

Made Of Felt Market



PROYECTO MOFM – Presentación

Pedro A. Leal Martínez

Consultor: Óscar Escudero Sánchez



INTRODUCCIÓN. Objetivos.

- En la realización de este Trabajo Final de Carrera se pretende realizar una aplicación web que gestione las necesidades de una tienda de productos realizados con fieltro.

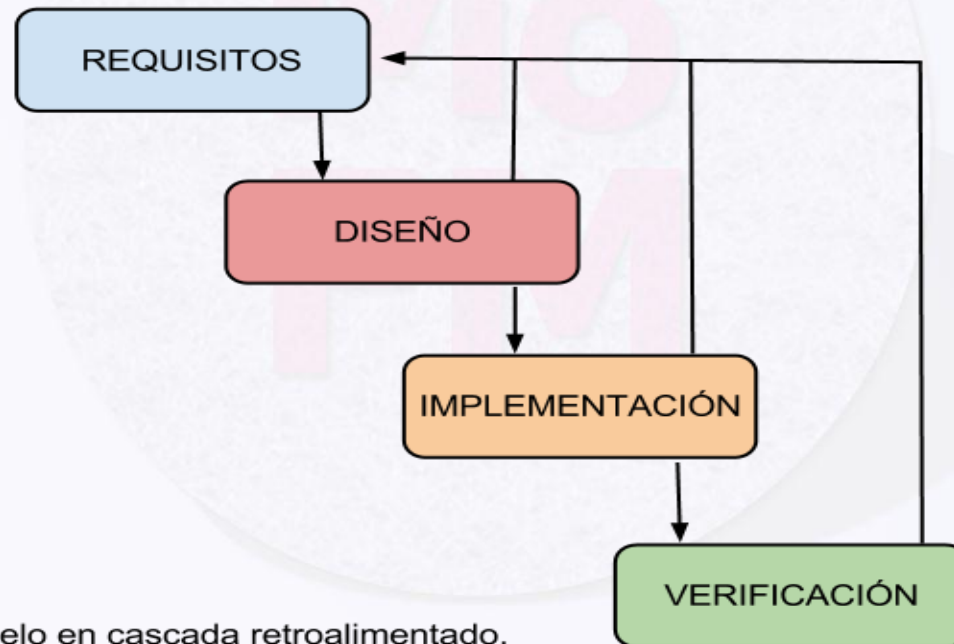
OBJETIVOS:

1. El manejo de herramientas que permitan la implementación de la aplicación en un entorno web: J2EE, JPA, MySQL, Servlets, html, css, modelo Vista Controlador MVC, ...
2. Realización de un proyecto de software completo y planificado a través de las diferentes fases de análisis, diseño e implementación, ...
3. Obtención de un producto que satisfaga las funcionalidades pedidas, además de una documentación completa.

PLANIFICACIÓN. Modelo.

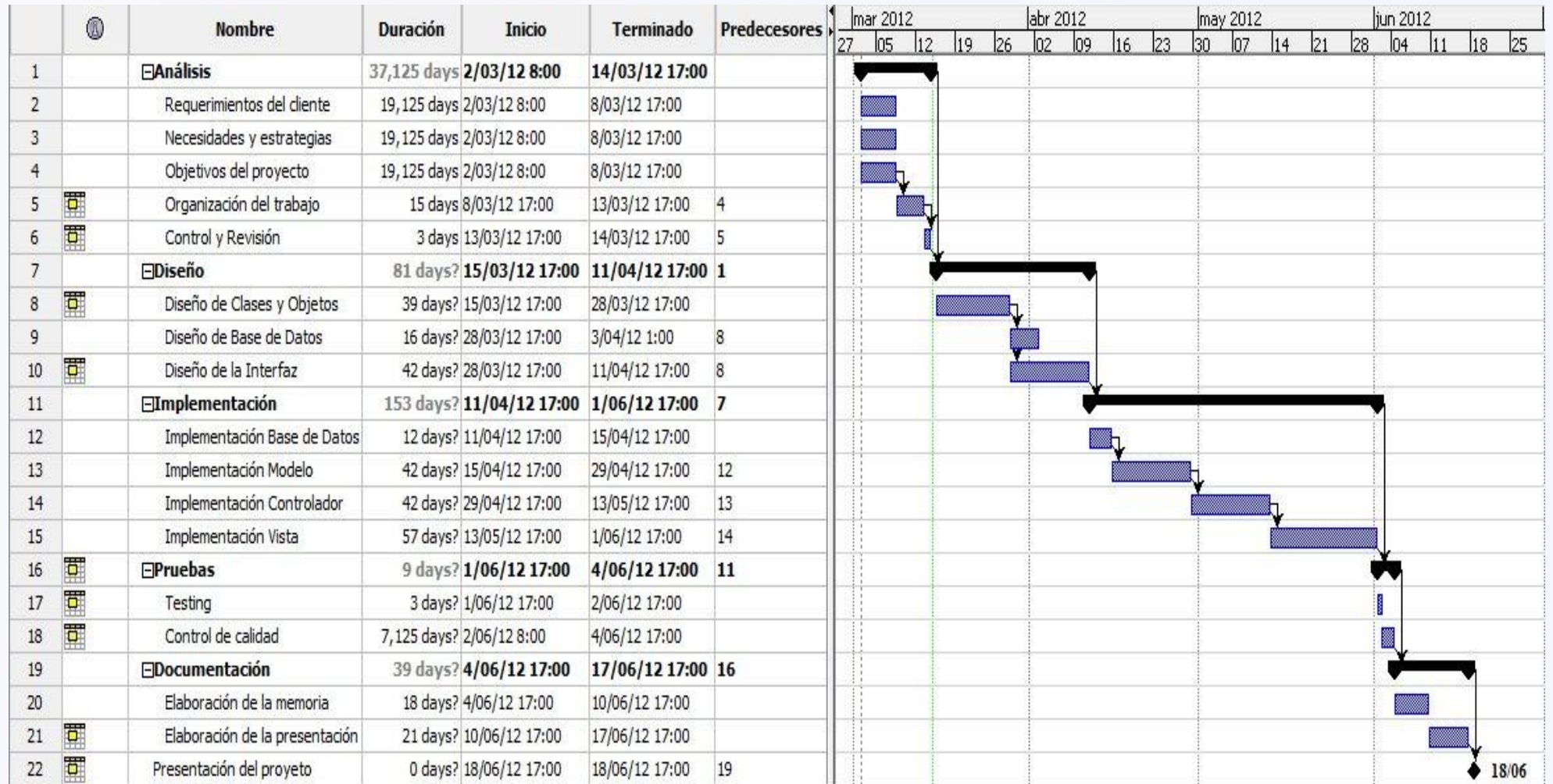
MODELO EN CASCADA RETROALIMENTADO

Donde se seguirá el modelo tradicional en cascada aunque se pretende ser más realista permitiendo que cualquier problema que surja en una etapa pueda influenciar una etapa anterior.



Modelo en cascada retroalimentado.

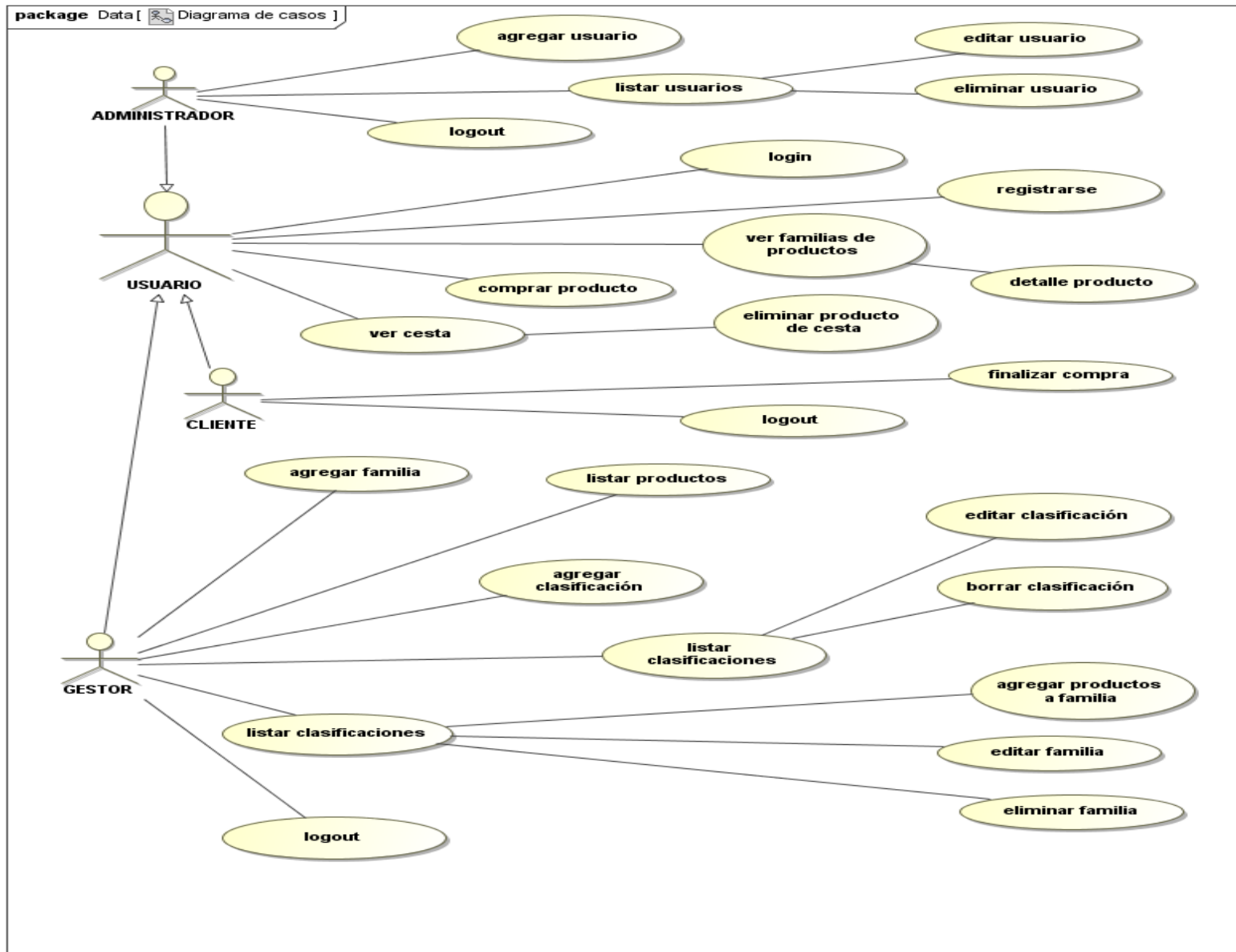
PLANIFICACIÓN. Diagrama Gantt.



ESPECIFICACIÓN. Descripción.

- Realización del **texto informal** que representa la información que viene desde el cliente y que describe las necesidades del producto.
- **Glosario de términos** que presenta y describe los términos más importantes utilizados en el sistema.
- **Modelo del dominio** donde se representan las clases y relaciones entre ellas del sistema.
- **Diagrama de casos de uso** donde se representan las funcionalidades del sistema.
- **Documentación textual de los casos de uso** donde se describe textualmente los casos de uso del sistema.
- **Requisitos de la interfaz de usuario** donde a través de una tabla se representan las necesidades de la interfaz para cada funcionalidad.

ESPECIFICACIÓN. Casos de uso.



ANÁLISIS. Clases y atributos.

- En esta fase se identifican las clases y sus atributos que forman en el sistema.
- Se utiliza una representación en cajas:

USUARIO
-idUserio : integer -nombre : string -apellidos : string -dni : string -dirección : string -login : string -password : string -estado : boolean -rol : integer
+usuario(id, nombre, apellidos, dni, dirección, login, password, estado) +getId(dni) +setId(dni) +getNombre(idUsuario) +setNombre(idUsuario, nombre) +getApellidos(idUsuario) +setApellidos(idUsuario, apellidos) +getDni(idUsuario) +setDni(idUsuario, dni) +getDirección(idUsuario) +setDirección(idUsuario, dirección) +setLogin(idUsuario, login) +getLogin(idUsuario) +getPassword(idUsuario) +setPassword(idUsuario, password) +getEstado(idUsuario) +setEstado(idUsuario, estado) +getRol(idUsuario) +setRol(idUsuario, rol)

FAMILIA
-idFamilia : integer -idClasificacion : integer -nombre : string -descripcion : string -urlImagen : string -urlImagenPeq : string -precio : Float -estado : boolean
+familia(id, clasificacion, nombre, descripcion, urlImagen, urlImagenPeq, precio) +getClasificacion(id) +setClasificacion(id, clasificacion) +getNombre(id) +setNombre(id, nombre) +getDescripcion(id) +setDescripcion(id, descripcion) +getURLImg1(id) +setURLImg1(id, url) +getURLImg2(id) +setURLImg2(id, url) +getPrecio(id) +setPrecio(id, precio) ...

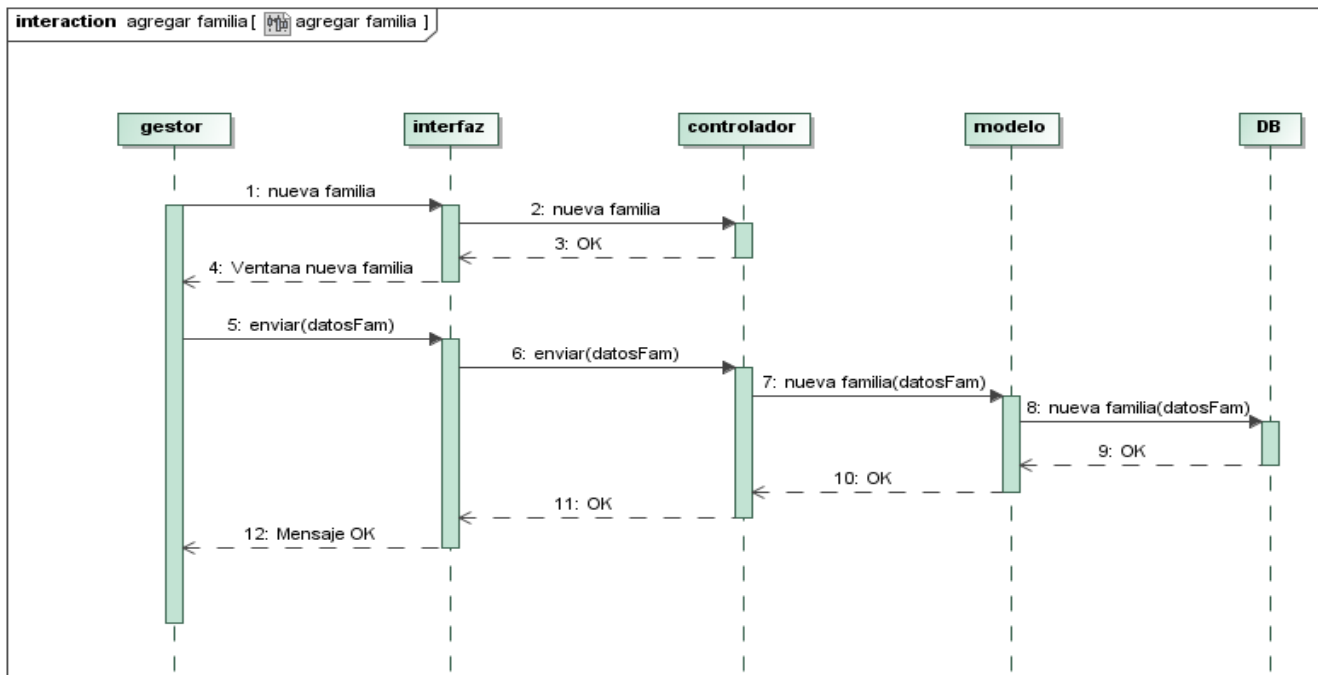
CLASIFICACIÓN
-idClasificacion : Integer -nombre : String
+clasificacion(id, nombre) +getNombre(idClasificacion) +setNombre(id, nombre)

PRODUCTO
-idProducto : integer -idFamilia : integer -estado : Byte
+producto(idProducto, idFamilia, estado) +getEstado(idProducto) +setEstado(idProducto, estado) +getFamilia(idProducto) +setFamilia(idProducto, familia)

COMPRA
-idCliente : integer -idProducto : integer -fecha : date -estado : byte
+compra(idCliente, idProducto, fecha, estado) +getFecha(idCliente, idProducto) +setFecha(idCliente, idProducto, fecha) +getEstado(idCliente, idProducto) +setEstado(idCliente, idProducto, estado)

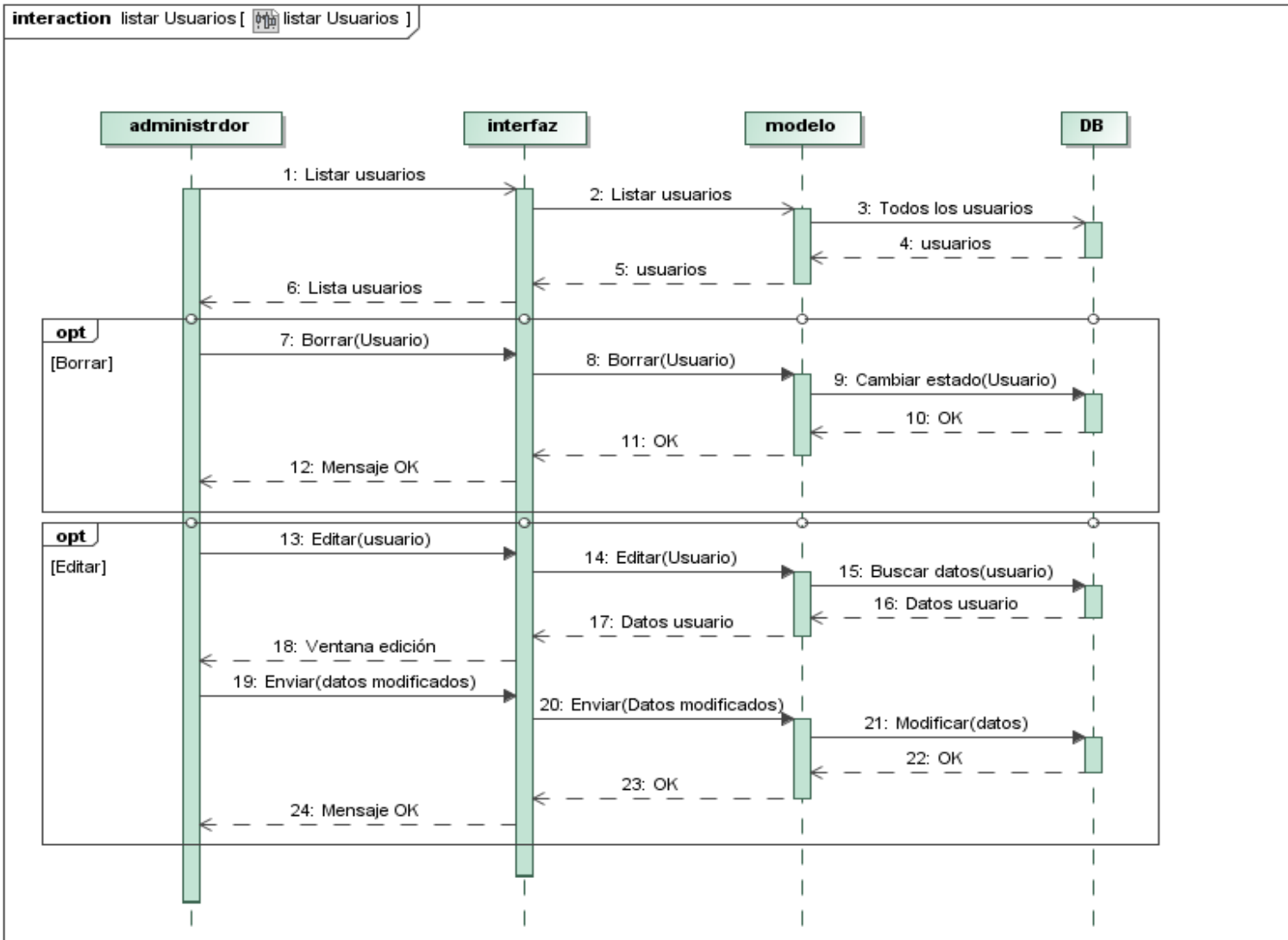
ANÁLISIS. Diagramas de secuencia.

- En esta etapa del análisis, se representa cada uno de los casos de uso del sistema a través de su respectivo diagrama de secuencia.
 - De esta manera modelamos la interacción de los distintos objetos que forman el sistema.
- EJEMPLO 1 (Nueva inserción):**

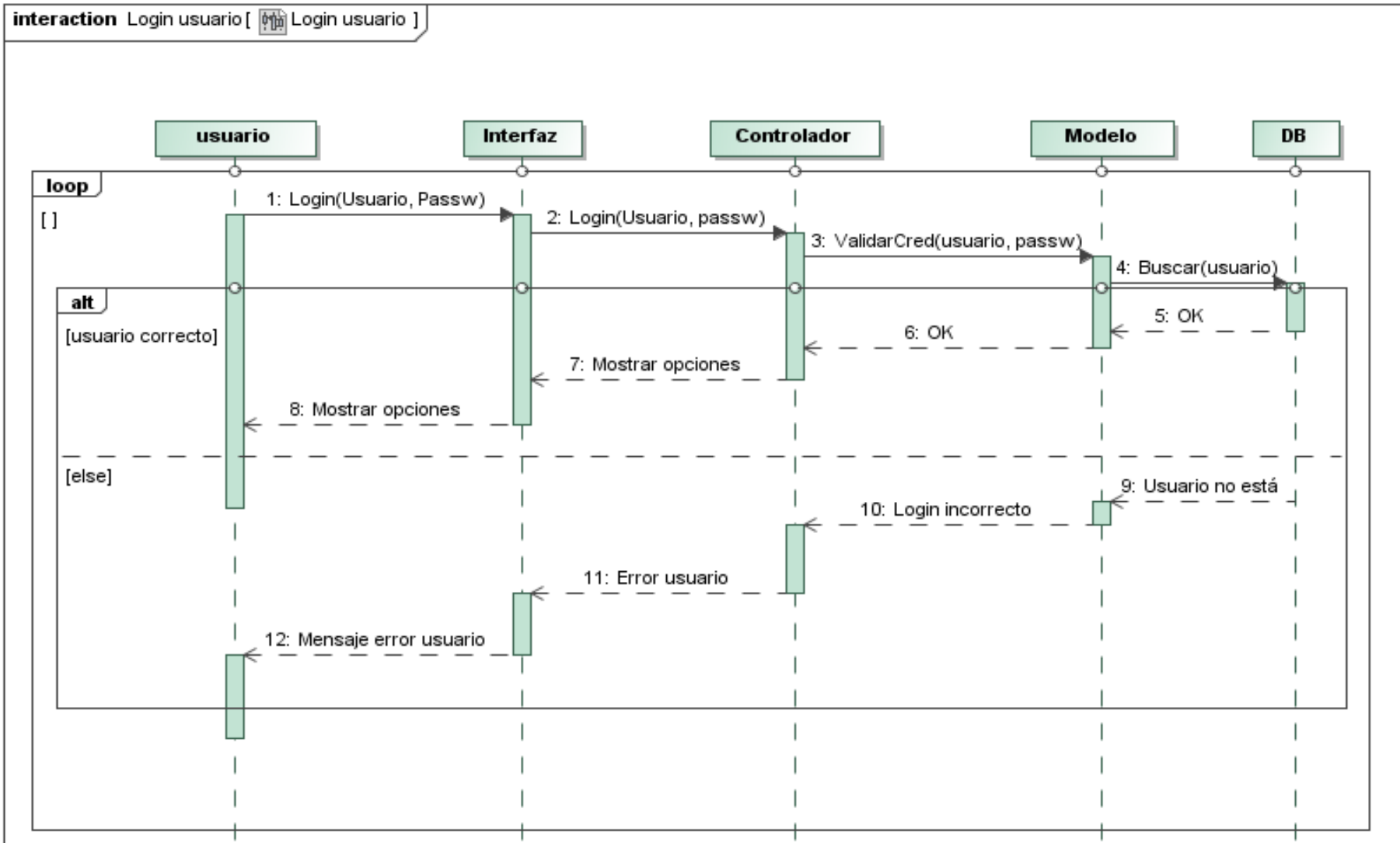


ANÁLISIS. Diagramas de secuencia II.

EJEMPLO 2. (Listar/borrar/Editar)

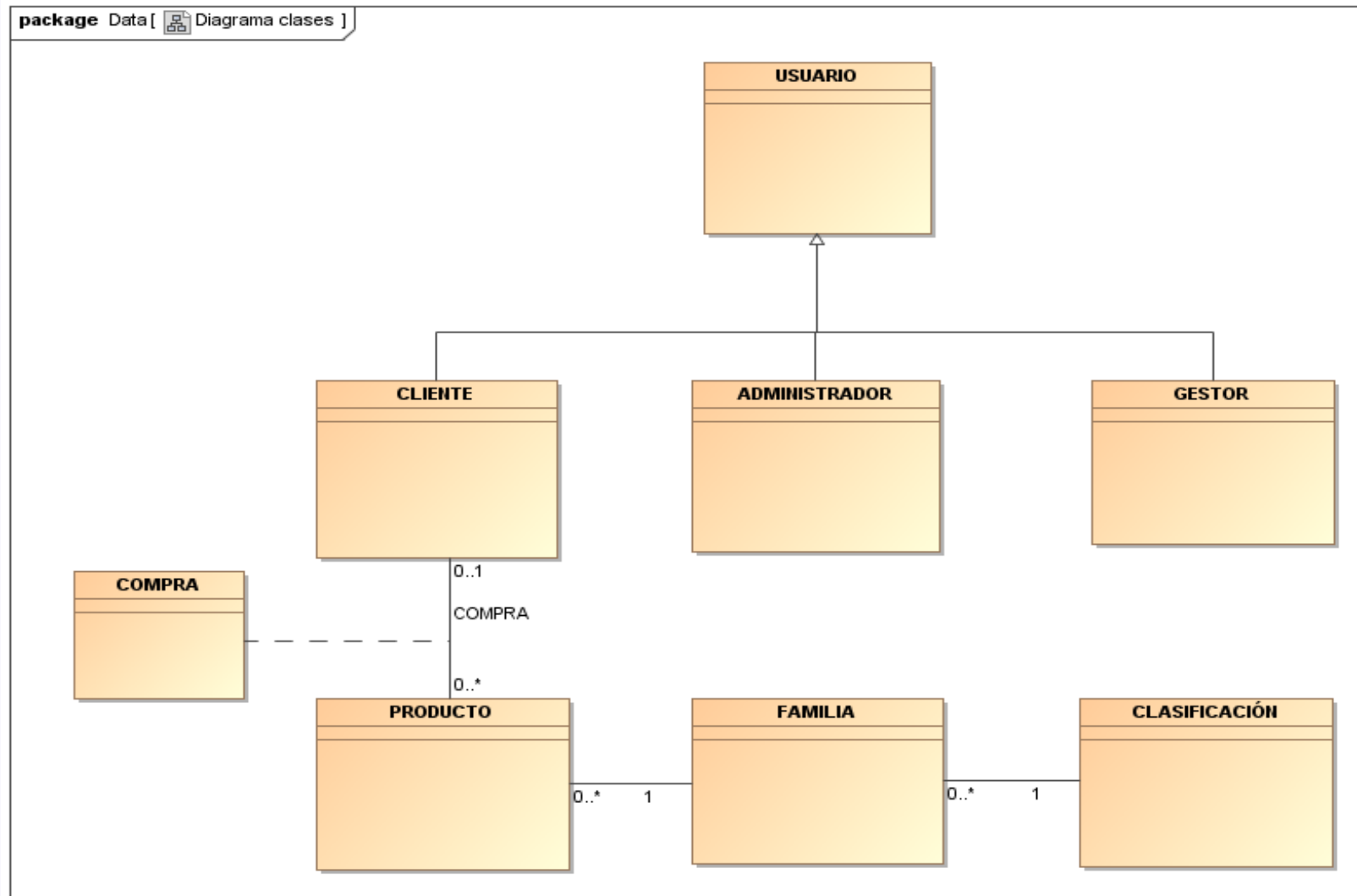


EJEMPLO 3. (Login)



ANÁLISIS. Modelo de datos.

Diagrama Entidad – Relación: Entidades del sistema de información

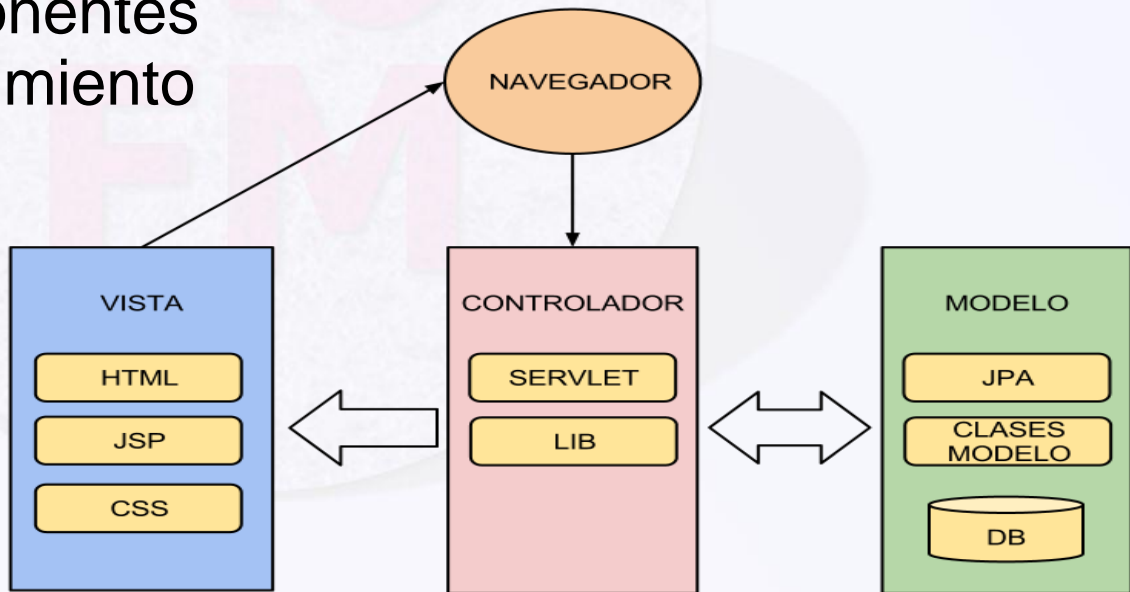


DISEÑO. Uso de patrones.

Se utiliza el **patrón MVC** (Modelo Vista Controlador) que es un patrón de diseño que separa los datos (modelo), la interfaz (vista) y la lógica de negocio (controlador)

VENTAJAS

- Escalabilidad
- Reutilización de componentes
- Sencillez en el mantenimiento



DISEÑO. Uso de patrones II.

Se utiliza el patrón **Singleton** que restringe la creación de objetos de una clase a un único objeto.

1. Objeto Entity Manager Factory.
2. Objeto HttpSession.

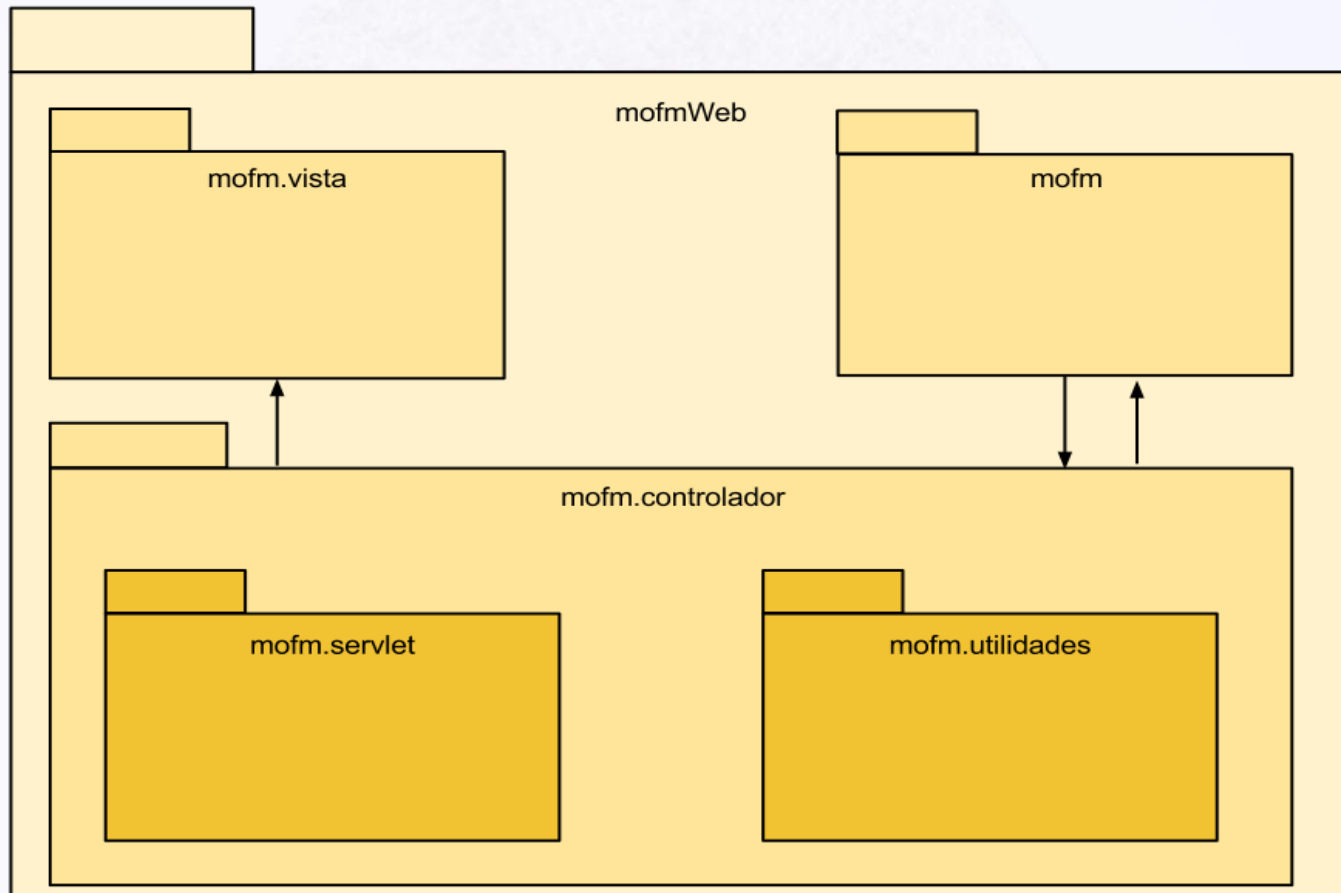
VENTAJAS

- Al tener sólo un objeto Entity Manager Factory garantizamos que una operación costosa como es la creación del objeto sólo se realiza una vez. Utilizando durante todo el código el mismo objeto.
- Al mantener siempre un único objeto HttpSession, durante todo el ciclo de conexión de un mismo usuario mantenemos el mismo objeto permitiéndonos su uso para el guardado de parámetros asociados a la sesión del usuario.

NOTA: El uso de el objeto httpsession es en si mismo otro patrón de diseño utilizado.

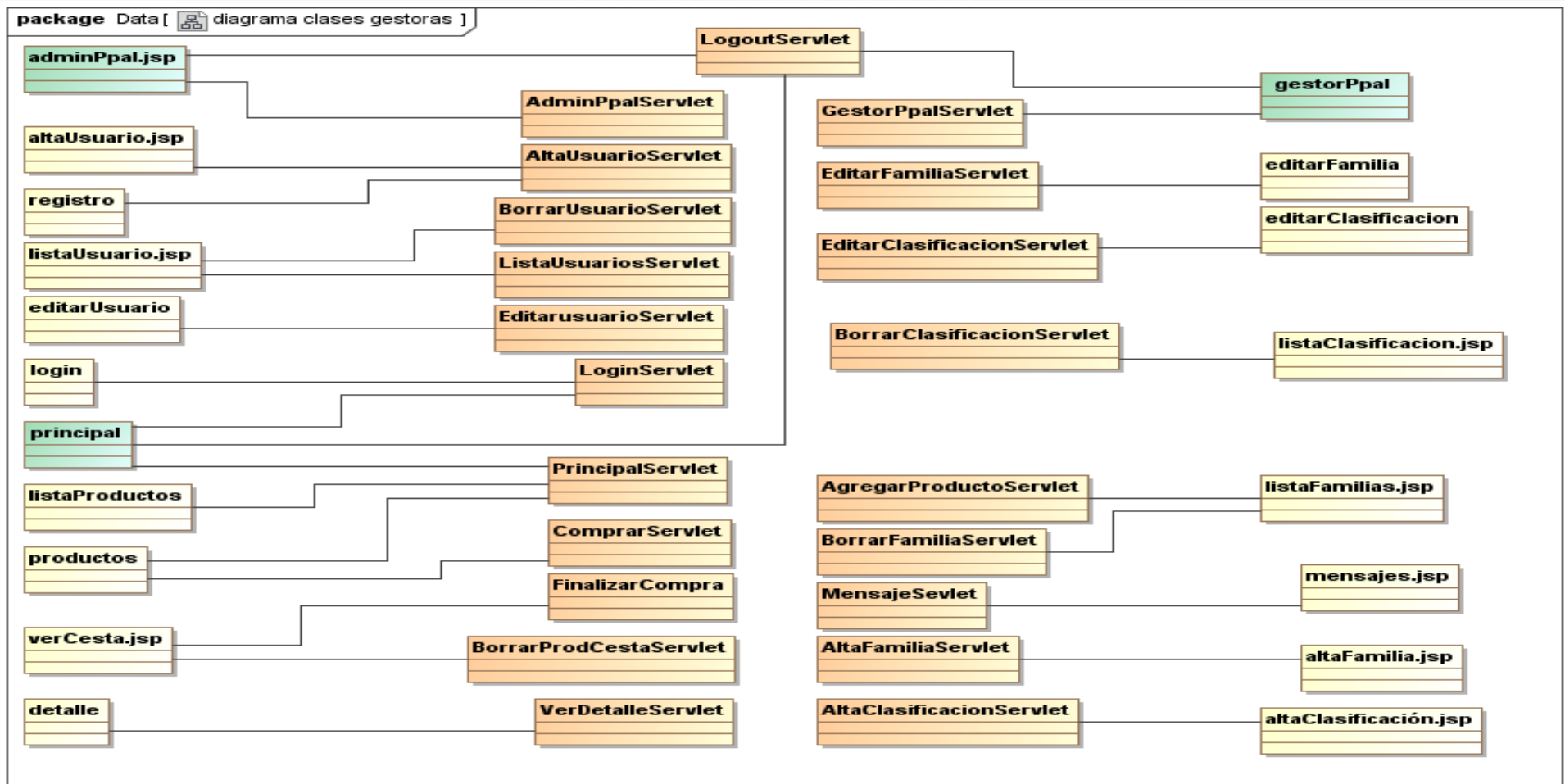
DISEÑO. Paquetes.

Se ha utilizado la siguiente estructura de paquetes basada en la estructura del modelo MVC:



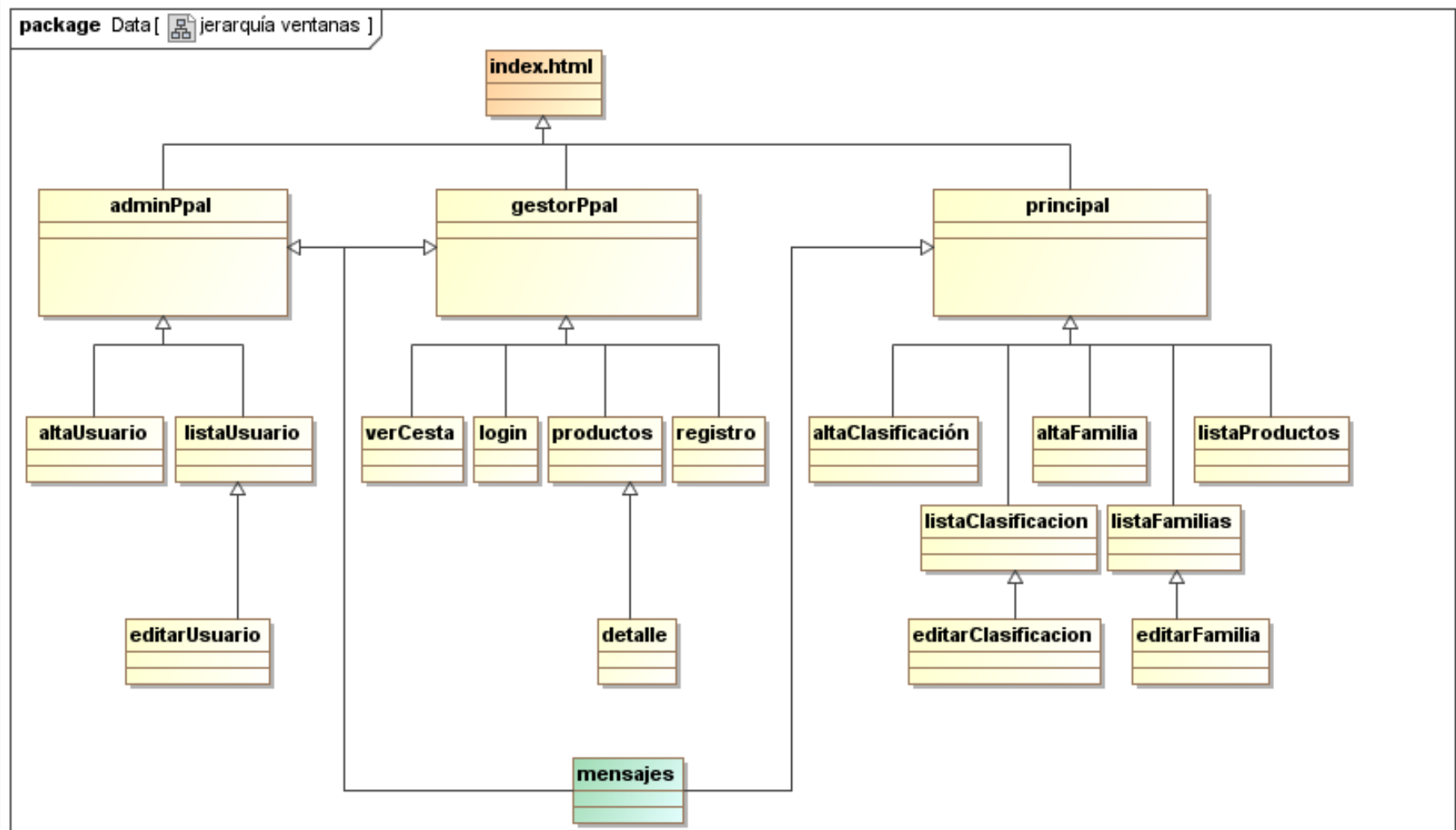
DISEÑO. Diagrama gestores.

En la parte del controlador, éstos son los gestores que realizan cada una de las funcionalidades del sistema.



DISEÑO. Jerarquía de ventanas.

En la parte de la vista, tenemos la siguiente jerarquía de ventanas:



El diseño de la interfaz de la aplicación tendrá el siguiente aspecto:

MADE OF FELT MARKET

Mo FM

MENÚ

- ACCEDER(LOGIN)
- REGISTRARSE
- VER CESTA
- PRODUCTOS**
- TODOS
- ANIMALUCHOS
- COSAS

Bienvenido. Seleccione su fieltro favorito de entre nuestros productos

Imagen no disponible Imagen no disponible Imagen no disponible

Referencia: 1
Nombre: Perro
Precio: 4,25 €
🛒 🔍

Referencia: 2
Nombre: gato
Precio: 1,75 €
🛒 🔍
¡AGOTADO!

Referencia: 3
Nombre: avión
Precio: 1,99 €
🛒 🔍
¡AGOTADO!

MADE OF FELT MARKET

Mo FM

MENÚ

- SALIR(LOGOUT)
- LISTAR USUARIOS
- ALTA USUARIO

LISTADO DE USUARIOS DEL SISTEMA

NOMBRE	APELLIDOS	DNI	LOGIN	EDITAR	BORRAR
David	Silva	44444888N	dsilva	✏️	🗑️
Sergio	Ramos	33242343G	sramos	✏️	🗑️
Gerard	Piqué	22222890Y	gpique	✏️	🗑️
Xabi	Alonso	55345654U	xalonso	✏️	🗑️

IMPLEMENTACIÓN. NetBeans.

La implementación se ha realizado con el entorno de desarrollo NetBeans.

LIBRERÍAS UTILIZADAS

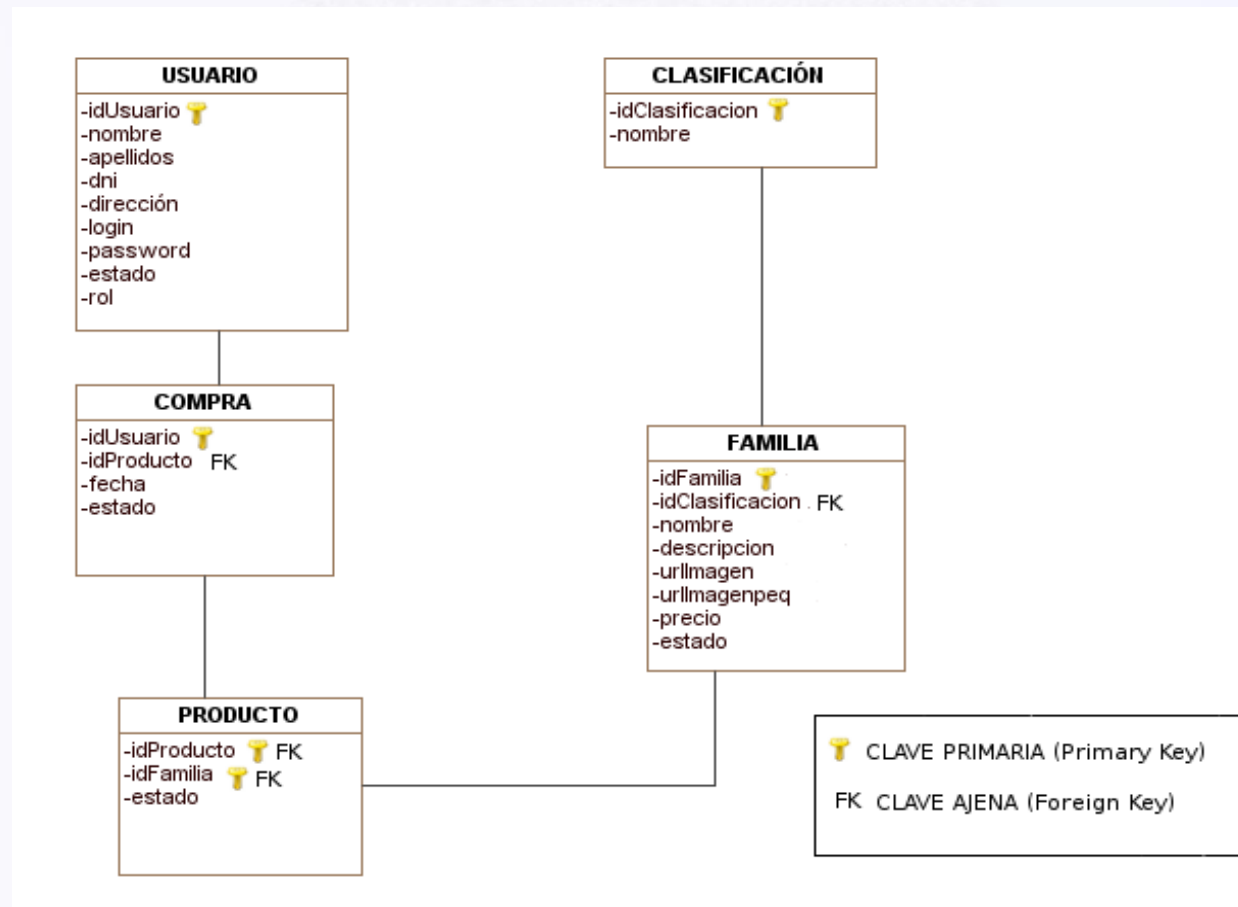
- Se ha optado como implementación de JPA la librería Hibernate.
- También se ha optado por el uso de Persistence como API de JPA.
- Necesitaremos tener el driver JDBC para el SGBD utilizado que en nuestro caso es MySQL.
- Por último utilizaremos una librería de TAGS para JSP. En este caso será JSTL 1.1.

EJECUCIÓN

- Para la ejecución del programa se ha utilizado Apache Tomcat. La ejecución se ha realizado a través del IDE.

IMPLEMENTACIÓN. MySQL.

Se ha optado por el sistema gestor de bases de datos **MYSQL**. Aunque utilizando la herramienta visual MySQL Workbench 5.2. El diseño de la base de datos queda de la siguiente manera:

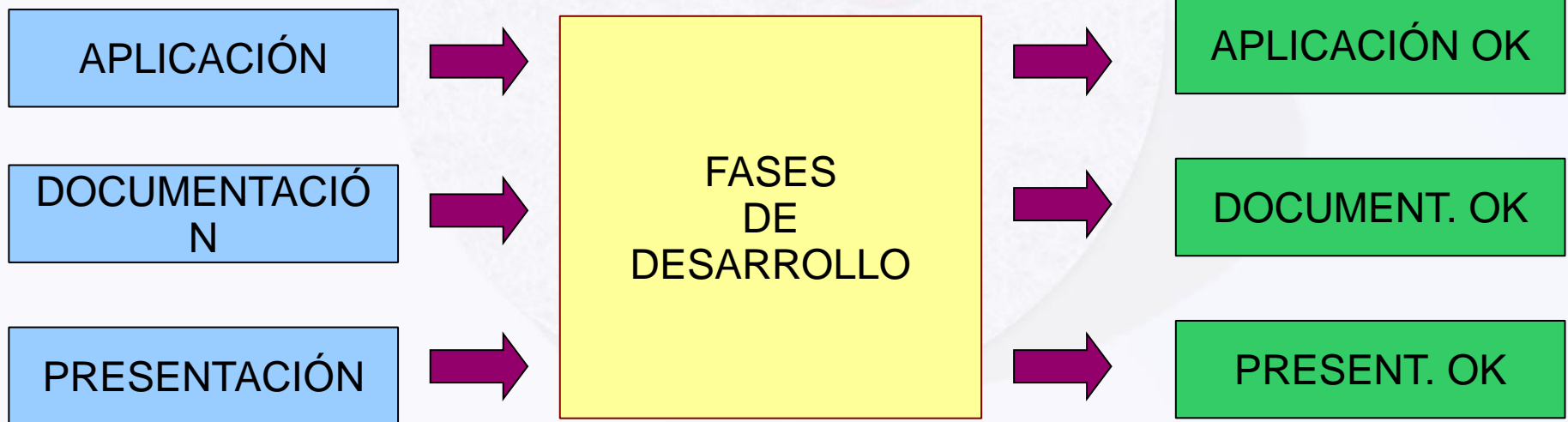


CONCLUSIONES.

ELECCIÓN DEL PROYECTO

- Intenso uso que se hace de los entornos web para la realización de aplicaciones.
- Alta demanda de aplicaciones con acceso remoto.
- Futuro prometedor para este tipo de tecnología.

OBJETIVOS DEL PROYECTO



Aprendizaje de herramientas J2EE, desarrollo de un proyecto completo OK

