

INGENIERIA DE SOFTWARE

Trabajo Final de Grado

ANÁLISIS Y DISEÑO DE UN PORTAL DE VENTA DE MATERIAL DEPORTIVO



Jordi Cid Rodríguez -GII -
Consultor: Oriol Martí Girona
Junio 2016



Objetivo

Análisis y diseño de una aplicación Web B2C (venta de material deportivo de montaña), de forma que a partir de dicha documentación, se puede iniciar las fases de implementación y verificación, antes de su puesta en producción.

La documentación se realizó 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 pudo aplicar de manera estricta y en particular el aspecto iterativo e incremental.

Otras tres características que se tuvieron 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 lado y tomando como referencia los principios de **MDA** (Model-Driven Architecture) se procuró separar los requerimientos funcionales y de diseño de la infraestructura o plataforma tecnológica, de forma que dicho análisis y diseño pudieran ser reutilizable. Es por ello, que en este trabajo se evitó hacer alusión a aspectos relacionados con una tecnología de implementación específica. Sobre este último aspecto, en este proyecto existe un apartado de requerimientos no funcionales que son los que a posteriori se tendrían que considerar de cara a concretar la plataforma tecnológica en que se quiera desarrollar e implementar dicha aplicación.

Por último y como consecuencia de considerar como marco de desarrollo UP, se procuró que fuesen 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, de forma, que la ingeniería de requisitos y en particular los casos de uso se convierten en el eje vertebrador de dicho trabajo.



Definición del proyecto

La empresa Adventure Sport S.A., está especializada en la venta de material deportivo de montaña. 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 material y venta de éste. También 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 realizado por éste.



Ciclo de vida del producto

1.3 Enfoque y método seguido

Todo y que en muchos aspectos UP fué 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 utilizó 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:

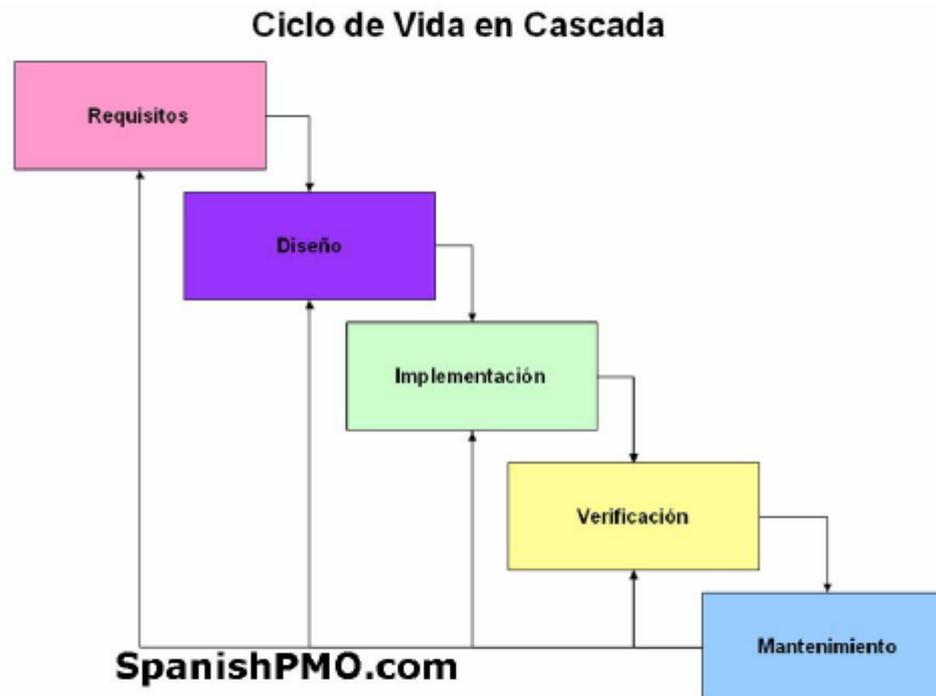


Figura 1. Ciclo de vida en Cascada

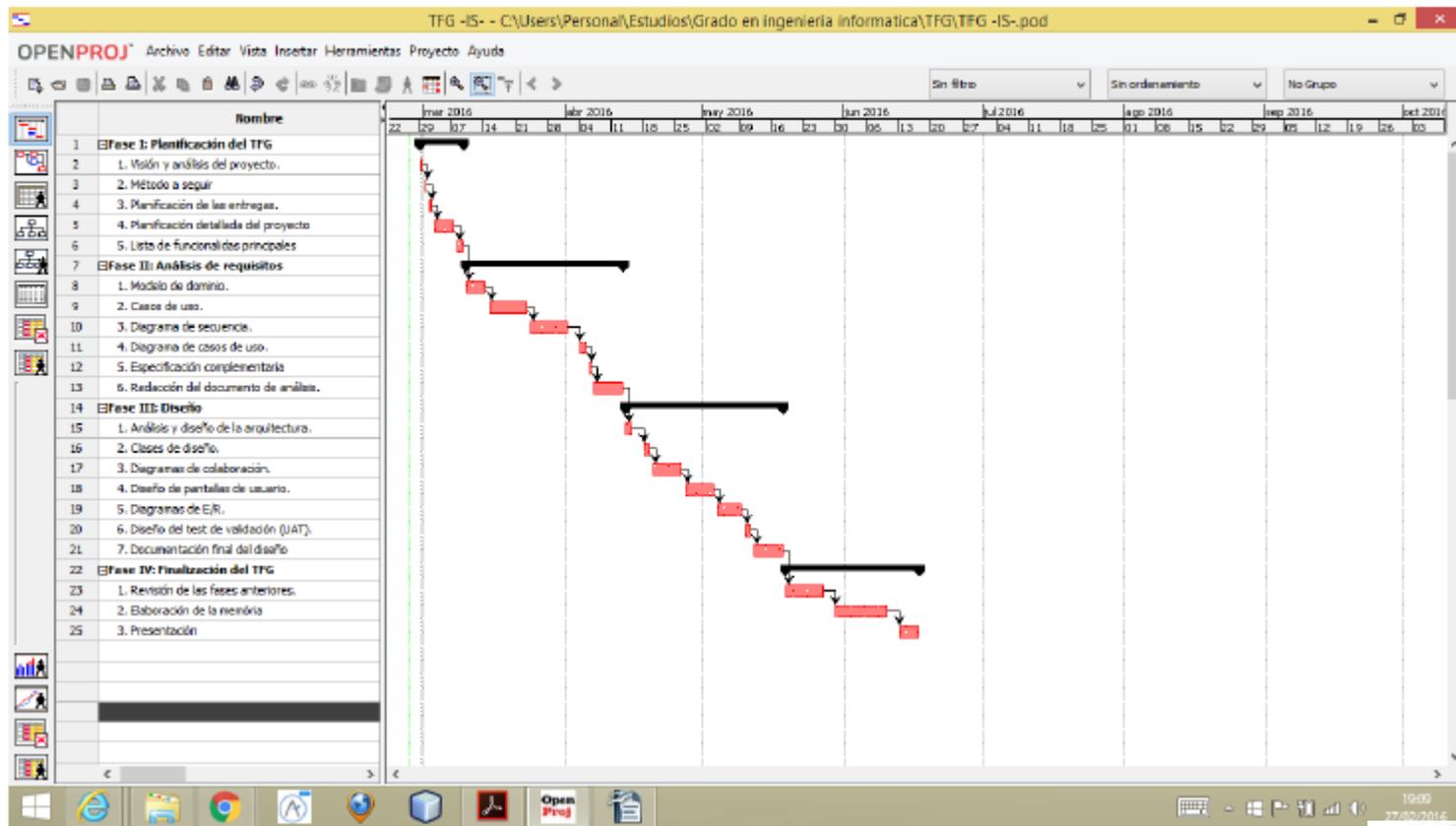
en este caso la etapa de implementación, verificación y mantenimiento no se tuvo en cuenta, centrándome exclusivamente en el análisis de requisitos y diseño.



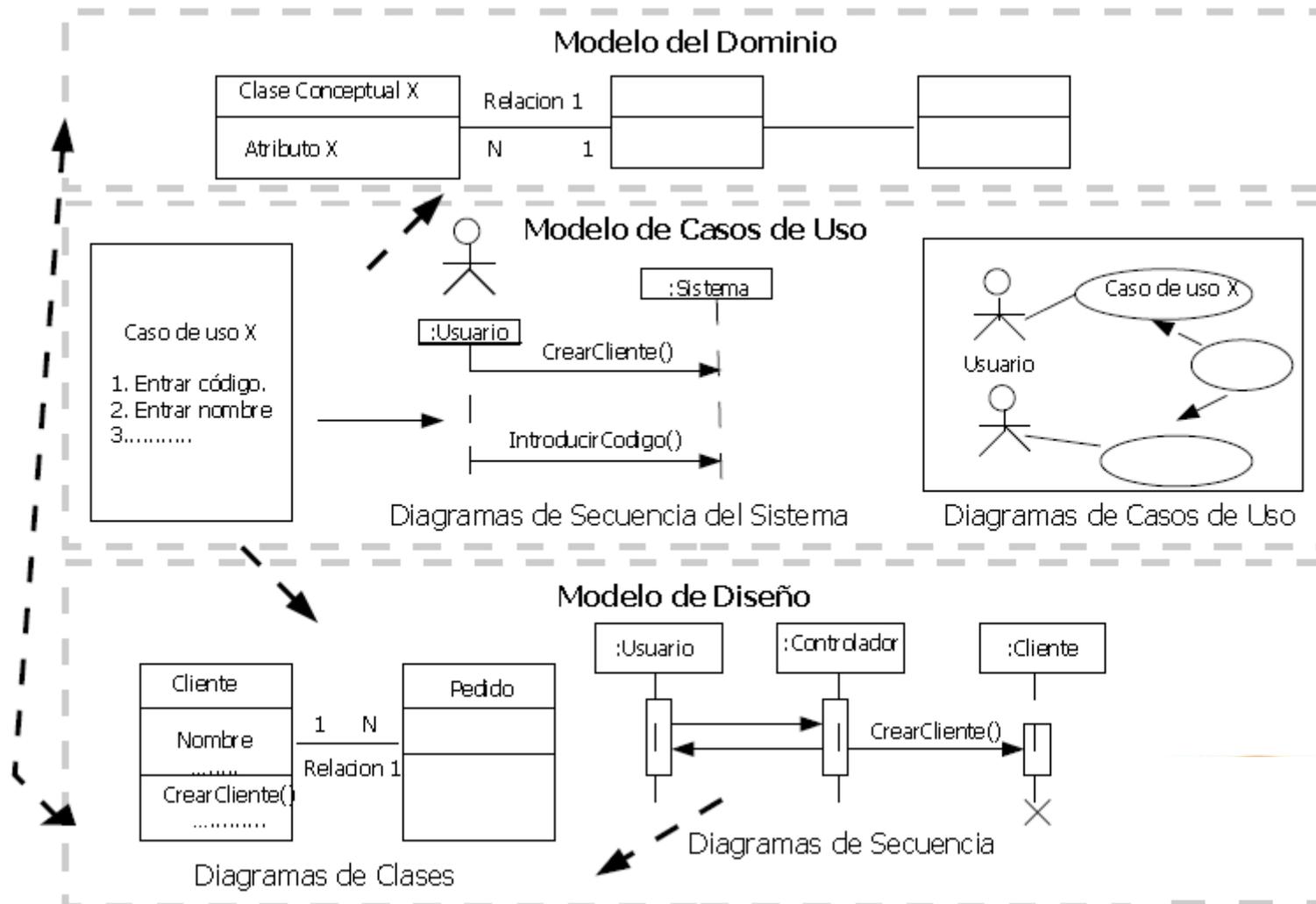
Planificación

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

Diagrama de Gantt.



Relación entre artefactos UML del UP



Análisis: Actores-Casos de Uso

Dentro del análisis también se han identificado los principales casos de uso relacionados con cada actor. Como resultado de ello se ha obtenido la siguiente relación:

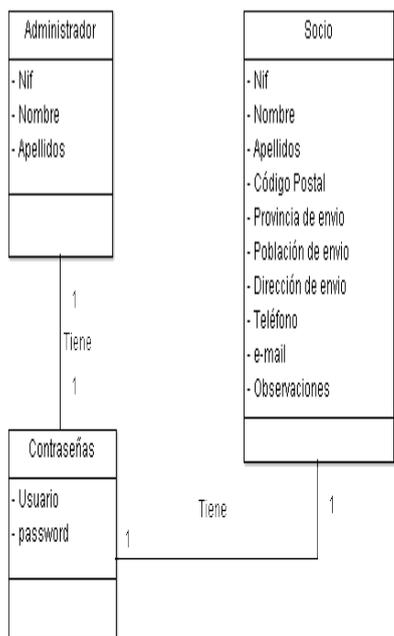
Actor-Caso de Uso			
N.CU	Actor	Funcionalidad	Descripción
CU01	Administrador o socio	Identificación	Permitirá identificarse.
CU02	Administrador	Alta de categorías	Permitirá dar de alta en el sistema las categorías.
CU03	Administrador	Consulta de categorías	Permitirá consultar las categorías.
CU04	Administrador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



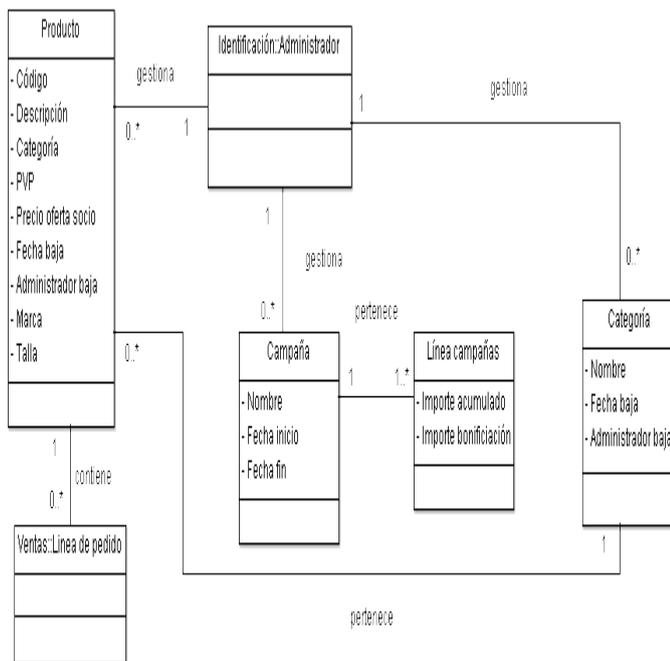
Análisis: Modelo de Dominio

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

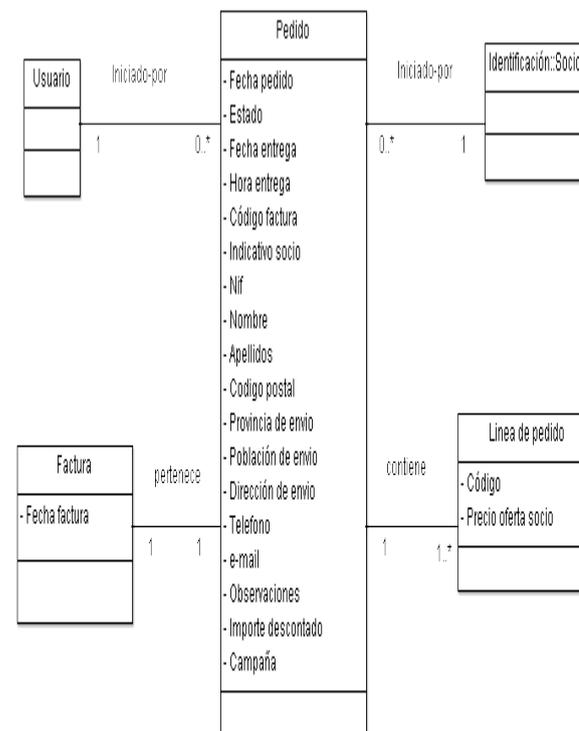
Paquete identificación



Paquete productos



Paquete ventas



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	CU 02
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	Día.
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 amanca 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>permítendole</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>permítendole</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.

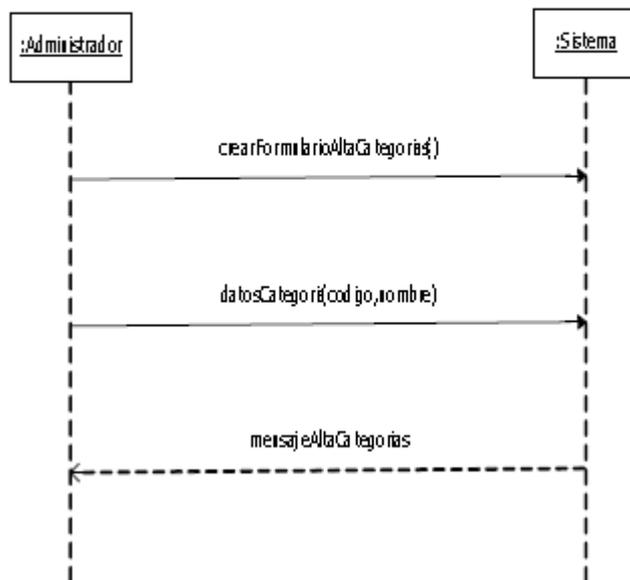
ID	CU 03
Identificador del CU	Consulta de categorías
Objetivo del CU	<u>Permitir</u> que el administrador consulte las categorías vigentes.
Caso de uso relacionado	CU04 Mantenimiento de categorías-
Actores	Administrador
Frecuencia	Día
Condiciones Previas	El administrador se ha identificado en el sistema 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 amanca 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 categoría vigentes.
3	El sistema espera 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.



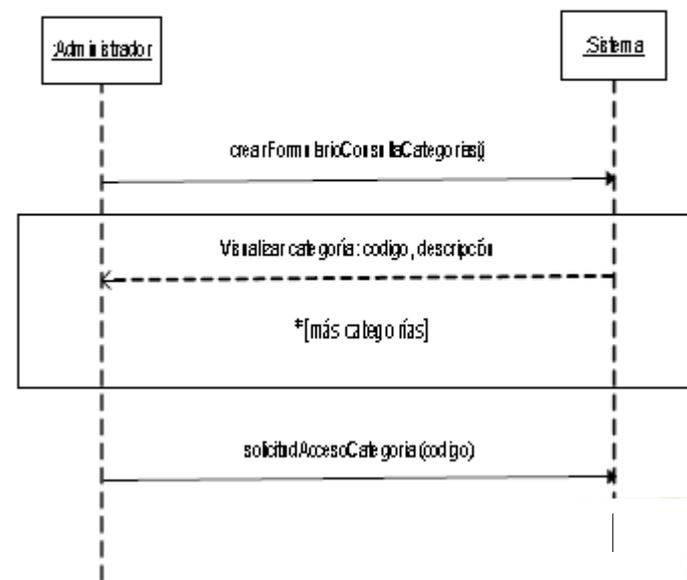
Análisis: Diagrama de secuencia

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

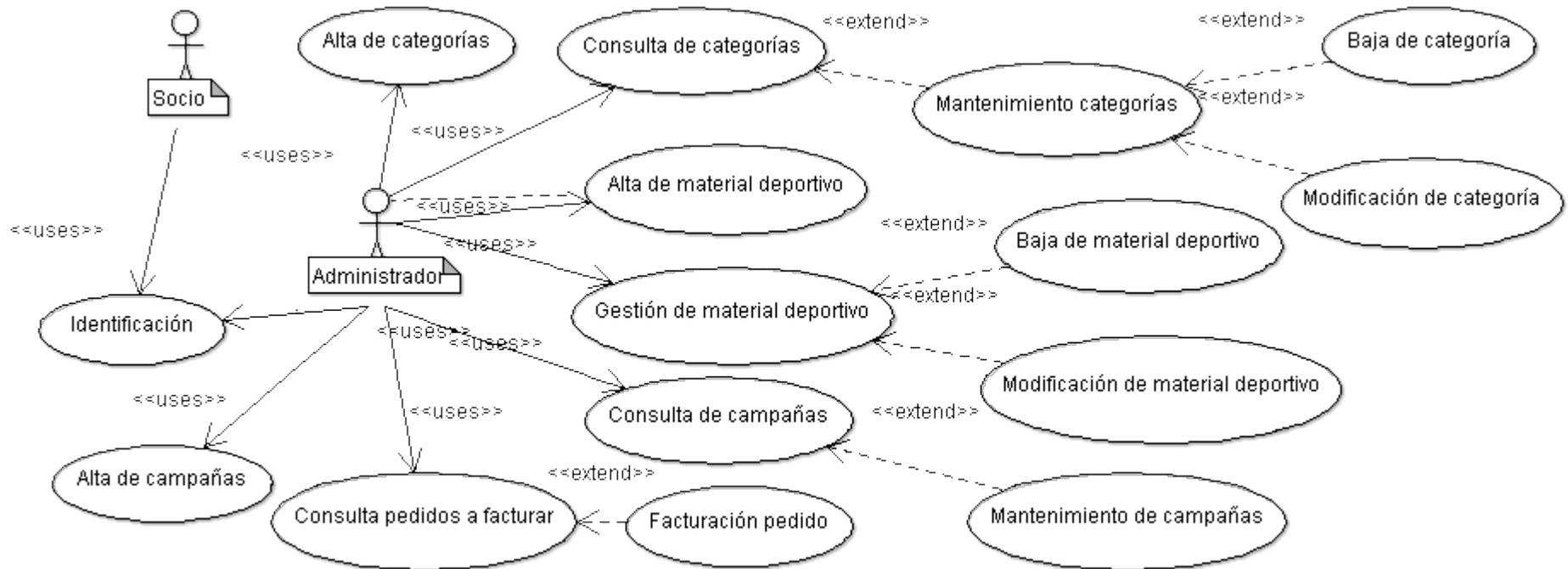


CU03 Consulta Categorías



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 también 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).

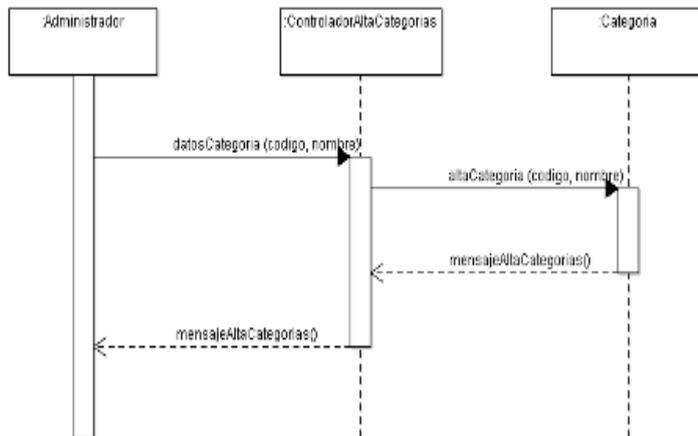
Por último y antes de entrar en los diagramas de interacción, he intentado basarme en los principios de diseño GRASP. GRASP es el acrónimo de General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades), y lo componen varios patrones, en el proyecto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



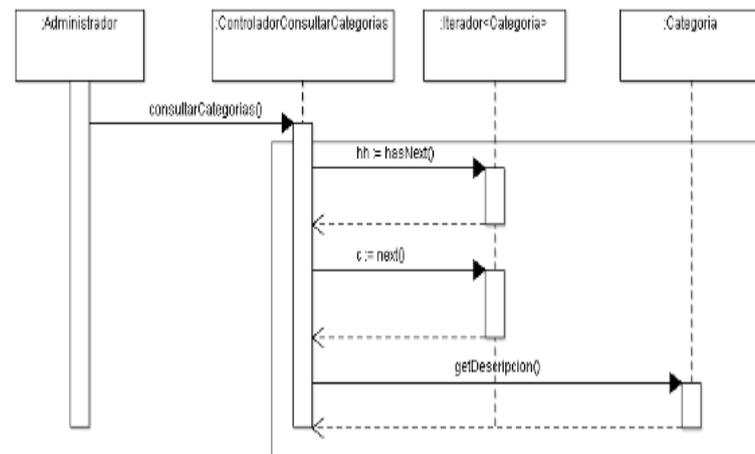
Diseño: Diagramas de secuencia

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

Vista: Administrador.
Controlador: ControladorAltaCategorias.
Modelo: Categoria.

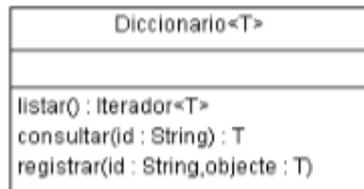
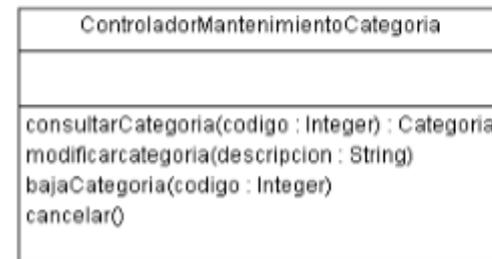
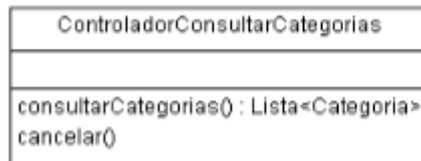
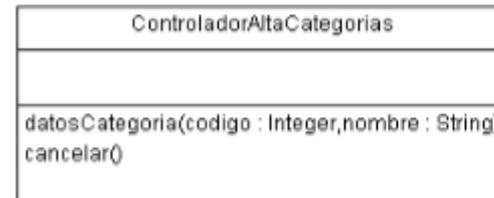
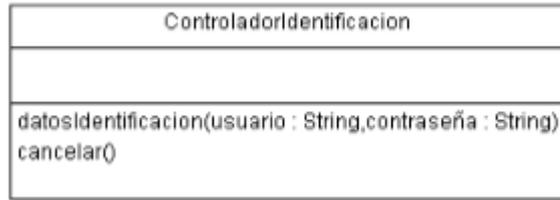


Vista: Administrador.
Controlador: ControladorConsultarCategorias.
Modelo: Iterador, Categoria.



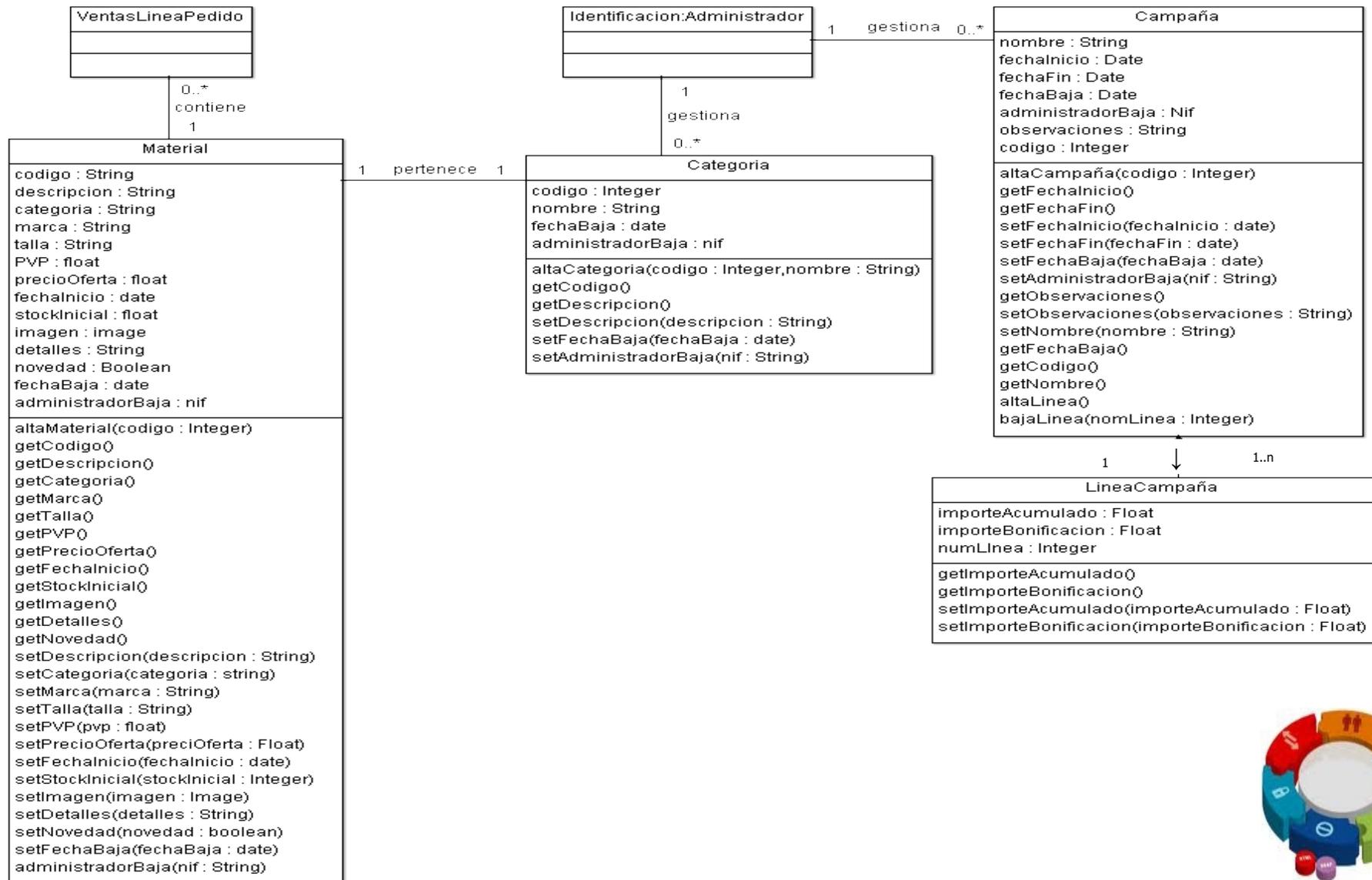
Diseño: Diagramas de clases I

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.



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 interface que denominaremos BDLibreria.

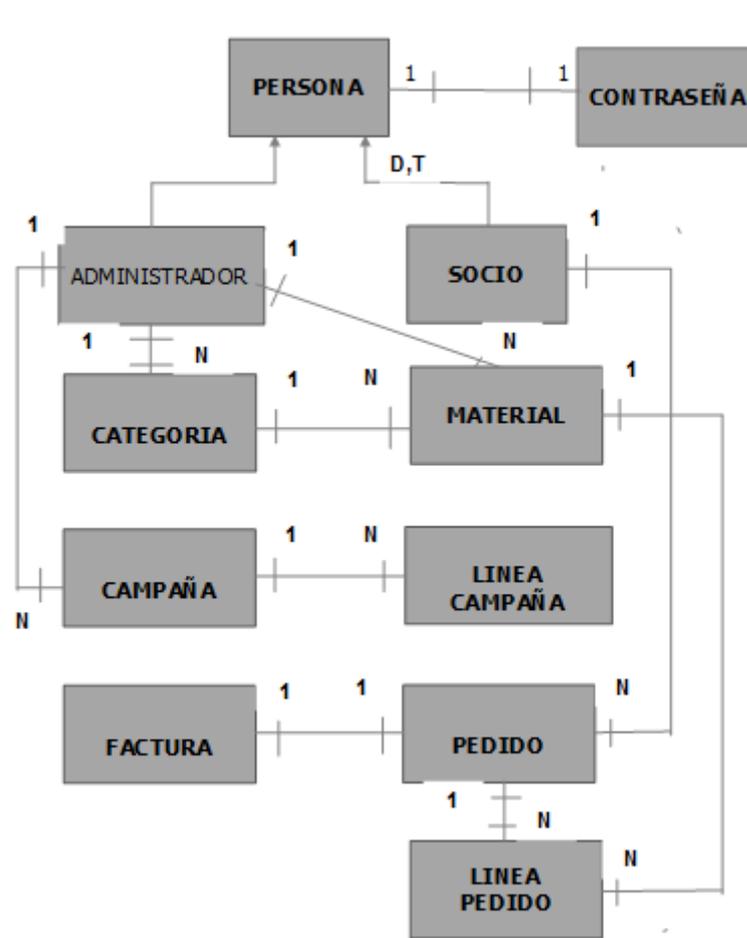
<<interface>> BDLibreria
consultarCategorias() : Iterador<Categoria>
consultarCategoria(codigo : Integer) : Categoria
almacenarCategoria(datos) : Categoria
eliminarCategoria(codigo : Integer) : Categoria
consultarMaterial(codigo : Integer) : Material
almacenarMaterial(datos) : Material
eliminarMaterial(codigo : Integer) : Material
consultarCampañas() : Iterador<Campañas>
consultarLineasCampañas(numLinea : Integer) : LineasCampaña
almacenarCampañas(codigo : Integer) : Campaña
almacenarLineasCampañas(datosLinea) : LineasCampaña
consultarCampaña(codigo : String) : Campaña
eliminarCampaña(codigo : String) : Campaña
consultarMaterial(criterios) : Iterador<Material>
consultarMaterial(codigo : Integer) : Material
almacenarPedido(datos) : Pedido
almacenarSocio(datos) : Socio
almacenarContraseña(usuario : String,password : String) : Contraseña
consultarPedidos() : Iterador<Pedidos>
consultarPedido(Id : Integer) : Pedido
consultarLineasPedidos(Id : Integer) : LineasPedido
facturarPedido(datos) : Pedido
facturarPedido(datos) : Factura
eliminarMaterial(codigo : Integer) : Material



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 II

A parte de tener testing en las diferentes fases del proyecto, también 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.. Escenario principal.	Step 1	Acceder a la nueva pantalla Alta de Categorías desde el menu 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 mantiene en formulario.
		Step5	Se pulsa el botón Cancelar	Salimos del formulario y volvemos al menú principal
		Step6	Visualizar el menu del administrador	El menú que aparece es el del administrador
Excepciones		Step 1	Acceder a la nueva pantalla Alta de Categorías desde el menu 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 mantiene en el formulario.
		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. Escenario principal.	Step 1	Acceder a la nueva pantalla Consulta de Categorías desde el menu principal del administrador.	Acceder a la pantalla sin problemas
		Step2	Visualizar la información de pantalla	El sistema visualiza solo el código 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
Acceso al mantenimiento de categorías		Step 1	Acceder a la nueva pantalla Consulta de Categorías desde el menu principal del administrador.	Acceder a la pantalla sin problemas
		Step 2	Visualizar la información de pantalla	El sistema visualiza solo el código 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 visualiza el caso de uso Mantenimiento Categorías y muestra la información sobre dicha categoría
		Step 4	El usuario da 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 sistem visualiza solo el código 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 parte de los conocimientos sobre ingeniería de software realizados 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. Intentar realizar el trabajo bajo una filosofía 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 los dos últimos puntos son los más interesantes, sobre todo porque como responsable de proyectos a nivel profesional, he visto como se han producido equívocos en los productos finales, por existir visiones distintas entre las necesidades reales manifestadas por los stakeholders y la implementación realizada bajo una visión, a veces, excesivamente tecnológica, al fin y al cabo, llevar adelante un proyecto es una tarea multidisciplinar y para ello es importante disponer de una visión holística del proyecto que estamos tratando.

