

Aplicación Web para la Gestión del Inventario de un Almacén

Autor: Jose Mario Celda Galbis

Director: Pedro J. Valderas

INDICE

1. Introducción	4
- Nacimiento del Proyecto	4
- Objetivos Generales del Proyecto	5
- Estructura del Documento	5
1.1. ¿Qué es un Almacén?	7
1.1.1. Definición	7
1.1.2. Clasificación Según Construcción	8
1.2. Actividades de Almacenamiento	8
1.2.1. Recepción: Responsabilidades que Incluye	9
1.2.2. Características de los Muelles y Áreas de Recepción	9
1.2.3. Almacenamiento: Sus Diferentes Actividades	10
1.2.4. Planificación de Almacenamientos	10
1.2.5. Inventarios.....	11
1.3. La Logística.....	12
1.3.1. El Origen	12
1.3.2. Definición	12
1.4. Operaciones en el Almacén	13
1.4.1. Operaciones de Entrada.....	13
1.4.2. Operaciones de Salida	14
1.4.3. Operaciones de Control.....	16
1.5. Los Stocks	16
1.5.1. Definición	17
2. Especificación del Problema	18
3. Arquitectura utilizada	20
- Concepto de Multicapa.....	20
4. Planificación.....	26
- Descomposición en Fases del Proyecto	27
5. Análisis detallado.....	33
- Casos de Uso.....	33
Pantalla de Login	35
Menú Principal	36
1.- Alta Producto	37
2.- Alta Ubicación	39
3.- Entrada Producto	41
4.- Salida Producto.....	43
5.- Movimiento Productos.....	45
6.- Listado de Productos	47
6.5. Operaciones en Ficha de Producto	52
7.- Listado de Ubicaciones	56
8.- Listado de Movimientos	65
9.- Listado de Inventario.....	69
- Diagrama de clases.....	73
- Modelo Relacional.....	75
6. Implementación	78
Entorno	78
STRUTS (Tecnología de Implementación J2EE + JSP)	86
Herramientas	95
7. Manual de Usuario.....	96
7.1. Introducción	96
7.2. Como acceder a la aplicación	97
7.2.1. Para acceder mediante el terminal (INTERMEC)	97
7.2.2. Para acceder desde cualquier PC	98

7.2.3. Validarse en la aplicación.....	99
7.3. Menú principal.....	100
7.3.1. Alta de producto	100
7.3.2. Alta Ubicación	103
7.3.3. Entrada producto.....	106
7.3.4. Salida producto	108
7.3.5. Movimiento de productos	111
7.3.6. Listado de productos	114
7.3.7. Listado de ubicaciones.....	120
7.3.8. Listado de movimientos.....	125
7.3.9. Listado Inventario	127
7.3.10. Cerrar Sesión	129
7.3.11. Funcionalidad “Buscar” en los listados	130
7.3.12. Funcionalidad “Orden” en los listados	133
8. Bibliografía.....	135

1. Introducción

El presente Proyecto Fin de Carrera se encuentra enmarcado en el área de la Logística y el almacenamiento de mercancía, mas concretamente en la recepción, gestión y salida de los productos en un almacén.

Para la correcta comprensión de las funcionalidades que se deberán implementar en la aplicación y los diferentes términos que se utilizaran en este documento podemos ver las definiciones en el apartado **“1.1. ¿Qué es un Almacén?”**.

En él se presenta el desarrollo de una aplicación para la gestión de un almacén. Para el correcto desarrollo de esta aplicación será necesario gestionar de manera correcta las diferentes actividades que se suelen llevar a cabo en un almacén, de este modo se podrá tener mucho más controlado el proceso de recepción y salida de mercancía, al igual que podremos tener una información exacta y en tiempo real del stock de productos del que disponemos en nuestro almacén. Esta información será muy útil para el personal de compras y la dirección de la empresa, ayudándoles en los procesos de toma de decisión y en la realización de las previsiones.

El desarrollo del proyecto se ha llevado a cabo utilizando el lenguaje de programación Java junto con los entornos/herramientas: MS Project para la planificación, ArgoUML para el modelado resultante del análisis, así como las herramientas para el desarrollo Eclipse HELIOS, MySQL y JasperReports.

- **Nacimiento del Proyecto**

El cliente es una empresa que dispone de un almacén, en dicho almacén hay un encargado y cinco operarios. La empresa se plantea la implantación de un sistema informático tras surgir problemas tales como los inventarios periódicos y las perdidas de tiempo en búsquedas de productos que no aparecen en ninguna parte y nadie sabe donde están ubicados.

Se busca automatizar el inventario con el nuevo sistema de gestión, y obtener la información necesaria para la gestión del negocio mediante la extracción de informes.

El Gerente de la empresa y el encargado del almacén expusieron el problema, explicando que es lo que esperaban del programa, después de varias reuniones y tras estudiar el proyecto, se les pidió autorización para organizar reuniones con los operarios y que estos demostrasen como desempeñaban su trabajo diario, de esta forma podría tener una visión mas amplia de su forma de trabajar y podría adecuar el nuevo sistema tanto a las necesidades de sus superiores como a las suyas. Una vez analizada la forma de trabajar y siempre ciñéndome a las peticiones del gerente y el encargado realice una propuesta de proyecto y este fue aceptado.

- **Objetivos Generales del Proyecto**

El proyecto a realizar debe cubrir las necesidades de la empresa.

Se necesita un programa para gestionar la Recepción y Salida de productos en un almacén, también necesitaremos acceder a la información de donde están ubicados los diferentes productos mientras se encuentran dentro de nuestros almacenes y poder revisar un histórico de los diferentes movimientos o cambios de ubicación, para poder hacer un seguimiento de las ubicaciones por las que estos han ido pasando y poder localizar dichos productos en caso de pérdida.

La información podrá ser consultada por los usuarios a través del terminal, un PC o un Smartphone conectado a la red local de la empresa.

Se les ha ofrecido como mejora la extracción de listados en formato .PDF y .XLS para imprimir si el usuario lo desea ó poder trabajar con la información de los listados en una hoja de cálculo.

- **Estructura del Documento**

A continuación se explica la estructura y breve descripción del contenido del resto de este documento:

- **Capítulo 2:** Especificación del problema. En este capítulo se describe de forma informal el problema al que nos hemos enfrentado en el desarrollo del presente proyecto.

- **Capítulo 3:** Arquitectura utilizada. Descripción de la arquitectura de tres capas que se ha seguido para este proyecto, la cual es de gran utilidad en aplicaciones Java.

- **Capítulo 4:** Planificación. Especificación de las fases en las que se ha dividido el proyecto y estructuración de las diferentes tareas en cada fase con su correspondiente diagrama de duración.

- **Capítulo 5:** Análisis detallado. Estudio del proyecto y modelado de los diferentes diagramas que describen la estructura del sistema, necesarios para su desarrollo.

- **Capítulo 6:** Implementación. Herramientas y entornos utilizados para la implementación del proyecto

- **Capítulo 7:** Manual de usuario. Guía detallada para el uso de la aplicación por el cliente.

- **Capítulo 8:** Bibliografía. Autores, títulos y referencias de los textos utilizados como soporte para la realización del proyecto.

1.1. ¿Qué es un Almacén?

1.1.1. Definición

Para poder hablar de un almacén, en primer lugar deberíamos definir su **concepto**: un **almacén** básicamente es un espacio, recinto, edificio, o instalación donde se suele guardar la mercancía, pero al mismo tiempo puede hacer otras funciones, como por ejemplo el acondicionamiento de productos determinados, hacer recambios (tanto para el mantenimiento como para la existencia técnica), etc., más profundamente diríamos que el término almacén viene derivado del árabe (almaizan) y es una casa o edificio donde se guardan géneros de cualquier clase.

Por tanto, un almacén fundamentalmente se encarga de guardar el stock, pero no debemos de confundir los términos. La gestión del stock no será la misma que la gestión del almacén. La primera se encarga de aprovisionar para un buen nivel de servicio mientras que la segunda intenta realizar las operaciones de almacenamiento (algunas veces también de preparación y producción) con los mínimos recursos propios del almacén (como son el espacio, la maquinaria y el personal).

De esta forma, para la gestión del almacén, la gestión del stock se convertirá en proveedora de servicios logísticos de almacenaje y preparación.

Tras el análisis podemos llegar a las siguientes conclusiones:

El almacén no solo servirá para almacenar sino también para preparar la entrega al cliente y algunas veces operaciones de producción.

Finalmente el almacén es un recinto (tanto abierto como cerrado) ordenado para cumplir las funciones de almacenamiento y acondicionamiento que se hayan definido previamente.

A continuación definiremos los siguientes términos:

- Alveolo o compartimento → Huecos donde entran los palets o se depositan los materiales de la misma clase

- Ciclo Combinado→ Son una serie de operaciones que sufre un artículo desde que llega al almacén hasta que se deposita en su alveolo. Formar de moverlo.

- Ciclo Simple→ Es cada una de las partes en las que se divide el Ciclo Combinado.

- Inventario→ Operación que permite saber la cantidad y emplazamiento del material.

- Inventario Permanente→ Método con el que en cualquier momento se sabe la cantidad de material que se tiene.

- Lifo→ Regla de gestión de stock.

- Picking→ Operación de carga de artículos desde su lugar de almacenamiento para completar el pedido.

- Terminal a bordo→ Pantalla con teclado instalado en una carretilla para conocer la situación de un almacén.

- Almacén→ Es el edificio o local donde se depositan géneros de cualquier categoría.

1.1.2. Clasificación Según Construcción

a) Patios de materiales y contenedores→

No se pueden guardar en almacenes corrientes y se deben depositar al aire libre debido a su volumen (ladrillos, etc).

c) Naves de todo tipo→

Grandes edificios muy elementales y según el tipo de nave, hay grúas para el movimiento del material. Tienen que ser adecuadas al fin que se ocupan.

1.2. Actividades de Almacenamiento

1.2.1. Recepción: Responsabilidades que Incluye

Producción→ Se almacena la producción.

Planta Física.

Administración.

Recepción→ de todas las materias primas.

Se necesita el Almacén de Entrada, Almacén de Salidas y las Expediciones.

Se necesita la Oficina de Producción y por último el Material de Manejo.

Obtenemos el material y lo tenemos que llevar al almacén.

Almacén de Entrada para producción.

Almacén de Salida para servir el stock necesario.

Expediciones→ aquellos movimientos hasta que llega al cliente.

Actividad de Recepción

- a) Descargar el material de los transportistas.
- b) Desempacar el material
- c) Identificar y seleccionar el material.
- d) Comprobar los albaranes del transportista con lo que recibimos.
- e) Comprobación.
- f) Anotar daños y defectos del material.
- g) Llevar los controles adecuados.
- h) Despachar el material para el área de uso o estantería correspondiente.

1.2.2. Características de los Muelles y Áreas de Recepción

Se debe tener en cuenta la capacidad de llegada para no tener retrasos o aglomeraciones en el muelle o punto de descarga.

El diseño de los muelles ha de ser adecuado a nuestro volumen de recepción.

En las áreas de recepción se controla la mercancía recibida para ver si corresponde con lo solicitado.

1.2.3. Almacenamiento: Sus Diferentes Actividades

Almacenar cualquier producto hasta su uso:

Durante la llegada.

Almacén de material primas y partes→

M.P. (material primas)→ normalmente requieren una transformación posterior.

Partes→ estos son Productos Acabados como motores, pinturas.

Almacén de suministros →

Materiales para el mantenimiento de equipos.

Oficinas→ material de oficina.

Se guardan cerca del área de uso.

1.2.4. Planificación de Almacenamientos

Uso efectivo del tiempo, M.O. (Mano de Obra), equipo.

Usar al máximo el cubo (forma de almacenaje), siempre que sea posible ya que es la mejor.

Usar bien el tiempo, hacer buen uso de él.

Buen uso de materiales, hemos de estar seguros de suministrarlos a tiempo y en el lugar adecuado, en la cantidad solicitada.

Uso efectivo de la Mano de Obra→ si logro ordenar el tiempo haré que la M.O. (Mano de Obra) está bien organizada durante su jornada.

Equipo→ necesitaremos todo el necesario.

Acceso rápido a los items.

Items→ productos almacenados.

Se tiene que almacenar de manera adecuada (A,B,C) del más uso al menos uso.

Facilidad para MM (Mover Materiales).

Necesitaremos que los materiales puedan ser movidos con gran facilidad.

No por ahorrar un trozo de espacio vamos a moverlo mejor, se tendrá en cuenta la manera de mover los materiales (toro, carretilla).

El almacén y área colindante deben estar perfectamente coordinados.

Identificación positiva.

No haya ninguna duda de que los materiales estén perfectamente identificados.

Es absolutamente vital para un almacén medio o grande.

Máxima protección.

Los costes, si no se protege, son muy grandes.

Tanto en cuestión de golpes, contaminación.

Deben llevar cierre, seguridad, etc. Con la máxima protección.

Evitar todo lo relacionado con el tipo de material que tratamos.

Orden.

No debe haber nunca nada fuera de su sitio y un sitio para cada cosa.

Facilitar la buena marcha del almacén.

1.2.5. Inventarios

Definición.

Consiste en una enumeración precisa de lo que contiene el almacén, en calidad y cantidad.

Esta operación es indispensable para saber que se dispone.

El inventario debe ser permanente, tanto en salidas como en entradas.

El inventario internamente se realiza sólo una vez al año.

Desviaciones reales.

Despistes en la introducción de datos.

Productos que se dañen en el almacén (caducados, rotos, etc).

Daños durante la carga en un nicho que no le corresponde.

El inventario físico ha de ser igual al inventario administrativo.

Control.

Debe coincidir los datos introducidos con lo que realmente tenemos en el almacén.

Comprobación de cantidades, peso (neto).

Control de marcaje → la marca debe corresponder al producto.

Control del embalaje → para preservar la mercancía.

1.3. La Logística

1.3.1. El Origen

Etimológicamente, el término logística viene del griego *logistikos* (aquella que sabe aplicar el cálculo). Posteriormente el logístico era el administrador o intendente del ejército del imperio romano; comenzando en este periodo a utilizarse como término militar, y que durará a lo largo de la existencia del imperio bizantino.

A finales del siglo XVI se introdujeron las expresiones "logística numerosa" y "log. espaciosa" (para el cálculo mediante los números y las letras respectivamente) pero este carácter militar llegará hasta el siglo XX.

Es después de la Segunda Guerra Mundial cuando el concepto logístico entra en el término civil con tal de identificar las actividades de la empresa dedicadas a utilizar los recursos de forma racional y dentro de ellos los dedicados a la operativa diaria de la empresa (aprovisionar y suministrar productos).

1.3.2. Definición

La logística es un conjunto de actividades que tenían la finalidad de conseguir la disponibilidad de todos los recursos necesarios en el momento, lugar y cantidad concreta, sin duda al mínimo coste. Así, el concepto de logística va asociado con la óptima gestión de aquellos recursos necesarios con tal de conseguir su finalidad (poner productos de utilidad a disposición de los consumidores, prestación de servicios al cliente...).

Estos recursos que se van a utilizar pueden agruparse en las siguientes categorías:

- Recursos materiales tipo producto/mercancía.
- Recursos materiales operativos (ej: infraestructuras – fabricas, almacenes, maquinaria...-)

- Recursos humanos operativos(ej: personal)

- Recursos de utilidad a consumir (ej: agua, electricidad...)

Su definición nos indicaba que se trata de que los recursos relatados anteriormente estuvieran en el momento (cuando), la cantidad (cuanto) y el sitio (dónde) adecuado al menor coste posible; es decir, la logística busca el equilibrio entre el servicio(cuando, cuánto y dónde), y el coste.

En conclusión, la actividad logística no solo es un nivel equivalente de calidad al mínimo coste, sino también un servicio.

1.4. Operaciones en el Almacén

Las empresas necesitan tener un sistema de estudio de las entradas y las salidas tanto para posibles pérdidas en ventas (carencia de mercancías), como por los costos del mantenimiento y conservación, capital invertido (existencia excesiva de stocks)

1.4.1. Operaciones de Entrada

Lo primero que hay que hacer cuando entran productos es identificar el origen y destino para comprobar que no es un error. Seguidamente se identifica el envío para descargarlo por la zona adecuada.

Ahora ya esta todo preparado para descargar la mercancía. Se avisa al personal cual será la ubicación de destino, “la playa” de descarga, el área de devoluciones, zona de preparación de envíos, o el área de cuarentena (es la zona dónde las mercancías reposan un tiempo antes de ser aptas para vender), como por ejemplo, las cámaras frigoríficas en los almacenes de naranjas.

Durante la descarga hay que hacer una comprobación o recepción externa del género (lo que esté dañado externamente tiene que ser devuelto, pudiendo descargarlo o no, según el acuerdo establecido).

Una vez finalizada esta fase se firma la documentación del transportista dándole las copias correspondientes. A partir de este momento el transportista puede abandonar la instalación.

Seguidamente se comprueba la carga (cantidad y calidad) y se separan las mercancías aceptadas, devueltas y las pendientes de algún control. Las aceptadas (normalmente con una etiqueta verde) tienen que ser almacenadas, por lo que hay que buscarle la ubicación más idónea para su peso forma y dimensión.

Finalmente debemos introducir la información sobre la actividad realizada en el sistema informático con las características que se hayan podido dar (incidencias o errores).

1.4.2. Operaciones de Salida

Igual que en el apartado anterior, analizaremos las diferentes fases de las operaciones de salida del almacén:

1.4.2.1. Extracción.

La extracción de mercancías para finalmente enviarlas al cliente se puede hacer de dos maneras

- Extraer los productos en las unidades logísticas especializadas.
- Extraer unidades sueltas (picking). Suele ser más habitual mientras más nos acercamos al consumidor o usuario. Es decir, el producto suele entrar en pocos envíos de mucha cantidad, y sale en muchos de poca. (Ver Figura 1)

- Entregarlos al transportista, habiendo firmado antes la documentación correspondiente.

1.4.3. Operaciones de Control

1.4.3.1. Recuentos físicos: el inventario.

El inventario es la lista dónde se inscriben y describen, artículo por artículo, todos los bienes que pertenecen a una persona o están en una casa. Hablando en términos contables, es la relación y valoración de los bienes, derechos y obligaciones de una empresa, que expresa la estructura de su patrimonio en un momento dado. Podemos hablar de dos casos de inventarios:

- **contable:** estado de cuentas que permiten conocer, durante todo el periodo de explotación, las existencias en stock.

- **extracontable:** recuento en el almacén de las existencias realmente mantenidas en stock.

El procedimiento casi siempre es el mismo:

- Preparación de una orden para cada ubicación.
- Programación de ordenes de recuento y también de colocación / extracción.
- Incorporación de los resultados al sistema informático.

Entre todos nombramos tres tipos de inventarios:

- Completo (normalmente una vez al año)
- Rotativo. Consiste en verificar cada vez unos productos.
- Permanente. Consiste en contar las existencias cada vez que se da una operación sobre una ubicación.

1.5. Los Stocks

No podríamos hablar completamente de almacén si no intentáramos descubrir su esencia: los artículos almacenados, por lo que los siguientes apartados los dedicaremos única y exclusivamente a esta parte.

1.5.1. Definición

Los stocks los podemos definir como los artículos en espera de su utilización posterior.

La posesión de estos, es la manera que tienen las empresas para garantizar el buen funcionamiento de su actividad, no solo a la hora de servir a un cliente sino también para poder tener materias primas con las que fabricar.

Esta necesidad de tener existencias en el almacén, es para disponer del producto en la cantidad necesaria, en el momento oportuno, con una seguridad de calidad y un precio más económico.

Una pequeña reducción del porcentaje de los stocks significará un gran aumento de los beneficios.

2. Especificación del Problema

Este proyecto consiste en la realización de una aplicación para el control del stock de productos en un almacén. En dicha aplicación se pretende dar una información al usuario a cerca de la ubicación y cantidad de todos los artículos de un almacén, al igual que se le permitirá realizar movimientos de material, y tener controladas las entradas y salidas de mercancía.

He analizado las tareas habituales que se llevan a cabo en un almacén, la distribución de sus ubicaciones y la gran variedad de productos que se suelen manejar, así he podido sacar unas conclusiones previas de las necesidades que tendría que cubrir y de los usuarios a los que va dirigida la aplicación.

El usuario tipo para el que se desarrollada la aplicación, es un operario de almacén, sea este más veterano en el puesto o menos, con mayor experiencia o menos en el campo de la informática. Las tareas que estos usuarios realizaban por pura lógica y a base de la experiencia, para la ubicación y posterior búsqueda de los artículos, gestión de inventario, etc. A partir de la puesta en marcha de la aplicación se realizara automáticamente, ahorrándoles búsquedas innecesarias y perdidas de tiempo en realizar inventarios de forma continuada.

El usuario avanzado o “Administrador”, se encarga de añadir y modificar los usuarios y sus contraseñas para que puedan acceder al sistema.

Las funcionalidades que se proponen son:

MANTENIMIENTO UBICACIONES

- Alta de ubicación
- Edición de ubicación
- Baja de ubicación
- Ver ficha de ubicación

MANTENIMIENTO PRODUCTOS

- Alta de producto
- Edición de producto
- Baja de producto
- Ver ficha de producto

CONTROL INVENTARIO

- Entrada almacén
- Salida almacén
- Movimiento interno de producto

LISTADOS

- Listado histórico de movimientos
- Listado del inventario
- Listado de productos
- Listado de ubicaciones
- Impresión de etiquetas de código de barras.

Dicha aplicación podría ser accedida por el personal de administración y compras de la empresa, a este tipo de usuario le son de gran ayuda los informes de stock actual de los productos y los históricos de movimientos para llevar a cabo sus análisis y posteriormente tomar las decisiones correctas a la hora de comprar material.

La aplicación esta diseñada para acceder a ella mediante terminales de radiofrecuencia con lector de códigos de barras, aunque debido a que esta en un entorno Web se puede acceder a ella desde cualquier ordenador o SmartPhone introduciéndole la URL apropiada.

Respecto a las técnicas y estándares que se van a utilizar para la especificación se utilizará el estándar IEEE 830-1998, en el modelado del análisis, es decir casos de uso y diagrama de clases, se usará UML, respecto al diseño se implementará con CSS y HTML validados por w3.org. La aplicación se programa con J2EE y se utilizarán base de datos MySQL, de esta forma utilizaremos tanto para el lenguaje de programación como para la base de datos código libre.

También se pretende utilizar la tecnología JavaScript y Ajax donde sea útil como por ejemplo en la validación de campos de formularios. Para esta aplicación Web se intentará usar una arquitectura de 3 capas, una será la parte del aspecto Web y su navegabilidad, otra el nivel de negocio y la última la de persistencia, es decir, la base de datos. Con dicha arquitectura se desea conseguir una independencia entre capas.

3. Arquitectura utilizada

El lenguaje en el que se ha realizado el proyecto es Java orientado a objetos, con accesos a una base de datos relacional mediante MySQL.

El recorrido que se va a hacer en esta memoria nos presentará, en un primer momento, la arquitectura que va a seguir la aplicación, a fin de conseguir comunicar el código fuente con la base de datos de este sistema de información.

- **Concepto de Multicapa**

Para la estructuración del presente proyecto se ha optado por una organización en niveles (capas). De esta forma, la partición del sistema en componentes y la relación existente entre ellos siguen un esquema incremental, donde los niveles inferiores proporcionan servicios a los superiores, existiendo una comunicación bidireccional entre las capas. Mediante esta definición de la arquitectura del sistema de información, se consigue un cierto nivel de aislamiento entre capas que facilitará el mantenimiento, consistencia y versatilidad de la aplicación.

Las capas en las que se ha dividido el proyecto poseen una funcionalidad determinada dentro del sistema. A continuación se presentan las capas con las tareas para las que han sido diseñadas:

- **Capa Presentación:** Es la más externa. En ella se implementan todas las interfaces gráficas que interaccionarán con el usuario, capturando todos sus eventos y comunicaciones y pasándolos a los niveles inferiores.

- **Capa Negocio:** En esta capa lógica se contiene los objetos del dominio del problema, así como las colecciones de estos objetos. A ella le llegan todos los mensajes que produce la capa 'Presentación'. En este nivel se definen todas las funciones y cálculos intermedios que se realizarán con los datos origen antes de presentarlos al usuario.

- **Capa Tuplas-Objeto:** Este nivel contiene todas las funciones necesarias para el paso de la información que se extrae e introduce (si ha habido alguna modificación) en la Base de Datos.

- **Capa de Acceso a BD:** Contiene toda la información referente a las consultas que deben hacerse sobre el sistema de información (la Base de Datos) y transfiere a los objetos de la capa superior.

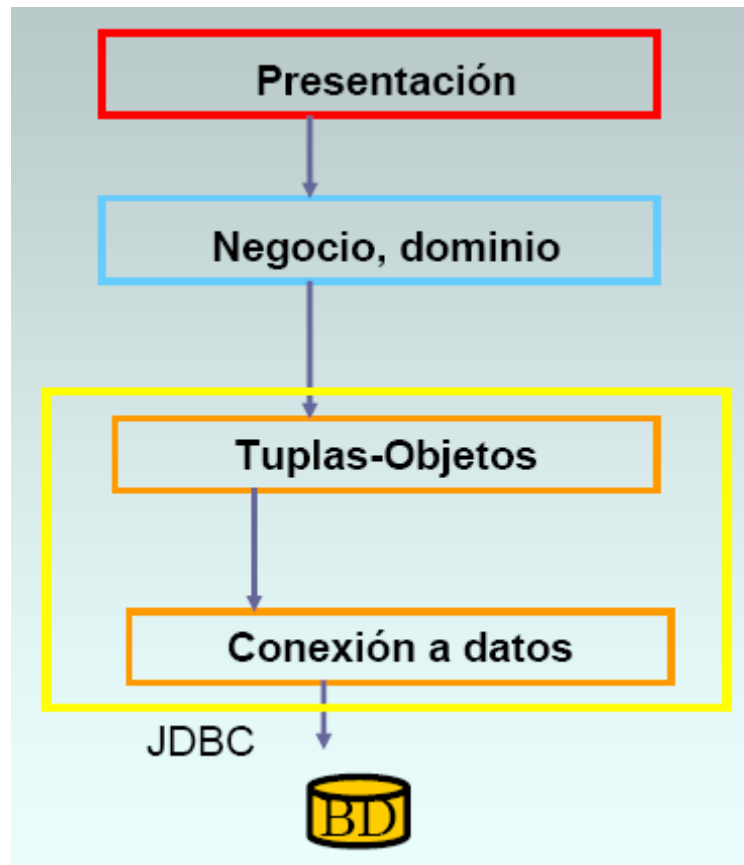


Fig.2 División en Capas

Por lo tanto, las comunicaciones entre los distintos niveles quedarían de la siguiente forma:

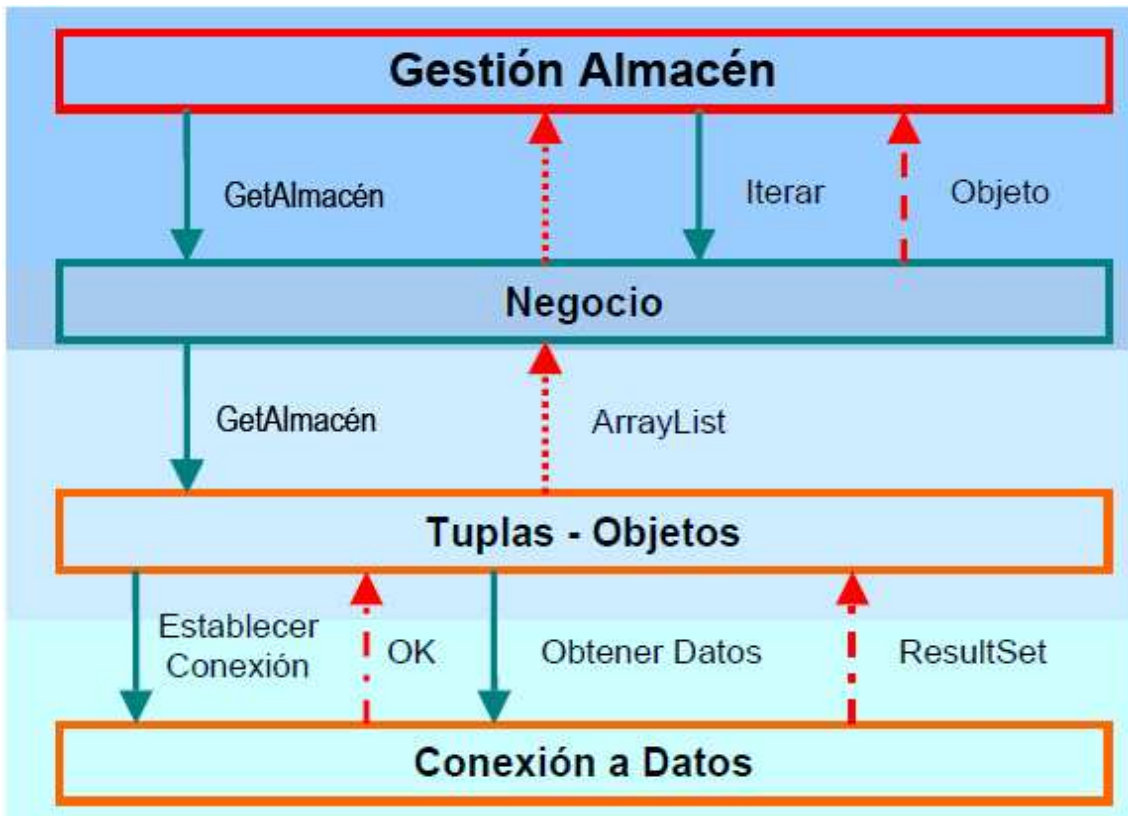


Fig.3 Comunicación entre capas

Dada la estructura de datos empleada (multicapa), la introducción de una nueva clase en el sistema implica la inserción de numerosas secciones de código en los distintos niveles a fin de conseguir que la comunicación entre la interfaz de usuario y la base de datos sea completa y consistente.

La relación de ficheros creados y modificados por cada nueva clase introducida es la siguiente:

- En la aplicación de la base de datos debe de crearse una tabla con el nombre de la clase, que contenga como campos el nombre de los atributos especificados con sus tipos correspondientes.

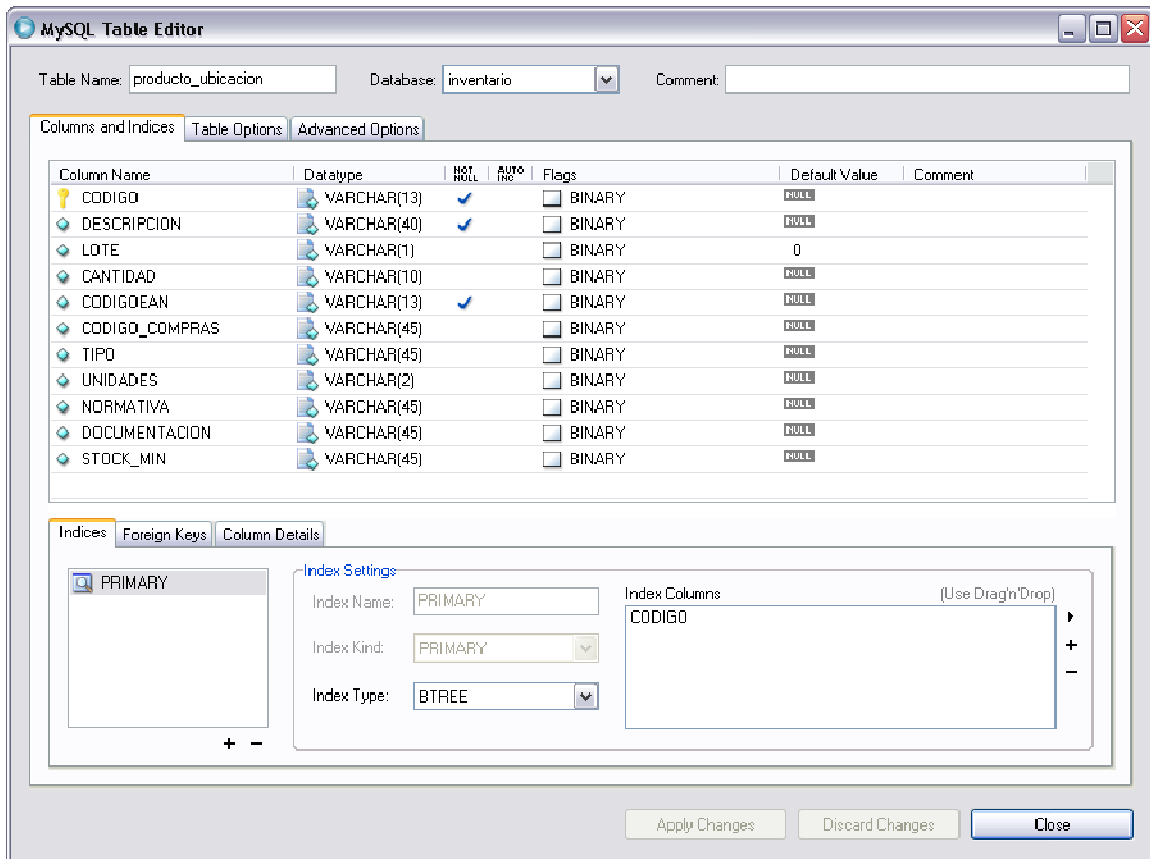


Fig.4 Ejemplo de definición de tabla en MySQL para la clase Producto.

Obtener DatosResultSet

- Se crea un nuevo fichero, que tiene el nombre de la clase, donde se definen ésta, así como todos sus atributos, métodos para poder acceder a ellos y métodos para su mantenimiento (crear, modificar e indicar su estado dentro del sistema de información).

- El nivel de 'Acceso a Datos' está contenido en un fichero general dentro del proyecto, ya que contiene todas las funciones que permiten la creación, establecimiento y cierre de la comunicación con la Base de Datos. Por cada clase, se introducirán dos funciones que contendrán consultas SQL que permitan extraer de la BD los datos referentes a la clase y borrar algún objeto de dicha clase.

- El siguiente nivel, 'Tuplas-objetos', es el encargado de extraer y preparar la información que a salido o va a entrar en la Base de Datos. Por lo tanto, se deberá añadir funciones encargadas de extraer la información de la BD para pasárselas a

'Capa Negocio', insertar nueva información o borrarla, para cada una de las clases del proyecto.

- La 'Capa Negocio' tendrá una lista dinámica de objetos de cada clase. Esta lista será cargada con toda la información de la Base de Datos sobre esa clase y se realizarán los cambios y modificaciones sobre ella, hasta que se decida actualizar la BD con la nueva información. También contiene todas las funciones que pueden hacer falta a la hora de trabajar con la información, por lo que se crean numerosos métodos, tanto sobre el estado de los datos del sistema (obtener todos los elementos de una clase determinada, el número total u obtener datos de una instancia en concreto) como sobre el tratamiento de los eventos que pueden ocurrir (nuevo, borrar, modificar algún atributo, calcular identificadores,...).

- Por último, respecto al nivel de 'Presentación', se generará, por cada clase, un fichero que contendrá una interfaz y todos los elementos necesarios para su gestión, para que el usuario se pueda comunicar con el sistema. El nombre de dicha ventana representara la funcionalidad de la misma, en caso de mostrar un listado, se llamaría "listado" más el nombre de la clase a mostrar en plural, ya que en ella se mostrará un listado completo en forma de tabla de todos los elementos de esa clase que hay en el sistema.



Listado Productos

Fecha: 27/9/2011
Usuario: MARIO

Buscar:

Orden:

Código	Descripción	Lote	Cantidad	Código EAN13
000000000001	BARRERA DOBLE ONDA 4318MM	0	1	0000000000017
000000000002	POSTE TUBULAR 1500MM	0	1	0000000000024
000000000003	SEPARADOR ESTANDAR IZQUIERDO	0	1	0000000000031
000000000004	SEPARADOR ESTANDAR DERECHO	0	1	0000000000048
000000000005	CONECTOR EN ""U"" PARA POSTE TUBULAR	0	1	0000000000055
000000000006	TORNILLO REDONDO CUELLO OVAL M16X30 5.8	0	1	0000000000062
000000000007	TORNILLO REDONDO CUELLO OVAL M16X40 5.8	0	1	0000000000079
000000000008	TORNILLO HEXAGONAL M16X35 5.8	0	1	0000000000086
000000000009	TORNILLO HEXAGONAL M10X40 DIN 933 6.8	0	1	0000000000093
000000000010	TUERCA HEXAGONAL M16 CLASE 5	0	1	0000000000109

Pag. 1 [siguiente](#) >>

Fig.5 Ejemplo de interfaz de la capa de Presentación asociada a la clase Producto

Por otro lado, se incluirá también la funcionalidad necesaria (nuevas ventanas para la creación de nuevas instancias, posibilidad de selección de elementos de la tabla, botones,...) para poder abrir fichas de detalle y realizar las operaciones de crear, modificar y borrar, así como toda la lógica para que el sistema detecte y lleve a cabo todas estas operaciones en la Base de Datos y la actualice manteniéndola en un estado consistente.

4. Planificación

El objetivo de la Planificación del proyecto de Software es proporcionar un marco de trabajo que permita al gestor hacer estimaciones razonables de recursos, costos y planificación temporal. Estas estimaciones se hacen dentro de un marco de tiempo limitado al comienzo de un proyecto de software, y deberían actualizarse regularmente a medida que progresa el proyecto. Además las estimaciones deberían definir los escenarios del mejor caso, y peor caso, de modo que los resultados del proyecto pueden limitarse.

Para realizar la planificación, se divide el proyecto en Fases, Tareas e Hitos. Para este proyecto de desarrollo de software, se han seguido las fases de desarrollo del proceso software, es decir, creando las fases que define una buena planificación de proyectos software. Por tanto, se podría decir que se ha empleado el modelo de creación de software denominado 'Modelo cascada Realimentado'. Ya difícilmente se utiliza, tal cual, el Modelo Cascada, pues esto implica un previo y absoluto conocimiento de los requisitos, la no volatilidad de los mismos y etapas subsiguientes libres de errores, ya que lo ideal sería pasar de una etapa a otra sin tener que rectificar nada de las anteriores. Pero eso es utópico, ya que intrínsecamente el software es de carácter evolutivo, cambiante y difícilmente libre de errores, tanto durante su desarrollo como durante su vida operativa. Por eso es conveniente que se produzca alguna realimentación entre etapas, que no es completamente predecible no rígida; lo cual da oportunidad al desarrollo de productos software en los cuales hay ciertas incertidumbres, cambios o evoluciones durante el ciclo de vida. Para verlo con mayor claridad, podemos observar la figura 6.

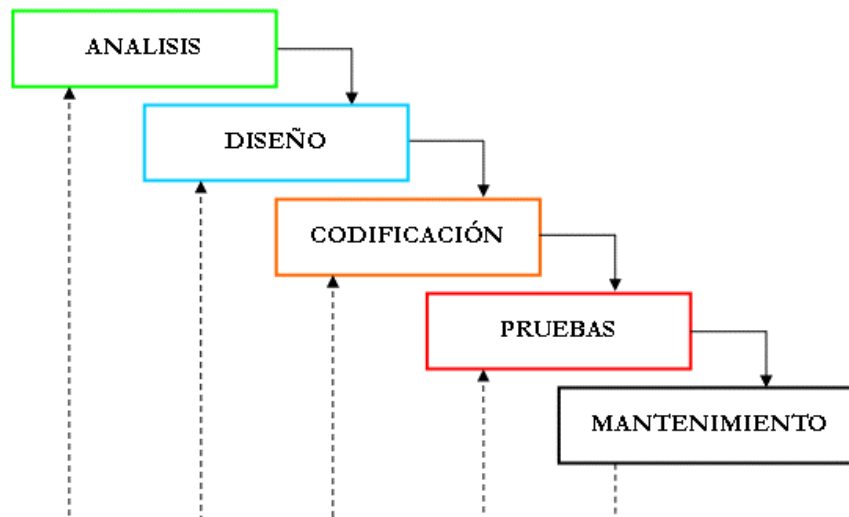


Fig.6 Modelo Cascada Realimentado

- Descomposición en Fases del Proyecto

Para ver con mayor claridad la planificación, se puede observar la figura 7, para poder ver las fechas, la duración y los componentes que llevan a cabo cada tarea.

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1	- Análisis	20 días	lun 02/05/11	vie 27/05/11		Mario
2	Obtencion Necesidades	10 días	lun 02/05/11	vie 13/05/11		Mario
3	Especificacion de Requisitos	3 días	lun 16/05/11	mié 18/05/11	2	Mario
4	Desarrollo de Casos de Uso	5 días	jue 19/05/11	mié 25/05/11	3	Mario
5	Desarrollo Diagrama de Clases	1 día	jue 26/05/11	jue 26/05/11		Mario
6	Desarrollo Diagrama Entidad-Rel	1 día	vie 27/05/11	vie 27/05/11	5	Mario
7	- Diseño	8 días	lun 30/05/11	mié 08/06/11		Mario
8	Identificación de Arquitectura del	2 días	lun 30/05/11	mar 31/05/11		Mario
9	Diseño BBDD	1 día	mié 01/06/11	mié 01/06/11	8;5	Mario
10	Diseño Interfaz	5 días	jue 02/06/11	mié 08/06/11	3;8	Mario
11	- Diseño Detallado	10 días	jue 09/06/11	mié 22/06/11		Mario
12	Diseño de la Capa de Presentaci	10 días	jue 09/06/11	mié 22/06/11	10	Mario
13	Diseño de la Capa de Persistenci	2 días	jue 09/06/11	vie 10/06/11		Mario
14	- Construccion	45 días	jue 23/06/11	mié 24/08/11		Mario
15	Codificacion	35 días	jue 23/06/11	mié 10/08/11	10;12;13	Mario
16	Depuracion	5 días	jue 18/08/11	mié 24/08/11	19	Mario
17	- Pruebas	15 días	jue 11/08/11	mié 31/08/11		Mario
18	Testeo	5 días	jue 25/08/11	mié 31/08/11		Mario
19	Corrección de Errores	5 días	jue 11/08/11	mié 17/08/11	15	Mario
20	- Implantacion y Mantenimiento	2 días	jue 01/09/11	vie 02/09/11		Mario
21	Implantacion y Mantenimiento	2 días	jue 01/09/11	vie 02/09/11	16	Mario

Fig. 7 Tareas de la Planificación

Al finalizar cada etapa se ha realizado una reunión tanto con el cliente como con el director de proyecto, la idea básica de estas es que proporcionen suficiente información para generar confianza en el cliente y el jefe de proyecto, de que el

proyecto avanza notoriamente, que la fecha de culminación se respetará (o en caso contrario, cuál es la tardanza real) y que el coste del desarrollo está dentro de los márgenes previstos.

Para que se vea más detalladamente la faena realizada en cada fase, pasamos a relatar una breve descripción de cada tarea.

Análisis:

- **Obtención necesidades:** para comenzar lo primero que se necesita es toda la información que se pueda obtener del cliente sobre sus necesidades y el sistema de gestión actual para su clínica, y tras ello una exposición al director del proyecto con la exposición de la información recolectada. Y posteriormente documentación sobre la rama de sanidad con la que se ha trabajado.

- **Desarrollo de Casos de uso:** fase donde se detallan los roles de las personas que van a participar en el sistema así como las funciones que van a llevar a cabo cada uno y pasar la información que hemos encontrado anteriormente al diagrama (en el capítulo tres se detalla el Diagrama de Casos de Uso).

- **Desarrollo de Diagrama De Clase:** fase donde introduciremos las clases que hemos identificado anteriormente y pasamos a relacionar las clases entre ellas.

- **Desarrollo de Diagrama Entidad-Relación:** en base al los anteriores diagramas y pensando en la base de datos, se realiza el esquema que guiará la estructura de esta última.

- **Especificación de los Requisitos:** en esta tarea se realiza una reunión con el director de proyecto en la que se hace referencia a la primera reunión que hubo con el cliente para que nos contara cuales eran los requisitos fundamentales del proyecto, y así poder realizar la especificación de requisitos. A partir de ella se plasman en papel las ideas principales del sistema, que problemas se deben resolver y cuáles serán las funciones que se llevaran a cabo.

Diseño:

- Identificación de la Arquitectura del Sistema: una ardua tarea, ya que es importantísimo una arquitectura adecuada y consecuente con el proyecto que deseamos desarrollar. De esta manera llegamos a la conclusión que la mejor arquitectura es la basada en tres capas.

- Diseño BD: es preciso saber cuales son los datos necesarios a almacenar y la relación de unas tablas con otras para poder empezar a estructurar las diferentes pantallas.

- Definir Interfaz: fase en la que se van a realizar los bocetos de cómo se va a enfocar las interfaces. Tras esto se realiza una reunión con el cliente para comprobar su conformidad con la estructura grafica que seguirá la aplicación.

Diseño Detallado:

- Diseño de la capa de Presentación: una faena laboriosa, ya que es importantísimo realizar unas interfaces intuitivas y que aporten la información necesaria al usuario.

- Diseño de la capa de Persistencia: importantísima tarea, ya que debemos de tener un acceso óptimo a la base de datos, tanto como para recuperar datos, como para crearlos o actualizarlos.

Implementación:

- Codificación: En esta fase se implementa toda la funcionalidad expuesta en los casos de uso, hay que seguir fielmente la especificación de requisitos, para que la aplicación desarrollada se adecue a las solicitudes de los clientes. Una vez finalizada esta fase, se realiza una reunión con el cliente para presentarle la primera versión de la aplicación y que este pueda dar su opinión al respecto. Generalmente en la presentación de la primera versión de cualquier aplicación al cliente aparecen nuevas peticiones y sugerencias de modificaciones. En este punto debemos de dejarle muy claro el coste adicional que conlleva modificar un proyecto en fase de desarrollo, y que puede solicitar alguna ampliación o mejora, siempre y cuando nosotros la validemos, pero que una gran modificación podría tener consecuencias catastróficas,

obligándonos a rediseñar todo el análisis inicial y tener que modificar toda la estructura del proyecto.

- Depuración: Realización de las mejoras a petición del cliente como formatos de fechas, modificación en la visualización de informes, etc.

Pruebas:

- Testeo: fase en donde hay que testear todo lo anterior para comprobar que el sistema funciona a la perfección, en esta fase se deberían de detectar todos los posibles Bugs, en el caso de que existiesen. Para un correcto "Testeo" deberíamos de coger los casos de uso y seguirlos paso a paso, marcando todo lo que esta testeado y funcionando correctamente, para asegurarnos de que no nos olvidamos de testear ninguna funcionalidad.

- Corrección de Errores: se realiza prácticamente en paralelo al testeo para solucionar los errores derivados de este, todos los Bugs detectados deben de ser depurados antes de la entrega de la versión final al cliente.

Implantación y Mantenimiento

- Se implanta en el servidor del cliente y se da una formación básica a los usuarios para el correcto manejo de la aplicación, aunque durante un corto periodo de tiempo se les dará soporte.

Tras esta etapa se realiza una entrega de la aplicación al director del proyecto: para que él determine si el proyecto ha llegado a los objetivos marcados de la manera más eficiente.

Como se puede observar, el proyecto se comienza a principios de Mayo de 2011 y se termina a principios de Septiembre de 2011. Para tener el avance del proyecto de forma más visual, se ha realizado un Diagrama de Gantt. Ésta es una popular herramienta gráfica cuyo objetivo es mostrar el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado. A pesar de que, en principio, el Diagrama de Gantt indica las relaciones existentes entre actividades, y la posición de cada tarea a lo largo del tiempo hace que se puedan identificar dichas relaciones e interdependencias. Por ello, en gestión de proyectos,

dicho diagrama muestra el origen y el final de las diferentes unidades mínimas de trabajo y los grupos de tareas o las dependencias entre unidades mínimas de trabajo. Seguidamente en la figura 8 se puede ver con detalle cual es el resultado del Diagrama de Gantt.

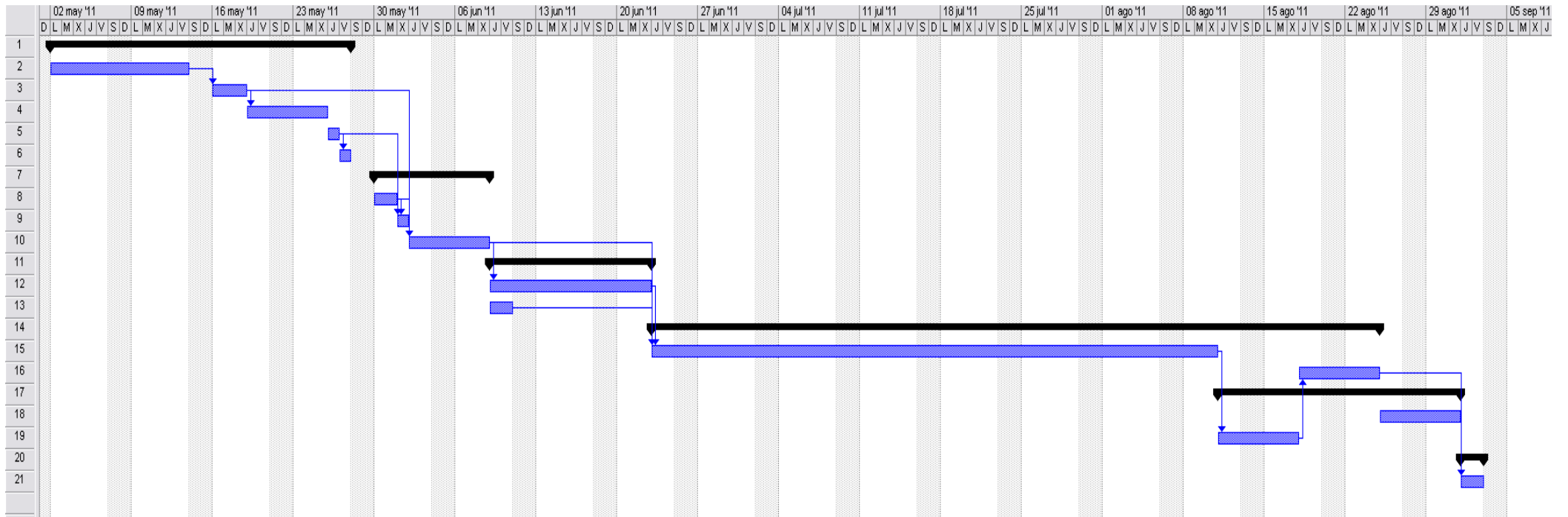


Fig. 8 Diagrama de Gantt del Proyecto

5. Análisis detallado

- Casos de Uso

Un caso de uso representa una interacción típica entre un usuario y un sistema informático. Se utilizan para capturar los requisitos funcionales del sistema. Por ello los casos de uso no son parte del diseño (cómo), sino parte del análisis (qué). De forma que al ser parte del análisis nos ayudan a describir qué es lo que el sistema debe hacer. Por tanto son lo que hace el sistema desde el punto de vista del usuario. Es decir, describen un uso del sistema y cómo este interactúa con el usuario. Pero lo realmente útil de los casos de uso es el documento que describe el caso de uso, ya que en él se explica la forma de interactuar entre el sistema y el usuario.

Haciendo referencia a una descripción más técnica, se puede decir que un caso de uso es un grafo con dos tipos de nodos: Actor, que representa cualquier entidad externa (persona, dispositivos, otros sistemas) que intercambia información con el sistema, por lo que está fuera de él, Caso de uso, es una secuencia de intercambios en diálogo con el sistema que se encuentran relacionadas por su comportamiento. Los arcos entre los actores y los casos de uso se denominan arcos de comunicación. El actor es un agente externo. Un actor representa un cierto papel que un usuario puede jugar. Una máquina también puede ser un actor.

La notación es muy importante, véase la figura 9. En el caso de los casos de uso se representan por una elipse conteniendo el nombre. Los actores se representan con un monigote y el nombre del actor al pie de la figura.

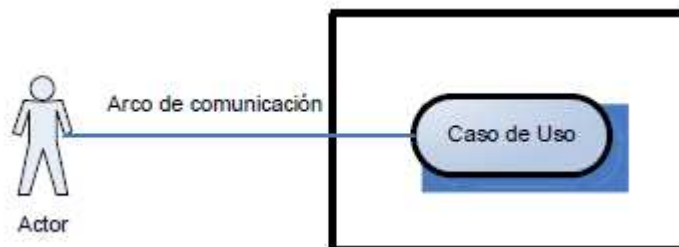
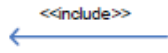


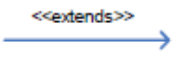
Fig.9 Notación Casos de Usos

De manera que, un caso de uso describe una funcionalidad más una interacción entre un actor y un sistema en forma de secuencia de acciones. La descripción se centra en lo que debe hacerse, no en la manera de hacerlo. Se debe de expresar de forma sencilla y clara.


Otros aspectos importantes referentes a los casos de usos, son las relaciones entre los propios casos de uso. Estas relaciones son: Incluye (<<include>>)



– es un estereotipo de dependencia. Indica que un caso de uso es incluido en otro. Típicamente ocurre cuando unos casos de uso comparten unos pasos

comunes – Extiende (<<extend>>) –  es un estereotipo de dependencia.

Ofrece una forma de extensión más controlada que la relación de generalización. El

caso de uso base declara un conjunto de puntos de extensión –  y

Generalización – indica que in caso de uso es una variante de otro. El caso de uso especializado puede variar cualquier aspecto del caso de uso base.

El primer nivel del diagrama de casos de uso, en este nivel únicamente vamos a encontrar paquetes para después ir explicando cada uno de ellos.

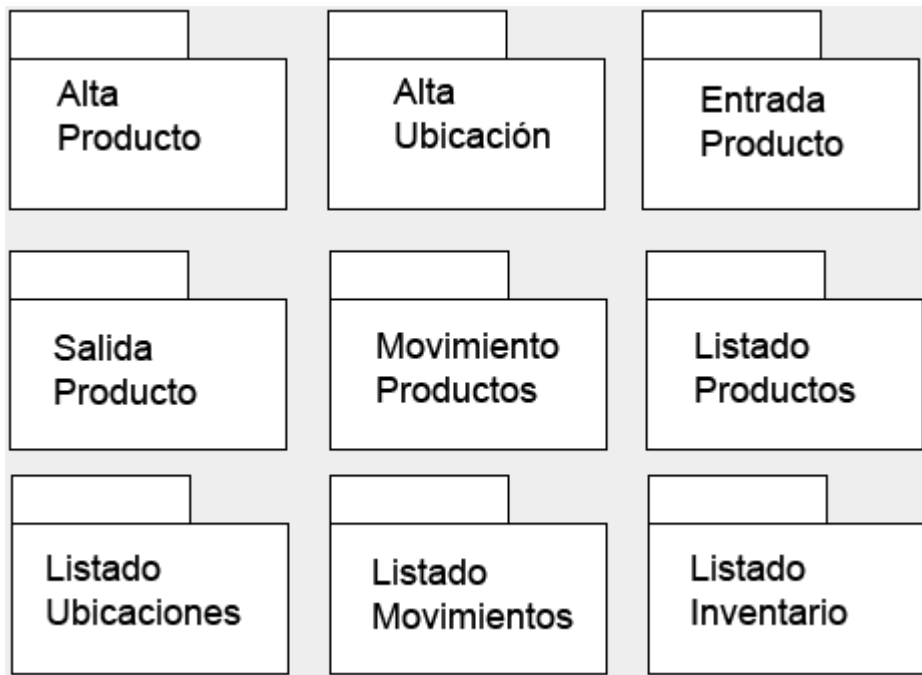


Fig.10 Primer Nivel Diagrama Casos de Uso

Seguidamente vamos a explicar cada uno de los paquetes.

Pantalla de Login

Iniciador: Usuario

Actor secundario: -

Precondición: El servidor Tomcat y el servidor de Base de datos deben estar arrancados y accesibles.

Poscondición: -

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario quiere acceder a la aplicación.	2	El servidor de la aplicación recibe una solicitud y devuelve al usuario la pantalla de Login.
3	El usuario introduce el Login y Password, y lo valida presionando el botón aceptar.	4	El sistema comprueba que dicho usuario existe en la BBDD y si este existe comprueba que el Password es el correcto.

Puntos de Extensión

#1.- Si el usuario valida el formulario de Login sin introducir Login o Password, aparecerá un mensaje emergente indicándole los datos obligatorios para proceder a la validación del formulario.

#2.- Si el Login no existe en nuestra BBDD o el Password no se corresponde, la aplicación nos redireccionara a una pantalla de error con un mensaje para informarnos del error producido.

Menú Principal

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada y el usuario se ha validado correctamente.

Poscondición: El sistema detecta las solicitudes del usuario y reacciona correctamente a estas peticiones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario se ha validado en la pantalla de Login.	2	El servidor de la aplicación recibe una solicitud y devuelve al usuario la pantalla principal.
3	El usuario selecciona cualquier entrada del menú o bien teclea el número que precede a dicha entrada a modo de "atajo de teclado".	4	El sistema captura el número o entrada seleccionada y redirecciona el navegador a la pantalla correspondiente.

Puntos de Extensión

#1.- En la pantalla principal también se encuentra disponible el botón de "**Cerrar Sesión**", que nos sacaría de la aplicación, redireccionandonos de nuevo a la pantalla de Login.

1.- Alta Producto

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada y el usuario se ha validado correctamente.

Poscondición: El sistema registra las modificaciones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario selecciona la opción "1.- Alta Producto" o teclea "1" en el Menú principal.	2	El servidor de la aplicación recibe una solicitud y devuelve al usuario la pantalla de alta de producto.
3	El usuario rellena los campos obligatorios y valida el formulario.	4	El sistema comprueba que los campos obligatorios están rellenos y que los datos tienen el formato correcto, si todo es correcto almacena los datos en la BBDD y devuelve la pantalla de "Éxito".
5	El usuario presiona el botón aceptar.	6	El sistema devuelve al usuario a la pantalla principal de la aplicación.

Puntos de Extensión

#1.- En la pantalla alta aparecerá un botón que nos permita volver a la pantalla principal sin guardar cambios.

#2.- Los campos "Código" y "Código EAN" no podrán ser introducidos manualmente por el usuario, ya que estos campos los gestiona automáticamente el sistema.

#3.- Si los datos de los campos obligatorios están en blanco, o no tienen el formato correcto, el sistema avisara mediante mensajes emergentes al usuario, para que este complete o modifique los datos necesarios.

#4.- Si durante la inserción en la base de datos ocurre algún problema, la aplicación nos mostraría una pantalla de error, para informarnos de que la transacción no se ha realizado.

2.- Alta Ubicación

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada y el usuario se ha validado correctamente.

Poscondición: El sistema registra las modificaciones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario selecciona la opción "2.- Alta Ubicación" o teclea "2" en el Menú principal.	2	El servidor de la aplicación recibe una solicitud y devuelve al usuario la pantalla de alta de ubicación.
3	El usuario rellena los campos obligatorios y valida el formulario.	4	El sistema comprueba que los campos obligatorios están rellenos y que los datos tienen el formato correcto, si todo es correcto almacena los datos en la BBDD y devuelve la pantalla de "Éxito".
5	El usuario presiona el botón aceptar.	6	El sistema devuelve al usuario a la pantalla principal de la aplicación.

Puntos de Extensión

#1.- En la pantalla aparecerá un botón que nos permita volver a la pantalla principal sin guardar cambios.

#2.- Los campos "Código" y "Código EAN" no podrán ser introducidos manualmente por el usuario, ya que estos campos los gestiona automáticamente el sistema.

#3.- Si los datos de los campos obligatorios están en blanco, o no tienen el formato correcto, el sistema avisara mediante mensajes emergentes al usuario, para que este complete o modifique los datos necesarios.

#4.- Si durante la inserción en la base de datos ocurre algún problema, la aplicación nos mostraría una pantalla de error, para informarnos de que la transacción no se ha realizado.

3.- Entrada Producto

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada y el usuario se ha validado correctamente.

Poscondición: El sistema registra las modificaciones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario selecciona la opción "3.- Entrada Producto" o teclea "3" en el Menú principal.	2	El servidor de la aplicación recibe una solicitud y devuelve al usuario la pantalla de Entrada de Producto.
3	El usuario rellena el campo "COD EAN UBICACION" y presiona "Enter".	4	El sistema comprueba que dicho Código de ubicación existe en la BBDD y nos rellena la descripción.
5	El usuario rellena el campo "COD EAN Producto" y presiona "Enter".	6	El sistema comprueba que dicho Código de producto existe en la BBDD y nos rellena la descripción.
7	Por ultimo el usuario rellena la cantidad y presiona el botón aceptar.	8	El sistema comprueba que los campos obligatorios están rellenos y que los datos tienen el formato correcto, si todo es correcto almacena los datos en la BBDD y devuelve la pantalla de "Éxito".
9	El usuario presiona el botón aceptar.	10	El sistema devuelve al usuario a la pantalla principal de la aplicación.

Puntos de Extensión

#1.- En la pantalla aparecerá un botón que nos permita volver a la pantalla principal sin guardar cambios.

#2.- Si los datos de los campos obligatorios están en blanco, o no tienen el formato correcto, el sistema avisara mediante mensajes emergentes al usuario, para que este complete o modifique los datos necesarios.

#3.- Si durante la inserción en la base de datos ocurre algún problema, la aplicación nos mostraría una pantalla de error, para informarnos de que la transacción no se ha realizado.

4.- Salida Producto

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada y el usuario se ha validado correctamente.

Poscondición: El sistema registra las modificaciones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario selecciona la opción "4.- Salida Producto" o teclea "4" en el Menú principal.	2	El servidor de la aplicación recibe una solicitud y devuelve al usuario la pantalla de Salida de Producto.
3	El usuario rellena el campo "COD EAN UBICACION" y presiona "Enter".	4	El sistema comprueba que dicho Código de ubicación existe en la BBDD y nos rellena la descripción.
5	El usuario rellena el campo "COD EAN Producto" y presiona "Enter".	6	El sistema comprueba que dicho Código de producto existe en la BBDD y nos rellena la descripción.
7	Por ultimo el usuario rellena la cantidad y presiona el botón aceptar.	8	El sistema comprueba que los campos obligatorios están rellenos y que los datos tienen el formato correcto, si todo es correcto almacena los datos en la BBDD y devuelve la pantalla de "Éxito".
9	El usuario presiona el botón aceptar.	10	El sistema devuelve al usuario a la pantalla principal de la aplicación.

Puntos de Extensión

#1.- En la pantalla aparecerá un botón que nos permita volver a la pantalla principal sin guardar cambios.

#2.- Si los datos de los campos obligatorios están en blanco, o no tienen el formato correcto, el sistema avisara mediante mensajes emergentes al usuario, para que este complete o modifique los datos necesarios.

#3.- Si durante la inserción en la base de datos ocurre algún problema, la aplicación nos mostraría una pantalla de error, para informarnos de que la transacción no se ha realizado.

5.- Movimiento Productos

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada y el usuario se ha validado correctamente.

Poscondición: El sistema registra las modificaciones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario selecciona la opción "5.- Movimiento Productos" o teclaea "5" en el Menú principal.	2	El servidor de la aplicación recibe una solicitud y devuelve al usuario la pantalla de Movimiento de Producto.
3	El usuario rellena el campo "Ubicación Origen" y presiona "Enter".	4	El sistema comprueba que dicho Código de ubicación existe en la BBDD y nos rellena la descripción.
5	El usuario rellena el campo "COD EAN Producto" y presiona "Enter".	6	El sistema comprueba que dicho Código de producto existe en la BBDD y nos rellena la descripción.
7	El usuario rellena el campo "Ubicación Destino" y presiona "Enter".	8	El sistema comprueba que dicho Código de ubicación existe en la BBDD y nos rellena la descripción.
9	Por ultimo el usuario rellena la cantidad y presiona el botón aceptar.	10	El sistema comprueba que los campos obligatorios están rellenos y que los datos tienen el formato correcto, si todo es correcto almacena los datos en la BBDD y devuelve la pantalla de "Éxito".
11	El usuario presiona el botón	12	El sistema devuelve al

	aceptar.		usuario a la pantalla principal de la aplicación.
--	----------	--	---

Puntos de Extensión

#1.- En la pantalla aparecerá un botón que nos permita volver a la pantalla principal sin guardar cambios.

#2.- Si los datos de los campos obligatorios están en blanco, o no tienen el formato correcto, el sistema avisara mediante mensajes emergentes al usuario, para que este complete o modifique los datos necesarios.

#3.- Si durante la inserción en la base de datos ocurre algún problema, la aplicación nos mostraría una pantalla de error, para informarnos de que la transacción no se ha realizado.

6.- Listado de Productos

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada y el usuario se ha validado correctamente.

Poscondición: El sistema registra las modificaciones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario selecciona la opción "6.- Listado de Productos" o teclea "6" en el Menú principal.	2	El servidor de la aplicación recibe una solicitud y devuelve al usuario la pantalla de Listado de Productos.

Puntos de Extensión

#1.- En la pantalla aparecerá un botón que nos permita volver a la pantalla principal sin guardar cambios.

#2.- Los listados se mostraran paginados, organizando los productos en grupos de 10 por página y permitiendo al usuario una navegación mediante botones siguiente, anterior, primera y ultima.

6.1. Búsqueda en Listado de Productos

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Listado de Productos.

Poscondición: El sistema registra las modificaciones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario inserta un código, descripción o parte de una descripción en el campo de búsqueda de la pantalla de Listado de Productos y presiona el botón "Buscar".	2	El sistema recupera el parámetro de búsqueda y devuelve un listado de productos filtrado por dicho parámetro.

Puntos de Extensión

#1.- Los listados aparecerán paginados, organizando los productos en grupos de 10 por pagina y permitiendo al usuario una navegación mediante botones siguiente, anterior, primera y ultima.

#2.- El parámetro de búsqueda deberá conservarse durante la navegación por el listado.

6.2. Ordenación en Listado de Productos

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Listado de Productos.

Poscondición: El sistema registra las modificaciones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario selecciona un campo por el que organizar y el orden (ASC - ascendente o DESC - descendente) en la pantalla de Listado de Productos y presiona el botón "Ordenar".	2	El sistema recupera los parámetros de ordenación y devuelve un listado de productos filtrado por dichos parámetros.

Puntos de Extensión

#1.- Los listados aparecerán paginados, organizando los productos en grupos de 10 por pagina y permitiendo al usuario una navegación mediante botones siguiente, anterior, primera y ultima.

#2.- Los parámetros de ordenación deberán conservarse durante la navegación por el listado.

6.3. Abrir Ficha de Producto

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Listado de Productos.

Poscondición: -

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario selecciona una fila del listado de productos y pulsa el botón "Abrir", o simplemente hace "doble click" sobre la fila.	2	El sistema recupera los datos de la fila seleccionada y abre la ficha del producto que se corresponde con dicha fila.

Puntos de Extensión

#1.- La ficha del producto abierta dispondrá de un botón "volver", que nos permitirá volver al listado de productos.

#2.- Desde la ficha de producto podremos realizar el mantenimiento de los productos (modificar o eliminar el producto).

#3.- En la ficha del producto debe de aparecer un botón "Imprimir Etiqueta", que sacara por la impresora de etiquetas o en PDF la etiqueta de código de barras correspondiente a dicho producto.

6.4. Imprimir Listado de Producto

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Listado de Productos.

Poscondición: -

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario pulsa el botón "Imprimir PDF" en la pantalla del listado de productos.	2	El sistema generara un informe en PDF con el listado de productos.
3	El usuario selecciona imprimir o guardar el informe.		

Puntos de Extensión

6.5. Operaciones en Ficha de Producto

6.5.1. Modificar Producto

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Ficha de Producto.

Poscondición: El sistema registra las modificaciones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario se encuentra en la Ficha del Producto, modifica el contenido de los campos del formulario y pulsa "Guardar".	2	El sistema comprueba que los campos obligatorios están rellenos y que los datos tienen el formato correcto, si todo es correcto almacena los datos en la BBDD y devuelve la pantalla de "Éxito".
3	El usuario presiona el botón aceptar.	4	El sistema devuelve al usuario a la pantalla principal de la aplicación.

Puntos de Extensión

#1.- En la pantalla aparecerá un botón que nos permita volver a la pantalla principal sin guardar cambios.

#2.- Los campos "Código" y "Código EAN" no se podrán modificar, ya que estos campos los gestiona automáticamente el sistema.

#3.- Si los datos de los campos obligatorios están en blanco, o no tienen el formato correcto, el sistema avisara mediante mensajes emergentes al usuario, para que este complete o modifique los datos necesarios.

#4.- Si durante la inserción en la base de datos ocurre algún problema, la aplicación nos mostraría una pantalla de error, para informarnos de que la transacción no se ha realizado.

6.5.2. Eliminar Producto

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Ficha de Producto.

Poscondición: El sistema registra las modificaciones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario se encuentra en la Ficha del Producto y pulsa "Eliminar".	2	El sistema comprueba que el producto a eliminar no tiene tuplas relacionadas en otras tablas, y si no existen se procede a la eliminación de dicho producto de la BBDD y se devuelve la pantalla de "Éxito".
3	El usuario presiona el botón aceptar.	4	El sistema devuelve al usuario a la pantalla principal de la aplicación.

Puntos de Extensión

#1.- En la pantalla aparecerá un botón que nos permita volver a la pantalla principal sin guardar cambios.

#2.- Si durante la inserción en la base de datos ocurre algún problema, la aplicación nos mostraría una pantalla de error, para informarnos de que la transacción no se ha realizado.

6.5.3. Imprimir Etiqueta Producto

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Ficha de Producto.

Poscondición: -

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario pulsa el botón "Imprimir Etiqueta" en la pantalla de Ficha de producto.	2	El sistema generara un informe en PDF con la descripción y el código de barras del producto.
3	El usuario selecciona imprimir o guardar el informe.		

Puntos de Extensión

7.- Listado de Ubicaciones

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada y el usuario se ha validado correctamente.

Poscondición: El sistema registra las modificaciones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario selecciona la opción "7.- Listado de Ubicaciones" o teclea "7" en el Menú principal.	2	El servidor de la aplicación recibe una solicitud y devuelve al usuario la pantalla de Listado de Ubicaciones.

Puntos de Extensión

#1.- En la pantalla aparecerá un botón que nos permita volver a la pantalla principal sin guardar cambios.

#2.- Los listados se mostraran paginados, organizando los productos en grupos de 10 por página y permitiendo al usuario una navegación mediante botones siguiente, anterior, primera y ultima.

7.1. Búsqueda en Listado de Ubicaciones

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Listado de Ubicaciones.

Poscondición: El sistema registra las modificaciones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario inserta un código, descripción o parte de una descripción en el campo de búsqueda de la pantalla de Listado de Ubicaciones y presiona el botón "Buscar".	2	El sistema recupera el parámetro de búsqueda y devuelve un listado de Ubicaciones filtrado por dicho parámetro.

Puntos de Extensión

#1.- Los listados aparecerán paginados, organizando los productos en grupos de 10 por pagina y permitiendo al usuario una navegación mediante botones siguiente, anterior, primera y ultima.

#2.- El parámetro de búsqueda deberá conservarse durante la navegación por el listado.

7.2. Ordenación en Listado de Ubicaciones

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Listado de Ubicaciones.

Poscondición: El sistema registra las modificaciones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario selecciona un campo por el que organizar y el orden (ASC - ascendente o DESC - descendente) en la pantalla de Listado de Ubicaciones y presiona el botón "Ordenar".	2	El sistema recupera los parámetros de ordenación y devuelve un listado de Ubicaciones filtrado por dichos parámetros.

Puntos de Extensión

#1.- Los listados aparecerán paginados, organizando los productos en grupos de 10 por pagina y permitiendo al usuario una navegación mediante botones siguiente, anterior, primera y ultima.

#2.- Los parámetros de ordenación deberán conservarse durante la navegación por el listado.

7.3. Abrir Ficha de Ubicación

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Listado de Ubicaciones.

Poscondición: -

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario selecciona una fila del listado de Ubicaciones y pulsa el botón "Abrir", o simplemente hace "doble click" sobre la fila.	2	El sistema recupera los datos de la fila seleccionada y abre la ficha de la Ubicación que se corresponde con dicha fila.

Puntos de Extensión

#1.- La ficha de la ubicación abierta dispondrá de un botón "volver", que nos permitirá volver al listado de productos.

#2.- Desde la ficha de ubicación podremos realizar el mantenimiento de las ubicaciones (modificar o eliminar el producto).

#3.- En la ficha de la ubicación debe de aparecer un botón "Imprimir Etiqueta", que sacara por la impresora de etiquetas o en PDF la etiqueta de código de barras correspondiente a dicha ubicación.

7.4. Imprimir Listado de Ubicación

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Listado de Ubicaciones.

Poscondición: -

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario pulsa el botón "Imprimir PDF" en la pantalla del listado de Ubicaciones.	2	El sistema generara un informe en PDF con el listado de Ubicaciones.
3	El usuario selecciona imprimir o guardar el informe.		

Puntos de Extensión

7.5. Operaciones en Ficha de Ubicación

7.5.1. Modificar Ubicación

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Ficha de Ubicación.

Poscondición: El sistema registra las modificaciones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario se encuentra en la Ficha de la ubicación, modifica el contenido de los campos del formulario y pulsa "Guardar".	2	El sistema comprueba que los campos obligatorios están rellenos y que los datos tienen el formato correcto, si todo es correcto almacena los datos en la BBDD y devuelve la pantalla de "Éxito".
3	El usuario presiona el botón aceptar.	4	El sistema devuelve al usuario a la pantalla principal de la aplicación.

Puntos de Extensión

#1.- En la pantalla aparecerá un botón que nos permita volver a la pantalla principal sin guardar cambios.

#2.- Los campos "Código" y "Código EAN" no se podrán modificar, ya que estos campos los gestiona automáticamente el sistema.

#3.- Si los datos de los campos obligatorios están en blanco, o no tienen el formato correcto, el sistema avisara mediante mensajes emergentes al usuario, para que este complete o modifique los datos necesarios.

#4.- Si durante la inserción en la base de datos ocurre algún problema, la aplicación nos mostraría una pantalla de error, para informarnos de que la transacción no se ha realizado.

7.5.2. Eliminar Ubicación

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Ficha de Ubicación.

Poscondición: El sistema registra las modificaciones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario se encuentra en la Ficha de la ubicación y pulsa "Eliminar".	2	El sistema comprueba que la ubicación a eliminar no tiene tuplas relacionadas en otras tablas, y si no existen se procede a la eliminación de dicha ubicación de la BBDD y se devuelve la pantalla de "Éxito".
3	El usuario presiona el botón aceptar.	4	El sistema devuelve al usuario a la pantalla principal de la aplicación.

Puntos de Extensión

#1.- En la pantalla aparecerá un botón que nos permita volver a la pantalla principal sin guardar cambios.

#2.- Si durante la inserción en la base de datos ocurre algún problema, la aplicación nos mostraría una pantalla de error, para informarnos de que la transacción no se ha realizado.

7.5.3. Imprimir Etiqueta Ubicación

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Ficha de Ubicación.

Poscondición: -

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario pulsa el botón "Imprimir Etiqueta" en la pantalla de Ficha de Ubicación.	2	El sistema generara un informe en PDF con la descripción y el código de barras de la Ubicación.
3	El usuario selecciona imprimir o guardar el informe.		

Puntos de Extensión

8.- Listado de Movimientos

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada y el usuario se ha validado correctamente.

Poscondición: El sistema registra las modificaciones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario selecciona la opción "8.- Listado de Movimientos" o teclea "8" en el Menú principal.	2	El servidor de la aplicación recibe una solicitud y devuelve al usuario la pantalla de Listado de Movimientos.

Puntos de Extensión

#1.- En la pantalla aparecerá un botón que nos permita volver a la pantalla principal sin guardar cambios.

#2.- Los listados se mostraran paginados, organizando los productos en grupos de 10 por página y permitiendo al usuario una navegación mediante botones siguiente, anterior, primera y ultima.

8.1. Búsqueda en Listado de Movimientos

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Listado de Movimientos.

Poscondición: El sistema registra las modificaciones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario inserta un código, descripción o parte de una descripción en el campo de búsqueda de la pantalla de Listado de Movimientos y presiona el botón "Buscar".	2	El sistema recupera el parámetro de búsqueda y devuelve un listado de movimientos filtrado por dicho parámetro.

Puntos de Extensión

#1.- Los listados aparecerán paginados, organizando los productos en grupos de 10 por pagina y permitiendo al usuario una navegación mediante botones siguiente, anterior, primera y ultima.

#2.- El parámetro de búsqueda deberá conservarse durante la navegación por el listado.

8.2. Abrir Ficha de Producto

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Listado de Movimientos.

Poscondición: -

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario selecciona una fila del listado de movimientos y pulsa el botón "Abrir", o simplemente hace "doble click" sobre la fila.	2	El sistema recupera los datos de la fila seleccionada y abre la ficha del producto que se corresponde con dicha fila.

Puntos de Extensión

#1.- La ficha del producto abierta dispondrá de un botón "volver", que nos permitirá volver al listado de movimientos.

#2.- Desde la ficha de producto podremos realizar el mantenimiento de los productos (modificar o eliminar el producto).

#3.- En la ficha del producto debe de aparecer un botón "Imprimir Etiqueta", que sacara por la impresora de etiquetas o en PDF la etiqueta de código de barras correspondiente a dicho producto.

8.3. Imprimir Listado de Movimientos

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Listado de Movimientos.

Poscondición: -

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario pulsa el botón "Imprimir PDF" en la pantalla del listado de movimientos.	2	El sistema generara un informe en PDF con el listado de movimientos.
3	El usuario selecciona imprimir o guardar el informe.		

Puntos de Extensión

9.- Listado de Inventario

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada y el usuario se ha validado correctamente.

Poscondición: El sistema registra las modificaciones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario selecciona la opción "9.- Listado de Inventario" o teclea "9" en el Menú principal.	2	El servidor de la aplicación recibe una solicitud y devuelve al usuario la pantalla de Listado de Inventario.

Puntos de Extensión

#1.- En la pantalla aparecerá un botón que nos permita volver a la pantalla principal.

#2.- Los listados se mostraran paginados, organizando los productos en grupos de 10 por página y permitiendo al usuario una navegación mediante botones siguiente, anterior, primera y ultima.

9.1. Búsqueda en Listado de Inventario

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Listado de Inventario.

Poscondición: El sistema registra las modificaciones.

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario inserta un código, descripción o parte de una descripción en el campo de búsqueda de la pantalla de Listado de Inventario y presiona el botón "Buscar".	2	El sistema recupera el parámetro de búsqueda y devuelve un listado de inventario filtrado por dicho parámetro.

Puntos de Extensión

#1.- Los listados aparecerán paginados, organizando los productos en grupos de 10 por pagina y permitiendo al usuario una navegación mediante botones siguiente, anterior, primera y ultima.

#2.- El parámetro de búsqueda deberá conservarse durante la navegación por el listado.

9.2. Abrir Ficha de Producto

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Listado de Inventario.

Poscondición: -

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario selecciona una fila del listado de Inventario y pulsa el botón "Abrir", o simplemente hace "doble click" sobre la fila.	2	El sistema recupera los datos de la fila seleccionada y abre la ficha del producto que se corresponde con dicha fila.

Puntos de Extensión

#1.- La ficha del producto abierta dispondrá de un botón "volver", que nos permitirá volver al listado de inventario.

#2.- Desde la ficha de producto podremos realizar el mantenimiento de los productos (modificar o eliminar el producto).

#3.- En la ficha del producto debe de aparecer un botón "Imprimir Etiqueta", que sacara por la impresora de etiquetas o en PDF la etiqueta de código de barras correspondiente a dicho producto.

9.3. Imprimir Listado de Inventario

Iniciador: Usuario

Actor secundario: -

Precondición: La aplicación esta arrancada, el usuario se ha validado correctamente y nos encontramos en la pantalla de Listado de Inventario.

Poscondición: -

Flujo de Eventos

INTENCION DE USUARIO		OBLIGACION DEL SISTEMA	
1	El caso de uso se inicia cuando el usuario pulsa el botón "Imprimir PDF" en la pantalla del listado de inventario.	2	El sistema generara un informe en PDF con el listado de inventario.
3	El usuario selecciona imprimir o guardar el informe.		

Puntos de Extensión

- Diagrama de clases

El Diagrama de Clases representa la estructura y comportamiento de cada uno de los objetos del sistema y sus relaciones con los demás objetos. El objetivo principal es representar los aspectos estáticos del sistema. Al mismo tiempo nos representa la dinámica de los objetos.

Los elementos de un diagrama de clases son las clases, las relaciones y las interfaces. Las clases son las que describen un conjunto de objetos con propiedades y comportamientos comunes.

Dentro de la estructura de una clase se han de definir los siguientes elementos: los atributos, son los datos asociados a los elementos y que toman valor al instanciar objetos de una clase, y las operaciones (métodos), que son funciones o procesos propios de los objetos de una clase.

La notación es semejante a la que aparece en la figura 11.

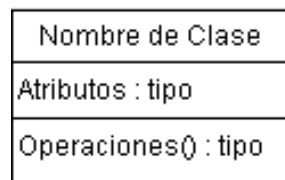


Fig.11 Notación Clase

Las relaciones son los enlaces entre los distintos elementos de los diagramas. A varios tipos de relaciones, por lo que nos vamos a centrar en explicar aquellas que nosotros hemos utilizado en nuestro diagrama. Estas son:

- Asociación: relación genérica entre dos clases que representa un conjunto de enlaces entre objetos. Denota generalmente una dependencia semántica.
- Herencia: establecimiento de una jerarquía padre/hijo entre las clases implicadas. Permite a una clase (hijo o subclase) heredar los atributos de otra (padre o superclase) siendo posible añadir atributos y operaciones específicas.

- Composición: representación de una relación de composición entre un elemento y sus componentes (rombo relleno).
- Agregación: parte de una asociación en la cual los objetos representan componentes o partes de otro objeto (Composición débil).
- Restricciones: permiten establecer limitaciones o 'restricciones' entre los elementos del diagrama de clases (or exclusivo).

Seguidamente se puede comprobar, en la figura 12 la utilización de las relaciones comentadas anteriormente en nuestro diagrama de clases. En él se pueden contabilizar las 7 clases que hemos llevado a cabo, o lo que es lo mismo, los objetos que modelan nuestro trabajo.

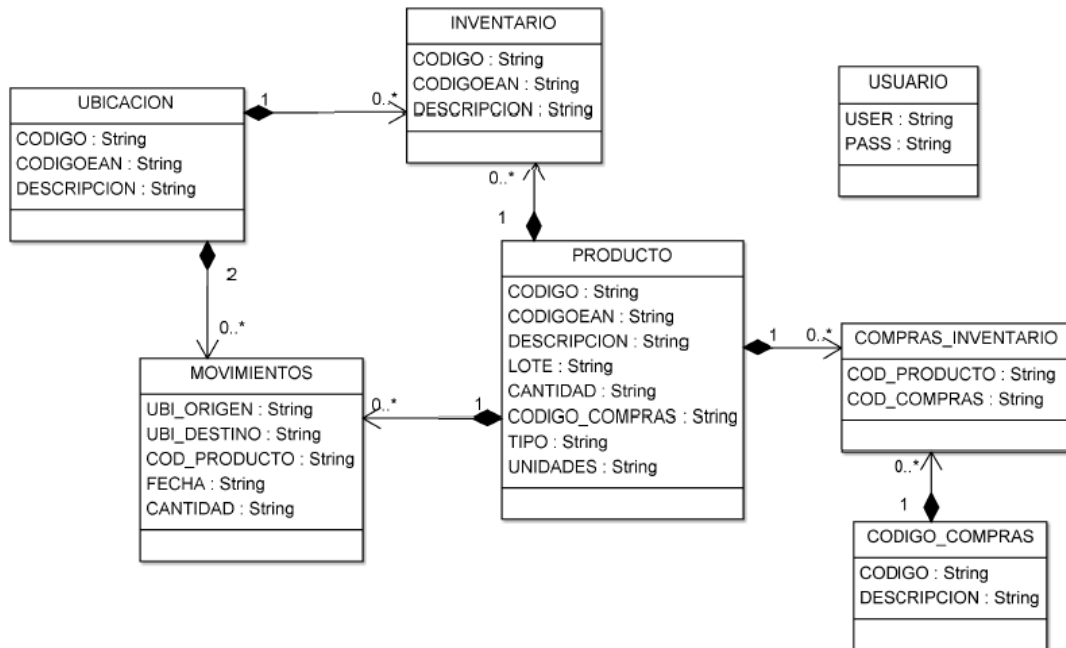


Fig.12 Diagrama de clases

- **Modelo Relacional**

Las bases de datos relacionales son el tipo de bases de datos actualmente más difundido. Los motivos de este éxito son fundamentalmente dos: ofrecer un sistema simple y eficaz para representar y manipular los datos, y que se basan en un modelo, el relacional, con sólidas bases teóricas.

El modelo relacional para la gestión de una base de datos, es un modelo de datos basado en la lógica de predicado y en la teoría de conjuntos. La estructura fundamental del modelo relacional es precisamente esa, la “relación”. Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados “tuplas”. Pese a que ésta es la teoría de las bases de datos relacionales, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar, esto es, pensando en cada relación como si fuese una tabla que está compuestas por registros (cada fila de la tabla sería un registro o tupla), y columnas (también llamadas campos).

En este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en el que estos se almacenen no tiene mayor relevancia.

Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar por un usuario no experto. La información puede ser recuperada o almacenada por medio de “consultas” que ofrecen una amplia flexibilidad y poder para administrar la información. Para manipular la información utilizamos un lenguaje relacional, actualmente se cuenta con dos lenguajes formales el álgebra relacional y el cálculo relacional. El álgebra relacional permite describir la forma de realizar una consulta, en cambio, el cálculo relacional sólo indica lo que se desea devolver.

El lenguaje más común (el que nosotros hemos utilizado) para construir las consultas a bases de datos relacionales es SQL (Structured Query Language), un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

A continuación, en la figura 13, se puede ver la nomenclatura utilizada para representar nuestro modelo relacional.

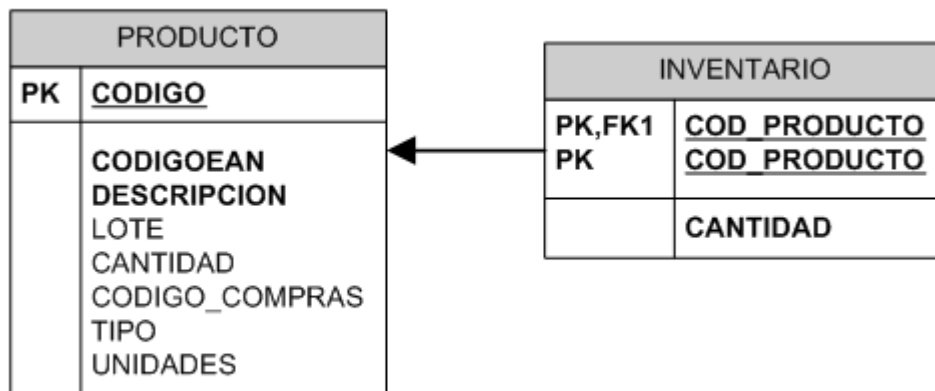


Fig.13 Notación Modelo Relacional

Como vemos en la imagen anterior, cada 'caja' hace referencia a una tabla. El nombre que aparece en la parte superior de la tabla es el nombre de la tabla en cuestión. La primera fila que aparece, donde pone PK (PRIMARY KEY), expresa cual es la clave primaria dentro de la tabla. Seguidamente, en la siguiente fila podemos ver una serie de nombres, los cuales hacen referencia a los atributos de la tabla. Pero en aquellos atributos a los que aparezcan al lado FK (FOREIGN KEY) quiere decir cuál o cuáles son las claves ajenas de la tabla. Para finalizar, la flecha ← significa con que tabla se relaciona dicha clave ajena, en este ejemplo sería la Tabla Paciente se relaciona con Usuario.

Normalmente se pueden identificar las tablas del modelo relacional a través del diagrama de clases que hemos obtenido. En este caso es casi similar, ya que tenemos las mismas 7 tablas que tenemos en el diagrama de clases.

El modelo relacional que hemos obtenido en nuestro proyecto es el que se puede observar en la figura 14.

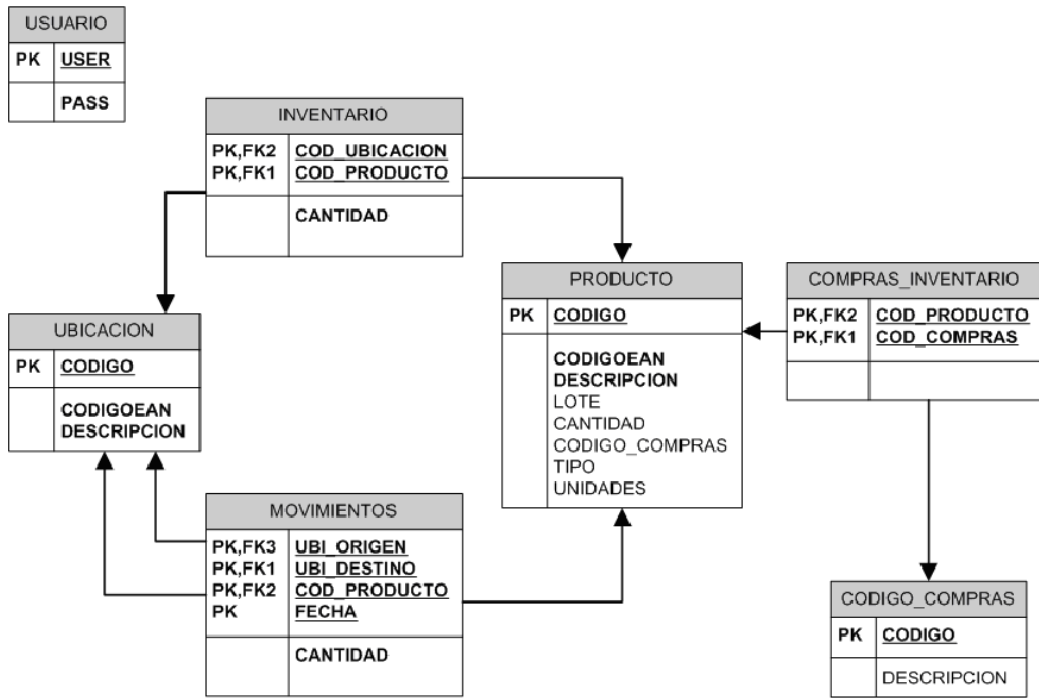


Fig.14 Modelo Relacional

6. Implementación

Entorno

Eclipse HELIOS es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent o Azureus.

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project, cubriendo casi todas las áreas de Model Driven Engineering.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

Eclipse fue liberado originalmente bajo la Common Public License, pero después fue re-licenciado bajo la Eclipse Public License. La Free Software Foundation ha dicho que ambas licencias son licencias de software libre, pero son incompatibles con Licencia pública general de GNU (GNU GPL).

Eclipse es un potente entorno de desarrollo integrado Java (IDE, por sus siglas en inglés), es multiplataforma, (disponible para Windows, Linux, Solaris, y Mac OS X) que agiliza el desarrollo de las aplicaciones en lenguaje Java.

Se puede utilizar para el desarrollo completo de la aplicación (accesos a la base de datos (DAO), lógica de la aplicación y la interfaz gráfica de usuario) y para realizar las pruebas durante y después del proceso de desarrollo, ya que integra el diferentes servidores como Apache, IBM WebSphere, JBoss, etc. y te permite arrancar

un servidor embebido (en nuestro caso un Tomcat), para la realización de dichas pruebas.

He optado por Eclipse Helios como IDE de desarrollo, ya que es una herramienta OpenSource, muy robusta que integra todo lo necesario para el desarrollo web (editores de texto para XML, JavaServer Pages y Struts, refactorización distribuida, auditoria de código, testeo de unidades, depuración avanzada, sistemas de control de versiones, etc.).

Algunas de las características más importantes, y que hemos usado para el proyecto para facilitar nuestro trabajo han sido:

- ♣ Editor de Código: al igual que cualquier otro IDE, incluye un editor de código que soporta resaltado de sintaxis y el uso de código de IntelliSense, no sólo en variables, funciones y métodos, sino también como lenguaje que construye bucles y preguntas. Cosa que resulta muy útil, ya que te ayuda a programar de una forma más rápida y eficaz, sin tener que estar buscando por las clases los nombre de las propiedades o de los métodos que quieras usar. También ha sido útil a la hora de crear las propiedades, ya que únicamente colocando 'prop' y tabular, te crea la estructura para realizar una propiedad. De igual manera pasa con la creación de bucles, como ya hemos comentado anteriormente. Además apoya el establecimiento de marcadores en el código para facilitar una navegación más rápida.

- ♣ Debugger: es una herramienta muy util para poder encontrar los problemas en un determinado punto del código de nuestra aplicación. Colocando un 'punto de interrupción' en la zona de código donde se produce el error, o creemos que se puede producir, se va ejecutando paso por paso para ver qué valores van tomando las variables y demás propiedades, y así comprobar si es lo esperado, y si no lo es, detectar con más certeza donde nos hemos podido equivocar, para después poder reparar el error. Estos 'breakpoints' pueden ser condicionales, es decir, que coloquemos una condición para cuando esta se cumpla, entonces se produzca dicha interrupción. Además, se puede ir recorriendo en código de varias formas, haciendo una ejecución 'paso a paso por instrucciones' o 'paso a paso por procedimientos'. La primera hace referencia a que pasará por todas las líneas de código por las cuales deba pasar, y la segunda, hace el recorrido más rápido sin introducirse en los métodos por los que vaya pasando.

♣ Diseño: en nuestra aplicación hemos utilizado JSP (HTML con Java y JavaScript). Además de usar “estilos” y CSS para adecuar la interfaz gráfica de usuario.

○ Independencia de plataforma. Al tratarse de una aplicación Web, puede ejecutarse en cualquier navegador (Internet Explorer, Firefox, Opera, Chrome, ...), independientemente del sistema operativo de la máquina.

Otro aspecto importante referido al entorno, es el lenguaje que hemos utilizado para llevar a cabo nuestro proyecto, **Java**. Se trata de un lenguaje orientado a objetos elegante y con seguridad de tipos que permiten a los desarrolladores crear una amplia gama de aplicaciones sólidas y seguras.

La segunda característica, la independencia de la plataforma, significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Este es el significado de ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo. Java dispone de una herramienta llamada el recolector automático de basura (o automatic garbage collector). El programador determina cuándo se crean los objetos y el entorno en tiempo de ejecución de Java (Java Runtime) es el responsable de gestionar el ciclo de vida de los objetos. El programa, u otros objetos pueden tener localizado un objeto mediante una referencia a éste (que, desde un punto de vista de bajo nivel es una dirección de memoria). Cuando no quedan referencias a un objeto, el recolector de basura de Java borra el objeto, liberando así la memoria que ocupaba previniendo posibles fugas (ejemplo: un objeto creado y únicamente usado dentro de un método sólo tiene entidad dentro de éste; al salir del método el objeto es eliminado).

El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir muchos errores, como la manipulación directa de punteros o memoria. La sintaxis de Java se deriva en gran medida de C++. Pero a diferencia de éste, que combina la sintaxis para programación genérica, estructurada y orientada a objetos, Java fue construido desde el principio para ser completamente orientado a objetos. Todo en Java es un objeto (salvo algunas excepciones), y todo en Java reside en alguna clase (recordemos que una clase es un molde a partir del cual pueden crearse varios objetos).

También admite métodos y tipos genéricos, que proporcionan mayor rendimiento y seguridad de tipos, e iteradores, que permiten a los implementadores de clases de colección definir comportamientos de iteración personalizados que el código de cliente puede utilizar fácilmente.

Como lenguaje orientado a objetos que es, admite los conceptos de encapsulación, herencia y polimorfismo. Todas las variables y métodos, incluido el método 'Main' que es el punto de entrada de la aplicación, se encapsulan dentro de definiciones de clase. Una clase puede heredas directamente de una clase primaria, pero puede implementar cualquier número de interfaces.

iReport es un diseñador visual de código abierto para crear informes complejos utilizando la biblioteca JasperReports, el más avanzado motor de fuente abierta de información disponibles, aprovechando toda su potencia. Los datos para imprimir, se pueden adquirir en cualquier tipo de fuente de datos incluidas las bases de datos, OLAP XML / A conexiones, archivos XML, CSV, las colecciones de JavaBeans o incluso la costumbre fuentes proporcionadas por el usuario. Proporciona un soporte incorporado para varios lenguajes como consulta SQL, HQL, XPath, EJBQL y MDX y permite conectar a la costumbre en los idiomas necesarios como PL / SQL. iReport ofrece un entorno completo para crear y probar las fuentes de datos, diseño de los informes, previo y la exportación en todos los formatos incluyendo PDF, HTML, XML, Word (RTF), Excel, OpenOffice (ODF), texto y CSV. Para los desarrolladores, iReport Jasper ayuda a generar los archivos que se pueden desplegar en cualquier aplicación Java (web y aplicación de escritorio) para generar informes rica dinámica en cualquier formato. La integración con JasperReports requiere sólo unas líneas de código de proporcionar al instante de corte de edad de presentación de informes a cualquier aplicación Java. iReport puede ser usado para desplegar y editar informes en JasperServer, un alto rendimiento de código abierto servidor de informes que proporciona servicios adicionales como la seguridad, repositorio de recursos, informe de la programación, web para la integración con otras aplicaciones y una interfaz web para gestionar, ejecutar y generar el informes.

Los pasos para crear y desplegar un informe en una aplicación Java son los siguientes. El primer paso es crear la plantilla de informe, un archivo XML con extensión jrxml (XML JasperReports). Esta plantilla se puede crear a mano o utilizando una herramienta como iReport. El jrxml contiene toda la información de diseño y otras cosas como la posibilidad de consulta que deben ser utilizados para adquirir los datos

y la definición de parámetros de entrada. Jrxml El archivo está compilado en un formato binario que producen un archivo Jasper. Jasper el archivo puede ser cargado en una aplicación Java utilizando el JasperRepors API, y se utiliza para generar el informe con una fuente de datos (es decir, una conexión JDBC) y si es necesario los parámetros de entrada (que puede ser utilizada por ejemplo para filtrar los registros). El resultado puede ser exportado en un formato final (como PDF) o ver con el JasperReports construido en espectador. Todas estas operaciones se pueden realizar directamente en iReport como parte del proceso de elaboración de informe.

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre.

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad.

MySQL funciona sobre múltiples plataformas, incluyendo: AIX, BSD, FreeBSD, HP-UX, GNU/Linux, NetBSD, Novell Netware, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, eBD, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista, y otras versiones de Windows.

Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente. Un amplio subconjunto de ANSI SQL 99, y varias extensiones.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
 - Transacciones y claves foráneas.
 - Conectividad segura.
 - Replicación.
 - Búsqueda e indexación de campos de texto.

- Procedimientos almacenados
- Disparadores (triggers)
- Cursores
- Vistas actualizables
- Soporte a VARCHAR
- INFORMATION_SCHEMA
- Modo Strict
- Soporte X/Open XA de transacciones distribuidas; transacción en dos fases como parte de esto, utilizando el motor InnoDB de Oracle
- Transacciones con los motores de almacenamiento InnoDB, BDB Y Cluster; puntos de recuperación (savepoints) con InnoDB
- Soporte para SSL
- Query caching
- Sub-SELECTs (o SELECTs anidados)
- Réplica con un maestro por esclavo, varios esclavos por maestro, sin soporte automático para múltiples maestros por esclavo.
- Embedded database library
- Soporte completo para Unicode
- Conforme a las reglas ACID usando los motores InnoDB, BDB y Cluster
- Shared-nothing clustering through MySQL Cluster Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi (vía dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby, Gambas, REALbasic (Mac), (x)Harbour (Eagle1), FreeBASIC, y Tcl; cada uno de estos utiliza una API específica. También existe un interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL. También se puede acceder desde el sistema SAP, lenguaje ABAP.

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos. Para agregar, acceder a y procesar datos guardados en un computador, se necesita un administrador como MySQL Server.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están

conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir que puede hacer y que no puede hacer con el software en diferentes situaciones. Si usted no se ajusta al GPL o requiere introducir código MySQL en aplicaciones comerciales, usted puede comprar una versión comercial licenciada.

Las siguientes características son implementadas únicamente por MySQL:

- Múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example en 5.x), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

Hay tres tipos de compilación del servidor MySQL:

- Estándar: Los binarios estándar de MySQL son los recomendados para la mayoría de los usuarios, e incluyen el motor de almacenamiento InnoDB.
- Max (No se trata de MaxDB, que es una cooperación con SAP): Los binarios incluyen características adicionales que no han sido lo bastante probadas o que normalmente no son necesarias.
- MySQL-Debug: Son binarios que han sido compilados con información de depuración extra. No debe ser usada en sistemas en producción porque el código de depuración puede reducir el rendimiento.

En el proyecto se han utilizado MySQL Drivers para ODBC llamado MyODBC, MySQL Tools y MySQL Server 5.1

Apache Tomcat (también llamado **Jakarta Tomcat** o simplemente **Tomcat**) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la

Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems.

Tomcat es mantenido y desarrollado por miembros de la Apache Software Foundation y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la *Apache Software Licence*. Las primeras distribuciones de Tomcat fueron las versiones 3.0.x. Las versiones más recientes son las 7.x, que implementan las especificaciones de Servlet 3.0 y de JSP 2.2. A partir de la versión 4.0, Jakarta Tomcat utiliza el contenedor de servlets Catalina.

Tomcat es un servidor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

Estructura de directorios:

La jerarquía de directorios de instalación de Tomcat incluye:

bin - arranque, cierre, y otros scripts y ejecutables

common - clases comunes que pueden utilizar Catalina y las aplicaciones web
conf-ficheros XML y los correspondientes DTD para la configuración de Tomcat

logs - logs de Catalina y de las aplicaciones

server - clases utilizadas solamente por Catalina

shared - clases compartidas por todas las aplicaciones web

webapps - directorio que contiene las aplicaciones web

work - almacenamiento temporal de ficheros y directorios

STRUTS (Tecnología de Implementación J2EE + JSP) es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma Java EE (Java Enterprise Edition). Struts se desarrollaba como parte del proyecto Jakarta de la Apache Software Foundation, pero actualmente es un proyecto independiente conocido como Apache Struts.

Struts permite reducir el tiempo de desarrollo. Su carácter de "software libre" y su compatibilidad con todas las plataformas en las que Java Enterprise esté disponible lo convierten en una herramienta altamente disponible.

Con la versión 2 del framework se introdujeron algunas mejoras sobre la primera versión, de cara a simplificar las tareas más comunes en el desarrollo de aplicaciones web, así como mejorar su integración con AJAX, etc.

Estructura de los Archivos e Instalación

Las aplicaciones Web poseen una estructura de archivos definida para poder ejecutarse en un contenedor de Servlets, como el Tomcat de Apache, esta estructura por lo regular no se modifica y solamente se le agregan algunas librerías para poder correr una aplicación utilizando Struts.

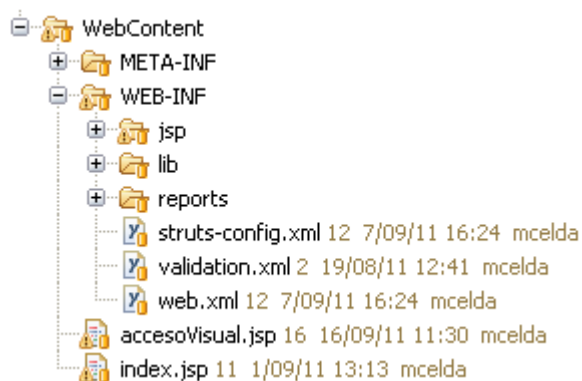


Fig.15 Estructura de archivos

Se han incorporado en la carpeta "lib", dentro de la carpeta WEB-INF todos los archivos JAR necesarios para el correcto funcionamiento de la aplicación. Al momento de compilar los archivos java se debe asegurar que los archivos contenidos dentro de la carpeta "lib" estén en el classpath.

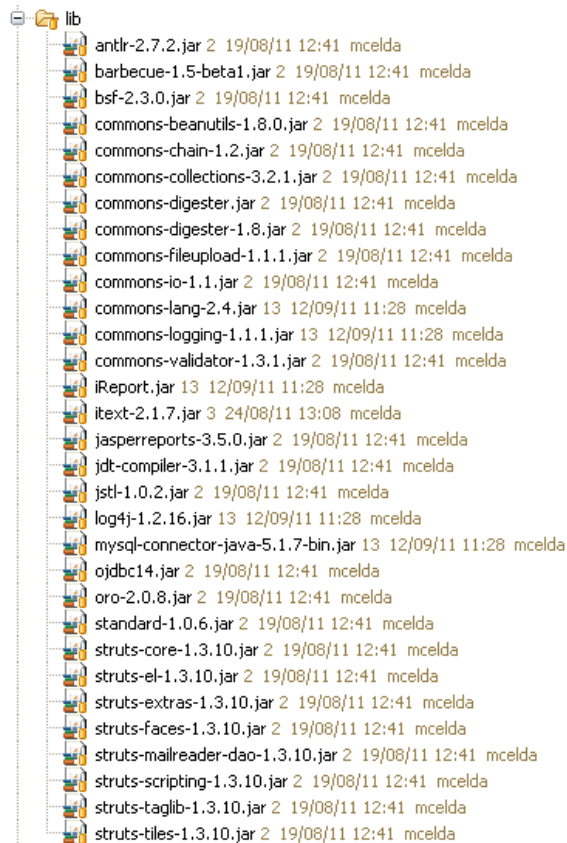


Fig.16 Librerías

Componentes de Struts

MVC es parte fundamental de Struts, para explicar como encajan las partes de este patrón, se explicará un ejemplo de la “Alta de Producto”.

Elementos de la Vista

La vista esta conformada de dos JSPs: altaProducto.jsp y exito.jsp, dentro de index.jsp se declara una forma que hace referencia a la acción altaProducto.do, esta compuesta de un par de campos de texto y un botón para enviar la forma. El campo de texto tiene un atributo name = “nombre” que especifica un nombre lógico para la forma.

```
<form name="formularioAltaProducto" action="altaProducto.do" method="POST">
  <%@ include file='cabecera.jsp' %>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title><bean:message key="altaProducto.TITULO"/></title>
  </head>
```

```

<table width="100%" border="0" cellpadding="0" cellspacing="0">
  <tr class="etiquetaTexto4">
    <td width="60">
      <bean:message key="altaProducto.CODIGO" />
    </td>
    <td>
      <input type="textfield" name="value(CODIGO)" id="CODIGO" disabled/>
    </td>
  </tr>
  ...
</form>

```

Para el manejo de formas, Struts provee una clase llamada ActionForm que se debe extender para darle funcionalidad. Dentro de la subclase, por cada campo dentro de la forma, se debe declarar una variable instancia con el mismo nombre que se asignó al atributo name en el código HTML, para cada variable se deben escribir métodos get y set como se muestra en el siguiente ejemplo:

```

package es.form;

import org.apache.struts.action.ActionForm;

/**
 * @author informatica4
 *
 */
public class ProductoForm extends ActionForm{

    private String codigo;
    private String codigoean;
    private String descripcion;
    private String lote;
    private String cantidad;
    public String getCodigo() {
        return codigo;
    }
    public void setCodigo(String codigo) {
        this.codigo = codigo;
    }
    public String getDescripcion() {
        return descripcion;
    }
    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
    ...
}

```

Adicionalmente se pueden sobrescribir los métodos reset y validate que tienen funcionalidades específicas. El método reset es invocado para reinicializar las variables a un valor dado por el programador, mientras el método validate se usa para validar que los datos no tengan errores de captura o valores nulos.

Mediante la siguiente línea de código se pueden recuperar atributos dentro del request, Ejemplo:

```
<h1>Hola <%= request.getAttribute("nombre")%></h1>
```


La línea anterior pide un atributo llamado “nombre” dentro del request y lo despliega en la página.

Elementos del controlador

Struts implementa el controlador mediante la clase ActionServlet del paquete org.apache.struts.action, es importante aclarar que esta clase es concreta, es el servlet principal de nuestra aplicación y no se necesita extender para poder usar el framework.

Toda aplicación Web debe tener un descriptor de despliegue (conocido en inglés como deployment descriptor) llamado web.xml, en este archivo se debe especificar el uso de Struts de la siguiente manera:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app id="WebApp_ID">
  <display-name>Gestión Almacén</display-name>
  <servlet>
    <servlet-name>action</servlet-name>
    <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
    <!-- <init-param>
      <param-name>application</param-name>
      <param-value>ApplicationResources</param-value>
    </init-param> -->

    <init-param>
      <param-name>config</param-name>
      <param-value>/WEB-INF/struts-config.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet>
    <servlet-name>ServletReporte</servlet-name>
    <display-name>ServletReporte</display-name>
    <description></description>
    <servlet-class>es.postigo.action.ServletReporte</servlet-class>
  </servlet>
  <servlet>
    <servlet-name>Conexion</servlet-name>
    <display-name>Conexion</display-name>
    <description></description>
    <servlet-class>es.postigo.utility.Conexion</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.do</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>ServletReporte</servlet-name>
    <url-pattern>/ServletReporte</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>Conexion</servlet-name>
    <url-pattern>/Conexion</url-pattern>
  </servlet-mapping>

  <session-config>
```

```

    <session-timeout>
      30
    </session-timeout>
  </session-config>

  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>

```

El desarrollador indica al framework las acciones, las formas y otros atributos mediante un archivo de configuración que por convención se llama struts-config.xml y tiene los siguientes campos para la aplicación "Hola Mundo":

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE struts-config PUBLIC
  "-//Apache Software Foundation//DTD Struts Configuration 1.3//EN"
  "http://struts.apache.org/dtds/struts-config_1_3.dtd">
<struts-config>
  <form-beans>
    <form-bean name="registroForm"
type="org.apache.struts.validator.DynaValidatorForm">
      <form-property name="login" type="java.lang.String"></form-property>
      <form-property name="pass" type="java.lang.String"></form-property>
    </form-bean>
    <form-bean name="formularioInventario"
type="es.postigo.struts.forms.FormularioMAP">
    </form-bean>
    <form-bean name="formularioAltaProducto"
type="es.postigo.struts.forms.FormularioMAP">
      <form-property name="codigo" type="java.lang.String"></form-property>
      <form-property name="descripcion" type="java.lang.String"></form-
property>
      <form-property name="lote" type="java.lang.String"></form-property>
      <form-property name="cantidad" type="java.lang.String"></form-
property>
      <form-property name="codigoean" type="java.lang.String"></form-
property>
    </form-bean>
    <form-bean name="formularioAltaUbicacion"
type="es.postigo.struts.forms.FormularioMAP">
      <form-property name="codigo" type="java.lang.String"></form-property>
      <form-property name="descripcion" type="java.lang.String"></form-
property>
      <form-property name="codigoean" type="java.lang.String"></form-
property>
    </form-bean>
    ...
  </form-beans>

  <action-mappings>
    <action path="/productoUbicacion"
      type="es.postigo.action.ProductoUbicacionAction"
      name="registroForm"
      scope="session"
      validate="true"
      input="/index.jsp">
      <forward name="ok" path="/WEB-INF/jsp/inventario.jsp"/>
      <forward name="SUCCESS" path="/WEB-INF/jsp/inventario.jsp"/>
      <forward name="cancel" path="/WEB-INF/jsp/cancel.jsp"/>
    </action>
    <action path="/listadoProductos"
      type="es.postigo.action.ProductoUbicacionAction"
      name="registroForm"
      scope="session"
      validate="true"
      input="/index.jsp">
      <forward name="ok" path="/WEB-INF/jsp/listadoProductos.jsp"/>
      <forward name="volver" path="/WEB-INF/jsp/inventario.jsp"/>

```

```

        <forward name="SUCCESS"           path="/WEB-INF/jsp/listadoProductos.jsp" />
        <forward name="ERROR"            path="/WEB-INF/jsp/error.jsp" />
        <forward name="inicio"           path="/index.jsp" />
    </action>
    ...
</action-mappings>

<!-- Message Resources-->
<message-resources parameter="es.postigo.resources.ApplicationResources" />

<plug-in className="org.apache.struts.validator.ValidatorPlugIn">
    <set-property
        property="pathnames"
        value="/org/apache/struts/validator/validator-rules.xml, /WEB-
INF/validation.xml" />
</plug-in>
</struts-config>

```

La etiqueta form-beans sirve para declarar las clases que representan formas HTML, en el atributo name se escribe un nombre lógico con el que se le hará una referencia futura y en type se especifica el nombre completo de la clase.

Dentro de la etiqueta action-mappings se especifican las acciones, en este caso tenemos una sola acción pero por lo regular se tendrá más de una. La etiqueta action describe los atributos de una acción, en path se indica el nombre de como se llamará al recurso sin el ".do" (dado que esta terminación sirve para llamar al ActionServlet) y agregándose una diagonal al principio. El atributo name sirve para especificar el ActionForm que se usará por esta Action mediante su nombre lógico, validate indica si se llamará o no al método validate del ActionForm, así como input debe tener el nombre del JSP al que se redireccionará si validate regresa algún error.

La etiqueta forward especifica un nombre lógico y una página JSP, que el Action utiliza para continuar con el flujo de la navegación.

Para crear una acción se debe extender la clase org.apache.struts.Action y sobrescribir el método execute que tiene la siguiente firma:

```

public ActionForward execute( ActionMapping mapping, ActionForm form,
    HttpServletRequest request, HttpServletResponse response )

```

El siguiente ejemplo muestra como implementar el Action de la aplicación para la pantalla de Login:

```

package es.action;
public class InventarioAction extends GenericoAction{

    public ActionForward execute(ActionMapping mapping, ActionForm form,
    HttpServletRequest request, HttpServletResponse response) throws Exception {
        DynaActionForm registerForm = (DynaActionForm) form;
    }
}

```

```

String lStrResult = "ok";
/**Comprobar sesión.*/
if(request.getSession().getAttribute("registerUser") == null){
    return mapping.findForward("inicio");
}
// Extraemos los datos del formulario
String login = (String) registerForm.get("login");
String pass = (String) registerForm.get("pass");

// Configurariamos los objetos Request, Session, etc. que necesita la siguiente
vista a mostrar
User user = new User();
user.setLogin(login.toUpperCase());
user.setPass(pass);
lStrResult = validarUsuario(user);
request.getSession().setAttribute("registerUser", user);
// Mostramos la siguiente vista
return mapping.findForward(lStrResult);
}
}
}

```

Elementos del Modelo

El modelo es lo que al desarrollador le interesa. En el modelo se lleva acabo la lógica aplicativa del sistema Web que se esta desarrollando. Struts no provee elementos que faciliten al desarrollador la implementación del modelo, ya que es responsabilidad del programador hacerlo. A continuación se muestra el modelo utilizado por la aplicación ejemplo:

```

package es.dao.impl;

public class ProductoUbicacionDaoImpl extends AbstractDataAccessObject implements
ProductoUbicacionDao
{
    ...

    public long obtenerId(String pStrIdCampo, String pStrTabla, String pStrWhere,
Connection con) throws ProductoUbicacionDaoException {

        String lStrSQL = " SELECT MAX(" + pStrIdCampo + ") FROM " + pStrTabla;
        if(pStrWhere.length()>0){
            lStrSQL += " WHERE "+ pStrWhere;
        }
        //return (getLong(lStrSQL) + 1);
        PreparedStatement stmt = null;
        ResultSet rs = null;
        long max = 0;
        try {
            stmt = con.prepareStatement(lStrSQL);
            rs = stmt.executeQuery();
            while (rs.next()) {
                max = Long.parseLong(rs.getString("MAX(CODIGO)") + 1);
            }
        } catch (SQLException e) {
            e.printStackTrace();
            throw new ProductoUbicacionDaoException( "SQLException: " +
e.getMessage(), e );
        }
        return max;
    }

    /**TODO: String index out of range: -10*/
    public int generaEAN13(String pStrCodigo) throws ProductoUbicacionDaoException {
        // Cálculo del dígito de control EAN
        int iSum = 0;
        int iSumInpar = 0;
    }
}

```

```

        int iDigit = 0;
        String EAN = pStrCodigo;
        EAN = "000000000001";

        for (int i = EAN.length(); i > 0; i--)
        {
            iDigit = Integer.parseInt(EAN.substring(i - 1, i));
            if (i % 2 != 0)
            {
                iSumInpar += iDigit;
            }
            else
            {
                iSum += iDigit;
            }
        }

        iDigit = (iSumInpar*3) + iSum ;

        int iCheckSum = (10 - (iDigit % 10)) % 10;
        System.out.println("Digito de control: " + iCheckSum);
        return iCheckSum;
    }

    /**
     * Cálculo del dígito de control EAN
     * @param lStrCodigo
     * @return checksum
     */
    public int generaChecksumEAN(String lStrCodigo){
        int lIntPares = 0;
        int lIntImpares = 0;
        int lIntSuma = 0;
        int lIntResultado = 0;

        //lStrCodigo = reverse(lStrCodigo);

        for (int i = lStrCodigo.length(); i > 0; i--) {
            int lIntDigito = Integer.parseInt(lStrCodigo.substring(i - 1, i));
            if (i % 2 != 0) {
                lIntPares += lIntDigito;
            } else {
                lIntImpares += lIntDigito;
            }
        }

        lIntResultado = (10 - (((lIntImpares * 3) + lIntPares) % 10)) % 10;

        return lIntResultado;
    }

    public String reverse(String str) {
        if ((null == str) || (str.length() <= 1)) {
            return str;
        }
        StringBuffer reverse = new StringBuffer(str.length());
        for (int i = str.length() - 1; i >= 0; i--) {
            reverse.append(str.charAt(i));
        }
        return reverse.toString();
    }

    public ProductoUbicacion findByCodigoEAN(String codigoEAN)
        throws ProductoUbicacionDaoException {
        List<ProductoUbicacion> ret = findByDynamicSelect( SQL_SELECT + " WHERE
CODIGOEAN = ?", new Object[] { codigoEAN } );
        return ret.size()==0 ? null : ret.get(0);
    }
}

```

Correr la Aplicación

Para correr la aplicación se deben compilar las clases y copiar los paquetes generados al directorio classes dentro de WEB-INF, por otra parte se deben copiar también las páginas o JSPs encargadas de la vista a la carpeta raíz, una vez obtenida la estructura, se procede a copiar la carpeta de la aplicación al directorio de nuestro contenedor de Servlets. La aplicación funciona de la siguiente manera:

1. El sistema pregunta al usuario su nombre
2. El usuario lo escribe y presiona el botón de enviar
3. El sistema procesa la solicitud
4. El sistema despliega el resultado



The screenshot shows the login interface for 'Gestión Almacén'. It features a blue logo with a white 'P' on a square background. Below the logo, the text 'Gestión Almacén' is displayed in a bold, black font. Underneath, there are two input fields: 'Login:' and 'Pass:'. A blue button labeled 'Enviar' is positioned to the right of the password field.

Fig.17 Pantalla Login



The screenshot displays the main menu of the 'Gestión Almacén' application. At the top left, the title 'Gestión Almacén' is shown. To its right, the user's name 'Usuario: MARIO' is displayed. In the top right corner, there is a 'CERRAR SESIÓN' button with a red arrow icon. Below the title, the text 'Seleccione una opción:' is followed by a list of nine menu items, each with a blue underline:

- 1.- [Alta producto](#)
- 2.- [Alta ubicación](#)
- 3.- [Entrada producto](#)
- 4.- [Salida producto](#)
- 5.- [Movimiento productos](#)
- 6.- [Listado productos](#)
- 7.- [Listado ubicaciones](#)
- 8.- [Listado movimientos](#)
- 9.- [Listado Inventario](#)

Fig.18 Menú Principal

Herramientas

En nuestro proyecto, hemos utilizado la herramienta informática denominada **MS Project 2003 Professional** para realizar la planificación. Dicha herramienta es muy conocida dentro del ámbito de la planificación de proyectos. Permite realizar un seguimiento y analizar los proyectos de una forma eficaz. Además de aportar un mejor control financiero y un análisis más completo, y dar opción a poder presentar la información del proyecto de un modo efectivo. Ayuda a usar informes predefinidos para seleccionar, obtener vistas previas e imprimir la información de uso más frecuente y necesaria para administrar los proyectos, realizar un seguimiento de los recursos, mantener los costos y comunicar el progreso a los miembros del equipo. Destaca que es bastante intuitivo y fácil de utilizar.

También hemos usado la herramienta **Microsoft Visio**, que es un software de dibujo vectorial. Las herramientas que lo componen permiten realizar diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas, Entidad - Relación y más, que permiten iniciarse en los lenguajes de programación. Aunque originalmente apuntaba a ser una aplicación para dibujo técnico para el campo de Ingeniería. En nuestro proyecto nos ha facilitado la tarea de crear el diagrama del Entidad - Relación y el diagrama de Casos de Usos. Ha resultado ser una herramienta muy útil.

Y para obtener el diagrama de clases se ha utilizado una herramienta de código abierto, orientada al modelado UML, llamada **ArgoUML**.

7. Manual de Usuario

7.1. Introducción

En las siguientes páginas mostraremos el manejo de la aplicación para el Mantenimiento del Inventario del Almacén, y sus diferentes funcionalidades.

Esta aplicación nos permitirá:

- Tener controlado el Stock de cualquier articulo/producto que tengamos en nuestro almacén.
- Dar de alta productos nuevos.
- Saber la ubicación exacta de cada producto en todo momento (desde que entra a nuestro almacén, hasta que le damos salida).
- Ubicar o cambiar de ubicación productos
- Dar la entrada/salida productos
- Seguir los movimientos internos de los productos mediante un histórico de los movimientos.
- Consultar la ficha del producto

Tener controlado en todo momento las cantidades en stock, permite hacer mejor las previsiones, evita las grandes perdidas de tiempo en buscar material que no se sabe donde esta y nos ayuda a todos a llevar a cabo nuestro trabajo de manera más rápida y eficiente.

Se han implementado las siguientes funciones:

- 1.- Alta producto
- 2.- Alta ubicación
- 3.- Entrada producto
- 4.- Salida producto
- 5.- Movimiento productos
- 6.- Listado productos
- 7.- Listado ubicaciones
- 8.- Listado movimientos
- 9.- Listado Inventario

Como posibles propuestas de ampliación para cubrir mejor las necesidades de los operarios del almacén, y que la aplicación les resulte de mayor utilidad en su día a día, se me ocurren los siguientes puntos a desarrollar:

- La impresión de informes de los listados en PDF
- La exportación a EXCEL de los listados
- Gestión automática de las ubicaciones del almacén
- Actualizar el aspecto de nuestra aplicación mediante CSS, imágenes,...
- ...

Mediante los siguientes apartados iremos profundizando en el manejo de cada una de las funcionalidades.

7.2. Como acceder a la aplicación

Para acceder a la aplicación podemos utilizar el terminal con lector de código de barras (INTERMEC CK3) o un PC normal con acceso a la red local.

7.2.1. Para acceder mediante el terminal (INTERMEC)

Se ha instalado un punto de acceso en el almacén (ANTENA1) y se ha comprobado que el terminal tiene cobertura en los diferentes espacios de trabajo.

Una vez encendido el terminal y conectado al punto de acceso,

- 1.- Pinchar en el menú "Start", situado en la esquina superior izquierda.
- 2.- A continuación seleccionar "Internet Explorer"
- 3.- Pinchar en "Menu" esquina inferior derecha
- 4.- Seleccionar Favorites
- 5.- Seleccionar "Inventario"

Y ya estamos en disposición de validarnos en la aplicación.



Fig.19 Pantalla de Login

7.2.2. Para acceder desde cualquier PC

Desde un PC normal, simplemente comprobar que estamos conectados en la misma red local, y si disponemos de conexión, abrir un explorador de Internet (Internet Explorer, Mozilla Firefox, Google Chrome, ...) y teclear:

<http://192.168.0.205:8080/GestionAlmacen>

Esta dirección nos cargara en el navegador la pantalla de inicio de la aplicación en la que debemos de validarnos.



Fig.20 Pantalla de Login

7.2.3. Validarse en la aplicación

Para validarse en la aplicación es necesario introducir un usuario y contraseña validos, la gestión de los usuarios la realiza el administrador de la BBDD, por lo que si no tiene usuario asignado deberá solicitarlo al departamento de informática.



The image shows a login form for 'Gestión Almacén'. The form consists of two text input fields, one for 'Login:' and one for 'Pass:'. To the right of the 'Pass:' field is a blue button with the text 'Enviar'. The entire form is centered on a light gray background.

Fig.21 Pantalla de Login

Hay que tener en cuenta que la validación es sensible a mayúsculas y minúsculas, por lo que tanto el usuario como la contraseña deben de escribirse de manera correcta y respetando dicha restricción.

7.3. Menú principal

Una vez nos hayamos validado correctamente accedemos a la pantalla principal de la aplicación, consiste en un menú con las diferentes opciones disponibles.



Fig.22 Menú Principal

Para acceder a la funcionalidad deseada podemos seleccionarla mediante el puntero o el ratón, o simplemente pulsar el número que tiene asignado (Accesos rápidos).

7.3.1. Alta de producto

Para acceder al alta de un producto nuevo, podemos “clickar” sobre esta opción en el menú principal, o pulsar “1”.

Una vez seleccionada esta opción nos aparecerá la pantalla de alta de producto.

Alta Producto

Usuario: MARIO

Código:

EAN13:

Cód.
Compras:

Descripción:

Lote:

Cantidad:

Unidades:

Ubicación:

Fig.23 Pantalla de Alta de Producto

Esta pantalla nos permite crear productos que hasta el momento no existían en nuestra base de datos. Para crear un nuevo producto tenemos que tener en cuenta un par de conceptos:

- El código del producto lo genera el programa
- El código EAN del producto también es generado mediante el programa.
- El resto de campos los debe de rellenar el usuario

Por lo tanto nos aparecen deshabilitados los campos "CODIGO" y "CODIGO EAN", y tendremos el foco en "CODIGO COMPRAS".

Tenemos que rellenar obligatoriamente los campos "CODIGO COMPRAS", "DESCRIPCION" y "CANTIDAD".

El campo “UNIDADES” es un desplegable que nos permite seleccionar la unidad de medida deseada (Por defecto saldrá seleccionada la opción “UD”):

- UD – Unidades
- ML – Metros lineales
- M2 – Metros cuadrados
- KG – Kilogramos
- RO – Rollo

También se añadirá por defecto el valor “1” en el campo “CANTIDAD”, ya que se introduce un “1” en la mayoría de los casos, y de esta forma no existirá la necesidad de teclearlo manualmente.

Una vez introducidos los datos en los diferentes campos procederemos a registrar el alta, pulsando sobre el botón “Aceptar”, que nos mostrara una pantalla de éxito para que sepamos que el nuevo producto se ha insertado correctamente en la base de datos.



Fig.24 Pantalla de Éxito

Si se ha producido algún problema, nos aparecerá la pantalla de error, y tendremos que repetir la introducción de los datos para poder llevar a cabo la acción.

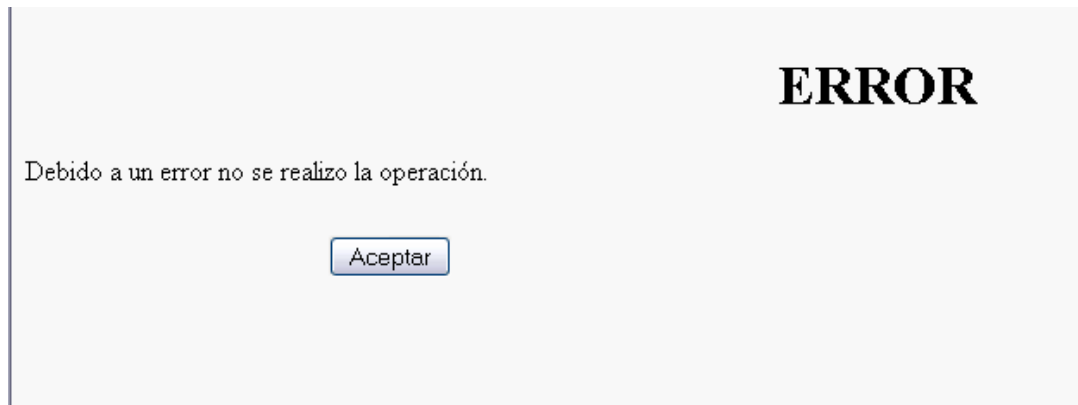


Fig.25 Pantalla de Error

En el caso de no querer llevar a cabo el alta del producto, también podemos pulsar sobre el botón “Volver”, que nos devolverá al menú principal.



Fig.26 Botón Volver

Existen validaciones JavaScript para los campos obligatorios, por lo que si no se introduce valor en dichos campos o el valor introducido no es el correcto, nos mostrara un Popup con el correspondiente mensaje de error y nos colocara el foco en el campo que debemos modificar.

7.3.2. Alta Ubicación

Para crear una nueva ubicación, solo tenemos que pulsar sobre esta opción en el menú principal, o seleccionarla mediante el “2”.

Una vez seleccionada esta opción nos aparecerá la pantalla de alta de ubicación.

Alta Ubicación Usuario: MARIO

Código:

EAN13:

Descripción:

Fig.27 Pantalla Alta Ubicación

Esta pantalla nos permite crear nuevas ubicaciones teniendo en cuenta un par de conceptos:

- El código de la ubicación lo genera el programa
- El código EAN de la ubicación también es generado mediante el programa

Por lo tanto nos aparecen deshabilitados los campos “CODIGO” y “CODIGOEAN”.

Tenemos que rellenar obligatoriamente el campo “DESCRIPCIÓN”.

Una vez rellenada la descripción procederemos a registrar el alta, pulsando sobre el botón “Aceptar”, que nos mostrara una pantalla de éxito para que sepamos que la nueva ubicación se ha insertado correctamente en la base de datos.

Exito

La operación se ha realizado con éxito.

Fig.28 Pantalla de Exito

Si se ha producido algún problema, nos aparecerá la pantalla de error, y tendremos que repetir la introducción de los datos para poder llevar a cabo la acción.

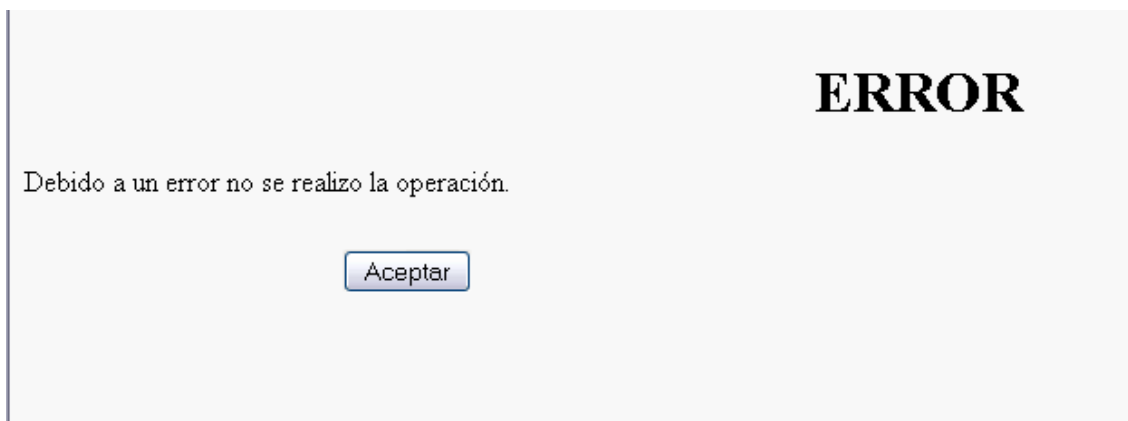


Fig.29 Pantalla de Error

En el caso de no querer llevar a cabo el alta del producto, también podemos pulsar sobre el botón “Volver”, que nos devolverá al menú principal.



Fig.30 Botón Volver

Existen validaciones JavaScript para comprobar que hemos introducido algun valor descripción de la ubicación, por lo que si no se introduce valor o no es el correcto, nos mostrara un Popup con el correspondiente mensaje de error y nos colocara el foco en el campo descripción para que lo modifiquemos.

7.3.3. Entrada producto

Para dar de entrada un producto, solo tenemos que pulsar sobre esta opción en el menú principal, o seleccionarla mediante el “3”.



The screenshot shows a web form titled "Entrada Producto" with the user name "MARIO" in the top right. The form contains two sets of input fields. The first set includes "Cód. EAN Ubicación:" with an empty text box, "Desc. Ubicación:" with an empty text area, "Cód. EAN Producto:" with an empty text box, "Desc. Producto:" with an empty text area, and "Cantidad:" with an empty text box. The second set of fields is identical but for the product. At the bottom, there are two buttons: "Aceptar" and "Volver".

Fig.31 Pantalla Entrada de Producto

En esta pantalla tenemos tres campos habilitados, “COD. EAN UBICACIÓN”, “COD. EAN PRODUCTO” y “CANTIDAD”, todos ellos obligatorios.

Y dos campos deshabilitados “DESC. UBICACIÓN” y “DESC. PRODUCTO”, estos campos se rellenaran automáticamente recogiendo el COD. EAN UBICACIÓN en el primero de los casos, y el COD. EAN PRODUCTO en el segundo. Con el código EAN obtenido se extrae la descripción de la Base de Datos y se coloca en el campo correspondiente.



The screenshot shows the same "Entrada Producto" form, but now with data entered. The "Cód. EAN Ubicación:" field contains "0100000000007", and the "Desc. Ubicación:" field contains "RETECPIN". The "Cód. EAN Producto:" field contains "0000000000017", and the "Desc. Producto:" field contains "BARRERA DOBLE ONDA 4318MM". The "Cantidad:" field contains "1". The "Aceptar" and "Volver" buttons are still present at the bottom.

Fig.32 Pantalla Entrada de Productos con Datos

Si utilizamos el terminal con lector de código de barras, podemos pulsar el “gatillo” (botón azul ubicado en la parte inferior de dicho terminal) y mediante el haz de luz que saca por la parte frontal, leer el código de barras de la etiqueta correspondiente, tanto a la ubicación como al producto.

En caso de utilizar un PC normal, tendríamos que teclear dichos códigos manualmente, el resultado final es el mismo en ambos casos.

Una vez rellenados los campos pulsaremos el botón “Aceptar” que nos mostrara una pantalla de éxito para que sepamos que el producto se ha ubicado correctamente.



Fig.33 Pantalla de Exito

Si ha ocurrido algún problema, nos aparecerá la pantalla de error, y tendremos que volver a introducir los datos para poder llevar a cabo la acción.

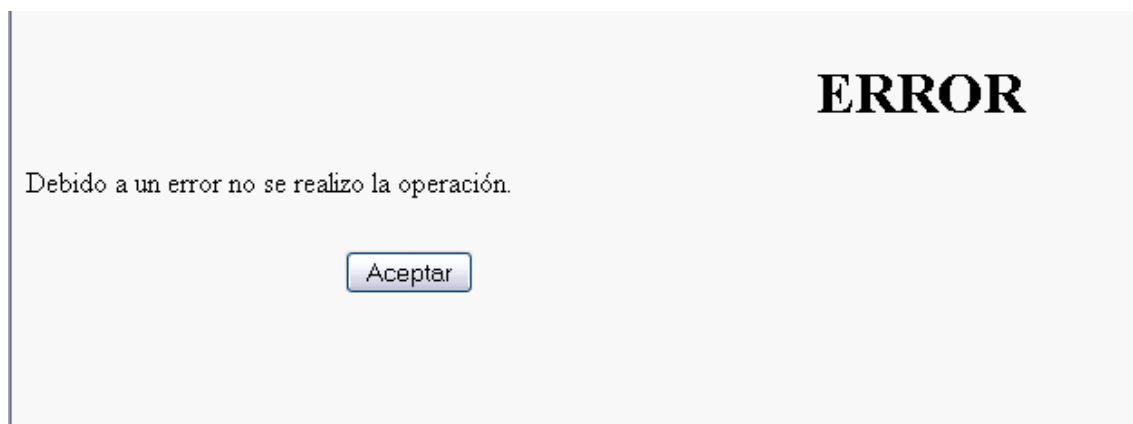


Fig.34 Pantalla de Error

En el caso de no querer llevar a cabo la entrada del producto, también podemos pulsar sobre el botón “Volver”, que nos devolverá al menú principal.



Fig.35 Botón Volver

7.3.4. Salida producto

Para la salida de un producto, solo tenemos que pulsar sobre esta opción en el menú principal, o seleccionarla mediante el “4”.

A screenshot of a web application interface titled "Salida Producto". The user is identified as "MARIO". The form contains two sets of input fields. The first set includes "Cód. EAN Ubicación:" with a text input, and "Desc. Producto:" with a disabled text area. The second set includes "Cód. EAN Producto:" with a text input, "Desc. Producto:" with a disabled text area, and "Cantidad:" with a text input. At the bottom, there are two buttons: "Aceptar" and "Volver".

Fig.36 Pantalla Salida de Producto

En esta pantalla tenemos tres campos habilitados, “COD. EAN UBICACIÓN”, “COD. EAN PRODUCTO” y “CANTIDAD”, todos ellos obligatorios.

Y dos campos deshabilitados “DESC. UBICACIÓN” y “DESC. PRODUCTO”, estos campos se rellenaran automáticamente recogiendo el COD. EAN UBICACIÓN en el primero de los casos, y el COD. EAN PRODUCTO en el segundo. Con el código EAN obtenido se extrae la descripción de la Base de Datos y se coloca en el campo correspondiente.

Salida Producto Usuario: MARIO

Cód. EAN Ubicación: 0100000000007

Desc. Producto: RETECPIN

Cód. EAN Producto: 0000000000017

Desc. Producto: BARRERA DOBLE ONDA 4318MM

Cantidad: 1

Aceptar Volver

Fig.37 Pantalla Salida de Producto con datos

Si utilizamos el terminal con lector de código de barras, podemos pulsar el “gatillo” (botón azul ubicado en la parte inferior de dicho terminal) y mediante el haz de luz que saca por la parte frontal, leer el código de barras de la etiqueta correspondiente, tanto a la ubicación como al producto.

En caso de utilizar un PC normal, tendríamos que teclear dichos códigos manualmente, el resultado final es el mismo en ambos casos.

Una vez rellenados los campos pulsaremos el botón “Aceptar” que nos mostrara una pantalla de éxito para que sepamos que se ha generado correctamente la salida del producto de nuestro almacén.

Exito

La operación se ha realizado con éxito.

Aceptar

Fig.38 Pantalla de éxito

Si ha ocurrido algún problema, nos aparecerá la pantalla de error, y tendremos que volver a introducir los datos para poder llevar a cabo la acción.

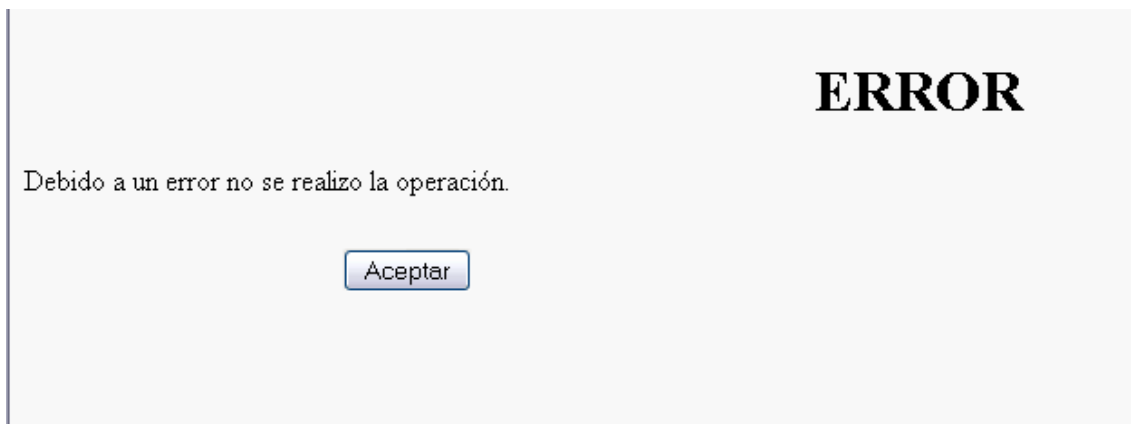


Fig.39 Pantalla de error

En el caso de no querer llevar a cabo la salida del producto, también podemos pulsar sobre el botón "Volver", que nos devolverá al menú principal.



Fig.40 Botón Volver

7.3.5. Movimiento de productos

Para la pantalla de movimiento de productos, solo tenemos que pulsar sobre esta opción en el menú principal, o seleccionarla mediante el “5”.

The screenshot shows a web form titled "Movimiento Productos". In the top right corner, it displays "Usuario: MARIO". The form contains the following fields and controls:

- Ubicación Origen:** A text input field.
- Desc. Origen:** A disabled text area.
- Cód. EAN Producto:** A text input field.
- Desc. Producto:** A disabled text area.
- Ubicación Destino:** A text input field.
- Desc. Destino:** A disabled text area.
- Cantidad:** A text input field.
- Buttons:** "Aceptar" and "Volver" buttons at the bottom.

Fig.41 Pantalla de Movimiento de Productos

En esta pantalla tenemos cuatro campos habilitados, “UBICACIÓN ORIGEN”, “COD. EAN PRODUCTO”, “UBICACIÓN DESTINO” y “CANTIDAD”, todos ellos obligatorios.

Y dos campos deshabilitados “DESC. ORIGEN”, “DESC. PRODUCTO” y “DESC. DESTINO”, estos campos se rellenaran automáticamente recogiendo el UBICACIÓN ORIGEN en el primero de los casos, y el COD. EAN PRODUCTO en el segundo y la UBICACIÓN DESTINO en el tercero. Con el código EAN obtenido se extrae la descripción de la Base de Datos y se coloca en el campo correspondiente.

Movimiento Productos Usuario: MARIO

Ubicación Origen: 0100000000007

Desc. Origen: RETECPIN

Cód. EAN Producto: 0000000000017

Desc. Producto: BARRERA DOBLE ONDA 4318MM

Ubicación Destino: 0110000000006

Desc. Destino: MAFER

Cantidad: 1

Aceptar Volver

Fig.42 Pantalla de Movimiento de productos

Si utilizamos el terminal con lector de código de barras, podemos pulsar el “gatillo” (botón azul ubicado en la parte inferior de dicho terminal) y mediante el haz de luz que saca por la parte frontal, leer el código de barras de la etiqueta correspondiente, tanto a las ubicaciones como al producto.

En caso de utilizar un PC normal, tendríamos que teclear dichos códigos manualmente, el resultado final es el mismo en ambos casos.

Una vez rellenados los campos pulsaremos el botón “Aceptar” que nos mostrara una pantalla de éxito para que sepamos que se ha generado correctamente el movimiento del producto de la ubicación origen a la ubicación destino.

Exito

La operación se ha realizado con éxito.

Aceptar

Fig.43 Pantalla de exito

Si ha ocurrido algún problema, nos aparecerá la pantalla de error, y tendremos que volver a introducir los datos para poder llevar a cabo la acción.

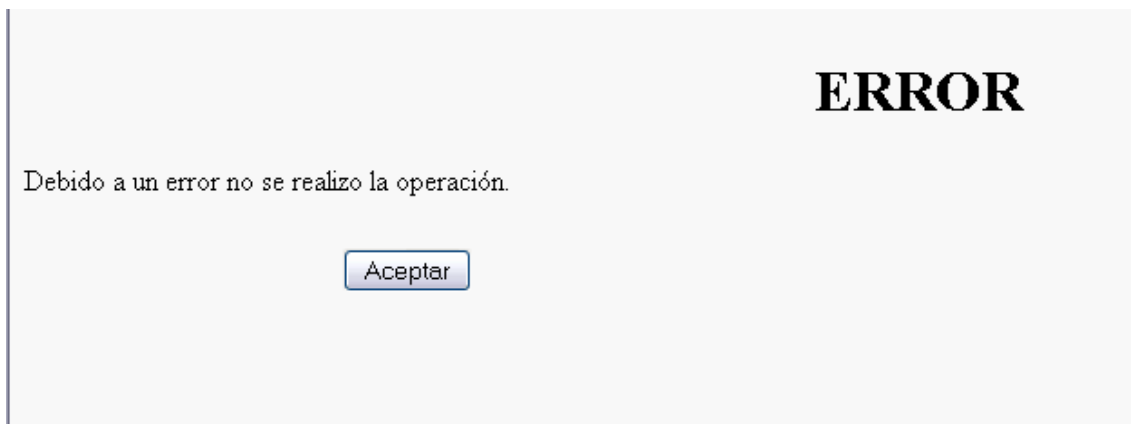


Fig.44 Pantalla de error

En el caso de no querer llevar a cabo el movimiento de los productos, también podemos pulsar sobre el botón "Volver", que nos devolverá al menú principal.



Fig.45 Botón Volver

7.3.6. Listado de productos

Para ver el listado de productos, solo tenemos que pulsar sobre esta opción en el menú principal, o seleccionarla mediante el “6”.

Se nos cargara un listado en la pantalla, en el que se mostraran los productos que tenemos disponibles en nuestra base de datos.

Listado Productos Fecha: 16/9/2011
Usuario: MARIO

Buscar:

Orden:

Código	Descripción	Lote	Cantidad	Código EAN13
000000000001	BARRERA DOBLE ONDA 4318MM	0	1	0000000000017
000000000002	POSTE TUBULAR 1500MM	0	1	0000000000024
000000000003	SEPARADOR ESTANDAR IZQUIERDO	0	1	0000000000031
000000000004	SEPARADOR ESTANDAR DERECHO	0	1	0000000000048
000000000005	CONECTOR EN "U" PARA POSTE TUBULAR	0	1	0000000000055
000000000006	TORNILLO REDONDO CUELLO OVAL M16X30 5.8	0	1	0000000000062
000000000007	TORNILLO REDONDO CUELLO OVAL M16X40 5.8	0	1	0000000000079
000000000008	TORNILLO HEXAGONAL M16X35 5.8	0	1	0000000000086
000000000009	TORNILLO HEXAGONAL M10X40 DIN 933 6.8	0	1	0000000000093
000000000010	TUERCA HEXAGONAL M16 CLASE 5	0	1	0000000000109

Pag. 1 [siguiente](#) [>>](#)

Fig.46 Listado de Productos

Esta pantalla tiene además unas funcionalidades añadidas:

- Paginación del listado en grupos de 10 productos
- Función de búsqueda entre los productos existentes
- Función de ordenar entre los productos del listado
- Ver ficha del producto

Al cargar el listado nos aparecerán los primeros diez productos ordenados por código, ya que las páginas del listado van de diez en diez. En la parte inferior del listado podremos ver en todo momento en que pagina nos encontramos y los links a las paginas siguiente, anterior primera y ultima (en caso de que las haya).

Pulsando sobre el link de “siguiente” podremos cargar los siguientes diez productos del listado, y pulsando sobre el link “anterior” se cargaran los diez productos previos.

En la parte superior derecha de la pantalla podremos ver un campo de texto y un botón “Buscar”, si introducimos en este campo la descripción o parte de dicha descripción, nos filtrara el listado, mostrándonos un nuevo listado con las coincidencias.

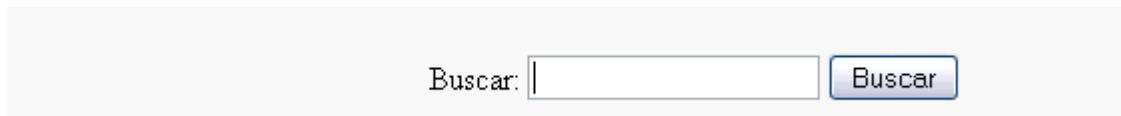


Fig.47 Funcionalidad Buscar

En la parte superior derecha de la pantalla, justo debajo de “Buscar” podremos ver un par de desplegados con el texto Orden y un botón “Ordenar”, si seleccionamos en el primer desplegable el campo a filtrar y en el segundo el orden (Ascendente o descendente), nos filtrara el listado, mostrándonos un nuevo listado con la ordenación elegida.



Fig.48 Funcionalidad Orden

Si queremos ver la ficha detallada de un producto, deberemos seleccionar la fila de dicho producto y pulsar el botón “Abrir”, que nos cargara una pantalla con la ficha del producto asociado a la fila seleccionada.



Listado Productos

Fecha: 19/9/2011
Usuario: MARIO

Buscar:

Orden:

Código	Descripción	Lote	Cantidad	Código EAN13
000000000001	BARRERA DOBLE ONDA 4318MM	0	1	0000000000017
000000000002	POSTE TUBULAR 1500MM	0	1	0000000000024
000000000003	SEPARADOR ESTANDAR IZQUIERDO	0	1	0000000000031
000000000004	SEPARADOR ESTANDAR DERECHO	0	1	0000000000048
000000000005	CONECTOR EN "U" PARA POSTE TUBULAR	0	1	0000000000055
000000000006	TORNILLO REDONDO CUELLO OVAL M16X30 5.8	0	1	0000000000062
000000000007	TORNILLO REDONDO CUELLO OVAL M16X40 5.8	0	1	0000000000079
000000000008	TORNILLO HEXAGONAL M16X35 5.8	0	1	0000000000086
000000000009	TORNILLO HEXAGONAL M10X40 DIN 933 6.8	0	1	0000000000093
000000000010	TUERCA HEXAGONAL M16 CLASE 5	0	1	0000000000109

Pag. 1 [siguiente](#) [>>](#)

Fig.49 Fila seleccionada Listado de Productos

Ficha Producto

Usuario: MARIO

Código:

Código EAN13:

Código Compras:

Descripción:

Lote:

Cantidad:

Unidades:

Fig.50 Ficha de producto

Y por ultimo, podemos volver al menú principal pulsando el botón “Volver”.



Fig.51 Botón Volver

7.3.6.1. Ficha de producto

Para acceder a la ficha de un producto, debemos seleccionar en el listado de productos la fila correspondiente al producto del cual queremos abrir su ficha, y una vez seleccionada pulsar el botón “Abrir”, esto nos cargara una pantalla con los datos referentes a dicho producto.



Fecha: 19/9/2011
Usuario: MARIO

Buscar:

Orden:

Código	Descripción	Lote	Cantidad	Código EAN13
000000000001	BARRERA DOBLE ONDA 4318MM	0	1	0000000000017
000000000002	POSTE TUBULAR 1500MM	0	1	0000000000024
000000000003	SEPARADOR ESTANDAR IZQUIERDO	0	1	0000000000031
000000000004	SEPARADOR ESTANDAR DERECHO	0	1	0000000000048
000000000005	CONECTOR EN "U" PARA POSTE TUBULAR	0	1	0000000000055
000000000006	TORNILLO REDONDO CUELLO OVAL M16X30 5.8	0	1	0000000000062
000000000007	TORNILLO REDONDO CUELLO OVAL M16X40 5.8	0	1	0000000000079
000000000008	TORNILLO HEXAGONAL M16X35 5.8	0	1	0000000000086
000000000009	TORNILLO HEXAGONAL M10X40 DIN 933 6.8	0	1	0000000000093
000000000010	TUERCA HEXAGONAL M16 CLASE 5	0	1	0000000000109

Pag. 1 [siguiente](#) [>>](#)

Fig.52 Listado de Productos

Ficha Producto

Usuario: MARIO

Código:

Código EAN13:

Código Compras:

Descripción:

Lote:

Cantidad:

Unidades: ▼

Fig.53 Ficha de Producto

Para volver al listado disponemos en la parte inferior de la pantalla de un botón “Volver”.



Fig.54 Botón Volver

A la derecha del botón “Volver” veremos el botón “Guardar”, que nos permite guardar en la base de datos las modificaciones realizadas en la ficha.



Fig.55 Botón Guardar

A la derecha del botón “Guardar” veremos el botón “Eliminar”, que nos permite borrar el producto de la base de datos, siempre y cuando no tenga registros asociados en la tabla de inventario.



Fig.56 Botón Eliminar

Y por ultimo tenemos el botón “Imprimir etiqueta”, que nos imprime automáticamente la etiqueta con el código de barras del artículo, para su posterior identificación mediante el terminal.



Fig.57 Botón Imprimir Etiqueta

La etiqueta impresa quedaría así:

DESC: BARRERA DOBLE ONDA 4318MM

EAN13:



Fig.58 Ejemplo Impresión Etiqueta Producto

7.3.7. Listado de ubicaciones

Para ver el listado de ubicaciones, solo tenemos que pulsar sobre esta opción en el menú principal, o seleccionarla mediante el “7”.

Código	Descripción	Lote	Cantidad	Código EAN13
009000000000	IMPEVA	0	0	0090000000001
010000000000	RETECPIN	0	0	0100000000007
011000000000	MAFER	0	0	0110000000006
012000000000	GTS	0	0	0120000000005
013000000000	AD HOC	0	0	0130000000004
014000000000	FUNTAM	0	0	0140000000003
015000000000	GALESA	0	0	0150000000002
016000000000	C-0	0	0	0160000000001
017000000000	C-1	0	0	0170000000000
018000000000	D-0	0	0	0180000000009

Fig.59 Listado de Ubicaciones

Se nos cargara un listado en la pantalla, en el que se mostraran las ubicaciones que tenemos disponibles en nuestra base de datos.

Esta pantalla tiene además unas funcionalidades añadidas:

- Paginación del listado en grupos de 10 ubicaciones
- Función de búsqueda entre las ubicaciones existentes
- Ver ficha de la ubicación

Al cargar la pantalla nos aparecerán las primeras diez ubicaciones ordenadas por código. En la parte inferior del listado podremos ver en todo momento en que pagina nos encontramos y los links a las paginas siguiente, anterior, primera y ultima (en caso de que las haya).

Pulsando sobre el link de “siguiente” podremos cargar las siguientes diez ubicaciones del listado, y pulsando sobre el link “anterior” se cargaran las diez ubicaciones previas.

En la parte superior derecha de la pantalla podremos ver un campo de texto y un botón “Buscar”, si introducimos en este campo la descripción o parte de dicha descripción, nos filtrara el listado, mostrándonos un nuevo listado con las coincidencias.



Fig.60 Funcionalidad Buscar

En la parte superior derecha de la pantalla, justo debajo de “Buscar” podremos ver un par de desplegables con el texto Orden y un botón “Ordenar”, si seleccionamos en el primer desplegable el campo a filtrar y en el segundo el orden (Ascendente o descendente), nos filtrara el listado, mostrándonos un nuevo listado con la ordenación elegida.



Fig.61 Funcionalidad Orden

Si queremos ver la ficha detallada de una ubicación, deberemos seleccionar la fila correspondiente a dicha ubicación y pulsar el botón “Abrir”, que nos cargara una pantalla con la ficha de la ubicación asociada a la fila seleccionada.



Listado Ubicaciones

Fecha: 19/9/2011
Usuario: MARIO

Buscar:

Orden:

Código	Descripción	Lote	Cantidad	Código EAN13
009000000000	IMPEVA	0	0	0090000000001
010000000000	RETECPIN	0	0	0100000000007
011000000000	MAFER	0	0	0110000000006
012000000000	GTS	0	0	0120000000005
013000000000	AD HOC	0	0	0130000000004
014000000000	FUNTAM	0	0	0140000000003
015000000000	GALESA	0	0	0150000000002
016000000000	C-0	0	0	0160000000001
017000000000	C-1	0	0	0170000000000
018000000000	D-0	0	0	0180000000009

Pag. 1 [siguiente](#) >>

Fig.62 Listado de Ubicaciones

Ficha Ubicación

Usuario: MARIO

Código:

Código EAN13:

Descripción:

Fig.63 Ficha de ubicación

Y por ultimo, podemos volver al menú principal pulsando el botón "Volver".



Fig.64 Botón Volver

7.3.7.1. Ficha de ubicación

Para acceder a la ficha de una ubicación, debemos seleccionar en el listado de ubicaciones la fila correspondiente a la ubicación de la cual queremos abrir su ficha, y una vez seleccionada pulsar el botón “Abrir”, esto nos cargara una pantalla con los datos referentes a dicha ubicación.

Listado Ubicaciones Fecha: 19/9/2011
Usuario: MARIO

Buscar:

Orden:

Código	Descripción	Lote	Cantidad	Código EAN13
009000000000	IMPEVA	0	0	0090000000001
010000000000	RETECPIN	0	0	0100000000007
011000000000	MAFER	0	0	0110000000006
012000000000	GTS	0	0	0120000000005
013000000000	AD HOC	0	0	0130000000004
014000000000	FUNTAM	0	0	0140000000003
015000000000	GALESA	0	0	0150000000002
016000000000	C-0	0	0	0160000000001
017000000000	C-1	0	0	0170000000000
018000000000	D-0	0	0	0180000000009

Pag. 1 [siguiente](#) >>

Fig.65 Listado de ubicaciones con fila seleccionada

Ficha Ubicación Usuario: MARIO

Código:

Código EAN13:

Descripción:

Fig.66 Ficha de ubicación

Para volver al listado disponemos en la parte inferior de la pantalla de un botón “Volver”.



Fig.67 Botón Volver

A la derecha del botón “Volver” veremos el botón “Guardar”, que nos permite guardar en la base de datos las modificaciones realizadas en la ficha.



Fig.68 Botón Guardar

A la derecha del botón “Guardar” veremos el botón “Eliminar”, que nos permite borrar el producto de la base de datos, siempre y cuando no tenga registros asociados en la tabla de inventario.



Fig.69 Botón Eliminar

Y por ultimo tenemos el botón “Imprimir etiqueta”, que nos imprime automáticamente la etiqueta con el código de barras del artículo, para su posterior identificación mediante el terminal.

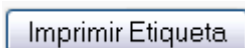


Fig.70 Botón Imprimir Etiqueta

La etiqueta impresa quedaría así:



Fig.71 Ejemplo Impresión etiqueta de Ubicación

7.3.8. Listado de movimientos

Para ver el listado de movimientos, solo tenemos que pulsar sobre esta opción en el menú principal, o seleccionarla mediante el “8”.

Se nos cargara un listado en la pantalla, en el que se mostraran el histórico de los movimientos que tenemos registrados en nuestra base de datos.

Listado Movimientos							Usuario: MARIO
Ubicación Origen	Desc. Ubicación Origen	Ubicación Destino	Desc. Ubicación Destino	Cód. Producto	Desc. Cód. Producto	Cantidad	Fecha
999000000000	Ubic. Externa	020000000000	A-0	000000000217	CABLE ELEC. GENERAL 1000V	200	2011-09-16 13:08:29.0
999000000000	Ubic. Externa	028000000000	C-2	000000000193	ESPEJO PARABOLICO O60	5	2011-09-15 17:11:39.0
031000000000	D-1	999000000000	Ubic. Externa	000000000210	CELULOSA INDUSTRIAL RECI-100	-2	2011-09-15 17:11:15.0
031000000000	D-1	999000000000	Ubic. Externa	000000000210	CELULOSA INDUSTRIAL RECI-100	-2	2011-09-15 16:57:05.0
019000000000	A-1	999000000000	Ubic. Externa	000000000174	ARANDELA M12 ZN DIN125A	-14	2011-09-14 18:49:03.0
999000000000	Ubic. Externa	026000000000	B-2	000000000216	CONO DE 50 CM	149	2011-09-14 16:35:15.0
999000000000	Ubic. Externa	025000000000	B-1	000000000215	PERNO M14 X 400 MM	90	2011-09-14 12:35:23.0
999000000000	Ubic. Externa	031000000000	D-1	000000000205	TORNILLO HEX. M12X40 ZN DIN933 4.8	520	2011-09-14 12:00:24.0
024000000000	B-0	999000000000	Ubic. Externa	000000000205	TORNILLO HEX. M12X40 ZN DIN933 4.8	-520	2011-09-14 12:00:03.0
999000000000	Ubic. Externa	031000000000	D-1	000000000167	ABRAZADERA SIMPLE O50 (OMEGA)	195	2011-09-14 11:59:23.0

Pag. 1 [siguiente](#) >>

Fig.72 Listado de Movimientos

Esta pantalla tiene además unas funcionalidades añadidas:

- Paginación del listado en grupos de 10 movimientos
- Función de búsqueda entre los movimientos existentes

Al cargar la pantalla nos aparecerán los primeros diez movimientos ordenados por fecha, de manera que los movimientos más recientes aparecerán los primeros y los más antiguos los últimos.

En la parte inferior del listado podremos ver en todo momento en que pagina nos encontramos y los links a las paginas siguiente, anterior, primera y ultima (en caso de que las haya).

Pulsando sobre el link de “siguiente” podremos cargar los siguientes diez movimientos del listado, y pulsando sobre el link “anterior” se cargaran los diez movimientos previos.

En la parte superior derecha de la pantalla podremos ver un campo de texto y un botón “Buscar”, si introducimos en este campo la ubicación origen, la ubicación destino o el código del producto, nos filtrara el listado, mostrándonos un nuevo listado con las coincidencias.

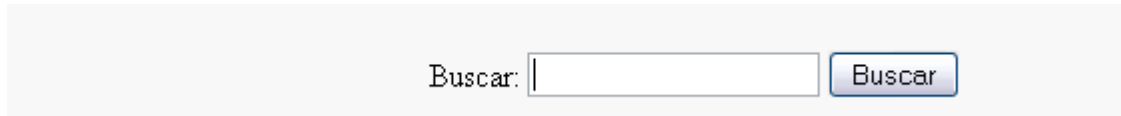


Fig.73 Funcionalidad Buscar

Y por ultimo, podemos volver al menú principal pulsando el botón “Volver”.



Fig.74 Botón Volver

7.3.9. Listado Inventario

Para ver el listado de inventario, solo tenemos que pulsar sobre esta opción en el menú principal, o seleccionarla mediante el “9”.

Se nos cargara un listado en la pantalla, en el que se mostraran el inventario que tenemos registrados en nuestra base de datos.

Inventario						Usuario: MARIO
Cód. Ubicación	Desc.	Cód. Ubicación	Cód. Producto	Desc.	Cód. Producto	Cantidad
009000000000	IMPEVA		000000000001	BARRERA DOBLE ONDA 4318MM		3
009000000000	IMPEVA		000000000068	LANA ROCA 1960X490X40MM D=100KG		2090
009000000000	IMPEVA		000000000136	HEB 140 H=4050 MM CON PLACA DE 350X350x3		2
009000000000	IMPEVA		000000000137	HEB 140 H=4070 MM CON PLACA DE 400X400x3		1
009000000000	IMPEVA		000000000138	HEB 140 H=4070 MM CON PLACA DE 400X400x3		1
009000000000	IMPEVA		000000000139	HEB 140 H=4200 MM CON PLACA DE 350X350x3		1
009000000000	IMPEVA		000000000140	HEB 180 H=5100 mm. CON PLACA DE 300X340X		1
009000000000	IMPEVA		000000000141	HEB 180 H=5050 mm. CON PLACA DE 300X340X		4
009000000000	IMPEVA		000000000142	HEB 180 H=4700 mm. CON PLACA DE 400X400X		1
009000000000	IMPEVA		000000000143	HEB 180 H=4550MM CON PLACA DE 400X400X30		1

Pag. 1 [siguiente](#) [>>](#)

Fig.75 Listado de Inventario

Esta pantalla tiene además unas funcionalidades añadidas:

- Paginación del listado en grupos de 10 entradas en el inventario
- Función de búsqueda entre las entradas de inventario existentes

Al cargar la pantalla nos aparecerán las primeras diez entradas de inventario ordenadas por Cód. Ubicación.

En la parte inferior del listado podremos ver en todo momento en que pagina nos encontramos y los links a las paginas siguiente, anterior, primera y ultima (en caso de que las haya).

Pulsando sobre el link de “siguiente” podremos cargar las siguientes diez entradas en el inventario del listado, y pulsando sobre el link “anterior” se cargaran las diez entradas previas.

En la parte superior derecha de la pantalla podremos ver un campo de texto y un botón “Buscar”, si introducimos en este campo el código EAN de la ubicación, la descripción total/parcial de la ubicación, el código EAN del producto o la descripción total/parcial del producto, nos filtrara el listado, mostrándonos un nuevo listado con las coincidencias.



Fig.76 Funcionalidad Buscar

Y por ultimo, podemos volver al menú principal pulsando el botón “Volver”.



Fig.77 Botón Volver

7.3.10. Cerrar Sesión

El botón de “Cerrar sesión” esta disponible en el menú principal, una vez pulsado este botón se cerrara la sesión, borrando todos los datos de registro del usuario y mostrándonos la pantalla de inicio, en la que nos volverá a pedir usuario y contraseña.

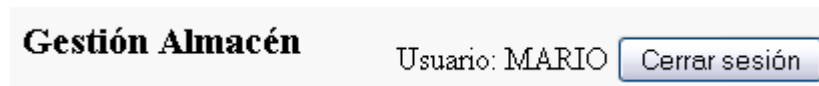


Fig.78 Funcionalidad Cerrar Sesión



Fig.79 Pantalla de Login

También se ha implementado un sistema automático de cierre de sesión por “Time out”, el funcionamiento es el siguiente:

Si pasan 10 minutos sin que hagamos uso activo de la aplicación, al intentar realizar cualquier acción, esta nos mandara a la pantalla de inicio en la que deberemos de volver a introducir el usuario y la contraseña.



Fig.80 Pantalla de Login con datos

Una vez nos hayamos vuelto a validar ya podremos llevar a cabo las acciones deseadas.

El cierre automático de sesión por “time out” se implementa como sistema de seguridad, para evitar que si un usuario por una urgencia debe de abandonar temporalmente el puesto de trabajo y se deja el terminal u ordenador con la aplicación arrancada y sin cerrar sesión, esta se cierre automáticamente a los 10 min. y no permita a un usuario no validado realizar ninguna operación.

7.3.11. Funcionalidad “Buscar” en los listados

Todos los listados están paginados para facilitar la navegación y la revisión de los datos, pero conforme se va incrementando el número de filas de nuestros listados surge la necesidad de filtrar un poco más la información proporcionada por el programa.

Detectada esta necesidad se ha implementado en los listados la funcionalidad de “Buscar”, en dichos listados dispondremos en la parte superior derecha de la pantalla un campo de texto y un botón “Buscar”, si introducimos en este campo el código, el código EAN o la descripción (total o parcial) por el que queremos buscar, nos filtrara el listado, mostrándonos un nuevo listado con las coincidencias.

Buscar:

Fig.81 Funcionalidad Buscar

Ejemplo de búsqueda por descripción parcial:



Listado Productos

Fecha: 19/9/2011
 Usuario: MARIO


Buscar:
 Orden:

Código	Descripción	Lote	Cantidad	Código EAN13
000000000006	TORNILLO REDONDO CUELLO OVAL M16X30 5.8	0	1	0000000000062
000000000007	TORNILLO REDONDO CUELLO OVAL M16X40 5.8	0	1	0000000000079
000000000008	TORNILLO HEXAGONAL M16X35 5.8	0	1	0000000000086
000000000009	TORNILLO HEXAGONAL M10X40 DIN 933 6.8	0	1	0000000000093
000000000062	TORNILLO HEXAGONAL M16X50 DIN 933 8.8	1	1	0000000000628
000000000066	TORNILLO M16X70 5.6, TUERCA Y ARANDELA	0	1	0000000000666
000000000067	TORNILLO M22X70 8.8, TUERCA Y 2 ARAND	0	1	0000000000673
000000000135	TORNILLO CABEZA HEXAGONAL 6.8 M8x80	0	1	0000000001359
000000000160	TORNILLO CAB.HEXAG. M16X70 DIN933 8.8 ZN	1	1	0000000001601
000000000163	TORNILLO CAB.HEXA. M8X40 DIN933 8.8 HDG	0	1	0000000001632

Pag. 1 [siguiente](#)

Fig.82 Ejemplo Búsqueda por descripción parcial

Ejemplo de búsqueda por Código EAN:



Listado Productos

Fecha: 19/9/2011
 Usuario: MARIO

Buscar:
 Orden:

Código	Descripción	Lote	Cantidad	Código EAN13
000000000001	BARRERA DOBLE ONDA 4318MM	0	1	0000000000017

Pag. 1

Fig.83 Ejemplo de búsqueda por Código EAN

Ejemplo de búsqueda por código:

Listado Productos Fecha: 19/9/2011
Usuario: MARIO

Buscar: 000000000002

Orden:

Código	Descripción	Lote	Cantidad	Código EAN13
000000000002	POSTE TUBULAR 1500MM	0	1	0000000000024

Pag. 1

Fig.84 Ejemplo de Búsqueda por Código

Esta funcionalidad nos será mucho más útil cuando vaya incrementando el número de filas en las que tengamos que buscar, ya que revisar fila a fila un listado, e ir mirando en las diferentes páginas es una tarea un poco tediosa.


7.3.12. Funcionalidad “Orden” en los listados

Los diferentes listados que nos proporciona el programa vienen ordenados por defecto, pero al usuario puede interesarle cambiar los parámetros de ordenación, y para ello se ha implementado la funcionalidad de “Orden”, que se compone de dos desplegables, uno para seleccionar el campo por el que queremos ordenar, y otro para decirle si el orden será ascendente o descendente:



Fig.85 Funcionalidad de orden

Primero seleccionamos el campo:



Listado Productos

Fecha: 19/9/2011
Usuario: MARIO

Orden:

▼

▼

Código	Descripción	Cantidad	Código EAN13
000000000001	BARRERA DOBLE ONDA 4318MM		0000000000017
000000000002	POSTE TUBULAR 1500MM		0000000000024
000000000003	SEPARADOR ESTANDAR IZQUIERDO	0 1	0000000000031
000000000004	SEPARADOR ESTANDAR DERECHO	0 1	0000000000048
000000000005	CONECTOR EN "U" PARA POSTE TUBULAR	0 1	0000000000055
000000000006	TORNILLO REDONDO CUELLO OVAL M16X30 5.8	0 1	0000000000062
000000000007	TORNILLO REDONDO CUELLO OVAL M16X40 5.8	0 1	0000000000079
000000000008	TORNILLO HEXAGONAL M16X35 5.8	0 1	0000000000086
000000000009	TORNILLO HEXAGONAL M10X40 DIN 933 6.8	0 1	0000000000093
000000000010	TUERCA HEXAGONAL M16 CLASE 5	0 1	0000000000109

Pag. 1 [siguiente](#) [>>](#)

Fig.86 Campo selección de columna para ordenar

Después seleccionamos el orden:



The screenshot shows a web interface titled "Listado Productos". At the top right, it displays the date "Fecha: 19/9/2011" and the user "Usuario: MARIO". Below the title, there is a search bar with a "Buscar" button and a sorting section with a dropdown menu set to "Código" and an "Ordenar" button. A second dropdown menu is open, showing "ASC" and "DESC" options. Below this is a table with columns: "Código", "Descripción", "Lote", "Cantidad", and "EAN13". The table contains 10 rows of product data. At the bottom of the table, there are navigation links: "Pag. 1", "[siguiente](#)", and ">>". Below the table are two buttons: "Volver" and "Abrir".

Código	Descripción	Lote	Cantidad	EAN13
000000000001	BARRERA DOBLE ONDA 4318MM	0	1	0000000000017
000000000002	POSTE TUBULAR 1500MM	0	1	0000000000024
000000000003	SEPARADOR ESTANDAR IZQUIERDO	0	1	0000000000031
000000000004	SEPARADOR ESTANDAR DERECHO	0	1	0000000000048
000000000005	CONECTOR EN "U" PARA POSTE TUBULAR	0	1	0000000000055
000000000006	TORNILLO REDONDO CUELLO OVAL M16X30 5.8	0	1	0000000000062
000000000007	TORNILLO REDONDO CUELLO OVAL M16X40 5.8	0	1	0000000000079
000000000008	TORNILLO HEXAGONAL M16X35 5.8	0	1	0000000000086
000000000009	TORNILLO HEXAGONAL M10X40 DIN 933 6.8	0	1	0000000000093
000000000010	TUERCA HEXAGONAL M16 CLASE 5	0	1	0000000000109

Fig.87 Selección de orden ascendente o descendente

Una vez seleccionados los criterios de ordenación, solo nos quedaría pulsar el botón "Ordenar" y tendríamos la ordenación del listado adaptada a nuestras necesidades.

8. Bibliografía

Para la ayuda en el análisis, desarrollo y documentación de este proyecto se han utilizado:

- Sommerville, I. "Ingeniería del software" (6ª Edición) Addison-Wesley; 2002.
- Eclipse Site. Pagina Oficial Eclipse Helios [Acceso: 21-09-2011]. Disponible en: <http://www.eclipse.org/helios/>
- Open Source Software Engineering Tools. Plugin Subclipse para control de versiones Eclipse [Acceso: 21-09-2011]. Disponible en: <http://subclipse.tigris.org/>
- The Apache Software Foundation. Pagina Oficial Apache Tomcat [Acceso: 21-09-2011]. Disponible en: <http://tomcat.apache.org/>
- MySQL.COM. Pagina Oficial Open Source Database MySQL. [Acceso: 21-09-2011]. Disponible en: <http://www.mysql.com/>
- MySQL.COM. MySQL GUI Tools. [Acceso: 21-09-2011]. Disponible en: <http://dev.mysql.com/downloads/gui-tools/5.0.html>
- JasperReports. Open Source Reporting Engine. [Acceso: 21-09-2011]. Disponible en: <http://jasperforge.org/projects/jasperreports>
- W3Schools.com. Web Development Site. [Acceso: 21-09-2011]. Disponible en: <http://www.w3schools.com/>
- Javahispano.com. Foro de Java en Castellano. [Acceso: 21-09-2011]. Disponible en: <http://www.javahispano.org/forum/>
- Forosdelweb.com. Foro de programación en Castellano. [Acceso: 21-09-2011]. Disponible en: <http://www.javahispano.org/forum/>
- Lawebdelprogramador.com. Foro de Programación en Castellano. [Acceso: 21-09-2011]. Disponible en: <http://www.lawebdelprogramador.com/foros/Java/index1.html>

- Foro Chuidang. Foro de programación en Castellano. [Acceso: 21-09-2011].
Disponible en: <http://foro.chuidiang.com/>

-Java.com. Pagina Oficial Descargas Java de Oracle. [Acceso: 21-09-2011].
Disponible en: <http://java.com/es/download/>

- Manual MySQL. DEV MySQL 5.0 Reference Manual. [Acceso: 21-09-2011].
Disponible en: <http://dev.mysql.com/doc/refman/5.0/es/index.html>

-Adictosaltrabajo.com. Mi primera web con Struts. [Actualizado: 31-07-2006;
Acceso: 21-09-2011]. Disponible en:
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=miPrimeraWebStruts>

-Struts. Definicion Apache Struts. [Acceso: 21-09-2011]. Disponible en:
http://es.wikipedia.org/wiki/Apache_Struts

-Modelo Vista Controlador (MVC). Presentación del modelo [Acceso:
21-09-2011]. Disponible en: http://es.wikipedia.org/wiki/Modelo_Vista_Controlador