



Gestión de productos de una empresa dedicada a la moda

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per

Marina Lerma Gómez

i dirigit per

Joan Sorribes Gomis

Bellaterra, 2 de febrer de 2007

El sotasignat, Joan Sorribes Gomis

Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per la Marina Lerma Gómez

I per tal que consti firma la present.

Signat:

Bellaterra, 2 de febrer de 2007

1.- Introducción	6
1.1.- Estudio de viabilidad.....	6
1.2.- Presentación de la aplicación:	7
1.3.- Objetivos:	11
1.4.- Ventajas e inconvenientes de la aplicación	11
1.4.1.- Ventajas	11
1.4.2.- Inconvenientes	12
1.5.- Recursos necesarios:	12
1.5.1.- Servidor	12
1.5.2.- Cliente.....	12
1.6.- Evaluación de riesgos:	13
1.7.- Organización:	14
1.8.- Coste	15
1.9.- Alternativas	16
2.- Planificación	24
2.1.- Definición de la jornada laboral	24
2.2.- Determinación de actividades	24
2.3.- Planificación inicial	25
2.4.- Coste Inicial.....	27
2.4.1.- Estimación inicial como trabajador por cuenta ajena.....	27
2.4.2.- Estimación inicial como trabajador por cuenta propia	28
2.4.2.1.- Elección de la forma jurídica de empresa	29
2.4.2.2.- Trámites y costes derivados por la constitución de la empresa	30
2.4.2.3.- Costes derivados del desarrollo	33
2.4.2.3.1.- Hardware	34
2.4.2.3.2.- Software	36
2.4.2.4.- Cuadro estimación resumen de costes y coste del proyecto.	36
2.5.- Desviación de la planificación inicial.....	38
2.6.- Coste final y análisis de desviaciones	38
2.6.1.- Coste final como trabajador por cuenta ajena.....	38
2.6.2.- Coste final como trabajador por cuenta propia.....	39
3.- Análisis de Requisitos	40
3.1.- Descripción de la aplicación.....	40
3.1.1.-Descripción del módulo de administración.....	43
3.1.2.- Acceso a la aplicación	43
3.2.- Descripción del entorno de ejecución	43
3.3.- Arquitectura de la aplicación	44
3.4.- Casos de uso:.....	44
3.4.1.- Diagrama de casos	44
3.4.1.1.- Diagrama de casos de uso de perfil Administrador	44

3.4.1.2.-Diagrama de casos de uso de perfil usuario.....	49
3.4.2.- Especificación de casos de uso	50
3.4.2.1.- Validarse.....	50
3.4.2.2.- Listar familias	51
3.4.2.3.- Dar de Alta Familias.....	52
3.4.2.4.- Dar de Baja Familias.....	53
3.4.2.5.- Modificar familias	54
3.4.2.6.- Buscar por código familias	55
3.4.2.7.- Buscar por descripción de familias	56
3.4.2.8.- Selección de familias por árbol	57
3.4.2.9.- Listar modelos	58
3.4.2.10.- Dar de Alta Modelos.....	59
3.4.2.11.- Dar de Baja Modelos	60
3.4.2.12.- Modificar Modelos	61
3.4.2.13.- Buscar por código modelos	62
3.4.2.14.- Buscar por descripción de modelos.....	63
3.4.2.15.- Selección de familias por árbol de modelos.....	64
3.4.2.16.- Listar artículos	65
3.4.2.17.- Dar de Alta Artículos	66
3.4.2.18.- Dar de Baja Artículos	67
3.4.2.19.- Modificar Artículos	69
3.4.2.20.- Buscar por código artículo.....	70
3.4.2.21.- Buscar por descripción de artículos.....	71
3.4.2.22.- Selección de modelos-familias por árbol.....	72
3.4.2.23.- Listar Tiendas	73
3.4.2.24.- Dar de Alta Tiendas	74
3.4.2.25.- Dar de Baja Tiendas.....	75
3.4.2.26.- Modificar Tiendas.....	77
3.4.2.27.- Buscar por código de tienda	78
3.4.2.28.- Buscar por Nombre de tienda	79
3.4.2.29.- Buscar por población de tienda	80
3.4.2.30.- Listar usuarios	81
3.4.2.31.- Dar de Alta Usuarios	82
3.4.2.32.- Dar de Baja Usuarios	83
3.4.2.33.- Modificar Usuarios	85
3.4.2.34.- Buscar por código de usuario	86
3.4.2.35.- Buscar por Nombre de usuario.....	87
3.4.2.36.- Listar temporadas.....	88
3.4.2.37.- Dar de Alta Temporadas	89
3.4.2.38.- Dar de Baja Temporadas	90
3.4.2.39.- Modificar Temporadas	92
3.4.2.40.- Listar Arreglos de modista	92
3.4.2.41.- Dar de Alta Arreglos de modista	94
3.4.2.42.- Dar de Baja Arreglos de modista.....	95
3.4.2.43.- Modificar Arreglos de modista.....	96
3.4.2.44.- Buscar por descripción de arreglos de modista	97
3.4.2.45.- Listar Stock	98
3.4.2.46.- Dar de Alta Stock	99
3.4.2.47.- Dar de Baja Stock.....	100
3.4.2.48.- Modificar Stock	101
3.4.2.49.- Buscar por código artículo.....	102
3.4.2.50.- Buscar por código de tienda	103
3.4.2.51.- Ir a la página siguiente.....	104
3.4.2.52.- Ir a la página anterior	105
3.4.2.53.- Ir a la página número N	106
3.4.2.54.- Informes	107

4.- Análisis de la aplicación	108
4.1.- Diagrama de secuencia	109
4.1.1.- Diagrama de secuencia de validación	109
4.1.2.- Diagrama de secuencia de inicDB	110
4.1.3.- Diagrama de secuencia de listar Familias	111
4.1.4.- Diagrama de secuencia de la función ObtenerDatos()	112
4.1.5.- Diagrama de secuencia de listar Modelos	113
4.1.6.- Diagrama de secuencia de listar Artículos	114
4.1.7.- Diagrama de secuencia de listar Tiendas	115
4.1.8.- Diagrama de secuencia de listar Usuarios	116
4.1.9.- Diagrama de secuencia de listar Temporadas	117
4.1.10.- Diagrama de secuencia de listar Arreglos de modista	118
4.1.11.- Diagrama de secuencia de listar Stock	119
4.1.12.- Diagrama de secuencia de dar de alta Familia	120
4.1.13.- Diagrama de secuencia de selección de familias por árbol	122
4.1.14.- Diagrama de secuencia de dar de baja Modelos	124
4.1.15.- Diagrama de secuencia de Modificar Artículos	127
4.1.16.- Diagrama de secuencia de Búsqueda por población de Tiendas	131
4.1.17.- Diagrama de secuencia de Búsqueda por código de Usuario	132
4.1.18.- Diagrama de secuencia de ir a la página siguiente	133
4.1.19.- Diagrama de secuencia de ir a la página número N	133
4.2.- Diagrama de Clases	134
4.2.1.- Diagrama de Clases general	134
4.2.2.- Diagrama de clases de MySqlAdmin y de las Excepciones	135
4.2.3.- Diagrama de clases específico de la clase de dicTemporada	136
4.2.4.- Diagrama de clases específicos, diagrama número 0	137
4.2.5.- Diagrama de clases específicos, diagrama número 1	138
4.2.6.- Diagrama de clases específico, diagrama número 2	139
4.2.7.- Diagrama de clases específico, diagrama número 3	140
4.2.8.- Diagrama de clases específico, diagrama número 4	141
4.2.9.- Diagrama de clases específico, diagrama número 5	142
5.- DISEÑO	144
5.1.- Especificaciones de la aplicación del módulo de administración ..	144
5.1.1. Validación	144
5.1.2.- Panel de control	145
5.1.3.- Mantenimiento de Familias	146
5.1.4.- Mantenimiento de Modelos	147
5.1.5.- Mantenimiento de Artículos	148
5.1.6.- Mantenimiento de Tiendas	151
5.1.7.- Mantenimiento de Usuarios	152
5.1.8.- Mantenimiento de temporadas	153
5.1.9.- Mantenimiento de Arreglos modista	154
5.1.10.- Mantenimiento de Stock	155
5.2.- Diseño de la base de datos	157
5.2.1.- Diagrama de clases de la base de datos	157
5.2.2.- Definición de las tablas	158
5.2.3.- Script de creación	166
6.- Manual de usuario	170
6.1.- Instalación de la aplicación	170
6.2.- Demostración del funcionamiento de la aplicación	171

6.2.1.- Validación y acceso a la aplicación	171
6.2.2.- Frames del diseño de la aplicación	175
6.2.3.- Elementos comunes de los mantenimientos.....	177
6.2.4.- Mantenimientos de la aplicación	178
6.2.4.1.- Mantenimiento de familias	178
6.2.4.1.1.- Búsqueda por familias	180
6.2.4.1.2.- Búsqueda por Descripción	181
6.2.4.1.3.- Alta de familias	182
6.2.4.1.4.- Modificar Familias	185
6.2.4.1.5.- Eliminar Familias	186
6.2.4.2.- Mantenimiento de modelos	187
6.2.4.2.1.- Alta de modelos	189
6.2.4.2.2.- Modificar Modelos	192
6.2.4.2.3.- Eliminar Modelos	192
6.2.4.3.- Mantenimiento de artículos	193
6.2.4.3.1.- Alta de Artículos	194
6.2.4.3.2.- Modificar Artículos	199
6.2.4.3.3.- Eliminar Artículos	200
6.2.4.4.- Mantenimiento de tiendas.....	200
6.2.4.4.1.- Alta de tiendas	201
6.2.4.4.2.- Modificar Tiendas	203
6.2.4.4.3.- Eliminar Tiendas	203
6.2.4.5.- Mantenimiento de usuarios	204
6.2.4.5.1.- Alta de usuarios	205
6.2.4.5.2.- Modificar Usuarios	208
6.2.4.5.3.- Eliminar Usuarios	210
6.2.4.6.- Mantenimiento de Temporadas.....	210
6.2.4.6.1.- Alta de temporadas	211
6.2.4.6.2.- Modificar Temporadas	212
6.2.4.6.3.- Eliminar temporadas	212
6.2.4.7.- Mantenimiento de Arreglos de modista	213
6.2.4.7.1.- Alta de Arreglos de modista	214
6.2.4.7.2.- Modificar Arreglos de modista	215
6.2.4.7.3.- Eliminar Arreglos de modista	215
6.2.4.8.- Mantenimiento de stock	216
6.2.4.8.1.- Alta de stock	217
6.2.4.8.2.- Modificar Stock	220
6.2.4.8.3.- Eliminar Stock	220
7.- Conclusiones	222
7.1.- Objetivos conseguidos	222
7.2.- Mejoras de la aplicación.....	222
8.- Agradecimientos.....	224
9.- Bibliografía.....	226
Anexos.....	227
Anexo I:- Instalación del PHP	227
Resumen de la memoria	248

1.- Introducción

1.1.- Estudio de viabilidad

Se necesita un programa poder gestionar un grupo de tiendas o franquicias llevando los artículos desde fábrica donde se producen hasta las tiendas que es donde se venden. El control de estos productos actualmente se gestiona a través varios programas, todos ellos comprados por separado y que ayudan de forma específica a llevar cada uno de los temas para los que fueron creados. Los temas de gestión del almacén, la entrada de productos, las ventas de las tiendas a través de las tpv,... Se pretende integrar un solo programa que ayude a llevar en tiempo real todo esto.

La dificultad inicial será poner en marcha todos los productos, introduciendo todos los datos a través de formularios en la base de datos, pero una vez perdido este tiempo inicial la aplicación dará todos los servicios en un tiempo no excesivamente grande debido a su fácil instalación y utilización. Lo que se ha intentando es que la interficie de la aplicación sea lo más intuitiva y sencilla posible para que pueda llegar a todo tipo de usuarios de forma que el aprendizaje de la utilización de la aplicación y la puesta en marcha de ésta sean muy rápidas.

Además una vez introducidos todos los datos en la aplicación la actualización y dada de baja de productos es muy rápida y esto permite aumentar la velocidad de la entrada de nuevos datos.

Para todo esto se decide crear un servicio por Internet que ofrezca una gestión y administración totalmente automatizada. Este servicio está orientado a empresas que poseen un grupo de tiendas o franquicias que comercialicen productos como prendas de ropa y complementos (cinturones, relojes, zapatos,...)

La versión 1.0.0 contempla potencialmente empresas que gestionen sus productos mediante Talla, Color, Código de barras y Precio. Existe la posibilidad de realizar un conjunto de propiedades estándares de productos, configurables según las necesidades del cliente.

El proyecto consta de dos partes diferenciadas:

1. Módulo de administración
2. Módulo de gestión.

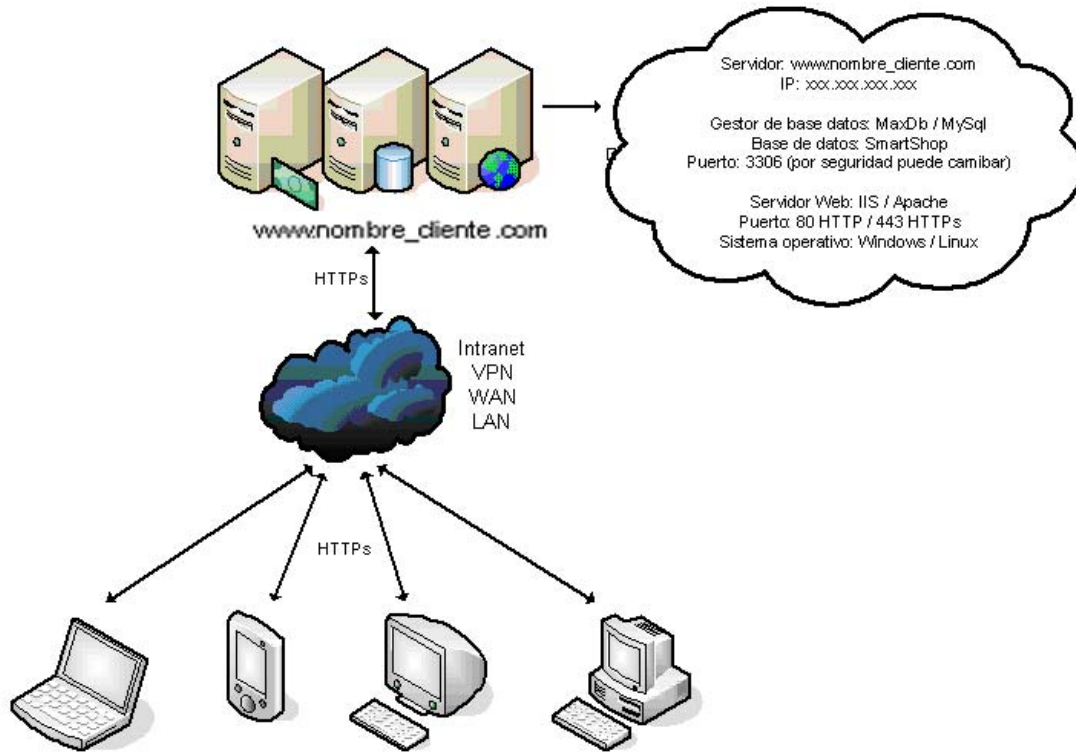
3.

1.2.- Presentación de la aplicación:

La aplicación una vez creada y probada será presentada a través de pantallas de entrada de datos y de gestión de compra y venta muy intuitivas y con poco margen de error a nivel de usuario.

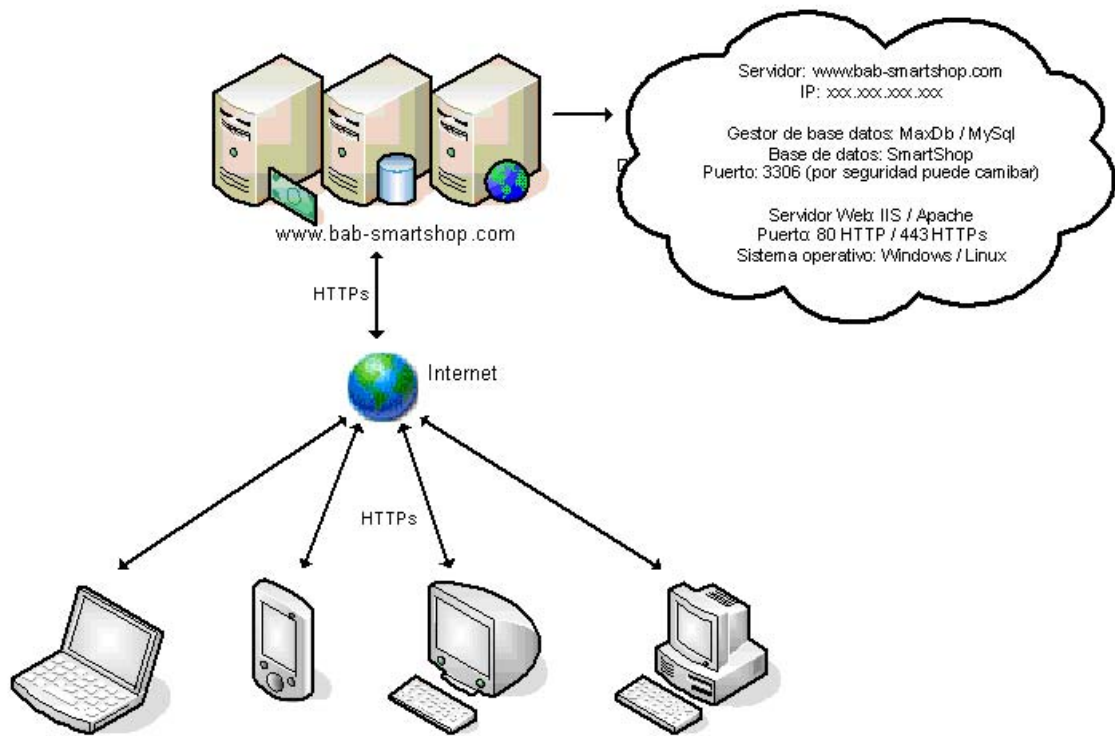
Entorno de la aplicación

1. WAN, VPN, Intranet,...



Esta opción será implementada en el caso de que el cliente desee tener dentro de su infraestructura la aplicación. El cliente será el responsable del mantenimiento del hardware, software, así como de las copias de seguridad.

2. Internet



Esta opción será implementada en el caso de que el cliente desee utilizar nuestro servicio siendo responsable BabSoft del software, hardware y copias de seguridad. En este caso el cliente abonará una mensualidad que incluirán:

- Actualizaciones gratuitas
- Reporting mensual
- Soporte telefónico en horario comercial
- Copias de seguridad

- **Arquitectura de la aplicación**

La arquitectura de la aplicación se divide en tres capas bien diferenciadas:

1. Capa de interfície

Esta capa hace referencia a la presentación de la información. Podemos interpretar esta capa como el conjunto de páginas HTML que debemos diseñar, ya sea estática como dinámicamente.

2. Capa de negocio

Esta capa está encargada a resolver toda la lógica de negocio. El nivel de lógica de negocio es la porción de la arquitectura realizada por componentes para crear y garantizar las reglas de clientes, productos y de negocio. Existen diversas tecnologías que pueden aparecer en esta parte de la arquitectura como SMTP, FTP, WebServices,...

3. Capa de datos

El nivel de gestión de acceso a datos es la porción de la arquitectura que proporciona intercambio de datos en tiempo real (o casi) entre los diversos sistemas responsables del flujo de datos. En nuestro caso será una clase en php que ofrecerá acceso a la base de datos de forma reutilizable, estructurada, estable, transaccional, control de excepciones,...

1.3.- Objetivos:

El objetivo es poder conseguir que una persona este donde este con un ordenador pueda encargarse de la producción y de la gestión de las ventas de todos sus productos a través de internet y en tiempo real pueda tener una visión clara del stock de sus tiendas, de las ventas de éstas, de los productos mas vendidos y de los que están a punto de acabarse por si se puede volver a reponer stock de otra tienda o de la misma fábrica. De esta manera se ganará en tiempo de repartos y en tiempo de coordinar los productos entre tiendas y fábrica.

La usabilidad del módulo de Administración debe ser ágil y sobre todo rápido pero se debe tener en cuenta que su uso no será diario durante 12 horas. La resolución de pantalla será de 1024 x 768 aceptando 800x600 como resolución mínima.

La usabilidad del módulo de Tpv debe ser extremadamente ágil y aún más rápida, ya que su uso será diario durante 12 o más horas. La resolución de pantalla será de 640x480 como resolución mínima, debido a que la mayoría de estos comercios utilizan mini monitores.

Crearemos una nomenclatura de programación para conseguir un único modo de desarrollo.

Implementaremos un generador de código a partir de la base de datos. Con ello obtendremos la capa de datos.

1.4.- Ventajas e inconvenientes de la aplicación

1.4.1.- Ventajas

1. Centralización de los datos en una única base de datos
2. Datos en tiempo real
3. Administración desde cualquier ubicación geográfica con una conexión a Internet
4. Administración de todas las tiendas (stock, ventas, reservas,...)
5. Consulta de stock entre tiendas
6. Reservas de productos entre tiendas
7. Control de arreglos de modistas
8. Control de vales y pagas y señales
9. Control de caja
10. Control de ventas (por vendedor/a, artículo, familia,...)
11. No se instala software en las estaciones clientes (teóricamente)

12. Copias de seguridad
13. Restauración de las copias de seguridad

1.4.2.- Inconvenientes

1. El servicio depende de una conexión a Internet estable
2. Usabilidad (velocidad de uso, respuesta vía https,...)
3. Interactuar con impresora de ticket + cajón
4. Mantenimiento de las estaciones de trabajo
5. Desarrollar una aplicación Windows (en C++ o VB) para ser utilizada cuando se caiga la conexión a Internet

1.5.- Recursos necesarios:

Los recursos hardware/software mínimos necesarios para la implantación de este proyecto son:

1.5.1.- Servidor

- PC Pentium III con 512 Mb de memoria RAM con tarjeta de red que realizará las funciones de servidor.
- Conexión a internet (ADSL o cable).
- Sistema operativo Windows 2000 Server con Service Pack 2.
- Un servidor donde se instalarán el IIS y el servidor de la base de datos de MySql Server.
- PHP 5.0 o superior

1.5.2.- Cliente

- PC Pentium III con 256 Mb de memoria RAM con tarjeta de red.
- Conexión a internet (ADSL o cable)
- Navegador web Internet Explorer 5.0.

1.6.- Evaluación de riesgos:

Como cualquier aplicación destinada a ejecutarse a través de internet se corren los siguientes riesgos:

- La aplicación puede ejecutarse de dos maneras con un servidor que el propio cliente establezca en su fábrica o a través de un servicio que la empresa babSoft pone a disposición para gestionar y guardar los datos en un servidor propio. Sea cual sea de las dos existe la posibilidad de que este servidor puede estropearse y provocar un mal funcionamiento de la aplicación, perdiéndose datos y actualización de ventas y de entrada a la base de datos
- Si deja de funcionar internet por un problema en la línea de teléfono nos encontraríamos que se dejarían de poder hacer ventas, ya que toda la aplicación corre vía internet.
- Los errores que pueden haber en el software del servidor o un error interno de hardware puede provocar que la aplicación o parte de esta deje de funcionar.

Para estos problemas se propone:

- Para solucionar un posible problema en el disco duro se debería tener un servidor con dos discos duros SCSI y en paralelo donde se encuentran copias simultáneas de los datos, si el mal funcionamiento es de la máquina, esto lo suplen teniendo un servidor conectado en paralelo dando el mismo servicio al unísono. También existe la posibilidad de que el mal funcionamiento sea debido a un corte eléctrico de luz esto se suplirá con un SAI, que permitirá el funcionamiento del servidor durante unas horas a la espera de recuperar el servicio eléctrico. También se propone guardar en modo fichero posibles datos de compra y venta en franquicias y entrada de datos dentro de la propia máquina, este fichero de fallo se recuperará actualizando los datos una vez se reponga el fallo del servidor.
- En el caso de que el fallo sea de internet se propone el hacer correr una aplicación instalada en las propias máquinas que guarde toda la información de las ventas hasta que se vuelva a recuperar el servicio de internet. Se entiende que cada vez que hay un problema con la línea telefónica en las tiendas siempre hay un poco desfase debido a que dejan de funcionar otros temas importantes como son el pago de los productos a través de tarjetas de crédito.

1.7.- Organización:

Principales etapas del proyecto:

- Análisis del proyecto: Obtenemos los requisitos del sistema y las especificaciones por parte del cliente de lo que necesita que realice la aplicación.
 - o Definición del sistema
 - o Análisis de los requisitos del sistema
- Especificación del proyecto:
 - o Modelo de casos de uso -> Funcionalidades de nuestro sistema
 - o Modelo conceptual -> Definición de las clases y de las relaciones entre ellas
 - o Diagramas de secuencia -> Secuencia de iteración de las diferentes clases para cada uno de los casos de uso
 - o Contrato de las operaciones -> Definición de los requisitos de cada una de las operaciones
- Diseño del proyecto: Del modelo conceptual sacamos el diseño de nuestra aplicación
 - o Diagrama de clases de diseño
 - o Diagrama de secuencia
 - o Contrato de las operaciones
 - o Diseño de la base de datos
- Implementación de los módulos: Codificación de nuestras clases.
 - o Implementación de cada uno de los formularios de las entradas de datos para el área de administración, todos ellos implementados con PHP.
 - o Diseño de la apariencia de la interficie con la que operaran los usuarios con perfil usuario
 - o Informes estadísticos
 - o Introducción inicial de ciertos datos necesarios para la puesta en marcha de la aplicación
 - o Pruebas exhaustivas de todos los procesos
- Análisis y diseño de la aplicación orientado a objetos utilizando un modelo "Unified Modeling Language" UML que es un modelo para la construcción de programas orientado a objetos que el OMG ha propuesto como un estándar de ISO y que en estos últimos años se ha impuesto por todo el mundo. El UML consta de un conjunto de diagramas interrelacionados que sirven para describir diversos aspectos de la estructura y dinámica de la programación.

- Redactar la documentación necesaria.
- Preparar la presentación y demostración de la aplicación.
- Integración y puesta en funcionamiento de la aplicación en el website.

1.8.- Coste

Para mi ha sido importante realizar este proyecto por el aporte personal. En primer lugar, por tener que aprender un nuevo lenguaje de programación PHP además de ayudarme de otros lenguajes como Java, Html, XML aparte de los estilos. También ha habido un proceso de aprendizaje de UML para la parte de diseño y análisis de la aplicación. Además de las herramientas que he utilizado para el diseño de las pantallas de presentación de datos, aprendiendo a utilizar programas como el PhotoShop. También la creación de bases de datos dándoles integridad referencial y normalizándolas. Pero sobre todo lo más importante a nivel personal es el poder empezar y sacar adelante y acabar un proyecto por mi sola.

Por otra parte el coste económico de este proyecto ha sido mínimo ya que el software utilizado es de libre distribución y por tanto gratuito, este también es uno de los motivos principales en el momento de escoger los lenguajes de programación. El coste más grande viene dado por la empresa BabSoft que es la encargada de distribuir y vender el programa y que se tiene que encargar de dar mantenimiento, actualizaciones e incluso configurar un servidor de la empresa para dar soporte a aquellas empresas que compren el software y que utilicen el servicio vía internet, teniendo las bases de datos en la empresa BabSoft.

1.9.- Alternativas

Esta aplicación se ha realizado utilizando PHP y MySQL. Esta ha sido la mejor opción desde mi punto de vista para realizar el proyecto debido a su potencia y su orientación hacia internet. También se podrían haber utilizado otros lenguajes como ASP, JSP, etc

La elección del lenguaje se justifica a continuación:

Características que debía tener para dar un buen funcionamiento a la aplicación

- Tenía que interactuar con Bases de datos
- Tenía que ser un lenguaje del lado del servidor: "es lenguaje que se ejecuta en el servidor web, justo antes de que se envíe la página a través de internet al cliente" De esta manera se pueden realizar accesos a bases de datos, conexiones de red y otras tareas para crear la página final que verá el cliente.
- Y debía ser capaz de hacer una página web dinámica.
- Debía tener el mínimo coste económico posible.

Estos son algunos de los **motivos de la elección de PHP** como lenguaje de programación

1. Es software libre, lo que implica menores costes y servidores más baratos que otras alternativas, a la vez que el tiempo entre el hallazgo de un fallo y su resolución es más corto. Además, el volumen de código PHP libre es mucho mayor que en otras tecnologías, siendo superado por Perl, que es más antiguo. Esto permite construir sitios realmente interesantes con sólo instalar scripts libres como el conocido PHP Nuke.
2. Es muy rápido. Su integración con la base de datos MySQL, también veloz, le permite constituirse como una de las alternativas más atractivas para sitios de tamaño medio-bajo.
3. Su sintaxis está inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja, de modo que si estás familiarizado con esa sintaxis, PHP o JSP son las opciones más atractivas.
4. Su librería estándar es realmente amplia, lo que permite reducir los llamados "costes ocultos", uno de los principales defectos de ASP.

5. PHP es relativamente multiplataforma. Funciona en toda máquina que sea capaz de compilar su código, entre ellas diversos sistemas operativos para PC y diversos Unix. El código escrito en PHP en cualquier plataforma funciona exactamente igual en cualquier otra.
6. Su acceso a las bases de datos es muy heterogéneo, pues dispone de un juego de funciones distinto por cada gestor.
7. Es suficientemente versátil y potente como para hacer tanto aplicaciones grandes que necesiten acceder a recursos a bajo nivel del sistema como pequeños scripts que envíen por correo electrónico un formulario rellenado por el usuario.

Una vez elegido PHP por sus múltiples ventajas paso a definir el lenguaje y sus características:

Definición de PHP:

Es un lenguaje de secuencia de comandos de servidor diseñado específicamente para la Web. Dentro de una página Web se puede incrustar código PHP que se ejecutará cada vez que se visite la página. El código PHP es interpretado en el servidor Web y genera código HTML y otros contenidos que verá el usuario

PHP es un lenguaje de código abierto, lo que quiere decir que puede acceder a su código. Se puede utilizar, modificar y redistribuir sin coste alguno.

Características de PHP:

Rendimiento:

Mediante el uso de un único servidor, puede servir millones de accesos al día.

Integración de base de datos:

Dispone de una conexión propia a todos los sistemas de base de datos, además de MySQL, puede conectarse a las bases de datos de PostgreSQL, mSQL, Oracle,...

Bibliotecas incorporadas:

Como se ha diseñado para su uso en la Web, PHP incorpora una gran cantidad de funciones integradas para realizar útiles tareas relacionadas con la Web. Puede generar imágenes GIF al instante, establecer conexiones a otros servicios de red, enviar correos electrónicos, trabajar con cookies y generar documentos PDF, todo con unas pocas líneas de código.

Coste:

PHP es gratuito.

Aprendizaje:

La sintaxis de PHP se basa en otros lenguajes de programación, principalmente C y Perl.

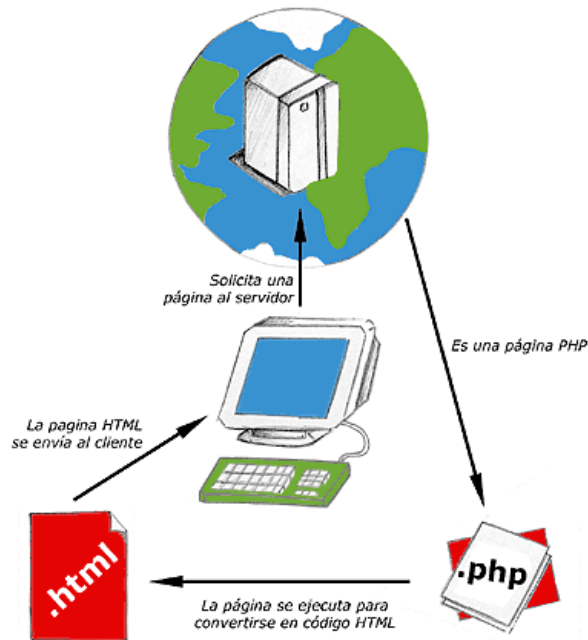
Portabilidad:

PHP está disponible para una gran cantidad de sistemas operativos diferentes. Se puede escribir código PHP en todos los sistemas operativos gratuitos del tipo Unix como Linux y FreeBSD, versiones comerciales de Unix, como Solaris e IRIX o en las diferentes versiones de Microsoft Windows.

Su código funcionará sin necesidad de aplicar ninguna modificación a los diferentes sistemas que ejecute PHP.

Código fuente:



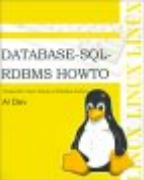
Se puede acceder al código fuente de PHP. A diferencia de los productos comerciales y de código cerrado, si se desea modificar algo o agregar un elemento al programa, puede hacerse con total libertad.



Esquema del funcionamiento de las páginas PHP.

MySql Vs PostgreSQL Vs MaxDB

Comparativa entre tres de las bases de datos de código abierto que ofrecen soluciones actualmente. MySql 4.0, PostgreSQL 7.4 y MaxDB 7.5

Bases De Datos	MySql	PostgreSQL	MaxDB
			
Versión:	Mysql-4.0.	PostgreSQL 7.4.x	MaxDB Versión 7.5
Licencia:	Dual: GPL y Comercial	BSD	GPL/LGPL
Plataformas:	Linux, Solaris, HP-UX, MacOS, AIX, SCO, IRIX, FreeBSD, NetBSD, OpenBSD, Windows, BSDI, DEC, OS/2, Compaq Tru64	Linux, Solaris, HP-UX, AIX, IRIX, FreeBSD, OpenBSD, NetBSD, MacOS, SCO OpenServer, SCO Unixware, BeOS, BSDI, Compaq Tru64, QNX, Windows	Linux, Solaris, HP-UX, Compaq Tru64, AIX, Windows
Velocidad	Media/Alta	Media	?
Estabilidad	Alta / Muy Alta	Alta	Media/Alta
Integridad de Datos	No	Si	Si
Características de Seguridad	Alta	Media/Alta	Media
Métodos de Autenticación soportados	Desconocido	md5, crypt, password and Kerberos	Desconocido
Acceso a SSL	Si	Si	?
Soporte para Concurrencia y Bloqueo	Medio/Alto	Alto	Alto
Soporte de Vistas	No	Si	Si
Soporte de Esquemas	No	Si	?
Soporte de Subselect	No	Si	Si
Procedimientos Almacenados	No	Si	Si
Soporte UniCode	No	Si	Si
Soporte de Triggers	No	Si	Si
Interfaces de Programación	ODBC, JDBC, C/C++, OLEDB, Delphi, Perl, Python, PHP	ODBC, JDBC, C/C++, Embedded SQL (in C), Tcl/Tk, Perl, Python, PHP	ODBC, JDBC, C/C++, Precompiler (Embedded SQL), Perl, Python, PHP
Tipos de Tablas Alternativas	ISAM, MYISAM, BerkeleyDB, InnoDB, HEAP, MERGE, Gemini	PostgreSQL tiene sus propios tipo de tablas construidas y ni permites ningunas otras	
Transacciones	Si	Si	Si
Claves Foráneas	No	Si	?

Replicación	Si	No	No
Balaceo de Cargas	No	No	No
Clustering	No	No	No
Hot Backups	Si	Si	Si

Características de Velocidad:

MySQL es muy rápido tanto para SELECTS simples como complejos, pero en cambio tiene que cambiar de base de datos a MySQL /MyISAM a MySQL /InnoDB para aplicaciones con muchos UPDATE. Los handles de conexión de MySQL son muy rápidos, esto hace que MySQL sea perfecta para su uso Web. Si tenemos centenares de CGI's conexiones y desconexiones continuamente, nos interesa evitar lanzamientos de procedimientos largos. MySQL es generalmente mucho más rápido que PostgreSQL.

Como tengo como fin que los accesos a las bases de datos sean lo más rápido posible para que no relanticen el trabajo esto me obliga a descartar PostgreSQL y continuar el estudio con los otros dos tipos de Bases de datos.

Estudio de las características entre MySql y MaxDB

MaxDB: Lista de características de las cuales carece MySQL: las vistas, los cursores en el servidor, procedimientos almacenados y disparadores (Triggers), sinónimos,...

MySQL: Las principales virtudes del MySQL son su gran velocidad, robustez y facilidad de uso. Posee replicación.

MySQL:

Ventajas principales:

Aumento de velocidad general y en especial en la búsqueda y creación de índices

Las tablas InnoDB se ofrecen como opción en el servidor estándar

Los clientes podrán conectar usando Secure Sockets Layer, conexiones seguras

Mayor compatibilidad con otras bases de datos

Es un servidor de bases de datos multiusuario

MaxDB

Ventajas principales:

Se reducen considerablemente el número de iteraciones entre cliente y servidor. La comunicación Cliente/Servidor sólo necesitará una iteración para cada operación y no para cada declaración SQL.

La capa de acceso de SQL contendrá los procedimientos de integridad y reglas comerciales. Concentrando estas reglas en el lado del servidor y eliminándolas de las aplicaciones de bases de datos, las modificaciones pueden estar centralizadas y de esta manera hacerse válidas inmediatamente en todas las aplicaciones de bases de datos. De esta manera, la integridad y las reglas de decisión también se hacen parte del catálogo en el sistema de base de datos.

Una capa de acceso de SQL en el formulario de procedimientos transferidos de la base de datos en el lado del servidor, es una esencial herramienta de personalización, esto permite incluir funcionalidades a un usuario específico de la base de datos.

Una vez estudiadas todas las características detenidamente de **MaxDB** y nombradas las principales de **MySQL** me encuentro que las diferencian las siguientes particularidades.

Comparativa de las partes importantes de bases de datos:

Integridad referencial	
<p>La integridad referencial es un sistema de reglas que utilizan la mayoría de las bases de datos relacionales para asegurarse que los registros de tablas relacionadas son válidos y que no se cambien datos relacionados de forma accidental produciendo errores de integridad. Característica que no ofrece la versión de MySQL 4.0.</p>	
Replicación	
<p>La Replicación nos ofrece Robustez y Velocidad.</p> <p>Para aumentar la robustez de dos o más sistemas existe la posibilidad de realizar un backup del servidor por si existen problemas con el servidor principal. Respecto a la velocidad se puede enviar parte de la consulta (no Update) al servidor replicado. La replicación también puede beneficiar las operaciones de backup eliminando así tiempos muertos</p> <p>Esta característica la ofrece MySql mientras que MaxDB solamente posibilita la carga y descarga completa de la tabla, pero no permite la replica en la línea de envío de una consulta o técnicas similares.</p>	
Soporte para Concurrencia y Bloqueo	
<p>MySQL: Produce un bloque de toda la tabla la cual se esta consultando.</p> <p>MaxDB: Sólo produce un bloqueo de la fila que se está consultando.</p>	
Triggers y Procesamientos Almacenados	
<p>Para que una consulta se lleve a cabo de manera eficiente es necesario que esté adecuadamente optimizada. El proceso de optimización es generalmente uno de los más costosos en el ciclo de trabajo del gestor en la obtención de datos. En entornos Clientes/ servidor las aplicaciones clientes enviarán consultas al gestor que deberá optimizarlas, procesarlas y devolverlos resultados. Una manera de optimizar es que el administrador cree por su cuenta las consultas más repetidas, las almacene en el servidor, previa optimización y las ofrezca a los clientes, facilitando su operación y uniformando los formatos de las peticiones de datos. Y es por eso que se utilizan los procedimientos Almacenados.</p> <p>Procedimientos Almacenados:</p> <p>Son una colección de sentencias Transact SQL que se constituye como si se tratase de una función de lenguaje estructurado. Es posible llamarla mediante un identificador, puede recibir argumentos y devolver un valor de retorno. La</p>	

característica principal de los procedimientos almacenados es que se optimizan en el momento de su creación, con lo que proporciona más velocidad en la ejecución. **MySql** no posee esta característica mientras que **MaxDB** si que la posee.

Los triggers:

También es importante de dotar a la base de datos de una serie de mecanismos para que ella misma se mantenga siempre coherente, respondiendo de manera automática a las modificaciones de los datos y realizando las actualizaciones que sean necesarias. Estos mecanismos son los triggers.

Un trigger no es más que un tipo especial de procedimiento almacenado que en lugar de ejecutarse como respuesta a una llamada, se pone en funcionamiento automáticamente como respuesta a ciertas modificaciones de los datos, cuya naturaleza se especifica en el momento de la creación del trigger.

Los triggers son una herramienta poderosa para centralizar en la base de datos. Las acciones que motivarán que un trigger se ponga en ejecución, es decir aquellas acciones que disparan el trigger, son modificaciones que puedan llevarse a cabo en los datos, tanto la adición de datos nuevos como la eliminación de datos existentes. Dichas modificaciones se llevan a cabo, evidentemente, mediante la ejecución de las sentencias UPDATE, DELETE o INSERT.

Después de listar las características llego a la conclusión que el mejor sistema gestor de bases de datos es **MaxDB** debido a que nos ofrece más prestaciones que **MySQL** y son prestaciones útiles para cualquier aplicación que interaccione con bases de datos.

Pero debido a lo altamente que está ligado **PHP con MySql** y todas las funciones que éste presenta para poder acceder a este gestor de base de datos y además teniendo en cuenta que también tiene características de velocidad que es lo que más nos interesa para un formato Web, intentaré suplir las características importantes como las integridad referencial y las coherencias de datos a través de código.

2.- Planificación

En este apartado he querido realizar una planificación con los costes reales de este proyecto, tanto si lo hubiera realizado para una empresa BabSoft o si lo hubiera realizado por cuenta propia.

A continuación paso a explicar las diferentes definiciones de los términos que utilizaré en mi planificación.

2.1.- Definición de la jornada laboral

La jornada laboral la definiré como 8 horas durante 5 días a la semana.

2.2.- Determinación de actividades

A continuación especificaré las diferentes actividades que se realizarán a lo largo del proyecto:

- **Análisis:** Esta etapa nos proporcionará una visión general de lo que es este proyecto y nos determinará el alcance y los objetivos de éste. También detallaré minuciosamente las funciones que realizará este proyecto.
- **Especificación:** Esta parte del proyecto se detallarán las funcionalidades que se proponen (Modelos de casos de uso), se determinará el dominio del sistema (Modelo conceptual) y se realizará un estudio exhaustivo del comportamiento del sistema (Modelo de comportamiento del sistema).
- **Diseño:** Durante esta etapa, aplicaremos los diferentes patrones arquitectónicos y de diseño vistas a lo largo de la carrera, normalizaremos el Modelo Conceptual (Diagrama de clases de diseño), se revisará el Modelo de comportamiento del sistema siguiendo los cambios realizados en el Modelo Conceptual, es decir, siguiendo el Diagrama de Clases de diseño y se realizará el diseño de la base de datos que sean necesaria siguiendo el modelo relacional.
- **Implementación:** Esta etapa se codificará nuestro sistema de acuerdo con la especificación y diseño establecido en etapas anteriores. Al final de esta etapa, se realizarán estrictas pruebas reales para determinar el correcto funcionamiento de nuestro sistema.

2.3.- Planificación inicial

Para el cálculo de planificación de este proyecto, utilizaremos un ciclo de vida clásico. Este ciclo determina que no podemos realizar la siguiente actividad hasta no haber acabado la actual.

Cada actividad la realizará un determinado perfil profesional. En función de estos perfiles, se asignan los recursos a las diferentes actividades que se contemplan en este proyecto.

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1	Proyecto TPV	86 días	lun 26/06/06	dom 31/12/06		
2	Análisis	10 días	lun 26/06/06	mié 19/07/06		
3	Definición del sistema	4 días	lun 26/06/06	vie 07/07/06		Analista
4	Análisis de requisitos del sistema	4 días	vie 07/07/06	sáb 15/07/06	3	Analista
5	Documentación	2 días	sáb 15/07/06	mié 19/07/06	4	Analista
6	Especificación	12 días	jue 20/07/06	lun 14/08/06		
7	Modelo de casos de uso	3 días	jue 20/07/06	mié 26/07/06	5	Especificador
8	Modelo conceptual	2 días	mié 26/07/06	dom 30/07/06	7	Especificador
9	Diagramas de secuencia	3 días	dom 30/07/06	dom 06/08/06	8	Especificador
10	Contrato de las operaciones	2 días	dom 06/08/06	sáb 12/08/06	9	Especificador
11	Documentación	2 días	sáb 12/08/06	lun 14/08/06	10	Especificador
12	Diseño	15 días	lun 14/08/06	dom 17/09/06		
13	Diagrama de clases de diseño	2 días	lun 14/08/06	dom 20/08/06	11	Diseñador
14	Diagramas de secuencia	4 días	dom 20/08/06	dom 27/08/06	13	Diseñador
15	Contrato de las operaciones	4 días	dom 27/08/06	jue 07/09/06	14	Diseñador
16	Diseño de la BD	2 días	vie 08/09/06	dom 10/09/06	15	Diseñador
17	Documentación	3 días	dom 10/09/06	dom 17/09/06	16	Diseñador
18	Implementación de los módulos	49 días	dom 17/09/06	dom 31/12/06		
19	Login	1 día	dom 17/09/06	mié 20/09/06	17	Programador
20	Panel de control o frames	2 días	mié 20/09/06	dom 24/09/06	19	Programador
21	Mantenimiento de familias	4 días	dom 24/09/06	lun 02/10/06	20	Programador
22	Mantenimiento de modelos	3 días	mar 03/10/06	lun 09/10/06	21	Programador
23	Mantenimiento de artículos	6 días	lun 09/10/06	dom 22/10/06	22	Programador
24	Mantenimiento de usuarios	2 días	dom 22/10/06	sáb 28/10/06	23	Programador
25	Mantenimiento de tiendas	2 días	sáb 28/10/06	jue 02/11/06	24	Programador
26	Mantenimiento de temporadas	1 día	vie 03/11/06	sáb 04/11/06	25	Programador
27	Mantenimiento de arreglos de modista	1 día	sáb 04/11/06	dom 05/11/06	26	Programador
28	Mantenimiento de stock	3 días	dom 05/11/06	dom 12/11/06	27	Programador
29	Envío y/o traspaso de stock a tiendas	3 días	dom 12/11/06	dom 19/11/06	28	Programador
30	Informes estadísticos	10 días	dom 19/11/06	jue 07/12/06	29	Programador
31	Introducción inicial de los datos	1 día	jue 07/12/06	vie 08/12/06	30	Programador
32	Pruebas finales	4 días	vie 08/12/06	sáb 16/12/06	31	Programador
33	Documentación	6 días	sáb 16/12/06	dom 31/12/06	32	Programador

Planificación prevista del proyecto desglosada por tarea y perfil profesional

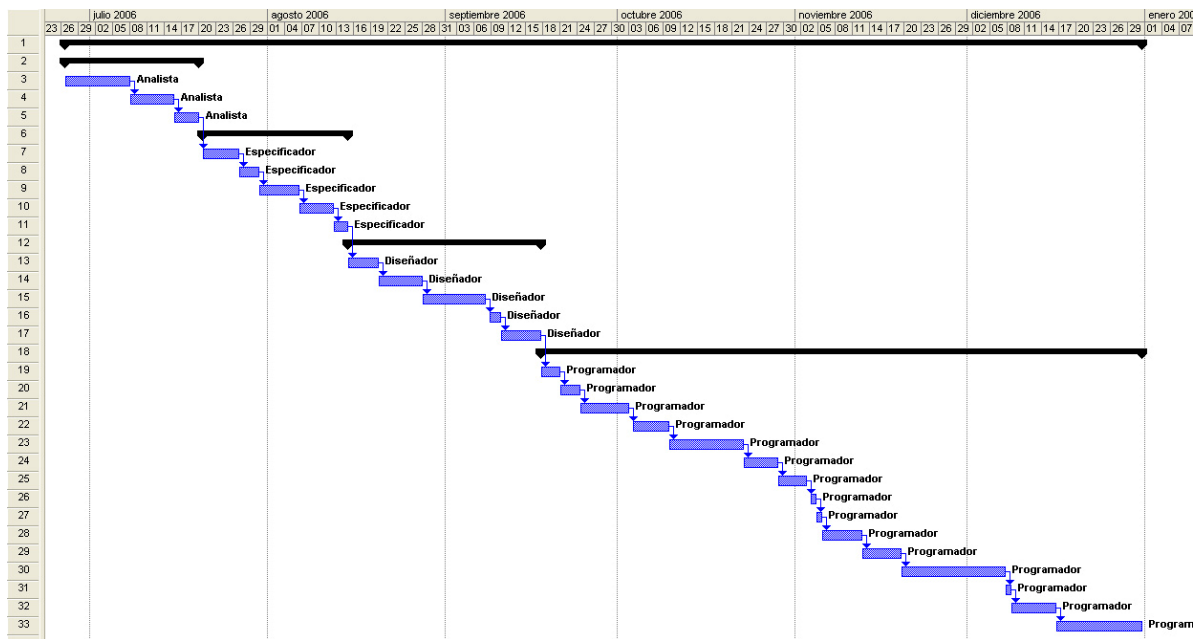


Diagrama de Gantt del proyecto

Estas dos imágenes nos muestran las tareas a realizar. También se puede observar las tareas a realizar por cada perfil.

Aunque hagamos la distinción de perfiles, todo el trabajo aquí planificado será realizado por una sola persona. Se han dividido las tareas por perfiles para poder dar un coste económico del proyecto, ya que el coste de cada perfil es diferente.

Mediante esta estimación, se ha establecido que el proyecto podrá ser finalizado en un plazo de 86 días hábiles. Esta estimación se ha realizado utilizando un ciclo de vida clásico y realizando una estimación de horas de trabajo basado en experiencias anteriores.

2.4.- Coste Inicial

2.4.1.- Estimación inicial como trabajador por cuenta ajena

Para la estimación inicial del coste del proyecto sólo voy a tener en cuenta el coste de mano de obra que supone la realización del mismo.

Por otra parte, hay otros costes que no incluiremos en ningún momento en el coste del proyecto. Estos costes son:

- Software utilizado para el desarrollo de las aplicaciones ya que es software gratuito.
- Software utilizado para el desarrollo y diseño de la base de datos.
- Software utilizado para el diseño del modulo.
- Licencias de los sistemas operativos utilizados en el desarrollo.

En un principio, no se tendrá que adquirir ningún nuevo elemento hardware necesario para el desarrollo de la aplicación. Las licencias de los sistemas operativos, software utilizado para el desarrollo de la aplicación ya están sufragados por la empresa antes del inicio del proyecto, por eso no se incluirán como costes directos de la realización del mismo.

A continuación mostramos la estimación inicial del coste de realización del proyecto.

Nombre Recurso	Tasa Estándar	Trabajo Previsto	Coste Previsto
Analista	30,00 €/hora	80 Horas	2.400,00 €
Especificador	30,00 €/hora	96 horas	2.880,00 €
Diseñador	30,00 €/hora	120 Horas	3.600,00 €
Programador	24,00 €/hora	392 Horas	9.408,00 €
Total		688 Horas	18.288,00 €

Desglose de los costes por tarea.

Nombre	Duración	Trabajo previsto	Coste previsto
Proyecto TPV	86 días	688 horas	18.288,00 €
Análisis del sistema	10 días	80 horas	2.400,00 €
Definición del sistema	4 días	32 horas	
Obtención de información de cliente	2 días	16 horas	
Recogida de información del cliente	1 días	8 horas	
Visita a la fábrica del cliente	1 días	8 horas	
Definición de las características principales del sistema	2 días	16 horas	
Análisis de requisitos del sistema	4 días	32 horas	
Análisis de requisitos del cliente	1 día	8 horas	
Documentación	2 días	16 horas	
Especificación	12 días	96 horas	2.880,00 €
Modelo de casos de uso	4 días	24 horas	
Modelo conceptual	2 días	16 horas	
Diagramas de secuencia	4 días	24 horas	
Contrato de las operaciones	2 días	16 horas	
Documentación	2 días	16 horas	
Diseño	15 días	120 horas	3.600,00 €
Diagrama de clases de diseño	2 días	16 horas	
Diagramas de secuencia	8 días	32 horas	
Contrato de las operaciones	8 días	32 horas	
Diseño de Base de datos	2 días	16 horas	
Documentación	3 días	24 horas	
Programador	49 días	392 horas	9.408,00 €
Validación	1 día	8 horas	
Panel de control o frames	2 días	16 horas	
Mantenimiento	25 días	200 horas	
Informes	10 días	80 horas	
Introducción inicial de los datos	1 días	8 horas	
Pruebas finales	4 días	32 horas	
Documentación	6 días	48 horas	

2.4.2.- Estimación inicial como trabajador por cuenta propia

En esta estimación se va a tratar la estimación del coste del proyecto desde una vista empresarial propia. Para ello seguiremos todos los trámites necesarios para la constitución de una empresa propia, calculando todos los gastos necesarios para ello.

Creo que es importante saber hacer esta gestión y así poder sacar partido a otras asignaturas que realicé durante la carrera de créditos libres.

También realizaré los cálculos de los costes fijos de una empresa, de esta manera imputaremos dichos costes al proyecto.

Para todas las decisiones y cálculos de costes, tomaremos como premisa que la empresa residirá en la ciudad de Barcelona.

2.4.2.1.- Elección de la forma jurídica de empresa

Teniendo en cuenta las siguientes características seleccionaremos la forma jurídica mejor para este proyecto:

Partiremos de la idea que el proyecto se llevará a cabo por una sola persona.

Todos los riesgos los asumirá la persona que realice el proyecto.

La empresa tiene que ser ágil y se puede dedicar a otros trabajos para poder financiar el proyecto a realizar.

La inversión inicial debe ser mínima.

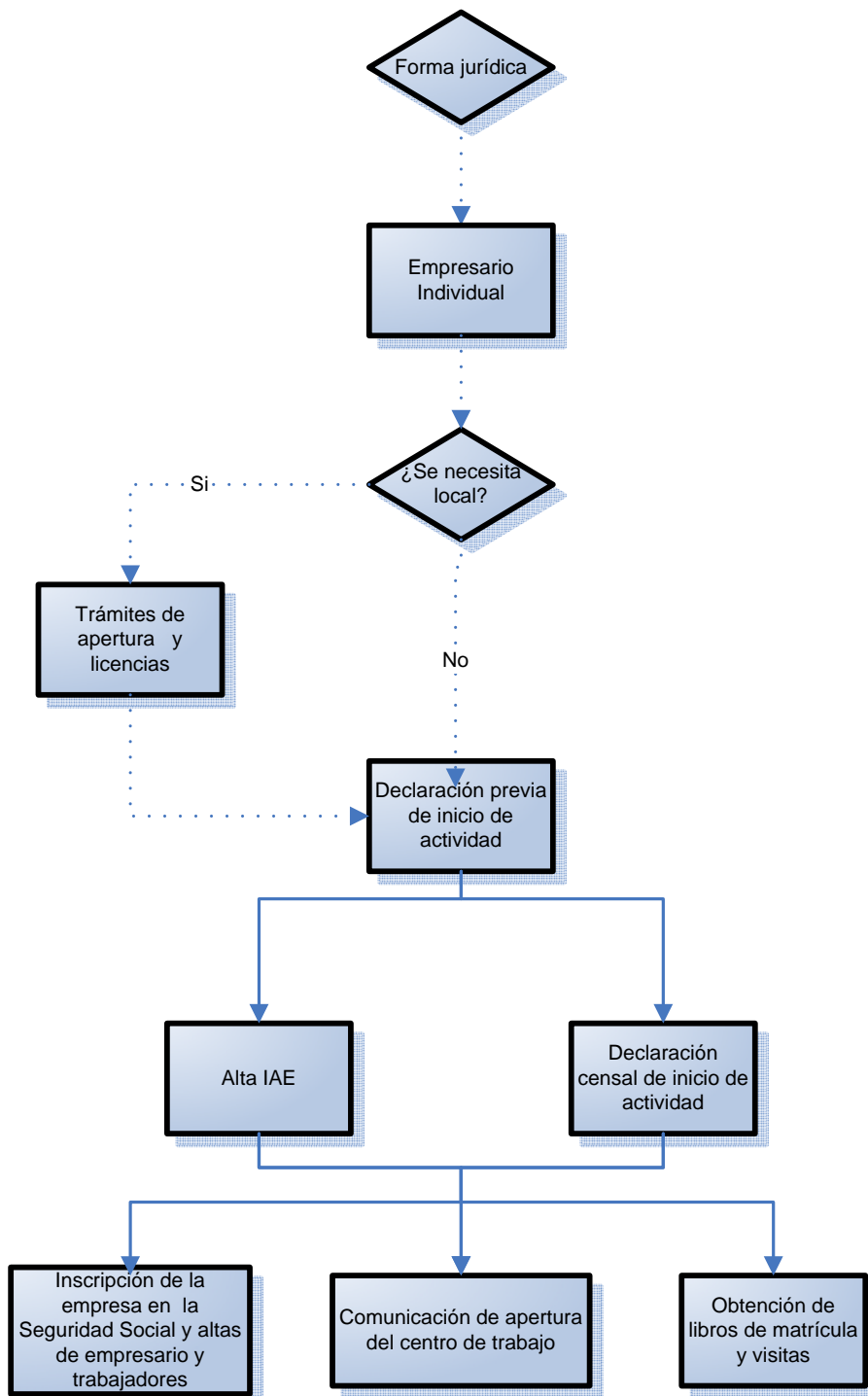
Teniendo en cuenta todas estas características la forma elegida será la de empresario individual, popularmente llamado autónomo.

Tomada la decisión de la forma jurídica pasaremos a describir todos los pasos necesarios para la creación de esta, y los gastos que esta creación conlleva.

2.4.2.2.- Trámites y costes derivados por la constitución de la empresa

Para la creación de la empresa tendremos en cuenta que necesitaremos un local para atender a los clientes y para desarrollar nuestro proyecto.

Los trámites para la constitución de la empresa son los siguientes:



1. **Decisión de forma jurídica de la empresa:** Como ya se ha explicado anteriormente, la forma jurídica seleccionada es la de autónomo. Los costes derivados de esta decisión son los siguientes:
 - a. **Cuotas a la seguridad social:** para realizar el cálculo de los gastos de seguridad social primero tendremos que seleccionar una base de cotización. Nosotros tomaremos como base de cotización la mínima permitida. Teniendo cuenta que el empresario tendrá menos de 30 años, la base de cotización mínima es de 631,20 € mensuales y el tipo de cotización es del 26,50 %. Una vez indicado esto, calculamos que el gastos de esta cuota mensual es de: **167€ mensuales**.
 - b. **Pago de impuesto IRPF:** el cálculo de este impuesto se basa en los rendimientos de y se pueden realizar deducciones por varios motivos. Debido a que la empresa es de nueva constitución, y no habrán grandes rendimientos, y que se podrán realizar deducciones por distintos motivos, este pago los vamos a descartar para el cálculo de los gastos.
 - c. **Pago de impuesto IVA:** para realizar este cálculo decidimos entre todas las posibilidades fiscales que tenemos la de Estimación Directa. Los rendimientos de actividades profesionales y empresariales se van pagando a cuenta trimestralmente. Se paga el 20% del rendimiento de cada trimestre. Básicamente, consiste en restar el IVA repercutido (la cuantía cobrada a los clientes) y el IVA soportado (la que le han cobrado a usted los proveedores). Como en un principio no vamos a tener ningún rendimiento, este pago lo vamos a descartar.
 - d. **Sueldo:** debido a la decisión de crear una empresa propia, tenemos que definir un sueldo. Realizando una comparativa de los sueldos actuales del mercado, calculamos un sueldo de **1500 €**, en los que irían incluidos las dietas del trabajador.

2. **Decisión de necesidad de local:** sí que es necesario un local para las reuniones con clientes y el desarrollo del proyecto. Los costes derivados de esta decisión son los siguientes:
 - a. **Coste del alquiler de la oficina:** elegimos una oficina en Eixample Izquierdo de Barcelona de 65 m². 680 € Mensuales.
 - b. **Mobiliario:** en un principio no sabemos el mobiliario necesario. Para empezar necesitaremos poco mobiliario. Visitando las paginas de fabricantes, estimamos un coste de 600 € para empezar.
 - c. **Luz:** teniendo en cuenta la media de una pequeña oficina, incluiremos un gastos de 300 € mensuales.
 - d. **Agua:** la tarifa de contratación supone un gasto de 49,46 €. Y el gasto medio mensual es de 30 €.
 - e. **Teléfono:** realizamos el cálculo haciendo pequeño estudio de las diferentes ofertas telefónicas. El alta de la línea la consideraremos gratuita y el gasto mensual será 200 € mensuales.
 - f. **Viajes:** en este apartado contabilizaremos el gastos medio en viajes y visitas a clientes. Estimando que tenemos un vehículo con un gasto de 8 litros cada 100 Km., el combustible utilizado es diesel y realizamos unos 30 Km. Diarios: $(30 \text{ Km./Día} \times 30 \text{ Días}) * (8 \text{ Litros/100 Km.}) * 1\text{€/Litro} = 112,5 \text{ € mensuales}$.
3. **Trámites de apertura y licencias:** por lo que hemos podido estudiar, para este tipo de negocio en Barcelona, debemos realizar una solicitud de licencia municipal de apertura de establecimientos. El coste es de 688.30 € más un 3,25 % del presupuesto de las instalaciones. Esta ultima tasa se pagará al retirar la resolución del expediente, por ello no lo tendremos en cuenta.
4. **Declaración previa de inicio de actividad:** trámite que debemos realizar en la Agencia Tributaria. Nos habilita a recibir facturas y por lo tanto deducir cuotas de IVA soportados, además de poder liquidar posteriormente para el caso de Sociedades el impuesto de Transmisiones Patrimoniales y Actos jurídicos documentados. El coste de este tramite es de 0,54 €.
5. **Alta IAE:** trámite que debemos realizar en la Agencia Tributaria. Debido a nuestro modelo de empresa, quedamos exentos de este impuesto. Mediante esta alta, nuestra empresa quedara clasificada automáticamente en la Agencia Tributaria. Conoceremos la cuota inicial de su IAE (a la que luego se aplican unos coeficientes correctores); si nuestra actividad está o no sometida al Impuesto sobre el Valor Añadido (IVA); y cuáles son las opciones que tiene para

calcular los rendimientos (ganancias o pérdidas) de nuestro negocio a efectos del IRPF.

6. **Inscripción al censo de la Agencia Tributaria:** obligatoria para ejercer una actividad empresarial, profesional o artística.
7. **Inscripción de la empresa en la Seguridad Social y altas de empresario y trabajadores:** tramite que debemos cursar en la Tesorería de la Seguridad social. Este trámite es gratuito.
8. **Comunicación de apertura del centro de trabajo:** esta comunicación tiene que ser presentada cualquier empresa que realice una actividad empresaria y tenga o no tenga trabajadores a su cargo. Se realiza en la Delegación Territorial del departamento de trabajo y es gratuita.
9. **Obtención de libros de matrícula y visitas:** este libro se tiene que adquirir en la Delegación de Trabajo y Seguridad Social. Tramite gratuito. Dicho libro lo tenemos que tener siempre disponible para las posibles Inspecciones de trabajo.

2.4.2.3.- Costes derivados del desarrollo

Para realizar el desarrollo de este proyecto necesitamos el software necesario para este, y hardware para la empresa. La elección del software necesario para el desarrollo, vendrá dada por las características del proyecto. Para el hardware, se ha utilizado un elemento de financiación mensual, que nos permite poder cambiar de hardware con facilidad, renting. Pasamos a detallar la elección y costes de software y hardware.

2.4.2.3.1.- Hardware

Para el hardware hemos elegido un servidor y una estación de trabajo que nos permitirá realizar el desarrollo del proyecto y realizar las pruebas necesarias.

1. Servidor: Dell PowerEdge SC430 SATA



Características	
Modelo	Procesador Intel® Pentium® D 820 (2.80 GHz, 2x1MB cache , 800 MHz FSB) Dual Core
Servicios de Garantía	Garantía de 1 año - Respuesta In Situ el próximo día laborable
Servicios de Garantía	Base warranty
Memoria	512MB DDR2 SDRAM 533MHz (2X256MB Single Ranked DIMMs), ECC
Licencias	NEW! Microsoft Windows Server 2003 R2, x64 Standard Edition with 5 Client Licenses (incl. CD&doc)
Systems Management Software	Dell Server Assistant for PowerEdge SC Servers CD electronic documentation
Raid Connectivity	C1 Onboard SATA, One Drive connected to Onboard Controller
Disco duro	160GB 7200rpm SATA hard drive
2º Disco Duro	160GB 7,200rpm SATA hard drive
Unidad óptica	CD Drive, Internal, Half Height
Coste	Renting 20 € mensuales

2. Estación de trabajo: Precision 380 ESSENTIAL



Características	
Modelo	Intel® Pentium® D Processor 920 (2.80GHz, 800MHz, 2x2 MB Cache, Dual Core Technology)
Sistema operativo	Windows® XP Professional original, SP2 (NTFS)(+ Media) Hyper - Threading: Procesador Hyper Threading
Servicios de Garantía	3Y NBD (Next Business Day On-site)
Vertical Chassis Orientation	(Minitower) with 2 x 5.25" and 2 x 3.5" Front Bays
Memoria	Memoria de 512MB (2 x 256MB DIMM) 533MHz DDR2 Non- ECC SDRAM
Disco duro	160GB (7,200 rpm) Serial ATA II Hard Drive with NCQ
Unidad óptica	48x DVD-ROM/CD-RW Combo Drive
Disquetera y Almacenamientos adicionales	Media Card Reader integrada
Sin Documentación	Dell OpenManage Client Instrumentation
Monitor	Dell 17" E773c (15.9 VIS) Value TCO'03 DualTone Monitor
Tarjeta gráfica	64MB PCIe x16 (DVI/VGA) nVidia Quadro NVS 285, Dual Monitor DVI or VGA Graphics Card
IEEE 1394	1394 (Firewire) Controller Card
Altavoces	Altavoces internos
Tarjeta de Sonido	Info, no Soundcard, no IEEE1394
Teclado	Quikey™ USB Spacesaver Midnight Grey keyboard
Ratón	Ratón Dell Midnight Grey USB de 2 botones
Servicio Técnico Gold	Support 3 años (con e-Learning Pack)
Documentación	{Spanish} - System Documentation (Spanish)
Impresora	Impresora Multifunción Personal 924
Coste	Renting 41 € mensuales

3. Gastos varios:

Realizamos una estimación mensual de 100 €, para gastos en consumibles y otros gastos relacionados con el hardware de la empresa.

2.4.2.3.2.- Software

La elección del software la hemos realizado teniendo en cuenta las características del proyecto, los lenguajes de programación y los entornos de utilización de la aplicación. El sistema seleccionado tanto para el servidor como para la estación de trabajo ha sido Windows. El motivo principal ha sido la realización de otros proyectos desarrollados con Microsoft Visual Studio.

Para este proyecto sólo se utilizan las herramientas especificadas a continuación, pero para la empresa se decidió comprar un paquete que también incluyen herramientas de desarrollo de Microsoft. El coste del conjunto de este paquete es de 1195 €.

1. Herramientas de desarrollo: Todas las herramientas de desarrollo que hemos seleccionado, son gratuitas. Esto lo hemos hecho así para poder minimizar los costes.
2. Software de gestión y sistemas operativos:

Para estos propósitos hemos elegido un paquete que nos ofrece estas características:

- Acceso continuo a nuevo software y recursos técnicos. Durante la vigencia de su suscripción, tiene derecho a actualizaciones del software incluido en su suscripción y a utilizar nuevas versiones y betas de producto, además de otros recursos como paquetes de desarrollo de software y hardware, y service packs.
- Los últimos sistemas operativos de Microsoft incluyen las últimas versiones de Windows Server 2003, Windows 2000 Server, Windows 2000 Professional, Windows 2000 Advanced Server, Windows XP Professional Edition, Windows XP Home Edition, y futuras versiones de Windows. Los derechos de uso están limitados al desarrollo y a la realización de pruebas únicamente.

2.4.2.4.- Cuadro estimación resumen de costes y coste del proyecto.

Para realizar los siguientes cálculos se tomará la jornada laboral definida de 8 horas por día ,20 días laborables por mes y la duración del proyecto es de 688 h.

Concepto	Importe	Pago	Años Amortización	Formula coste hora	Coste hora	Coste repercutido proyecto
Costes derivados de la empresa						
Costes creación empresa						
Trámites de apertura y licencias	688,30 €	Único	2	Coste / (AñosAmortiz.x 240 días x 8 h)	0,18 €	123,32 €
Declaración previa de inicio de actividad	0,54 €	Único	2	Coste / (AñosAmortiz.x 240 días x 8 h)	0,00 €	0,10 €
Mobiliario	600,00 €	Único	2	Coste / (AñosAmortiz.x 240 días x 8 h)	0,16 €	107,50 €
Instalación Agua	49,26 €	Único	2	Coste / (AñosAmortiz.x 240 días x 8 h)	0,01 €	8,83 €
Alta IAE	0,00 €	Único	2	Coste / (AñosAmortiz.x 240 días x 8 h)	0,00 €	0,00 €
Impuestos y sueldo						
Cuotas a la seguridad social	167,00 €	Mensual		Coste / (20 días x 8 h)	1,04 €	718,10 €
Sueldo	1.500,00 €	Mensual		Coste / (20 días x 8 h)	9,38 €	6.450,00 €
Gastos mensuales varios						
Alquiler de la oficina	680,00 €	Mensual		Coste / (20 días x 8 h)	4,25 €	2.924,00 €
Agua	30,00 €	Mensual		Coste / (20 días x 8 h)	0,19 €	129,00 €
Luz	300,00 €	Mensual		Coste / (20 días x 8 h)	1,88 €	1.290,00 €
Teléfono	200,00 €	Mensual		Coste / (20 días x 8 h)	1,25 €	860,00 €
Transporte	112,50 €	Mensual		Coste / (20 días x 8 h)	0,70 €	483,75 €
Software y Hardware						
Hardware						
Servidor	20,00 €	Mensual		Coste / (20 días x 8 h)	0,13 €	86,00 €
Estación de trabajo	41,00 €	Mensual		Coste / (20 días x 8 h)	0,26 €	176,30 €
Software	1.195,00 €	Único	2	Coste / (AñosAmortiz.x 240 días x 8 h)	0,31 €	214,10 €
Consumibles	100,00 €	Mensual		Coste / (20 días x 8 h)	0,63 €	430,00 €
Total coste					20,35 €	14.001,00 €

Con este estudio de cálculo de coste como trabajador por cuenta propia, podemos observar que el proyecto podría salir más económico, 4287 € menos. Todo y esto, cabe remarcar que en este cálculo no contabilizamos los riesgos. Tampoco contabilizamos otros elementos que se incluirían a lo largo del proyecto, puesto que esto es sólo una estimación y carezco de experiencia previa en la creación y gestión de una empresa.

2.5.- Desviación de la planificación inicial

En este punto, comentaré el tiempo real dedicado a la realización de este proyecto y la desviación que se ha producido respecto a la planificación inicial.

La desviación se ha producido básicamente en el apartado de implementación de la aplicación, ya que han surgido problemas que han retrasado los desarrollos de los diferentes mantenimientos. Los problemas surgidos han hecho que me estancará, hasta poder conseguir encontrar una solución realizando búsquedas a páginas de ayuda en internet o a documentación de libros de php. Al realizar el proyecto por mi cuenta, es algo que podía pasar, pero el encontrar dificultades es algo que refuerza el aprendizaje al encontrar la solución.

La desviación de la planificación inicial ha sido aproximadamente un mes más de trabajo. 20 días laborales.

La desviación ha sido de muchos días eso demuestra la falta de experiencia por mi parte de un proyecto de estas características. Ya que anteriormente no había realizado ninguna aplicación que se pareciera en nada a la actual.

2.6.- Coste final y análisis de desviaciones

2.6.1.- Coste final como trabajador por cuenta ajena

El incremento del coste final serán los 20 días de más que ha aumentado la parte de desarrollo de la aplicación.

Esto hace que la parte de programación pase de 49 días y 392 horas

Programador	49 días	392 horas	9.408,00 €
-------------	---------	-----------	------------

A 69 días y 552 horas.

Programador	69 días	552 horas	13.248,00 €
-------------	---------	-----------	-------------

De tal manera que el total del coste del proyecto TPV pasa de

Proyecto TPV	86 días	688 horas	18.288,00 €
--------------	---------	-----------	-------------

A

Proyecto TPV	106 días	848 horas	22.128,00 €
--------------	----------	-----------	-------------

2.6.2.- Coste final como trabajador por cuenta propia

El incremento del coste final serán los 20 días de más que ha aumentado la parte de desarrollo de la aplicación.

Cambia parte de la tabla, ya que al aumentar un mes más de desarrollo, es un mes mas de gastos que tiene que sufragar la empresa.

Concepto	Importe	Pago	Años Amortización	Formula coste hora	Coste hora	Coste repercutido proyecto
Costes derivados de la empresa						
Costes creación empresa						
Trámites de apertura y licencias	688,30 €	Único	2	Coste / (AñosAmortiz.x 240 días x 8 h)	0,18 €	152,00 €
Declaración previa de inicio de actividad	0,54 €	Único	2	Coste / (AñosAmortiz.x 240 días x 8 h)	0,00 €	0,12 €
Mobiliario	600,00 €	Único	2	Coste / (AñosAmortiz.x 240 días x 8 h)	0,16 €	132,50 €
Instalación Agua	49,26 €	Único	2	Coste / (AñosAmortiz.x 240 días x 8 h)	0,01 €	10,88 €
Alta IAE	0,00 €	Único	2	Coste / (AñosAmortiz.x 240 días x 8 h)	0,00 €	0,00 €
Impuestos y sueldo						
Cuotas a la seguridad social	167,00 €	Mensual		Coste / (20 días x 8 h)	1,04 €	885,10 €
Sueldo	1.500,00 €	Mensual		Coste / (20 días x 8 h)	9,38 €	7.950,00 €
Gastos mensuales varios						
Alquiler de la oficina	680,00 €	Mensual		Coste / (20 días x 8 h)	4,25 €	3.604,00 €
Agua	30,00 €	Mensual		Coste / (20 días x 8 h)	0,19 €	159,00 €
Luz	300,00 €	Mensual		Coste / (20 días x 8 h)	1,88 €	1.590,00 €
Teléfono	200,00 €	Mensual		Coste / (20 días x 8 h)	1,25 €	1.060,00 €
Transporte	112,50 €	Mensual		Coste / (20 días x 8 h)	0,70 €	596,25 €
Software y Hardware						
Hardware						
Servidor	20,00 €	Mensual		Coste / (20 días x 8 h)	0,13 €	106,00 €
Estación de trabajo	41,00 €	Mensual		Coste / (20 días x 8 h)	0,26 €	217,30 €
Software	1.195,00 €	Único	2	Coste / (AñosAmortiz.x 240 días x 8 h)	0,31 €	263,90 €
Consumibles	100,00 €	Mensual		Coste / (20 días x 8h)	0,63 €	530,00 €
Total coste					20,35 €	17.257,04 €

Pasamos de un coste de

Total coste					20,35 €	14.001,00 €
--------------------	--	--	--	--	---------	-------------

A un coste de

Total coste					20,35 €	17.257,04 €
--------------------	--	--	--	--	---------	-------------

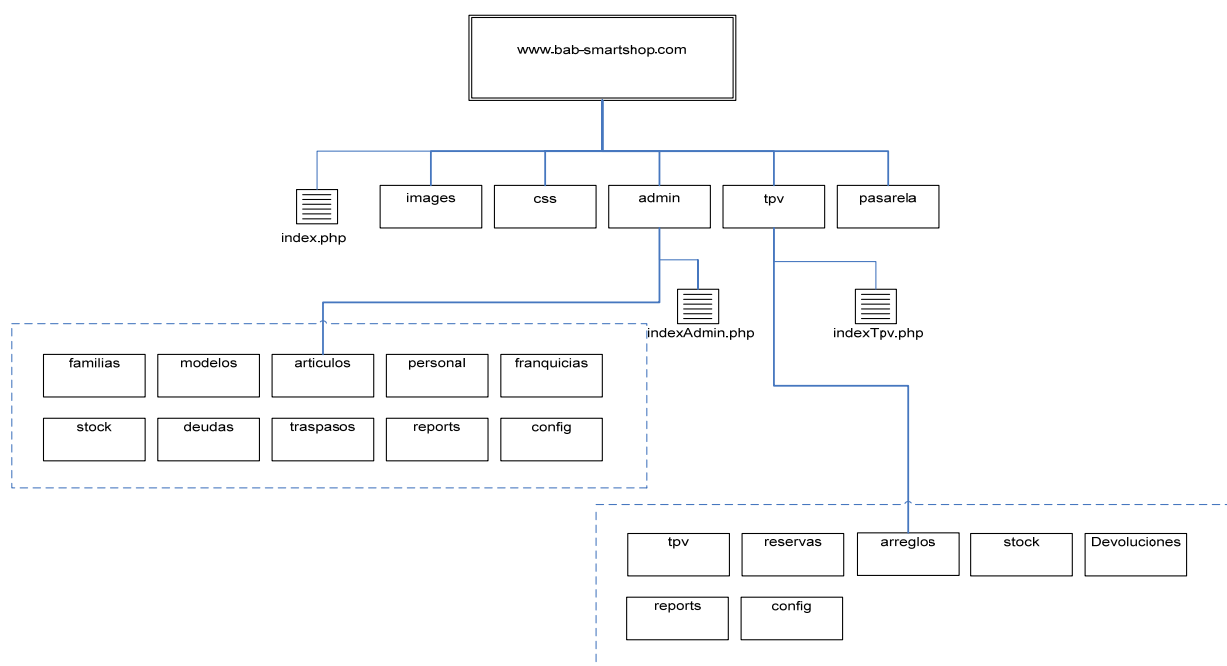
El coste de la desviación ha aumentado en **3.256,00 €**, el coste final de la aplicación.

3.- Análisis de Requisitos

Primero analizaré las funcionalidades que tiene que realizar tanto el módulo de administración como el de TPV. También empezaré a mostrar la arquitectura de la aplicación y la tecnología que utilizaremos. En el siguiente esquema se muestra la estructura de la aplicación.

- **Estructura del Sitio web**

La estructura de directorios del sitio web



3.1.- Descripción de la aplicación

La aplicación consta de dos módulos el de administración y el de TPV.

Para ambos módulos diferenciaremos dos tipos de perfil de usuarios:

1. El Administrador: que podrá encargarse de gestionar toda la aplicación sin ningún tipo de barreras.
2. El usuario: que tendrá limitaciones para entrar en algunas partes de la aplicación. Las partes a las que tendrá acceso será el propio dueño de la aplicación quien nos lo facilitará, configurando la aplicación a su gusto.

De esta manera intentaremos que no se eliminen datos de productos ni de cantidades que puedan hacer perder la coherencia de los datos, además de evitar

que usuarios mal intencionados puedan modificar datos para así obtener beneficios personales. Esto también le permitirá al administrador ver informes y estadísticas del trabajo de sus vendedores, ya que todas las entradas a la base de datos tienen un campo que indica el número de usuario que ha hecho la gestión.

Por defecto en la aplicación se introducen estos dos tipos de perfiles, pero no hay ningún impedimento para que se pueden añadir más tipos de perfiles y darles a cada uno de estos los privilegios que el Cliente que compra la aplicación proponga más adecuado para el buen funcionamiento y gestión de la misma.

Funcionalidades de la aplicación:

1. Módulo de Administración (Web):

Módulo que gestiona en tiempo real toda la información referente a la red comercial de la empresa. Realiza un mantenimiento de tiendas o franquicias, del personal, stock, reservas de prendas, vales, pagas y señales, familias, artículos, tallas, colores, arreglos,... Este módulo será de acceso restringido y en él, también se obtendrán informes, estadísticas, gráficas,...

Los Sub-Módulos a realizar serán:

1. Validación
2. Panel de control
3. Mantenimiento de Familias (familias)
4. Mantenimiento de Modelos (modelos)
5. Mantenimiento de Artículos (artículos)
6. Envío de Stock a franquicias (desde el mantenimiento de artículos si le das al check de agregar al stock de una tienda, sale el formulario del mantenimiento de stock de tal manera que se agrega al stock de la tienda y con la cantidad decidida)
7. Mantenimiento de franquicias/tiendas (tiendas)
8. Mantenimiento del personal (vendedores/as) (usuarios)
9. Mantenimiento del Stock en tiendas y almacén (hay un mantenimiento para el mantenimiento)
10. Mantenimiento de Arreglos de modista (individual por tienda/general por empresa) (arreglosmodista)
11. Traspasos de stock entre tiendas (entre las tiendas hay la posibilidad de distribuir artículos a otras tiendas, no se controla a la tienda que se distribuye pero si la cantidad de artículos distribuidos por la tienda en concreto)

12. Informes:

- i. Artículos por familias
- ii. Stock por tienda o franquicia

2. Módulo TpvVirtual (Web):

Módulo que gestiona en tiempo real toda la información referente a un punto de venta de la red comercial de la empresa.

Las sub-módulos a realizar serán:

1. Ventas (Tpv Clásico)
2. Arreglos de modista
3. Reservas (locales y entre franquicias)
4. Devoluciones
5. Vales/Abonos
6. Control de Stock
7. Gestión de Caja
 - i. Cajón
 - ii. Ticket
 - iii. Caja inicial
 - iv. Cierre diario
 - v. Cierre semanal (Confirmar)
 - vi. Cierre mensual (Confirmar)
8. Informes y estadísticas (Confirmar)
9. Opciones generales de la franquicia (depende de Opciones generales de la empresa) (Confirmar)
 - i. Ticket
 - ii. Caducidad de devoluciones
 - iii. Caducidad de pagas y señales
 - iv. Caducidad de reservas
 - v. Gestión de vales y abonos

Todo esto es lo que hará la aplicación todas estas funcionalidades se podrán gestionar según el cliente dando privilegios a los usuarios según su perfil.

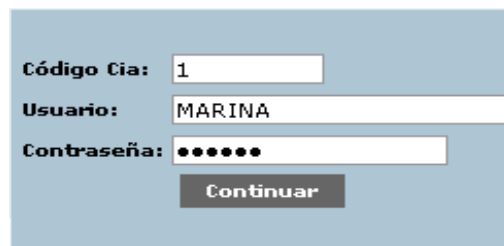
3.1.1.-Descripción del módulo de administración

Cuando el cliente compra esta aplicación entrará por primera vez con un usuario que está siempre dado de alta por defecto y con perfil de usuario administrador. El cliente una vez se introduzca como usuario podrá modificar o borrar este usuario por defecto.

El código de compañía dependerá de si el cliente compra la aplicación y se la instala dentro de su propia infraestructura o si decide que sea la empresa BabSoft la que se haga responsable de esto. Si se la instala dentro de su infraestructura el código de compañía será por defecto 1.

El Usuario será: MARINA

Y la contraseña: marina



The image shows a login form with a light blue background. It contains three input fields: 'Código Cia' with the value '1', 'Usuario' with the value 'MARINA', and 'Contraseña' with seven dots. Below the fields is a dark grey button labeled 'Continuar'.

3.1.2.- Acceso a la aplicación

A la aplicación se accede a través del website de la página, antes de poder ejecutarla el usuario tendrá que estar validado con el perfil correspondiente, y de esta manera podrá acceder a las partes diferenciadas de la aplicación, que puede acceder con sus privilegios.

3.2.- Descripción del entorno de ejecución

El entorno de funcionamiento será internet. De esta manera el cliente podrá trabajar con la aplicación desde cualquier sitio conectado a internet.

La plataforma que se utilizará para llevar a cabo la aplicación será:

- IIS funcionando sobre Windows 2000 Server con Service Pack 2.
- El servidor de la base de datos de MySql Server.
- PHP 5.0

Esta plataforma será instalada como servidor y será desde donde se ejecutará nuestra aplicación.

En este mismo se instalarán las bases de datos, (se hará de la misma manera tanto si el cliente se gestiona su propia aplicación o si se hace desde la empresa babSoft). Cualquier ordenador que se encuentre dentro de la red podrá acceder al servidor y acceder a la aplicación conectándose a este.

3.3.- Arquitectura de la aplicación

La arquitectura de la aplicación se divide en tres capas bien definidas en el website de la aplicación:

1. Capa de interfície con el usuario: Esta capa es la que hace referencia a la presentación de la información. En nuestro caso la capa de interfície es el diseño de las páginas web.
2. Capa de negocio: Esta capa hace referencia al todo el tratamiento de los datos. En nuestro caso esta capa hace referencia a los servlets, ya que se trata de una aplicación web dinámica que se ejecuta en el servidor.
3. Capa de datos: Esta capa hace referencia a los datos del sistema, a la base de datos del sistema, en nuestro caso se implementa con MySqlServer.

3.4.- Casos de uso:

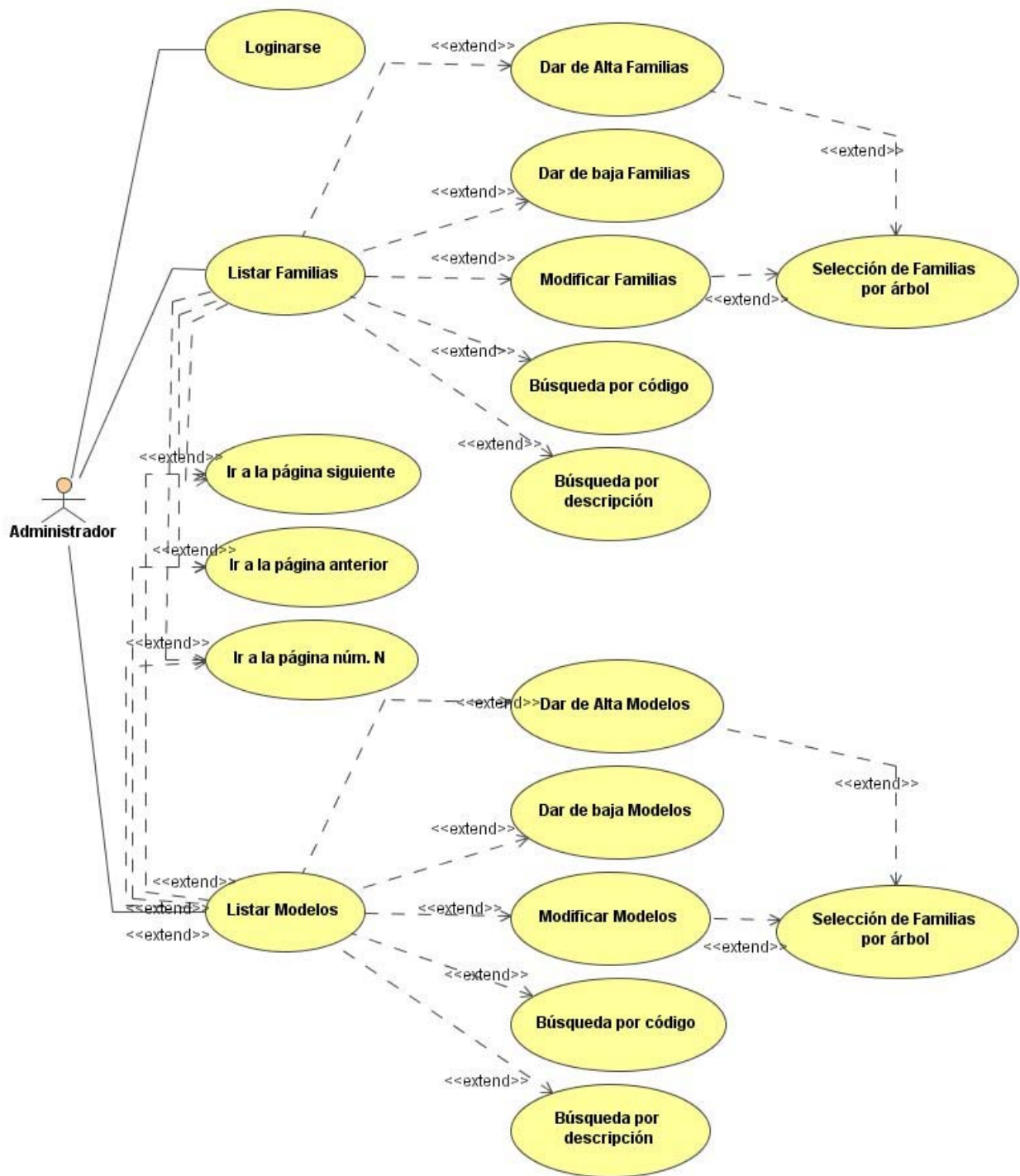
En este apartado definiré cuales son las acciones que se realizan, a muy alto nivel, en la aplicación. La documentación textual de los requisitos de los casos de uso se hace según un esquema común que comprende, una estructura estándar. Los casos de uso que se muestran a continuación son los resultantes de la aplicación para el perfil de administrador.

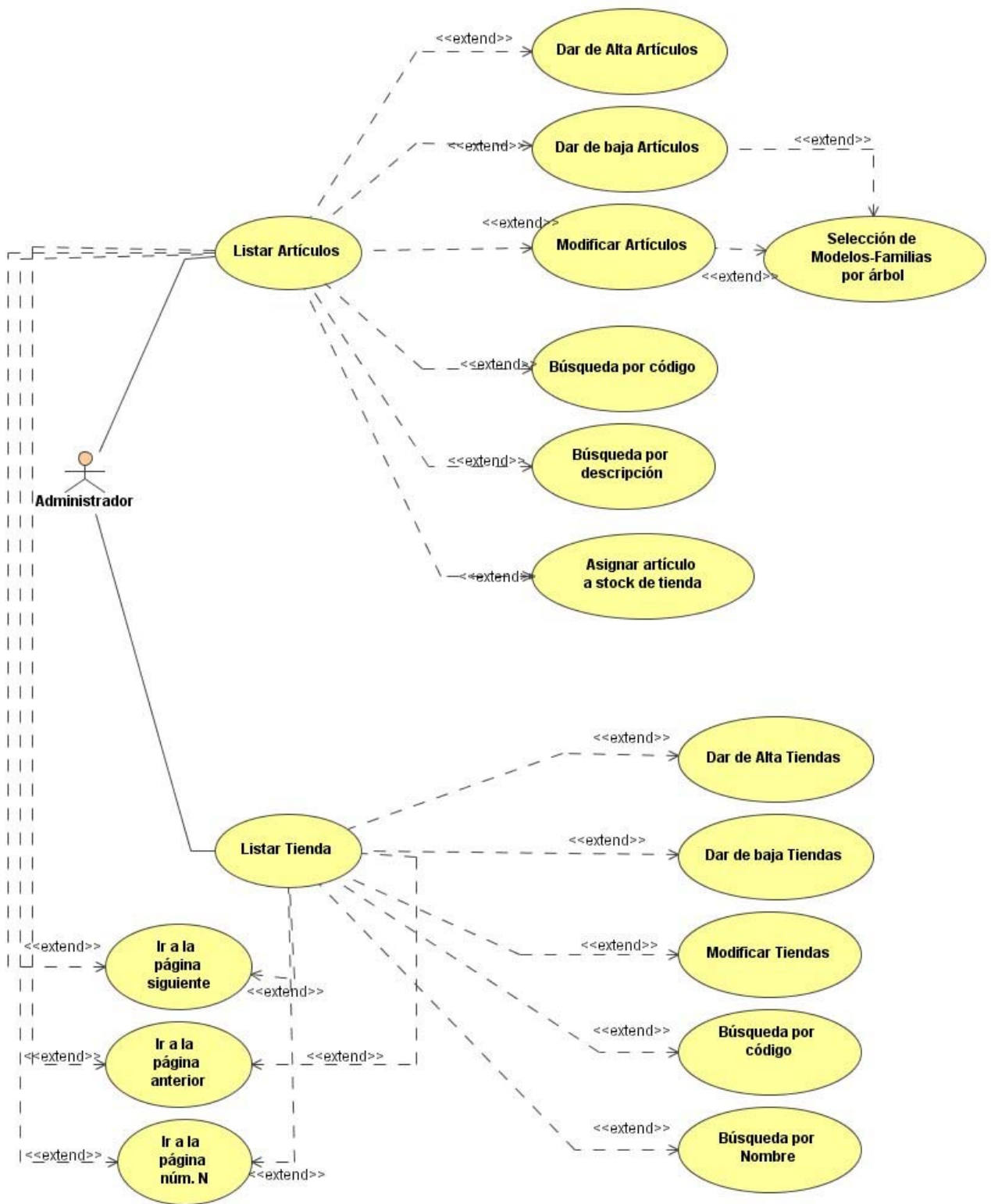
3.4.1.- Diagrama de casos

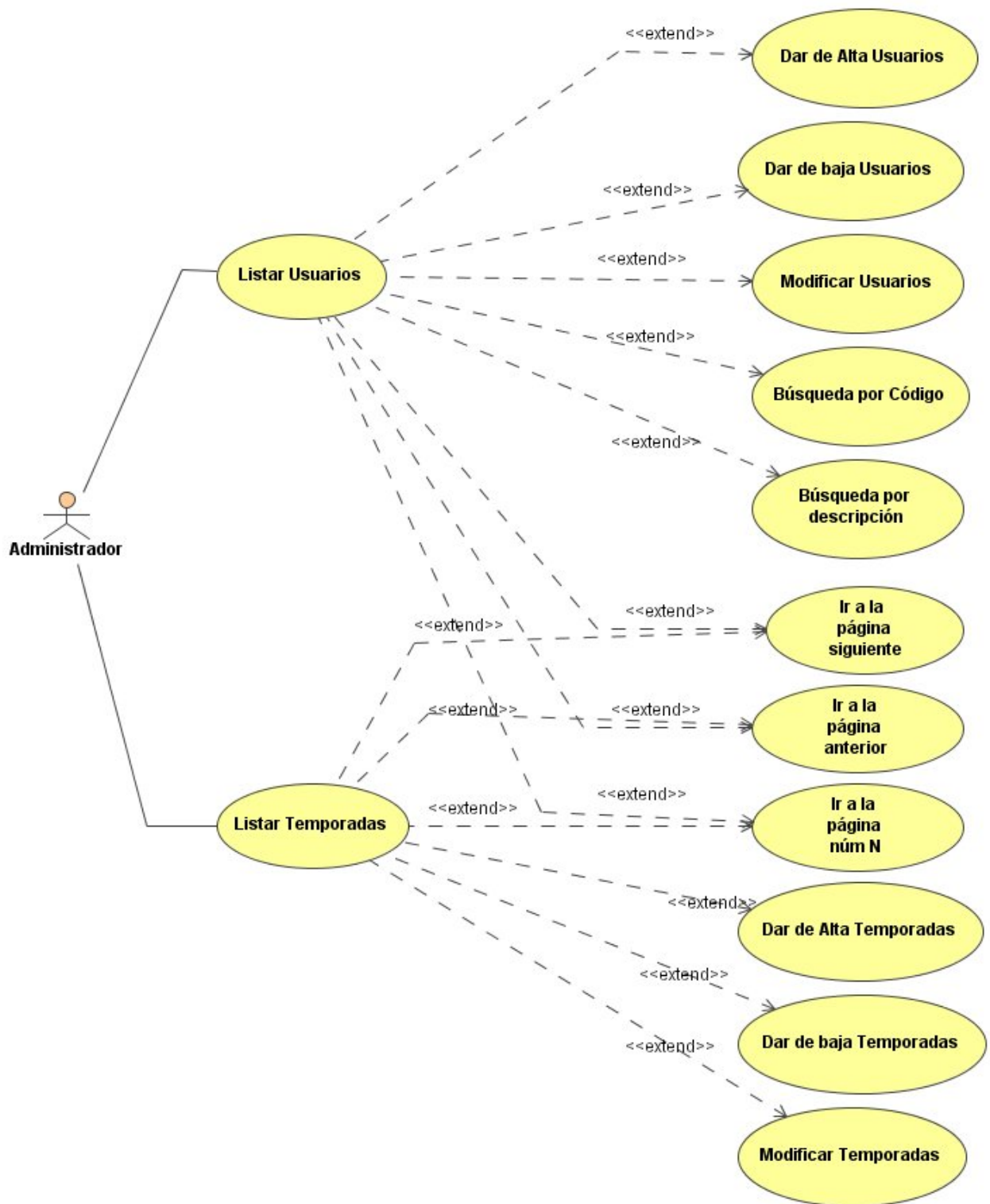
Diagrama de los casos de uso para el usuario con perfil administrador y con perfil usuario.

3.4.1.1.- Diagrama de casos de uso de perfil Administrador

A continuación muestro los diferentes diagramas de caso de uso para el perfil de administrador.









3.4.1.2.-Diagrama de casos de uso de perfil usuario

En el caso de un perfil diferente, como el de tipo usuario, cambiarían algunas de las acciones que permite hacer la aplicación. Todo esto se configura por parte del cliente que compra la aplicación, pero una posible configuración de caso de uso para ese perfil se mostrará a continuación del de administrador.



3.4.2.- Especificación de casos de uso

3.4.2.1.- Validarse

Caso de uso: Validarse

Resumen de la funcionalidad: El cliente desea entrar a la aplicación.

Actores: Administrador (iniciador)

Precondición: -

Postcondición: El usuario si sus datos son correctos se valida y puede entrar al sistema.

Descripción detallada: El usuario debe introducir el código de la compañía, el código de usuario y la contraseña.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario teclea sus datos de usuario para poder validarse y entrar en el sistema, y clica al botón de aceptar	
	2. El sistema valida que los datos introducidos son los correctos y permite entrar al usuario al sistema.

Cursos alternativos:

Línea 2: El usuario entrará a la aplicación y podrá realizar las funciones de ésta dependiendo del perfil que el usuario tenga dentro de la aplicación.

3.4.2.2.- Listar familias

Caso de uso: Listar Familias

Resumen de la funcionalidad: El cliente desea listar las familias.

Actores: Administrador (iniciador)

Precondición: Tienen que existir datos en la base de datos en la tabla familias

Postcondición: Ninguna

Descripción detallada: El usuario selecciona el botón de familias para poder listarlas. Una vez entrado en el listado de familias podrá verlas todas ordenadas por código familia y paginadas de 10 en 10, con los botones de página anterior y siguiente podrá moverse a través de las diferentes páginas.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de familias	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de familias para poder listar las páginas de datos agrupadas de 10 en 10

3.4.2.3.- Dar de Alta Familias

Caso de uso: Dar de alta familias

Resumen de la funcionalidad: El cliente desea dar de alta familias

Actores: Administrador (iniciador)

Precondición: -

Postcondición: Se ha dado de alta una nueva familia.

Descripción detallada: El usuario una vez que se encuentra en el listado de familias, podrá dar de alta la familia. Para ello deberá dar al botón de dar de alta y a continuación se mostrará un formulario que deberá rellenar con los datos adecuados.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de familias	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de familias para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario hace clic al botón de dar de alta.	
	4. Aparece un formulario con los campos de la tabla familia
5. El usuario rellena el mantenimiento con los datos adecuados.	
	6. El sistema valida que los datos introducidos por el usuario son correctos.

Cursos alternativos:

Línea 6: Si el usuario introduce algún dato incorrecto o algún código repetido el sistema mostrará un mensaje de error.

3.4.2.4.- Dar de Baja Familias

Caso de uso: Dar de baja familias

Resumen de la funcionalidad: El cliente desea dar de baja familias

Actores: Administrador (iniciador)

Precondición: La familia que desea dar de baja debe estar listada.

Postcondición: Se habrán borrado todas las familias seleccionadas y los registros que dependan de ésta.

Descripción detallada: El usuario una vez que se encuentra en el listado de familias, podrá dar de baja una familia. Para ello se puede hacer de dos maneras, borrando el registro o anulándolo de tal manera que continua existiendo pero ya no se considera. Para anularlo hay que dar al botón de modificar el registro y dar al check de anulado.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de familias	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de familias para poder listar las páginas de datos agrupadas de 10 en 10
3.a. El usuario marca todos los registros que quiere eliminar.	
4. El usuario da al botón de eliminar	
	5. El sistema borra de la base de datos todos los registros y con estos todos los datos que deba para continuar manteniendo la integridad de los datos.
3.b El usuario da al check de modificar el registro	
	6. El sistema muestra un formulario con los datos de ese registro.
7. El usuario marca el chek de anulado	

	8. El sistema guarda los datos modificados
--	--

Cursos alternativos:

Línea 5: Si el usuario borra los registros de la familia, el sistema generará una actualización en cadena borrando todos aquellos modelos y artículos que dependan de esta familia, para guardar la integridad y la coherencia de los datos.

3.4.2.5.- Modificar familias

Caso de uso: Modificar familias

Resumen de la funcionalidad: El cliente desea modificar familias

Actores: Administrador (iniciador)

Precondición: la familia a modificar debe estar listada

Postcondición: Se habrán modificado otros registros que dependan de este.

Descripción detallada: El usuario una vez que se encuentra en el listado de familias, podrá modificar cualquiera de estas familias, para ello deberá apretar el botón de modificar. Lo único que no podrá modificar será el código de familia, que aparecerá deshabilitado.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clicca al botón del content frame de familias	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de familias para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario pincha sobre el botón de modificar.	
	4. El sistema modifica el registro seleccionado con los datos modificados.

Cursos alternativos:

Línea 4: Si el usuario al modificar los datos, introduce algún dato que no sea correcto para el sistema, este mostrará un mensaje de error para que el usuario pueda corregirlo.

3.4.2.6.- Buscar por código familias

Caso de uso: Búsqueda por código familias

Resumen de la funcionalidad: El cliente desea realizar una búsqueda de código familias

Actores: Administrador (iniciador)

Precondición: se debe encontrar en el listado de familias

Postcondición: Ninguna.

Descripción detallada: El usuario una vez que se encuentra en el listado de familias, podrá realizar la búsqueda por código de familia. Para ello deberá escribir la cadena por la que quiere realizar la búsqueda y pulsar el botón de búsqueda.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de familias	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de familias para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario introduce la cadena de caracteres sobre la que quiere realizar la búsqueda.	
4. El usuario deberá apretar el botón de búsqueda.	
	5. El sistema devuelve los datos encontrados en la búsqueda agrupados de 10 en 10

Cursos alternativos:

Línea 5: El sistema devuelve todos los códigos de familia que se encuentre que contengan todos los caracteres que el usuario ha introducido para la búsqueda. Tanto si los encuentra al principio en medio o al final del código de familia dado de alta en la base de datos.

3.4.2.7.- Buscar por descripción de familias

Caso de uso: Búsqueda por descripción de familias

Resumen de la funcionalidad: El cliente desea realizar una búsqueda de descripción de familias

Actores: Administrador (iniciador)

Precondición: se debe encontrar en el listado de familias

Postcondición: Ninguna.

Descripción detallada: El usuario una vez que se encuentra en el listado de familias, podrá realizar la búsqueda por descripción de familias. Para ello deberá escribir la cadena por la que quiere realizar la búsqueda y pulsar el botón de búsqueda.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de familias	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de familias para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario introduce la cadena de caracteres sobre la que quiere realizar la búsqueda.	
4. El usuario deberá apretar el botón de búsqueda.	
	5. El sistema devuelve los datos encontrados en la búsqueda agrupados de 10 en 10

Cursos alternativos:

Línea 5: El sistema devuelve todas las descripciones de familias que se encuentre que contengan todos los caracteres que el usuario ha introducido para la búsqueda. Tanto si los encuentra al principio en medio o al final de la descripción de la familia dada de alta en la base de datos.

3.4.2.8.- Selección de familias por árbol

Caso de uso: selección de familias por árbol.

Resumen de la funcionalidad: El usuario clic al botón que nos mostrará un árbol para poder agrupar las familias.

Actores: Administrador (iniciador)

Precondición:

Postcondición:

Descripción detallada: El usuario selecciona una familia a la que quiere que pertenezca el código de familia actual.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1.a. El usuario dentro del formulario de dar de alta familias selecciona el botón de familia padre.	
	2. El sistema muestra un árbol con las jerarquías actuales de las familias.
1.b. El usuario dentro del formulario de dar de modificación familias selecciona el botón de familia padre.	
	3. El sistema muestra un árbol con las jerarquías actuales de las familias.

Cursos alternativos:

Línea 2,3: Si no existen familias el árbol aparecerá vacío. Se puede seleccionar cualquiera de los niveles para que forme parte de la familia padre.

3.4.2.9.- Listar modelos

Caso de uso: Listar modelos

Resumen de la funcionalidad: El cliente desea listar las modelos.

Actores: Administrador (iniciador)

Precondición: Tienen que existir datos en la base de datos en la tabla modelos

Postcondición: Ninguna

Descripción detallada: El usuario clicla el botón de modelos del content frame para poder listarlos. Una vez entrado en el listado de modelos podrá verlos todos ordenados por código modelos y paginadas de 10 en 10, con los botones de página anterior y siguiente podrá moverse a través de las diferentes páginas.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clicla al botón del content frame de modelos	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de modelos para poder listar las páginas de datos agrupadas de 10 en 10

3.4.2.10.- Dar de Alta Modelos

Caso de uso: Dar de alta modelos

Resumen de la funcionalidad: El cliente desea dar de alta modelos

Actores: Administrador (iniciador)

Precondición: -

Postcondición: Se ha dado de alta un nuevo modelo.

Descripción detallada: El usuario una vez que se encuentra en el listado de modelos, podrá dar de alta un modelo. Para ello deberá dar al botón de dar de alta y a continuación se mostrará un formulario que deberá rellenar con los datos adecuados.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de modelos	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de modelos para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario hace clic en el botón de dar de alta.	
	4. Aparece un formulario con los campos de la tabla modelos
5. El usuario rellena el mantenimiento con los datos adecuados.	
	6. El sistema valida que los datos introducidos por el usuario son correctos.

Cursos alternativos:

Línea 6: Si el usuario introduce algún dato incorrecto o algún código repetido el sistema mostrará un mensaje de error.

3.4.2.11.- Dar de Baja Modelos

Caso de uso: Dar de baja modelos

Resumen de la funcionalidad: El cliente desea dar de baja modelos

Actores: Administrador (iniciador)

Precondición: El modelo que desea dar de baja debe estar listado.

Postcondición: Se habrán borrado todos los modelos seleccionados y los registros que dependan de éste.

Descripción detallada: El usuario una vez que se encuentra en el listado de modelos, podrá dar de baja un modelo. Para ello se puede hacer de dos maneras, borrando el registro o anulándolo de tal manera que continua existiendo pero ya no se considera. Para anularlo hay que dar al botón de modificar el registro y dar al check de anulado.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clicó al botón del content frame de modelos	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de modelos para poder listar las páginas de datos agrupadas de 10 en 10
3.a El usuario marca todos los registros que quiere eliminar	
4. El usuario da al botón de eliminar	
	5. El sistema borra de la base de datos todos los registros y con estos todos los datos que deba para continuar manteniendo la integridad de los datos.
3.b El usuario da al check de modificar el registro	
	6. El sistema muestra un formulario con los datos de ese registro.
7. El usuario marca el check de anulado	

	8. El sistema guarda los datos modificados
--	--

Cursos alternativos:

Línea 5: Si el usuario borra los registros de modelo, el sistema generará una actualización en cadena borrando todos aquellos artículos que dependan de este modelo, para guardar la integridad y la coherencia de los datos.

3.4.2.12.- Modificar Modelos

Caso de uso: Modificar modelos

Resumen de la funcionalidad: El cliente desea modificar modelos

Actores: Administrador (iniciador)

Precondición: el modelo a modificar debe estar listada

Postcondición: Se habrán modificado otros registros que dependan de este.

Descripción detallada: El usuario una vez que se encuentra en el listado de modelos, podrá modificar cualquiera de estos modelos, para ello deberá apretar el botón de modificar. Lo único que no podrá modificar será el código de modelo, que aparecerá deshabilitado.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clicla al botón del content frame de modelos	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de modelos para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario pincha sobre el botón de modificar.	
	4. El sistema modifica el registro seleccionado con los datos modificados.

Cursos alternativos:

Línea 4: Si el usuario al modificar los datos, introduce algún dato que no sea correcto para el sistema, este mostrará un mensaje de error para que el usuario pueda corregirlo.

3.4.2.13.- Buscar por código modelos

Caso de uso: Búsqueda por código modelos

Resumen de la funcionalidad: El cliente desea realizar una búsqueda de código modelos

Actores: Administrador (iniciador)

Precondición: se debe encontrar en el listado de modelos

Postcondición: Ninguna.

Descripción detallada: El usuario una vez que se encuentra en el listado de modelos, podrá realizar la búsqueda por código de modelo. Para ello deberá escribir la cadena por la que quiere realizar la búsqueda y pulsar el botón de búsqueda.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de modelos	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de modelos para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario introduce la cadena de caracteres sobre la que quiere realizar la búsqueda.	
4. El usuario deberá apretar el botón de búsqueda.	
	5. El sistema devuelve los datos encontrados en la búsqueda agrupados de 10 en 10

Cursos alternativos:

Línea 5: El sistema devuelve todos los códigos de modelo que se encuentre que contengan todos los caracteres que el usuario ha introducido para la búsqueda. Tanto si los encuentra al principio, en medio o al final del código de modelo dado de alta en la base de datos.

3.4.2.14.- Buscar por descripción de modelos

Caso de uso: Búsqueda por descripción de modelos

Resumen de la funcionalidad: El cliente desea realizar una búsqueda de descripción de modelos.

Actores: Administrador (iniciador)

Precondición: se debe encontrar en el listado de modelos

Postcondición: Ninguna.

Descripción detallada: El usuario una vez que se encuentra en el listado de modelos, podrá realizar la búsqueda por descripción de modelos. Para ello deberá escribir la cadena por la que quiere realizar la búsqueda y pulsar el botón de búsqueda.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de modelos	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de modelos para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario introduce la cadena de caracteres sobre la que quiere realizar la búsqueda.	
4. El usuario deberá apretar el botón de búsqueda.	
	5. El sistema devuelve los datos encontrados en la búsqueda agrupados de 10 en 10

Cursos alternativos:

Línea 5: El sistema devuelve todas las descripciones de modelos que se encuentre que contengan todos los caracteres que el usuario ha introducido para la búsqueda. Tanto si los encuentra al principio, en medio o al final de la descripción del modelo que esta dada de alta en la base de datos.

3.4.2.15.- Selección de familias por árbol de modelos

Caso de uso: selección de familias por árbol.

Resumen de la funcionalidad: El usuario clicla al botón que nos mostrará un árbol para seleccionar la familia a la que pertenecerá el modelo.

Actores: Administrador (iniciador)

Precondición:

Postcondición:

Descripción detallada: El usuario selecciona una familia a la que quiere que pertenezca el código de familia actual.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1.a. El usuario dentro del formulario de dar de alta modelos selecciona el botón de familia.	
	2. El sistema muestra un árbol con las jerarquías actuales de las familias.
1.b. El usuario dentro del formulario de dar de modificación familias selecciona el botón de familia.	
	3. El sistema muestra un árbol con las jerarquías actuales de las familias.

Cursos alternativos:

Línea 2,3: Si no existen familias el árbol aparecerá vacío. Sólo se puede seleccionar una de las familias finales.

3.4.2.16.- Listar artículos

Caso de uso: Listar Artículos

Resumen de la funcionalidad: El cliente desea listar los artículos.

Actores: Administrador (iniciador)

Precondición: Tienen que existir datos en la base de datos en la tabla artículos

Postcondición: Ninguna

Descripción detallada: El usuario selecciona el botón de artículos para poder listarlos. Una vez entrado en el listado de artículos podrá verlos todos ordenados por código artículo y paginadas de 10 en 10, con los botones de página anterior y siguiente podrá moverse a través de las diferentes páginas.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de artículos	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de artículos para poder listar las páginas de datos agrupadas de 10 en 10

3.4.2.17.- Dar de Alta Artículos

Caso de uso: Dar de alta artículos

Resumen de la funcionalidad: El cliente desea dar de alta artículos

Actores: Administrador (iniciador)

Precondición: -

Postcondición: Se ha dado de alta un nuevo artículo.

Descripción detallada: El usuario una vez que se encuentra en el listado de artículos, podrá dar de alta el artículo. Para ello deberá dar al botón de dar de alta y a continuación se mostrará un formulario que deberá rellenar con los datos adecuados.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de artículos	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de artículo para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario hace clic en el botón de dar de alta.	
	4. Aparece un formulario con los campos de la tabla artículo
5. El usuario rellena el mantenimiento con los datos adecuados.	
	6. El sistema valida que los datos introducidos por el usuario son correctos.

Cursos alternativos:

Línea 6: Si el usuario introduce algún dato incorrecto o algún código repetido el sistema mostrará un mensaje de error.

3.4.2.18.- Dar de Baja Artículos

Caso de uso: Dar de alta artículos

Resumen de la funcionalidad: El cliente desea dar de baja artículos

Actores: Administrador (iniciador)

Precondición: El artículo que desea dar de baja debe estar listado.

Postcondición: Se habrán borrado todos los artículos seleccionados y los registros que dependan de éste.

Descripción detallada: El usuario una vez que se encuentra en el listado de artículos, podrá dar de baja un artículo. Para ello se puede hacer de dos maneras, borrando el registro o anulándolo de tal manera que continua existiendo pero ya no se considera. Para anularlo hay que dar al botón de modificar el registro y dar al check de anulado.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clicó al botón del content frame de artículos	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de artículo para poder listar las páginas de datos agrupadas de 10 en 10
3.a El usuario marca todos los registros que quiere eliminar	
4. El usuario da al botón de eliminar	
	5. El sistema borra de la base de datos todos los registros y con estos todos los datos que deba para continuar manteniendo la integridad de los datos.
3.b El usuario da al check de modificar el registro	
	6. El sistema muestra un formulario con los datos de ese registro.
7. El usuario marca el check de anulado	

	8. El sistema guarda los datos modificados
--	--

Cursos alternativos:

Línea 5: Si el usuario borra los registros de artículo, el sistema generará una actualización en cadena borrando todas aquellas referencias a stock de tiendas que dependan de este artículo, para guardar la integridad y la coherencia de los datos.

3.4.2.19.- Modificar Artículos

Caso de uso: Modificar artículos

Resumen de la funcionalidad: El cliente desea modificar artículos.

Actores: Administrador (iniciador)

Precondición: El artículo a modificar debe estar listado.

Postcondición: Se habrá modificado los datos del artículo.

Descripción detallada: El usuario una vez que se encuentra en el listado de artículos, podrá modificar cualquiera de estos artículos, para ello deberá apretar el botón de modificar. Lo único que no podrá modificar será el código de artículo, que aparecerá deshabilitado.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de artículos	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de artículo para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario pincha sobre el botón de modificar.	
	4. El sistema modifica el registro seleccionado con los datos modificados.

Cursos alternativos:

Línea 4: Si el usuario al modificar los datos, introduce algún dato que no sea correcto para el sistema, este mostrará un mensaje de error para que el usuario pueda corregirlo. Además si se modifica agregando stock a tienda se modificarán las relaciones de la tabla de artículo-tienda, donde se guarda el stock de estas.

3.4.2.20.- Buscar por código artículo

Caso de uso: Búsqueda por código artículo

Resumen de la funcionalidad: El cliente desea realizar una búsqueda de código artículo

Actores: Administrador (iniciador)

Precondición: se debe encontrar en el listado de artículos

Postcondición: Ninguna.

Descripción detallada: El usuario una vez que se encuentra en el listado de artículos, podrá realizar la búsqueda por código de artículo. Para ello deberá escribir la cadena por la que quiere realizar la búsqueda y pulsar el botón de búsqueda.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de artículo	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de artículos para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario introduce la cadena de caracteres sobre la que quiere realizar la búsqueda.	
4. El usuario deberá apretar el botón de búsqueda.	
	5. El sistema devuelve los datos encontrados en la búsqueda agrupados de 10 en 10

Cursos alternativos:

Línea 5: El sistema devuelve todos los códigos de artículo que se encuentre que contengan todos los caracteres que el usuario ha introducido para la búsqueda. Tanto si los encuentra al principio, en medio o al final del código de artículo dado de alta en la base de datos.

3.4.2.21.- Buscar por descripción de artículos

Caso de uso: Búsqueda por descripción de artículos

Resumen de la funcionalidad: El cliente desea realizar una búsqueda de descripción de artículos.

Actores: Administrador (iniciador)

Precondición: se debe encontrar en el listado de artículos.

Postcondición: Ninguna.

Descripción detallada: El usuario una vez que se encuentra en el listado de artículos, podrá realizar la búsqueda por descripción de artículo. Para ello deberá escribir la cadena por la que quiere realizar la búsqueda y pulsar el botón de búsqueda.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de artículo	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de artículos para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario introduce la cadena de caracteres sobre la que quiere realizar la búsqueda.	
4. El usuario deberá apretar el botón de búsqueda.	
	5. El sistema devuelve los datos encontrados en la búsqueda agrupados de 10 en 10

Cursos alternativos:

Línea 5: El sistema devuelve todas las descripciones de artículos que se encuentre que contengan todos los caracteres que el usuario ha introducido para la búsqueda. Tanto si los encuentra al principio, en medio o al final de la descripción del artículo dada de alta en la base de datos.

3.4.2.22.- Selección de modelos-familias por árbol

Caso de uso: selección de modelos-familias por árbol.

Resumen de la funcionalidad: El usuario clica al botón que nos mostrará un árbol seleccionar el modelo del artículo.

Actores: Administrador (iniciador)

Precondición:

Postcondición:

Descripción detallada: El usuario selecciona uno de los modelos a la que quiere que pertenezca el código de su artículo, al mismo tiempo el campo de familia se rellenará con la familia a la que pertenece el modelo.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1.a. El usuario dentro del formulario de dar de alta artículos selecciona el botón de modelos.	
	2. El sistema muestra un árbol con las con los modelos y sus familias.
1.b. El usuario dentro del formulario de dar de modificación artículos selecciona el botón de modelos.	
	3. El sistema muestra un árbol con las con los modelos y sus familias.

Cursos alternativos:

Línea 2,3: Si no existen modelos el árbol aparecerá vacío. Se seleccionará uno de los modelos que viene relacionado con su familia.

3.4.2.23.- Listar Tiendas

Caso de uso: Listar Tiendas

Resumen de la funcionalidad: El cliente desea listar las tiendas.

Actores: Administrador (iniciador)

Precondición: Tienen que existir datos en la base de datos en la tabla tiendas

Postcondición: Ninguna

Descripción detallada: El usuario selecciona el botón de tiendas para poder listarlos. Una vez entrado en el listado de tiendas podrá verlos todos ordenados por código tienda y paginadas de 10 en 10, con los botones de página anterior y siguiente podrá moverse a través de las diferentes páginas.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de tiendas	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de tiendas para poder listar las páginas de datos agrupadas de 10 en 10

3.4.2.24.- Dar de Alta Tiendas

Caso de uso: Dar de alta tiendas

Resumen de la funcionalidad: El cliente desea dar de alta tiendas

Actores: Administrador (iniciador)

Precondición: -

Postcondición: Se ha dado de alta una nueva tienda.

Descripción detallada: El usuario una vez que se encuentra en el listado de tiendas, podrá dar de alta la tienda. Para ello deberá dar al botón de dar de alta y a continuación se mostrará un formulario que deberá rellenar con los datos adecuados.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clicca al botón del content frame de tiendas	
	2. El sistema pide los datos necesarios a la base de datos de la tabla tiendas para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario apreta el botón de dar de alta.	
	4. Aparece un formulario con los campos de la tabla tiendas
5. El usuario rellena el mantenimiento con los datos adecuados.	
	6. El sistema valida que los datos introducidos por el usuario son correctos.

Cursos alternativos:

Línea 6: Si el usuario introduce algún dato incorrecto o algún código repetido el sistema mostrará un mensaje de error.

3.4.2.25.- Dar de Baja Tiendas

Caso de uso: Dar de baja tiendas

Resumen de la funcionalidad: El cliente desea dar de baja tiendas.

Actores: Administrador (iniciador)

Precondición: La tienda que desea dar de baja debe estar listada.

Postcondición: Se habrán borrado todas las tiendas seleccionadas y los registros que dependan de ésta.

Descripción detallada: El usuario una vez que se encuentra en el listado de tiendas, podrá dar de baja una tienda. Para ello se puede hacer de dos maneras, borrando el registro o anulándolo de tal manera que continua existiendo pero ya no se considera. Para anularlo hay que dar al botón de modificar el registro y dar al check de anulado.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clicó al botón del content frame de tiendas	
	2. El sistema pide los datos necesarios a la base de datos de la tabla tiendas para poder listar las páginas de datos agrupadas de 10 en 10
3.a El usuario marca todos los registros que quiere eliminar	
4. El usuario da al botón de eliminar	
	5. El sistema borra de la base de datos todos los registros y con estos todos los datos que deba para continuar manteniendo la integridad de los datos.
3.b El usuario da al check de modificar el registro	
	6. El sistema muestra un formulario con los datos de ese registro.
7. El usuario marca el check de anulado	

	8. El sistema guarda los datos modificados
--	--

Cursos alternativos:

Línea 5: Si el usuario borra los registros de tienda, el sistema generará una actualización en cadena borrando todas aquellas referencias a stock de artículos que dependan de esta tienda, además de todas las relaciones de usuario-tienda y los arreglos de modista para esa tienda, para guardar la integridad y la coherencia de los datos.

3.4.2.26.- Modificar Tiendas

Caso de uso: Modificar tiendas

Resumen de la funcionalidad: El cliente desea modificar tiendas.

Actores: Administrador (iniciador)

Precondición: La tienda a modificar debe estar listada.

Postcondición: Se habrán modificado todos los registros que dependan de éste.

Descripción detallada: El usuario una vez que se encuentra en el listado de tiendas, podrá modificar cualquiera de estas tiendas, para ello deberá apretar el botón de modificar. Lo único que no podrá modificar será el código de tienda, que aparecerá deshabilitado.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de tiendas	
	2. El sistema pide los datos necesarios a la base de datos de la tabla tiendas para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario pincha sobre el botón de modificar.	
	4. El sistema modifica el registro seleccionado con los datos modificados.

Cursos alternativos:

Línea 4: Si el usuario al modificar los datos, introduce algún dato que no sea correcto para el sistema, este mostrará un mensaje de error para que el usuario pueda corregirlo.

3.4.2.27.- Buscar por código de tienda

Caso de uso: Búsqueda por código de tienda

Resumen de la funcionalidad: El cliente desea realizar una búsqueda por código de tienda

Actores: Administrador (iniciador)

Precondición: se debe encontrar en el listado de tiendas

Postcondición: Ninguna.

Descripción detallada: El usuario una vez que se encuentra en el listado de tiendas, podrá realizar la búsqueda por código de tienda. Para ello deberá escribir la cadena por la que quiere realizar la búsqueda y pulsar el botón de búsqueda.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clicla al botón del content frame de tienda	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de tiendas para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario introduce la cadena de caracteres sobre la que quiere realizar la búsqueda.	
4. El usuario deberá apretar el botón de búsqueda.	
	5. El sistema devuelve los datos encontrados en la búsqueda agrupados de 10 en 10

Cursos alternativos:

Línea 5: El sistema devuelve todos los códigos de tienda que se encuentre que contengan todos los caracteres que el usuario ha introducido para la búsqueda. Tanto si los encuentra al principio, en medio o al final del código de tienda dado de alta en la base de datos.

3.4.2.28.- Buscar por Nombre de tienda

Caso de uso: Búsqueda por nombre de tienda

Resumen de la funcionalidad: El cliente desea realizar una búsqueda de nombre de tienda.

Actores: Administrador (iniciador)

Precondición: se debe encontrar en el listado de tiendas.

Postcondición: Ninguna.

Descripción detallada: El usuario una vez que se encuentra en el listado de tiendas, podrá realizar la búsqueda por nombre de tienda. Para ello deberá escribir la cadena por la que quiere realizar la búsqueda y pulsar el botón de búsqueda.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de tiendas	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de tiendas para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario introduce la cadena de caracteres sobre la que quiere realizar la búsqueda.	
4. El usuario deberá apretar el botón de búsqueda.	
	5. El sistema devuelve los datos encontrados en la búsqueda agrupados de 10 en 10

Cursos alternativos:

Línea 5: El sistema devuelve todos los nombres de tiendas que se encuentre que contengan todos los caracteres que el usuario ha introducido para la búsqueda. Tanto si los encuentra al principio, en medio o al final del nombre de la tienda dada de alta en la base de datos.

3.4.2.29.- Buscar por población de tienda

Caso de uso: Búsqueda por población de tienda

Resumen de la funcionalidad: El cliente desea realizar una búsqueda por población de tienda.

Actores: Administrador (iniciador)

Precondición: se debe encontrar en el listado de tiendas.

Postcondición: Ninguna.

Descripción detallada: El usuario una vez que se encuentra en el listado de tiendas, podrá realizar la búsqueda por población de tienda. Para ello deberá escribir la cadena por la que quiere realizar la búsqueda y pulsar el botón de búsqueda.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de tiendas	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de tiendas para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario introduce la cadena de caracteres sobre la que quiere realizar la búsqueda.	
4. El usuario deberá apretar el botón de búsqueda.	
	5. El sistema devuelve los datos encontrados en la búsqueda agrupados de 10 en 10

Cursos alternativos:

Línea 5: El sistema devuelve todas las poblaciones de tiendas que se encuentre que contengan todos los caracteres que el usuario ha introducido para la búsqueda. Tanto si los encuentra al principio, en medio o al final de la población de la tienda dada de alta en la base de datos.

3.4.2.30.- Listar usuarios

Caso de uso: Listar Usuarios

Resumen de la funcionalidad: El cliente desea listar los usuarios.

Actores: Administrador (iniciador)

Precondición: Tienen que existir datos en la base de datos en la tabla usuarios

Postcondición: Ninguna

Descripción detallada: El usuario selecciona el botón de usuarios para poder listarlos. Una vez entrado en el listado de usuarios podrá verlos todos ordenados por código vendedor y paginadas de 10 en 10, con los botones de página anterior y siguiente podrá moverse a través de las diferentes páginas.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de usuarios	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de usuarios para poder listar las páginas de datos agrupadas de 10 en 10

3.4.2.31.- Dar de Alta Usuarios

Caso de uso: Dar de alta usuarios

Resumen de la funcionalidad: El cliente desea dar de alta usuarios

Actores: Administrador (iniciador)

Precondición: -

Postcondición: Se ha dado de alta un nuevo usuario.

Descripción detallada: El usuario una vez que se encuentra en el listado de usuarios, podrá dar de alta un usuario. Para ello deberá dar al botón de dar de alta y a continuación se mostrará un formulario que deberá rellenar con los datos adecuados.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de usuarios	
	2. El sistema pide los datos necesarios a la base de datos de la tabla usuarios para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario hace clic en el botón de dar de alta.	
	4. Aparece un formulario con los campos de la tabla usuarios
5. El usuario rellena el mantenimiento con los datos adecuados.	
	6. El sistema valida que los datos introducidos por el usuario son correctos.

Cursos alternativos:

Línea 6: Si el usuario introduce algún dato incorrecto o algún código repetido el sistema mostrará un mensaje de error.

3.4.2.32.- Dar de Baja Usuarios

Caso de uso: Dar de baja usuarios

Resumen de la funcionalidad: El cliente desea dar de baja usuarios

Actores: Administrador (iniciador)

Precondición: El usuario que desea dar de baja debe estar listado.

Postcondición: Se habrán borrado todos los usuarios seleccionados y los registros que dependan de éste.

Descripción detallada: El usuario una vez que se encuentra en el listado de usuarios, podrá dar de baja un usuario. Para ello se puede hacer de dos maneras, borrando el registro o anulándolo de tal manera que continua existiendo pero ya no se considera. Para anularlo hay que dar al botón de modificar el registro y dar al check de anulado.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de usuarios	
	2. El sistema pide los datos necesarios a la base de datos de la tabla usuarios para poder listar las páginas de datos agrupadas de 10 en 10
3.a El usuario marca todos los registros que quiere eliminar	
4. El usuario da al botón de eliminar	
	5. El sistema borra de la base de datos todos los registros y con estos todos los datos que deba para continuar manteniendo la integridad de los datos.
3.b El usuario da al check de modificar el registro	
	6. El sistema muestra un formulario con los datos de ese registro.
7. El usuario marca el chek de anulado	

	8. El sistema guarda los datos modificados
--	--

Cursos alternativos:

Línea 5: Si el usuario borra los registros de un usuario, el sistema generará una actualización en cadena borrando todas aquellas referencias a ese usuario además de las relaciones de usuario-tienda, para guardar la integridad y la coherencia de los datos.

3.4.2.33.- Modificar Usuarios

Caso de uso: Modificar usuarios

Resumen de la funcionalidad: El cliente desea modificar usuarios

Actores: Administrador (iniciador)

Precondición: El usuario que desea modificar debe estar listado.

Postcondición: Se habrán modificado todos los registros que dependan de éste.

Descripción detallada: El usuario una vez que se encuentra en el listado de usuarios, podrá modificar cualquiera de este usuario, para ello deberá apretar el botón de modificar. Lo único que no podrá modificar será el código de usuario, que aparecerá deshabilitado.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de usuarios	
	2. El sistema pide los datos necesarios a la base de datos de la tabla usuarios para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario pincha sobre el botón de modificar.	
	4. El sistema modifica el registro seleccionado con los datos modificados.

Cursos alternativos:

Línea 4: Si el usuario al modificar los datos, introduce algún dato que no sea correcto para el sistema, este mostrará un mensaje de error para que el usuario pueda corregirlo.

3.4.2.34.- Buscar por código de usuario

Caso de uso: Búsqueda por código de usuario

Resumen de la funcionalidad: El cliente desea realizar una búsqueda por código de usuario

Actores: Administrador (iniciador)

Precondición: se debe encontrar en el listado de usuarios

Postcondición: Ninguna.

Descripción detallada: El usuario una vez que se encuentra en el listado de usuarios, podrá realizar la búsqueda por código de usuario. Para ello deberá escribir la cadena por la que quiere realizar la búsqueda y pulsar el botón de búsqueda.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de usuarios	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de usuarios para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario introduce la cadena de caracteres sobre la que quiere realizar la búsqueda.	
4. El usuario deberá apretar el botón de búsqueda.	
	5. El sistema devuelve los datos encontrados en la búsqueda agrupados de 10 en 10

Cursos alternativos:

Línea 5: El sistema devuelve todos los códigos de usuario que se encuentre que contengan todos los caracteres que el usuario ha introducido para la búsqueda. Tanto si los encuentra al principio, en medio o al final del código de usuario dado de alta en la base de datos.

3.4.2.35.- Buscar por Nombre de usuario

Caso de uso: Búsqueda por nombre de usuario

Resumen de la funcionalidad: El cliente desea realizar una búsqueda de nombre de usuario.

Actores: Administrador (iniciador)

Precondición: se debe encontrar en el listado de usuarios.

Postcondición: Ninguna.

Descripción detallada: El usuario una vez que se encuentra en el listado de usuarios, podrá realizar la búsqueda por nombre de usuario. Para ello deberá escribir la cadena por la que quiere realizar la búsqueda y pulsar el botón de búsqueda.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de usuarios	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de usuarios para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario introduce la cadena de caracteres sobre la que quiere realizar la búsqueda.	
4. El usuario deberá apretar el botón de búsqueda.	
	5. El sistema devuelve los datos encontrados en la búsqueda agrupados de 10 en 10

Cursos alternativos:

Línea 5: El sistema devuelve todos los nombres de usuarios que se encuentre que contengan todos los caracteres que el usuario ha introducido para la búsqueda. Tanto si los encuentra al principio, en medio o al final del nombre del usuario dado de alta en la base de datos.

3.4.2.36.- Listar temporadas

Caso de uso: Listar Temporadas

Resumen de la funcionalidad: El cliente desea listar los temporadas.

Actores: Administrador (iniciador)

Precondición: Tienen que existir datos en la base de datos en la tabla dictemporadas

Postcondición: Ninguna

Descripción detallada: El usuario selecciona el botón de temporadas para poder listarlos. Una vez entrado en el listado de temporadas podrá verlos todos ordenados por descripción y paginadas de 10 en 10, con los botones de página anterior y siguiente podrá moverse a través de las diferentes páginas.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de temporadas	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de dictemporadas para poder listar las páginas de datos agrupadas de 10 en 10

3.4.2.37.- Dar de Alta Temporadas

Caso de uso: Dar de alta temporadas

Resumen de la funcionalidad: El cliente desea dar de alta temporadas

Actores: Administrador (iniciador)

Precondición: -

Postcondición: Se ha dado de alta una nueva temporada.

Descripción detallada: El usuario una vez que se encuentra en el listado de temporadas, podrá dar de alta una temporada. Para ello deberá dar al botón de dar de alta y a continuación se mostrará un formulario que deberá rellenar con la descripción adecuada.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de temporadas	
	2. El sistema pide los datos necesarios a la base de datos de la tabla dictemporada para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario hace clic en el botón de dar de alta.	
	4. Aparece un formulario con la descripción
5. El usuario rellena el mantenimiento.	
	6. El sistema valida que la descripción introducida es la correcta

Cursos alternativos:

Línea 6: Si el usuario introduce algún dato incorrecto o alguna descripción repetida el sistema mostrará un mensaje de error.

3.4.2.38.- Dar de Baja Temporadas

Caso de uso: Dar de baja temporadas

Resumen de la funcionalidad: El cliente desea dar de baja temporadas

Actores: Administrador (iniciador)

Precondición: La temporada que desea dar de baja debe estar listada.

Postcondición: Se habrán borrado todas las temporadas.

Descripción detallada: El usuario una vez que se encuentra en el listado de temporadas, podrá dar de baja una temporada. Para ello se puede hacer de dos maneras, borrando el registro o anulándolo de tal manera que continua existiendo pero ya no se considera. Para anularlo hay que dar al botón de modificar el registro y dar al check de anulado.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clic al botón del content frame de temporadas	
	2. El sistema pide los datos necesarios a la base de datos de la tabla dictemporada para poder listar las páginas de datos agrupadas de 10 en 10
3.a El usuario marca todos los registros que quiere eliminar	
4. El usuario da al botón de eliminar	
	5. El sistema borra de la base de datos todos los registros y con estos todos los datos que deba para continuar manteniendo la integridad de los datos.
3.b El usuario da al check de modificar el registro	
	6. El sistema muestra un formulario con los datos de ese registro.
7. El usuario marca el chek de anulado	

	8. El sistema guarda los datos modificados
--	--

Cursos alternativos:

Línea 5: Si el usuario borra los registros de una temporada, el sistema generará una actualización en cadena borrando todas aquellas referencias a esa temporada, además de todos los artículos que dependan de esta temporada, para guardar la integridad y la coherencia de los datos.

3.4.2.39.- Modificar Temporadas

Caso de uso: Modificar temporadas

Resumen de la funcionalidad: El cliente desea modificar temporadas

Actores: Administrador (iniciador)

Precondición: La temporada que desea modificar debe estar listada.

Postcondición: Se habrán modificado todos los registros que dependan de éste.

Descripción detallada: El usuario una vez que se encuentra en el listado de temporadas, podrá modificar cualquiera de estas temporadas, para ello deberá apretar el botón de modificar.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de temporadas	
	2. El sistema pide los datos necesarios a la base de datos de la tabla dictemporada para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario pincha sobre el botón de modificar.	
	4. El sistema modifica el registro seleccionado con los datos modificados.

Cursos alternativos:

Línea 4: Si el usuario al modificar los datos, introduce algún dato que no sea correcto para el sistema, este mostrará un mensaje de error para que el usuario pueda corregirlo. Además si se modifica las franquicias a las que pertenece el usuario, se modificará la tabla que relaciona usuarios-tiendas.

3.4.2.40.- Listar Arreglos de modista

Caso de uso: Listar Arreglos de modista

Resumen de la funcionalidad: El cliente desea listar los arreglos de modista.

Actores: Administrador (iniciador)

Precondición: Tienen que existir datos en la base de datos en la tabla arreglosmodista

Postcondición: Ninguna

Descripción detallada: El usuario selecciona el botón de arreglos modista para poder listarlos. Una vez entrado en el listado de arreglos modista podrá verlos todos ordenados por descripción y paginados de 10 en 10, con los botones de página anterior y siguiente podrá moverse a través de las diferentes páginas.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de arreglos de modista	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de arreglosmodista para poder listar las páginas de datos agrupadas de 10 en 10

3.4.2.41.- Dar de Alta Arreglos de modista

Caso de uso: Dar de alta arreglos de modista

Resumen de la funcionalidad: El cliente desea dar de alta arreglos de modista

Actores: Administrador (iniciador)

Precondición: -

Postcondición: Se ha dado de alta uno arreglo de modista.

Descripción detallada: El usuario una vez que se encuentra en el listado de arreglos de modista, podrá dar de alta un nuevo arreglo de modista. Para ello deberá dar al botón de dar de alta y a continuación se mostrará el formulario que deberá rellenar con los datos adecuados.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de arreglos de modista	
	2. El sistema pide los datos necesarios a la base de datos de la tabla arreglosmodista para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario hace clic en el botón de dar de alta.	
	4. Aparece el formulario de los datos a dar de alta en la tabla de arreglosmodista
5. El usuario rellena el mantenimiento con los campos necesarios.	
	6. El sistema valida que los campos introducidos son los correctos

Cursos alternativos:

Línea 6: Si el usuario introduce algún dato incorrecto o alguna descripción repetido el sistema mostrará un mensaje de error.

3.4.2.42.- Dar de Baja Arreglos de modista

Caso de uso: Dar de baja arreglos de modista

Resumen de la funcionalidad: El cliente desea dar de baja arreglos de modista

Actores: Administrador (iniciador)

Precondición: El Arreglo de modista que desea dar de baja debe estar listado.

Postcondición: Se habrán borrado todos los arreglos de modista.

Descripción detallada: El usuario una vez que se encuentra en el listado de arreglos de modista, podrá dar de baja el arreglo de modista. Para ello se puede hacer de dos maneras, borrando el registro o anulándolo de tal manera que continua existiendo pero ya no se considera. Para anularlo hay que dar al botón de modificar el registro y dar al check de anulado.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de arreglos de modista	
	2. El sistema pide los datos necesarios a la base de datos de la tabla arreglosmodista para poder listar las páginas de datos agrupadas de 10 en 10
3.a El usuario marca todos los registros que quiere eliminar	
4. El usuario da al botón de eliminar	
	5. El sistema borra de la base de datos todos los registros.
3.b El usuario da al check de modificar el registro	
	6. El sistema muestra un formulario con los datos de ese registro.
7. El usuario marca el chek de anulado	
	8. El sistema guarda los datos modificados

3.4.2.43.- Modificar Arreglos de modista

Caso de uso: Modificar arreglos de modista

Resumen de la funcionalidad: El cliente desea modificar arreglos de modista

Actores: Administrador (iniciador)

Precondición: El arreglo de modista que desea modificar debe estar listado.

Postcondición: Se habrán modificado todos los registros que dependan de éste.

Descripción detallada: El usuario una vez que se encuentra en el listado de arreglos de modista, podrá modificar cualquiera de estos arreglos de modista, para ello deberá apretar el botón de modificar.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clicca al botón del content frame de arreglos de modista	
	2. El sistema pide los datos necesarios a la base de datos de la tabla arreglosmodista para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario pincha sobre el botón de modificar.	
	4. El sistema modifica el registro seleccionado con los datos modificados.

Cursos alternativos:

Línea 4: Si el usuario al modificar los datos, introduce algún dato que no sea correcto para el sistema, este mostrará un mensaje de error para que el usuario pueda corregirlo.

3.4.2.44.- Buscar por descripción de arreglos de modista

Caso de uso: Búsqueda por descripción de arreglos de modista

Resumen de la funcionalidad: El cliente desea realizar una búsqueda de descripción de arreglos de modista.

Actores: Administrador (iniciador)

Precondición: se debe encontrar en el listado de arreglos de modista

Postcondición: Ninguna.

Descripción detallada: El usuario una vez que se encuentra en el listado de arreglos de modista, podrá realizar la búsqueda por descripción de arreglos de modista. Para ello deberá escribir la cadena por la que quiere realizar la búsqueda y pulsar el botón de búsqueda.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clicó al botón del content frame de arreglos de modista	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de arreglosmodista para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario introduce la cadena de caracteres sobre la que quiere realizar la búsqueda.	
4. El usuario deberá apretar el botón de búsqueda.	
	5. El sistema devuelve los datos encontrados en la búsqueda agrupados de 10 en 10

Cursos alternativos:

Línea 5: El sistema devuelve todas las descripciones de arreglos de modista que se encuentre que contengan todos los caracteres que el usuario ha introducido para la búsqueda. Tanto si los encuentra al principio, en medio o al final de la descripción de arreglos de modista que esta dada de alta en la base de datos.

3.4.2.45.- Listar Stock

Caso de uso: Listar Stock

Resumen de la funcionalidad: El cliente desea listar el Stock.

Actores: Administrador (iniciador)

Precondición: Tienen que existir datos en la base de datos en la tabla articlotienda

Postcondición: Ninguna

Descripción detallada: El usuario selecciona el botón de stock para poder listarlo. Una vez entrado en el listado de stock podrá verlo todo ordenados por código de artículo y paginadas de 10 en 10, con los botones de página anterior y siguiente podrá moverse a través de las diferentes páginas.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de stock	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de articlotienda para poder listar las páginas de datos agrupadas de 10 en 10

3.4.2.46.- Dar de Alta Stock

Caso de uso: Dar de alta stock

Resumen de la funcionalidad: El cliente desea dar de alta stock

Actores: Administrador (iniciador)

Precondición: -

Postcondición: Se ha dado de alta un nuevo stock.

Descripción detallada: El usuario una vez que se encuentra en el listado de stock, podrá dar de alta un nuevo stock. Para ello deberá dar al botón de dar de alta y a continuación se mostrará el formulario que deberá rellenar con los datos adecuados.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de stock	
	2. El sistema pide los datos necesarios a la base de datos de la tabla articulo tienda para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario hace clic en el botón de dar de alta.	
	4. Aparece el formulario de los datos a dar de alta en la tabla de articulo tienda
5. El usuario rellena el mantenimiento con los campos necesarios.	
	6. El sistema valida que los campos introducidos son los correctos

Cursos alternativos:

Línea 6: Si el usuario introduce algún dato incorrecto o algún artículo con una misma tienda el sistema mostrará un mensaje de error.

3.4.2.47.- Dar de Baja Stock

Caso de uso: Dar de baja stock

Resumen de la funcionalidad: El cliente desea dar de baja stock

Actores: Administrador (iniciador)

Precondición: El stock que desea dar de baja debe estar listado.

Postcondición: Se habrán borrado todos los stocks seleccionados.

Descripción detallada: El usuario una vez que se encuentra en el listado de stock, podrá dar de baja el stock. Para ello deberá chequear los registros que desee dar de baja y apretar el botón de eliminar.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de stock	
	2. El sistema pide los datos necesarios a la base de datos de la tabla articulotienda para poder listar las páginas de datos agrupadas de 10 en 10
3.a El usuario marca todos los registros que quiere eliminar	
4. El usuario da al botón de eliminar	
	5. El sistema borra de la base de datos todos los registros.

3.4.2.48.- Modificar Stock

Caso de uso: Dar de baja stock

Resumen de la funcionalidad: El cliente desea dar de baja stock

Actores: Administrador (iniciador)

Precondición: El stock que desea modificar debe estar listado.

Postcondición: Se habrán modificado todos los registros que dependan de éste.

Descripción detallada: El usuario una vez que se encuentra en el listado de stock, podrá modificar cualquiera de estos stocks, para ello deberá apretar el botón de modificar. Lo único que no podrá modificar será el código de artículo y de tienda, que aparecerá deshabilitado.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clicó al botón del content frame de stock	
	2. El sistema pide los datos necesarios a la base de datos de la tabla articulo tienda para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario pincha sobre el botón de modificar.	
	4. El sistema modifica el registro seleccionado con los datos modificados.

Cursos alternativos:

Línea 4: Si el usuario al modificar los datos, introduce algún dato que no sea correcto para el sistema, este mostrará un mensaje de error para que el usuario pueda corregirlo

3.4.2.49.- Buscar por código artículo

Caso de uso: Búsqueda por código artículo

Resumen de la funcionalidad: El cliente desea realizar una búsqueda de código de artículo

Actores: Administrador (iniciador)

Precondición: se debe encontrar en el listado de stock

Postcondición: Ninguna.

Descripción detallada: El usuario una vez que se encuentra en el listado de stock, podrá realizar la búsqueda por código de artículo. Para ello deberá escribir la cadena por la que quiere realizar la búsqueda y pulsar el botón de búsqueda.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de stock	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de articulotienda para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario introduce la cadena de caracteres sobre la que quiere realizar la búsqueda.	
4. El usuario deberá apretar el botón de búsqueda.	
	5. El sistema devuelve los datos encontrados en la búsqueda agrupados de 10 en 10

Cursos alternativos:

Línea 5: El sistema devuelve todos los códigos de artículo que se encuentre que contengan todos los caracteres que el usuario ha introducido para la búsqueda. Tanto si los encuentra al principio, en medio o al final del código de artículo dado de alta en la base de datos de articulotienda.

3.4.2.50.- Buscar por código de tienda

Caso de uso: Búsqueda por código de tienda

Resumen de la funcionalidad: El cliente desea realizar una búsqueda por código de tienda

Actores: Administrador (iniciador)

Precondición: se debe encontrar en el listado de stock

Postcondición: Ninguna.

Descripción detallada: El usuario una vez que se encuentra en el listado de stock, podrá realizar la búsqueda por código de tienda. Para ello deberá escribir la cadena por la que quiere realizar la búsqueda y pulsar el botón de búsqueda.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clica al botón del content frame de stock	
	2. El sistema pide los datos necesarios a la base de datos de la tabla de articulotienda para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario introduce la cadena de caracteres sobre la que quiere realizar la búsqueda.	
4. El usuario deberá apretar el botón de búsqueda.	
	5. El sistema devuelve los datos encontrados en la búsqueda agrupados de 10 en 10

Cursos alternativos:

Línea 5: El sistema devuelve todos los códigos de tienda que se encuentre que contengan todos los caracteres que el usuario ha introducido para la búsqueda. Tanto si los encuentra al principio, en medio o al final del código de tienda dado de alta en la base de datos de articulotienda.

3.4.2.51.- Ir a la página siguiente


Caso de uso: Ir a la página siguiente

Resumen de la funcionalidad: El usuario clic a las flechas que se encuentran en la parte inferior de los listados

Actores: Administrador (iniciador)

Precondición: se debe encontrar en alguno de los listados y estos deben superar la cantidad de 10 elementos y el usuario debe estar colocado en una página que no sea la última.

Postcondición: NINGUNA.

Descripción detallada: El usuario una vez ha entrado en alguno de los listados de mantenimiento y si en cualquiera de estos nos encontramos que supera los 10 elementos, en la parte de debajo de los listados se encontrará las flechas de movimiento para poder ir moviéndose por los diferentes datos del listado, una vez haga clic sobre el botón  se mostrará la siguiente página.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clic al botón del content frame de cualquiera de los mantenimientos	
	2. El sistema pide los datos necesarios a la base de datos de la tabla correspondiente al mantenimiento para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario clic el botón de siguiente	
	5. El sistema devuelve los datos de la siguiente página agrupados de 10 en 10

3.4.2.52.- Ir a la página anterior


Caso de uso: Ir a la página anterior

Resumen de la funcionalidad: El usuario clic a las flechas que se encuentran en la parte inferior de los listados.

Actores: Administrador (iniciador)

Precondición: se debe encontrar en alguno de los listados y estos deben superar la cantidad de 10 elementos y el usuario debe estar posicionado en una página que no sea la primera

Postcondición: NINGUNA.

Descripción detallada: El usuario una vez ha entrado en alguno de los listados de mantenimiento y si en cualquiera de estos nos encontramos que supera los 10 elementos, en la parte de debajo de los listados se encontrará las flechas de movimiento para poder ir moviéndose por los diferentes páginas del listado, una vez haga clic sobre el botón  se mostrará la página anterior.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clic al botón del content frame de cualquiera de los mantenimientos	
	2. El sistema pide los datos necesarios a la base de datos de la tabla correspondiente al mantenimiento para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario clic el botón de anterior	
	5. El sistema devuelve los datos de la siguiente página agrupados de 10 en 10

3.4.2.53.- Ir a la página número N

Caso de uso: Ir a la página número N

Resumen de la funcionalidad: El usuario clic a la página número N.

Actores: Administrador (iniciador)

Precondición: se debe encontrar en alguno de los listados y estos deben superar la cantidad de 10 elementos.

Postcondición: NINGUNA.

Descripción detallada: El usuario una vez ha entrado en alguno de los listados de mantenimiento y si en cualquiera de estos nos encontramos que supera los 10 elementos, en la parte de debajo de los listados se encontrará las flechas de y los números de las diferentes páginas, una vez haga clic en alguna de las páginas el sistema mostrará el listado de los datos de la página seleccionada por el usuario.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clic al botón del content frame de cualquiera de los mantenimientos	
	2. El sistema pide los datos necesarios a la base de datos de la tabla correspondiente al mantenimiento para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario clic el botón del número de la página	
	5. El sistema devuelve los datos de la página seleccionada agrupados de 10 en 10 y señala en negrita la página seleccionada.

3.4.2.54.- Informes

Caso de uso: Ir a la página número N

Resumen de la funcionalidad: El usuario clic a la página número N.

Actores: Administrador (iniciador)

Precondición: se debe encontrar en alguno de los listados y estos deben superar la cantidad de 10 elementos.

Postcondición: NINGUNA.

Descripción detallada: El usuario una vez ha entrado en alguno de los listados de mantenimiento y si en cualquiera de estos nos encontramos que supera los 10 elementos, en la parte de debajo de los listados se encontrará las flechas de y los números de las diferentes páginas, una vez hace clic en alguna de las páginas el sistema mostrará el listado de los datos de la página seleccionada por el usuario.

Curso de los sucesos

Acciones de los actores	Respuesta del sistema
1. El usuario clic al botón del content frame de cualquiera de los mantenimientos	
	2. El sistema pide los datos necesarios a la base de datos de la tabla correspondiente al mantenimiento para poder listar las páginas de datos agrupadas de 10 en 10
3. El usuario clic el botón del número de la página	
	5. El sistema devuelve los datos de la página seleccionada agrupados de 10 en 10 y señala en negrita la página seleccionada.

4.- Análisis de la aplicación

Una vez mostrados los casos de uso pasaré a mostrar los diagramas de secuencia más importantes de todo el proyecto. El resto de diagramas de secuencia no se han incluido porque entiendo que se basan en las mismas operaciones. Para realizar los diagramas de secuencia utilizaremos los modelos y diagramas del UML.

Las operaciones se encuentran en orden tal y como se realizan a través del código, junto a ellas vienen algunos textos explicativos que indican etiquetas y resultados importantes.

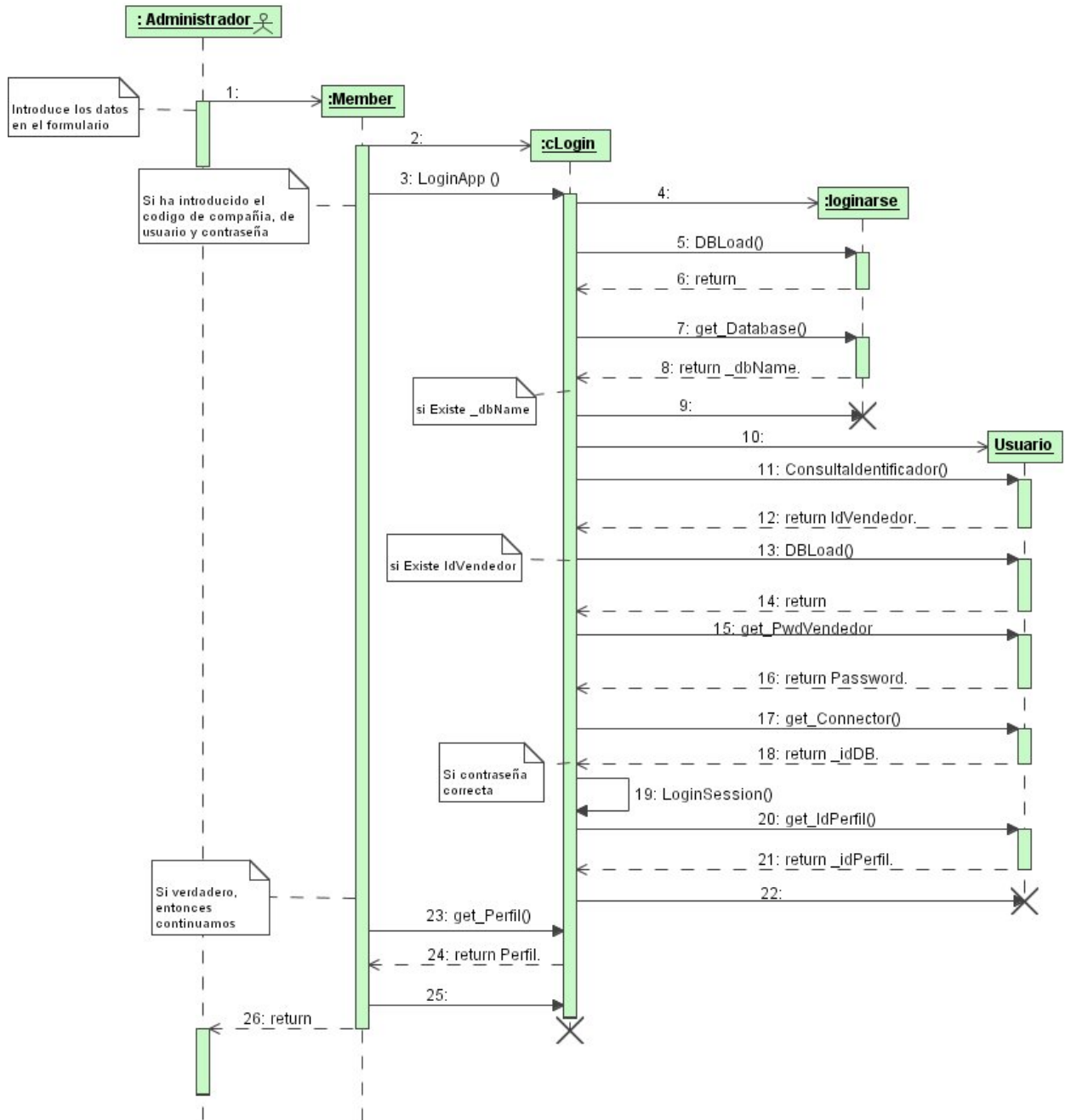
En los diagramas de secuencia se puede observar claramente como interactuamos entre las diferentes capas de interfície, negocio y de datos, a todas ellas se accede de forma jerárquica, a la capa de interfície es a la que accede el usuario a la de negocio accedemos a través de la de interfície y a la de datos accedemos a través de la de negocio.

Además nos encontramos con las diferentes clases de error que son extensiones de la clase MyException.

Mostraré todos los diagramas de secuencia de listar de los datos para mostrar los accesos a las diferentes clases. Una vez visto esto sólo enseñare uno de los diagramas de secuencia para dar de alta, baja, modificación y para las búsquedas.

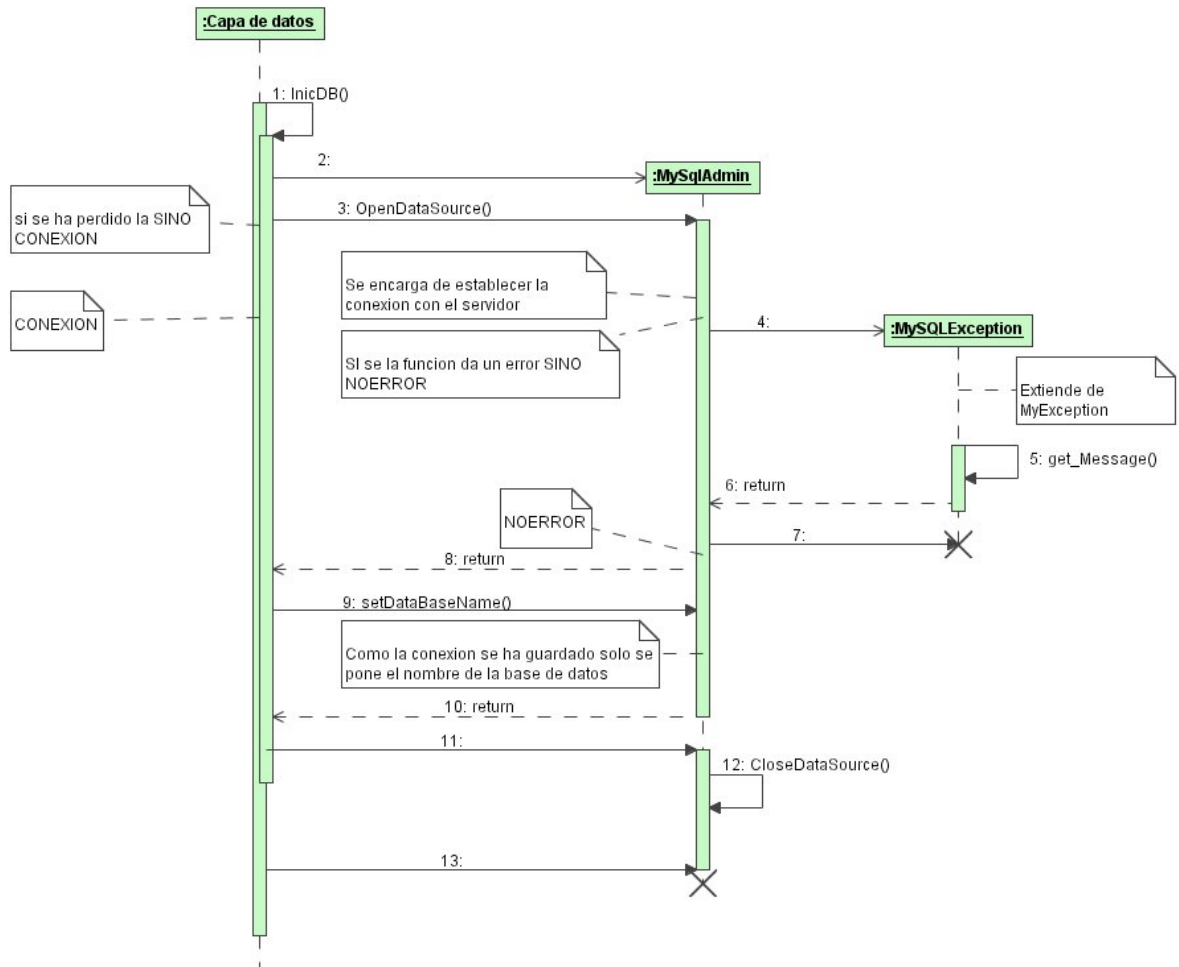
4.1.- Diagrama de secuencia

4.1.1.- Diagrama de secuencia de validación

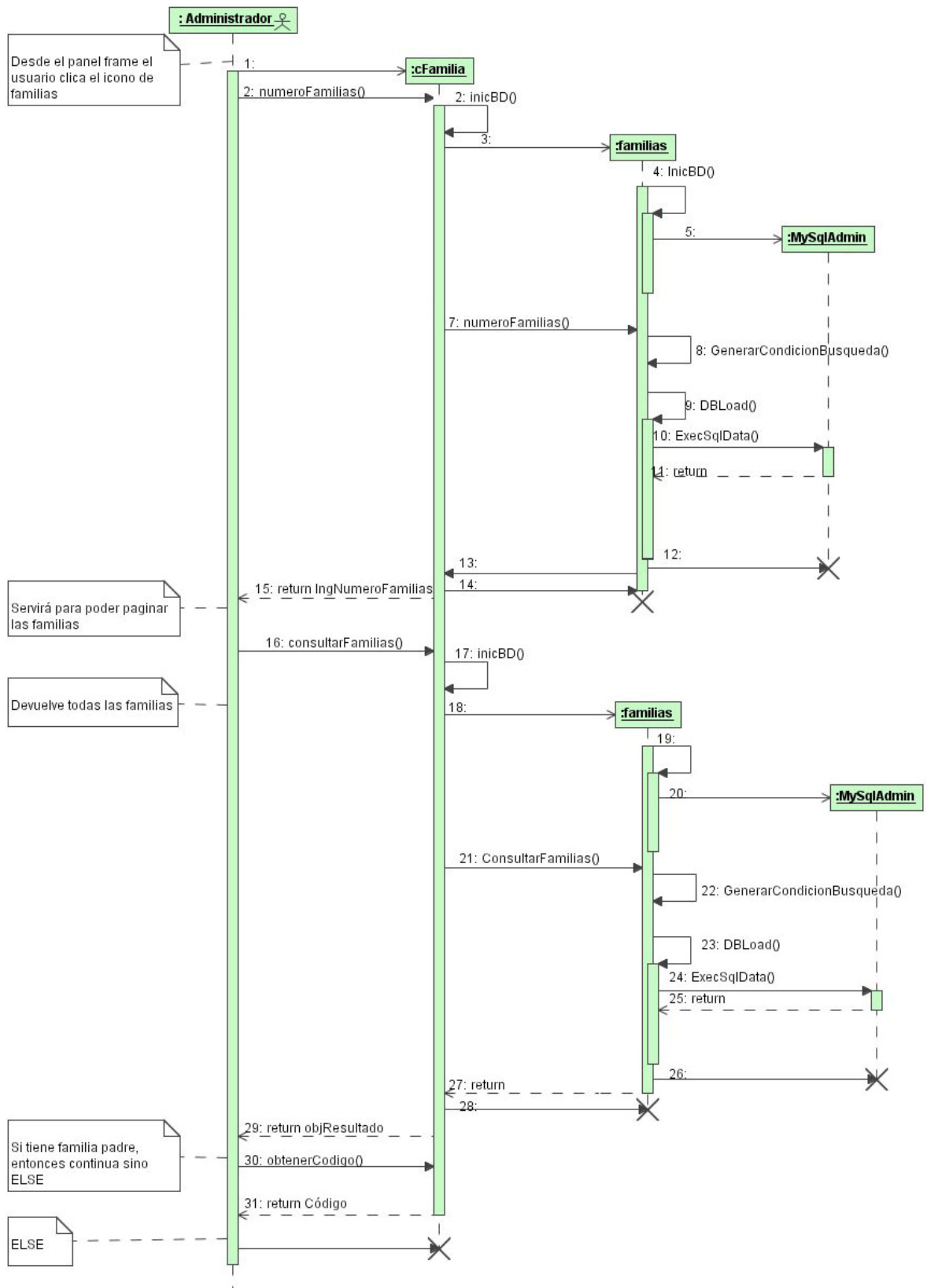


4.1.2.- Diagrama de secuencia de inicDB

Como se trata de una función que hacemos uso de ella en numerosas ocasiones la trataremos de forma concreta, para poder referirnos a ella solo con el nombre y no tener que rehacerla todas las veces que se llame desde el código.

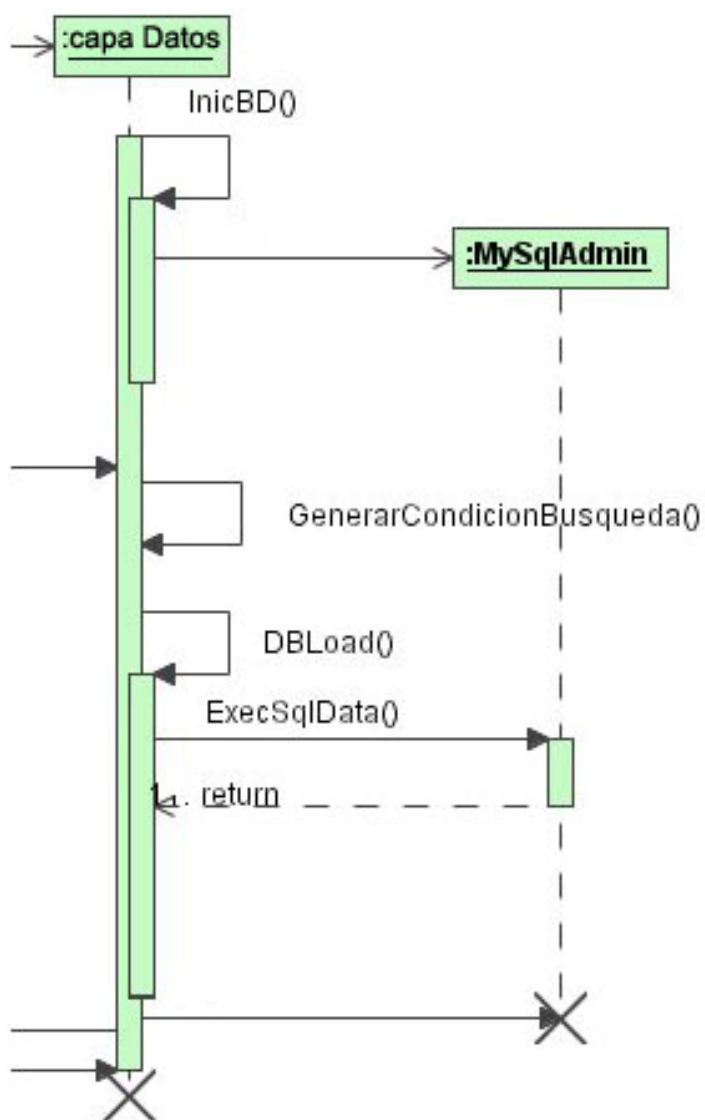


4.1.3.- Diagrama de secuencia de listar Familias

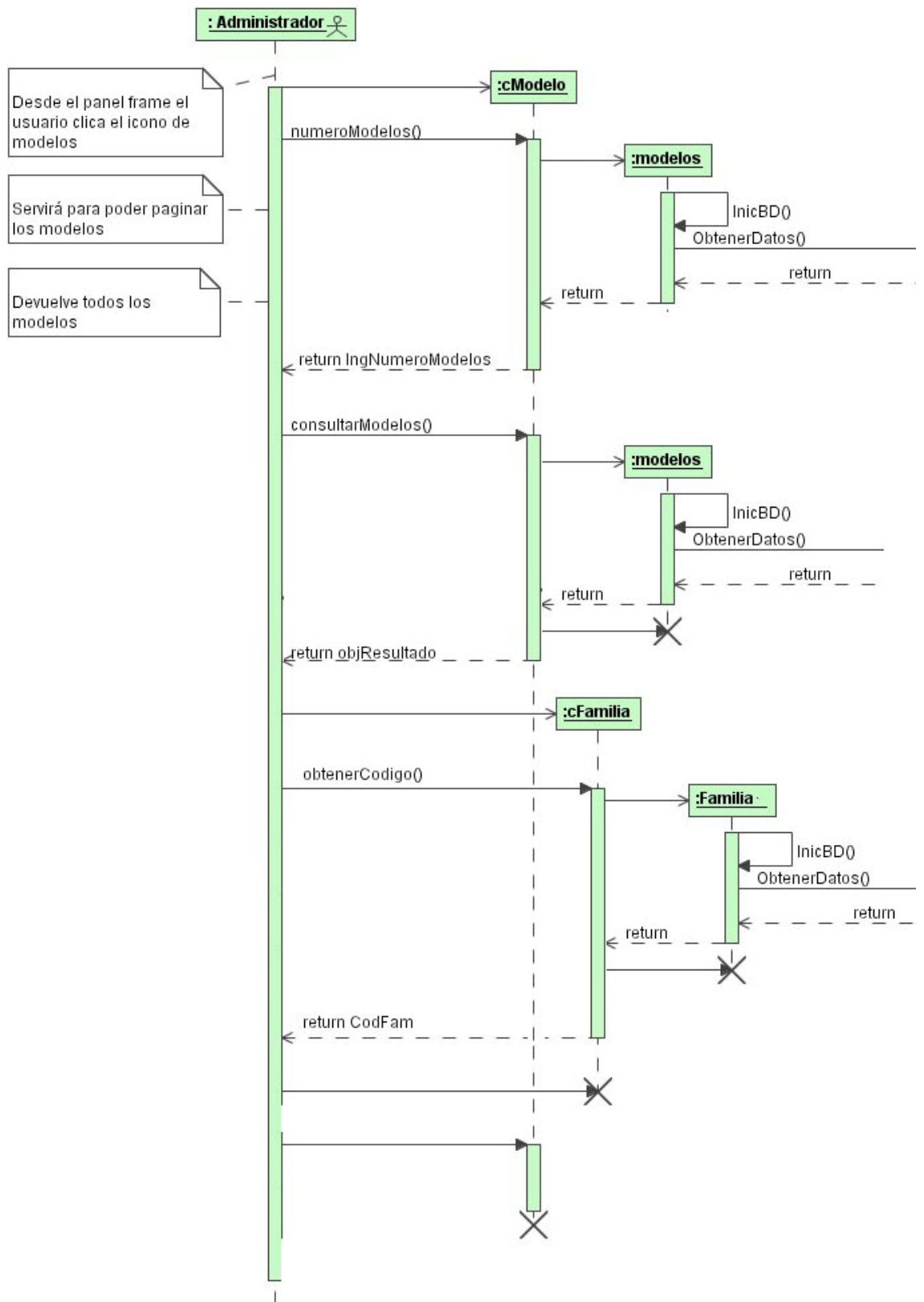


En los siguientes diagramas de secuencia sustituiremos toda la parte de obtención de datos por una llamada a la función ObtenerDatos(). Sustituiremos todo el siguiente diagrama por un llamada a una función ya que entiendo que se trata de lo mismo, crear un objeto a la clase de :MySQLAdmin realizar una consulta a la base de datos, retornar el valor y destruir el objeto.

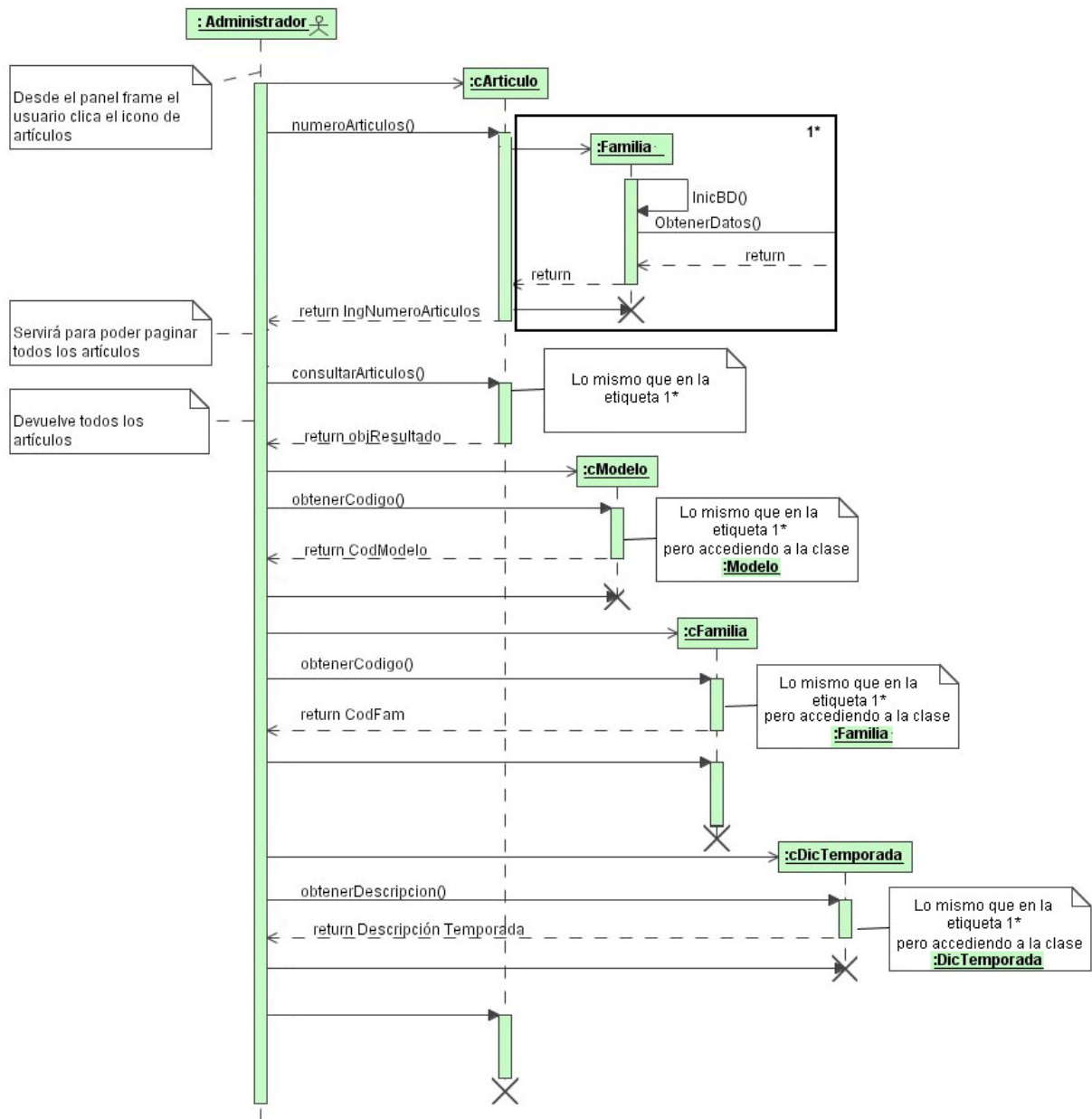
4.1.4.- Diagrama de secuencia de la función ObtenerDatos()



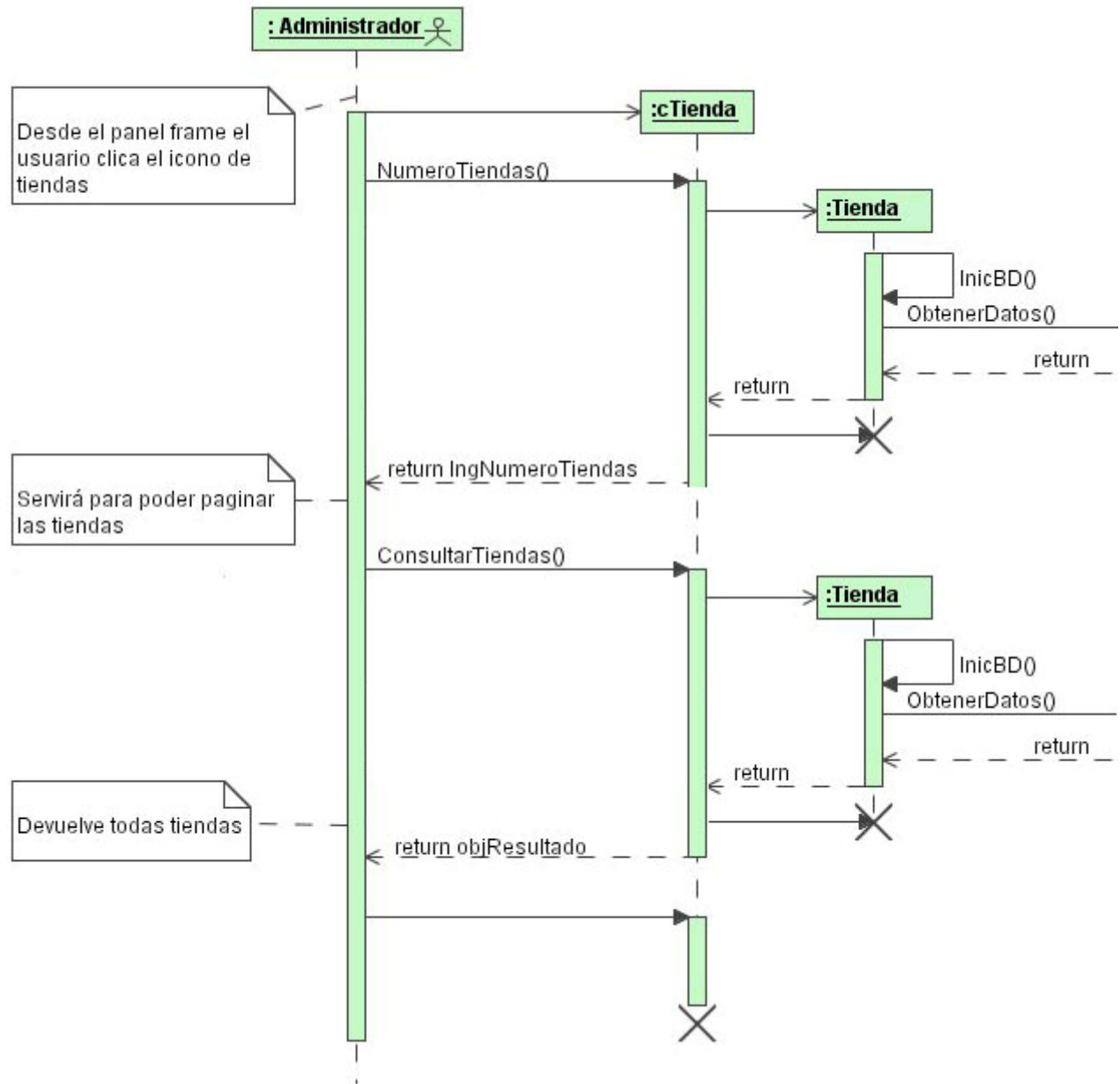
4.1.5- Diagrama de secuencia de listar Modelos



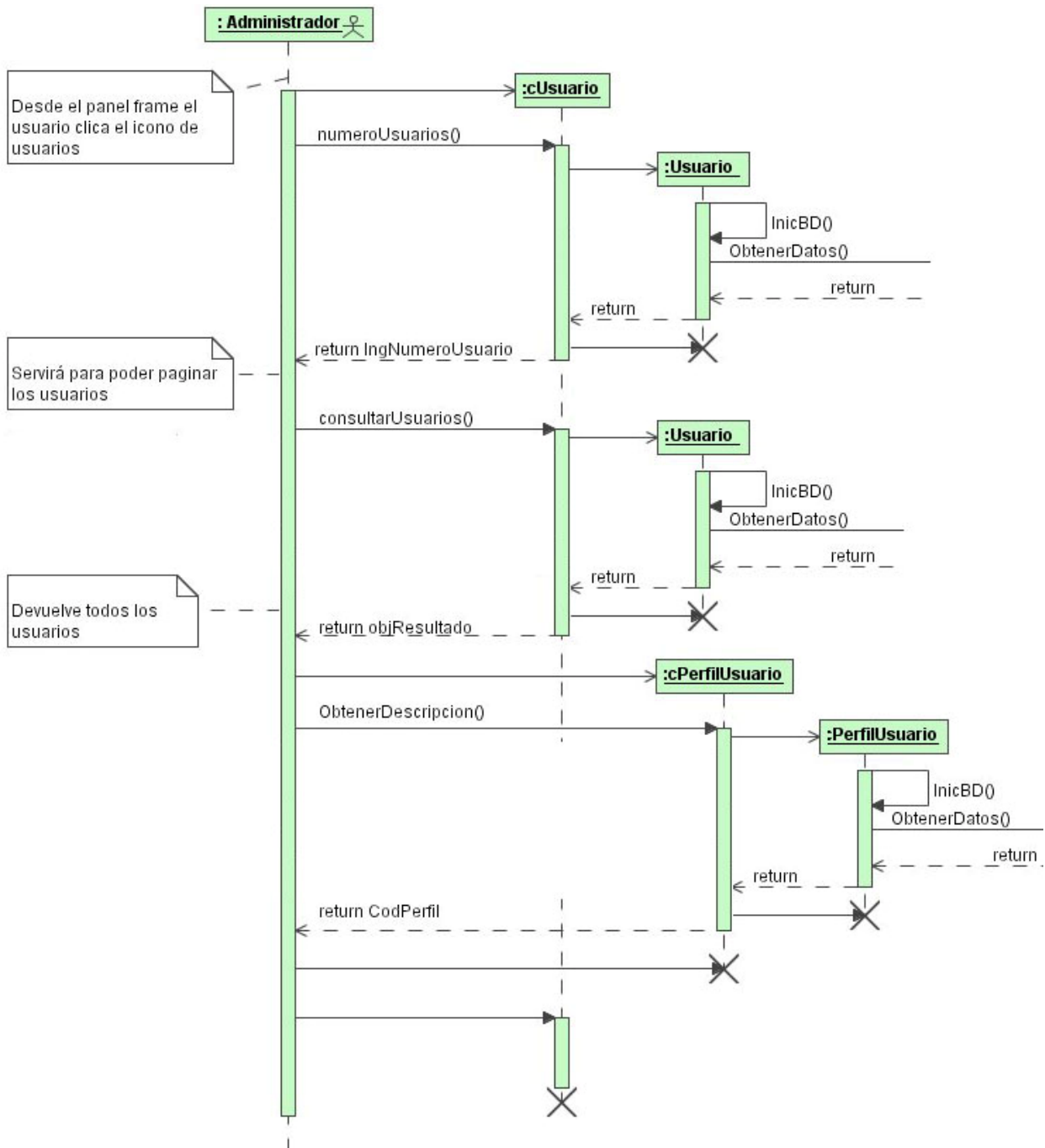
4.1.6- Diagrama de secuencia de listar Artículos



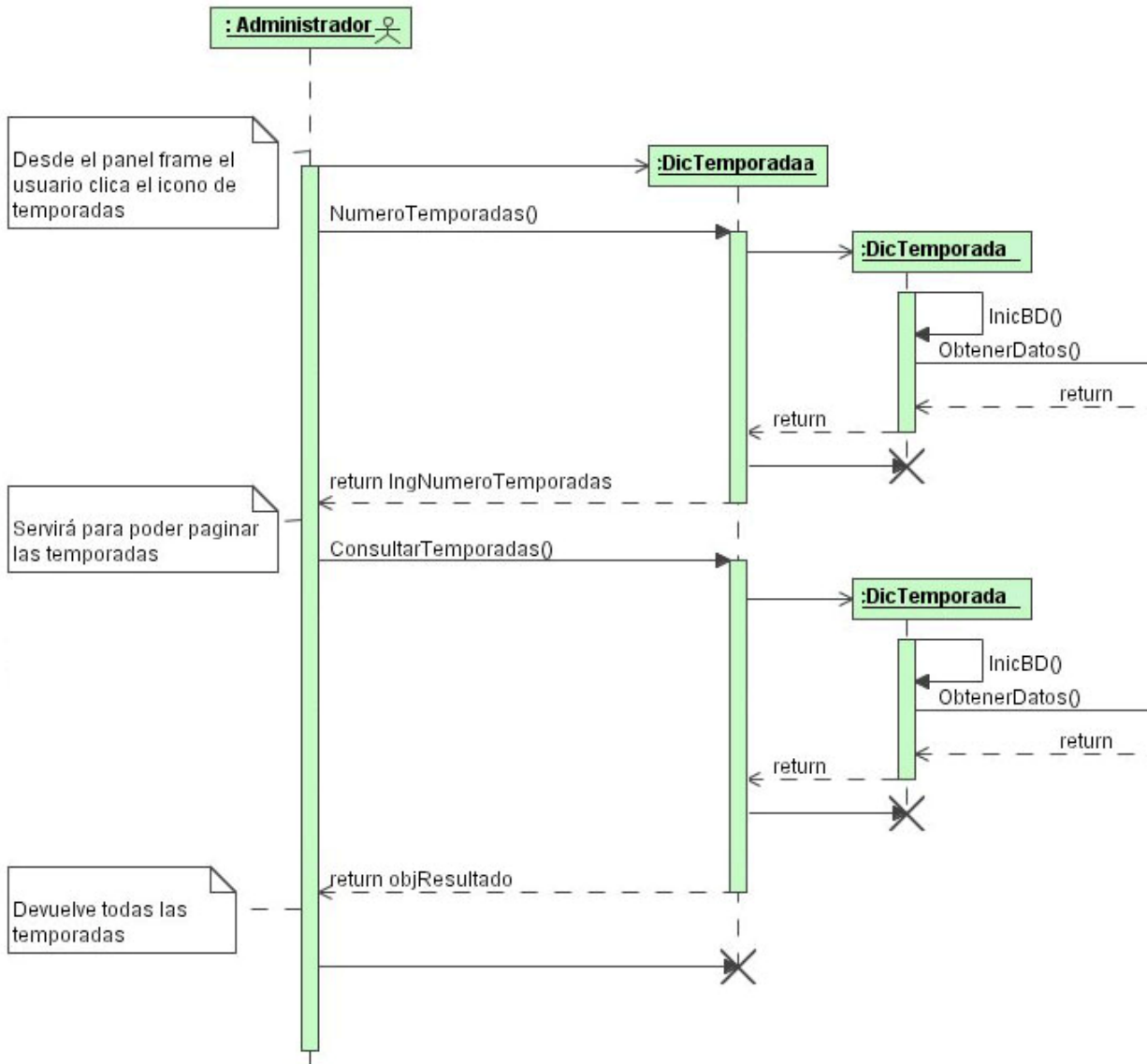
4.1.7.- Diagrama de secuencia de listar Tiendas



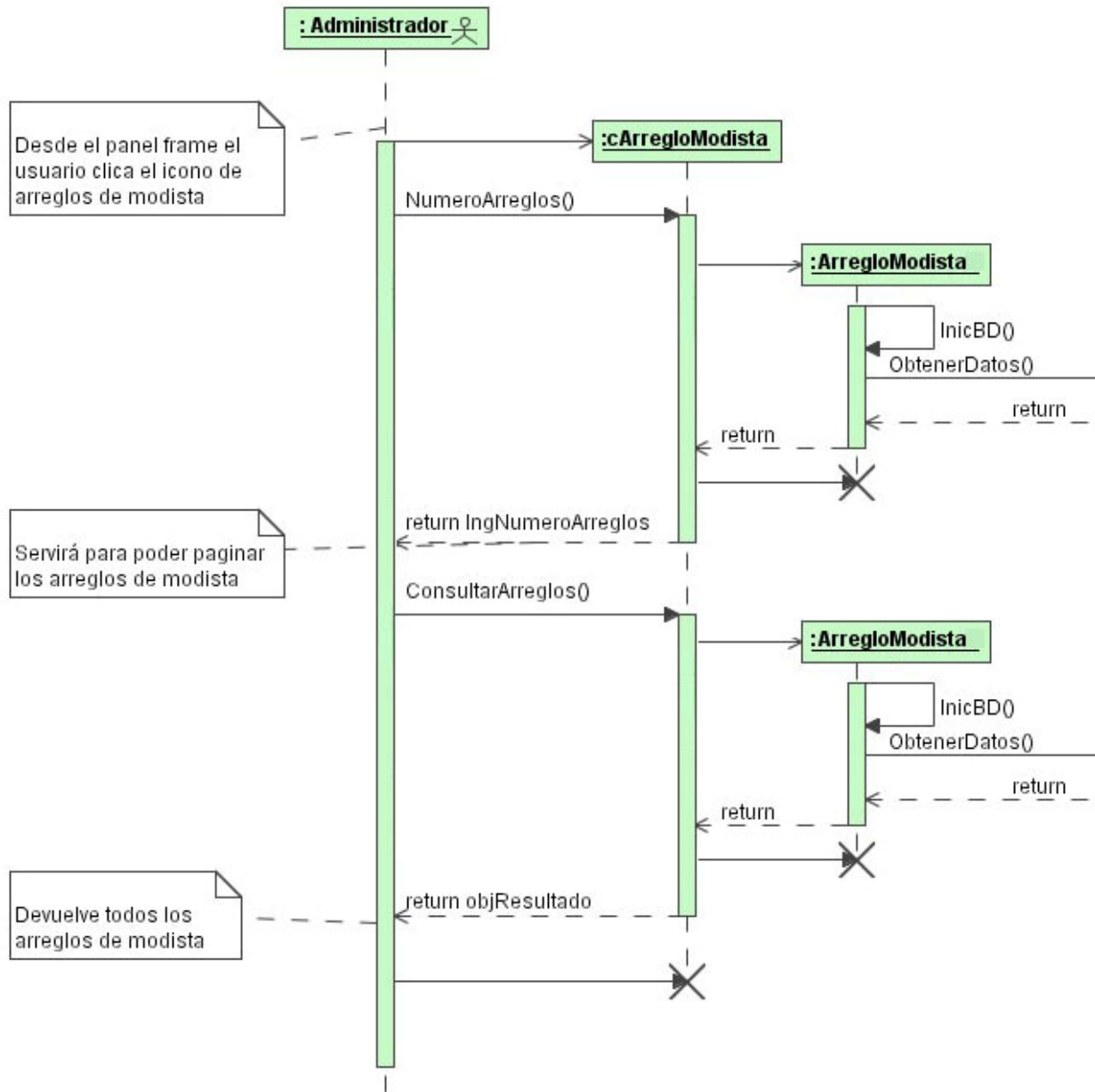
4.1.8.- Diagrama de secuencia de listar Usuarios



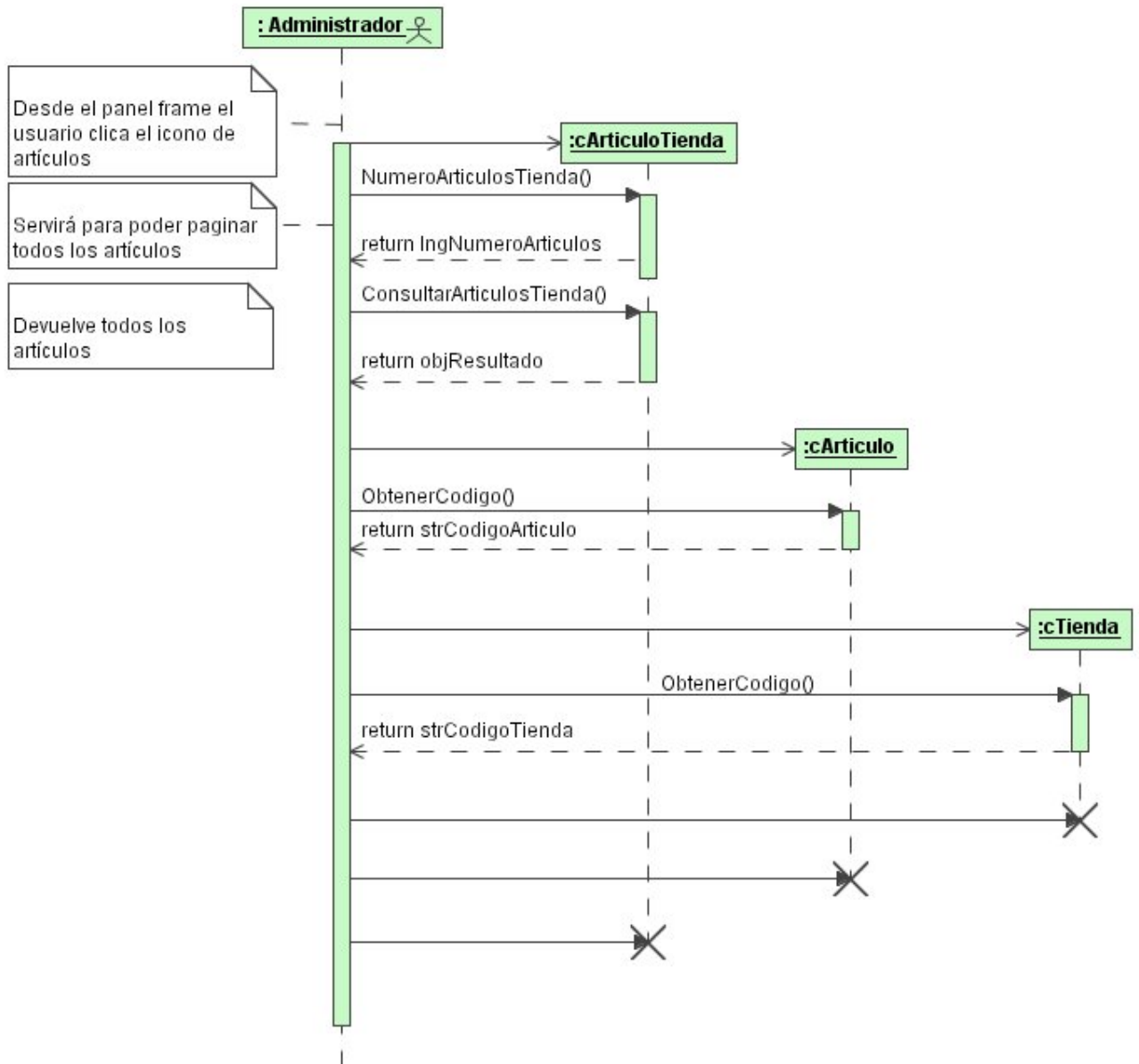
4.1.9.- Diagrama de secuencia de listar Temporadas



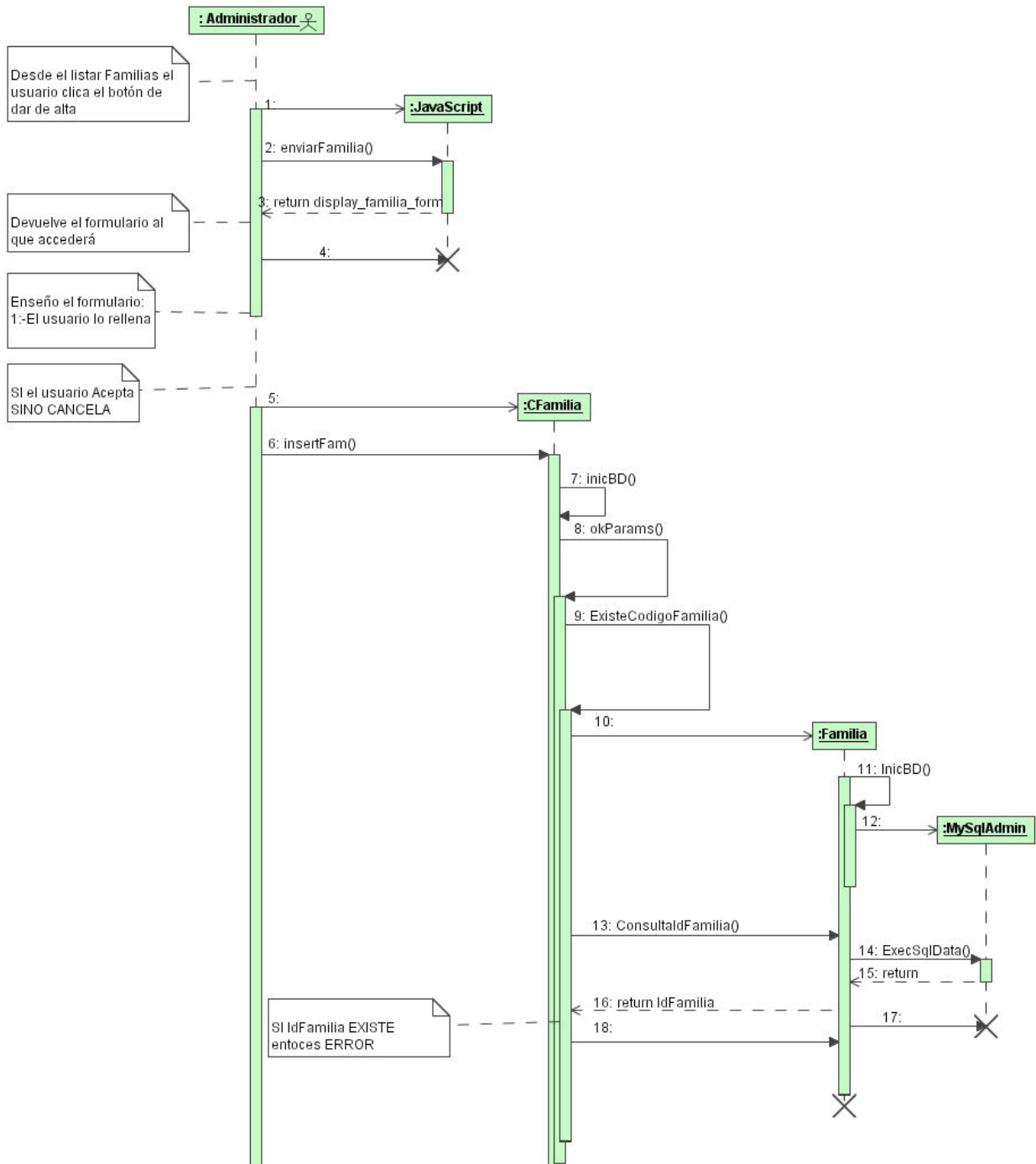
4.1.10.- Diagrama de secuencia de listar Arreglos de modista

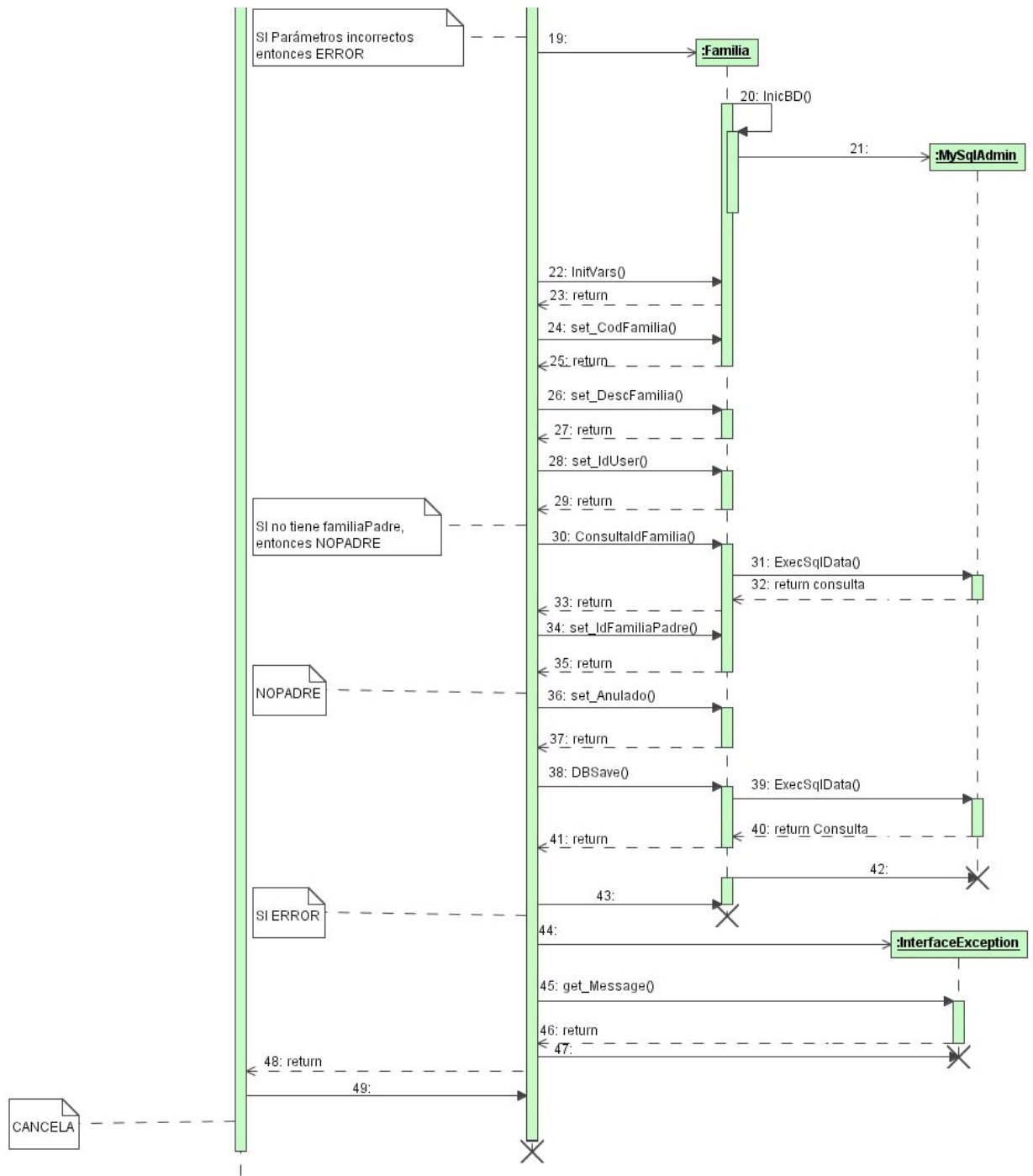


4.1.11.- Diagrama de secuencia de listar Stock

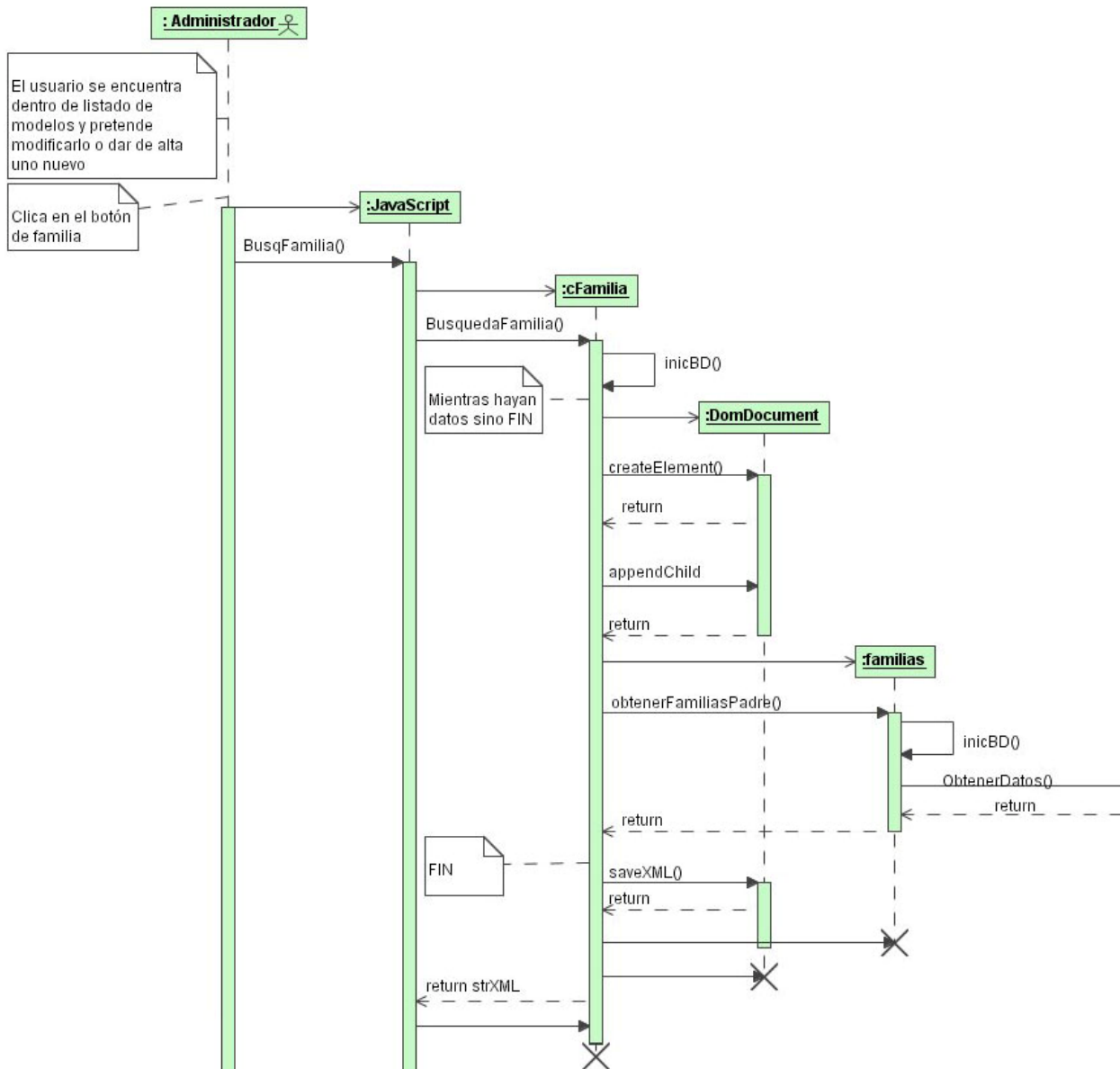


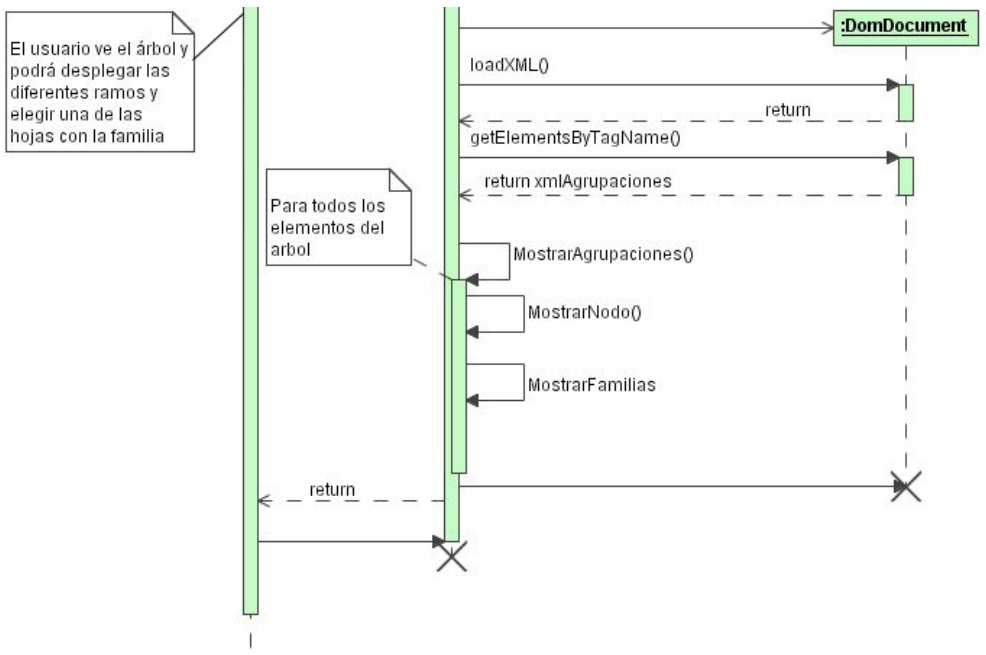
4.1.12.- Diagrama de secuencia de dar de alta Familia



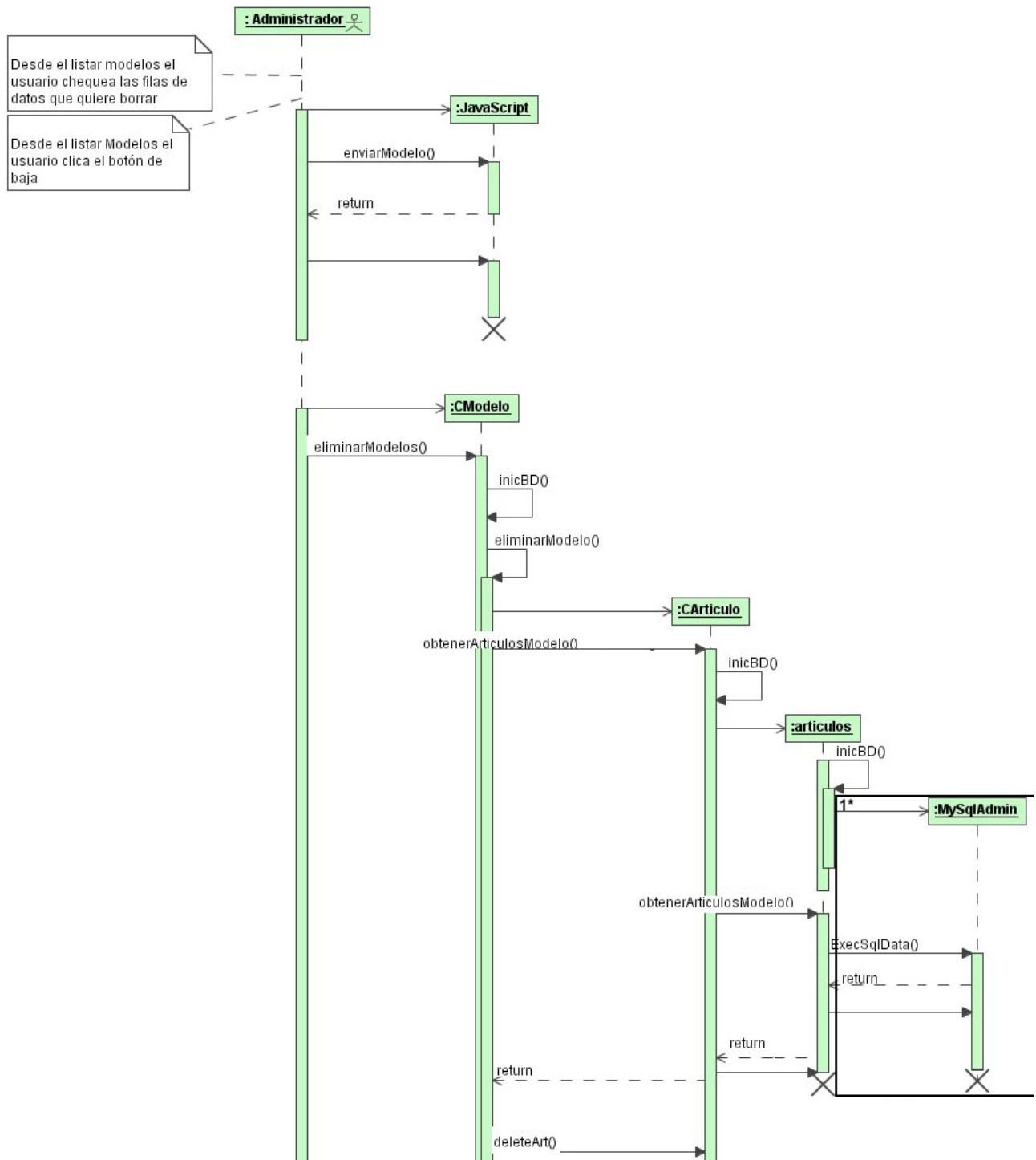


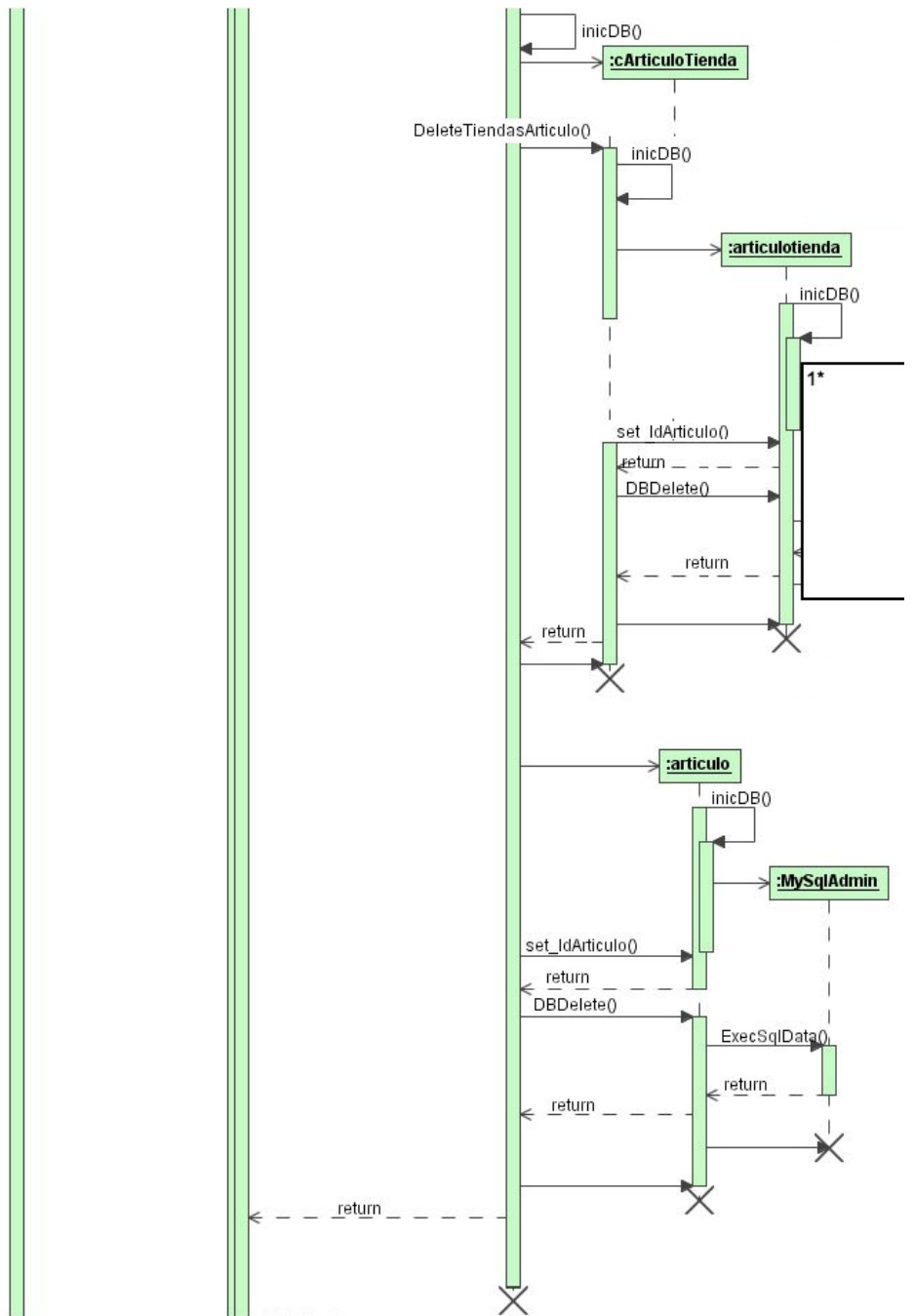
4.1.13.- Diagrama de secuencia de selección de familias por árbol

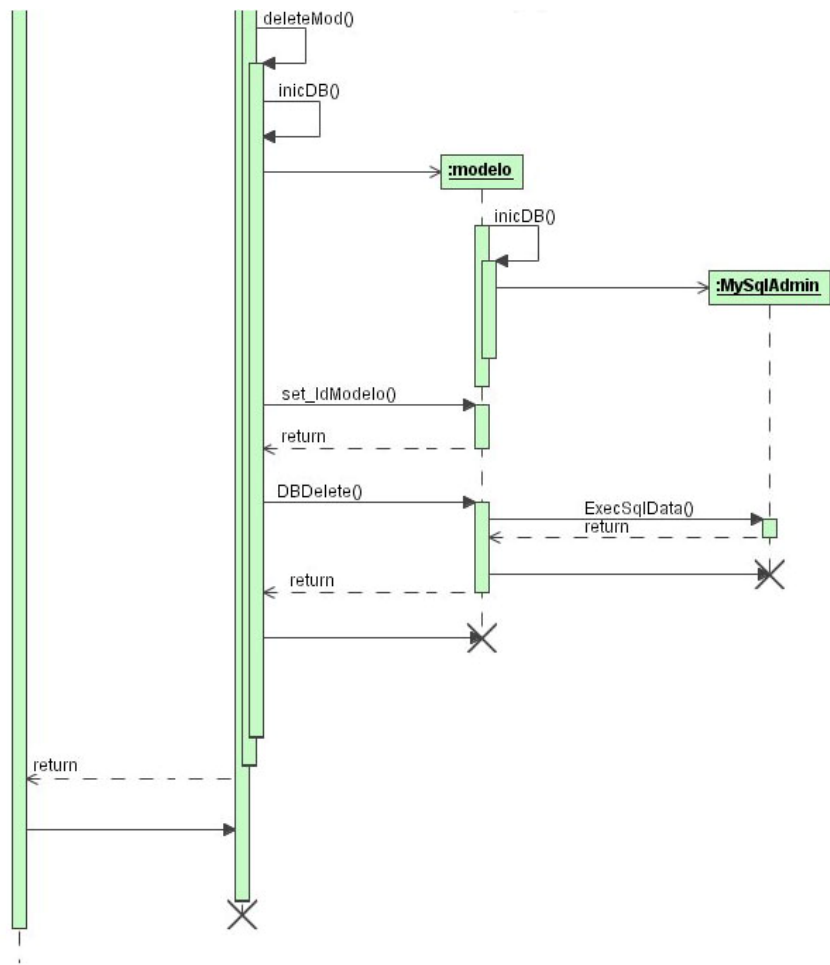




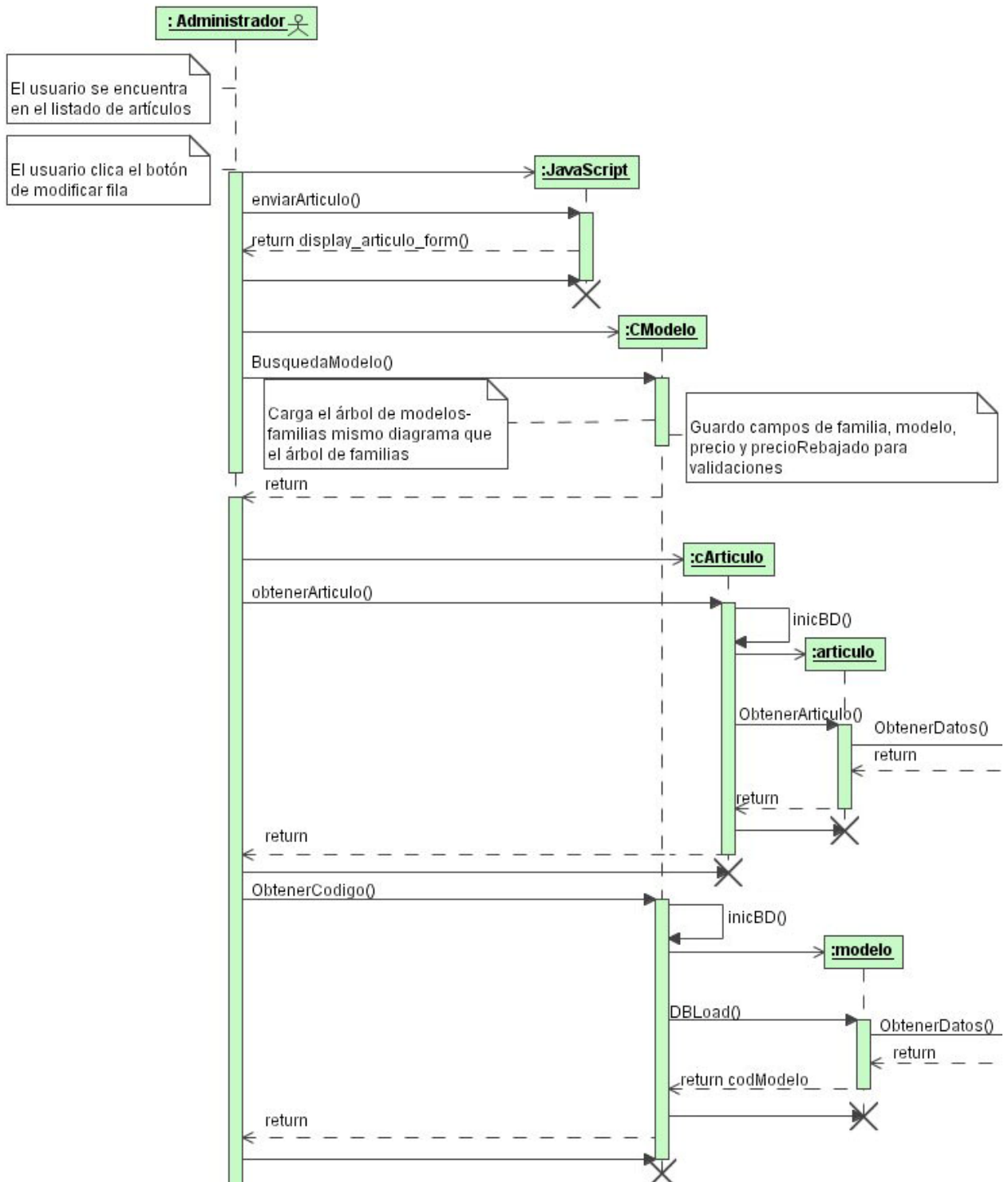
4.1.14.- Diagrama de secuencia de dar de baja Modelos

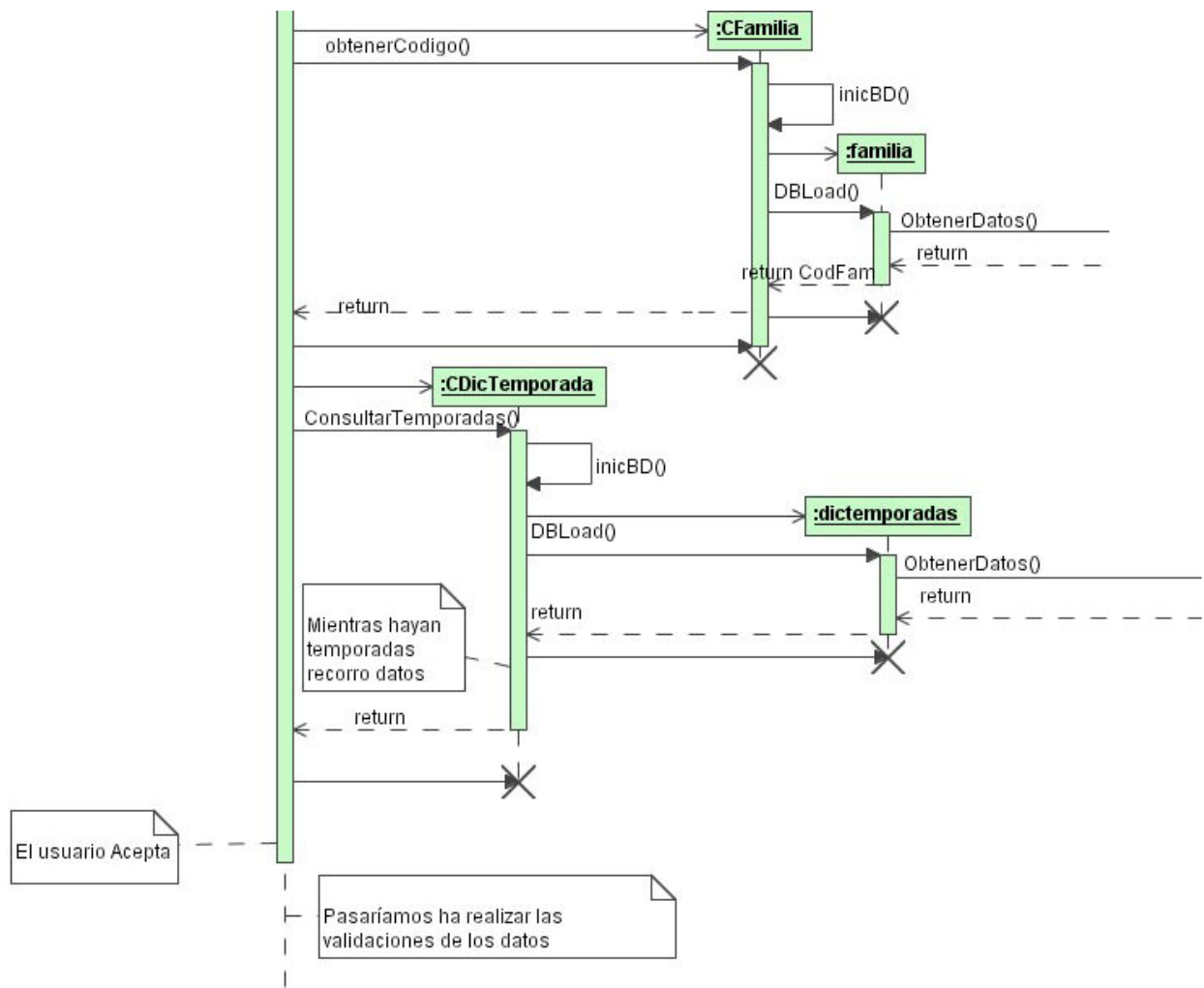


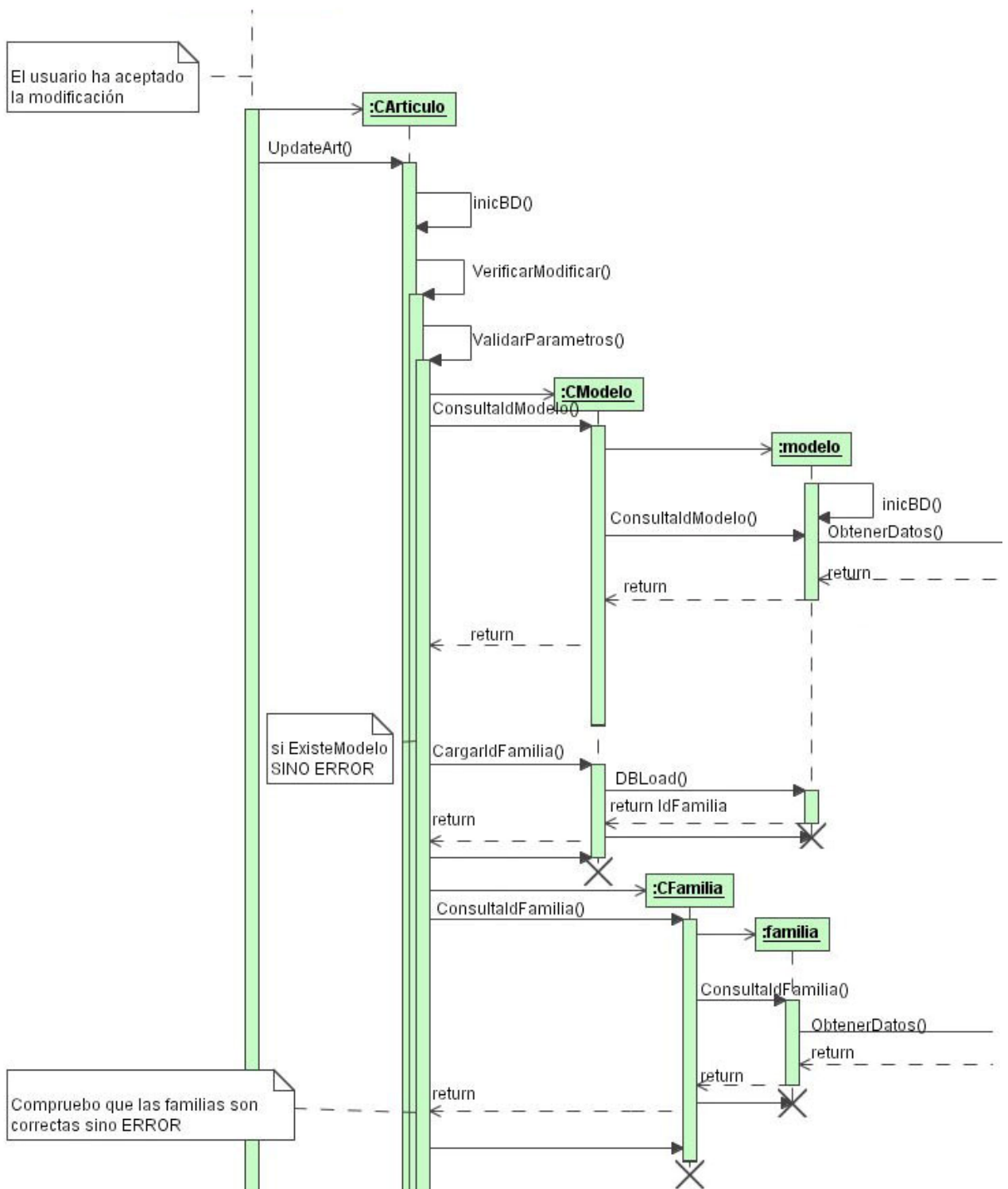


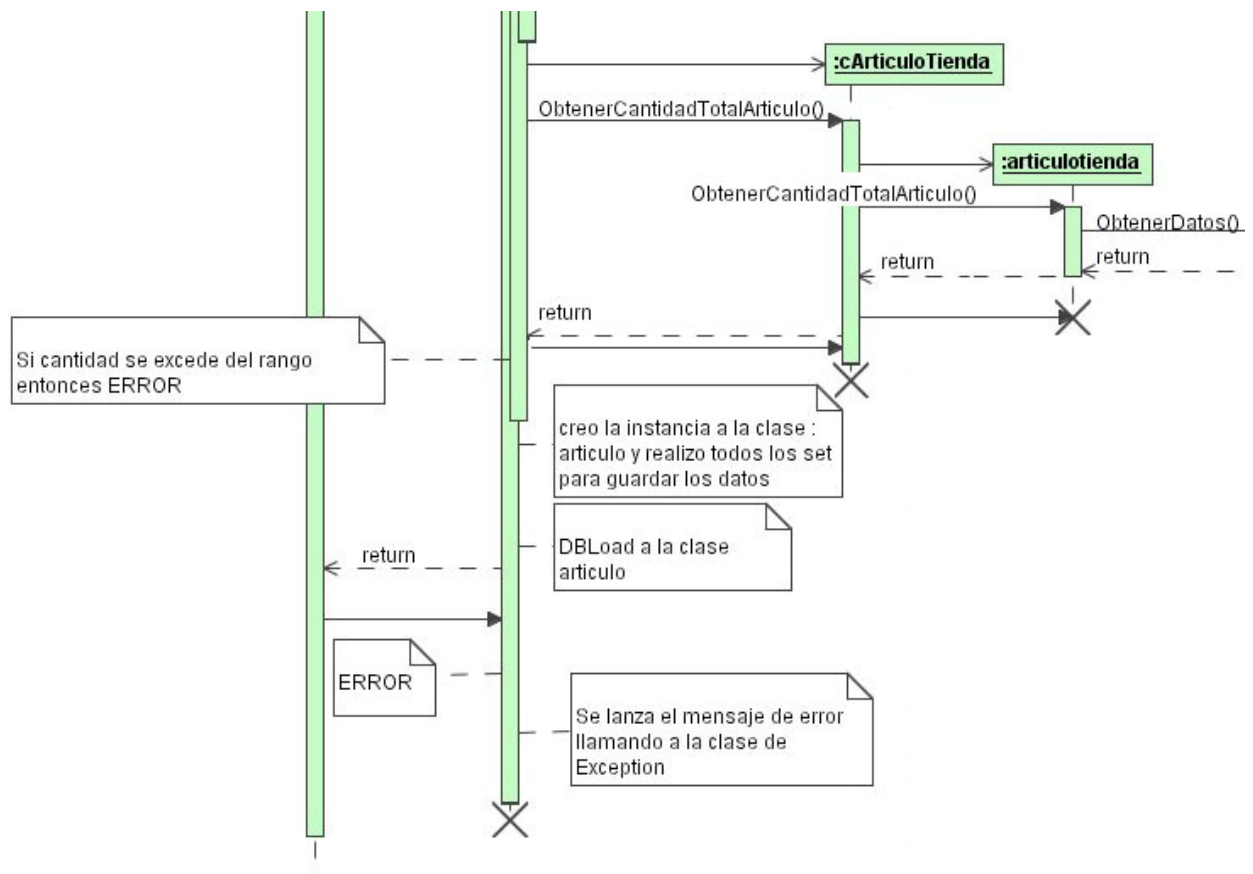


4.1.15.- Diagrama de secuencia de Modificar Artículos

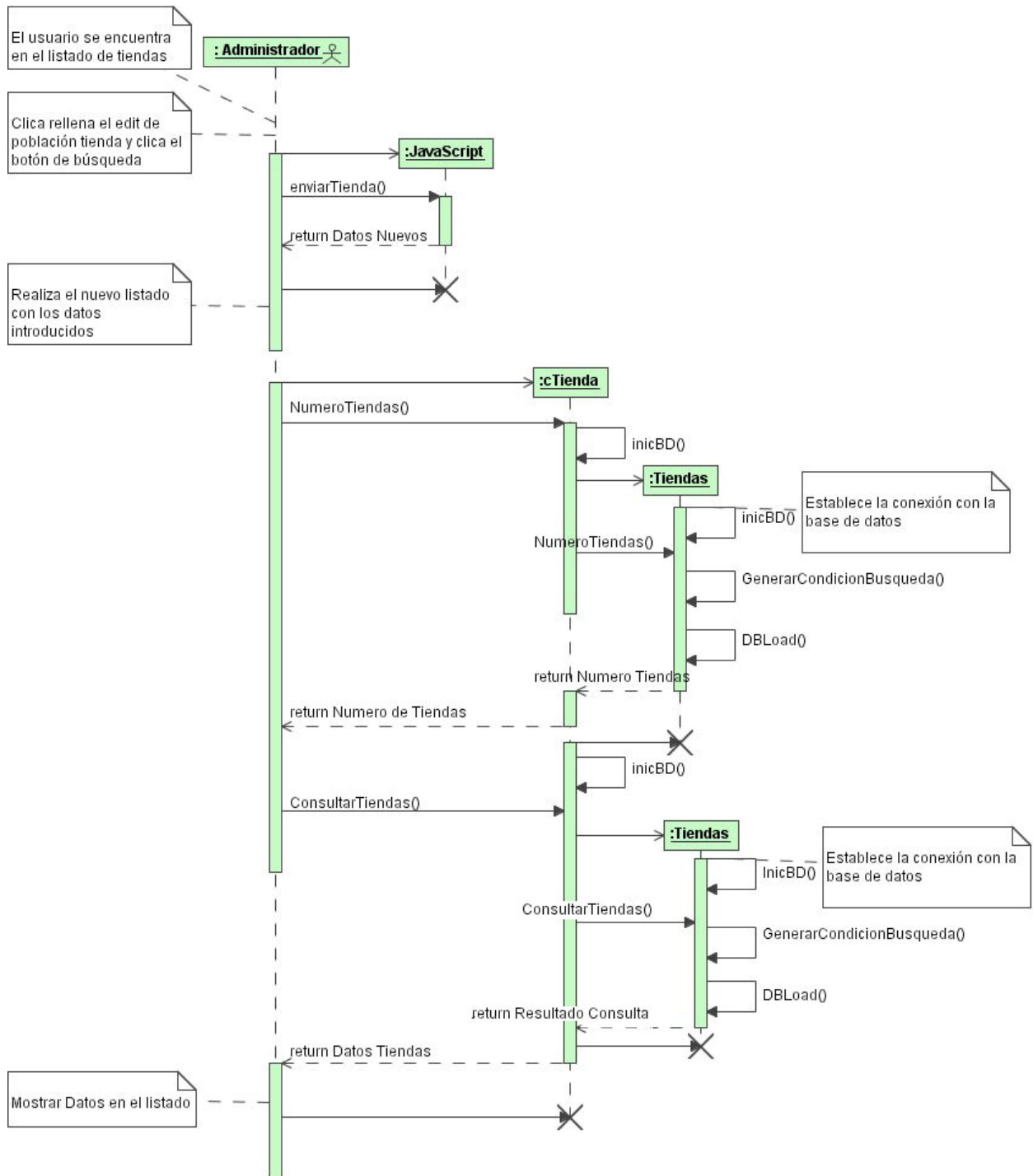




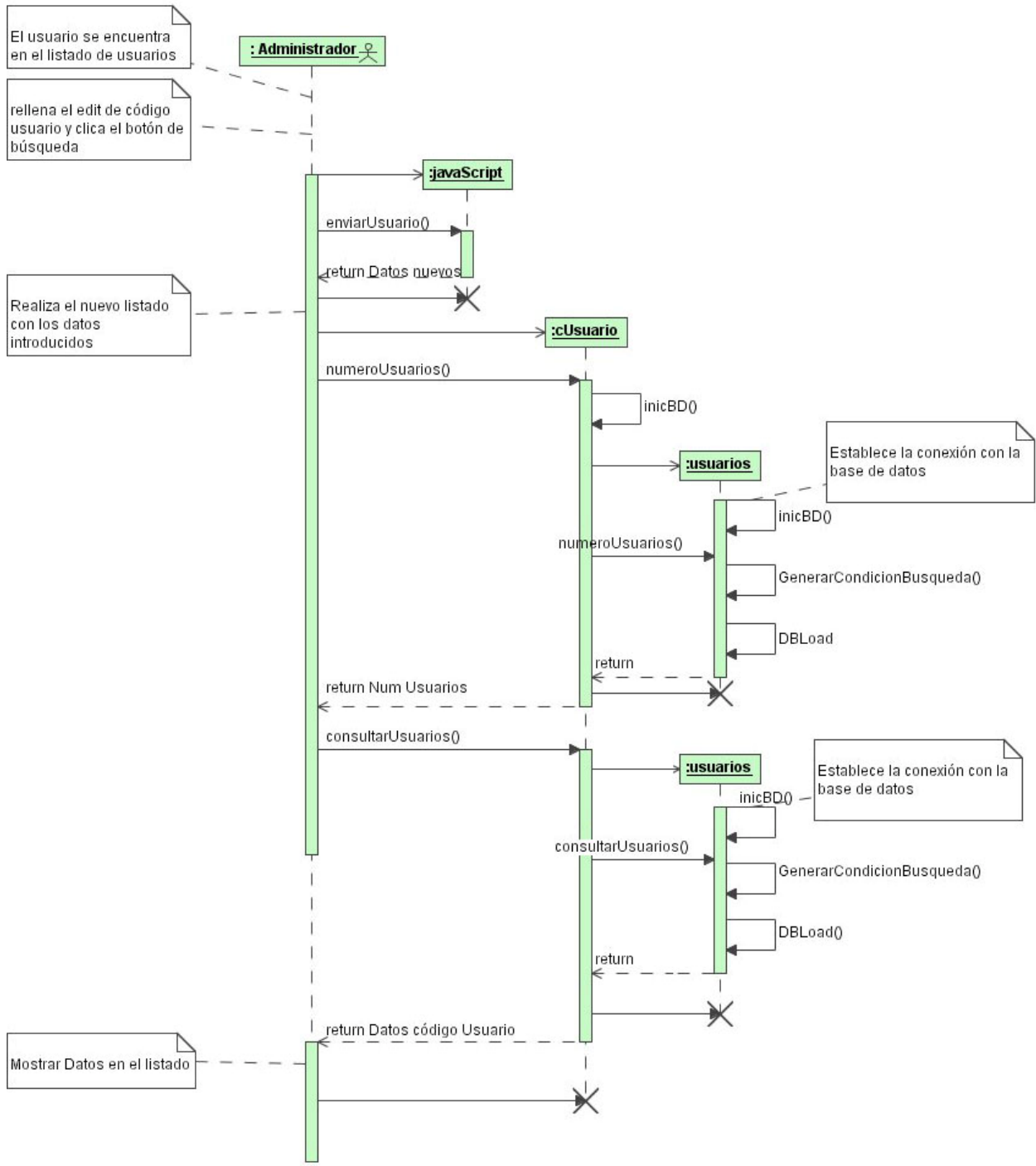





4.1.16.- Diagrama de secuencia de Búsqueda por población de Tiendas



4.1.17.- Diagrama de secuencia de Búsqueda por código de Usuario



4.1.18.- Diagrama de secuencia de ir a la página siguiente

El usuario hace clic sobre el botón  y el listado pasa a la página siguiente del Listar (familias, modelos, artículos,...) dependiendo del mantenimiento en el que se encuentre el usuario. Se la pasa página a la cual tiene que ir y el listar se encarga de agrupar los productos a listar de 10 en 10.

No muestro un diagrama de secuencia en concreto, porque este sería igual que el diagrama de listar alguno de los productos, excepto el inicio que sería el momento en el que el usuario hace clic sobre el botón de siguiente. Se entiende que al ir a la página anterior es exactamente lo mismo.

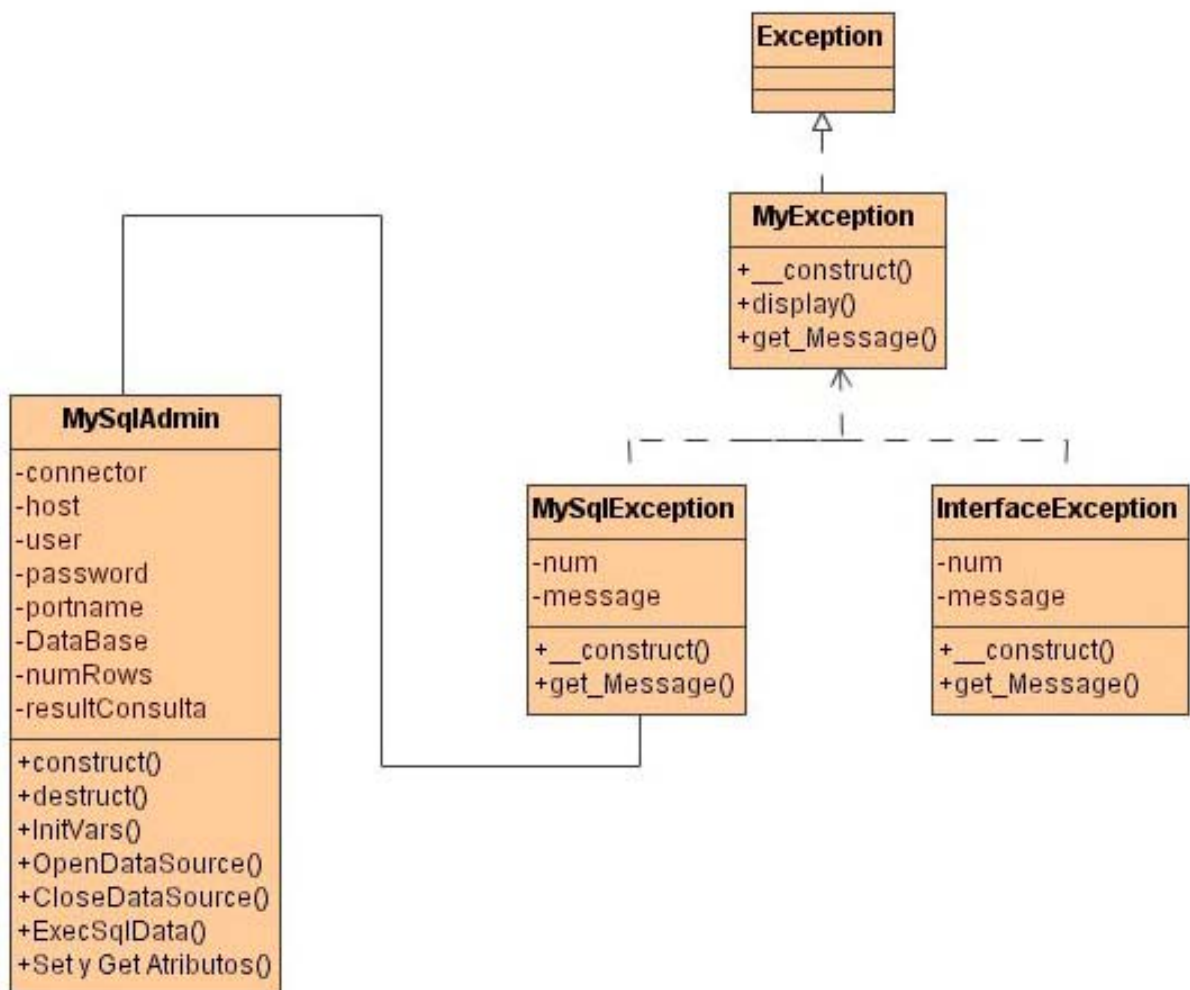
4.1.19.- Diagrama de secuencia de ir a la página número N

El usuario hace clic sobre el botón del número de la página en la que se quiere situar y el programa ejecuta la continuación del Listar (familias, modelos, artículos,...) dependiendo del mantenimiento en el que se encuentre el usuario. Se la pasa el número de la página a la cual tiene que ir y el listar se encarga de agrupar los productos a listar de 10 en 10.

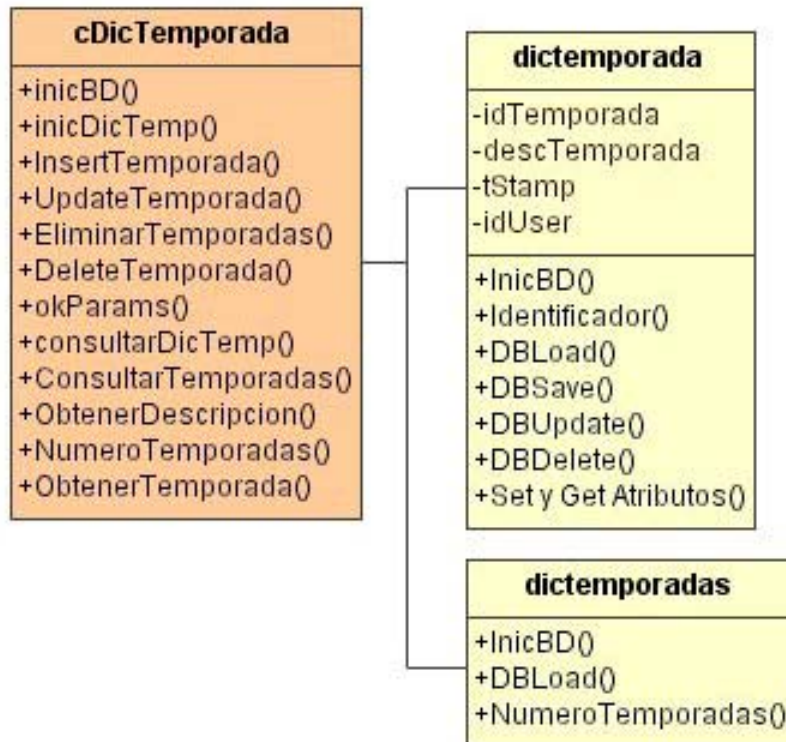
No muestro un diagrama de secuencia en concreto, porque este sería igual que el diagrama de listar alguno de los productos, excepto el inicio que sería el momento en el que el usuario hace clic en el número de la página a la que quiere ir.

4.2.2.- Diagrama de clases de MySqlAdmin y de las Excepciones

A continuación mostraré el diagrama de clases de las excepciones y de la clase de MySqlAdmin, esta clase es llamada (se crea una instancia de ella) desde todas las clases que están dibujadas de amarillo en el diagrama de clases. MySqlConnection es instanciada desde MySqlAdmin. InterfaceException es instanciada desde todas las clases restantes. Ambas Extends de MyException que a su vez Extends de la clase de php Exception la encargada de gestionar las excepciones.

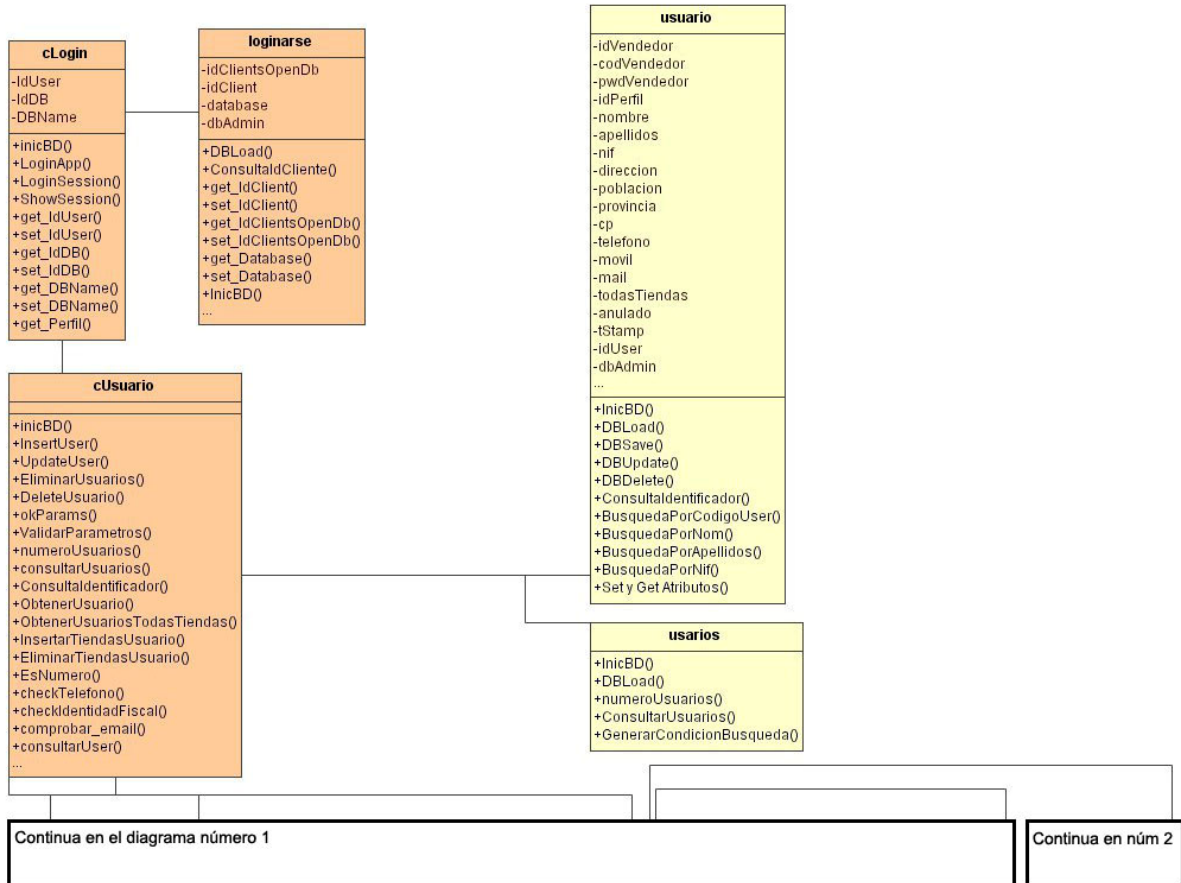


4.2.3.- Diagrama de clases específico de la clase de dicTemporada

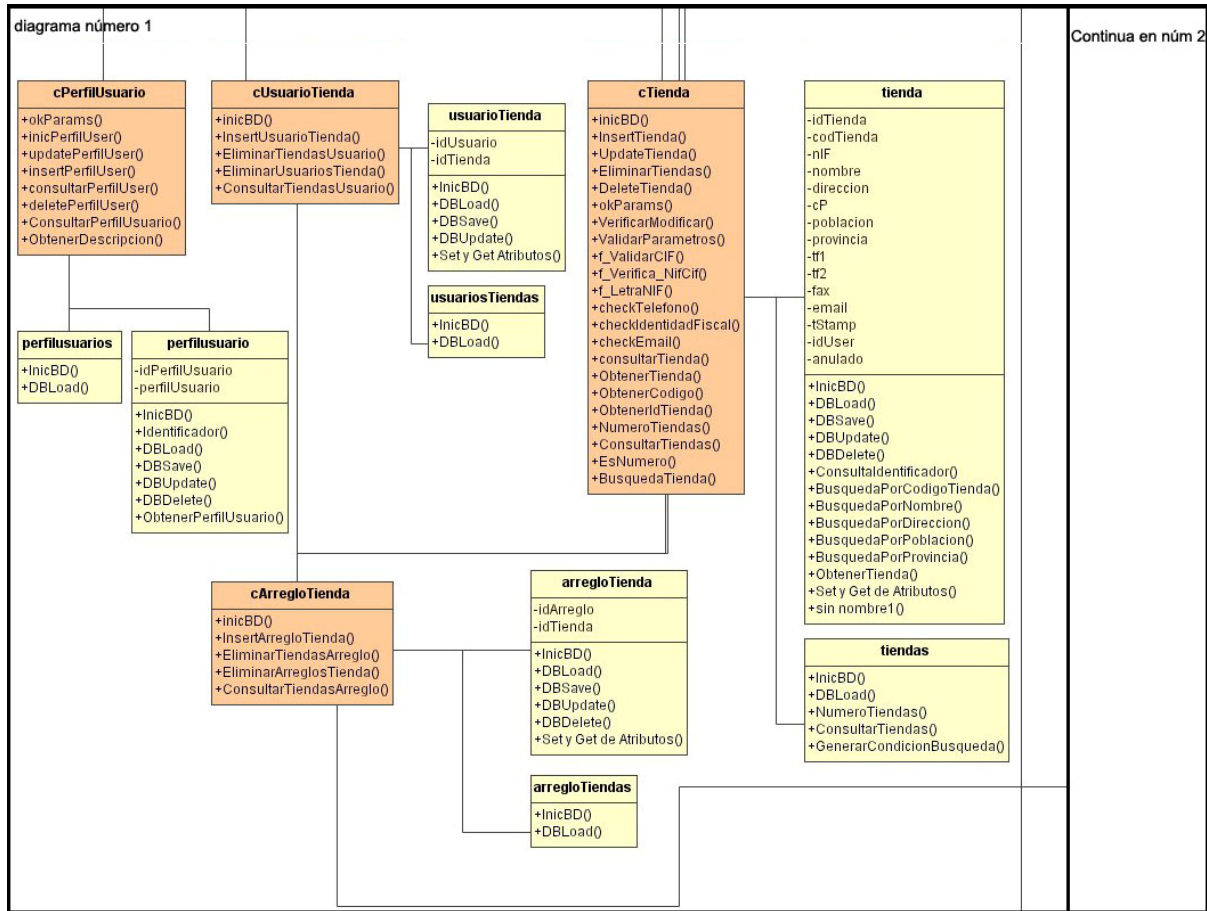


4.2.4.- Diagrama de clases específicos, diagrama número 0

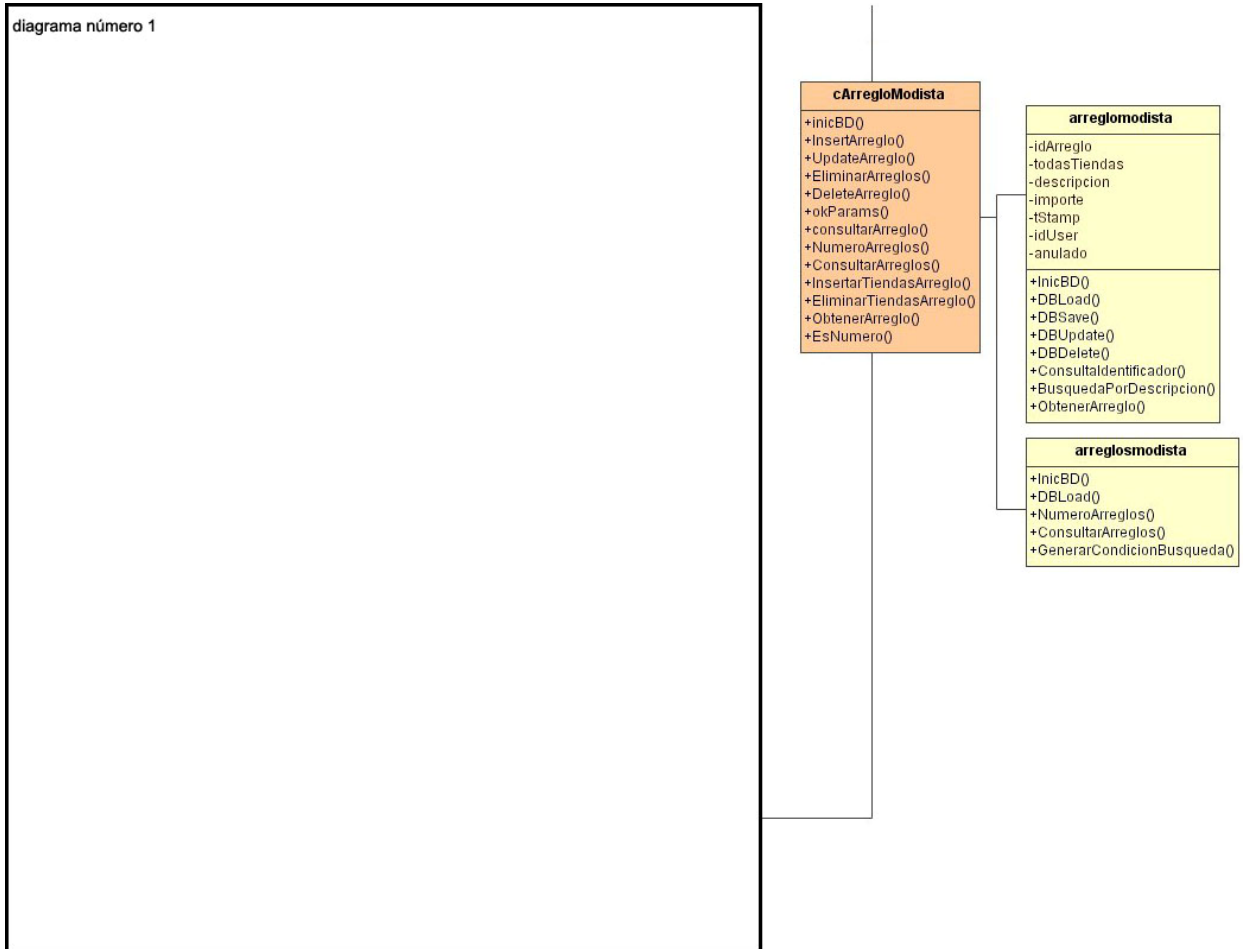
Diagrama número 0



4.2.5.- Diagrama de clases específicos, diagrama número 1

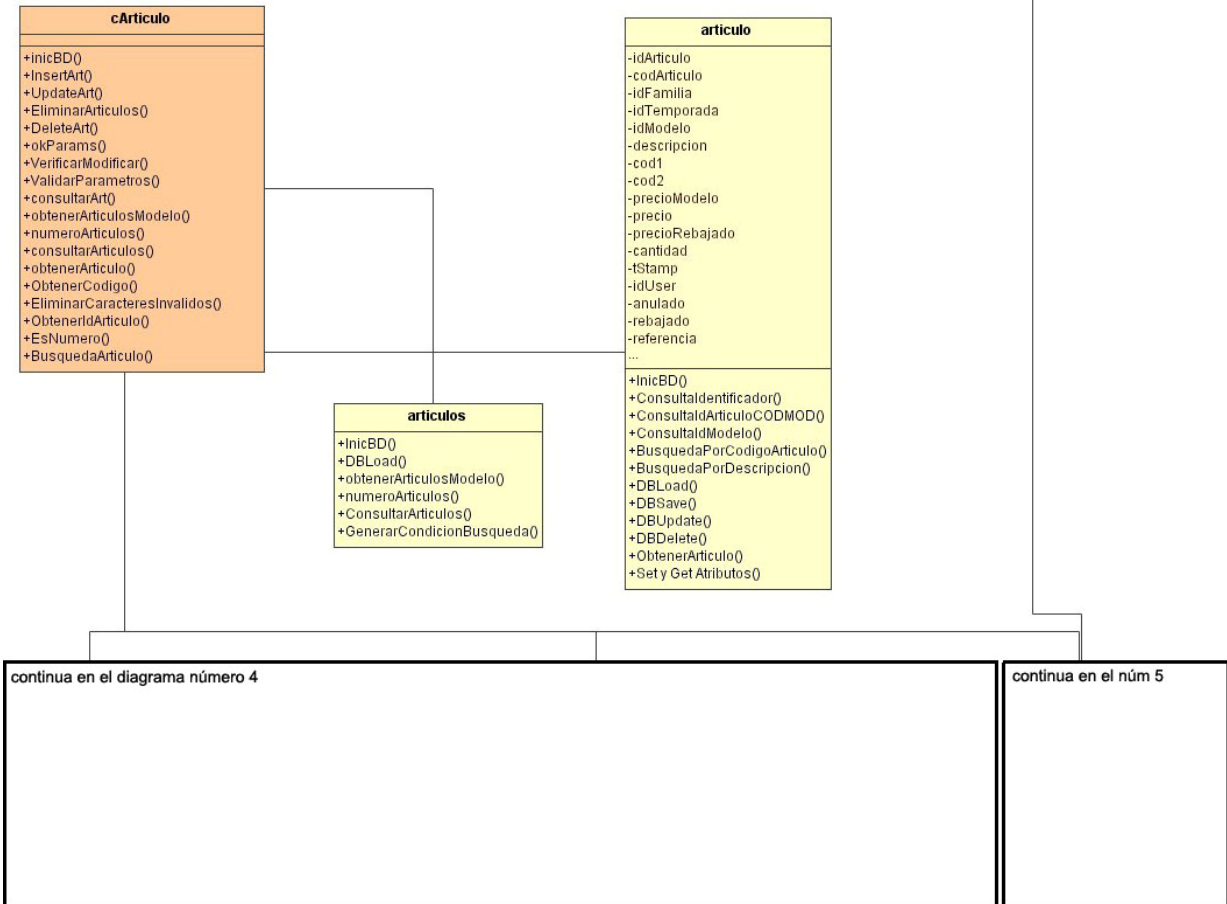


4.2.6.- Diagrama de clases específico, diagrama número 2



4.2.7.- Diagrama de clases específico, diagrama número 3

diagrama de clase número 3

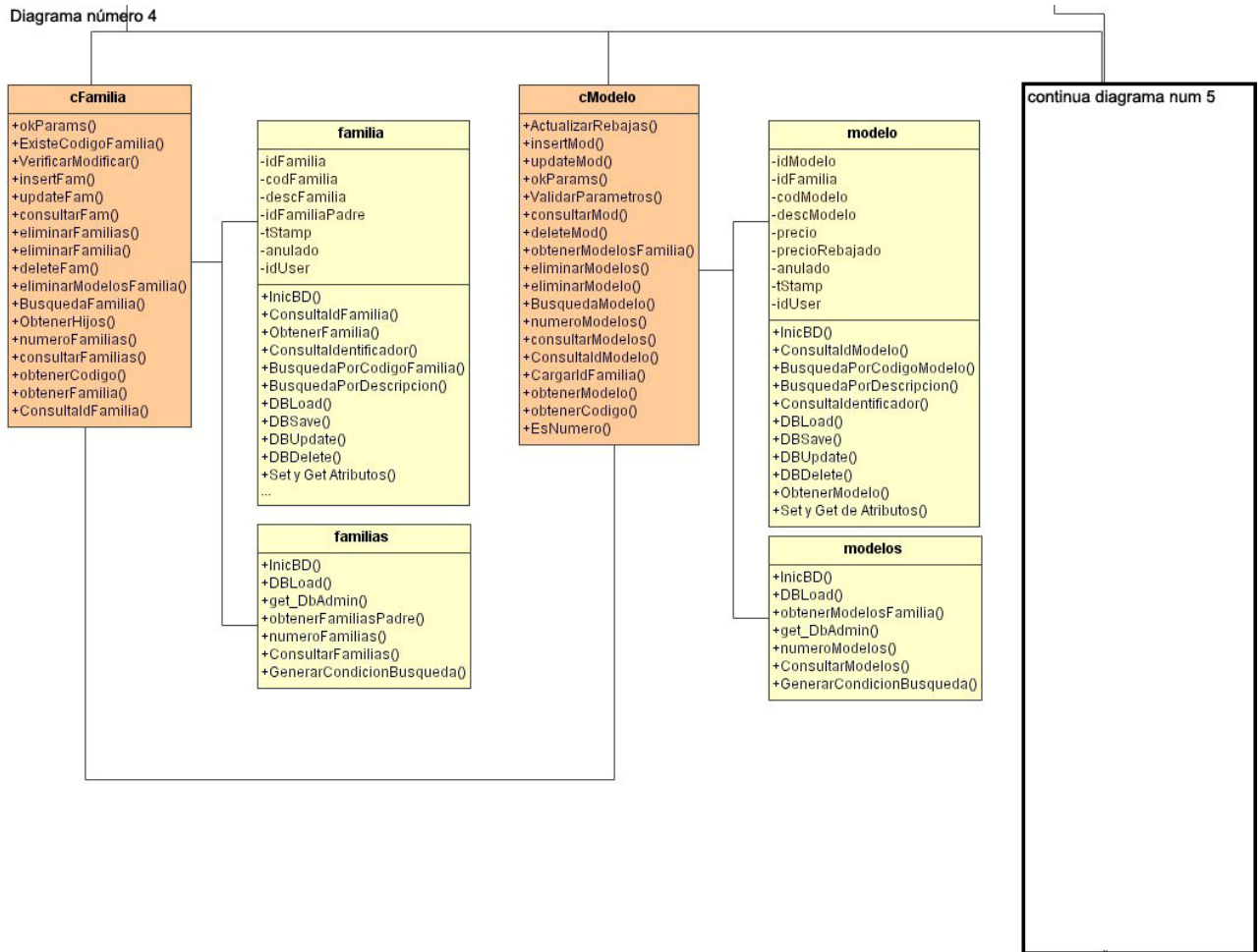


continua en el diagrama número 4

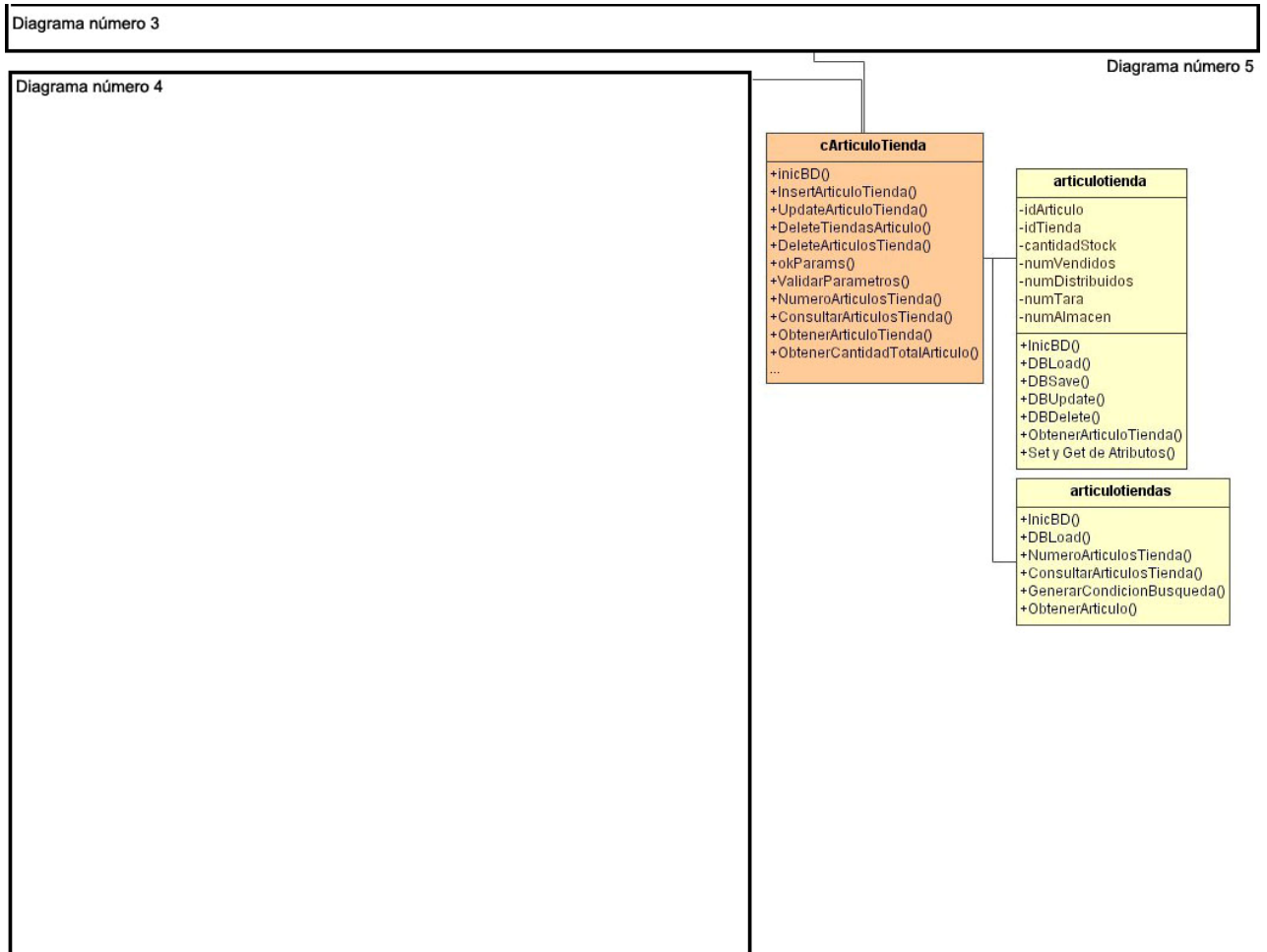
continua en el núm 5

4.2.8.- Diagrama de clases específico, diagrama número 4

Diagrama número 4



4.2.9.- Diagrama de clases específico, diagrama número 5



5.- DISEÑO

5.1.- Especificaciones de la aplicación del módulo de administración

Recordemos que la aplicación consta de dos módulos el de Administración y el de TPV, este proyecto se ha centrado en el módulo de Administración realizando todos los mantenimientos y gestiones que este necesitaba para poder llevarse a cabo.

A continuación mostraremos algunos de los requerimientos de este módulo que se han tenido que tener en cuenta para realizar el diseño.

5.1.1. Validación

Se deberá realizar un formulario web en el que el usuario introduzca el código de la compañía, el nombre del usuario que desea validar y la contraseña que este posee.

Una vez introducidos los datos la aplicación accederá a la base de datos centralizada smartShopLogindb donde se encuentran los datos de la empresa a la cual se desea acceder. La empresa tendrá su propia base de datos donde se encontraran todas las tablas que gestionarán su parte de administración.

Una vez accedido a la base de datos y obteniendo el nombre de la empresa, haremos una consulta a ésta para confirmar la validez del código de usuario y de la contraseña introducida por éste. La contraseña se encuentra encriptada por el método estándar de encriptación del Unix DES.

En la aplicación guardaremos los datos de la sesión por dos motivos:

- Histórico
- Control de sesión

Una vez realizada la validación se iniciará un control de sesión de usuarios realizada con php. Ésta puede ser a través de cookies o a través de variables de sesión.

Yo he utilizado variables de sesión ya que las cookies tienen el inconveniente de que algunos navegadores no las aceptan o que algunos usuarios las pueden tener desactivadas en sus navegadores.

Utilizaré las variables de sesión para guardar los siguientes datos:

- IdUsuario: identificador del usuario que se acaba de validar en nuestro sistema
- Nombre de la base de datos a la cual se conectará la aplicación para hacer todos los accesos.

- El id de conexión a las bases de datos para no tener que estar continuamente abriendo y cerrando el acceso.

De esta manera hasta que no haya una desconexión o un fin de sesión, el usuario no tendrá que volver a validarse.

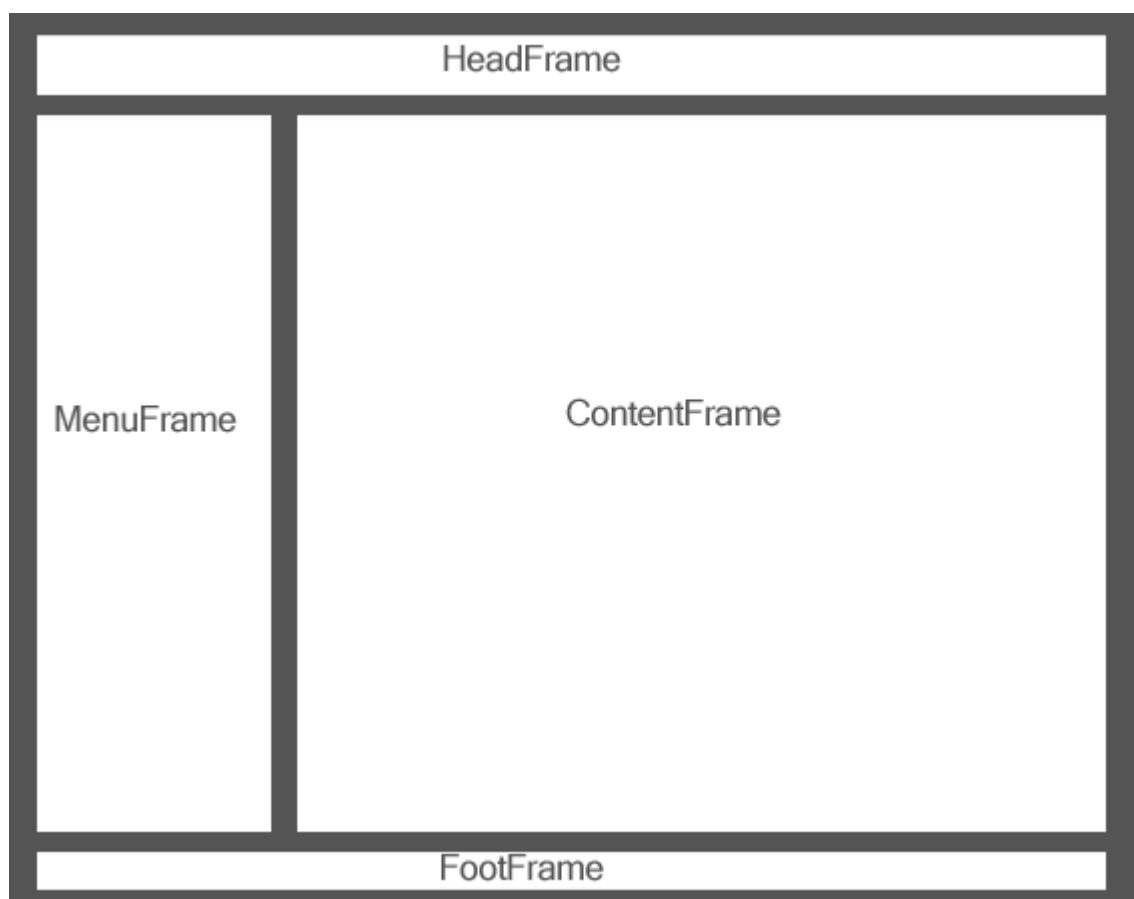
5.1.2.- Panel de control

El diseño del panel de control de la aplicación será el siguiente:

Se realizará una estructura de paneles para optimizar los refrescos de pantalla que se realicen al interactuar con los distintos formularios web.

Se realizará un panel de control parecido a los utilizados por los webmail, control de hostings,...

La estructura de frames será la siguiente:



El MenuFrame: Se encontrarán todos los mantenimientos y nos mostrará el mantenimiento actual, poniendo la imagen del mantenimiento de un color diferente al resto de mantenimientos.

El ContentFrame: Se encontrarán todos los datos del formulario seleccionado y al inicio de la aplicación nos mostrará con imágenes a estilo botones los diferentes formularios.

El FootFrame: que contendrá los botones de página inicio (volver al menú principal) y de salir.

El HeadFrame: que está destinado a insertar alguna imagen del empresa que ha adquirido el producto.

5.1.3.- Mantenimiento de Familias

Para el mantenimiento de familias se propone lo siguiente:

Realizar un formulario web en el que se podrán realizar:

- Búsquedas
- Inserciones
- Eliminaciones
- Actualizaciones

Los datos se listarán en agrupaciones de 10 registros con las familias que tengo dadas de alta en la base de datos y ordenadas por código de familia.

En el momento de realizar la inserción del nuevo dato en la base de datos, realizaré el siguiente diseño:

- Varios edits para introducir los nuevos datos
- Un árbol de búsqueda donde se podrá seleccionar cualquier nivel, para especificar el nivel al que pertenece la nueva familia insertada.

También realizaré las siguientes comprobaciones:

- Se hayan insertado los campos obligatorios
- Código Familia supere los 3 caracteres
- Que la descripción de la familia supere los 5 caracteres

En el momento de la eliminación de un campo, se mostrará un mensaje de confirmación y de los elementos que se eliminarán para continuar manteniendo la coherencia de los datos de la base de datos. Recordar que se ha programado para que se puedan borrar varios registros de datos a la vez, utilizando los checks que se encuentran en la parte izquierda del listado de los datos.

Se podrán realizar búsquedas por código de familia y de descripción de familia. Los datos listados hasta entonces se refrescarán mostrando los resultados de la búsqueda agrupados en páginas de 10 registros. La búsqueda se podrá realizar por patrones de caracteres contenidos en la palabra de búsqueda.

Las modificaciones de los datos se realizarán de aquellos registros deseados manteniendo algunos campos desactivados como el código de la familia.

Si se detecta algún error en algunas de las funcionalidades de este mantenimiento, se mostrará una pantalla de error con el mensaje correspondiente al error efectuado por parte del usuario. Una vez se acepte el error, el usuario podrá corregirlo y continuar con la operación que pretendía realizar.

Para poder aumentar la facilidad de uso de la aplicación, se permitirán al usuario jerarquizar las diferentes familias ubicándolas en diferentes niveles, y así aumentar la facilidad de búsqueda de estas en el momento de insertar modelos y artículos. Para poder realizar los diferentes niveles se hará a través del campo de familia padre.

5.1.4.- Mantenimiento de Modelos

Para el mantenimiento de modelos se propone lo siguiente:

Realizar un formulario web en el que se podrán realizar:

- Búsquedas
- Inserciones
- Eliminaciones
- Actualizaciones

Listaré los datos de los modelos contenidos en la base de datos en agrupaciones de 10 registros. Estos datos se mostrarán ordenados por código de modelo.

En el momento de realizar la inserción del nuevo dato en la base de datos, realizaré el siguiente diseño:

- Varios edits para introducir los nuevos datos
- Un árbol de búsqueda donde se podrá seleccionar el último nivel de la familia a la que queremos que pertenezca nuestro modelo.

También realizaré las siguientes comprobaciones:

- Que se hayan rellenado los campos obligatorios
 - Código de modelo
 - Descripción del modelo
 - Precio del modelo
- El código de modelo no deberá superar los 3 caracteres
- La descripción del modelo no supere los 5 caracteres
- Cada modelo solo podrá pertenecer a una familia.
- El descuento será un campo que no se guardará en la base de datos, pero nos permitirá realizar el cálculo del precio rebajado para este modelo.

- El formato del precio y del precio rebajado será un número con decimales separados por "." (punto)

En el momento de la eliminación de un campo, se mostrará un mensaje de confirmación y de los elementos que se eliminarán para continuar manteniendo la coherencia de los datos de la base de datos. Se ha programado para que se puedan borrar varios registros de datos a la vez, utilizando los checks que se encuentran en la parte izquierda del listado de los datos.

Se podrán realizar búsquedas por código de modelo y de descripción de modelo. Los datos listados hasta entonces se refrescarán mostrando los resultados de la búsqueda agrupados en páginas de 10 registros. La búsqueda se podrá realizar por patrones de caracteres contenidos en la palabra de búsqueda.

Las modificaciones de los datos se realizarán de aquellos registros deseados manteniendo algunos campos desactivados como el código de la modelo.

Si se detecta algún error en algunas de las funcionalidades de este mantenimiento, se mostrará una pantalla de error con el mensaje correspondiente al error efectuado por parte del usuario. Una vez se acepte el error, el usuario podrá corregirlo y continuar con la operación que pretendía realizar.

5.1.5.- Mantenimiento de Artículos

Para el mantenimiento de artículos se propone lo siguiente:

Realizar un formulario web en el que se podrán realizar:

- Búsquedas
- Inserciones
- Eliminaciones
- Actualizaciones

Listaré los datos de los artículos contenidos en la base de datos en agrupaciones de 10 registros. Estos datos se mostrarán ordenados por código de artículo.

En el momento de realizar la inserción del nuevo dato en la base de datos, realizaré el siguiente diseño:

- Varios edits para introducir los nuevos datos
- Un árbol para facilitar la inserción del modelo al que pertenece el artículo. Este árbol mostrará en el primer nivel el modelo y la familia en un nivel inferior y sangrado. Una vez seleccionado desde el árbol el modelo deseado, se rellenarán los campos en el formulario de alta.
- Un check que marcará que el artículo posee el mismo precio que su modelo, esto permitirá rellenar por defecto los campos de precio y de precio rebajado con los datos obtenidos del modelo, que se ha

seleccionado anteriormente. Para que este check esté habilitado se tiene que haber seleccionado un modelo.

- Un menú desplegable para seleccionar las temporadas a las que pueden estar ligados los artículos. Este menú desplegable está relleno con los datos que están dados de alta en el mantenimiento de Temporadas.
- Un campo de tipo texto, el cual se rellenará con la lectura del código de barras mediante un lector óptico.
- También realizaré las siguientes comprobaciones:
 - Que se hayan rellenado los campos obligatorios
 - Código de artículo
 - Descripción del artículo
 - Modelo con su familia correspondiente.
 - Precio del artículo
 - El código del artículo no supere los 3 caracteres
 - La descripción del artículo no supere los 5 caracteres
 - El descuento será un campo que no se guardará en la base de datos, pero nos permitirá realizar el cálculo del precio Rebajado para este modelo.
 - El formato del precio y del precio rebajado será un número decimal separado por "." (punto)

Una vez el usuario ha introducido los datos y acepta, se pasará a comprobar la validez de los datos y se dará de alta el artículo. En ese momento aparecerá un mensaje con el texto *"quiere agregar stock a una tienda"*. Si el usuario decide continuar, se mostrará un formulario para poder dar de alta el artículo en la tienda deseada, esto facilitará al usuario rellenar el mantenimiento de stock de las tiendas. El formulario es el mismo que el del mantenimiento de stock, pero con el campo de artículo ya relleno.

En el momento de la eliminación de un registro se mostrará un mensaje de confirmación con los elementos que se eliminarán para continuar manteniendo la coherencia de los datos. Se ha implementado para que se puedan borrar varios registros a la vez mediante los checks que se encuentran en la parte izquierda del listado de los datos.

Se podrán realizar búsquedas por código de artículo y descripción del artículo. Los datos listados hasta entonces se refrescarán mostrando los resultados de la búsqueda agrupados en páginas de 10 registros. La búsqueda se podrá realizar por patrones de caracteres contenidos en la palabra de búsqueda.

Las modificaciones de los datos se realizarán de aquellos registros deseados manteniendo algunos campos desactivados como el código del artículo.

Si se detecta algún error en algunas de las funcionalidades de este mantenimiento, se mostrará una pantalla de error con el mensaje correspondiente al error efectuado por parte del usuario. Una vez se acepte el error, el usuario podrá corregirlo y continuar con la operación que pretendía realizar.

5.1.6.- Mantenimiento de Tiendas

Para el mantenimiento de tiendas se propone lo siguiente:

Realizar un formulario web en el que se podrán realizar:

- Búsquedas
- Inserciones
- Eliminaciones
- Actualizaciones

Listaré los datos de las tiendas contenidas en la base de datos en agrupaciones de 10 registros. Estos datos se mostrarán ordenados por código de tienda.

En el momento de realizar la inserción del nuevo dato en la base de datos, realizaré el siguiente diseño:

- Varios edits para introducir los nuevos datos

También realizaré las siguientes comprobaciones:

- Que se hayan rellenado los campos obligatorios:
 - Código de tienda
 - NIF de la tienda
 - Dirección de la tienda
 - Código postal de la tienda
 - Población de la tienda
 - Provincia de la tienda
- El código tienda no debe superar los 3 caracteres
- Comprobación y validación el NIF
- Comprobación de que el código postal sea numérico
- Comprobación y validación del e.mail

Una vez se ha dado de alta la nueva tienda con los datos especificados, se hará una consulta a la tabla de vendedores. Todos aquellos vendedores que pertenezcan a todas las tiendas, se realizará un proceso para que estos también pertenezcan a la tienda acabada de introducir.

También se gestionará de la misma manera para los arreglos de modista. Si el arreglo de modista tiene activado el check de *"para todas las tiendas"*, al añadir la nueva tienda se añadirá la relación del identificador de la tienda con los arreglos de modista.

En el momento de la eliminación de un campo, se mostrará un mensaje de confirmación y de los elementos que se eliminarán para continuar manteniendo la coherencia de los datos. Se ha programado para que se puedan borrar varios

registros de datos a la vez, utilizando los checks que se encuentran en la parte izquierda del listado de los datos.

Se podrán realizar búsquedas por código de tienda, nombre de tienda y población. Los datos listados hasta entonces se refrescarán mostrando los resultados de la búsqueda agrupados en páginas de 10 registros. La búsqueda se podrá realizar por patrones de caracteres contenidos en la palabra de búsqueda.

Las modificaciones de los datos se realizarán de aquellos registros deseados manteniendo algunos campos desactivados como el código de tienda.

Si se detecta algún error en algunas de las funcionalidades de este mantenimiento, se mostrará una pantalla de error con el mensaje correspondiente al error efectuado por parte del usuario. Una vez se acepte el error, el usuario podrá corregirlo y continuar con la operación que pretendía realizar.

5.1.7.- Mantenimiento de Usuarios

Para el mantenimiento de usuarios se propone lo siguiente:

Realizar un formulario web en el que se podrán realizar:

- Búsquedas
- Inserciones
- Eliminaciones
- Actualizaciones

Listaré los datos de los usuarios contenidos en la base de datos en agrupaciones de 10 registros. Estos datos se mostrarán ordenados por código de usuario.

En el momento de realizar la inserción del nuevo dato en la base de datos, realizaré el siguiente diseño:

- Varios edits para introducir los nuevos datos
- Un combo Box para introducir el perfil de usuario

También realizaré las siguientes comprobaciones:

- Que se hayan insertado los campos obligatorios
 - Código de usuario
 - Nombre del usuario
- El código usuario no deberá superar los 3 caracteres
- El NIF deberá ser correcto.
- El código postal debe ser numérico
- Teléfono Fijo debe de estar compuesto por números
- Teléfono Móvil debe de estar compuesto por números
- El e.mail debe de ser correcto, con su @ y su dominio
- La contraseña para el usuario

- Si el campo "*todas las franquicias*" está chequeado, dará permisos al usuario para poder administrar todas las franquicias / tiendas, sino sólo tendrá privilegios para la franquicia / tienda a la cual esté vinculado. Si se quieren especificar una serie de franquicias para ese usuario, al desmarcar el check, el campo mostrará una lista con las franquicias actuales para seleccionar las deseadas.

En el momento de la eliminación de un registro se mostrará un mensaje de confirmación con los elementos que se eliminarán para continuar manteniendo la coherencia de los datos. Se ha implementado para que se puedan borrar varios registros a la vez mediante los checks que se encuentran en la parte izquierda del listado de los datos.

Se podrán realizar búsquedas por código de usuario y nombre de usuario. Los datos listados hasta entonces se refrescarán mostrando los resultados de la búsqueda agrupados en páginas de 10 registros. La búsqueda se podrá realizar por patrones de caracteres contenidos en la palabra de búsqueda.

Las modificaciones de los datos se realizarán de aquellos registros deseados manteniendo algunos campos desactivados como el código de usuario.

Si se detecta algún error en algunas de las funcionalidades de este mantenimiento, se mostrará una pantalla de error con el mensaje correspondiente al error efectuado por parte del usuario. Una vez se acepte el error, el usuario podrá corregirlo y continuar con la operación que pretendía realizar.

5.1.8.- Mantenimiento de temporadas

Este es un mantenimiento simple que nos servirá para poder rellenar la temporada en los artículos. Para el mantenimiento de temporadas se propone lo siguiente:

Realizar un formulario web en el que se podrán realizar:

- Inserciones
- Eliminaciones
- Actualizaciones

Listaré los datos de las temporadas contenidos en la base de datos en agrupaciones de 10 registros. Estos datos se mostrarán ordenados por descripción de temporada.

En el momento de realizar la inserción de la nueva temporada en la base de datos, realizaré el siguiente diseño:

- Varios edits para introducir los nuevos datos

En el momento de la eliminación de un registro se mostrará un mensaje de confirmación con los elementos que se eliminarán para continuar manteniendo la coherencia de los datos. Se ha implementado para que se puedan borrar varios registros a la vez mediante los checks que se encuentran en la parte izquierda del listado de los datos.

Las modificaciones de los datos se realizarán de aquellos registros deseados manteniendo algunos campos desactivados como el código de temporada.

Las modificaciones de los datos se realizarán de aquellos registros deseados.

Si se detecta algún error en algunas de las funcionalidades de este mantenimiento, se mostrará una pantalla de error con el mensaje correspondiente al error efectuado por parte del usuario. Una vez se acepte el error, el usuario podrá corregirlo y continuar con la operación que pretendía realizar.

5.1.9- Mantenimiento de Arreglos modista

Para el mantenimiento de Arreglos de modista se propone lo siguiente:

Realizar un formulario web en el que se podrán realizar:

- Búsquedas
- Inserciones
- Eliminaciones
- Actualizaciones

Listaré los datos de los arreglos de modista contenidos en la base de datos en agrupaciones de 10 registros. Estos datos se mostrarán ordenados por descripción del arreglo.

En el momento de realizar la inserción del nuevo dato en la base de datos, realizaré el siguiente diseño:

- Varios edits para introducir los nuevos datos
- Check para vincular las tiendas que se incluirá este arreglo de modista. El check de todas las tiendas vendrá marcado por defecto. Al desmarcarse, aparecerá un listado con todas las tiendas y de esta lista se podrán elegir todas las tiendas a las que se quiere relacionar este arreglo de modista. Entiendo que dependiendo de la zona donde nos encontremos las modistas cobran unos precios dependiendo de sus tarifas es por eso, que debemos ser capaces de diferenciar entre diferentes tiendas.

También realizaré las siguientes comprobaciones:

- Que se hayan rellenado los campos obligatorios
 - Descripción de los arreglos de modista.

- No se puede añadir un arreglo de modistas sin haber insertado, como mínimo, una tienda.
- La descripción debe superar los 5 caracteres
- El formato del precio será un número con decimales separados por "." (punto)

En el momento de la eliminación de un registro se mostrará un mensaje de confirmación con los elementos que se eliminarán para continuar manteniendo la coherencia de los datos.

Se podrán realizar búsquedas por la descripción del arreglo. Los datos listados hasta entonces se refrescarán mostrando los resultados de la búsqueda agrupados en páginas de 10 registros. La búsqueda se podrá realizar por patrones de caracteres contenidos en la palabra de búsqueda.

Las modificaciones de los datos se realizarán de aquellos registros deseados.

Si se detecta algún error en algunas de las funcionalidades de este mantenimiento, se mostrará una pantalla de error con el mensaje correspondiente al error efectuado por parte del usuario. Una vez se acepte el error, el usuario podrá corregirlo y continuar con la operación que pretendía realizar.

5.1.10.- Mantenimiento de Stock

Recordemos que el stock también lo podíamos gestionar en el momento de dar de alta un artículo. De esta manera evitamos tener que estar listando todos los artículos para tener que repartirlos por las franquicias. Al insertar un artículo nuevo, en el mantenimiento de artículos, aparecerá el mensaje de añadir al stock, si el usuario acepta aparecerá el mantenimiento de alta de stock pero con la diferencia de que el código de artículo aparecerá ya rellenado. Se gestionará de la misma manera que el mantenimiento de dar de alta stock.

Para el mantenimiento de stock se propone lo siguiente:

Realizar un formulario web en el que se podrán realizar:

- Búsquedas
- Inserciones
- Eliminaciones
- Actualizaciones

Listaré los datos de stock contenidos en la base de datos en agrupaciones de 10 registros. Estos datos se mostrarán ordenados por código de tienda.

En el momento de realizar la inserción del nuevo dato en la base de datos, realizaré el siguiente diseño:

- Varios edits para introducir los nuevos datos
- Un diálogo donde aparecerán los artículos con el modelo al que pertenecen, y así ayudar a la selección del artículo.
- Un diálogo donde aparecerá el código de la tienda con una pequeña descripción del nombre de la tienda y la población donde se encuentra.

También realizaré las siguientes comprobaciones:

- Se hayan rellenado los campos obligatorios
 - Código de artículo
 - Código de la tienda
- El código de artículo exista
- El código de la tienda exista
- Que el stock total de la tienda no supere la cantidad total de artículos
- El número de artículos vendidos, distribuidos, almacenados y con tara no deben superar el total de artículos de la tienda.
- El número de artículos debe ser un número entero.

En el momento de la eliminación de un registro se mostrará un mensaje de confirmación con los elementos que se eliminarán para continuar manteniendo la coherencia de los datos.

Se podrán realizar búsquedas por código de artículo y por código de tienda. Los datos listados hasta entonces se refrescarán mostrando los resultados de la búsqueda agrupados en páginas de 10 registros. La búsqueda se podrá realizar por patrones de caracteres contenidos en la palabra de búsqueda.

Las modificaciones de los datos se realizarán de aquellos registros deseados manteniendo el campo de código de artículo y de código de tienda desactivados.

Si se detecta algún error en algunas de las funcionalidades de este mantenimiento se mostrará una pantalla de error, con el mensaje correspondiente al error efectuado por parte del usuario. Una vez se acepte el error, el usuario podrá corregirlo y continuar con la operación que pretendía realizar.

Para el módulo de administración, este formulario sólo se encargará de repartir los artículos de la fábrica a las diferentes tiendas. Una vez en la tienda, el vendedor será en encargado de administrar los artículos en su lugar (almacén, distribuido, con tara o vendidos).

5.2.2.- Definición de las tablas

ClientsOpendb	
Es la tabla que se encarga de guardar el nombre de la base de datos donde se guardan el resto de las tablas con los datos dados de alta para poder gestionar la empresa.	
IdClientsOpenDb	Clave primaria, es una auto numérico que es el identificador de la tabla.
IdClient	Es un número que es el código que inserta el usuario cuando intenta validarse al entrar en la aplicación. Con este número sabremos a que base de datos tenemos que ir a gestionar los datos.
Database	Es el nombre de la base de datos a la cual accederemos para confirmar la validación del usuario.

Una vez hemos sabemos el nombre de la base de datos a la cual tenemos que acceder nos encontramos con todas las tablas que gestionan los datos de los productos de la empresa.

Familias	
Es la tabla que se encarga de guardar todas las familias de los productos que daremos de alta.	
IdFamilia	Clave primaria, es una auto numérico que es el identificador de la tabla.
CodFamilia	Es el código de la familia con la que identificaremos los productos
DescFamilia	Descripción de la familia que vamos a dar de alta
IdFamiliaPadre	Identificador para poder jerarquizar las familias, en varios niveles
Anulado	Nos sirve para anular la familia sin tener que borrarla, es para no tener que borrar familias y todos las eliminaciones en cascada que esto conlleva
Tstamp	Nos servirá para guardar la fecha de alta de la familia
IdUser	Nos indica el identificador de usuario que dio de alta la familia

Modelos	
Es la tabla que se encarga de guardar todos los modelos de los productos.	
IdModelo	Clave primaria, es una auto numérico que es el identificador de la tabla.
IdFamilia	Identificador de la familia a la que pertenece el modelo
CodModelo	Código del modelo con el que se identificará el producto
DescModelo	Descripción del modelo que vamos a dar de alta.
Precio	Precio fijado para este modelo
PrecioRebajado	Precio rebajado del modelo
Anualdo	Nos sirve para anular el modelo sin tener que borrarlo, es para no tener que borrar modelos y todas las eliminaciones en cascada que esto conlleva
Tstamp	Nos servirá para guardar la fecha de alta del modelo
IdUser	Nos indica el identificador de usuario que dio de alta el modelo

Artículos	
Es la tabla que se encarga de guardar todos los artículos que gestiona la empresa.	
IdArticulo	Clave primaria, es una auto numérico que es el identificador de la tabla.
IdFamilia	Identificador de la familia a la que pertenece el artículo
IdModelo	Identificador del modelo al que pertenece el artículo. Cada artículo pertenece a un modelo y un modelo pertenece a una familia, el idFamilia no sería necesario ya que podríamos encontrar la familia haciendo una búsqueda en modelos, pero lo guardamos para poder hacer la gestión más rápidamente.
CodArticulo	Código del artículo con el que identificaremos el producto
IdTemporada	Identificador con la temporada a la que pertenece el artículo
Descripcion	Descripción de artículo que vamos a dar de alta.
Referencia	Código de barras con el que identificaremos el artículo
Cod1	En este código guardaremos la talla del artículo
Cod2	En este código guardaremos el color del artículo
Cod3	No gestionado
Cod4	No gestionado
Cod5	No gestionado
Cod6	No gestionado
Cod7	No gestionado
PrecioModelo	Nos indicará que el precio del artículo es el mismo que el del modelo
Precio	Se guardará el precio del artículo
PrecioRebajado	Guardará el precio rebajado del artículo
Rebajado	Check que nos indica si el artículo se encuentra actualmente rebajado
Cantidad	Cantidad total de artículos
Anualdo	Nos sirve para anular el artículo sin tener que borrarlo, es para no tener que borrar artículos y todas las eliminaciones en cascada que esto conlleva
Tstamp	Nos servirá para guardar la fecha de alta del artículo
IdUser	Nos indica el identificador de usuario que dio de alta el artículo

Tiendas	
Es la tabla donde se guardan las tiendas/franquicias de la empresa	
IdTienda	Clave primaria, es una auto numérico que es el identificador de la tabla.
CodTienda	Código de la tienda con la que la identificaremos
NIF	Nif de la tienda
Nombre	Nombre de la tienda
Direccion	Dirección en la que se encuentra la tienda
CP	Código postal
Poblacion	Población en la que se encuentra la tienda
Provincia	Provincia en la que se encuentra la tienda
Tf1	Teléfono de la tienda
Tf2	Teléfono secundario de la tienda
Fax	Número de fax de la tienda
Email	Email de contacto de la tienda
Anulado	Nos sirve para anular la tienda sin tener que darla de baja en la base de datos y con esto todos los deletes que conlleva.
Tstamp	Nos servirá para guardar la fecha de alta la tienda
IdUser	Nos indica el identificador de usuario que dio de alta la tienda

Usuarios	
Es la tabla donde se guardan todos los usuarios/vendedores que se encuentran en plantilla o que han formado parte de la plantilla de la empresa.	
IdVendedor	Clave primaria, es una auto numérico que es el identificador de la tabla.
CodVendedor	Código del vendedor con el que lo identificamos
PwdVendedor	Contraseña que asignamos al vendedor para poder entrar en la aplicación
IdPerfil	Identificación del perfil de usuario para poder gestionar los permisos que tendrá el usuario
Nombre	Nombre del vendedor
Apellidos	Apellidos del vendedor
NIF	DNI del vendedor
Direccion	Dirección donde vive el vendedor
Población	Población en la que vive el vendedor
Provincia	Provincia en la que vive el vendedor
CP	Código postal donde vive el vendedor
Telefono	Teléfono fijo del vendedor
Movil	Teléfono móvil del vendedor
Mail	Email de contacto del vendedor
TodasTiendas	Marca que nos indica si el vendedor tiene permiso para acceder a los datos de todas las tiendas
Anulado	Nos sirve para anular al vendedor sin tener que darla de baja en la base de datos y con esto todas las eliminaciones que conlleva.
Tstamp	Nos servirá para guardar la fecha de alta/modificación del vendedor
IdUser	Nos indica el identificador de usuario que dio de alta al vendedor

Arreglosmodista	
Es la tabla donde se guardan todos los arreglos que se ofrecen en las tiendas o franquicias.	
IdArreglo	Clave primaria, es una auto numérico que es el identificador de la tabla.
TodasTiendas	Marca que nos indica si el arreglo es para todas las tiendas
Descripcion	Descripción del arreglo
Importe	Precio del arreglo, según la tienda
Anulado	Nos sirve para anular el arreglo sin tener que darla de baja en la base de datos.
Tstamp	Nos servirá para guardar la fecha de alta/modificación del arreglo
IdUser	Nos indica el identificador de usuario que dio de alta el arreglo

DicTemporada	
Es la tabla donde se guardan todos los arreglos que se ofrecen en las tiendas o franquicias.	
IdTemporada	Clave primaria, es una auto numérico que es el identificador de la tabla.
DescTemporada	Descripción de la temporada
Anulado	Nos sirve para anular la temporada sin tener que darla de baja en la base de datos.
Tstamp	Nos servirá para guardar la fecha de alta/modificación de la temporada
IdUser	Nos indica el identificador del usuario que dio de alta la temporada

PerfilUsuarios	
Es la tabla donde se guardan todos los arreglos que se ofrecen en las tiendas o franquicias.	
IdPerfilUsuario	Clave primaria, es una auto numérico que es el identificador de la tabla.
PerfilUsuario	Nombre que identifique el perfil de usuario y de los permisos que éste tendrá

arreglostienda	
<p>Tabla que nos permitirá relacionar las tiendas con los arreglos de modista.</p> <p>Esta tabla es importante porque dependiendo de la provincia o de la modista encargada de realizar un arreglo los precios de estos pueden ser diferentes y es por esto que se necesita una tabla para poder gestionar el arreglo y la tienda a la que pertenece.</p>	
IdArreglo	Identificador del arreglo con el que se quiere relacionar la tienda
IdTienda	Identificador de la tienda para el que se guarda el arreglo

articuloTienda	
<p>Tabla que nos permitirá relacionar las tiendas con los artículos.</p> <p>Esta tabla es la que nos permitirá gestionar el stock de las tiendas. En esta guardaremos el artículo que se encuentra en esta tienda en concreto con una cantidad de stock. Una vez nos encontremos en el módulo de tpv, a través de esta tabla gestionaremos el número de artículos vendidos, el numero de artículos distribuidos, los artículos devueltos con tara y los artículos que tenemos en almacén.</p>	
IdArticulo	Identificador del artículo que relacionamos con la tienda
IdTienda	Identificador de la tienda en la que se encuentra este artículo
CantidadStock	Cantidad total que se encuentra en la tienda
NumVendidos	Número de artículos que hemos vendido desde esta tienda
NumDistribuidos	Número de artículos que he distribuido desde la tienda
NumTara	Número de artículos que hemos catalogado como tarados
NumAlmacen	Número de artículos que tenemos guardados en el almacén

usuarioTienda	
<p>Tabla que nos permitirá relacionar las tiendas con los vendedores.</p> <p>Esta tabla es la que nos permitirá gestionar los permisos que tienen los diferentes vendedores para acceder a las diferentes tiendas/franquicias, ya que es posible que nosotros queramos tener un usuario que pueda gestionar todas las tiendas o tener otros usuarios que solo puedan gestionar la tienda/franquicia en la que trabajan.</p>	
IdUsuario	Identificador del usuario que relacionamos con la tienda
IdTienda	Identificador de la tienda en la que se encuentra trabajando el vendedor

5.2.3.- Script de creación

Familias	
	<pre>CREATE TABLE `familias` (`IdFamilia` int(7) NOT NULL auto_increment, `CodFamilia` varchar(50) default NULL, `DescFamilia` varchar(150) default NULL, `IdFamiliaPadre` int(7) unsigned default NULL, `Anulado` tinyint(1) default '0', `TStamp` datetime default '0000-00-00 00:00:00', `IdUser` int(4) default '0', PRIMARY KEY (`IdFamilia`)) TYPE=MyISAM;</pre>
Modelos	
	<pre>CREATE TABLE `modelos` (`IdModelo` int(7) NOT NULL auto_increment, `IdFamilia` int(7) default '0', `CodModelo` varchar(10) default NULL, `DescModelo` varchar(150) default NULL, `Precio` float default '0', `PrecioRebajado` float default '0', `Anulado` tinyint(1) default '0', `TStamp` datetime default '0000-00-00 00:00:00', `IdUser` int(4) default '0', PRIMARY KEY (`IdModelo`)) TYPE=MyISAM;</pre>
Articulos	
	<pre>CREATE TABLE `articulos` (`IdArticulo` int(7) NOT NULL auto_increment, `IdFamilia` int(7) NOT NULL default '0', `IdModelo` int(7) NOT NULL default '0', `CodArticulo` varchar(20) NOT NULL default '', `IdTemporada` int(4) unsigned default '0', `Descripcion` varchar(150) default NULL, `Referencia` varchar(100) default NULL, `Cod1` varchar(50) default NULL, `Cod2` varchar(50) default NULL, `Cod3` varchar(50) default NULL, `Cod4` varchar(50) default NULL, `Cod5` varchar(50) default NULL, `Cod6` varchar(50) default NULL, `Cod7` varchar(50) default NULL, `PrecioModelo` tinyint(1) default '0', `Precio` float default '0', `PrecioRebajado` float default '0', `Rebajado` tinyint(1) default '0', `Cantidad` int(4) default '0', `Anulado` tinyint(1) default '0', `TStamp` datetime default '0000-00-00 00:00:00', `IdUser` int(4) default '0', PRIMARY KEY (`IdArticulo`)) TYPE=MyISAM;</pre>

Tiendas	
<pre>CREATE TABLE `tiendas` (`IdTienda` int(7) NOT NULL auto_increment, `CodTienda` varchar(50) default NULL, `NIF` varchar(15) default NULL, `Nombre` varchar(100) default NULL, `Direccion` varchar(200) default NULL, `CP` varchar(10) default NULL, `Poblacion` varchar(100) default NULL, `Provincia` varchar(100) default NULL, `Tf1` varchar(25) default NULL, `Tf2` varchar(25) default NULL, `Fax` varchar(25) default NULL, `Email` varchar(100) default NULL, `Anulado` tinyint(1) default '0', `TStamp` datetime default '0000-00-00 00:00:00', `idUser` int(4) default '0', PRIMARY KEY (`IdTienda`)) TYPE=MyISAM;</pre>	
Usuarios	
<pre>CREATE TABLE `usuarios` (`IdVendedor` int(7) NOT NULL auto_increment, `CodVendedor` varchar(50) NOT NULL default "", `PwdVendedor` varchar(50) default NULL, `IdPerfil` int(7) default '0', `Nombre` varchar(100) default NULL, `Apellidos` varchar(100) default NULL, `Nif` varchar(9) default NULL, `Direccion` varchar(200) default NULL, `Poblacion` varchar(100) default NULL, `Provincia` varchar(100) default NULL, `CP` varchar(5) default NULL, `Telefono` varchar(20) default NULL, `Movil` varchar(20) default NULL, `Mail` varchar(100) default NULL, `TodasTiendas` tinyint(1) default '1', `Anulado` tinyint(1) default '0', `TStamp` datetime default '0000-00-00 00:00:00', `idUser` int(4) default '0', PRIMARY KEY (`IdVendedor`)) TYPE=MyISAM;</pre>	
ArreglosModista	
<pre>CREATE TABLE `arreglosmodista` (`IdArreglo` int(7) NOT NULL auto_increment, `TodasTiendas` tinyint(1) default '0', `Descripcion` varchar(100) default NULL, `Importe` float default '0', `Anulado` tinyint(1) default '0', `TStamp` datetime default '0000-00-00 00:00:00', `idUser` int(4) default NULL, PRIMARY KEY (`IdArreglo`)) TYPE=MyISAM;</pre>	
dictemporada	
<pre>CREATE TABLE `dictemporada` (`IdTemporada` int(7) NOT NULL auto_increment, `DescTemporada` varchar(50) default '0', `Anulado` tinyint(1) default '0',</pre>	

	<pre> `TStamp` datetime default '0000-00-00 00:00:00', `idUser` int(4) default '0', PRIMARY KEY (`IdTemporada`)) TYPE=MyISAM; </pre>
perfilUsuario	
	<pre> CREATE TABLE `perfilusuarios` (`IdPerfilUsuario` int(7) NOT NULL auto_increment, `PerfilUsuario` varchar(100) NOT NULL default '', PRIMARY KEY (`IdPerfilUsuario`)) TYPE=MyISAM; </pre>
arreglostienda	
	<pre> CREATE TABLE `arreglostienda` (`IdArreglo` int(7) NOT NULL default '0', `IdTienda` int(7) NOT NULL default '0', PRIMARY KEY (`IdArreglo`,`IdTienda`)) TYPE=MyISAM; </pre>
articulotienda	
	<pre> CREATE TABLE `articulotienda` (`IdArticulo` int(7) unsigned NOT NULL default '0', `IdTienda` int(7) unsigned NOT NULL default '0', `CantidadStock` int(4) unsigned default '0', `NumVendidos` int(4) unsigned default '0', `NumDistribuidos` int(4) unsigned default '0', `NumTara` int(4) unsigned default '0', `NumAlmacen` int(4) unsigned default '0', PRIMARY KEY (`IdArticulo`,`IdTienda`)) TYPE=MyISAM; </pre>
usuariotienda	
	<pre> CREATE TABLE `usuariotienda` (`IdUsuario` int(7) unsigned NOT NULL default '0', `IdTienda` int(7) unsigned NOT NULL default '0', PRIMARY KEY (`IdUsuario`,`IdTienda`)) TYPE=MyISAM; </pre>

clientsopendb	
	<pre> CREATE TABLE clientsopendb (IdClientsOpenDb int(7) NOT NULL auto_increment, IdClient int(7) NOT NULL default '0', Database varchar(50) NOT NULL default '', PRIMARY KEY (IdClientsOpenDb), UNIQUE KEY NewIndex (IdClient)) TYPE=MyISAM; </pre>

En la tabla de perfilusuario además de la creación de la tabla viene una inserción con los dos campos mínimos de tipos de usuarios.

perfilUsuario

<pre>INSERT INTO `perfilusuarios` (`IdPerfilUsuario`, `PerfilUsuario`) VALUES("0", "Administrador"); INSERT INTO `perfilusuarios` (`IdPerfilUsuario`, `PerfilUsuario`) VALUES("1", "Usuario");</pre>
--

En la tabla de usuarios también se inserta un usuario por defecto para poder entrar en la aplicación para dar de alta al resto de usuarios con los perfiles correspondientes.

usuarios

<pre>INSERT INTO `usuarios` (`IdVendedor`, `CodVendedor`, `PwdVendedor`, `IdPerfil`, `Nombre`, `Apellidos`, `Nif`, `Direccion`, `Poblacion`, `Provincia`, `CP`, `Telefono`, `Movil`, `Mail`, `TodasTiendas`, `Anulado`, `TStamp`, `idUser`) VALUES("1", "marina", " xx4B01VraBHdQ", "0", "marina", "", "", "", "", "", "", "", "", "", "", "0", "0", "2006-11-11 20:51:52", "1");</pre>

6.- Manual de usuario

6.1.- Instalación de la aplicación

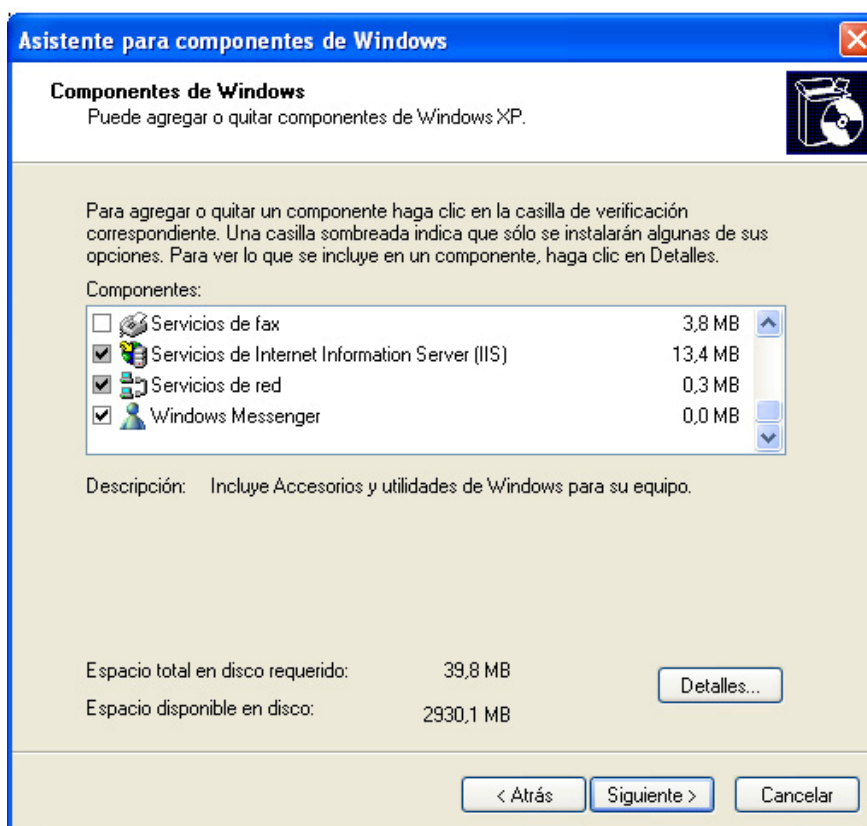
A continuación explicaré los pasos a seguir para la instalación de los diferentes componentes que permitirán el correcto funcionamiento de la aplicación.

Todos estos pasos hacen referencia a la configuración que debe tener el ordenador que realizará las funciones de servidor, ya que los clientes sólo tienen que tener instalado el IIS para poder acceder a la aplicación a través de internet.

Pasos a realizar:

En primer lugar verificar que la componente IIS se encuentra instalada en el sistema operativo Windows del servidor. En caso contrario, se debe añadir la componente de la siguiente manera:

1. Panel de control
2. Agregar o quitar programas
3. Agregar o quitar componentes de Windows
4. Seleccionar la componente IIS



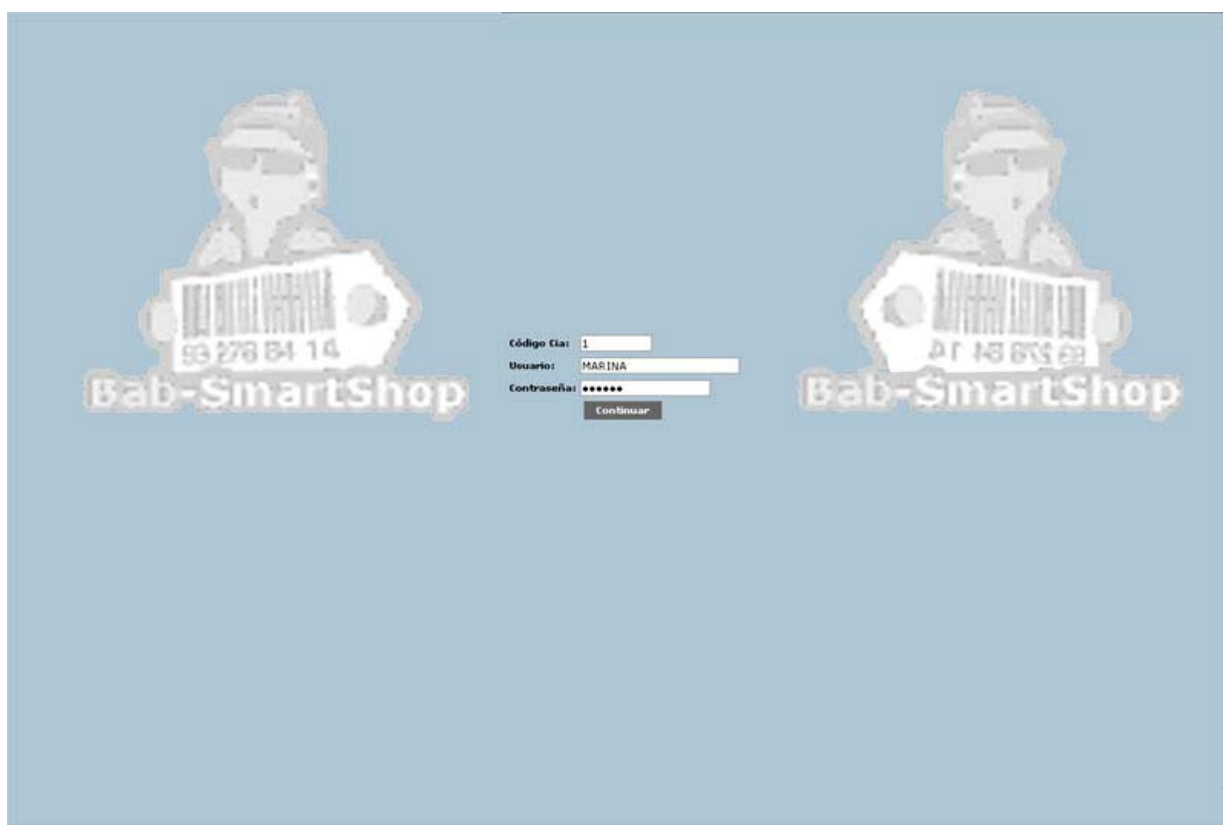
En segundo lugar, pasaríamos a instalar el SGBD MySQL que nos permitirá almacenar los datos de nuestra aplicación. No tendrá mayor dificultad que seguir los pasos de instalación del ejecutable que proporciona gratuitamente el grupo MySQL. <http://www.mysql.com/>

En tercer lugar se debe instalar y configurar el PHP 5.0. Para dicha instalación seguí los pasos que se indican en el [Anexo I](#). Una vez instalado el PHP, se tiene que editar el fichero con nombre "php.ini" y realizar las configuraciones necesarias para desarrollar la aplicación.

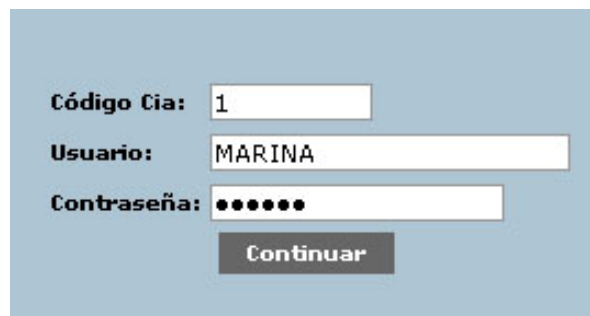
6.2.- Demostración del funcionamiento de la aplicación

6.2.1.- Validación y acceso a la aplicación

Una vez el usuario entra en la aplicación, nos aparecerá la pantalla de validación de usuario.



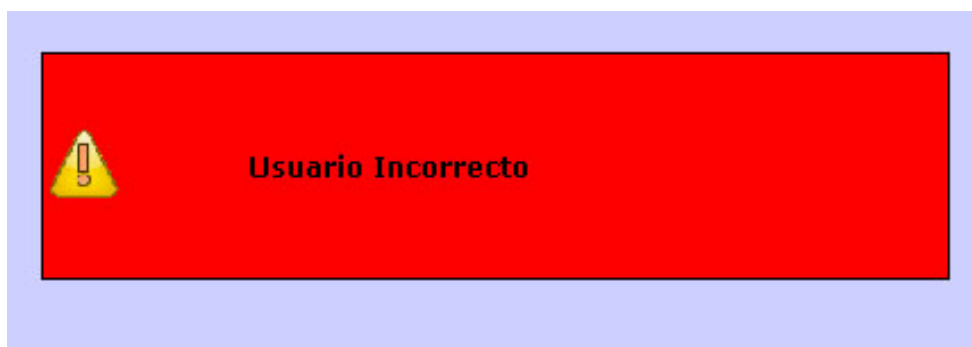
En esta pantalla el usuario deberá validarse para poder acceder a la aplicación. Deberá hacerlo rellenando los datos en el formulario que se muestra a continuación. La primera vez que se acceda en la aplicación, para que el usuario de la aplicación pueda darse de alta en la aplicación, habrá dado de alta por defecto un usuario junto con su contraseña.



Código Cia: 1
Usuario: MARINA
Contraseña: ●●●●●●
Continuar

Código de la compañía: 1
Nombre del usuario: MARINA
Contraseña: MARINA

Si la validación es incorrecta, la aplicación mostrará el siguiente mensaje de error:



Una vez validado el usuario, pasará a ver la siguiente pantalla que le permitirá acceder a los diferentes mantenimientos de la aplicación.

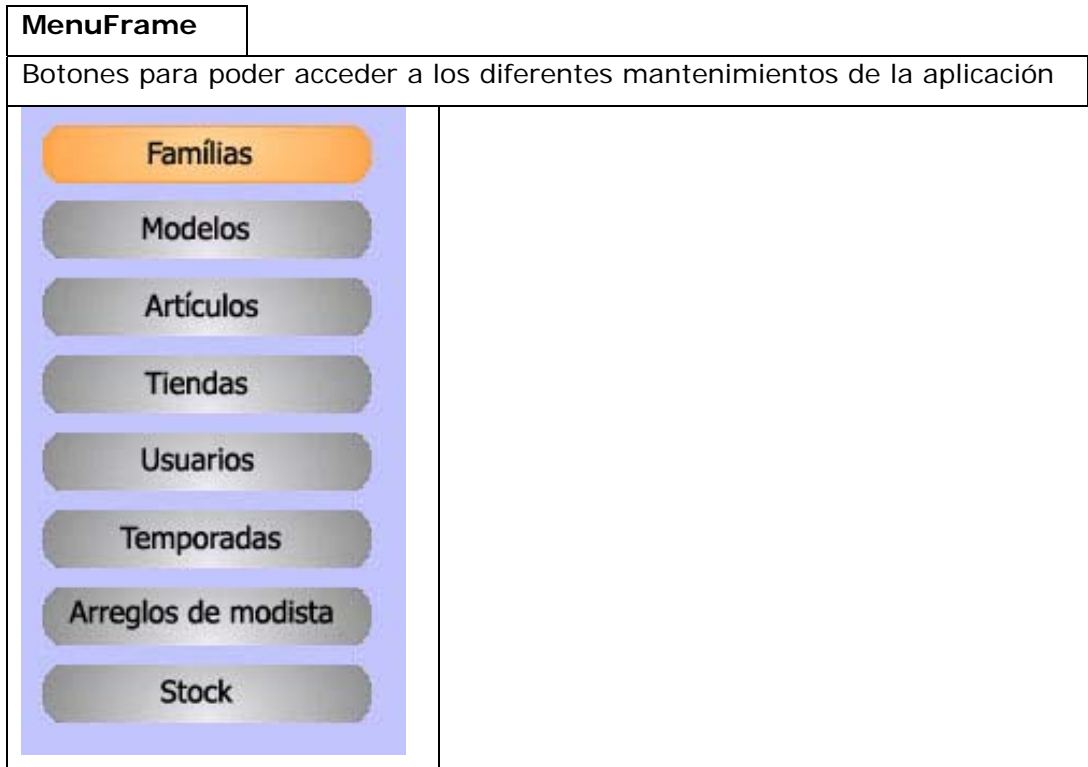
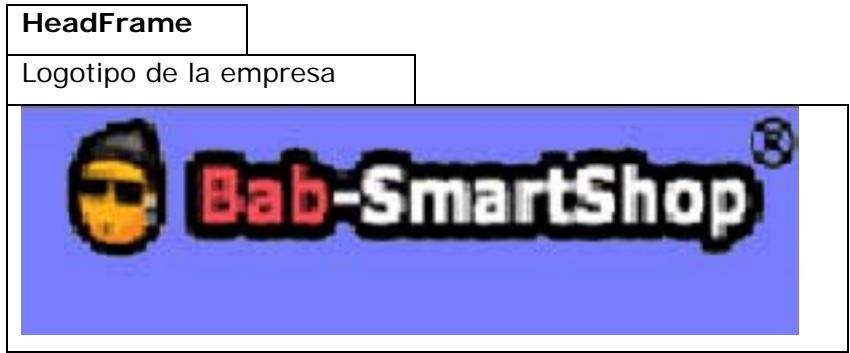


Utilidades de los botones del contentFrame

 <p>Familias</p>	<p>Nos permitirá acceder al mantenimiento de Familias</p>	 <p>Modelos</p>	<p>Nos permitirá acceder al mantenimiento de Modelos</p>
 <p>Artículos</p>	<p>Nos permitirá acceder al mantenimiento de Artículos</p>	 <p>Tiendas</p>	<p>Nos permitirá acceder al mantenimiento de Tiendas</p>
 <p>Usuarios</p>	<p>Nos permitirá acceder al mantenimiento de Usuarios</p>	 <p>Temporadas</p>	<p>Nos permitirá acceder al mantenimiento de Temporadas</p>
 <p>Arreglos</p>	<p>Nos permitirá acceder al mantenimiento de Arreglos de modista</p>	 <p>Stock</p>	<p>Nos permitirá acceder al mantenimiento de Stock</p>
 <p>Informes</p>	<p>Nos permitirá acceder a los Informes de la aplicación</p>		<p>Nos permitirá abandonar la aplicación</p>

6.2.2.- Frames del diseño de la aplicación



Dentro de cada mantenimiento se pueden distinguir los siguientes frames.













ContentFrame

Contiene los datos de la aplicación, listados, búsquedas, diálogos de altas e informes.

Código Familia Descripción

	Código Familia	Descripción	Familia Padre	Anulado	
<input type="checkbox"/>	PRUEBA	Familia de pruebas		No	
<input type="checkbox"/>	SDFHG	sdhgj		Sí	
<input type="checkbox"/>	SML	sueter de manga larga		No	
<input type="checkbox"/>	SML02	sueter de manga larga tipo 21	SML	No	
<input type="checkbox"/>	SML1	Sueter de manga larga Tommy	SML	No	
<input type="checkbox"/>	SML11	Sueter mejorado	SML1	Sí	
<input type="checkbox"/>	SML2	Sueter de manga larga XDYE	SML	No	
<input type="checkbox"/>	SML3	Sueter de manga larga Ralph Lauren	SML	No	
<input type="checkbox"/>	SML4	Sueter de manga larga NIKE	SML	No	
<input type="checkbox"/>	ST	sueter tirantes		No	

1 2 

FootFrame



Pie de página, contiene los iconos de página de inicio y de finalización de la sesión



6.2.3.- Elementos comunes de los mantenimientos

A continuación explicaré algunas de las funcionalidades comunes de todos los mantenimientos.


Los siguientes botones aparecen en la parte inferior de los listados cuando estos tienes varias páginas de datos para mostrar.

-  Este botón nos indica que el listado tiene varias páginas. Haciendo clic sobre este botón podremos pasar a la página siguiente.
-  Este botón nos indica que el listado tiene varias páginas. Haciendo clic sobre este botón podremos pasar a la página anterior.
- 1 Este botón nos indica la posibilidad por parte del usuario de ir a la página número 1.

Imágenes de búsqueda. Los parámetros de búsqueda cambiarán dependiendo del mantenimiento.



El formulario de búsqueda tiene un fondo azul claro. A la izquierda, el texto 'Código Familia' precede a un campo de entrada rectangular. A la derecha, el texto 'Descripción' precede a otro campo de entrada rectangular. Pequeños iconos de lupa se encuentran al final de cada campo de entrada.

La interficie que se muestra en la imagen permite al usuario realizar búsquedas por código de familia y por descripción. Para poder realizar la búsqueda, el usuario sólo deberá introducir el campo que quiere encontrar y hacer clic sobre el icono .

Botones de uso en la aplicación.



Botón que nos permite pasar al formulario para dar de alta.

Botón que nos permitirá borrar registros de la aplicación.

Botón que nos permitirá pasar al formulario de modificación de datos.

Botón que nos permitirá lanzar las búsquedas.

6.2.4.- Mantenimientos de la aplicación

6.2.4.1.- Mantenimiento de familias

A continuación se muestra la pantalla de mantenimiento de familias:

The screenshot displays the Bab-SmartShop application interface. On the left is a sidebar with navigation buttons: Familias (highlighted), Modelos, Artículos, Tiendas, Usuarios, Temporadas, Arreglos de modista, and Stock. The main area features search filters for 'Código Familia' and 'Descripción'. Below the filters is a table listing family records:

Código Familia	Descripción	Familia Padre	Anulado	
PRUEBA	Familia de pruebas		No	
SDFHG	sdhgg		Sí	
SML	sueter de manga larga		No	
SML02	sueter de manga larga tipo 21	SML	No	
SML1	Sueter de manga larga Tommy	SML	No	
SML11	Sueter mejorado	SML1	Sí	
SML2	Sueter de manga larga XDYE	SML	No	
SML3	Sueter de manga larga Ralph Lauren	SML	No	
SML4	Sueter de manga larga NIKE	SML	No	
ST	sueter tirantes		No	

At the bottom of the table, there is a pagination control showing '1 2' with arrows, indicating the current page is 1 of 2.

Podemos comprobar, que como la aplicación tiene datos de prueba, al hacer clic en el mantenimiento de familias la aplicación muestra un listado con todas las familias que están dadas de alta en la base de datos.

El listado muestra el código de familia, la descripción, la familia padre y el campo de Anulado. Los registros se muestran ordenados por código de familia.


A continuación muestro un ejemplo de cómo el usuario podrá moverse entre las diferentes páginas de datos. Los datos se encuentran listados en registros de 10 elementos.

En este ejemplo podemos ver como el listado de familias tiene dos páginas.


Para pasar de la página **1**, donde se encuentra actualmente el usuario, a la página **2** lo puede hacer de dos maneras. Estas dos maneras se explican a continuación:

Listado de familias, datos de la página 1

	Código Familia	Descripción	Familia Padre	Anulado	
<input type="checkbox"/>	PRUEBA	Familia de pruebas		No	
<input type="checkbox"/>	SML	sueter de manga larga		No	
<input type="checkbox"/>	SML02	sueter de manga larga tipo 2 1	SML	No	
<input type="checkbox"/>	SML1	Sueter de manga larga Tommy	SML	No	
<input type="checkbox"/>	SML11	Sueter mejorado	SML1	Sí	
<input type="checkbox"/>	SML2	Sueter de manga larga XDYE	SML	No	
<input type="checkbox"/>	SML3	Sueter de manga larga Ralph Lauren	SML	No	
<input type="checkbox"/>	SML4	Sueter de manga larga NIKE	SML	No	
<input type="checkbox"/>	ST	sueter tirantes		No	
<input type="checkbox"/>	ST1	Sueter tirantes Mango	ST	No	


Para pasar a la páginas siguiente el usuario deberá hacer clic al botón  , o bien podrá pulsar el botón [2](#). El listado enseñará los datos de la página siguiente, como se muestra en la imagen que encontramos a continuación.

	Código Familia	Descripción	Familia Padre	Anulado	
<input type="checkbox"/>	ST2	Sueter tirantes System Action	ST	No	
<input type="checkbox"/>	TEJPIR	tejanillos piratas		No	
<input type="checkbox"/>	TEJPIR1	Tejanos piratas Tommy	TEJPIR	No	
<input type="checkbox"/>	TEJPIR11	Tejanos piratas Tommy Desgastados	TEJPIR1	No	
<input type="checkbox"/>	TEJPIR12	Tejanos piratas Tommy Rajados	TEJPIR1	No	
<input type="checkbox"/>	TEJPIR13	Tejanos piratas Tommy Rotos	TEJPIR1	No	
<input type="checkbox"/>	TEJPIR2	Tejanos piratas Bullberry	TEJPIR	No	
<input type="checkbox"/>	TEJPIR3	Tejanos piratas NIKE	TEJPIR	No	
<input type="checkbox"/>	TEJPIR4	tejano pirata Roxy	TEJPIR	No	

Esta es la segunda página de nuestro listado de familias, el usuario apretando el botón  o [1](#) podrá volver a la página anterior.

6.2.4.1.1.- Búsqueda por familias

El usuario rellena el Edit de la búsqueda con el código de la familia a buscar, en este caso el usuario quiere encontrar todas aquellas familias que tengan SML como parte de su código de familia.

El usuario pulsa el botón 

El programa devuelve el siguiente listado, una vez ha realizado la búsqueda realizada por el usuario.




The screenshot shows a search interface with a light blue background. At the top, there are two search input fields: 'Código Familia' containing 'SML' and 'Descripción' which is empty. Below the inputs are two small icons: a blue folder icon on the left and a blue trash can icon on the right. The main content is a table with a light blue header and white rows. The table has five columns: 'Código Familia', 'Descripción', 'Familia Padre', 'Anulado', and an empty column with a small icon. The table contains seven rows of data. Below the table, the number '1' is centered.

	Código Familia	Descripción	Familia Padre	Anulado	
<input type="checkbox"/>	SML	sueter de manga larga		No	
<input type="checkbox"/>	SML02	sueter de manga larga tipo 21	SML	No	
<input type="checkbox"/>	SML1	Sueter de manga larga Tommy	SML	No	
<input type="checkbox"/>	SML11	Sueter mejorado	SML1	Sí	
<input type="checkbox"/>	SML2	Sueter de manga larga XDYE	SML	No	
<input type="checkbox"/>	SML3	Sueter de manga larga Ralph Lauren	SML	No	
<input type="checkbox"/>	SML4	Sueter de manga larga NIKE	SML	No	

1

6.2.4.1.2.- Búsqueda por Descripción

El usuario rellena el Edit de la búsqueda con la descripción de familia a buscar. En este caso el usuario quiere encontrar todas aquellas descripciones que tengan Suéter como parte de su descripción.

El usuario pulsa el botón 

El programa devuelve el siguiente listado, una vez ha realizado la búsqueda realizada por el usuario. Podemos comprobar que devuelve las búsquedas ordenadas por código de familia .




The screenshot shows a search interface with two input fields: 'Código Familia' (empty) and 'Descripción' (containing 'Suéter'). Below the fields is a table with the following data:

	Código Familia	Descripción	Familia Padre	Anulado	
<input type="checkbox"/>	SML	sueter de manga larga		No	
<input type="checkbox"/>	SML02	sueter de manga larga tipo 2 1	SML	No	
<input type="checkbox"/>	SML1	Sueter de manga larga Tommy	SML	No	
<input type="checkbox"/>	SML11	Sueter mejorado	SML1	Sí	
<input type="checkbox"/>	SML2	Sueter de manga larga XDYE	SML	No	
<input type="checkbox"/>	SML3	Sueter de manga larga Ralph Lauren	SML	No	
<input type="checkbox"/>	SML4	Sueter de manga larga NIKE	SML	No	
<input type="checkbox"/>	ST	sueter tirantes		No	
<input type="checkbox"/>	ST1	Sueter tirantes Mango	ST	No	
<input type="checkbox"/>	ST2	Sueter tirantes System Action	ST	No	

1

6.2.4.1.3.- Alta de familias

A continuación el usuario se dispone a dar de alta familias en la base de datos.

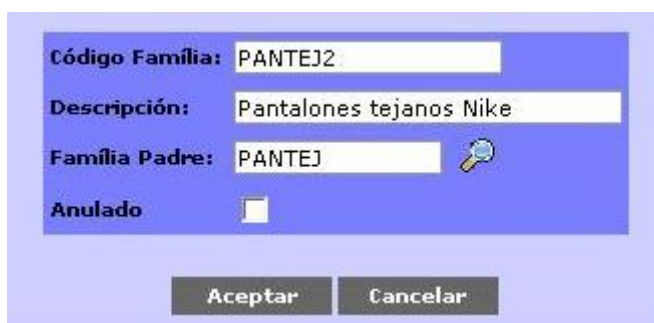
Para dar de alta una familia el usuario debe pulsar el botón que realiza esta funcionalidad  y en la pantalla visualizaremos un nuevo formulario para poder realizar esta función.

Formulario de alta




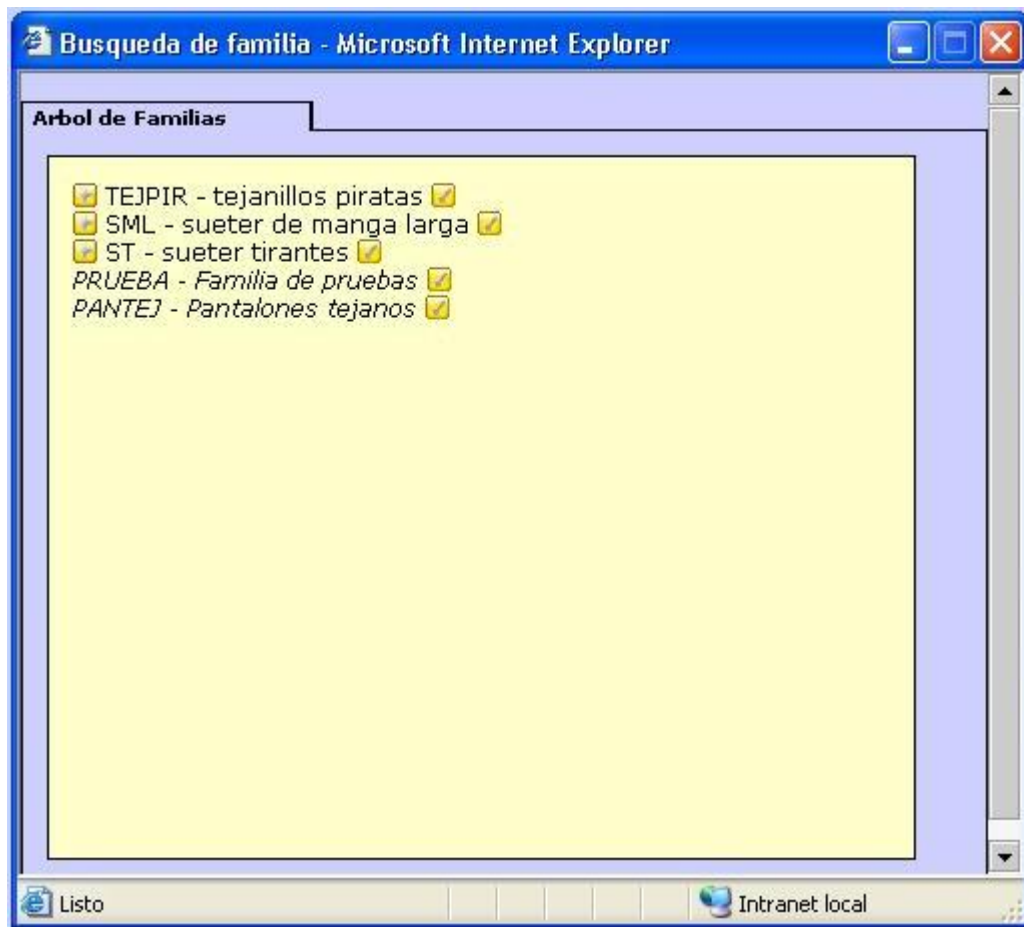
Formulario de alta vacío. Campos: Código Familia: [], Descripción: [], Familia Padre: [] (con icono de lupa), Anulado: . Botones: Aceptar, Cancelar.

Una vez el usuario introduzca los datos en el formulario aparecerá de la siguiente forma



Formulario de alta con datos. Campos: Código Familia: PANTEJ2, Descripción: Pantalones tejanos Nike, Familia Padre: PANTEJ (con icono de lupa), Anulado: . Botones: Aceptar, Cancelar.

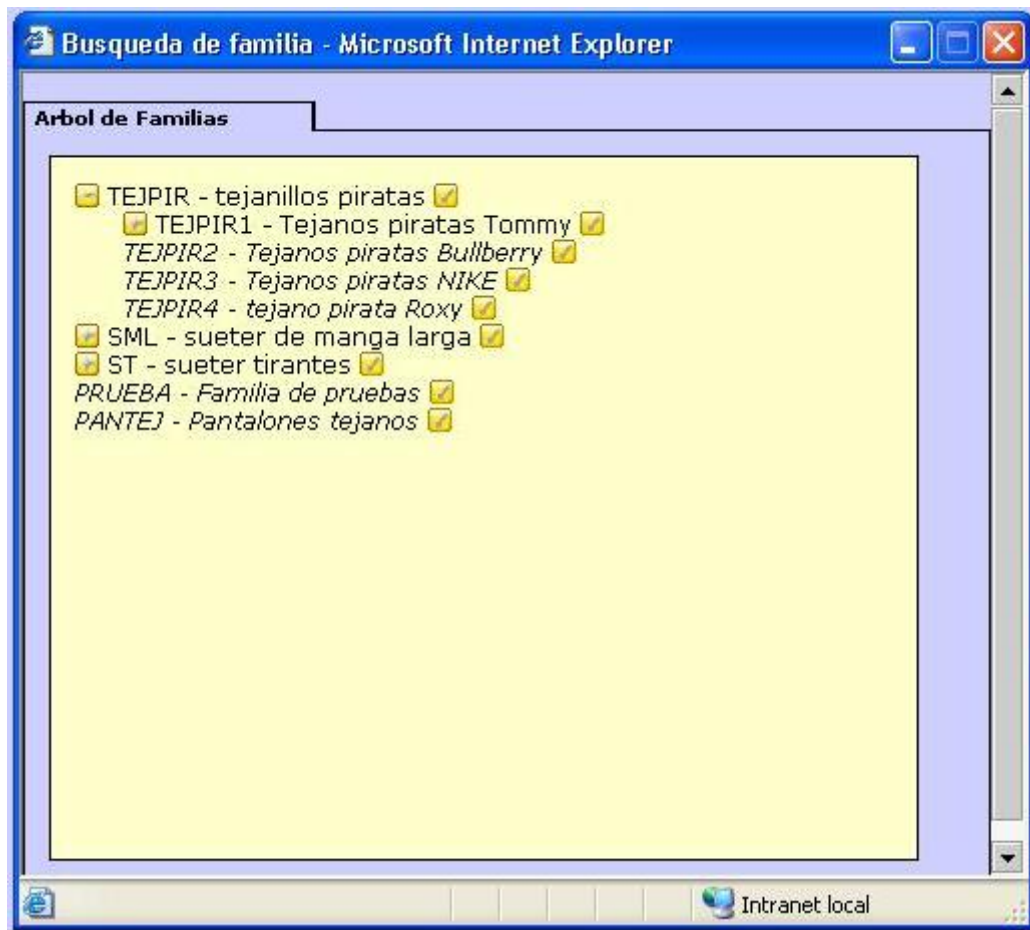
Como se puede ver el usuario ha rellenado la Familia Padre. Para poder rellenarla sin tener que recordar la familia dentro de la cual se encuentra jerarquizada, el usuario puede pulsar el botón  y aparecerá la siguiente ventana.



Este árbol de familias se puede expandir o seleccionar la familia de la línea actual, utilizando los siguiente iconos:

Para expandir la rama del árbol Para seleccionar la familia actual.

Si seleccionamos la opción de expandir, podemos ver el árbol de la siguiente forma:




Se entiende que en aquellas líneas donde encontramos el icono de expandir es porque esa rama tiene hijos, sino la rama es final y sólo dispone del icono de seleccionar.

Para esta familia he seleccionado como familia padre **PANTEJ**

Una vez que el usuario haya rellenado los campos obligatorios del formulario de alta, se dispondrá a aceptar el formulario y los datos se insertarán en la base de datos.

6.2.4.1.4.- Modificar Familias

El usuario que se encuentra en el listado de familias decide modificar una de éstas. Para ello se colocará en la fila de la familia que desea modificar y pulsará el

siguiente botón 

Formulario de modificación de familias



Código Familia:	SML2
Descripción:	Sueter de manga larga XDYE
Familia Padre:	SML 
Anulado	<input type="checkbox"/>

Aceptar **Cancelar**

En este formulario el usuario podrá modificar todos los campos excepto el de código familia. Para ello sólo deberá posicionarse en el edit del campo que desee modificar y realizar la misma. Una vez acabada la modificación deberá acabar pulsando el botón de aceptar para que esta se modifique en la base de datos.


6.2.4.1.5.- Eliminar Familias

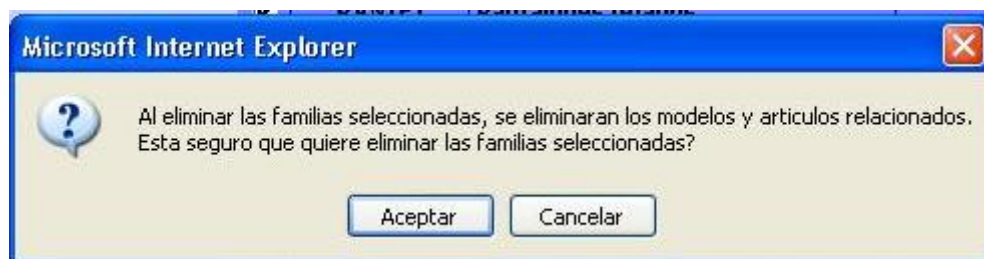
El usuario se encuentra en el mantenimiento de familias y decide eliminar una o varias de estas. Para ello deberá chequear la fila o filas de familias que desee eliminar y pulsar el botón destinado para esto.



	Código Familia	Descripción	Familia Padre	Anulado	
<input checked="" type="checkbox"/>	PANTEJ	Pantalones tejanos		Sí	
<input type="checkbox"/>	PANTEJ2	Pantalones tejanos Nike	PANTEJ	No	
<input type="checkbox"/>	PRUEBA	Familia de pruebas		No	
<input type="checkbox"/>	SML	sueter de manga larga		No	
<input type="checkbox"/>	SML02	sueter de manga larga tipo 21	SML	No	
<input type="checkbox"/>	SML1	Sueter de manga larga Tommy	SML	No	
<input type="checkbox"/>	SML11	Sueter mejorado	SML1	Sí	
<input type="checkbox"/>	SML2	Sueter de manga larga XDYE	SML	No	
<input type="checkbox"/>	SML3	Sueter de manga larga Ralph Lauren	SML	No	
<input type="checkbox"/>	SML4	Sueter de manga larga NIKE	SML	No	

El usuario chequea para eliminar la fila donde se encuentra la familia **PANTEJ**. Podemos ver que en la línea siguiente nos encontramos con la familia PANTEJ2 que tiene como familia padre **PANTEJ**, la familia que deseamos eliminar.

El usuario pulsa el botón 



La aplicación nos muestra el siguiente mensaje, para que el usuario conozca que realizando esta operación se borrarán todas aquellas familias seleccionadas y las que dependan de ella, junto con los modelos y artículos relacionados con la misma. Si el usuario acepta se pasará a borrar todos los registros de la base de datos. Una vez eliminada la familia el listado se muestra de la siguiente manera

Código Familia Descripción

	Código Familia	Descripción	Familia Padre	Anulado	
<input type="checkbox"/>	PRUEBA	Familia de pruebas		No	
<input type="checkbox"/>	SML	sueter de manga larga		No	
<input type="checkbox"/>	SML02	sueter de manga larga tipo 21	SML	No	
<input type="checkbox"/>	SML1	Sueter de manga larga Tommy	SML	No	
<input type="checkbox"/>	SML11	Sueter mejorado	SML1	Sí	
<input type="checkbox"/>	SML2	Sueter de manga larga XDYE	SML	No	
<input type="checkbox"/>	SML3	Sueter de manga larga Ralph Lauren	SML	No	
<input type="checkbox"/>	SML4	Sueter de manga larga NIKE	SML	No	
<input type="checkbox"/>	ST	sueter tirantes		No	
<input type="checkbox"/>	ST1	Sueter tirantes Mango	ST	No	

1 2

Podemos comprobar como se han borrado la familia **PANTEJ** y **PANTEJ2**

6.2.4.2.- Mantenimiento de modelos

A continuación se muestra la pantalla de mantenimiento de modelos:

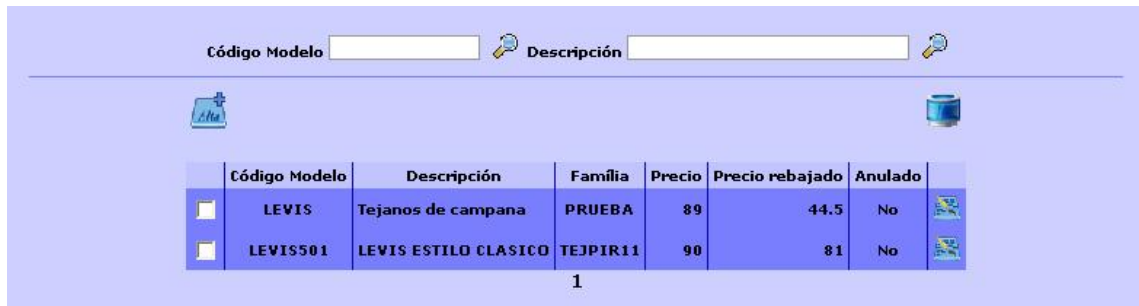
Familias
Modelos
Artículos
Tiendas
Usuarios
Temporadas
Arreglos de modista
Stock

Código Modelo Descripción



	Código Modelo	Descripción	Familia	Precio	Precio rebajado	Anulado	
<input type="checkbox"/>	LEVIS	Tejanos de campana	PRUEBA	89	44.5	No	
<input type="checkbox"/>	LEVIS001	LEVIS ESTILO CLASICO	TEJPIR11	90	81	No	

1

El contentFrame de este mantenimiento es el siguiente:



The screenshot shows a web application interface with a search form at the top and a table of data below. The search form has two input fields: "Código Modelo" and "Descripción", each with a magnifying glass icon. Below the search form is a table with the following columns: "Código Modelo", "Descripción", "Familia", "Precio", "Precio rebajado", "Anulado", and a small icon column. The table contains two rows of data. Below the table is a page number "1".

	Código Modelo	Descripción	Familia	Precio	Precio rebajado	Anulado	
<input type="checkbox"/>	LEVIS	Tejanos de campana	PRUEBA	89	44.5	No	
<input type="checkbox"/>	LEVIS501	LEVIS ESTILO CLASICO	TEJPIR11	90	81	No	

1


Podemos comprobar que como la aplicación tiene datos de prueba el usuario al hacer clic en el mantenimiento de modelos, la aplicación muestra un listado con los datos de modelos dados de alta en la base de datos.

El listado muestra el código de modelo, la descripción, la familia, el precio, el precio rebajado y el campo de Anulado. El listado de modelos se encuentra ordenado por código de modelo.

6.2.4.2.1.- Alta de modelos

A continuación el usuario se dispone a dar de alta modelos en la base de datos.

Para dar de alta un modelo el usuario debe pulsar el botón que realiza esta

funcionalidad  y en la pantalla visualizaremos un nuevo formulario para poder realizar esta función.

Formulario de alta



Formulario de alta vacío con los siguientes campos:

Código Modelo:	<input type="text"/>
Descripción:	<input type="text"/>
Familia:	<input type="text"/> 
Precio:	<input type="text"/>
Descuento %:	<input type="text"/>
Precio Rebajado:	<input type="text"/>
Anulado:	<input type="checkbox"/>

Botones: **Aceptar** **Cancelar**

Una vez el usuario introduzca los datos el formulario aparecerá la siguiente imagen




Formulario de alta con los siguientes datos:


Código Modelo:	CALVIN
Descripción:	Camisa CALVIN KLEIN
Familia:	PRUEBA 
Precio:	120
Descuento %:	<input type="text"/>
Precio Rebajado:	100
Anulado:	<input type="checkbox"/>


Botones: **Aceptar** **Cancelar**


Como se puede ver el usuario ha rellenado la Familia. Para poder rellenarla sin tener que recordar la familia dentro de la cual se encuentra el modelo, el usuario puede

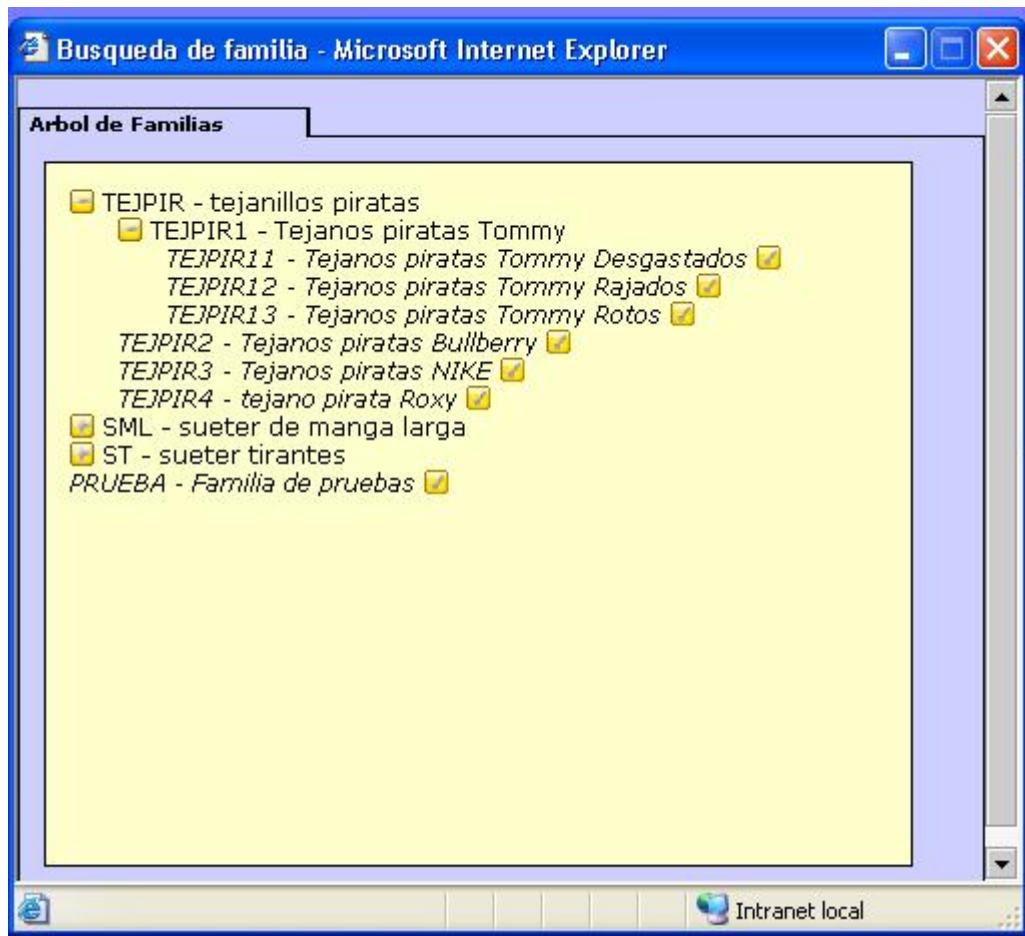
pulsar el botón  y aparecerá la siguiente pantalla.



Se trata de un árbol de familias. El usuario podrá seleccionar una de las familias rama de este árbol, esto lo hará apretando 

Para recorrer las diferentes ramas de las familias podrá hacerlo con apretando 

Esta es la imagen del árbol expandido, una vez que el usuario ha apretado el botón 



Para este modelo el usuario ha seleccionado como familia **PRUEBA**

Una vez que el usuario haya rellenado los campos obligatorios del formulario de alta, se dispondrá a aceptar el formulario y los datos se insertarán en la base de datos.

6.2.4.2.2.- Modificar Modelos

El usuario que se encuentra en el listado de modelos y decide modificar uno de éstos. Para ello se colocará en la fila del modelo que desea modificar y pulsará el

siguiente botón 

Formulario de modificación de modelos




Código Modelo:	LEVIS
Descripción:	Tejanos de campana
Familia:	PRUEBA 
Precio:	89
Descuento %:	
Precio Rebajado:	44.5
Anulado:	<input type="checkbox"/>

Aceptar Cancelar

En este formulario el usuario podrá modificar todos los campos excepto el de código modelo. Para ello sólo deberá posicionarse en el edit del campo que desee modificar y realizar la misma. Una vez acabada la modificación deberá acabar pulsando el botón de aceptar para que este se modifique en la base de datos.

6.2.4.2.3.- Eliminar Modelos

El eliminar sigue el mismo proceso para todos los mantenimientos seleccionar la fila deseada y proceder a eliminarla pulsando .

6.2.4.3.- Mantenimiento de artículos

A continuación se muestra la pantalla del contentFrame del mantenimiento de artículos:

Código Artículo Descripción

	Código Artículo	Código Modelo	Código Familia	Descripción	Referencia	Temporada	Talla	Color	Precio	Precio rebajado	Rebajado	Cantidad	Anulado
<input type="checkbox"/>	ART1	LEVIS	PRUEBA	Esto es un artículo de prueba	1234567890123	Invierno	38	Azul	89	44.5	No	10000	No
<input type="checkbox"/>	ART2	LEVIS501	TEJPIR11	Artículo de prueba		Invierno	40	Verde	90	81	No	200	No
<input type="checkbox"/>	ART3	LEVIS	PRUEBA	Artículo 3	1234565432178	Verano	46	Rojo	120	108	No	20093	No
<input type="checkbox"/>	ART4	LEVIS501	TEJPIR11	Prueba		Otoño	54	fuixia	90	81	No	2345	No
<input type="checkbox"/>	ART5	LEVIS	PRUEBA	Artículo 5		Verano	15	azul	89	44.5	No	10000	No
<input type="checkbox"/>	ART6	LEVIS501	TEJPIR11	artículo 6		Primavera	43	rojo	90	81	No	800	No
<input type="checkbox"/>	CAMISA	LEVIS501	TEJPIR11	Camisa de manga larga		Verano	XL	Negro	90	81	No	100	No


1

Podemos comprobar que como la aplicación tiene datos de prueba el usuario al hacer clic en el mantenimiento de artículos, esta muestra una serie de registros que se encuentran dados de alta en la base de datos.


El listado muestra el código de artículo, el código de modelo, el código familia, la descripción, la referencia, la temporada, la talla, el color, el precio, el precio rebajado, la cantidad y el campo de Anulado. El listado de artículos se encuentra ordenado por código de artículo.

6.2.4.3.1.- Alta de Artículos

A continuación el usuario se dispone a dar de alta artículos en la base de datos.

Para dar de alta un artículo el usuario debe pulsar el botón que realiza esta funcionalidad  y en la pantalla visualizaremos un nuevo formulario para poder realizar esta función.


Formulario de alta

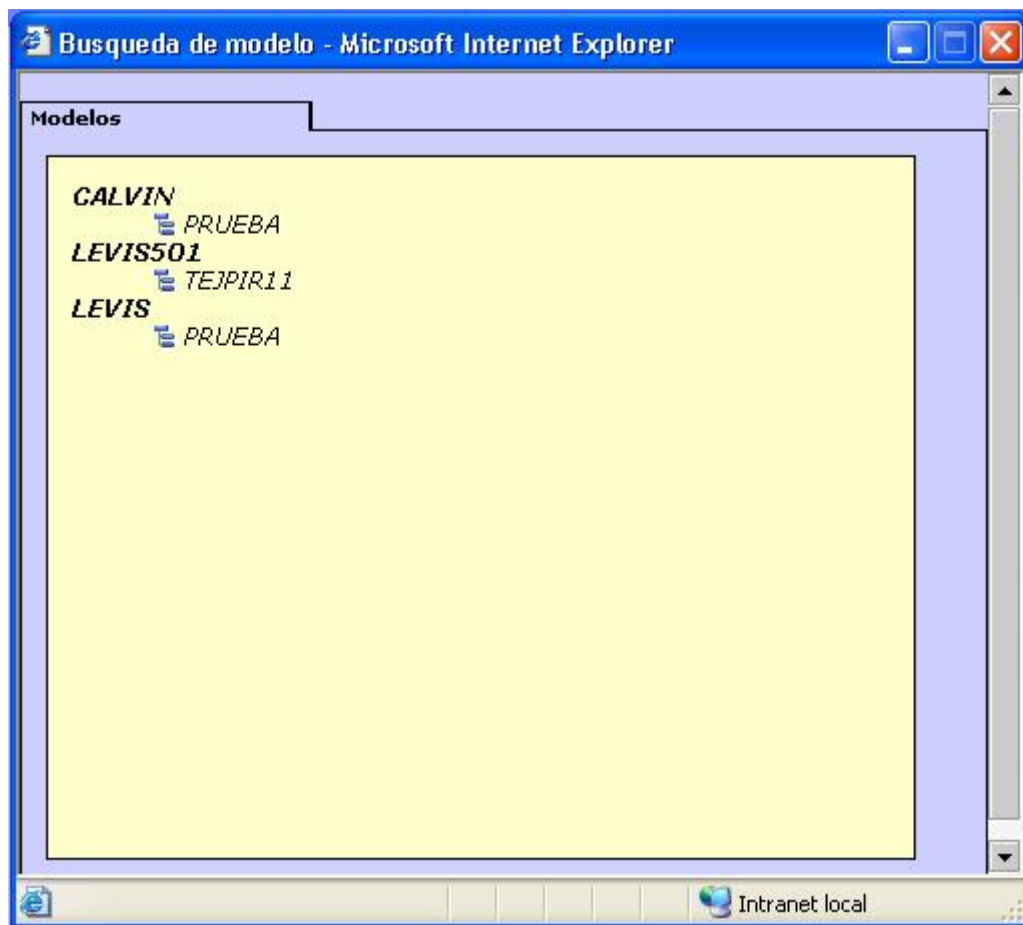
Artículo:	<input type="text"/>
Descripción:	<input type="text"/>
Modelo:	<input type="text"/> 
Familia:	<input type="text"/>
Temporada:	<input type="text"/> ▼
Talla:	<input type="text"/>
Color:	<input type="text"/>
Precio Modelo:	<input type="checkbox"/>
Precio:	<input type="text"/>
%Descuento:	<input type="text"/>
Precio Rebajado:	<input type="text"/>
Referencia:	<input type="text"/>
Cantidad:	<input type="text"/>
Anulado	<input type="checkbox"/>
Rebajado	<input type="checkbox"/>

Una vez el usuario introduzca los datos el formulario aparecerá como muestra la siguiente imagen

Artículo:	LEVIS512
Descripción:	Pantalones tejanos Levis 512
Modelo:	LEVIS 
Familia:	PRUEBA
Temporada:	Invierno 
Talla:	38
Color:	Azul
Precio Modelo:	<input checked="" type="checkbox"/>
Precio:	89
%Descuento:	
Precio Rebajado:	44,5
Referencia:	1234567890123
Cantidad:	1000
Anulado	<input type="checkbox"/>
Rebajado	<input type="checkbox"/>

Aceptar **Cancelar**

Como se puede ver el usuario ha rellenado el modelo. Para poder rellenarlo sin tener que recordar el modelo dentro del se encuentra el artículo, el usuario, puede pulsar el botón  y aparecerá la siguiente imagen.



El usuario podrá seleccionar un modelo de la siguiente lista, esto lo hará apretando sobre el modelo que desee. Como información debajo del modelo nos muestra la familia a la que pertenece el mismo.

Para este artículo he seleccionado el modelo **LEVIS** y la familia **PRUEBA**

Una vez que el usuario haya rellenado los campos obligatorios del formulario de alta, se dispondrá a aceptar el formulario y los datos se insertarán en la base de datos.

Una vez se haya aceptado el formulario el usuario tendrá la posibilidad de asignar este artículo al stock de alguna tienda.

Diálogo de confirmación de inserción de stock a una tienda



El diálogo mostrará el siguiente mensaje “¿Quiere agregar stock a tienda?” Si el usuario apreta el botón de aceptar aparecerá la siguiente imagen

Formulario de alta de stock



Formulario de alta de stock con los siguientes campos:

Código Artículo:	LEVIS512
Código Tienda:	
Stock:	
Vendidos:	
Distribuidos:	
Con tara:	
En almacen:	

Botones: Aceptar, Cancelar

En el formulario ya aparecerá el código del artículo y sólo tendremos que seleccionar la tienda de destino del stock del artículo, así como la cantidad de stock que se le desea asignar a esta tienda.



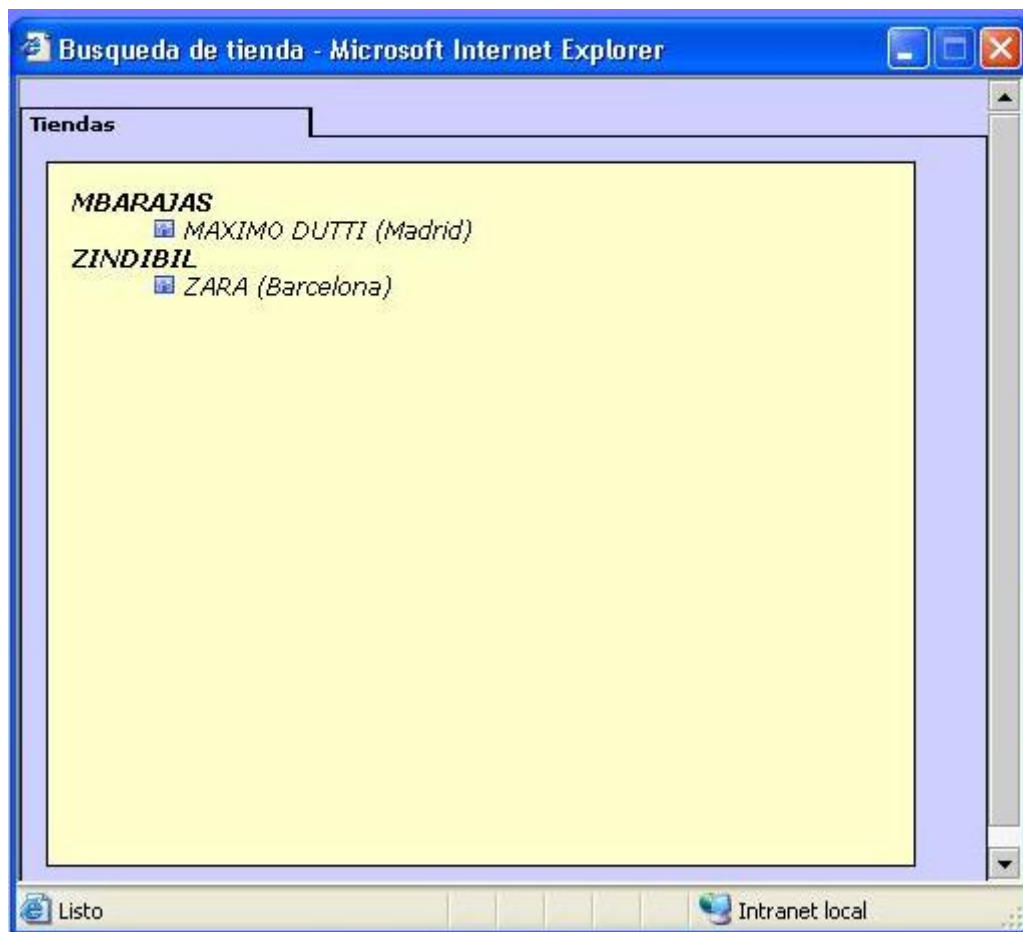
Formulario de alta de stock con los siguientes datos ingresados:

Código Artículo:	LEVIS512
Código Tienda:	ZINDIBIL
Stock:	15
Vendidos:	0
Distribuidos:	0
Con tara:	0
En almacen:	0

Botones: Aceptar, Cancelar

Como podemos ver el usuario ha seleccionado la tienda de código **ZINDIBIL**. Para facilitar esta tarea al usuario también disponemos de un listado con todas las tiendas.

Para esto el usuario debe pulsar  y aparecerá la siguiente pantalla.



El usuario podrá seleccionar la tienda deseada apretando con el botón del ratón sobre la misma. En la línea inferior, debajo del código de la tienda, mostramos información del nombre y la población de la misma.



Una vez asignado todo el stock de este artículo a las diferentes tiendas el usuario aceptará y los registros pasarán a darse de alta en la base de datos.

6.2.4.3.2.- Modificar Artículos

El usuario que se encuentra en el listado de artículos y decide modificar uno de éstos registros. Para ello se colocará en la fila del artículo que desea modificar y


pulsará el siguiente botón 

Formulario de modificación de artículos

Artículo:	<input type="text" value="CAMISA"/>
Descripción:	<input type="text" value="Camisa de manga larga"/>
Modelo:	<input type="text" value="LEVIS501"/> 
Familia:	<input type="text" value="TEJPIR11"/>
Temporada:	<input type="text" value="Verano"/> 
Talla:	<input type="text" value="XL"/>
Color:	<input type="text" value="Negro"/>
Precio Modelo:	<input type="checkbox"/>
Precio:	<input type="text" value="90"/>
% Descuento:	<input type="text"/>
Precio Rebajado:	<input type="text" value="81"/>
Referencia:	<input type="text"/>
Cantidad:	<input type="text" value="100"/>
Anulado	<input type="checkbox"/>
Rebajado	<input type="checkbox"/>

En este formulario el usuario podrá modificar todos los campos excepto el de código de artículo. Para ello sólo deberá posicionarse en el campo que desee modificar y realizar la operación, una vez acabada la modificación deberá acabar pulsando el botón de aceptar para que el cambio se modifique en la base de datos.

6.2.4.3.3.- Eliminar Artículos

El eliminar sigue el mismo proceso para todos los mantenimientos. El usuario debe seleccionar la fila deseada y proceder a eliminarla pulsando .

6.2.4.4.- Mantenimiento de tiendas

A continuación muestro la imagen del contentFrame del mantenimiento de tiendas:



	Código Tienda	NIF	Nombre	Dirección	Código postal	Población	Provincia	Teléfono	Fax	Email
	MBARAJAS	A08590333	MAXIMO DUTTI	Aeropuerto de Barajas	01145	Madrid	Madrid	917685432	917685434	maximodutti@terra
	ZINDIBIL	A08590333	ZARA	Indibil 3	08004	Barcelona	Barcelona	933385643	933385644	zindibil@zara.es


1

Podemos comprobar que como la aplicación tiene datos de prueba el usuario al hacer clic en el mantenimiento de tiendas, esta muestra una serie de registros que se encuentran dados de alta en la base de datos.

El listado muestra el código de tienda, el NIF, el nombre de la tienda, la dirección, el código postal, la población, la provincia, el teléfono, el fax, el Email y el campo de Anulado. El listado de tiendas se encuentra ordenado por código de tienda.

6.2.4.4.1.- Alta de tiendas

A continuación el usuario se dispone a dar de alta tiendas en la base de datos.

Para dar de alta una tienda el usuario debe pulsar el botón  y en la pantalla visualizaremos un nuevo formulario para poder realizar esta función.

Formulario de alta

Código tienda:	<input type="text"/>
Nif:	<input type="text"/>
Nombre:	<input type="text"/>
Dirección:	<input type="text"/>
CP:	<input type="text"/>
Población:	<input type="text"/>
Provincia:	<input type="text"/>
Teléfono:	<input type="text"/>
Teléfono 2:	<input type="text"/>
Fax:	<input type="text"/>
Email:	<input type="text"/>
Anulado:	<input type="checkbox"/>

Una vez el usuario introduzca los datos el formulario aparecerá de la siguiente imagen

Código tienda:	<input type="text" value="TOMYBCN"/>
Nif:	<input type="text" value="A08590325"/>
Nombre:	<input type="text" value="Tommy Hilfiger"/>
Dirección:	<input type="text" value="c/Barcelona 34"/>
CP:	<input type="text" value="08912"/>
Población:	<input type="text" value="Hospitalet de Llobregat"/>
Provincia:	<input type="text" value="Barcelona"/>
Teléfono:	<input type="text" value="933327865"/>
Teléfono 2:	<input type="text"/>
Fax:	<input type="text"/>
Email:	<input type="text" value="tommy.hilfiger@hotmail.com"/>
Anulado:	<input type="checkbox"/>

Una vez que el usuario haya rellenado los campos obligatorios del formulario, pasará a apretar el botón de aceptar. De esta manera se insertarán los nuevos campos en la base de datos

6.2.4.4.2.- Modificar Tiendas

El usuario que se encuentra en el listado de tiendas y decide modificar una de éstas. Para ello se colocará en la fila de la tienda que desea modificar y pulsará el


siguiente botón 

Formulario de modificación de tiendas

Código tienda:	<input type="text" value="MBAARAJAS"/>
Nif:	<input type="text" value="A08590333"/>
Nombre:	<input type="text" value="MAXIMO DUTTI"/>
Dirección:	<input type="text" value="Aeropuerto de Barajas"/>
CP:	<input type="text" value="01145"/>
Población:	<input type="text" value="Madrid"/>
Provincia:	<input type="text" value="Madrid"/>
Teléfono:	<input type="text" value="917685432"/>
Teléfono 2:	<input type="text" value="917685433"/>
Fax:	<input type="text" value="917685434"/>
Email:	<input type="text" value="maximodutti@terra.es"/>
Anulado:	<input checked="" type="checkbox"/>

En este formulario el usuario podrá modificar todos los campos excepto el de código de tienda. Para ello sólo deberá posicionarse en el campo que desee modificar y realizar el cambio, una vez acabada la modificación deberá pulsar el botón de aceptar para que esta quede registrada en la base de datos.

6.2.4.4.3.- Eliminar Tiendas

El eliminar sigue el mismo proceso para todos los mantenimientos. El usuario debe seleccionar la fila o filas deseadas y proceder a eliminarla pulsando  .

6.2.4.5.- Mantenimiento de usuarios

A continuación mostramos la imagen del contentFrame del mantenimiento de usuarios:



The screenshot shows a web interface for user management. At the top, there are two search input fields: 'Código Usuario' and 'Nombre', each with a magnifying glass icon. Below the search fields is a table with 13 columns: 'Código Usuario', 'Nombre', 'Apellidos', 'NIF', 'Dirección', 'Población', 'Provincia', 'CP', 'Teléfono', 'Móvil', 'Email', 'Perfil', and 'Anulado'. The table contains three rows of user data. Each row has a small square icon to its left. Below the table, the number '1' is centered.

	Código Usuario	Nombre	Apellidos	NIF	Dirección	Población	Provincia	CP	Teléfono	Móvil	Email	Perfil	Anulado
<input type="checkbox"/>	ANGEL	Angel	Romero Hernández	47787807V	Avda. Fraternitat 3	Viladecans	Barcelona	08917	936470647	606293849	anromher@hotmail.com	Usuario	No
<input type="checkbox"/>	DANI	Daniel	García lugo	47787807V	c/ Indibil 3	Barcelona	Barcelona	08004	932894045	655617384	sempy2@hotmail.com	Administrador	No
<input type="checkbox"/>	MARINA	Marina	Lerma Gómez	46767531K	c/ Indibil 3	Barcelona	Barcelona	08004	932894045	626409850	aniram_77@yahoo.es	Administrador	No


1

Como la aplicación tiene datos de prueba, el usuario al hacer clic en el mantenimiento de tiendas, la aplicación nos muestra un listado con todos de usuarios que están dados de alta en la base de datos.

El listado muestra el código de usuario, el nombre, los apellidos, el NIF, la dirección, la población, la provincia el código postal, el teléfono, el teléfono móvil, el email, el perfil del usuario y el campo de Anulado. El listado de usuarios se encuentra ordenado por código de usuario.

6.2.4.5.1.- Alta de usuarios

A continuación el usuario se dispone a dar de alta usuarios en la base de datos.

Para dar de alta un usuario se debe pulsar el botón  y en la pantalla visualizaremos un nuevo formulario para poder realizar esta función.

Formulario de alta

Código Usuario:	<input type="text"/>
Nombre:	<input type="text"/>
Apellidos:	<input type="text"/>
Nif:	<input type="text"/>
Dirección:	<input type="text"/>
Poblacion:	<input type="text"/>
Provincia:	<input type="text"/>
CP:	<input type="text"/>
Teléfono fijo:	<input type="text"/>
Móvil:	<input type="text"/>
Email:	<input type="text"/>
Perfil Usuario:	Usuario <input type="button" value="v"/>
Anulado:	<input type="checkbox"/>
Modificar contraseña:	<input checked="" type="checkbox"/> <input type="text"/>
Todas las franquicias:	<input checked="" type="checkbox"/>

En este formulario el vendedor nuevo puede formar parte de una o varias franquicias. Si deseamos que este vendedor pertenezca a alguna tienda en concreto y no a todas, deberemos deschequear "*Todas las franquicias*" y de esta manera aparecerá una lista con todas las franquicias existentes actualmente

Código Usuario:

Nombre:

Apellidos:

Nif:

Dirección:

Poblacion:

Provincia:

CP:

Teléfono fijo:

Móvil:

Email:

Perfil Usuario: Usuario

Anulado:

Modificar contraseña:

Todas las franquicias:

MBAARAJAS
TOMYBCN
ZINDIBIL

De esta lista de tiendas se pueden seleccionar una o varias.

Una vez el usuario introduzca todos los datos el formulario aparecerá de la siguiente forma

Código Usuario:	MARI
Nombre:	Maria José
Apellidos:	Maya Heredia
Nif:	46767531K
Dirección:	c/Igualdad 20
Poblacion:	Hospitalet de Llobregat
Provincia:	Barcelona
CP:	08902
Teléfono fijo:	933372890
Móvil:	654768976
Email:	mjmaya@hotmail.com
Perfil Usuario:	Usuario <input type="button" value="v"/>
Anulado:	<input type="checkbox"/>
Modificar contraseña:	<input checked="" type="checkbox"/> <input type="password" value="••••"/>
Todas las franquicias:	<input checked="" type="checkbox"/>

El usuario una vez haya rellenado todos los campos obligatorios del formulario de alta, apretará sobre el botón de aceptar para que estos se inserten en la base de datos.

6.2.4.5.2.- Modificar Usuarios

El usuario se encuentra en el listado de vendedores/usuarios y decide modificar algún registro. Para ello se colocará en la fila del usuario que desea modificar y

pulsará el siguiente botón 

Formulario de modificación de usuarios


Código Usuario:	<input type="text" value="MARINA"/>
Nombre:	<input type="text" value="Marina"/>
Apellidos:	<input type="text" value="Lerma Gómez"/>
Nif:	<input type="text" value="46767531K"/>
Dirección:	<input type="text" value="c/Indibil 3"/>
Poblacion:	<input type="text" value="Barcelona"/>
Provincia:	<input type="text" value="Barcelona"/>
CP:	<input type="text" value="08004"/>
Teléfono fijo:	<input type="text" value="932894045"/>
Móvil:	<input type="text" value="626409850"/>
Email:	<input type="text" value="aniram_77@yahoo.es"/>
Perfil Usuario:	<input type="text" value="Administrador"/>
Anulado:	<input type="checkbox"/>
Modificar contraseña:	<input type="checkbox"/>
Todas las franquicias:	<input type="checkbox"/>
	<input type="text" value="MBAARAJAS"/> <input type="text" value="ZINDIBIL"/>

Si la modificación que se desea realizar es la de la contraseña, se pulsará sobre el check de "modificar contraseña". Al chequear aparecerá un edit que nos permitirá realizar esta operación

Código Usuario:	MARINA
Nombre:	Marina
Apellidos:	Lerma Gómez
Nif:	46767531K
Dirección:	c/Indibil 3
Poblacion:	Barcelona
Provincia:	Barcelona
CP:	08004
Teléfono fijo:	932894045
Móvil:	626409850
Email:	aniram_77@yahoo.es
Perfil Usuario:	Administrador
Anulado:	<input type="checkbox"/>
Modificar contraseña:	<input checked="" type="checkbox"/> ●●●●●●
Todas las franquicias:	<input type="checkbox"/> MBARAJAS ZINDIBIL
<input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>	

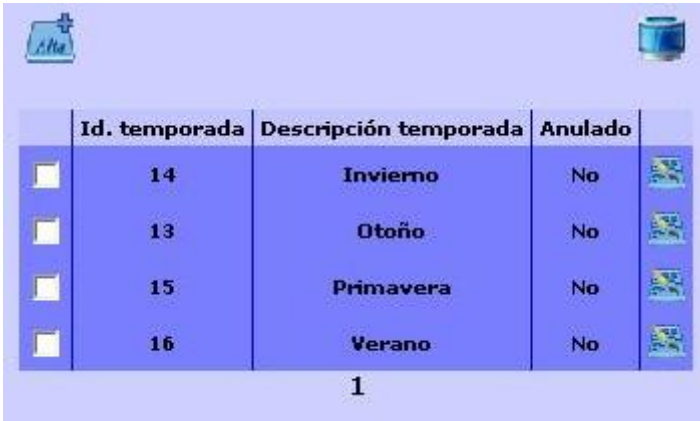
En este formulario el usuario podrá modificar todos los campos excepto el de código de usuario. Para ello sólo deberá posicionarse en el campo que desee modificar y realizar la misma, una vez acabado el cambio deberá acabar pulsando el botón de aceptar para que este registro se modifique en la base de datos.





6.2.4.5.3.- Eliminar Usuarios

El eliminar sigue el mismo proceso para todos los mantenimientos. El usuario debe seleccionar la fila o filas deseadas y proceder a eliminarlas pulsando .

6.2.4.6.- Mantenimiento de Temporadas

A continuación mostramos el contentFrame del mantenimiento de temporadas:



	Id. temporada	Descripción temporada	Anulado	
<input type="checkbox"/>	14	Invierno	No	
<input type="checkbox"/>	13	Otoño	No	
<input type="checkbox"/>	15	Primavera	No	
<input type="checkbox"/>	16	Verano	No	


1

Podemos comprobar que como la aplicación tiene datos de prueba, este listado aparece relleno con varios registros.

El listado muestra el "*Id de temporada*", la descripción de la temporada y el campo de Anulado. El listado de temporadas se encuentra ordenado por descripción de temporada.

6.2.4.6.1.- Alta de temporadas

A continuación el usuario se dispone a dar de alta temporadas en la base de datos.

Para dar de alta una temporada se debe pulsar el botón que realiza esta funcionalidad  y en la pantalla visualizaremos un nuevo formulario para poder realizar esta función.

Formulario de alta

Id. temporada:	<input type="text"/>
Descripción de temporada:	<input type="text"/>
Anulado	<input type="checkbox"/>

Una vez el usuario introduzca los datos el formulario aparecerá de la siguiente forma

Id. temporada:	<input type="text"/>
Descripción de temporada:	<input type="text" value="Temporada excepcional"/>
Anulado	<input type="checkbox"/>

Una vez rellenados los campos obligatorios del formulario, el usuario deberá apretar el botón de aceptar para que se inserten los datos nuevos en la base de datos.

6.2.4.6.2.- Modificar Temporadas

El usuario se encuentra en el listado de temporadas y decide modificar una de éstas. Para ello se colocará en la fila de la temporada que desea modificar y pulsará

el botón 


Formulario de modificación de temporadas



Id. temporada:	<input type="text" value="14"/>
Descripción de temporada:	<input type="text" value="Invierno"/>
Anulado	<input checked="" type="checkbox"/>

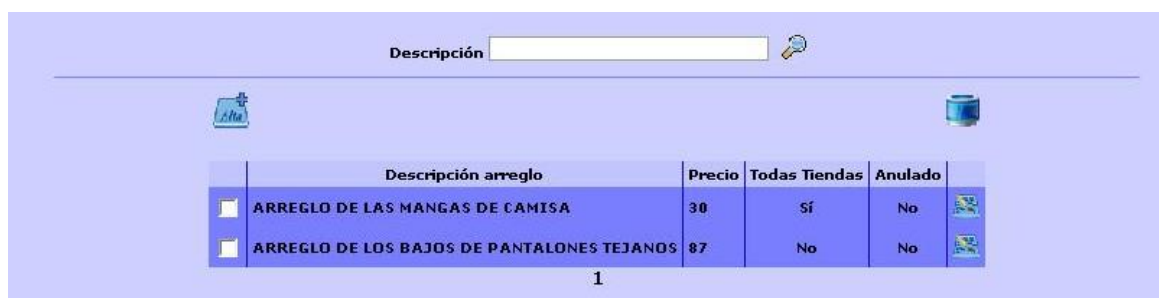
En este formulario el usuario podrá modificar todos los campos excepto el de id de temporada. Para ello sólo deberá posicionarse en el campo deseado y realizar la modificación, una vez realizada deberá acabar pulsando el botón de aceptar para que este cambio se realice en la base de datos.

6.2.4.6.3.- Eliminar temporadas



El eliminar sigue el mismo proceso para todos los mantenimientos. El usuario deberá seleccionar la fila o filas deseadas y proceder a la eliminación pulsando .

6.2.4.7.- Mantenimiento de Arreglos de modista

A continuación mostramos la imagen del contentFrame del mantenimiento de arreglos de modista:



The screenshot shows a web application interface with a light blue background. At the top, there is a search bar labeled "Descripción" with a magnifying glass icon. Below the search bar, there are two small icons: a blue square with a white plus sign and a blue cylinder. The main content is a table with the following data:

	Descripción arreglo	Precio	Todas Tiendas	Anulado	
<input type="checkbox"/>	ARREGLO DE LAS MANGAS DE CAMISA	30	Sí	No	
<input type="checkbox"/>	ARREGLO DE LOS BAJOS DE PANTALONES TEJANOS	87	No	No	


Below the table, the number "1" is centered, indicating the page number.

Podemos comprobar que como la aplicación tiene datos de prueba, este listado se muestra con varios registros.

El listado muestra la descripción del arreglo, el precio, y si es para todas las tiendas y el campo de Anulado. El listado de arreglos de modista se encuentra ordenado por descripción del arreglo.

6.2.4.7.1.- Alta de Arreglos de modista

A continuación el usuario se dispone a dar de alta arreglos de modista en la base de datos.

Para dar de alta un arreglo se debe pulsar el botón  y en la pantalla visualizaremos el siguiente formulario.

Formulario de alta

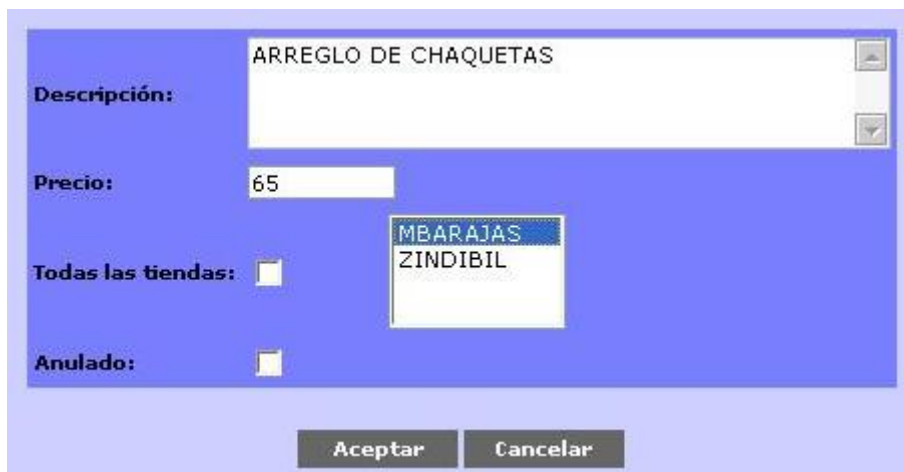


Formulario de alta de un arreglo de modista. El formulario tiene un fondo azul claro y contiene los siguientes campos:

- Descripción:** Un campo de texto grande y vacío.
- Precio:** Un campo de texto más pequeño y vacío.
- Todas las tiendas:** Un checkbox que está marcado con una 'x'.
- Anulado:** Un checkbox que no está marcado.

En la parte inferior del formulario hay dos botones: "Aceptar" y "Cancelar".

Una vez el usuario introduzca los datos el formulario aparecerá la siguiente imagen:



Formulario de alta de un arreglo de modista con datos introducidos. El formulario tiene un fondo azul claro y contiene los siguientes campos:


- Descripción:** Un campo de texto que contiene el texto "ARREGLO DE CHAQUETAS".
- Precio:** Un campo de texto que contiene el número "65".
- Todas las tiendas:** Un checkbox que no está marcado. Una lista desplegable está abierta, mostrando las opciones "MBARAJAS" y "ZINDIBIL".
- Anulado:** Un checkbox que no está marcado.

En la parte inferior del formulario hay dos botones: "Aceptar" y "Cancelar".

Si el usuario decide que el arreglo de modista no será para todas las tiendas, deberá desmarcar el check de "Todas las tiendas". Al realizar esa operación la aplicación mostrará un listado con todas las posibles tiendas. De este listado el usuario podrá escoger las que le convenga.

Una vez que el usuario haya rellenado los campos obligatorios del formulario de alta, deberá apretar el botón de aceptar para que estos se inserten en la base de datos.

6.2.4.7.2.- Modificar Arreglos de modista

El usuario se encuentra en el listado de arreglos de modista y decide modificar uno de los campos. Para ello se colocará en la fila de arreglos de modista que desea modificar y pulsará el siguiente botón 

Formulario de modificación de Arreglos de modista




The screenshot shows a form with a blue background. It contains the following fields and controls:

- Descripción:** A text input field containing the text "ARREGLO DE LAS MANGAS DE CAMISA".
- Precio:** A text input field containing the number "30".
- Todas las tiendas:** A checkbox that is checked.
- Anulado:** A checkbox that is unchecked.
- At the bottom, there are two buttons: "Aceptar" and "Cancelar".

En este formulario el usuario podrá modificar todos los campos. Una vez acabada la modificación deberá acabar pulsando el botón de aceptar para que se efectúen los cambios en la base de datos.

6.2.4.7.3.- Eliminar Arreglos de modista

El eliminar sigue el mismo proceso para todos los mantenimientos. El usuario deberá seleccionar la fila o filas deseada y proceder a eliminarlas pulsando .

6.2.4.8.- Mantenimiento de stock

A continuación mostramos la imagen del contentFrame del mantenimiento de stock:



Artículo	Tienda	Stock	Vendidos	Distribuidos	Con tara	En almacén
CAMISA	MBARAJAS	13	0	0	0	2
ART1	MBARAJAS	400	0	0	0	400
ART4	MBARAJAS	10	0	0	0	0
ART2	MBARAJAS	10	0	0	0	0
ART6	MBARAJAS	200	0	0	0	0
ART1	ZINDIBIL	200	2	2	2	2
CAMISA	ZINDIBIL	13	0	0	0	10
ART6	ZINDIBIL	500	0	0	0	0


1

Podemos comprobar que aparece un listado con varios registros. Esto es debido a que en la aplicación tenemos introducidos algunos datos de prueba.

El listado muestra el código de artículo, el código de la tienda, el stock de estos en cada tienda, el número de artículos vendidos, distribuidos, con tara y almacenados en la tienda. El listado de stock se encuentra ordenado por código de tienda.

6.2.4.8.1.- Alta de stock

A continuación el usuario se dispone a dar de alta stock en la base de datos.

Para dar de alta stock se debe pulsar el botón  y en la pantalla visualizaremos un nuevo formulario para poder realizar esta función.

Formulario de alta



Formulario de alta de stock vacío. El formulario tiene un fondo azul claro y contiene los siguientes campos de texto:

- Código Artículo:
- Código Tienda:
- Stock:
- Vendidos:
- Distribuidos:
- Con tara:
- En almacen:

En la parte inferior del formulario hay dos botones: "Aceptar" y "Cancelar".

Una vez el usuario ha rellenado todos los datos el formulario muestra la siguiente apariencia:



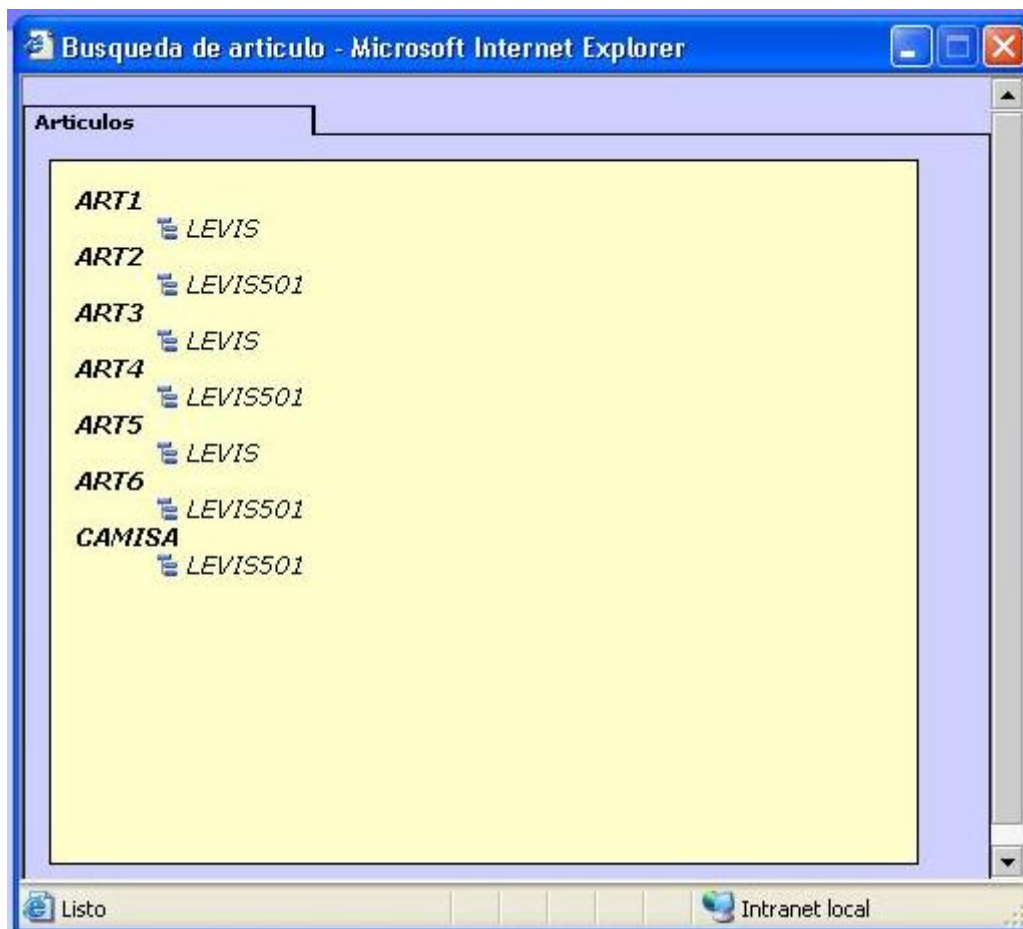
Formulario de alta de stock relleno. El formulario tiene un fondo azul claro y contiene los siguientes campos de texto:

- Código Artículo:
- Código Tienda:
- Stock:
- Vendidos:
- Distribuidos:
- Con tara:
- En almacen:


En la parte inferior del formulario hay dos botones: "Aceptar" y "Cancelar".

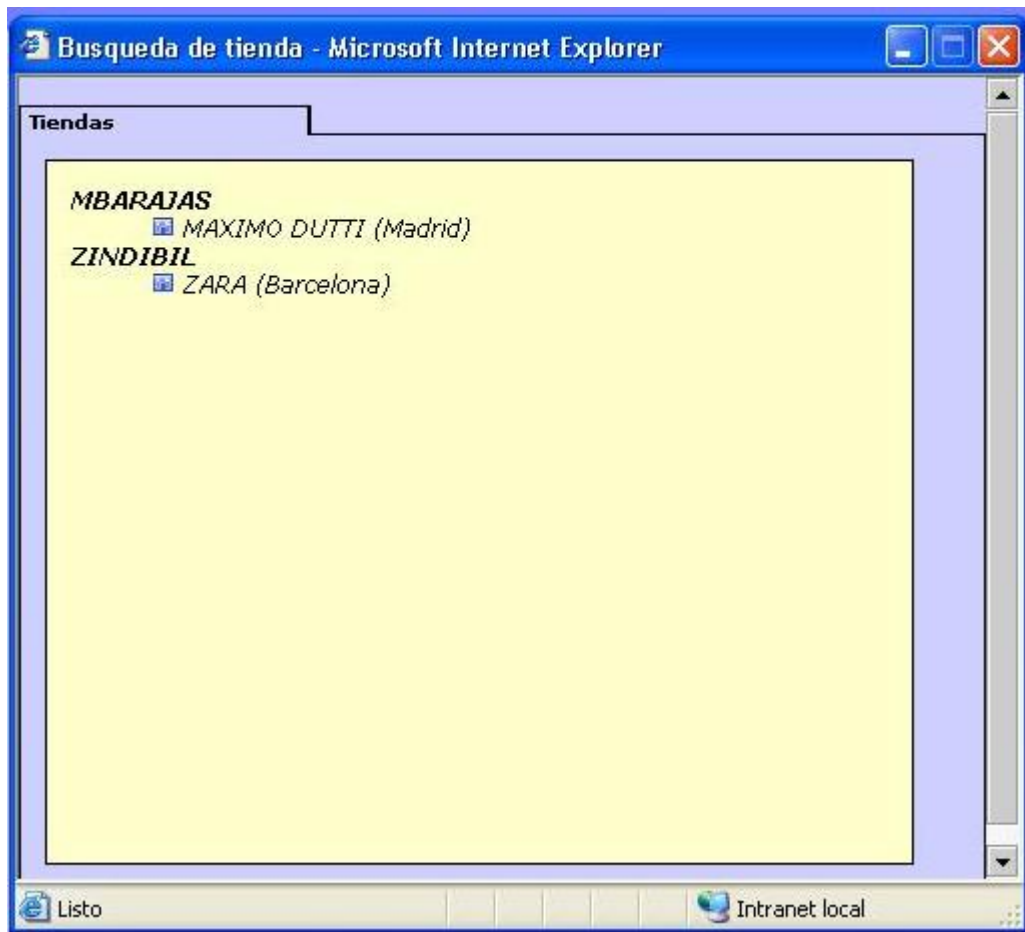
Para facilitar el trabajo de la búsqueda del código del artículo, la aplicación dispone de un listado de códigos de artículos, que se mostrará cuando el usuario pulse el botón





La información que se muestra en este formulario es el artículo y el modelo al que pertenece, una vez el usuario visualice el artículo deseado sólo tendrá que pulsar sobre éste, para que se seleccione, para este ejemplo el usuario ha seleccionado el artículo **ART1**

Algo parecido ocurre con las tiendas, si el usuario pulsa el botón de  aparecerá el siguiente listado que le permitirá seleccionar con facilidad la tienda a la cual quiere destinar el stock



Pulsando sobre la tienda deseada, esta quedará seleccionada en el formulario, para este ejemplo el usuario ha seleccionado **ZINDIBIL**. A parte del código de la tienda el listado nos muestra el nombre y la población.

Una vez que el usuario haya rellenado los campos obligatorios, se dispondrá a aceptar el formulario. Una vez aceptado se insertarán los nuevos campos en la base de datos

6.2.4.8.2.- Modificar Stock

El usuario que se encuentra en el listado de stock y decide modificar uno de éstos. Para ello se colocará en la fila del stock que desea modificar y pulsará el

siguiente botón 


Formulario de modificación de Stock



Código Artículo:	<input type="text" value="ART2"/>
Código Tienda:	<input type="text" value="MBARAJAS"/>
Stock:	<input type="text" value="10"/>
Vendidos:	<input type="text" value="0"/>
Distribuidos:	<input type="text" value="0"/>
Con tara:	<input type="text" value="0"/>
En almacén:	<input type="text" value="0"/>

En este formulario el usuario podrá modificar todos los campos excepto el de código de artículo y el código de tienda. Para ello sólo deberá posicionarse en el campo que desee modificar y realizar el cambio, una vez acabada la modificación deberá pulsar el botón de aceptar para que este se modifique en la base de datos.

6.2.4.8.3.- Eliminar Stock

El eliminar sigue el mismo proceso para todos los mantenimientos. El usuario debe seleccionar la fila o filas deseadas y proceder a eliminarlas pulsando .

7.- Conclusiones

Las conclusiones extraídas de este proyecto son muy positivas. Me ha encantado poder realizar un proyecto de esta envergadura, estando en todos los pasos de la realización, la especificación, el análisis, la implementación y el diseño de las pantallas.

También me ha ayudado a aprender otros lenguajes de programación como son el PHP, el HTML, el XML y el JavaScript. Menos del PHP de los demás habíamos hecho pequeñas pinceladas durante la carrera, pero siempre han sido parte de prácticas y nunca han llegado a ser una aplicación completa.

También he aprendido mucho en la parte de análisis y diseño de aplicaciones, en lenguaje UML y de cómo se tienen que separar los diferentes módulos para llegar a realizar una aplicación completa. Todo esto me ha enseñado que para poder hacer un diseño claro y fácil primero se debe realizar un completo y exhaustivo análisis.

En definitiva, yo buscaba ampliar mis conocimientos ya que siempre he pensado que la informática es un mundo que está en continuo desarrollo y por eso elegí realizar un proyecto que no estuviera relacionado con el trabajo que actualmente realizo y de esta manera seguir aprendiendo.

A continuación comentaré los objetivos conseguidos y cuales son las posibles ampliaciones y mejoras que se podrían realizar a la aplicación.

7.1.- Objetivos conseguidos

Los objetivos conseguidos han sido los marcados desde el principio conseguir realizar una aplicación que pueda gestionar el módulo de administración de una empresa dedicada al mundo de la moda y complementos. Desde un principio se ha intentado que la aplicación sea lo más intuitiva posible y fácil de utilizar, para que cualquier tipo de usuario puede sacar el máximo partido a la aplicación.

Todas las ideas han sido sacadas de información dada por un empresario que se dedica a gestionar sus productos fabricados desde fábrica para pasar más tarde a venderlos en sus franquicias. Se entiende que es una aplicación hecha para un cliente pero que es suficientemente general para que cualquier empresa dedicada al mundo de la moda y con varias tiendas o franquicias puedan utilizarla.

7.2.- Mejoras de la aplicación

Una vez que acabas la aplicación y te pones a hacer pruebas con ella te das cuenta de cuales podrían ser las mejoras y ampliaciones que se podrían dar para que esta aun fuera mas fácil de usar y ahorrara tiempo al usuario. Todas estas ampliaciones se pueden hacer con el mismo código, debido a la alta funcionalidad de este, ya que tiene una clara división de cada una de las capas y todo esta gestionado con funciones y con clases.

Me he quedado con las ganas de poder ver la aplicación finalizada totalmente con sus dos módulos e integrada en la empresa, para poder verla funcionar.

En un principio se pensó gestionar la base de datos con claves primarias que fueran los mismos códigos de los productos (familias, artículos, modelos,...) mientras se iba realizando la aplicación me di cuenta que sería mucho más fácil y que daría muchas más velocidad si se utilizaban índices auto numéricos como claves primarias de la base de datos. Pero esto no lo tuve en cuenta en el momento de las modificaciones de los productos, ya que al realizar la modificación no permito que el usuario pueda modificar el código manteniendo este campos deshabilitado, por lo tanto una de las mejoras sería esta.

Otra mejora sería el gestionar los anulados, he pensado que para todos aquellos productos que estén anulados, poder hacer una consulta a la base de datos, dejando todos estos productos fuera de los listados, y poder recuperarlos al realizar algún tipo de listado por anulados.

También me hubiera gustado gestionar el módulo para el perfil de usuario, ya que toda la aplicación y todas sus pantallas se encuentran gestionadas para el perfil de administrador. Creo que una manera de hacerlo hubiera sido no permitir realizar ciertas operaciones a un usuario con perfil usuario como puede ser evitar que este pueda eliminar productos. Hacer esta simple mejora en el programa es muy sencillo, porque sólo se debería deshabilitar los checks de los formularios junto con el botón de eliminar.

Otra posible mejora es poder permitir realizar copiados y pegados de los productos obligando a cambiar el código de éstos para que no se encuentren repetidos. Esto facilitaría y aumentaría la velocidad de introducir los datos para la persona encargada de esto.

También me hubiera gustado poder realizar la aplicación para que pudiera funcionar sobre HTTPS mediante certificados. Esto aumentaría la seguridad de la aplicación. También se debe estudiar la manera de proteger los directorios del sistema para ofrecer seguridad real.

Sería muy útil guardar imágenes de los artículos, y así poder ponerlos en la ficha de artículos.

8.- Agradecimientos

En este apartado me gustaría nombrar a toda la gente que ha estado continuamente apoyándome y exigiéndome para que acabara y presentara este proyecto. Es obvio que, gracias a ellos, he podido acabar el proyecto y así finalizar un ciclo que ya duraba muchos años.

Para poder finalizar el proyecto he necesitado del apoyo constante de Dani, mi pareja. Él ha sido mi apoyo continuo y la persona que siempre ha creído en mí para poder conseguir mi meta. No tengo palabras para expresar la gratitud que siento hacia él en estos momentos, así que simplemente diré que es del todo cierto que sin él no lo hubiera conseguido.

Me gustaría nombrar a mis padres que me han ayudado a que pudiera centrarme siempre en los estudios y que nunca me faltara de nada.

A todos mis amigos y compañeros de trabajo que siempre han sabido aconsejarme y me han estado apoyando en todo lo que ha estado en sus manos.

Y a mi director de proyecto Joan Sorribes que año tras año, me ha apoyado y me ha ayudado a poder llegar a finalizar mi carrera.

9.- Bibliografía

1. Para el desarrollo de PHP
 - Desarrollo Web con PHP y MySQL
 - Editorial: Anaya
 - Autores: Luke Welling y Laura Thomson
 - Ayuda proporcionada por PHP bajada de la página web
 - <http://www.php.net/manual/es/>
 - www.desarrolloweb.com
 - www.elgueroprogramador.com

2. Para el desarrollo Web
 - www.desarrolloweb.com
 - www.htmlhelp.com
 - www.w3.org
 - www.w3schools.com
 - www.webestilo.com
 - www.elguille.info

3. Para el diseño de la base de datos y enlaces con php
 - <http://www.mysql.com/>

4. Para realizar la documentación
 - <http://www.geocities.com/mailsoftware42/db/>
 - http://www.programacion.com/bbdd/tutorial/mysql_referencial/
 - <http://www.mysql-hispano.org/>
 - <http://www.geocities.com/mailsoftware42/db/>
 - <http://www.vivalinux.com.ar/article-2557.html>
 - <http://www.programacion.com/foros/28/msg/61633/>
 - <http://mx.grulic.org.ar/archiver/html/grulic/2004-02/msg00603.html>

Anexos

Anexo I:- Instalación del PHP

Installing PHP

Table of Contents

1. General Installation Considerations
2. Installation on Windows systems

Windows Installer

Manual Installation Steps

ActiveScript

Microsoft IIS / PWS

Apache 1.3.x on Microsoft Windows

Apache 2.0.x on Microsoft Windows

Sun, iPlanet and Netscape servers on Microsoft Windows

OmniHTTPd Server

Sambar Server on Microsoft Windows

Xitami on Microsoft Windows

Installation of extensions on Windows

3. Problems?

Read the FAQ

Other problems

Bug reports

4. Runtime Configuration

The configuration file

How to change configuration settings

Chapter 1. General Installation Considerations

Before starting the installation, first you need to know what do you want to use PHP for. There are three main fields you can use PHP, as described in the What can PHP do? section:

- Server-side scripting
- Command line scripting

- Client-side GUI applications

For the first and most common form, you need three things: PHP itself, a web server and a web browser. You probably already have a web browser, and depending on your operating system setup, you may also have a web server (e.g. Apache on Linux and MacOS X; IIS on Windows). You may also rent webspace at a company. This way, you don't need to set up anything on your own, only write your PHP scripts, upload it to the server you rent, and see the results in your browser.

While setting up the server and PHP on your own, you have two choices for the method of connecting PHP to the server. For many servers PHP has a direct module interface (also called SAPI). These servers include Apache, Microsoft Internet Information Server, Netscape and iPlanet servers. Many other servers have support for ISAPI, the

Microsoft module interface (OmniHTTPd for example). If PHP has no module support for your web server, you can always use it as a CGI or FastCGI processor. This means you set up your server to use the CGI executable of PHP to process all PHP file requests on the server.

If you are also interested to use PHP for command line scripting (e.g. write scripts autogenerating some pages for you offline, or processing text files depending on some arguments you pass to them), you always need the command line executable. For more information, read the section about writing command line PHP applications. In this case, you need no server and no browser.

With PHP you can also write desktop GUI applications using the PHP-GTK extension. This is a completely different approach than writing web pages, as you do not output any HTML, but manage windows and objects within them. For more information about PHP-GTK, please visit the site dedicated to this extension. PHP-GTK is not included in the official PHP distribution.

From now on, this section deals with setting up PHP for web servers on Unix and Windows with server module interfaces and CGI executables. You will also find information on the command line executable in the following sections.

PHP source code and binary distributions for Windows can be found at <http://www.php.net/downloads.php>. We recommend you to choose a mirror nearest to you for downloading the distributions.

Chapter 2. Installation on Windows systems

This section applies to Windows 98/Me and Windows NT/2000/XP/2003. PHP will not work on 16 bit platforms such as Windows 3.1 and sometimes we refer to the supported Windows platforms as Win32. Windows 95 is no longer supported as of PHP 4.3.0.

There are two main ways to install PHP for Windows: either manually or by using the installer. If you have Microsoft Visual Studio, you can also build PHP from the original source code.

Once you have PHP installed on your Windows system, you may also want to load various extensions for added functionality.

Warning

There are several all-in-one installers over the Internet, but none of those are endorsed by PHP.net, as we believe that the manual installation is the best choice to have your system secure and optimised.

Windows Installer

The Windows PHP installer is available from the downloads page at <http://www.php.net/downloads.php>. This installs the CGI version of PHP and for IIS, PWS, and Xitami, it configures the web server as well.

The installer does not include any extra external PHP extensions (php_*.dll) as you'll only find those in the Windows Zip Package and PECL downloads.

Note: While the Windows installer is an easy way to make PHP work, it is restricted in many aspects as, for example, the automatic setup of extensions is not supported. Use of the installer isn't the preferred method for installing PHP.

First, install your selected HTTP (web) server on your system, and make sure that it works.

Run the executable installer and follow the instructions provided by the installation wizard. Two types of installation are supported - standard, which provides sensible defaults for all the settings it can, and advanced, which asks questions as it goes along.

The installation wizard gathers enough information to set up the php.ini file, and configure certain web servers to use PHP. With IIS or PWS on a NT Workstation, a list of all the nodes on the server with script map settings is displayed, and you can choose those nodes to which you wish to add the PHP script mappings. One of the web servers the PHP installer does not configure for is Apache, so you'll need to configure it manually.

Once the installation has completed, the installer will inform you if you need to restart your system, restart the server, or just start using PHP.

Warning

Be aware, that this setup of PHP is not secure. If you would like to have a secure PHP setup, you'd better go on the manual way, and set every option carefully. This automatically working setup gives you an instantly working PHP installation, but it is not meant to be used on online servers.

Manual Installation Steps

This install guide will help you manually install and configure PHP with a web server on Microsoft Windows. To get started you'll need to download the zip binary distribution from the downloads page at <http://www.php.net/downloads.php>.

Although there are many all-in-one installation kits, and we also distribute a PHP installer for Microsoft Windows, we recommend you take the time to setup PHP yourself as this will provide you with a better understanding of the system, and enables you to install PHP extensions easily when needed.

Upgrading from a previous PHP version: Previous editions of the manual suggest moving various ini and DLL files into your SYSTEM (i.e. C:\WINDOWS) folder and while this simplifies the installation procedure it makes upgrading difficult. We advise you remove all of these files (like php.ini and PHP related DLLs from the Windows SYSTEM folder) before moving on with a new PHP installation. Be sure to backup these files as you might break the entire system.

The old php.ini might be useful in setting up the new PHP as well.

And as you'll soon learn, the preferred method for installing PHP is to keep all PHP related files in one directory and have this directory available to your systems PATH.

MDAC requirements: If you use Microsoft Windows 98/NT4 download the latest version of the Microsoft Data Access Components (MDAC) for your platform. MDAC is available at <http://msdn.microsoft.com/data/>. This requirement exists because ODBC is built into the distributed Windows binaries.

The following steps should be completed on all installations before any server specific instructions are performed:

Extract the distribution file into a directory of your choice. If you are installing PHP 4, extract to C:\, as the zip file expands to a foldername like php-4.3.7-Win32. If you are installing PHP 5, extract to C:\php

as the zip file doesn't expand as in PHP 4. You may choose a different location but do not have spaces in the path (like C:\Program Files\PHP) as some web servers will crash if you do.

The directory structure extracted from the zip is different for PHP versions 4 and 5 and look like as follows:

Example 2-1. PHP 4 package structure

```
c:\php
|
+--cli
| |
| |-php.exe      -- CLI executable - ONLY for commandline scripting
|
+--dlls          -- support DLLs required by some extensions
| |
| |-expat.dll
| |
| |-fdftk.dll
| |
| |-...
|
+--extensions    -- extension DLLs for PHP
| |
| |-php_bz2.dll
| |
| |-php_cpdf.dll
| |
| |-..
|
+--mibs          -- support files for SNMP
|
+--openssl       -- support files for Openssl
|
+--pdf-related   -- support files for PDF
|
+--sapi          -- SAPI (server module support) DLLs
| |
| |-php4activescript.dll
| |
| |-php4apache.dll
| |
| |-php4apache2.dll
| |
| |-..
|
+--PEAR          -- initial copy of PEAR
|
|
```

```

|-go-pear.bat      -- PEAR setup script
|
|..
|
|-php.exe          -- CGI executable
|
|..
|
|-php.ini-dist    -- default php.ini settings
|
|-php.ini-recommended -- recommended php.ini settings
|
|-php4ts.dll      -- core PHP DLL
|
|-...

```

Or:

Example 2-2. PHP 5 package structure

```

c:\php
|
+--dev
| |
| |-php5ts.lib
|
+--ext      -- extension DLLs for PHP
| |
| |-php_bz2.dll
| |
| |-php_cpdf.dll
| |
| |-..
|
+--extras
| |
| +--mibs      -- support files for SNMP
| |
| +--openssl  -- support files for Openssl
| |
| +--pdf-related -- support files for PDF
| |
| |-mime.magic
|
+--pear      -- initial copy of PEAR
|
|
|-go-pear.bat -- PEAR setup script

```



```

|
|-fdftk.dll
|
|...
|
|-php-cgi.exe      -- CGI executable
|
|-php-win.exe     -- executes scripts without an opened command prompt
|
|-php.exe         -- CLI executable - ONLY for command line scripting
|
|...
|
|-php.ini-dist    -- default php.ini settings
|
|-php.ini-recommended -- recommended php.ini settings
|
|-php5activescript.dll
|
|-php5apache.dll
|
|-php5apache2.dll
|
|...
|
|-php5ts.dll      -- core PHP DLL
|
|...

```

Notice the differences and similarities. Both PHP 4 and PHP 5 have a CGI executable, a CLI executable, and server modules, but they are located in different folders and/or have different names. While PHP 4 packages have the server modules in the sapi folder, PHP 5 distributions have no such directory and instead they're in the PHP folder root. The supporting DLLs for the PHP 5 extensions are also not in a separate directory.

Note: In PHP 4, you should move all files located in the dll and sapi folders to the main folder (e.g. C:\php).

Here is a list of server modules shipped with PHP 4 and PHP 5:

- sapi/php4activescript.dll (php5activescript.dll) - ActiveScript engine, allowing you to embed PHP in your Windows applications.
- sapi/php4apache.dll (php5apache.dll) - Apache 1.3.x module.
- sapi/php4apache2.dll (php5apache2.dll) - Apache 2.0.x module.
- sapi/php4isapi.dll (php5isapi.dll) - ISAPI Module for ISAPI compliant web servers like IIS 4.0/PWS 4.0 or newer.
- sapi/php4nsapi.dll (php5nsapi.dll) - Sun/iPlanet/Netscape server module.
- sapi/php4pi3web.dll (no equivalent in PHP 5) - Pi3Web server module.

Server modules provide significantly better performance and additional functionality compared to the CGI binary. The CLI version is designed to let you use PHP for command line scripting. More information about CLI is available in the chapter about using PHP from the command line.

Warning

The SAPI modules have been significantly improved as of the 4.1 release, however, in older systems you may encounter server errors or other server modules failing, such as ASP.

The CGI and CLI binaries, and the web server modules all require the php4ts.dll (php5ts.dll) file to be available to them. You have to make sure that this file can be found by your PHP installation. The search order for this DLL is as follows:

- The same directory from where php.exe is called, or in case you use a SAPI module, the web server's directory (e.g. C:\ProgramFiles\Apache Group\Apache2\bin).
- Any directory in your Windows PATH environment variable.

To make php4ts.dll / php5ts.dll available you have three options: copy the file to the Windows system directory, copy the file to the web server's directory, or add your PHP directory, C:\php to the PATH. For better maintenance, we advise you to follow the last option, add C:\php to the PATH, because it will be simpler to upgrade PHP in the future. Read more about how to add your PHP directory to PATH in the corresponding FAQ entry.

The next step is to set up a valid configuration file for PHP, php.ini. There are two ini files distributed in the zip file, php.ini-dist and php.ini-recommended. We advise you to use php.ini-recommended, because we optimized the default settings in this file for performance, and security. Read this well documented file carefully because it has changes from php.ini-dist that will drastically affect your setup. Some examples are display_errors being off and magic_quotes_gpc being off. In addition to reading these, study the ini settings and set every element manually yourself. If you would like to achieve the best security, then this is the way for you, although PHP works fine with these default ini files. Copy your chosen ini-file to a directory that PHP is able to find and rename it to php.ini. PHP searches for php.ini in the following locations (in order):

- PHPIniDir directive (Apache 2 module only)
- HKEY_LOCAL_MACHINE\SOFTWARE\PHP\IniFilePath
- The PHPRC environment variable
- Directory of PHP (for CLI), or the web server's directory (for SAPI modules)
- Windows directory (C:\windows or C:\winnt)

If you are running Apache 2, the simpler option is to use the PHPIniDir directive (read the installation on Apache 2 page), otherwise your best option is to set the PHPRC environment variable. This process is explained in the following FAQ entry.

Note: If you're using NTFS on Windows NT, 2000, XP or 2003, make sure that the user running the web server has read permissions to your php.ini (e.g. make it readable by Everyone).

The following steps are optional:

- Edit your new php.ini file. If you plan to use OmniHTTPd, do not follow the next step. Set the doc_root to point to your web servers document_root. For example:

```
doc_root = c:\inetpub // for IIS/PWS
```

```
doc_root = c:\apache\htdocs // for Apache
```

- Choose the extensions you would like to load when PHP starts. See the section about Windows extensions, about how to set up one, and what is already built in. Note that on a new installation it is advisable to first get PHP working and tested without any extensions before enabling them in php.ini.* On PWS and IIS, you can set the browscap configuration setting to point to: c:\windows\system\inetrv\browscap.ini on Windows 9x/Me,

c:\winnt\system32\inetrv\browscap.ini on NT/2000, and
c:\windows\system32\inetrv\browscap.ini on XP. For an up-to-date browscap.ini, read the
following FAQ.

PHP is now setup on your system. The next step is to choose a web server, and enable it to run PHP.
Choose a webserver from the table of contents.

ActiveScript

This section contains notes specific to the ActiveScript installation.

ActiveScript is a windows only SAPI that enables you to use PHP script in any ActiveScript compliant
host, like Windows Script Host, ASP/ASP.NET, Windows Script Components or Microsoft Scriptlet control.

As of PHP 5.0.1, ActiveScript has been moved to the PECL repository. You may download this PECL
extensions DLL from the PHP Downloads page or at <http://snaps.php.net/>.

Note: You should read the manual installation steps first!

After installing PHP, you should download the ActiveScript DLL (php5activescript.dll) and place it in
the main PHP folder (e.g. C:\php).

After having all the files needed, you must register the DLL on your system. To achieve this, open a
Command Prompt window (located in the Start Menu). Then go to your PHP directory by typing something
like `cd C:\php`. To register the DLL just type `regsvr32 php5activescript.dll`.

To test if ActiveScript is working, create a new file, named test.wsf (the extension is very important)
and type:

```
<job id="test">
<script language="PHPScript">
$WScript->Echo("Hello World!");
</script>
</job>
```

Save and double-click on the file. If you receive a little window saying "Hello World!" you're done.

Note: ActiveScript doesn't use the default php.ini file. Instead, it will look only in the same directory
as the .exe that caused it to load. You should create php-activescript.ini and place it in that folder, if you
wish to load extensions, etc.

Microsoft IIS / PWS

This section contains notes and hints specific to IIS (Microsoft Internet Information Server). We have
included installation instructions for PWS/IIS 3, PWS 4 or newer and IIS 4 or newer versions.

Important for CGI users: Read the `faq on cgi.force_redirect` for important details. This directive needs
to be set to 0.

Warning

By using the CGI setup, your server is open to several possible attacks. Please read our CGI security
section to learn how to defend yourself from those attacks.

Windows and PWS/IIS 3

The recommended method for configuring these servers is to use the REG file included with the
distribution (`pws-php4cgi.reg` in the SAPI folder for PHP 4, or `pws-php5cgi.reg` in the main folder for PHP
5). You may want to edit this file and make sure the extensions and PHP install directories match your
configuration. Or you can follow the steps below to do it manually.

Warning

These steps involve working directly with the Windows registry. One error here can leave your system in an unstable state. We highly recommend that you back up your registry first. The PHP Development team will not be held responsible if you damage your registry.

- Run Regedit.
- Navigate to: HKEY_LOCAL_MACHINE /System /CurrentControlSet /Services /W3Svc /Parameters /ScriptMap.
- On the edit menu select: New->String Value.
- Type in the extension you wish to use for your php scripts. For example .php
- Double click on the new string value and enter the path to php.exe in the value data field. ex: C:\php\php.exe for PHP 4, or C:\php\php-cgi.exe for PHP 5.
- Repeat these steps for each extension you wish to associate with PHP scripts.

The following steps do not affect the web server installation and only apply if you want your PHP scripts to be executed when they are run from the command line (ex. run C:\myscripts\test.php) or by double clicking on them in a directory viewer window. You may wish to skip these steps as you might prefer the PHP files to load into a text editor when you double click on them.

- Navigate to: HKEY_CLASSES_ROOT
- On the edit menu select: New->Key.
- Name the key to the extension you setup in the previous section. ex: .php
- Highlight the new key and in the right side pane, double click the "default value" and enter phpfile.
- Repeat the last step for each extension you set up in the previous section.
- Now create another New->Key under HKEY_CLASSES_ROOT and name it phpfile.
- Highlight the new key phpfile and in the right side pane, double click the "default value" and enter PHP Script.
- Right click on the phpfile key and select New->Key, name it Shell.
- Right click on the Shell key and select New->Key, name it open.
- Right click on the open key and select New->Key, name it command.
- Highlight the new key command and in the right side pane, double click the "default value" and enter the path to php.exe. ex: c:\php\php.exe -q %1. (don't forget the %1).
- Exit Regedit.
- If using PWS on Windows, reboot to reload the registry.

PWS and IIS 3 users now have a fully operational system. IIS 3 users can use a nifty tool from Steven Genusa to configure their script maps.

Windows and PWS 4 or newer

When installing PHP on Windows with PWS 4 or newer version, you have two options. One to set up the PHP CGI binary, the other is to use the ISAPI module DLL.

If you choose the CGI binary, do the following:

- Edit the enclosed pws-php4cgi.reg / pws-php5cgi.reg file (look into the SAPI folder for PHP 4, or in the main folder for PHP 5) to reflect the location of your php.exe / php-cgi.exe. Backslashes should be escaped, for example:
- [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w3svc\parameters\Script Map] ".php"="C:\php\php.exe" (change to C:\php\php-cgi.exe if you are using PHP 5) Now merge this registry file into your system; you may do this by double-clicking it.
- In the PWS Manager, right click on a given directory you want to add PHP support to, and select Properties. Check the 'Execute' checkbox, and confirm.
- If you choose the ISAPI module, do the following:

- Edit the enclosed pws-php4isapi.reg / pws-php5isapi.reg file (look into the SAPI folder for PHP 4, or in the main folder for PHP 5) to reflect the location of your php4isapi.dll / php5isapi.dll.
- Backslashes should be escaped, for example:
- [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w3svc\parameters\Script Map] ".php"="C:\\php\\sapi\\php4isapi.dll" (or C:\\php\\php5isapi.dll for PHP 5) Now merge this registry file into your system; you may do this by double-clicking it.
- In the PWS Manager, right click on a given directory you want to add PHP support to, and select Properties. Check the 'Execute' checkbox, and confirm.

Windows NT/2000/XP and IIS 4 or newer

To install PHP on an NT/2000/XP Server running IIS 4 or newer, follow these instructions. You have two options to set up PHP, using the CGI binary (php.exe in PHP 4, or php-cgi.exe in PHP 5) or with the ISAPI module.

In either case, you need to start the Microsoft Management Console (may appear as 'Internet Services Manager', either in your Windows NT 4.0 Option Pack branch or the Control Panel=>Administrative Tools under Windows 2000/XP). Then right click on your Web server node (this will most probably appear as 'Default Web Server'), and select 'Properties'.

If you want to use the CGI binary, do the following:

- Under 'Home Directory', 'Virtual Directory', or 'Directory', click on the 'Configuration' button, and then enter the App Mappings tab.
- Click Add, and in the Executable box, type: C:\php\php.exe for PHP4 or C:\php\php-cgi.exe for PHP 5 (assuming that you have unzipped PHP in c:\php\).
- In the Extension box, type the file name extension you want associated with PHP scripts. Leave 'Method exclusions' blank, and check the 'Script engine' checkbox. You may also like to check the 'check that file exists' box - for a small performance penalty, IIS (or PWS) will check that the script file exists and sort out authentication before firing up PHP. This means that you will get sensible 404 style error messages instead of CGI errors complaining that PHP did not output any data. You must start over from the previous step for each extension you want associated with PHP scripts. .php and .phtml are common, although .php3 may be required for legacy applications.
- Set up the appropriate security. (This is done in Internet Service Manager), and if your NT Server uses NTFS file system, add execute rights for I_USR_ to the directory that contains php.exe /php-cgi.exe.

To use the ISAPI module, do the following:

- If you don't want to perform HTTP Authentication using PHP, you can (and should) skip this step. Under ISAPI Filters, add a new ISAPI filter. Use PHP as the filter name, and supply a path to the php4isapi.dll / php5isapi.dll.
- Under 'Home Directory', click on the 'Configuration' button. Add a new entry to the Application Mappings. Use the path to the php4isapi.dll / php5isapi.dll as the Executable, supply .php as the extension, leave 'Method exclusions' blank, and check the 'Script engine' checkbox.
- Stop IIS completely (NET STOP iisadmin)
- Start IIS again (NET START w3svc)

Apache 1.3.x on Microsoft Windows

This section contains notes and hints specific to Apache 1.3.x installs of PHP on Microsoft Windows systems. We also have instructions and notes for Apache 2 on a separate page.

Note: You should read the manual installation steps first!

There are two ways to set up PHP to work with Apache 1.3.x on Windows. One is to use the CGI binary (php.exe for PHP 4 and php-cgi.exe for PHP 5), the other is to use the Apache module DLL. In either case you need to edit your httpd.conf to configure Apache to work with PHP, and then restart the server.

It is worth noting here that now the SAPI module has been made more stable under Windows, we recommend it's use above the CGI binary, since it is more transparent and secure.

Although there can be a few variations of configuring PHP under Apache, these are simple enough to be used by the newcomer. Please consult the Apache Documentation for further configuration directives.

After changing the configuration file, remember to restart the server, for example, NET STOP APACHE followed by NET START APACHE, if you run Apache as a Windows Service, or use your regular shortcuts.

Note: Remember that when adding path values in the Apache configuration files on Windows, all backslashes such as c:\directory\file.ext must be converted to forward slashes, as c:/directory/file.ext.

Installing as a CGI binary

If you unzipped the PHP package to C:\php\ as described in the Manual Installation Steps section, you need to insert these lines to your Apache configuration file to set up the CGI binary:

Example 2-3. PHP and Apache 1.3.x as CGI

```
ScriptAlias /php/ "c:/php/"
AddType application/x-httpd-php .php
# For PHP 4
Action application/x-httpd-php "/php/php.exe"
# For PHP 5
Action application/x-httpd-php "/php/php-cgi.exe"
# specify the directory where php.ini is
SetEnv PHPRC C:/php
```

Note that the second line in the list above can be found in the actual versions of httpd.conf, but it is commented out. Remember also to substitute the c:/php/ for your actual path to PHP.

Warning

By using the CGI setup, your server is open to several possible attacks. Please read our CGI security section to learn how to defend yourself from those attacks.

If you would like to present PHP source files syntax highlighted, there is no such convenient option as with the module version of PHP. If you chose to configure Apache to use PHP as a CGI binary, you will need to use the show_source() function. To do this simply create a PHP script file and add this code: <?php show_source("original_php_script.php"); ?>. Substitute original_php_script.php with the name of the file you wish to show the source of.

Installing as an Apache module

You should add the following lines to your Apache httpd.conf file:

```
Example 2-4. PHP as an Apache 1.3.x module
AddType application/x-httpd-php .php
# For PHP 4
LoadModule php4_module "c:/php/sapi/php4apache.dll"
# For PHP 5
LoadModule php5_module "c:/php/php5apache.dll"
```

```
# specify the directory where php.ini is
```

```
SetEnv PHPRC C:/php
```

You may find after using the Windows installer for Apache that you need to define the `AddModule` directive for `mod_php4.c`. This is especially important if the `ClearModuleList` directive is defined, which you will find by scrolling down a few lines. You will see a list of `AddModule` entries, add the following line at the end of the list:

```
AddModule mod_php4.c. For PHP 5, instead use AddModule mod_php5.c
```

If you would like to use the source code highlighting feature, you need to add the following line to your `httpd.conf`: `AddType application/x-httpd-php-source .phps`. This should be inserted at the same place where you inserted `AddType application/x-httpd-php .php` above). With this setup, all files served with the `.phps` extension will be syntax highlighted for the browser.

Apache 2.0.x on Microsoft Windows

This section contains notes and hints specific to Apache 2.0.x installs of PHP on Microsoft Windows systems. We also have instructions and notes for Apache 1.3.x users on a separate page.

Note: You should read the manual installation steps first!

Warning

Do not use Apache 2.0.x and PHP in a production environment neither on Unix nor on Windows. For information on why, read the following FAQ entry

You are highly encouraged to take a look at the Apache Documentation to get a basic understanding of the Apache 2.0.x Server. Also consider to read the Windows specific notes for Apache 2.0.x before reading on here.

PHP and Apache 2.0.x compatibility notes: The following versions of PHP are known to work with the most recent version of Apache 2.0.x:

- PHP 4.3.0 or later available at <http://www.php.net/downloads.php>.
- the latest stable development version. Get the source code <http://snaps.php.net/php4-latest.tar.gz> or download binaries for Windows <http://snaps.php.net/win32/php4-win32-latest.zip>.
- a prerelease version downloadable from <http://qa.php.net/>.
- you have always the option to obtain PHP through anonymous CVS.

These versions of PHP are compatible to Apache 2.0.40 and later.

Apache 2.0 SAPI-support started with PHP 4.2.0. PHP 4.2.3 works with Apache 2.0.39, don't use any other version of Apache with PHP 4.2.3. However, the recommended setup is to use PHP 4.3.0 or later with the most recent version of Apache2.

All mentioned versions of PHP will work still with Apache 1.3.x.

Warning

Apache 2.0.x is designed to run on Windows NT 4.0, Windows 2000 or Windows XP. At this time, support for Windows 9x is incomplete. Apache 2.0.x is not expected to work on those platforms at this time.

Download the most recent version of Apache 2.0.x and a fitting PHP version. Follow the Manual Installation Steps and come back to go on with the integration of PHP and Apache.

There are two ways to set up PHP to work with Apache 2.0.x on Windows. One is to use the CGI binary the other is to use the Apache module DLL. In either case you need to edit your `httpd.conf` to configure Apache to work with PHP and then restart the server.

Note: Remember that when adding path values in the Apache configuration files on Windows, all backslashes such as `c:\directory\file.ext` must be converted to forward slashes, as `c:/directory/file.ext`.

Installing as a CGI binary

You need to insert these three lines to your Apache httpd.conf configuration file to set up the CGI binary:

Example 2-5. PHP and Apache 2.0 as CGI ScriptAlias /php/ "c:/php/" AddType application/x-httpd-php .php

```
# For PHP 4
Action application/x-httpd-php "/php/php.exe"
# For PHP 5
Action application/x-httpd-php "/php/php-cgi.exe"
```

Warning

By using the CGI setup, your server is open to several possible attacks. Please read our CGI security section to learn how to defend yourself from those attacks.

Installing as an Apache module

You need to insert these two lines to your Apache httpd.conf configuration file to set up the PHP module for Apache 2.0:

Example 2-6. PHP and Apache 2.0 as Module

```
# For PHP 4 do something like this:
LoadModule php4_module "c:/php/sapi/php4apache2.dll" AddType application/x-httpd-php .php
# For PHP 5 do something like this:
LoadModule php5_module "c:/php/php5apache2.dll" AddType application/x-httpd-php .php
# configure the path to php.ini
PHPIniDir "C:/php"
```

Note: Remember to substitute the c:/php/ for your actual path to PHP in the above examples. Take care to use either php4apache2.dll or php5apache2.dll in your LoadModule directive and not php4apache.dll or php5apache.dll as the latter ones are designed to run with Apache 1.3.x.

Warning

Don't mix up your installation with DLL files from different PHP versions. You have the only choice to use the DLL's and extensions that ship with your downloaded PHP version.

Sun, iPlanet and Netscape servers on Microsoft Windows

This section contains notes and hints specific to Sun Java System Web Server, Sun ONE Web Server, iPlanet and Netscape server installs of PHP on Windows.

From PHP 4.3.3 you can use PHP scripts with the NSAPI module to generate custom directory listings and error pages. Additional functions for Apache compatibility are also available. For support in current web servers read the note about subrequests.

CGI setup on Sun, iPlanet and Netscape servers

To install PHP as a CGI handler, do the following:

- Copy php4ts.dll to your systemroot (the directory where you installed Windows)
- Make a file association from the command line. Type the following two lines:
- `assoc .php=PHPScript ftype PHPScript=c:\php\php.exe %1 %*`
- In the Netscape Enterprise Administration Server create a dummy shellcgi directory and remove it just after (this step creates 5 important lines in obj.conf and allow the web server to handle shellcgi scripts).

- In the Netscape Enterprise Administration Server create a new mime type (Category: type, Content-Type: magnus-internal/shellcgi, File Suffix: php).
- Do it for each web server instance you want PHP to run

More details about setting up PHP as a CGI executable can be found here:
<http://benoit.noss.free.fr/php/install-php.html>

NSAPI setup on Sun, iPlanet and Netscape servers

To install PHP with NSAPI, do the following:

- Copy php4ts.dll to your systemroot (the directory where you installed Windows)
- Make a file association from the command line. Type the following two lines:
- `assoc .php=PHPScript ftype PHPScript=c:\php\php.exe %1 %*`
- In the Netscape Enterprise Administration Server create a new mime type (Category: type, Content-Type: magnus-internal/x-httpd-php, File Suffix: php).
- Edit magnus.conf (for servers >= 6) or obj.conf (for servers < 6) and add the following: You should place the lines after mime types init.

```
Init          fn="load-modules"          funcs="php4_init,php4_execute,php4_auth_trans"
shlib="c:/php/sapi/php4nsapi.dll"
Init  fn="php4_init"    LateInit="yes"    errorString="Failed to initialise PHP!"
[php_ini="c:/path/to/php.ini"]
```

(PHP >= 4.3.3) The `php_ini` parameter is optional but with it you can place your `php.ini` in your webserver config directory.

- Configure the default object in `obj.conf` (for virtual server classes [Sun Web Server 6.0+] in their `vserver.obj.conf`): In the `<Object name="default">` section, place this line necessarily after all 'ObjectType' and before all 'AddLog' lines:
- Service `fn="php4_execute" type="magnus-internal/x-httpd-php" [inikey=value inikey=value ...]`
- (PHP >= 4.3.3) As additional parameters you can add some special `php.ini`-values, for example you can set a `docroot="/path/to/docroot"` specific to the context `php4_execute` is called. For boolean ini-keys please use 0/1 as value, not "On","Off",... (this will not work correctly), e.g. `zlib.output_compression=1` instead of `lib.output_compression="On"`
- This is only needed if you want to configure a directory that only consists of PHP scripts (same like a `cgi-bin` directory):
- `<Object name="x-httpd-php"> ObjectType fn="force-type" type="magnus-internal/x-httpd-php"`
- Service `fn=php4_execute [inikey=value inikey=value ...]`
- `</Object>`
- After that you can configure a directory in the Administration server and assign it the style `x-httpd-php`. All files in it will get executed as PHP. This is nice to hide PHP usage by renaming files to `.html`.
- Restart your web service and apply changes
- Do it for each web server instance you want PHP to run

Note: More details about setting up PHP as an NSAPI filter can be found here:
<http://benoit.noss.free.fr/php/install-php4.html>

Note: The stacksize that PHP uses depends on the configuration of the webserver. If you get crashes with very large PHP scripts, it is recommended to raise it with the Admin Server (in the section "MAGNUS EDITOR").

CGI environment and recommended modifications in php.ini

Important when writing PHP scripts is the fact that Sun JSWS/Sun ONE WS/iPlanet/Netscape is a multithreaded web server. Because of that all requests are running in the same process space (the space of the webserver itself) and this space has only one environment. If you want to get CGI variables like PATH_INFO, HTTP_HOST etc. it is not the correct way to try this in the old PHP 3.x way with getenv() or a similar way (register globals to environment, \$_ENV). You would only get the environment of the running webserver without any valid CGI variables!

Note: Why are there (invalid) CGI variables in the environment?

Answer: This is because you started the webserver process from the admin server which runs the startup script of the webserver, you wanted to start, as a CGI script (a CGI script inside of the admin server!). This is why the environment of the started webserver has some CGI environment variables in it. You can test this by starting the webserver not from the administration server. Use the command line as root user and start it manually - you will see there are no CGI-like environment variables.

Simply change your scripts to get CGI variables in the correct way for PHP 4.x by using the superglobal \$_SERVER. If you have older scripts which use \$HTTP_HOST, etc., you should turn on register_globals in php.ini and change the variable order too (important: remove "E" from it, because you do not need the environment here):

```
variables_order = "GPCS"
register_globals = On
```

Special use for error pages or self-made directory listings (PHP >= 4.3.3)

You can use PHP to generate the error pages for "404 Not Found" or similar. Add the following line to the object in obj.conf for every error page you want to overwrite:

```
Error fn="php4_execute" code=XXX script="/path/to/script.php" [inikey=value ini key=value...]
```

where XXX is the HTTP error code. Please delete any other Error directives which could interfere with yours. If you want to place a page for all errors that could exist, leave the code parameter out. Your script can get the HTTP status code with \$_SERVER['ERROR_TYPE'].

Another possibility is to generate self-made directory listings. Just create a PHP script which displays a directory listing and replace the corresponding default Service line for type="magnus-internal/directory" in obj.conf with the following:

```
Service fn="php4_execute" type="magnus-internal/directory" script="/path/to/script.php"
[inikey=value inikey=value...]
```

For both error and directory listing pages the original URI and translated URI are in the variables \$_SERVER['PATH_INFO'] and \$_SERVER['PATH_TRANSLATED'].

Note about nsapi_virtual() and subrequests (PHP >= 4.3.3)

The NSAPI module now supports the nsapi_virtual() function (alias: virtual()) to make subrequests on the webserver and insert the result in the webpage. The problem is, that this function uses some undocumented features from the NSAPI library.

Under Unix this is not a problem, because the module automatically looks for the needed functions and uses them if available. If not, nsapi_virtual() is disabled.

Under Windows limitations in the DLL handling need the use of a automatic detection of the most recent ns-httpdXX.dll file. This is tested for servers till version 6.1. If a newer version of the Sun server is used, the detection fails and nsapi_virtual() is disabled.

If this is the case, try the following: Add the following parameter to php4_init in magnus.conf/obj.conf:

Init `fn=php4_init ... server_lib="ns-httpdXX.dll"` where XX is the correct DLL version number. To get it, look in the server-root for the correct DLL name. The DLL with the biggest filesize is the right one.

You can check the status by using the `phpinfo()` function.

Note: But be warned: Support for `nsapi_virtual()` is EXPERIMENTAL!!!

OmniHTTPd Server

This section contains notes and hints specific to OmniHTTPd on Windows.

Note: You should read the manual installation steps first!

Warning

By using the CGI setup, your server is open to several possible attacks. Please read our CGI security section to learn how to defend yourself from those attacks.

You need to complete the following steps to make PHP work with OmniHTTPd. This is a CGI executable setup. SAPI is supported by OmniHTTPd, but some tests have shown that it is not so stable to use PHP as an ISAPI module.

Important for CGI users: Read the `faq` on `cgi.force_redirect` for important details. This directive needs to be set to 0.

1. Install OmniHTTPd server.
2. Right click on the blue OmniHTTPd icon in the system tray and select Properties
3. Click on Web Server Global Settings
4. On the 'External' tab, enter: `virtual = .php | actual = c:\php\php.exe` (use `php-cgi.exe` if installing PHP 5), and use the Add button.
5. On the Mime tab, enter: `virtual = wwwserver/stdcgi | actual = .php`, and use the Add button.
6. Click OK

Repeat steps 2 - 6 for each extension you want to associate with PHP.

Note: Some OmniHTTPd packages come with built in PHP support. You can choose at setup time to do a custom setup, and uncheck the PHP component. We recommend you to use the latest PHP binaries. Some OmniHTTPd servers come with PHP 4 beta distributions, so you should choose not to set up the built in support, but install your own. If the server is already on your machine, use the Replace button in Step 4 and 5 to set the new, correct information.

Sambar Server on Microsoft Windows

This section contains notes and hints specific to the Sambar Server for Windows.

Note: You should read the manual installation steps first!

This list describes how to set up the ISAPI module to work with the Sambar server on Windows.

- Find the file called `mappings.ini` (in the config directory) in the Sambar install directory.
- Open `mappings.ini` and add the following line under `[ISAPI]`:

Example 2-7. ISAPI configuration of Sambar

```
#for PHP 4
```

```
*.php = c:\php\php4isapi.dll
```

```
#for PHP 5
```

```
*.php = c:\php\php5isapi.dll
```

(This line assumes that PHP was installed in `c:\php`.)

* Now restart the Sambar server for the changes to take effect.

Xitami on Microsoft Windows

This section contains notes and hints specific to Xitami on Windows.

Note: You should read the manual installation steps first!

This list describes how to set up the PHP CGI binary to work with Xitami on Windows.

Important for CGI users: Read the faq on `cgi.force_redirect` for important details. This directive needs to be set to 0. If you want to use `$_SERVER['PHP_SELF']` you have to enable the `cgi.fix_pathinfo` directive.

Warning

By using the CGI setup, your server is open to several possible attacks. Please read our CGI security section to learn how to defend yourself from those attacks.

- Make sure the webserver is running, and point your browser to xitamias admin console (usually `http://127.0.0.1/admin`), and click on Configuration.
- Navigate to the Filters, and put the extension which PHP should parse (i.e. `.php`) into the field File extensions (`.xxx`).
- In Filter command or script put the path and name of your PHP CGI executable i.e. `C:\php\php.exe` for PHP 4, or `C:\php\php-cgi.exe` for PHP 5.
- Press the 'Save' icon.
- Restart the server to reflect changes.

Installation of extensions on Windows

After installing PHP and a webserver on Windows, you will probably want to install some extensions for added functionality. You can choose which extensions you would like to load when PHP starts by modifying your `php.ini`. You can also load a module dynamically in your script using `dl()`.

The DLLs for PHP extensions are prefixed with `php_`.

Note: In PHP 4.3.1 Bcmath, Calendar, COM, CType, FTP, MySQL, ODBC, Overload, PCRE, Session, Tokenizer, WDDX, XML and Zlib support is built in. You don't need to load any additional extensions in order to use these functions. See your distributions `README.txt` or `install.txt` or this table for a list of built in modules.

The default location PHP searches for extensions is `c:\php4\extensions` in PHP 4 and `c:\php5` in PHP 5. To change this setting to reflect your setup of PHP edit your `php.ini` file:

- You will need to change the `extension_dir` setting to point to the directory where your extensions lives, or where you have placed your `php_*.dll` files. Please do not forget the last backslash. For example: `extension_dir = c:/php/extensions/`
- Enable the extension(s) in `php.ini` you want to use by uncommenting the `extension=php_*.dll` lines in `php.ini`. This is done by deleting the leading `;` form the extension you want to load.

Example 2-8. Enable Bzip2 extension for PHP-Windows

```
// change the following line from ...;extension=php_bz2.dll
// ... to extension=php_bz2.dll
```

- Some of the extensions need extra DLLs to work. Couple of them can be found in the distribution package, in the `C:\php\dlls\` folder in PHP 4 or in the main folder in PHP 5, but some, for example Oracle (`php_oci8.dll`) require DLLs which are not bundled with the distribution package. If you are installing PHP 4, copy the bundled DLLs from `C:\php\dlls` folder to the main `C:\php` folder. Don't forget to include `C:\php` in the system PATH (this process is explained in a separate FAQ entry).

Note: If you are running a server module version of PHP remember to restart your webserver to reflect your changes to `php.ini`.

The following table describes some of the extensions available and required additional dlls.

Table 2-1. PHP Extensions

Extension Description Notes php_bz2.dll bzip2 compression functions None php_calendar.dll Calendar conversion functions Built in since PHP 4.0.3

php_cpdf.dll ClibPDF functions None

php_crack.dll Crack functions None

php_ctype.dll ctype family functions Built in since PHP 4.3.0

php_curl.dll CURL, Client URL library functions Requires: libeay32.dll, ssleay32.dll (bundled)

php_cybercash.dll Cybercash payment functions PHP <= 4.2.0

php_db.dll DBM functions Deprecated. Use DBA instead (php_dba.dll)

php_dba.dll DBA: DataBase (dbm-style) Abstraction layer functions None

php_dbase.dll dBase functions None

php_dbx.dll dbx functions

php_domxml.dll DOM XML functions PHP <= 4.2.0 requires: libxml2.dll (bundled) PHP >= 4.3.0 requires: iconv.dll (bundled)

php_dotnet.dll .NET functions PHP <= 4.1.1

php_exif.dll Read EXIF headers from JPEG None

php_fbsql.dll FrontBase functions PHP <= 4.2.0

php_fdf.dll FDF: Forms Data Format functions. Requires: fdfk.dll (bundled)

php_filepro.dll filePro functions Read-only access

php_ftp.dll FTP functions Built-in since PHP 4.0.3

php_gd.dll GD library image functions Removed in PHP 4.3.2. Also note that truecolor functions are not available in GD1, instead, use

php_gd2.dll.

php_gd2.dll GD library image functions GD2

php_gettext.dll Gettext functions PHP <= 4.2.0 requires

gnu_gettext.dll (bundled), PHP >= 4.2.3 requires libintl-1.dll, iconv.dll (bundled).

php_hyperwave.dll HyperWave functions None

php_iconv.dll ICONV charset conversion Requires: iconv-1.3.dll (bundled), PHP >=4.2.1 iconv.dll

php_ifx.dll Informix functions Requires: Informix libraries

php_iisfunc.dll IIS management functions None

php_imap.dll IMAP POP3 and NNTP functions None

php_ingres.dll Ingres II functions Requires: Ingres II libraries

php_interbase.dll InterBase functions Requires: gds32.dll (bundled)

php_java.dll Java functions PHP <= 4.0.6 requires: jvm.dll (bundled)

php_ldap.dll LDAP functions PHP <= 4.2.0 requires libsasl.dll (bundled), PHP >= 4.3.0 requires libeay32.dll, ssleay32.dll (bundled)

php_mbstring.dll Multi-Byte String functions None

php_mcrypt.dll Mcrypt Encryption functions Requires: libmcrypt.dll

php_mhash.dll Mhash functions PHP >= 4.3.0 requires: libmhash.dll (bundled)

php_mime_magic.dll Mimetype functions Requires: magic.mime (bundled)

php_ming.dll Ming functions for Flash None

php_msql.dll mSQL functions Requires: msql.dll (bundled)

php_mssql.dll MSSQL functions Requires: ntdbllib.dll (bundled)

php_mysql.dll MySQL functions PHP >= 5.0.0, requires libmysql.dll (bundled)

php_mysqli.dll MySQLi functions PHP >= 5.0.0, requires libmysqli.dll (bundled)

php_oci8.dll Oracle 8 functions Requires: Oracle 8.1+ client libraries

php_openssl.dll OpenSSL functions Requires: libeay32.dll (bundled)

php_oracle.dll Oracle functions Requires: Oracle 7 client libraries

php_overload.dll Object overloading functions Built in since PHP 4.3.0
php_pdf.dll PDF functions None
php_pgsql.dll PostgreSQL functions None
php_printer.dll Printer functions None
php_shmop.dll Shared Memory functions None
php_snmp.dll SNMP get and walk functions NT only!
php_soap.dll SOAP functions PHP >= 5.0.0
php_sockets.dll Socket functions None
php_sybase_ct.dll Sybase functions Requires: Sybase client libraries
php_tidy.dll Tidy functions PHP >= 5.0.0
php_tokenizer.dll Tokenizer functions Built in since PHP 4.3.0
php_w32api.dll W32api functions None
php_xmlrpc.dll XML-RPC functions PHP >= 4.2.1 requires: iconv.dll (bundled)
php_xslt.dll XSLT functions PHP <= 4.2.0 requires sablot.dll, expat.dll (bundled). PHP >= 4.2.1 requires sablot.dll, expat.dll, iconv.dll (bundled).
php_yaz.dll YAZ functions Requires: yaz.dll (bundled)
php_zip.dll Zip File functions Read only access
php_zlib.dll ZLib compression functions Built in since PHP 4.3.0

Chapter 3. Problems?

Read the FAQ

Some problems are more common than others. The most common ones are listed in the PHP FAQ, part of this manual.

Other problems

If you are still stuck, someone on the PHP installation mailing list may be able to help you. You should check out the archive first, in case someone already answered someone else who had the same problem as you. The archives are available from the support page on <http://www.php.net/support.php>. To subscribe to the PHP installation mailing list, send an empty mail to php-install-subscribe@lists.php.net. The mailing list address is php-install@lists.php.net.

If you want to get help on the mailing list, please try to be precise and give the necessary details about your environment (which operating system, what PHP version, what web server, if you are running PHP as CGI or a server module, safe mode, etc...), and preferably enough code to make others able to reproduce and test your problem.

Bug reports

If you think you have found a bug in PHP, please report it. The PHP developers probably don't know about it, and unless you report it, chances are it won't be fixed. You can report bugs using the bug-tracking system at <http://bugs.php.net/>. Please do not send bug reports in mailing list or personal letters. The bug system is also suitable to submit feature requests.

Read the How to report a bug document before submitting any bug reports!

Chapter 4. Runtime Configuration

The configuration file

The configuration file (called `php3.ini` in PHP 3, and simply `php.ini` as of PHP 4) is read when PHP starts up. For the server module versions of PHP, this happens only once when the web server is started. For the CGI and CLI version, it happens on every invocation.

The default location of `php.ini` is a compile time option (see the FAQ entry), but can be changed for the CGI and CLI version with the `-c` command line switch, see the chapter about using PHP from the command line. You can also use the environment variable `PHPRC` for an additional path to search for a `php.ini` file.

If `php-SAPI.ini` exists (where SAPI is used SAPI, so the filename is e.g. `php-cli.ini` or `php-apache.ini`), it's used instead of `php.ini`.

Note: The Apache web server changes the directory to root at startup causing PHP to attempt to read `php.ini` from the root filesystem if it exists.

The `php.ini` directives handled by extensions are documented respectively on the pages of the extensions themselves. The list of the core directives is available in the appendix. Probably not all the PHP directives are documented in the manual though. For a complete list of directives available in your PHP version, please read your well commented `php.ini` file. Alternatively, you may find the the latest `php.ini` from CVS helpful too.

Example 4-1. `php.ini` example; any text on a line after an unquoted semicolon (;) is ignored

[php] ; section markers (text within square brackets) are also ignored; Boolean values can be set to either:

```
; true, on, yes
; or false, off, no, none
register_globals = off
track_errors = yes
; you can enclose strings in double-quotes
include_path = ".:usr/local/lib/php"
; backslashes are treated the same as any other character
include_path = ".;c:\php\lib"
```

How to change configuration settings

Running PHP as an Apache module

When using PHP as an Apache module, you can also change the configuration settings using directives in Apache configuration files (e.g. `httpd.conf`) and `.htaccess` files. You will need "AllowOverride Options" or "AllowOverride All" privileges to do so.

With PHP 4 and PHP 5, there are several Apache directives that allow you to change the PHP configuration from within the Apache configuration files. For a listing of which directives are `PHP_INI_ALL`, `PHP_INI_PERDIR`, or `PHP_INI_SYSTEM`, have a look at the table found within the `ini_set()` documentation.

Note: With PHP 3, there are Apache directives that correspond to each configuration setting in the `php3.ini` name, except the name is prefixed by "php3_".

`php_value` name value Sets the value of the specified directive. Can be used only with `PHP_INI_ALL` and `PHP_INI_PERDIR` type directives. To clear a previously set value use `none` as the value.

Note: Don't use `php_value` to set boolean values. `php_flag` (see below) should be used instead.

`php_flag` name on|off Used to set a boolean configuration directive. Can be used only with `PHP_INI_ALL` and `PHP_INI_PERDIR` type directives.

`php_admin_value` name value Sets the value of the specified directive. This can not be used in `.htaccess` files. Any directive type set with `php_admin_value` can not be overridden by `.htaccess` or `virtualhost` directives.

To clear a previously set value use `none` as the value.

php_admin_flag name on|off

Used to set a boolean configuration directive. This can not be used in .htaccess files. Any directive type set with php_admin_flag can not be overridden by .htaccess or virtualhost directives.

Example 4-2. Apache configuration example

```
<IfModule mod_php5.c>
  php_value include_path ".:usr/local/lib/php"
  php_admin_flag safe_mode on
</IfModule>
<IfModule mod_php4.c>
  php_value include_path ".:usr/local/lib/php"
  php_admin_flag safe_mode on
</IfModule>
<IfModule mod_php3.c>
  php3_include_path ".:usr/local/lib/php"
  php3_safe_mode on
</IfModule>
```

Caution

PHP constants do not exist outside of PHP. For example, in httpd.conf you can not use PHP constants such as E_ALL or E_NOTICE to set the error_reporting directive as they will have no meaning and will evaluate to 0. Use the associated bitmask values instead. These constants can be used in php.ini

Changing PHP configuration via the Windows registry

When running PHP on Windows, the configuration values can be modified on a per-directory basis using the Windows registry. The configuration values are stored in the registry key HKLM\SOFTWARE\PHP\Per Directory Values, in the sub-keys corresponding to the path names. For example, configuration values for the directory c:\inetpub\wwwroot would be stored in the key HKLM\SOFTWARE\PHP\Per Directory Values\c:\inetpub\wwwroot. The settings for the directory would be active for any script running from this directory or any subdirectory of it. The values under the key should have the name of the PHP configuration directive and the string value. PHP constants in the values are not parsed.

Other interfaces to PHP

Regardless of how you run PHP, you can change certain values at runtime of your scripts through ini_set(). See the documentation on the ini_set() page for more information.

If you are interested in a complete list of configuration settings on your system with their current values, you can execute the phpinfo() function, and review the resulting page. You can also access the values of individual configuration directives at runtime using ini_get() or get_cfg_var().

Resumen de la memoria

Esta memoria ha sido realizada para dar a conocer el proyecto de título Gestión de productos de una empresa dedicada a la moda. Esta aplicación intentará poner en el mercado una solución para pequeñas empresas que quieren hacerse un puesto en el mundo de la moda y necesitan un software para poder gestionar sus tiendas. En este sector, existen pequeños empresarios que han empezado realizando las piezas de ropa en sus fábricas y han decidido hacer sus pequeñas colecciones y ponerlas a la venta al detalle en sus propias franquicias. Esta aplicación muestra un módulo de lo que sería un proyecto más grande. Este módulo es el que se encarga de gestionar los artículos desarrollados en la fábrica, para poder distribuirlos entre las tiendas. Una posible segunda parte del proyecto sería la gestión de las ventas en tiendas.

Aquesta memòria ha sigut realitzada per donar a conèixer el projecte de títol Gestió de productes d'una empresa dedicada a la moda. Aquesta aplicació intentarà introduir al mercat una solució per petites empreses que volen fer-se un lloc en el món de la moda i necessiten un software per poder gestionar les seves botigues. En aquest sector, existeixen petits empresaris que han començat a realitzar les peces de roba a les seves fàbriques i que han decidit fer petites coleccions y posar-les a la venda al detall, a les seves pròpies franquícies. Aquesta aplicació mostra un mòdul d'un projecte molt més gran. Aquest mòdul es l'encarregat de gestionar els articles creats a fàbrica, per poder distribuir-los entre les botigues. Un possible segon mòdul del projecte seria la gestió de les vendes a les botigues.

This memory has been made to bring to light the project named Products - Management of a company dedicated to Fashion. This application will put into the market a solution for small businesses that want to get positioned inside the fashion world and need a software to be able to manage their stores. In this sector there are small business-men that have begun with making the garments of clothes in their factories and have then decided to do their small collections and to sell them for the end client in their own franchising-stores. This application shows a module of what would be a larger project. This module is the one that takes charge of managing the articles developed inside the factory, to get then distributed among the stores. A possible second part of the project would be the sales management in the stores.