

INGENIERIA DE SOFTWARE

Trabajo Final de Carrera

ANÁLISIS Y DISEÑO DE UN PORTAL DE VENTA DE LIBROS EDUCATIVOS



Jordi Cid Rodríguez - ETIG -
Consultor: José Antonio Raya Martos
Septiembre 2011



Objetivo

El objetivo de este proyecto ha consistido en realizar el análisis y diseño de una aplicación **Web B2C**, de forma que a partir de dicha documentación, se puede iniciar las fases de implementación y verificación, antes de su puesta en producción.

Cabe señalar que la documentación se ha realizado basándome en el lenguaje de modelado **UML** (Unified Modeling Language), bajo el paradigma **OO** (Object-Oriented) y utilizando **UP** (Unified Process) como marco de desarrollo, si bien, en este último aspecto y dadas las características de secuencialidad del proyecto no se ha podido aplicar de manera estricta y en particular el aspecto iterativo e incremental.

Otras tres características que se han tenido en cuenta en el proyecto son:

Por un lado la aplicación de patrones de diseño y en particular los relacionados con la asignación de responsabilidades **GRASP** (General Responsibility Assignment Software Patterns).

Por otro e inspirándome en los principios de **MDA** (Model-Driven Architecture) se ha procurado separar los requerimientos funcionales y de diseño de la infraestructura o plataforma tecnológica, de forma que dicho análisis y diseño pueda ser reutilizable, es por ello, que en este trabajo se ha evitado hacer alusión a aspectos relacionados con una tecnología de implementación específica.

Por último y como consecuencia de considerar como marco de desarrollo UP, se ha procurado que sean los casos de uso el aspecto central y mas relevante sobre el cual se despliega todo el resto de artefactos de análisis y diseño utilizados en este trabajo. De forma que la ingeniería de requisitos y en particular los casos de uso se convierte en el eje vertebrador de dicho trabajo.



Definición del proyecto

La empresa Agora Junior S.A., está especializada en la venta de libros infantiles y juveniles. Con el objetivo de incrementar y globalizar el negocio se han planteado como primer paso abrir un nuevo canal de venta por Internet.

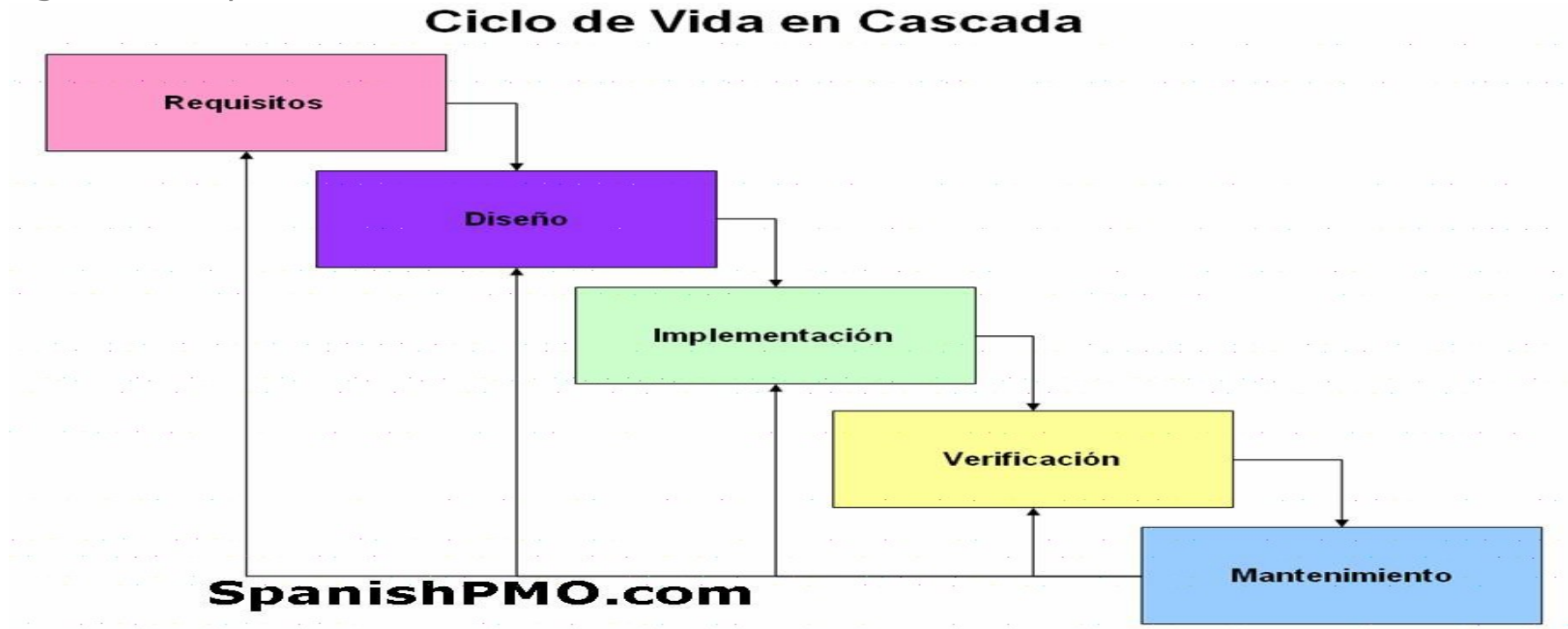
Consideran que este nuevo canal puede ser a largo plazo una buena plataforma que le permitirá darse a conocer y extender sus ventas a nivel mas global. Para ello y en el corto plazo, ha decidido abrir un sitio web básico para que se pueda consultar su catálogo de libros y venta de éstos. Tambien y con el objetivo de fidelizar a los clientes e incentivar las compras por Internet, se quiere definir y poner en marcha campañas que premiarán al cliente en función del volumen de compras realizadas por éste.

Como se ha comentado anteriormente se trata de crear un sitio web básico pero a su vez en el medio y largo plazo con capacidad de crecimiento y ampliación de nuevas funcionalidades y estos son aspectos que se han de considerar tanto en el análisis como en el diseño de la aplicación.



Ciclo de vida del producto

Todo y que en muchos aspectos UP ha sido el marco de referencia a la hora de realizar este trabajo. Debido a la secuencialidad para la finalización de los diferentes apartados del proyecto, se ha utilizado la metodología de producción mas común y conocida como **SDLC** (Systems development life cycle) que representa las fases secuenciales por las que pasa un sistema de información a lo largo de su vida útil. Este ciclo se compone de las siguientes etapas:

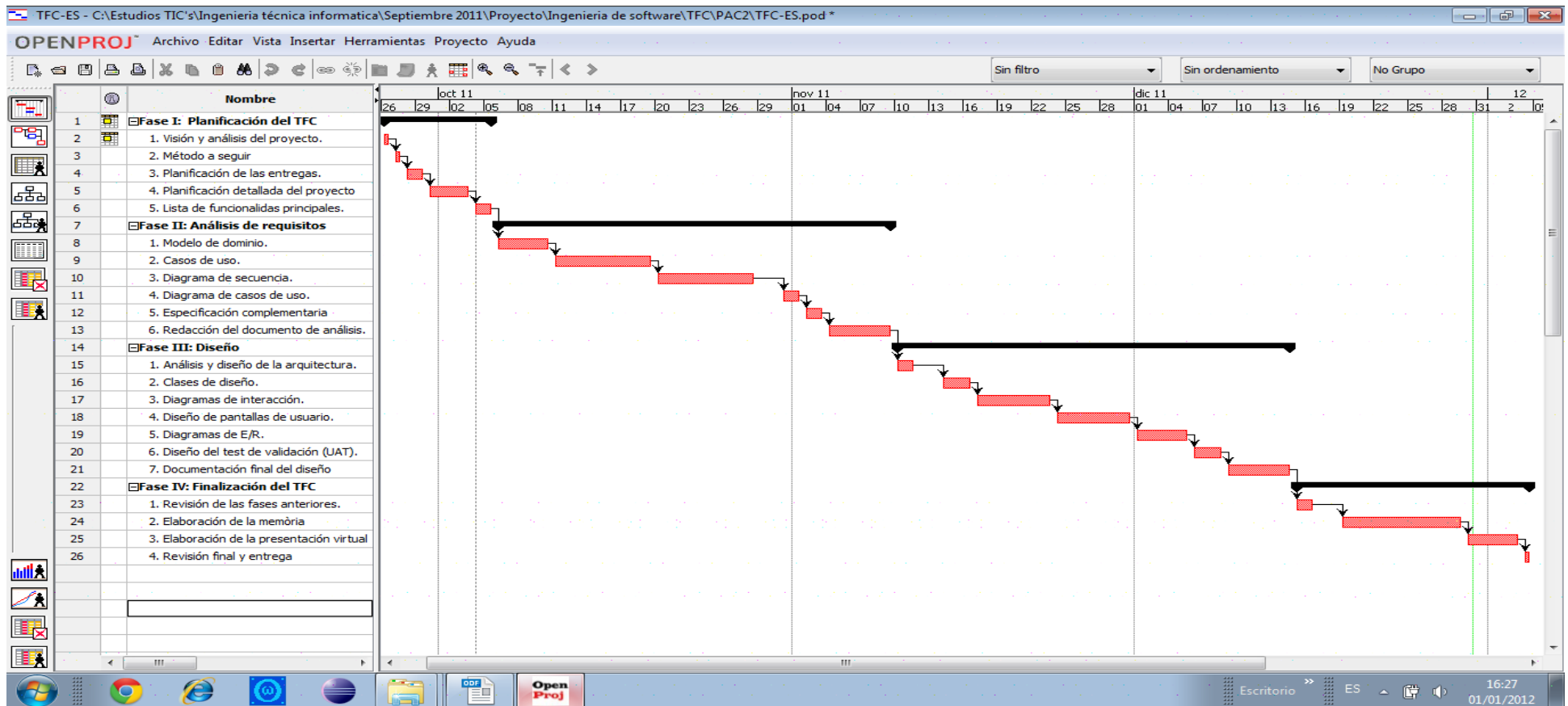


en nuestro caso la etapa de implementación, verificación y mantenimiento no se tendrán en cuenta, centrándonos exclusivamente en el análisis de requisitos y diseño.



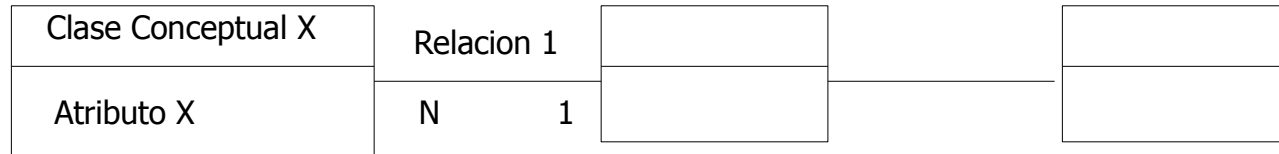
Planificación

La planificación detallada de las tareas del proyecto y los objetivos de cada una de ellas, quedó de la siguiente manera:

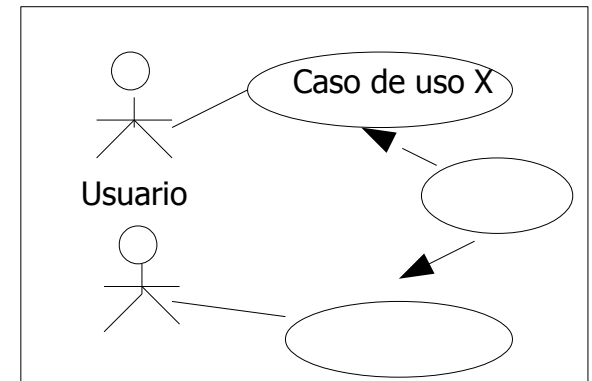
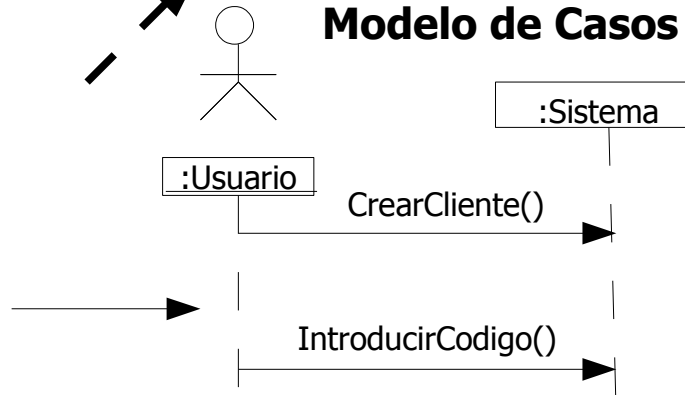
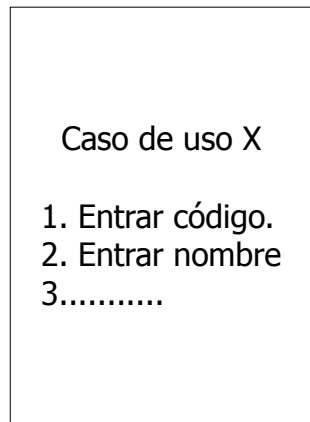


Relación entre artefactos UML del UP

Modelo del Dominio



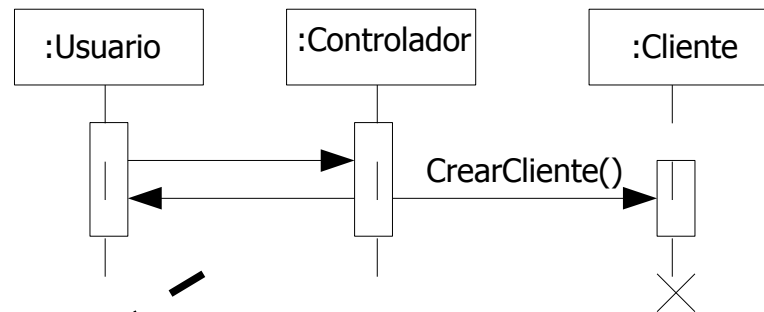
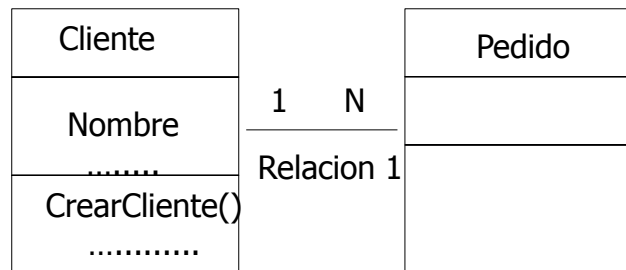
Modelo de Casos de Uso



Diagramas de Secuencia del Sistema

Diagramas de Casos de Uso

Modelo de Diseño



Diagramas de Clases

Diagramas de Secuencia



Análisis: Actores-Casos de Uso

Dentro del análisis el primer paso a consistido en identificar los casos de uso y sus actores. Como resultado de ello se ha obtenido la siguiente relación.

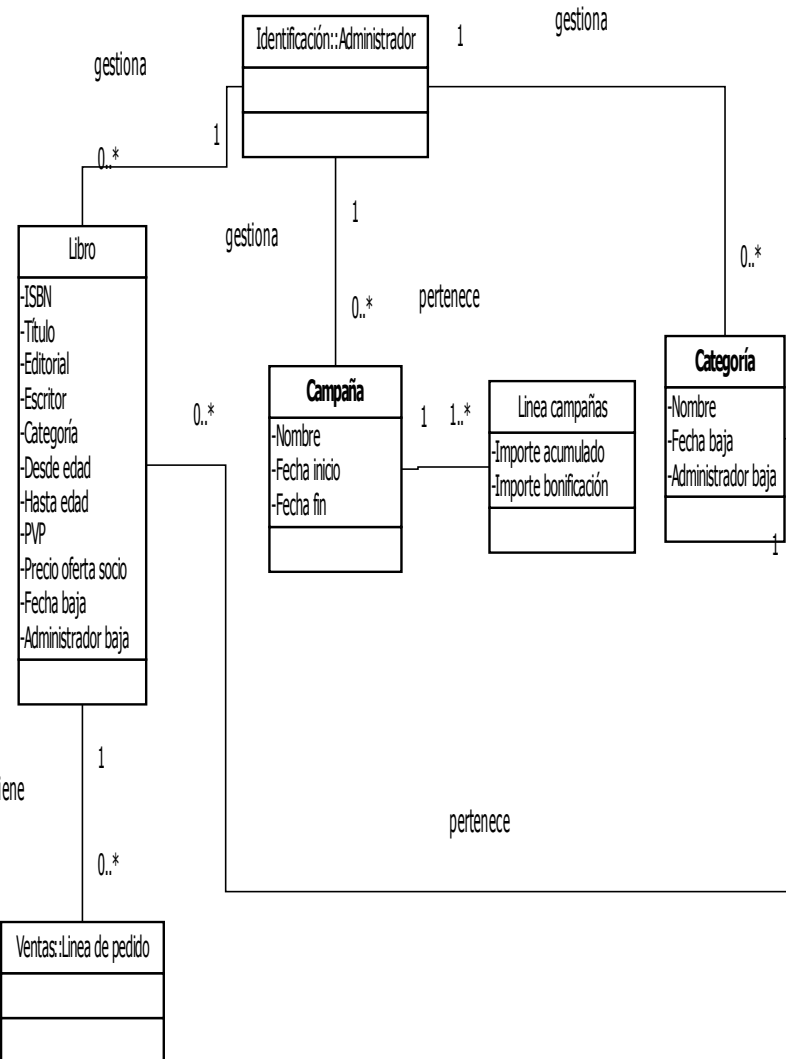
N.CU	Actor-Caso de uso		
	Actor	Funcionalidad	Descripción
CU01	Administrador o socio	Identificación	Permite identificarse.
CU02	Administrador	Alta de categorías	<u>Permite</u> dar de alta en el sistema las categorías.
CU03	Administrador	Consulta de categorías.	<u>Permitir</u> consultar las categorías.
CU04	Administrador	Mantenimiento de categorías	<u>Permitir</u> modificar o dar de baja una categoría.
CU05	Administrador	Alta de libros	<u>Permitir</u> dar de alta la información sobre libros.
CU06	Administrador	Mantenimiento de libros.	<u>Permitir</u> modificar dar de baja un libro
CU07	Administrador	Alta de campañas	<u>Permitir</u> dar de alta las campañas de marketing.
CU08	Administrador	Consulta campañas	<u>Permitir</u> consultar las campañas
CU09	Administrador	Mantenimiento de campañas.	<u>Permitir</u> modificar o dar de baja una campaña.
CU10	Usuario Socio	Criterios de consulta de libros.	<u>Permitir</u> indicar los criterios de consulta de libros.
CU11	Usuario Socio	Resultado consulta libros.	<u>Permitir</u> visualizar los libros según criterios previamente indicados en el caso de uso <u>Criterios consulta de libros</u> .
CU12	Usuario Socio	Gestión del pedido en curso.	<u>Permitir</u> visualizar pedido en curso, eliminar libros del pedido y/o formalizar el pedido en curso.
CU13	Usuario Socio	Datos de <u>envío</u>	Solicita o muestra en el caso de socio registrado los datos para el <u>envío</u> del pedido. <u>También</u> permite a un usuario registrarse como socio.
CU14	Administrador	Consulta pedidos formalizados.	<u>Permitir</u> los pedidos formalizados por un usuario o socio y que están pendientes de facturar.
CU15	Administrador	Facturación pedido	<u>Permitir</u> facturar el pedido para la entrega definitiva del material al cliente.



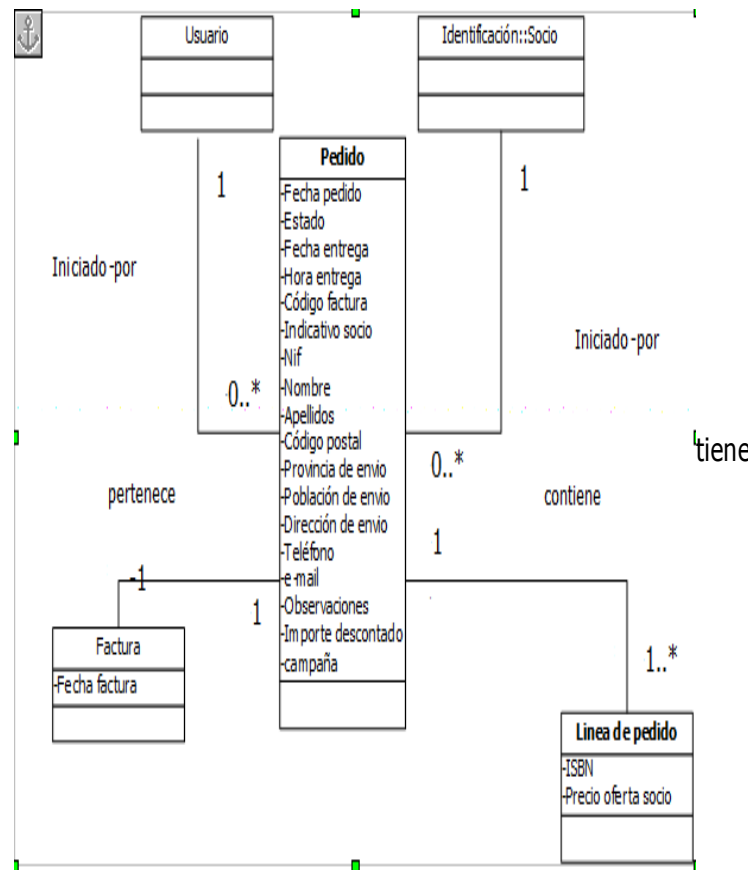
Análisis: Modelo de Dominio

En nuestro sistema se ha identificado y estructurado el modelo de dominio en los siguientes paquetes/módulos:

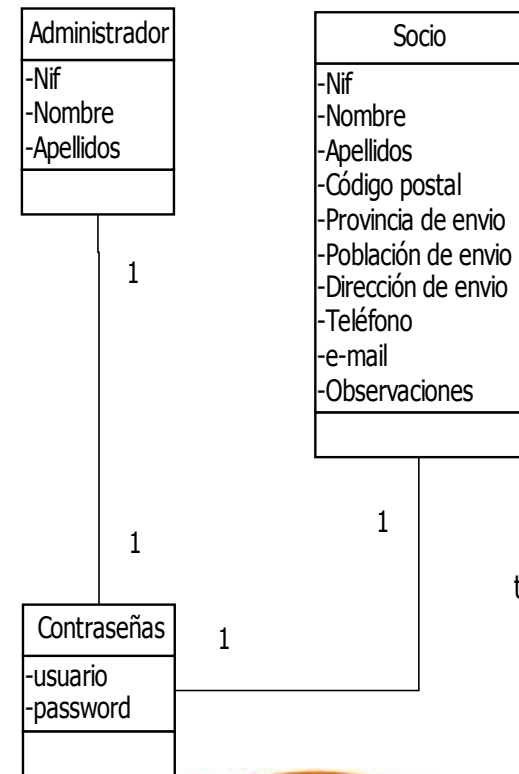
Paquete Productos



Paquete Ventas



Paquete Identificación



Análisis: Casos de Uso

En el proyecto se han identificado los 15 casos de uso comentados anteriormente. Por razones de espacio se muestran 2 de ellos:

ID	CU02	
Identificador del Caso de Uso	Alta de categoría.	
Objetivo del Caso de Uso	Permite que el administrador dé de alta en el sistema las categorías.	
Caso de uso relacionado		
Actores	Administrador.	
Frecuencia	Díaria.	
Condiciones Previas	El administrador se ha identificado en el sistema mediante un nombre de usuario y una <u>password</u> .	
Condiciones Posteriores	El sistema a registrado correctamente la información correspondiente a la categoría (código y nombre).	
Acción o evento que arranca Caso de Uso	Nuevo formulario <u>Alta de categorías</u> donde el administrador introducirá los datos para dar de alta una categoría.	
Paso	Descripción	
1	Acceder al formulario de Alta de categoría.	
2	Introduce el código de la categoría.	
3	Introduce el nombre de la categoría.	
4	El administrador solicita al sistema que registre los datos.	
5	El sistema registra los datos proporcionados e informa al administrador que el proceso ha finalizado con éxito.	
Excepciones		
Número	Descripción	
4	Si el sistema detecta que la categoría ya está registrada, el sistema informa de la situación al administrador <u>permitiéndole</u> modificar los datos proporcionados y el caso de uso continúa.	
4	Si el sistema detecta que falta alguno de los datos solicitados, informa de la situación al administrador <u>permitiéndole</u> modificar los datos proporcionados y el caso de uso continúa.	
4	Si el administrador solicita cancelar la operación, el sistema cancela la operación, y a continuación este caso de uso se cancela.	

ID	CU03	
Identificador del CU	Consulta de categorías.	
Objetivo del CU	<u>Permitir</u> que el administrador consulte las categorías vigentes.	
Caso de uso relacionado	CU04: Mantenimiento de categorías.	
Actores	Administrador	
Frecuencia	Díaria	
Condiciones Previas	El administrador se ha identificado en el sistema mediante un nombre de usuario y una <u>password</u> .	
Condiciones Posteriores	El sistema muestra correctamente la información correspondiente a las categorías vigentes, es decir, sin fecha de baja.	
Acción que arranca el CU	Formulario <u>Consulta de categorías</u> donde el administrador podrá visualizar los datos de las categorías vigentes.	
Paso	Descripción	
1	Acceder a la nueva pantalla Consulta de categorías.	
2	El sistema muestra la información (código y nombre) de las categorías vigentes.	
3	El sistema es para respuesta del usuario.	
Excepciones		
Número	Descripción	
3	Si el usuario solicita cancelar la operación, el sistema cancela la operación, y a continuación este caso de uso se cancela.	
Flujo alternativo		
Paso	Descripción	
3	Si el usuario selecciona una categoría accederá al caso de uso <u>Mantenimiento de categorías</u> . Al volver se refrescará en el formulario la información de las categorías vigentes y el caso de uso continúa.	



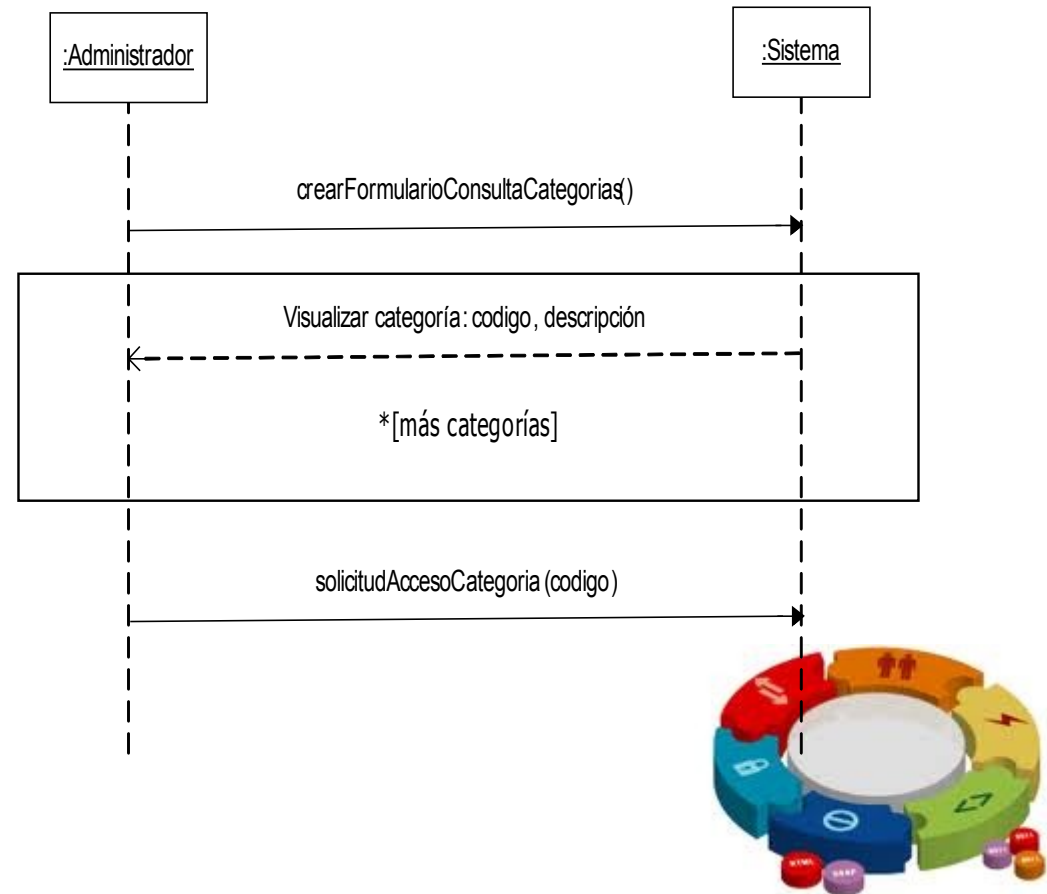
Análisis: Diagrama de secuencia

En esta fase los diagramas de secuencia del sistema han de mostrar los eventos del sistema de entrada y salida relacionados con los casos de uso. Es decir, se trata de representar los eventos que parten de los actores externos hacia el sistema. Es pues una visión de caja negra sobre el sistema, describiendo **que** hace el sistema, sin explicar **como** lo hace. Por cada caso de uso tenemos un diagrama de secuencia, que al igual que en el apartado anterior, por razones de espacio solo se representarán los diagramas correspondientes a los 2 anteriores casos de uso.

CU02 Alta de categorías

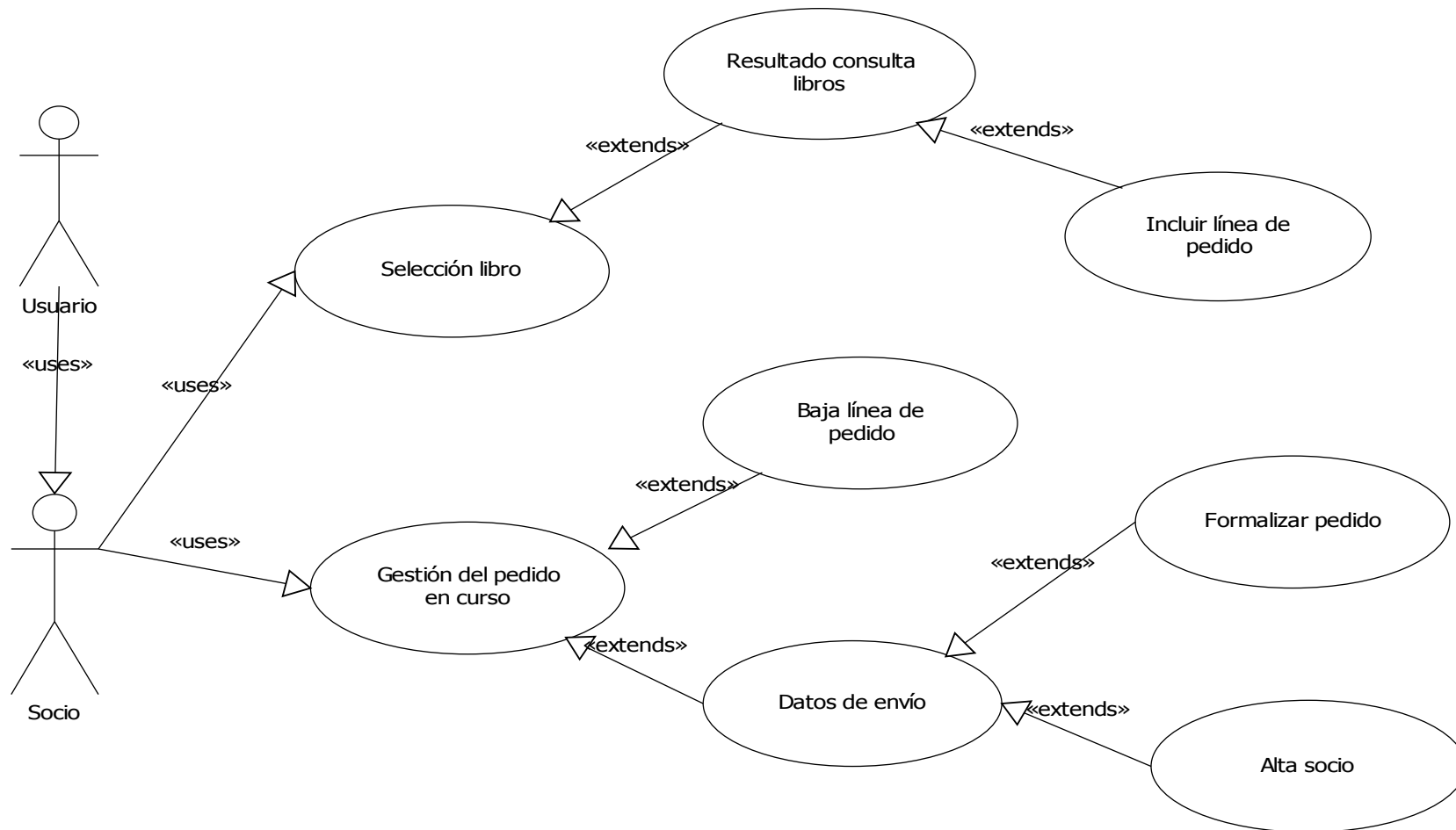


CU03 Consulta Categorías



Análisis: Diagrama de casos de uso

Dentro de UML los diagramas de casos de uso nos permite ilustrar de manera gráfica los nombres de los casos de uso y los actores , así como las relaciones entre ellos. En nuestro proyecto estos serían una parte de los diagramas de casos de uso identificados:



Diseño: Arquitectura y Patrones I

El primer libro que se dedicó al tema de los patrones de arquitectura fue POSA (Pattern-Oriented Software Architecture). En este libro se recogen las buenas prácticas de diseño arquitectural y de forma especial sobre la arquitectura lógica a gran escala. En POSA se da una clasificación de los patrones a diferentes niveles:

Patrones de arquitectura: Relacionados con el diseño a gran escala y de grano grueso.

Patrones de diseño: Relacionados con el diseño de los objetos y frameworks de pequeña y mediana escala. Aplicables al diseño de una solución para conectar los elementos de gran escala que se definen mediante los patrones de arquitectura y durante el trabajo de diseño detallado para cualquier aspecto de diseño local.

En nuestro caso aplicaremos el **patrón de arquitectura Capas** (Layers):

Las ideas del patrón Capas son:

- a) Organizar la estructura lógica de gran escala de un sistema en capas separadas de responsabilidades distintas y relacionadas, con una separación clara entre las capas mas bajas, es decir, los servicios de bajo nivel y las capas mas altas, las que son mas específicas de la aplicación.
- b) La colaboración y acoplamiento desde las capas mas altas hacia las mas bajas, de forma que se evite el acoplamiento entre ambas capas.

El patrón Capas se relaciona con la arquitectura lógica, es decir, describe la organización conceptual de los elementos del diseño en grupos.



Diseño: Arquitectura y Patrones II

En UML los diagramas de paquetes se utilizan para representar las capas. En UML una capa es un paquete. Las Capas definen un modelo general de N-niveles para la arquitectura lógica, en nuestro se ha seleccionado 3 de los mas comunes. Estos son:

Capa de presentación: lo compondrán tanto el aspecto externo de nuestros sistemas (interficies de usuario), como el interno que incluirá las clases que implementarán las pantallas y tambien las interacciones de estas clases con el resto del sistema. Para ello, dentro de esta capa aplicaremos el patrón de arquitectura modelo, vista, controlador.

Cada clase control representa un caso de uso. Recibe mensajes de las clases frontera (nunca de los actores directamente).

Capa de dominio: se puede decir que es la parte esencial del diseño, dado que dentro de esta se gestiona las peticiones de la capa de presentación, los flujos de trabajo, estado de la sesión, transiciones a ventanas/páginas y concentración /transformación de diferentes datos para la presentación.

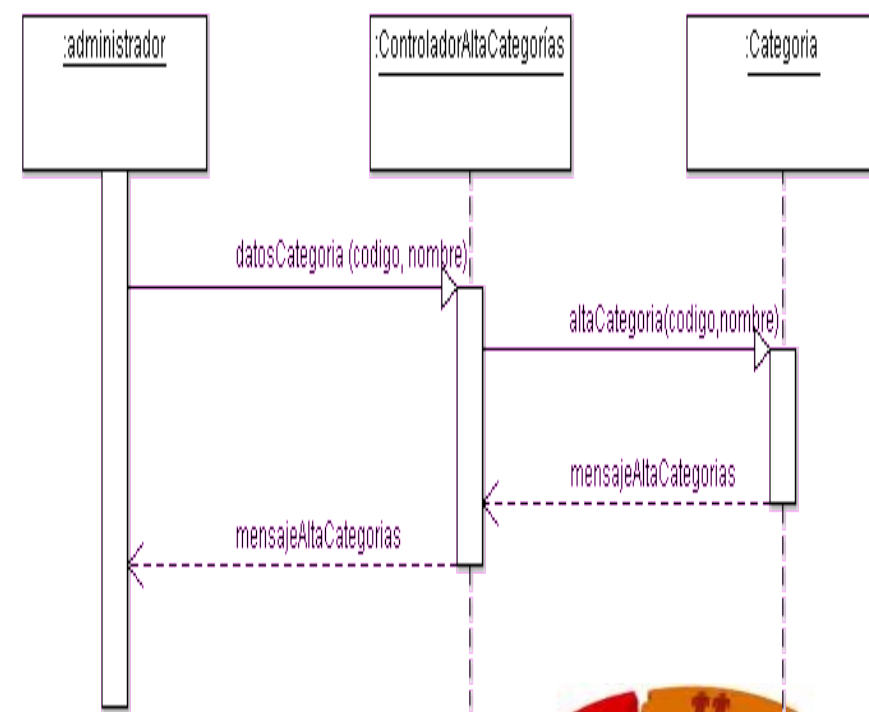
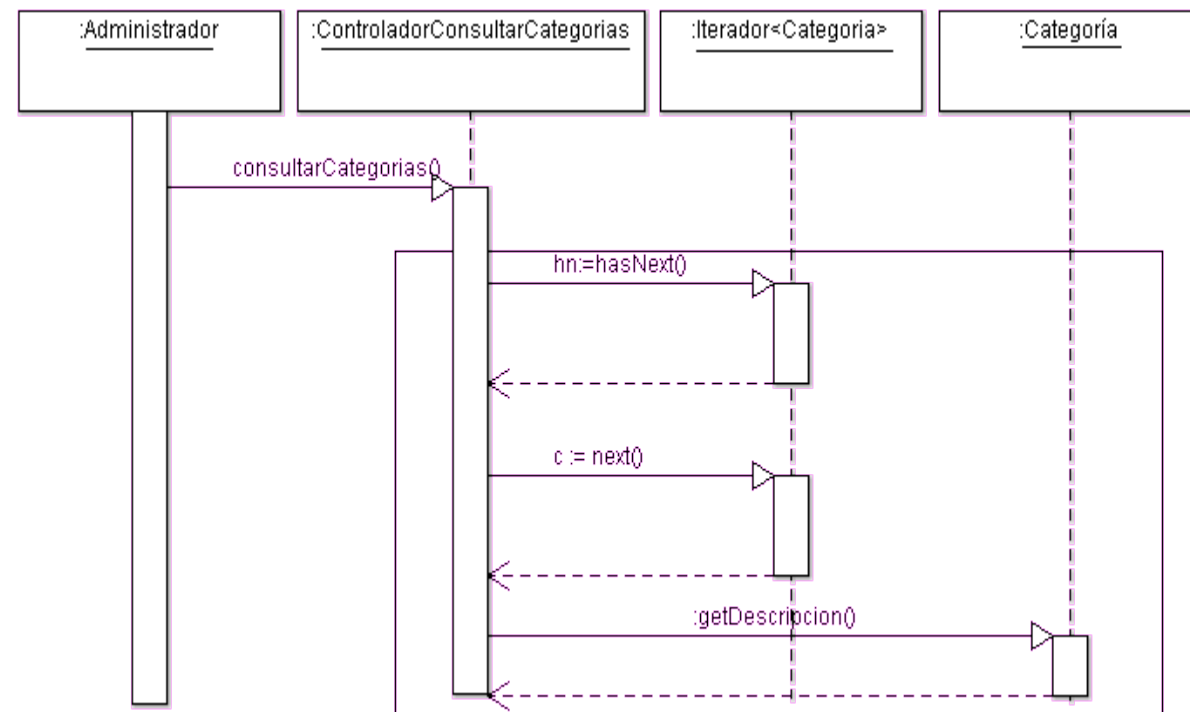
Capa de servicios técnicos: la forman los servicios técnicos de alto nivel y frameworks y persistencia. Esta capa es la que pone en conexión y relación las clases conceptuales de la capa de dominio con las entidades que forman la persistencia (base de datos).

Por último y antes de entrar en los diagramas de interacción, he intentado basarme en los principios de diseño GRASP. GRASP es el acrónimo de General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades), y lo componen varios patrones, en el proyecto sobre todo he tenido en cuenta los siguientes cinco patrones: **Experto en información, Creador, Bajo Acoplamiento, Alta Cohesión y Controlador.**



Diseño: Diagramas de secuencia

Dentro de los diagramas de interacción tenemos los de secuencia o colaboración, su objetivo es representar la interacción de mensajes. Cada uno de ellos tienen puntos fuertes y débiles, el hecho de decantarnos por el diagrama de secuencia es porque a mi entender reflejan de forma sintética mejor el envío de mensajes entre objetos. Por cada diagrama de secuencia de la fase de análisis, tenemos un diagrama de secuencia en la fase de diseño, en la fase de diseño los diagramas de secuencia, explican en el que sino el como y además en el diseño de estos se ha tenido en cuenta los principios de patrones anteriormente comentados. Al igual que en apartados anteriores, por razones de espacio solo se representarán los diagramas correspondientes a los 2 diagramas de Secuencia, seleccionados en el apartado de análisis.



Diseño: Diagramas de clases I

Una vez finalizados los diagramas de interacción partiendo de los casos de uso, y del Modelo de Dominio ya es posible identificar la especificación de las clases de software (e interfaces) que participan en la solución software y añadirles detalles de diseño, como los métodos. UML proporciona la notación para representar los detalles de diseño en los diagramas de clases de diseño o **DCD**. A diferencia de las clases conceptuales del Modelo del Dominio, las clases de diseño de los DCD muestran las definiciones de las clases software en lugar de los conceptos del mundo real. En el trabajo se han identificado por un lado los controladores de dominio y por otro las clases. Como controladores y por razones de espacio solo presentamos una parte de ellos.

ControladorIdentificacion
datosIdentificacion(usuario : String,contraseña : String) cancelar()

ControladorAltaCategorias
datosCategoria(codigo : Integer,nombre : String) cancelar()

ControladorConsultarCategorias
consultarCategorias() : Lista<Categoria> cancelar()

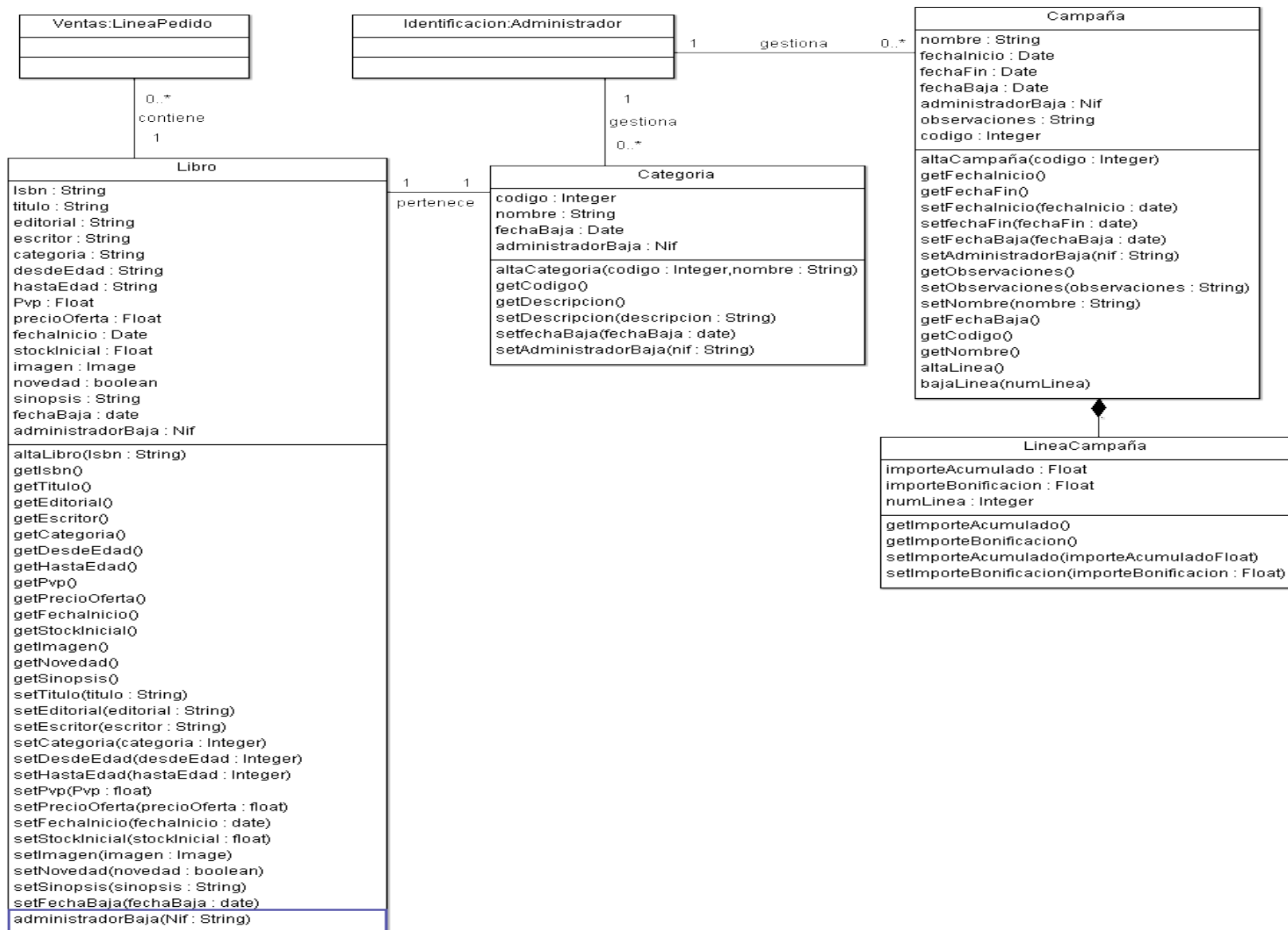
ControladorMantenimientoCategoria
consultarCategoria(codigo : Integer) : Categoria modificarcategoria(descripcion : String) bajaCategoria(codigo : Integer) cancelar()

Diccionario<T>
listar() : Iterador<T> consultar(id : String) : T registrar(id : String,objete : T)



Diseño: Diagramas de clases II

En cuanto a los diagramas de clases, dentro del proyecto se han agrupado/estructurado atendiendo a la estructura por paquetes indicadas anteriormente. Por razón de espacio, aquí se refleja solo la estructura de clases correspondiente al paquete de productos.



Diseño: Servicios Técnicos

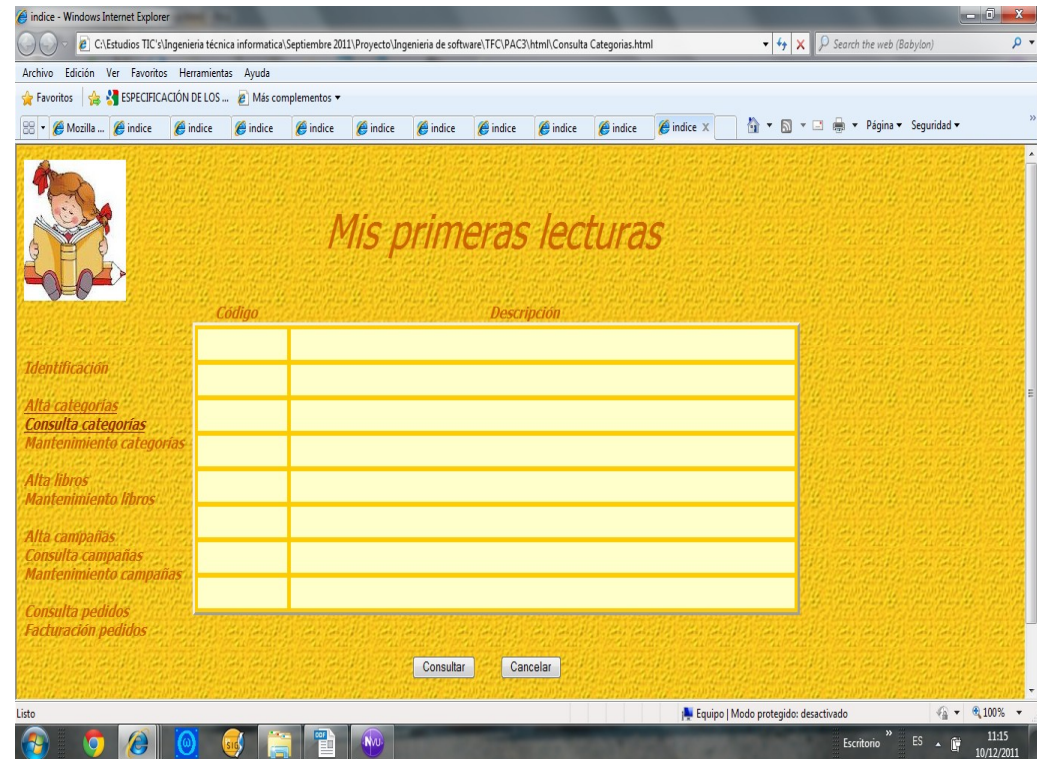
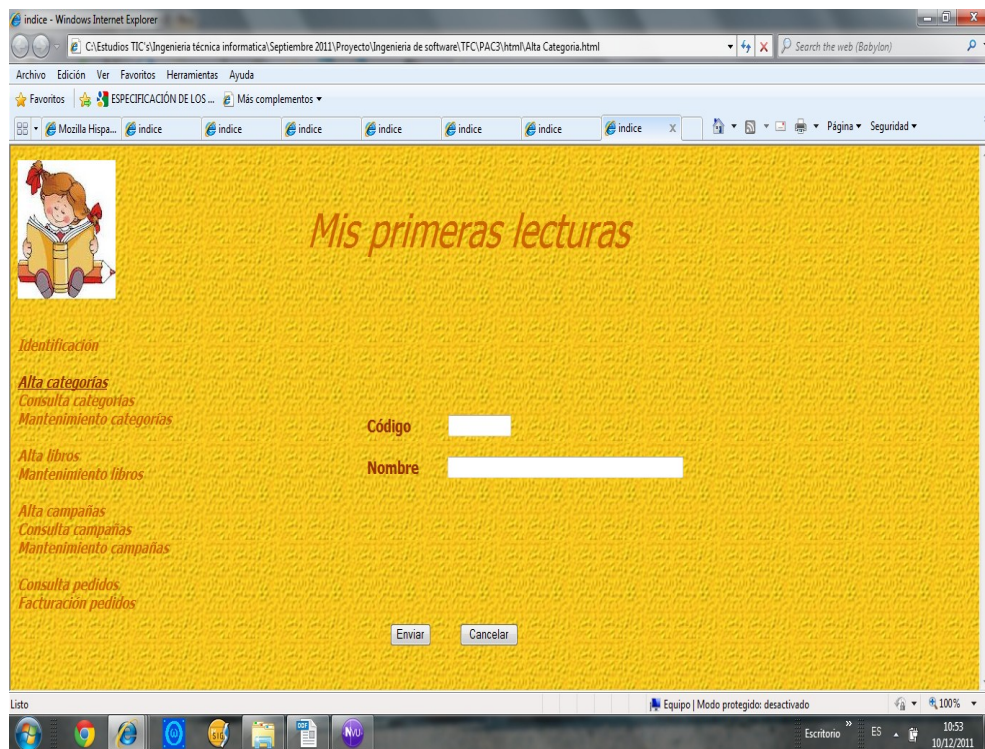
La aplicación requiere que se almacene y recuperen la información en mecanismos de almacenamiento persistente, como una base de datos relacional, y este es en nuestro caso el objetivo básico del nivel de servicios técnicos. en este estudio tan solo se reflejará la interface del dominio de forma que queden indicados cuales han de ser los servicios que tendría o tendrá que implementar el framework que se seleccione en su momento. Teniendo en cuenta esto, concentraremos todos los servicios necesarios en una única clase que denominaremos BDLibreria.

<div><<interface>></div> <div>BDLibreria</div>
<div><div><div>consultarCategorias() : Iterador<Categoria></div><div>consultarCategoria(codigo : Integer) : Categoria</div><div>almacenarCategoria(datos) : Categoria</div><div>eliminarCategoria(codigo : Integer) : Categoria</div><div>consultarLibro(isbn : String) : Libro</div><div>almacenarLibro(datos) : Libro</div><div>eliminarLibro(isbn : String) : Libro</div><div>consultarCampañas() : Iterador<Campañas></div><div>consultarLineasCamapañas(numLinea : Integer) : <LineasCampaña></div><div>almacenarCampañas(codigo : String) : Campaña</div><div>almacenarLineasCamapaña(datosLinea) : LineasCampaña</div><div>consultarCampaña(codigo : String) : Campaña</div><div>eliminarCampaña(codigo : String) : Campaña</div><div>consultarLibros(criterios) : Iterado<Libros></div><div>consultarLibro(Isbn : String) : Libro</div><div>almacenarPedido(datos) : Pedido</div><div>almacenarSocio(datos) : Socio</div><div>almacenarContraseña(usuario : String,password : String) : Contraseña</div><div>consultarPedidos() : Iterador<Pedidos></div><div>consultarPedido(id : Number) : Pedido</div><div>consultarLineasPedidos(Id : Integer) : LineasPedidos</div><div>facturarPedido(datos) : Pedido</div><div>facturaPedido(datos) : Factura</div><div>eliminarLibro(Isbn : String) : Libro</div></div></div>



Diseño: Interficie gráfica (UI)

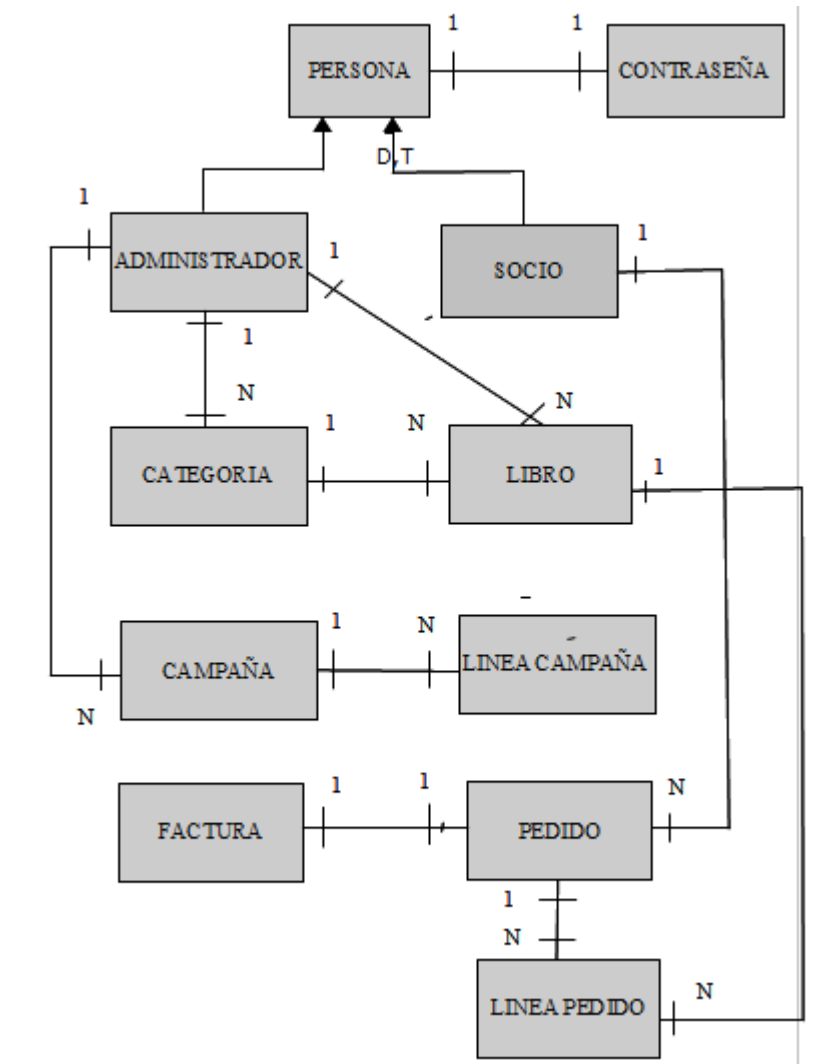
Para hacer el análisis de la interficie gráfica del usuario nos basamos en los diagramas de secuencia simplificados de los casos de uso. Cabe señalar que estamos hablando de un primer diseño y comportamiento de las interfaces de usuario y que sería la idea base para una posterior revisión detallada por un experto en usabilidad. Al igual que en apartados anteriores, por razones de espacio solo se representarán los diagramas correspondientes a los 2 diagramas de secuencia, seleccionados en el apartado de análisis.



Diseño: Diagrama de E/R

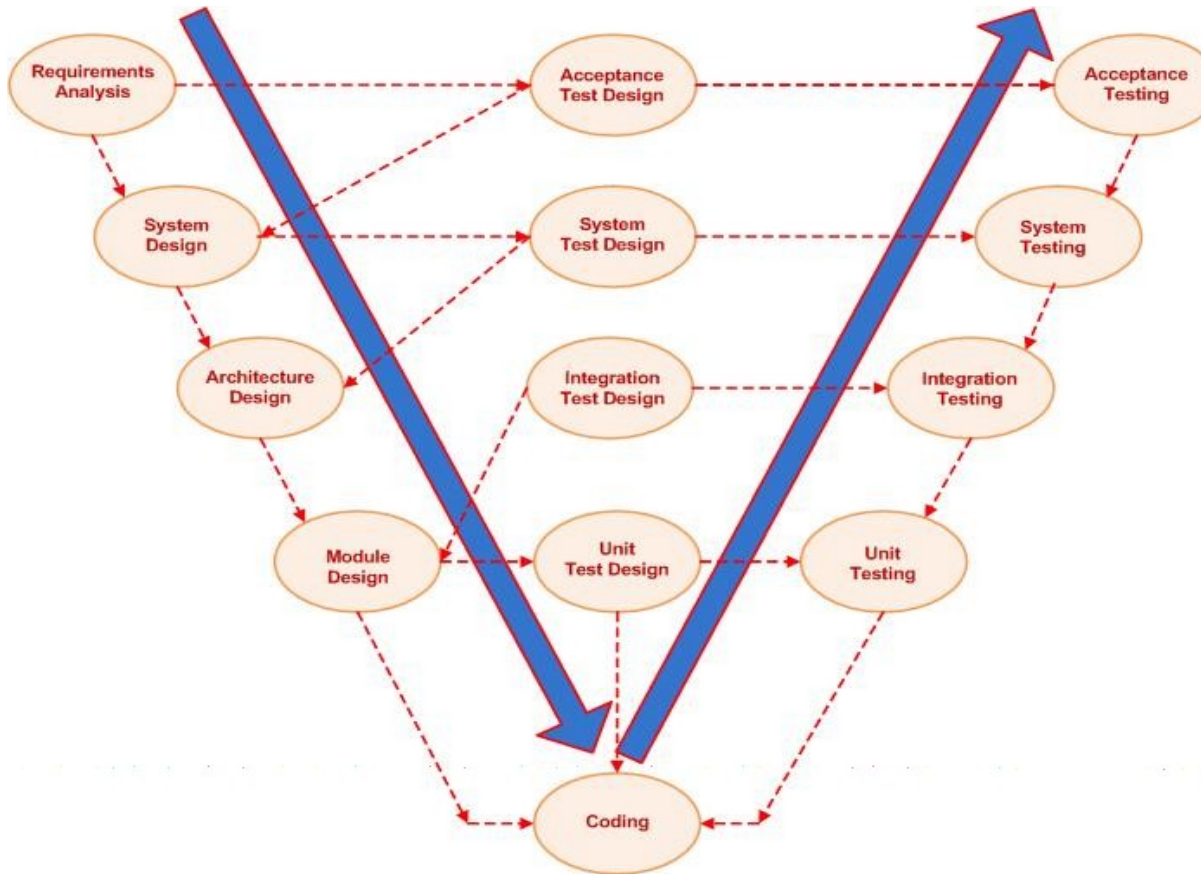
En cualquier negocio e-commerce, en nuestro caso estaríamos hablando de un B2C, se necesita, presentar un catálogo de nuestros productos, guardar la información mas relevante de nuestros clientes/socios, etc.. y además realizar un mantenimiento continuo de esta información, para poder visualizarla en la web a nuestros usuarios. Es por ello, que se requiere de un sistema de persistencia de datos y para ello necesitamos el uso de una base de datos.

En este apartado, lo que reflejamos es la estructura y necesidad de información que se requiere en la base de datos para nuestro negocio.



Diseño: Test I

Dentro de la ingeniería de software testear es una actividad orientada a reducir los riesgos y mejorar la calidad del software a partir de la localización de fallos. Para poder realizar la tarea de testeo, es conveniente que comience desde los primeros momentos, es decir, en los requerimientos. Para ello se ha de considerar que éstos han de cumplir una condiciones básicas para que sean testeables. Estas son: los requerimientos del proyecto han de estar indicados de tal forma que puedan ser testeables, los requerimientos han de ser precisos y detallados para evitar requerimientos ambiguos, garantizar que no hay ningún requerimiento implícito o inexistente y los requerimientos han de ser completos, es decir, recoger todas las condiciones y comportamiento que el usuario espera que tenga el sistema. Teniendo en cuenta lo anterior tambien se puede ver el testing como un proceso que está relacionado con el proceso de desarrollo.



Diseño: Test II

A parte de tener testing en las diferentes fases del proyecto, tambien tenemos diferentes tipos de técnicas: testing de caja negra (black-box testing), testing de la caja blanca (white-box testing). El testing se realiza bajo un escenario concreto, y sobre todo bajo la perspectiva del usuario, es decir, diseñar pruebas en función de lo que hace (necesita) el usuario y no en base a lo que hace el software. Para obtener estas pruebas, nos podemos basar en los casos de uso que ya disponemos de las fases anteriores del proyecto.

Teniendo en cuenta todo lo anterior en nuestros casos hemos seleccionado definir el test en la fase previa a la implementación y dentro de los posibles test (unit test, integration test, stress test, regresion test, etc..) hemos seleccionado el UAT o test case de aceptación. Estamos hablando de un test de caja negra y el escenario lo determinarán los cada caso de uso, tanto el escenario principal como las excepciones. Al igual que en apartados anteriores, por razones de espacio solo se representarán los diagramas correspondientes a los 2 diagramas de secuencia, seleccionados en el apartado de análisis.

Funcionalidad cubierta	Descripción	Paso	Descripción del paso	Resultados esperados
CU02 Alta Categorías	Permite que el administrador dé de alta en el sistema las categorías.. <u>Escenario principal.</u>	Step 1	Acceder a la nueva pantalla Alta de Categorías desde el <u>menu</u> principal del administrador.	Acceder a la pantalla sin problemas
		Step2	Introduce un código de categoría inexistente.	Se puede introducir el código sin problemas
		Step3	Introduce la descripción de la categoría	Se puede introducir una descripción sin problemas
		Step4	Se pulsa el botón Enviar	El sistema registra la información en el sistema e informa de que es correcto. Se <u>mantiene</u> en <u>formulario.</u>
		Step5	Se pulsa el botón Cancelar	Salimos del formulario y volvemos al menú principal
		Step6	Visualizar el <u>menu</u> del administrador	El menú que aparece es el del administrador
<u>Excepciones</u>		Step 1	Acceder a la nueva pantalla Alta de Categorías desde el <u>menu</u> principal.	Acceder a la pantalla sin problemas
		Step 2	Introduce el código de categoría que acaba de dar de alta	Se puede introducir el código sin problemas
		Step 3	Introduce una descripción de la categoría	Se puede introducir la descripción sin problemas
		Step 4	Se pulsa el botón Enviar	El sistema verifica la información e informa al usuario de que la categoría ya existe y se mantiene en el formulario.
		Step 5	Se introduce un nuevo código de categoría	Se puede introducir sin problemas.
		Step 6	Se deja el campo descripción de la categoría sin información	Se puede dejar sin información sin problemas.
		Step 7	Se pulsa el botón Enviar	El sistema verifica la información e informa al usuario de que la descripción es obligatoria. Se <u>mantiene</u> en el <u>formulario.</u>
		Step 8	Se pulsa el botón Cancelar	Salimos del formulario y volvemos al menú principal

Funcionalidad cubierta	Descripción	Paso	Descripción del paso	Resultados esperados
CU03 Consulta Categorías	Permitir que el administrador consulte las categorías vigentes. <u>Escenario principal.</u>	Step 1	Acceder a la nueva pantalla Consulta de Categorías desde el <u>menu</u> principal del administrador.	Acceder a la pantalla sin problemas
		Step2	Visualizar la información de pantalla	El sistema visualiza solo el <u>código</u> y la descripción de las categorías que no tienen fecha de baja.
		Step3	Se pulsa el botón Cancelar	El menú que aparece es el del administrador
<u>Acceso al mantenimiento de categorías</u>		Step 1	Acceder a la nueva pantalla Consulta de Categorías desde el <u>menu</u> principal del administrador.	Acceder a la pantalla sin problemas
		Step 2	Visualizar la información de pantalla	El sistema visualiza solo el <u>código</u> y la descripción de las categorías que no tienen fecha de baja.
		Step 3	El usuario hace doble clic sobre una línea de categoría	El sistema visualizar el caso de uso Mantenimiento Categorías y muestra la información sobre dicha categoría
		Step 4	El usuario <u>dá</u> de baja la Categoría desde el caso de uso Mantenimiento categorías.	El sistema avisa de que ha dado de baja la categoría y abandona el caso de uso Mantenimiento Categorías
		Step 5	Visualizar la información de pantalla	El <u>sistem</u> visualiza solo el <u>código</u> y la descripción de las categoría que no tienen fecha de baja y por tanto la categoría que se ha dado de baja anteriormente no tiene que visualizarse
		Step 6	Se pulsa el botón Cancelar	El menú que aparece es el del administrador



Conclusiones

El objetivo cuando inicie el trabajo era triple:

1. Por un lado poner en práctica los conocimientos sobre ingeniería de software y base de datos realizado durante la carrera.
2. Desarrollar el proyecto focalizando todo el análisis, diseño y test a partir de los casos de uso, es decir, dar relevancia a los aspectos de la ingeniería de requisitos de forma que se convirtiera en el hilo conductor de todo el trabajo.
3. Introducirme en temas de la ingeniería de patrones, así como intentar realizar el trabajo bajo la perspectiva de MDA, que sin aplicar la metodología propiamente dicha, si tener en perspectiva el poder realizarlo evitando aludir a cualquier plataforma tecnológica Específica.

Si los tres objetivos eran importantes, para mi el último ha resultado ser el mas interesante, sobre todo, por la novedad de aspectos de la ingeniería de software novedosos para mi y que he podido introducirme a traves de este trabajo. Sin ninguna duda, todos ellos son aspectos que espero profundizar y poner en práctica en un futuro inmediato. Desde este punto de vista, valoro el trabajo realizado como un elemento motivador para seguir profundizando y aprendiendo en este apasionante mundo de la ingeniería del software.

