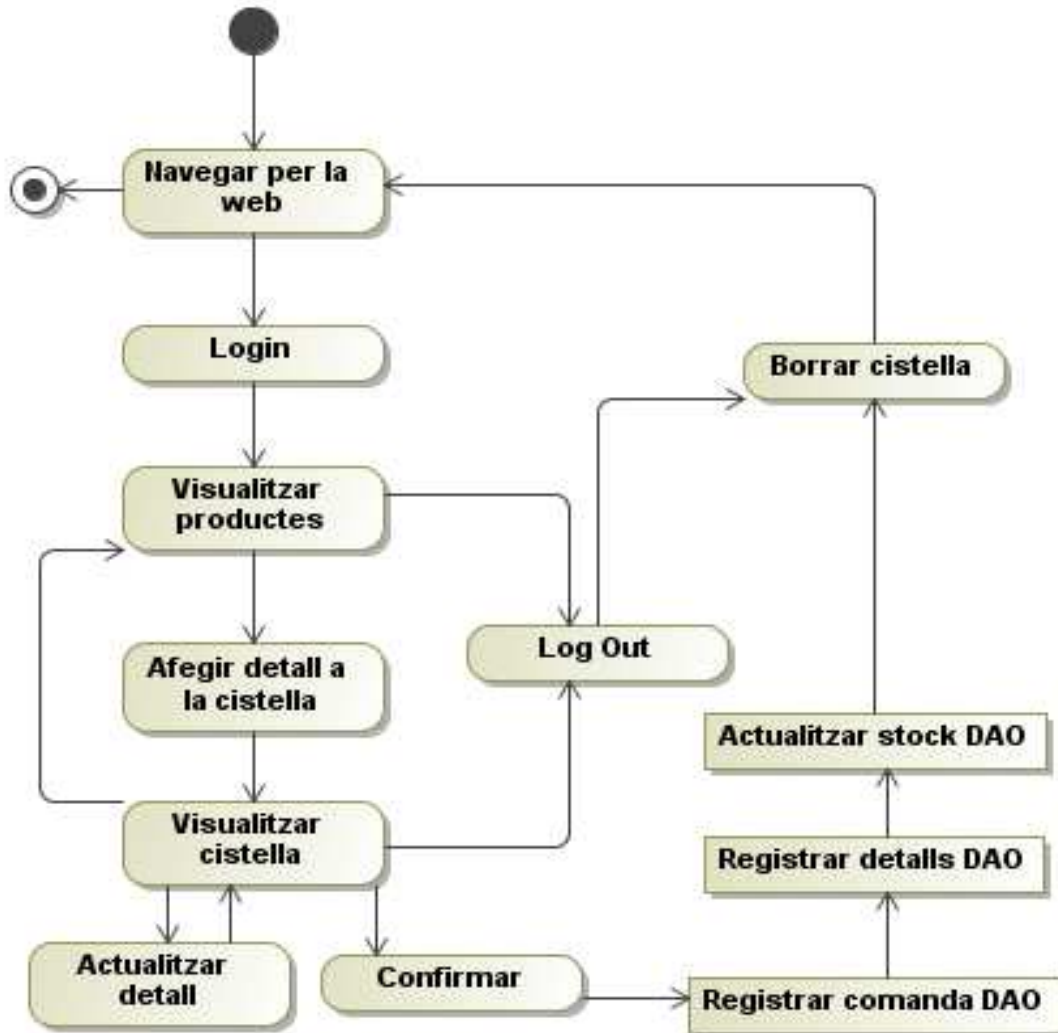
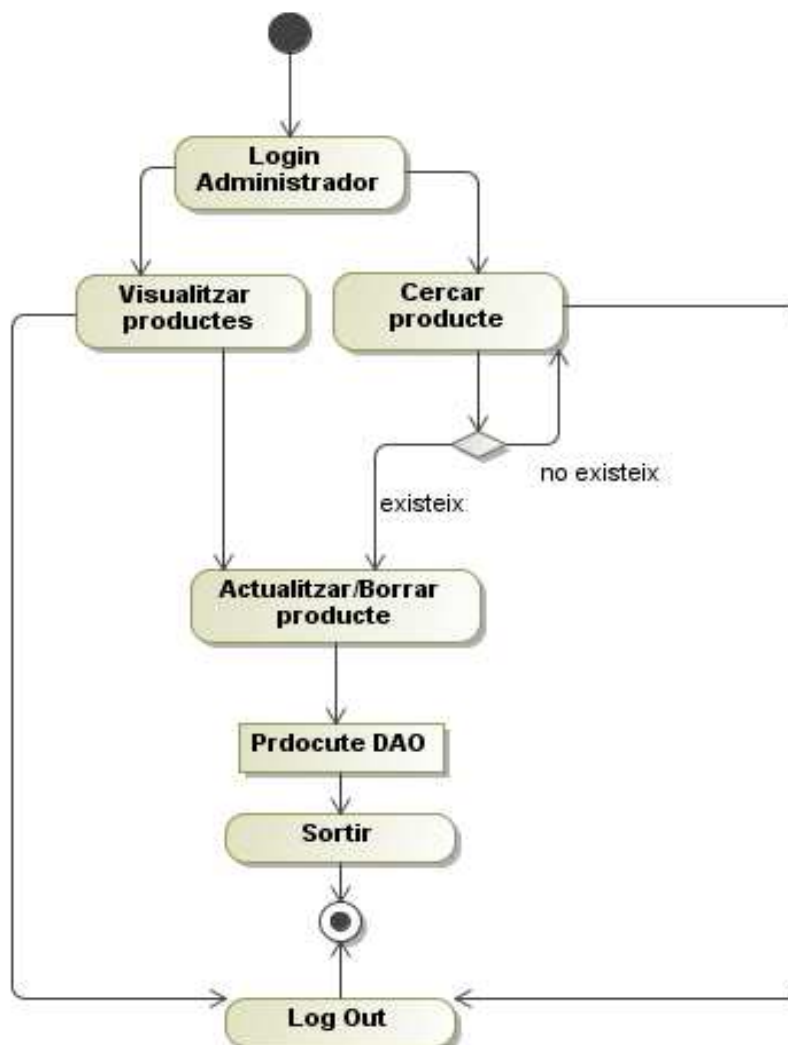


Diagrama que realitzaran els clients de la botiga quan vulguin fer una compra. Podran afegir els productes desitjats junt amb la quantitat i posteriorment podran modificar-ne els detalls. Podran confirmar la compra o bé realitzar Log Out si volen sortir sense confirmar els canvis realitzats anteriorment. Un cop confirmada la compra es registraran els detalls i la comanda, el camp stock dels articles que formen part de la comanda serà actualitzat.

### 3.4.2 Diagrama d'activitats "Actualitzar/Eliminar producte"

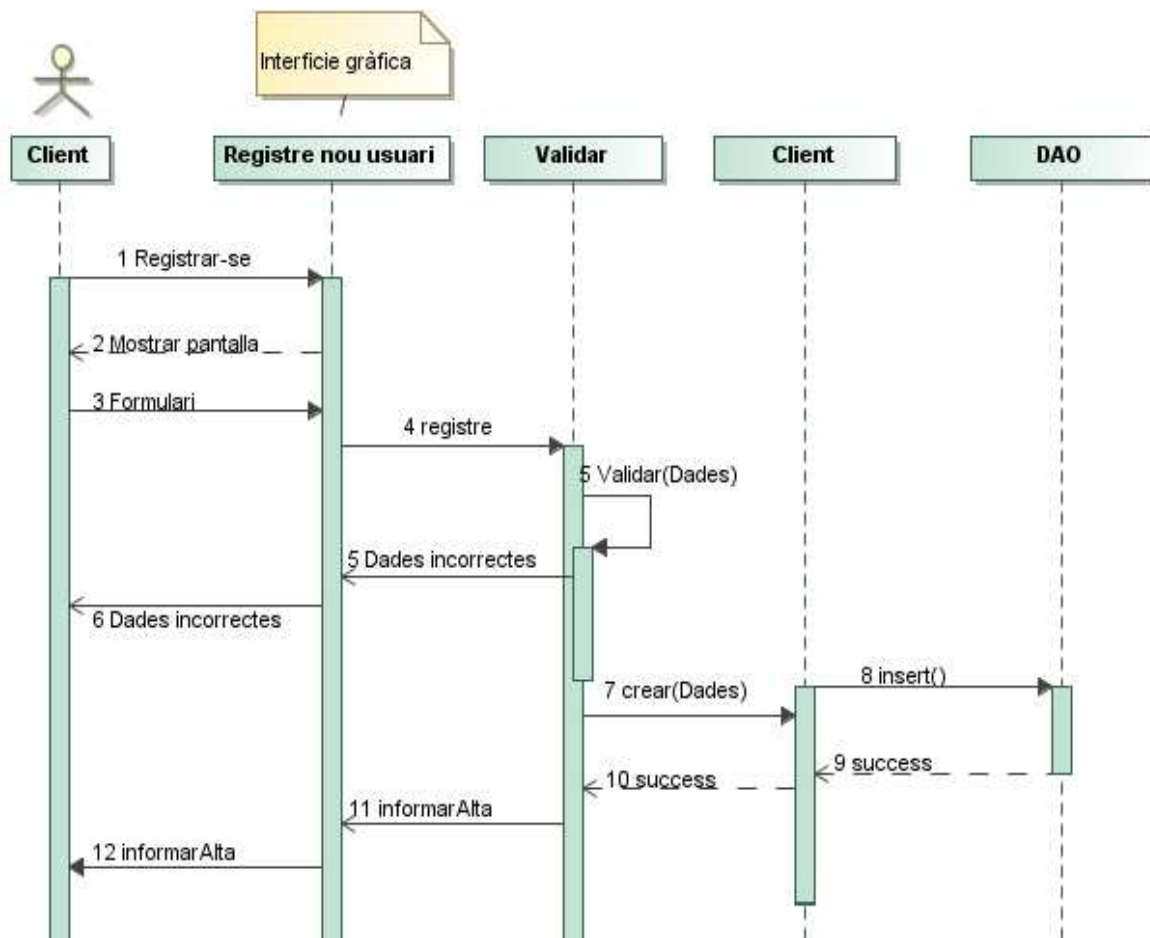


L'administrador de la tenda a l'hora d'actualitzar o eliminar un producte ho podrà fer per dos vies diferents. Pot visualitzar els productes, però a diferència del client que només pot veure el detall, l'administrador podrà eliminar o actualitzar el producte desitjat. Es cercarà el producte per el seu codi de referència. També, més endavant es podria afegir un "select list" amb la categoria de producte, però inicialment la idea es cercar un únic producte determinat, ja que si existissin molts productes d'una determinada categoria seria més difícil trobar el producte en qüestió. Un cop trobat el producte s'elimina o s'actualitza.

ProducteDAO en el diagrama seria "Update" o "Drop" depenent del que es vulgui fer.

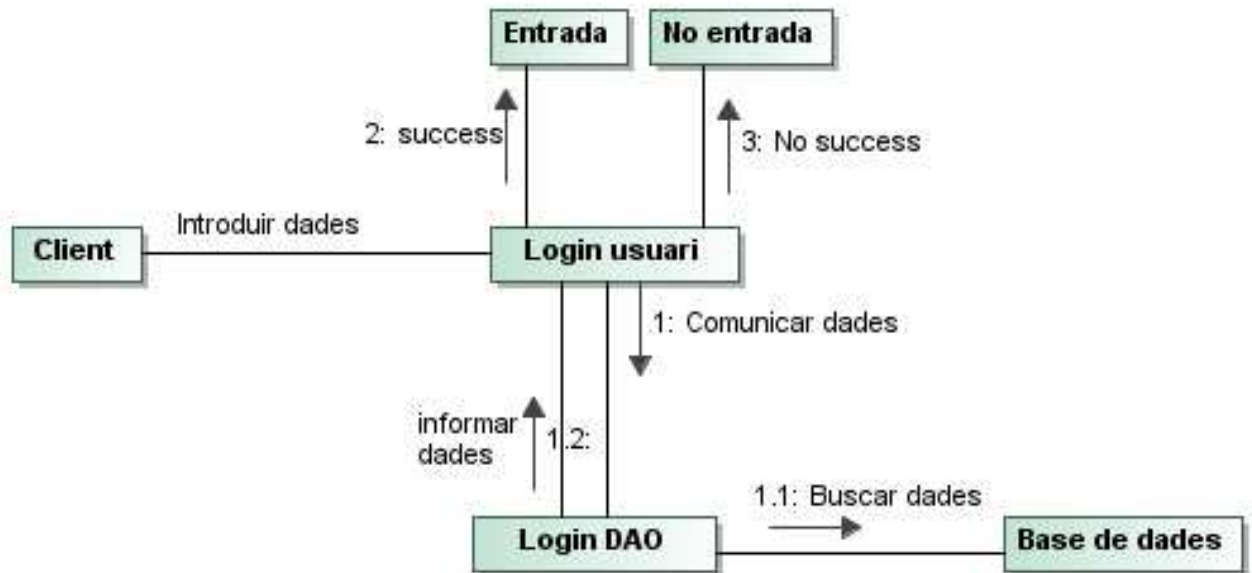


### 3.5 Diagrama de seqüència “Alta de client”



Quan es confirmen les dades, es validaran tots els camps amb “validators”. Quan tots els camps introduïts per el client siguin correctes (acceptables), es guardarà el nou usuari a la base de dades. A la classe validar.java es trobaran les condicions per validar un camp d’un registre.

### 3.6 Diagrama de col·laboració “Login”



En aquest diagrama es pot observar el model (Base de dades), el controlador (Login DAO) encarregat de fer la consulta a la base de dades, i la vista (Login usuari).

Depenent de si l'usuari existeix o no anirà a la vista (Entrada), accedint a la part de client o (No entrada) accedint a una pàgina .jsp informant que l'usuari o contrasenya es incorrecte.

En el cas del client les dades d'identificació es trobaran guardades a la base de dades, les dades del administrador es trobaran dins Login.java. Mitjançant un condicional (if) es verificarà el usuari i la contrasenya per tal d'accedir a l'apartat administrador.

### 3.7 Interfícies d'usuari final

Interfície registre d'usuari:

## INT STOCKIST

inicio socket 1150 socket 1155 socket 2011 socket 478 socket 775 identificate registre
admin

### Registro de usuario

Nombre:	<input type="text" value="joan"/>	
Apellidos:	<input type="text" value="gibert"/>	
Sexo:	<input type="text" value="Hombre"/>	
NIF:	<input type="text" value="12345678A"/>	
Fecha Nacimiento:	<input type="text" value="1983-11-01"/>	
Direccion:	<input type="text" value="prat"/>	
Codigo Postal:	<input type="text" value="08232"/>	
Poblacion:	<input type="text" value="Vila"/>	
Provincia:	<input type="text" value="Barcelona"/>	
Email:	<input type="text" value="joan@gmail.com"/>	
Telefono:	<input type="text" value="649566564"/>	
Nombre de Usuario:	<input type="text" value="clientuoc"/>	El usuario ya existe
Password:	<input type="password"/>	
Repite Password:	<input type="password"/>	


Interfície Detall de producte amb stock:

inicio socket 1150 socket 1155 socket 2011 socket 478 socket 775
carrito logout

### Cantidad a pedir

Articulo:	Asus P9X79 LE OEM
Referencia:	p9x79le
Descripcion:	socket 2011
Cpu:	Intel Socket 2011 for 2nd Generation Core i7 Processors
Precio:	165.0
Status:	En stock
Cantidad disponible:	7
Cantidad a pedir:	<input type="text" value="8"/>

La cantidad deseada supera al stock




Interfície cistell de la compra:

<a href="#">inicio</a> <a href="#">socket 1150</a> <a href="#">socket 1155</a> <a href="#">socket 2011</a> <a href="#">socket 478</a> <a href="#">socket 775</a>						<a href="#">carrito</a> <a href="#">logout</a>
Estado actual del pedido						
Nombre:	Descripcion:	Precio:	Cantidad:	Disponibles:	SubTotal:	
Asus P9X79 LE OEM	socket 2011	165.0 €	<input type="text" value="2"/>	7	330,00 €	<input type="button" value="Borrar"/>
GigaByte GA-H87-HD3	socket 1150	75.0 €	<input type="text" value="2"/>	4	150,00 €	<input type="button" value="Borrar"/>
Coste:				480,00 €		
Coste con IVA:				580,80 €		
<input type="button" value="Guardar Pedido"/> <input type="button" value="Recalcular"/>						

Interfície buscar article a modificar i posteriorment Interfície amb el formulari amb l'article trobat:

<a href="#">inicio</a> <a href="#">socket 1150</a> <a href="#">socket 1155</a> <a href="#">socket 2011</a> <a href="#">socket 478</a> <a href="#">socket 775</a> <a href="#">insertar producto</a> <a href="#">actualizar producto</a> <a href="#">eliminar producto</a> <a href="#">ver pedidos</a> <a href="#">logout</a>									
Buscar articulo									
Referencia <input type="text"/>									
<input type="button" value="Buscar"/>									

<a href="#">inicio</a> <a href="#">socket 1150</a> <a href="#">socket 1155</a> <a href="#">socket 2011</a> <a href="#">socket 478</a> <a href="#">socket 775</a> <a href="#">insertar producto</a> <a href="#">actualizar producto</a> <a href="#">eliminar producto</a> <a href="#">ver pedidos</a> <a href="#">logout</a>						
Articulo encontrado						
MUESTRA:	Referencia	Nombre	Descripcion	Cpu	Stock	Precio
	p4i65g	ssssss	socket 1150	ssssss	3	3.0
Modificar						
Nombre:	<input type="text"/>					
Descripcion	<input type="text" value="Socket 1150"/>					
Cpu:	<input type="text"/>					
Stock:	<input type="text" value="0"/>					
Precio	<input type="text" value="0.0"/>					
<input type="button" value="Registrar"/>						

Interfície registre de producte:

**INT STOCKIST**

[inicio](#)
[socket 1150](#)
[socket 1155](#)
[socket 2011](#)
[socket 478](#)
[socket 775](#)
[insertar producto](#)
[actualizar producto](#)
[eliminar producto](#)
[ver pedidos](#)
[logout](#)

**Registro de articulo**

Referencia:  La referencia ya existe

Nombre:

Descripcion:

Cpu:

Precio:

Stock:

Imagen: Hay que subir la imagen (referencia.jpg) a la carpeta imagenes !

Interfície veure comandes:

**INT STOCKIST**

[inicio](#)
[socket 1150](#)
[socket 1155](#)
[socket 2011](#)
[socket 478](#)
[socket 775](#)
[insertar producto](#)
[actualizar producto](#)
[eliminar producto](#)
[ver pedidos](#)
[logout](#)

**Pedidos**

Pedido	Id articulo	Cantidad	Fecha pedido	Cliente Id
177532978	2	2.0	2013-12-17	1
177532978	17	3.0	2013-12-17	1
538312126	16	1.0	2013-12-11	1
654747915	2	1.0	2013-12-17	2
654747915	18	1.0	2013-12-17	2

Interfície veure productes per categoria (administrador):

**INT STOCKIST**

[inicio](#)
[socket 1150](#)
[socket 1155](#)
[socket 2011](#)
[socket 478](#)
[socket 775](#)
[insertar producto](#)
[actualizar producto](#)
[eliminar producto](#)
[ver pedidos](#)
[logout](#)

**Socket 1150**

MUESTRA	ARTICULO	STATUS	STOCK	PRECIO	
	Gigabyte GA-B85M-D3	En stock	44	56.0 €	<input type="button" value="Eliminar"/>
	GigaByte GA-H87-HD3	En stock	1	75.0 €	<input type="button" value="Eliminar"/>

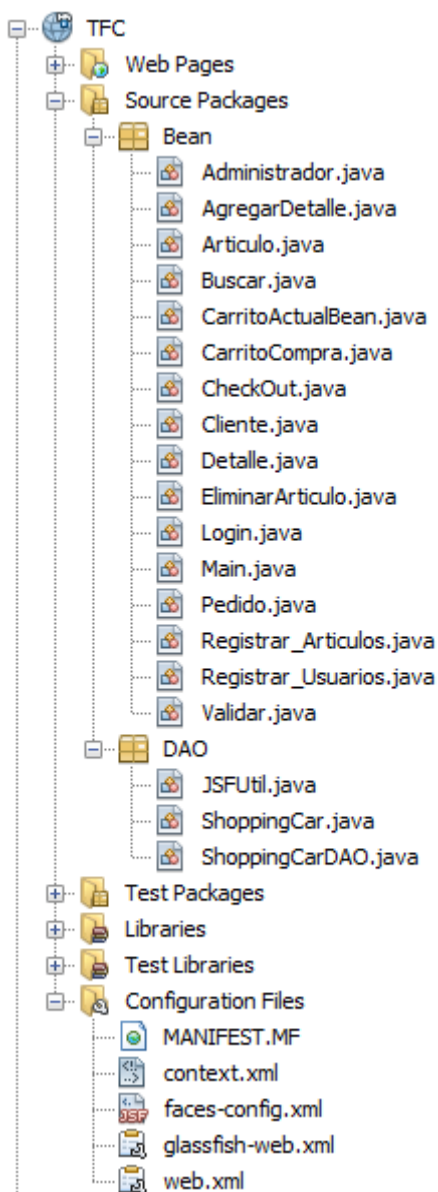
## 4. Implementació

### 4.1 Estructura del projecte

Vista: En el directori Web Pages es troben les pàgines JavaServer Pages, l'arxiu estils.css i la carpeta imatges

Controlador: En el directori Bean hi ha les classes que fan d'intermediàries entre la vista i el model

Model: En el directori DAO es troben les classes que realitzen consultes a la base de dades de l'aplicació



En la carpeta Configuration files, hi trobem els fitxers de configuració. L'arxiu faces-config.xml defineix la navegació entre pàgines, en el context.xml definim la base de dades amb el propi controlador i l'usuari i contrasenya per accedir-hi. L'arxiu web.xml defineix la pàgina inicial del projecte. Tots aquest fitxers també es troben a la carpeta WEB-INF dins del directori Web Pages.

Context.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
] <Context path="/TFC">
  <Resource driverClassName="com.mysql.jdbc.Driver" maxActive="8" maxIdle="4" name="jdbc/stockist"
    password="root" type="javax.sql.DataSource" url="jdbc:mysql://localhost/stockist" username="root"/>
- </Context>
```

Web.xml:

```
<welcome-file-list>
  <welcome-file>faces/main.jsp</welcome-file>
</welcome-file-list>
```

Faces-config.xml

```
  <navigation-case>
    <from-outcome>comandes</from-outcome>
    <to-view-id>/Comandes.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
<managed-bean>
  <managed-bean-name>Login</managed-bean-name>
  <managed-bean-class>Bean.Login</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>Registrar_Usuarios</managed-bean-name>
  <managed-bean-class>Bean.Registrar_Usuarios</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
</managed-bean>
```

## 4.2 Proves

Durant les proves del funcionament de l'aplicació, s'ha pogut comprovar el correcte funcionament de tots els rols dels actors. La funcionalitat que no funciona el 100% bé és la de registre de producte. En aquest cas es registre correctament el producte a la base de dades excepte de la imatge del producte, la qual s'ha d'afegir manualment a la carpeta d'imatges.

La funcionalitat que no he pogut implementar és la de actualització de producte a partir de la llista de productes d'una determinada categoria. Cal dir, que un producte es pot eliminar si s'accedeix al menú eliminació de producte, on apareix un caixó de text a on s'escriu la referència del producte i elimina el producte en qüestió en cas d'existir.

L'script de la base de dades s'ha modificat respecte al presentat a la PAC2, on hi havia una relació forana a la taula detalls que feia que l'aplicació fallés. L'eliminació d'aquesta relació fa que es pugui eliminar un producte, el qual es troba en un o més registres de detalls.

En l'aplicació es realitza la visualització, inserció, modificació i eliminació de productes per part del administrador, no s'ha fet per als clients, tot i que la metodologia seria la mateixa.

S'ha dut a terme el control d'estoc dels productes. En cas de que no quedin unitats, l'aplicació mostra "Fora d'estoc". En cas de demanar una quantitat superior a l'estoc, s'informa de que no es disposa de tantes unitats.



## 5. Conclusions

Tenint en compte que tenia unes nocions bàsiques sobre el framework JSF amb managed beans, la realització d'aquest treball la considero molt enriquidora i positiva, ja que he consolidat coneixements anteriors i n'he adquirit de nous.

Al principi m'ha costat crear funcionalitats, però un cop vist el funcionament m'ha costat cada cop menys crear-ne de noves. Personalment penso que és una tecnologia molt completa, i més simple comparada amb altres J2EE, encara que és l'única amb la que hi tinc experiència programant.

He assolit totes les funcionalitats proposades al principi, però es podria millorar, com per exemple la inserció i visualització d'imatges a la base de dades. Tot i això em considero satisfet amb el producte final, portant al dia les tasques plantejades i finalitzant la meva primera aplicació J2EE.

La creació de la botiga virtual m'ha servit per veure el funcionament o comunicació que hi ha entre l'aplicació i la base de dades, per un hipotètic futur projecte es podrien aplicar els coneixements adquirits. Consultes, insercions, modificacions i eliminacions utilitzant mysql i jsf.

## 6. Glossari

Administrador: Persona que gestiona els productes de la botiga virtual

Client: Persona registrada que pot efectuar compres

Visitant: Persona que consulta els productes sense haver-se registrat

MVC: Patró Model-Vista-Controlador

DAO: Patró Data Access Object

JDBC: Java Database Connectivity

JSF: Java Server Faces

J2EE: Java 2 Enterprise Edition

## 7. Bibliografia

<http://www.mkyong.com/jsf2/custom-validator-in-jsf-2-0/>

<http://www.sicuma.uma.es/sicuma/Formacion/documentacion/JSF.pdf>

<http://tomcat.apache.org/tomcat-7.0-doc/jndi-resources-howto.html>

<http://www.roseindia.net/jsf/dataTable.shtml>

<http://www.jsftoolbox.com/documentation/help/12-TagReference/html/index.jsf>

<http://www.jsftutorials.net/jsf-navigation-by-examples.html>

## 8. Annex – Manual d'instal·lació

El sistema de gestió de bases de dades que he escollit és MySQL 5.6. Per tal de crear les taules i afegir-hi valors, s'ha d'executar l'script un cop s'ha iniciat sessió a MySQL Command Line Client. La comanda ha escriure és: source "directori on es troba l'arxiu .sql ". Per exemple source c:\script.sql, si es trobés a c:

Per connectar-se a la base de dades s'utilitzarà l'usuari "root", amb contrasenya "root".

L'entorn de desenvolupament es NetBeans IDE 7.4

Crear un nou projecte amb el nom TFC, File / NewProject / javaWeb / WebApplication / framework Java Server Faces. Un cop creat s'han de sobreesciure tots els arxius de la carpeta TFC, a la carpeta TFC que s'ha acabat de crear, la qual s'ubica a la carpeta NetBeansProjects. D'aquesta manera tindrem a l'abast tots els fitxers necessaris.

Les llibreries necessàries venen incloses a la carpeta build/web/WEB-INF, si el programa avises de que no es troben es podrien afegir. Botó dret sobre el projecte(TFC) i "Resolve Problems", hauríem d'afegir les llibreries que no es trobessin.

Com ha servidor s'utilitza Apache Tomcat 7.0.41.0

Per tal de córrer l'aplicació amb aquest servidor, cal fer botó dret sobre TFC i anar a propietats. Un cop allà en el menú categories, seleccionar "Run" i escollir el servidor.

Un cop executem l'aplicació el navegador obrirà la pàgina <http://localhost:8084/TFC/>.

Per tal d'iniciar com a client s'ha d'utilitzar l'usuari "clientuoc" i la contrasenya "uocclient", com a administrador s'ha d'utilitzar l'usuari "adminuoc" i la contrasenya "uocadmin" i com a administrador de comandes usuari "comanda" i contrasenya "comanda".