



**Universitat Autònoma
de Barcelona**

Gestextil

**Sistema de planificación de recursos
empresariales (*ERP*)**

Memoria del proyecto
de Ingeniería Técnica en
Informática de Gestión
realizado por
Mireia González Jorge
i dirigido por
Marc Talló Sendra

Escola Universitària d'Informàtica
Sabadell, Junio de 2009

Prefacio

En la actualidad, el mercado actual y las nuevas tecnologías hacen que las empresas sean más competitivas entre ellas. Frente a esto, las empresas que quieren sobrevivir en el mercado y competir con un mínimo de garantía de éxito deberían de evaluar la mejor manera de trabajar con la información. Con esto nos referimos a disponer de un sistema de información (SI)¹ eficiente que permita la obtención rápida y precisa de los datos. Este puede ser un punto vital que puede ayudar significativamente en la productividad de una empresa, y así, satisfacer objetivos.

El presente proyecto pretende realizar el estudio y desarrollo de un sistema de estas características con la intención de sustituir parcialmente el trabajo de terceros, reduciendo significativamente los costes económicos por parte de la empresa.

Por otra parte, el desarrollo del proyecto se basará en la aplicación de los conceptos adquiridos durante esta carrera, que desde mi punto de vista considero sólo una base necesaria y útil para desenvolverse en el mundo real. Gracias a ello, esto me permite el aprendizaje de nuevos conceptos a partir de la base adquirida. Es por eso que el desarrollo de este sistema no lo he querido centrar solamente en un caso práctico, sino también en aportar un valor añadido al proyecto aprendiendo tecnologías desconocidas por mí hasta la actualidad.

¹ Un sistema de información (SI) es un conjunto organizado de elementos. Pueden ser de 4 tipos diferentes: personas, datos, actividades o recursos materiales en general.

Contenido

Introducción.....	1
Presentación	2
Objetivos	2
Estado del arte.....	2
Motivación	3
Conceptos básicos	3
Estructura de la memoria.....	4
Connotación usada.....	5
Estudio de viabilidad	7
Introducción	8
Situación a tratar.....	8
Perfil del usuario	10
Objetivos.....	10
Descripción del sistema	11
Descripción.....	11
Recursos	13
Evaluación de riesgos	14
Organización del proyecto	14
Requisitos del sistema.....	15
Requisitos funcionales	15
Requisitos no funcionales	16
Requisitos técnicos	16
Viabilidad legal	16
<i>Protección de datos</i>	<i>16</i>
<i>Propiedad intelectual.....</i>	<i>16</i>
Estudio de alternativas	17
Planificación del proyecto.....	18
Planificación de las etapas de desarrollo proyecto.....	18
Calendario laboral	19
Horario laboral	20
Recursos software	21
Entrega de documentación.....	21
Análisis coste-beneficio.....	22
Coste de las tareas	22
Coste de los recursos software.....	23
Presupuesto	24
Fundamentos teóricos.....	25
Integración de tecnologías	26

PostgreSQL	27
Hibernate	28
<i>¿Qué es Hibernate?</i>	28
<i>¿Por qué usar Hibernate?</i>	29
Framework Struts	30
<i>Funcionamiento de Struts</i>	31
Tags Struts-EL y JSTL.....	33
Tag Display 1.1	34
Struts Layout.....	34
Librería iText	35
Cewolf tag library	35
Análisis del sistema	37
Casos de uso.....	38
Inicio de sesión	38
Gestión de usuarios	39
Configuración del sistema	42
Módulos de datos maestros	43
<i>Gestión de proveedores</i>	43
<i>Gestión de clientes</i>	46
<i>Gestión de artículos</i>	49
<i>Gestión de materiales</i>	51
<i>Gestión de escandallos</i>	54
Módulo de gestión de aprovisionamientos	56
<i>Gestión de pedidos de compra</i>	56
<i>Gestión de recepción de mercancías</i>	58
Módulo de gestión de proyectos.....	63
<i>Gestión de proyectos</i>	63
<i>Gestión de pedidos de venta</i>	67
Modelo conceptual	69
Análisis de requerimientos	69
<i>Requisitos de datos</i>	69
Diseño.....	75
Diagramas de clases	76
Package ERP.....	76
Package data.....	76
<i>Package Empresa</i>	77
Diagrama de secuencia	78
Inicio de sesión	78
Interfaz gráfica de usuario	79
Implementación	87
Entorno de desarrollo.....	88
Estructura del directorio	88
Funcionamiento de Hibernate	90
Mapeo de objetos	94
<i>Identificador autoincrement</i>	94
<i>Herencia</i>	95
<i>Relaciones Muchos-a-muchos (Many-to-many)</i>	97
<i>Relaciones Muchos-a-uno (many-to-one)</i>	98
<i>Lenguaje HQL de Hibernate</i>	100

Display tag: funcionamiento	101
<i>Archivo de configuración de Struts</i>	101
Pruebas del sistema	103
Seguridad.....	104
Inyecciones en Hibernate	105
Prevenciones	105
<i>Parametrización de la consulta</i>	105
<i>Almacenamiento de procedures</i>	106
<i>Escape de los datos introducidos por el usuario</i>	106
StrutsTestCase.....	107
Test: inicio de sesión	107
Test: gestión de proyectos	108
Test: pedidos	108
Usabilidad	109
Conclusiones y trabajo futuro	111
Conclusiones	112
Desviaciones	112
Trabajo futuro	113
Bibliografía.....	115
Libros	116
Webs	116
Anexo	117
Diagrama de clases	118
Package Login.....	118
Package Datos Maestros	119
<i>Package Artículo</i>	119
<i>Package Material</i>	120
Package Aprovisionamientos	120
<i>Package Recepción de mercancías</i>	121
<i>Package Pedidos de compra</i>	122
Package Proyectos	123
<i>Package Proyectos</i>	123
<i>Package Pedidos de venta</i>	124
Package Estadísticas.....	125
Package Export	125
Manual de usuario	126

I

Introducción

- Presentación
- Objetivos
- Estado del arte
- Motivación
- Conceptos básicos
- Estructura de la memoria

Presentación

Actualmente en el ámbito empresarial se utiliza mucho el término de planificación de recursos empresariales, conocido por el acrónimo **ERP** (*Enterprise Resource Planning*). Este término no es otro que un sistema de información que integra y maneja muchos de los negocios relacionados con las operaciones de producción y distribución de una empresa. Principalmente, estos sistemas manejan el control de producción, logística, distribución, inventario, envíos, facturas e incluso la contabilidad.

Los ERP son considerados sistemas *back office* (trastienda) ya que el usuario y el cliente en general no están directamente relacionados, es decir, el sistema se centra en proporcionar unos beneficios directos al usuario y no al cliente.

Por otra parte, cabe destacar la diferencia entre sistemas de planificación de recursos empresariales y los utilizados para la gestión de las relaciones con los clientes, llamados **CRM** (*Customer Relationship Management*). Este último, se centra en una estrategia basada en la administración de las relaciones con el cliente y enfocados a sistemas de negocios electrónicos. Ocasionalmente, estos sistemas son llamados *front office*², por crear un lazo administrativo entre usuario y cliente.

Este proyecto se basa en la implementación de un sistema ERP. Para ello, se desarrollará la base principal del sistema, la cual es genérica para todas las empresas. A partir de aquí se realizará el estudio personalizado e individual de los módulos necesarios para una PYME³ que ofrece servicio en el sector del textil y se implementarán algunos de estos módulos acorde con las necesidades particulares.

Para su desarrollo se aplicará una arquitectura basada en tecnologías Java y un entorno de desarrollo que ofrezca soluciones apropiadas para este tipo de proyecto. Se tendrá en cuenta factores como el coste económico de implantación, plataforma necesaria e incluso el rendimiento del propio sistema.

Objetivos

Los objetivos principales de este proyecto se centran en tres puntos claves:

- ▶ Control de pedidos y proyectos que se llevan a cabo.
- ▶ Gestión de artículos y materiales.
- ▶ Información centralizada.

Con este sistema se pretende solucionar los principales problemas que conlleva un método de gestión no adecuado para este tipo de empresa.

Estado del arte

Actualmente, existen en el mercado varias soluciones que podrían utilizarse para este tipo de empresa, pero no todas resultan adecuadas. Básicamente, nos podemos decantar por dos tipos de sistemas.

- ▶ Sistemas de gestión
- ▶ Sistemas ERP

² En los negocios el término *front office* se refiere a la parte de ventas y marketing de una compañía. También se puede referir a otras secciones que proporcionan interacción con los clientes.

³ Acrónimo para denominar a las Pequeñas y Medianas Empresas.

Motivación

Actualmente existe la tendencia de implantar una interfaz Web en muchas aplicaciones. Quizás el motivo de esto es el auge que ha habido en el sector de las comunicaciones y las ventajas que nos proporciona esta tecnología. Gracias a esto, han ido apareciendo numerosas tecnologías Web que ofrecen una amplia variedad de soluciones. Uno de los motivos por los que he llevado a cabo este proyecto es precisamente este.

Partiendo desde ese punto, busqué la manera de aplicar estos conocimientos a un caso práctico real que como mínimo tuviera unos fundamentos reales. He tenido la oportunidad de enfocar mi proyecto en un sector concreto como es el textil y basarme en las necesidades específicas de una PYME.

Conceptos básicos

Muchas empresas están utilizando ERPs ya que se han dado cuenta que un sistema como este otorga muchas ventajas a nivel empresarial reduciendo costes y aumentando la rentabilidad de la misma. Se caracterizan por estar compuestos por diferentes partes que permanecen integradas en una única aplicación y con ellas se pueden llevar a cabo todos los procesos de negocio de una empresa. En muchas ocasiones estas soluciones pueden resultar complejas y difíciles de implantar ya que se necesita un desarrollo personalizado e individual para cada tipo de empresa, pero que gracias a su alta modularidad y adaptación esto es posible.

Las tres reglas básicas que deben cumplir estos sistemas son:

- Modularidad
- Adaptabilidad
- Integridad

A la hora de implantar un sistema ERP es necesario hacer un exhaustivo estudio para cada caso. Muchas veces una solución puede ser válida para más de una empresa pero casi siempre existe una personalización concreta que la diferencia de las otras. Es necesario evaluar las necesidades en cada caso, definir los resultados esperados con la implantación del sistema, plataforma tecnológica, estrategia de implantación, análisis del cambio organizativo así como de los controles de calidad y seguridad de la información en general.

Estructura de la memoria

La estructura básica de la memoria está dividida en cuatro grandes capítulos. A continuación se detallan cada uno de ellos.

..... **Capítulo 1: Estudio de viabilidad**

Este capítulo trata sobre el plan de viabilidad. Se realiza un estudio sobre las especificaciones tanto funcionales como técnicas, la planificación que se llevará a cabo para desarrollar el proyecto y los riesgos que conlleva.

..... **Capítulo 2: Fundamentos teóricos**

En este capítulo se habla sobre los fundamentos teóricos del proyecto. Se explican que tipo de tecnologías se han utilizado, cómo funcionan y el porqué de su elección. También se detallan las alternativas que se han tenido en cuenta para cada una de las decisiones tomadas.

..... **Capítulo 3: Análisis del sistema**

En este capítulo se explica la fase de análisis del sistema. Se detallan las necesidades específicas del sistema y cuales son las funciones que realiza. Se incluyen también los diagramas de casos de uso y el modelo conceptual de la base de datos. Se especifican también los requisitos de datos y que es lo que necesitamos almacenar.

..... **Capítulo 4: Diseño del sistema**

En este capítulo se detalla la fase de diseño del sistema. Se especifica la interfaz gráfica de usuario y el modelo lógico de la base de datos.

..... **Capítulo 5: Implementación**

En este capítulo se habla sobre la fase de implementación. En este apartado se explica como se han integrado las diferentes tecnologías y su uso en el sistema.

..... **Capítulo 6: Pruebas del sistema**

En este capítulo se habla sobre la fase de pruebas. Se detallan las pruebas que se han realizado, los resultados esperados y los reales.

..... **Capítulo 7: Conclusiones**

En este capítulo se habla sobre las conclusiones y las desviaciones del proyecto. También se detallan las mejoras y ampliaciones posibles del sistema.

..... **Capítulo 8: Bibliografía**

En este capítulo se incluye toda la bibliografía utilizada para la realización del proyecto.


Connotación usada

A lo largo de esta memoria se utilizan diferentes símbolos y connotaciones con la finalidad de aclarar términos técnicos o especificaciones concretas. A continuación se detallan los diferentes tipos que se pueden encontrar.

 Este símbolo se utiliza para aclarar alguna información que se considera importante.

¹ Este tipo de numeración hace referencia a la definición de algún término en pie de pagina.

código Este tipo de letra se utiliza para indicar un fragmento de código o en algunos casos, ejemplos.

 Este símbolo indica una cita de un libro. Normalmente usada para definir algún concepto concreto.

Para citar se utilizará un título dependiendo de los siguientes casos:

Diagramas y esquemas: *Figura X: Título de la figura*

Imágenes: *Imagen X: Título de la imagen*

Las palabras en MAYUSCULAS indican un acrónimo y vendrá seguido de la definición de sus iniciales en cursiva -tan solo la primera vez que aparezcan en el documento-.

Las palabras en *cursiva* indican una palabra en inglés que no ha sido traducida.

Las palabras importantes aparecen en **negrita**.

En el texto, también podemos encontrar palabras que resaltan por su tipografía como por ejemplo **esta**. Será utilizada para referirnos a un término técnico como una clase, entidad, tabla de la base de datos, etc.

1

1. Estudio de viabilidad

Este capítulo contiene

- Introducción
- Situación a tratar
- Objetivos
- Descripción del sistema
- Requisitos del sistema
- Planificación del proyecto
- Viabilidad legal

1.1. Introducción

Este proyecto se basa en el estudio e implementación de un sistema de planificación de recursos empresariales, conocido por el acrónimo ERP (*Enterprise Resource Planning*) que pretende ser una solución para una PYME del sector del textil. El sistema tiene como finalidad mejorar la gestión de los procesos que se llevan a cabo en esta empresa reduciendo tiempo, errores y costes innecesarios.

Las ventajas principales que presenta este sistema son los siguientes:

- Optimizar los procesos empresariales.
- Facilitar la toma de decisiones (ejecución más rápida).
- Mantener la información actualizada para un rápido acceso.
- Compartir información entre las diferentes secciones y componentes de la empresa para obtener una mayor eficiencia y automatización de los procesos (integridad de datos).
- Proporcionar una solución adaptada y específica a la necesidad de la empresa.
- Reducir costes y tiempo (lo que lleva a un aumento de la productividad).
- Mejorar el servicio con el cliente.

En cuanto a las desventajas hay que tener en cuenta los siguientes puntos:

- Coste de implantación.
- Formación previa para el uso del sistema.

1.2. Situación a tratar

La actividad que desempeña la empresa textil se basa principalmente en el diseño y venta de tejidos. Para ello un comercial se encarga de gestionar los pedidos que un cliente realiza y lo pone en conocimiento del diseñador. El diseñador se encarga de realizar el diseño del artículo pedido –en algunos casos no será necesario porque puede que ya haya sido diseñado anteriormente- y busca el mejor proveedor según criterios como: calidad, precio, tiempo de servicio o transporte. Posteriormente existirá un proceso de fabricación del tejido, que será llevado a cabo por uno o más proveedores, ya que las fases de fabricación del tejido pueden ser llevadas a cabo por proveedores diferentes.

Finalmente, se entrega el tejido diseñado y fabricado al cliente y se da por finalizado el proyecto. Cabe la posibilidad que no todo el tejido sea entregado al cliente y la parte sobrante se tenga que guardar en el almacén.

En la figura 1 se muestra un pequeño esquema de la operativa de la empresa.

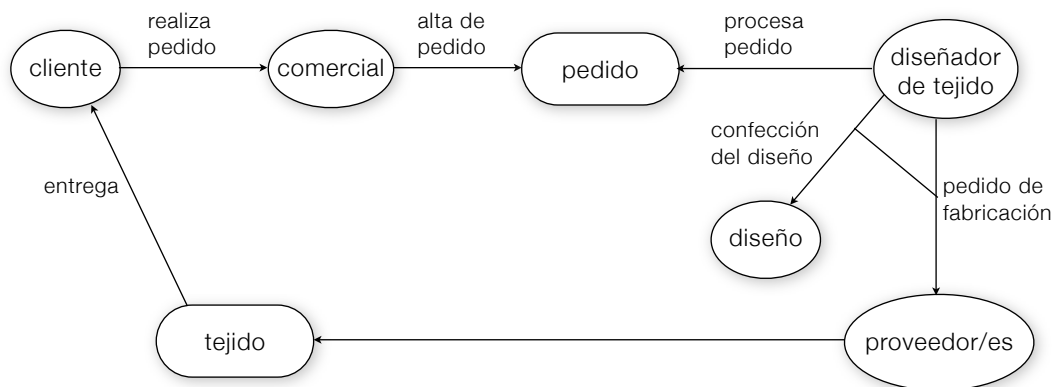


Figura 1: Operativa de la empresa

Frente a la necesidad de registrar y gestionar todos los elementos relacionados con la actividad de la empresa, se opta por la implantación de este sistema de planificación de recursos empresariales. Con este nuevo sistema se pretende sustituir el método actual de gestión basado en documentos Excel que resulta muy rudimentario y poco eficiente.

Para obtener una buena conexión entre los diferentes departamentos de la empresa, las funcionalidades del sistema se clasificarán por módulos. El diseño e implementación de un ERP desde cero es un proceso muy costoso, ya que desarrollar cada módulo es complejo. El alcance de este proyecto es mostrar la integración de algunos módulos que componen parte de un ERP.

Por este motivo, se han escogido para desarrollar los módulos más relevantes en la operativa de esta empresa: datos maestros, aprovisionamiento y gestión de proyectos. El resto de módulos que pudieran ser útiles se considerarían en una fase posterior de ampliación y mejoras.

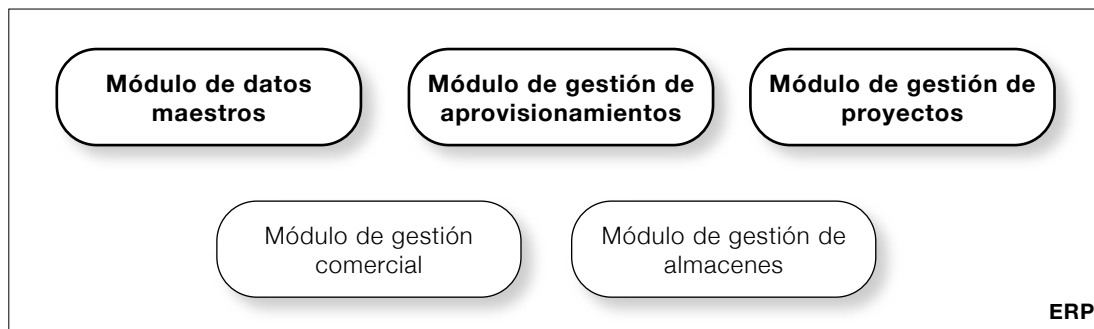


Figura 2: Módulos del sistema

1.3. Perfil del usuario

El uso del sistema está orientado principalmente a todas las personas que intervienen directamente en las funciones de la empresa –como es el caso del comercial y el diseñador de los tejidos- y a la persona encargada de administrar el sistema. Es por eso que se tiene en cuenta los siguientes tipos de usuarios.

Usuario Administrador

Este usuario es el que dispone de todos los privilegios y permisos para configurar el sistema. Entre otras funciones, podrá crear grupos de usuarios, asignar diferentes permisos –si se precisa- y dar de alta a nuevos usuarios.

Usuario normal

Este usuario dispone de unos permisos más restrictivos que el administrador. Podrá acceder a todas las funcionalidades que tengan que ver estrictamente con la actividad de la empresa pero no a aquellas que supongan un riesgo, por ejemplo, las copias de seguridad.

Para este tipo de usuario no se precisa un conocimiento específico en informática salvo lo principal para un usuario básico. La complejidad de uso reside en el aprendizaje de la aplicación sin necesidad de conocimientos extras.

1.4. Objetivos

Los objetivos los podemos clasificar según el módulo del sistema ya que aún habiendo relación entre ellos cada uno tiene una funcionalidad muy específica.

Módulo de datos maestros

- Consultar información referente a clientes, proveedores productos y materiales.
- Disponer de un catálogo de productos.
- Generación de informes de productos.
- Elaborar un producto a partir de otro disponible en el catálogo (ahorro de tiempo).
- Facilitar la consulta de datos generando listados a partir de búsquedas específicas.

Módulo de aprovisionamientos

- Disponer de un listado de pedidos de compra.
- Controlar la recepción de las mercancías y detectar errores de entrega.
- Controlar los retrasos de los pedidos y las devoluciones que se puedan producir.
- Obtener evaluaciones de un determinado proveedor.
- Generar estadísticas referentes a un proveedor.

Módulo de gestión de proyectos

- Realizar seguimiento de un proyecto.
- Consultar proyectos para obtener información útil sobre ellos.
- Control de los gastos asociados a un proyecto.
- Planificación de tareas para cada proyecto.
- Obtener informes sobre los proyectos.

1.5. Descripción del sistema

1.5.1. Descripción

El sistema permitirá llevar un control de los proyectos que se lleven a cabo realizando un exhaustivo seguimiento de cada fase de fabricación –desde que el cliente realiza un pedido hasta que éste es servido-. Por el momento, toda la contabilidad de la empresa se gestiona externamente por terceras personas y no se dispone de almacén propio, es por eso que no se incluyeron en el proyecto los módulos correspondientes a estas funciones.

Módulo de datos maestros

Este módulo consiste en la gestión de todos los datos principales de la empresa que intervienen en el resto de módulos de la aplicación. Se considera que estos datos son un pilar importante en la aplicación, ya que sin ellos el resto carece de sentido. Vamos a explicar la funcionalidad de cada uno de ellos.

Nombre	Descripción
Proveedores	Se dispondrá de todos los datos de los proveedores con los que la empresa trabaje. Para ello se guardarán los datos útiles que serán utilizados posteriormente para realizar pedidos de compra así como otras gestiones con los que sea necesario relacionarlos.
Clientes	Hace referencia a los datos de los clientes de la empresa. Esta información es necesaria para llevar a cabo cualquier proyecto.
Artículos	La empresa necesita una lista de los artículos que se han ido fabricando bajo petición de los clientes o por otros motivos -por ejemplo, prototipos-. En este caso los artículos de esta empresa son los tejidos.
Lista de materiales	La lista de materiales estará compuesta por aquellos materiales que la empresa necesita para la elaboración de artículos. En este caso, la materia primera que se utiliza son los hilos que se compran a diferentes proveedores.
Escandallos	Consiste en la planificación de los costes que conlleva la elaboración de un artículo. Cada artículo dispondrá de un escandallo - viene a ser como un presupuesto -. En él se especifica que proveedor realiza cada fase y el coste de ella. También se podrán anotar los gastos asociados como por ejemplo, el transporte.

Módulo de aprovisionamiento

El módulo de aprovisionamientos servirá para gestionar la logística de la empresa. Permitirá llevar un control de los pedidos de compra que se realicen para abastecerse de materias primas así como de la compra de servicios. Con esto nos referimos a la necesidad de que terceras personas fabriquen un determinado diseño de tejido que posteriormente utilizaremos para la elaboración de otros productos o para la venta final a un cliente.

Se llevará también un control de la recepción de las mercancías, teniendo en cuenta los retardos en las entregas y las posibles devoluciones que se produzcan. También se podrán realizar evaluaciones referentes a un determinado proveedor que nos servirá de utilidad como referencia a la hora de escoger uno u otro basándose en las experiencias anteriores.

Nombre	Descripción
Pedidos de compra	Permite llevar un control de los pedidos que la empresa realiza a los proveedores y mantener un registro sobre ello. Por el tipo de actividad de esta empresa los pedidos se basarán en la elaboración de tejidos a partir de un diseño proporcionado por ellos mismos o en algunos casos la compra de materia primera -hilos-.
Recepción de mercancías	Esta funcionalidad permitirá llevar un control de las mercancías que se compran a los proveedores teniendo en cuenta posibles retrasos. Estas se basarán en la recepción de los pedidos anteriormente citados.
Devoluciones	Permite llevar un control de las devoluciones que se produzcan con una determinada mercancía. Se llevará también el control de los gastos que que pueda acarrear.
Evaluación de proveedores	Permite evaluar a un proveedor con el que hemos tratado cualificándolo y permitiendo añadir anotaciones que pueden ser consultadas en un futuro.

Módulo de gestión de proyectos

Éste módulo nos permite gestionar todas las fases de las que se compone un proyecto. La empresa necesita registrar todos los datos referentes a un pedido de venta que un cliente realiza.

Cuando un cliente hace un pedido este se basa principalmente en la elaboración de un determinado tejido. La empresa deberá registrar este pedido de venta y podrá estimar el coste de éste a partir del escandallo del artículo. Una vez que el pedido se ha realizado se pondrá en marcha un proyecto. Éste tendrá unas fases que el sistema facilitará su seguimiento y se llevará también el control de los gastos asociados a dicho proyecto -gastos que no se tuvieron en cuenta en el escandallo-.

De esta manera se podrá acceder a un proyecto en concreto y saber rápidamente en que fase se encuentra, si ha sido entregado o si por algún motivo éste se ha cancelado o retrasado.

Nombre	Descripción
Proyectos	Permite registrar un proyecto y llevar el control de todos los elementos relacionados con él.
Pedidos de venta	Permite registrar los pedidos de venta que realizan los clientes.

1.5.2. Recursos

En este apartado se mostrarán los recursos necesarios para desarrollar el sistema y la implantación del mismo en la empresa.

Ordenador cliente

- Sistema operativo: Indiferente.
- Navegador de Internet: Mozilla Firefox o Internet Explorer.

Servidor

- Base de datos: Postgre (uso de la herramienta Hibernate)
- Sistema operativo: Indiferente.
- Plataforma de Internet: Tomcat.
- Entorno de programación.
- J2EE (Java Enterprise Edition), JSP (Java Server Pages), Framework Struts.
- IDE Eclipse.

Generación de documentación

- Pages
- Omniplan 1.5

Hardware

recursos mínimos del cliente

- Disponer de tarjeta de red
- Teclado, ratón
- Monitor

recursos mínimos del servidor

- Memoria RAM: 4GB mínimo
- RAID 1: 2HD de 100 GB
- DVD-ROM

Recursos humanos

En este caso, por tratarse de un proyecto de carácter académico el único recurso humano es el propio estudiante. Este realizará las funciones de analista, diseñador, programador y equipo de pruebas.

1.5.3. Evaluación de riesgos

Teniendo en cuenta el tipo de sistema y los factores internos y externos a la empresa hay que tener en cuenta los siguientes riesgos:

- Incompatibilidad entre navegadores de Internet.
- Problemas en el servidor o red local.
- Medidas de seguridad:
- Diferentes roles en la base de datos para delimitar permisos.
- Identificación en el sistema mediante nombre de usuario y contraseña.
- Encriptación de la comunicación HTTP mediante el protocolo SSL (HTTPS).

1.5.4. Organización del proyecto

Inicialmente, es conveniente realizar un estudio de la situación actual de la empresa y los problemas que conlleva el sistema actual. Para ello, se planificará un seguido de entrevistas con el personal.

Para una primera visión general será necesario realizar una entrevista con el responsable de la empresa. Una vez iniciada la fase de captura de requisitos, se entrevistará al personal adecuado, lo que permitirá obtener una visión más detallada de las funciones que se llevan a cabo.

En una segunda fase se realizará un prototipo inicial del sistema. Este punto se considera de gran importancia teniendo en cuenta que anteriormente no se utilizaba ningún sistema informático. Un prototipo puede resultar realmente útil para captar mejor las necesidades y obtener un sistema que se adapte totalmente a los objetivos marcados.

A partir de aquí se continuarán con las diferentes fases de diseño e implementación del sistema. A medida que se vaya avanzando en el desarrollo se realizará otro prototipo evolucionado para evaluar de nuevo las funciones de cada módulo, siempre intentando que se ajuste a las necesidades reales.

Finalmente, se realizarán las pruebas antes y después de la implantación del sistema en la empresa y se entregará toda la documentación y manual de usuario.

1.6. Requisitos del sistema

1.6.1. Requisitos funcionales

Enumeramos los requisitos funcionales según las funciones principales del sistema.

Función: Altas en el sistema.

- El sistema deberá registrar los datos introducidos por el usuario.
- El sistema deberá validar los datos introducidos por el usuario.

Función: Modificaciones en el sistema.

- El sistema deberá permitir modificar los datos teniendo en cuenta la persistencia.
- El sistema deberá validar los datos modificados por el usuario.
- Función: Bajas en el sistema.
- El sistema deberá permitir eliminar datos teniendo en cuenta la persistencia.
- El sistema deberá tener en cuenta si el dato a eliminar no afecta a otros elementos del sistema. De ser así, deberá informar al usuario.
- El sistema deberá permitir asignar a un dato el estado "inactivo" en caso de no poder ser eliminado.

Función: Consultas en el sistema.

- El sistema deberá mostrar un formulario de búsqueda.
- El sistema deberá realizar listados a partir de parámetros de búsqueda.
- El sistema deberá poder listar datos usando paginación.
- El sistema deberá permitir ordenar los listados.
- El sistema deberá permitir exportar los listados en varios formatos.
- El sistema deberá informar al usuario si no se encuentran resultados para un listado.

Función: Creación, seguimiento y control de proyectos.

- El sistema deberá permitir realizar un proyecto a partir de un presupuesto aceptado.
- El sistema deberá registrar los gastos asociados a un proyecto.
- El sistema deberá permitir crear tareas específicas para cada fase del proyecto.
- El sistema deberá permitir el seguimiento de las fases anotando también los imprevistos que surjan -retrasos, costes extras, devoluciones, etc.-.
- El sistema deberá permitir obtener listados de proyectos según búsquedas acotadas.
- El sistema deberá permitir generar informes sobre proyectos detallando toda la información sobre el.

1.6.2. Requisitos no funcionales

En cuanto a los requisitos funcionales del sistema tenemos los siguientes:

- El sistema deberá visualizarse y funcionar correctamente como mínimo en los navegadores Internet Explorer y Mozilla Firefox.
- El sistema deberá cumplir las normas recogidas en la Ley Orgánica de Datos Personales (LODP).
- El sistema deberá tener medidas de seguridad.
- El sistema deberá tener un tiempo de respuesta aceptable (no superior a 5 seg.)

1.7. Requisitos técnicos

1.7.1. Viabilidad legal

En este apartado se estudiará la viabilidad legal de nuestro sistema. Existe la posibilidad de sufrir riesgos legales que nuestro sistema pueda albergar o provocar por falta de seguridad. En este caso, hay que tener cuidado con la información que se maneja o los derechos de autor entre otras cosas.

1.7.1.1. Protección de datos

Según el marco legal, la ley orgánica de protección de datos (LOPD) 15/1999 tiene como objetivo proteger los datos de carácter personal registrados en cualquier soporte que los haga susceptibles de tratar. Teniendo en cuenta el tipo de datos que nuestro sistema ERP almacena, podemos catalogarlos de nivel básico, ya que no almacenamos información de carácter crítico como pueden ser ideologías políticas, religión, salud o vida sexual –entre otras-.

Cuando registramos el nombre de una empresa (ya sea cliente o proveedor), se le debe permitir el derecho de acceso a sus datos, modificarlos e incluso cancelarlos. Toda la información es de carácter privado y debemos almacenarla de forma segura para que terceras persona no puedan acceder a ella. Queda también excluido el derecho de suministrar esta información sin consentimiento a otras personas. Se tomarán medidas técnicas y organizativas para asegurar la integridad de los datos, la disponibilidad y la confidencialidad.

1.7.1.2. Propiedad intelectual

La propiedad intelectual del proyecto desarrollado es del profesor o profesores que dirigen y coordinan el trabajo así como del estudiante que lo lleva a cabo. Ambos son los coautores del proyecto.

En cuanto a los derechos de explotación, estos corresponden en su totalidad a la universidad. Si se llevará a cabo esta explotación y se obtuvieran unos beneficios económicos, los autores del proyecto recibirán un 50% de los beneficios netos obtenidos.

1.8. Estudio de alternativas

Un sistema ERP se caracteriza principalmente por ser modular y adaptarse a cualquier tipo de empresa. Es por eso que se estudian alternativas que satisfazcan las mismas necesidades pero sin tratarse de un sistema ERP, ya que de lo contrario el estudio de alternativas carecería de sentido por lo mencionado anteriormente.

Productos comerciales

- Software Gestión Pymes: Permite alta de clientes y proveedores, control de stock, facturación, compras y pagos.
- Full Network: Dispone de módulos de gestión, contabilidad y CRM.

Evaluación de alternativas

- Software Gestión Pymes: Solución no viable ya que no se adapta a las necesidades específicas de la empresa en cuestión. La facturación es innecesaria – ya que por el momento la realizan terceras personas- y no dispone de las herramientas necesarias para llevar a cabo un proyecto y relacionar datos con el resto del sistema.
- Full Network: Solución no viable ya que no permite llevar una gestión de proyectos que se adapte a las características del tipo de proyectos que se realiza en la empresa.

1.9. Planificación del proyecto

1.9.1. Planificación de las etapas de desarrollo proyecto

La metodología que se llevará a cabo para el desarrollo de este proyecto será siguiendo el modelo lineal. La estructura estará dividida en módulos y se diseñará por etapas.

En la siguiente figura se muestra la planificación que se llevará a cabo y las dependencias entre tareas.

Tarea	Duración	Inicio	Fin	Dependencias
▼ 1) Proyecto ERP	80d 1h 26,25m	20/11/08	21/05/09	
▼ 1.1) Análisis de requerimientos	16d 2h 15m	20/11/08	19/12/08	
• 1.1.1) Estudio de viabilidad	3d 5h	20/11/08	25/11/08	
• 1.1.2) Entrevistas con el personal	5h	20/11/08	20/11/08	
• 1.1.3) Especificación UML	4d 5h	20/11/08	28/11/08	1.1.2
• 1.1.4) Prototipo	11d 15m	28/11/08	19/12/08	1.1.3
▼ 1.2) Diseño	23d 5h 41,25m	20/11/08	09/01/09	
• 1.2.1) Diseño de la base de datos	4d	20/11/08	27/11/08	
▼ 1.2.2) Diseño del sistema	15d 4h 56,25m	04/12/08	09/01/09	
• 1.2.2.1) Módulo de gestión de datos	5d	04/12/08	12/12/08	
• 1.2.2.2) Módulo de aprovisionamientos	5d	11/12/08	19/12/08	
• 1.2.2.3) Módulo de gestión de...	6d 5h	19/12/08	09/01/09	
▼ 1.3) Implementación	49d 4h 41,25m	04/12/08	30/03/09	
• 1.3.1) Desarrollo de la base de datos	2d	04/12/08	08/12/08	1.2.1
• 1.3.2) Módulo de gestión de datos	10d	08/01/09	12/02/09	1.2.2.1
• 1.3.3) Módulo de aprovisionamientos	10d	12/02/09	03/03/09	1.2.2.2
• 1.3.4) Módulo de gestión de proyectos	15d	03/03/09	30/03/09	1.2.2.3
▼ 1.4) Pruebas	12d	30/03/09	30/04/09	
▼ 1.4.1) Pruebas locales del sistema	6d	30/03/09	20/04/09	
• 1.4.1.1) Módulo de gestión de datos	2d	30/03/09	02/04/09	1.3.2
• 1.4.1.2) Módulo de aprovisionamientos	2d	02/04/09	14/04/09	1.3.3
• 1.4.1.3) Módulo de gestión de...	2d	14/04/09	20/04/09	1.3.4
• 1.4.2) Implantación del sistema	1d	20/04/09	21/04/09	1.4.1
• 1.4.3) Pruebas finales y corrección de...	5d	21/04/09	30/04/09	1.4.2
• 1.5) Documentación	15d	21/04/09	21/05/09	

Imagen 1: Planificación de las tareas

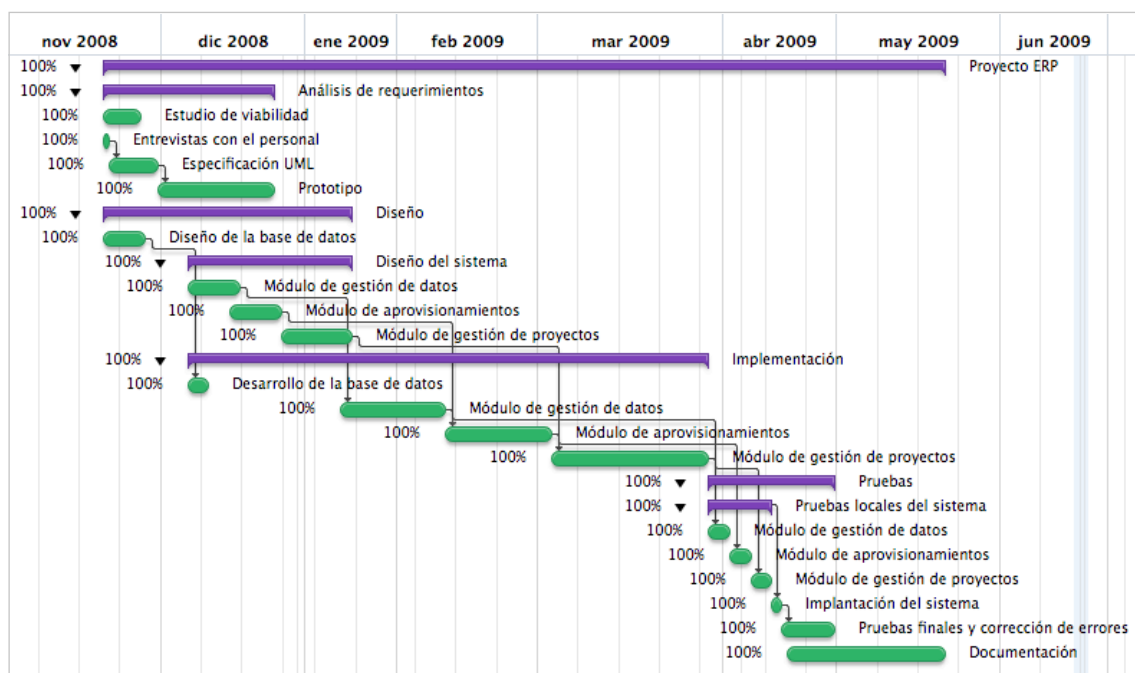


Imagen 2: Diagrama de Gantt

1.9.2. Calendario laboral

En la planificación de las tareas se ha tenido en cuenta los días de fiestas y los no laborales por varios motivos, entre ellos las semanas de exámenes.

Noviembre 2008						
L	M	MX	J	V	S	D
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Diciembre 2008						
L	M	MX	J	V	S	D
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Enero 2009						
L	M	MX	J	V	S	D
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Febrero 2009						
L	M	MX	J	V	S	D
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	

Marzo 2009						
L	M	MX	J	V	S	D
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Abril 2009						
L	M	MX	J	V	S	D
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Mayo 2009						
L	M	MX	J	V	S	D
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Junio 2009						
L	M	MX	J	V	S	D
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Los días recuadrados en rojo no se trabaja.

1.9.3. Horario laboral

El horario laboral es el que se muestra a continuación, el cual se adapta de la mejor manera a la disponibilidad. Por regla general ese será el horario para desarrollar el proyecto salvo que no exista algún contratiempo puntual.

Semana laboral normal de Proyecto							
	lunes	martes	miércoles	jueves	viernes	sábado	domingo
2							
4							
6							
8							
10		martes 8:00- 13:00			viernes 8:00- 13:00		
12							
14							
16	lunes - martes 15:00 - 21:00			jueves 15:00- 21:00	viernes 15:00- 18:00		
18							
20							
22							

Imagen 3: Horario laboral

1.10. Recursos software

Dependiendo de las tareas se hará uso de un determinado software. Para cada una de ellas se ha estudiado la mejor opción en cuanto a coste y características.

Tarea	Software
Análisis de requerimientos	
Estudio de viabilidad	OmniPlan, Eclipse
Especificación UML	OmniGraffle Profesional
Prototipo	Eclipse
Diseño	
Diseño de la base de datos	OmniGraffle Profesional
Diseño del sistema	OmniGraffle Profesional
Implementación	
Desarrollo de la base de datos	Postgre
Módulos del sistema	Eclipse
Pruebas	
Pruebas locales del sistema	StrutsTestCase for JUnit
Implantación del sistema	Linux, Tomcat
Documentación	
	Pages

1.11. Entrega de documentación

La documentación a entregar se basa en los documentos de especificación y diseño del sistema. Se ha utilizado como lenguaje de especificación UML (*Unified Modeling Language*). Estos se harán entrega en formato digital y en papel.

A continuación se detalla de la composición de cada uno de ellos.

- Documento del análisis de requerimientos, UML
 - Casos de uso
- Documento del diseño del sistema, UML
 - Diagramas de clases
 - Diagramas de secuencia
- Manual de usuario del sistema ERP (formato digital).

1.12. Análisis coste-beneficio

En el siguiente apartado se realizará el análisis de costes que supone el desarrollo de este sistema. Se tendrá en cuenta los costes de los recursos empleados así como del hardware necesario para la implantación del sistema.

1.12.1. Coste de las tareas

Según la planificación de las tareas anteriormente vista se muestra el coste total de cada una de ellas teniendo en cuenta un coste por hora de 20€.

Tarea	Duración	Inicio	Fin	Coste total	Dependencias
▼ 1) Proyecto ERP	80d 1h 26,25m	20/11/08	21/05/09	21.256,25 €	
▼ 1.1) Análisis de requerimientos	16d 2h 15m	20/11/08	19/12/08	4.131,25 €	
● 1.1.1) Estudio de viabilidad	3d 5h	20/11/08	25/11/08	800,00 €	
● 1.1.2) Entrevistas con el personal	5h	20/11/08	20/11/08	125,00 €	
● 1.1.3) Especificación UML	4d 5h	20/11/08	28/11/08	1.000,00 €	1.1.2
● 1.1.4) Prototipo	11d 15m	28/11/08	19/12/08	2.206,25 €	1.1.3
▼ 1.2) Diseño	23d 5h 41,25m	20/11/08	09/01/09	4.325,00 €	
● 1.2.1) Diseño de la base de datos	4d	20/11/08	27/11/08	875,00 €	
▼ 1.2.2) Diseño del sistema	15d 4h 56,25m	04/12/08	09/01/09	3.450,00 €	
● 1.2.2.1) Módulo de gestión de datos	5d	04/12/08	12/12/08	1.000,00 €	
● 1.2.2.2) Módulo de aprovisionamientos	5d	11/12/08	19/12/08	1.000,00 €	
● 1.2.2.3) Módulo de gestión de proyectos	6d 5h	19/12/08	09/01/09	1.450,00 €	
▼ 1.3) Implementación	49d 4h 41,25m	04/12/08	30/03/09	7.400,00 €	
● 1.3.1) Desarrollo de la base de datos	2d	04/12/08	08/12/08	400,00 €	1.2.1
● 1.3.2) Módulo de gestión de datos	10d	08/01/09	12/02/09	2.000,00 €	1.2.2.1
● 1.3.3) Módulo de aprovisionamientos	10d	12/02/09	03/03/09	2.000,00 €	1.2.2.2
● 1.3.4) Módulo de gestión de proyectos	15d	03/03/09	30/03/09	3.000,00 €	1.2.2.3
▼ 1.4) Pruebas	12d	30/03/09	30/04/09	2.400,00 €	
▼ 1.4.1) Pruebas locales del sistema	6d	30/03/09	20/04/09	1.200,00 €	
● 1.4.1.1) Módulo de gestión de datos	2d	30/03/09	02/04/09	400,00 €	1.3.2
● 1.4.1.2) Módulo de aprovisionamientos	2d	02/04/09	14/04/09	400,00 €	1.3.3
● 1.4.1.3) Módulo de gestión de proyectos	2d	14/04/09	20/04/09	400,00 €	1.3.4
● 1.4.2) Implantación del sistema	1d	20/04/09	21/04/09	200,00 €	1.4.1
● 1.4.3) Pruebas finales y corrección de errores	5d	21/04/09	30/04/09	1.000,00 €	1.4.2
● 1.5) Documentación	15d	21/04/09	21/05/09	3.000,00 €	

Imagen 4: Coste del proyecto desglosado por tareas

1.12.2. Coste de los recursos software

A continuación se detalla todo el software necesario para llevar acabo el proyecto. En este se incluye también el sistema operativo así como otros programas que se amortizan anualmente.

Descripción	Coste en €
Mac OS X Leopard	129,00
OmniGraffle Profesional	396
OmniPlan 1.5	119
Tomcat 6.0.18	0
Postgre	0
Eclipse	0
Pages	79
Total	723

Este software se amortiza anualmente durante 5 años con una cuota anual de 128,8€.

$$644,00 / 5 = 128,8 \text{ €}$$

1.12.3. Presupuesto

En el presupuesto se incluyen los costes del sistema hardware necesario para la implantación del ERP así como la configuración e instalación. Se incluye también el propio sistema en si y la base de datos.

	Descripción	Coste en €
1	Dell Precision™ T7400 Esencial Procesador Quad Core Intel® Xeon® Processor E5405 (2.00GHz,2X6M L2,1333). 4GB, DDR2 SDRAM FBD Memory. 16X DVD+/-RW. RAID-2: 500GB , 80GB SATA.	2970,00
2	Instalación del servidor al cliente Instalar distribución Linux RedHat Configurar Tomcat y Samba Instalar el sistema ERP	363
3	Aplicación Sistema de planificación de recursos empresariales (ERP) Módulo de datos maestros Módulo de aprovisionamientos Módulo de gestión de proyectos Base de datos	38.217
	Total del presupuesto	41.550

1 Precio coste * beneficio * margen de negociación cliente:

$$1.800 * 1,5 * 1,1$$

2 Precio coste * desviación * Beneficio * margen de negociación:

$$200 * 1,1 * 1,5 * 1,1$$

3 Precio coste * desviación * Beneficio * margen de negociación:

$$(21.256,25 - 200) * 1,1 * 1,5 * 1,1$$

El pago se realizará en dos términos: un 40% al inicio del proyecto y el 60% restante después de la entrega de documentación e implantación del sistema en la empresa.

* Los precios no incluyen IVA.

2

2. Fundamentos teóricos

Este capítulo contiene

- Tecnologías usadas
 - PostgreSQL
 - Hibernate
 - *Framework* Struts
 - Librerías de tags
 - Librería iText

2.1. Integración de tecnologías

En este apartado vamos a hablar sobre las tecnologías que se han usado para la implementación del sistema.

- PostgreSQL
- Hibernate
- *Framework* Struts
- Tags Struts-EL y JSTL
- Display tag library
- Struts Layout
- Librería iText
- Cewolf tag library

Para ello vamos a explicar cada una de estas y el porqué de su elección. En la figura de abajo podemos ver un esquema de la arquitectura utilizada para el desarrollo del sistema.

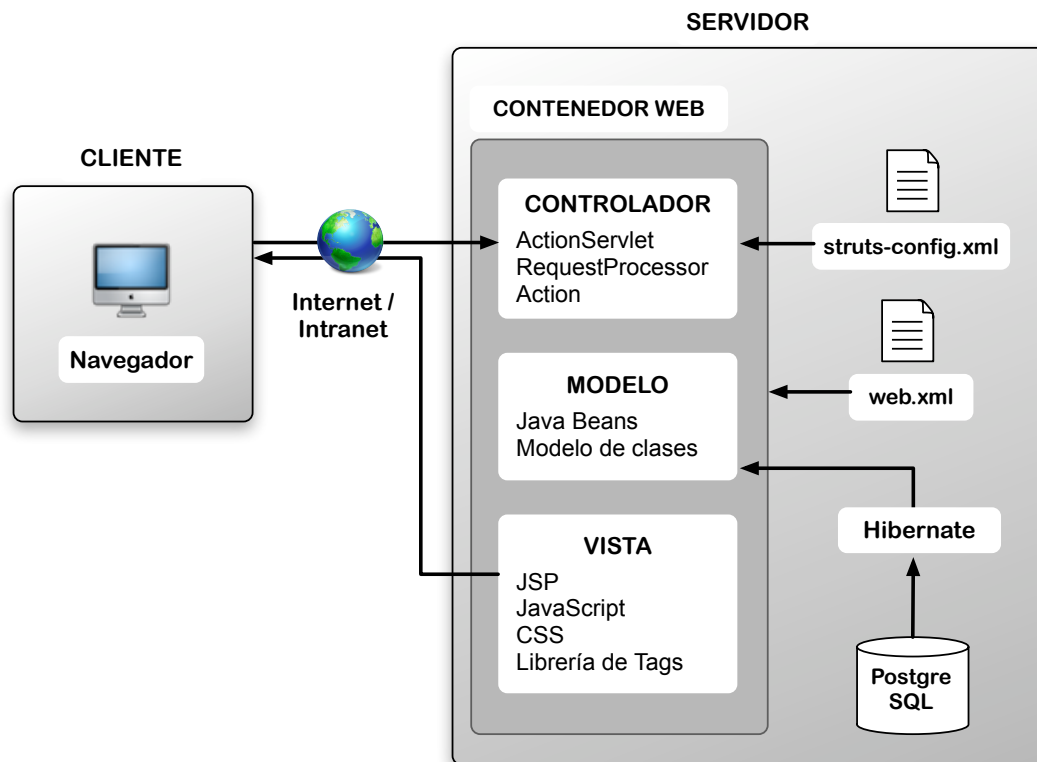


Figura 3: Esquema de la arquitectura utilizada

2.1.1. PostgreSQL

Escoger un sistema de base de datos es un proceso importante y hay que estudiar bien que opción es la mejor. En este caso he optado por usar PostgreSQL y me he basado principalmente en las características que ofrece éste y en las necesidades del propio sistema.



PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo licencia BSD.

[Fuente: Wikipedia]

En un principio se pensó en varios sistemas de base de datos en los que estudiar sus características. En la tabla de abajo podemos ver de forma esquemática los puntos que he considerado más importantes a tener en cuenta.

Soporte SO						
BBDD	Licencia	Mac OS	Linux	Herencia	Autoincrement	Múltiple PK
MySQL	BSD	✓	✓	✗	✓	✓
PostgreSQL	GPL o Propietario	✓	✓	✓	✓	✓
Oracle	Propietario	✗*	✓	✓	✗	✓
Microsoft SQL Server	Propietario	✗	✗	✓	✓	✓

Una de las características que condicionaba la elección de la base de datos es el sistema operativo. El proyecto está desarrollado utilizando Mac OS X, es por eso, que se descarto la utilización de Microsoft SQL Server y Oracle. A la hora de implantar el sistema hay que tener en cuenta que se hará uso de una distribución de Linux para el servidor y Windows para los ordenadores clientes.

*

❗ Hay que matizar en el soporte de Oracle para Mac OS X. En realidad no es cierto que éste no soporte este sistema operativo. Oracle dispone de un cliente para Mac OS X pero no de un servidor. En nuestro caso lo que nos importa es el servidor, por lo tanto se descarta la posibilidad de su uso.

Por otro lado, está el tema de la Licencia. Con el fin de reducir costes se ha optado por decantarse a sistemas de bajo coste o ninguno. Teniendo en cuenta las características comentadas nos quedamos con sólo dos opciones: MySQL y PostgreSQL.

A partir de aquí se debate que sistema es mejor teniendo en cuenta las funcionalidades que proporcionan. Una de las más importantes en este caso, es la posibilidad de realizar **herencia** en nuestro sistema de base de datos. Esta característica es de gran utilidad para la realización de tablas que derivan de otras - como la herencia en la relación de objetos -. Teniendo en cuenta esta necesidad de uso, MySQL se descarta por no permitir este tipo de asociación entre tablas.

Después de descartar los sistemas de base de datos que no se ajustan a las necesidades se opta por escoger PostgreSQL.

Otras características a destacar de PostgreSQL son las siguientes:

- Alta concurrencia
- Amplia variedad de tipos nativos
- Uso de disparadores (*Triggers*)



Un trigger en una base de datos, es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación de inserción, actualización o borrado.

[Fuente: Wikipedia]

2.1.2. Hibernate

La persistencia de nuestro sistema es una pieza fundamental. No solo es necesaria sino que tendremos que tener cuidado de cómo llevaremos a cabo el proceso de almacenamiento de datos –ya que una pérdida de información resulta inadmisible para cualquier empresa-.

Después de debatir sobre que tecnología sería la más adecuada para este tipo de sistema, se llegó a la conclusión de utilizar la herramienta Hibernate. Vamos a ver con detalle en que consiste y que ventajas nos aporta.

2.1.2.1. ¿Qué es Hibernate?

Podemos definir el siguiente término como:



Hibernate es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones.

[Fuente: Wikipedia]

Visto así puede resultar poco clara esta definición pero vallamos por pasos. Cuando utilizamos programación orientada a objetos y a su vez una base de datos relacional, nos damos cuenta de que estos son dos paradigmas diferentes. El modelo relacional se basa en relaciones y conjuntos, en cambio el paradigma orientado a objetos trabaja con objetos, atributos y las relaciones entre ellos. Puede parecer que esto no afecta a la hora de trabajar con ellos conjuntamente pero no es así. Cuando queremos tener una

persistencia de los objetos utilizando una base de datos relacional, nos damos cuenta que hay una desavenencia entre estos dos paradigmas conocida como **diferencia objeto-relacional**. Es aquí donde la herramienta Hibernate nos ayuda a solucionar este problema.

2.1.2.2. ¿Por qué usar Hibernate?

Como hemos visto antes existe la necesidad de solventar el problema de la diferencia objeto-relacional. El hecho de usar esta herramienta de mapeo⁴ nos proporciona muchas ventajas que hacen que nuestra base de datos no se sobrecargue y agilice el tiempo de respuesta con el sistema.

Cuando queremos almacenar información en una base de datos como por ejemplo MySQL, normalmente realizamos una conexión, creamos una sentencia SQL y la lanzamos a la base de datos. Lo mismo haríamos a la hora de recuperar la información para tratarla. Este método es correcto pero no es eficiente para todos los casos. Si tenemos que cargar de la base de datos un objeto pequeño no hay ningún problema, pero si por el contrario el objeto a cargar es muy grande –por ejemplo que tenga muchos atributos– esta tarea no resulta tan sencilla. Por otra parte, tenemos que considerar la posibilidad de que un objeto también tenga asociaciones con otros objetos, entonces no sólo tendríamos que tener en cuenta la persistencia de este sino de todos con los que esté asociado. Una situación como esta conlleva a tener que guardar y cargar los datos de una manera tediosa y lenta que resulta poco eficiente.

⁴ El mapeo es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

2.1.3. Framework Struts

Para la implementación de la aplicación se ha utilizado como lenguaje de programación J2EE (*Java Enterprise Edition*). El motivo de su elección se fundamenta principalmente por las ventajas que nos ofrece este lenguaje frente a otros como PHP o ASP.

Se descarto la opción de desarrollar el sistema con el lenguaje ASP por condicionar la plataforma de desarrollo. Como he comentado anteriormente, el sistema operativo que se lleva a cabo para implementar el sistema es Mac OS X. Otros lenguajes como Java o PHP pueden ser utilizados en cualquier sistema operativo y eso es un punto a favor en su elección.

A continuación, se explica con detalle las principales diferencias entre Java y PHP que justifican en gran parte la elección tomada.

Búsqueda de errores

Un punto importante a la hora de desarrollar una aplicación es el control de errores y *warnings*. En este caso, PHP es interpretado y no compilado, lo que lleva a que la aplicación se interrumpa en tiempo de ejecución por problemas de sintaxis. En cambio, Java es interpretado y compilado, lo que permite solucionar problemas antes del *deploy*.

Includes

En Java, cuando realizamos un *import* de una clase, sólo se importan las clases que se necesitan y de manera casi automática. En PHP se realizan *includes* de páginas en general, lo cual hace que tengamos que tener cuidado con lo que importamos. El uso de éste es más susceptible de errores a la hora de manejarlo. En este sentido, Java facilita esta tarea.

Sintaxis

PHP utiliza una sintaxis demasiado sencilla, no es hasta la versión 5 de PHP que se puede utilizar orientación a objetos. En cambio, en Java la sintaxis es mucho más compleja y robusta. Permite aprovechar todas las ventajas que ofrece la orientación a objetos.

Entornos de desarrollo

En Java podemos utilizar entornos de desarrollo muy completos como Eclipse, JDeveloper, Netbeans, etc. Para el desarrollo en PHP también se puede utilizar Eclipse pero se tiende a utilizar herramientas más simple como PHPEditor o Notepad++.

Podemos acabar esta comparativa con un gráfico que muestra de manera resumida los puntos más importantes a tener en cuenta.

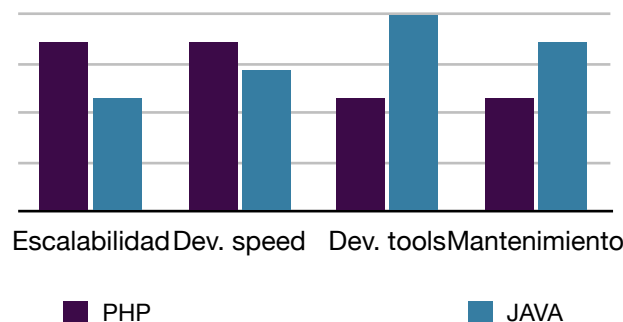


Figura 4: Comparativa de lenguajes

En vista de la necesidad de aplicar el patrón MVC (*Model View Controller*) y teniendo en cuenta la elección del lenguaje Java nos encontramos con dos posibilidades a la hora de utilizarlo. Una de ellas y quizás no tan óptima sería “manualmente”, mediante el uso de Java Beans para la parte del modelo, páginas JSP para la vista y Java Servlets para el controlador. Esta no es una opción a descartar pero existen herramientas que nos facilitan este trabajo permitiendo aplicar el patrón de una manera más fácil. Esta herramienta es el *framework Struts*.

2.1.3.1. Funcionamiento de Struts

Struts es un framework open-source que permite a los desarrolladores crear páginas Web utilizando la arquitectura MVC. Para ello, Struts proporciona tres componentes claves:

- Un manejador de peticiones
- Un manejador de respuestas que transfiera el control a otra fuente para completar la respuesta.
- Una librería de tags que ayuda a los desarrolladores a crear aplicaciones basadas en formularios interactivos con páginas servidor (*Server Pages*).

Vamos a detallar el esquema de la figura 5 para entender el funcionamiento de cada parte.

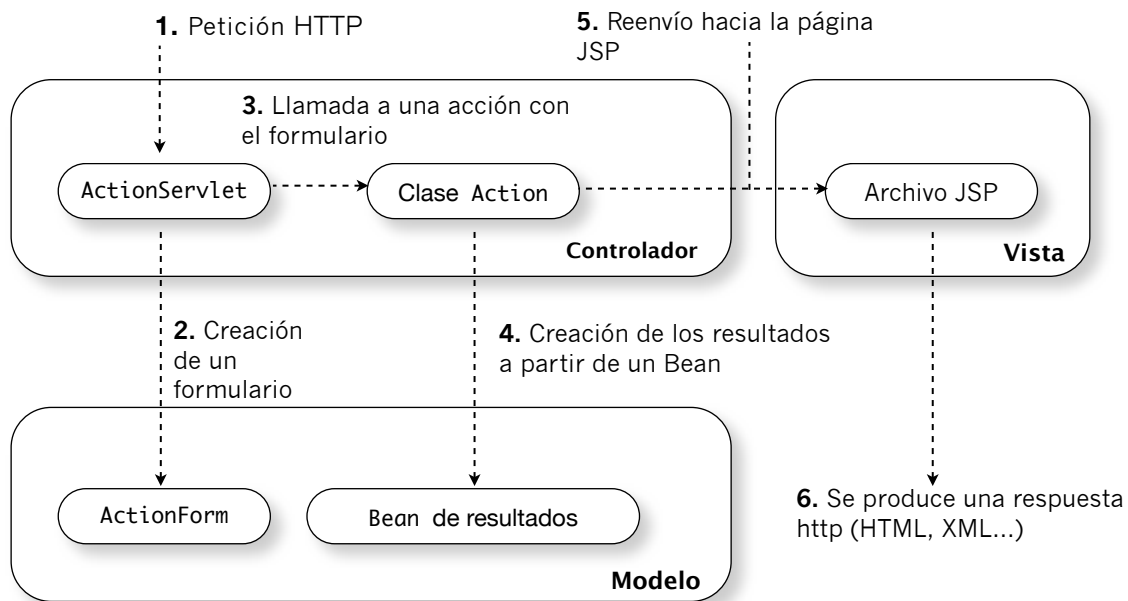


Figura 5: Funcionamiento del framework Struts

El controlador

La clase principal del **controlador** es un Java Servlet llamado **ActionServlet**. Esta clase es el manejador de todas las peticiones que un usuario realiza mediante el fichero administrador Struts de URLs. De esta manera, el **ActionServlet** utiliza la configuración de este fichero para pasar la información de la petición a una clase **ActionForm**, es decir, un *Bean* que tendrá la información que el usuario envió.

A continuación, el siguiente paso del controlador es delegar el control al manejador específico para ese tipo de petición. Este manejador siempre es una subclase de la clase **Action**. La función principal de la clase **Action** es la de mostrar la información de la petición del usuario –ahora residiendo en un **ActionForm**– y determina que acciones se necesitan tomar. Se llamará a la lógica de negocio de la parte del **modelo** para procesar la acción o enviará la petición a otra **vista**.

Después de que la acción haya sido tratada, la subclase **Action** escoge que vista es la adecuada para mostrarla al usuario como respuesta a su petición.

Vista

Para generar las vistas utilizamos páginas JSP (*Java Server Pages*) juntamente con librerías de *tags* como Struts (*Struts and Struts-EL*) y JSTL (*JSP Standard Tag Library*). La especificación de las páginas JSP proporciona la creación de HTML a partir de tags que extienden la funcionalidad de JSP.

Según las necesidades se han utilizado una serie de tags que proporcionan las librerías necesarias para realizar tareas de manera más sencilla. En el capítulo X se explican las tags que se han utilizado y su utilidad.

Modelo

No hay nada que dictamine a Struts como construir el **modelo** pero la mejor manera de llevar esto a cabo es encapsulando la información y las operaciones en *JavaBeans*. Para ello utilizaremos varias clases que nos permitirá manipular mejor los datos. De esta manera las operaciones que se realicen representaran la lógica de negocio de la aplicación.

El **modelo** se comunica con la **vista** siempre que los datos del modelo cambian y así poder mostrar los cambios en la vista para que el usuario pueda verlos.

2.1.4. Tags Struts-EL y JSTL

Las librerías Struts-EL y JSTL consisten en numerosas librerías separadas, cada una organizada acorde con una función. Struts-EL contiene las siguientes tags: Beans-EL, HTML-EL y Logic-EL.

La librería JSTL en cambio se basa en librerías de núcleo, formato, SQL y XML. Vamos a ver con más detalle las que se han utilizado en el sistema y su funcionalidad.

Bean-EL

Esta librería proporciona las tags necesarias para definir y usar los *Beans* disponibles en la página JSP. Un ejemplo de su uso es el siguiente:

```
<bean:write name="cliente" property="nombre" />
```

Con esta declaración mostramos en la página la propiedad “nombre” del *Bean* Cliente. De esta manera podemos mostrar la información almacenada en un *Bean* cuando la necesitamos. En algunos casos, necesitaremos mostrar información que no reside en ningún *Bean* sino en la *request* o la *session*. El funcionamiento sería el mismo, indicando en la propiedad **name** la variable a mostrar.

HTML-EL

Esta librería permite crear formularios Struts así como muchas de las tags de HTML utilizadas para generar la interfaz de usuario. Algunos ejemplos de su uso son los siguientes:

```
<html:text property="nombre" size="30" maxlength="30" />
```

Este es un ejemplo de declaración de un *textbox* en Struts. La principal diferencia con un *textbox* normal de HTML es la inclusión de propiedades como *property* que nos permite relacionar el componente con el resto del framework Struts. La utilidad principal de esta propiedad permite guardar en un *ActionForm* el valor de ese campo, llamado “nombre”. Para ello será necesario que la clase *ActionForm* –en este caso *ProveedorActionForm.java*– contenga el atributo “nombre” y los métodos *get()* y *set()* que permitirán coger los valores de este *textbox*. De esta manera se obtienen todos los datos de un formulario y permanecen almacenados en un *Bean*.

Un ejemplo de etiquetas de HTML es el siguiente, en el que se muestra un componente de tipo *submit* con el valor “Aceptar”.

```
<html:submit value="Aceptar" />
```

Logic-EL

Esta librería consiste en una serie de tags muy útiles para generar condicionalmente la salida de texto de página. Por ejemplo, es útil si queremos mostrar una lista de nombres que tenemos almacenados en un *Bean*.

```
<logic:iterate name="listaProveedorForm" Collection="listaProveedores"/>
```

FMT

Esta librería está incluida en el grupo de JSTL. Permite implementar soporte i18N para varios lenguajes y personalización. Su uso resulta muy práctico para llevar a cabo esa finalidad. En nuestra aplicación se utilizan varios archivos como por ejemplo *ApplicationResource.properties* que contienen todo el texto mostrado en la aplicación. Para mostrar este texto en el lugar correspondiente de la página se utilizan las tags *fmt*. Un ejemplo de su uso es el siguiente:

```
<fmt:message key="proveedor.nif"/>
```

La propiedad *key* sirve para introducir el nombre de la clave que se utilizó en el fichero *.properties* y se mostrará el texto que contenga. Para ver el ejemplo más claro, nos podemos fijar en la figura 6 que muestra un ejemplo del fichero *ApplicationResource.properties*.

```
proveedor.nif=NIF  
proveedor.nombre=Nombre  
proveedor.mensaje=El proveedor ha sido dado de alta correctamente.
```

Figura 6: Ejemplo del fichero *ApplicationResource.properties*.

De esta manera, cuando demos de alta a un proveedor y queramos el mostrar el mensaje de confirmación haremos lo siguiente:

```
<fmt:message key="proveedor.mensaje"/>
```

2.1.5. Tag Display 1.1

La librería de tags *Display* es una suite de tags personalizables *open source* que proporciona un alto nivel de patrones para la presentación de la Web, la cual trabaja con el patrón MVC. Esta nos permite mostrar una tabla a partir de los datos que nosotros le proporcionamos. Para ello gestiona la información de las columnas, permite ordenación, paginación e incluso exportación a otros formatos como PDF, CVS o EXCEL.

2.1.6. Struts Layout

Struts Layout es otra librería para Struts que proporciona la creación de una interfaz sencilla y rápida. A diferencia de la anterior, incluye muchos componentes -no solo la posibilidad de hacer tablas, aunque éstas proporcionan menos opciones-. Permite utilizar componentes para mostrar la información como paneles, campos de entrada, tablas, árboles, listas ordenables, datagrids, calendarios, etc.

2.1.7. Librería iText

iText es una librería *open-source* para crear y manipular archivos PDF, RTF y HTML en Java. Esta librería resulta realmente útil en combinación con la tecnología basada en Servlets de Java.

Teniendo en cuenta la necesidad de poder imprimir los informes y información del sistema -desde listados personalizados de clientes, proveedores, materiales, etc. hasta informes completos de proyectos realizados.- se a optado por una herramienta de estas características, la cual se integra muy bien con el resto de tecnologías utilizadas. En el capítulo 5 se detalla como funciona esta librería y como se usa.

2.1.8. Cewolf tag library

Cewolf es una librería de tags *open-source* que permite crear gráficos dentro dentro de aplicaciones web basadas en Servlets/JSP. Esta librería proporciona herramientas muy completas para la creación de muchos tipos de gráficos (barra, lineal, queso, etc.) con la que podemos definir varias propiedades como colores y leyenda.

Una de las ventajas de esta librería es la manera en la que se incrusta el gráfico en la página JSP. A diferencia de otras herramientas, Cewolf no necesita usar código JAVA dentro de la JSP, su uso es mediante tags XML.

Cewolf está basado en JFreeChart y usa su sistema de renderizado para renderizar la imagen final del gráfico dentro del flujo de respuesta del cliente. Los ficheros no son creados en el lado del servidor. Todo se basa en objetos de sesión y análisis de información dinámica. El funcionamiento consiste en un servlet que maneja el renderizado de los gráficos y una libreria de tags que traslada la definición del gráfico incluida en la JSP a una tag de imagen en HTML, la cual consulta el servlet para recuperar el gráfico apropiado. En la imagen de abajo podemos ver un ejemplo de gráfico de tipo queso.



Imagen 5: Ejemplo de gráfico

3

3. Análisis del sistema

Este capítulo contiene

- Especificación de requerimientos
 - Casos de uso
 - Modelo conceptual de la BBDD

3.1. Casos de uso

3.1.1. Inicio de sesión

Inicio de sesión

Descripción	Permite iniciar sesión a un usuario en el sistema permitiéndole el acceso.
Actores	Invitado.
Objetivos	Controlar el acceso a la aplicación y tener conocimiento de las acciones que realiza cada usuario. Permite seguridad en los datos según los permisos de cada usuario.
Precondiciones	Ninguna.
Flujo normal	<ol style="list-style-type: none">1. El invitado rellena el formulario de inicio de sesión.2. El sistema valida que los datos introducidos sean válidos.3. El invitado pasa a ser de invitado a usuario registrado.
Flujo alternativo	<ol style="list-style-type: none">1. Si los datos son incorrectos el sistema informará al invitado sobre ello.2. El sistema vuelve a comprobar los datos.
Poscondiciones	El sistema permitirá el acceso al usuario en el sistema.

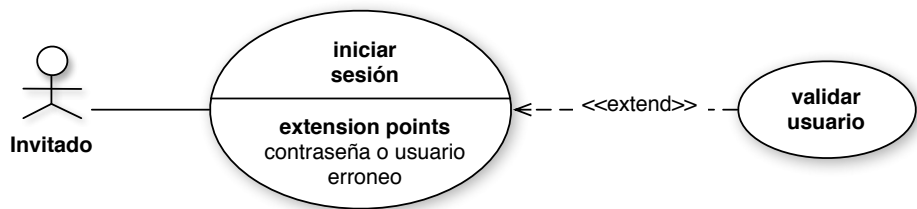


Figura 7: Inicio de sesión

3.1.2. Gestión de usuarios

Alta de usuarios

Descripción	Permite dar de alta en el sistema a nuevos usuarios.
Actores	Administrador.
Objetivos	Permitir el acceso a diferentes personas en el sistema.
Precondiciones	Haber iniciado sesión como Administrador.
Flujo normal	<ol style="list-style-type: none"> 1. El Administrador accede al panel de gestión de usuarios. 2. Selecciona la opción de dar de alta a un nuevo usuario. 3. Rellena los campos necesarios para su alta y le asigna un grupo. 4. El sistema comprobará que el usuario no exista previamente en el sistema.
Flujo alternativo	Si el usuario ya existe, el sistema informará al Administrador.
Poscondiciones	El sistema dará de alta el nuevo usuario y estará disponible para que pueda iniciar sesión.

Baja de usuarios

Descripción	Permite eliminar a un usuario en el sistema.
Actores	Administrador.
Objetivos	Poder eliminar un usuario que ya no debe tener acceso en el sistema.
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado sesión como Administrador. 2. El usuario debe existir. 3. El usuario no debe de haber realizado ninguna operación en el sistema (por ejemplo una compra o proyecto).
Flujo normal	<ol style="list-style-type: none"> 1. El Administrador busca al usuario que desea eliminar. 2. Selecciona la opción de darlo de baja. 3. El sistema da de baja al usuario.
Flujo alternativo	-
Poscondiciones	-

Desactivar usuario

Descripción	Permite desactivar a un usuario en el sistema.
Actores	Administrador.
Objetivos	Poder evitar a un usuario el acceso en el sistema. Si no podemos eliminar al usuario se hará uso de esta función.
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado sesión como Administrador. 2. El usuario debe existir.
Flujo normal	<ol style="list-style-type: none"> 1. El Administrador busca al usuario que desea eliminar. 2. Selecciona la opción de desactivar usuario. 3. El sistema guarda los cambios.
Flujo alternativo	-
Poscondiciones	-

Modificación de usuarios

Descripción	Permite modificar los datos de un usuario.
Actores	Administrador.
Objetivos	<ol style="list-style-type: none"> 1. Poder modificar los permisos (grupo al que pertenece) de un usuario así como los datos personales. 2. Asignar una nueva contraseña si el usuario perdió la antigua.
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado sesión como Administrador. 2. El usuario debe existir. 3. El usuario no debe formar parte de una operación que se encuentre activa si se asignan nuevos permisos que no permiten realizar esa operación.
Flujo normal	<ol style="list-style-type: none"> 1. El Administrador busca al usuario en el sistema y modifica los datos convenientes. 2. Al confirmar los nuevos datos el sistema comprobará que son válidos y que se pueden llevar a cabo. 3. El sistema pide confirmación y almacena los datos modificados.
Flujo alternativo	-
Poscondiciones	-

Listados

Descripción	Permite realizar listados de grupos o usuarios.
Actores	Administrador.
Objetivos	Permite obtener un resumen de los grupos y usuarios que hay en el sistema en formato lista.
Precondiciones	Haber iniciado sesión como Administrador.
Flujo normal	<ol style="list-style-type: none"> 1. Se selecciona la opción de listar grupos o usuarios. 2. Se seleccionan los campos que se desean incluir en el listado. 3. El sistema realiza la búsqueda y muestra un listado de la información.
Flujo alternativo	-
Poscondiciones	-

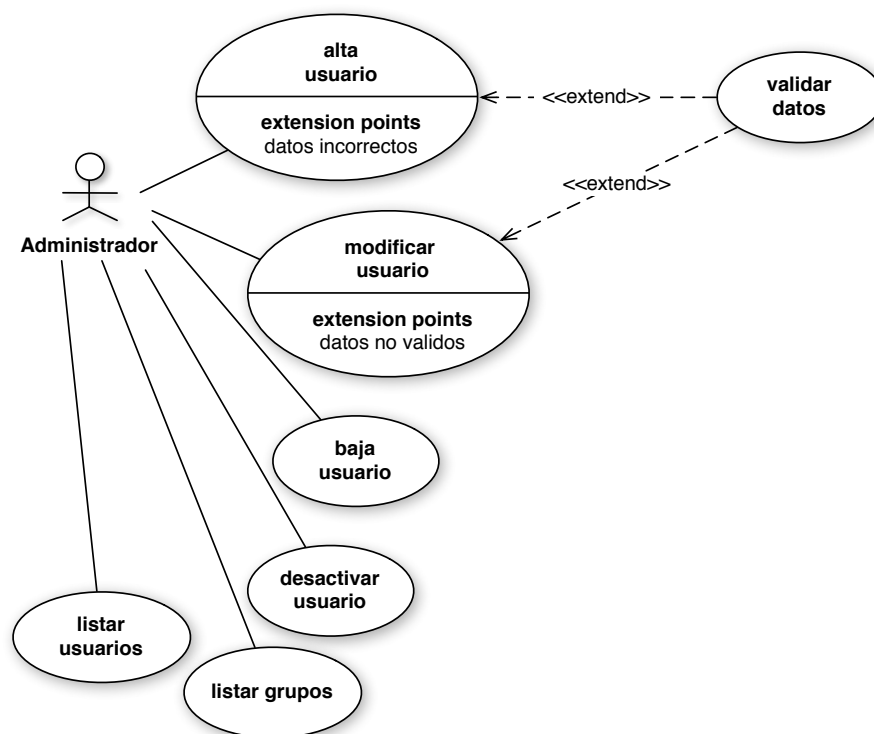


Figura 8: Gestión de usuarios

3.1.3. Configuración del sistema

Copias de seguridad

Descripción	Permite realizar copias de seguridad de la base de datos.
Actores	Administrador.
Objetivos	Mantener copias de seguridad de la base de datos.
Precondiciones	Haber iniciado sesión como Administrador.
Flujo normal	<ol style="list-style-type: none">1. El administrador configura los datos para que el sistema realice la copia de seguridad de la base de datos.2. El sistema realiza la copia de seguridad y la almacena en el destino especificado.
Flujo alternativo	-
Poscondiciones	-

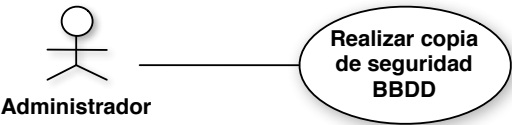


Figura 9: Copia de seguridad

3.1.4. Módulos de datos maestros

3.1.4.1. Gestión de proveedores

Alta de proveedor

Descripción	Permite dar de alta un proveedor en el sistema.
Actores	Administrador y Usuario.
Objetivos	Disponer de la información completa de un proveedor que es necesaria para realizar gestiones como pedidos de compra, evaluación de proveedores, etc.
Precondiciones	Haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. El actor pulsa el botón de alta de proveedor. 2. El sistema muestra un formulario para rellenar los datos del proveedor. 3. El actor rellena los datos del formulario. 4. El sistema valida el formulario y guarda los datos.
Flujo alternativo	Si los datos introducidos son incorrectos el sistema pedirá que se rectifiquen antes de guardar.
Poscondiciones	Se muestra notificación de que el proveedor ha sido dado de alta.

Baja de proveedor

Descripción	Permite eliminar a un proveedor del sistema.
Actores	Administrador y Usuario.
Objetivos	Poder dar de baja en el sistema a un proveedor que por ejemplo, ya no exista como razón social.
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado sesión. 2. El proveedor debe existir para poder ser modificado. 3. El proveedor no debe formar parte de un proyecto o pedido de compra.
Flujo normal	<ol style="list-style-type: none"> 1. El actor busca el proveedor a dar de baja. 2. Selecciona la opción de dar de baja al proveedor. 3. El sistema pide confirmación y da de baja al proveedor.
Flujo alternativo	-
Poscondiciones	Se muestra notificación de que el proveedor ha sido dado de baja.

Desactivar proveedor

Descripción	Permite marcar a un proveedor como inactivo.
Actores	Administrador y Usuario.
Objetivos	Poder marcar en el sistema a un proveedor como inactivo. Este caso se puede dar cuando por ejemplo un proveedor deja de existir como razón social. Si el cliente forma parte de un proyecto el sistema no permite eliminarlo, es por eso que utilizaremos la desactivación del cliente.
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado sesión. 2. El proveedor debe existir para poder ser modificado.
Flujo normal	<ol style="list-style-type: none"> 1. Se busca al proveedor y se cambia el estado a inactivo. 2. El sistema pide confirmación y guarda los cambios del proveedor.
Flujo alternativo	-
Poscondiciones	Se muestra notificación de que el proveedor ha sido dado de baja o marcado como inactivo.

Modificar proveedor

Descripción	Permite modificar los datos de un proveedor.
Actores	Administrador y Usuario.
Objetivos	Es posible que un proveedor cambie o modifique cualquier dato referente a su empresa. Esta funcionalidad permitirá actualizar los datos.
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado sesión. 2. El proveedor debe existir para poder ser modificado.
Flujo normal	<ol style="list-style-type: none"> 1. El actor pulsa el botón de modificar los datos del proveedor. 2. El sistema muestra un formulario para la búsqueda del proveedor. 3. El actor realiza una búsqueda del proveedor por algún campo clave. 4. El sistema muestra los datos del proveedor. 5. El actor modifica los datos. 6. El sistema pide confirmación y actualiza los datos.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si los datos introducidos son incorrectos el sistema pedirá que se rectifiquen antes de guardar. 2. El sistema almacena los datos validados.
Poscondiciones	Se muestra notificación de que el proveedor ha sido modificado.

Listar proveedor

Descripción	Permite realizar listados de proveedores.
Actores	Administrador y Usuario.
Objetivos	Tener la información de los usuarios resumida en listados personalizados.
Precondiciones	Haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. Se selecciona la opción de realizar listados de proveedores. 2. Se marcan los campos que se quieren incluir en el listado. 3. Se realiza una búsqueda para acotar los resultados (si se desea). 4. El sistema muestra el listado de proveedores teniendo en cuenta los puntos anteriores.
Flujo alternativo	-
Poscondiciones	-

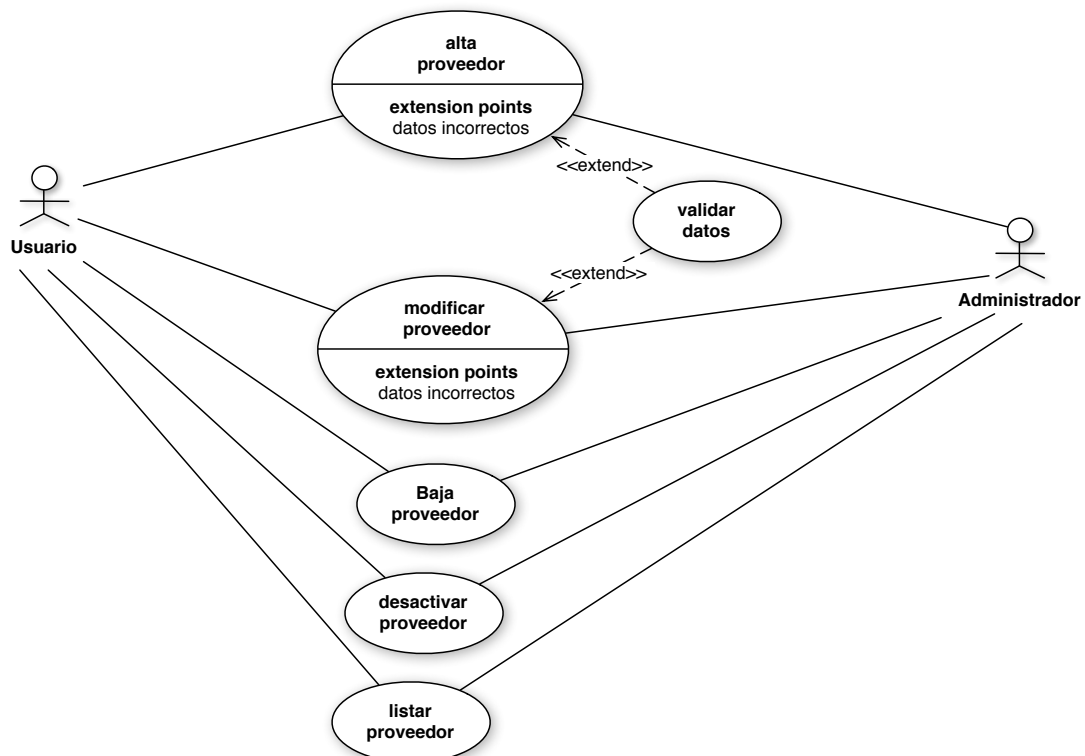


Figura 10: Gestión de proveedores

3.1.4.2. Gestión de clientes

Alta cliente

Descripción	Permite dar de alta un cliente en el sistema.
Actores	Administrador y Usuario.
Objetivos	Disponer de la información completa de un cliente que es necesaria para realizar gestiones como presupuestos, proyectos, facturación, etc.
Precondiciones	Haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. El actor pulsa el botón de alta de cliente 2. El sistema muestra un formulario para rellenar los datos del cliente. 3. El actor rellena los datos del formulario 4. El sistema valida el formulario y guarda los datos.
Flujo alternativo	Si los datos introducidos son incorrectos el sistema pedirá que se rectifiquen antes de guardar.
Poscondiciones	Se guarda la información del cliente en el sistema.

Baja cliente

Descripción	Permite eliminar a un cliente del sistema.
Actores	Administrador y Usuario.
Objetivos	Poder dar de baja en el sistema a un cliente que por ejemplo, ya no exista como razón social.
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado sesión. 2. El cliente debe existir para poder ser modificado. 3. El cliente no debe formar parte de un proyecto o pedido.
Flujo normal	<ol style="list-style-type: none"> 1. El actor busca el cliente a dar de baja. 2. Selecciona la opción de dar de baja al cliente. 3. El sistema pide confirmación y da de baja al cliente.
Flujo alternativo	-
Poscondiciones	Se muestra notificación de que el cliente ha sido dado de baja o marcado como inactivo.

Desactivar cliente

Descripción	Permite marcar a un cliente como inactivo.
Actores	Administrador y Usuario.
Objetivos	Poder marcar en el sistema a un cliente como inactivo. Este caso se puede dar cuando por ejemplo un cliente deja de existir como razón social. Si el cliente forma parte de un proyecto el sistema no permite eliminarlo, es por eso que utilizaremos la desactivación del cliente.
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado sesión. 2. El cliente debe existir para poder ser modificado.
Flujo normal	<ol style="list-style-type: none"> 1. Se busca al cliente y se cambia el estado a inactivo. 2. El sistema pide confirmación y guarda los cambios del cliente.
Flujo alternativo	-
Poscondiciones	Se muestra notificación de que el cliente ha sido dado de baja o marcado como inactivo.

Modificar cliente

Descripción	Permite modificar los datos de un cliente.
Actores	Administrador y Usuario.
Objetivos	Es posible que un cliente cambie de domicilio o se modifique cualquier otro dato personal. Esta funcionalidad permitirá actualizar los datos.
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado sesión. 2. El cliente debe existir para poder ser modificado.
Flujo normal	<ol style="list-style-type: none"> 1. El actor pulsa el botón de modificar los datos del cliente. 2. El sistema muestra un formulario para la búsqueda del cliente. 3. El actor realiza una búsqueda del cliente por algún campo clave. 4. El sistema muestra los datos del cliente. 5. El actor modifica los datos. 6. El sistema pide confirmación y actualiza los datos.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si los datos introducidos son incorrectos el sistema pedirá que se rectifiquen antes de guardar. 2. El sistema almacena los datos validados correctamente.
Poscondiciones	Se muestra notificación de que el cliente ha sido modificado.

Listar cliente

Descripción	Permite realizar listados de clientes.
Actores	Administrador y Usuario.
Objetivos	Tener la información de los usuarios resumida en listados personalizados.
Precondiciones	Haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none">1. Se selecciona la opción de realizar listados de clientes.2. Se marcan los campos que se quieren incluir en el listado.3. Se realiza una búsqueda para acotar los resultados (si se desea).4. El sistema muestra el listado de clientes teniendo en cuenta los puntos anteriores.
Flujo alternativo	-
Poscondiciones	-

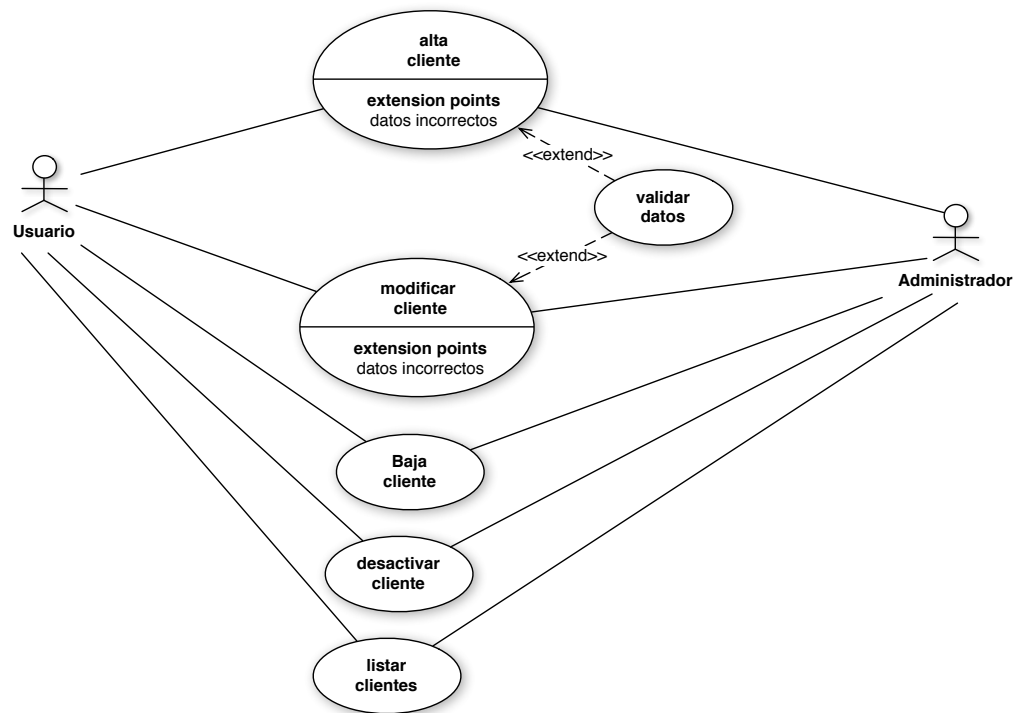


Figura 11: Gestión de cliente

3.1.4.3. Gestión de artículos

Alta de artículo

Descripción	Permite crear un artículo.
Actores	Administrador y Usuario.
Objetivos	Disponer de un catálogo de artículo que podemos consultar cuando sea necesario.
Precondiciones	El usuario debe haber iniciado sesión en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El actor pulsa sobre el botón para dar de alta un artículo. 2. El sistema muestra un formulario para rellenar los datos requeridos de los que se compone un artículo. 3. El actor rellena el formulario. 4. El sistema comprueba la validez de los datos y los almacena.
Flujo alternativo	Si la validación de los datos es incorrecta se avisa al usuario que debe modificar los datos introducidos.
Poscondiciones	Se muestra notificación de que el artículo ha sido dado de alta correctamente.

Baja de artículo

Descripción	Permite eliminar un artículo.
Actores	Administrador y Usuario.
Objetivos	Permite eliminar un artículo que no se desea tener en el sistema.
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe haber iniciado sesión en el sistema. 2. El artículo no debe formar parte de ningún proyecto.
Flujo normal	<ol style="list-style-type: none"> 1. Se busca el artículo y se elimina. 2. El sistema pide confirmación y se elimina el artículo del sistema.
Flujo alternativo	-
Poscondiciones	Se muestra notificación de que el artículo ha sido dado de baja correctamente.

Modificar artículo

Descripción	Permite modificar los datos de un artículo.
Actores	Administrador y Usuario.
Objetivos	Rectificar la introducción de datos incorrectos sobre un determinado artículo o la simple modificación de alguno de ellos.
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado sesión. 2. El producto debe existir para poder modificarlo. 3. No debe formar parte de ningún proyecto.
Flujo normal	<ol style="list-style-type: none"> 1. El actor pulsa el botón de modificar productos. 2. El sistema muestra un formulario para la búsqueda de uno o más productos. 3. El actor realiza una búsqueda del producto por un campo clave. 4. El sistema muestra los datos del producto. 5. El actor modifica los datos. 6. El sistema pide confirmación y actualiza los datos.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si el actor modifica un campo e introduce un valor no válido el sistema avisará del error para que se revisen los datos introducidos. 2. El sistema almacena los datos validados.
Poscondiciones	Se muestra notificación de que el artículo ha sido modificado.

Listar artículos

Descripción	Permite realizar listados de artículo.
Actores	Administrador y Usuario.
Objetivos	Tener la información de los artículo resumida en listados personalizados.
Precondiciones	Haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. Se selecciona la opción de realizar listados de artículos. 2. Se marcan los campos que se quieren incluir en el listado. 3. Se realiza una búsqueda para acotar los resultados (si se desea). 4. El sistema muestra el listado de artículos teniendo en cuenta los puntos anteriores.
Flujo alternativo	-

Listar artículos

Poscondiciones

-

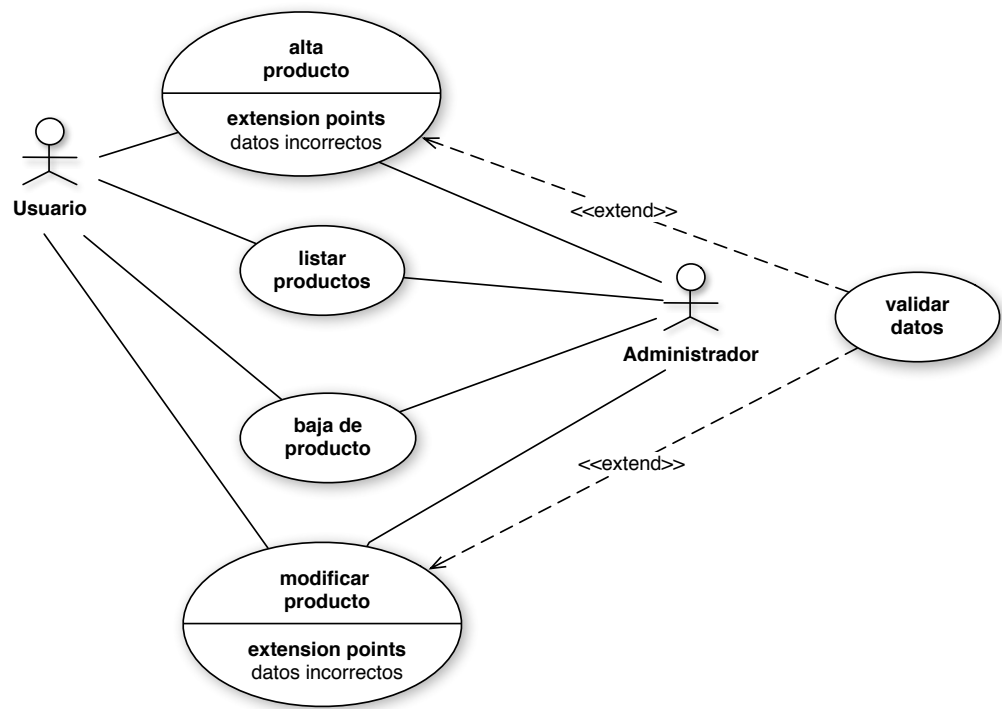


Figura 12: Gestión de productos

3.1.4.4. Gestión de materiales

Alta de material

Descripción	Permite crear un material.
Actores	Administrador y Usuario.
Objetivos	Disponer de un catálogo de los materiales con los que solemos trabajar y visualizar rápidamente que proveedores lo suministran.
Precondiciones	Haber iniciado sesión en el sistema.
Flujo normal	<div>1. Se selecciona la opción de dar de alta un material.</div> <div>2. El sistema muestra un formulario para rellenar los datos requeridos de los que se compone un material.</div> <div>3. El sistema comprueba la validez de los datos y los almacena en el sistema.</div>
Flujo alternativo	Si la validación de los datos es incorrecta se avisa al actor que debe modificar los datos introducidos.
Poscondiciones	-

Baja de material

Descripción	Permite eliminar un material del sistema.
Actores	Administrador y Usuario.
Objetivos	Permitir eliminar un material que hemos introducido por error y no queremos utilizarlo nunca.
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado sesión. 2. El material debe de existir. 3. El material no debe formar parte de ningún producto o proyecto activo.
Flujo normal	<ol style="list-style-type: none"> 1. Se busca en el catálogo el material y lo eliminamos. 2. El sistema pide confirmación y lo elimina del sistema.
Flujo alternativo	-
Poscondiciones	Se muestra notificación de que el material ha sido dado de baja.

Modificación de material

Descripción	Permite modificar los datos de un material.
Actores	Administrador y Usuario.
Objetivos	Poder modificar un material que hemos dado de alta con algún dato incorrecto sin necesidad de eliminarlo y crearlo de nuevo.
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado sesión. 2. El material debe de existir. 3. El material no debe formar parte de ningún producto o proyecto activo.
Flujo normal	<ol style="list-style-type: none"> 1. Se busca en el catálogo el material y se modifican los datos. 2. El sistema valida los datos introducidos, pide confirmación y almacena los cambios.
Flujo alternativo	-
Poscondiciones	Se muestra notificación de que el material ha sido dado de baja.

Listar materiales

Descripción	Permite obtener listados de materiales.
Actores	Administrador y Usuario.
Objetivos	Permite obtener listados de materiales.
Precondiciones	Haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. Se selecciona la opción de listados de materiales. 2. Se realiza una búsqueda sin parámetros o con parámetros. 3. El sistema muestra el listado de materiales.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si los parámetros son incorrectos el sistema informará sobre ello para que se modifiquen. 2. El sistema muestra el listado de materiales una vez los parámetros son correctos.
Poscondiciones	-

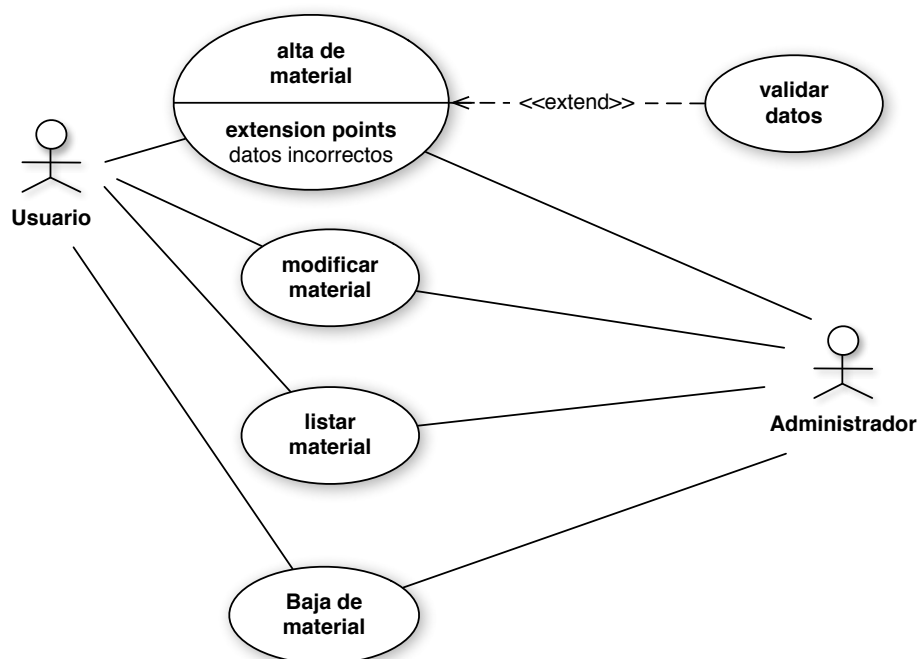


Figura 13: Gestión de materiales

3.1.4.5. Gestión de escandallos

Alta de escandallo

Descripción	Creación de un escandallo relacionado con un artículo.
Actores	Administrador y usuario.
Objetivos	Permite calcular el precio de un artículo.
Precondiciones	Haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. Se selecciona la opción de generar escandallo. 2. Se introducen los datos necesarios para realizar un escandallo. 3. El sistema valida los datos y los almacena en el sistema.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si el los datos introducidos no son válidos el sistema informará sobre ello para que se modifiquen. 2. El sistema almacena los datos en el sistema.
Poscondiciones	-

Modificar escandallo

Descripción	Modificar los datos de un escandallo.
Actores	Administrador y usuario.
Objetivos	Poder modificar los datos introducidos en la creación de un escandallo sin tener que crear uno de nuevo.
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado sesión. 2. El escandallo debe existir.
Flujo normal	<ol style="list-style-type: none"> 1. Se busca el escandallo a modificar. 2. Se introducen los nuevos datos. 3. El sistema valida los datos y los almacena en el sistema.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si el los datos introducidos no son válidos el sistema informará sobre ello para que se modifiquen. 2. El sistema almacena los datos en el sistema.
Poscondiciones	-

Eliminar escandallo

Descripción	Eliminar un escandallo creado.
Actores	Administrador y usuario.
Objetivos	Permite eliminar un escandallo que no se desea tener en el sistema.
Precondiciones	1. Haber iniciado sesión. 2. El escandallo debe existir. 3. El escandallo no debe formar parte de ningún proyecto.
Flujo normal	1. Se busca el escandallo a eliminar. 2. Se elimina y el sistema modifica los datos.
Flujo alternativo	-
Poscondiciones	-

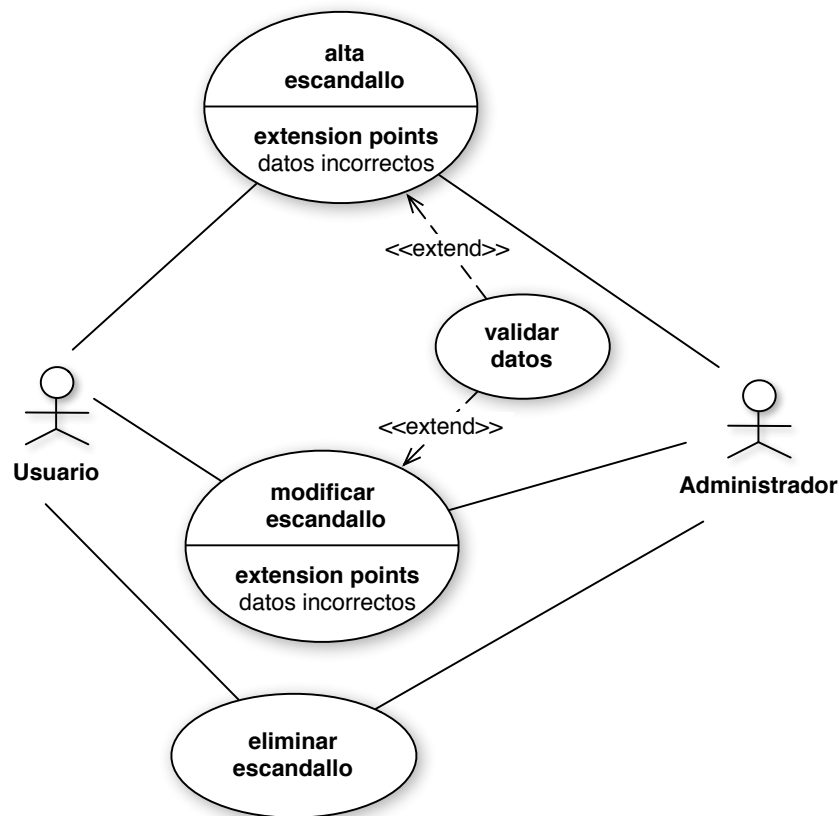


Figura 14: Gestión de escandallos

3.1.5. Módulo de gestión de aprovisionamientos

3.1.5.1. Gestión de pedidos de compra

Alta de pedido de compra

Descripción	Permite dar de alta un pedido de compra a un proveedor.
Actores	Administrador y Usuario.
Objetivos	Tener constancia de los pedidos que se realizan a los proveedores y relacionarlos con un proyecto.
Precondiciones	Haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. Se selecciona la opción de pedidos de compra. 2. Se introducen los campos necesarios para realizar el pedido y el sistema valida los datos. 3. El sistema almacena el pedido en el sistema.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si los parámetros son incorrectos el sistema informará sobre ello para que se modifiquen. 2. El sistema almacena el pedido en el sistema.
Poscondiciones	-

Modificación de pedido de compra

Descripción	Permite modificar un pedido de compra.
Actores	Administrador y Usuario.
Objetivos	Permite modificar campos de un pedido que se hayan modificado o introducido erróneamente.
Precondiciones	Haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. Se busca el pedido que se quiere modificar. 2. Se modifican los campos que se quieren modificar y el sistema valida los datos. 3. El sistema almacena las modificaciones del pedido en el sistema.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si los parámetros son incorrectos el sistema informará sobre ello para que se modifiquen. 2. El sistema almacena las modificaciones del pedido en el sistema.
Poscondiciones	-

Cancelación de pedido de compra

Descripción	Permite cancelar un pedido de compra.
Actores	Administrador y Usuario.
Objetivos	Reflejar la cancelación de un pedido que no va a ser recibido nunca.
Precondiciones	Haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. Se busca el pedido que se quiere cancelar. 2. Se marcar el pedido como cancelado y el sistema almacena los cambios. 3. El sistema mostrará los problemas que causan la cancelación de dicho pedido.
Flujo alternativo	-
Poscondiciones	-

Listados de pedidos de compra

Descripción	Permite listar los pedidos de compra que se han realizado.
Actores	Administrador y Usuario.
Objetivos	Obtener información resumida sobre los pedidos de compra.
Precondiciones	Haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. Se selecciona la opción de listados de pedidos de compra. 2. Se realiza una búsqueda sin parámetros o con parámetros. 3. El sistema muestra el listado de materiales.
Flujo alternativo	-
Poscondiciones	-

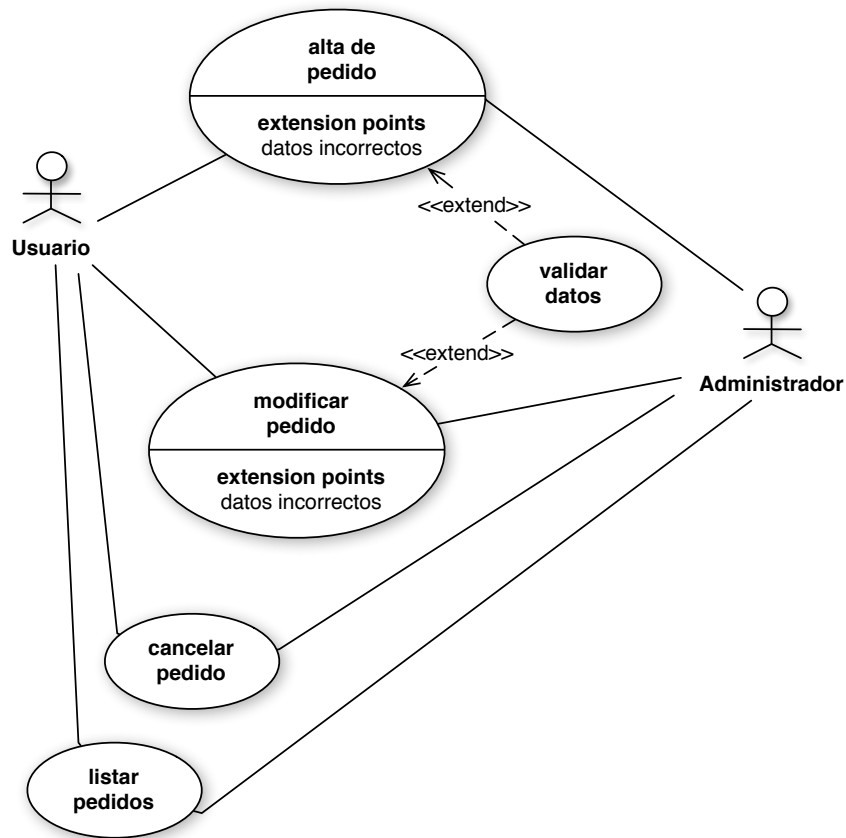


Figura 15: gestión de pedidos de compra

3.1.5.2. Gestión de recepción de mercancías

Registrar retrasos

Descripción	Permite controlar los retrasos de un pedido para tomar las acciones pertinentes.
Actores	Administrador y Usuario.
Objetivos	Tener constancia de los retrasos que se producen en los pedidos realizados y reflejar las soluciones que se han llevado a cabo para solventarlo.
Precondiciones	1. Haber iniciado sesión. 2. El pedido debe tener un retraso de entrega.
Flujo normal	1. Se selecciona la opción de retrasos de entrega. 2. Se anotan los motivos y las modificaciones que se hayan podido producir. 3. El sistema almacena los datos.
Flujo alternativo	-
Poscondiciones	-

Listar pedidos de compra pendientes de entrega

Descripción	Realizar listados sobre los pedidos que están en curso.
Actores	Administrador y Usuario.
Objetivos	Tener constancia de los pedidos que se han realizado y están pendientes de ser entregados.
Precondiciones	Haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. Se selecciona la opción de listar mercancías pendientes de entrega. 2. Se realiza una búsqueda con parámetros o no. 3. El sistema valida los parámetros de acotamiento (si se han introducido) y muestra el listado en base a ello.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si los datos introducidos son erróneos el sistema informará de ello para que sean modificados. 2. El sistema muestra el listado.
Poscondiciones	-

Devoluciones de pedidos de compra

Descripción	Permite realizar la devolución de pedidos o parte de ellos al proveedor.
Actores	Administrador y Usuario.
Objetivos	Devolver mercancías que son erróneas o que están defectuosas para que sean cambiadas.
Precondiciones	Haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. Se busca el pedido de compra con el que queremos realizar una devolución. 2. Se anotan los motivos y se rellena los campos necesarios para la devolución. 3. El sistema almacena los datos.
Flujo alternativo	-
Poscondiciones	-

Informes de pedidos de compra

Descripción	Permite realizar la informes de los pedidos de compra que se han realizado.
Actores	Administrador y Usuario.
Objetivos	Obtener información detallada sobre los pedidos que se han realizado.
Precondiciones	Haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. Se selecciona la opción de informes de pedidos. 2. Se escogen las opciones para la generación del informe. 3. El sistema muestra el informe generado teniendo en cuenta el punto anterior.
Flujo alternativo	-
Poscondiciones	-

Evaluar proveedor

Descripción	Permite evaluar el servicio ofrecido por un proveedor en un determinado pedido de compra.
Actores	Administrador y Usuario.
Objetivos	Obtener información resumida sobre el trato con un determinado proveedor y tenerlo en cuenta en futuros pedidos.
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado sesión. 2. Haber realizado como mínimo un pedido al proveedor.
Flujo normal	<ol style="list-style-type: none"> 1. Se selecciona la opción evaluación de proveedores. 2. Se rellenan los campos necesarios para realizar la evaluación. 3. El sistema almacena los datos.
Flujo alternativo	-
Poscondiciones	-

Modificar estado de pedido de compra

Descripción	Permite modificar el estado de un pedido de compra: pendiente o recibido.
Actores	Administrador y Usuario.
Objetivos	Permite especificar en que estado se encuentra un pedido.
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado sesión. 2. El pedido debe existir.
Flujo normal	<ol style="list-style-type: none"> 1. Se busca el pedido de compra. 2. Se modifica el estado del pedido. 3. El sistema almacena los datos.
Flujo alternativo	-
Poscondiciones	-

Modificar evaluación de proveedor

Descripción	Permite modificar los datos de una evaluación de un proveedor.
Actores	Administrador y Usuario.
Objetivos	Rectificar valoraciones erróneas.
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado sesión. 2. El proveedor debe tener una evaluación realizada.
Flujo normal	<ol style="list-style-type: none"> 1. Se busca el proveedor que queremos modificar la evaluación. 2. Se modifican los datos. 3. El sistema almacena las modificaciones.
Flujo alternativo	-
Poscondiciones	-

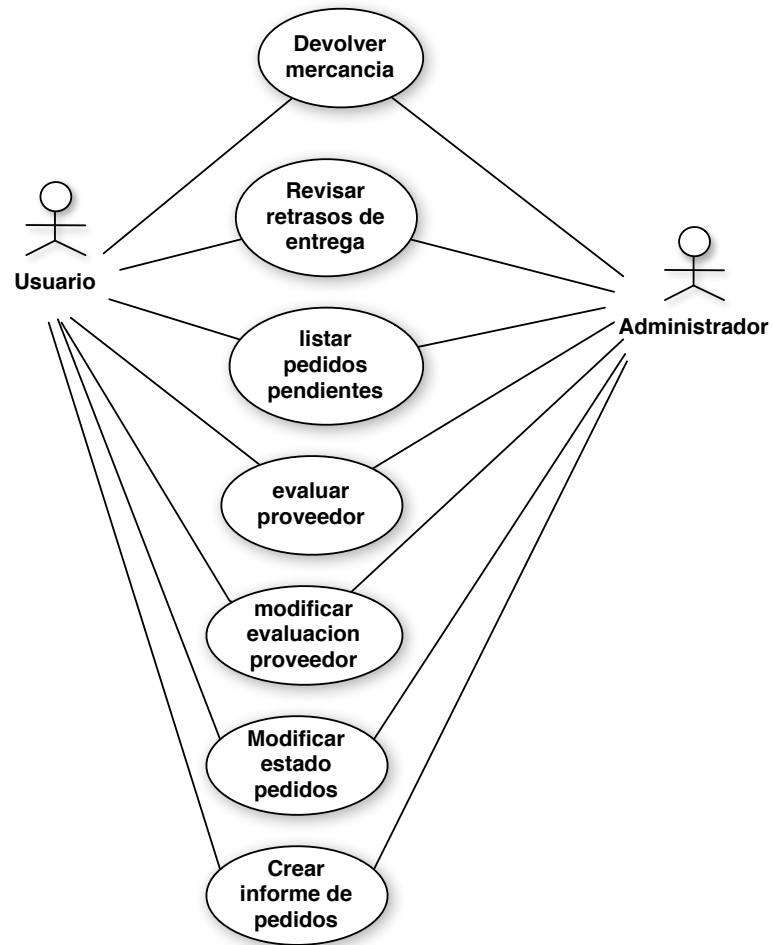


Figura 16: Gestión de recepción de mercancías

3.1.6. Módulo de gestión de proyectos

3.1.6.1. Gestión de proyectos

Alta de proyecto

Descripción	Creación de un proyecto.
Actores	Administrador y usuario.
Objetivos	Iniciar un proyecto para llevar un seguimiento y control exhaustivo.
Precondiciones	Haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. Se selecciona la opción de crear un proyecto. 2. Se rellenan los datos necesarios para iniciar un proyecto. 3. El sistema valida los datos y los almacena en el sistema.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si los datos introducidos no son válidos el sistema informará sobre ello para que se modifiquen. 2. El sistema almacena los datos en el sistema.
Poscondiciones	-

Cancelar proyecto

Descripción	Permite cancelar un proyecto.
Actores	Administrador y usuario.
Objetivos	Poder suspender un proyecto que por algunos motivos no se llevará a cabo hasta su finalización.
Precondiciones	Haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. Se selecciona la opción de crear un proyecto. 2. Se rellenan los datos necesarios para iniciar un proyecto. 3. El sistema valida los datos y los almacena en el sistema.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si los datos introducidos no son válidos el sistema informará sobre ello para que se modifiquen. 2. El sistema almacena los datos en el sistema.
Poscondiciones	-

Crear tareas

Descripción	Permite definir las tareas que se llevan a cabo en cada fase del proyecto.
Actores	Administrador y usuario.
Objetivos	Planificar las tareas de un proyecto que forman parte de una fase y asegurarse de que se van cumpliendo en las fechas señaladas. Permitirá llevar un control del proyecto.
Precondiciones	Haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. Se selecciona la fase a la que se le quiere añadir tareas. 2. Se rellenan los campos necesarios para crear una tarea y el sistema valida los datos introducidos. 3. El sistema almacena los datos.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si el los datos introducidos no son válidos el sistema informará sobre ello para que se modifiquen. 2. El sistema almacena los datos en el sistema.
Poscondiciones	-

Seguimiento del proyecto

Descripción	Permite marcar las tareas realizadas y comprobar las fases finalizadas.
Actores	Administrador y usuario.
Objetivos	Llevar un control del proyecto y detectar problemas de manera fácil.
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado sesión. 2. El proyecto debe haberse iniciado. 3. Para marcar una tarea como finalizada como mínimo debe de haber pasado la fecha de inicio.
Flujo normal	<ol style="list-style-type: none"> 1. Se busca el proyecto y se marcan las tareas realizadas. 2. El sistema almacena los datos.
Flujo alternativo	-
Poscondiciones	-

Anotar gastos

Descripción	Permite anotar los gastos relacionados con un proyecto.
Actores	Administrador y usuario.
Objetivos	Registrar los gastos que se han producido en la fabricación de un artículo y que no han sido contemplados en el escandallo. De esta manera se controla si el precio estimado de un artículo y el precio real son iguales o no.
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado sesión. 2. El proyecto debe de haberse iniciado.
Flujo normal	<ol style="list-style-type: none"> 1. Se busca el proyecto al que se le quieren asociar gastos. 2. Se introducen los campos necesarios para añadir un gasto. 3. El sistema almacena los datos.
Flujo alternativo	-
Poscondiciones	-

Crear informes

Descripción	Permite crear informes sobre un proyecto.
Actores	Administrador y usuario.
Objetivos	Crear informes detallados sobre todos los elementos de los que se compone un proyecto.
Precondiciones	Haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. Seleccionar la opción de realizar informes sobre proyectos. 2. Especificar las opciones de las que se puede componer un informe. 3. El sistema mostrará el informe generado teniendo en cuenta el punto anterior.
Flujo alternativo	-
Poscondiciones	-

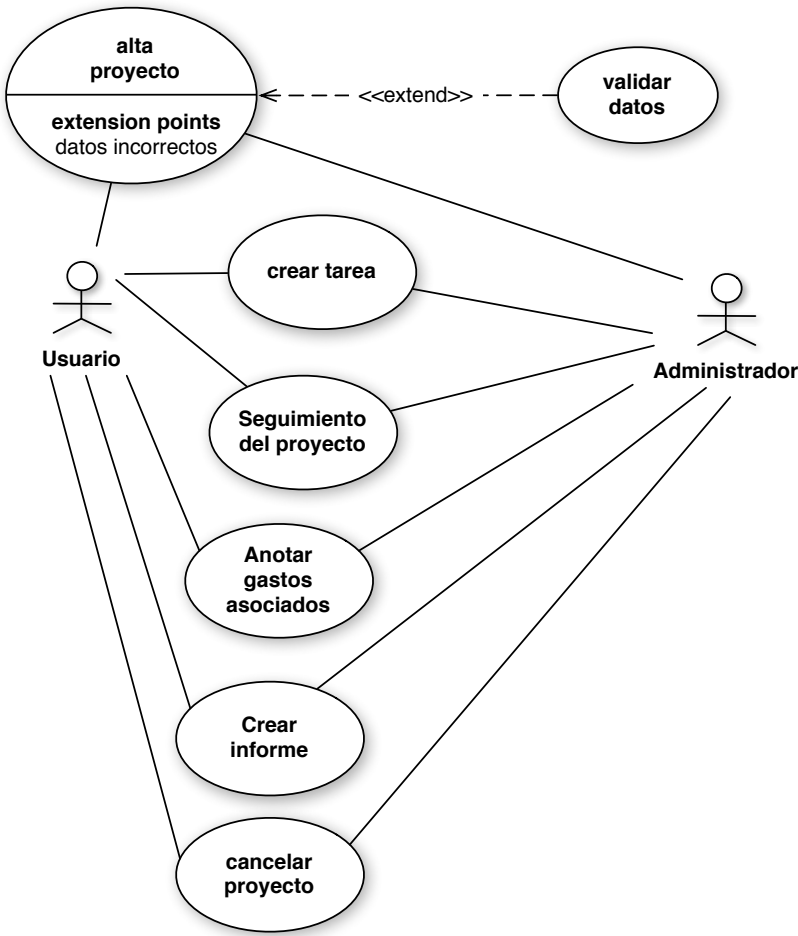


Figura 17: Gestión de proyectos

3.1.6.2. Gestión de pedidos de venta

Alta de pedido de venta

Descripción	Creación de un pedido de venta bajo petición de un cliente.
Actores	Administrador y usuario.
Objetivos	Permite registrar los pedidos que realizan los clientes.
Precondiciones	Haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. Se selecciona la opción de generar pedido de venta. 2. Se introducen los datos necesarios para realizar un pedido. 3. El sistema valida los datos y los almacena en el sistema.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si los datos introducidos no son válidos el sistema informará sobre ello para que se modifiquen. 2. El sistema almacena los datos en el sistema.
Poscondiciones	-

Modificar pedido de venta

Descripción	Modificar los datos de un pedido de venta.
Actores	Administrador y usuario.
Objetivos	Poder modificar los datos erróneos de un pedido de venta.
Precondiciones	<ol style="list-style-type: none"> 1. Haber iniciado sesión. 2. Si el pedido de venta ha generado un proyecto y éste está iniciado, no se podrá modificar. No debe estar asociado a ningún proyecto.
Flujo normal	<ol style="list-style-type: none"> 1. Se busca el pedido a modificar. 2. Se introducen los nuevos datos. 3. El sistema valida los datos y los almacena en el sistema.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si los datos introducidos no son válidos el sistema informará sobre ello para que se modifiquen. 2. El sistema almacena los datos en el sistema.
Poscondiciones	-

Eliminar pedido de venta

Descripción	Eliminar un pedido de venta creado.
Actores	Administrador y usuario.
Objetivos	Permite eliminar un pedido de venta que no se desea tener en el sistema.
Precondiciones	1. Haber iniciado sesión. 2. El pedido de venta debe existir. 3. El pedido de venta no debe formar parte de ningún proyecto.
Flujo normal	1. Se busca el pedido de venta a eliminar. 2. Se elimina y el sistema modifica los datos.
Flujo alternativo	1. Si los datos introducidos no son válidos el sistema informará sobre ello para que se modifiquen. 2. El sistema almacena los datos en el sistema.
Poscondiciones	-

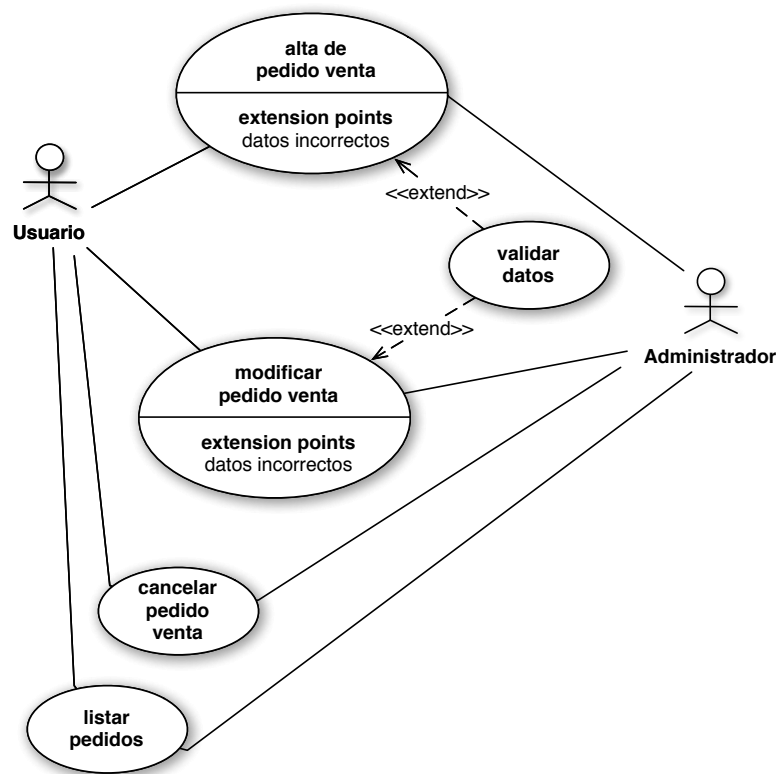


Figura 18: gestión de pedidos de venta

3.2. Modelo conceptual

3.2.1. Análisis de requerimientos

Antes de definir nuestro modelo conceptual de la base de datos vamos a ver que información necesitamos almacenar y cual es su finalidad.

3.2.1.1. Requisitos de datos

- Se distinguen dos tipos de **usuarios**, Administrador y usuario normal. Para cada tipo se almacena el *nombre de usuario*, la *contraseña*, *nombre y apellidos*, *dni*, *dirección*, *email* y el *estado del usuario*: activado o desactivado. Cada usuario pertenecerá a un grupo de usuarios.
- Para gestionar el control de privilegios sobre los usuarios necesitamos almacenar **grupos** de usuarios, compuestos por el nombre del grupo y el **permiso** asociado al grupo. Un *permiso* tendrá un *nivel* específico.
- Será de gran utilidad almacenar los **países**. De cada uno guardamos el identificador ISO numérico, el identificador ISO abreviado de letras, un segundo identificador ISO abreviado de letras y el nombre completo. Por ejemplo: 724, ES, ESP, España.
- Interesa guardar los datos principales de una **empresa** -haciendo distinción entre cliente y proveedor-. Los datos más relevantes a guardar son: nombre, dirección, NIF y servicio que ofrece. En la siguiente tabla se muestran los diferentes servicios que puede ofrecer un proveedor -hay que tener en cuenta que un proveedor puede ofrecer todos los servicios-.

Servicio	Descripción
Urdir	Esta operación consiste en urdir uno o muchos hilos. Como resultado obtenemos uno o muchos plegadores -dependiendo de los metros-.
Tejer	Esta operación consiste en tejer uno o muchos hilos con un plegador. Como resultado obtenemos uno o varios tubos de tejido. En muchas ocasiones este será el artículo final que se entregará al cliente. En otros casos hará falta aplicarle algún acabado.
Acabar	Esta operación consiste en darle un acabado a un tubo de tejido. El acabado puede ser: preparar o estampar.
Compra de hilo	Este servicio consiste en el abastecimiento de materias primas, en este caso los hilos.
Otros	Este tipo nos permite definir un servicio personalizado para ser usado en caso de que tengamos un proveedor que realice un servicio diferente a los anteriores.

- Se necesita guardar un catálogo de los materiales con los que trabaja la empresa. En este caso es un único tipo de material, los **hilos**. Cada hilo se compone de un *identificador*, *nombre*, *coste*, *proveedores* que lo suministran y una serie de *propiedades* que se detallan en tabla 2. Hay que tener en cuenta la diferencia entre hilos de catálogo -los cuales sólo nos sirven para saber sus propiedades- y los hilos comprados -son los hilos físicamente como tal-.

Propiedad del hilo	Descripción
/título comercial	Este es el título que figura en la descripción del hilo, por tanto, sirve para identificarlo. Se expresa en Nm.
longitud	Número real que representa la longitud del hilo cortado, expresada en milímetros.
color	Representa el color del hilo.
brillo	Define el brillo del hilo. Puede ser: brillante, semi-mate o mate.
vuelatas	Representa el número de vueltas que tiene el hilo.
sentido	Representa el sentido del hilo: izquierda o derecha.

- También será necesario disponer de un catálogo de **artículos**. Un artículo es un tejido que se compone de una trama y un urdimbre concreto. La **trama** y el **urdimbre** están compuestos de diferentes hilos. Por otra parte, el artículo está definido por una serie de operaciones que son necesarias para su elaboración. Las operaciones que se le pueden realizar son: **acabar**, **urdir** y **tejer**, siendo estas dos últimas obligatorias para que un artículo pueda fabricarse. Se distingue entre dos tipos de artículos: prototipo y final.
- A cada artículo de nuestro catálogo se le asocia con un **escandallo**. En éste se detallará que proveedores formarán parte de la elaboración del tejido -ver el ejemplo de abajo- y los gastos asociados que se prevean. En definitiva, cuanto cuesta cada una de las partes materiales y del proceso necesario para elaborarlo. Esto nos sirve para determinar el precio estimado del artículo y fijar un beneficio. El precio será estimado porque cuando se lleve a cabo la elaboración del tejido pueden surgir nuevos gastos que no estaban previstos.

Ejemplo: El cliente nos pide el artículo Omega con ref.: 903450. Este artículo necesita las operaciones de urdir, tejer y acabado. Especificamos para cada operación del artículo que proveedor la realizará:

urdir : Urtextil,S.L	coste: 0,5€/m
tejer : Urtextil,S.L	coste: 0,2€/m
acabar: Filatura,S.A	coste: 0,15€/m

En este caso, para la elaboración del artículo Omega han intervenido 2 proveedores diferentes.

- También nos interesa registrar los pedidos que se realizan. Los pedidos pueden ser de dos tipos: pedidos de venta y pedidos de compra. Hay que tener en cuenta que un pedido puede tener un **retraso** y habrá que registrar el motivo y el tiempo esperado del retraso.

- **Pedidos de venta:** Son los que se llevan a cabo bajo petición del cliente. Un cliente nos pedirá uno o varios artículos a comprar. Normalmente los artículos pedidos están en nuestro catálogo. De no ser así, se crearía la especificación de uno nuevo y lo daríamos de alta en nuestro catálogo -elaborando también el escandallo-. Si el cliente está de acuerdo con el precio del artículo se formaliza el pedido de venta. El pedido puede tener varios estados: borrador, enviado, en producción, realizado o servido -en la tabla 3 se detalla cada estado-. De un pedido de compra nos interesa almacenar: el *cliente*, el *artículo* y la *fecha de pedido* y *entrega*. Cuando el pedido deja de estar en estado *borrador* se inicia un proyecto para llevar a cabo la producción del artículo pedido.

Estado del pedido	Descripción
Borrador	Este estado es el inicial. Cuando un cliente realiza un pedido éste permanece en estado borrador hasta que cambia a <i>enviado</i> .
Aceptado	Este estado indica que el pedido de venta se llevará a cabo y está a la espera de iniciar un proyecto.
Cursando	Este estado se da cuando se ha iniciado un proyecto relacionado con el pedido de venta y al menos alguna de sus fases está en producción. Indica que el pedido se está procesando para su entrega.
Finalizado	Indica que el pedido se ha realizado, es decir, el proyecto ha finalizado y ya se tiene la pieza/s a entregar al cliente.
Entregado	Indica que el pedido ha sido servido al cliente.

- **Pedidos de compra:** Son los que se realizan a los proveedores. Un pedido de compra consiste en la compra de un servicio. En la siguiente tabla se detallan los servicios que puede ofrecer un proveedor. Los pedidos de compra pueden estar relacionados con uno o varios proyectos. Por ejemplo: podemos pedirle a un proveedor que nos fabrique un plegador y éste lo podemos utilizar en varios proyectos. Por tanto, el pedido de compra del plegador, formará parte de dos proyectos distintos. En otros casos, se pueden realizar pedidos que no estén vinculados a ningún proyecto, por ejemplo, la compra de hilo. Los pedidos de compra también tienen un estado y son los siguientes:

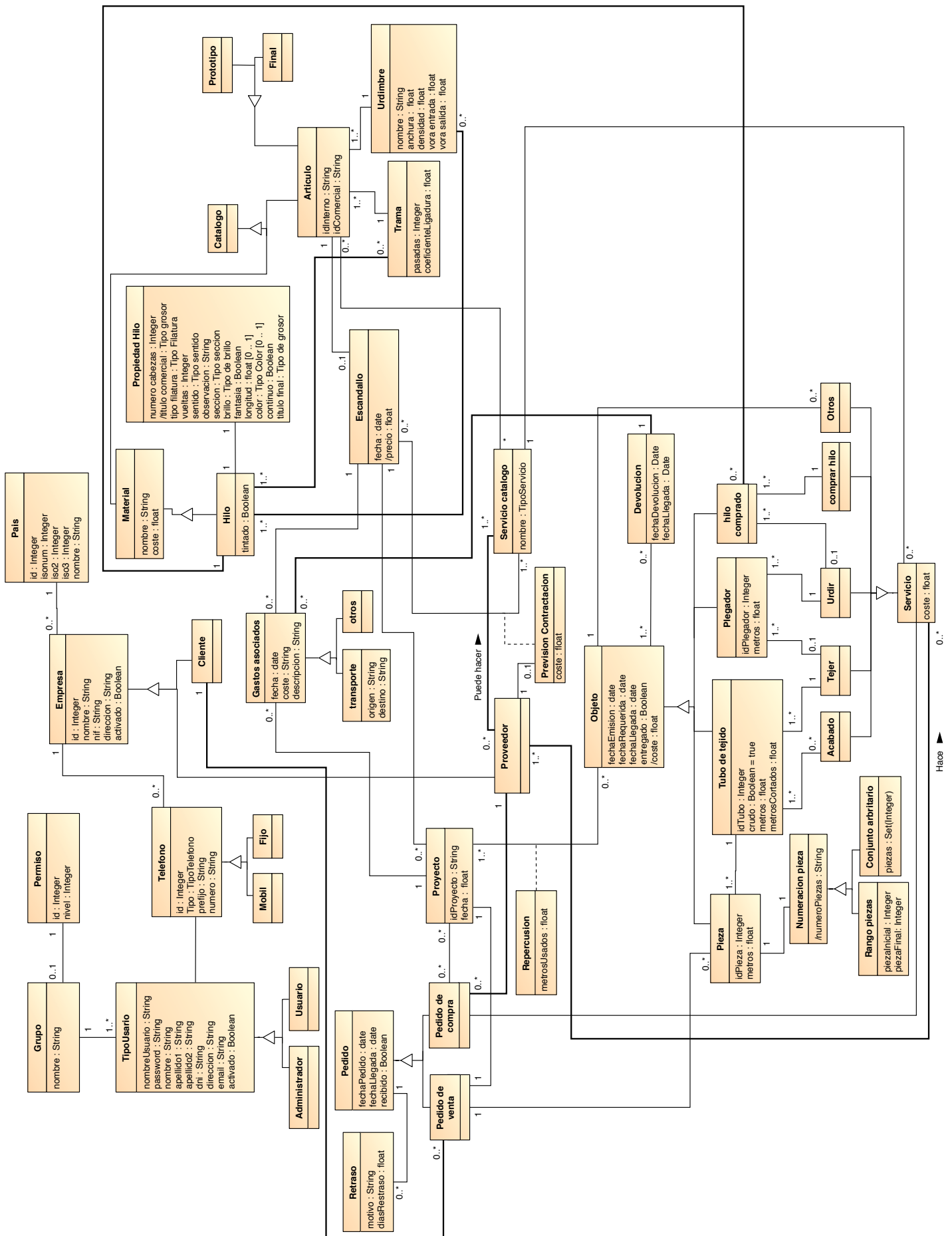
Estado del pedido	Descripción
Cursando	Este estado se da cuando se ha creado el pedido de compra y el proveedor lo está tramitando.
Finalizado	Indica que el pedido se ha realizado, es decir, el proveedor nos ha entregado el pedido.

- Cuando realizamos un pedido de compra obtenemos un **objeto**. Un objeto puede ser de varios tipos según del servicio del que proceda. En la tabla 4 se detalla esta relación.

Objeto	Resultado del servicio...	Servicio a aplicar
Plegador	urdir	-
Tubo de tejido	tejer	acabar (opcional)
Pieza	cortar el tubo de tejido	-
hilo comprado	compra de hilo	-

El objeto *pieza* es el artículo que se le entrega al cliente. Si el cliente no quiere el tejido cortado en piezas, se le entrega el tubo de tejido.

- De cada **pieza** guardaremos el identificador del tubo del que proviene y los metros que tiene. Para su numeración, podemos tener dos tipos: rango de piezas -pieza inicial y final- o conjunto arbitrario -números aleatorios-.
- Cuando hacemos un pedido de compra a un proveedor puede que este se demore y se produzca un **retraso**. De un retraso nos interesa saber de que pedido se trata, el motivo y el tiempo de retardo aproximado.
- Se puede dar el caso que el proveedor nos entregue un objeto erróneo o defectuoso, por lo que se producirá una **devolución**. En este caso, nos interesa saber a que proveedor se le realiza la devolución, la fecha de devolución, la fecha de llegada -del objeto descambiado- y almacenar los posibles gastos que se hayan producido.
- Se necesitará también almacenar la información de un **proyecto**. Un proyecto consta de un identificador, una fecha inicial y final, el *identificador de presupuesto* con el que se le asocia, las fases de las que se compone el proyecto -son las operaciones necesarias para fabricar el artículo- y un *estado*: iniciado, finalizado o cancelado. También se registrarán los **gastos asociados** a la fabricación del artículo que no se previnieron en el escandallo del artículo.
- Los gastos asociados pueden ser de dos tipos: gastos de transporte y otros. Se necesita saber si el gasto era previsto o por el contrario es un gasto extra no previsto como por ejemplo, el coste de transporte de una devolución.
- Los **gastos de transporte** se componen de una *fecha*, *proveedor del servicio*, *dirección destino-origen* y *coste*. Los gastos de transportes se producirán cuando la empresa envíe el/los objetos al proveedor para que realice las operaciones necesarias y cuando el proveedor nos devuelva el producto acabado.
- Los **gastos "otros"** se componen de una fecha, descripción y coste. De esta manera se pueden registrar costes varios como por ejemplo, un incremento de costes por variación de las tarifas del proveedor, etc.



4

4. Diseño

Este capítulo contiene

- Diagramas de clases y secuencia
- Interfaz gráfica de usuario

4.1. Diagramas de clases

4.1.1. *Package* ERP

Este *package* contiene todos los módulos principales del sistema. En cada uno de estos módulos están implementadas las clases relacionadas con las funciones específicas de cada uno. (*Para ver en más detalle los diferentes package consultar el anexo*).

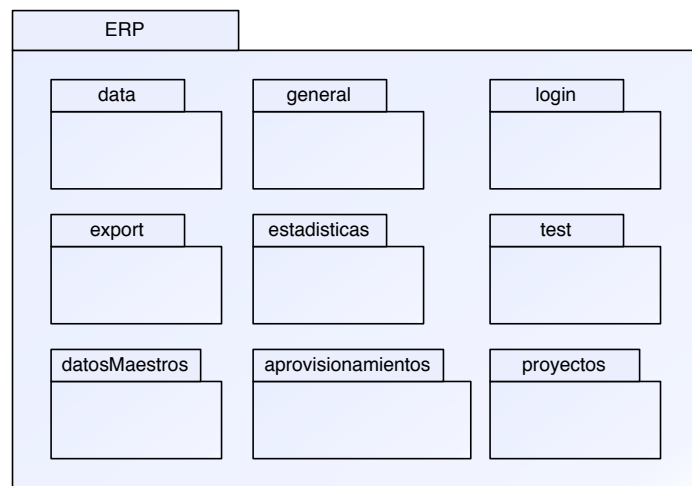


Figura 20: Diagrama de clase package ERP

4.1.2. *Package* data

El package data contiene todas las clases del modelo lógico de la base de datos. Para evitar la redundancia sólo se ha incluido en el diagrama la clase **SessionFactory**. Esta clase es la encargada de establecer la conexión con la base de datos. Para su uso, se ha implementado el patrón *Singleton* para evitar tener varias sesiones abiertas de nuestra base de datos, evitando así posibles errores.

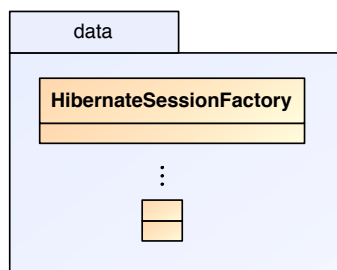


Figura 21: Diagrama de clase del package data

4.1.2.1. Package Empresa

El *package* empresa esta dentro del paquete “datos maestros” y desde aquí se gestiona todo lo relacionado con las empresas. En el diagrama **X** se muestra las clases utilizadas del *framework* Struts. En este caso se ha utilizado como controlador un *DispatchAction*. A diferencia del *Action* que explicábamos en el funcionamiento de Struts - capítulo X-, éste permite trabajar mejor cuando necesitamos un controlador que gestione diferentes peticiones, por ejemplo: sería inapropiado utilizar un *Action* diferente para cada función (alta, baja, modificación, eliminación...). Con esta clase podemos gestionar las diferentes funciones con un único controlador.

Por otra lado, también se ha utilizado la clase *SuggestAction* de la librería de *Struts Layout* -ver capítulo 2- que permite implementar un tipo de *Action* específico para el uso del componente de autocompletado -ver interfaz gráfica en el capítulo 4-.

El resto de clases y vistas JSP utilizadas siguen el funcionamiento de Struts explicado en el capítulo 2 “Funcionamiento de Struts”. Para ver el resto de *package* del sistema vea “Diagramas de clases” en el anexo.

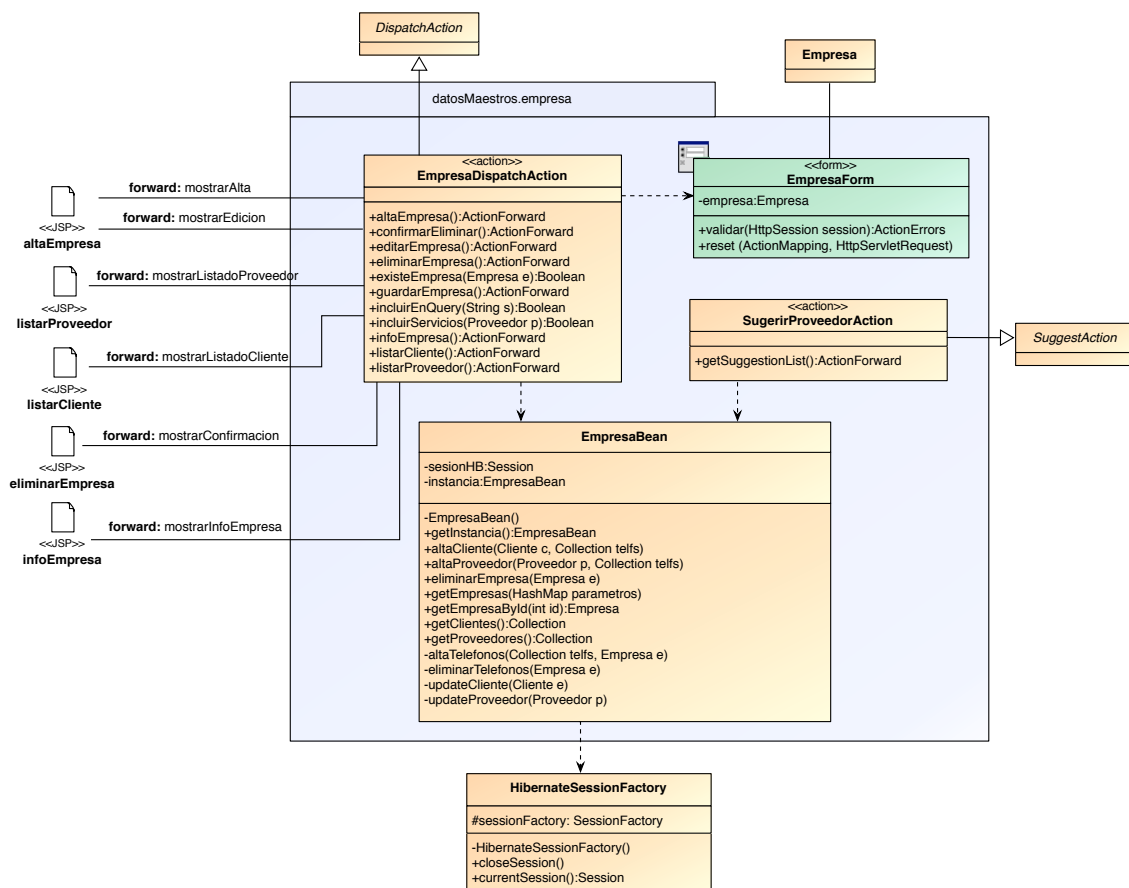


Figura 22: Diagrama de clase: Package Empresa

4.2. Diagrama de secuencia

Con la finalidad de entender mejor el funcionamiento del *framework* Struts se muestra un diagrama de secuencia sencillo donde se puede apreciar las relaciones entre los diferentes objetos desde que el usuario realiza una petición hasta que esta es procesada y se muestra una respuesta. Como ejemplo se ha utilizado el caso de uso “iniciar sesión”.

4.2.1. Inicio de sesión

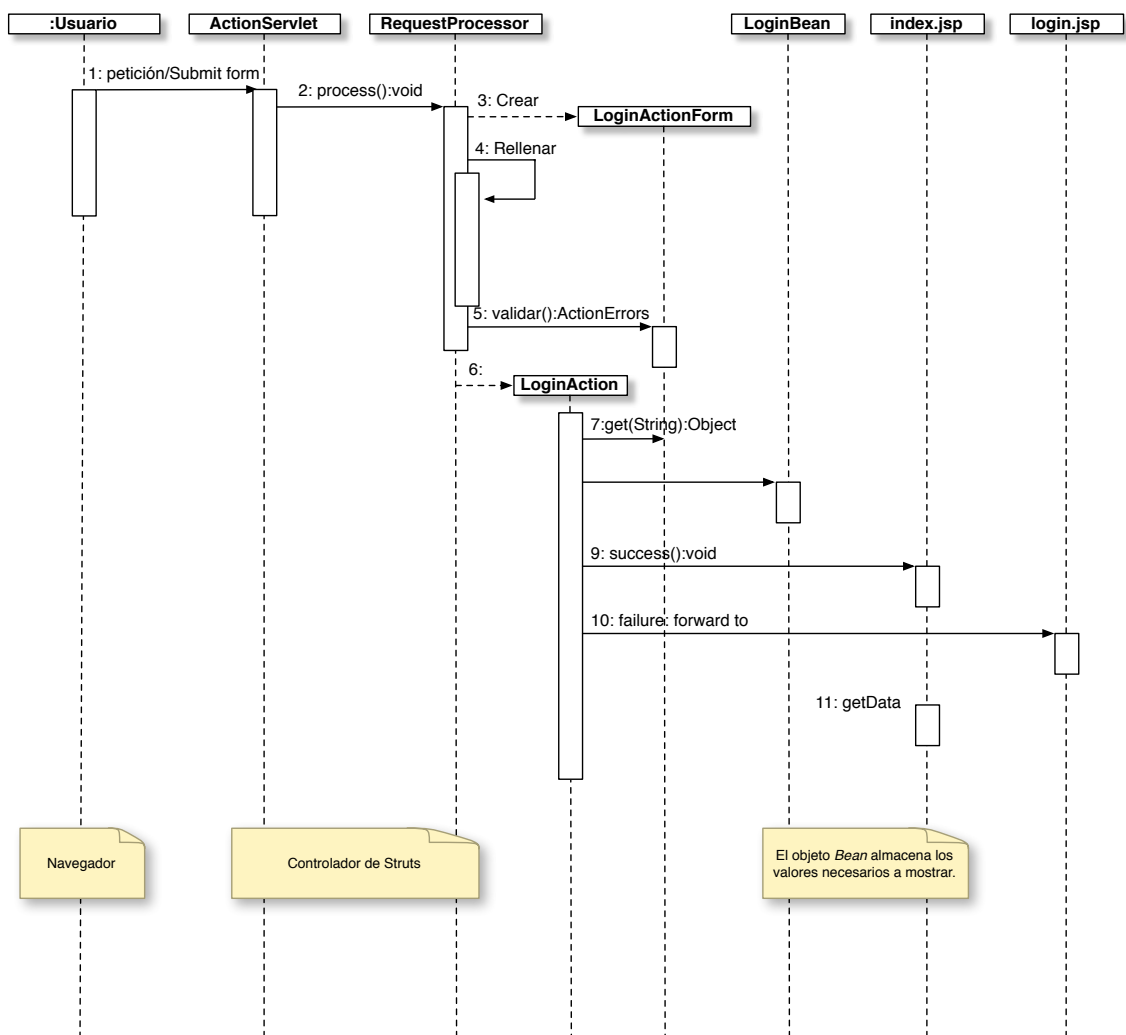


Figura 23: diagrama de secuencia: Inicio de sesión.

4.3. Interfaz gráfica de usuario

La interfaz de usuario del sistema está basada en páginas JSPs. Para lograr una interfaz rica en componentes y de fácil usabilidad se han utilizado diferentes técnicas. A continuación se detallan cada una de ellas y las ventajas que proporcionan.

Interfaz clara

Una de las cosas más importantes a tener en cuenta en el diseño de una interfaz de usuario es su facilidad de uso. Una buena GUI será aquella que facilite al usuario el uso de la aplicación. La interfaz no pretende eliminar la necesidad de aprendizaje por parte del usuario pero si reducir el tiempo a la hora de usar el sistema y evitar que el usuario dude sobre lo que la aplicación le está mostrando.

Por eso, se ha intentado realizar una interfaz clara y a la vez sencilla que permita localizar las funciones de manera ordenada. En la imagen 6 se muestra la pantalla principal del sistema una vez que el usuario a iniciado sesión. En ella podemos ver las diferentes secciones de la aplicación.

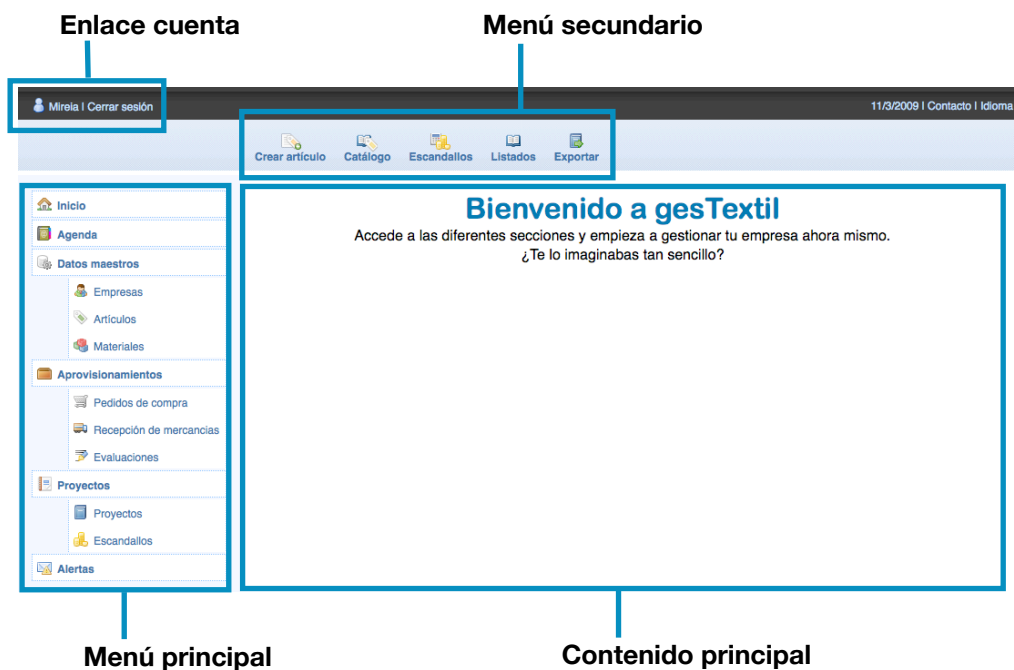


Imagen 6: Pantalla principal

En la parte superior se encuentra la información relacionada con nuestra cuenta. Desde allí podremos acceder a nuestro perfil, modificar nuestros datos y cerrar la sesión.

El resto de funcionalidades del sistema se encuentran repartidas en dos tipos de menús. El menú **principal** y el **secundario**.

Menú principal

Con la finalidad de facilitar la localización de las funciones de la aplicación, se ha optado por el uso de un menú principal de dos niveles. En cada nivel podemos encontrar las funciones más generales mostrando en cada una de ellas las más específicas. Para evitar tener un menú con infinitos niveles se ha utilizado un segundo menú que aparece en la parte superior mostrando las opciones del *ítem* que hayamos seleccionado.



Imagen 7: Menú principal

Menú secundario

Este es el menú que se encuentra en la parte superior de la Web. Este menú se muestra cuando accedemos a algunos de los *ítems* del menú principal -tal y como comentábamos antes-. En la imagen 8 podemos ver un ejemplo de este menú al seleccionar la opción **artículos** del menú principal - dentro del apartado *Datos Maestros*-.



Imagen 8: Ejemplo de menú secundario

Uso de CSS

El uso de CSS (*Cascade Style Sheet*) es fundamental y desde mi punto de vista, necesario para lograr una interfaz agradable. El uso de CSS nos permite aplicarle un estilo a nuestra aplicación de manera que si queremos modificar el estilo de algún elemento, por ejemplo, un determinado título, lo podemos hacer fácilmente sin necesidad de modificar texto a texto en la web.

Vamos a ver un ejemplo para entender su funcionamiento. Tenemos por ejemplo el siguiente fragmento de código en nuestra página JSP:

```
<h2> <fmt:key="titulo.formulario" /> </h2>
<div class="formulario">
    <html:form action="formularioAction">
        <html:text property="nombre"/>
    </html:form>
</div>
```

Figura 23: ejemplo de código usando CSS

En este ejemplo vemos que utilizamos el tipo de título `<h2>`⁵ y una clase de estilo llamada `formulario`. En el fichero CSS especificamos como se mostrarán estos elementos.

```
h2{
    color: blue;
    font-weight: 20px;
}
.formulario{
    color: gray;
    font-family: arial, tahoma, verdana;
}
```

Figura 24: ejemplo fichero CSS

De esta manera, cada vez que utilizamos la etiqueta `<h2>` o la clase de estilo `formulario` se aplicará el estilo definido en el fichero CSS al fragmento de la página que lo utilice. Si queremos modificar el formato de estilo de un elemento bastará con modificar el fichero y los cambios se verán reflejados en todas las páginas JSP que tengan referenciado el fichero CSS. Para usar el fichero CSS es necesario enlazarlo en cada página como se muestra a continuación:

```
<head>
    <link src="css/style.css"/>
</head>
```

Figura 25: Referencia al fichero css

⁵ Etiqueta reservada de html que permite utilizar un título

Existen otras maneras de usar CSS en las páginas pero la más adecuada en este caso es mediante un fichero externo y el enlace como hemos indicado. Otra opción es definir el estilo CSS directamente en la página mediante las etiquetas `<style>`. Este método podría ser útil en el caso de necesitar definir algo en concreto en una página, pero si lo que queremos es definir un estilo mucho más complejo y para ser utilizado en varias páginas, no resulta una buena opción. Si necesitáramos modificar algo tendríamos que hacerlo página por página y el mantenimiento resultaría una pérdida de tiempo.

Componentes ricos

Nuestro sistema se basa en gran parte en formularios de datos y en la consulta de datos. Es por eso que se han utilizado componentes gráficos que permitan tratar con la información de manera rápida y cómoda a la hora de visualizarla. Para ello se han hecho uso de diferentes componentes. A continuación se explican los diferentes tipos.

Tablas indexadas

Cuando realizamos una búsqueda y necesitamos mostrar muchos resultados, resulta de gran utilidad utilizar componentes como éste que permiten al usuario visualizar los datos de manera ordenada y clara. De esta manera evitamos que los resultados hagan crecer la Web hasta dimensiones realmente molestas.

13 proveedores encontrados. Mostrando del 1 al 10.

1, 2

Nombre	NIF	Dirección	Ciudad	País		
Alatros, S.L	454001212	c/Fuencarral, 150	Madrid	España		
Altaz, S.L	156045000	c/ Caja de Ahorros, 1	Torrent	España		
Artextil,S.A	321540000	c/Salgado, 22, bajos	Barcelona	España		
Bucarest,S.A	212454111	Plz. de la Riera, 34	Rubi	España		
Cardedeu torras, S.A	484212120	c/ Isabel de Villena, 14 - loc.	Granollers	España		
Cortex, S.A	121540014	c/ Joaquim Auger, 14	Barcelona	España		
Fextil Hnos., S.L	154877741	Rda. de Sant Pau, 76	Barcelona	España		
Industrias textil, S.L	414679981	c/ Arnau D'oms, 19	Barcelona	España		
Jiménez Textiles, S.A	445450005	c/Salamanca , 54	Madrid	España		
L. H. sartex, S.A	145440078	c/ Sicilia, 253	Terrassa	España		

Imagen 9: Ejemplo de tabla indexada

Calendarios

Con este tipo de calendarios evitamos que el usuario introduzca fechas erróneas y a la vez facilitamos la introducción de la misma. De esta manera podemos visualizar las fechas en forma de calendario, cosa que puede resultar muy útil en ocasiones.

Si en un formulario tenemos que introducir varias fechas o simplemente un par puede resultar tedioso el uso de desplegables, ya que supone más tiempo que hacer uso de un calendario de este tipo. De todas formas, este componente permite introducir la fecha de las dos maneras.

The image shows a web form for date selection. It has three dropdown menus for 'Dia' (Day), 'Mes' (Month), and 'Año' (Year). The 'Dia' is set to 12, 'Mes' to 'Abr' (April), and 'Año' to 2009. Below these is a text input field labeled 'Artículo' containing 'Artículo A'. To the right of the date fields is a calendar widget for April 2009. The calendar shows days of the week (L, M, MX, J, V, S, D) and dates from 1 to 30. The date 12 is highlighted with a red border.

Imagen 10: Ejemplo de calendario

Auto completables


Otro componente que se ha utilizado es el auto completable. Esto nos facilita la entrada de un campo en el formulario. Por ejemplo, si miramos la imagen 11 vemos que en campo necesitamos introducir el nombre de un artículo existente. Al empezar a escribir el nombre de un artículo se nos muestra un listado de las posibles opciones. De esta manera, podemos introducir los datos de manera más rápida en caso de no acordarse del nombre exactamente almacenado.

The image shows an auto-complete dropdown menu. It has a label 'Artículo' above a text input field. The input field contains 'Artículo A'. Below the input field is a dropdown list showing three options: 'Artículo A', 'Artículo A', and 'Artículo proto'. The first two options are highlighted with a blue background.

Imagen 11: Ejemplo de auto completable

Listas con scroll

En algunos formularios de nuestro sistema, es necesario la selección de varios objetos y una manera de visualizarlos para que el usuario los seleccione es mediante el uso de listas con *scroll*. En la imagen de abajo podemos ver un ejemplo de esto.

 **Alta de artículo**

Rellene los formularios para definir un nuevo artículo. Para finalizar, pulse sobre el botón aceptar.

Datos generales	Fases	Trama	Urdimbre
Composición de la trama <div> <input type="text" value="0.0"/> <input type="text" value="0.0"/> </div> <div> <small>Nº pasadas</small> <small>Coefficiente de ligadura</small> </div> <div> <div> <div>Hilos</div> <div> BAUSCHGARN PA SGD Garment Dyeing Saba Cor Saba-Saba C Sabaflex Serafil Strongfil TechX </div> </div> <div> <div><<</div> <div>>></div> </div> <div> <div>Hilos seleccionados</div> <div> Rasant-metrofil Bright Floss 440 Corneta </div> </div> <div> <div>Detalles del hilo</div> <div> Nombre Garment Dyeing Longitud 5000.0 Color - Brillo - Vueltas 1.0 Nº de cabezas 2 Sentido </div> </div> </div> <div> <div>Guardar</div> </div>			

Imagen 12: Ejemplo de listas con scroll

En este caso nos encontramos en la alta de un artículo donde tenemos que especificar una trama. Para ello debemos rellenar los atributos y seleccionar los hilos que la componen. Si mostráramos un lista de los hilos sin utilizar un *scroll* podríamos obtener una lista demasiado larga - ya que en el sistema pueden haber dados de alta muchos hilos-. Para facilitar la selección de los elementos, se ha utilizado otro componente que nos permite visualizar la información del elemento del cual estamos apuntando con el ratón. En el ejemplo se trata del hilo “Omega simple” y podemos ver sus propiedades en la parte derecha del formulario.

Uso de pestañas

Otro componente bastante útil en formularios donde se necesita introducir muchos datos es el uso de pestañas. Cuando tenemos formularios muy grandes resulta cómodo clasificar los campos en varias pestañas. De esta manera reducimos la longitud de la página y facilitamos al usuario la visualización de los datos.

Alta de hilo

Datos generales **Propiedades** Proveedores

Especifique las propiedades del hilo.

Longitud (en metros)

Color Brillo

Vueltas Nº de cabezas

Guardar

Imagen 13: Ejemplo de pestañas

5

5. Implementación

Este capítulo contiene

- Entorno de desarrollo
- Funcionamiento de Hibernate
- Funcionamiento Display Tag
- Configuración de Struts

5.1. Entorno de desarrollo

Una vez escogido el lenguaje de programación busqué un IDE adecuado que proporcionara las herramientas necesarias para el desarrollo de la aplicación y que facilitará las tareas más comunes como el *debug*.

Existen varios IDEs para el desarrollo de aplicaciones web pero no todos resultan igual de útiles. En este caso, me decidí por el uso de Eclipse por las siguientes razones:

- Permite la generación de los métodos *Getters* y *Setters* de manera automática. Lo cual es muy útil si una clase tiene muchos atributos.
- Auto completa el código en varias ocasiones:
 - Cuando abrimos un paréntesis o una llave lo cierra automáticamente. Además, si seleccionamos un paréntesis nos indica donde se cierra.
 - Cuando accedemos a los métodos de una clase (*en uso en muchos IDEs*).
- Permite indentar el código automáticamente.
- Permite generar diferentes vistas. Una vista consiste en la distribución de ventanas de manera que faciliten el uso de ellas según la ocasión. Por ejemplo, para *debuggear* resulta más cómodo utilizar una distribución de ventanas donde se muestren las variables que se inspeccionan, los *breakpoints* o la consola.
- Esta basado en un sistema de plugins, lo cual permite integrar muchas herramientas útiles para el desarrollo. En este caso se ha hecho uso del plugin **MyEclipse** para desarrollar páginas web, permitiendo soporte para proyectos webs basados en el framework Struts con reconocimiento de sintaxis.

5.1.1. Estructura del directorio

Cuando se desarrollan aplicaciones web, estás deben tener una estructura mínima obligatoria. A partir de esa estructura se puede personalizar con carpetas y subcarpetas que sean necesarias pero siempre respetando la jerarquía.

En el directorio principal deben haber todas las páginas -en nuestro caso JSP- y una carpeta llamada *WEB-INF*. Dentro de esta carpeta es donde se guardan las librerías que utilizamos y el fichero *web.xml* necesario para que la aplicación funcione.

En esta misma carpeta, también se introducirán el resto de ficheros que necesiten estar ahí, como por ejemplo, las librerías de tags y otros ficheros XML como el de *struts-config.xml*.

Todas las clases Java se deben guardar en la carpeta *src*, situada en el directorio principal. Ahí residen los diferentes *packages* que contienen todas las clases necesarias para la correcta ejecución de la aplicación.

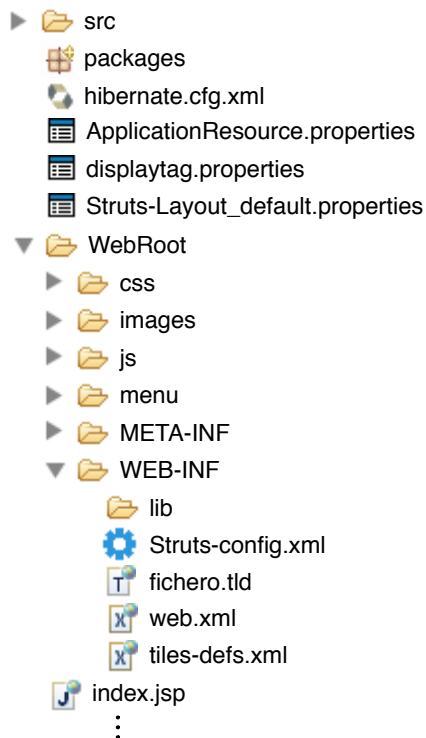


Imagen X: Estructura del directorio del sistema

En la imagen X se muestra la jerarquía de directorios del sistema. A continuación se especifica con más detalle el contenido de cada uno.

- **src:** Contiene los *package* que contiene las clases Java. También incluye el fichero de configuración de Hibernate y otros ficheros de configuración de las diferentes librerías de tag.
- **css:** contiene todos los ficheros de estilos que se utilizan en el sistema.
- **js:** contiene los ficheros javascript que se utilizan en el sistema.
- **menu:** contiene todas los menús secundarios que se utilizan en el sistema.

5.2. Funcionamiento de Hibernate

La herramienta Hibernate nos permite detallar como es nuestro modelo de datos, qué relaciones existen y que forma tienen. El funcionamiento es sencillo. A partir de ficheros XML especificamos que tablas de la base de datos hay que acceder, y que columnas de una tabla usar. De esta manera cuando queremos obtener los datos de un objeto, Hibernate sabrá en que tabla debe buscar y a que tipo de objeto corresponde para poder ser cargado en él.

En la figura 26 se muestra un ejemplo de la estructura básica de estos ficheros de mapeo en XML.

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
[... ]
</hibernate-mapping>
```

Figura 26: Estructura del fichero XML

A modo de ejemplo vamos a ver como se realizaría en nuestro caso la especificación del fichero XML para el objeto *usuario* de nuestro sistema.

Disponemos de la clase **Usuario.java** que representa la información de un usuario. Esta clase contendrá los siguientes atributos:

```
private Integer id;
private String nombre;
private String password;
```

Para cada uno de los atributos será necesario implementar los métodos *getter* y *setter* que servirán para que Hibernate pueda rellenar el objeto una vez lea la información de la base de datos.

Para relacionar un objeto con una tabla de la base de datos se utiliza la etiqueta `<class>`. En los atributos de la etiqueta detallamos el nombre de la clase y el nombre de la tabla con el que lo relacionaremos. En la figura 27 se muestra como relacionamos la clase **Usuario** con la tabla **USUARIO**.

```
<hibernate-mapping Packaged="data">
  <class name="Usuario" table="USUARIO">
    <id name="id" column="id" type="java.lang.Integer">
      <generator class="native"/>
    </id>
    <property name="nombre" column="nombre" type="java.lang.String"/>
    <property name="password" column="password" type="java.lang.String"/>
  </class>
</hibernate-mapping>
```

Figura 27: Fichero usuario.hbm.xml: ejemplo de relación objeto-tabla.

No sólo tenemos que relacionar el objeto con la tabla sino que también tendremos que hacer lo mismo con cada atributo de la clase **Usuario**.

Como ya sabemos, en cada tabla de la base de datos hay que especificar un campo como clave primaria (*PK*). Esto debe reflejarse en el fichero XML utilizando la etiqueta `<id>`. En la propiedad *name* de la etiqueta ponemos el nombre del atributo del objeto con el que lo queremos relacionar, en este caso "id". En la *column* especificamos el nombre del campo de la tabla **USUARIO** que en este caso se llama igual que el atributo y por último especificamos de que tipo es el atributo "id" de la clase **Usuario**. Lo mismo tenemos que hacer para el resto de los atributos pero para estos utilizando la etiqueta `<property>`.

Hasta aquí hemos especificado relaciones sencillas entre atributos de un objeto y campos de una tabla. Pero en este caso necesitamos otra relación que nos permita reflejar una clave foránea entre la tabla **USUARIO** y la tabla **GRUPO**, ya que un usuario pertenece a un grupo específico (administrador, usuario, ...). Para realizar esto utilizamos la etiqueta `<many-to-one>` que indica el tipo de relación entre una tabla y otra.

❗ En los ejemplos no se incluye la cabecera del fichero XML visto en la figura X para evitar exceso de código, pero es necesario en todos los casos incluirla.

Este proceso se tendrá que realizar para cada clase de nuestra aplicación que queramos almacenar en la base de datos y nombrar el archivo con la extensión *hbm.xml*.

Hay que destacar que la definición del mapeo en los ficheros XML sólo será necesario realizarla una única vez. Posteriormente, Hibernate se encargará automáticamente de realizar las sentencias SQL necesarias para almacenar y cargar datos.

Finalmente, es necesario crear un fichero con el nombre *hibernate.cfg.xml* donde le indicamos a Hibernate todos los ficheros de mapeo que debe utilizar y la configuración necesaria para la conexión con la base de datos. En la siguiente figura 28 se muestra las etiquetas necesarias para la configuración.

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 2.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-2.0.dtd">

<hibernate-configuration>
  <session-factory>
    <property name="connection.username">root</property>
    <property name="connection.password"></property>
    <property name="connection.url">jdbc:postgresql://Localhost/ERP</property>
    <property name="connection.driver.class">org.postgresql.Driver</property>
    <property name="dialect">net.sf.hibernate.dialect.PostgreDialect</property>

    <mapping resource="data/usuario.hbm.xml" />
    <mapping resource="data/empresa.hbm.xml" />
    ...
  </session-factory>
</hibernate-configuration>
```

Figura 28: Ejemplo fichero *hibernate.cfg.xml*

Hibernate proporciona un gran número de propiedades para poder realizar su configuración, pero sólo nos centraremos en las necesarias para obtener un buen funcionamiento.

Observemos las etiquetas que hay dentro de **<session-factory>**. Las dos primeras propiedades especifican datos de autenticación para la base de datos. La tercera especifica la cadena de conexión que utilizará Hibernate para acceder a la base de datos, en nuestro caso para PostgreSQL. Las otras dos que le siguen son para indicar el *driver* de conexión a la base de datos -en nuestro caso PostgreSQL- y el dialecto de la base de datos que utilizaremos, necesaria para que Hibernate pueda traducir correctamente los tipos de datos y sintaxis de las consultas.

En el siguiente esquema podemos ver como funciona Hibernate de manera resumida.

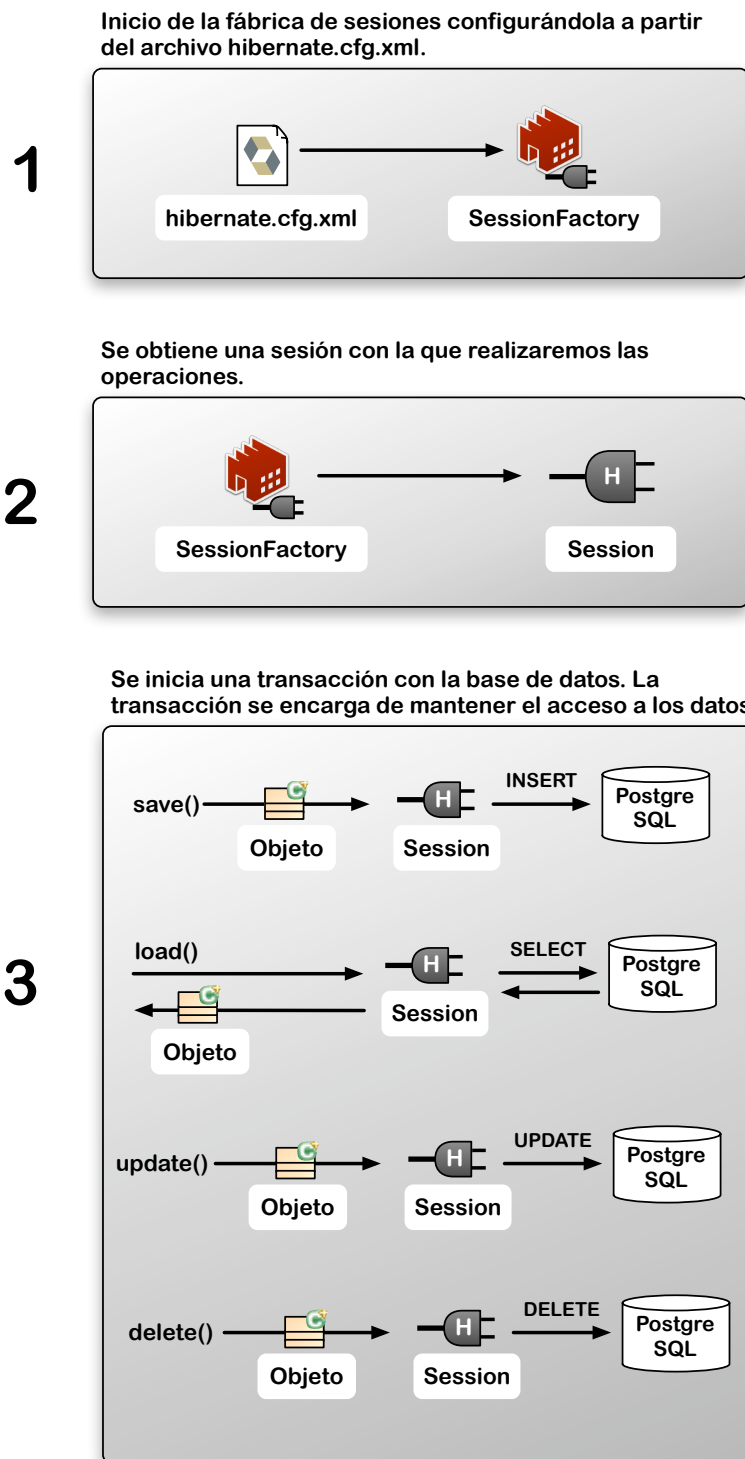


Figura 29: Funcionamiento Hibernate

En el punto 3 se muestran algunas operaciones que se pueden realizar mediante el objeto *Session*.

5.2.1. Mapeo de objetos

El uso de esta herramienta ha supuesto muchas ventajas pero no ha sido fácil realizar el correcto mapeo de las diferentes tablas de la base de datos. El ejemplo visto en la figura 29 es un caso muy sencillo pero cuando existen relaciones más complejas el mapeo se complica. Al desconocer el uso de esta herramienta, esta tarea a supuesto una inversión de tiempo mayor a la esperada pero a medida que se ha ido avanzando en el proyecto, se han ido solventando todos los casos que se han ido dando. Vamos a ver algunos de ellos y la explicación de su funcionamiento.

5.2.1.1. Identificador *autoincrement*

En nuestro modelo de base de datos casi todos los identificadores de claves primarias son de tipo numérico y autoincrementable. En PostgreSQL no existe un tipo nativo que permita generar identificadores autoincrementables pero si que proporciona funciones que permiten conseguir este propósito. Para su uso es tan sencillo como utilizar el tipo de datos **serial** -aunque existen otros como *bigserial*-. En Hibernate, la declaración de la clave primaria se realiza de la siguiente manera:

```
<hibernate-mapping package="data">
  <class name="Empresa" table="empresa">
    <id name="id" column="id" type="java.lang.Integer">
      <generator class="native">
        <param name="sequence">empresa_id_seq</param>
      </generator>
    </id>
    ...
  </class>
</hibernate-mapping>
```

Figura 30: ejemplo de identificador *autoincrement*

La etiqueta `<id>` permite definir el campo de la tabla que será la clave primaria. La etiqueta `<generator>` se encarga de indicar que tipo de generador utilizará Hibernate para trabajar con la clave primaria. Algunos de los tipos que existe son:

- **sequence:** Utiliza una secuencia en PostgreSQL. El identificador devuelto es de tipo `long`, `short` o `int`. En nuestro caso es el que utilizaremos por lo que hemos mencionado antes.
- **assigned:** La aplicación asigna el identificador al objeto antes de hacer el `save()`.
- **identity:** Soporta columnas identidad en DB2, MySQL, MS SQL Server.
- **native:** Comprueba el uso de `identity`, `sequence` o `hilo` dependiendo de las características de la base de datos.

En el ejemplo vemos que la etiqueta `<generator>` contiene la propiedad `<param>`. Ésta es para indicar el nombre del parámetro que utilizará la función de PostgreSQL.

5.2.1.2. Herencia

Para ver el uso de herencia en Hibernate, vamos a centrarnos en un ejemplo de la aplicación. Para ello nos fijamos en el objeto *Empresa* el cual tiene dos clases hijas *Cliente* y *Proveedor*. En nuestra base de datos tenemos las siguientes tablas declaradas de la siguiente manera:

```
CREATE TABLE empresa
(
  id serial NOT NULL,
  pais character varying(80),
  CONSTRAINT empresa_pkey PRIMARY KEY (id),
  CONSTRAINT empresa_pais_fkey FOREIGN KEY (pais)
    REFERENCES pai_pais (pai_nombre),
  CONSTRAINT empresa_nif_key UNIQUE (nif)
)

CREATE TABLE proveedor
(
  id serial NOT NULL,
  CONSTRAINT proveedor_pkey PRIMARY KEY (id),
  CONSTRAINT proveedor_id_fkey FOREIGN KEY (id)
    REFERENCES empresa (id)
)

CREATE TABLE cliente
(
  id serial NOT NULL,
  CONSTRAINT cliente_pkey PRIMARY KEY (id),
  CONSTRAINT cliente_id_fkey FOREIGN KEY (id)
    REFERENCES empresa (id)
)
```

Vemos que las tablas *Proveedor* y *Cliente* tienen el mismo identificador que la clase madre *Empresa*. Para realizar esto, hacemos que el atributo *id* sea clave primaria de la tabla y a la vez clave foránea de la tabla *Empresa*.

❗ En el ejemplo utilizado obviamos el resto de campos de las tablas (excepto el "id") para reducir el código y centrarnos en la parte que nos interesa, la herencia.

Cuando pasamos estos objetos a Java tenemos lo siguiente:

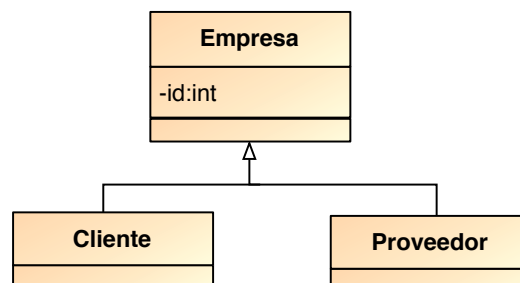


Figura 31: Ejemplo de herencia.

Como vemos, la clase *Cliente* y *Proveedor* heredan de *Empresa*. El paso de tablas a clases es sencillo, cada tabla corresponde a una clase del sistema.

La herencia en Hibernate es muy sencilla de mapear y se puede aplicar de dos maneras distintas:

- ▶ Utilizando un fichero para cada clase.
- ▶ Utilizando un único fichero para las tres clases.

En este caso, como las clases hijas no tienen demasiadas propiedades propias se ha optado por usar un único fichero. En la figura 32 podemos verlo.

```
<hibernate-mapping package="data">
  <class name="Empresa" table="empresa">
    <id name="id" column="id" type="java.lang.Integer">
      <generator class="native">
        <param name="sequence">empresa_id_seq</param>
      </generator>
    </id>
    <joined-subclass name="Proveedor" extends="Empresa">
      <key column="id"/>
      ...
    </joined-subclass>
    <joined-subclass name="Cliente" extends="Empresa">
      <key column="id"/>
    </joined-subclass>
  </class>
</hibernate-mapping>
```

Figura 32: Ejemplo fichero empresa.hbm.xml

Como vemos, utilizamos la etiqueta `<joined-subclass>` para indicar el nombre de la clase hija y el atributo `<extends>` para indicar de que clase deriva.

5.2.1.3. Relaciones Muchos-a-muchos (*Many-to-many*)

Siguiendo con el ejemplo anterior vamos a explicar las relaciones *many-to-many* del objeto *Proveedor* y *hilo*. Las tablas de la base de datos se declaran como se muestra a continuación.

```
CREATE TABLE proveedor
(
  id serial NOT NULL,
  CONSTRAINT proveedor_pkey PRIMARY KEY (id),
  CONSTRAINT proveedor_id_fkey FOREIGN KEY (id)
    REFERENCES empresa (id)
)

CREATE TABLE proveedor_hilo
(
  id_proveedor integer NOT NULL,
  id_hilo integer NOT NULL,
  precio real,
  CONSTRAINT proveedor_hilo_pkey PRIMARY KEY (id_proveedor, id_hilo),
  CONSTRAINT proveedor_hilo_id_hilo_fkey FOREIGN KEY (id_hilo)
    REFERENCES hilo (id_material),
  CONSTRAINT proveedor_hilo_id_proveedor_fkey FOREIGN KEY (id_proveedor)
    REFERENCES proveedor (id)
)

CREATE TABLE hilo
(
  id_material integer NOT NULL,
  tintado boolean NOT NULL DEFAULT true,
  propiedad_fk integer,
  CONSTRAINT hilo_pkey PRIMARY KEY (id_material),
  CONSTRAINT hilo_id_material_fkey FOREIGN KEY (id_material)
    REFERENCES material (id_catalogo),
  CONSTRAINT hilo_propiedad_fk_fkey FOREIGN KEY (propiedad_fk)
    REFERENCES propiedad (id)
)
```

Como podemos ver en las declaraciones, aparece una tercera clase de la relación entre *Proveedor* y *Hilo*. Esta clase es necesaria si queremos permitir la relación *many-to-many*. De esta manera, podemos saber de un proveedor que hilos suministra, y de un hilo saber que proveedores lo venden. En el diagrama de clase Java tenemos lo siguiente:

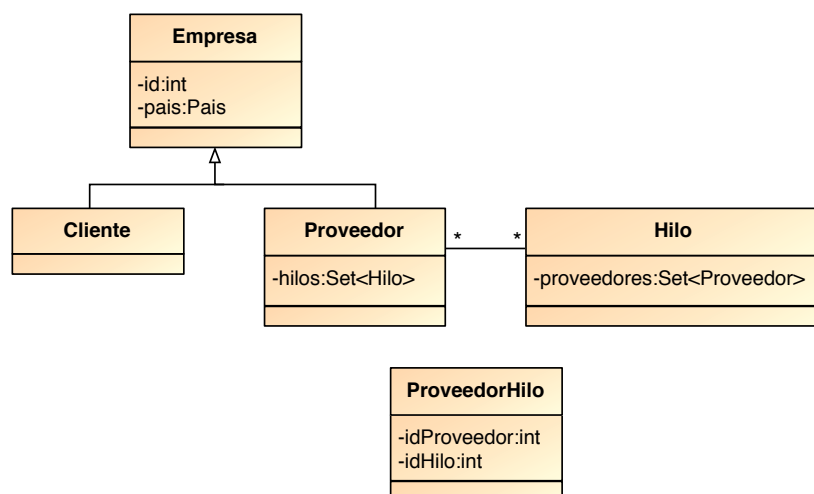


Figura 33: Ejemplo de relación *many-to-many*.

Si nos fijamos en el tipo de dato de la colección de hilos que tiene *Proveedor* o la colección de proveedores que tiene *Hilo* vemos que es de tipo `java.util.Set`. Quizás la opción más acertada sea utilizar un tipo como `java.util.ArrayList` que permite tener una colección de objetos ordenados. A diferencia del anterior, este permite hacer cosas como `get(index)` pero su uso con Hibernate no es tan sencillo. Para poder utilizar un tipo `java.util.ArrayList` o `java.util.Collection` sería necesario incluir una columna en la tabla que sirviera de índice, lo cual tendríamos que controlar los índices de manera correlativa constantemente. Descartando esta opción nos quedan otras como `Set`, `Bag`, `Array` o `Map`. En la figura 34 podemos ver el fichero que mostrábamos en el ejemplo anterior pero incluyendo las propiedades del proveedor.

```
<hibernate-mapping package="data">
  ...
  <joined-subclass name="Proveedor" extends="Empresa">
    <key column="id"/>
    <set name="hilos" table="proveedor_hilo" inverse="true" lazy="true">
      <key column="id_proveedor"/>
      <many-to-many class="Hilo" column="id_hilo"/>
    </set>
    <set name="servicios" table="proveedor_sc">
      <key column="id_proveedor"/>
      <many-to-many class="ServicioCatalogo" column="id_sc"/>
    </set>
  </joined-subclass>
  ...
</hibernate-mapping>
```

Figura 34: Ejemplo fichero empresa.hbm.xml

5.2.1.4. Relaciones Muchos-a-uno (*many-to-one*)

Otro tipo de relación entre tablas es la de uno a muchos. Un ejemplo de ello es la relación que existe entre la tabla *Empresa* y *País*. Una empresa pertenece a un sólo país y un país puede ser parte de una, muchas o ninguna empresa. En la figura **x** vemos la declaración de las tablas.

```
CREATE TABLE empresa
(
  id serial NOT NULL,
  pais character varying(80),
  CONSTRAINT empresa_pkey PRIMARY KEY (id),
  CONSTRAINT empresa_pais_fkey FOREIGN KEY (pais)
    REFERENCES pai_pais (pai_nombre),
  CONSTRAINT empresa_nif_key UNIQUE (nif)
)

CREATE TABLE pai_pais
(
  pai_pk serial NOT NULL,
  pai_isonum numeric(6) DEFAULT NULL::numeric,
  pai_iso2 character(2) DEFAULT NULL::bpchar,
  pai_iso3 character(3) DEFAULT NULL::bpchar,
  pai_nombre character varying(80) NOT NULL DEFAULT NULL::character varying,
  CONSTRAINT pai_pais_pkey PRIMARY KEY (pai_nombre)
)
```

El diagrama de clases en Java es el siguiente:

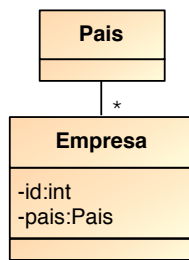


Figura 35: Ejemplo de many-to-one

El mapeo de esta relación es muy sencilla. A continuación podemos ver el ejemplo.

```

<hibernate-mapping package="data">
  <class name="Empresa" table="empresa">
    <id name="id" column="id" type="java.lang.Integer">
      <generator class="native">
        <param name="sequence">empresa_id_seq</param>
      </generator>
    </id>
    <many-to-one name="pais" class="Pais" column="pais"/>
    ...
  </class>
</hibernate-mapping>
  
```

5.2.2. Lenguaje HQL de Hibernate

El lenguaje HQL no es más que un lenguaje similar al SQL que permite realizar consultas a la base de datos. Hibernate utiliza este lenguaje específico y el se encarga de hacer el mapeo con los ficheros necesarios y interpretar la consulta HQL como si de una SQL se tratara. Vamos a ver como se lanza una consulta a la base de datos y como usar algunas de las cláusulas más comunes.

Para crear una consulta tendremos que hacer uso del objeto **Session** y **Query** de las librerías de Hibernate. El primero será necesario para abrir una sesión con la base de datos y poder ejecutar consultas y el segundo será necesario para poder invocarlas. Véase en la figura 36 un ejemplo donde se pretende obtener todos los clientes existentes en la base de datos. En la consulta se hace uso de la cláusula **from** y se le indica a que objeto se hace referencia, es decir, el tipo de objeto al que se le aplicará la consulta. Cliente se refiere a la clase **Cliente** y no a la tabla **CLIENTE** de la base de datos. Hibernate sabrá a que tabla corresponde la clase Cliente según la especificación que realizamos en el fichero *cliente.hbm.xml* anteriormente.

Una vez ejecutada la consulta tenemos que recoger el resultado de esta y para hacerlo hacemos uso del método **list()**, tal como muestra la segunda línea del ejemplo.

```
Query consulta = sesion.createQuery("from Cliente");  
List listaClientes = query.list();
```

Figura 36: Ejemplo de consulta

Cláusula WHERE

Esta cláusula tiene la misma función que la de SQL. Permite filtrar los resultados en base a unas condiciones. Si se desea hacer múltiples condiciones se hará uso de las palabras "and", "or" y "not".

```
from Cliente cliente where cliente.nombre like '%Juan%' and  
cliente.nif is not null
```

En el ejemplo mostramos todos los clientes que se llaman Juan y que el campo nif no esté vacío.

Cláusula SELECT

De igual manera que en SQL podemos utilizar la cláusula SELECT. Algunos ejemplos de su uso son:

```
select cliente.nombre from Cliente cliente  
select avg(producto.precio) from Producto producto
```

Estos son algunos ejemplos de las cláusulas que se utilizan en HQL pero en general se utilizan las mismas que en SQL.

5.2.3. Display tag: funcionamiento

En la figura 37 se muestra un ejemplo de su uso. En este caso se muestra un listado de clientes que contiene el *Bean* `listarClienteForm`. La propiedad `sortable=true` indica que la columna "nombre" se podrá ordenar alfabéticamente al pulsar sobre la columna. Podríamos hacer lo mismo para el resto de columnas si se cree necesario.

```
<display:table id="cliente" pagesize="10" name="listarClienteForm.clientes">
  <display:column title="Nombre" sortable="true">
    <bean:write name="cliente" property="nombre" />
  </display:column>

  <display:column title="NIF">
    <bean:write name="cliente" property="nif" />
  </display:column>

  <display:column title="Dirección">
    <bean:write name="cliente" property="direccion" />
  </display:column>

  <display:column title="Ciudad">
    <bean:write name="cliente" property="ciudad" />
  </display:column>

  <display:column title="País">
    <bean:write name="cliente" property="pais" />
  </display:column>
</display:table>
```

Figura 37: Ejemplo de uso de la tag *Display*.

5.2.3.1. Archivo de configuración de Struts

El archivo de configuración de Struts es un archivo XML en el cual se define exactamente cual de las subclases de **Action** debería ser usada, cuando y que **ActionForm** debería ser proporcionado por la subclase **Action**. Nosotros especificamos parte de la interacción del controlador en este fichero de configuración.

Cuando el controlador decide que vista debe mostrar la información al usuario, se escoge la vista que se especifico en el fichero. En definitiva, el fichero define las conexiones entre los componentes del MVC.

La principal ventaja del uso de este fichero es la posibilidad de cambiar las conexiones sin tener que modificar nuestro código.

6

6. Pruebas del sistema

Este capítulo contiene

- Seguridad
- StrutsTestCase
- Usabilidad

6.1. Seguridad

Las inyecciones de SQL -conocidas como *SQL injections*- vulneran la seguridad de las aplicaciones. Estos ataques consisten en insertar comandos SQL dentro de otro código SQL para alterar su funcionamiento normal sin necesidad de utilizar aplicaciones adicionales. Esta es una de las más peligrosas y extendidas vulnerabilidades que existen en las aplicaciones.

Supongamos el siguiente ejemplo: tenemos un formulario de *login* donde hay que introducir un usuario y una contraseña.

La consulta normal que se lanza a la base de datos para validar los datos introducidos es la que se muestra en la figura 38.

```
select * from Usuarios where NombreUsuario='$nombreUsuario' and  
password='$password'
```

Figura 38: Ejemplo de consulta

En el ejemplo se utiliza la concatenación de variables para hacer la consulta. Esta práctica no previene ningún tipo de ataque y tiene un alto riesgo de vulnerabilidad. Si un usuario mal intencionado introduce una *SQL injection* como la que se muestra en la figura 39, podría acceder a la aplicación como si hubiese hecho *login* correctamente cuando en realidad no debería poder acceder al sistema.

```
$nombreUsuario = 1' or '1' = '1  
$password = 1' or '1' = '1
```

Figura 39: Ejemplo de SQL injection

Cuando el usuario introduce esos valores en los campos de usuario y contraseña realmente vemos que la consulta que se lanzará a la base de datos no es la misma que la de la figura 40 sino que esta ha sido alterada permitiendo el acceso. Esto es debido a que la condición de la consulta ha cambiado y siempre será válida, ya que 1 siempre es igual a 1.

```
select * from Usuarios where NombreUsuario='1' or '1'='1'  
and password='1' or '1'='1'
```

Figura 40: Ejemplo de consulta alterada

6.1.1. Inyecciones en Hibernate

Una de las ventajas de Hibernate es el uso de capas ORM (*Object Relational Mapping*). Esta manera de trabajar permite evitar las inyecciones de SQL siendo inmune a ellas pero esto no quiere decir que con Hibernate no tengamos problemas con ataques de inyección de otros tipos. Hibernate incluye un *subset* de SQL llamado HQL -lenguaje explicado en el capítulo anterior) y este es el que permite realizar las consultas nativas de SQL. Debido a esto llegamos a la conclusión de que las capas ORM solo minimizan la manipulación de las consultas -ya que antes de procesarlas la base de datos es manejada por esta capa- pero no resuelve nuestro problema de vulnerabilidad.

Después de ver el riesgo que existe incluso con el uso de herramientas con capas ORM detectamos los siguientes casos donde se puede producir una vulnerabilidad:

- ▶ En consultas nativas de SQL que contienen información introducida directamente por el usuario -como en el ejemplo visto anteriormente-.
- ▶ En consultas generadas dinámicamente por Hibernate que no están sujetas a parámetros.
- ▶ Si se utilizan métodos de la API que usan como parámetros de la consulta un *String* los cuales son propensos de ataques, como por ejemplo:

`Session.iterate(String, ...)` o `Session.delete(String, ...)`

6.1.2. Prevenciones

Existen varias opciones para prevenir las inyecciones SQL. A continuación se enumeran tres alternativas:

- ▶ Opción 1: Parametrización de la consulta
- ▶ Opción 2: Almacenamiento de *procedures*
- ▶ Opción 3: Escaping all user supplied input

6.1.2.1. Parametrización de la consulta

Esta manera de generar consultas es mucho más simple y fácil de entender que las consultas dinámicas. Las consultas parametrizadas obligan al desarrollador a definir todo el código SQL primero y luego los parámetros que se pasan en la consulta. Este estilo de código permite a la base de datos distinguir entre código y datos a pesar de lo que el usuario haya introducido.

La parametrización de la consulta te asegura que un atacante no podrá cambiar una consulta incluso si los comandos SQL son insertados por él. Si volvemos al ejemplo anterior de *SQL injection* y el atacante introdujera los mismos valores, la consulta no podría ser vulnerada porque de esta manera realmente se buscaría un nombre de usuario que fuera igual al *String* completo (`1'` or `'1' = '1`).

En nuestro caso, la mejor opción es el uso de la parametrización. Hibernate permite realizar esta tarea de manera efectiva y sobretodo permite un buen mantenimiento del sistema. Para ver como funciona este método en Hibernate vamos a ver un ejemplo.

```
Query consulta = session.createQuery("from Usuarios u where  
u.NombreUsuario = ' " + nombreUsuario + "'");
```

Figura 41: Ejemplo de consulta en HQL (Hibernate Query Language)

En este ejemplo se muestra una consulta normal en Hibernate donde se concatena la variable recogida del formulario para comprobar el nombre de usuario. Para hacer uso de la parametrización utilizaríamos el siguiente método:

```
Query consulta = session.createQuery("from Usuarios u where  
u.NombreUsuario =:nombreUsuario");  
consulta.setParameter("nombreUsuario", nombre);
```

Figura 42: Ejemplo de consulta en HQL usando parametrización

En vez de concatenar la consulta con la variable que contiene el nombre, lo que hacemos es parametrizarla definiendo un nombre. Con el objeto **Query** lo que hacemos es indicarle el identificador de la variable como primer parámetro y la variable que contiene el nombre del usuario que queremos buscar como segundo parámetro.

6.1.2.2. Almacenamiento de *procedures*

El almacenamiento de procesos tiene el mismo efecto que la parametrización de la consulta. También se requiere que el desarrollador defina primero el código SQL y luego el paso de los parámetros. La diferencia entre uno y otro es que el código SQL para el almacenamiento de los procesos es definido y almacenado en la propia base de datos y por lo tanto, llamado desde la aplicación. En cambio la parametrización de la consulta se define en la propia aplicación. Ambas técnicas tienen el mismo grado de efectividad contra las inyecciones SQL, por lo tanto dependerá de nosotros el uso de una u otra.

6.1.2.3. Escape de los datos introducidos por el usuario

Esta técnica tiene la ventaja de poder utilizarse en cualquier aplicación sin afectar a su estructura. Su funcionamiento es el siguiente: Cada sistema gestor de base de datos soporta un esquema de caracteres de escape diferente. Esto permite indicarle a nuestra base de datos que caracteres deben tratarse como datos en vez de código. De esta manera, la base de datos no confundirá los datos introducidos como parte de la consulta y se evitara la inyección SQL.

6.2. StrutsTestCase

StrutsTestCase es una extensión del estándar JUnit TestCase que proporciona las facilidades necesarias para hacer pruebas en una aplicación basada en el *framework* Struts. Esta herramienta ejecuta el **ActionServlet** permitiendo que podamos realizar pruebas de nuestro código sin necesidad de ejecutar el *Servlet*. Gracias a esto podemos comprobar no solo la implementación de nuestros objetos **Action** sino también de los *mapeos*, *form beans* y las declaraciones de *forward*.

Después de finalizar la fase de implementación del sistema se han estudiado que módulos y clases deberían de ser testadas para detectar posibles errores. Por esta razón se ha creído conveniente hacer un juego de pruebas para las clases de inicio de sesión, módulo de proyectos y para los pedidos, tanto de compra como de venta.

6.2.1. Test: inicio de sesión

Las pruebas que se han realizado para el inicio de sesión han consistido en: comprobar que el usuario sea almacenado correctamente en la sesión, que no se produjesen errores en el **Action** y comprobar que los *forward* se realizan correctamente.

Lo más importante de este caso es asegurarse que los *forward* se hacen bien, ya que de no ser así un usuario podría acceder al sistema sin haber hecho un *login* correcto. También se ha tenido en cuenta el tema de las inyecciones en SQL y se ha hecho uso de la parametrización de la consulta. En la figura 43 se muestra el código que se ha empleado para realizar el test y en la imagen 14 el resultado obtenido.

```
public void testSuccessfulLogin(){
    setRequestPathInfo("/login");
    addRequestParameter("userName","mireia");
    addRequestParameter("password","1234");
    setActionForm(lf);
    actionPerform();
    assertSame("mireia",String)getSession().getAttribute("autenticacion"));
    verifyForward("success");
    verifyNoActionErrors();
}
```

Figura 43: Código del test: inicio de sesión

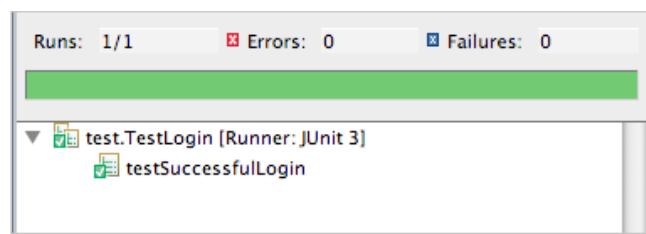


Imagen 14: Resultados del test: inicio de sesión

6.2.2. Test: gestión de proyectos

Este es uno de los módulos más importantes del sistema, ya que desde aquí se gestiona gran parte de los recursos. Realizar pruebas en este caso es fundamental y se ha optado por testar los siguientes casos de usos:

- ▶ Escoger un pedido de venta para iniciar un proyecto. Hay que asegurarse que el pedido este aceptado por el cliente y que no tenga ya un proyecto iniciado.
- ▶ Estados de un proyecto. Asegurarse que los estados de un proyecto son correctos según los datos relacionados con él. Un proyecto no puede estar finalizado si algunas de sus fases no estan finalizadas.
- ▶ Consulta de proyectos. Asegurarse que al consultar la información de un proyecto es la del proyecto correcto.

En este caso de prueba existen diferentes test y para hacerlos se han hecho diferentes casos englobados en una suite de test. El resultado del test se puede ver en la imagen 15.

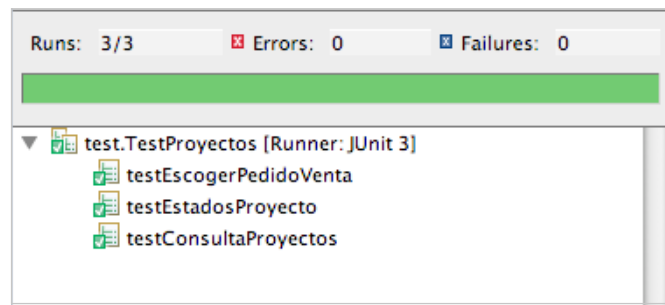


Imagen 15: Resultados del test: gestión de proyectos

6.2.3. Test: pedidos

Las pruebas a realizar sobre los pedidos se basa en los siguientes puntos:

- ▶ Comprobar el estado. Si un pedido ha finalizado tiene que tener fecha final.
- ▶ Alta pedido compra. No se puede crear un pedido de un artículo que no tiene escandallo.

Los resultados se muestran en la imagen 16.

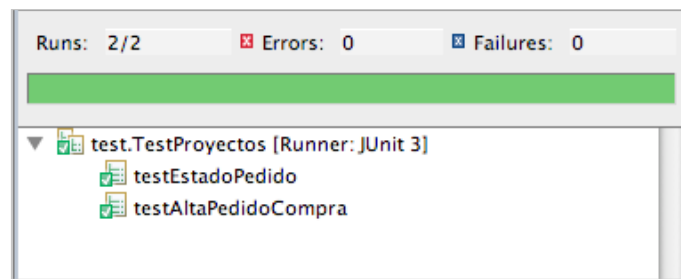


Imagen 16: Resultado del test: pedidos

6.3. Usabilidad

Otra de las cosas importantes a probar es la propia usabilidad del sistema. A medida que se ha ido desarrollando el sistema se han hecho una serie de prototipos que han sido probados para determinar si la facilidad e intuitividad del sistema eran los adecuados.

Después de varias pruebas y con el sistema finalmente acabado se han obtenido los siguientes resultados:

Navegación

★★★★	Rápido acceso	Acceso a la información evitando múltiples clicks.
★★★★	Jerarquía de menús	Menús organizados según temática.
★★☆☆	Uso de text links	Información en los links de la aplicación.
★★★★	Estructura del <i>layout</i>	Estructura de la interfaz gráfica de la aplicación.

Diseño

★★★★	Interfaz	Interfaz intuitiva y clara.
★★★★	Iconos y texto	Iconos de menú y opciones acorde con su función.
★★★★	Uso de componentes	Componentes que facilitan el uso del sistema.
★★★☆☆	Formularios	Formularios que permiten rellenarlos de manera cómoda.

Las tablas que se muestran son las valoraciones de varios usuarios que han probado la aplicación. En general, la valoración ha sido muy positiva incluso en aquellos usuarios con poco conocimiento informático -a nivel de usuario normal- que han encontrado la aplicación intuitiva y de fácil manejo.



7. Conclusiones y trabajo futuro

Este capítulo contiene

- Conclusiones
- Trabajo futuro

7.1. Conclusiones

Después de finalizar el proyecto y de realizar las pruebas del sistema se ha obtenido un producto que ha cumplido con los objetivos marcados en un inicio. Durante su desarrollo han habido varios problemas que en un principio no se creía que lo fueran a ser. Con esto me refiero al aprendizaje de las tecnologías utilizadas en el desarrollo del sistema y sobre todo a su uso y configuración.

La principal conclusión a la que he llegado con este proyecto es la siguiente: siempre que se inicie un proyecto es necesario planificar hasta el último recurso y prever las posibles dificultades o problemas basándose en experiencias anteriores o similares. Es de vital importancia no empezar la casa por el tejado y es que es muy habitual querer empezar a programar sin antes haber realizado el correcto análisis y diseño del sistema. Hacer las cosas en el orden adecuado realmente evita problemas graves a la hora de implementar el código y sobre todo en el funcionamiento de la propia aplicación.

7.2. Desviaciones

En general se ha seguido la planificación inicial del proyecto pero en algunas etapas el tiempo planificado no ha sido suficiente y se han producido demoras. Básicamente, los mayores retrasos se han producido en las siguientes etapas:

► **Especificación UML:** En esta fase de especificación del sistema se han producido grandes retrasos a causa de la tecnología utilizada. El uso del *framework Struts* juntamente con el resto de librerías usadas han requerido un tiempo de aprendizaje superior al previsto. A causa del desconocimiento, se ha invertido mucho tiempo en la integración y corrección de errores que de tener experiencia con el *framework* podrían haber sido resueltos más rápidamente.

► **Desarrollo de la base de datos:** El desarrollo de la base de datos ha supuesto el mismo problema que el mencionado anteriormente. En este caso la causa ha sido del uso de la herramienta Hibernate. Esta herramienta proporciona muchas ventajas pero por la falta de experiencia y uso han surgido muchos problemas de configuración y implementación que han supuesto una demora de unas dos semanas.

A continuación se muestra una tabla de desviaciones donde se aprecia la demora de tiempo para cada fase.

Fase	Tiempo planificado	Tiempo real
Análisis de requerimientos	130,25	130,25
Diseño		
Diseño de la base de datos	32	126
Diseño del sistema	125	219
Implementación		
Desarrollo de la base de datos	16	40
Módulo de gestión de datos	80	80
Módulo de gestión de aprovisionamientos	80	85
Módulo de gestión de proyectos	120	140
Pruebas	96	100
Documentación	120	140

Aunque han habido retrasos importantes, en otras etapas se ha trabajado más rápido y esto ha permitido implementar otras funciones que inicialmente no estaban previstas de ser implementadas. Algunas de ellas son:

- ▶ Inclusión de gráficos en las estadísticas
- ▶ Componentes gráficos fácil y cómodos de usar
- ▶ Exportación de documentos a formato PDF

7.3. Trabajo futuro

En general, todo el trabajo planificado se ha cumplido en mayor o menor medida aunque existen algunas funciones que podrían mejorarse y otras podrían ser añadidas. Por ejemplo, al sistema se le podrían añadir formularios de búsqueda más ricos incluyendo más posibles parámetros de búsqueda lo que permitiría acotar mejor los resultados.

Otro de los puntos pendientes es el desarrollo del módulo del control de *stock*, el cual permitiría llevar el control de todos los recursos generados de los pedidos de venta y de compra y de las provisiones de materiales.

8

8. Bibliografía

Este capítulo contiene

- Bibliografía utilizada
 - Libros
 - Webs

8.1. Libros

Bauer, Christian and King, Gavin. **Java Persistence with Hibernate**. Second Edition oh Hibernate in Action. Manning, 2006.

Husted, Ted N., Dumoulin, Cedric, Franciscus, George, Winterfeldt, David. **Struts in Action: Building web applications with the leading Java framework**. Manning, 2002.

Hanna, Phil. **JSP: the complete reference**. Translated by Ana M^a Macía Ovalle Pedro P. Sánchez Villalón. Madrid, Madrid: McGraw-Hill Osborne, cop., 2002.

Iverson, Will. **Hibernate: A J2EE Developer's Guide**. Addison-Wesley, 2004.

Kurniawan, Budi. **Java for the Web with Servlets, JSP, and EJB: A Developer's Guide to J2EE Solutions**. 12 de Abril de 2002.

Schach, Stephen R. **Análisis y diseño orientado a objetos con UML y el proceso unificado**. MacGraw-Hill, cop., 2005.

US, Springer, ed. **Enterprise Resource Planning (ERP): The Dynamics of Operations Management**. 2002.

Lowagie, Bruno. **iText in Action, Creating and Manipulating PDF**. Manning Publications, December, 2006.

Bauer, Christian and King, Gavin. **Hibernate in Action**. Manning Publications, 2004.

8.2. Webs

W3Schools. AJAX Tutorial. <http://www.w3schools.com>, 2008.

PostgreSQL Global Development Group, PostgreSQL 8.3.7 Documentation. <http://www.postgresql.org/docs/8.3/interactive/index.html>

Hibernate API Documentation. https://www.hibernate.org/hib_docs/v3/api/

<http://strutstestcase.sourceforge.net/api/index.html>

Displaytag Team, Display tag library 1.2. <http://displaytag.sourceforge.net/1.2/>

Bruno Lowagie, Adolf Baeyensstraat, iText Project. <http://www.lowagie.com/iText/>

Apache Software Foundation, Struts. <http://struts.apache.org/1.x/userGuide/>

Improve Institute, Struts Layout. <http://struts.improve-technologies.com/>

9

9. Anexo

Este capítulo contiene

- Diagramas de clases
- Manual de usuario

9.1. Diagrama de clases

9.1.1. Package Login

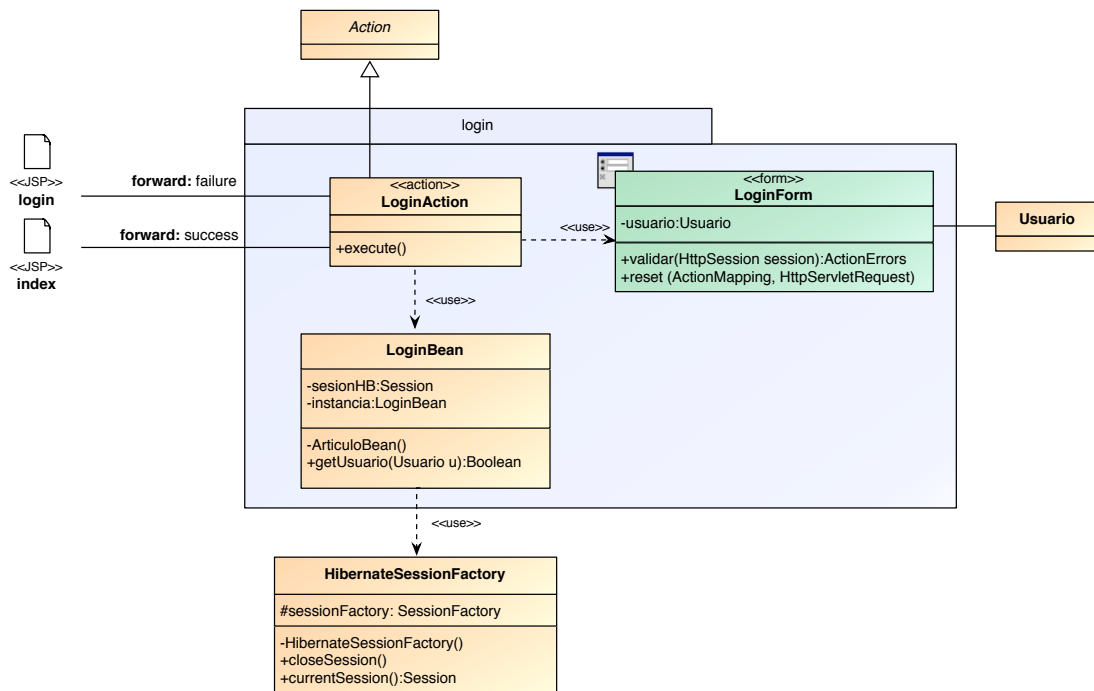


Diagrama X: Diagrama de clase package Login

9.1.2. Package Datos Maestros

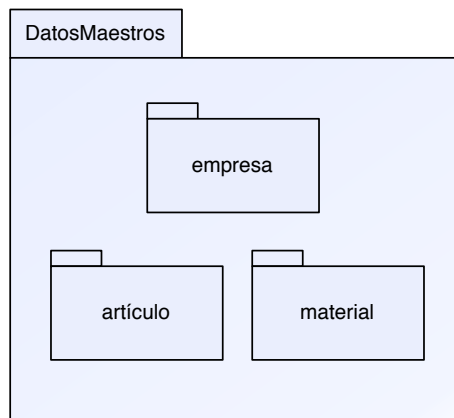


Diagrama X: Diagrama de clase del package `datosMaestros`

9.1.2.1. Package Artículo

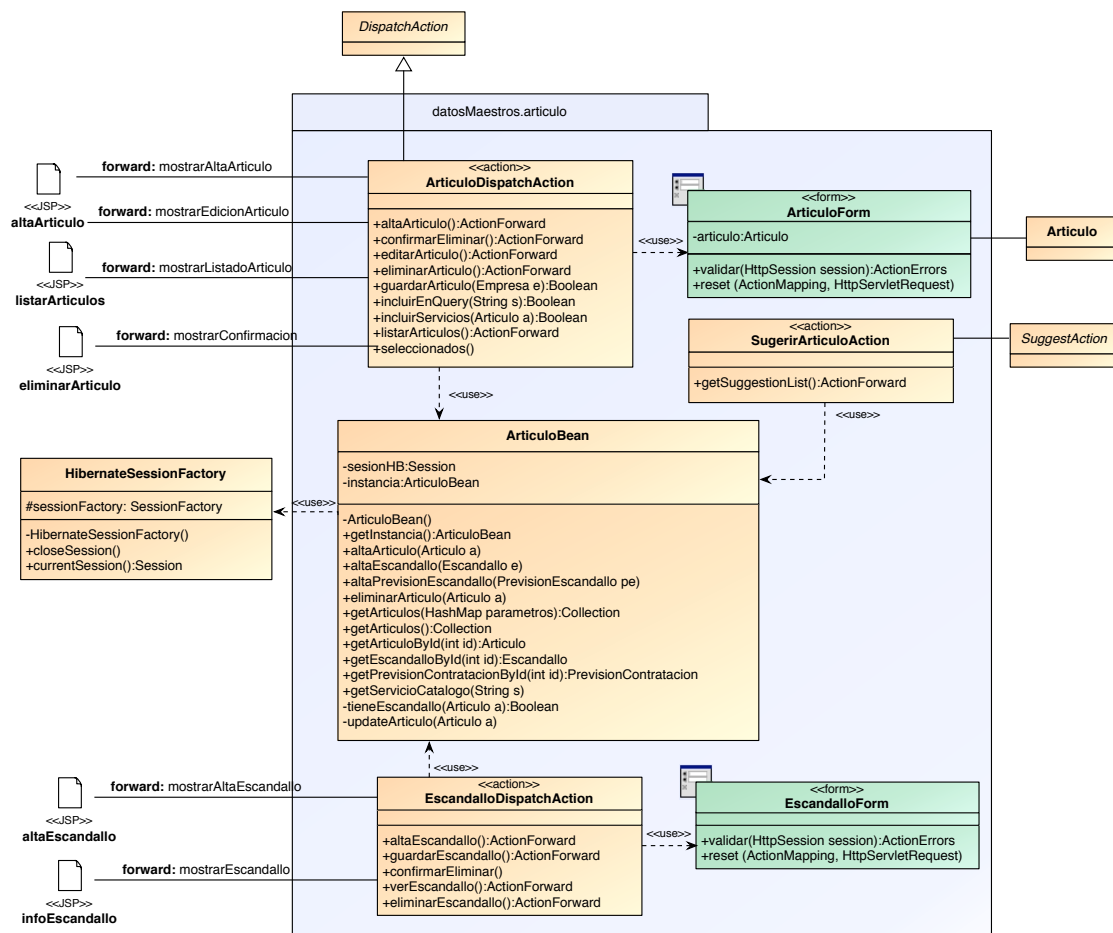


Diagrama X: Diagrama de clase package `articulo`

9.1.2.2. Package Material

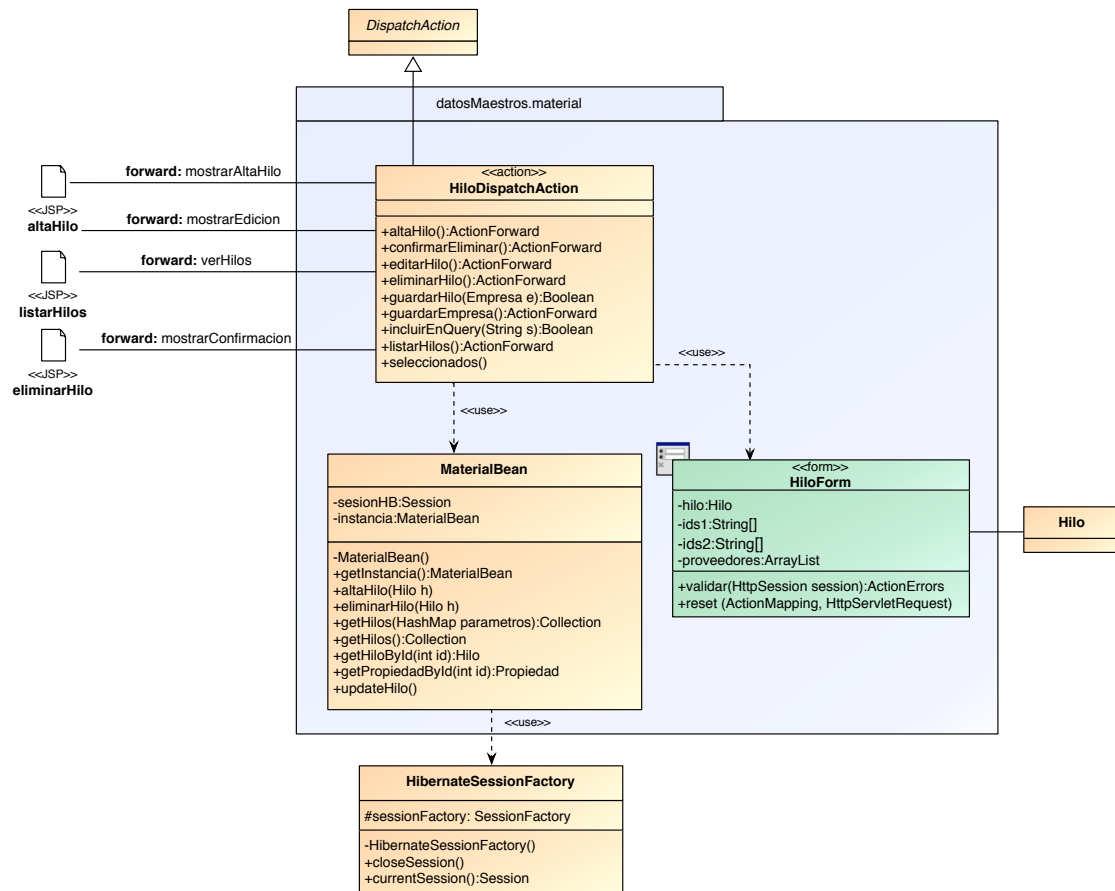


Diagrama X: Diagrama de clase package Material

9.1.3. Package Aprovisionamientos

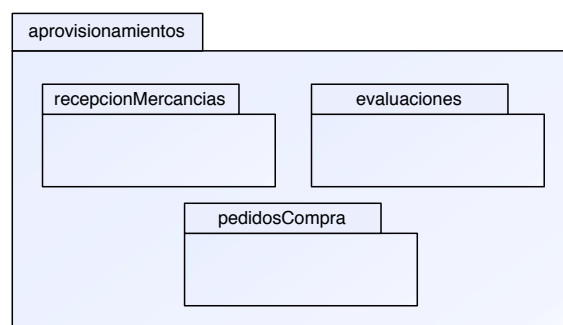


Diagrama X: Diagrama de clase package Aprovisionamientos

9.1.3.1. Package Recepción de mercancías

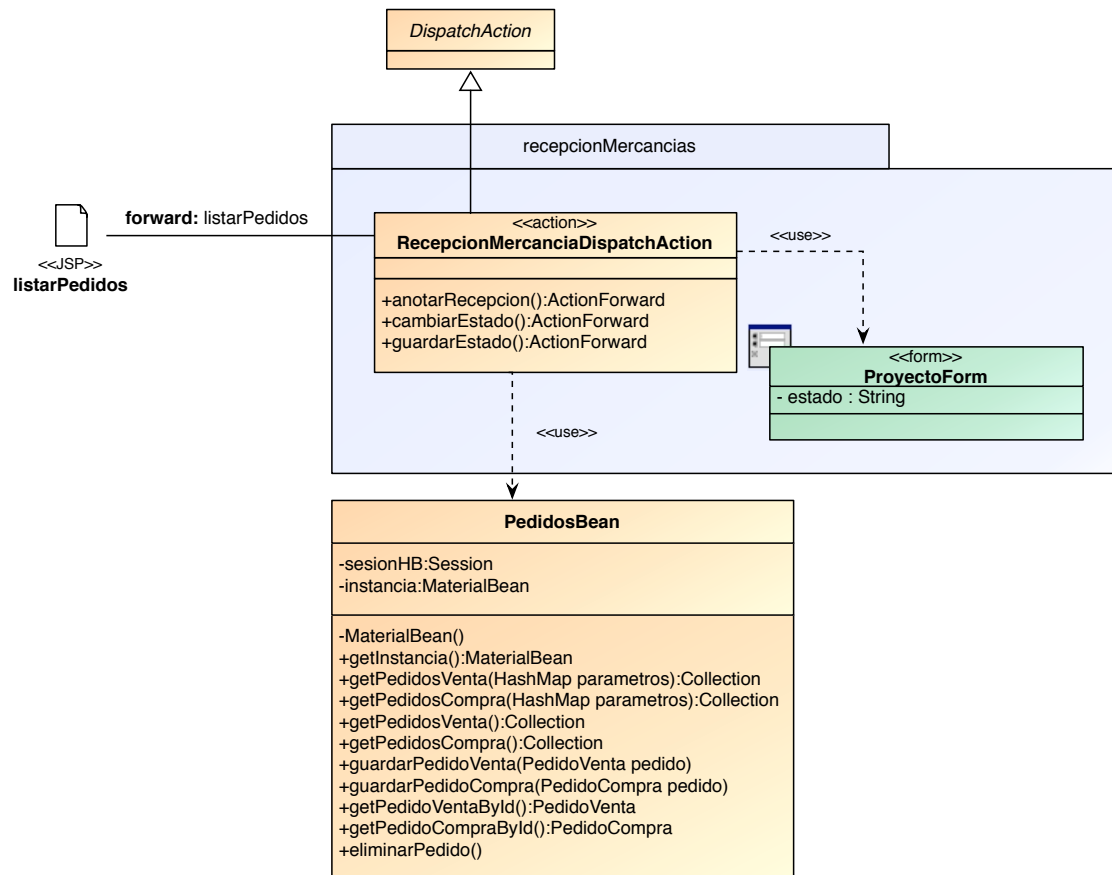
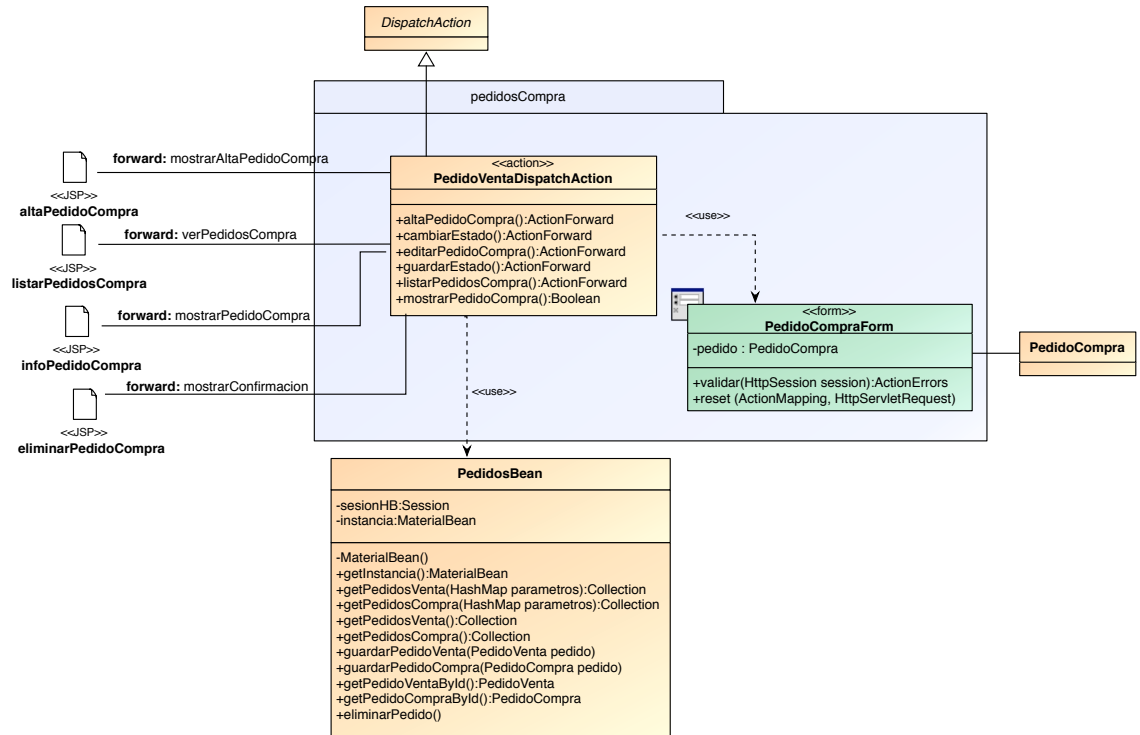


Diagrama X: Diagrama de clase package Recepción de mercancías



9.1.4. Package Proyectos

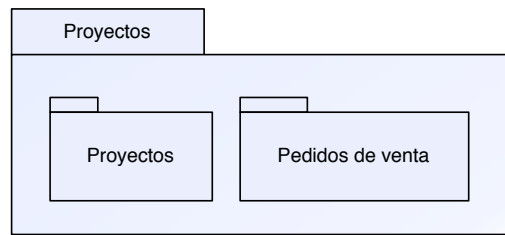


Diagrama X: Diagrama de clase package Proyectos

9.1.4.1. Package Proyectos

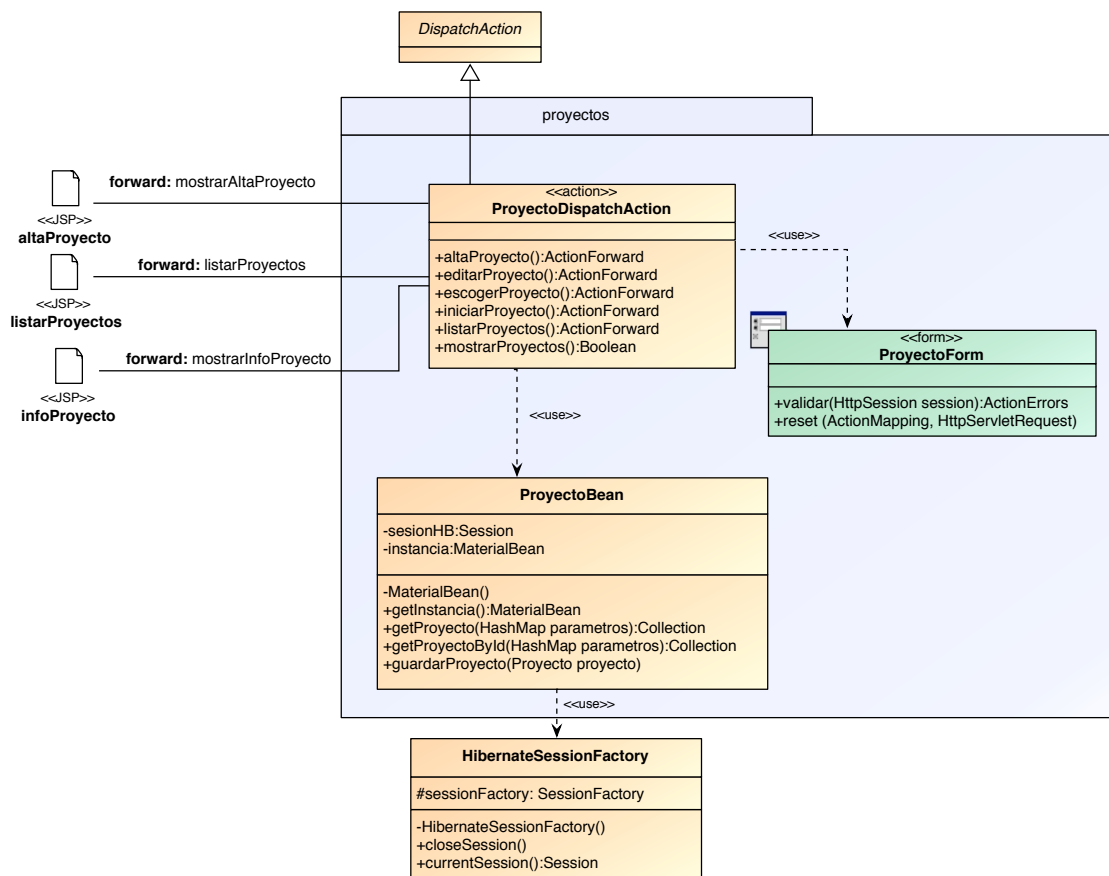


Diagrama X: Diagrama de clase package Proyectos

9.1.4.2. Package Pedidos de venta

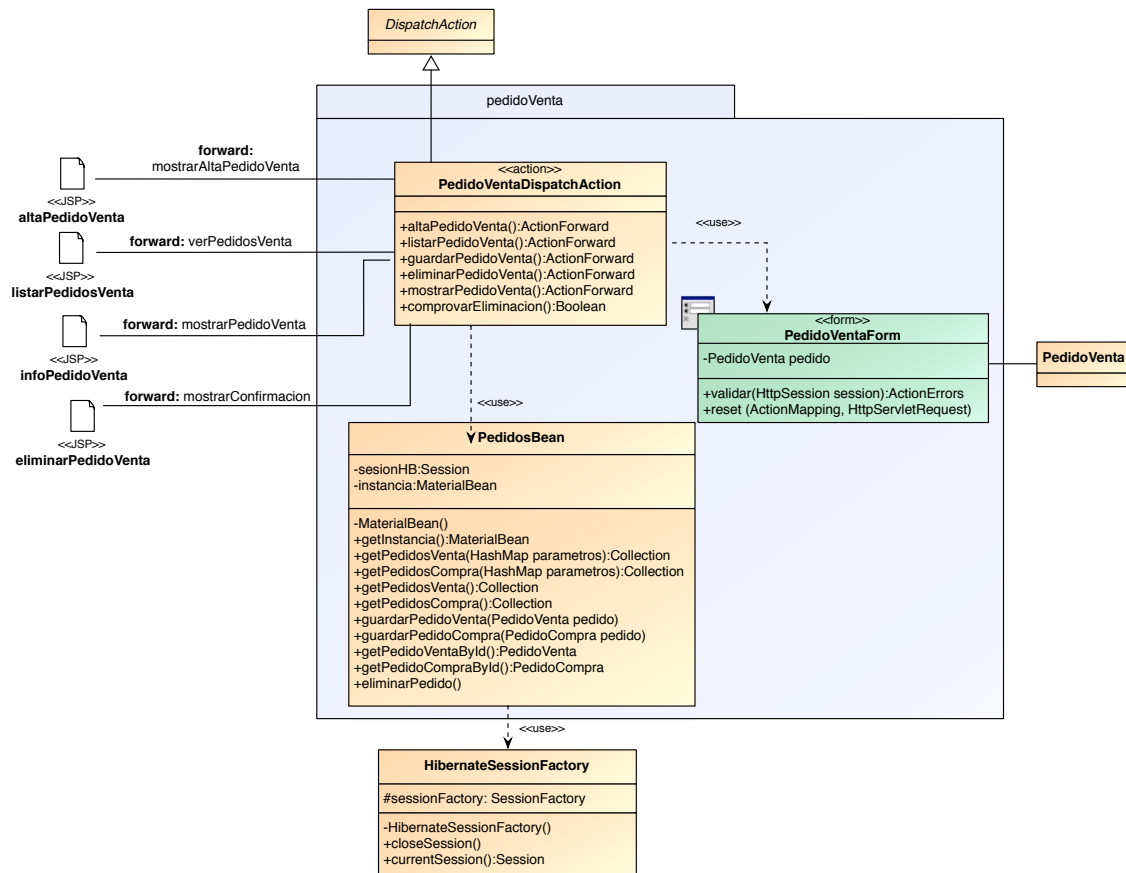


Diagrama X: Diagrama de clase package Pedidos de venta

9.1.5. Package Estadísticas

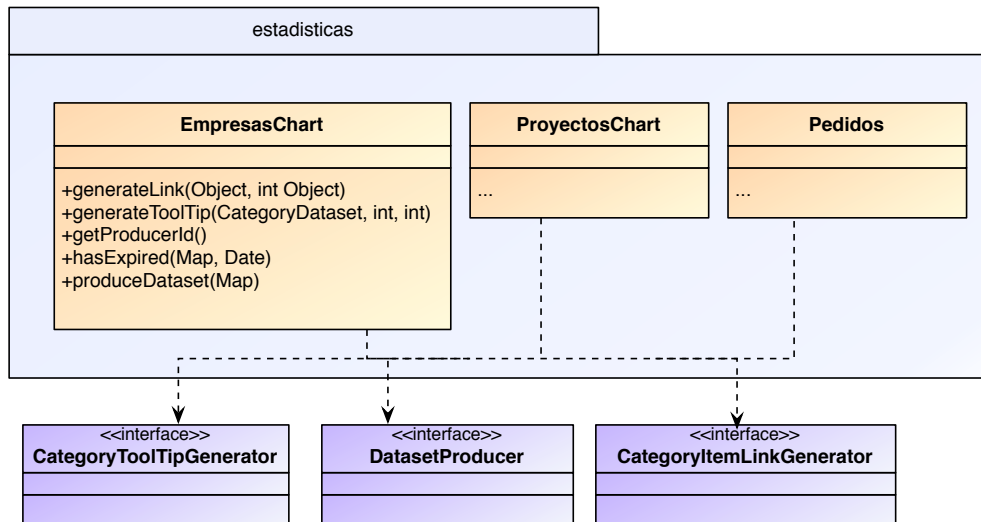


Diagrama X: Diagrama de clase package estadísticas

9.1.6. Package Export

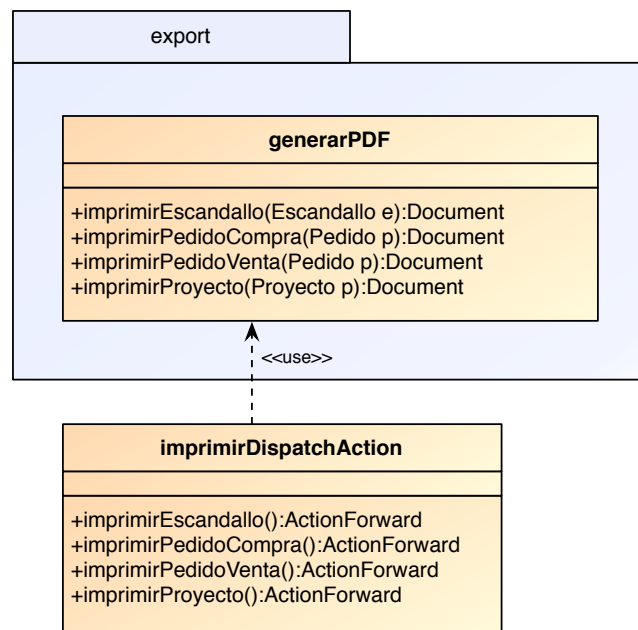
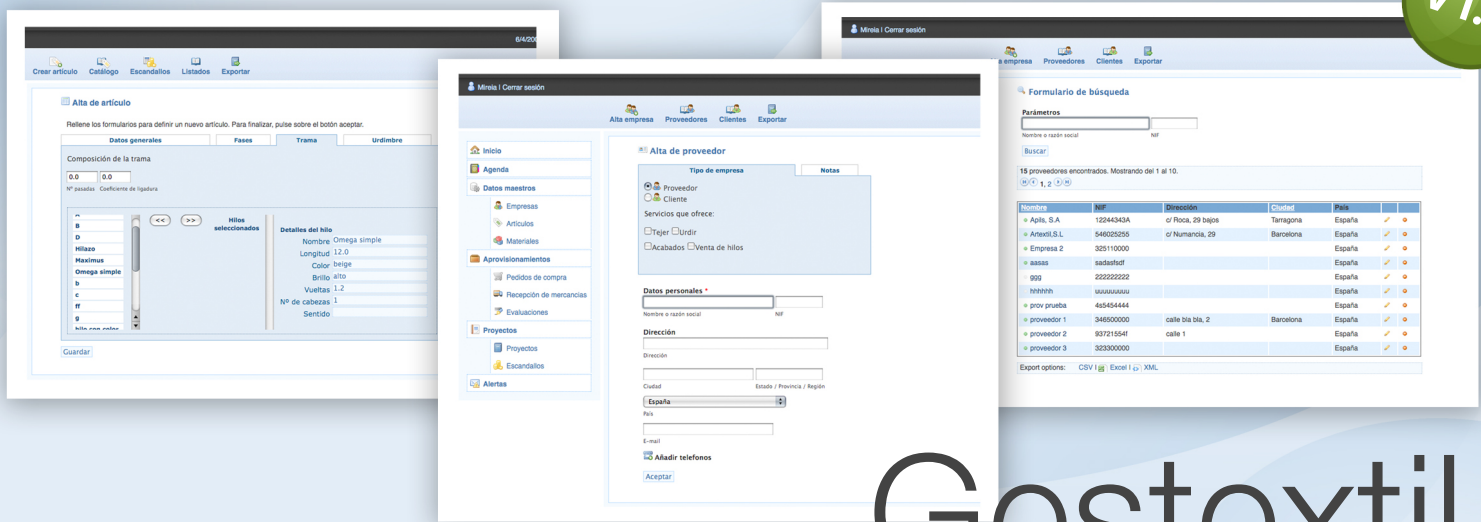


Diagrama X: Diagrama de clase package Export

V1.0



Gestextil

BETA

Gestextil

Manual de usuario

Bienvenido a Gestextil

Gestextil es un sistema de planificación de recursos empresariales funcional y potente a la vez, que permite al usuario gestionar con facilidad todos los recursos de una PYME del sector del textil. Este prefacio le dará una visión general de las características principales de Gestextil, además de proporcionarle una guía de recursos para aprender a usarlo.



Seguridad

Administre los privilegios de sus usuarios con cuentas personalizadas. Controle el acceso al sistema y evita intrusiones no deseadas.



Clientes y proveedores

Información rápida y centralizada de los clientes y proveedores. Acceda de manera rápida a toda la información de su empresa.



Catálogo

Mantenga sus artículos catalogados y listos para ser consultados en cualquier momento. Podrá gestionar tanto artículos como materiales.



Proyectos

Con Gestextil podrá gestionar y controlar todos los proyectos de su empresa de una manera eficaz.



Pedidos de compra/venta

Gestione los pedidos que se realicen en su empresa. Lleve el control de los cobros/pagos y de las mercancías que llegan.



Avisos

Utilice el panel principal para visualizar los recursos que están pendientes de ser atendidos con urgencia.

Contenido

Visión general	4
Ventana principal	4
Menú principal	5
Menú secundario	6
Herramientas del sistema	7
Panel principal	7
Agenda	8
Estadísticas	9
Exportación de informes a PDF	10
Ayuda	10
Datos maestros	11
Empresas	11
Cómo dar de alta una empresa	11
Cómo añadir teléfonos a una empresa	12
Cómo buscar y listar empresas	13
Cómo editar o eliminar una empresa	13
Artículos	15
Cómo crear un artículo	15
Cómo consultar un artículo	17
Cómo editar o eliminar un artículo	17
Cómo crear un escandallo	18
Cómo consultar un escandallo	19
Materiales	20
Cómo crear un material	20
Cómo consultar un material	21
Cómo editar y eliminar un material	21
Aprovisionamientos	22
Pedidos de compra	22
Cómo crear un pedido	22
Cómo consultar un pedido de compra	24
Cómo cambiar el estado de pago de un pedido de compra	25
Recepción de mercancías	26
Cómo anotar las recepciones de mercancías	26
Cómo modificar el estado de un pedido de compra	27
Evaluaciones	27
Cómo crear evaluaciones	28
Cómo consultar una evaluación	28
Cómo modificar una evaluación	29
Proyectos	30
Pedidos de venta	30
Cómo crear un pedido de venta	30
Cómo consultar un pedido de venta	32
Cómo modificar el estado de un pedido de venta	33
Proyectos	34
Cómo iniciar un proyecto	34
Cómo iniciar las fases de un proyecto	36
Cómo consultar un proyecto	38
¿Que estados tienen un proyecto?	38
Administrador	39
Panel del administrador	39
Gestión de usuarios	39
Cómo desactivar un usuario	40

1.Visión general

Ventana principal

Al iniciar Gestextil lo primero que verá es el formulario de *login*, tal y como muestra la imagen 1.



Imagen 1: Pantalla de login

Si es la primera vez que utiliza Gestextil deberá entrar con el nombre de usuario "gestextil" y dejare en blanco el campo de la contraseña. Aprete el botón de iniciar sesión para acceder al sistema cómo usuario registrado.

Lo primero que debería hacer es cambiar su contraseña (para evitar que otros puedan utilizar el sistema bajo su nombre). Antes de esto, vamos a ver el área de trabajo de Gestextil y donde encontrar cada función.

En la imagen 2 se puede ver la aplicación al completo, tal y como se muestra una vez se ha iniciado sesión.

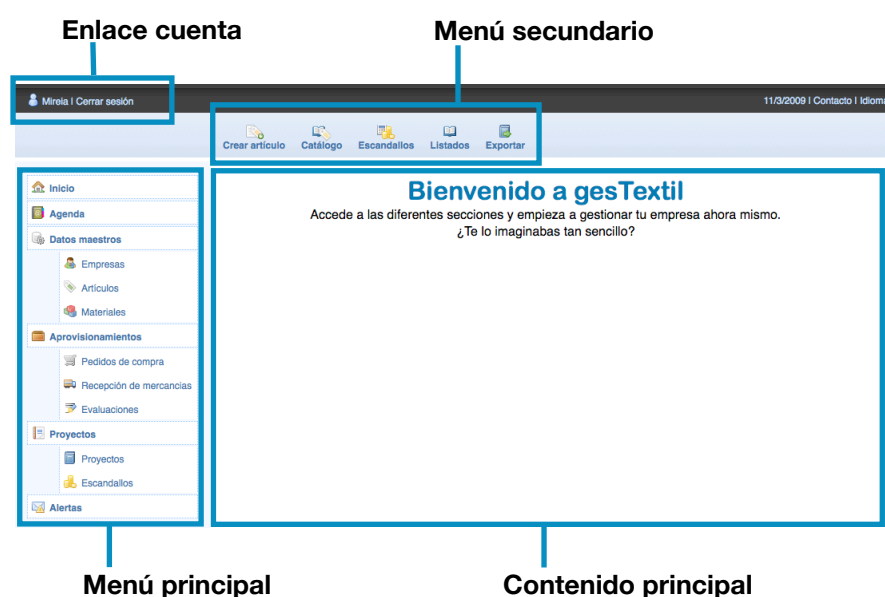


Imagen 2: Ventana principal de Gestextil

Menú principal

En este apartado se explica en que consiste cada menú de la aplicación y como está dividida. En la parte superior izquierda hay información sobre su usuario. Aprete sobre su nombre de usuario para acceder a su perfil. Allí podrá configurar datos referentes a su sesión en el programa, el cambio de contraseña o datos personales. En la imagen 3 se muestra un ejemplo sobre esto.

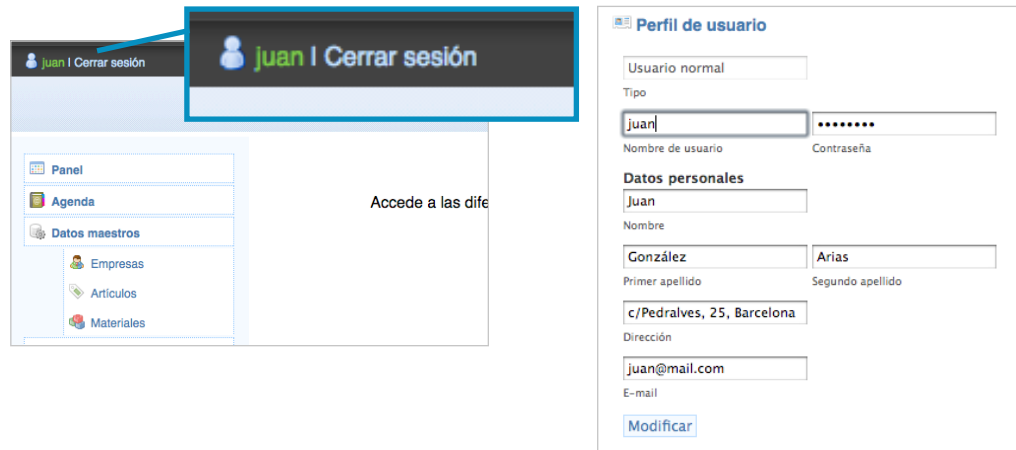


imagen 3: Datos del usuario

Con la finalidad de facilitar la localización de las funciones de la aplicación, existe un menú principal de dos niveles. En cada nivel podemos encontrar las funciones más generales mostrando en cada una de ellas las más específicas. Cada módulo del sistema tiene su menú principal, tal y como se muestra en la imagen 4.



Imagen 4: Menú principal

A continuación, se detalla el contenido de cada menú:

- ▶ **Datos maestros:** Desde aquí podrá gestionar los datos principales de la empresa, desde los proveedores y clientes hasta los artículos y materiales.
- ▶ **Aprovisionamientos:** En este apartado podrá gestionar todo lo relacionado con los proveedores (pedidos de compra, recepción de mercancías y evaluaciones).
- ▶ **Proyectos:** Desde aquí podrá gestionar todo lo relacionado con los clientes, es decir, pedidos de venta y los proyectos que se llevan a cabo para cada pedido.

Menú secundario

En este menú se encuentran las funciones concretas de cada elemento del menú principal. Mientras que trabaje en una sección se mostrará el menú secundario en la parte de arriba donde podremos realizar las diferentes funciones y acceder a ellas de manera rápida. En la imagen 5 se muestra un ejemplo del menú secundario del menú principal “Empresas”.



Imagen 5: Ejemplo de menú secundario

Herramientas del sistema

Panel principal

El panel principal está situado al principio del menú principal. Esta función le permite visualizar información importante referente a pedidos de venta y a cobros y pagos.

La finalidad principal del panel consiste en avisar al usuario de la aplicación de recursos que tienen que ser atendidos con una prioridad mayor a otras.

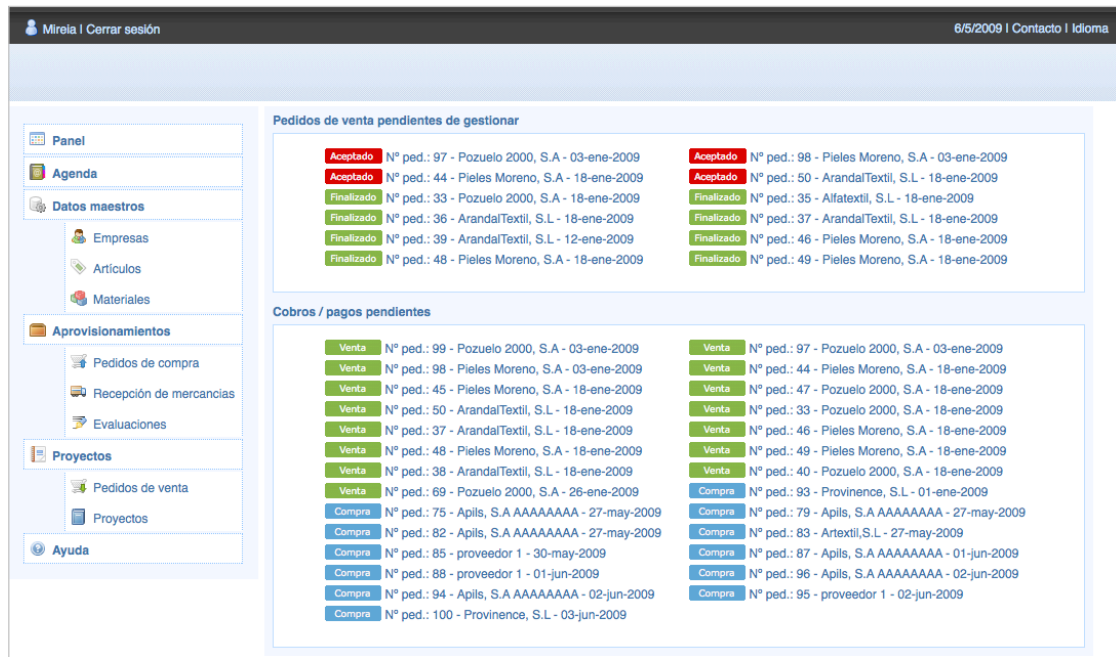


Imagen 6: Panel principal

Existen dos tipos de avisos: los de pedidos de venta y los de cobros pendientes.

- **Pedidos de venta:** en esta primera parte del panel se muestran aquellos pedidos de venta que están pendientes de ser gestionados. Dependiendo del estado del pedido de venta la gestión cambia. Existen dos estados:

Aceptado El estado Aceptado significa que el pedido de venta ha sido aceptado por el cliente y esta a la espera de que se inicie cuanto antes un proyecto para fabricar el artículo pedido.

Finalizado Este estado significa que el pedido de venta ha finalizado, es decir, ya se ha elaborado el artículo y está pendiente de entregarse al cliente.

En cada elemento de la lista se muestra su estado, el número de pedido, el nombre del cliente y la fecha del pedido. Pulse sobre el elemento para visualizar el pedido de venta al completo.

- **Cobros y pagos:** en esta parte del panel se muestran aquellos pedidos que no han sido cobrados o pagados. Existen dos tipos de pedidos:

Compra Estos son los pedidos de compra. En este caso se muestra el número del pedido, el proveedor al que se le ha realizado el pedido de compra y la fecha del pedido. Pulse sobre estos elementos para visualizar el pedido de compra completo.

Venta Estos son los pedidos de venta. En estos elementos se muestran el número de pedido, el cliente al que se le realiza la venta y la fecha del pedido. Pulse sobre estos elementos para visualizar el pedido de venta completo.

Agenda

En el sistema se dispone de una herramienta muy útil a la hora de buscar la información de contacto de una empresa, la agenda.

La agenda se encuentra en el segundo icono del menú principal. Pulse sobre él para mostrar la agenda en el contenido principal tal y como se muestra en la imagen 7.

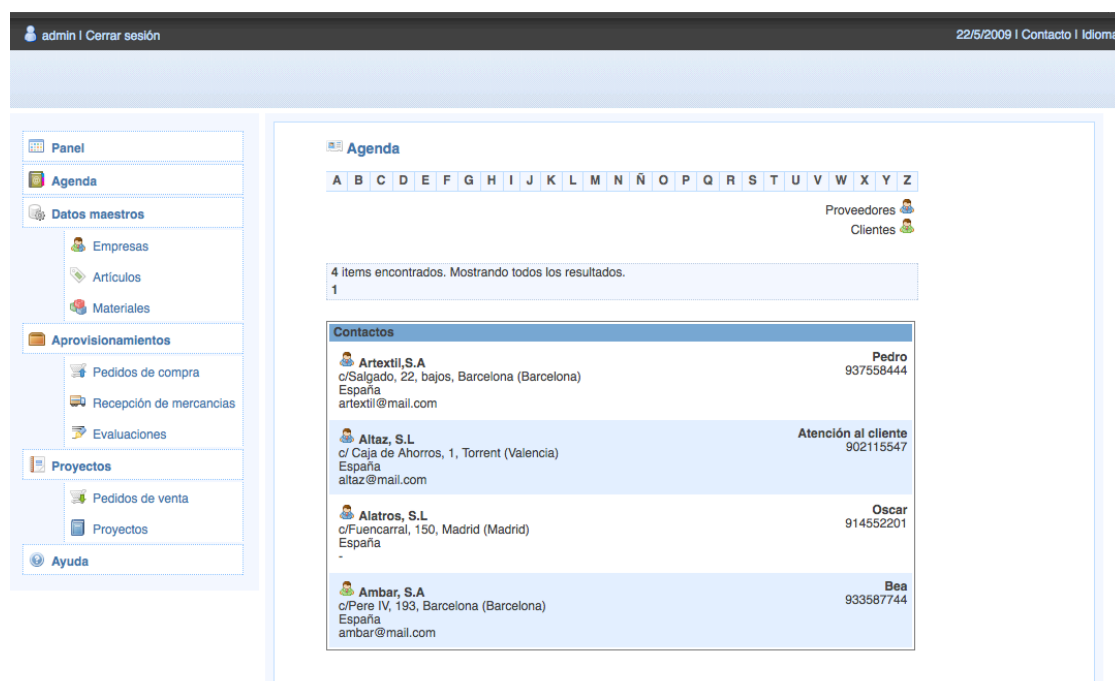


Imagen 7: Agenda

En la parte superior hay un índice alfabético para facilitar la búsqueda de las empresas. Según el tipo de empresa (cliente o proveedor) existe una pequeña leyenda donde se indica con un icono cada tipo.

Para ver como dar de alta un nuevo contacto véase en el capítulo 2, "Cómo dar de alta una empresa".

Estadísticas

Otra de las herramientas que hay en el sistema son las estadísticas. Algunos recursos del sistema disponen de gráficos estadísticos que muestran información útil sobre ellos. Para acceder a ellos sólo debe seleccionar el icono “Estadísticas” en el menú secundario. En la imagen 8 se muestra un ejemplo de gráfico de la sección de proyectos.

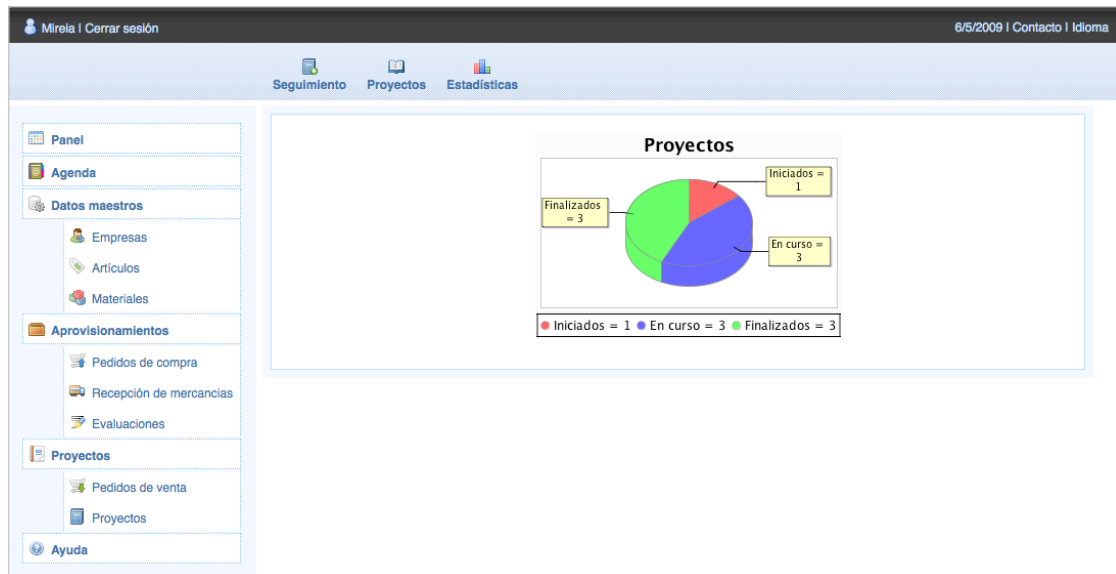



Imagen 8: Ejemplo de estadísticas

En este ejemplo se muestran la cantidad de proyectos que hay según el estado en que estén (iniciados, en curso, finalizados).

NOTA: No todos los elementos del sistema disponen de esta opción.

Exportación de informes a PDF

La exportación de los informes a formato PDF es una de las herramientas más útiles del sistema. Esta función permite exportar los informes de los pedidos, escandallos y proyectos a formato PDF para poder ser imprimidos o simplemente almacenados fuera del sistema.

Para acceder a esta función es necesario consultar alguno de los elementos mencionados y pulsar sobre el icono de exportar .

En la imagen 9 se muestra un ejemplo de exportación de un escandallo.

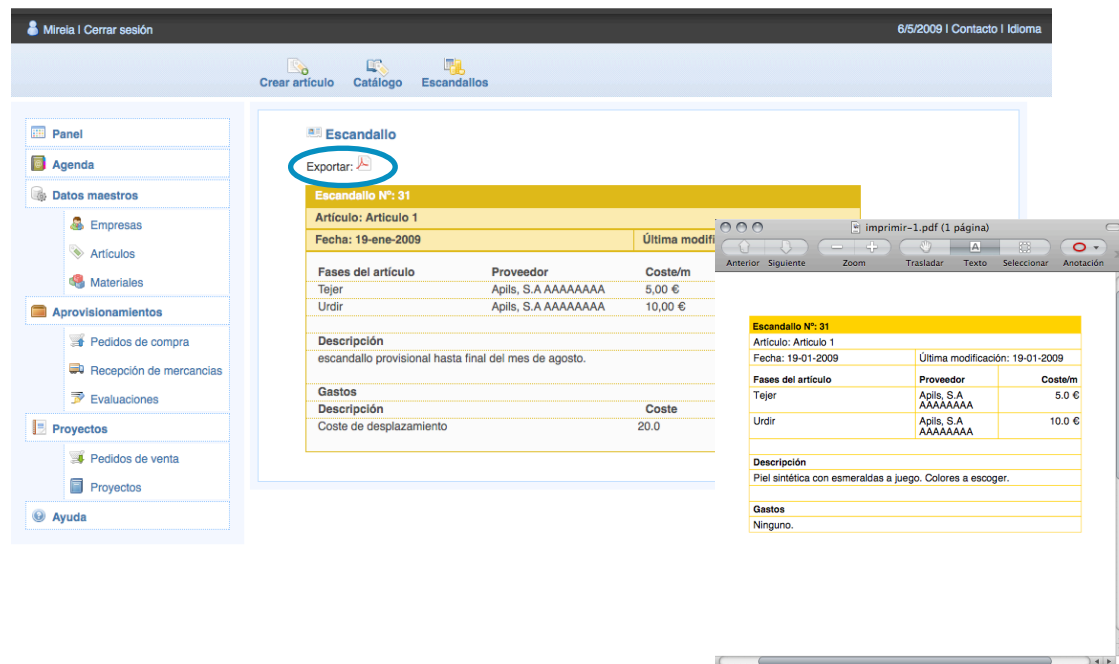


Imagen 9: Exportación a PDF

Ayuda

La ayuda del sistema está situada en el último icono del menú principal. Desde allí podrá consultar este manual de usuario en formato digital.

2.Datos maestros





Empresas

La gestión de las empresas se realiza desde el menú principal en la sección de datos maestros. En el menú secundario tiene los iconos que corresponden a las funciones que se pueden realizar con las empresas. En la imagen 10 podemos ver este menú.



Imagen 10: Menú secundario

A continuación, se detalla la funcionalidad de cada icono:

-  **Alta empresa:** Esta función permite dar de alta una empresa.
-  **Proveedores:** Esta función permite buscar proveedores y listarlos.
-  **Cliente:** Esta función permite buscar clientes y listarlos.
-  **Estadísticas:** Esta función permite visualizar los tipos de empresas que existen en el sistema de manera gráfica.

Cómo dar de alta una empresa

Para dar de alta una empresa en el sistema pulse sobre el botón “Alta empresa” del menú secundario de “Empresas” (ver imagen 10). Según el tipo de empresa que quiera dar de alta seleccione la opción de “proveedor” o “cliente” en la pestaña “Tipo de empresa” tal y como se muestra en la imagen 11.

En la pestaña de “notas” podemos anotar algún comentario referente a la empresa que se va a dar de alta.

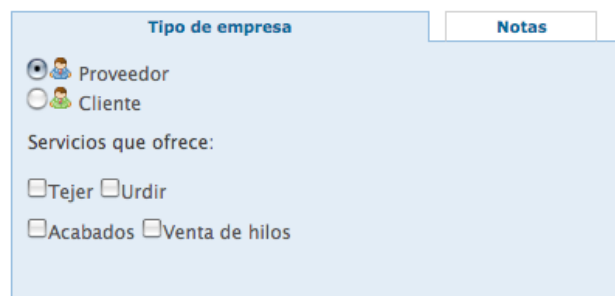
A screenshot of a web form titled 'Tipo de empresa'. It has two tabs: 'Tipo de empresa' (active) and 'Notas'. Under the 'Tipo de empresa' tab, there are two radio buttons: 'Proveedor' (selected) and 'Cliente'. Below these, there is a section titled 'Servicios que ofrece:' with four checkboxes: 'Tejer', 'Urdir', 'Acabados', and 'Venta de hilos'. The 'Tejer' and 'Urdir' checkboxes are checked.

Imagen 10: Tipo de empresa

En el formulario existen algunos campos que son obligatorios y estos están señalizados como tal con un asterisco rojo.

Cómo añadir teléfonos a una empresa

Cuando se realiza el alta de una empresa o una edición, se pueden añadir teléfonos de contacto. Para darlos de alta seleccione el icono “Añadir teléfonos” del formulario. Véase la imagen 12.

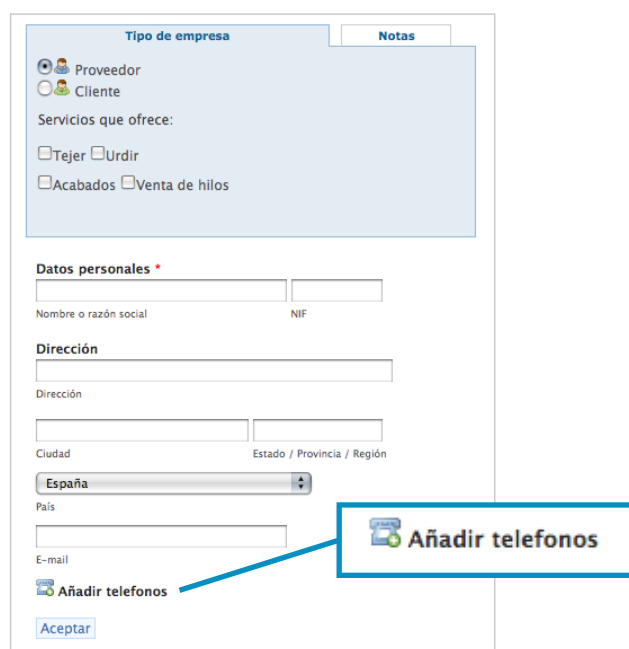

The image shows a web form for company registration. At the top, there are two tabs: 'Tipo de empresa' and 'Notas'. Under 'Tipo de empresa', there are radio buttons for 'Proveedor' and 'Cliente', and a section 'Servicios que ofrece:' with checkboxes for 'Tejer', 'Urdir', 'Acabados', and 'Venta de hilos'. Below this is the 'Datos personales' section, which includes fields for 'Nombre o razón social', 'NIF', 'Dirección', 'Ciudad', 'Estado / Provincia / Región', 'País' (set to 'España'), and 'E-mail'. At the bottom of this section is a button labeled 'Añadir teléfonos' with a telephone icon. A blue box highlights this button, and a blue arrow points from it to a larger box on the right that also contains the 'Añadir teléfonos' button and icon.

Imagen 12: Cómo añadir teléfonos


Al pulsar sobre el botón aparecerá una nueva fila de campos para dar de alta el número de teléfono. Si finalmente no se desea añadir el teléfono después de haber pulsado sobre el botón de añadir, pulse el botón  para eliminarlo. Véase la imagen 13.

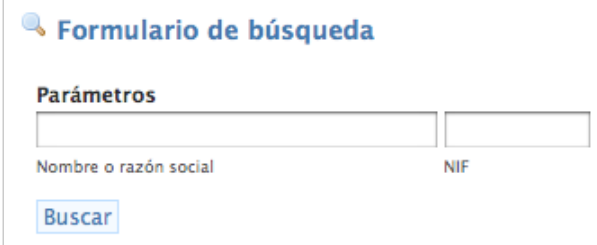
The image shows a close-up of the 'Añadir teléfonos' form. It has a title 'Añadir teléfonos' with a telephone icon. Below the title are two rows of input fields. Each row has a 'Número' label and a 'Fijo' label with a dropdown arrow. There are also minus icons to the right of each row, indicating they can be removed.

Imagen 13: Filas nuevas para añadir teléfonos

Nota: Se pueden añadir varios teléfonos para una misma empresa.

Cómo buscar y listar empresas

Para buscar y listar una empresa seleccione el icono . Si se desea acotar el listado de las empresas se puede utilizar el formulario de búsqueda (véase la imagen 14). Si realiza una búsqueda sin rellenar el formulario se listarán todas las empresas sin ningún tipo de restricción.



Formulario de búsqueda

Parámetros

Nombre o razón social NIF

Buscar

Imagen 14: Formulario de búsqueda de empresas

Cómo editar o eliminar una empresa

Para editar o eliminar una empresa primero es necesario buscarla mediante la función de listar proveedores o clientes (como se ha visto en el punto anterior). Una vez se listan las empresas, seleccione el icono de edición o de eliminar (Véase imagen 15).

13 proveedores encontrados. Mostrando del 1 al 10.

1 2

Nombre	NIF	Dirección	Ciudad	País		
Alatros, S.L	454001212	c/Fuencarral, 150	Madrid	España		
Altaz, S.L	156045000	c/ Caja de Ahorros, 1	Torrent	España		
Artextil,S.A	321540000	c/Salgado, 22, bajos	Barcelona	España		
Bucarest,S.A	212454111	Plz. de la Riera, 34	Rubi	España		
Cardedeu torras, S.A	484212120	c/ Isabel de Villena, 14 - loc.	Granollers	España		
Cortex, S.A	121540014	c/ Joaquim Auger, 14	Barcelona	España		
Fextil Hnos., S.L	154877741	Rda. de Sant Pau, 76	Barcelona	España		
Industrias textil, S.L	414679981	c/ Arnau D'oms, 19	Barcelona	España		
Jiménez Textiles, S.A	445450005	c/Salamanca , 54	Madrid	España		
L. H. sartex, S.A	145440078	c/ Sicilia, 253	Terrassa	España		

Imagen 15: Listado de proveedores

Al seleccionar el icono de edición 🖋 se mostrará el formulario de la empresa seleccionada para poder modificar los datos, tal como se muestra en la imagen 16.



Formulario de edición de empresa. El formulario contiene los siguientes campos:

- Datos personales ***
 - Nombre o razón social:
 - NIF:
- Dirección**
 - Dirección:
 - Ciudad:
 - Estado / Provincia / Región:
 - País:
 - E-mail:
- Añadir telefonos** (icono de teléfono)
- Aceptar** (botón)

Imagen 16: Formulario de edición de empresa

Si pulsa sobre el botón de eliminar 🗑 se mostrará confirmación de que realmente se desea eliminar la empresa seleccionada (Véase la imagen 17).



Confirmación de eliminación de una empresa. La pantalla muestra un mensaje de confirmación en un recuadro rojo:

¿Está seguro de que quiere eliminar la siguiente información? (iconos de confirmación y cancelación)

Debajo del mensaje, se muestra el formulario de datos de la empresa a eliminar:

- Datos personales**
 - Nombre o razón social:
 - NIF:
- Dirección**
 - Dirección:
 - Ciudad:
 - Estado / Provincia / Región:
 - País:
 - E-mail:

Imagen 18: Confirmación de eliminación de una empresa

Artículos

Toda la gestión de los artículos se realiza desde el apartado de artículos situado en el menú principal dentro de la sección “Datos maestros”.

Desde aquí se realizan las altas de los artículos, las modificaciones y las previsiones económicas de fabricación de un artículo, es decir, los escandallos.



Imagen 19: Menú secundario de artículos

Cómo crear un artículo

Para crear un artículo pulse sobre el icono “Crear artículo del menú secundario” (imagen 19). En el contenido principal se mostrará la ficha necesaria para crear un artículo. Existen cuatro pestañas diferentes para la definición de un artículo: datos generales, fases, trama y urdimbre.

Datos generales

En la pestaña de datos generales se especifican los datos principales para identificar a un artículo: el nombre comercial, tipo y descripción.

Si el artículo a dar de alta es un prototipo, seleccione la opción de “prototipo” del formulario (Véase la imagen 20).

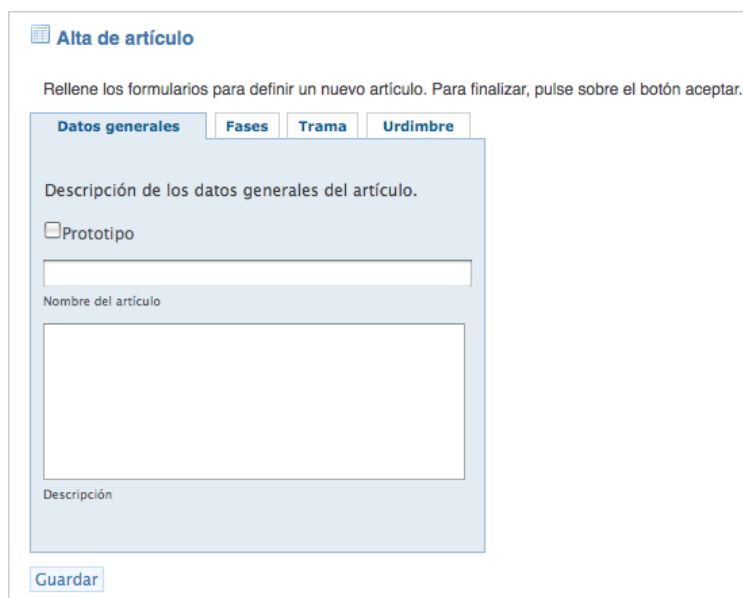
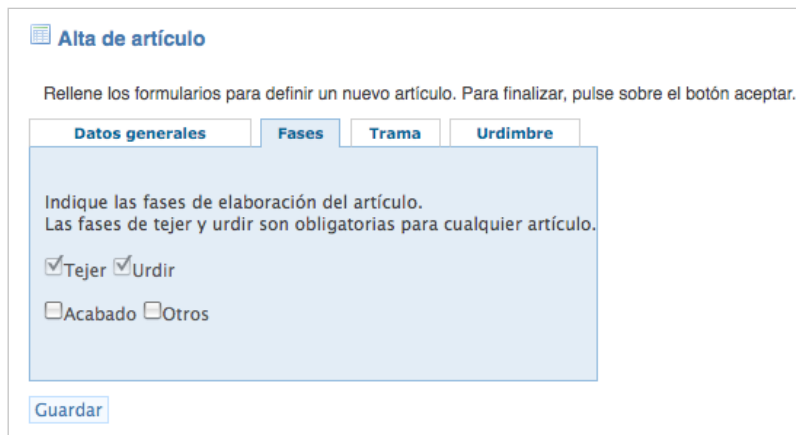
A screenshot of a web application form titled 'Alta de artículo'. Below the title is a instruction: 'Rellene los formularios para definir un nuevo artículo. Para finalizar, pulse sobre el botón aceptar.' There are four tabs: 'Datos generales' (selected), 'Fases', 'Trama', and 'Urdimbre'. The 'Datos generales' tab contains a section 'Descripción de los datos generales del artículo.' with a checkbox for 'Prototipo'. Below this is a text input field for 'Nombre del artículo' and a larger text area for 'Descripción'. At the bottom left of the form is a 'Guardar' button.

Imagen 20: Pestaña “Datos generales” del formulario alta de artículo

Fases

En la pestaña de fases se indica las fases necesarias para elaborar un artículo. Por defecto las fases de “tejer” y “urdir” permanecen seleccionadas ya que son imprescindibles para fabricar un artículo. En la imagen 21 se muestra esta pestaña.



The screenshot shows the 'Alta de artículo' form with the 'Fases' tab selected. The form has four tabs: 'Datos generales', 'Fases', 'Trama', and 'Urdimbre'. The 'Fases' tab contains the following text: 'Rellene los formularios para definir un nuevo artículo. Para finalizar, pulse sobre el botón aceptar.' Below this, there is a section titled 'Indique las fases de elaboración del artículo. Las fases de tejer y urdir son obligatorias para cualquier artículo.' with four checkboxes: 'Tejer' (checked), 'Urdir' (checked), 'Acabado' (unchecked), and 'Otros' (unchecked). A 'Guardar' button is at the bottom.

Imagen 21: Pestaña de “fases” del formulario alta de artículo

Trama y urdimbre

En la pestaña de trama y urdimbre se especifican propiedades necesarias para realizar la trama y el urdimbre. En cada una de las dos pestañas a demás de los campos de las características principales se escogerán los hilos necesarios para la fabricación.



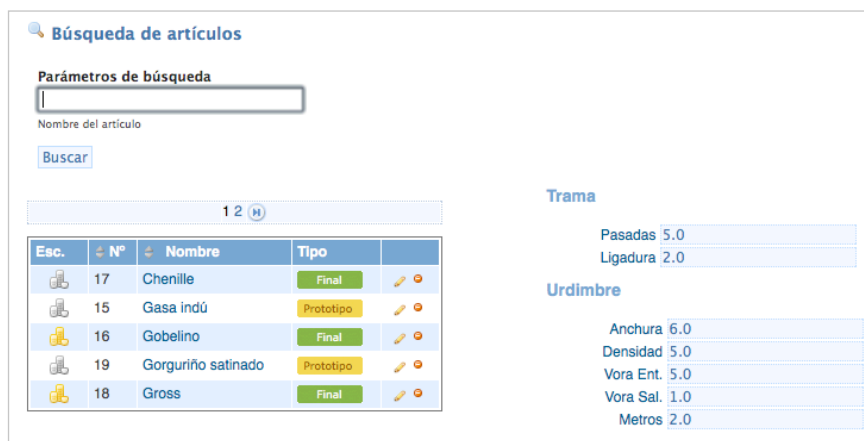
The screenshot shows the 'Alta de artículo' form with the 'Trama' tab selected. The form has four tabs: 'Datos generales', 'Fases', 'Trama', and 'Urdimbre'. The 'Trama' tab contains the following sections: 'Composición de la trama' with two input fields (0.0 and 0.0) and labels 'Nº pasadas' and 'Coeficiente de ligadura'; 'Hilos' with a list of thread names (Corneta, Rasant-metrofil, Saba Cor, Saba-Saba C, Sabaflex, Serafil, Strongfil, TechX); 'Hilos seleccionados' with a list of selected thread names (Garment Dyeing, Bright Floss 440, BAUSCHGARN PA SGD); and 'Detalles del hilo' with fields for 'Nombre' (Sabaflex), 'Longitud' (0.5), 'Color' (Blanco), 'Brillo' (-), 'Vueltas' (12.0), 'Nº de cabezas' (1), and 'Sentido'. A 'Guardar' button is at the bottom.

Imagen 22: Pestaña “Trama” del formulario alta de artículo

Cómo consultar un artículo

Para consultar un artículo seleccione el icono “Catálogo” del menú secundario de artículos (Véase imagen 19).

Para acotar la lista de resultados utilice el formulario de búsqueda o pulse directamente el botón de buscar para mostrar todos los artículos (Véase imagen 23).



Esc.	N°	Nombre	Tipo	
	17	Chenille	Final	
	15	Gasa indú	Prototipo	
	16	Gobelino	Final	
	19	Gorguriño satinado	Prototipo	
	18	Gross	Final	

Trama

Pasadas 5.0
Ligadura 2.0

Urdimbre

Anchura 6.0
Densidad 5.0
Vora Ent. 5.0
Vora Sal. 1.0
Metros 2.0

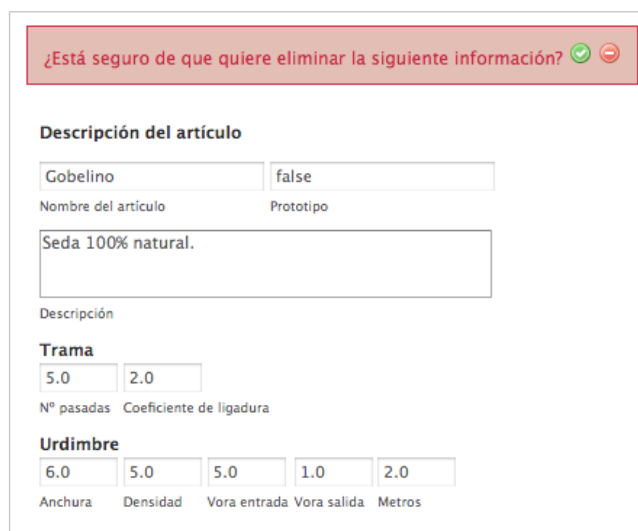
Imagen 23: Listado de artículos

Cómo editar o eliminar un artículo

Para editar un artículo o eliminarlo primero es necesario buscarlo mediante el listado de artículos (visto en el punto anterior). Una vez buscado el artículo deseado, pulse sobre el icono de edición o de eliminar.

Al seleccionar el icono de edición se le mostrará el formulario con los datos del artículo para poder modificar los campos.

Al pulsar sobre el icono de eliminar se le pedirá confirmación sobre el artículo que desea eliminar (Véase imagen 24).



¿Está seguro de que quiere eliminar la siguiente información?

Descripción del artículo

Gobelino false
Nombre del artículo Prototipo

Seda 100% natural.
Descripción

Trama

5.0 2.0
Nº pasadas Coeficiente de ligadura

Urdimbre

6.0 5.0 5.0 1.0 2.0
Anchura Densidad Vora entrada Vora salida Metros

Imagen 24: Confirmación de eliminación de un artículo

Cómo crear un escandallo

Para crear un escandallo busque el artículo del cual desea crear la previsión económica. En el listado de artículos se mostrará el icono 🏠 para indicar que el artículo ya tienen un escandallo. Si el artículo no tiene escandallo aparecerá el siguiente icono 🏠. Pulse sobre él para crear un nuevo escandallo. Véase la imagen 25 donde se muestran los dos tipos de iconos.

Esc.	Nº	Nombre	Tipo	
🏠	17	Chenille	Final	🏠
🏠	15	Gasa indú	Prototipo	🏠
🏠	16	Gobelino	Final	🏠
🏠	19	Gorguriño satinado	Prototipo	🏠
🏠	18	Gross	Final	🏠

Imagen 25: Listado de artículos con y sin escandallo.

En la imagen 26 se muestra el formulario de alta de un escandallo después de pulsar sobre el icono mencionado. En este caso se trata del artículo Omega XL. Cada artículo tiene especificadas las fases necesarias para su elaboración, de manera que al crear un escandallo se mostrará de manera automática cada fase para planificarla.

Alta de escandallo

Fases

Gastos

Omega XL

Artículo

Tejer

Artextil,S.L.

0.0

€

Fase

Proveedor

Coste

Urdir

Artextil,S.L.

0.0

€

Fase

Proveedor

Coste

Notas

Guardar

Imagen 26: Formulario de alta de un escandallo

Cuando se planifica el coste de un artículo mediante el escandallo, también se pueden anotar los gastos que se prevean. Para anotar estos gastos seleccione la pestaña “Gastos” del formulario de alta de un escandallo y especifique el tipo de gasto. Incluya una descripción que detalle la causa del gasto y su coste.

Alta de escandallo

Fases | **Gastos**

☒ Transporte
Origen Destino

☐ Otros

0.0 €
Coste

Descripción

Guardar

Imagen 27: Pestaña de gastos del formulario de alta de un escandallo

Cómo consultar un escandallo

Para consultar un escandallo busqué un artículo desde el icono “Catálogo” del menú secundario de artículos. Liste los artículos o el artículo del cual desee consultar su escandallo. Una vez listados, seleccione el icono 📄 para ver todos los datos del escandallo (Véase imagen 28).

Escandallo

Exportar:

Escandallo Nº: 1		
Artículo: Gobelino		
Fecha: 18-jun-2009		Última modificación: 18-jun-2009
Fases del artículo	Proveedor	Coste/m
Tejer	Artextil,S.A	,50 €
Urdir	Rostex textil, S.L	,20 €
Descripción		
-		
Gastos		
Descripción	Coste	
Transporte del proveedor Artextil hasta nuestro almacén.	10.0 €	

Imagen 28: Detalles de un escandallo

Materiales

Los materiales se encuentran en el menú principal en la sección de “Datos maestros”. Desde aquí podrá gestionar los materiales de la empresa y consultar sus características. En la imagen 29 se muestra el menú secundario de materiales.

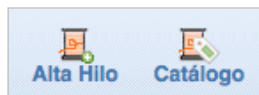


Imagen 29: Menú secundario de materiales

Cómo crear un material

Para crear un material pulse sobre el icono de “Alta de hilo”. A continuación, rellene el formulario de alta (véase imagen 30).

El formulario de alta de hilo tiene una pestaña activa 'Alta de hilo' y tres sub-pestañas: 'Datos generales', 'Propiedades' y 'Proveedores'. En la pestaña 'Datos generales', hay un campo de texto para 'Nombre comercial' y un campo de texto más grande para 'Descripción detallada'. En la parte inferior hay un botón 'Guardar'.

Imagen 30: Formulario de alta de hilos

En la pestaña de proveedores si lo desea podrá especificar que proveedores suministran el hilo que va a dar de alta (Véase imagen 31).

La pestaña 'Proveedores' muestra la lista de proveedores que suministran el hilo. Hay una lista de 'Proveedores' a la izquierda y una lista de 'Proveedores seleccionados' a la derecha. Entre ellas hay botones '<<' y '>>'. La lista de proveedores incluye: Jiménez Textiles, S.A, Portes textil, S.A, Textiler, S.L, Altaz, S.L (seleccionado), Artextil, S.A, Rostex textil, S.L, y Cortex, S.A. La lista de proveedores seleccionados incluye: L. H. sartex, S.A, Industrias textil, S.L, y Alatros, S.L. En la parte inferior hay un botón 'Guardar'.

Imagen 31: Pestaña proveedores del formulario de alta de hilos

Cómo consultar un material

Pulse sobre el botón “Catálogo” del menú secundario de materiales. Utilice el formulario de búsqueda para localizar un material más rápido (Véase imagen 32).

Búsqueda de hilos

Parámetros de búsqueda

Nombre comercial

1 2 (H)

Nombre	Longitud	Descripción	Brillo	Vueltas		
BAUSCHGARN PA SGD	10000.0	Bauschgarn PA SGD es un hilo texturado de poliamida extremadamente suave y de tacto muy agradable. En combinación con nuestro hilo Garment Dyeing, de poliamida, es idóneo para tintar en prenda.	-	4.0		
Bright Floss 440	5000.0	Bright Floss440 es un hilo decorativo exclusivo de Donisthorpe. Este hilo mejora la apariencia de los productos acabados sin necesidad de añadir ningún otro adorno más costoso.	-	4.0		
Corneta	1000.0	Corneta es el único hilo de coser fabricado en España hilado a partir de fibras de algodón egipcio tipo Jumel. Esta calidad de hilo está especialmente recomendada en la fabricación de hilo de coser.	-	2.0		
Garment Dyeing	5000.0	Garment Dyeing es un hilo de gran elasticidad idóneo para tintar en prenda y en combinación con hilo texturado Bauschgarn SGD, también de poliamida, ofrece unos resultados de costurabilidad y tintura excelentes.	-	1.0		
Rasant-metofil	2.0	Poliéster. Recubierto de algodón. Alta tenacidad	-	3.0		
Saba Cor	5000.0	Sabacor es el hilo de coser convencional que supera el rendimiento medio. El hilo recubierto de poliéster 100% posee unas cualidades para coser óptimas y una máxima calidad de costura.	4	2.0		
Saba-Saba C	800.0	Saba-Sabac es el hilo de coser universal fabricado a partir de fibras poliéster de alta tenacidad y bajo el sistema de 3 cilindros, en el cual el Grupo AMANN fue pionero en cuanto a su aplicación al campo del hilo de coser.	-	2.0		
Sabaflex	0.5	Poli(trimetil)terefalato 100%	-	12.0		
Serafil	2.0	Filamento continuo. Poliéster. Alta tenacidad.	-	5.0		
Strongfil	1000.0	Strongfil es el robusto torzal para coser hecho a partir de filamentos continuos de poliamida 6.6 de alta tenacidad.	-	1.0		

Imagen 32: Búsqueda de material

Cómo editar y eliminar un material

Para modificar o eliminar un material primero deberá listarlos (visto en el punto anterior). Una vez haya listado el material seleccione el botón de edición o eliminar. Si selecciona el botón de editar se mostrará el formulario del material que le permitirá modificar sus datos. Si selecciona el botón de eliminar se le pedirá confirmación antes de eliminar el material (Véase figura 33).

¿Está seguro de que quiere eliminar la siguiente información?

Datos del hilo

Bright Floss 440

Nombre comercial

Bright Floss440 es un hilo decorativo exclusivo de Donisthorpe. Este hilo mejora la apariencia de los productos acabados sin necesidad de añadir ningún otro adorno más costoso.

Descripción detallada

5000.0

Longitud (en metros)

-

Color

Brillo

4.0

5

Vueltas

Nº de cabezas

Imagen 34: Confirmación de eliminación de un material

3.Aprovisionamientos

Pedidos de compra

En la sección de aprovisionamientos se encuentra la función de pedidos de compra. Con los pedidos de compra podrá registrar todos los pedidos que realiza a sus proveedores y llevar un control de los pagos y recepción de las mercancías.

Cómo crear un pedido

Para crear un pedido de compra diríjase al menú principal y pulse sobre “Pedidos de compra”, dentro de la sección de aprovisionamientos. En el menú secundario observará el icono de “Alta de pedido”. Pulse sobre el para mostrar el formulario de alta (véase imagen 35).

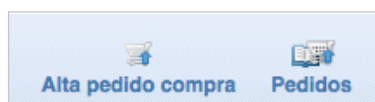


Imagen 35: Menú secundario de Pedidos de compra

En el formulario de alta del pedido deberá indicar los siguientes datos:

- ▶ **Proveedor:** Indique el proveedor al que se le va a hacer el pedido de compra.
- ▶ **Fecha del pedido:** Indique la fecha en que se realiza el pedido de compra.
- ▶ **Pagado:** Este campo solo podrá ser modificado una vez exista el pedido, es decir, se de de alta en el sistema.
- ▶ **Fecha recibido:** Este campo no es editable. La fecha de recepción se asignará automáticamente cuando se especifique mediante la “Recepción de mercancías” que el pedido ha sido recibido.
- ▶ **Estado del pedido:** Indique el estado en que se encuentra el pedido. Los estados posibles son los siguientes:


Borrador El estado de borrador indica que el pedido no ha sido enviado al proveedor y está pendiente de ser informado.

Cursando El estado de cursando indica que el pedido ha sido notificado al proveedor y éste lo está cursando.

Finalizado El estado recibido indica que el pedido ha sido recibido y por lo tanto, el pedido ha finalizado.

► **Servicio:** Indique que tipo de servicio solicita al proveedor. Existen cuatro tipos diferentes:

- **Acabado:** Este tipo sirve para realizar un acabado sobre un tejido. Seleccione la opción de “Acabado” y pulse sobre el botón de añadir para rellenar los campos del servicio. Especifique en el cuadro “Descripción” el tipo de acabado (Véase imagen 36).

 Pedido de compra

Pedido de compra N°: -			
Proveedor:	Alatros, S.L.	Pagado:	-
Fecha del pedido:			
Día:	22	Mes:	Jun 2009
Fecha recibido: -		Estado del pedido: Cursando	
Servicio	Coste	Metros	Recurso
Acabado	12.0 € / m	255.0	
<div>Realizar acabado brillante color negro.</div> <div>Tipo de acabado</div> <div>eliminar</div>			

 Guardar

Imagen 36: Ejemplo de servicio “Acabado”

- **Hilos:** Este tipo de servicio sirve para realizar la compra de hilos. Seleccione esta opción en el desplegable de servicio. Pulse sobre el botón añadir para indicar que hilo desea comprar (Véase imagen 37).

 Pedido de compra

Pedido de compra N°: -			
Proveedor:	Alatros, S.L.	Pagado:	-
Fecha del pedido:			
Día:	22	Mes:	Jun 2009
Fecha recibido: -		Estado del pedido: Cursando	
Servicio	Coste	Metros	Recurso
Hilos	0.5 € / m	150.0	Hilo: Bright Floss 440
<div>eliminar</div>			

 Guardar

Imagen 37: Ejemplo de servicio “Hilos”

- **Tejer y Urdir:** Este servicio sirve para realizar las fases de tejer y urdir de un artículo.

Cómo consultar un pedido de compra

Para consultar un pedido de compra pulse sobre el botón “Pedidos” del menú secundario de “Pedidos de compra”. Una vez haya pulsado le aparecerá un formulario de búsqueda que le permitirá acotar los resultados. Si no desea acotar el resultado pulse directamente sobre el botón buscar. En la tabla de abajo se listarán los pedidos de compra que existen (Véase imagen 38).

Lista de pedidos de compra

Parámetros de búsqueda

Cliente Fecha del pedido

Buscar

28 pedidos de compra encontrados. Mostrando del 1 al 10.

1, 2, 3

Estado	Nº	Cliente	Proveedor	Servicio	Proyecto	Fecha del pedido	Pagado
Borrador	93	Pieles Moreno, S.A	Provinence, S.L	Hilos	-	01-ene-2009	
Borrador	102	Pieles Moreno, S.A	Provinence, S.L	Acabado	-	07-ene-2009	
Borrador	101	Pieles Moreno, S.A	Provinence, S.L	Tejer	-	09-ene-2009	
Finalizado	79	Alfatextil, S.L	Apils, S.A AAAAAAAA	Urdir	4	27-may-2009	
Finalizado	83	Pozuelo 2000, S.A	Artextil,S.L	Tejer	5	27-may-2009	
Finalizado	82	ArandalTextil, S.L	Apils, S.A AAAAAAAA	Tejer	2	27-may-2009	
Finalizado	75	ArandalTextil, S.L	Apils, S.A AAAAAAAA	Urdir	2	27-may-2009	
Finalizado	85	Pozuelo 2000, S.A	proveedor 1	Urdir	5	30-may-2009	
Finalizado	88	Alfatextil, S.L	proveedor 1	Tejer	4	01-jun-2009	
Cursando	87	Pozuelo 2000, S.A	Apils, S.A AAAAAAAA	Tejer	6	01-jun-2009	

Imagen 38: Listado de pedidos de compra

Ahora pulse sobre el icono del pedido de compra que quiere consultar. A continuación, verá un pedido de compra similar al de la imagen 39.

Pedido de compra

Pedido de compra Nº: 118

Proveedor: Provinence, S.L

Cliente: Alfatextil, S.L **Pagado:**

Fecha del pedido: 08-jun-2009

Fecha recibido: **Estado del pedido: Cursando**

Servicio	Coste	Metros	Total
Acabado	1,00 € / m	500,00	500,00 €

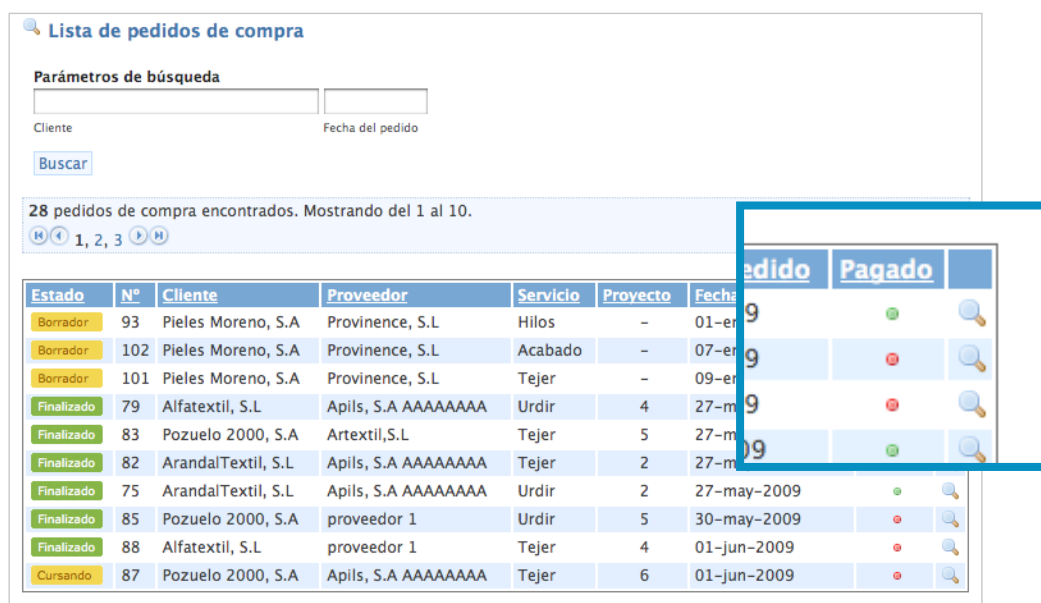
Tipo de acabado

Acabado simple brillante de color negro. Ribete de las costuras con hilo brillante.

Imagen 39: Consulta de un pedido de compra

Cómo cambiar el estado de pago de un pedido de compra

Para cambiar el estado de pago de un pedido de compra pulse sobre el botón "Pedidos" del menú secundario de "Pedidos de compra" para listar los pedidos. Busque el pedido del cual quiere cambiar su estado y pulse sobre el icono de la columna "Pagado" para cambiar su estado (Véase imagen 40).



Lista de pedidos de compra

Parámetros de búsqueda

Cliente: Fecha del pedido:


Buscar

28 pedidos de compra encontrados. Mostrando del 1 al 10.

Estado	Nº	Cliente	Proveedor	Servicio	Proyecto	Fecha	Pedido	Pagado
Borrador	93	Pieles Moreno, S.A	Provinence, S.L	Hilos	-	01-en	9	
Borrador	102	Pieles Moreno, S.A	Provinence, S.L	Acabado	-	07-en	9	
Borrador	101	Pieles Moreno, S.A	Provinence, S.L	Tejer	-	09-en	9	
Finalizado	79	Alfatextil, S.L	Apils, S.A AAAAAAAA	Urdir	4	27-m	9	
Finalizado	83	Pozuelo 2000, S.A	Artextil, S.L	Tejer	5	27-m	9	
Finalizado	82	ArandalTextil, S.L	Apils, S.A AAAAAAAA	Tejer	2	27-m	9	
Finalizado	75	ArandalTextil, S.L	Apils, S.A AAAAAAAA	Urdir	2	27-may-2009		
Finalizado	85	Pozuelo 2000, S.A	proveedor 1	Urdir	5	30-may-2009		
Finalizado	88	Alfatextil, S.L	proveedor 1	Tejer	4	01-jun-2009		
Cursando	87	Pozuelo 2000, S.A	Apils, S.A AAAAAAAA	Tejer	6	01-jun-2009		

Imagen 40: Cambio del estado "Pagado"

Otra posibilidad para modificar el estado de pago es directamente sobre el pedido de compra. Pulse sobre el botón y desde el pedido pulse sobre el botón para cambiar su estado (Véase imagen 41). Para realizar una gestión más rápida puede consultar los pedidos impagados desde el panel principal (vea el punto "Herramientas del sistema").



Pedido de compra

Pedido de compra Nº: 118

Proveedor: Provinence, S.L

Cliente: Alfatextil, S.L

Fecha del pedido: 08-jun-2009

Fecha recibido: Estado del pedido: Cursando

Servicio	Coste	Metros	Total
Acabado	1,00 € / m	500,00	500,00 €

Tipo de acabado

Acabado simple brillante de color negro. Ribete de las costuras con hilo brillante.

Imagen 41: Cambio del estado de pago desde el pedido de compra

Recepción de mercancías

Desde la recepción de mercancías se puede llevar un control de los pedidos de compra realizados y gestionar su recepción. Esta función se encuentra en el menú principal dentro de la sección de "Aprovisionamientos".

Cómo anotar las recepciones de mercancías

Para anotar la recepción de una mercancía pulse sobre el botón de "Recepción de mercancías" de menú principal. A continuación, busque el pedido de compra que quiere anotar como recibido. Si lo desea puede utilizar el formulario de búsqueda para acotar los resultados.

Una vez haya buscado el pedido pulse sobre el icono de estado para cambiarlo a "recibido". Seleccione el nuevo estado y pulse sobre el botón guardar (Véase imagen 42).

En el listado solo aparecen los pedidos de compra que están en estado "Cursando", ya que son los únicos que pueden pasar a estado "Recibido". Si se equivoca cambiando el estado de un pedido a "Recibido" no podrá modificarlo desde la recepción de mercancías. Vea el siguiente punto "Cómo modificar el estado de un pedido".

Listado de mercancías pendientes

6 pedidos encontrados. Mostrando todos los resultados.

1

Nº pedido	Tipo	Empresa	Estado	Proyecto	Fecha
1	Compra	Industrias textil, S.L	Cursando		18-jun-2009
4	Compra	Artextil,S.A	Cursando	1	18-jun-2009
5	Compra	Alatros, S.L	Cursando		22-jun-2009
6	Compra	Alatros, S.L	Cursando		22-jun-2009
11	Compra	Cortex, S.A	Cursando	2	22-jun-2009
15	Compra	Alatros, S.L	Cursando	3	25-jun-2009

Cursando

Borrador

Cursando

Recibido

Imagen 42: Anotar recepción de mercancías

Cómo modificar el estado de un pedido de compra

Si desea modificar el estado de un pedido de compra pulse sobre el botón “Pedidos” del menú secundario de “Pedidos de compra”. Busque el pedido del cual desea modificar su estado y pulse sobre el icono de estado para modificarlo (Véase imagen 43).

8 pedidos de compra encontrados. Mostrando todos los resultados.

1

Estado	Nº	Proveedor	Servicio	Proyecto	Fecha del pedido	Pagado	
Cursando	1	Industrias textil, S.L	Hilos	-	18-jun-2009		
Finalizado	2	L. H. sartex, S.A	Acabado	-	18-jun-2009		
Cursando	4	Artextil, S.A	Tejer	1	18-jun-2009		
Borrador	5	Alatros, S.L	Hilos	-	22-jun-2009		
Cursando	6	Alatros, S.L	Hilos	-	22-jun-2009		
Recibido	11	Cortex, S.A	Tejer	2	22-jun-2009		
Cursando	14	Alatros, S.L	Tejer	3	25-jun-2009		
Finalizado	15	Alatros, S.L	Urdir	3	25-jun-2009		

Imagen 43: Modificar estado de un pedido de compra

Evaluaciones

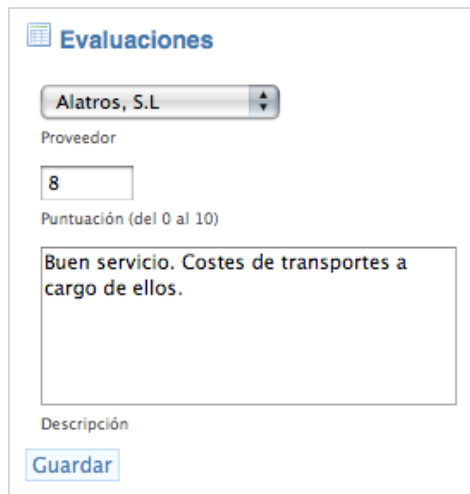
Las evaluaciones le permiten puntuar a los proveedores. De esta manera podrá almacenar el grado de satisfacción con cada proveedor que haya trabajado. Esta función se encuentra en el menú principal en la sección de “Aprovisionamientos”.



Imagen 44: Menú secundario de “Evaluaciones”

Cómo crear evaluaciones

Para crear una evaluación pulse sobre el botón “Evaluar” del menú secundario de “Evaluaciones”. A continuación, seleccione el proveedor del cual desea puntuar. Rellene el resto del formulario especificando una puntuación y una descripción (opcional). Para finalizar, pulse sobre el botón guardar (Véase imagen 45).




El formulario se titula "Evaluaciones". Incluye un menú desplegable para seleccionar el proveedor, actualmente mostrando "Alatros, S.L.". Debajo, hay un campo de texto para la puntuación con el valor "8" y la etiqueta "Puntuación (del 0 al 10)". A continuación, hay un área de texto para la descripción con el contenido "Buen servicio. Costes de transportes a cargo de ellos.". En la parte inferior, hay un botón "Guardar".

Imagen 45: Formulario de alta de una evaluación

Cómo consultar una evaluación

Para consultar una evaluación pulse sobre el botón “Listar” del menú secundario de “Evaluaciones”.

En la tabla se listan todas las evaluaciones existentes especificando la puntuación, el proveedor y los comentarios anotados (Véase imagen 46).

 **Listado de evaluaciones**

5 evaluaciones encontrados. Mostrando todos los resultados.

1






Puntuación	Proveedor	Descripción	
★★★★	Alatros, S.L	Buen servicio. Costes de transportes a cargo de ellos.	
★★★★★	Cardedeu torras, S.A	Transporte gratuito incluso en devoluciones. Entregas puntuales y en pocos días.	
★	Artextil,S.A	Servicio normal. La secretaria con poco trato al cliente.	
★★★	Rostex textil, S.L	Servicio correcto. Entregas en 5 o 6 días laborables.	
★	Textiler, S.L	Mala respuesta ante una devolución por tara en el acabado realizado. Los cargos de transporte extras fueron a nuestro cargo.	

Imagen 46: Listado de evaluaciones

Cómo modificar una evaluación

Para modificar una evaluación pulse sobre el botón “Listar” del menú secundario de “Evaluaciones” y busqué la evaluación a editar. A continuación, pulse sobre el botón de la evaluación que desee modificar (*Véase imagen 47*).

Listado de evaluaciones

5 evaluaciones encontrados. Mostrando todos los resultados.

1




Puntuación	Proveedor	Descripción	
☆☆☆	Alatros, S.L	Buen servicio. Costes de transportes a cargo de ellos.	
☆☆☆☆	Cardedeu torras, S.A	Transporte gratuito incluso en devoluciones. Entregas puntuales en pocos días.	
☆☆	Artextil, S.A	Servicio normal. La secretaria con poco trato al cliente.	
☆☆☆	Rostex textil, S.L	Servicio correcto. Entregas en 5 o 6 días laborables.	
☆☆	Textiler, S.L	Mala respuesta ante una devolución por tara en el acabado realizado. Los cargos de transporte extras fueron a nuestro cargo.	

Imagen 47: Edición de una evaluación

4.Proyectos

Pedidos de venta

Desde la sección de proyectos podemos crear pedidos de venta y gestionarlos mediante proyectos. Para acceder a los pedidos de venta pulse sobre el botón "Pedidos de venta" del menú principal dentro de la sección "Proyectos".



Imagen 48: Menú secundario de "Pedidos de venta"

Cómo crear un pedido de venta

Para crear un pedido de venta pulse sobre el icono "Alta pedido" del menú secundario de "Pedidos de venta". A continuación, rellene el formulario de alta (Véase imagen 49).

 **Pedido de venta**

Pedido de venta N°: -	
Cliente: <input type="text" value="Alfatextil, S.L."/>	Cobrado: -
Fecha del pedido: Dia: <input type="text" value="8"/> Mes: <input type="text" value="Jun"/> 2009 	
Fecha recibido: -	Estado del pedido: <input type="text" value="Aceptado"/>
Artículo: <input type="text" value="Prototipo inicial"/>	Metros: <input type="text" value="0.0"/> <input type="button" value="añadir"/>

Imagen 49: Formulario de alta de pedido de venta

En el formulario de alta del pedido deberá indicar los siguientes datos:

- ▶ **Cliente:** Indique el cliente al que se le va a hacer el pedido de venta.
- ▶ **Fecha del pedido:** Indique la fecha en la que se realiza el pedido.
- ▶ **Cobrado:** Este estado no podrá ser editado hasta que el pedido de venta no se cree.
- ▶ **Fecha recibido:** Esta fecha no puede ser asignada de manera manual. La fecha de finalización del pedido de venta se asignará automáticamente cuando el proyecto que tiene asociado finalice.
- ▶ **Estado del pedido:** Indique con que estado quiere dar de alta el pedido de venta. Existen cuatro estados diferentes:

Borrador El estado de borrador indica que el cliente no ha dado su consentimiento final para llevar a cabo el pedido.

Aceptado El estado aceptado indica que el cliente ha dado su consentimiento para que el pedido sea procesado lo antes posible y se inicie un proyecto.

Cursando El estado cursando indica que se ha iniciado un proyecto para elaborar el artículo del pedido.

Finalizado El estado finalizado indica que el artículo está listo para ser entregado al cliente.

Entregado El estado entregado indica que el pedido de venta ha finalizado totalmente y se ha hecho entrega al cliente.

- ▶ **Artículo:** Indique el artículo que quiere vender al cliente. Si el artículo no existe, debe darlo de alta desde la función "Artículos" del menú principal dentro de la sección "Datos maestros" (Véase el capítulo "Cómo crear un artículo"). Una vez haya escogido el artículo indique cuantos metros de tejido desea y pulse sobre el botón "añadir" para ver los detalles del artículo (Véase imagen 50).

 **Pedido de venta**

Pedido de venta N°: -	
Cliente: Pozuelo 2000, S.A	Cobrado: -
Fecha del pedido:	
Dia: 8	Mes: Jun 2009
Fecha recibido: -	Estado del pedido: Aceptado
Artículo: Prototipo inicial	Metros: 250.0 eliminar
Fase	
Tejer	Coste estimado: ,02 € / m
Urdir	2,00 € / m

Guardar

Imagen 50: Detalles del artículo en el pedido de venta

Cómo consultar un pedido de venta

Para consultar un pedido de venta pulse sobre el botón “Pedidos” del menú secundario de “Pedidos de venta”. A continuación, busque el pedido que quiere consultar. Si lo desea puede utilizar el formulario de búsqueda para acotar los resultados (Véase imagen 51).

 **Lista de pedidos de venta**

Parámetros de búsqueda

Cliente

6 pedidos de venta encontrados. Mostrando todos los resultados.

1

Estado	Nº	Cliente	Artículo	Metros	Fecha del pedido	Cobrado	
Cursando	9	Burdos, S.A	Gasa indú	500.0	03-jun-2009		
Aceptado	8	Cortinaje industrial, S.L	Gobelino	560.0	12-jun-2009		
Cursando	10	Dertel folc, S.L	Gross	900.0	15-jun-2009		
Cursando	3	Burdos, S.A	Gobelino	1000.0	18-jun-2009		
Aceptado	12	Burdos, S.A	Gasa indú	500.0	25-jun-2009		
Aceptado	13	Ambar, S.A	Gross	1000.0	25-jun-2009		

Imagen 51: Listado de pedidos de venta

Una vez haya buscado el pedido que desea consultar pulse sobre el icono para ver los detalles. A continuación se le mostrará los datos completos del pedido de venta como se muestra en la imagen 52.

 **Pedido de venta**

Pedido de venta Nº: 116	
Cliente: Pozuelo 2000, S.A	Cobrado: 
Fecha del pedido: 15-ene-2009	
Fecha recibido:	Estado del pedido: Aceptado
Artículo: Artículo 1	
Metros: 25,00	Coste estimado/m: 15,00
Fase	
Tejer	5,00 € / m
Urdir	10,00 € / m
Descripción	
Piel sintética con esmeraldas a juego. Colores a escoger.	
Total	375,00 €

Imagen 52: Consulta de un pedido de venta

Cómo modificar el estado de un pedido de venta

A diferencia de los pedidos de compra, los pedidos de venta tienen más estados y algunos de ellos no pueden ser asignados manualmente. Para modificar un pedido de venta pulse sobre el botón "Pedidos" del menú secundario de "Pedidos de venta" y busque el pedido del cual quiere modificar su estado. A continuación, pulse sobre el icono de "estado" para cambiarlo (Véase imagen 53).

6 pedidos de venta encontrados. Mostrando todos los resultados.

1

Estado	Nº	Cliente	Artículo	Metros	Fecha del pedido	Cobrado	
Cursando	9	Burdos, S.A	Gasa indú	500.0	03-jun-2009		
Borrador	8	Cortinaje industrial, S.L	Gobelino	560.0	12-jun-2009		
Borrador	10	Dertel folc, S.L	Gross	900.0	15-jun-2009		
Aceptado	3	Burdos, S.A	Gobelino	1000.0	18-jun-2009		
Cursando	12	Burdos, S.A	Gasa indú	500.0	25-jun-2009		
Aceptado	13	Ambar, S.A	Gross	1000.0	25-jun-2009		

Imagen 53: Cambio del estado de un pedido de venta

Tenga en cuenta que dependiendo del estado en que se encuentre un pedido de venta, éste no podrá ser modificado. Solo podrá modificar el estado en los siguientes casos:

- ▶ De Borrador a Aceptado (y *viceversa*)
- ▶ De Finalizado a Entregado (y *viceversa*)

El estado "Cursando" se asignará de manera automática cuando se inicie un proyecto asociado al pedido de venta.

Proyectos

La función de proyectos nos permite realizar un seguimiento para cada pedido de venta. Con la gestión de proyectos usted podrá consultar las fases del artículo, en que estado se encuentran los pedidos de compra y anotar los gastos extras.


La función de proyectos se encuentra dentro del menú principal en la sección "Proyectos".



Imagen 54: Menú secundario de Proyectos

Cómo iniciar un proyecto

Para iniciar un proyecto pulse sobre el botón de "Seguimiento" del menú secundario de "Proyectos". A continuación, se le mostrará un listado de los pedidos de venta que están esperando que se les asocie un proyecto (*estado "Aceptado"*) y los que ya tienen un proyecto asociado y no han finalizado (*estado "Cursando"*).

Busque el pedido de venta del cual quiera iniciar un proyecto y pulse sobre el icono  (Véase figura 55).

Alta de proyecto

Seleccione el pedido de venta del cual quiere iniciar un nuevo proyecto.

4 pedidos de venta encontrados. Mostrando todos los resultados.

1





Estado	Nº	Cliente	Artículo	Metros	Fecha del pedido	
Cursando	3	Burdos, S.A	Gobelino	1000.0	18-jun-2009	
Aceptado	8	Cortinaje industrial, S.L	Gobelino	560.0	12-jun-2009	
Aceptado	9	Burdos, S.A	Gasa indú	500.0	03-jun-2009	
Cursando	10	Dertel folc, S.L	Gross	900.0	15-jun-2009	

Imagen 55: Listados de pedidos de venta para iniciar o editar un proyecto

A continuación se le mostrará los datos del pedido de venta y los datos del proyecto. Pulse sobre el botón “Guardar” para dar de alta el proyecto (Véase imagen 56).

Alta de proyecto

Pedido de venta Nº: 10						
Cliente: Dertel folc, S.L						
Fecha del pedido: 15-jun-2009						
Artículo: Gross						

Fases		Gastos previstos		Gastos extras		
Fase	Proveedor	Coste previsto	Coste real	Iniciar		Estado
Tejer	Cortex, S.A	0.05 €/m	 0.0 €	 0.0 metros		-
Urdir	Fextil Hnos., S.L	0.8 €/m	 0.0 €	 0.0 metros		-


 Guardar

Imagen 56: Alta de proyecto


A continuación, se mostrará los datos del proyecto ya creado. A partir de este momento, ya puede usted gestionar el proyecto y empezar a iniciar las diferentes fases (Véase imagen 57).

Proyecto

Proyecto N°: 2				
Cliente: Dertel folc, S.L				
N° pedido: 10 		Fecha pedido: 15-jun-2009		
Fecha de inicio: 22-jun-2009		Fecha finalizado: -		
Estado: Cursando				
Artículo: Gross		Coste estimado/m: 0.85		
Fase	Proveedor	Coste estimado	Coste real	Estado
Tejer	Cortex, S.A	0.05 €	0.05 €	-
Urdir	Fextil Hnos., S.L	0.8 €	0.8 €	-
Gastos estimados		Coste		
ninguno.		-		
Gastos extras				
ninguno.				

Imagen 57: Detalles del proyecto

Cómo iniciar las fases de un proyecto

Para iniciar las fases de un proyecto pulse sobre el botón “Seguimiento” del menú secundario de “Proyectos”. A continuación, seleccione un pedido de venta que ya tenga asociado un proyecto para empezar a gestionarlo pulsando en el icono  (Véase imagen 58).

Alta de proyecto

Seleccione el pedido de venta del cual quiere iniciar un nuevo proyecto.

4 pedidos de venta encontrados. Mostrando todos los resultados.


1

Estado	Nº	Cliente	Artículo	Metros	Fecha del pedido	
Cursando	3	Burdos, S.A	Gobelino	1000.0	18-jun-2009	
Aceptado	8	Cortinaje industrial, S.L	Gobelino	560.0	12-jun-2009	
Aceptado	9	Burdos, S.A	Gasa indú	500.0	03-jun-2009	
Cursando	10	Dertel folc, S.L	Gross	900.0	15-jun-2009	

Imagen 58: Listado de pedidos de venta con proyectos

Una vez seleccionado se le mostrará los datos del proyecto, tal y como se muestra en la imagen 59.

Alta de proyecto

Proyecto Nº: 3	
Cliente: Burdos, S.A	
Nº pedido: 9 	Fecha pedido: 03-jun-2009
???FechaProyecto??? 22-jun-2009	






Fases		Gastos previstos		Gastos extras	
Fase	Proveedor	Coste previsto	Coste real	Iniciar	Estado
Tejer	Alatros, S.L	0.02 €/m	 0.0 €	 0.0 metros	-
Urdir	Alatros, S.L	0.02 €/m	 0.0 €	 0.0 metros	-

Imagen 59: Gestión de un proyecto

En el encabezado del proyecto se muestran los siguientes datos:

- ▶ **Proyecto Nº:** Indica el número de proyecto como identificador único.
- ▶ **Cliente:** Indica a que cliente se le esta llevando acabo el proyecto.
- ▶ **Nº pedido:** Indica el pedido que tiene asociado el proyecto.
- ▶ **Fecha del pedido:** Indica la fecha en que se inició el pedido de venta.
- ▶ **Fecha del proyecto:** Indica la fecha en que se inició el proyecto.

En la parte inferior del proyecto se muestra una serie de pestañas que le permitirán gestionar las fases de fabricación del artículo y los gastos. A continuación se detallan las diferentes pestañas:

- ▶ **Fases:** En esta pestaña se muestran las fases del artículo. Para cada fase se detalla lo siguiente (*Véase imagen 60*):
 - **Fase:** nombre de la fase.
 - **Proveedor:** Nombre del proveedor.
 - **Coste previsto:** El coste previsto por metro que se estima en el escandallo.
 - **Coste real:** Permite anotar un coste real en caso de que el coste estimado hubiese cambiado.
 - **Iniciar:** Le permite iniciar la fase. Para ello es necesario especificar cuantos metros. Al pulsar sobre el icono  se generará automáticamente el pedido de compra necesario para llevar a cabo la fase.
 - **Estado:** Indica (*si lo hay*) el estado en que se encuentra el pedido de compra de la fase.

Fases					
Gastos previstos			Gastos extras		
Fase	Proveedor	Coste previsto	Coste real	Iniciar	Estado
Tejer	Cortex, S.A	0.05 C/m	 0.0 C		Cursando
Urdir	Fextil Hnos., S.L	0.8 C/m	 0.0 C	 0.0 metros	-

Imagen 60: Pestaña de “Fases” en la gestión de un proyecto

- ▶ **Gastos previstos:** En esta pestaña se muestran los gastos que se planificaron en el escandallo (*Véase imagen 61*).

Fases		
Gastos previstos		Gastos extras
Estos son los gastos previstos en el escandallo del producto.		
Descripción	Coste	Tipo
Coste extra en la fase de tejer por cambio de hilo (espesor más grande).	5.0 C	Transporte

Imagen 61: Pestaña “Gastos previstos” en la gestión de un proyecto

- **Gastos extras:** En esta pestaña podrá anotar los gastos extras que se hayan producido de manera no prevista, es decir, todos aquellos que no se estimaron en el escandallo del artículo (Véase imagen 62).

Anote los gastos que no estaban planificados.

Transporte
Origen Destino

Otros 0.0 €
Coste

Descripción

Añadir gasto

Descripción	Coste	Fecha	Tipo
Coste de transporte extra por devolución del tubo de tejido (mostraba taras).	20.0 €	08-jun-2009	

Imagen 62: Pestaña "Gastos extras" en la gestión de un proyecto

Cómo consultar un proyecto

Para consultar un proyecto pulse sobre el botón "Proyectos" del menú secundario de "Proyectos". A continuación, busque el proyecto que desee consultar y pulse sobre el icono para ver los detalles (Véase imagen 63).

3 proyectos encontrados. Mostrando todos los resultados.

1

Estado	Nº	Cilente	PV	Artículo	Fecha de inicio	Fecha fin	
Cursando	1	Burdos, S.A	3	Gobelino	18-jun-2009	-	
Cursando	2	Dertel folc, S.L	10	Gross	22-jun-2009	-	
Iniciado	3	Burdos, S.A	9	Gasa indú	22-jun-2009	-	

Imagen 63: Listado de proyectos

¿Que estados tienen un proyecto?

Un proyecto tiene tres estados diferentes y son los siguientes:

Iniciado El estado iniciado significa que se ha creado el proyecto pero todavía no se está llevando a cabo su gestión.

Cursando El estado cursando indica que alguna de las fases del proyecto se ha iniciado, es decir, tienen asociado un pedido de compra.

Finalizado El estado finalizado indica que el proyecto ha finalizado, es decir, todas sus fases a finalizado.

5.Administrador

Panel del administrador

Los usuarios administrador permiten tener acceso a todas las secciones del sistema sin restricciones. Desde el panel del administrador podrá gestionar los diferentes usuarios y consultar los datos de su perfil. El panel del administrador está situado en el menú secundario que aparece después de pulsar sobre nuestro nombre de usuario en la parte superior izquierda (*Véase imagen 64*).

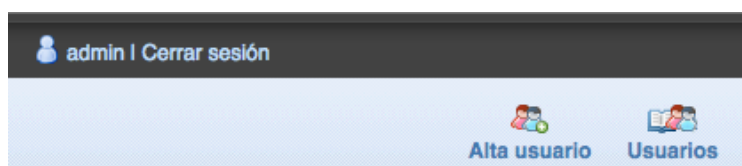




Imagen 64: Panel del administrador

Gestión de usuarios

Para gestionar los usuarios pulse sobre su nombre de usuario como se detalla en el punto anterior y pulse sobre "Alta de usuario" o "Usuarios" dependiendo de lo que quiera gestionar. Si desea dar de alta un nuevo usuario pulse sobre el botón "Alta de usuario"  del panel del administrador. A continuación, especifique que tipo de usuario será "administrador" o "usuario normal" y rellene el resto del formulario (*Véase imagen 65*).

A screenshot of the 'Alta de usuario' (Add user) form. It has a title 'Alta de usuario' with a plus icon. Below the title are two radio buttons: 'Usuario normal' (selected) and 'Administrador'. Underneath is a 'Tipo' label. Then there are two input fields for 'Nombre de usuario' and 'Contraseña'. A section titled 'Datos personales' follows, containing four input fields: 'Nombre', 'Primer apellido', 'Segundo apellido', and 'Dirección'. Below these is an 'E-mail' input field. At the bottom is an 'Aceptar' button.

Imagen 65: Formulario de alta de usuario

Si lo que desea es gestionar un usuario ya existente pulse sobre el botón de “Usuarios”  donde se le listarán todos los usuarios del sistema.

 **Listado de usuarios**

Parámetros de búsqueda

Nombre

4 usuarios found, displaying all usuarios.
1

Tipo	Activado	Nombre usuario	Nombre	1r Apellido	2º Apellido	Direccion	E-mail	
Normal		juan	Juan	González	Arias	c/Pedralves, 25, Barcelona	juan@mail.com	
Normal		bea	Beatriz	López	Santiago	c/Sardenya, 10, Barcelona	beatriz@mail.com	
Admin		admin	Administrador	-	-	-	admin@mail.com	
Normal		cristina	Cristina	Gener	Sanz	-	-	

Imagen 66: Listado de usuarios

Cómo desactivar un usuario

Para desactivar un usuario pulse sobre el icono “Usuarios” del panel del administrador. A continuación, busque el usuario que quiere desactivar y pulse sobre el icono de activado (Véase imagen 67).

Tipo	Activado	Nombre usuario	Nombre	1r Apellido	2º Apellido	Direccion	E-mail	
Normal		juan	Juan	González	Arias	c/Pedralves, 25, Barcelona	juan@mail.com	
Normal		bea	Beatriz	López	Santiago	c/Sardenya, 10, Barcelona	beatriz@mail.com	
Admin		admin	Administrador	-	-	-	admin@mail.com	
Normal		cristina	Cristina	Gener	Sanz	-	-	

Nota: Si desactiva un usuario éste no podrá iniciar sesión en el sistema. Si lo que desea es eliminar el usuario definitivamente, pulse sobre el icono de eliminar. 