



# TFG – Ingeniería de Software

## ANÁLISIS Y DISEÑO DE UN PORTAL DE VENTA DE MATERIAL DEPORTIVO

**Autor:** *Jordi Cid Rodríguez*

**Consultor:** *Oriol Martí Girona*

**Fecha de entrega:** *06/2016*

**Área del trabajo:** *Ingeniería del software*

**Titulación:** *Grado en ingeniería informática*



Esta obra está sujeta a una licencia de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](#).

### **GNU Free Documentation License (GNU FDL)**

Copyright © ANY EL-TEU-NOM.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

### **C) Copyright**

© (el autor/a)

Reserva todos los derechos. Está prohibida la reproducción total o parcial de esta obra por cualquier medio o procedimiento, incluida la impresión, la reprografía, el microfilm, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

## Agradecimientos

Tanto este proyecto como toda la carrera no hubiera sido posible tirar adelante sin aquellas personas que me han apoyado, animado y porque no soportado, en todo momento. En especial a mi mujer Merce y a mis hijos Adriá y Laura.

Muchas gracias por vuestra paciencia.

## Resumen del proyecto

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 pudiesen 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.

**Palabras clave:** B2C, UML, OO, UP.

**Abstract**

Analysis and design of a B2C Web application (selling mountain sports equipment), so that from this documentation, you can start the implementation and verification phases, before putting them into production.

The documentation was made based on the modeling language UML (Unified Modeling Language), under the paradigm OO (Object-Oriented) and using UP (Unified Process) as a development framework, although in the latter and given the characteristics of secuencialidad the project can not be applied strictly and in particular the iterative and incremental aspect.

Three other features that will be considered in the project are:

On the one hand the application of design patterns and in particular those related to the allocation of responsibilities GRASP (General Responsibility Assignment Software Patterns).

On the other hand and with reference to the principles of MDA (Model-Driven Architecture) will seek to separate the functional and design requirements infrastructure or technology platform, so that the analysis and design can be reusable. That is why, in this work should be avoided to refer to aspects related to technology specific implementation. On this last point, this project will be a section of nonfunctional requirements are those that post would have to consider concrete facing technological platform you want to develop and implement the application.

Finally, as a result of considered development framework UP, it sought to be the use cases the focus and more relevant on which all other artifacts of analysis and design used is despliega so that engineering particular requirements and use cases becomes the backbone of the work.

**Key words:** B2C, UML, OO, UP.

## Índice

1	Introducción .....	1
1.1	Contexto y justificación del Trabajo .....	1
1.2	Objetivos del Trabajo .....	1
1.3	Enfoque y método seguido .....	2
1.4	Planificación del Trabajo .....	3
1.5	Productos obtenidos .....	5
1.6	Resumen de los otros capítulos de la memoria .....	5
2	Análisis .....	6
2.1	Actores .....	6
2.2	Identificación y relación actores-casos de uso .....	6
2.3	Modelo del dominio .....	7
2.3.1	Paquete Identificación .....	9
2.3.2	Paquete productos .....	9
2.3.3	Paquete Ventas .....	10
2.4	Casos de uso .....	10
2.4.1	CU01 Identificación .....	11
2.4.2	CU02 Alta de categorías .....	12
2.4.3	CU03 Consulta de categorías .....	12
2.4.4	CU04 Mantenimiento de categorías .....	13
2.4.5	CU05 Alta de material deportivo .....	14
2.4.6	CU06 Mantenimiento de material deportivo .....	15
2.4.7	CU07 Alta de campañas .....	16
2.4.8	CU08 Consulta campañas .....	17
2.4.9	CU09 Mantenimiento de campañas .....	17
2.4.10	CU10 Selección de material deportivo .....	18
2.4.11	CU11 Resultado consulta material deportivo .....	18
2.4.12	CU12 Gestión del pedido en curso .....	19
2.4.13	CU13 Datos de envío .....	20
2.4.14	CU14 Consulta pedidos formalizados .....	21
2.4.15	CU15 Facturación pedido .....	21
2.5	Diagrama de secuencia (del sistema) .....	22
2.5.1	Identificación .....	22
2.5.2	Alta de categorías .....	23
2.5.3	Consulta de categorías .....	23
2.5.4	Mantenimiento de categorías .....	24
2.5.5	Alta material deportivo .....	24
2.5.6	Mantenimiento material deportivo .....	25
2.5.7	Alta Campaña .....	26
2.5.8	Consulta Campañas .....	26
2.5.9	Mantenimiento Campañas .....	27
2.5.10	Selección de material deportivo .....	27
2.5.11	Resultado Consulta material deportivo .....	28
2.5.12	Gestión del Pedido en Curso .....	28
2.5.13	Datos de Envío .....	29

2.5.14 Consulta pedidos formalizados .....	29
2.5.15 Facturación del Pedido .....	30
2.6 Diagrama de casos de uso .....	31
2.7 Especificación complementaria o requisitos no funcionales .....	32
3 Diseño .....	33
3.1 Análisis y diseño de la arquitectura .....	33
3.2 Diagramas de interacción .....	35
3.2.1 Responsabilidades en UML .....	35
3.2.2 Responsabilidades y objetos .....	35
3.2.3 Responsabilidades y diagramas de interacción .....	35
3.2.4 Patrones GRASP .....	36
3.2.4.1 Experto .....	36
3.2.4.2 Creador .....	36
3.2.4.3 Bajo acoplamiento .....	36
3.2.4.4 Alta cohesión .....	36
3.2.4.5 Controlador .....	36
3.3 Diagramas de secuencia (diseño) .....	37
3.3.1 CU01 Identificación .....	38
3.3.2 CU02 Alta de categorías .....	38
3.3.3 CU03 Consulta de categorías .....	39
3.3.4 CU04 Mantenimiento de categorías .....	39
3.3.5 CU05 Alta de material deportivo .....	40
3.3.6 CU06 Alta de material deportivo .....	41
3.3.7 CU07 Alta de campañas .....	42
3.3.8 CU08 Consulta campañas .....	42
3.3.9 CU09 Mantenimiento de campañas .....	43
3.3.10 CU10 Selección de material deportivo .....	44
3.3.11 CU11 Resultado consulta material deportivo .....	44
3.3.12 CU12 Gestión del pedido en curso .....	45
3.3.13 CU13 Datos de envío .....	46
3.3.14 CU14 Consulta pedidos formalizados .....	47
3.3.15 CU15 Facturación pedido .....	47
3.4 Clases de diseño .....	48
3.4.1 Controladores del dominio .....	48
3.4.2 Paquete identificación .....	51
3.4.3 Paquete productos .....	52
3.4.4 Paquete ventas .....	53
3.5 Servicios técnicos .....	53
3.6 Diseño de la interficie gráfica de usuario .....	54
3.6.1 Menú usuario .....	55
3.6.2 Menú administrador .....	55
3.6.3 CU01 Identificación .....	56
3.6.4 CU02 Alta Categorías .....	56
3.6.5 CU03 Consultar categorías .....	57
3.6.6 CU04 Mantenimiento categorías .....	57
3.6.7 CU05 Alta material deportivo .....	58
3.6.8 CU06 Mantenimiento material deportivo .....	58
3.6.9 CU07 Alta campañas .....	59

3.6.10 CU08 Consulta campañas .....	59
3.6.11 CU09 Mantenimiento campañas .....	60
3.6.12 CU10 Consulta material deportivo .....	60
3.6.13 CU11 Resultado consulta material deportivo .....	61
3.6.14 CU12 Gestión del pedido en curso .....	61
3.6.15 CU13 Datos de envío .....	62
3.6.16 CU14 Consulta pedidos .....	62
3.6.17 CU15 Facturación pedidos .....	63
3.7 Diagrama de E/R (persistencia) .....	63
3.8 Diseño del test .....	68
4 Aspectos relacionados con la gestión del proyecto .....	82
5 Conclusiones .....	84
6 Glosario .....	85
7 Bibliografía .....	87
8 Anexos .....	88



## Lista de figuras

Figura 1. Ciclo de vida en Cascada.....	2
Figura 2. Planificación del proyecto.....	5
Figura 3. Relación actores-casos de uso.....	7
Figura 4. Módulos del modelo de dominio.....	8
Figura 5. Módulo Identificación.....	9
Figura 6. Módulo productos.....	9
Figura 7. Módulo Ventas.....	10
Figura 8. Diagrama de casos de uso y actores.....	31
Figura 9. Controladores de dominio.....	50
Figura 10. Diagrama de clase Módulo Identificación.....	51
Figura 11. Diagrama de clases módulo productos.....	52
Figura 12. Diagrama de clases módulo ventas.....	53
Figura 13. Servicios técnicos.....	54
Figura 14. Diagrama E/R.....	64
Figura 15. Modelo de Testing.....	69

# 1. Introducción

## 1.1 Contexto y justificación del Trabajo

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.

## 1.2 Objetivos del Trabajo

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 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 tuvo en cuenta en el proyecto son:

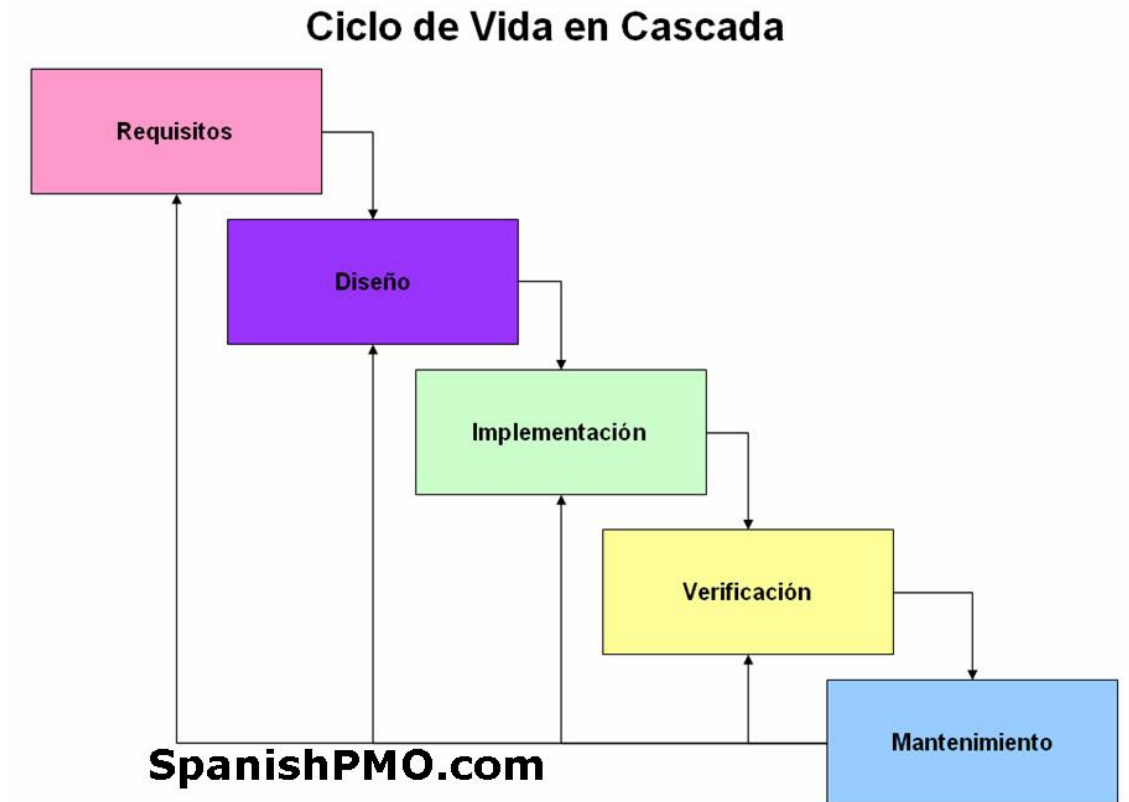
Por un lado la aplicación de patrones de diseño y en particular los relacionados con la asignación de responsabilidades **GRASP** (General Responsibility Assignment Software Patterns).

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 pueda 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.

### 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.

## 1.4 Planificación del Trabajo

Teniendo en cuenta la metodología a seguir, la distribución del contenido de las entregas queda de la siguiente forma:

**Fase 1. Planificación del proyecto:** En esta fase, se indicaron los objetivos del proyecto, se identificaron las funcionalidades principales y se dió una planificación del proyecto.

**Fase 2. Análisis de requisitos:** En esta fase, se analizará e identificará de forma detallada los requisitos/funcionalidades del sistema. Para ello se utilizará los diferentes tipos de diagramas existentes en UML, y también se diseñarán los test de **UAT** (User Acceptance Testing) que se tendrán que ejecutar en la etapa de validación del sistema desarrollados y en particular nos permitirá validar los requisitos del sistema.

**Fase 3. Diseño:** En esta fase, se analizará en detalle el patrón arquitectónico a seguir. Se realizará la identificación tanto de las clases como de las entidades de persistencia en cuanto a su número, relación, cardinalidad, atributos, etc..., así como de los patrones de diseño a utilizar, y el detalle de las pantallas gráficas de usuario.

En cuanto al número y fechas de entrega queda resumido de la siguiente forma:

PAC	Objetivo	Fecha inicio	Fecha fin
1	Plan de trabajo y análisis preliminar de requerimientos	29/02/16	09/03/16
2	Análisis	10/03/16	13/04/16
3	Diseño	14/04/16	18/05/16
Entrega final	Memoria del proyecto y presentación	19/05/16	17/06/16

Teniendo en cuenta todo lo anterior, la planificación detallada de las tareas del proyecto y los objetivos de cada una de ellas, queda de la siguiente forma:

LISTA DE TAREAS	Fecha inicio	Fecha fin
<b>Fase I: Planificación del TFG</b>	29/02/16	09/03/16
1. Visión y análisis del proyecto.	29/02/16	29/02/16
2. Método a seguir	01/03/16	01/03/16
3. Planificación de las entregas.	02/03/16	02/03/16
4. Planificación detallada del proyecto.	03/03/16	07/03/16
5. Lista de funcionalidades principales.	08/03/16	09/03/16
<b>HITO: Entrega de la PAC1</b>		
<b>Fase II: Análisis de requisitos</b>	10/03/16	13/04/16
1. Modelo de dominio.	10/03/16	14/03/16

2. Casos de uso.	15/03/16	23/03/16
3. Diagrama de secuencia.	24/03/16	01/04/16
4. Diagrama de casos de uso.	04/04/16	05/04/16
5. Especificación complementaria.	06/04/16	06/04/16
6. Redacción del documento de análisis.	07/03/16	13/04/16
<b>HITO: Entrega de la PAC2</b>		
<b>Fase III: Diseño</b>	14/04/16	18/05/16
1. Análisis y diseño de la arquitectura.	14/04/16	15/04/16
2. Clases de diseño.	18/04/16	19/04/16
3. Diagramas de colaboración.	20/04/16	26/04/16
4. Diseño de pantallas de usuario.	27/04/16	03/05/16
5. Diagramas de E/R.	04/05/16	09/05/16
6. Diseño del test de validación (UAT)	10/05/16	11/05/16
7. Documentación final del diseño	12/05/16	18/05/16
<b>HITO: Entrega de la PAC3</b>		
<b>Fase IV: Finalización del TFG</b>	19/05/16	17/06/16
1. Revisión de las fases anteriores.	19/05/16	27/05/16
2. Elaboración de la memoria.	30/05/16	10/06/16
3. Elaboración de la presentación.	13/06/16	17/06/16
<b>HITO: Entrega final</b>		

## Diagrama de Gantt.

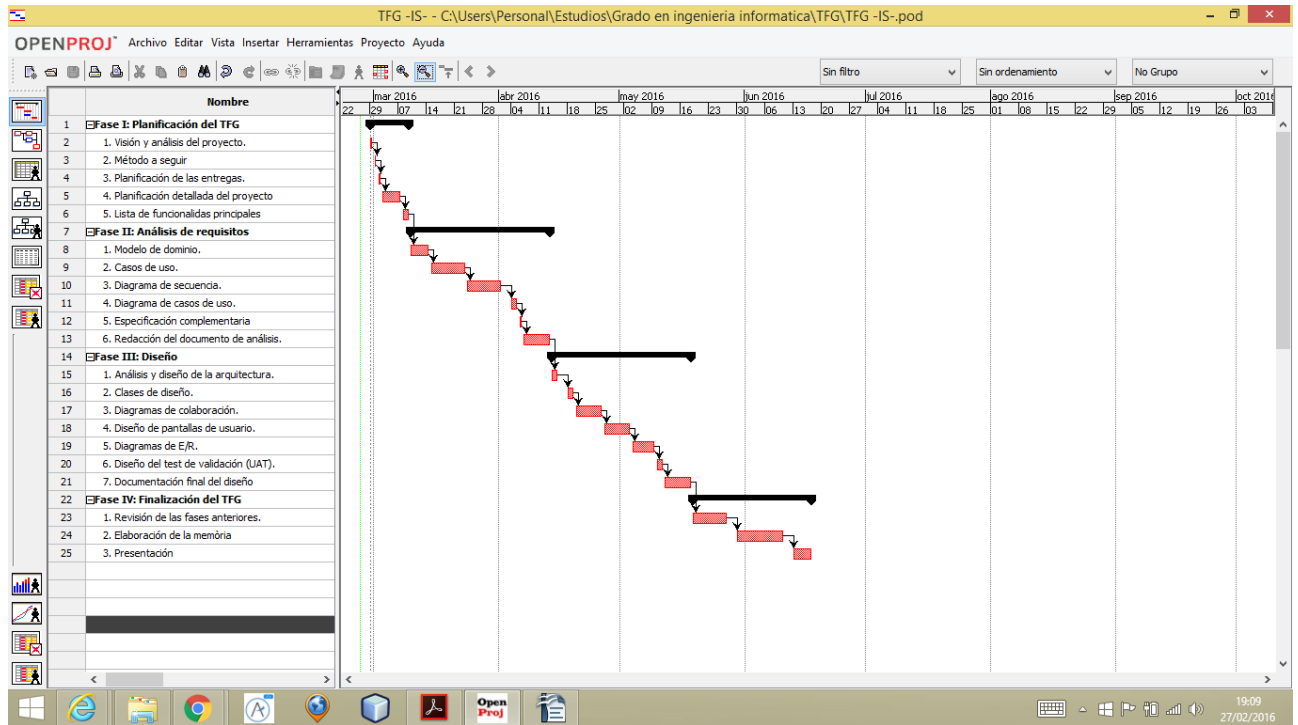


Figura 2. Planificación del proyecto

## 1.5 Productos obtenidos

**Documento de análisis:** Contiene a las especificaciones del proyecto y donde se utilizará los diferentes tipos de diagramas existentes en UML, y también se diseñarán los test de **UAT** (User Acceptance Testing) que se tendrán que ejecutar en la etapa de validación del sistema desarrollados y en particular nos permitirá validar los requisitos del sistema.

**Documento de diseño:** Contiene las especificaciones técnicas de diseño del software, partiendo del documentos anterior. Se analiza detalle el patrón arquitectónico a seguir. Se realiza la identificación tanto de las clases como de las entidades de persistencia en cuanto a su número, relación, cardinalidad, atributos, etc..., así como de los patrones de diseño a utilizar, y el detalle de las pantallas gráficas de usuario.

## 1.6 Resumen de los otros capítulos de la memoria

En los próximos capítulos de la memoria se describirán de forma detallada los siguientes temas:

**Capítulo 2. Análisis funcional.** Donde se recogen los requerimientos solicitados por los *stakeholders* al sistema.

**Capítulo 3. Diseño.** Donde se recogen principalmente los aspectos de la arquitectura, aplicación de patrones, diseño técnico, clases, diagramas E/R (persistencias BB.DD), interface de usuario y diseño del test de aceptación (UAT).

**Capítulo 4. Coste económico.** Donde se recoge una valoración aproximada del coste del proyecto.

**Capítulo 5. Conclusiones.** Resumen final sobre el proyecto, así como reflexión entre lo previsto y lo realizado.

**Glosario.** Donde se realizará una relación de los principales conceptos que figuran en el proyecto.

**Bibliografía.** Listado con referencias a la bibliografía consultada y cualquier otra fuentes de información.

## 2. Análisis

Después de haber realizado la toma de datos mediante entrevistas con usuarios, revisión de documentos, investigación sobre fuentes externas con la misma línea de negocio y aplicación de técnicas de grupos, se han identificado tanto los actores así como las principales funcionalidades que deben contemplarse en el sistema.

### 2.1. Actores.

El concepto de actor permite identificar cualquier entidad externa que interactúa con el sistema con el objetivo de que éste dé respuesta a una funcionalidad determinada. Los actores pueden ser humanos, sistemas externos y entidades abstractas. En el actual proyecto se han identificado los siguientes actores que coinciden con el tipo de usuario que accederá a la aplicación:

**Administrador:** Es todo aquel empleado de la tienda cuyo objetivo es acceder al sistema para mantener la información de categorías, material deportivo, campañas y cursar la facturación de pedidos formalizadas a través de la web. Por su singularidad, la información del administrador será mantenida a través de una operativa no contemplada en este trabajo.

**Usuario:** Es toda aquella persona que entra en la web con permisos para poder ejecutar cualquiera de las funcionalidades, excepto las propias del administrador.

**Socio:** Es toda persona que está registrada en el sistema. Esto significa que el sistema dispone de forma permanente de la siguiente información personal: DNI, nombre y apellidos y dirección de envío de los pedidos. Para ello es suficiente que índice su usuario y password. A nivel funcional tendrá las mismas funcionalidades que el usuario, y además podrá beneficiarse de las campañas de marketing, así como agilizar la petición de sus pedidos al no tener que introducir siempre su información personal.

### 2.2. Identificación y relación 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.

**Figura 3. Relación actores-casos de uso**

### 2.3. Modelo del dominio.

Un modelo del dominio es en esta fase una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. Utilizando la notación UML, un modelo de dominio se representa con un conjunto de diagramas de clases en los que no se define ninguna operación. Donde se pueden mostrar:

Objetos del dominio o clases conceptuales.

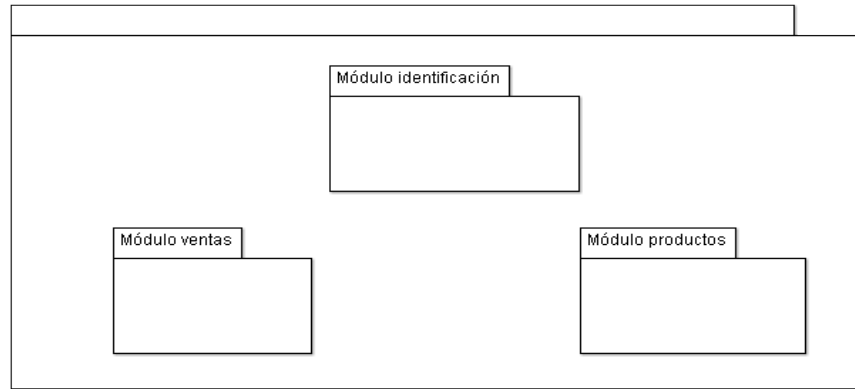
Asociaciones entre clases conceptuales.

Atributos de las clases conceptuales.

Los modelos del dominio en ningún momento son modelos de componentes de software, sin embargo, servirán de base para establecer en la fase de diseño el diagrama de clases de la aplicación.

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





**Figura 4. Módulos del modelo de dominio**

2.3.1. Paquete Identificación.

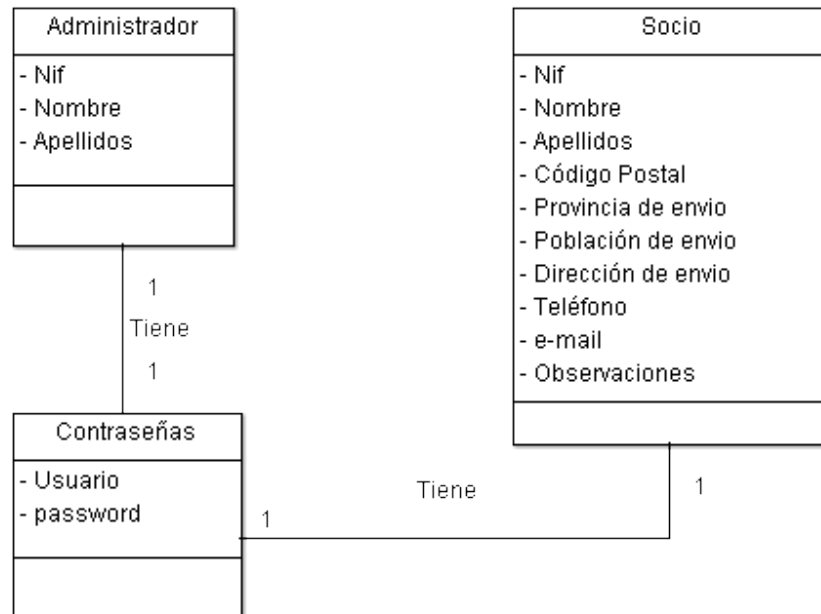


Figura 5. Módulo Identificación

2.3.2. Paquete productos

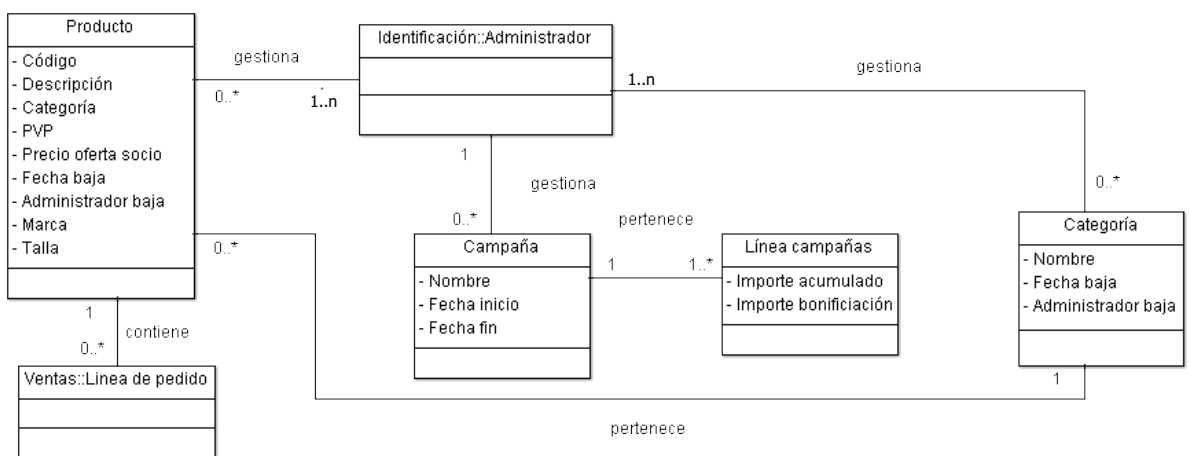


Figura 6. Módulo productos

### 2.3.3. Paquete Ventas



Figura 7. Módulo Ventas

### 2.4. Casos de uso.

Los casos de uso se describen desde el punto de vista de un observador externo (caso de uso de caja negra), sin describir el funcionamiento interno del sistema y poniendo énfasis en qué debe hacer el sistema y no en cómo lo hará. Para ello, conceptualmente en los casos de uso se han de identificar dos elementos esenciales:

**Actor:** Algo con comportamiento (persona, otro programa, organización..), que interactúa con el sistema.

**Escenario:** (instancia de caso de uso): Secuencia de acciones e interacciones entre el actor y el sistema.

Teniendo en cuenta esto, el caso de uso lo podemos definir como la colección de escenarios (éxito y fracaso) que describen actores que usan el sistema para conseguir un objetivo.

Los casos de uso serán el elemento central que determinará cualquier decisión de análisis y diseño, dado que ambas actividades han de estar orientadas a cumplir con las indicaciones del caso de uso. Además, y en este trabajo, los casos de pruebas de caja negra (pruebas que están orientadas a verificar que el comportamiento del sistema satisface los requisitos sin conocer como están implementados) también se diseñan a partir de los casos de uso. Como se podrá ver los casos de uso nos describen mediante una secuencia de pasos como se ha de llevar a cabo la ejecución de la funcionalidad correspondiente.

Dentro de los casos de uso, cabe señalar por su relevancia, el apartado de condiciones posteriores, dado que describen cambios en el estado de los objetos del Modelo del Dominio.

Los cambios de estado del Modelo de Dominio, comprenden:

Creación y eliminación de instancias.

Modificación de atributos.

Formación y rotura de asociaciones (enlaces UML).

En el proyecto que nos ocupa los casos de uso identificados son:

#### 2.4.1. CU01 Identificación.

ID	CU01
Identificador del CU	Identificarse en el sistema como administrador o socio
Objetivo del CU	Permite que el administrador o socio se identifique.
Caso de uso relacionado	
Actores	Administrador o socio
Frecuencia	Diario.
Condiciones Previas	Exista previamente registrado en el sistema el administrador o socio que va a identificarse.
Condiciones Posteriores	Identificado como administrador o socio.
Acción que arranca el CU	Nuevo formulario Identificación donde el administrador o socio introducirá los datos para identificarse.
Paso	Descripción
1	El usuario accede al formulario de Identificación.
2	Introduce el usuario.
3	Introduce la contraseña.
4	El administrador o socio solicita al sistema que verifique los datos
5	El sistema verifica que los datos son correctos e informa al usuario.
6	En el caso de que el usuario sea el administrador, el sistema mostrará las opciones de acceso a los casos de uso propios del administrador.
7	El caso de uso se cancela.
Excepciones	
Número	Descripción
5	Si el sistema detecta que el usuario o la contraseña no es correcta informa de la situación permitiéndole modificar los datos proporcionados y el caso de uso continua.
5	Si el sistema detecta que falta alguno de los datos solicitados, informa de la situación permitiendo modificar los datos proporcionados y el caso de uso continua.
5	Si el usuario solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso se cancela.

**2.4.2. CU02 Alta de categorías.**

ID	CU02
Identificador del CU	Alta de categoría
Objetivo del CU	Permite que el administrador dé de alta en el sistema las categorías.
Caso de uso relacionado	
Actores	Administrador.
Frecuencia	Diaria.
Condiciones Previas	El administrador se ha identificado en el sistema mediante un nombre de usuario y una password.
Condiciones Posteriores	El sistema a registrado correctamente la información correspondiente a la categoría (código y nombre).
Acción que arranca el CU	Nuevo formulario Alta de categorías 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 permitiéndole modificar los datos proporcionados y el caso de uso continua.
4	Si el sistema detecta que falta alguno de los datos solicitados, informa de la situación al administrador permitiéndole modificar los datos proporcionados y el caso de uso continua.
4	Si el administrador solicita cancelar la operación, el sistema cancela la operación, y a continuación este caso de uso se cancela.

**2.4.3. CU03 Consulta de categorías.**

ID	CU03
Identificador del CU	Consulta de categorías
Objetivo del CU	Permitir que el administrador consulte las categorías vigentes.
Caso de uso relacionado	CU04 Mantenimiento de categorías
Actores	Administrador
Frecuencia	Diaria

Condiciones Previas	El administrador se ha identificado en el sistema mediante un nombre de usuario y una password.
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 Consulta de categorías donde el administrador podrá visualizar lo 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 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 Mantenimiento de categorías. Al volver se refrescará en el formulario la información de las categorías vigentes y el caso de uso continua.

#### 2.4.4. CU04 Mantenimiento de categorías.

ID	CU04
Identificador del CU	Mantenimiento de categoría.
Objetivo del CU	Permitir que el administrador modifique o dé de baja una categoría.
Caso de uso relacionado	CU03 Consulta de categorías.
Actores	Administrador.
Frecuencia	Diaria.
Condiciones Previas	El administrador se ha identificado en el sistema mediante un nombre de usuario y una password.
Condiciones Posteriores	El sistema a registrado correctamente la información modificada correspondiente a la categoría (descripción). En el caso de darla de baja se registra el nombre del usuario y la fecha en que se dió de baja.
Acción que arranca el CU	Nuevo formulario Consulta categoría donde el administrador ha seleccionado la categoría a modificar o a dar de baja.
Paso	Descripción
1	Accede al nuevo formulario Mantenimiento de categoría.
2	El sistema muestra el código de la categoría y no se puede modificar.
3	El sistema muestra la descripción de la categoría y se puede modificar.

4	El sistema solicita si desea o no dar de baja la categoría y se puede modificar (por defecto el sistema propone que no desea darla de baja).
5	El administrador modifica la información y solicita al sistema que registre los datos
6	El sistema registra los datos proporcionados e informa al administrador que el proceso ha finalizado con éxito y el caso de uso se cancela.
<b>Excepciones</b>	
<b>Número</b>	<b>Descripción</b>
5	Si el sistema detecta que falta la descripción, informa de la situación al usuario permitiéndole modificar los datos proporcionados y modificables y el caso de uso continua.
5	Si el usuario solicita cancelar la operación, el sistema cancela la operación, y a continuación este caso de uso se cancela.

#### 2.4.5. CU05 Alta de material deportivo.

ID	CU05
Identificador del CU	Alta de material deportivo
Objetivo del CU	Permitir que el administrador dé de alta la información sobre material deportivo.
Caso de uso relacionado	
Actores	Administrador
Frecuencia	Diario
Condiciones Previas	El material deportivo no está registrado en el sistema.
Condiciones Posteriores	El sistema ha almacenado la información correspondiente al nuevo material deportivo.
Acción que arranca el CU	Nuevo formulario Alta de material deportivo donde el actor introducirá los datos para darlos de alta.
<b>Paso</b>	
<b>Descripción</b>	
1	El administrador solicita al sistema comenzar el proceso de alta de material deportivo.
2	Introduce el código del material deportivo es obligatorio.
3	Introduce descripción es obligatorio.
4	Introduce categoría es obligatoria.
5	Introduce marca es obligatorio.
6	Introduce talla.
7	Introduce PVP es obligatoria.
8	Introduce precio oferta al socio.
9	Introduce la fecha de inicio de vigencia es obligatoria.
10	Introduce el stock inicial es obligatorio.
11	Introduce/asocia la imagen del material deportivo es obligatoria.

12	Introduce detalles del material deportivo es obligatoria.
13	Introduce indicativo de si es o no una novedad. Por defecto el sistema indicará que no es novedad.
14	El administrador solicita al sistema que registre los datos.
15	El sistema registra los datos proporcionados e informa al usuario de que el proceso ha finalizado con éxito.
<b>Excepciones</b>	
<b>Número</b>	<b>Descripción</b>
14	Si el sistema detecta que el material deportivo ya está registrado, el sistema informa de la situación al administrador permitiéndole modificar los datos proporcionados y el caso de uso continua.
14	Si el sistema detecta que falta alguno de los datos obligatorios, informa de la situación al administrador permitiéndole modificar los datos proporcionados y el caso de uso continua.
14	Si el administrador solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso se cancela.
Notas: Le fecha inicio de vigencia servirá para mostrar al usuario el material deportivo en la web. Esto permite que la información del material deportivo pueda ser modificada hasta que no entre la fecha de vigencia.	

#### 2.4.6. CU06 Mantenimiento de material deportivo.

<b>ID</b>	<b>CU06</b>
<b>Identificador del CU</b>	Mantenimiento de la información de un material deportivo.
<b>Objetivo del CU</b>	Permitir que el administrador modifique o dé de baja un material deportivo.
<b>Caso de uso relacionado</b>	
<b>Actores</b>	Administrador
<b>Frecuencia</b>	Diario
<b>Condiciones Previas</b>	El material deportivo está registrado en el sistema y no está dado de baja.
<b>Condiciones Posteriores</b>	El sistema ha almacenado la información modificada correspondiente al <b>material deportivo</b> .
<b>Acción que arranca el CU</b>	Nuevo formulario <b>Mantenimiento de material deportivo</b> donde el actor introducirá/modificará los datos.
<b>Paso</b>	<b>Descripción</b>
1	El administrador solicita al sistema comenzar el proceso de modificación de material deportivo.
2	El usuario introduce el código y el sistema recupera la información del material deportivo.
3	El sistema muestra la descripción y no se puede modificar si ya ha entrado en vigencia.
4	El sistema muestra la categoría y no se puede modificar si ya ha entrado en vigencia.
5	El sistema muestra la marca y no se puede modificar si ya ha entrado en vigencia.
6	El sistema muestra la talla y no se puede modificar si ya ha entrado en vigencia.
7	El sistema muestra el PVP y no se puede modificar si ya ha entrado en vigencia.
8	El sistema muestra el precio oferta al socio y no se puede modificar si ya ha entrado en vigencia.
9	El sistema muestra la fecha de inicio de vigencia y no se puede modificar si ya ha entrado en vigencia.
10	El sistema muestra el stock inicial y no se puede modificar si ya ha entrado en vigencia.
11	El sistema muestra el stock actual y no se puede modificar en ningún momento.
12	El sistema muestra la imagen del material deportivo y la puede modificar en cualquier



	momento.
13	El sistema muestra los detalles del material deportivo y los puede modificar en cualquier momento.
14	El sistema muestra el indicativo de si es o no una novedad y los puede modificar en cualquier momento (por defecto estará marcado que no desea).
15	El administrador solicita al sistema que registre los datos.
16	El sistema registra los datos proporcionados e informa al usuario de que el proceso ha finalizado con éxito.
Excepciones	
Número	Descripción
15	Si el sistema detecta que la fecha de inicio de vigencia es anterior a la fecha actual el sistema informa de la situación al administrador permitiéndole modificar los datos proporcionados y el caso de uso continua.
15	Si el sistema detecta que falta alguno de los datos obligatorios, informa de la situación al administrador permitiéndole modificar los datos proporcionados y el caso de uso continua.
15	Si el administrador solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso se cancela.

#### 2.4.7. CU07 Alta de campañas.

ID	CU07
<b>Identificador del CU</b>	Alta de campañas.
<b>Objetivo del CU</b>	Permitir que el administrador dé de alta en el sistema las campañas de marketing.
<b>Caso de uso relacionado</b>	
<b>Actores</b>	Administrador.
<b>Frecuencia</b>	Trimestral.
<b>Condiciones Previas</b>	El administrador está identificado en el sistema. La campaña no existe en el sistema.
<b>Condiciones Posteriores</b>	El sistema a registrado correctamente la información correspondiente a la nueva <b>campaña</b>
<b>Acción que arranca el CU</b>	Nuevo formulario <b>Alta de campañas</b> donde el administrador introducirá los datos para dar de alta una campaña.
Paso	Descripción
1	Acceder a la nueva pantalla Alta de campañas.
2	Introduce el código de la campaña dato obligatorio
3	Introduce el nombre de la campaña dato obligatorio
4	Introduce la fecha inicio de la campaña dato obligatorio
5	Introduce la fecha fin de la campaña dato obligatorio
6	Introduce el importe acumulado hasta dato obligatorio
7	Introduce el importe a abonar dato obligatorio
8	El administrador repite los pasos 7 y 8 hasta que se indique. Como mínimo ha de existir un rango informado.
9	Introduce observaciones
10	El administrador solicita al sistema que registre los datos
11	El sistema almacena los datos proporcionados e informa al administrador de que el proceso ha finalizado con éxito.
Excepciones	
10	Si el sistema detecta que ya existe una campaña entre las fechas de inicio y fin informa de la situación al administrador permitiéndole modificar los datos proporcionados y el caso de uso continua.
10	Si el sistema detecta que falta alguno de los datos obligatorios, informa de la situación al administrador permitiéndole modificar los datos proporcionados y el caso de uso continua.
10	Si el sistema detecta que la fecha inicio no es estrictamente $\leq$ a la fecha fin o bien que las fechas no son $>$ a la fecha actual informa de la situación al administrador permitiéndole modificar los datos proporcionados y el caso de uso continua.
10	Si el sistema detecta que se solapan los rangos de importe acumulado informa de la situación al administrador permitiéndole modificar los datos proporcionados y el caso de uso continua.
10	Si el administrador solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso se cancela.

## 2.4.8. CU08 Consulta campañas.

ID	CU08
Identificador del CU	Consulta campaña.
Objetivo del CU	Permitir que el administrador pueda consultar las campañas .
Caso de uso relacionado	CU09 Mantenimiento de campañas.
Actores	Administrador
Frecuencia	Diaria
Condiciones Previas	El administrador está identificado.
Condiciones Posteriores	El sistema muestra la información sobre las campañas.
Acción que arranca el CU	Nuevo formulario <b>Consulta campañas</b>
Paso	Descripción
1	Acceder a la nueva pantalla <b>Consulta de campañas</b> .
2	El sistema muestra los datos de las campañas: código, nombre, fecha inicio, fecha fin, fecha baja y observaciones.
3	El sistema espera respuesta del usuario.
Excepciones	
3	Si el administrador solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso se cancela.
Flujo alternativo	
3	Si el usuario clics sobre una campaña, navegará al caso de uso <b>Mantenimiento de campañas</b> . Al volver del mantenimiento de campañas, siempre se refrescará la información visualizada en pantalla y se mantiene en el caso de uso.

## 2.4.9. CU09 Mantenimiento de campañas.

ID	CU09
Identificador del CU	Mantenimiento de campaña.
Objetivo del CU	Permitir que el administrador modificar o dar de baja una campaña.
Caso de uso relacionado	CU08 Consulta campañas
Actores	Administrador.
Frecuencia	Diaria.
Condiciones Previas	El administrador está identificado. La campaña existe en el sistema y no es aún vigente.
Condiciones Posteriores	El sistema registra correctamente la información modificada de una campaña. En el caso de dar de baja una campaña, se registra el nombre del usuario y la fecha en que se dio de baja.
Acción que arranca el CU	Nuevo formulario <b>Consulta de campañas</b> donde el administrador ha seleccionado la campaña a modificar o dar de baja.
Paso	Descripción
1	Acceder a la nueva pantalla Modificación de campaña. En el caso de que la campaña esté dada de baja, no se podrá modificar ningún dato.
2	El sistema muestra el código de la campaña y no lo puede modificar.
3	El sistema muestra la descripción de la campaña y la puede modificar si no está vigente.
4	El sistema muestra la fecha de inicio y la puede modificar si no está vigente.
5	El sistema muestra la fecha fin y la puede modificar si no está vigente.
6	El sistema muestra los rangos de importe acumulado hasta e importe a abonar y los puede modificar si no está vigente.
7	El sistema solicita si desea o no dar de baja la campaña y la pueda modificar en cualquier momento (por defecto estará marcado que no desea darla de baja).
8	El administrador solicita al sistema que registre los datos
9	El sistema registra los datos proporcionados e informa al administrador que el proceso ha finalizado con éxito y el caso de uso se cancela.
Excepciones	
8	Si el sistema detecta que ya existe una campaña entre las fechas de inicio y fin informa de la situación al administrador permitiéndole modificar los datos proporcionados y el caso de uso continúa.
8	Si el sistema detecta que falta alguno de los datos obligatorios, informa de la situación al administrador permitiéndole modificar los datos proporcionados y el caso de uso continúa.
8	Si el sistema detecta que la fecha inicio no es estrictamente $\leq$ a la fecha fin o bien que

	las fechas no son > a la fecha actual informa de la situación al administrador permitiéndole modificar los datos proporcionados y el caso de uso continua.
8	Si el sistema detecta que el rango de importe acumulado hasta no es estrictamente creciente informa de la situación al administrador permitiéndole modificar los datos proporcionados y el caso de uso continua.
8	Si el administrador solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso se cancela.

#### 2.4.10. CU10 Selección de material deportivo.

ID	CU10
<b>Identificador del CU</b>	Selección de material deportivo.
<b>Objetivo del CU</b>	Permitir que el usuario o el socio pueda indicar los criterios de consulta de material deportivo.
<b>Caso de uso relacionado</b>	CU11 Resultado consulta material deportivo.
<b>Actores</b>	Usuario o socio.
<b>Frecuencia</b>	Diaria.
<b>Condiciones Previas</b>	El material deportivo está registrado en el sistema y no ha sido dado de baja.
<b>Condiciones Posteriores</b>	El sistema ha permitido indicar al usuario los criterios de selección para la consulta de material deportivo.
<b>Acción que arranca el CU</b>	Nuevo formulario <b>Selección de material deportivo</b> donde el usuario introducirá los datos para realizar la consulta.
<b>Paso</b>	<b>Descripción</b>
1	El usuario solicita al sistema comenzar el proceso de consulta de material deportivo.
2	El sistema solicita introducir marca
3	El sistema solicita introducir la categoría
4	El sistema solicita introducir la descripción
5	El sistema solicita introducir la talla.
6	El sistema una vez el usuario a introducido información en uno o varios de los diferentes criterios espera confirmación de consulta por parte del usuario.
<b>Excepciones</b>	
6	Si el administrador solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso se cancela.
<b>Flujo alternativo</b>	
6	Si el usuario confirma la consulta el sistema navegará al caso de uso Resultado consulta Material deportivo

#### 2.4.11. CU11 Resultado consulta material deportivo.

ID	CU11
<b>Identificador del CU</b>	Resultado consulta material deportivo.
<b>Objetivo del CU</b>	Permitir que el usuario o el socio pueda visualizar el material deportivo según criterios previamente indicados en el caso de uso <b>Criterios consulta de material deportivo</b> .
<b>Caso de uso relacionado</b>	CU10 Selección de material deportivo
<b>Actores</b>	Usuario o socio.
<b>Frecuencia</b>	Diaria.
<b>Condiciones Previas</b>	El material deportivo está registrado en el sistema y no está dado de baja y la fecha de vigencia es superior o > a la fecha actual.
<b>Condiciones Posteriores</b>	El sistema ha mostrado la información correspondiente, al material deportivo según selección indicada por el usuario en el caso de uso <b>Criterios consulta de material deportivo</b> . El material deportivo también ha podido abrir o ser incluido en un pedido. El pedido quedará en estado pendiente de formalizar.
<b>Acción que arranca el CU</b>	Nuevo formulario <b>Criterios de consulta de material deportivo</b> donde el actor introducirá los datos para realizar la consulta.
<b>Paso</b>	<b>Descripción</b>
1	El sistema muestra todo aquel material deportivo que responde a los criterios indicados por el usuario en el caso de uso Criterios de consulta de material deportivo.
2	El sistema muestra la imagen del material deportivo
3	El sistema muestra la marca
4	El sistema muestra la descripción

5	El sistema muestra la categoría.
6	El sistema muestra la talla.
7	El sistema muestra el PVP.
8	El sistema muestra el precio de oferta al socio
9	El sistema muestra una explicación del material
10	El sistema muestra el stock actual.
11	El sistema muestra el indicativo incluir en el pedido.
12	El sistema espera respuesta del usuario
<b>Excepciones</b>	
11	Si el usuario selecciona esta opción y el stock actual es = 0 el sistema avisa al usuario y se mantiene en el caso de uso.
12	Si el administrador solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso se cancela.
<b>Flujo alternativo</b>	
12	Si el usuario solicita registrar la información, esta se registra y salimos del caso de uso. En el caso de haber marcado algún material deportivo para incluir en el pedido, se decrementará el stock actual y se dará de alta o se incluirá el artículo en el pedido.

#### 2.4.12. CU12 Gestión del pedido en curso.

ID	CU12
<b>Identificador del CU</b>	Gestión del pedido en curso.
<b>Objetivo del CU</b>	Permitir que el usuario o socio pueda visualizar, eliminar y formalizar la información del pedido en curso.
<b>Caso de uso relacionado</b>	CU13 Datos de envío
<b>Actores</b>	Usuario o socio
<b>Frecuencia</b>	Diaria.
<b>Condiciones Previas</b>	Tenemos un pedido en curso.
<b>Condiciones Posteriores</b>	El sistema habrá mostrado la información correspondiente, del pedido en curso.
<b>Acción que arranca el CU</b>	Nuevo formulario <b>Gestión del pedido</b> donde el usuario introducirá los datos para realizar la consulta del <b>pedido en curso</b> .
<b>Paso</b>	<b>Descripción</b>
1	El sistema muestra el contenido actual del pedido.
2	Por cada línea de pedido el sistema muestra la siguiente información.
3	la imagen del material deportivo.
4	la marca
5	la descripción
6	la categoría
7	la talla
8	el PVP.
9	el precio de oferta socio
10	El sistema muestra el indicativo dar de baja del pedido.
11	Muestra total pedido
12	El sistema solicita formalizar el pedido.
<b>Excepciones</b>	
12	Si el administrador solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso se cancela.
<b>Flujo alternativo</b>	
10	El usuario selecciona la baja del sistema elimina la línea de pedido e incrementará el stock actual en una unidad, manteniéndose en el caso de uso.
<b>Flujo alternativo</b>	
12	El sistema pasa al caso de uso <b>datos de envío</b>
12	Si el pedido se ha formalizado se abandona el caso de uso, sino nos mantenemos.
<b>Notas:</b> La oferta de socio se visualizar solo en el caso de que esté identificado como tal en el momento de acceder al caso de	

uso. El total pedido, se calculará calculando con el PVP o bien con el precio de oferta socio en el caso de existir dicha oferta y estar identificado en el sistema como tal.

#### 2.4.13. CU13 Datos de envío.

ID	CU13
<b>Identificador del CU</b>	Datos de envío.
<b>Objetivo del CU</b>	Solicita los datos envío del pedido
<b>Caso de uso relacionado</b>	CU12 Gestión del pedido en curso
<b>Actores</b>	Usuario o socio
<b>Frecuencia</b>	Diaria.
<b>Condiciones Previas.</b>	El usuario está conectado y ha solicitado la formalización de un pedido en curso
<b>Condiciones Posteriores</b>	Las acciones se han realizado correctamente y los datos informados por el usuario han quedado registrados correctamente. El pedido ha quedado en estado pendiente de facturar.
<b>Acción que arranca el CU</b>	Nuevo formulario <b>Datos de envío</b> donde el usuario introducirá los datos para el envío.
Paso	Descripción
1	Usuario accede a la nueva pantalla.
2	Introduce el DNI, obligatorio y modificable
3	Introduce el nombre, obligatorio y modificable
4	Introduce apellidos, obligatorio y modificable
5	Introduce el código postal de envío, obligatorio y modificable.
6	Introduce la provincia de envío es obligatorio y modificable
7	Introduce la población de envío es obligatorio y modificable.
8	Introduce la dirección de envío es obligatorio y modificable.
9	Introduce un teléfono es obligatorio y modificable.
10	Introduce e-mail es obligatorio y modificable.
11	Introduce observaciones
12	Introducir contraseña (solo se visualizará en caso de no ser socio)
13	Repetir contraseña (solo se visualizará en caso de no ser socio)
14	Muestra el total del pedido no modificable. Según se haya identificado o no como socio, el total lo obtiene del PVP o bien del precio oferta socio.
15	Muestra, si hay, el importe a descontar al socio por campaña. No modificable
16	En caso de existir importe a descontar, Solicita importe a descontar (no puede ser superior al total pedido).
17	Muestra el importe neto a liquidar no modificable (17 = 14 - 16)
18	Fecha de entrega del pedido es obligatorio.
19	Hora de entrega del pedido es obligatorio.
20	El sistema solicita al usuario que registre los datos
21	El sistema registra la información.
Excepciones	
2	Si el usuario se ha identificado como socio, el sistema recupera los datos y los muestra para su validación y/o modificación.
12	Si el usuario introduce contraseña, cuando solicite el registro de los datos quedará registrado como socio.
21	El DNI no es correcto. El sistema muestra el error y rechaza la entrada, el caso de uso continua.
21	La dirección de envío no es correcta. El sistema muestra el error y rechaza la entrada, el caso continua.
21	Si el sistema detecta que falta alguno de los datos solicitados como obligatorios informa de la situación al usuario permitiéndole modificar los datos proporcionados y el caso de uso continua.
21	Si el sistema detecta que ya existe un usuario registrado con el mismo nombre muestra el error y rechaza la entrada, el caso continua.
21	Si se solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso se cancela.
<b>Notas:</b> El importe a descontar y el importe neto a liquidar se puede actualizar en el formulario en el caso que el usuario introduzca los datos para darse de alta como socio. El importe a descontar siempre se calculará teniendo en	

cuenta el importe total bruto de todos los pedidos formalizados siendo socio y correspondientes a una campaña y en base a esto se localiza el tramo correspondiente para identificar el importe a abonar y a éste último se le descuenta el importe abonado hasta la fecha.

#### 2.4.14. CU14 Consulta pedidos formalizados.

ID	CU14
<b>Identificador del CU</b>	Consulta pedidos formalizados.
<b>Objetivo del CU</b>	Permitir que el administrador pueda visualizar los pedidos pendientes de facturar.
<b>Caso de uso relacionado</b>	CU15 Facturación pedido.
<b>Actores</b>	Administrador.
<b>Frecuencia</b>	Diaria.
<b>Condiciones Previas</b>	Tenemos pedidos formalizados pendientes de facturar.
<b>Condiciones Posteriores</b>	El sistema habrá mostrado la información correspondiente, de los pedidos formalizados pendientes de facturar. Y el estado del pedido pasará a facturado.
<b>Acción que arranca el CU</b>	Nuevo formulario <b>Consulta pedidos formalizados</b> donde el administrador introducirá los datos para realizar la consulta de los <b>pedidos formalizados</b> .
<b>Paso</b>	<b>Descripción</b>
1	El sistema muestra todos los pedidos pendientes de facturar por orden de menor a mayor fecha de formalización.
2	Por cada pedido el sistema muestra una línea con la siguiente información.
3	DNI cliente
4	Nombre y apellidos
5	Dirección de envío completa
6	Fecha y hora de entrega.
7	Total importe neto pedido.
8	Seleccione pedido a facturar
9	El sistema solicita confirmación facturación
<b>Excepciones</b>	
9	Si el administrador solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso se cancela.
<b>Flujo alternativo</b>	
9	El usuario selecciona confirmar facturación, accederá al caso de uso <b>facturación pedido</b> , refrescando la información a la salida del caso de uso anterior y manteniéndose en el caso de uso actual.

#### 2.4.15. CU15 Facturación pedido.

ID	CU15
<b>Identificador del CU</b>	Facturación del pedido
<b>Objetivo del CU</b>	Permitir que el administrador facture el pedido para su entrega al cliente.
<b>Caso de uso relacionado</b>	CU14 Consulta pedidos pendientes de facturar
<b>Actores</b>	Administrador.
<b>Frecuencia</b>	Diaria.
<b>Condiciones Previas</b>	Tenemos un pedido formalización y pendiente de facturar.
<b>Condiciones Posteriores</b>	El sistema habrá facturado el pedido.
<b>Acción que arranca el CU</b>	Nuevo formulario <b>Facturación del pedido</b> donde el usuario podrá visualizar los datos del pedido y su detalle para su facturación.
<b>Paso</b>	<b>Descripción</b>
1	El sistema muestra el contenido actual del pedido.
2	DNI cliente
3	Nombre y apellidos
4	Dirección de envío
5	la imagen del material deportivo.
6	la marca
7	la descripción
8	la categoría
9	la talla

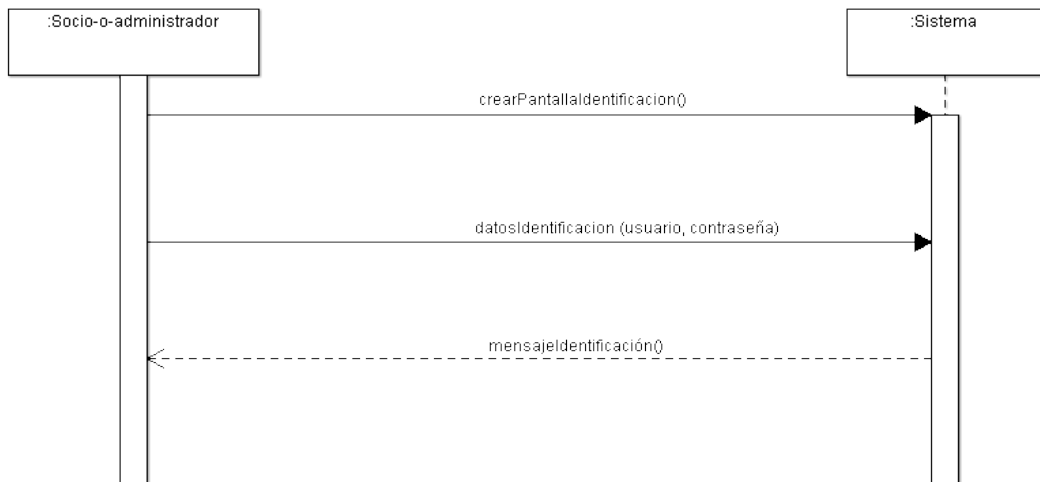
10	el PVP.
11	el precio de oferta.
12	Muestra el total del pedido
13	Muestra, si hay, el importe a descontar por campaña indicado por socio
14	Muestra el total del pedido liquidar una vez aplicando el descuento anterior
15	Fecha de entrega del pedido.
16	Hora de entrega del pedido.
17	El sistema solicita confirmación facturación.
<b>Excepciones</b>	
17	Si el administrador solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso se cancela.
<b>Flujo alternativo</b>	
17	Se emite la factura del pedido. El pedido queda registrado como facturado. El caso de uso se cancela.

## 2.5. Diagrama de secuencia del sistema.

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.

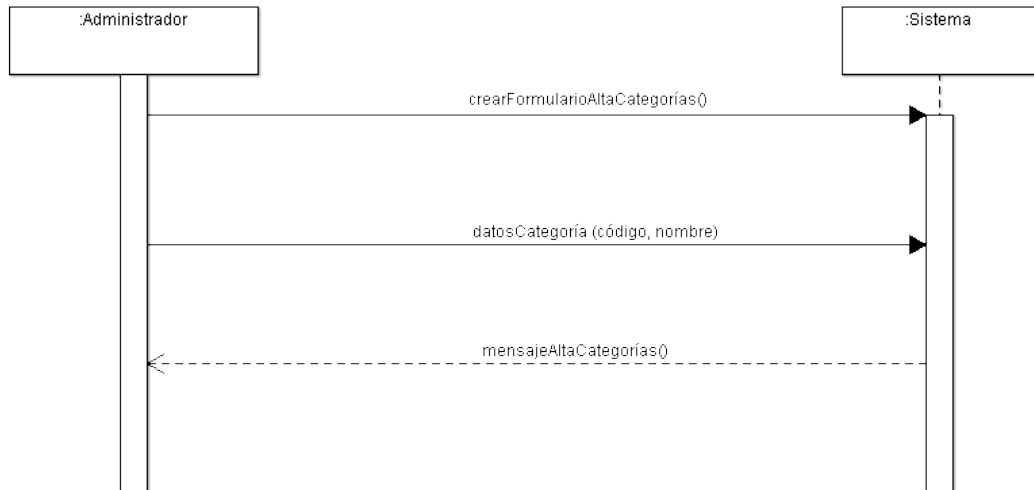
### 2.5.1. Identificación.

#### CU01 Identificación



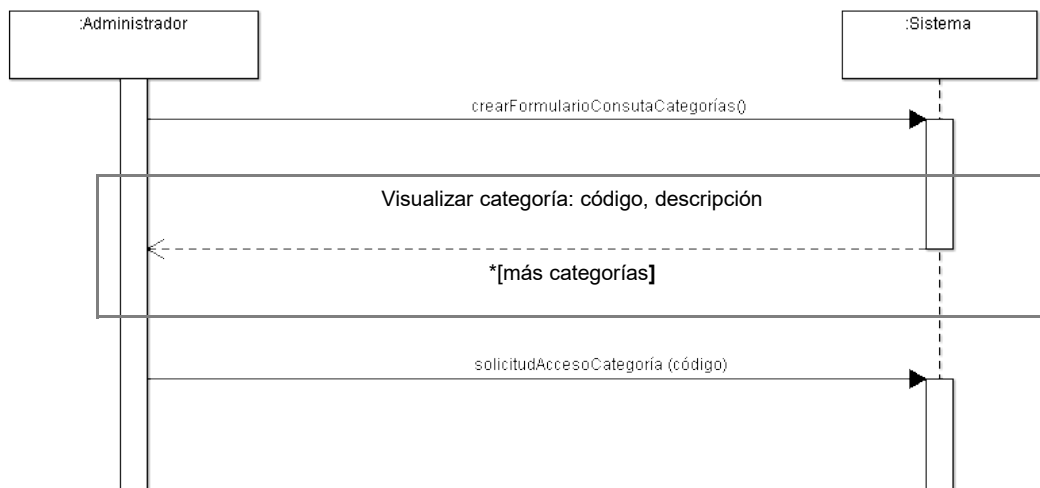
### 2.5.2. Alta de categorías.

#### CU02 Alta Categorías



### 2.5.3. Consulta de categorías.

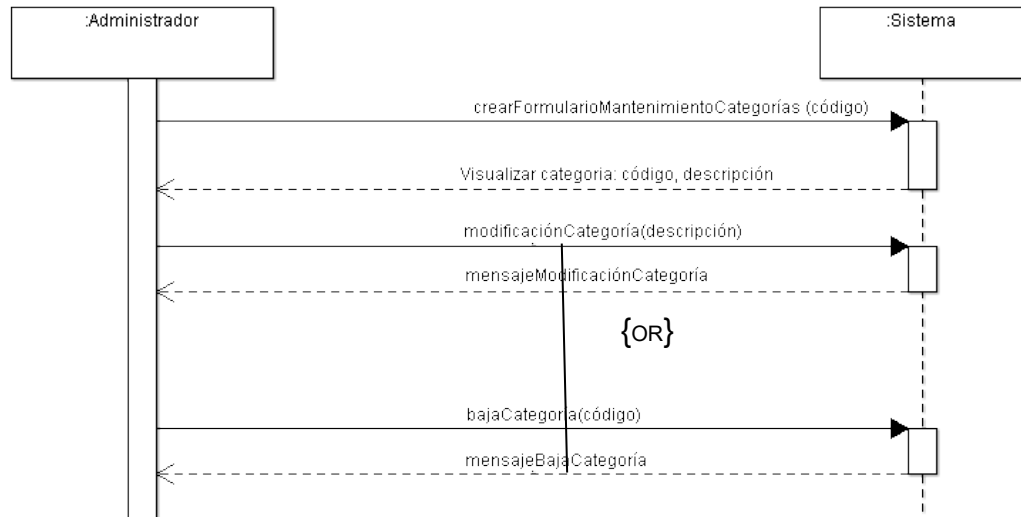
#### CU03 Consulta categorías





2.5.4. Mantenimiento de categorías.

CU04 Mantenimiento categorías



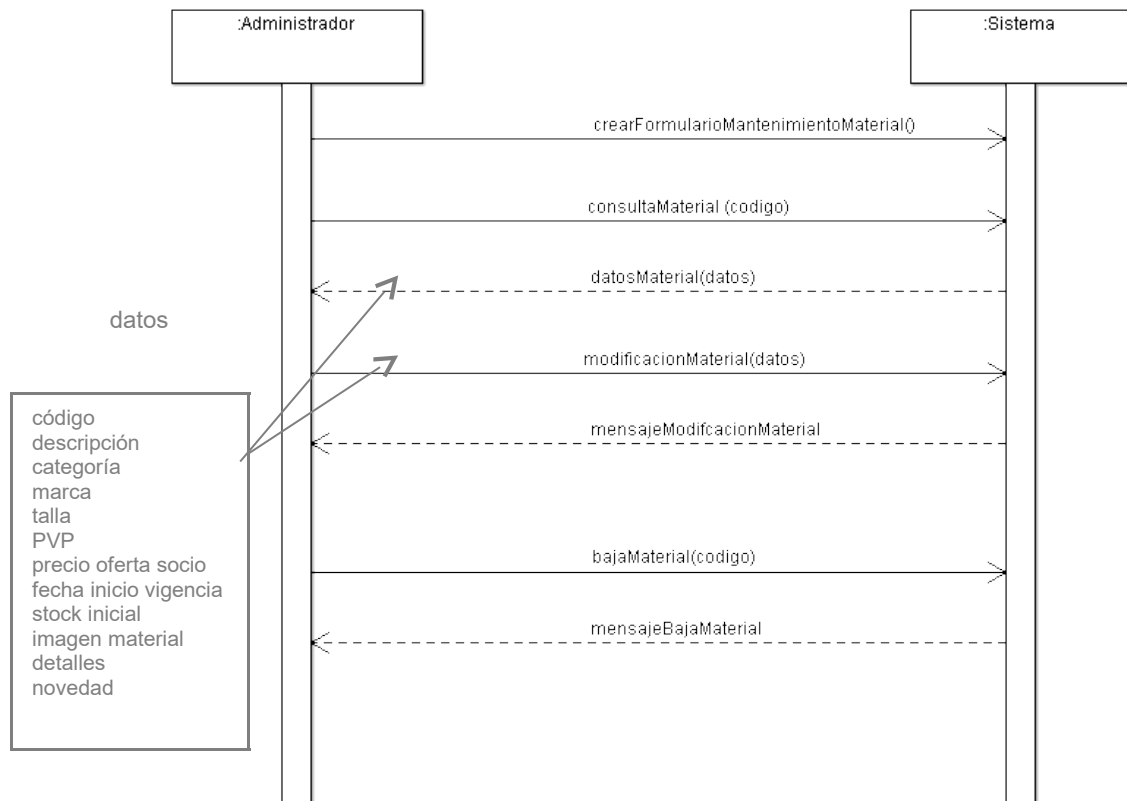
2.5.5. Alta material deportivo.

CU05 Alta material deportivo



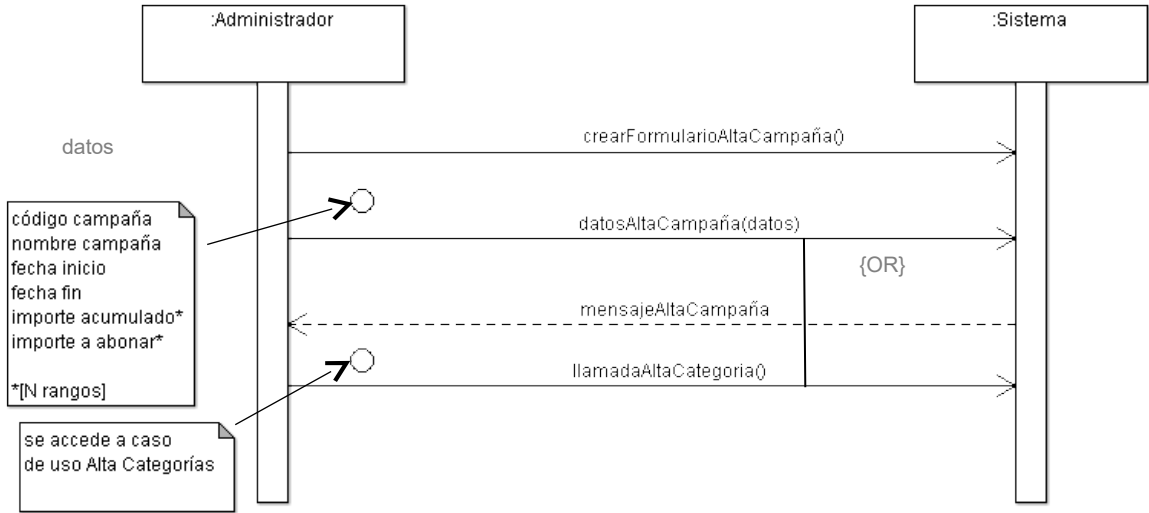
## 2.5.6. Mantenimiento material deportivo.

## CU06 Mantenimiento material deportivo



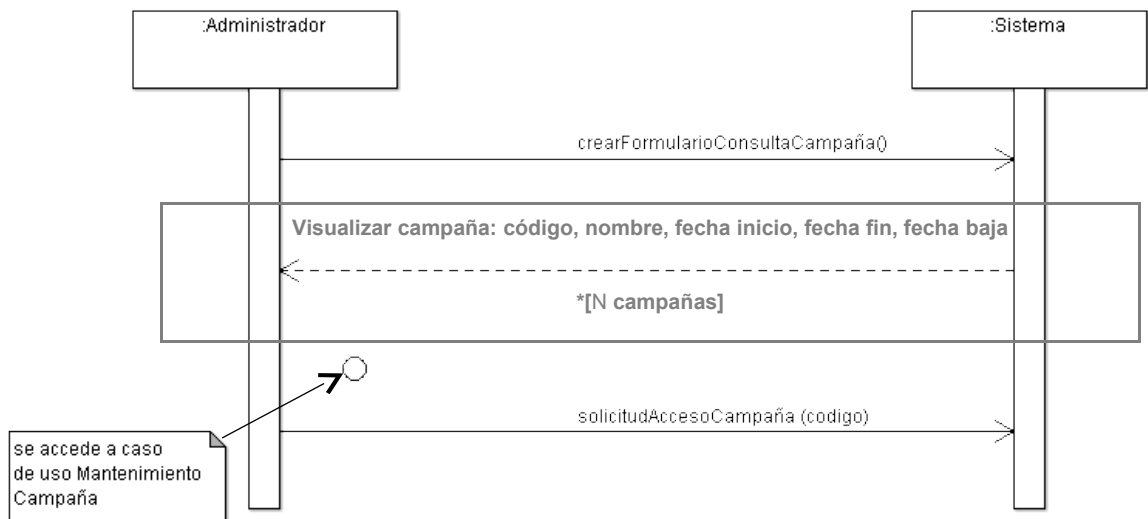
2.5.7. Alta Campaña.

CU07 Alta campaña



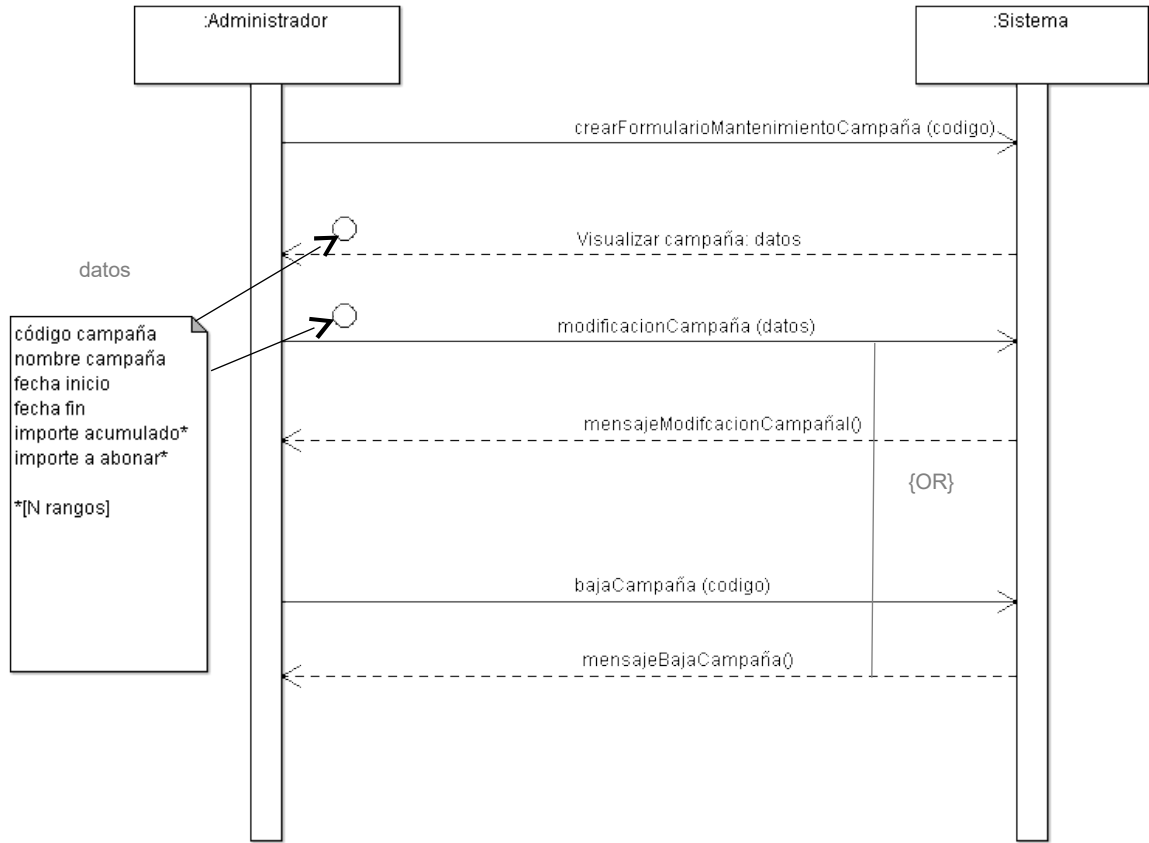
2.5.8. Consulta Campañas.

CU08 Consulta Campañas



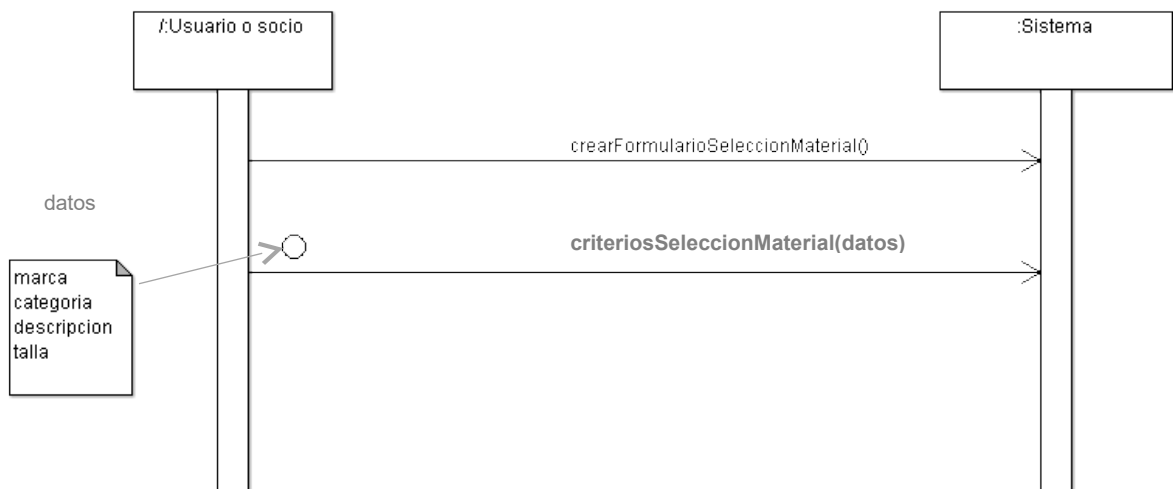
2.5.9. Mantenimiento Campañas.

CU09 Mantenimiento Campañas



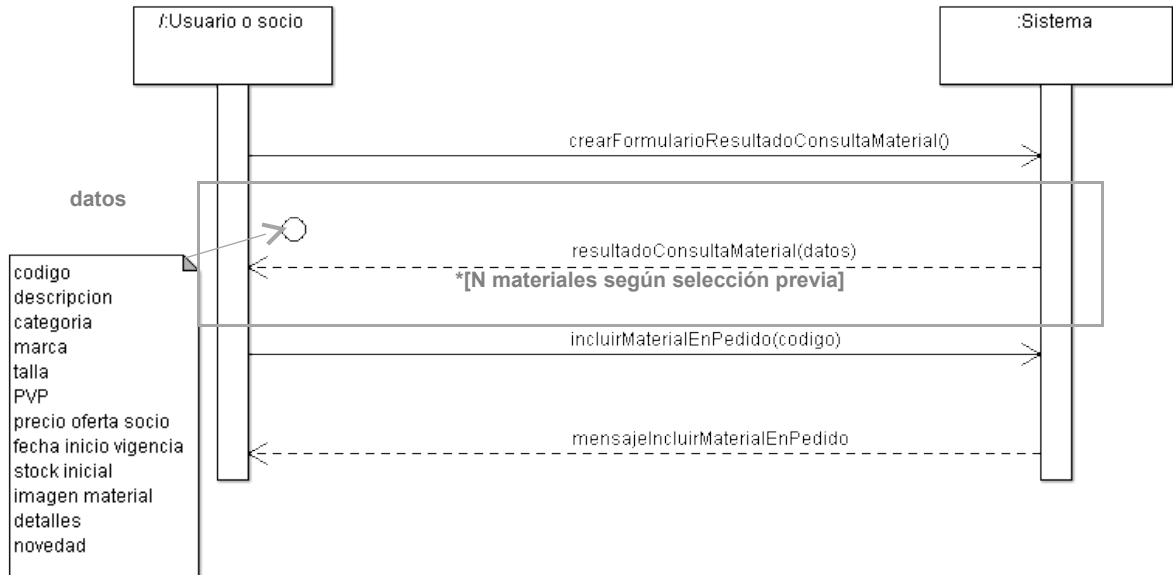
2.5.10. Selección de material deportivo.

CU010 Selección material deportivo



2.5.11. Resultado Consulta material deportivo.

CU011 Resultado consulta material deportivo



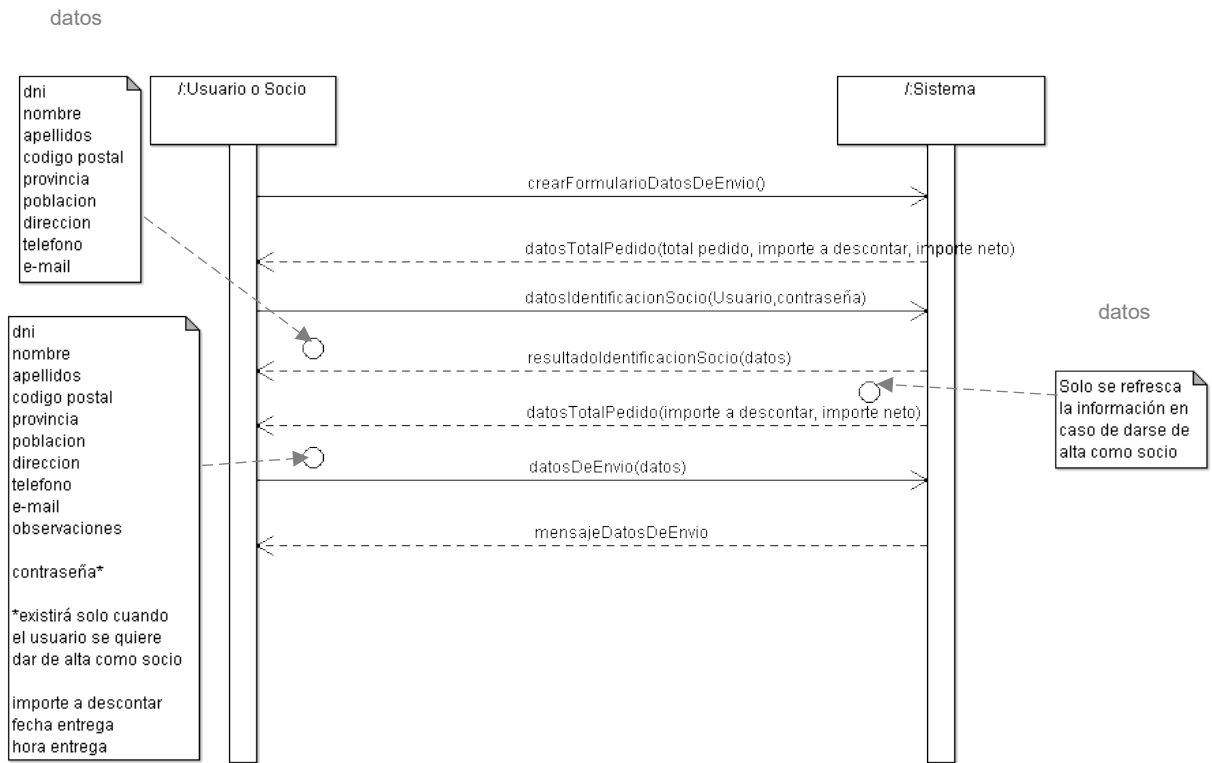
2.5.12. Gestión del Pedido en Curso.

CU012 Gestión del pedido en curso



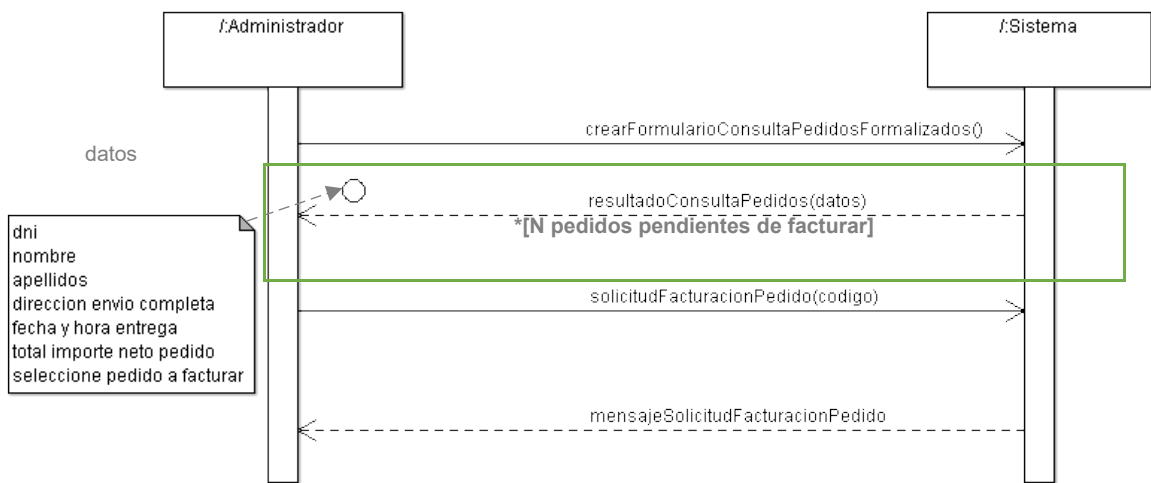
2.5.13. Datos de Envío.

CU013 Datos de envío.



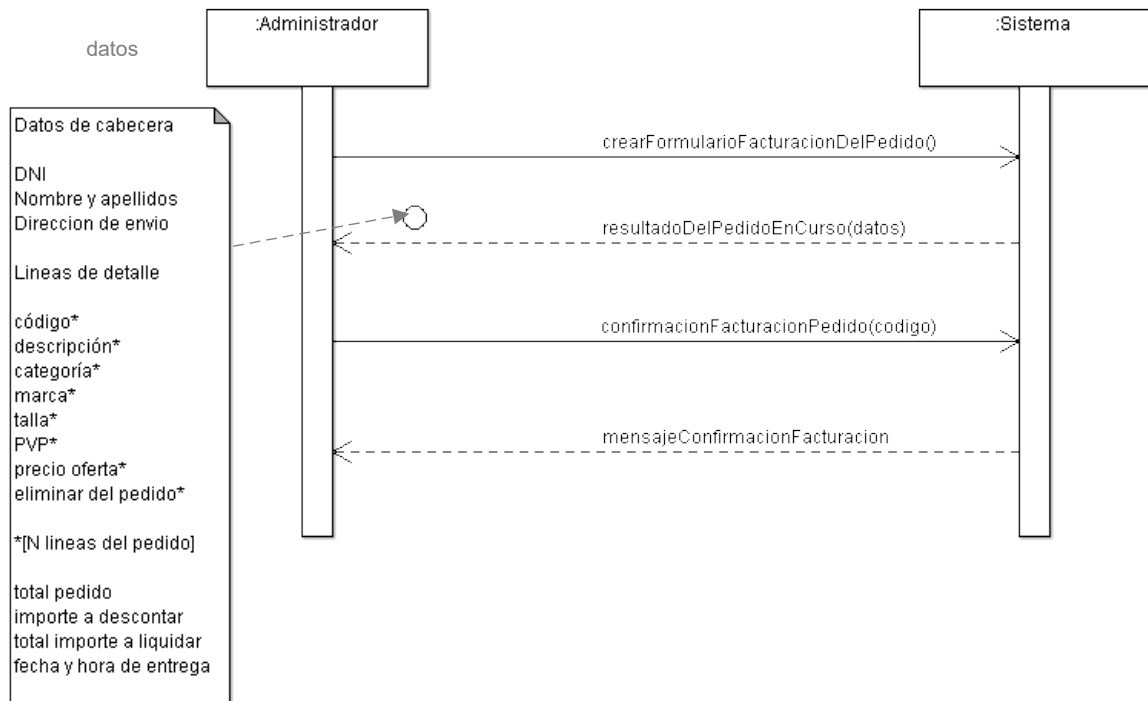
2.5.14. Consulta pedidos formalizados.

CU014 Consulta pedidos formalizados



## 2.5.15. Facturación del Pedido.

## CU015 Facturación del pedido



### 2.6. 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 los diagramas de casos de uso identificados:

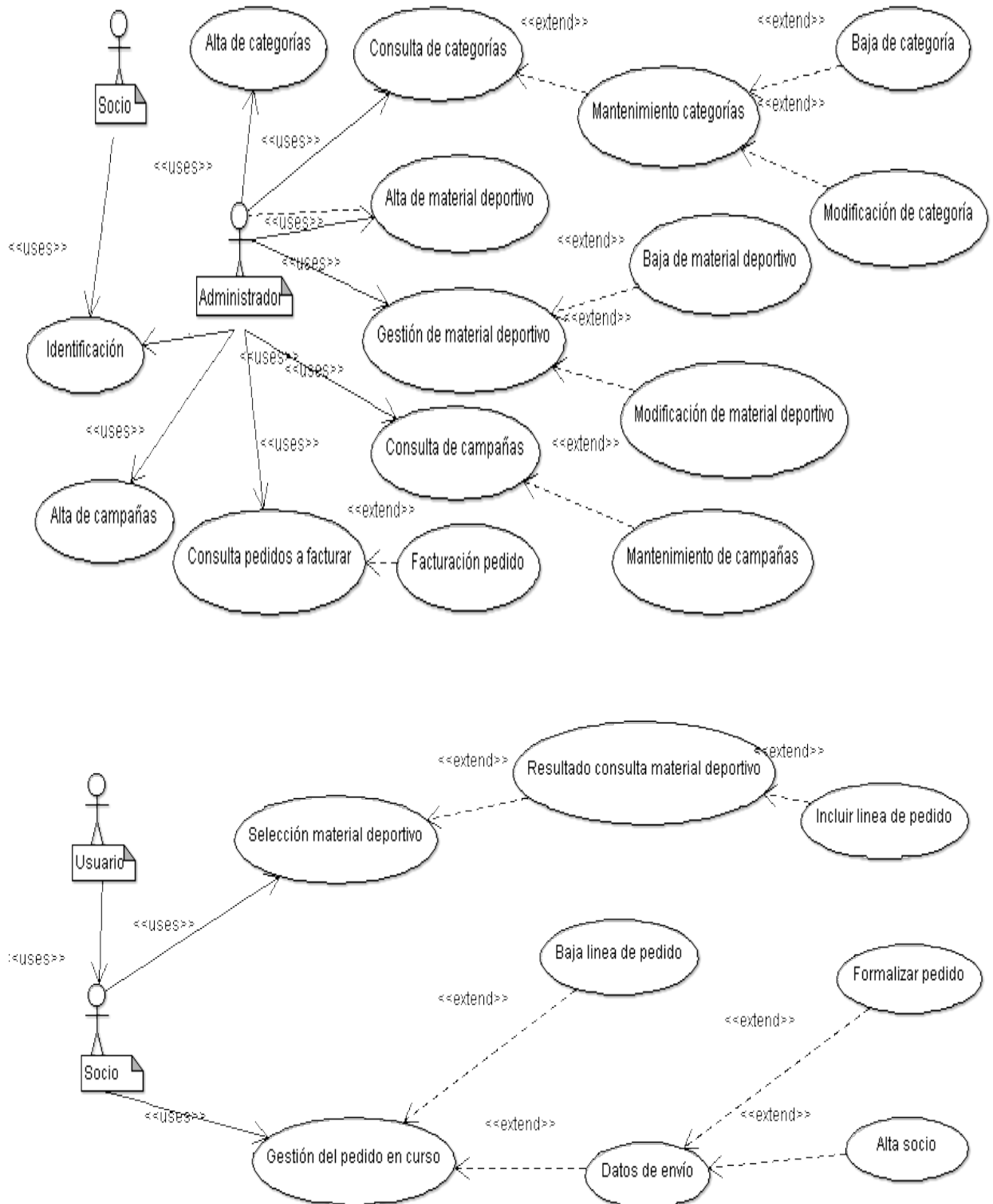


Figura 8. Diagrama de casos de uso y actores



## 2.7. Especificación complementaria o requisitos no funcionales.

Las especificaciones complementarias o requisitos no funcionales en los sistemas hacen sobre todo referencia a los requisitos y atributos de calidad. Dentro de los diferentes sistemas de clasificación de estos requisitos, a los que pertenecería este apartado, se ha seleccionado seguir el modelo **FURPS+** (Functionality, Usability, Reliability, Performance, Supportability model), por ser el criterio de clasificación utilizado en UP.

Para este proyecto los requisitos de calidad se han ido identificando durante la toma de requerimientos con los usuarios interesados en el proyecto. No se trata tanto de requerimientos funcionales, sino de requisitos de calidad o aspectos incluso técnicos que afectan en general a la totalidad del sistema y que se han de tener en cuenta a la hora de diseñar, implementar o bien en el diseño de la infraestructura que tendrá que soportarlo.

**Funcionalidad:** Se solicita que el sistema se construya de forma que pueda ser mañana fácilmente ampliable y reutilizable.

**Usabilidad:** se solicita facilidad de uso en el diseño de la web y que esté acorde con los standards de la **W3C** -Word Wide Web Consortium-.

**Fiabilidad:** Se solicita también que el sistema tenga una alta fiabilidad en cuanto a tolerancia a fallos.

**Rendimiento:** Se solicita que el tiempo de respuesta al usuario no supere los 5' en ningún caso de uso.

**Soporte:** Se solicita que el software que se utilice sea un standar aceptado en el mercado de forma que en el futuro no tengan problemas para la localización de técnicos para su mantenimiento.

**Restricción en el software:** Se solicita que en la medida de lo posible el software sea **open source** debido principalmente a las limitaciones de presupuesto.

Como consecuencia de lo anterior se tendrán que considerar los siguientes aspectos:

**1.Herramientas de software:** Se deberán seleccionar herramientas de desarrollo que cumplan tanto con los criterios de soporte como de restricción en el software.

**2.Hardware:** Se deberán considerar un sistema redundante para dar soporte a la fiabilidad solicitada.

**3.Desarrollo del software e implementación:** Se tendrán que considerar desde un punto de GUI's el requerimiento indicado en Usabilidad.

**4.Arquitectura del software e implementación:** Se tendrá que considerar para dar soporte al apartado de funcionalidad y rendimiento.

### 3. Diseño.

Durante esta fase se trata de desarrollar una solución lógica basada en el paradigma orientado a objetos.

Lo primero será definir la arquitectura general del sistema. Por otro lado realizaremos la creación de los diagramas de interacción que representan el modo en el que los objetos colaboran para satisfacer los requisitos. Posteriormente, aunque en realidad se realiza en paralelo, se representarán los diagramas de clase.

Tanto los diagramas de interacción como los diagramas de clases son artefactos que forman parte del Modelo de diseño. Durante la creación de los diagramas de interacción y clases es donde se aplicará patrones de diseño.

Posteriormente, se realizará el diseño de las interfaces de usuario, diseño de la persistencia mediante la aplicación del modelo Entidad-Relación.

Por último se incluirá el diseño de unos de los posibles tipos de test que se pueden realizar durante el proceso de desarrollo de software. En nuestro caso, nos basaremos en el test de caja negra orientado a la validación de los casos de usos por parte del usuario, es decir, el **UAT**, el cual se explica con mas detalle mas adelante.

#### 3.1. Análisis y diseño de la arquitectura.

El objetivo de este apartado es dar una visión de general de como se ha de contruir el sistema. Es una idea de altos vuelos que condicionan todo el diseño y posterior desarrollo del sistema. En UP las decisiones de diseño arquitectónico se recogen en el **SAD** (Documento de la Arquitectura del Software). En este documento la arquitectura se describe mediante el concepto de **vistas**, siendo éstas especializaciones de la información que destacan una perspectiva particular. En base a esto se organiza la información de la arquitectura en seis vistas: **Lógica, Proceso, Despliegue, Datos, Casos de uso e Implementación**. De todas las vistas anteriores nosotros nos centraremos principalmente en la vista Lógica o Arquitectura del software.

La arquitectura de software podemos definirla como el conjunto de decisiones significativas sobre la organización del sistema software, la selección de los elementos estructurales y sus interfaces, con los que se compone el sistema, junto con su comportamiento tal como se especifica en las colaboraciones entre estos elementos. Independientemente de la definición, lo mas importante es que la arquitectura de software tiene que ver con la gran escala.

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 dá 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.

Bajo esta idea en este apartado nos referiremos a los patrones de arquitectura y la aplicación de los patrones de diseño para realizar las conexiones entre las estructuras a gran escala.

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.

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 caso seleccionaremos 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**. Este patrón se aplicará a cada pantalla de la aplicación, a la que se le asociará un controlador que se encargará de gestionar todas las peticiones realizadas desde la pantalla por el usuario. Este controlador en función del tipo de petición decidirá realizar las llamadas correspondientes a la capa de dominio y/o una llamada a la vista que se encargará de mostrar el resultado en la pantalla al usuario. Por tanto, por cada clase frontera tenemos una vista y un controlador.

Como reglas prácticas a seguir para preservar este patrón podemos señalar tres:

Los actores solo interactuan con la clase frontera.

Las clases frontera solo interactuan con los actores y las clases de control.

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.

Resumiendo esta capa la formarán:

a) El conjunto de clases denominadas controladores que implementarán las operaciones que la capa de presentación necesita. Estas operaciones quedarán definidas en términos de la capa de dominio y serán independientes de cualquier tecnología utilizada en la capa de presentación.

b) Conjunto de clases de software que representarán las clases conceptuales detectadas durante el análisis y que implementarán toda la lógica del sistema.

**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).

### 3.2. Diagramas de interacción.

Antes de entrar en los diagramas de interacción, me gustaría comentar de forma resumida los principios de diseño más relevantes que he intentado basarme a la hora de realizarlos, principios que también se han considerado de cara al diseño de las clases.

#### 3.2.1. Responsabilidades en UML.

UML define una responsabilidad como "un contrato u obligación de un clasificador". Las responsabilidades están relacionadas con las obligaciones de un objeto en cuanto a su comportamiento. Básicamente estas responsabilidades son de los siguientes dos tipos:

**Hacer:** dentro de las responsabilidades de hacer de un objeto se encuentran:

Hacer algo él mismo, por ejemplo, crear un objeto o hacer un cálculo.  
Iniciar una acción en otros objetos.  
Controlar y coordinar actividades en otros objetos.

**Conocer:** dentro de las responsabilidades de conocer de un objeto se encuentran:

Conocer los datos privados encapsulados.  
Conocer los objetos relacionados.  
Conocer las cosas que puede derivar o calcular.

#### 3.2.2. Responsabilidades y objetos.

Las responsabilidades se asignan a las clases de los objetos durante el diseño de objetos. Las responsabilidades relevantes relacionadas con "conocer" a menudo se pueden inferir a partir del modelo del dominio, debido a los atributos y asociaciones que describe.

Por otro lado, las responsabilidades se implementan utilizando métodos que o actúan solos o colaboran con otros métodos u objetos.

Teniendo en cuenta los dos aspectos anteriores podemos decir que las responsabilidades influyen en la conversión de éstas en clases y métodos.

#### 3.2.3. Responsabilidades y diagramas de interacción.

En los artefactos UML, un contexto habitual donde se tiene en cuenta las responsabilidades (implementadas como métodos) es durante la creación de los diagramas de interacción (que forman parte del Modelo de Diseño del UP). Cuando estamos realizando diagramas de interacción en parte estamos eligiendo también asignación de responsabilidades a los objetos, y esto lo reflejamos mediante los mensajes que se envían a diferentes clases de objetos.

Teniendo en cuenta lo anterior para guiar las elecciones sobre donde asignar responsabilidades nos basaremos en los patrones **GRASP**.

### 3.2.4. Patrones GRASP.

**GRASP** es el acrónimo de **General Responsibility Assignment Software Patterns** (patrones generales de software para asignar responsabilidades), y lo componen varios patrones, en nuestro caso sobre todo utilizaremos en el diseño los siguientes cinco patrones:

**Experto en información**

**Creador**

**Bajo Acoplamiento**

**Alta Cohesión**

**Controlador**

**3.2.4.1. Experto:** Guía la idea de a que objetos hay que asignar responsabilidades. La respuesta sería al experto en información, es decir, a la clase que tiene la información necesaria para realizar la responsabilidad.

En realidad se trata de un principio de guía básico que se ha de utilizar siempre en el diseño de objetos.

**3.2.4.2. Creador:** Guía la idea de que objeto sería el responsable de la creación de una instancia de alguna clase. La respuesta sería asignar a la clase B la responsabilidad de crear una instancia de la clase A si se cumple uno o más de los casos siguientes:

B agrega objetos de A

B contiene objetos de A

B registra instancias de objetos de A

B utiliza más estrechamente objetos de A

B tiene los datos de inicialización que se pasarán a un objeto de A cuando sea creado.

Si se puede aplicar más de una opción, nos inclinaremos por una clase B que agregue o contenga la clase A.

**3.2.4.3. Bajo acoplamiento:** Guía la idea de cómo soportar bajas dependencias, bajo impacto del cambio e incrementar la reutilización. La respuesta sería asignar una responsabilidad de manera que el acoplamiento permanezca bajo.

El acoplamiento es una medida de la fuerza con que un elemento está conectado o tiene conocimiento de, confía en, otros elementos. Un elemento con bajo acoplamiento no depende de demasiadas otras clases.

**3.2.4.4. Alta cohesión:** Guía la idea de como mantener una complejidad manejable. La respuesta sería asignar una responsabilidad de manera que la cohesión permanezca alta.

La cohesión es una medida de la fuerza con la que se relacionan y del grado de focalización de las responsabilidades de un elemento. Un elemento con responsabilidades altamente relacionadas y que no hace una gran cantidad de trabajo, tiene alta cohesión. Por el contrario, una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo, pues resultan difíciles de entender, de reutilizar, mantener y se ven afectadas constantemente por los cambios.

**3.2.4.5. Controlador:** Guía la idea de quien debe ser el responsable de gestionar un evento de entrada al sistema. La respuesta sería asignar la responsabilidad de recibir o manejar un mensaje de evento al sistema a una clase que representa una de las siguientes opciones:

Representa el sistema global, dispositivo o subsistema (controlador de fachada).

Representa un escenario de caso de uso en el que tiene lugar el evento del sistema a menudo denominado controlador de sesión o de caso de uso. En este caso, utiliza la misma clase controlador para todos los eventos del sistema en el mismo escenario de caso de uso. En este trabajo se ha optado por aplicar este criterio.

Sesión se ha de entender como una instancia de una conversación con un actor. Las sesiones pueden tener cualquier duración, pero se organizan a menudo en función de los casos de uso (sesiones de casos de uso).

Se ha de tener en cuenta que el controlador es un objeto que no pertenece a la interfaz de usuario, responsable de recibir o manejar un evento del sistema. Generalmente un controlador delegará en otros objetos el trabajo que se necesita hacer, y también realiza labores de coordinación.

En el UP existen los conceptos de clases frontera, control y entidad. Los objetos frontera son abstracciones de las interfaces. Los objetos entidad son los objetos de software del dominio independiente de la aplicación y normalmente persistentes. Los objetos control son los manejadores de los casos de uso tal y como se describen en el patrón controlador.

Resumiendo el patrón controlador nos indica que los objetos interfaz (ventanas, elementos gráficos, etc..) y la capa de presentación no deberían ser responsables de llevar a cabo los eventos del sistema, es decir, las operaciones del sistema se deberían manejar en la lógica de la aplicación o capas del dominio en lugar de en la capa de interfaz del sistema.

### 3.3. Diagramas de secuencia (diseño).

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.

Antes de entrar en el diseño específico de los diagramas de secuencia conviene aclarar que para el diseño de estos se ha tenido en cuenta los principios de diseño anteriormente comentados.

Por último, comentar que por cada diagrama de secuencia, se indicará de forma explícita el rol de cada uno de ellos, teniendo en cuenta el patrón de arquitectura de software: modelo, vista, controlador, donde:

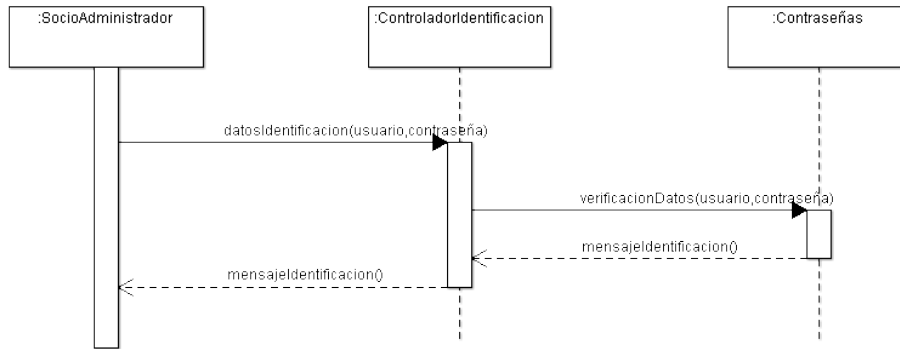
**Modelo:** Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.

**Controlador:** Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario (middleware) entre la 'vista' y el 'modelo'.

**Vista:** Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.

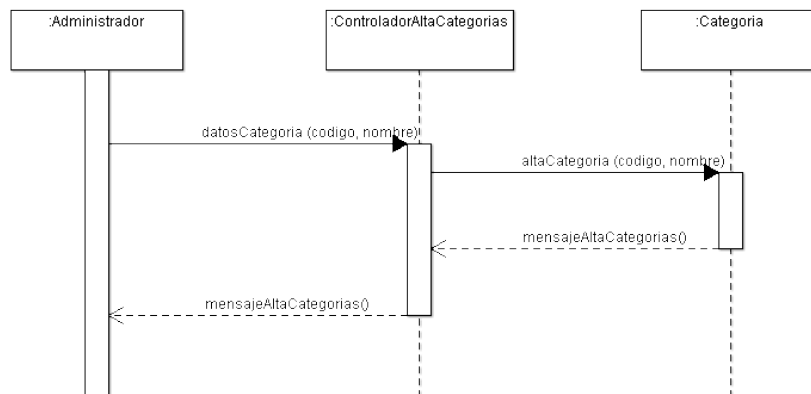
### 3.3.1. CU01 Identificación.

**Vista:** SocioAdministrador.  
**Controlador:** ControladorIdentificacion.  
**Modelo:** Contraseñas.



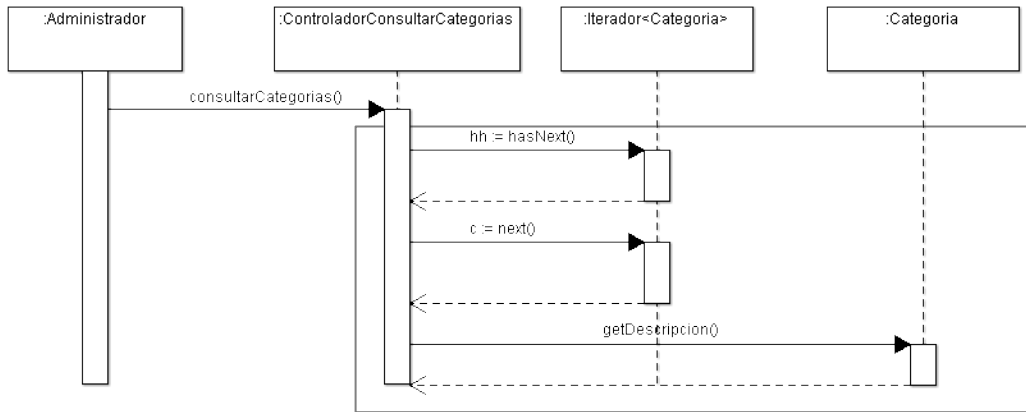
### 3.3.2. CU02 Alta de categorías.

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



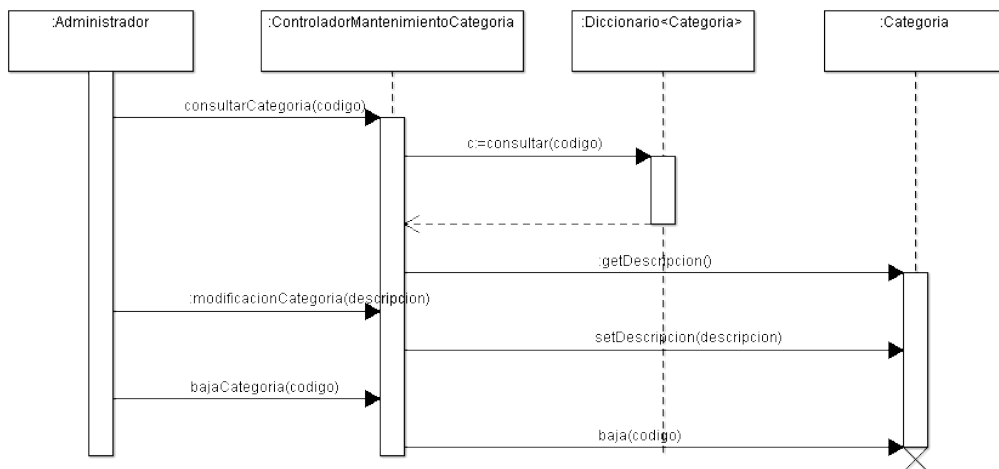
### 3.3.3. CU03 Consulta de categorías.

**Vista:** Administrador.  
**Controlador:** ControladorConsultarCategorias.  
**Modelo:** Iterador, Categoría.



### 3.3.4. CU04 Mantenimiento de categorías

**Vista:** Administrador.  
**Controlador:** ControladorMantenimientoCategoria.  
**Modelo:** Dicionario, Categoría.



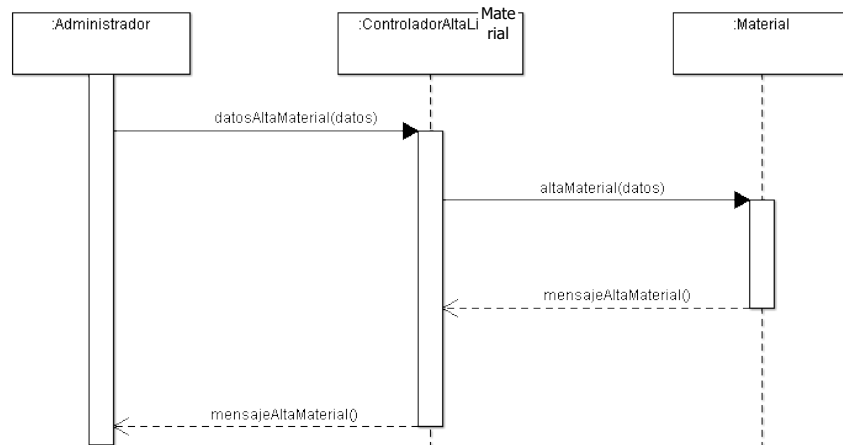


### 3.3.5. CU05 Alta de material deportivo.

**Vista:** Administrador.

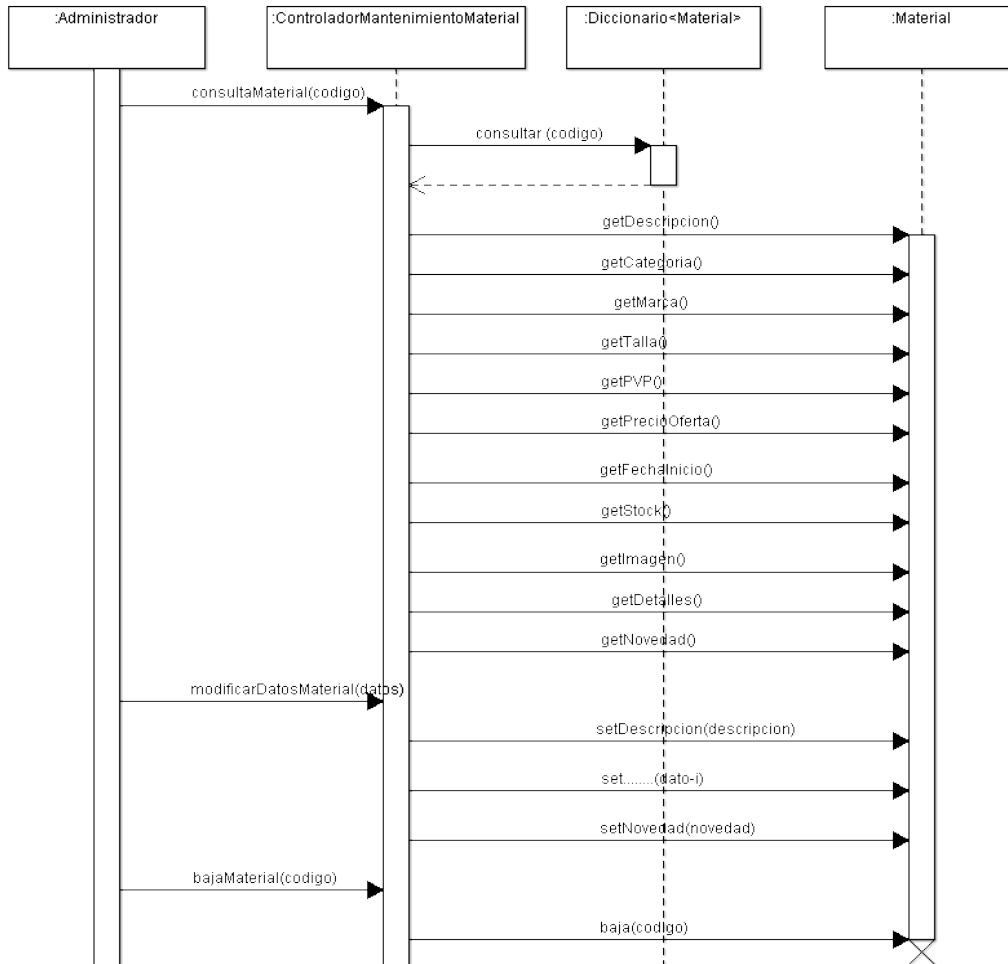
**Controlador:** ControladorAltaMaterial.

**Modelo:** Material.



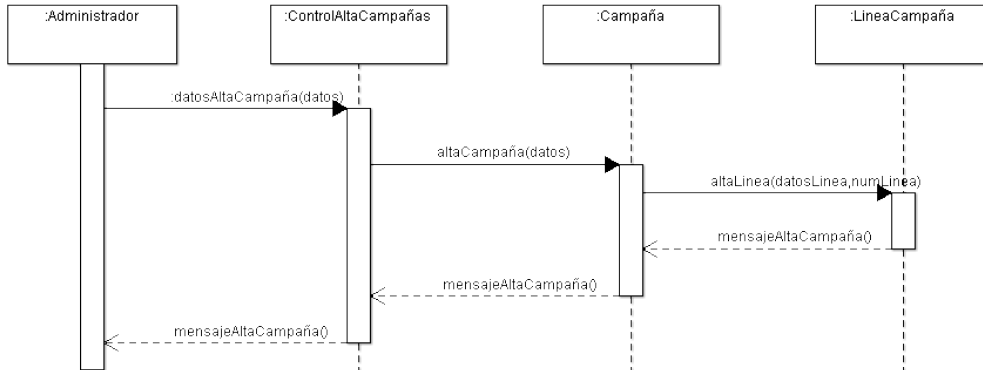
3.3.6. CU06 Mantenimiento de material deportivo.

**Vista:** Administrador.  
**Controlador:** ControladorMantenimientoMaterial.  
**Modelo:** Diccionario<Material>, Material.



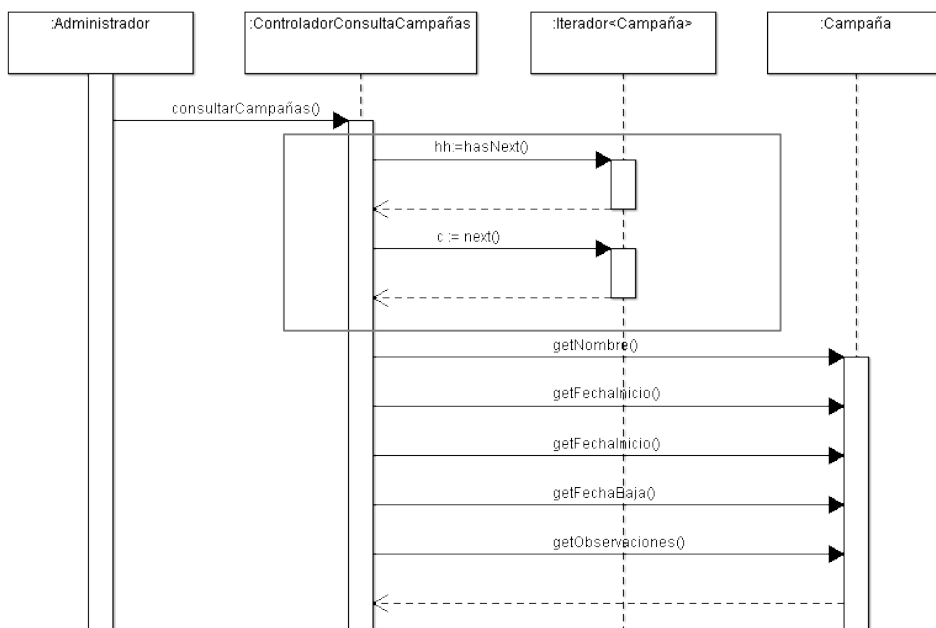
### 3.3.7. CU07 Alta de campañas.

**Vista:** Administrador.  
**Controlador:** ControladorAltaCampañas.  
**Modelo:** Campaña, LineaCampaña.



### 3.3.8. CU08 Consulta campañas.

**Vista:** Administrador.  
**Controlador:** ControladorConsultaCampañas.  
**Modelo:** Iterador<Campaña>, Campaña.

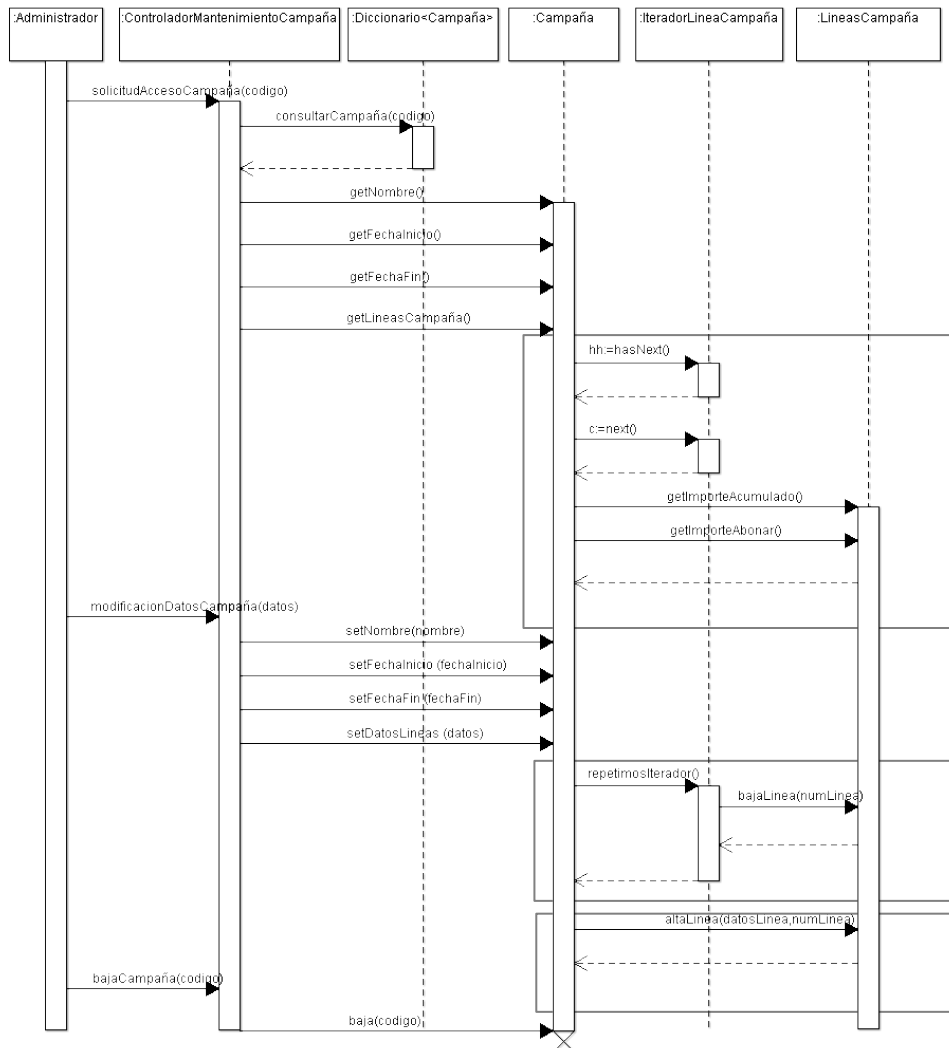


### 3.3.9. CU09 Mantenimiento de campañas.

**Vista:** Administrador.

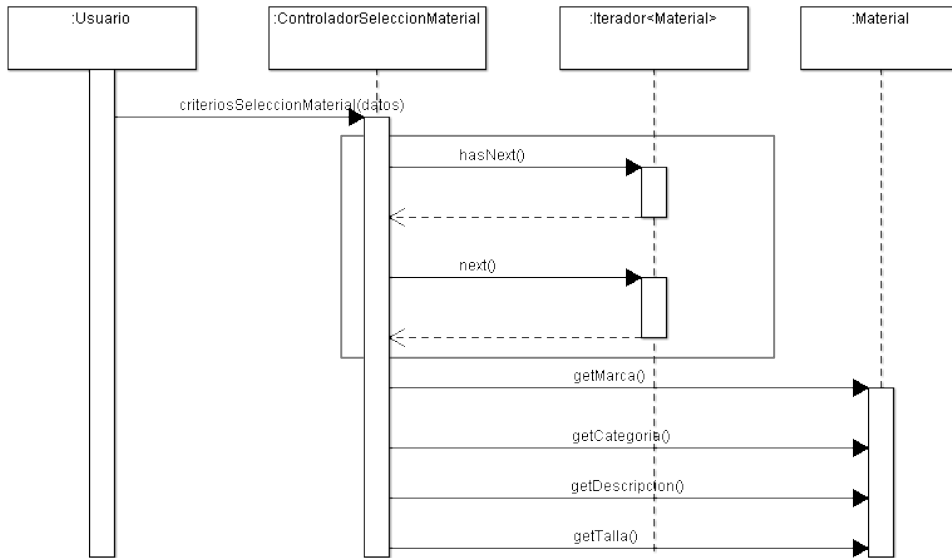
**Controlador:** ControladorMantenimientoCampaña.

**Modelo:** Diccionario<Campaña>, Campaña, IteradorLineaCampaña, LineasCampaña.



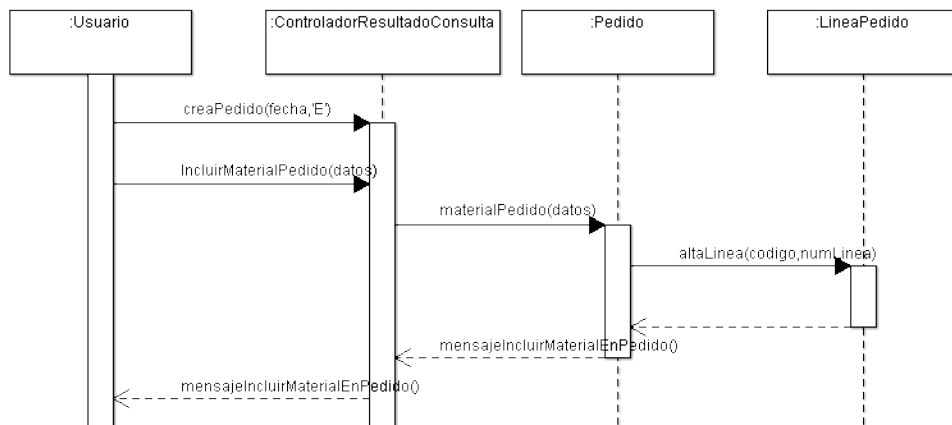
### 3.3.10. CU10 Selección de material deportivo.

**Vista:** Usuario.  
**Controlador:** ControladorSeleccionMaterial.  
**Modelo:** Iterador<Material>, Material.



### 3.3.11. CU11 Resultado consulta material deportivo.

**Vista:** Usuario.  
**Controlador:** ControladorResultadoConsulta.  
**Modelo:** Pedido, LineaPedido.

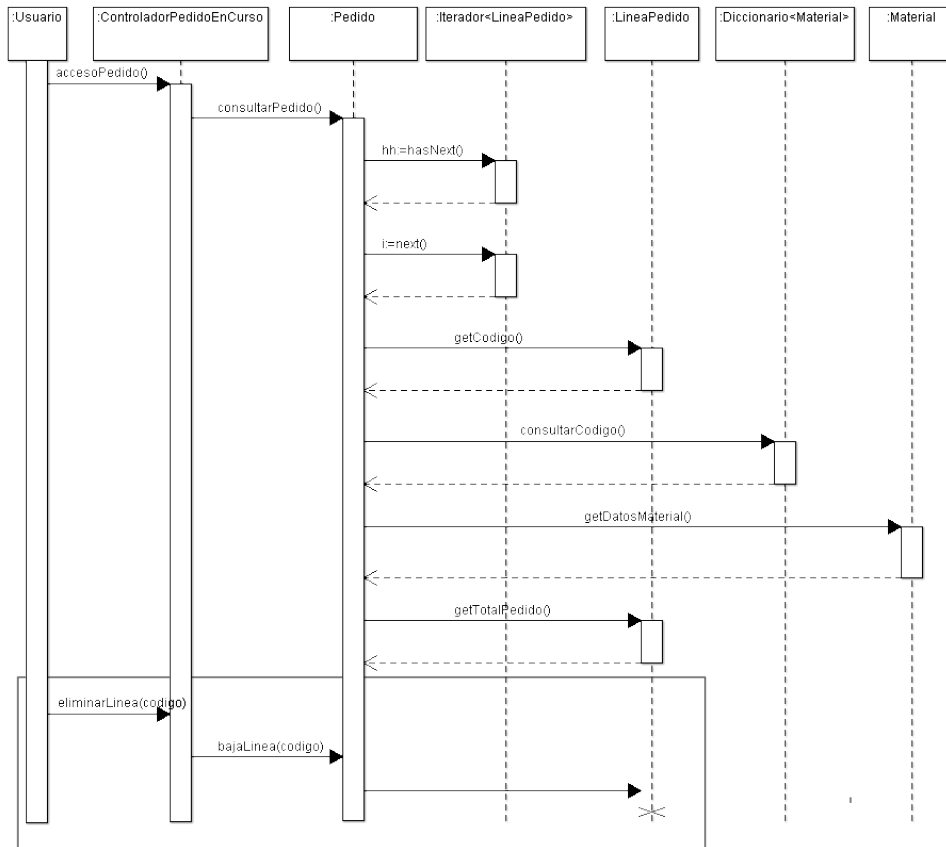


### 3.3.12. CU12 Gestión del pedido en curso

**Vista:** Usuario.

**Controlador:** ControladorPedidoEnCurso.

**Modelo:** Pedido, Iterador<LineaPedido>, LineaPedido, Diccionario<Material>, Material.

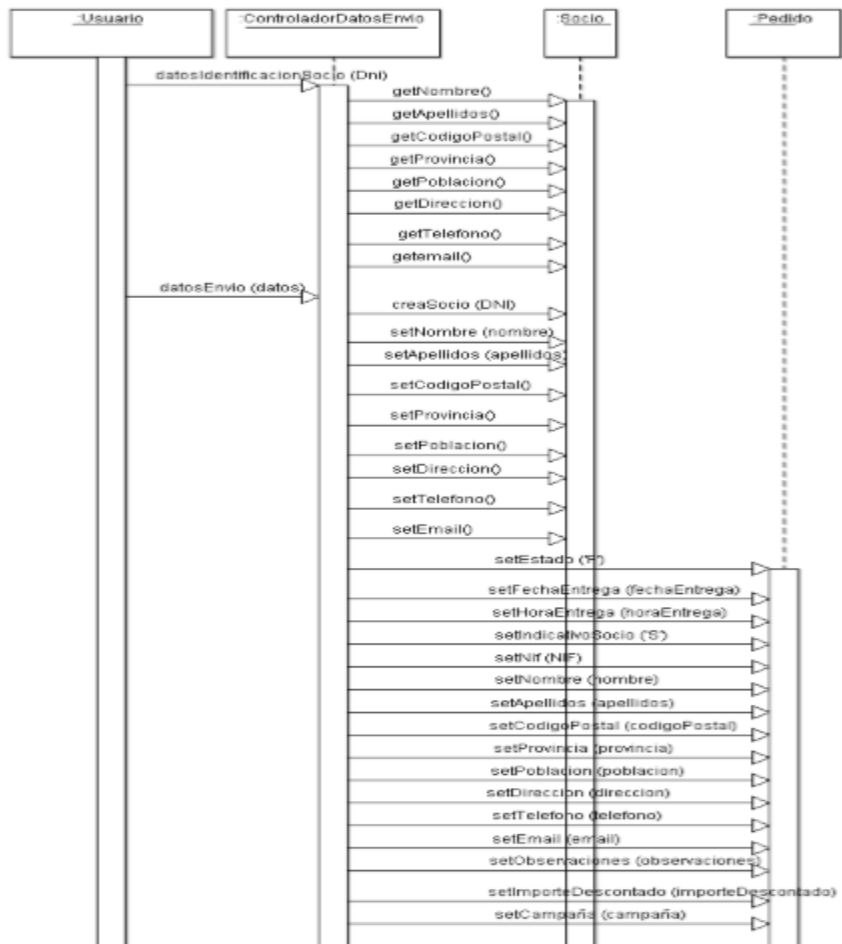


### 3.3.13. CU13 Datos de envío

Vista: Usuario.

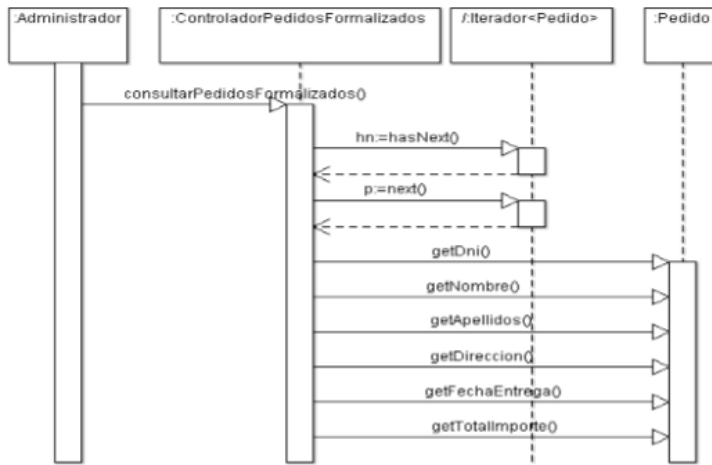
Controlador: ControladorDatosEnvio.

Modelo: Socio, Pedido.



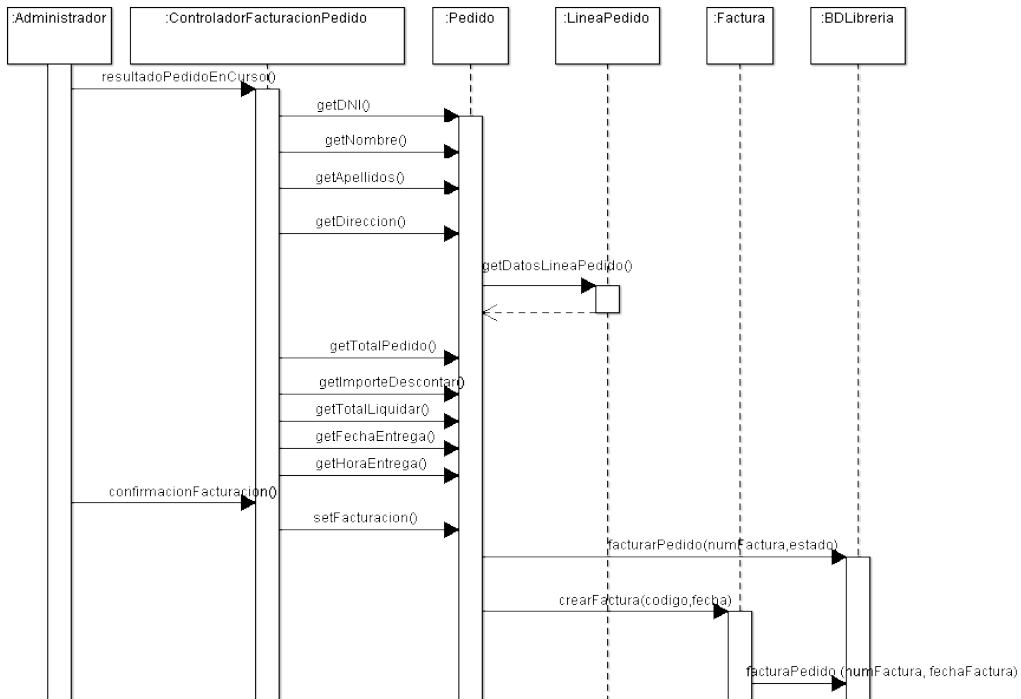
### 3.3.14. CU14 Consulta pedidos formalizados

**Vista:** Administrador.  
**Controlador:** ControladorPedidosFormalizados.  
**Modelo:** Iterador<Pedido>, Pedido.



### 3.3.15. CU15 Facturación pedido.

**Vista:** Administrador.  
**Controlador:** ControladorFacturacionPedido.  
**Modelo:** Pedido, Iterador<LineaPedido>, LineaPedido, Factura.





### 3.4. Clases de diseño.

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**. Aunque esta presentación de los DCD viene después de los diagramas de interacción, en la práctica se realizan en paralelo.

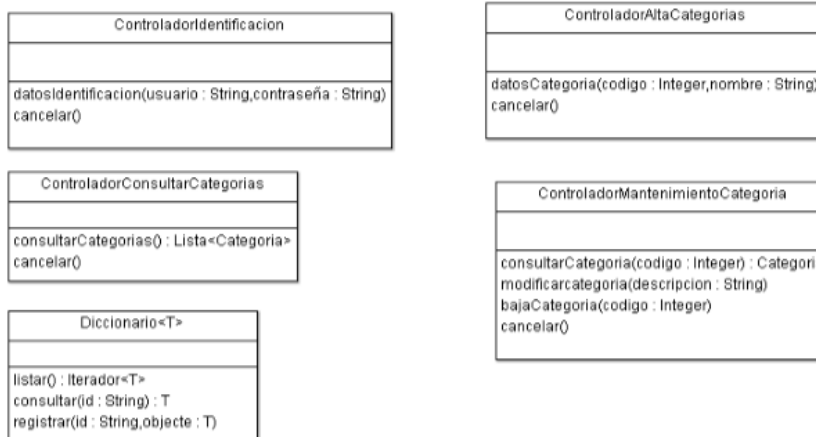
Un diagrama de clases de diseño o DCD representa las especificaciones de las clases e interfaces software en una aplicación. Entre la información que ha de reflejar se encuentra:

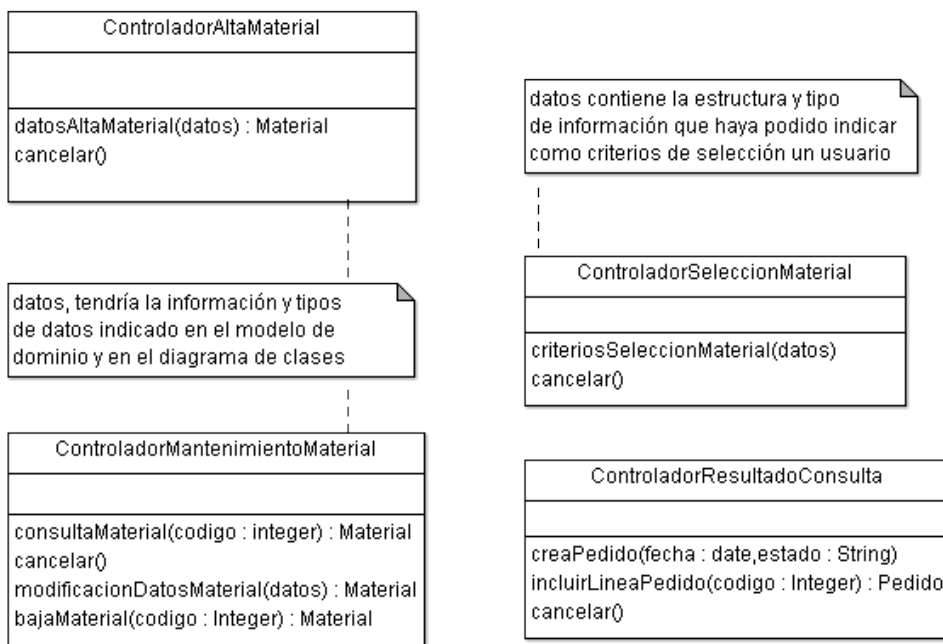
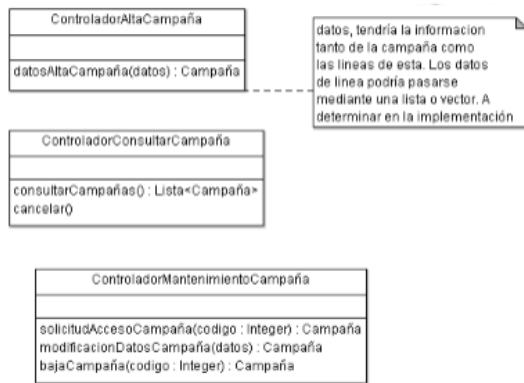
- a) Clases, asociaciones y atributos.
- b) Interfaces, con sus operaciones y constante.
- c) Métodos.
- d) Información acerca del tipo de los atributos.
- e) Navegabilidad.
- f) Dependencias.

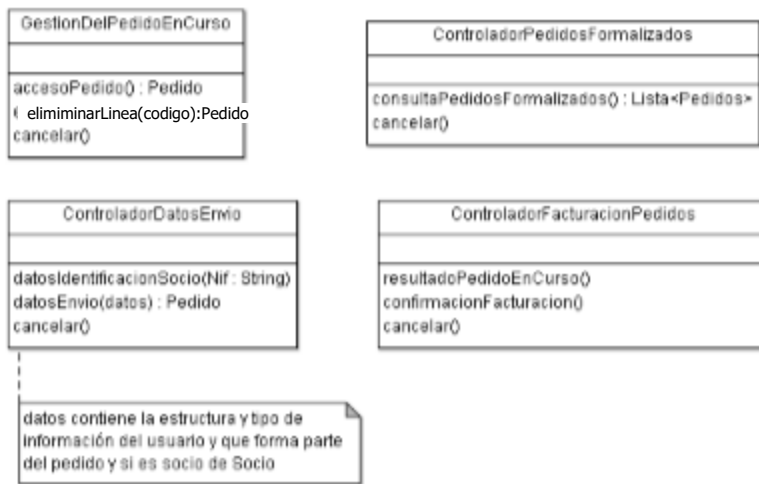
A diferencia de las clases conceptuales del Modelo del Dominio, las clases de diseño de los **DCD** muestran las definiciones de las clases software en lugar de los conceptos del mundo real.

En nuestro caso y como resultado de los casos de uso, del Modelo del Dominio y de los diagramas de interacción obtenemos las siguientes estructuras:

#### 3.4.1. Controladores del dominio.





**Figura 9. Controladores de dominio**

### 3.4.2. Paquete identificación.

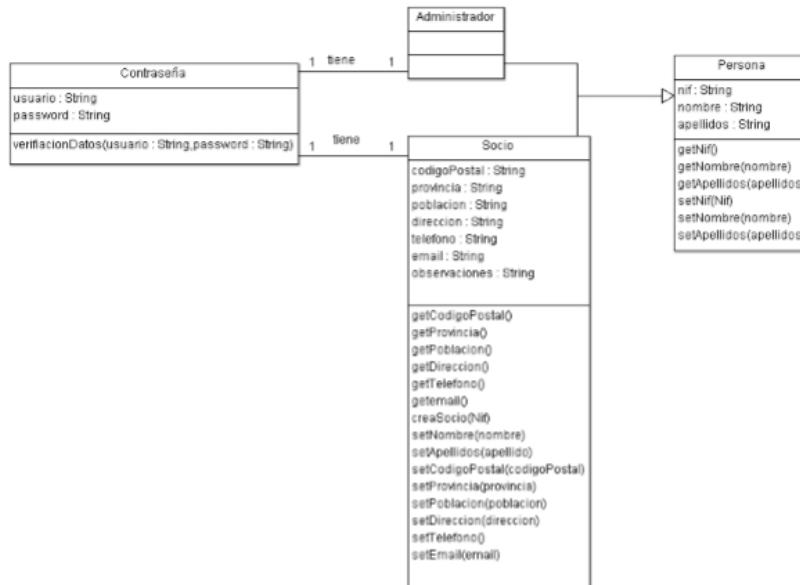


Figura 10. Diagrama de clase Módulo Identificación

3.4.3 Paquete productos.

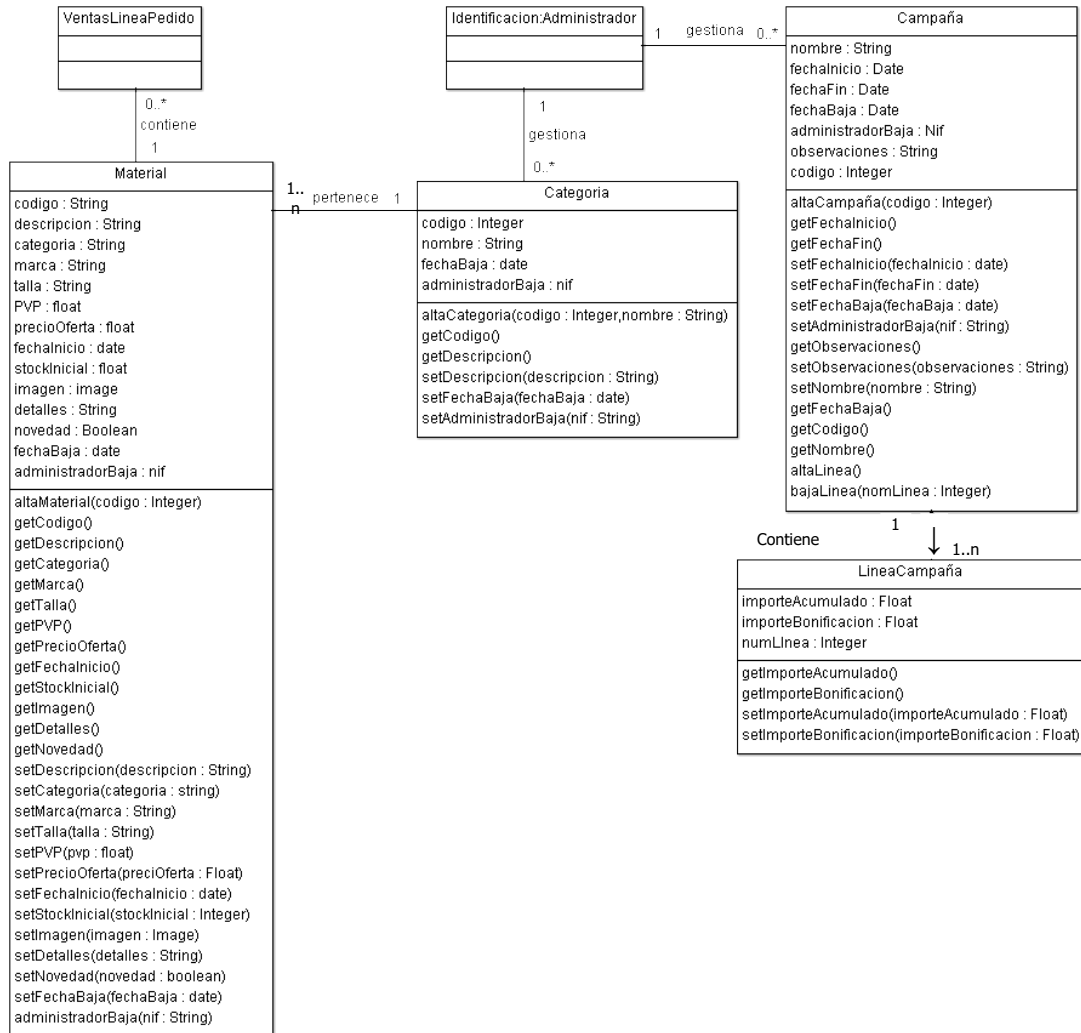
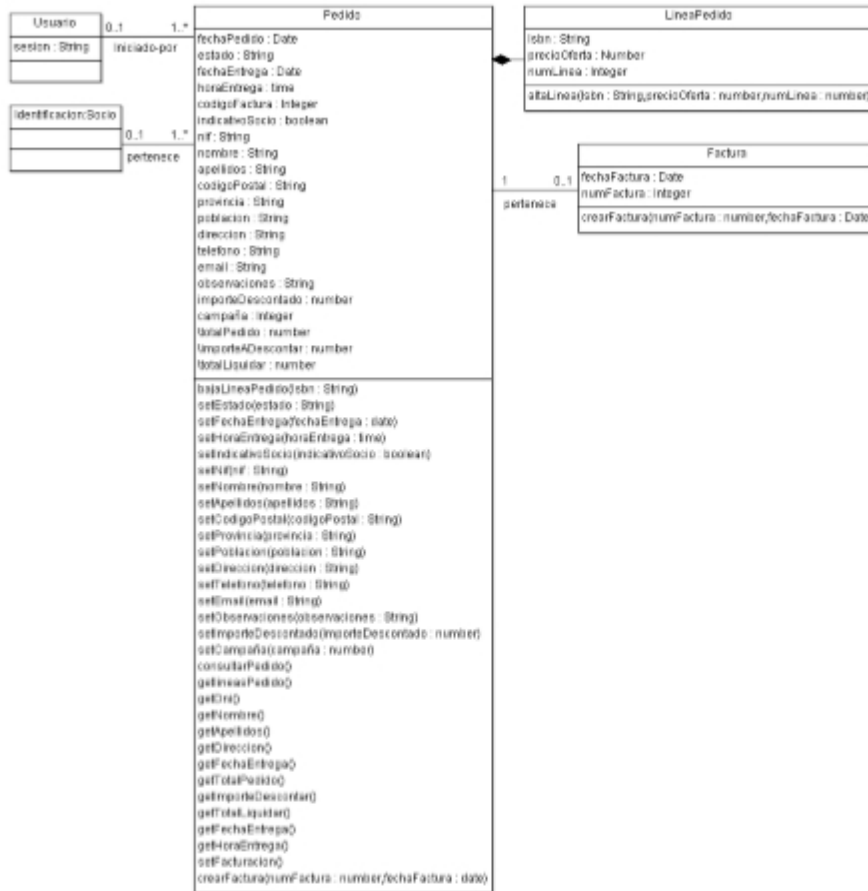


Figura 11. Diagrama de clases módulo productos

### 3.4.4. Paquete ventas.

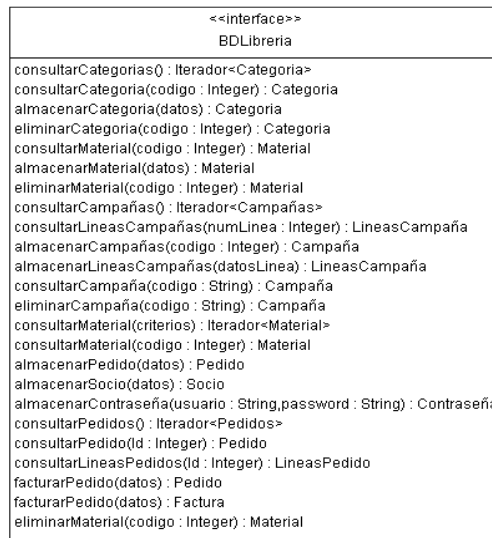


### 3.5. Servicios técnicos.

La aplicación requiere que se almacene y recupere la información en mecanismos de almacenamiento persistente, como, por ejemplo, una base de datos relacional, y este es en nuestro caso el objetivo básico del nivel de servicios técnicos.

Para ello actualmente tendríamos que construir un framework de almacenamiento de objetos persistentes, pero esto puede llevar mucho esfuerzo y actualmente en el mercado se puede conseguir o comprar según la tecnología uno de estos. Es por ello, que 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 (*en el diagrama de secuencia CU15 Facturación pedido, se ha realizado un detalle ilustrativo de como se podría acceder a los métodos, independientemente de la tecnología y/o framework que se utilice en la implementación, de la interface BDLibreria*).



**Figura 13. Servicios técnicos**

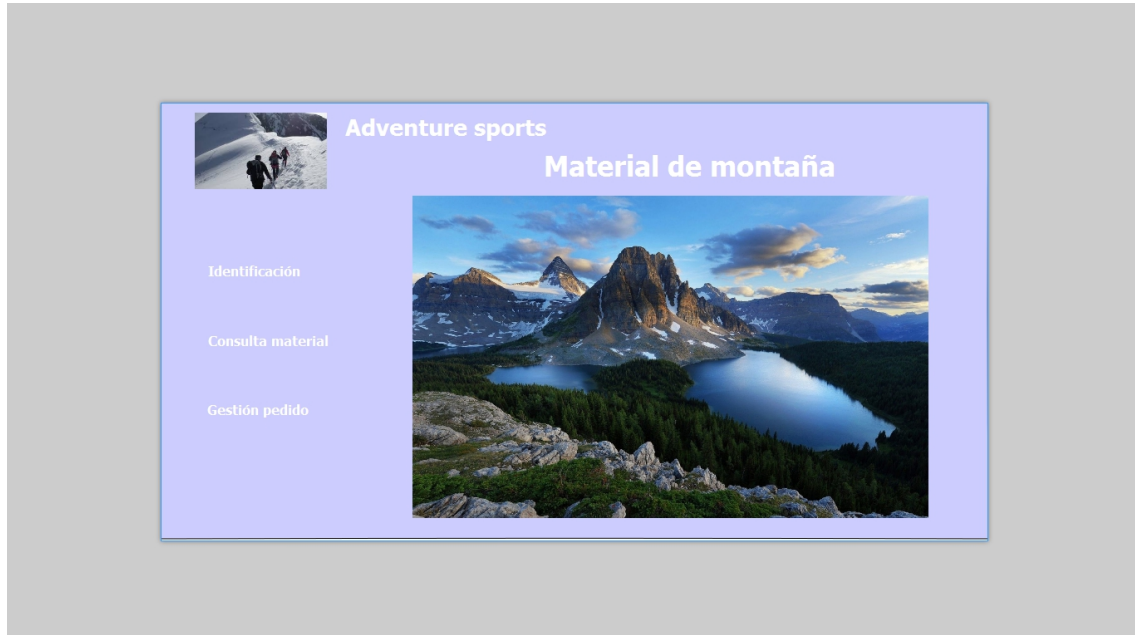
### 3.6 Diseño de la interficie gráfica del usuario.

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 y sin perder de vista que ha de cumplir parte de los requisitos no funcionales recogidos durante la fase de análisis.

### 3.6.1 Menu usuario.

El sistema presenta inicialmente esta pantalla. Si se trata de un socio, puede en cualquier momento identificarse como tal.



### 3.6.2 Menu administrador.

Si el usuario que se identifica es un administrador aparecerá el menú del administrador.





### 3.6.3 CU01 Identificación.

Es la interficie que utiliza tanto un socio, como el administrador para identificarse en el sistema.



### 3.6.4 CU02 Alta Categorías.

Desde esta pantalla se puede exclusivamente dar de alta nuevas categorías.



### 3.6.5 CU03 Consultar Categorías.

Desde esta pantalla se puede realizar la consulta de todas las categorías. Cabe señalar que si el usuario hace doble clic en la línea de detalle se navega al caso de uso del mantenimiento de la categorías, correspondiente a dicha categoría.



### 3.6.6 CU04 Mantenimiento categorías.

Permite modificar o dar de baja una categoría. Se puede acceder o bien directamente o bien a partir de la pantalla de consulta de categorías con una categoría previamente seleccionada.



### 3.6.7 CU05 Alta material deportivo.

Permite exclusivamente dar de alta un material.

Adventure sports

## Material de montaña

Identificación Código  Descripción

Alta categorías Categoría  Marca

Consulta categorías Talla  PVP  Precio oferta

Mantenimiento categorías Fecha inicio vigencia  Stock inicial  Novedad

**Alta material**

Mantenimiento material

Alta campañas

Consulta campañas

Mantenimiento campañas

Consulta pedidos

Facturación pedidos

Imagen 

Detalles

Enviar Cancelar

### 3.6.8. CU06 Mantenimiento material deportivo.

Permite exclusivamente consultar, modificar o dar de baja un material, Es requisito imprescindible que el material esté dado de alta previamente.

Adventure sports

## Material de montaña

Identificación Código  Descripción

Alta categorías Categoría  Marca

Consulta categorías Talla  PVP  Precio oferta

Mantenimiento categorías Fecha inicio vigencia  Stock inicial  Novedad

**Mantenimiento material**

Alta campañas

Consulta campañas

Mantenimiento campañas

Consulta pedidos

Facturación pedidos

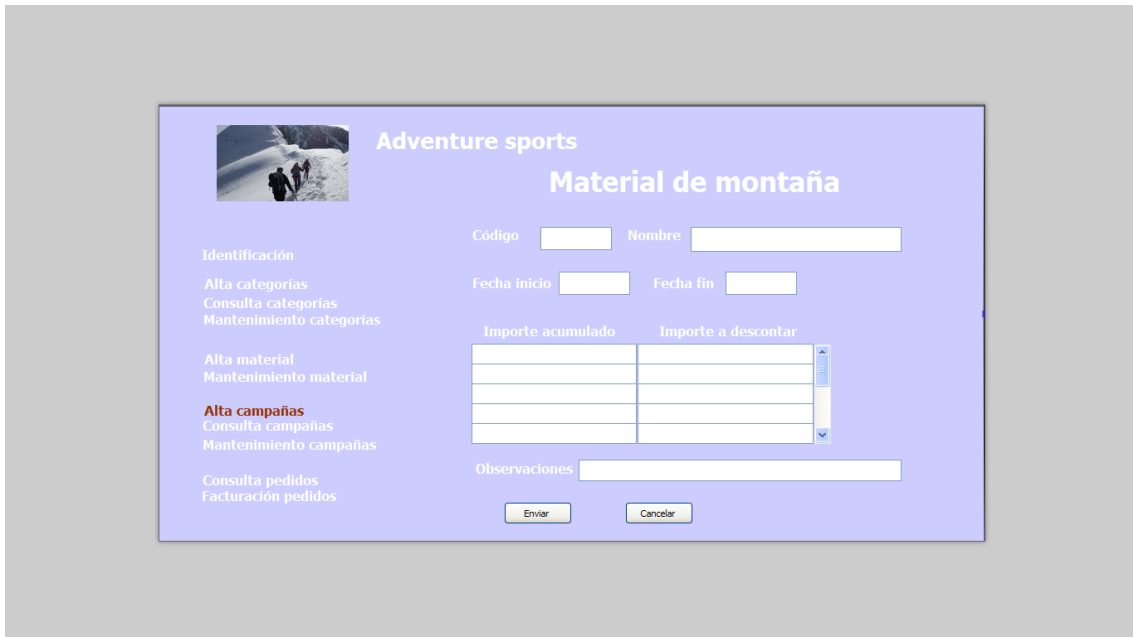
Imagen 

Detalles

Enviar Baja Cancelar

**3.6.9. CU07 Alta campañas.**

Permite exclusivamente dar de alta una campaña.




**3.6.10. CU08 Consulta campañas.**

Permite consultar las campañas y acceder al mantenimiento de éstas, mediante un doble clic en la línea de detalle correspondiente.



### 3.6.11. CU09 Mantenimiento campañas.

Permite consultar en detalle una campaña, modificar sus datos o darla de baja. Es imprescindible que la campaña exista previamente.



The screenshot displays a web interface for 'Adventure sports' with the title 'Material de montaña'. On the left, a navigation menu includes: 'Identificación', 'Alta categorías', 'Consulta categorías', 'Mantenimiento categorías', 'Alta material', 'Mantenimiento material', 'Alta campañas', 'Consulta campañas', 'Mantenimiento campañas' (highlighted in red), 'Consulta pedidos', and 'Facturación pedidos'. The main form contains the following fields: 'Código' and 'Nombre' (text boxes), 'Fecha inicio' and 'Fecha fin' (date pickers), 'Importe acumulado' and 'Importe a descontar' (a table with 4 rows and 2 columns), and 'Observaciones' (text area). At the bottom are 'Enviar', 'Baja', and 'Cancelar' buttons.

### 3.6.12. CU10 Consulta material deportivo.

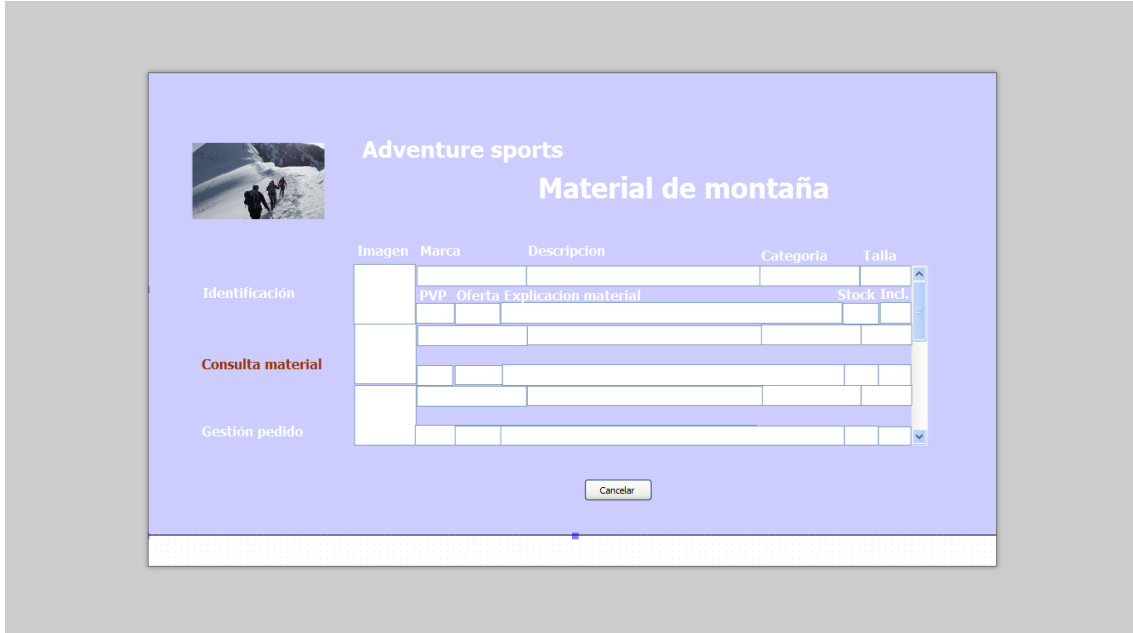
Permite introducir por parte de cualquier usuario, unos criterios de selección para la consulta del material deportivo. Es el único punto de acceso a dicha consulta.



The screenshot displays a web interface for 'Adventure sports' with the title 'Material de montaña'. On the left, a navigation menu includes: 'Identificación', 'Consulta material' (highlighted in red), and 'Gestión pedido'. The main form contains the following fields: 'Marca' (text box), 'Categoría' (two text boxes), 'Descripción' (text area), and 'Talla' (text box). At the bottom are 'Consultar' and 'Cancelar' buttons.

**3.6.13. CU11 Resultado consulta material deportivo.**

Permite visualizar la información mas relevante de cara a la adquisición de un material y también permite incluir material en el pedido de un usuario.



**3.6.14. CU12 Gestión del pedido en curso.**

Permite a un usuario visualizar el contenido de su pedido pendiente de formalizar. A su vez también permite eliminar material del pedido en curso.

Mediante el botón formalizar, pasará a el caso de uso de datos de envío, donde rellenará la información para su entrega y elaboración de la factura correspondiente.



**3.6.15. CU13 Datos de envío.**

Forma parte del proceso de formalización del pedido por parte del usuario, en esta interface se acaban de recoger los datos de entrega y facturación, así como información económica sobre el pedido. Se permite también mediante la entrada de una password que el usuario se pueda registrar si lo desea como socio, con los consiguientes beneficios económicos que le pueda reportar.

**3.6.16. CU14 Consulta pedidos.**

Permite al administrador consultar todos los pedidos formalizados por los usuarios o socios y que están pendientes de su entrega/facturación. Mediante la selección previo de un pedido y al clicar sobre el botón facturar pasará al caso de uso facturación pedidos.

### 3.6.17. CU15 Facturación pedidos.

Es el caso de uso mediante el cual el administrador puede repasar la complitud del pedido realizado por el usuario o socio para pasar a su entrega domiciliaria con la correspondiente factura.

Al clicar sobre facturar, el estado del pedido pasa a facturado y se emitirá la factura correspondiente, cerrando el ciclo del proceso de compra.

The screenshot shows a web application interface for 'Adventure sports' with the title 'Material de montaña'. The interface is divided into several sections:

- Header:** Includes a logo of two people on a snowy mountain and the title 'Material de montaña'.
- Form Fields:**
  - DNI:
  - Nombre y apellidos:
  - Dirección envío:
- Table:** A table with columns: Imagen, Marca, Descripción, Categoría, Talla, PVP, Oferta. The table is currently empty.
- Summary Fields:**
  - Total pedido:
  - Importe a descontar:
  - Total a liquidar:
  - Fecha y hora entrega:
- Buttons:** 'Facturar' and 'Cancelar' buttons are located at the bottom.
- Navigation Menu (Left):**
  - Identificación
  - Alta categorías
  - Consulta categorías
  - Mantenimiento categorías
  - Alta material
  - Mantenimiento material
  - Alta campañas
  - Consulta campañas
  - Mantenimiento campañas
  - Consulta pedidos
  - Facturación pedidos** (highlighted in red)

### 3.7. Diagrama de E/R (persistencia).

En cualquier negocio e-commerce, en nuestro caso estaríamos hablando de un B2C, se necesita, tal como hemos visto, 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.

Si bien podemos tener base de datos de objetos y o bases de datos relacionales, en nuestro caso nos inclinamos por éstas últimas por la fuerte implementación y predominio aún tienen.

En este apartado, lo que reflejamos es la estructura y necesidad de información que se requiere en la base de datos para nuestro negocio y teniendo en cuenta todo lo comentado hasta el momento.



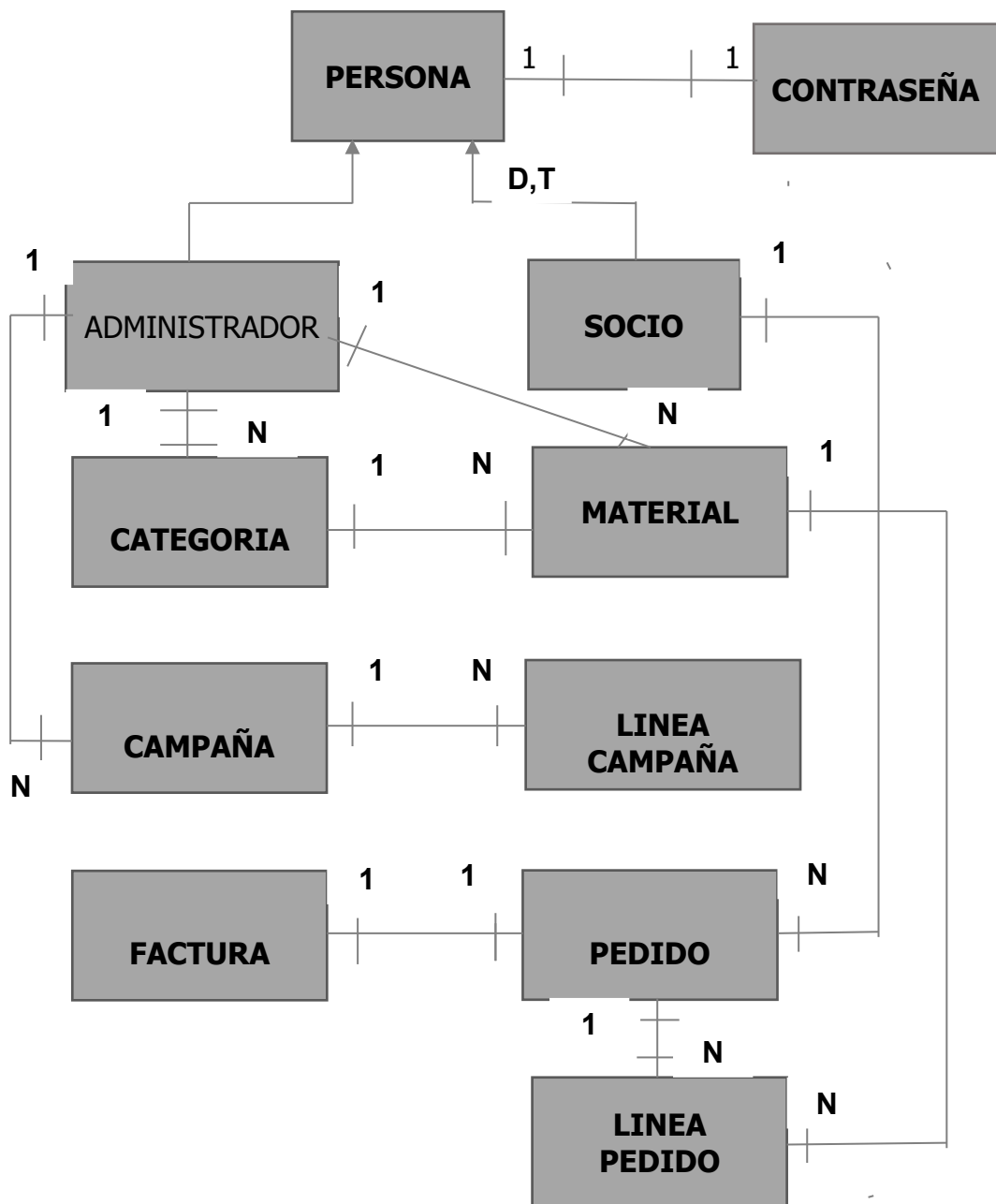


Figura 14. Diagrama E/R

## Relación de tablas de la base de datos

<b>Tabla PERSONA</b>				
<b>Objetivo:</b> Contiene la última información común de administradores o socios registrados				
Atributo	Tipo	Long.	Null	Constraint
Nif	String	10	N	PK
Nombre	String	30	N	
Apellidos	String	50	N	

<b>Tabla ADMINISTRADOR</b>				
<b>Objetivo:</b> Contiene el nif del tipo persona administrador				
Atributo	Tipo	Long.	Null	Constraint
Nif	String	10	N	PK y FK a PERSONA

<b>Tabla SOCIO</b>				
<b>Objetivo:</b> Contiene información del tipo persona socio				
Atributo	Tipo	Long.	Null	Constraint
Nif	String	10	N	PK y FK a PERSONA
Codigo_Postal	String	5	N	
Provincia	String	30	N	
Población	String	35	N	
Direccion	String	50	N	
Telefono	String	10	N	
Email	String	10	N	
Observaciones	String	70		

<b>Tabla CONTRASEÑA</b>				
<b>Objetivo:</b> Contiene información del usuario y contraseña de administradores o socios				
Atributo	Tipo	Long.	Null	Constraint
Nif	String	10	N	PK y FK a PERSONA
Usuario	String	10	N	
Password	String	10	N	

<b>Tabla CATEGORIA</b>				
<b>Objetivo:</b> Contiene información sobre las categorías.				
Atributo	Tipo	Long.	Null	Constraint
Codigo	String	10	N	PK
Nombre	String	30	N	
Fecha_baja	Date			
Administrador_baja	String	10		FK a ADMINISTRADOR

<b>Tabla MATERIAL</b>				
<b>Objetivo:</b> Contiene información de material deportivo.				
Atributo	Tipo	Long.	Null	Constraint
Codigo	Number	6	N	PK

Descripción	String	30	N	
Categoría	String	10	N	FK a CATEGORIA
Marca	String	25	N	
Talla	String	3		
Pvp	Number	5,2	N	
Precio_oferta	Number	5,2	N	
Fecha_inicio	Date		N	
Stock_actual	Number	4	N	
Stock_inicial	Number	4	N	
Imagen	Image		N	
Novedad	varchar	1	N	
Detalles	String	100	N	
Fecha_baja	Date			
Administrador	String	10		FK a ADMINISTRADOR

<b>Tabla CAMPAÑA</b>				
<b>Objetivo:</b> Contiene información sobre las campañas				
<b>Atributo</b>	<b>Tipo</b>	<b>Long.</b>	<b>Null</b>	<b>Constraint</b>
Codigo	String	10	N	PK
Nombre	String	30	N	
Fecha_inicio	Date		N	
Fecha_fin	Date		N	
Observaciones	String	70		
Fecha_baja	Date			
Administrador	String	10		FK a ADMINISTRADOR

<b>Tabla LINEAS_CAMPAÑA</b>				
<b>Objetivo:</b> Contiene información sobre las Líneas campañas				
<b>Atributo</b>	<b>Tipo</b>	<b>Long.</b>	<b>Null</b>	<b>Constraint</b>
Codigo	String	10	N	PK y FK CAMPAÑA
Num_linea	Number	2	N	PK
Importe_acumulado	Number	5,2	N	
Importe_bonificacion	Number	5,2	N	

Tabla PEDIDO				
<b>Objetivo:</b> Contiene información sobre el pedido				
Atributo	Tipo	Long.	Null	Constraint
Codigo	Number	6	N	PK
Estado	Varchar	1	N	P = Pendiente de facturar, F = Facturado
Fecha_entrega	Date		N	
Hora_entrega	Date		N	
Codigo_factura	Number	6		FK a FACTURA
Indicativo_socio	Varchar	1	N	S o N
Nif	String	10	N	FK a SOCIO
Nombre	String	30	N	
Apellidos	String	30	N	
Codigo_postal	String	5	N	
Provincia	String	30	N	
Poblacion	String	35	N	
Direccion	String	50	N	
Telefono	String	10	N	
Email	String	10	N	
Observaciones	String	70		
Total_pedido	Number	5,2	N	
Importe_a_descontar	Number	5,2	N	
Total_a_liquidar	Number	5,2	N	

Tabla LINEA_PEDIDO				
<b>Objetivo:</b> Contiene información sobre las Líneas del pedido				
Atributo	Tipo	Long.	Null	Constraint
Codigo	Number	6	N	PK y FK PEDIDO
Num_linea	Number	2	N	PK
Material	Number	6	N	FK MATERIAL
Precio_oferta	Number	5,2	N	

Tabla FACTURA				
<b>Objetivo:</b> Contiene información sobre el pedido				
Atributo	Tipo	Long.	Null	Constraint
Codigo	Number	6	N	PK
Fecha_factura	Date		N	

### 3.8. Diseño del test.

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.

Es importante la distinción entre testear y depurar:

**Depurar:** es el proceso que utiliza un programador para localizar defectos en el código y abordar su modificación.

**Testear:** es la exploración sistemática del comportamiento de un sistema, con el objetivo de localizar y reportar fallos. El testeo no incluye la corrección de errores.

Algunos datos estadísticos nos muestran la importancia del testeo:

Por cada 1.000 líneas de código puede haber hasta 30-40 errores. Mediante el proceso de testing se detectan y fijan unos cuantos errores.

La mayoría de errores (64%) se cometen durante la fase de diseño y un número más reducido (36%) durante la fase de implementación.

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 unas condiciones básicas para que sean testeables. Estas son:

- a) Los requerimientos del proyecto han de estar indicados de tal forma que puedan ser testeables.
- b) Los requerimientos han de ser precisos y detallados para evitar requerimientos ambiguos.
- c) Garantizar que no hay ningún requerimiento implícito o inexistente.
- d) Los requerimientos han de ser completos, es decir, recoger todas las condiciones y comportamiento que el usuario espera que tenga el sistema.

Si se dan las condiciones anteriores y mediante el testeo de los requerimientos se puede romper el ciclo error-defecto-fallo y detectamos a tiempo los defectos antes de pasar al próximo proceso dentro del ciclo de desarrollo.

Teniendo en cuenta lo anterior también se puede ver el testing como un proceso que está relacionado con el proceso de desarrollo.

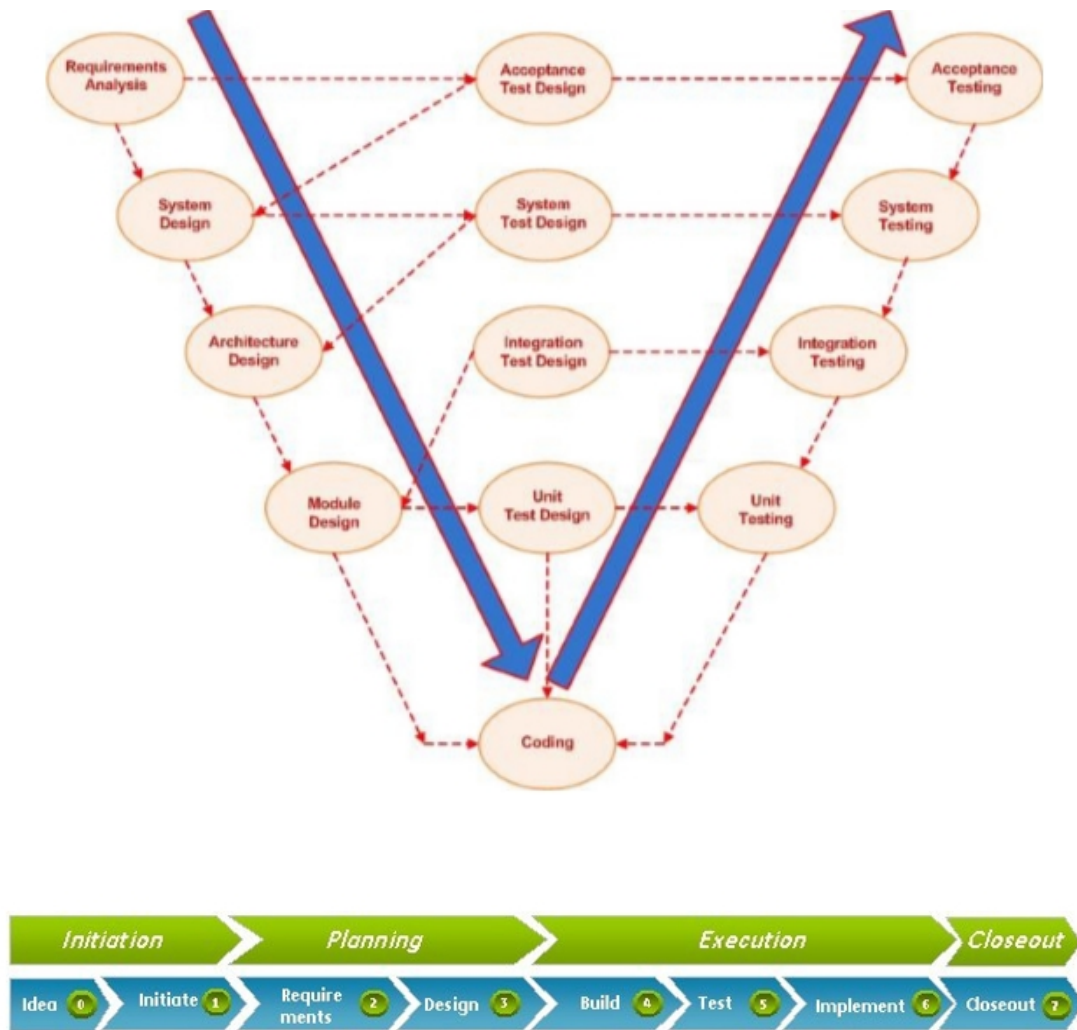


Figura 15. Modelo de Testing

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):

**Testing de caja blanca:** la estrategia de testing se define por la estructura del software. El objetivo es diseñar pruebas (entradas) de tal forma que garantice que toda línea de código se ejecuta al menos una vez, que toda función es probada por separado, etc.. Las técnicas usuales del testing de caja blanca son las pruebas de cobertura con el objetivo de asegurar que toda línea de código es probada considerando todos los posibles caminos del flujo de control. El testing de la caja blanca también se le denomina testing de la implementación. El testing que garantiza que toda línea de código se ejecuta al menos una vez se denomina testing de cobertura (**testing coverage**).

**Testing de caja negra:** la estrategia del testing viene definida por los requerimientos de software. En otros términos, escogemos entradas para someter el programa P que ahora es una caja negra. El resultado obtenido (la salida) se compara con lo esperado (según dice la especificación). En esta estrategia de test lo que nos interesa es que el software produzca unos resultados según lo que se espera del requerimiento. No importa, pues, como se han obtenido estos resultados de salida. El testing de caja negra se le conoce también como el testing de la especificación o testing de la omisión.

Como es imposible realizar un testintg exhaustivo, es decir, considerar todas las entradas posibles, nuestro objetivo será encontrar el conjunto de entradas de dimensiones razonables que nos permite deducir el comportamiento del programa como si hubieramos hecho un test exhaustivo.

El conjunto total de las posibles entradas están formados por los tres subconjuntos siguientes:

- a) Conjunto de **entradas válidas**: también se denominan valores dentro del rango.
- b) Conjunto de **entradas no válidas**: fuera del rango de valores permitidos.
- c) Conjunto de **entradas frontera**, que se encuentra entre los dos rangos de valores.

De los tres rangos de entrada que se pueden probar, en general hay un número infinito de valores, es necesario, entonces, escoger una muestra (un subconjunto finito) para cada uno de ellos.

Para las entradas válidas, esperamos que el programa dé cómo salida el resultado indicado por la especificación.

Para las entradas no válidas, esperamos que el programa nos indique "entrada no válida por motivo X".

Las entradas frontera tratan de probar el comportamiento del programa en valores que se encuentran en la frontera entre los dos subconjuntos de entradas: válidas y no válidas.

Por último, el **testing se realiza bajo un escenario concreto**, y sobre todo bajo la perspectiva del usuario. En este caso, consiste, pues, en ponerse en el lugar del usuario y 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 basamos en los casos de uso** que ya disponemos de las fases anteriores del proyecto.

El objetivo de las **pruebas basadas en escenarios** es comprobar que el software funciona de la manera que se espera que ha de funcionar. Por tanto, el software ha de cumplir con el caso de uso definido, el cual también habrá contemplado el diseño e implementación de las excepciones en caso de errores. El juego de pruebas que construyamos ha de revelar si el tratamiento de las excepciones se ha realizado.

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.

Lo que se defina en el UAT, nos servirá para que una vez implementado el producto, el usuario ejecute las pruebas de aceptación diseñadas en su momento, con el objetivo de validar y dar conformidad por su parte al producto final en cuanto a lo que ha requerimientos funcionales se refiere.

Dado que el objetivo del apartado de test dentro de este trabajo es reflejar/ilustrar someramente su finalidad y encaje dentro de la ingeniería de software, así como su relación con el ciclo de desarrollo. A la hora de plasmar el UAT dentro del proyecto, no se trataba de ser exhaustivo con todos los casos de uso, y es por ello que se han reflejado los 8 primeros casos de uso a modo de ejemplos.

## TEST CASE UAT.

Funcionalidad cubierta	Descripción	Paso	Descripción del paso	Resultados esperados
CU01 Identificación	Permite que el administrador se identifique. Escenario principal.	Step 1	Acceder a la nueva pantalla Identificación desde el menú principal.	Acceder a la pantalla sin problemas
		Step 2	Introduce un usuario correcto.	Se puede introducir el usuario sin problemas
		Step 3	Introduce el password correcto	Se puede introducir el password sin problemas
		Step 4	Se pulsa el botón verificar	Se verifican los datos y son correctos.
		Step 5	Se pulsa el botón Cancelar	Se sale del caso de uso y se vuelve al menú principal.
		Step 6	Visualizar el menu de administrador	Se visualiza el menu de administrador.
	Permite que el socio se identifique. Escenario principal.	Step 1	Acceder a la nueva pantalla Identificación desde el menú principal.	Acceder a la pantalla sin problemas
		Step 2	Introduce un usuario correcto.	Se puede introducir el usuario sin problemas.
		Step 3	Introduce el password correcto	Se puede introducir el password sin problemas
		Step 4	Se pulsa el botón Verificar.	Se verifican los datos y son correctos.
		Step 5	Se pulsa el botón Cancelar	Se sale del caso de uso y se vuelve al menu principal.
		Step 6	Visualizar el menu de administrador	Se visualiza el menú de Socio.
	Excepciones	Step 1	Acceder a la nueva pantalla	Identificación desde el menú principal. Acceder a la pantalla sin problemas.
		Step 2	Introduce un usuario correcto.	Se puede introducir el usuario sin problemas
		Step 3	Introduce el password incorrecta	Se puede introducir el password sin problemas
		Step 4	Se pulsa el botón verificar	Se verifican los datos y son incorrectos. Se dá el mensaje que el password es incorrecto y nos mantenemos en el caso de uso.
		Step 5	Introduce un usuario incorrecto.	Se puede introducir el usuario sin problemas
		Step 6	Introduce el password	Se puede introducir el password sin problemas



			correcto.	
		<b>Step 7</b>	Se pulsa el botón verificar	Se verifican los datos y son incorrectos. Se dá el mensaje que el usuario es incorrecto y nos mantenemos en el caso de uso.
		<b>Step 8</b>	Se deja el campo de usuario sin información	Se puede introducir el usuario sin problemas
		<b>Step 9</b>	Se deja el password sin información	Se puede introducir el password sin problemas
		<b>Step 10</b>	Se pulsa el botón verificar	Se verifican los datos y son incorrectos. Se dá el mensaje que el usuario y el password son datos obligatorios y nos mantenemos en el caso de uso.

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.	<b>Step 1</b>	Acceder a la nueva pantalla Alta de Categorías desde el menu principal del administrador.	Acceder a la pantalla sin problemas
		<b>Step 2</b>	Introduce un código de categoría inexistente.	Se puede introducir el código sin problemas
		<b>Step 3</b>	Introduce la descripción de la categoría	Se puede introducir una descripción sin problemas
		<b>Step 4</b>	Se pulsa el botón Enviar	El sistema registra la información en el sistema e informa de que es correcto. Se mantiene en formulario.
		<b>Step 5</b>	Se pulsa el botón Cancelar	Salimos del formulario y volvemos al menú principal.
		<b>Step 6</b>	Visualizar el menu del administrador.	El menú que aparece es el del administrador
	<b>Excepciones</b>	<b>Step 1</b>	Acceder a la nueva pantalla Alta de Categorías desde el menu principal.	Acceder a la pantalla sin Problemas.

		<b>Step 2</b>	Introduce el código de categoría que acaba de dar de alta.	Se puede introducir el código sin problemas
		<b>Step 3</b>	Introduce una descripción de la categoría.	Se puede introducir la descripción sin problemas.
		<b>Step 4</b>	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.
		<b>Step 5</b>	Se introduce un nuevo código de categoría.	Se puede introducir sin problemas.
		<b>Step 6</b>	Se deja el campo descripción de la categoría sin información	Se puede dejar sin información sin problemas.
		<b>Step 7</b>	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.
		<b>Step 8</b>	Se pulsa el botón Cancelar	Salimos del formulario y volvemos al menú principal.

Funcionalidad cubierta	Descripción	Paso	Descripción del paso	Resultados esperados
<b>CU03 Consulta Categorías</b>	<b>Pemitir que el administrador consulte las categorías vigentes. Escenario principal.</b>	<b>Step 1</b>	Acceder a la nueva pantalla Consulta de Categorías desde el menu principal del administrador.	Acceder a la pantalla sin problemas
		<b>Step 2</b>	Visualizar la información de pantalla	El sistema visualiza solo el código y la descripción de las categorías que no tienen fecha de baja.
		<b>Step 3</b>	Se pulsa el botón Cancelar	El menú que aparece es el del administrador.
	<b>Acceso al mantenimiento de categorías</b>	<b>Step 1</b>	Acceder a la nueva pantalla Consulta de Categorías desde el menu principal del administrador.	Acceder a la pantalla sin problemas
		<b>Step 2</b>	Visualizar la información de	El sistema visualiza solo el código y la descripción

			pantalla	de las categorías que no tienen fecha de baja.
		<b>Step 3</b>	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.
		<b>Step 4</b>	El usuario dá 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.
		<b>Step 5</b>	Visualizar la información de pantalla	El sistema 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.
		<b>Step 6</b>	Se pulsa el botón Cancelar	El menú que aparece es el del administrador.

Funcionalidad cubierta	Descripción	Paso	Descripción del paso	Resultados esperados
CU04 Mantenimiento Categorías	Permite que el administrador dé de alta en el sistema las categorías. Escenario principal	<b>Step 1</b>	Acceder a la nueva pantalla Consulta de Categorías desde el menú principal del administrador.	Acceder a la pantalla sin Problemas.
		<b>Step 2</b>	Visualizar la información de Pantalla.	El sistema visualiza solo el código y la descripción de las categorías que no tienen fecha de baja.
		<b>Step 3</b>	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.
		<b>Step 4</b>	El usuario modificar la descripción de la categoría.	Permite modificar la descripción de la categoría.
		<b>Step 5</b>	El usuario intenta modificar el código de la categoría.	El sistema no le permite modificarlo.
		<b>Step 6</b>	El usuario clica el botón Enviar	El sistema registra los datos y navega al caso de uso Consulta Categorías.

		<b>Step 7</b>	El usuario visualiza la información en pantalla.	El usuario vé que la descripción de la categoría modificada es correcta.
		<b>Step 8</b>	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
		<b>Step 9</b>	El usuario deja sin información el campo descripción.	El sistema deja hacerlo sin problemas.
		<b>Step 10</b>	El usuario clica el botón Enviar	El sistema informa de que la descripción es obligatoria y se mantiene en el caso de uso
		<b>Step 11</b>	El usuario clica el botón Cancelar	El sistem vuelve al caso de uso Consulta Categorías.
		<b>Step 12</b>	Visualizar la información de Pantalla.	El sistema visualiza correctamente la descripción de la categoría que había dejado sin informar anteriormente
		<b>Step 13</b>	El usuario clica el botón Cancelar.	El sistema vuelve al menú de administrador.

Funcionalidad cubierta	Descripción	Paso	Descripción del paso	Resultados esperados
CU05 Alta material	Pemitir que el administrador dé de alta la información sobre material. Escenario principal	<b>Step 1</b>	Acceder a la nueva pantalla Alta de material desde el menu principal del administrador.	Acceder a la pantalla sin problemas
		<b>Step 2</b>	Introduce el código.	Se puede introducir sin problemas
		<b>Step 3</b>	Introduce la descripción	Se puede introducir sin problemas
		<b>Step 4</b>	Introduce categoría.	Se puede introducir sin problemas
		<b>Step 5</b>	Introduce la marca	Se puede introducir sin Problemas.
		<b>Step 6</b>	Introduce la talla	Se puede introducir sin Problemas.
		<b>Step 7</b>	Introduce PVP .	Se puede introducir sin Problemas.
		<b>Step 8</b>	Introduce precio oferta al socio.	Se puede introducir sin Problemas.
		<b>Step 9</b>	Introduce la fecha de inicio de vigencia.	Se puede introducir sin Problemas.

		<b>Step 10</b>	Introduce el stock inicial	Se puede introducir sin Problemas.
		<b>Step 11</b>	Introduce/asocia la imagen del material	Se puede introducir sin Problemas.
		<b>Step 12</b>	Introduce detalles del material	Se puede introducir sin Problemas.
		<b>Step 13</b>	El sistema está marcando por defecto que no es una novedad. El usuario introduce que si lo és.	Se puede introducir sin Problemas.
		<b>Step 14</b>	El usuario clica en Enviar	El sistema registra los datos e informa de ello. El sistema se mantiene en el caso de uso.
	<b>Excepciones</b>	<b>Step 1</b>	El usuario con la información que está en pantalla del anterior material clica en Enviar.	El sistema informa que ya existe el material en el sistema y se mantiene en el caso de uso.
		<b>Step 2</b>	El usuario modifica el código.	Se puede modificar sin problemas.
		<b>Step 3</b>	El usuario modifica la descripción y la deja en blanco.	Se puede dejar en blanco sin problemas.
		<b>Step 4</b>	El usuario clica en Enviar	El sistema avisa de que la descripción es un campo obligatorio y se mantiene en el caso de uso.
		<b>Step 5</b>	El usuario repite los dos pasos anteriores para el resto de campos	El sistema avisará de que es un dato obligatorio y se mantiene en el caso de uso.
		<b>Step 6</b>	El usuario clica Cancelar	El sistema vuelve al menú de Administrador.

<b>Funcionalidad cubierta</b>	<b>Descripción</b>	<b>Paso</b>	<b>Descripción del paso</b>	<b>Resultados esperados</b>
<b>CU06 Mantenimiento o baja de material</b>	<b>Pemitir que el administrador modificar dar de baja un material. Escenario principal</b>	<b>Step 1</b>	Acceder a la nueva pantalla Mantenimiento de material desde el menú principal del administrador.	Acceder a la pantalla sin problemas
		<b>Step 2</b>	Introduce un código que ya existe y no está dado de baja.	Se puede introducir sin problemas y el sistema recupera y visualiza la información del material.
		<b>Step 3</b>	El usuario clica	El sistema informa al

			botón Baja.	usuario que el material ha sido dado de baja y se mantiene en el caso de uso.
		<b>Step 4</b>	Introduce un código que ya existe y que no ha entrado en vigencia y que no está dado de baja.	El sistema recupera y visualiza la información sin problemas.
		<b>Step 5</b>	El usuario modifica la descripción	Se puede modificar sin Problemas.
		<b>Step 6</b>	El usuario clicca botón Enviar.	El sistema informa que se ha registrado la información sin problemas y se mantiene en el caso de uso.
		<b>Step 7</b>	Introduce un código que ya existe y que no ha entrado en vigencia y que no está dado de baja.	El sistema recupera y visualiza la información sin problemas.
		<b>Step 8</b>	El usuario intenta modificar el stock actual.	El sistema informa al usuario que el dato no se puede modificar y se mantiene en el caso de uso.
		<b>Step 9</b>	Introduce un código que ya ha sido dado de baja.	El sistema informa al usuario que el material ha sido dado de baja y se mantiene en el caso de uso.
		<b>Step 10</b>	Introduce un código que no ha sido dado de baja y ya ha entrado en vigencia.	El sistema recupera y visualiza la información sin problemas.
		<b>Step 11</b>	El usuario intenta modificar alguno de los campos que no son modificables al entrar en vigencia	El sistema lo deja hacer sin problemas.
		<b>Step 12</b>	El usuario clicca sobre botón Enviar.	El sistema avisa que el dato no es modificable y se mantiene en el caso de uso.
		<b>Step 13</b>	Introduce el código que ha modificado la descripción	El sistema recupera y visualiza la información sin problemas, viendo que

			anteriormente	la información sobre la descripción es correcta.
	<b>Excepciones</b>	<b>Step 1</b>	Introduce un código que no existe	El sistema informa al usuario que el material no existe y se mantiene en el caso de uso.
		<b>Step 2</b>	Introduce un código que ya existe.	Se puede introducir sin problemas y el sistema recupera y visualiza la información del material.
		<b>Step 3</b>	Modifica la fecha de inicio vigencia poniéndola anterior a la fecha actual.	Permite introducirla sin problema.
		<b>Step 4</b>	El usuario clica botón Enviar	El sistema informa que la fecha de inicio vigencia es incorrecta y se mantiene en el caso de uso.
		<b>Step 5</b>	El usuario modifica la descripción y la deja en blanco.	Se puede dejar en blanco sin problemas.
		<b>Step 6</b>	El usuario clica en Enviar	El sistema avisa de que la descripción es un campo obligatorio y se mantiene en el caso de uso.
		<b>Step 7</b>	El usuario repite los dos pasos anteriores para el resto de campos	El sistema avisará de que es un dato obligatorio y se mantiene en el caso de uso.
		<b>Step 8</b>	El usuario clica Cancelar	El sistema vuelve al menu de Administrador

<b>Funcionalidad cubierta</b>	<b>Descripción</b>	<b>Paso</b>	<b>Descripción del paso</b>	<b>Resultados esperados</b>
<b>CU07 Alta Campañas</b>	<b>Pemitir que el administrador dé de alta en el sistema las campañas de marketing. Escenario principal</b>	<b>Step 1</b>	Acceder a la nueva pantalla Alta Campañas desde el menu principal del administrador.	Acceder a la pantalla sin problemas
		<b>Step 2</b>	Introduce un código de campaña que no exista	Se puede introducir sin problemas

		<b>Step 3</b>	Introduce el código de la campaña	Se puede introducir sin problemas
		<b>Step 4</b>	Introduce la fecha inicio	Se puede introducir sin problemas
		<b>Step 5</b>	Introduce la fecha fin	Se puede introducir sin problemas
		<b>Step 6</b>	Introduce el importe acumulado	Se puede introducir sin problemas
		<b>Step 7</b>	Introduce el importe a abonar	Se puede introducir sin problemas
		<b>Step 8</b>	Repetir los pasos 6 y 7 para 4 intervalos más	Se puede introducir sin problemas
		<b>Step 9</b>	El usuario introduce observaciones	Se puede introducir sin problemas
		<b>Step 10</b>	El usuario clic el botón Enviar	Se registra la información y se informa al usuario. El sistema se mantiene en la pantalla.
	<b>Excepciones</b>	<b>Step 1</b>	Introduce un código de campaña que no exista	Se puede introducir sin problemas
		<b>Step 2</b>	Introduce una fecha de inicio que ya existe	Se puede introducir sin problemas
		<b>Step 3</b>	Introduce una fecha fin que ya exista	Se puede introducir sin problemas
		<b>Step 4</b>	Introduce el importe acumulado	Se puede introducir sin problemas
		<b>Step 5</b>	Introduce el importe a abonar	Se puede introducir sin problemas
		<b>Step 6</b>	El usuario clic el botón Enviar	El sistema le indica que ya existe una campaña con solapamiento de las fechas de inicio y fin, no se registra la información y nos mantenemos en la pantalla.
		<b>Step 7</b>	Introducir una fecha de inicio vacía	Se puede introducir sin problemas
		<b>Step 8</b>	Introducir una fecha fin vacía	Se puede introducir sin problemas
		<b>Step 9</b>	El usuario clic el botón Enviar	El sistema indica que existe información que se ha de introducir, no se registra la información y nos mantenemos en la pantalla
		<b>Step 10</b>	Introducir una fecha de inicio	Se puede introducir sin problemas
		<b>Step 11</b>	Introducir una fecha fin inferior a la fecha de inicio	Se puede introducir sin problemas
		<b>Step 12</b>	El usuario clic el	El sistema indica que



			botón Enviar	existe información que se la fecha inicio y fecha fin no son coherentes, no se registra la información y nos mantenemos en la pantalla
		<b>Step 13</b>	El usuario introduce una fecha inicio que no se solapan con ninguna campaña	Se puede introducir sin problemas
		<b>Step 14</b>	El usuario Introduce una fecha fin que no se solape con ninguna campaña	Se puede introducir sin problemas
		<b>Step 15</b>	Introduce algun rango de importe acumulado que se solapan con alguno de los ya introducidos	Se puede introducir sin problemas
		<b>Step 16</b>	El usuario clica el botón Enviar	El sistema indica que existe información que existen rangos de importe acumulado que se solapan, el sistema no registra la información y nos mantenemos en la pantalla.
		<b>Step 17</b>	El usuario clica Cancelar	El sistema vuelve al menu de Administrador

Funcionalidad cubierta	Descripción	Paso	Descripción del paso	Resultados esperados
<b>CU08 Consulta Campañas</b>	<b>Pemitir que el administrador pueda consultar las campañas. Escenario principal</b>	<b>Step 1</b>	Acceder a la nueva pantalla Consulta de Campañas desde el menu principal del administrador.	Acceder a la pantalla sin problemas
		<b>Step 2</b>	Visualizar la información de pantalla	El sistema visualiza solo el código, nombre, fecha inicio, fecha fin, fecha baja y observaciones.
		<b>Step 3</b>	Se pulsa el botón Cancelar	El menú que aparece es el del administrador
	<b>Acceso al mantenimiento de categorías</b>	<b>Step 1</b>	Acceder a la nueva pantalla Consulta de Categorías desde el menu principal del administrador.	Acceder a la pantalla sin problemas
		<b>Step 2</b>	Visualizar la	El sistema visualiza solo

			información de pantalla	el código, nombre, fecha inicio, fecha fin, fecha baja y observaciones.
		<b>Step 3</b>	El usuario hace doble clic sobre una línea de campaña	El sistema visualiza el caso de uso Mantenimiento Campañas y muestra la información sobre dicha campaña.
		<b>Step 4</b>	El usuario da de baja la Campaña desde el caso de uso Mantenimiento campañas.	El sistema avisa de que ha dado de baja la categoría y abandona el caso de uso Mantenimiento Campañas.
	<b>Excepciones</b>	<b>Step 1</b>	Visualizar la información de pantalla	El sistema visualiza solo el código, nombre, fecha inicio, fecha fin, fecha baja y observaciones. En pantalla aparece la campaña dada de baja anteriormente
		<b>Step 2</b>	El usuario hace doble clic sobre la línea de campaña con fecha de baja	El sistema le indica que dicha campaña está dada de baja.
		<b>Step 3</b>	Se pulsa el botón Cancelar	El menú que aparece es el del administrador

## 4. Aspectos relacionados con la gestión del proyecto.

En un proyecto real de éstas características, hay muchos temas que se tendrían que conocer para poder desarrollar otros aspectos relacionados con la gestión de un proyecto. El mismo proyecto indicado en esta Memoria, dependiendo de la dimensión, estructura y ámbito de actuación de la empresa, puede variar en muchos aspectos, en los que no se han entrado a considerar, pues para ellos necesitaríamos que contextualizar mucho más todo el proyecto, por ejemplo, si se ha de trabajar en otros idiomas, que nivel de servicio ha de dar, con que calidad, nivel de respuesta, etc..

Al tratarse de un trabajo académico, es difícil y sería un ejercicio totalmente ficticio pretender dar una valoración económica o riesgos asociados al proyecto, porque además de los nombrados en el párrafo anterior también figuraría la plataforma tecnológica, que se va a utilizar, si se externalizará o no parte o la totalidad del servicio y de su desarrollo, etc.. Todos, aspectos imposibles de determinar con la información incluida en dicha Memoria y para los cuales necesitaríamos del conocimiento de muchos otros temas que van más allá del alcance y objetivo perseguidos en una Memoria relacionado con la Ingeniería del Software, y no sobre Gestión de Proyectos.

En cualquier caso, si veo interesante, hacer una relación somera de las partes que necesitarían ser consideradas, de una forma parcial y/o total, para poder abarcar todos los aspectos relevantes relacionados con la gestión del proyecto y entre los que cabría considerar, evidentemente, los aspectos de riesgo y presupuesto. Entre todos ellos y siguiendo los métodos profesionales más habituales y que están referenciados en PMBOK (Project Management Body of Knowledge), podríamos destacar:

**Definición detallada del alcance del proyecto.** En la Memoria y en términos reales, faltaría muchos otros aspectos a detallar y considerar.

**Estructuración de éste.** En la Memoria y en términos reales está desarrollado más como ejercicio académico que no como planificación real aunque sea de forma simulada de lo que tendríamos que considerar en un entorno profesional.

**Planificación temporal.** En la Memoria y en términos reales está desarrollado más como ejercicio académico que no como planificación real en un entorno profesional.

**Presupuesto detallado.** Imposible de establecer pues depende de muchos aspectos que no están considerados, ni han sido desarrollados en el proyecto.

**Plan de gestión de riesgos.** Imposible de establecer pues depende al igual que el punto anterior de muchos otros aspectos que no se han considerado.

**Plan de calidad.** Imposible de establecer pues depende al igual que el punto anterior de muchos otros aspectos que no se han considerado.

**Plan de administración de contratos y compras.** Imposible de establecer pues depende al igual que el punto anterior de muchos otros aspectos que no se han considerado.

### 4.1 Acta de constitución del proyecto:

Teniendo en cuenta lo comentado anteriormente, el primer paso que tendríamos que dar para poder completar los capítulos anteriormente comentados, y llevar adelante en términos profesionales el proyecto, sería realizar un acto de constitución del proyecto donde tendría que figurar, principalmente:

#### 4.1.1. Estructura de la gestión del proyecto:

**Modelo de relación.** Breve resumen de los principios de gestión que se aplicarán.

**Coordinador general del proyecto desde un punto de vista del negocio.** Quien es y que hace.

**Coordinador general del proyecto desde un punto de vista de IT.** Quien es, que hace y que responsabilidades tiene.

**Comité ejecutivo.** Quienes son, que tareas y que responsabilidades tiene cada uno.

**Responsables de los grupos de trabajo.** Quienes son, que tareas y que responsabilidades tiene cada uno.

**Socios del proyecto.** Quienes son y que tareas tienen encomendadas.

#### 4.1.2. Estrategias de gestión.

**Política de planificación.** es decir, explicar que aproximación se hará servir para planificar el proyecto.

**Metodologías del proyecto.** que tecnologías se emplearán (gestión y desarrollo).

**Análisis y gestión de riesgos.** Sumario.

**Control de la calidad.** Sumario.

**Procedimientos para resolver conflictos.** Sumario.

**Reuniones de seguimiento.** Sumario.

Los dos apartados anteriores son fundamentales y a tener en cuenta, dado que ningún proyecto de IT, se maneja y funciona en términos reales de forma independiente de todo un contexto (entidad pública o privada de la que depende, estructura, ámbito de actuación, necesidades de dicha entidad, etc...) y por tanto es todo ese contexto, junto con la aplicación de la praxis de la gestión de proyectos, la que nos permitiría desarrollar con rigor los aspectos anteriormente indicados.

En la Memoria, y por motivos académicos se ha realizado, con sentido, la suposición de un proyecto de desarrollo de software para la aplicación de aspectos relacionados con la ingeniería del software (no la gestión de proyectos) e independiente de un contexto real o ficticio de lo que sería las consideraciones integrales de la gestión de un proyecto. Esto, si bien nos permite poner en juego aspectos metodológicos y relacionados estrictamente con la ingeniería de software, no veo que nos pueda dar pie a más consideraciones de otros temas relacionados con la gestión de proyectos como pueden ser los de tipo presupuestario, calidad, riesgo, etc...a menos de no querer realizar un trabajo totalmente ficticio/especulativo.

## 5. 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 tecnológico. Al fin y al cabo, llevar adelante un proyecto es una tarea multidisciplinar y para ello es importante disponer de una visión holística de lo que estamos tratando.

Siempre he entendido que la tecnología de sistemas, y en particular la ingeniería de software ha de estar focalizada, sobre todo, en los requerimientos del sistema, a los que no se ha de perder nunca de vista y que han de actuar como hilo conductor en todo momento. En cambio la ingeniería de software ha de actuar como medio (y no fin) en todo momento para dar él soporte y servicio técnico necesario.

Aunque se trata de un hecho obvio, por experiencia profesional he visto muchas veces problemas que han provocado retrasos, costes añadidos, compra o desarrollo de software que no se ajusta a las necesidades reales, etc.. y no tanto por problemas técnicos, que también, sino por tener una visión excesivamente técnica y priorizar dicha técnica desfocalizando los requerimientos reales del proyecto. De hecho, la búsqueda de éstas necesidades reales se ha de convertir en todo momento en el objetivo constante y prioritario de todo proyecto. No se puede poner encima de la mesa, como a veces pasa, primero la herramienta técnica, sea por disponibilidad y/o dominio, o ambas cosas a la vez, y despues adaptar los requerimientos a las posibilidades de dicha herramienta.

Bajo mi punto de vista, a veces estos problemas se producen porque en muchos casos, nos convertimos o somos especialistas en unas tecnologías específicas que dominamos y conocemos en profundidad, y esto hace que se piensen los requerimientos/necesidades del usuario en base a las posibilidades de la tecnología que se domina y no al revés, es decir, recoger e incluso intentar extraer y verificar lo mas fidedignamente los requerimientos funcionales reales, diseñar la arquitectura necesaria que dé respuesta a éstos, independientemente de la plataforma tecnológica, y despues buscar, dentro del amplio abánico de tecnologías disponibles, aquellas que puedan dar mejor respuesta a dichas necesidades, así como la localización de los recursos técnicos humanos que mas se ajusten a ellas.

Dicho ésto, sin ninguna duda, todos los aspectos tratados durante la carrera son aspectos que espero continuar tratando y ampliando en un futuro inmediato en el Master en Ingeniería Informática, tanto técnicos, como metodológicos y de gestión, y desde este punto de vista, valoro todo el trabajo realizado en la Memoria como un elemento motivador para seguir profundizando y aprendiendo en este interesante mundo de la ingeniería del software.

## 6. Glosario

**Actor.** Algo con comportamiento (persona, otro programa, organización..), que interactúa con el sistema.

**Administrador.** Usuario del sistema con privilegios especiales sobre el módulo de mantenimiento que está encargado de realizar la gestión de los datos maestros de la aplicación.

**Arquitectura del software.** Es el diseño de más alto nivel de la estructura de un sistema.

**B2C.** Acrónimo de la expresión **Business-to-Consumer** («del negocio al consumidor», en inglés). Se refiere a la estrategia que desarrollan las empresas comerciales para llegar directamente al cliente o consumidor final.

**Casos de uso.** Es una descripción de las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores. En el contexto de ingeniería del software, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

**Cliente.** Usuario del sistema que interactúa con el módulo de ventas de la aplicación.

**DCD.** Diagrama de clases de diseño que representa las especificaciones de las clases e interfaces de software en una aplicación.

**Diagrama de secuencia.** Es un tipo de diagrama usado para modelar interacción entre objetos en un sistema según UML.

**Diagrama de secuencia de sistema.** Muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada **caso de uso**. Permite el modelado de una vista *business* del escenario.

**Diagramas de secuencia de diseño.** Muestra detalles de implementación del escenario, incluyendo los **objetos y clases** que se usan para implementar el escenario y mensajes intercambiados entre los objetos.

**Diagrama de E/R:** A veces denominado por sus siglas en inglés, E-R "Entity relationship", o del español DER "Diagrama de Entidad Relación") es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades.

**Escenario:** En UML es una circunstancia o situación, en la que se puede encontrar un sistema. Los escenarios, definen situaciones en las que se encuentra un sistema, esto quiere decir que, el conjunto de escenarios, definen el sistema, o al menos, lo que puede hacer el sistema, por lo que nos sirve para definir, que ha de hacer y que no ha de hacer un sistema.

**GRASP.** Acrónimo de *General Responsibility Assignment Software Patterns*, y son patrones de diseño relacionados con asignación de responsabilidades a los objetos.

**MDA.** Acrónimo de *Model-Driven Architecture*, es una arquitectura que proporciona un conjunto de guías para estructurar especificaciones expresadas como modelos. La funcionalidad del sistema será definida en primer lugar como un modelo independiente de la plataforma técnica.

**MVC.** Acrónimo de *Model View Controller* (Modelo, Vista, Controlador. Patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos.

**OO.** Acrónimo de *Object-Oriented*, es una filosofía de la ingeniería de software que afecta a buena parte del proceso de desarrollo de software. Trata de reflejar objetos reales en objetos de software. Los objetos se comunicarán entre ellos, transmitiendo órdenes, información o referencias a otros objetos.

**PMBOK.** Acrónimo de *Project Managment Body of Knowledge*, se trata de la *Guía de los Fundamentos de Gestión de Proyectos*. Es un libro en el que se presentan estándares, pautas y normas a seguir para uan adecuada gestión de proyectos.

**UAT.** Acrónimo de *User acceptance testing* , es una prueba llevada a cabo para determinar si los requisitos de una especificación o contrato se cumplen en base a los casos de uso.

**UML.** Acrónimo de *Unified Model Language*. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

**UP.** Acrónimo de *Unified Process*. Es un marco de desarrollo de software que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental.

## 7. Bibliografía

Material de las asignaturas:

Enginyeria de programari.

Bases de dades.

Disseny d'estructures de dades.

Enginyeria de requisits.

Anàlisi i disseny amb patrons.

Enginyeria del programari de components i sistemes distribuïts.

Projecte de desenvolupament del programari.

Craig Larman, UML y Patrones 2ª edición.

Object Management Group (<http://www.uml.org/>).

Wikipedia.



## **8. Anexos**