

Herramienta de Gestión

Integral de e-Business

Alumno: David López Martín

Consultor: Antoni Oller Arcas

Quería dejar expresada mi gratitud a aquellas personas que de una u otra manera me han ayudado a que a día de hoy pueda escribir estas líneas.

En primer lugar a Antoni Oller Arcas, por haberme guiado y ayudado a lo largo del proyecto, para que fuese capaz de lograr realizarlo de forma adecuada. Sin su ayuda, me habría sido imposible conseguir dicho objetivo. También quiero agradecer el apoyo a mi tutor Pere Manzanos González, el que no pocas veces me ha tenido que sacar de algún apuro burocrático, sin el que seguramente el tiempo que he tardado en llegar aquí habría sido bastante más cuantioso.

Por otro lado, no me puedo olvidar de mis amigos, con los cuales he “desconectado” de las preocupaciones relativas a los estudios o el trabajo en mi tiempo libre, algo que indudablemente me hacía mucho bien. Además, les estoy muy agradecido por su comprensión ya que en muchas ocasiones no pude cumplir con lo prometido debido a algún imprevisto relacionado con los estudios y, sin embargo, nunca hubo un mal gesto por su parte.

Por último, quiero dedicar unas líneas a mi familia. Sin lugar a dudas ellos son los responsables de que este documento exista, ya que me habría sido imposible llegar hasta aquí sin su apoyo. Cuando yo dudaba de si había elegido el camino correcto, ellos siempre tuvieron palabras de ánimo que hicieron que siguiese adelante. Cuando algo salía mal, ellos estaban ahí para recordarme aquellas cosas que sabían que me impulsarían a intentar superar los problemas. Quiero que tengáis claro que si he podido llegar hasta el final de este camino ha sido gracias a vosotros.

Contenido

Contenido	4
Licencia	7
Introducción	8
Contextualización	8
Descripción del proyecto	9
Modulo de gestión de inmuebles	9
Modulo de gestión de stock	9
Modulo de administración de la aplicación	9
Descripción del funcionamiento	11
Fundamentos Tecnológicos	12
El Lenguaje Java	12
Características	12
Struts	15
Spring	16
Hibernate	16
Motivación	17
Material	19
Eclipse	19
Microsoft Office Word 2007	19
Herramientas CASE: MagicDraw UML	19
OpenProj	20
MySQL	20
Requisitos funcionales	22
Casos de uso relacionados con el actor Sistemas	22
Casos de uso relacionados con el actor Gerencia	26
Casos de uso relacionados con el actor Mercado	28
Casos de uso relacionados con el actor Administrativo	30
Interfaz de usuario	32
Pantalla de acceso	32
Pantalla de Inicio	32
Pantalla de Gestión de Perfiles	33

Pantalla de Creación/Modificación de Perfiles _____	33
Pantalla de Gestión de Usuarios _____	34
Pantalla de Creación/Modificación de Usuarios _____	34
Pantalla de Gestión de Inmuebles _____	35
Pantalla de Creación/Modificación de Inmuebles _____	35
Pantalla de Gestión de Productos _____	36
Pantalla de Creación/Modificación de Productos _____	36
Pantalla de Gestión de Stock _____	37
Arquitectura de la Aplicación _____	38
Frameworks empleados en el desarrollo de la aplicación _____	39
Struts _____	39
Spring _____	39
Hibernate _____	40
Patrones de diseño utilizados _____	40
Struts _____	40
Spring _____	44
Hibernate _____	46
Estructura de paquetes de la aplicación _____	48
La capa de la lógica de negocio _____	50
Planificación _____	52
Manual de usuario _____	54
Página de Bienvenida _____	55
Pantalla de Consulta de Perfiles _____	55
Pantalla de Creación de Perfil _____	57
Pantalla de Modificación de Perfil _____	59
Pantalla de Consulta de Usuarios _____	61
Pantalla de Creación de Usuario _____	62
Pantalla de Modificación de Usuario _____	64
Pantalla de Consulta de Inmuebles _____	65
Pantalla de Creación de Inmueble _____	66
Pantalla de Modificación de Inmueble _____	68
Pantalla de Consulta de Productos _____	69
Pantalla de Creación de Producto _____	70

Pantalla de Modificación de Producto	72
Pantalla de Consulta de Stocks	73
Pantalla de Modificación de Stocks de Inmuebles	74
Pantalla de Modificación de Stocks de Productos	75
Conclusiones	77
Bibliografía	78

Licencia

Este TFC, la documentación y el código fuente de las aplicaciones que han sido desarrolladas queda licenciado bajo la licencia: Reconocimiento-NoComercial-CompartirIgual 3.0 España (CC BY-NC-SA 3.0).

Usted es libre de:

copiar, distribuir y comunicar públicamente la obra

Remezclar — transformar la obra

Bajo las condiciones siguientes:



Reconocimiento — Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciadador (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).



No comercial — No puede utilizar esta obra para fines comerciales.



Compartir bajo la misma licencia — Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

Más información en: <http://creativecommons.org/licenses/by-nc-sa/3.0/es/>

Introducción

A continuación expondré los objetivos concretos del presente proyecto, si bien creo conveniente un acercamiento previo tanto al dominio en el que va a ser utilizada la aplicación como a la situación actual que se encuentra el sector para entender la motivación de la creación de una herramienta como la propuesta.

Contextualización

Desde hace años se viene hablando, en todos los ámbitos, de un nuevo modelo de negocio basado en la web, ya que en dicha red existe un potencial mercado donde el usuario le da un mayor valor añadido a cuestiones como la comodidad y rapidez de uso, accesibilidad a los productos o a la rápida actualización de los mismos. Estos últimos años hemos ido viendo como en muchos países este tipo de negocio, no sólo se iba asentando, sino que en muchos casos, incluso arrebatava parte del mercado a los negocios de corte tradicional. Sin embargo, España parecía quedarse rezagada sin la aparición de grandes apuestas por ese tipo de negocio ya fuese por el asentamiento de las ya existentes en nuestro territorio o por la aparición de alguna desarrollada directamente por alguna empresa española.

Afortunadamente, estos últimos años se han dado los primeros pasos, con dos propuestas importantes, curiosamente cada una de ellas de uno de los tipos señalados anteriormente. Por un lado tenemos la aparición en Internet del nuevo portal de la multinacional textil con sede española, Inditext, con el que la compañía quiere dar una gran importancia al negocio en la red de redes dentro de su franquicia y con noticias de que aumentarán las funciones disponibles en el portal de forma que no solo sea un nuevo modelo de negocio, sino que sirva de sustento para el negocio tradicional. Por otro lado, recientemente se ha producido el desembarco en España ni más ni menos de Amazon, que no solo es que sea fácilmente la mayor empresa comercial de e-business, sino que además es un baluarte de dicho modelo de negocio. Por lo tanto, podemos decir que los primeros pasos dados, tienen la característica de ser pasos firmes, puesto que estamos hablando de que una de las empresas más importantes de España ha reconocido la necesidad de estar en la web con sus productos y, por otro lado, una de las principales compañías de e-business a nivel mundial ha visto que en España hay mercado suficiente como para que asentarse en ella sea rentable. Por lo tanto, viendo el comportamiento del mercado, seguramente poco tendremos que esperar para ver como otras compañías las siguen y la importancia del modelo de negocio de venta online cobra un peso específico en nuestro país.

Sin embargo, conforme dicho modelo de negocio cobra mayor peso, también supone un pequeño escollo para la pequeña y mediana empresa, puesto que los sistemas que se están implantando ahora mismo en España responden a las características de las grandes empresas que están dando el paso, y que no se adaptan a las necesidades de las empresas de naturaleza más modesta. Por eso, la propuesta de realizar una herramienta que permita la gestión de los distintos emplazamientos disponibles de la empresa, así como el stock disponible en cada uno de ellos, que sirva para en un futuro realizar un módulo que permita la venta online de dichos productos. En el siguiente punto se desarrollan los requisitos funcionales de la aplicación a desarrollar.

Descripción del proyecto

El objetivo del proyecto es llevar a cabo el desarrollo de una herramienta que permita a las pequeñas y medianas empresas informatizar su gestión de forma que un futuro salto al e-business pueda ser llevado a cabo sin suponer un proceso traumático. Para ello se desarrollará una herramienta que permita gestionar los distintos inmuebles pertenecientes a la empresa, así como el stock existente en cada uno de ellos.

Si bien, en el punto anterior ya se ha señalado de forma rápida el objetivo principal del proyecto tratado en este documento, creo adecuado señalar de forma detallada las distintas funcionalidades que debe aportar la aplicación al final de su desarrollo. Podemos destacar tres divisiones funcionales claras dentro de la aplicación objetivo:

- El módulo de gestión de inmuebles
- El módulo de gestión de stock
- El módulo de administración de la aplicación.

Una vez diferenciados esos tres bloques detallaremos que deben permitir realizar cada uno de ellos.

Modulo de gestión de inmuebles

Este modulo permitirá realizar una gestión sencilla de los locales pertenecientes a la empresa (o que trabajan a la misma), de esta forma deben permitir realizar las siguientes operaciones:

- Dar de alta un inmueble con todos los datos necesarios a nivel administrativo.
- Modificar los datos asociados a un inmueble
- Eliminar del sistema un inmueble creado con anterioridad, realizando las comprobaciones necesarias para ver si la operación se puede realizar (por ejemplo, un inmueble no se debería poder eliminar si tiene stock asociado)
- Busca inmuebles en el sistema, permitiendo distintos criterios de búsqueda.

Señalar que los datos asociados a un inmueble, además de los evidentes, como podría ser su dirección, deberá tener campos que permitan indicar si puede almacenar stock, si es de venta directa, etc.

Modulo de gestión de stock

Este modulo permitirá realizar una gestión simple del catálogo de productos existentes, por lo que tiene que permitir llevar a cabo las siguientes operaciones:

- Dar de alta un producto en el catalogo con todos los datos necesarios.
- Modificar los datos asociados a dicho producto.
- Eliminar del catalogo un producto creado anteriormente.
- Realizar búsquedas de los productos en el catalogo, permitiendo distintos criterios de búsqueda.

Modulo de administración de la aplicación

Este modulo permitirá realizar la administración de la aplicación, incluyendo el acceso a la misma y las acciones permitidas, por lo que debe proveer las siguientes funcionalidades:

- La gestión de perfiles dentro de la aplicación.
- Asignar/desasignar las acciones que pueden llevar a cabo dichos perfiles
- Realizar una gestión de las acciones.
- Dar de alta/baja usuarios de la aplicación, pudiendo asociarlos a uno o más perfiles de los datos de alta.

Además, habrá una serie de funcionalidades que surgirán de la integración de los distintos módulos en la aplicación final.

Por otro lado, el diseño de la aplicación debe llevarse a cabo teniendo presente que la aplicación debe ser fácilmente ampliable para que en un futuro permita el desarrollo de un nuevo módulo, que de forma integrada con el producto resultante del proyecto, se pueda llevar a cabo la venta online de los productos del catálogo.

Descripción del funcionamiento

A continuación describiremos de forma resumida el funcionamiento esperado por parte de la aplicación

- El encargado\los encargados de realizar la gestión de la herramienta darán de alta los distintos perfiles existentes en la herramienta de modo que le asocien a cada uno de ellos del listado de acciones disponibles, las que tienen permiso para realizar.
- Posteriormente, darán de alta en el sistema los distintos usuarios de la herramienta asociándolos a los perfiles que pueden realizar. De este modo, cada usuario podrá realizar aquellas acciones que estén asociadas a, al menos, uno de los perfiles que pueden desempeñar
- Los gerentes podrán realizar las operaciones necesarias de gestión de inmuebles, lo que implica que podrán darlos de alta en el sistema con toda la información necesaria, modificar dichos datos y darlos de baja de la herramienta. Será importante la opción de marcar que inmuebles pueden llevar a cabo ventas al usuario final y cuales únicamente pueden realizar tareas de almacenaje.
- Habrá un grupo de personas dentro de la empresa que serán los encargados de tomar las decisiones de que productos se comercializan y las características de los mismos. De este modo, la herramienta deberá permitirles gestionar los distintos productos a la venta en la herramienta, por lo que podrán crearlos en el sistema con las características necesarias, modificar la información de los mismos una vez dados de alta e incluso eliminarlos del sistema.
- Por último, habrá una serie de personas que se encargarán de gestionar el stock de cada uno de los productos, de este modo, podrán moverlo desde cada uno de los inmuebles a otro, así como dar nuevo stock de alta (aumentar el número de productos existentes dentro del sistema), así como darlo de baja (por distintos motivos venta, deterioro, pérdida...).

Fundamentos Tecnológicos

Una vez expuestos los objetivos del proyecto, a lo largo de este capítulo se expondrán las tecnologías que se utilizarán en el proyecto señalando sus principales características, para finalmente justificar la elección de dichos elementos recalcando las ventajas que las hacen adecuadas para este proyecto, frente a otras alternativas.

El Lenguaje Java

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 1990. Las aplicaciones Java están típicamente compiladas en un *bytecode*, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el *bytecode* es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del *bytecode* por un procesador Java también es posible. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel como punteros.

Características

Las características principales que nos ofrece Java respecto a cualquier otro lenguaje de programación, serían:

Simple

Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de éstos. C++ no es un lenguaje conveniente por razones de seguridad, pero C y C++ son los lenguajes más difundidos, por ello Java se diseñó para ser parecido a C++ y así facilitar un rápido y fácil aprendizaje.

Java elimina muchas de las características de otros lenguajes como C++, para mantener reducidas las especificaciones del lenguaje y añadir características muy útiles como el recolector de basura. No es necesario preocuparse de liberar memoria, el recolector se encarga de ello y como es de baja prioridad, cuando entra en acción, permite liberar bloques de memoria muy grandes, lo que limita en mucho la fragmentación de la memoria.

Orientado a Objetos

Java implementa la tecnología básica de C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje. Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo.

Distribuido

Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales.

Java en sí no es distribuido, sino que proporciona las librerías y herramientas para que los programas puedan ser distribuidos, es decir, que se corran en varias máquinas, interactuando.

Robusto

Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria.

También implementa los arrays auténticos, en vez de listas enlazadas de punteros, con comprobación de límites, para evitar la posibilidad de sobrescribir o corromper memoria resultado de punteros que señalan a zonas equivocadas. Estas características reducen drásticamente el tiempo empleado en el desarrollo de aplicaciones Java.

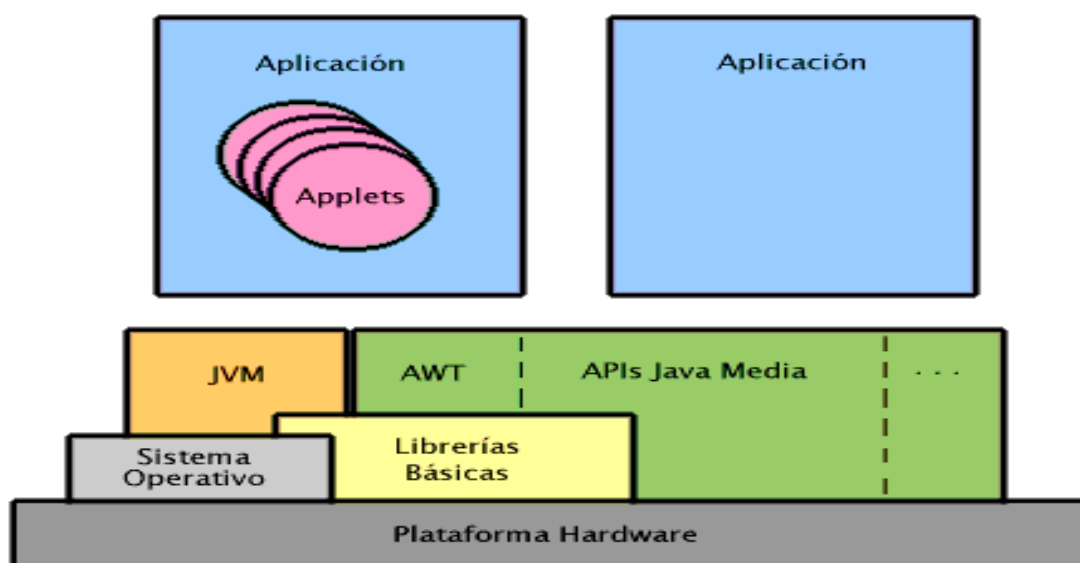
Además, para asegurar el funcionamiento de la aplicación, realiza una verificación de los ByteCodes, que son el resultado de la compilación de un programa Java. Es un código de máquina virtual que es interpretado por el intérprete Java. No es el código máquina directamente entendible por el hardware, pero ya ha pasado todas las fases del compilador: análisis de instrucciones, orden de operadores, etc., y ya tiene generada la pila de ejecución de órdenes.

Arquitectura Neutral

Para establecer Java como parte integral de la red, el compilador Java compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará. Cualquier máquina que tenga el sistema de ejecución (runtime) puede ejecutar ese código objeto, sin importar en modo alguno la máquina en que ha sido generado.

El código fuente Java se "compila" a un código de bytes de alto nivel independiente de la máquina. Este código (ByteCode) está diseñado para ejecutarse en una máquina hipotética que es implementada por un sistema runtime, que sí es dependiente de la máquina.

En una representación en que tuviésemos que indicar todos los elementos que forman parte de la arquitectura de Java sobre una plataforma genérica, obtendríamos una imagen como la siguiente:



En ella podemos ver que lo verdaderamente dependiente del sistema es la Máquina Virtual Java (JVM) y las librerías fundamentales, que también permitirían acceder directamente al hardware de la máquina. Además, siempre habrá APIs de Java que también entren en contacto directo con el hardware y serán dependientes de la máquina.

Seguro

El código Java pasa muchos tests antes de ejecutarse en una máquina. El código se pasa a través de un verificador de ByteCode que comprueba el formato de los fragmentos de código y aplica un probador de teoremas para detectar fragmentos de código ilegal -código que falsea punteros, viola derechos de acceso sobre objetos o intenta cambiar el tipo o clase de un objeto-.

Si los ByteCodes pasan la verificación si generar ningún mensaje de error, entonces sabemos que:

- El código no produce desbordamiento de operandos en la pila
- El tipo de los parámetros de todos los códigos de operación son conocidos y correctos
- No ha ocurrido ninguna conversión ilegal de datos, tal como convertir enteros en punteros
- El acceso a los campos de un objeto se sabe que es legal: public, private, protected
- No hay ningún intento de violar las reglas de acceso y seguridad establecidas

El Cargador de Clases también ayuda a Java a mantener su seguridad, separando el espacio de nombres del sistema de ficheros local del de los recursos procedentes de la red. Esto limita cualquier aplicación del tipo Caballo de Troya, ya que las clases se buscan primero entre las locales y luego entre las procedentes del exterior.

Las clases importadas de la red se almacenan en un espacio de nombres privado, asociado con el origen. Cuando una clase del espacio de nombres privado accede a otra clase, primero se busca en las clases predefinidas (del sistema local) y luego en el espacio de nombres de la clase que hace la referencia. Esto imposibilita que una clase suplante a una predefinida.

En resumen, las aplicaciones de Java resultan extremadamente seguras, ya que no acceden a zonas delicadas de memoria o de sistema, con lo cual evitan la interacción de ciertos virus. Java no posee una semántica específica para modificar la pila de programa o la memoria libre o utilizar objetos y métodos de un programa sin los privilegios del kernel del sistema operativo. El Cargador de Clases puede verificar una firma digital antes de realizar una instancia de un objeto. Por tanto ningún objeto se crea y almacena en memoria, sin que se validen los privilegios de acceso. Es decir, la seguridad se integra en el momento en que se interpreta, con el nivel de detalle y de privilegio que sea necesario. Con código compilado no sería posible establecer estos niveles de seguridad entre dos objetos pertenecientes al mismo proceso, porque al compartir un único espacio de direcciones, el código de uno podría acceder tranquilamente a cualquier dirección del otro.

Portable

Más allá de la portabilidad básica por ser de arquitectura independiente, Java implementa otros estándares de portabilidad para facilitar el desarrollo. Los enteros son siempre enteros y además, enteros de 32 bits en complemento a 2. Además, Java construye sus interfaces de usuario a través de un sistema abstracto de ventanas de forma que las ventanas puedan ser implantadas en entornos Unix, Pc o Mac.

Interpretado

Se puede decir que un lenguaje es interpretado si sus instrucciones se ejecutan secuencialmente a partir de código fuente. Para ejecutar el código de un lenguaje interpretado, necesitamos un intérprete de ese lenguaje. El intérprete irá recibiendo líneas de código que traducirá a lenguaje máquina para que se ejecute. A diferencia de los lenguajes compilados, no se produce un ejecutable. De este modo, de una plataforma a otra, sólo habrá que cambiar el intérprete, no el código.

Multihilo

Al ser Multihilo, Java permite muchas actividades simultáneas en un programa. El beneficio de ser multihilo consiste en un mejor rendimiento interactivo y mejor comportamiento en tiempo real. Aunque el comportamiento en tiempo real está limitado a las capacidades del sistema operativo subyacente de la plataforma, aún supera a los entornos de flujo único de programa tanto en facilidad de desarrollo como en rendimiento.

Struts

Struts es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma J2EE. Struts se desarrollaba como parte del proyecto Jakarta de la Apache Software Foundation, pero actualmente es un proyecto independiente conocido como Apache Struts.

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

Entre las características de Struts se pueden mencionar:

- Configuración del control centralizada.
- Interrelaciones entre acciones y página u otras acciones se especifican por tablas XML en lugar de codificarlas en los programas o páginas.
- Componentes de aplicación, que son el mecanismo para compartir información bidireccionalmente entre el usuario de la aplicación y las acciones del modelo.
- Librerías de entidades para facilitar la mayoría de las operaciones que generalmente realizan las páginas JSP.

- Struts contiene herramientas para validación de campos de plantillas bajo varios esquemas que van desde validaciones locales en la página hasta las validaciones de fondo hechas a nivel de las acciones.

Spring

Spring Framework es, de forma resumida, es un framework ligero pero completo que facilita la creación de aplicaciones Java SE y Java EE. Las principales características que hacen de este framework sea uno de los más difundidos en la actualidad, son las siguientes:

- Gestión de la configuración basada en JavaBeans, aplicando los principios de la Inversión de Control, mediante la técnica de inyección de dependencia.
- Capa abstracta genérica para la gestión de transacciones de bases de datos
- Integra estrategias genéricas para JTA y JDBC
- Integración con los principales frameworks de persistencia Hibernate, JDO e iBatis.
- Programación orientada a aspectos (AOP) para proporcionar servicios tales como gestión de transacciones

Hibernate

Hibernate es una de las soluciones más populares y más completas de mapeos entre objetos y base de datos relacionales para entornos Java. Hibernate no sólo se ocupa de la asignación de las clases Java a tablas de bases de datos (y de tipos de datos Java a tipos de datos SQL), sino que también proporciona un lenguaje de consulta y recuperación de datos y puede reducir significativamente el tiempo de desarrollo, lo contrario que sucede con la manipulación manual de datos en SQL y JDBC. Administra las bases de datos y el mapeo entre las bases de datos y los objetos.

Sus principales características:

- Se integra con todos los servidores de aplicaciones J2EE, contenedores Web y en aplicaciones independientes.
- Es open source
- Sigue el modelo natural de programación orientada a objetos
- Es altamente escalable
- Ofrece un lenguaje de consultas
- Implementa la API y el lenguaje de consultas de EJB 3.0

Motivación

Tras exponer las tecnologías utilizadas en este proyecto, la pregunta que resta por contestar es el porqué de su utilización. Como se ha podido observar a lo largo de este punto el presente proyecto está muy ligado al uso de Java como lenguaje para llevar a cabo la implementación, ya que el resto de las tecnologías son elementos que se apoyan en el mismo, por lo que es lógico comenzar justificando su uso.

Se ha elegido utilizar un lenguaje como Java por varios motivos descritos a continuación:

- Su portabilidad nos permitirá trasladar la aplicación entre distintas plataformas sin tener que realizar prácticamente ninguna modificación en los elementos ya desarrollados.
- Al ser un lenguaje orientado a objetos facilitará la creación de elementos reutilizables con poco acoplamiento con otros elementos de la misma aplicación. Una cuestión a tener muy en cuenta ya que no solo ayuda a la mencionada reutilización, sino también a una posible extensión en la funcionalidad de dicha aplicación.
- Es un lenguaje que permite la programación de aplicaciones Web, vital teniendo en cuenta que nuestra aplicación será de este tipo.
- Java es uno de los lenguajes que mayor cantidad de librerías que extienden la API estándar del lenguaje tiene, como se ha podido ver a lo largo de este punto. Esta característica es muy importante, ya prácticamente nos asegura el hecho de que podremos utilizar una librería que simplifica la implementación de alguna funcionalidad de nuestra aplicación.
- Como se trata de uno de los lenguajes más usados actualmente, Java es uno de los lenguajes de los que más documentación esta a nuestra disposición. Hecho importantísimo cuando se posee de poca experiencia en algunos campos abarcados por el proyecto y de poco tiempo para llevarlo a cabo.

Por lo tanto, teniendo claro que el lenguaje que más se adapta a nuestras necesidades, hay que entrar a razonar el uso de las restantes tecnologías utilizadas. Entrando a valorar el motivo de optar por el uso del framework Struts para la capa de vista, podemos señalar una serie de ventajas que conlleva a continuación:

- Refuerza la modularidad y partición de la aplicación.
- Incrementa la separación de roles.
- Incrementa la manejabilidad del código.
- Incrementa la extensibilidad del código.

- Centra al desarrollador en la lógica de negocio.

Por otro lado, usaremos Spring para integrar las distintas capas de la aplicación, principalmente porque usaremos la inyección de dependencias para relacionar los distintos componentes, lo que nos asegura un alto desacoplamiento, y porque nos ofrece la posibilidad de integrar los distintos frameworks que utilizaremos (principalmente Struts e Hibernate).

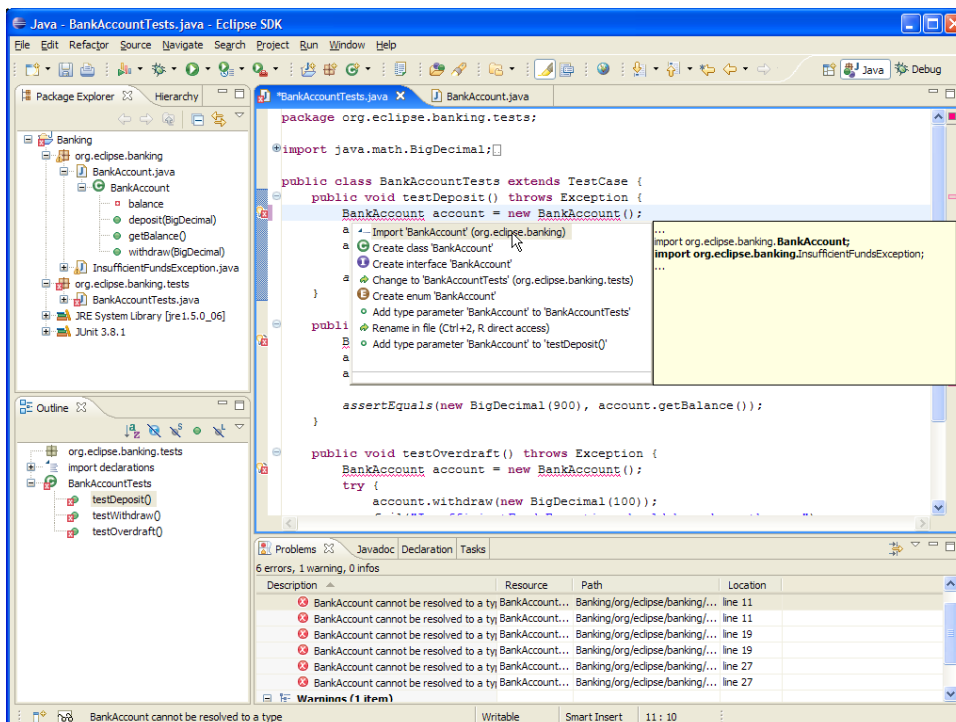
Por último, señalar que se utilizará Hibernate para la capa de acceso a datos, ya que nos ofrece todas las utilidades necesarias para realizar las operaciones correspondientes a dicha capa sin salirnos de la forma natural de la orientación objetos, además de proporcionarnos una alta independencia del gestor de base de datos que optemos por utilizar.

Material

A continuación se comentará de manera breve las herramientas que se utilizarán a lo largo del desarrollo de este proyecto, tanto la aplicación como su documentación.

Eclipse

Eclipse es una plataforma de software de código abierto independiente de la arquitectura en la que se ejecuta. Esta plataforma, típicamente ha sido usada para desarrollar entornos integrados de desarrollo (del Inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se enmarca como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente.



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

Microsoft Office Word 2007

Microsoft Word un procesador de textos creado por Microsoft, que nos permite editar, dar formato, grabar y modificar documentos de texto, si bien ofrece numerosas funciones avanzadas como la inserción y edición de imágenes, creación de referencias cruzadas, etc. Actualmente integrado en la suite ofimática Microsoft Office, que desgraciadamente se trata de software no libre.

Herramientas CASE: MagicDraw UML

Las herramientas CASE (ingeniería de software asistida por ordenador) son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo

el coste de las mismas en tiempo y de dinero. A modo de esquema podríamos señalar como objetivos de las herramientas CASE los siguientes:

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software.
- Mejorar el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto
- Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
- Automatizar, desarrollo del software, documentación, generación de código, pruebas de errores y gestión del proyecto.
- Ayuda a la reutilización del software, portabilidad y estandarización de la documentación
- Gestión global en todas las fases de desarrollo de software con una misma herramienta.
- Facilitar el uso de las distintas metodologías propias de la ingeniería del software.

Para el desarrollo del presente proyecto se ha optado por la utilización de MagicDraw como herramienta CASE de apoyo, debido a que es una herramienta que permite modelar mediante diagramas UML de una forma rápida e intuitiva.

OpenProj

En todo proyecto es necesario apoyarse en una aplicación que permita la administración de administración de proyectos y en este caso se ha elegido el programa OpenProj, principalmente porque reúne todas las funcionalidades necesarias tratándose de una herramienta OpenSource. Aunque debido al tamaño del proyecto y a los recursos utilizados en el mismo únicamente se utilizará para desarrollar los diagramas Gantt, la herramienta incluye:

- Costos de valor acumulado
- Diagrama de Gantt
- Gráfico PERT
- Estructura de descomposición del recurso gráfico (EDR)
- Informes de uso de tareas
- Diagrama de Estructura de descomposición del trabajo (EDT)

MySQL

Como sistema de gestión de bases de datos se ha optado por MySQL debido a que se trata de uno de los más utilizados siendo una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, que únicamente puede provocar problemas de integridad en

entornos de alta concurrencia en la modificación. Algo que la hace ideal para las aplicaciones web ya que en estas hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos. Sin olvidar que se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia.

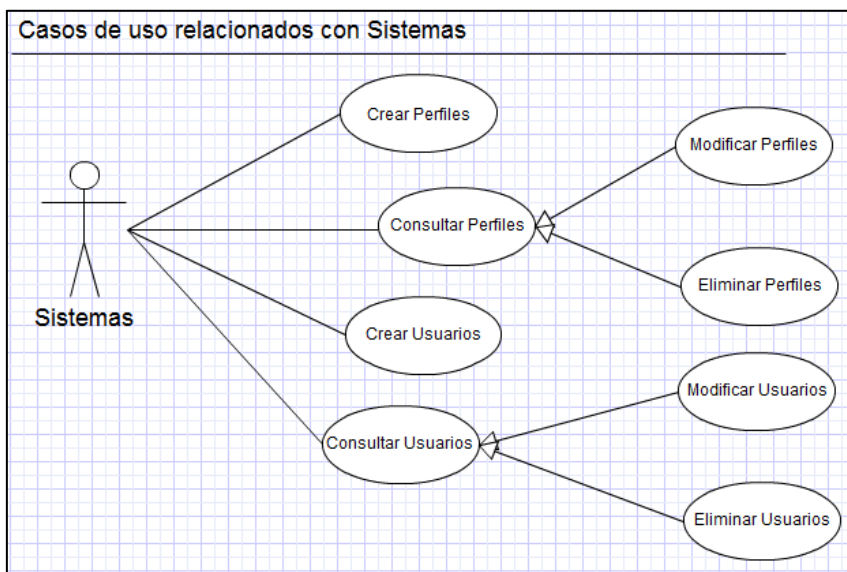
Requisitos funcionales

El primer paso que vamos a llevar a cabo es la definición de los distintos actores que van a interactuar con el sistema, que serán los siguientes:

- Sistemas: Nombraremos de este modo a los encargados de gestionar tanto los distintos perfiles existentes en el sistema como los distintos usuarios que tienen acceso al mismo.
- Gerencia: Denominaremos así a aquellos individuos de la empresa con capacidad de gestionar los inmuebles que forman parte de la empresa.
- Mercado: Aquellas personas de la empresa encargados de gestionar los distintos productos con los que trabaja dicha compañía.
- Administrativos: Encargados de gestionar el stock a lo largo de los distintos inmuebles que posee el sistema.

A lo largo de este apartado estudiaremos los casos de uso que se pueden dar de la aplicación.

Casos de uso relacionados con el actor Sistemas



Nombre del Caso de Uso	Consultar Perfiles
Resumen de la funcionalidad	El usuario busca un perfil utilizando distintos criterios
Papel del Actor	Necesario para efectuar la acción de gestión sobre el perfil
Casos de uso relacionados	Modificar Perfiles, Eliminar Perfiles
Precondiciones	El usuario debe estar conectado al sistema
Postcondiciones	Se muestra un listado de perfiles que cumplen los criterios de búsqueda
Descripción	Un usuario busca un perfil que cumple determinados criterios, como puede ser nombre, fecha de creación... Al final del proceso se obtiene una lista de los perfiles que cumplen todos los criterios introducidos.

Nombre del Caso de Uso	Consultar Usuarios
Resumen de la funcionalidad	El usuario busca un usuario utilizando distintos criterios
Papel del Actor	Necesario para efectuar la acción de gestión sobre el usuario
Casos de uso relacionados	Modificar Usuarios, Eliminar Usuarios
Precondiciones	El usuario debe estar conectado al sistema
Postcondiciones	Se muestra un listado de usuarios que cumplen los criterios de búsqueda
Descripción	Un usuario busca un usuario que cumple determinados criterios, como puede ser nombre, perfil asociado... Al final del proceso se obtiene una lista de los usuarios que cumplen todos los criterios introducidos.

Nombre del Caso de Uso	Crear Perfiles
Resumen de la funcionalidad	Dar de alta un perfil en la base de datos
Papel del Actor	Necesario para crear perfiles en la base de datos.
Casos de uso relacionados	Ninguno
Precondiciones	El usuario debe estar conectado al sistema
Postcondiciones	El perfil debe estar creado en la base de datos.
Descripción	El usuario introduce en un formulario la información sobre el perfil a crear. Una vez introducidos los datos se almacenan en la base de datos o se produce un error en caso de no ser posible

Nombre del Caso de Uso	Crear Usuarios
Resumen de la funcionalidad	Dar de alta un usuario en la base de datos
Papel del Actor	Necesario para crear usuarios en la base de datos.
Casos de uso relacionados	Ninguno
Precondiciones	El usuario debe estar conectado al sistema
Postcondiciones	El usuario debe estar creado en la base de datos.
Descripción	El usuario introduce en un formulario la información sobre el perfil a crear. Una vez introducidos los datos se almacenan en la base de datos o se produce un error en caso de no ser posible

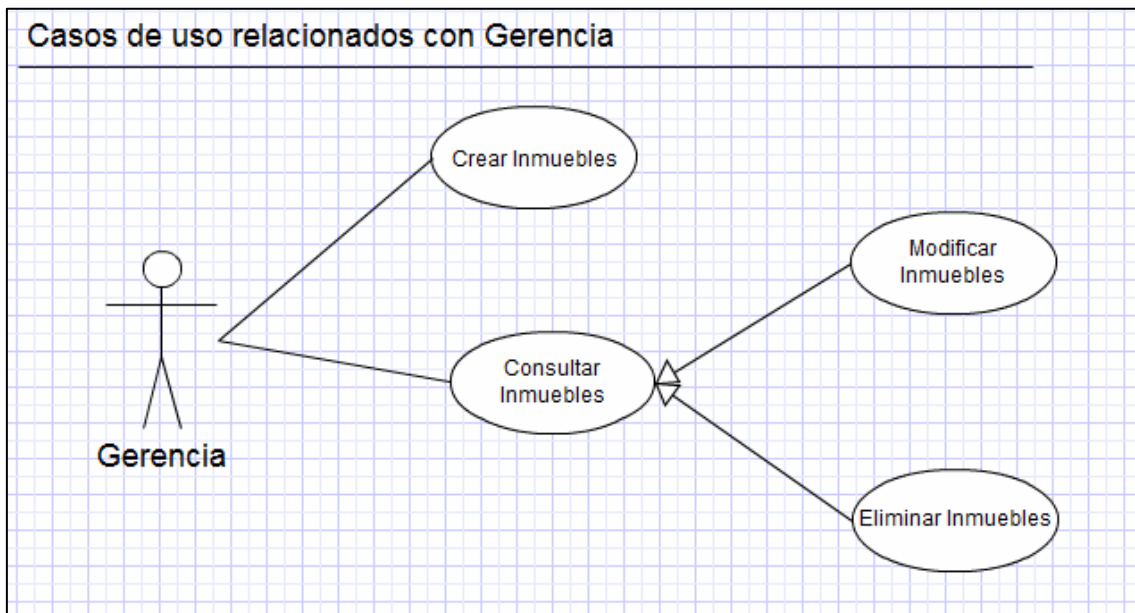
Nombre del Caso de Uso	Modificar Perfiles
Resumen de la funcionalidad	Se modifican los datos introducidos sobre un perfil creado previamente
Papel del Actor	Necesario para efectuar la acción de gestión sobre el perfil
Casos de uso relacionados	Consultar Perfiles
Precondiciones	El perfil debe existir en la base de datos.
Postcondiciones	Los datos deben haber sido almacenados en base de datos o mostrar los errores producidos en pantalla.
Descripción	Tras haber realizado una consulta de perfiles y seleccionar uno, se verá la ficha de dicho perfil con los datos almacenados que podrán ser modificados por el usuario. Una vez modificados se almacenan en base de datos o se muestran los errores producidos.

Nombre del Caso de Uso	Modificar Usuarios
Resumen de la funcionalidad	Se modifican los datos introducidos sobre un usuario creado previamente
Papel del Actor	Necesario para efectuar la acción de gestión sobre el usuario
Casos de uso relacionados	Consultar Usuarios
Precondiciones	El usuario debe existir en la base de datos.
Postcondiciones	Los datos deben haber sido almacenados en base de datos o mostrar los errores producidos en pantalla.
Descripción	Tras haber realizado una consulta de usuarios y seleccionar uno, se verá la ficha de dicho usuario con los datos almacenados que podrán ser modificados por el usuario. Una vez modificados se almacenan en base de datos o se muestran los errores producidos.

Nombre del Caso de Uso	Eliminar Perfiles
Resumen de la funcionalidad	Se elimina de la base de datos la información del perfil creado previamente.
Papel del Actor	Necesario para efectuar la acción de gestión sobre el perfil
Casos de uso relacionados	Consultar Perfiles
Precondiciones	El perfil debe existir previamente en base de datos.
Postcondiciones	La información sobre el perfil debe haber sido eliminada de la base de datos o mostrar los errores producidos en pantalla.
Descripción	Tras haber realizado una consulta de perfiles y seleccionar uno, se verá la ficha de dicho perfil donde el usuario podrá seleccionar eliminarlo. En caso de eliminarse el perfil, se volverá a la pantalla de consulta de perfiles, en caso contrario se mostraran los errores producidos en la pantalla de la ficha del perfil

Nombre del Caso de Uso	Eliminar Usuarios
Resumen de la funcionalidad	Se elimina de la base de datos la información del usuario creado previamente.
Papel del Actor	Necesario para efectuar la acción de gestión sobre el usuario
Casos de uso relacionados	Consultar Usuarios
Precondiciones	El usuario debe existir previamente en base de datos.
Postcondiciones	La información sobre el usuario debe haber sido eliminada de la base de datos o mostrar los errores producidos en pantalla.
Descripción	Tras haber realizado una consulta de usuarios y seleccionar uno, se verá la ficha de dicho usuario donde el actor podrá seleccionar eliminarlo. En caso de eliminarse el usuario, se volverá a la pantalla de consulta de usuarios, en caso contrario se mostraran los errores producidos en la pantalla de la ficha del usuario

Casos de uso relacionados con el actor Gerencia



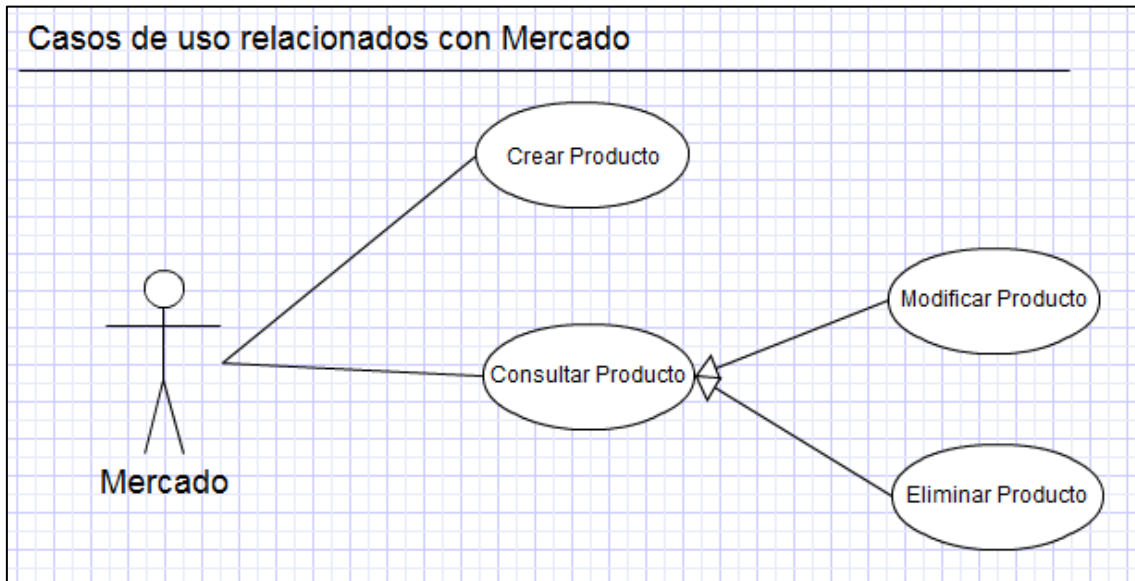
Nombre del Caso de Uso	Consultar Inmuebles
Resumen de la funcionalidad	El usuario busca un inmueble utilizando distintos criterios
Papel del Actor	Necesario para efectuar la acción de gestión sobre el inmueble
Casos de uso relacionados	Modificar Inmuebles, Eliminar Inmuebles
Precondiciones	El usuario debe estar conectado al sistema
Postcondiciones	Se muestra un listado de inmuebles que cumplen los criterios de búsqueda
Descripción	Un usuario busca un inmueble que cumple determinados criterios, como puede ser dirección, el tipo de inmueble que es... Al final del proceso se obtiene una lista de los inmuebles que cumplen todos los criterios introducidos.

Nombre del Caso de Uso	Crear Inmuebles
Resumen de la funcionalidad	Dar de alta un inmueble en la base de datos
Papel del Actor	Necesario para crear inmuebles en la base de datos.
Casos de uso relacionados	Ninguno
Precondiciones	El usuario debe estar conectado al sistema
Postcondiciones	El inmueble debe estar creado en la base de datos.
Descripción	El usuario introduce en un formulario la información sobre el inmueble a crear. Una vez introducidos los datos se almacenan en la base de datos o se produce un error en caso de no ser posible

Nombre del Caso de Uso	Modificar Inmuebles
Resumen de la funcionalidad	Se modifican los datos introducidos sobre un inmueble creado previamente
Papel del Actor	Necesario para efectuar la acción de gestión sobre el inmueble
Casos de uso relacionados	Consultar Inmuebles
Precondiciones	El inmueble debe existir en la base de datos.
Postcondiciones	Los datos deben haber sido almacenados en base de datos o mostrar los errores producidos en pantalla.
Descripción	Tras haber realizado una consulta de inmuebles y seleccionar uno, se verá la ficha de dicho inmueble con los datos almacenados que podrán ser modificados por el usuario. Una vez modificados se almacenan en base de datos o se muestran los errores producidos.

Nombre del Caso de Uso	Eliminar Inmuebles
Resumen de la funcionalidad	Se elimina de la base de datos la información del inmueble creado previamente.
Papel del Actor	Necesario para efectuar la acción de gestión sobre el inmueble
Casos de uso relacionados	Consultar Inmuebles
Precondiciones	El inmueble debe existir previamente en base de datos.
Postcondiciones	La información sobre el inmueble debe haber sido eliminada de la base de datos o mostrar los errores producidos en pantalla.
Descripción	Tras haber realizado una consulta de inmuebles y seleccionar uno, se verá la ficha de dicho inmueble donde el usuario podrá seleccionar eliminarlo. En caso de eliminarse el inmueble, se volverá a la pantalla de consulta de inmuebles, en caso contrario se mostraran los error producidos en la pantalla de la ficha del inmueble

Casos de uso relacionados con el actor Mercado



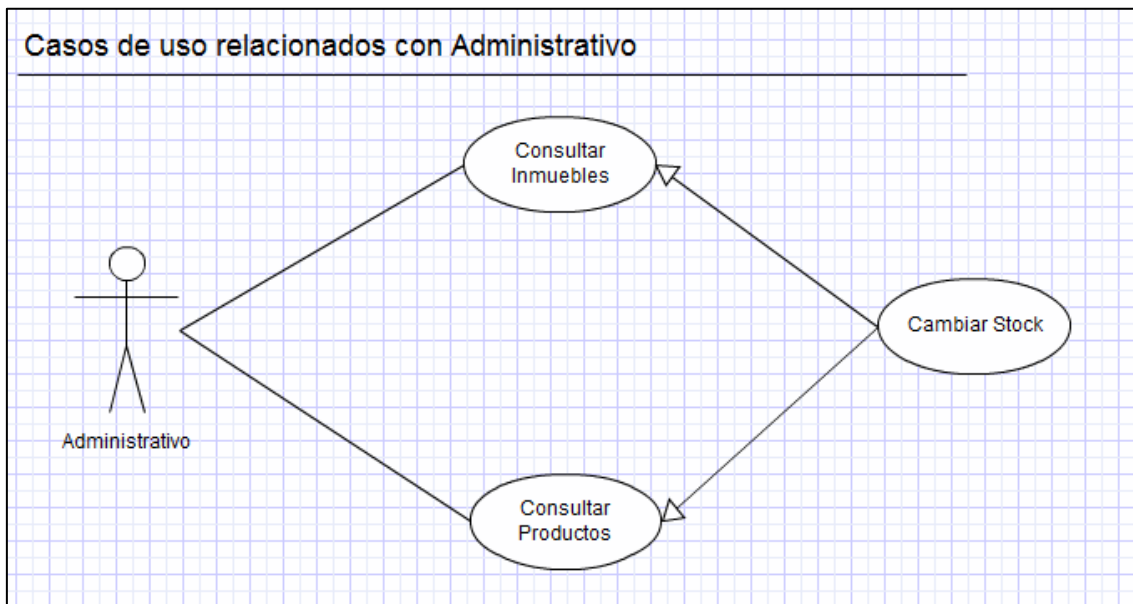
Nombre del Caso de Uso	Crear Producto
Resumen de la funcionalidad	Dar de alta un producto en la base de datos
Papel del Actor	Necesario para crear productos en la base de datos.
Casos de uso relacionados	Ninguno
Precondiciones	El usuario debe estar conectado al sistema
Postcondiciones	El producto debe estar creado en la base de datos.
Descripción	El usuario introduce en un formulario la información sobre el producto a crear. Una vez introducidos los datos se almacenan en la base de datos o se produce un error en caso de no ser posible

Nombre del Caso de Uso	Consultar Productos
Resumen de la funcionalidad	El usuario busca un producto utilizando distintos criterios
Papel del Actor	Necesario para efectuar la acción de gestión sobre el producto
Casos de uso relacionados	Modificar Producto, Eliminar Producto
Precondiciones	El usuario debe estar conectado al sistema
Postcondiciones	Se muestra un listado de productos que cumplen los criterios de búsqueda
Descripción	Un usuario busca un producto que cumple determinados criterios, como puede ser nombre, marca, categoría... Al final del proceso se obtiene una lista de los productos que cumplen todos los criterios introducidos.

Nombre del Caso de Uso	Eliminar Producto
Resumen de la funcionalidad	Se elimina de la base de datos la información del producto creado previamente.
Papel del Actor	Necesario para efectuar la acción de gestión sobre el producto
Casos de uso relacionados	Consultar Productos
Precondiciones	El producto debe existir previamente en base de datos.
Postcondiciones	La información sobre el producto debe haber sido eliminada de la base de datos o mostrar los errores producidos en pantalla.
Descripción	Tras haber realizado una consulta de productos y seleccionar uno, se verá la ficha de dicho producto donde el usuario podrá seleccionar eliminarlo. En caso de eliminarse el producto, se volverá a la pantalla de consulta de productos, en caso contrario se mostraran los errores producidos en la pantalla de la ficha del producto

Nombre del Caso de Uso	Modificar Producto
Resumen de la funcionalidad	Se modifican los datos introducidos sobre un producto creado previamente
Papel del Actor	Necesario para efectuar la acción de gestión sobre el producto
Casos de uso relacionados	Consultar Productos
Precondiciones	El producto debe existir en la base de datos.
Postcondiciones	Los datos deben haber sido almacenados en base de datos o mostrar los errores producidos en pantalla.
Descripción	Tras haber realizado una consulta de productos y seleccionar uno, se verá la ficha de dicho inmueble con los datos almacenados que podrán ser modificados por el usuario. Una vez modificados se almacenan en base de datos o se muestran los errores producidos.

Casos de uso relacionados con el actor Administrativo



Nombre del Caso de Uso	Consultar Productos
Resumen de la funcionalidad	El usuario busca un producto utilizando distintos criterios
Papel del Actor	Necesario para efectuar la acción de gestión sobre el producto
Casos de uso relacionados	Cambiar Stock
Precondiciones	El usuario debe estar conectado al sistema
Postcondiciones	Se muestra un listado de productos que cumplen los criterios de búsqueda
Descripción	Un usuario busca un producto que cumple determinados criterios, como puede ser nombre, marca, categoría... Al final del proceso se obtiene una lista de los productos que cumplen todos los criterios introducidos.

Nombre del Caso de Uso	Consultar Inmuebles
Resumen de la funcionalidad	El usuario busca un inmueble utilizando distintos criterios
Papel del Actor	Necesario para efectuar la acción de gestión sobre el inmueble
Casos de uso relacionados	Cambiar Stock
Precondiciones	El usuario debe estar conectado al sistema
Postcondiciones	Se muestra un listado de inmuebles que cumplen los criterios de búsqueda
Descripción	Un usuario busca un inmueble que cumple determinados criterios, como puede ser dirección, el tipo de inmueble que es... Al final del proceso se obtiene una lista de los inmuebles que cumplen todos los criterios introducidos.

Nombre del Caso de Uso	Cambiar Stock
Resumen de la funcionalidad	Se modifican el stock asociado al inmueble o producto seleccionado anteriormente
Papel del Actor	Necesario para efectuar la acción de gestión sobre el stock
Casos de uso relacionados	Consultar Productos, Consultar Inmuebles
Precondiciones	El producto y el inmueble deben existir en la base de datos.
Postcondiciones	Los datos deben haber sido almacenados en base de datos o mostrar los errores producidos en pantalla.
Descripción	Tras haber realizado una consulta de productos o inmuebles y seleccionar uno, se verá o el stock de los distintos productos existentes si se llega desde la consulta de un inmueble o el stock en los distintos inmuebles si se llega desde la consulta de productos. Una vez modificados se almacenan en base de datos o se muestran los errores producidos.

Interfaz de usuario

A lo largo de este apartado iremos viendo los prototipos de las distintas pantallas que formarán parte de la herramienta.

Pantalla de acceso

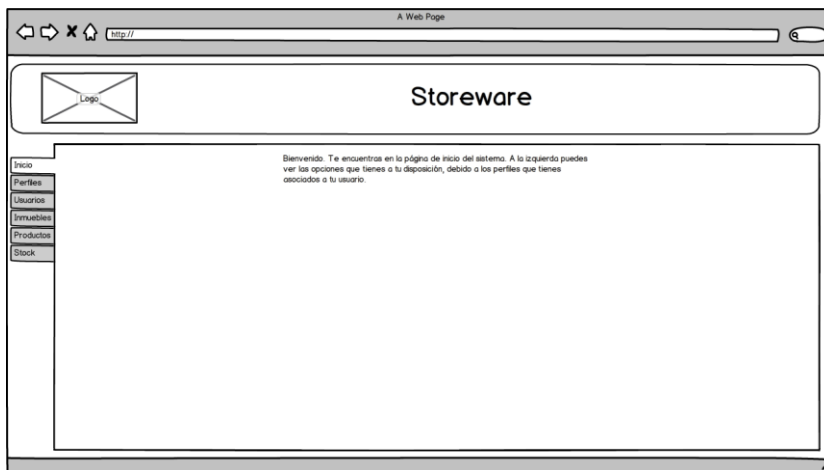
Se trata de la pantalla inicial de la herramienta, donde cada uno de los usuarios deberá introducir su identificador y su contraseña.



En caso de que los datos introducidos incorrectos se mostrarán el mensaje de error en la misma pantalla, con un contador de intentos disponibles antes del bloqueo de la cuenta.

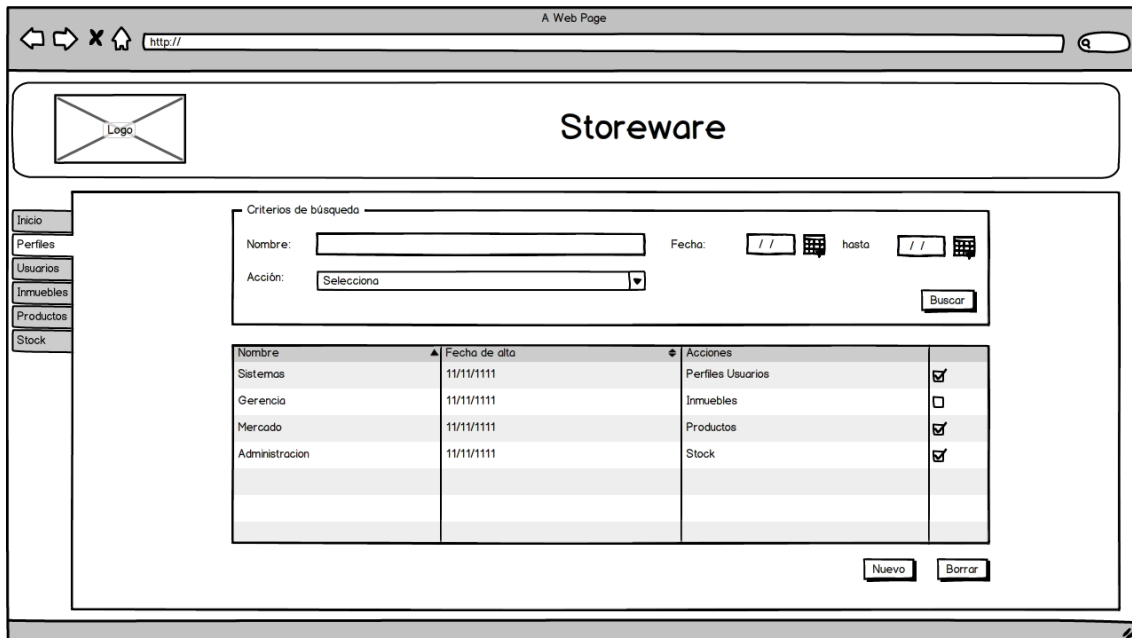
Pantalla de Inicio

Se trata de la pantalla que se mostrará cuando el usuario introduzca correctamente su identificador y su contraseña en la pantalla de acceso. Comentar que en las opciones del menú de la izquierda aparecerán aquellas acciones que pueden realizar los perfiles asociados al usuario.



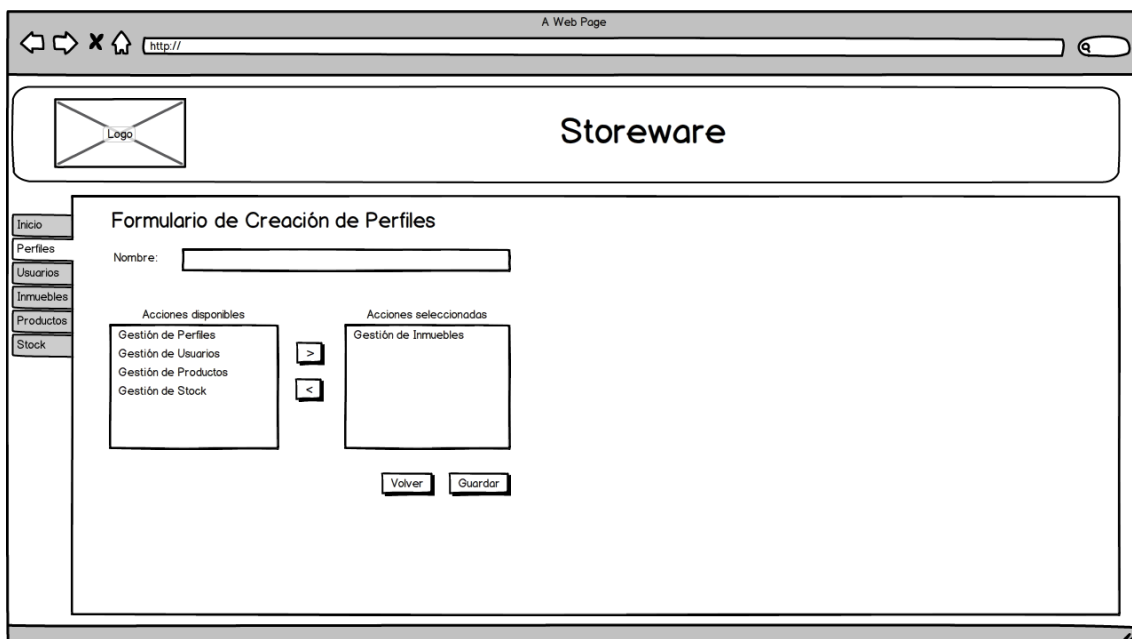
Pantalla de Gestión de Perfiles

Se trata de la pantalla que el usuario visualizará cuando pulse sobre la opción de “Perfiles” en el menú de la derecha. En la misma se permitirá realizar consultas sobre los perfiles dados de alta en el sistema, así como eliminar aquellos perfiles seleccionados de la lista de resultados, acceder a la pantalla de modificación de uno de los perfiles pulsando sobre su nombre o acceder a la pantalla de creación de un perfil pulsando sobre el botón “Nuevo”.



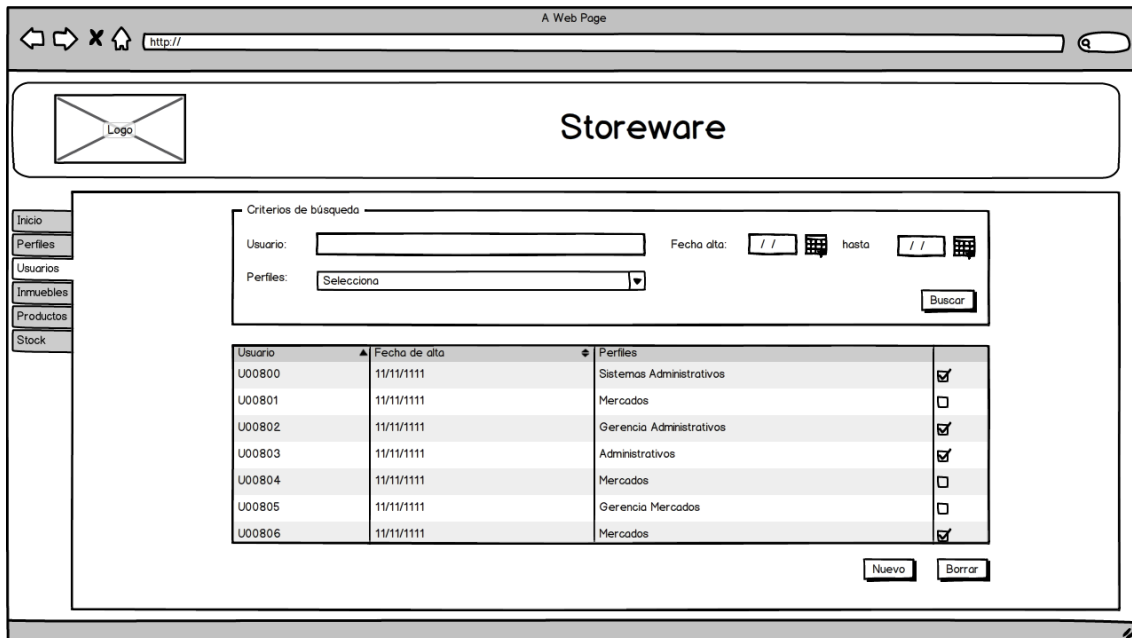
Pantalla de Creación/Modificación de Perfiles

La pantalla de creación y modificación de perfiles tienen la misma apariencia y funcionalidades, con la salvedad de que en el caso de la modificación de un perfil aparecerán los datos de dicho perfil precargados. Los cambios que se realicen en los datos introducidos en el formulario se almacenarán pulsando sobre el botón “Guardar”, mientras que el botón “Volver” nos permitirá acceder a la pantalla de Gestión de Perfiles.



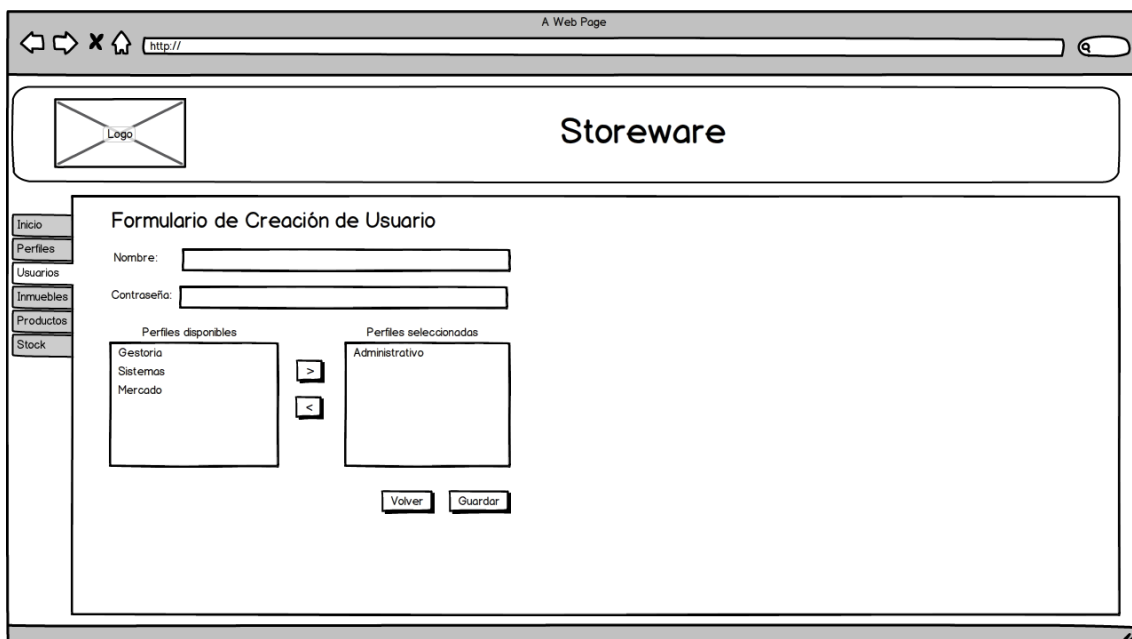
Pantalla de Gestión de Usuarios

Se trata de la pantalla que el usuario visualizará cuando pulse sobre la opción de “Usuarios” en el menú de la derecha. En la misma se permitirá realizar consultas sobre los usuarios dados de alta en el sistema, así como eliminar aquellos usuarios seleccionados de la lista de resultados, acceder a la pantalla de modificación de uno de los usuarios pulsando sobre su nombre o acceder a la pantalla de creación de un usuario pulsando sobre el botón “Nuevo”.



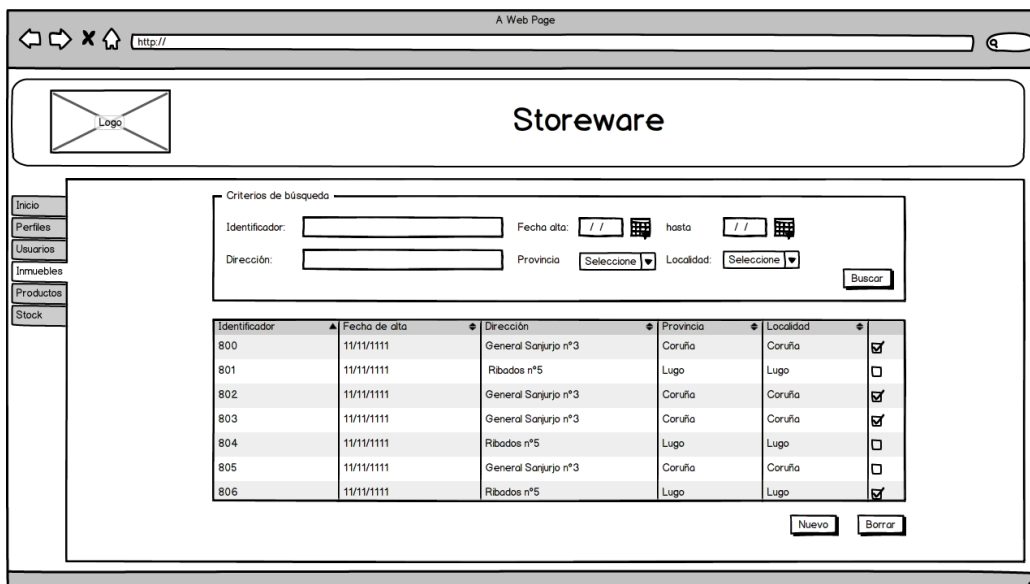
Pantalla de Creación/Modificación de Usuarios

La pantalla de creación y modificación de usuarios tienen la misma apariencia y funcionalidades, con la salvedad de que en el caso de la modificación de un usuario aparecerán los datos de dicho usuario precargados. Los cambios que se realicen en los datos introducidos en el formulario se almacenarán pulsando sobre el botón “Guardar”, mientras que el botón “Volver” nos permitirá acceder a la pantalla de Gestión de Usuarios.



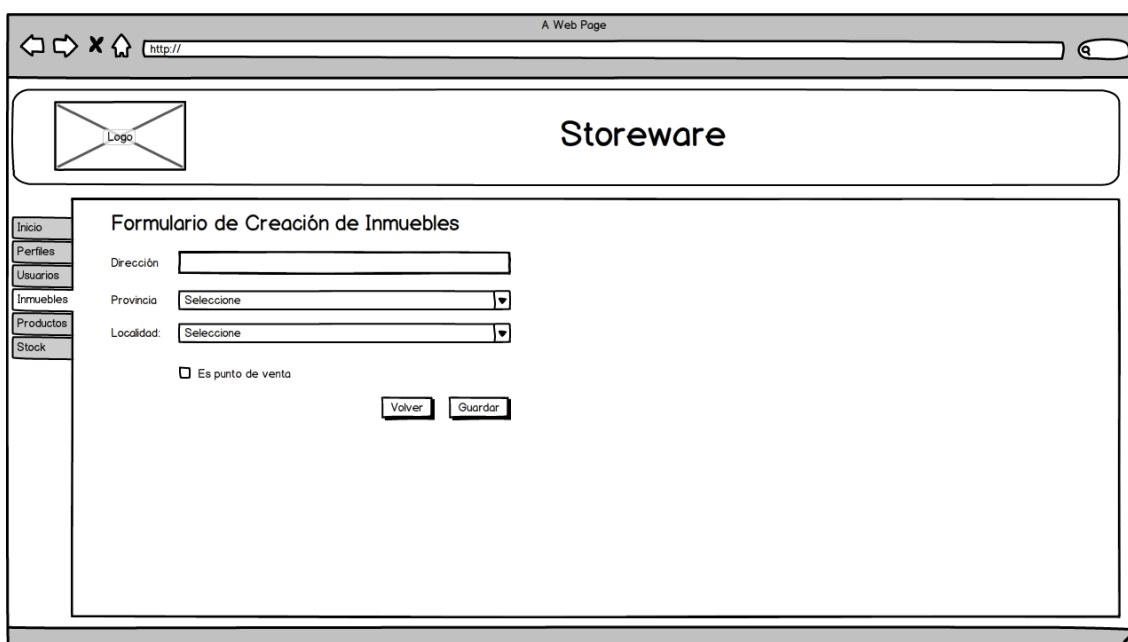
Pantalla de Gestión de Inmuebles

Se trata de la pantalla que el usuario visualizará cuando pulse sobre la opción de “Inmuebles” en el menú de la derecha. En la misma se permitirá realizar consultas sobre los inmuebles dados de alta en el sistema, así como eliminar aquellos inmuebles seleccionados de la lista de resultados, acceder a la pantalla de modificación de uno de los inmuebles pulsando sobre su identificador o acceder a la pantalla de creación de un inmueble pulsando sobre el botón “Nuevo”.



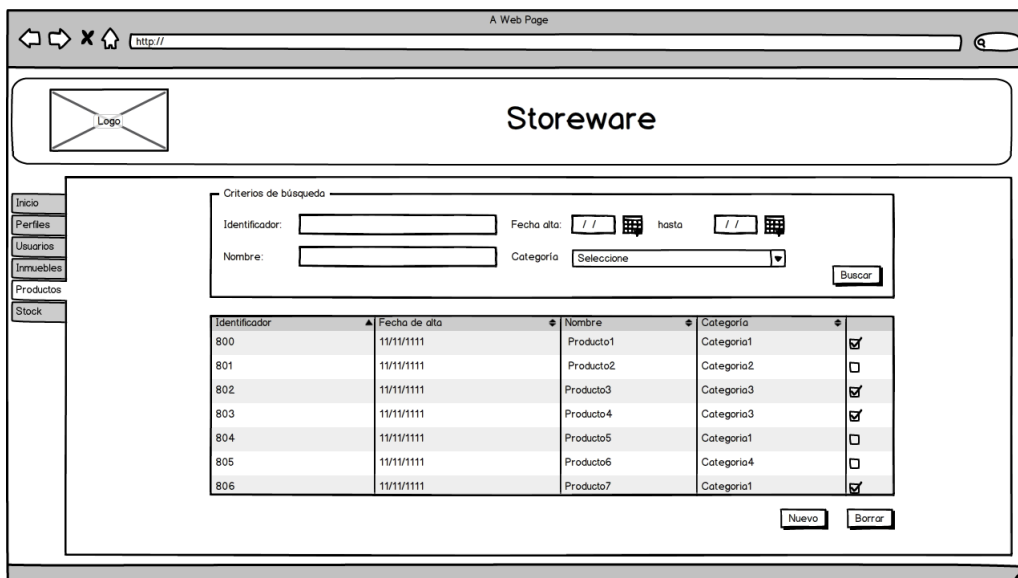
Pantalla de Creación/Modificación de Inmuebles

La pantalla de creación y modificación de inmuebles tienen la misma apariencia y funcionalidades, con la salvedad de que en el caso de la modificación de un inmueble aparecerán los datos de dicho inmueble precargados. Los cambios que se realicen en los datos introducidos en el formulario se almacenarán pulsando sobre el botón “Guardar”, mientras que el botón “Volver” nos permitirá acceder a la pantalla de Gestión de Inmuebles.



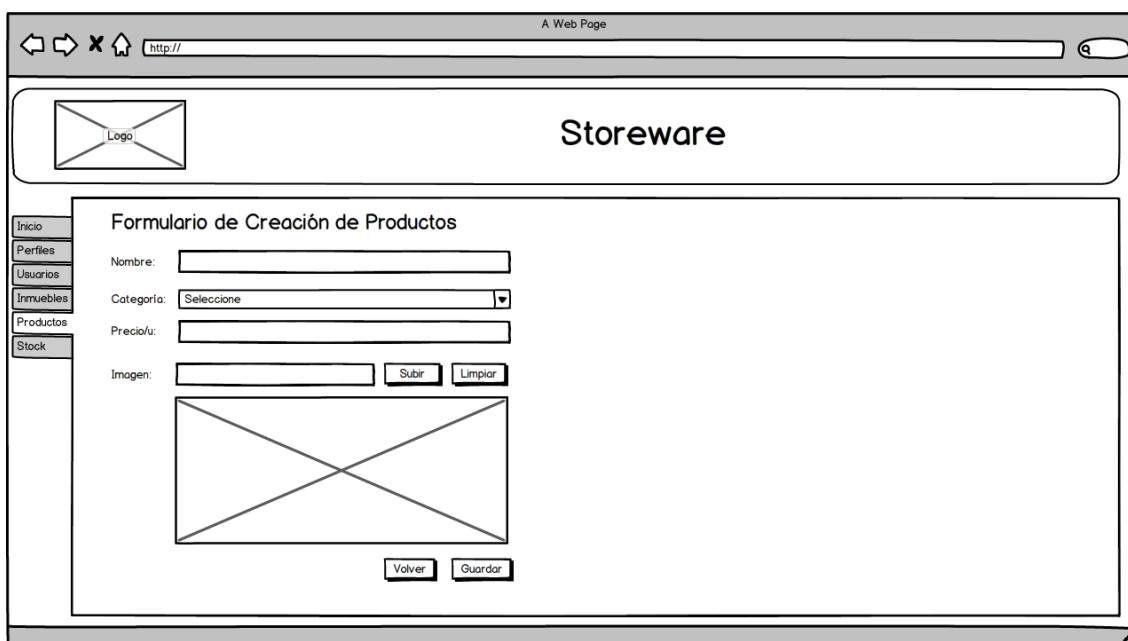
Pantalla de Gestión de Productos

Se trata de la pantalla que el usuario visualizará cuando pulse sobre la opción de “Productos” en el menú de la derecha. En la misma se permitirá realizar consultas sobre los productos dados de alta en el sistema, así como eliminar aquellos productos seleccionados de la lista de resultados, acceder a la pantalla de modificación de uno de los productos pulsando sobre su identificador o acceder a la pantalla de creación de un producto pulsando sobre el botón “Nuevo”.



Pantalla de Creación/Modificación de Productos

La pantalla de creación y modificación de productos tienen la misma apariencia y funcionalidades, con la salvedad de que en el caso de la modificación de un producto aparecerán los datos de dicho producto precargados. Los cambios que se realicen en los datos introducidos en el formulario se almacenarán pulsando sobre el botón “Guardar”, mientras que el botón “Volver” nos permitirá acceder a la pantalla de Gestión de Productos.



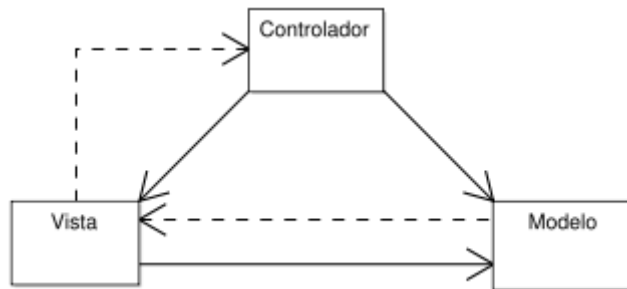
Pantalla de Gestión de Stock

Se trata de la pantalla que el usuario visualizará cuando pulse sobre la opción de “Stock” en el menú de la derecha. En la misma se permitirá realizar consultas sobre los productos o inmuebles dados de alta en el sistema de modo que permita acceder a las pantallas donde se pueden gestionar los stocks asociados a los mismos.



Arquitectura de la Aplicación

Comenzamos estudiando la arquitectura que se seguirá para el diseño del sistema. Se ha optado por seguir el patrón arquitectónico denominado MVC (Modelo – Vista - Controlador) debido a que se puede considerar que es un estándar de facto en el desarrollo del software. Este patrón, como señala su nombre, separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes/capas distintos buscando la mayor reutilización posible de los elementos implementados y facilitar la evolución del sistema debido a que los cambios y mejoras afectarán a uno o más componentes del mismo, pero nunca al núcleo.



Un ejemplo del flujo que sigue el control en una de las posibles implementaciones del patrón sería el siguiente:

- El usuario interactúa con la interfaz de alguna forma.
- El controlador recibe por parte de los objetos de la vista la notificación de la acción solicitada por el usuario.
- El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario.
- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo.
- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

Frameworks empleados en el desarrollo de la aplicación

Una vez que tenemos claro que la aplicación tendrá una estructura dividida en tres capas, lo conveniente es definir los frameworks que se van a utilizar en el desarrollo y en los cuales nos vamos a apoyar para facilitar el llegar a las metas marcadas. A lo largo de este punto se verán cuales han sido los elegidos.

Struts

Struts es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma J2EE. Struts se desarrollaba como parte del proyecto Jakarta de la Apache Software Foundation, pero actualmente es un proyecto independiente conocido como Apache Struts.

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

Entre las características de Struts se pueden mencionar:

- Configuración del control centralizada.
- Interrelaciones entre acciones y página u otras acciones se especifican por tablas XML en lugar de codificarlas en los programas o páginas.
- Componentes de aplicación, que son el mecanismo para compartir información bidireccionalmente entre el usuario de la aplicación y las acciones del modelo.
- Librerías de entidades para facilitar la mayoría de las operaciones que generalmente realizan las páginas JSP.
- Struts contiene herramientas para validación de campos de plantillas bajo varios esquemas que van desde validaciones locales en la página hasta las validaciones de fondo hechas a nivel de las acciones.

Spring

Spring Framework es, de forma resumida, un framework ligero pero completo que facilita la creación de aplicaciones Java SE y Java EE. Las principales características que hacen de este framework sea uno de los más difundidos en la actualidad, son las siguientes:

- Gestión de la configuración basada en JavaBeans, aplicando los principios de la Inversión de Control, mediante la técnica de inyección de dependencia.
- Capa abstracta genérica para la gestión de transacciones de bases de datos
- Integra estrategias genéricas para JTA y JDBC
- Integración con los principales frameworks de persistencia Hibernate, JDO e iBatis.

- Programación orientada a aspectos (AOP) para proporcionar servicios tales como gestión de transacciones

Hibernate

Hibernate es una de las soluciones más populares y más completas de mapeos entre objetos y base de datos relacionales para entornos Java. Hibernate no sólo se ocupa de la asignación de las clases Java a tablas de bases de datos (y de tipos de datos Java a tipos de datos SQL), sino que también proporciona un lenguaje de consulta y recuperación de datos y puede reducir significativamente el tiempo de desarrollo, lo contrario que sucede con la manipulación manual de datos en SQL y JDBC. Administra las bases de datos y el mapeo entre las bases de datos y los objetos.

Sus principales características:

- Se integra con todos los servidores de aplicaciones J2EE, contenedores Web y en aplicaciones independientes.
- Es open source
- Sigue el modelo natural de programación orientada a objetos
- Es altamente escalable
- Ofrece un lenguaje de consultas
- Implementa la API y el lenguaje de consultas de EJB 3.0

Patrones de diseño utilizados

A continuación se comentarán de forma breve los patrones de diseño que siguen en su implementación los frameworks utilizados. Señalar que son frameworks, especialmente en el caso de Spring, que abarcan un número elevado de patrones, por lo que únicamente se abarcarán alguno de ellos.

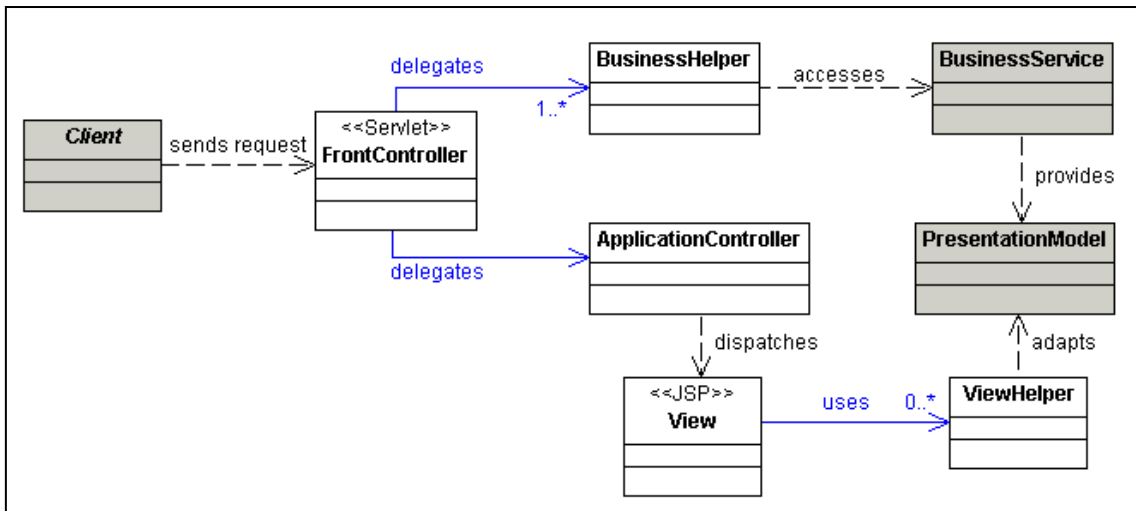
Struts

- Service To Worker

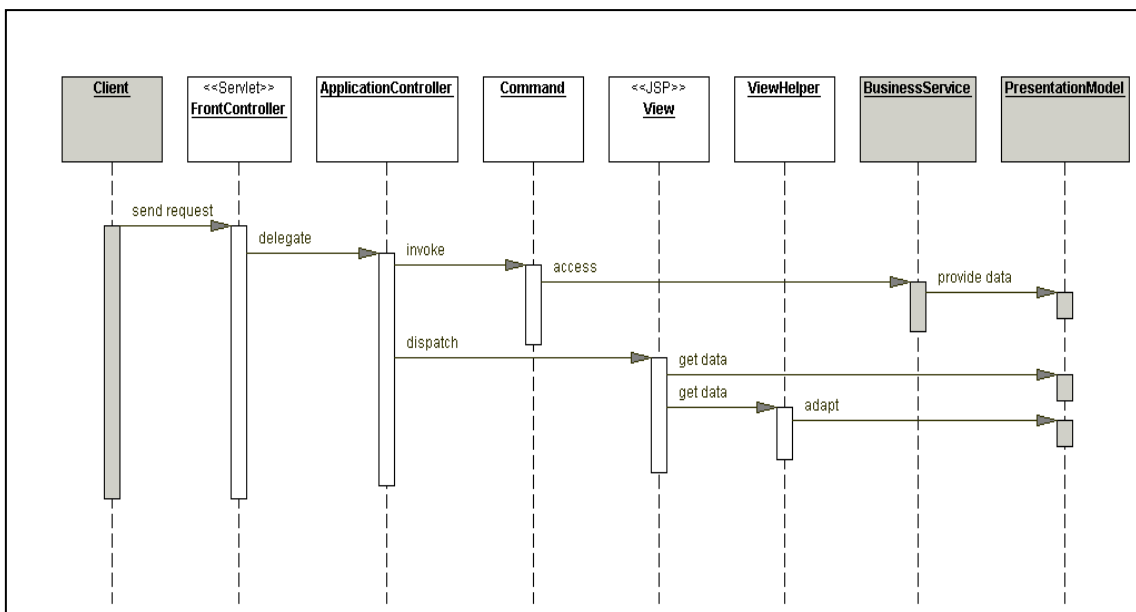
Se desea realizar el manejo básico solicitud e invocar la lógica de negocio antes de que el flujo de control llegue a la vista. Esto implica

- Ejecutar la lógica de negocio específico para atender una petición con el fin de recuperar el contenido que va a ser usado para generar una respuesta dinámica.
- Ver las selecciones que pueden depender de las respuestas de las invocaciones del servicio de negocio.

El patrón Service to Worker centraliza el control y el manejo de solicitud para recuperar un modelo de presentación antes de devolver el control a la vista. La vista genera una respuesta dinámica basada en el modelo de presentación.



Y como ejemplo, una secuencia de ejecución

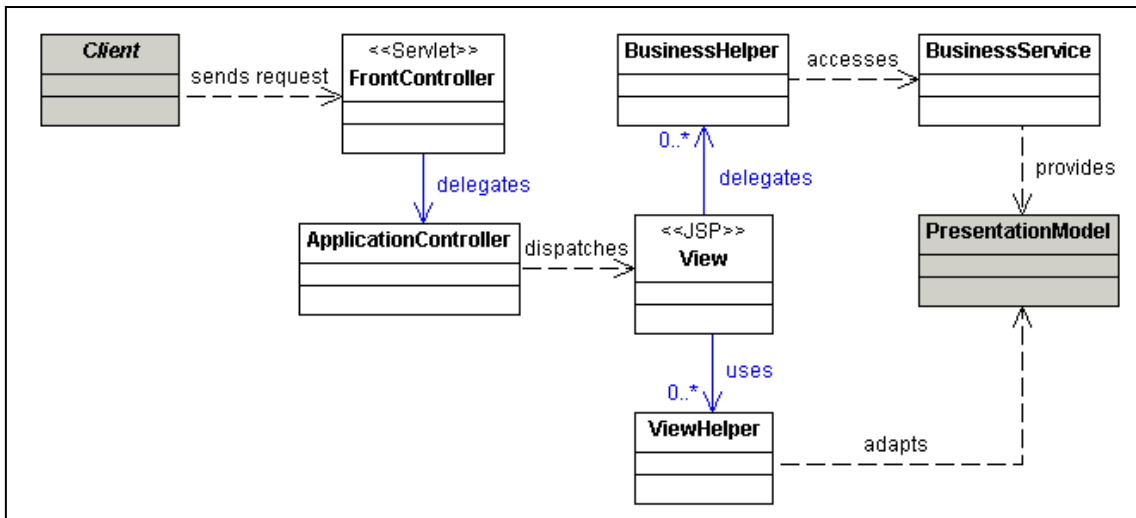


- Dispatcher View

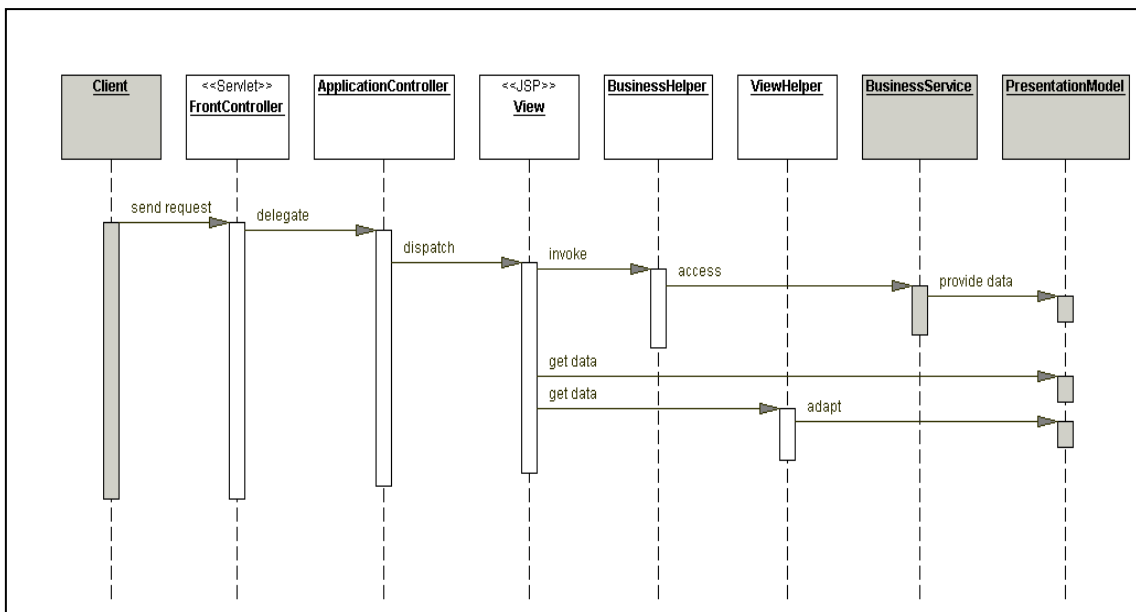
Se utiliza cuando las respuestas a las peticiones implican poco proceso a nivel de la lógica de negocio. Esto conlleva que para que su uso sea adecuado

- Nuestras jsp deben ser estáticas
- Los datos a mostrar en nuestras jsp estarán generados previamente.
- Nuestras jsp son independientes de la lógica de negocio.
- Los datos a mostrar en las jsp conllevan poco proceso.

Cuando se sigue este patrón, las propias jsp servirán como punto de acceso a la aplicación, lo que conlleva que el acceso a la capa de negocio se deba mantener en mínimos.



Y como ejemplo, una secuencia de ejecución

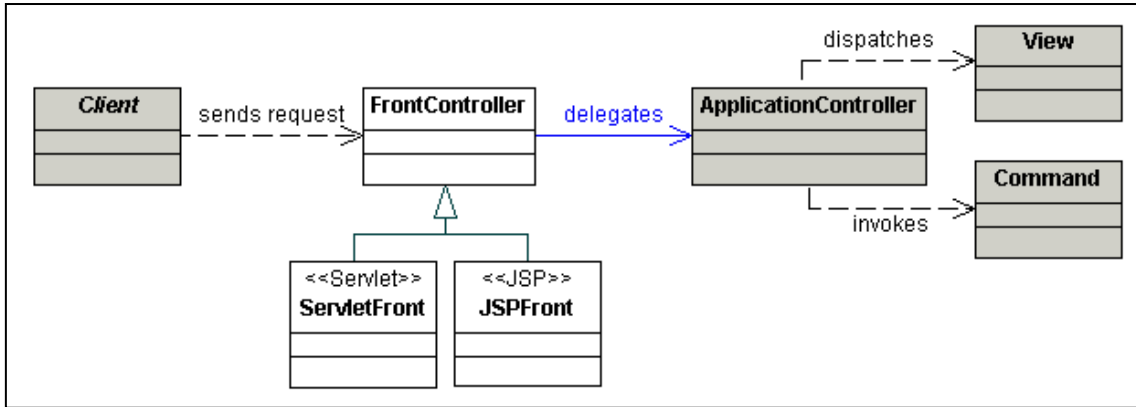


- Front Controller

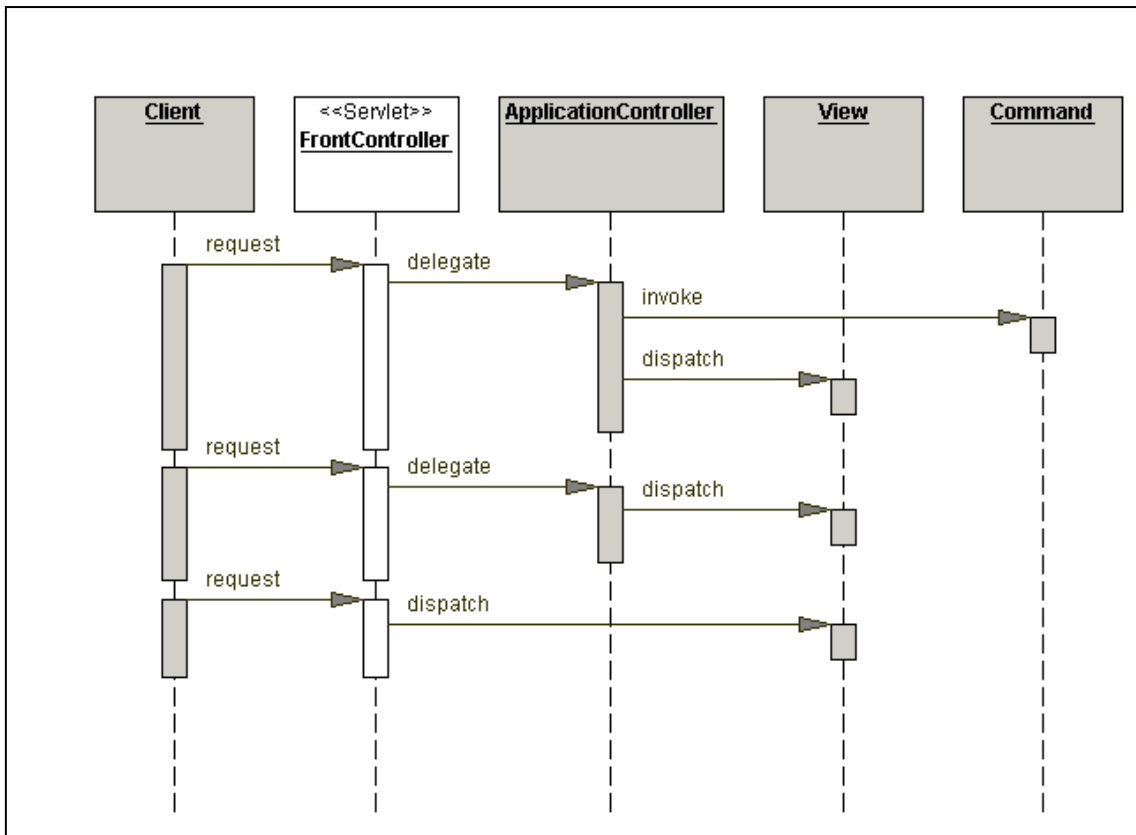
Se utiliza cuando se quiere centralizar el acceso a la capa de la presentación. Es decir, cuando quieres

- Evitar duplicar la lógica de control.
- Aplicar una lógica común a varios tipos de solicitud.
- Separar la lógica de negocio de la capa de vista de la aplicación

Usando un Front Controller se centraliza toda la lógica de control que de otro modo podría duplicarse, así como gestiona los distintos modos de tratar las peticiones que se reciben. Podemos comprobar que los patrones de diseño anteriores parten del uso de este.



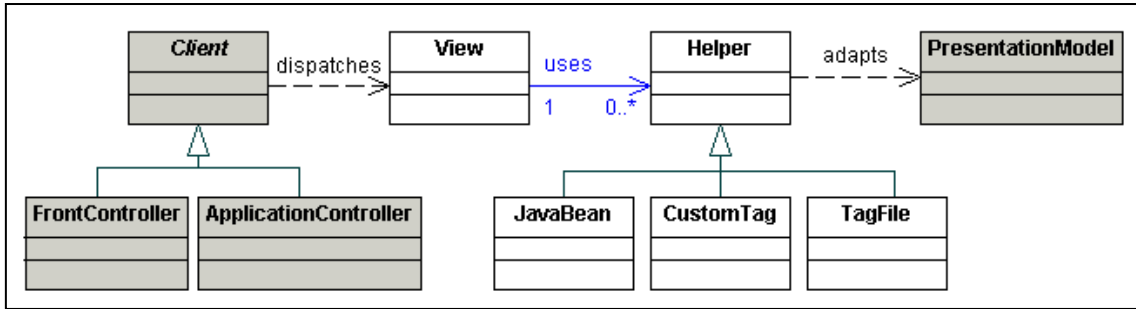
Y un ejemplo de una secuencia de ejecución que muestra la gestión de varios tipos de petición:



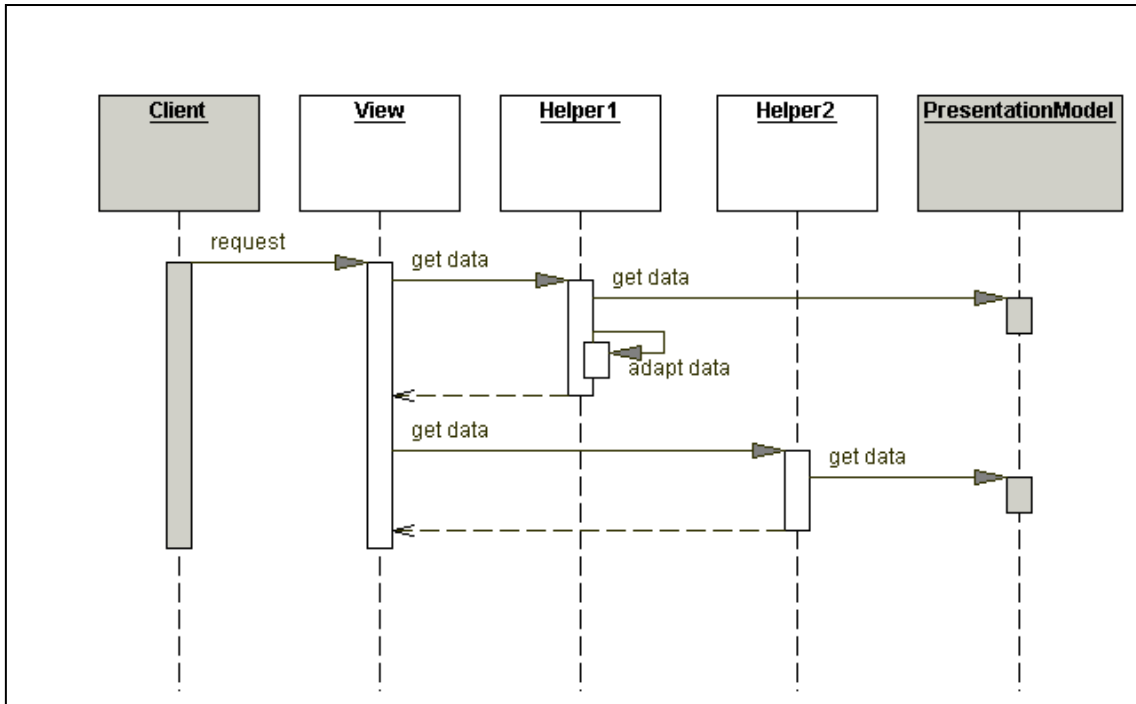
- View Helper

Se utiliza cuando se quiere separar la vista de la lógica de negocio, es decir, se quiere evitar embeber lógica de negocio en la propia vista y separar lo que es la labor de un diseñador web y un programador.

Usando este patrón, las vistas se encargarán únicamente de lo que es puramente formateo de la información a mostrar mientras que los Helpers encapsularán la lógica de negocio necesaria para la vista. Los Helpers pueden estar implementados como POJOs o custom tags.



Y un ejemplo de una secuencia de ejecución.



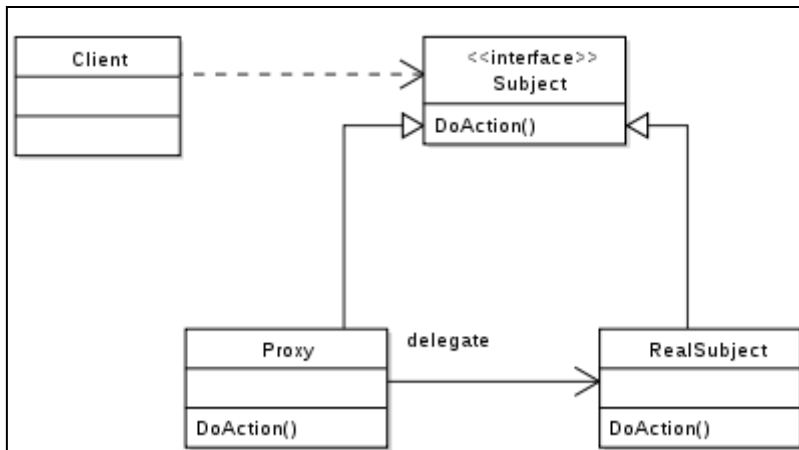
Spring

- Proxy

Este patrón de diseño se sigue cuando

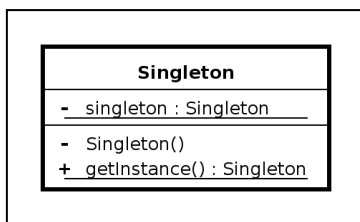
- Se quiere añadir acceso de seguridad a un objeto existente. El *proxy* determinará si el cliente puede acceder al objeto de interés (*proxy* de protección).
- Se quiere proporcionar interfaz de recursos remotos como el servicio web o recursos REST.
- Se quiere coordinar las operaciones costosas en recursos remotos pidiendo los recursos a distancia para iniciar la operación tan pronto como sea posible antes de acceder a los recursos.
- Se quiere agregar una operación segura para los subprocessos a una clase existente sin cambiar el código de la clase existente.

De este modo, el cliente realmente trabajará contra el *Proxy* de modo que se podrán realizar operaciones de control en la capa intermedia que añade, de modo transparente para las capas que separa, debido a que el contrato será el mismo para el *Proxy* que el objeto que representa.



- Singleton

El patrón de diseño Singleton (instancia única) está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.

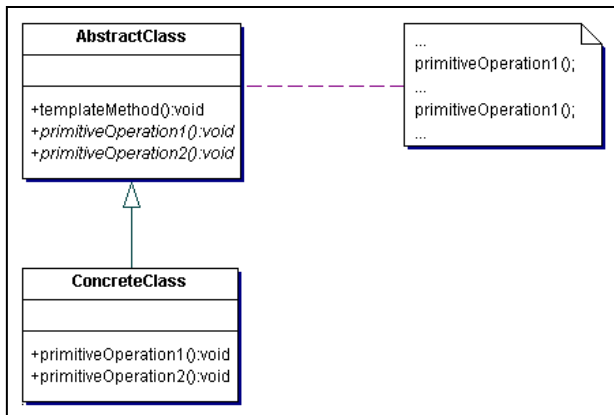


- Template Method (Método Plantilla)

Dentro del marco de la programación orientada a objetos, el patrón Template Method es un patrón de diseño enmarcado dentro de los llamados patrones de comportamiento, que se caracteriza por la definición, dentro de una operación de una superclase, de los pasos de un algoritmo, de forma que todos o parte de estos pasos son redefinidos en las subclases herederas de la citada superclase.

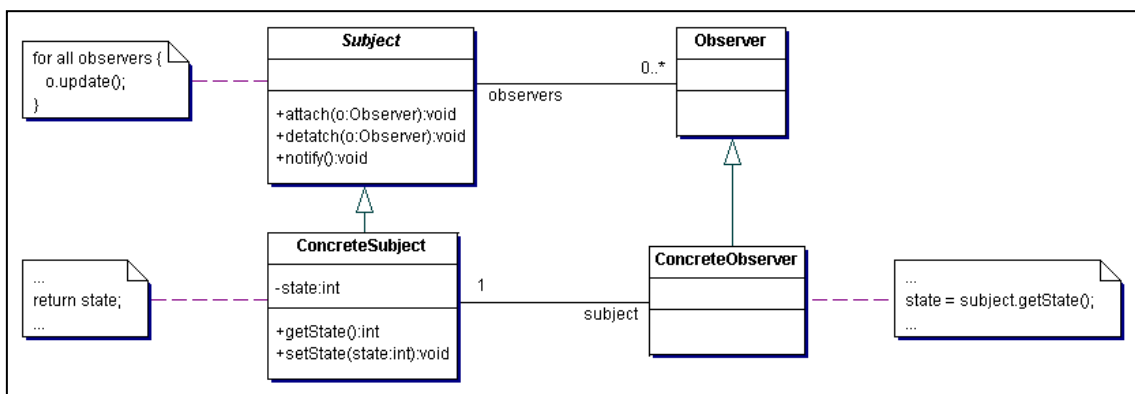
La utilización de este patrón es adecuada en los siguientes casos:

- Cuando contamos con un algoritmo con varios pasos que no cambian, de modo que dichos pasos invariantes serían implementados en una superclase, dejando la implementación de los pasos que cambian para las subclases.
- Para evitar la replicación de código mediante generalización: se factoriza el comportamiento común de varias subclases en una única superclase.
- Para controlar las extensiones de las subclases. El Método Plantilla utiliza métodos especiales (métodos de enganche o hooks) en ciertos puntos, siendo los únicos puntos que pueden ser redefinidos y, por tanto, los únicos puntos donde es posible la extensión.



- Observer

Se utiliza este patrón cuando se quiere definir una dependencia uno-a-muchos entre objetos, de tal forma que cuando el objeto cambie de estado, todos sus objetos dependientes sean notificados automáticamente. Se trata de desacoplar la clase de los objetos clientes del objeto, aumentando la modularidad del lenguaje, creando las mínimas dependencias y evitando bucles de actualización.



Hibernate

- Lazy Loading

Es un patrón de diseño comúnmente utilizado para aplazar la inicialización de un objeto hasta el momento en que se necesita. Puede contribuir a la eficiencia en el funcionamiento del programa, si se utiliza adecuadamente y correctamente.

- Unit of Work

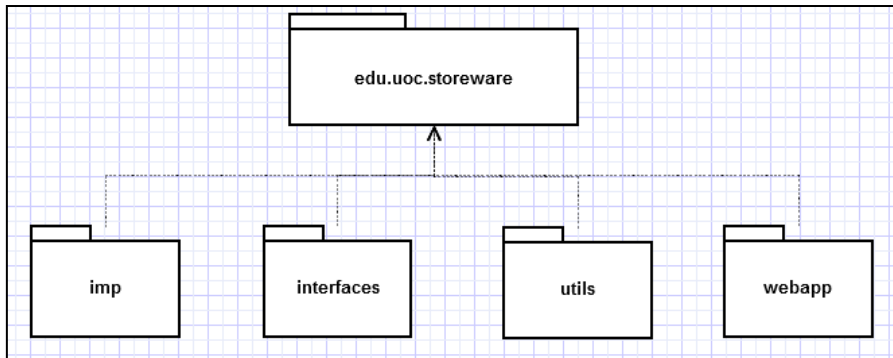
Es un patrón de diseño que se encarga de mantener la lista de los objetos afectados por una transacción y coordina las operaciones de escritura de los cambios y resuelve los problemas de concurrencia. Si se sigue este patrón se debe

- Gestionar las transacciones
- Ordenar las inserciones, borrados y modificaciones en la base de datos.
- Prevenir actualizaciones duplicadas. Si en una unidad de trabajo se marca un objeto como modificado varias veces, se debería generar una única orden de actualización en

la base de datos de modo que la información almacenada represente correctamente todos los cambios realizados.

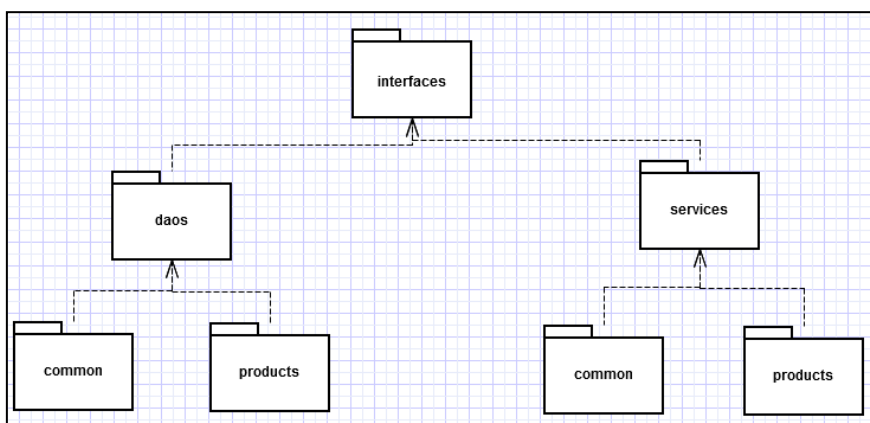
Estructura de paquetes de la aplicación

Empezamos a ver como se estructurará la implementación de la aplicación planteada para este trabajo fin de carrera, por lo que, creo conveniente especificar la estructura de paquetes en los que se estructurará la aplicación, ya que permite también ver las necesidades que se tienen a la hora de implementar la solución. A continuación podemos ver el primer nivel de jerarquización de paquetes:



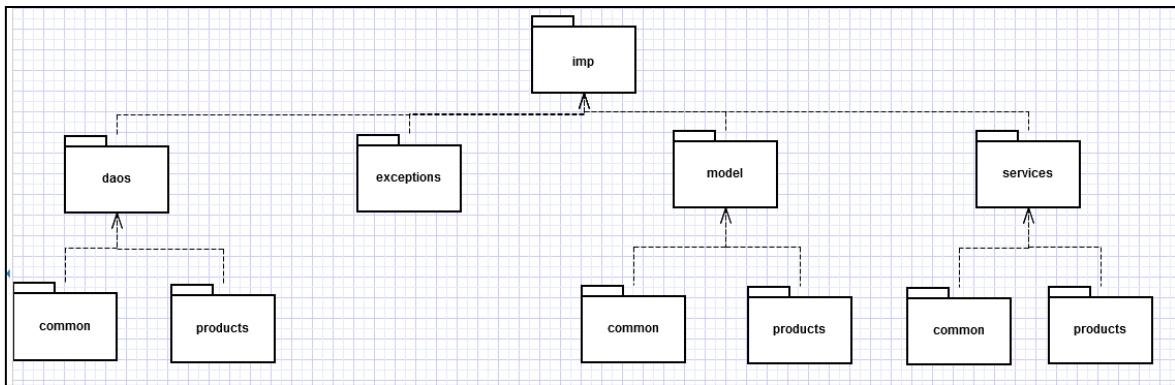
Como vemos, la estructura de la aplicación se dividirá en cuatro grandes bloques. El paquete **interfaces** que será en el que se definan por medio de interfaces los contratos que deben cumplir aquellas clases que implementen la lógica de negocio, así como la capa de acceso a datos. El paquete **imp** que será el que incluya aquellas clases que implementen los contratos definidos en el paquete anterior. El paquete **utils** que será el que albergue las clases de utilidad que sirvan de soporte a operaciones que van más allá de la lógica de negocio. Por último, el paquete **webapp** contendrá todo lo relacionado a la aplicación como herramienta web.

A continuación se desglosa el contenido del paquete **interfaces**:



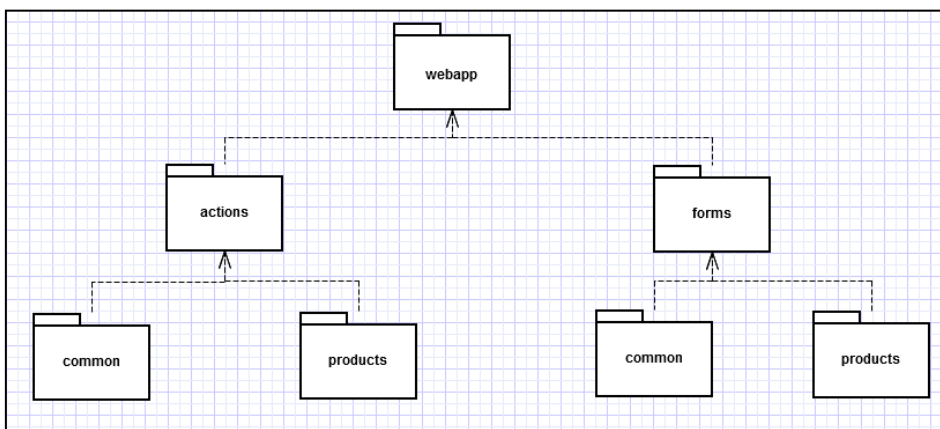
Podemos ver que el paquete se dividirá en dos subpaquetes. El paquete **services** que albergará, estructurado en dos subpaquetes los contratos que definirán las fachadas de la lógica de negocio desde las que accederán a la misma las clases controladoras. A su vez, el paquete **daos**, seguirá cierto paralelismo definiendo los contratos de las clases que ofrezcan el acceso a la capa de acceso a los datos.

A continuación se muestra el contenido del paquete **imp**:



Si bien, en general, hay un paralelismo claro con el paquete **interfaces**, podemos comprobar que aparecen dos nuevos paquetes. El de **exceptions**, no hay nada que comentar, ya que es claro que únicamente almacenará las excepciones que se definirán para manejar la aplicación. El paquete **model**, contendrá las clases necesarias para representas la lógica de negocio de la aplicación.

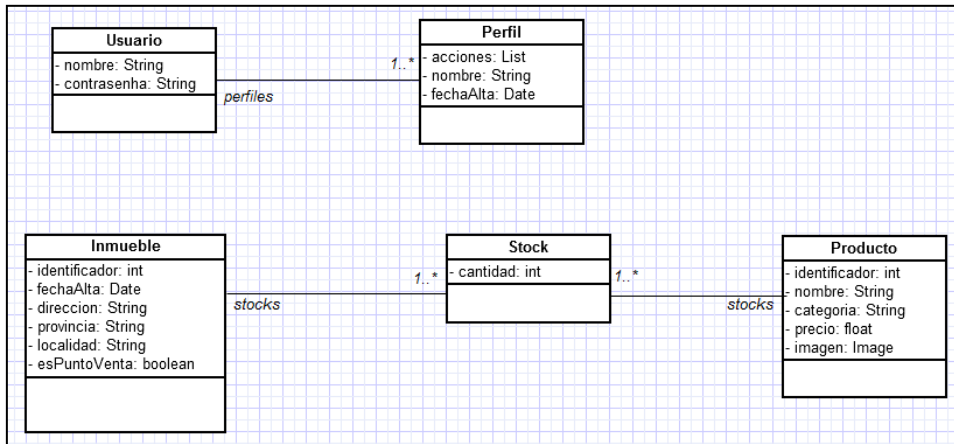
A continuación se pueden ver el contenido del paquete **webapp**:



El paquete está dividido en dos subpaquetes. El primero de ellos, el **actions**, contendrá las clases encargadas de realizar el papel de controlador de la aplicación, atendiendo las peticiones entrantes y usando las fachadas de la lógica de negocio para obtener la información necesaria para la vista. El paquete **forms**, contendrá las clases en las que se apoyará la vista para mostrar la información de la lógica de negocio.

La capa de la lógica de negocio

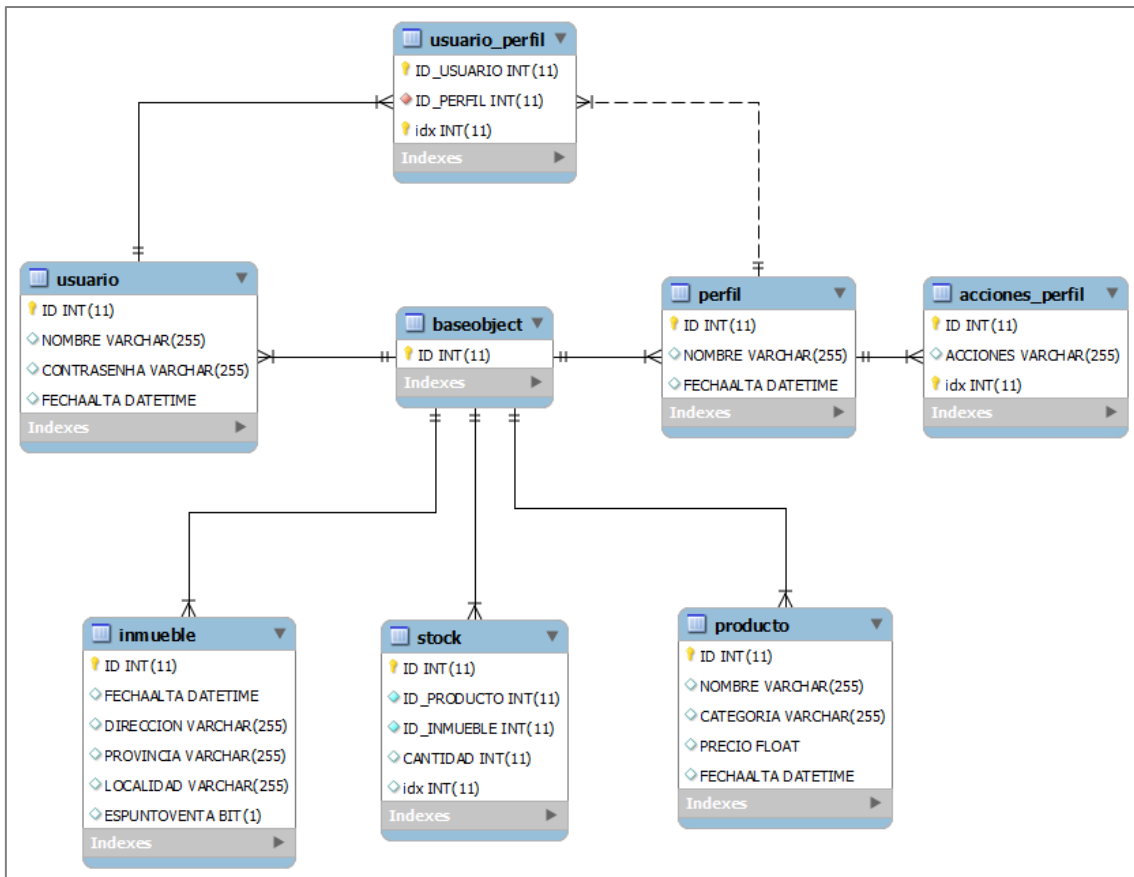
Una vez que tenemos una percepción global de lo que va a conllevar la implementación de la aplicación, entramos a ver como se llevaría a cabo la representación de la lógica de negocio. A continuación un pequeño diagrama que representa las clases necesarias, así como sus relaciones:



Como podemos observar, en la aplicación hay dos partes bien separadas. Por un lado tenemos lo referente propiamente a la aplicación en sí misma, donde como ya hemos visto que debe permitir las operaciones básicas de gestión de perfiles, que serán los que especifiquen que acciones pueden realizar cada usuario. Por tanto, afloran dos entidades que son claramente necesarias, aquellas que representan dichos perfiles y otra que representa a cada uno de los usuarios de la herramienta y que tendrá una relación 1 a N con los perfiles. Señalar en este punto que a día de hoy se deja la puerta abierta a que finalmente el concepto de acción termine ganando peso y teniendo una clase que la represente, ya que aunque en este proyecto dichas acciones se tratarán como un elemento estático de la herramienta, podría llegar a resultar interesante una gestión de las mismas de forma dinámica, por lo que la ampliación se facilitaría en caso de que ya estuviese planteada inicialmente como un entidad.

Por otro lado tenemos la lógica de negocio de la aplicación de forma estricta, es decir, aquella que representa los elementos finales a gestionar, que no son otros que los inmuebles y los productos. No hay mucho que señalar más allá de lo que el diagrama representa, comentar que se está estudiando la opción de que la dirección, provincia y localidad se encapsule en una clase diferenciada, para su posible reutilización futura.

A continuación se muestra el diagrama que muestra la representación en base de datos de la estructura de la lógica de negocio:



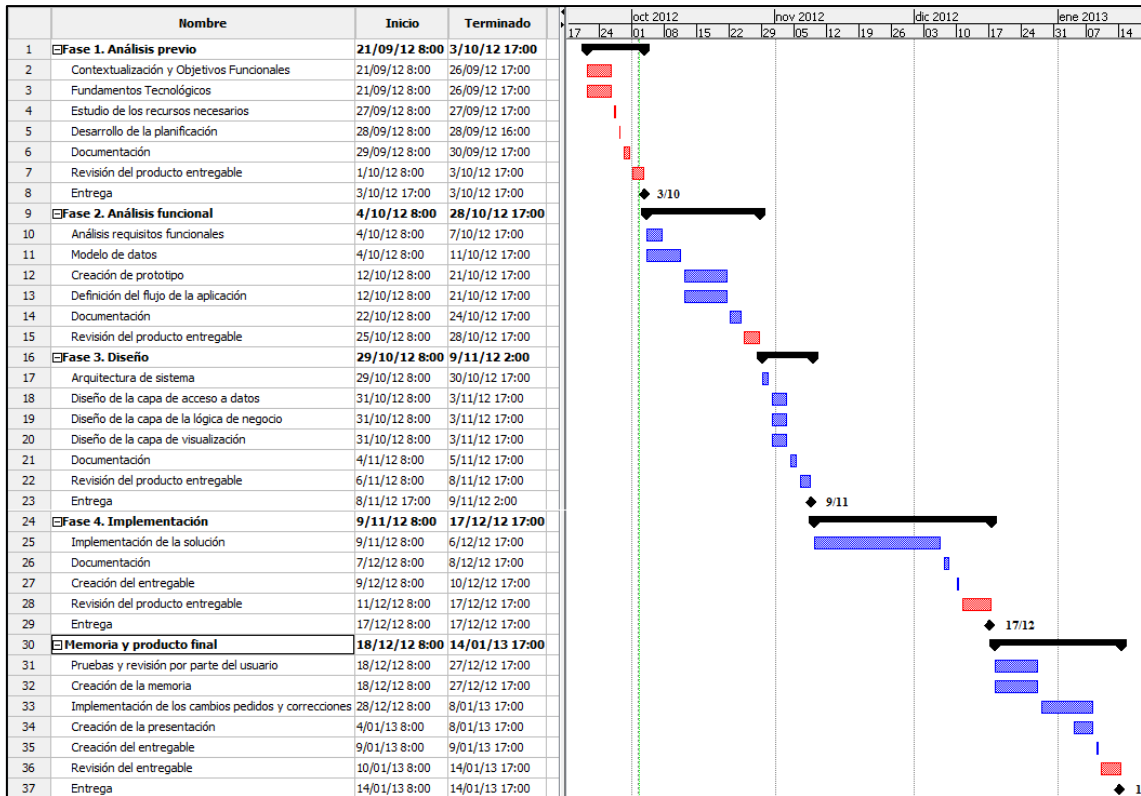
Planificación

A lo largo de este apartado se señalará la planificación que se plantea para el proyecto y que debe servir de guía a lo largo del proceso de desarrollo para saber si estamos sufriendo una desviación que imposibilite la entrega del producto final. Obviamente se ha llevado a cabo teniendo en cuenta que hay hitos que se deben cumplir para aprobar la evaluación continua de la asignatura.

La tabla siguiente muestra la distribución de las diferentes tareas. Señalar que se ha partido de la idea de que existirá un periodo tras la tercera prueba de evaluación hasta la entrega final del producto tomando la fecha de dicho ejemplo como la real, en el caso de que esto no sea así habría que replanificar el proyecto. Por otro lado, debido a que los hitos de entrega son inalterables (no se le puede aplicar ningún factor de riesgo), se ha optado por reservar un tiempo proporcional a cada fase en forma de revisión del producto entregable (se absorbe el riesgo en la planificación del proyecto).

Tarea	Previsión Días	Fecha Inicio	Fecha Fin
Fase 1. Análisis previo	12	21/09/2012	03/10/2012
Contextualización y Objetivos Funcionales	6	21/09/2012	26/09/2012
Fundamentos Tecnológicos	6	21/09/2012	26/09/2012
Estudio de los recursos necesarios	1	27/09/2012	27/09/2012
Desarrollo de la planificación	1	28/09/2012	28/09/2012
Documentación	2	29/09/2012	30/09/2012
Revisión del producto entregable	3	01/10/2012	03/10/2012
Entrega	-	03/10/2012	03/10/2012
Fase 2. Análisis funcional	23	04/10/2012	28/10/2012
Análisis requisitos funcionales	4	04/10/2012	07/10/2012
Modelo de datos	4	08/10/2012	11/10/2012
Creación de prototipo	10	12/10/2012	21/10/2012
Definición del flujo de la aplicación	10	12/10/2012	21/10/2012
Documentación	3	22/10/2012	24/10/2012
Revisión del producto entregable	4	25/10/2012	28/10/2012
Fase 3. Diseño	11	29/10/2012	08/11/2012
Arquitectura de sistema	2	29/10/2012	30/10/2012
Diseño de la capa de acceso a datos	4	31/10/2012	03/11/2012
Diseño de la capa de la lógica de negocio	4	31/10/2012	03/11/2012
Diseño de la capa de visualización	4	31/10/2012	03/11/2012
Documentación	2	04/11/2012	05/11/2012
Revisión del producto entregable	3	06/11/2012	08/11/2012
Entrega	-	08/11/2012	08/11/2012
Fase 4. Implementación	36	09/11/2012	17/12/2012
Implementación de la solución	25	09/11/2012	06/12/2012
Documentación	2	07/12/2012	08/12/2012
Creación del entregable	2	09/12/2012	10/12/2012
Revisión del producto entregable	7	11/12/2012	17/12/2012
Entrega	-	17/12/2012	17/12/2012
Memoria y producto final	28	18/12/2012	14/01/2013
Pruebas y revisión por parte del usuario	10	18/12/2012	27/12/2012
Creación de la memoria	10	18/12/2012	27/12/2012
Implementación de los cambios pedidos y correcciones	12	28/12/2012	08/01/2013
Creación de la presentación	5	04/01/2013	08/01/2013
Creación del entregable	1	09/01/2013	09/01/2013
Revisión del entregable	5	10/01/2013	14/01/2013
Entrega	-	14/01/2013	14/01/2013

A continuación se puede ver, por medio de un diagrama de Gantt, como se van paralelizando las distintas tareas a realizar a lo largo del proyecto:



Manual de usuario

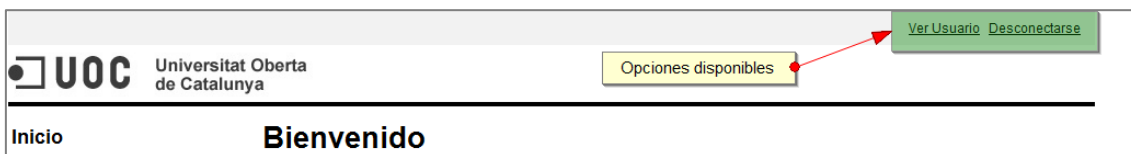
Para ayudar al usuario en su primera aproximación a la aplicación del presente proyecto, en este punto se señalará como se debe proceder para realizar las actividades más utilizadas de la aplicación.

Con el objetivo de facilitar la explicación que se llevará a cabo a lo largo de este capítulo, se definirán la estructura principal y compartida por todas las pantallas que forman parte de la aplicación. Para ello nos apoyaremos en la siguiente captura:



La captura se corresponde con la pantalla que ve el usuario tras conectarse a la aplicación y como se puede observar, la estructura de dicha pantalla se compone de cuatro grandes bloques:

- **Cabecera:** Ocupa la parte superior de la pantalla, y muestra la imagen que hace la función del logo de la aplicación, además de las dos opciones disponibles para todo usuario que se conecte a la aplicación:



- **Ver usuario:** Que permite ver la información del usuario con el que está trabajando en la aplicación.
 - **Desconectarse:** Es la opción que el usuario debe utilizar en caso de desear salir de la aplicación
- **Menú Lateral:** Permite el acceso a todas las opciones que tiene disponibles el usuario. De este modo, dependiendo del usuario el menú será diferente. En la captura utilizada de ejemplo se visualizan todas las opciones de la aplicación, sin embargo con otro usuario podría ver un menú como el siguiente:



- **Pie:** Parte común a todas las páginas y cuyo contenido es completamente estático.
- **Región Central:** Será el que muestre la funcionalidad de cada una de las pantallas.

Una vez que hemos visto la estructura general de las páginas podemos hacer un recorrido de las mismas viendo las distintas opciones que nos permite, teniendo, como resultado final una comprensión total de la funcionalidad de la aplicación. Señalar que a la hora de realizar este manual se ha utilizado un usuario con todos los permisos de acceso a la aplicación, por lo que en todo momento se verán todas las opciones en el menú lateral pero puede darse que el usuario no tenga porque verlas, hay que tenerlo presente.

Página de Bienvenida

A continuación, una captura de la misma:



No hay mucho que comentar de la misma, puesto que únicamente tiene la función de dirigir al usuario que ha accedido a la aplicación al menú lateral para que sepa cómo llegar a las pantallas que le permite realizar las tareas que tiene asignadas.

Pantalla de Consulta de Perfiles

Se trata de la pantalla que se alcanza pulsando sobre la opción del menú lateral etiquetada como "Perfiles". A continuación se muestra la apariencia de dicha pantalla:

Ver Usuario Desconectarse

UOC Universitat Oberta de Catalunya

Consulta de Perfiles

Identificador Acciones

Fecha desde hasta

Buscar

Mostrar 10 registros

Nombre	Fecha de Alta	Acciones
Administracion	11/01/2013	Perfiles, Usuarios,
Almacenistas	11/01/2013	Stocks,
Comerciales	11/01/2013	Productos, Stocks,
Contabilidad	11/01/2013	Inmuebles, Productos,
Sistemas	11/01/2013	Perfiles, Usuarios, Inmuebles, Productos, Stocks,

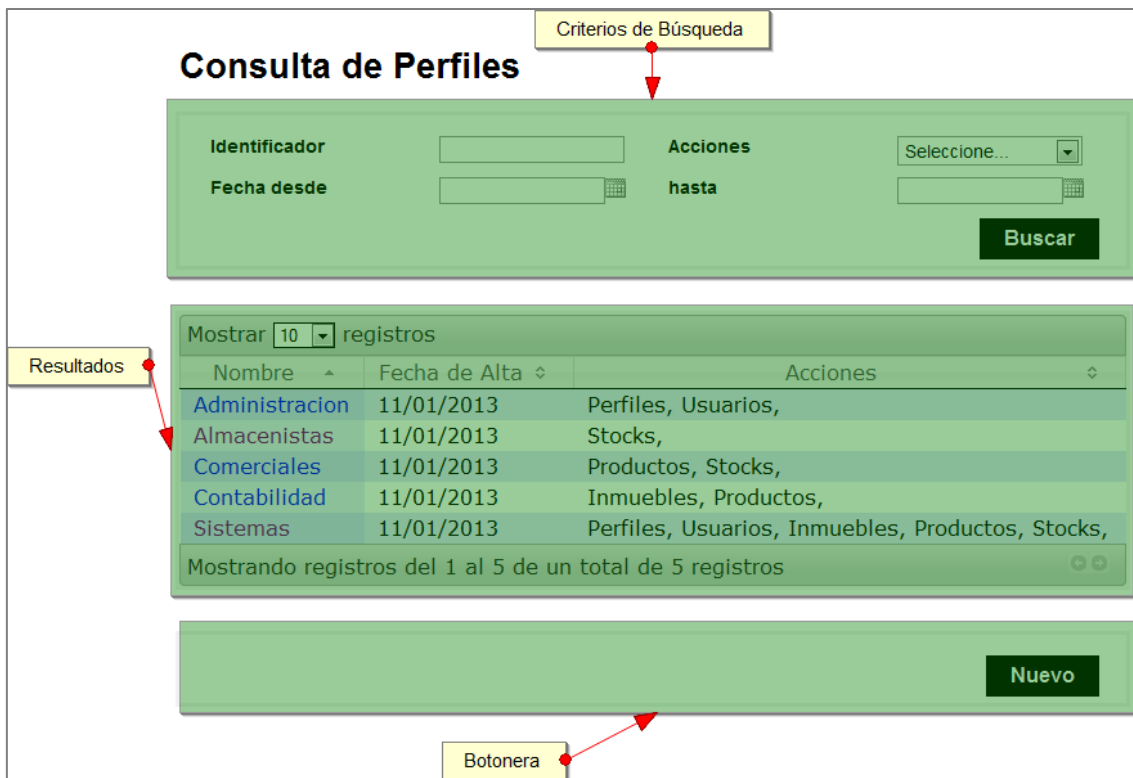
Mostrando registros del 1 al 5 de un total de 5 registros

Nuevo

Autor: David López Martín

Antes de comenzar a explicar el funcionamiento de la pantalla es imprescindible comprender el significado y el uso del concepto de perfil dentro de la aplicación. Cuando se gestionan los usuarios de la aplicación, realmente no se le asignarán las acciones directamente a ellos, si no que se harán a través de los perfiles, ya que el mantenimiento de las funciones de los usuarios se volvería un problema en cuanto la empresa tenga un número pequeño de empleados. Por lo tanto, lo que se hace es asignarle a los usuarios uno o varios perfiles y en caso de querer modificar las tareas asignadas a un usuario concreto, únicamente habría que asignarle un perfil con esas tareas, pero si por ejemplo quisiésemos añadir una tarea a un grupo de empleados de la empresa únicamente tendríamos que modificar el perfil que todos ellos deberían tener asignados en la aplicación.

En cuanto a la pantalla, lo primero que podemos observar, es que, como se había comentado sigue la estructura general definida al iniciar el manual, por lo que nos centraremos en la región central, que es donde se ubican las funcionalidades específicas de la pantalla. En la siguiente imagen se puede ver un desglose de las zonas que la componen:



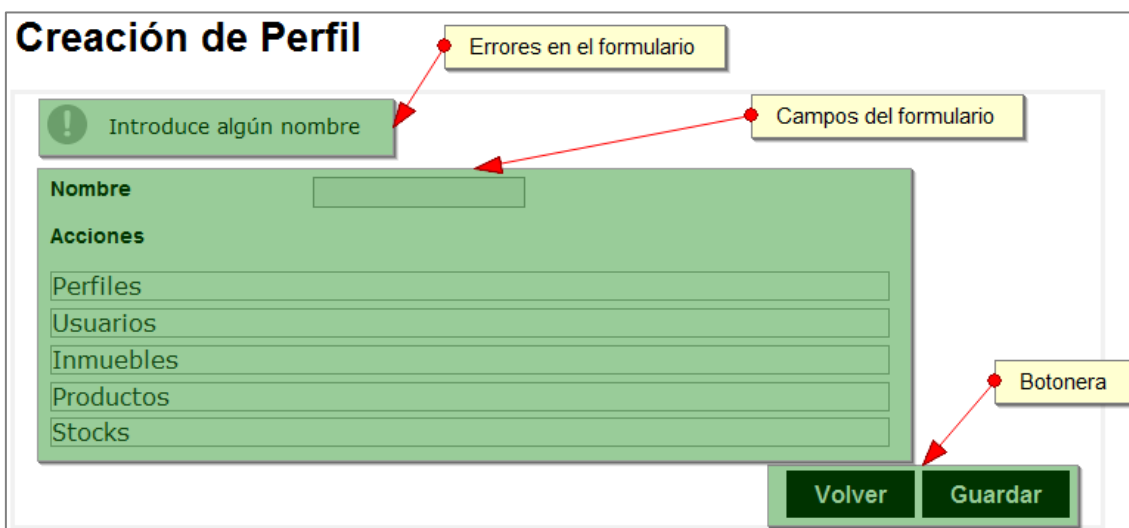
- **Criterios de Búsqueda:** Permite especificar que parámetros deben cumplir los perfiles que se muestran. Las condiciones que se pueden introducir, están el identificador del perfil, una acción asociada, o un intervalo de fechas entre las que se dio de alta dicho perfil.
- **Resultados:** Muestra aquellos perfiles que cumplen los criterios de búsqueda especificados y permite acceder a la pantalla de modificación de los mismos pulsando sobre su nombre.
- **Botonera:** Permite acceder a la pantalla de creación de un nuevo perfil.

Pantalla de Creación de Perfil

Se trata de la pantalla que se alcanza pulsando sobre el botón etiquetado como “Nuevo” en la pantalla de consulta de perfiles. A continuación se muestra la apariencia de dicha pantalla:



En cuanto a la pantalla, lo primero que podemos observar, es que, como se había comentado sigue la estructura general definida al iniciar el manual, por lo que nos centraremos en la región central, que es donde se ubican las funcionalidades específicas de la pantalla. En la siguiente imagen se puede ver un desglose de las zonas que la componen:



- En la zona superior se mostrarán los errores al intentar crear los perfiles, es decir, tras pulsar el botón etiquetado como guardar. Los errores irán desapareciendo conforme se vayan solucionando sin ser necesario volver a pulsar sobre el botón para verificar si están corregidos.
- El formulario de creación está formado por los campos “Nombre” y “Acciones”. Como su propio nombre indica, se corresponde con el nombre del perfil y las acciones permitidas al mismo. Comentar que el campo de las acciones es una lista de selección múltiple que sombrearán aquellas que estén seleccionadas:



- En la botonera tendremos a nuestra disposición la opción de volver a la pantalla de listado de los perfiles sin guardar ningún cambio mediante el botón etiquetado como “Volver” o crear el nuevo perfil pulsando sobre el botón etiquetado como “Guardar”, lo que nos llevará de nuevo a la página de consulta, donde veremos el nuevo perfil creado. Un ejemplo, si en la captura anterior pulsamos sobre guardar:

Consulta de Perfiles

Identificador: Acciones:

Fecha desde: hasta:

Buscar

Mostrar registros

Nombre	Fecha de Alta	Acciones
Administracion	11/01/2013	Perfiles, Usuarios,
Almacenistas	11/01/2013	Stocks,
Comerciales	11/01/2013	Productos, Stocks,
Contabilidad	11/01/2013	Inmuebles, Productos,
Prueba	13/01/2013	Usuarios, Productos,
Sistemas	11/01/2013	Perfiles, Usuarios, Inmuebles, Productos, Stocks,

Mostrando registros del 1 al 6 de un total de 6 registros

Nuevo perfil creado

Nuevo

Autor: David López Martín

Pantalla de Modificación de Perfil

Se trata de la pantalla que se alcanza pulsando sobre el nombre del perfil en la pantalla de consulta de perfiles. A continuación se muestra la apariencia de dicha pantalla:

Modificación de Perfil

Nombre:

Acciones:

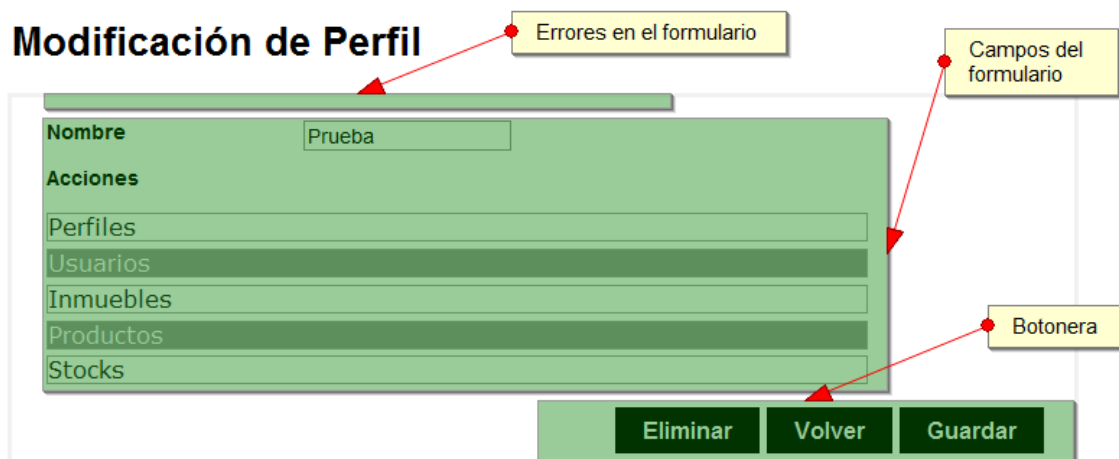
- Perfiles
- Usuarios
- Inmuebles
- Productos
- Stocks

Eliminar **Volver** **Guardar**

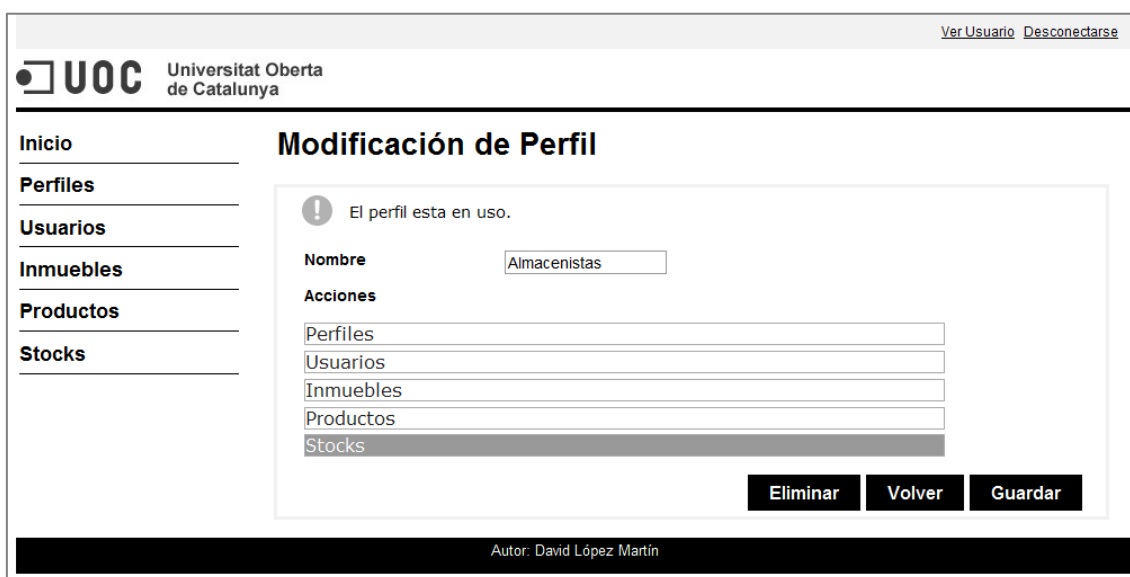
Autor: David López Martín

En cuanto a la pantalla, lo primero que podemos observar, es que, como se había comentado sigue la estructura general definida al iniciar el manual, por lo que nos centraremos en la región central, que es donde se ubican las funcionalidades específicas de la pantalla. En la siguiente imagen se puede ver un desglose de las zonas que la componen:

Modificación de Perfil



- En la zona superior se mostrarán los errores al intentar modificar los perfiles, es decir, tras pulsar el botón etiquetado como guardar. Los errores irán desapareciendo conforme se vayan solucionando sin ser necesario volver a pulsar sobre el botón para verificar si están corregidos.
- El formulario de creación está formado por los campos “Nombre” y “Acciones”. Como su propio nombre indica, se corresponde con el nombre del perfil y las acciones permitidas al mismo. Comentar que el campo de las acciones es una lista de selección múltiple que sombrearán aquellas que estén seleccionadas como se ve en la captura anterior.
- En la botonera tendremos a nuestra disposición la opción de volver a la pantalla de listado de los perfiles sin guardar ningún cambio mediante el botón etiquetado como “Volver”, modificar el perfil pulsando sobre el botón etiquetado como “Guardar”, lo que nos llevará de nuevo a la página de consulta, donde veremos el nuevo perfil modificado o eliminar el perfil que estamos modificando. En el caso de intentar eliminar un perfil que está en uso, la aplicación nos dará un aviso, y no permitirá su borrado hasta que se hagan las operaciones necesarias para solucionar la situación. Un ejemplo:



Pantalla de Consulta de Usuarios

Se trata de la pantalla que se alcanza pulsando sobre la opción del menú lateral etiquetada como "Usuarios". A continuación se muestra la apariencia de dicha pantalla:

Ver Usuario Desconectarse

UOC Universitat Oberta de Catalunya

Consulta de Usuarios

Usuario Perfiles Seleccione...
 Fecha desde hasta
 Buscar

Mostrar 10 registros

Usuario	Fecha de Alta	Perfiles
David	12/01/2013	Sistemas,
Francisco	13/01/2013	Administracion,
Jorge	11/01/2013	Almacenistas,
Maria	13/01/2013	Comerciales,

Mostrando registros del 1 al 4 de un total de 4 registros

Nuevo

Autor: David López Martín

Lo primero que podemos observar, es que, como se había comentado sigue la estructura general definida al iniciar el manual, por lo que nos centraremos en la región central, que es donde se ubican las funcionalidades específicas de la pantalla. En la siguiente imagen se puede ver un desglose de las zonas que la componen:

Criterios de búsqueda

Consulta de Usuarios

Usuario Perfiles Seleccione...
 Fecha desde hasta
 Buscar

Resultados

Mostrar 10 registros

Usuario	Fecha de Alta	Perfiles
David	12/01/2013	Sistemas,
Francisco	13/01/2013	Administracion,
Jorge	11/01/2013	Almacenistas,
Maria	13/01/2013	Comerciales,

Mostrando registros del 1 al 4 de un total de 4 registros

Botonera

Nuevo

- **Criterios de Búsqueda:** Permite especificar que parámetros deben cumplir los usuarios que se muestran. Las condiciones que se pueden introducir, están el nombre del

usuario, un perfil asociado, o un intervalo de fechas entre las que se dio de alta dicho usuario.

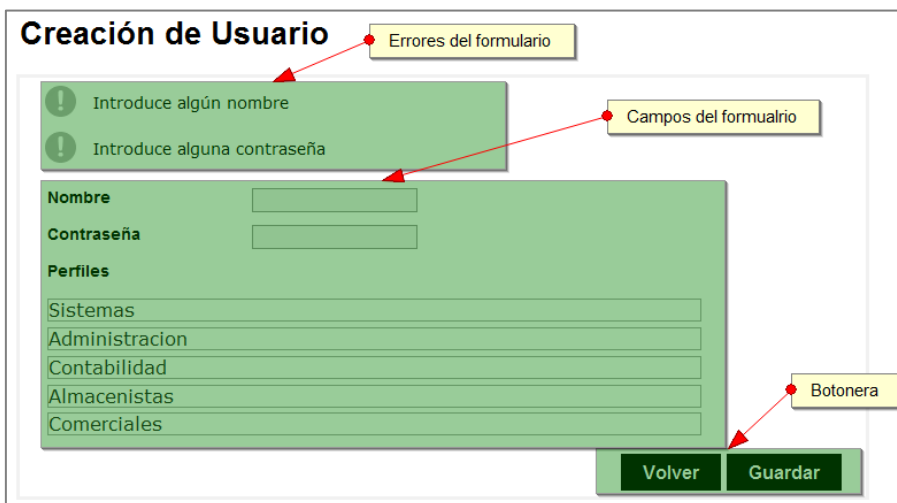
- **Resultados:** Muestra aquellos usuarios que cumplen los criterios de búsqueda especificados y permite acceder a la pantalla de modificación de los mismos pulsando sobre su nombre.
- **Botonera:** Permite acceder a la pantalla de creación de un nuevo usuario.

Pantalla de Creación de Usuario

Se trata de la pantalla que se alcanza pulsando sobre el botón etiquetado como “Nuevo” en la pantalla de consulta de usuarios. A continuación se muestra la apariencia de dicha pantalla:



En cuanto a la pantalla, lo primero que podemos observar, es que, como se había comentado sigue la estructura general definida al iniciar el manual, por lo que nos centraremos en la región central, que es donde se ubican las funcionalidades específicas de la pantalla. En la siguiente imagen se puede ver un desglose de las zonas que la componen:



- En la zona superior se mostrarán los errores al intentar crear los usuarios, es decir, tras pulsar el botón etiquetado como guardar. Los errores irán desapareciendo conforme se vayan solucionando sin ser necesario volver a pulsar sobre el botón para verificar si están corregidos.
- El formulario de creación está formado por los campos “Nombre”, “Contraseña” y “Perfiles”. Como su propio nombre indica, se corresponde con el nombre del usuario, su contraseña y los perfiles asociados al mismo. Comentar que el campo de los perfiles es una lista de selección múltiple que sombrearán aquellos que estén seleccionados:

Ver Usuario Desconectarse

UOC Universitat Oberta de Catalunya

Creación de Usuario

Nombre

Contraseña

Perfiles

- Sistemas
- Administracion
- Contabilidad
- Almacenistas
- Comerciales

Volver Guardar

Autor: David López Martín

- En la botonera tendremos a nuestra disposición la opción de volver a la pantalla de listado de los usuarios sin guardar ningún cambio mediante el botón etiquetado como “Volver” o crear el nuevo usuario pulsando sobre el botón etiquetado como “Guardar”, lo que nos llevará de nuevo a la página de consulta, donde veremos el nuevo usuario creado. Un ejemplo, si en la captura anterior pulsamos sobre guardar:

Ver Usuario Desconectarse

UOC Universitat Oberta de Catalunya

Consulta de Usuarios

Usuario Perfiles Seleccione...
 Fecha desde hasta
 Buscar

Mostrar 10 registros

Usuario	Fecha de Alta	Perfiles
David	12/01/2013	Sistemas,
Francisco	13/01/2013	Administracion,
Jorge	11/01/2013	Almacenistas,
Maria	13/01/2013	Comerciales,
Prueba	13/01/2013	Contabilidad,

Mostrando registros del 1 al 5 de un total de 5 registros

Nuevo usuario creado

Nuevo

Autor: David López Martín

Pantalla de Modificación de Usuario

Esta pantalla se puede alcanzar desde la opción “Ver Usuario” de la cabecera de la pantalla o pulsando sobre el nombre del perfil en la pantalla de consulta de usuarios. A continuación se muestra la apariencia de dicha pantalla:

En cuanto a la pantalla, lo primero que podemos observar, es que, como se había comentado sigue la estructura general definida al iniciar el manual, por lo que nos centraremos en la región central, que es donde se ubican las funcionalidades específicas de la pantalla. En la siguiente imagen se puede ver un desglose de las zonas que la componen:

- En la zona superior se mostrarán los errores al intentar modificar los usuarios, es decir, tras pulsar el botón etiquetado como guardar. Los errores irán desapareciendo conforme se vayan solucionando sin ser necesario volver a pulsar sobre el botón para verificar si están corregidos.
- El formulario de creación está formado por los campos “Nombre”, “Contraseña” y “Perfiles”. Como su propio nombre indica, se corresponde con el nombre del usuario, su contraseña y los perfiles asociados al mismo. Comentar que el campo de los perfiles

es una lista de selección múltiple que sombrearán aquellos que estén seleccionados como se ve en la captura anterior.

- En la botonera tendremos a nuestra disposición la opción de volver a la pantalla de listado de los usuarios sin guardar ningún cambio mediante el botón etiquetado como “Volver”, modificar el usuario pulsando sobre el botón etiquetado como “Guardar”, lo que nos llevará de nuevo a la página de consulta, donde veremos el nuevo perfil modificado o eliminar el perfil que estamos modificando.

Señalar que en el caso de que se acceda desde la opción de la cabecera de las pantallas y el usuario no tenga permisos para acceder a la pantalla de consulta de perfiles, en esta pantalla no se visualizarán los mismos.

Pantalla de Consulta de Inmuebles

Se trata de la pantalla que se alcanza pulsando sobre la opción del menú lateral etiquetada como “Inmueble”. A continuación se muestra la apariencia de dicha pantalla:

Identificador	Fecha de Alta	Dirección	Provincia	Localidad
41	12/01/2013	Centro sur	3	Vigo
6	06/01/2013	Centro Noroeste	1	Culleredo

Lo primero que podemos observar, es que, como se había comentado sigue la estructura general definida al iniciar el manual, por lo que nos centraremos en la región central, que es

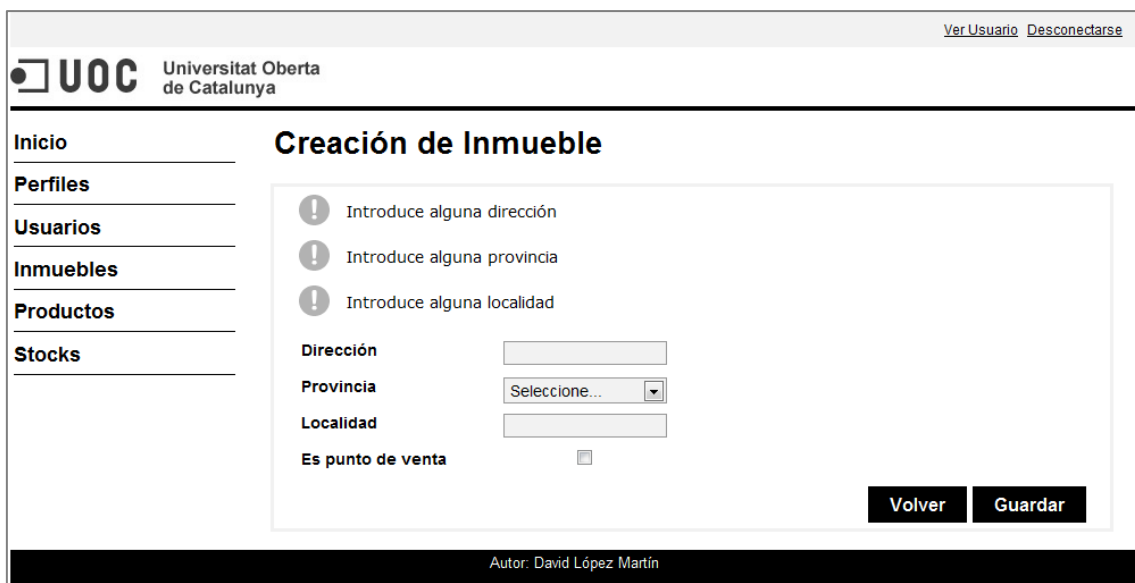
donde se ubican las funcionalidades específicas de la pantalla. En la siguiente imagen se puede ver un desglose de las zonas que la componen:



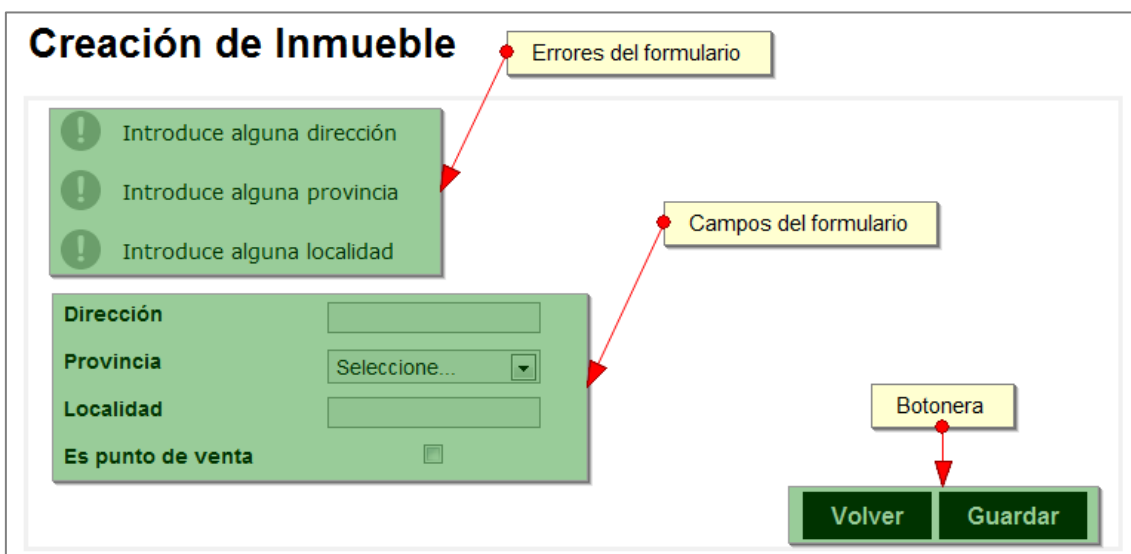
- **Criterios de Búsqueda:** Permite especificar que parámetros deben cumplir los inmuebles que se muestran. Las condiciones que se pueden introducir, están el identificador del inmueble, así como su dirección, provincia o localidad, o un intervalo de fechas entre las que se dio de alta dicho inmueble.
- **Resultados:** Muestra aquellos inmuebles que cumplen los criterios de búsqueda especificados y permite acceder a la pantalla de modificación de los mismos pulsando sobre su identificador.
- **Botonera:** Permite acceder a la pantalla de creación de un nuevo usuario.

Pantalla de Creación de Inmueble

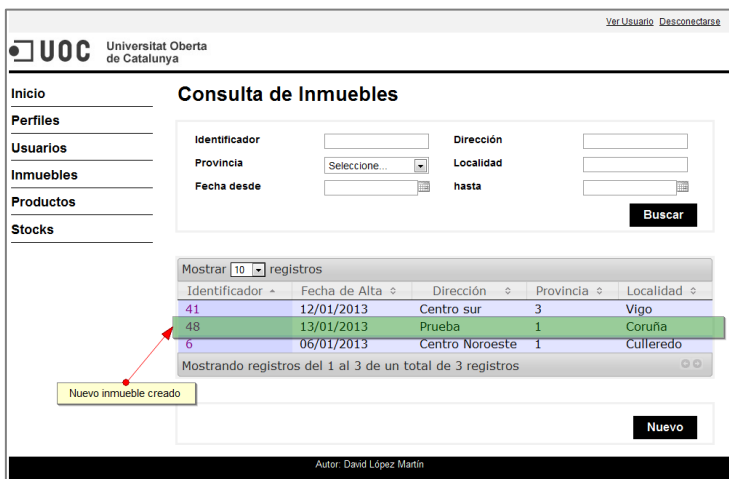
Se trata de la pantalla que se alcanza pulsando sobre el botón etiquetado como “Nuevo” en la pantalla de consulta de inmuebles. A continuación se muestra la apariencia de dicha pantalla:



En cuanto a la pantalla, lo primero que podemos observar, es que, como se había comentado sigue la estructura general definida al iniciar el manual, por lo que nos centraremos en la región central, que es donde se ubican las funcionalidades específicas de la pantalla. En la siguiente imagen se puede ver un desglose de las zonas que la componen:



- En la zona superior se mostrarán los errores al intentar crear los inmuebles, es decir, tras pulsar el botón etiquetado como guardar. Los errores irán desapareciendo conforme se vayan solucionando sin ser necesario volver a pulsar sobre el botón para verificar si están corregidos.
- El formulario de creación está formado por los campos “Dirección”, “Provincia”, “Localidad” y “Es punto de venta”. Como su propio nombre indica, se corresponde con la dirección del inmueble, su provincia y localidad, así como el indicador de si se trata de un punto de venta o no.
- En la botonera tendremos a nuestra disposición la opción de volver a la pantalla de listado de los inmuebles sin guardar ningún cambio mediante el botón etiquetado como “Volver” o crear el nuevo inmueble pulsando sobre el botón etiquetado como “Guardar”, lo que nos llevará de nuevo a la página de consulta, donde veremos el nuevo inmueble creado. Por ejemplo:



Pantalla de Modificación de Inmueble

Se trata de la pantalla a la que se accede pulsando sobre el identificador del inmueble en la pantalla de consulta de los mismos. A continuación se muestra la apariencia de dicha pantalla:

En cuanto a la pantalla, lo primero que podemos observar, es que, como se había comentado sigue la estructura general definida al iniciar el manual, por lo que nos centraremos en la región central, que es donde se ubican las funcionalidades específicas de la pantalla. En la siguiente imagen se puede ver un desglose de las zonas que la componen:

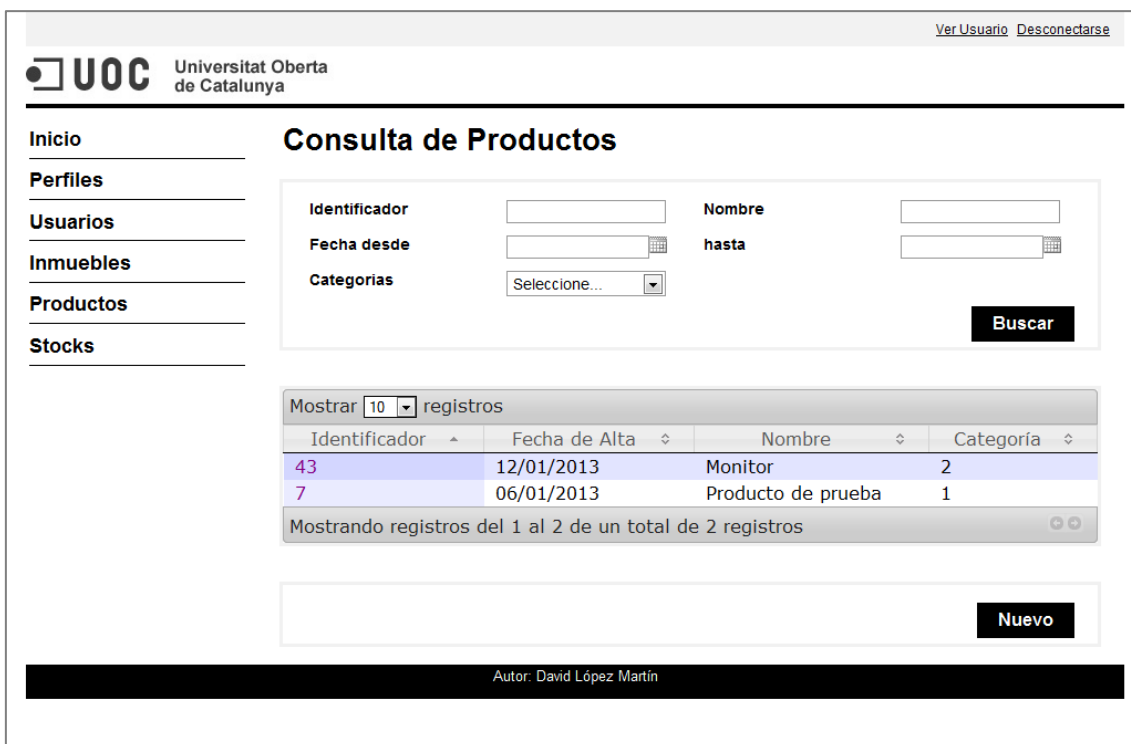
- En la zona superior se mostrarán los errores al intentar modificar los inmuebles, es decir, tras pulsar el botón etiquetado como guardar. Los errores irán desapareciendo conforme se vayan solucionando sin ser necesario volver a pulsar sobre el botón para verificar si están corregidos.
- El formulario de creación está formado por los campos “Dirección”, “Provincia”, “Localidad” y “Es punto de venta”. Como su propio nombre indica, se corresponde con la dirección del inmueble, la provincia y la localidad del mismo, así como el indicador de si se trata de un punto de venta.
- En la botonera tendremos a nuestra disposición la opción de volver a la pantalla de listado de los inmuebles sin guardar ningún cambio mediante el botón etiquetado como “Volver”, modificar el inmueble pulsando sobre el botón etiquetado como “Guardar”, lo que nos llevará de nuevo a la página de consulta, donde veremos el inmueble modificado o eliminar el inmueble que estamos modificando. Señalar que si el inmueble tiene algún stock asociado, no permitirá su borrado hasta que se le

desasigne dicho stock y al intentar borrarlo, la aplicación mostrará un aviso de esta situación en pantalla:



Pantalla de Consulta de Productos

Se trata de la pantalla que se alcanza pulsando sobre la opción del menú lateral etiquetada como "Productos". A continuación se muestra la apariencia de dicha pantalla:



Lo primero que podemos observar, es que, como se había comentado sigue la estructura general definida al iniciar el manual, por lo que nos centraremos en la región central, que es donde se ubican las funcionalidades específicas de la pantalla. En la siguiente imagen se puede ver un desglose de las zonas que la componen:



- **Criterios de Búsqueda:** Permite especificar que parámetros deben cumplir los productos que se muestran. Las condiciones que se pueden introducir, están el identificador del producto, así como su nombre y categoría, o un intervalo de fechas entre las que se dio de alta dicho producto.
- **Resultados:** Muestra aquellos productos que cumplen los criterios de búsqueda especificados y permite acceder a la pantalla de modificación de los mismos pulsando sobre su identificador.
- **Botonera:** Permite acceder a la pantalla de creación de un nuevo producto.

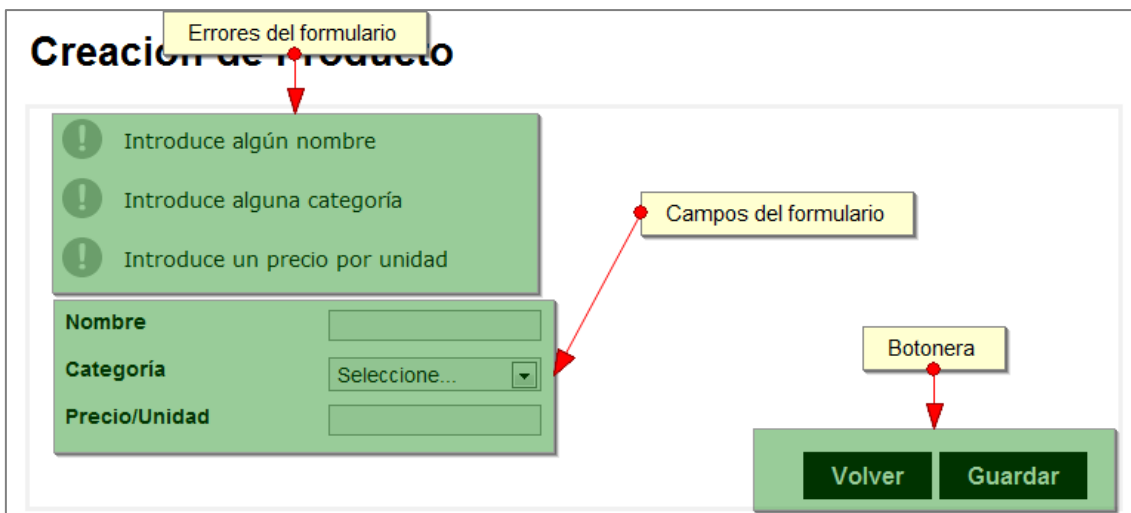
Pantalla de Creación de Producto

Se trata de la pantalla que se alcanza pulsando sobre el botón etiquetado como “Nuevo” en la pantalla de consulta de productos. A continuación se muestra la apariencia de dicha pantalla:

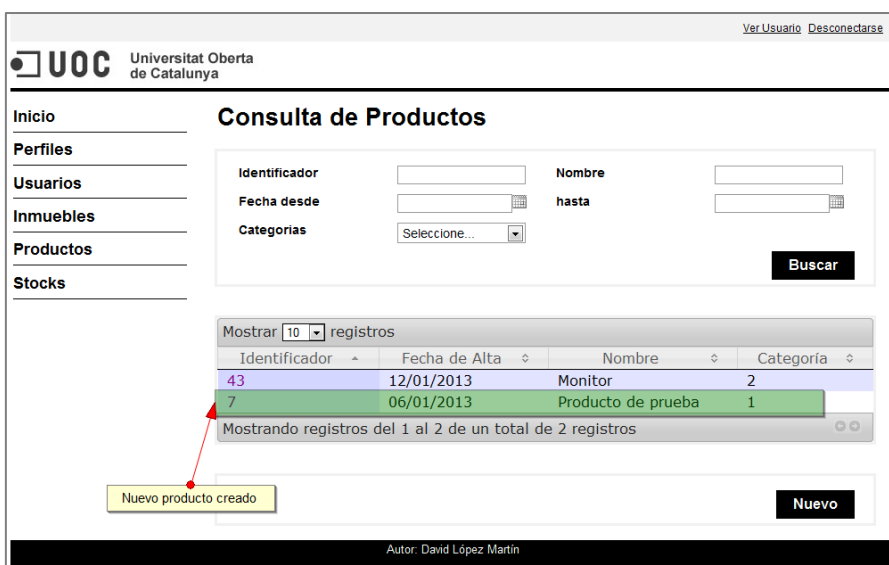


En cuanto a la pantalla, lo primero que podemos observar, es que, como se había comentado sigue la estructura general definida al iniciar el manual, por lo que nos centraremos en la

región central, que es donde se ubican las funcionalidades específicas de la pantalla. En la siguiente imagen se puede ver un desglose de las zonas que la componen:



- En la zona superior se mostrarán los errores al intentar crear los productos, es decir, tras pulsar el botón etiquetado como guardar. Los errores irán desapareciendo conforme se vayan solucionando sin ser necesario volver a pulsar sobre el botón para verificar si están corregidos.
- El formulario de creación está formado por los campos “Nombre”, “Categoria” y “Precio/Unidad”. Como su propio nombre indica, se corresponde con el nombre del producto, su categoría y el precio por unidad del mismo.
- En la botonera tendremos a nuestra disposición la opción de volver a la pantalla de listado de los productos sin guardar ningún cambio mediante el botón etiquetado como “Volver” o crear el nuevo producto pulsando sobre el botón etiquetado como “Guardar”, lo que nos llevará de nuevo a la página de consulta, donde veremos el nuevo producto creado. Por ejemplo:

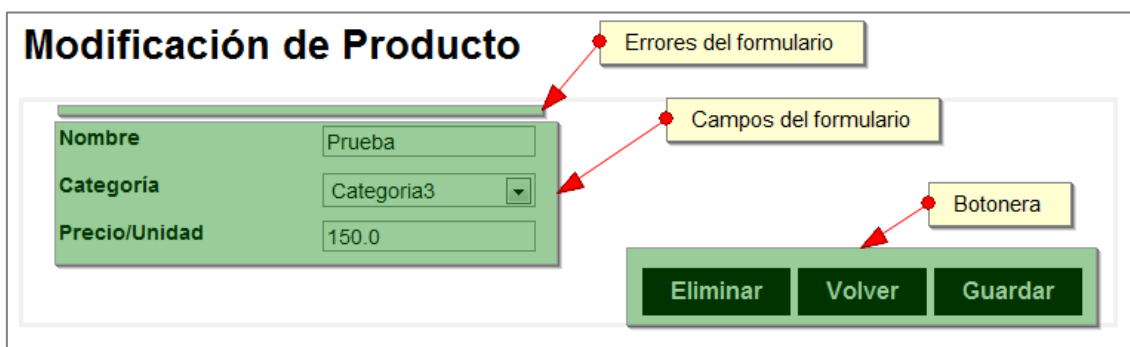


Pantalla de Modificación de Producto

Se trata de la pantalla a la que se accede pulsando sobre el identificador del producto en la pantalla de consulta de los mismos. A continuación se muestra la apariencia de dicha pantalla:



En cuanto a la pantalla, lo primero que podemos observar, es que, como se había comentado sigue la estructura general definida al iniciar el manual, por lo que nos centraremos en la región central, que es donde se ubican las funcionalidades específicas de la pantalla. En la siguiente imagen se puede ver un desglose de las zonas que la componen:



- En la zona superior se mostrarán los errores al intentar modificar los productos, es decir, tras pulsar el botón etiquetado como guardar. Los errores irán desapareciendo conforme se vayan solucionando sin ser necesario volver a pulsar sobre el botón para verificar si están corregidos.
- El formulario de creación está formado por los campos “Nombre”, “Categoría” y “Precio/Unidad”. Como su propio nombre indica, se corresponde con el nombre del producto, la categoría a la que pertenece y el precio por unidad del mismo.
- En la botonera tendremos a nuestra disposición la opción de volver a la pantalla de listado de los productos sin guardar ningún cambio mediante el botón etiquetado como “Volver”, modificar el producto pulsando sobre el botón etiquetado como “Guardar”, lo que nos llevará de nuevo a la página de consulta, donde veremos el producto modificado o eliminar el producto que estamos modificando. Señalar que si el producto tiene algún stock asociado, no permitirá su borrado hasta que se le desasigne dicho stock y al intentar borrarlo, la aplicación mostrará un aviso de esta situación en pantalla:

Pantalla de Consulta de Stocks

Se trata de la pantalla que se alcanza pulsando sobre la opción del menú lateral etiquetada como "Stocks". A continuación se muestra la apariencia de dicha pantalla:

Identificador	Fecha de Alta	Tipo
41	13/01/2013	Inmueble
43	12/01/2013	Producto
6	06/01/2013	Inmueble
7	06/01/2013	Producto

Lo primero que podemos observar, es que, como se había comentado sigue la estructura general definida al iniciar el manual, por lo que nos centraremos en la región central, que es donde se ubican las funcionalidades específicas de la pantalla. En la siguiente imagen se puede ver un desglose de las zonas que la componen:



- **Criterios de Búsqueda:** Permite especificar que parámetros deben cumplir los elementos mostrados en la lista de resultados. En las condiciones que se pueden introducir, están el tipo de elemento a mostrar (inmuebles o productos), así como su identificador o un intervalo de fechas entre las que se dio de alta dicho elemento.
- **Resultados:** Muestra aquellos elementos que cumplen los criterios de búsqueda especificados y permite acceder a la pantalla de modificación de los mismos pulsando sobre su identificador.

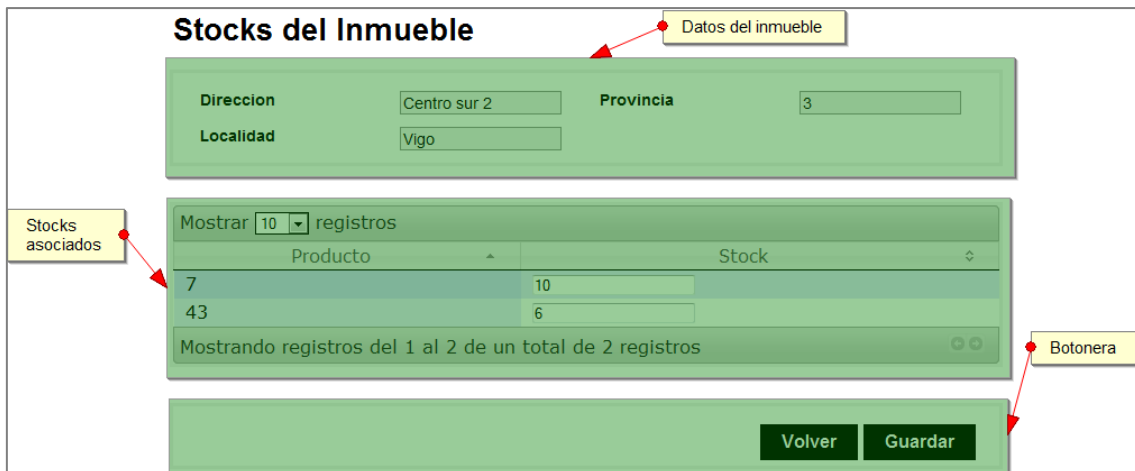
Pantalla de Modificación de Stocks de Inmuebles

Se trata de la pantalla a la que se accede pulsando sobre el identificador del inmueble en la pantalla de consulta de los stocks. A continuación se muestra la apariencia de dicha pantalla:



En cuanto a la pantalla, lo primero que podemos observar, es que, como se había comentado sigue la estructura general definida al iniciar el manual, por lo que nos centraremos en la

región central, que es donde se ubican las funcionalidades específicas de la pantalla. En la siguiente imagen se puede ver un desglose de las zonas que la componen:



- En la zona superior se mostrarán los datos del inmueble cuyos stocks se están mostrando.
- Se muestra un listado con todos los productos dados de alta en el sistema, asociados a sus stocks en el inmueble con el que se está trabajando.
- En la botonera tendremos a nuestra disposición la opción de volver a la pantalla de listado de los distintos elementos sin guardar ningún cambio mediante el botón etiquetado como “Volver”, o actualizar los distintos stocks del inmueble pulsando sobre el botón etiquetado como “Guardar”.

Pantalla de Modificación de Stocks de Productos

Se trata de la pantalla a la que se accede pulsando sobre el identificador del producto en la pantalla de consulta de los stocks. A continuación se muestra la apariencia de dicha pantalla:



En cuanto a la pantalla, lo primero que podemos observar, es que, como se había comentado sigue la estructura general definida al iniciar el manual, por lo que nos centraremos en la región central, que es donde se ubican las funcionalidades específicas de la pantalla. En la siguiente imagen se puede ver un desglose de las zonas que la componen:



- En la zona superior se mostrarán los datos del producto cuyos stocks se están mostrando.
- Se muestra un listado con todos los inmuebles dados de alta en el sistema, asociados a sus stocks del producto con el que se está trabajando.
- En la botonera tendremos a nuestra disposición la opción de volver a la pantalla de listado de los distintos elementos sin guardar ningún cambio mediante el botón etiquetado como "Volver", o actualizar los distintos stocks del producto pulsando sobre el botón etiquetado como "Guardar".

Conclusiones

Al comenzar este proyecto, uno de los objetivos del mismo era el familiarizarme con el proceso de desarrollo de software. Si bien hasta el día de hoy, no pocas veces se me ha presentado de forma teórica todo el ciclo de desarrollo de un sistema informático, ha sido con este proyecto cuando he experimentado por primera vez lo que es llevarlo a cabo.

Creo que, a pesar de mi inexperiencia tanto en prácticamente la totalidad de las fases del proceso de desarrollo, como en el uso de patrones para mejorar el diseño de la aplicación resultante, el presente proyecto muestra que he conseguido llevar a cabo con éxito su uso. Aunque he de reconocer, que si hasta ahora únicamente en la teoría ciertas prácticas ayudaban a llevar a cabo el desarrollo de una aplicación, a lo largo del proyecto se han mostrado ante mí como unas herramientas indispensables para el proceso de creación de una aplicación como la generada en este proyecto, y es que me es difícil considerar que hubiese sido capaz de lograr terminar en buen puerto este proyecto sin tener a mano los patrones utilizados en el mismo.

Por otro lado, en cuanto a la aplicación en sí misma. Pienso que los objetivos marcados inicialmente para el mismo han sido finalmente cubiertos. Como se señalaba al comenzar este documento, se buscaba principalmente desarrollar lo que pudiese tratarse de una base sobre la que empezar a cimentar una herramienta que permitiese al pequeño y mediano comercio el iniciar su andadura en el e-business, y si bien es cierto que la parte desarrollada se centra en la gestión de acceso a la propia plataforma, así como al sistema de inventariado, no es menos cierto que como aplicación está desarrollada para que sea fácilmente ampliable a tener una sección pública que permite la venta de los productos dados de alta.

En conclusión, el presente proyecto ha servido tanto como para instruirme en el uso de técnicas, procesos y herramientas que hasta el día de hoy no había utilizado, como para generar una aplicación conforme a los requisitos iniciales que se habían especificado.

Bibliografía

Algunos de los utilizados a lo largo de este proceso de desarrollo han sido:

- Jacobson, I., Booch, G., Rumbaugh J., El Proceso Unificado de Desarrollo de Software, 2000 Addison Wesley
- Kruchten, P., The Rational Unified Process: An Introduction, 2000 Addison Wesley
- Kruchten, P. Architectural Blueprints—The “4+1” View Model of Software Architecture. IEEE Software 12 (6), November 1995, pp. 42-50.
- E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994
- F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal, Pattern-Oriented Software Architecture: A System of Patterns, Wiley & Sons, 1996
- J. W. Cooper, Java Design Patterns: A Tutorial, Addison-Wesley, 2000
- M. Grand, Patterns in Java: Catalogue of Reusable Design Patterns Illustrated with UML - Vol 1, Wiley & Sons, 2002
- J. Crupi, D. Alur, D. Malks, Core J2EE Patterns, 2nd edition, Prentice Hall, 2003
- F. Marinescu, EJB Design Patterns, John Wiley & Sons, 2002
- Tomas Van de Velde, Bruce Snyder, Christian Dupuis, Sing Li, Anne Horton, Naceen Balani, Spring Framework 2, Wiley Publishing, 2008
- Christian Bauer, Gavin King, Java Persistence with Hibernate, Manning, 2007