

**UOC TRABAJO FIN DE CARRERA ITIG**  
**PROYECTO DE CATÁLOGO DE PRODUCTOS**  
**DESARROLLADO EN .NET**  
**"Surf eCatalogo"**

**Alumna:**  
**María Odriozola Astigarraga**  
**ITIG**

**Consultor:**  
**David Gañan Jiménez**

**10 Junio 2007**

*Gracias por el esfuerzo,  
perdona las exigencias,  
olvida las frustraciones y  
recuerda los triunfos.*

*A ti, MAJA*

## Resumen

El objetivo de este proyecto fin de carrera ha sido el diseño e implementación de un catálogo de productos, en concreto un catálogo de productos de artículos relacionados con el surf "Surf eCatalogo".

La solución consta de dos aplicaciones:

La aplicación de gestión y mantenimiento "**aplicación escritorio**" que se utilizará por el personal de la empresa para:

- ✚ Crear un nuevo cliente o modificar los datos existentes, así mismo poder consultar los datos referentes a los clientes existentes.
- ✚ Poder insertar nuevos productos en el catálogo, así como las categorías correspondientes. El sistema nos permitirá así mismo realizar el mantenimiento de dichas categorías y productos, lo que permitirá por ejemplo modificar los precios de venta-
- ✚ Al dar de alta un producto definiremos un stock mínimo de seguridad, que podremos controlar y consultar desde el "Control del stock"
- ✚ Realizar la actualización y control de los pedidos realizados por los clientes a través de la web.

La aplicación cliente "**aplicación web**" que constará de una parte pública a la que podrá acceder cualquier usuario, y una parte privada a la que podrán acceder los usuarios registrados en el sistema.

- ✚ El parte pública todo usuario podrá realizar consultas en el catálogo de productos, registrarse como un nuevo usuario.
- ✚ En la parte privada, el usuario registrado accederá mediante la introducción de clave y contraseña, una vez validado podrá consultar los pedidos en curso, así como a la realización de nuevos pedidos (pendiente de implementar).

## INDICE

<b>I. INTRODUCCIÓN</b>	<b>5</b>
1. JUSTIFICACIÓN DEL TFC	5
2. OBJETIVOS DEL TFC	5
3. ENFOQUE Y MÉTODO SEGUIDO	6
4. TECNOLOGÍAS UTILIZADAS	7
5. PLANIFICACIÓN DEL PROYECTO	11
6. PRODUCTOS OBTENIDOS	14
7. BREVE INTRODUCCIÓN DE OTROS CAPÍTULOS	14
<b>II. CUERPO DE LA MEMORIA</b>	<b>15</b>
<b>A. ANÁLISIS DEL PROYECTO</b>	<b>15</b>
1. DESCRIPCIÓN	15
2. ANÁLISIS DE REQUISITOS Y FUNCIONALIDADES	16
3. DIAGRAMA DE CASOS DE USO	19
4. MODELO CONCEPTUAL DE CLASES	32
<b>B. DISEÑO</b>	<b>34</b>
1. ARQUITECTURA DEL SISTEMA EN N-CAPAS	34
2. DIAGRAMAS DE SECUENCIA	36
3. DIAGRAMA DE CLASES DEL DISEÑO	38
4. DISEÑO DE LA INTERFAZ GRÁFICA	39
5. DISEÑO DE LA BASE DE DATOS	49
<b>C. IMPLEMENTACIÓN</b>	<b>52</b>
<b>III. CONCLUSIÓN</b>	<b>52</b>
1. CONCLUSIÓN Y OBJETIVOS ALCANZADOS	55
2. AMPLIACIONES Y MEJORAS DEL PROYECTO	55
<b>IV. GLOSARIO</b>	<b>57</b>
<b>V. BIBLIOGRAFIA</b>	<b>58</b>

# I. INTRODUCCIÓN

---

## 1. JUSTIFICACIÓN DEL TFC

El trabajo fin de carrera esta basado en la elaboración de un Proyecto como ejercicio integrador o de síntesis de la formación recibida a lo largo de la carrera. Siendo éste, un trabajo práctico, vinculado al ejercicio profesional de la informática.

El objetivo principal del trabajo es alcanzar un buen conocimiento de la plataforma .NET de Microsoft, concretamente utilizando el lenguaje de programación Visual Basic y la tecnología ADO.NET de acceso a datos. Así como consolidar los conocimientos adquiridos en las distintas asignaturas de la carrera que afectan al análisis y desarrollo de proyectos informáticos tales como:

- ✚ Ingeniería de software
- ✚ Programación orientada a objetos
- ✚ Bases de datos I y II
- ✚ Sistemas de gestión de bases de datos
- ✚ Técnicas de desarrollo de software.

El hecho de que el objetivo central sea aprender estas tecnologías nos obliga a buscar una aplicación sencilla, de manera que se ha simplificado, en la medida de lo posible, el futuro diseño de clases y estructura de base de datos para centrarnos en la parte que más nos interesa: la implementación.

## 2. OBJETIVOS DEL TFC

La aplicación que se ha escogido para desarrollar permite la gestión de un **catalogo de productos**, donde la empresa gestiona los artículos, su stock y los pedidos recibidos, los clientes pueden consultar el catálogo y realizar sus pedidos a través de Internet. Para ello se ha dividido la aplicación en varios subsistemas, separando claramente las tareas que habrá que realizar en el lado de la empresa, de las tareas que desempeñará el cliente Web.

En los siguientes capítulos se explicará de forma detallada las diferentes etapas que se seguirán para desarrollar el trabajo.

### **3. ENFOQUE Y MÉTODO SEGUIDO**

El enfoque y el método seguido para el desarrollo del trabajo, viene en gran parte delimitado por las indicaciones y propuestas realizadas por el tutor en plazos y entregas.

El proyecto se ha dividido en cuatro fases diferenciadas:

- ✚ Planificación
- ✚ Análisis de requisitos y funcionalidades de la aplicación
- ✚ Diseño
- ✚ Implementación de la aplicación escritorio
- ✚ Implementación de la aplicación web

En los procesos de desarrollo de los productos previstos en el trabajo se han utilizado los métodos, técnicas y herramientas aprendidos en el curso de los estudios y que más adecuados resultaban para obtener cada uno de los productos a desarrollar:

- ✚ Parte fundamental de la planificación y análisis se ha realizado con los métodos aprendidos en la asignatura de Técnicas de Desarrollo del Software.
- ✚ Se han aplicado los métodos y técnicas de diseño de bases de datos relacionales aprendidos en la asignatura Bases de Datos I (diagramas Entidad/Relación, diseño lógico, normalización de tablas, etc.) .
- ✚ Se han empleado los métodos adquiridos en Ingeniería de Software I, Técnicas de Desarrollo de Software y las técnicas aprendidas en las diferentes asignaturas de programación y Base de Datos II para el desarrollo de aplicaciones orientadas a objetos.
- ✚ Finalmente, he tenido que aprender nuevas tecnologías que desconocía: Visual Studio 2005, SQL Server, WPF, ASP.NET, ADO.NET y AJAX.

El seguimiento de este enfoque y la aplicación de los métodos previstos ha permitido la consecución exitosa de los objetivos del trabajo final de carrera.

Antes de detallar el plan de desarrollo de la aplicación, se hará un breve resumen explicativo de las tecnologías usadas en él, basadas, todas ellas en la plataforma .NET de Microsoft, como acceso a datos ADO.NET y ASP.NET.

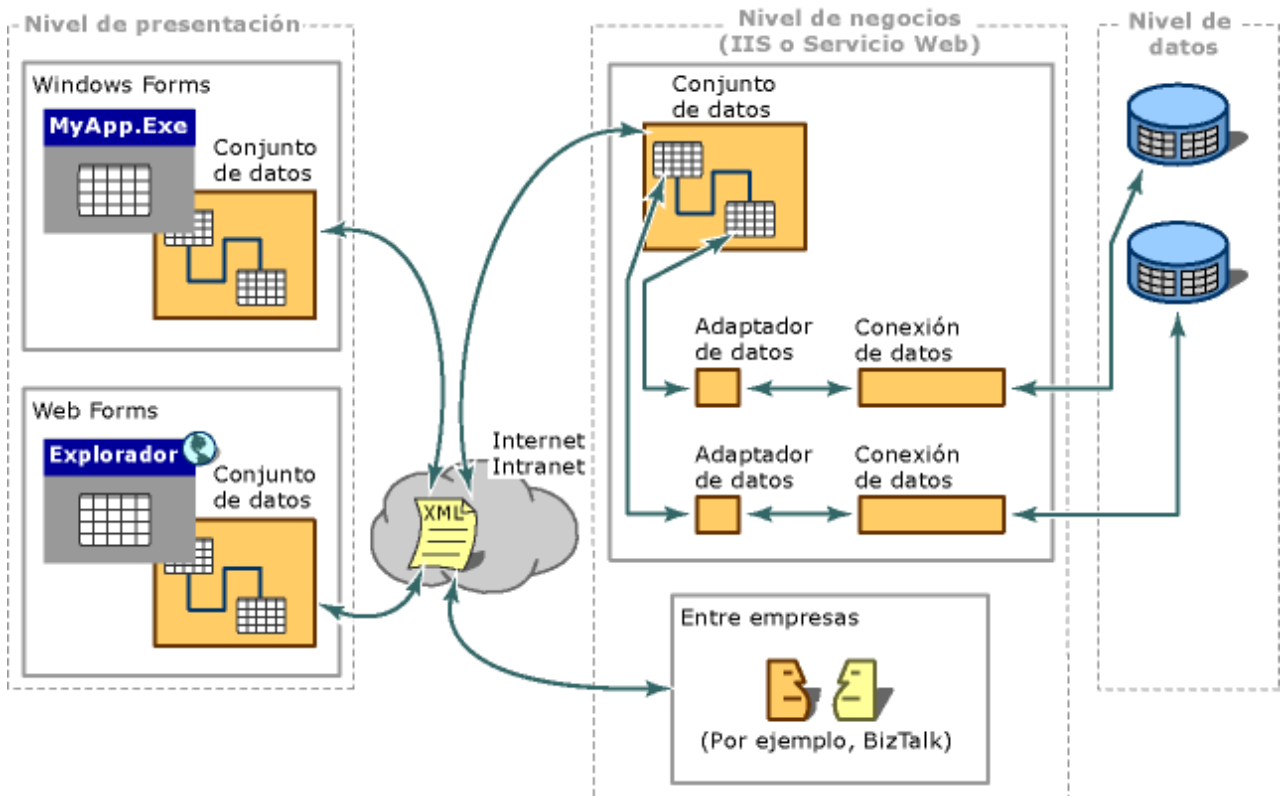
#### **4. TECNOLOGÍAS UTILIZADAS**

✚ **Microsoft.NET** es la arquitectura o conjunto de nuevas tecnologías lanzadas por Microsoft con el objetivo de mejorar tanto su sistema operativo como su modelo de componentes para obtener una plataforma con la que sea sencillo el desarrollo de software en forma de servicios web.

✚ La tecnología **ADO.NET** es la solución propuesta por Microsoft para permitir el acceso a bases de datos desde su plataforma .NET. Lo que hacen los objetos ADO (Objetos de Datos ActiveX) es permitir que una aplicación escrita en un lenguaje de programación cualquiera, pueda comunicarse con una base de datos usando el SQL (Structured Query Language). Se podría decir que ADO.NET es un conjunto de clases que exponen servicios de acceso a datos al programador de .NET.

Un aspecto importante de ADO.NET es el hecho de que utilice un modelo de acceso a datos pensado para entornos desconectados. Esto quiere decir que la aplicación se conecta al origen de datos, realiza las operaciones que necesite, carga en memoria los resultados obtenidos utilizando variables y se desconecta del origen de datos.

La siguiente ilustración muestra los componentes principales de una aplicación ADO.NET.



✚ **ASP.NET** es el componente de Microsoft.Net totalmente orientado al desarrollo de aplicaciones Web. El funcionamiento de esta tecnología es el siguiente: cuando el usuario, desde su navegador, realiza una petición a una página ASP.NET, el servidor devuelve una página html generada en ese mismo momento, basándose en el código ASP.NET alojado en el servidor. Esto permite alcanzar un grado de interactividad y personalización total entre la aplicación y el usuario.

Algunas de las posibilidades más interesantes que nos permite el ASP.NET son el poder acceder a bases de datos o generar gráficos y correos electrónicos de forma dinámica.

✚ **Visual Basic .NET** es el lenguaje de programación creado por Microsoft que he considerado conveniente utilizar para la implementación del proyecto, debido a que es un lenguaje con el que me encuentro bastante familiarizada por mi experiencia en versiones anteriores (Visual Basic 6).



✚ **Windows Presentation Foundation:** Se ha optado por incluir la tecnología Windows Presentation Foundation dada la eficacia para la creación de interfaces que ofrecen beneficios apreciables a los usuarios y las organizaciones que se sirven de ellas. El objetivo principal de WPF es ayudar a los desarrolladores a convertir estos beneficios potenciales en realidad.

WPF ofrece una plataforma unificada para interfaces de usuario modernas, convierte a los diseñadores en participantes activos a la hora de crear estas interfaces y proporciona un modelo de programación común para aplicaciones independientes y de explorador. Con todo ello, WPF contribuirá a mejorar considerablemente la experiencia de los usuarios de Windows. Algunas de las tecnologías que reemplaza fueron base de interfaces de usuario de Windows durante veinte años. La intención de WPF es establecer una base para los próximos veinte años.

Los usuarios dan gran importancia a las interfaces. La interfaz de una aplicación constituye una parte fundamental de la experiencia global del usuario con el software particular. En lo que respecta a los usuarios, la aplicación es la experiencia. La experiencia mejorada de los usuarios mediante una interfaz optimizada puede contribuir al incremento de la productividad, a la generación de clientes leales y a una ampliación de las ventas en línea, entre muchas otras ventajas.

Los usuarios, que un día se conformaban con interfaces basadas en caracteres, ahora se han acostumbrado a las interfaces gráficas. No obstante, los requisitos que deben cumplir estas interfaces siguen aumentando. El uso de gráficos y componentes multimedia se ha hecho más generalizado. Además, la Web ha condicionado a una generación de usuarios que esperan obtener una interacción con software sencilla. La relevancia de las interfaces de las aplicaciones crece con el aumento del tiempo que los usuarios dedican a interactuar con las mismas. Para satisfacer las crecientes expectativas, la tecnología de creación de interfaces de usuario debe avanzar a la par.

El objetivo de Windows Presentation Foundation (WPF) es proporcionar estos avances en el entorno de Windows. WPF se incluye en la versión 3.0 de Microsoft .NET Framework y permite crear interfaces que incorporan documentos, componentes multimedia, gráficos bidimensionales y tridimensionales, animaciones, características tipo web, etc.

✚ **AJAX:** Se ha optado por incluir la tecnología Asynchronous JavaScript And XML (JavaScript y XML asíncronos, donde XML es un acrónimo de eXtensible Markup Language), es una técnica de desarrollo web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

## **5. PLANIFICACIÓN DEL PROYECTO**

### ***1. Descomposición de las fases***

---

El método seguido para la realización del trabajo ha venido marcado por las limitaciones de tiempo y la necesidad de cumplir los plazos de entregas. Este hecho, unido a las entregas intermedias que se tenían que hacer de las diferentes Pecs, ha provocado la decisión de dividir el proyecto en tres grandes bloques o tareas:

- ✚ Obtención de requisitos, análisis y diseño de la aplicación.
- ✚ Implementación de la interfaz de usuario.
- ✚ Implementación de las funcionalidades de la aplicación

Dentro de cada uno de estos tres bloques se ha seguido un proceso paralelo en su realización, debido a la existencia de dos tipos de usuarios muy diferenciados por las tareas que realizarán en la aplicación: los gestores y los clientes. A grandes rasgos se podría decir que las tareas de los gestores serán las de definir al resto de usuarios, y el mantenimiento de los ficheros maestros (ABM de artículos y clientes), se encargarán del seguimiento de los pedidos y el control del stock y los clientes, lógicamente, serán los que consultarán el catalogo de productos y podrán realizar los pedidos que deseen, así como modificar sus datos, todo ello previa identificación de los mismos.

1º Tareas relacionadas con los gestores

2º Tareas relacionadas con los clientes

Esto facilitará a su vez, la etapa de pruebas que se realizarán en cada una de las fases de la implementación.

## 2. Calendario

Id	Nombre de tarea	Duración	Comienzo	Fin
1	<b>Proyecto ePaquet</b>	<b>103 días</b>	<b>jue 01/03/07</b>	<b>lun 11/06/07</b>
2	<b>Fase 1: Plan de Trabajo y Análi</b>	<b>13 días</b>	<b>jue 01/03/07</b>	<b>mar 13/03/07</b>
3	Justificación del TFC	2 días	jue 01/03/07	vie 02/03/07
4	Estudio del Software	3 días	sáb 03/03/07	lun 05/03/07
5	Definición de Tecnología	1 día	mar 06/03/07	mar 06/03/07
6	Planificación Trabajo	2 días	mié 07/03/07	jue 08/03/07
7	Análisis y Requisitos	2 días	vie 09/03/07	sáb 10/03/07
8	Entrega PEC1	3 días	dom 11/03/07	mar 13/03/07
9	<b>Fase 2: Especificación y Diseñ</b>	<b>27 días</b>	<b>mié 14/03/07</b>	<b>lun 09/04/07</b>
10	Diagrama de clases	5 días	mié 14/03/07	dom 18/03/07
11	Casos de uso	5 días	lun 19/03/07	vie 23/03/07
12	Diagrama ER	3 días	sáb 24/03/07	lun 26/03/07
13	Persistencia	3 días	mar 27/03/07	jue 29/03/07
14	Elaboración del documento	5 días	vie 30/03/07	mar 03/04/07
15	Revisión Final	4 días	mié 04/04/07	sáb 07/04/07
16	Entrega PEC2	2 días	dom 08/04/07	lun 09/04/07
17	<b>Fase 3: Implementación</b>	<b>49 días</b>	<b>mar 10/04/07</b>	<b>lun 28/05/07</b>
18	Script de creación de la BD	2 días	mar 10/04/07	mié 11/04/07
19	Implementación del proyecto	30 días	jue 12/04/07	vie 11/05/07
20	Pruebas de verificación	4 días	sáb 12/05/07	mar 15/05/07
21	Revisión Final	7 días	mié 16/05/07	mar 22/05/07
22	Entrega	6 días	mié 23/05/07	lun 28/05/07
23	<b>Memoria y Manual de Instalci</b>	<b>10 días</b>	<b>mar 29/05/07</b>	<b>jue 07/06/07</b>
24	Elaboración de la memoria	2 días	mar 29/05/07	mié 30/05/07
25	Elaboración del Manual de ins	1 día	jue 31/05/07	jue 31/05/07
26	Revisión	7 días	vie 01/06/07	jue 07/06/07
27	<b>Presentación</b>	<b>4 días</b>	<b>vie 08/06/07</b>	<b>lun 11/06/07</b>
28	Preparación presentación	2 días	vie 08/06/07	sáb 09/06/07
29	Revisión	1 día	dom 10/06/07	dom 10/06/07
30	Entrega Final	1 día	lun 11/06/07	lun 11/06/07



## **6. PRODUCTOS OBTENIDOS**

Como resultado del trabajo se han obtenido:

- ✚ Documentación relativa al análisis de requisitos y funcionalidades y diseño de las aplicaciones.
- ✚ La aplicación escritorio implementada.
- ✚ La aplicación Web implementada.

## **7. BREVE INTRODUCCIÓN DE OTROS CAPÍTULOS**

En los siguientes capítulos se explicará de forma detallada las diferentes etapas que se han seguido para desarrollar el trabajo.

El detalle de la aplicación empezará por dar una visión general de las funcionalidades esperadas de ella, lo que nos ha de conducir a detallar de una forma más concreta sus requisitos funcionales y de usabilidad hallados durante la fase de análisis. Estos requisitos funcionales desembocarán en la redacción de los casos de uso requeridos por la aplicación.

En el siguiente capítulo se detallará el diseño de la aplicación, concretamente el diseño de la implementación de los casos de uso, el diseño de la persistencia y el diseño de la interfaz de usuario.

Y finalmente se explicarán las conclusiones resultantes del trabajo, así como futuras mejoras y añadidos que serían posibles para completar la aplicación obtenida.

## II. CUERPO DE LA MEMORIA

---

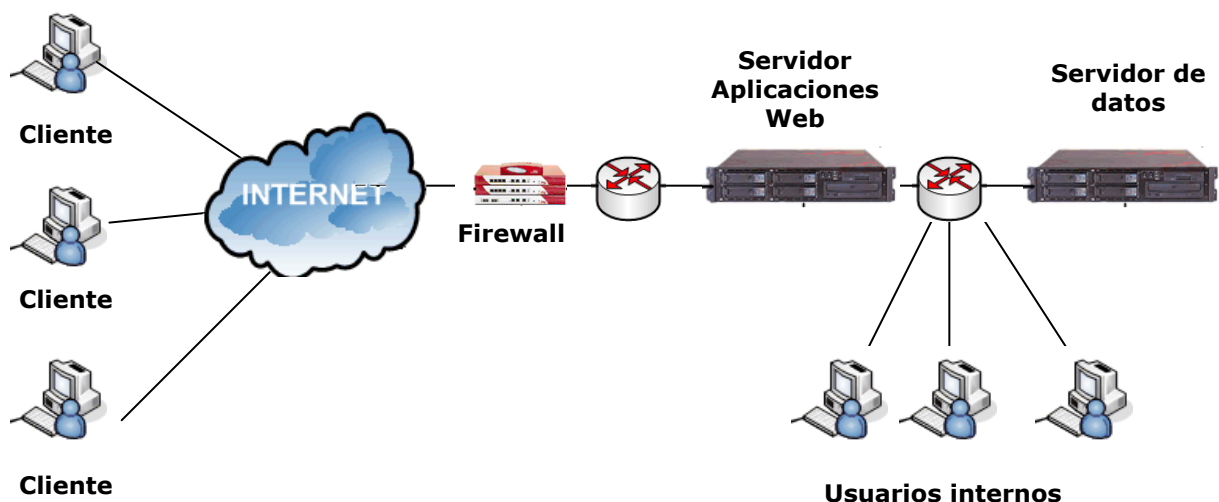
### A. ANÁLISIS DEL PROYECTO

---

En el análisis de requisitos y funcionalidades del proyecto se ha diferenciado entre las funcionalidades básicas del proyecto (limitadas por las características y tiempo necesario para el desarrollo del mismo) y las funcionalidades ampliadas, aquellas funcionalidades que se irían incorporando a medida que el proyecto se fuera desarrollando (funcionalidades que no se desarrollan en este proyecto).

#### 1. DESCRIPCIÓN

El sistema de información "Catalogo online" pretende ser un sistema informático mediante el cual se puedan llevar a cabo la gestión de pedidos de un catalogo de productos. La aplicación constará de tres programas, ejecutados en un número indeterminado de ordenadores. Todos los ordenadores deberán estar conectados en red (local o Internet) con un ordenador central que almacenará los datos. A continuación profundizaremos en los detalles.



Para empezar, la empresa distribuidora del catálogo deberá introducir los datos de los productos, el stock, los usuarios y hacer el mantenimiento de los mismos. Es

imprescindible introducir todos los datos necesarios para el correcto funcionamiento del programa. Para ello dispondrá de un programa específico para tal fin.

## 2. ANÁLISIS DE REQUISITOS Y FUNCIONALIDADES

### I. FUNCIONALIDADES BÁSICAS:

#### A. Subsistema operador:

Este subsistema tendrá una serie de opciones para realizar el mantenimiento de los datos de los ficheros maestros de productos y clientes de la empresa.

Funcionalidades:

- **Alta, baja / modificación del producto:** opción que permite dar de alta, baja o modificar un producto en el sistema.
- **Consulta de productos:** mediante esta opción se visualizan los productos que se encuentran almacenados en el sistema con distintos criterios de selección.
- **Entrada de unidades de productos a stock:** esta opción permita introducir el detalle del stock mínimo por producto, se accede a la misma en el alta o consulta de un producto.
- **Alta /Baja/ modificación de una categoría:** opción que permite dar de alta, baja o modificar una categoría en nuestro sistema.
- **Consulta de categorías:** mediante esta opción se visualizan las categorías definidas.
- **Alta /Baja/ modificación del cliente:** opción que permite dar de alta, baja o modificar los datos un nuevo cliente en nuestro sistema.
- **Consulta de clientes:** mediante esta opción se visualizan los clientes activos en el sistema.

#### B. Subsistema de gestión:

Este subsistema tendrá una serie de opciones para realizar el seguimiento de los pedidos por parte del personal de gestión de la empresa distribuidora.



Funcionalidades:

- **Consulta/Listado de stock de productos:** la finalidad de esta opción es obtener un listado de los productos mostrando el stock, en la misma se puede actualizar el stock actual y el stock mínimo de los productos.
- **Seguimiento de pedidos:**
  - o **Consulta/listado pedidos pendientes de entregar:** opción que muestra, los datos más relevantes de los pedidos pendientes de entrega.

### C. Subsistema cliente:

Este subsistema será el que se encontrará accesible a través de Internet para que los clientes, tras previa identificación puedan realizar sus pedidos.

Funcionalidades:

- **Consulta de catálogo de productos :** el cliente visualiza los productos del catálogo de forma ordenada y/o agrupada por categorías
- **Identificación del cliente:** esta opción permite al cliente remoto identificarse para tener acceso a nuestro sistema.
- **Alta nuevo cliente:** opción que permite a un cliente nuevo, darse de alta en nuestro sistema.
- **Consulta de pedidos realizados:** el cliente puede consultar los pedidos realizados con anterioridad que figuran en nuestro sistema.

## II. FUNCIONALIDADES AMPLIADAS:

### A. Subsistema de gestión:

Adicionalmente este subsistema tendrá una serie de opciones para realizar el seguimiento de los pedidos por parte del personal de gestión de la empresa distribuidora.

Funcionalidades:

- **Seguimiento de pedidos:**
  - o **Entrega de pedido:** opción que modifica el estado del pedido del cliente como entregado

## B. Subsistema cliente:

Este subsistema será el que se encontrará accesible a través de Internet para que los clientes, tras previa identificación puedan realizar sus pedidos.

Funcionalidades:

- **Modificación de datos personales:** opción que permite a un cliente modificar sus datos personales después de haberse identificado.
- **Realización de nuevo pedido:** opción que permite, al cliente hacer un nuevo pedido de productos, que tras aceptarlo llegará a registrarse en el sistema para que se pueda proceder a la entrega.
- **Búsqueda/Consulta de productos:** permite al cliente localizar un producto en el sistema tras haber indicado un criterio de búsqueda correcto.

### 3. DIAGRAMA DE CASOS DE USO

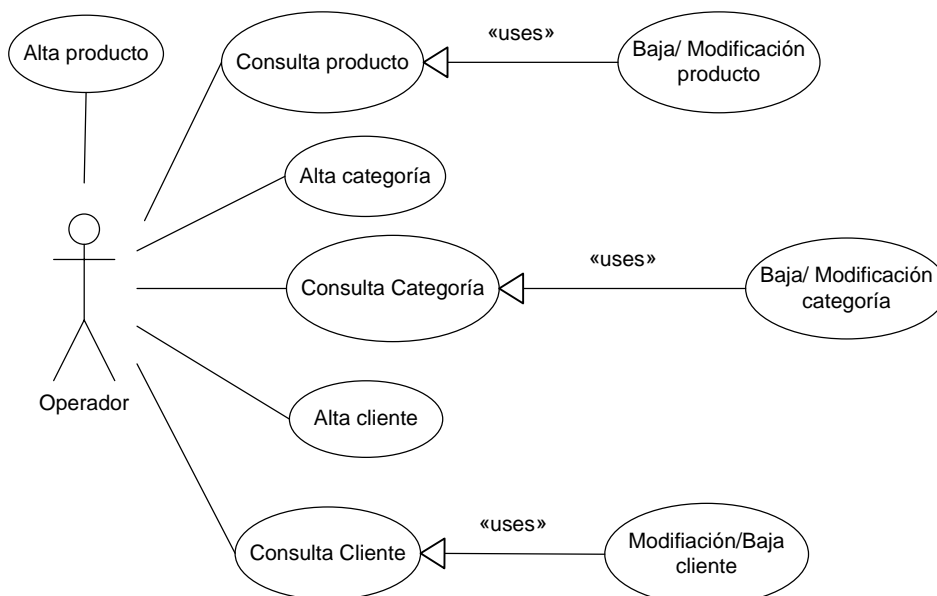
Una vez analizadas las funcionalidades y requisitos del sistema, construimos un diagrama de casos de uso sobre la base de las actividades o funcionalidades de cada actor. Se han definido tres actores, correspondiendo con los tres subsistemas:

- Cliente (subsistema cliente)
- Gestor (subsistema de gestión)
- Operador (subsistema de mantenimiento)

Podemos considerar cada funcionalidad como un caso de uso y el agente que la realiza como el actor del caso de uso.

#### I. FUNCIONALIDADES BÁSICAS:

##### A. Operador:



**+ Alta del producto:**

<b>Caso de uso:</b> Alta Producto
<b>Objetivo:</b> Dar de alta un nuevo producto en el sistema
<b>Actores:</b> Operador
<b>Precondiciones:</b> El producto no esta dado de alta en el sistema
<b>Casos de uso relacionados:</b>
<b>Pasos:</b> 1. Se introducen los datos del nuevo producto 2. Se relaciona el producto con una categoría concreta existente 3. El sistema valida que los datos introducidos son correctos y que el producto no existe previamente.
<b>Variaciones:</b> 2. a La categoría no existe y se crea una nueva 4. El sistema no valida los datos: a. Ya existe un producto igual b. Los datos introducidos no son correctos
<b>Poscondiciones:</b> Se ha creado un nuevo producto en el sistema
<b>Extensiones:</b>

**+ Baja o modificación del producto:**

<b>Caso de uso:</b> Baja/ Modificación Producto
<b>Objetivo:</b> Modificar los atributos de un producto o darlo de baja.
<b>Actores:</b> Operador
<b>Precondiciones:</b> El producto ya existe en el sistema
<b>Casos de uso relacionados:</b> Consulta producto
<b>Pasos:</b> 1.a Se selecciona un producto que se desea modificar 1, b Se selecciona el producto que se desea dar de baja 2. a. Se modifican los atributos del producto 3. a Se confirma la modificación de los atributos del producto 3. b Se confirma la eliminación del producto
<b>Variaciones:</b>
<b>Poscondiciones:</b> Se ha modificado o eliminado un producto en el sistema
<b>Extensiones:</b>

**✚ Consulta de un producto:**

<b>Caso de uso:</b> Consulta Producto
<b>Objetivo:</b> Se introduce el nombre o referencia del producto para poder ver el detalle del mismo.
<b>Actores:</b> Operador
<b>Precondiciones:</b> No hay
<b>Casos de uso relacionados:</b> baja o modificación producto
<b>Pasos:</b> 1. Indicar el producto que se quiere visualizar. 2. a El sistema localiza el producto y lo muestra. 2. b El sistema no localiza el producto y muestra el mensaje de error
<b>Variaciones:</b>
<b>Poscondiciones:</b> El sistema muestra el producto solicitado
<b>Extensiones:</b> Baja o modificación producto

**✚ Alta de categoría:**

<b>Caso de uso:</b> Alta Categoría
<b>Objetivo:</b> Dar de alta una nueva categoría en el sistema
<b>Actores:</b> Operador
<b>Precondiciones:</b> La categoría no esta dada de alta en el sistema
<b>Casos de uso relacionados:</b> baja o modificación categoría, consulta categoría
<b>Pasos:</b> 1. Se introducen los datos de la nueva categoría 2. El sistema valida que los datos introducidos son correctos y que la categoría no existe previamente.
<b>Variaciones:</b> 2. El sistema no valida los datos: a. Ya existe una categoría igual b. Los datos introducidos no son correctos
<b>Poscondiciones:</b> Se ha creado una nueva categoría en el sistema
<b>Extensiones:</b> alta producto

**✚ Baja o modificación de una categoría:**

<b>Caso de uso:</b> Baja/ Modificación Categoría
<b>Objetivo:</b> Modificar los atributos de una categoría o la da de baja.
<b>Actores:</b> Operador
<b>Precondiciones:</b> La categoría ya existe en el sistema

<b>Casos de uso relacionados:</b> Consulta categoría
<b>Pasos:</b> 1.a Se selecciona una categoría que se desea modificar 1, b Se selecciona la categoría que se desea dar de baja 2. a. Se modifican los atributos de la categoría 3. a Se confirma la modificación de los atributos de la categoría 3. b Se confirma la eliminación de la categoría
<b>Variaciones:</b>
<b>Poscondiciones:</b> Se ha modificado o eliminado una categoría en el sistema
<b>Extensiones:</b>

### ✚ Consulta de una categoría:

<b>Caso de uso:</b> Consulta Categoría
<b>Objetivo:</b> Se introduce el nombre o referencia de una categoría para poder ver el detalle de la misma.
<b>Actores:</b> Operador
<b>Precondiciones:</b> No hay
<b>Casos de uso relacionados:</b> baja o modificación categoría
<b>Pasos:</b> 1. Indicar la categoría que se quiere visualizar. 2. a El sistema localiza la categoría y la muestra. 2. b El sistema no localiza la categoría y muestra el mensaje de error
<b>Variaciones:</b>
<b>Poscondiciones:</b> El sistema muestra la categoría buscada
<b>Extensiones:</b> Baja o modificación categoría

### ✚ Alta clientes:

<b>Caso de uso:</b> Alta Cliente
<b>Objetivo:</b> Dar de alta a un nuevo cliente en el sistema
<b>Actores:</b> Operador
<b>Precondiciones:</b> El cliente no esta dado de alta en el sistema
<b>Casos de uso relacionados:</b> consulta cliente, modificación/baja cliente
<b>Pasos:</b> 1. Se introducen los datos del nuevo cliente 2. El sistema valida que los datos introducidos son correctos y que el cliente no existe previamente.
<b>Variaciones:</b> 2. El sistema no valida los datos: a. Ya existe el cliente b. Los datos introducidos no son correctos
<b>Poscondiciones:</b> Se ha creado una nuevo cliente en el sistema
<b>Extensiones:</b>

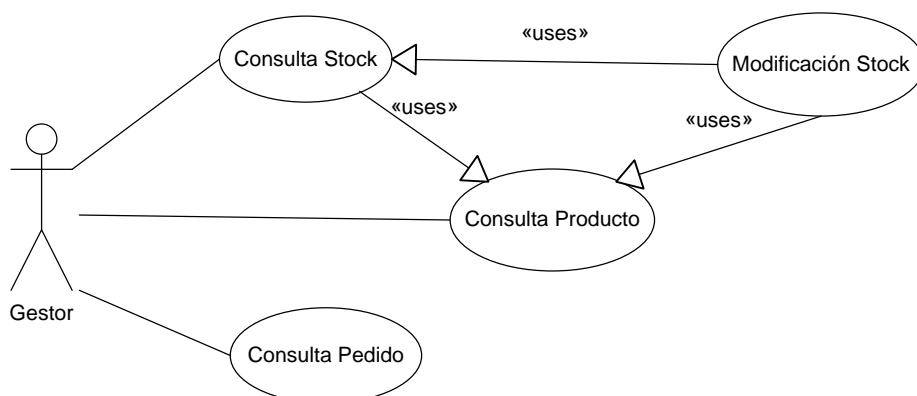
**✚ Baja o modificación de un cliente:**

<b>Caso de uso:</b> Baja/ Modificación Cliente
<b>Objetivo:</b> Modificar los atributos de un cliente o darlo de baja.
<b>Actores:</b> Operador
<b>Precondiciones:</b> El cliente ya existe en el sistema
<b>Casos de uso relacionados:</b> Consulta cliente
<b>Pasos:</b> 1.a Se selecciona el cliente del que se desean modificar datos 1, b Se selecciona el cliente que se desea dar de baja 2. a. Se modifican los atributos del cliente 3. a Se confirma la modificación de los atributos del cliente 3. b Se confirma la eliminación del cliente
<b>Variaciones:</b>
<b>Poscondiciones:</b> Se ha modificado o eliminado un cliente
<b>Extensiones:</b>

**✚ Consulta de un cliente:**

<b>Caso de uso:</b> Consulta Cliente
<b>Objetivo:</b> Se introduce el nombre o identificador de un cliente para ver sus datos.
<b>Actores:</b> Operador
<b>Precondiciones:</b> No hay
<b>Casos de uso relacionados:</b> baja o modificación del cliente
<b>Pasos:</b> 1. Indicar el cliente que se quiere visualizar. 2. a El sistema localiza el cliente y lo muestra. 2. b El sistema no localiza el cliente y muestra el mensaje de error
<b>Variaciones:</b>
<b>Poscondiciones:</b> El sistema muestra el cliente buscado
<b>Extensiones:</b> Baja o modificación cliente

**B. Gestor:**



**✚ Consulta de un producto:**

<b>Caso de uso:</b> Consulta Producto
<b>Objetivo:</b> Se introduce el nombre o referencia del producto para poder ver el detalle del mismo.
<b>Actores:</b> Gestor
<b>Precondiciones:</b> No hay
<b>Casos de uso relacionados:</b> consulta stock, modificación stock
<b>Pasos:</b> 1. Indicar el producto que se quiere visualizar. 2. a El sistema localiza el producto y lo muestra. 2. b El sistema no localiza el producto y muestra el mensaje de error
<b>Variaciones:</b>
<b>Poscondiciones:</b> El sistema muestra el producto solicitado
<b>Extensiones:</b> Modificación stock, consulta stock

**✚ Consulta del stock:**

<b>Caso de uso:</b> Consulta Stock
<b>Objetivo:</b> El usuario quiere conocer el nivel de stock de los productos
<b>Actores:</b> Gestor
<b>Precondiciones:</b> No hay
<b>Casos de uso relacionados:</b> Consulta Producto, Modificación Stock
<b>Pasos:</b> 1. a. El usuario accede al nivel de stock de todos los productos. 1. b. El usuario tras haber realizado la búsqueda de un producto, consulta el nivel de stock del mismo.
<b>Variaciones:</b>
<b>Poscondiciones:</b> El sistema muestra el nivel de stock de todos los productos, o del producto concreto seleccionado
<b>Extensiones:</b> Modificación Stock

**✚ Modificación del stock:**

<b>Caso de uso:</b> Modificación Stock
<b>Objetivo:</b> El gestor modifica el nivel de stock correspondiente a un producto.
<b>Actores:</b> Gestor
<b>Precondiciones:</b> El producto tiene que existir en el sistema
<b>Casos de uso relacionados:</b> Consulta stock, Consulta producto



**Pasos:**

1. a. El usuario desde la pantalla que le muestra el nivel de stock de un producto realizar la modificación del nivel de stock.
1. b. El usuario sobre la consulta del producto realiza la modificación del nivel de stock (una entrada o salida)
2. Aceptan y guardan la modificación

**Variaciones:**

**Poscondiciones:** El sistema guarda el nuevo nivel del stock del producto modificado.

**Extensiones:**
**+ Consulta de un pedido:**

**Caso de uso:** Consulta Pedido

**Objetivo:** Se quiere conocer el estado y datos de un pedido concreto.

**Actores:** Gestor

**Precondiciones:** No hay.

**Casos de uso relacionados:** estado pedido

**Pasos:**

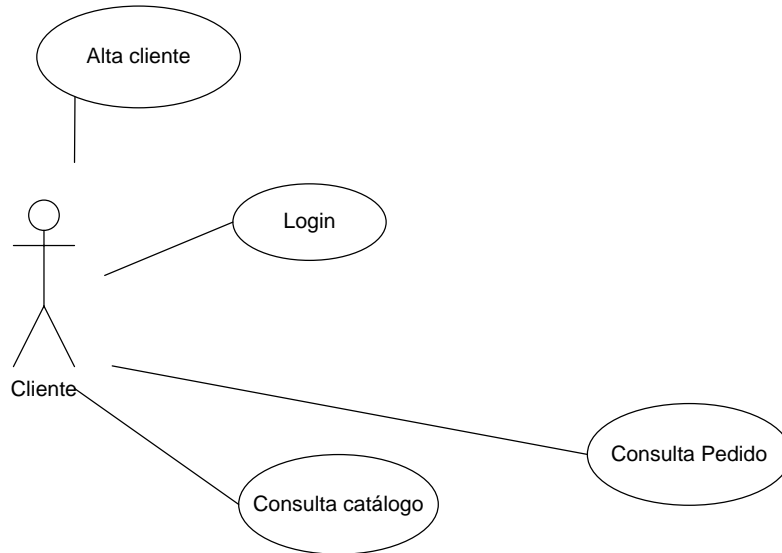
1. Indicar el pedido que se quiere visualizar.
2. a El sistema localiza el pedido y lo muestra.
2. b El sistema no localiza el pedido y muestra el mensaje de error

**Variaciones:**

**Poscondiciones:** El sistema muestra el pedido solicitado

**Extensiones:** Estado pedido

### C. Cliente:



#### ✚ Consulta de catálogo:

<b>Caso de uso:</b> Consulta de Catálogo
<b>Objetivo:</b> El usuario quiere visualizar el catálogo de productos, accediendo mediante las categorías.
<b>Actores:</b> Cliente
<b>Precondiciones:</b> No hay
<b>Casos de uso relacionados:</b>
<b>Pasos:</b> 1. El usuario accede a las distintas categorías de productos 2. Selecciona una categoría y le muestra el detalle de productos de la misma 3. Selecciona un producto de la categoría y le muestra el detalle del producto concreto.
<b>Variaciones:</b>
<b>Poscondiciones:</b> El sistema muestra el detalle del producto buscado
<b>Extensiones:</b>

#### ✚ Login:

<b>Caso de uso:</b> Login
<b>Objetivo:</b> Identifica al usuario mediante username y password
<b>Actores:</b> Cliente
<b>Precondiciones:</b> No hay

<b>Casos de uso relacionados:</b> Consulta Datos
<b>Pasos:</b> 1. El cliente introduce su username y password. 2. El sistema valida los datos y permite el acceso 3. En el caso de que el sistema no valide los datos, no da acceso y permite crear un nuevo usuario..
<b>Poscondiciones:</b> El sistema ha validado el usuario y le ha dado acceso
<b>Variaciones:</b> El sistema no valida al usuario y lo rechaza
<b>Extiende/ Incluye:</b> Consulta Datos

### **Alta Cliente:**

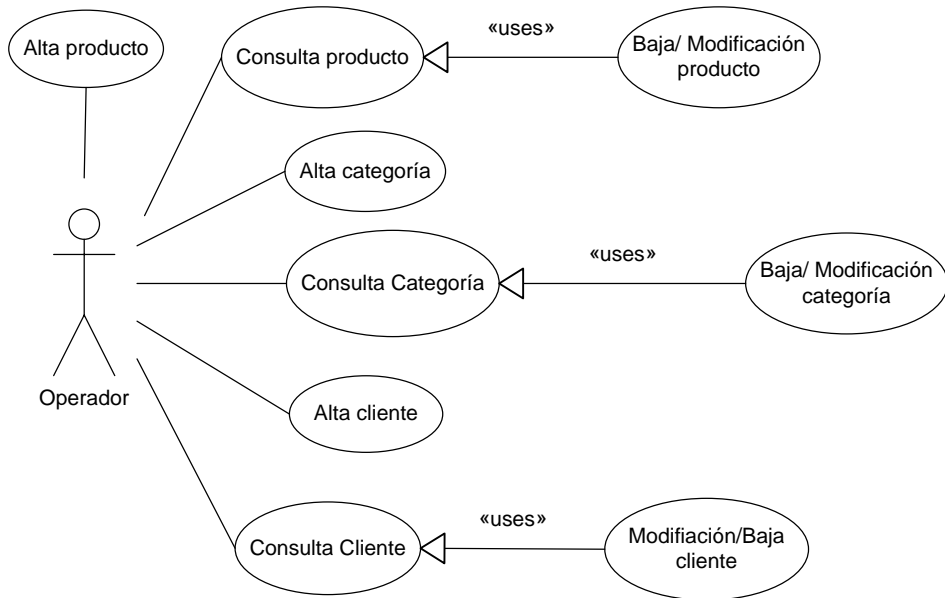
<b>Caso de uso:</b> Alta Cliente
<b>Objetivo:</b> Registrar a un nuevo cliente en el sistema
<b>Actores:</b> Cliente
<b>Precondiciones:</b> No hay
<b>Casos de uso relacionados:</b>
<b>Pasos:</b> 1. El cliente registra sus datos 2. El sistema da al cliente de alta en el sistema
<b>Poscondiciones:</b> Se ha dado de alta un nuevo cliente en el sistema
<b>Variaciones:</b>
<b>Extensiones:</b>

### **Consulta de un pedido:**

<b>Caso de uso:</b> Consulta de Pedido
<b>Objetivo:</b> El usuario quiere visualizar un pedido que previamente ha realizado.
<b>Actores:</b> Cliente
<b>Precondiciones:</b> Tiene que existir un pedido pendiente de entrega realizado por este cliente
<b>Casos de uso relacionados:</b> Nuevo pedido
<b>Pasos:</b> 1. El sistema muestra los pedidos realizados pendientes de entrega de este cliente
<b>Variaciones:</b>
<b>Poscondiciones:</b> El cliente ha visualizado los pedidos previamente realizados y no entregados.
<b>Extensiones:</b>

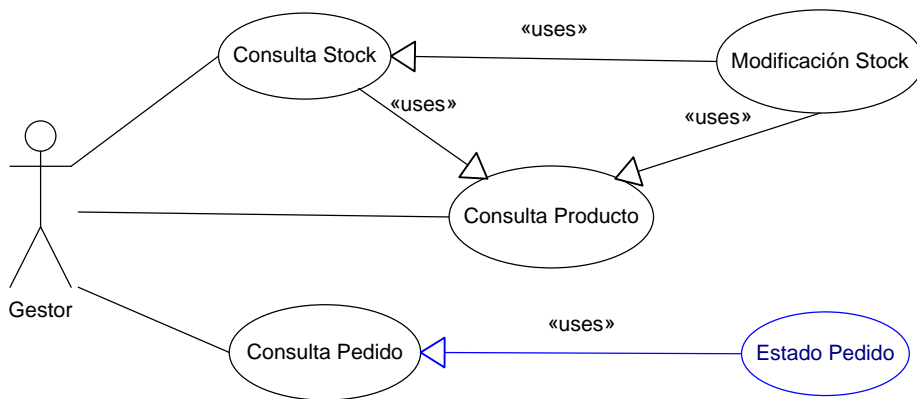
II. FUNCIONALIDADES AMPLIADAS:

A. Operador:



Los casos de uso son los ya contemplados como "funcionalidades básicas"

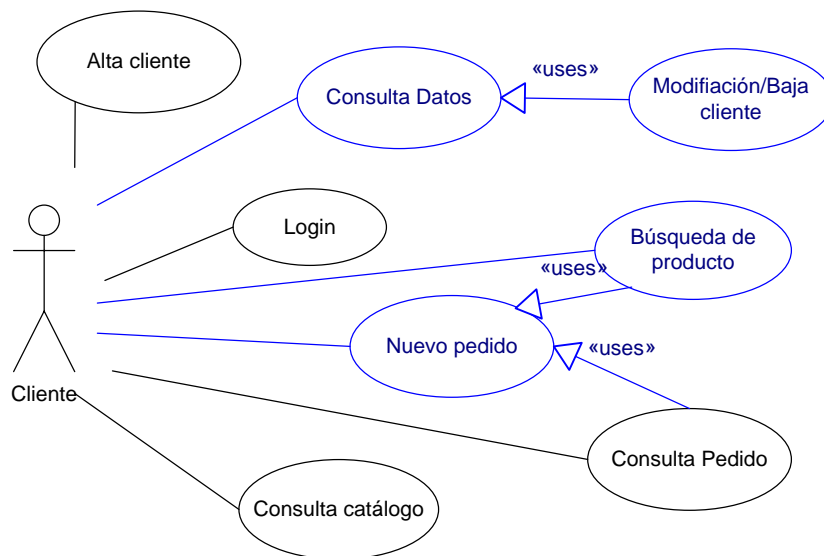
B. Gestor:



**Estado de un pedido:**

<b>Caso de uso:</b> Estado pedido
<b>Objetivo:</b> Se quiere modificar el estado de un pedido concreto, como consecuencia de la entrega o devolución del mismo
<b>Actores:</b> Gestor
<b>Precondiciones:</b> El pedido existe en el sistema
<b>Casos de uso relacionados:</b> consulta pedido
<b>Pasos:</b> 1. Se selecciona el pedido del que se desean modificar su estado. 2. Se modifican el estado del cliente 3. Se confirma la modificación del estado del pedido
<b>Variaciones:</b>
<b>Poscondiciones:</b> Se ha modificado el estado pedido
<b>Extensiones:</b>

**C. Cliente:**



**Búsqueda de productos:**

<b>Caso de uso:</b> Búsqueda de Productos
<b>Objetivo:</b> Se introduce el nombre o referencia del producto para poder ver el detalle
<b>Actores:</b> Cliente
<b>Precondiciones:</b> No hay

<b>Casos de uso relacionados:</b>
<b>Pasos:</b> 1. Indicar el producto que se quiere visualizar. 2. a El sistema localiza el producto y lo muestra. 2. b El sistema no localiza el producto y muestra el mensaje de error
<b>Variaciones:</b>
<b>Poscondiciones:</b> El sistema muestra el producto solicitado
<b>Extensiones:</b>

### **Consulta de datos:**

<b>Caso de uso:</b> Consulta de datos
<b>Objetivo:</b> El usuario accede a los datos del cliente
<b>Actores:</b> Cliente y Operador
<b>Precondiciones:</b> El cliente ha sido validado por el sistema
<b>Casos de uso relacionados:</b> login
<b>Pasos:</b> 1. El usuario realiza la consulta de los datos del cliente. 2. El usuario muestra los datos del cliente 3. El usuario del sistema accede a los datos del cliente
<b>Poscondiciones:</b> El sistema muestra los datos del cliente.
<b>Variaciones:</b>
<b>Extensiones:</b> Modificación /Baja Cliente

### **Modificación/ baja de Clientes:**

<b>Caso de uso:</b> Modificación/ Baja de Clientes
<b>Objetivo:</b> El usuario modifica los datos del cliente o da de baja al cliente
<b>Actores:</b> Cliente y Operador
<b>Precondiciones:</b> El usuario ha realizado la consulta de los datos del usuario y el sistema le ha mostrado los datos.
<b>Casos de uso relacionados:</b> Consulta Cliente
<b>Pasos:</b> 1. El usuario realiza una modificación de los datos del cliente 2. El usuario guarda los datos pulsando a "Aceptar"
<b>Variaciones:</b> 1 b El usuario realiza la baja/eliminación del cliente: 2. b. El usuario elimina el usuario pulsando "Baja Cliente"
<b>Poscondiciones:</b> El sistema modifica o elimina los datos del Cliente
<b>Extensiones:</b>

✚ **Realización de un nuevo pedido:**

<b>Caso de uso:</b> Nuevo pedido
<b>Objetivo:</b> Realizar un nuevo pedido
<b>Actores:</b> Cliente
<b>Precondiciones:</b> El cliente tiene que estar dado de alta en el sistema
<b>Casos de uso relacionados:</b> búsqueda de producto, consulta pedido
<b>Pasos:</b> 1. Seleccionar el/los productos de los que se desea realizar el pedido 2. a El sistema da la conformidad del pedido realizado b El sistema indica que no es posible realizar dicho pedido 3. Se confirma la realización del pedido
<b>Variaciones:</b>
<b>Poscondiciones:</b> Se ha realizado un nuevo pedido
<b>Extensiones:</b> búsqueda de producto, consulta pedido

#### 4. - MODELO CONCEPTUAL DE CLASES

A partir de la lista de informaciones que hemos obtenido del diagrama de usos empezamos a construir el modelo conceptual de clases.

Para lo cual a partir de los objetivos del proyecto y de la descripción de los casos de uso, utilizamos la técnica de identificación de nombres y encontramos los siguientes conceptos:

Cliente, Operador, Gestor, Pedido, Producto, Categoría y Stock.

Analizamos a continuación las asociaciones fundamentales:

- Operador - Realiza – Alta/ Baja/ Modificación del Producto
- Operador - Realiza – Alta/ Baja/ Modificación de Categoría
- Operador - Realiza – Alta/ Baja/ Modificación de Cliente
- Cliente - Realiza su - Alta/ Baja / Modificación de datos
- Cliente - Realiza – Pedido
- Cliente – Consulta – Pedido
- Gestor - Consulta/Modifica - Pedido
- Gestor - Consulta/Modifica - Stock
- Pedido - Contiene - Productos
- Categoría - Contiene - Producto
- Producto – Tiene - Stock

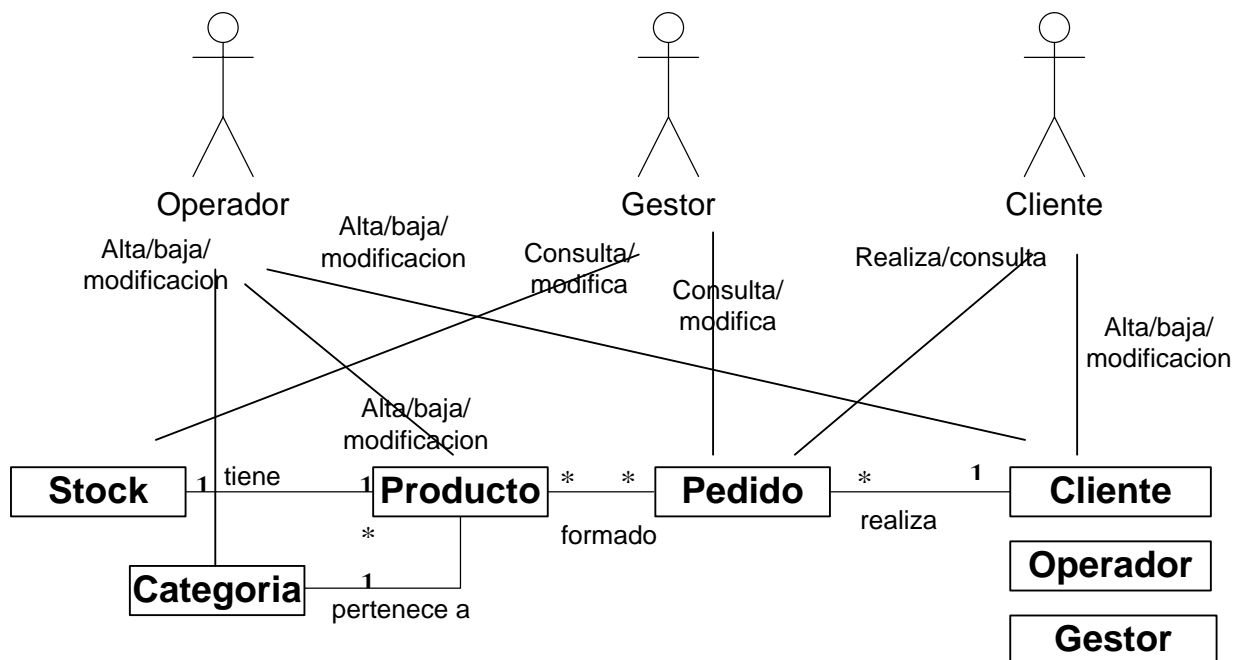
A continuación identificamos los atributos de estos conceptos. En la especificación sólo se hace referencia a los siguientes atributos:

- **Cliente:** password, nombre, dirección, cpostal, población, provincia, fechaAlta, fechaBaja
- **Pedido:** cliente, producto, fecha, estado
- **Producto:** descripción, características, precio, foto,
- **Categoría:** descripción



- **Stock:** unidades, entrada-salida, fecha
- **Operador:**
- **Gestor:**

El resto de atributos irán apareciendo durante las siguientes fases del desarrollo.



## ***B. DISEÑO***

---

### ***1. ARQUITECTURA DEL SISTEMA EN N-CAPAS***

Por una parte, es relativamente sencillo reconocer por qué nos podría interesar una disposición en dos capas, como mínimo: no queremos colocar equipos con las características de un servidor ante los usuarios, ya que iría en detrimento de la centralización de los datos y la eficacia desde un punto de vista económico. Sin embargo, en la mayoría de casos se introduce una tercera capa, en la que se alojan los componentes empresariales o la lógica; en el diagrama canónico de las aplicaciones web aparece en ocasiones una cuarta capa, con lo que tenemos la de cliente, la de servidor web, la de lógica empresarial y la de base de datos.

Se combinan dos fuerzas para crear el método de n niveles. El primero es la necesidad de escalabilidad: Con el crecimiento de Internet y una mayor accesibilidad para los usuarios, anteriormente era inviable pensar en que un cliente pudiera realizar los pedidos accediendo a nuestro sistema, pero actualmente podemos permitir a los clientes acceder a nuestro sistema para realizar pedidos a través de su navegador, con total seguridad.

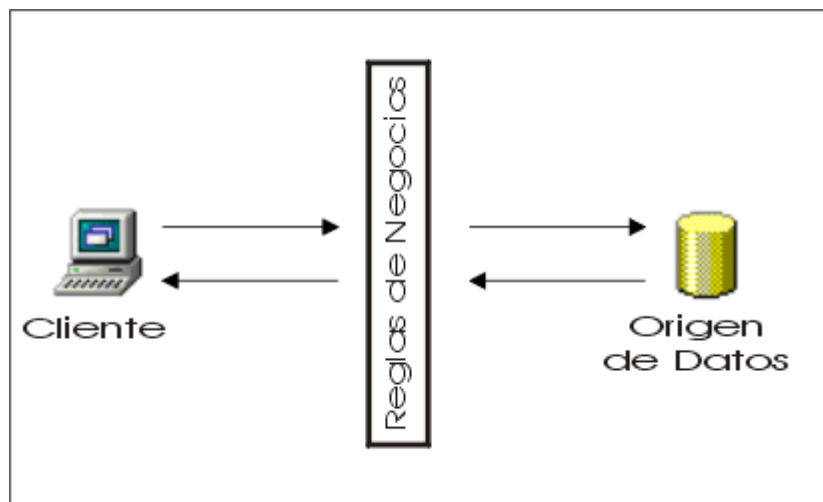
Un motivo para considerar la arquitectura de n módulos es también la idea de que el nivel intermedio sea un punto de reunión de lógica independiente de la presentación a la que se puede tener acceso desde distintos niveles de presentación. El ejemplo canónico que lo ilustra es la aplicación de combinación intranet/extranet, en que los empleados internos usan una aplicación de formularios WinForms (o, en nuestro caso Windows Presentation Foundation) para obtener acceso a un sistema de nivel intermedio que, a su vez, tiene acceso a la base de datos; mientras tanto, los clientes externos usan ASP.NET para hacer lo mismo: obtener acceso al nivel intermedio que, a su vez, tiene acceso a la base de datos.

El .NET Framework está repleto de nuevas tecnologías y nuevas versiones de las tecnologías existentes, incluyendo una serie de cambios en la óptima arquitectura de

una aplicación en 'n' niveles y proporcionando ensamblados y procedimientos almacenados. Para aplicaciones basadas en navegador, los ensamblados pueden crearse utilizando páginas .aspx. A diferencia de las aplicaciones tradicionales, se puede escribir toda la capa de negocio utilizando ASP.NET, una razón para ello es la opción de código 'por detrás' (code-behind).

Resumen de arquitectura en 'n' capas escogida:

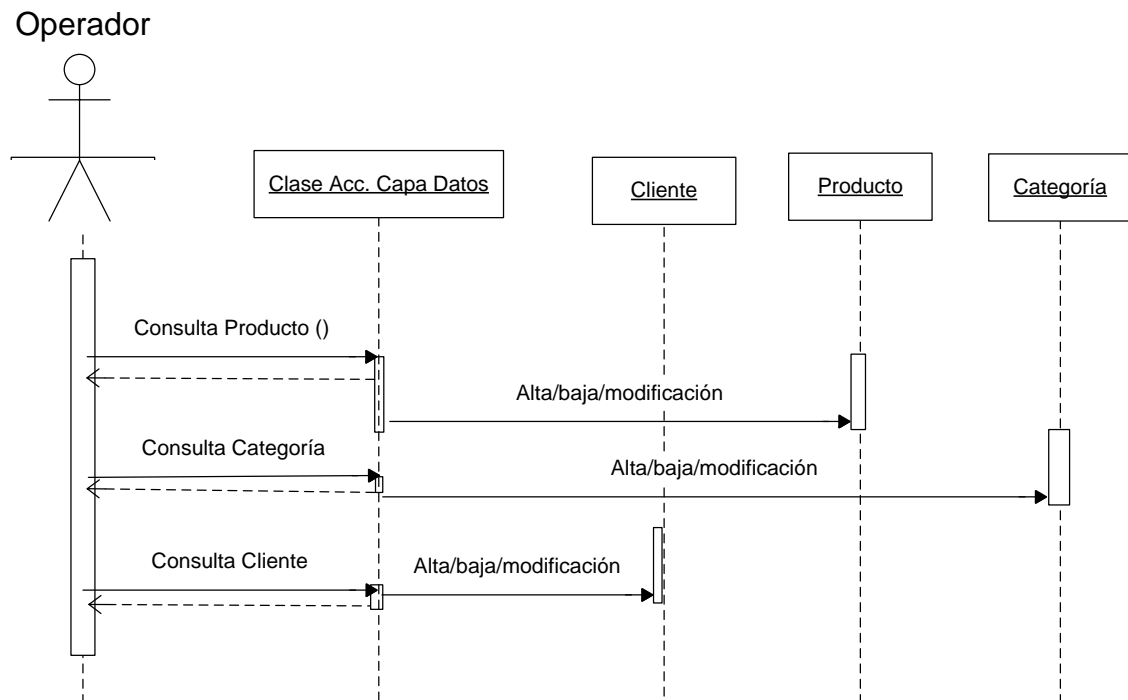
- Desarrollo de clientes (navegador, aplicaciones windows o wpf y AJAX)
- Lógica de negocio (code-Behind)
- Acceso a datos almacenados (ADO.NET)



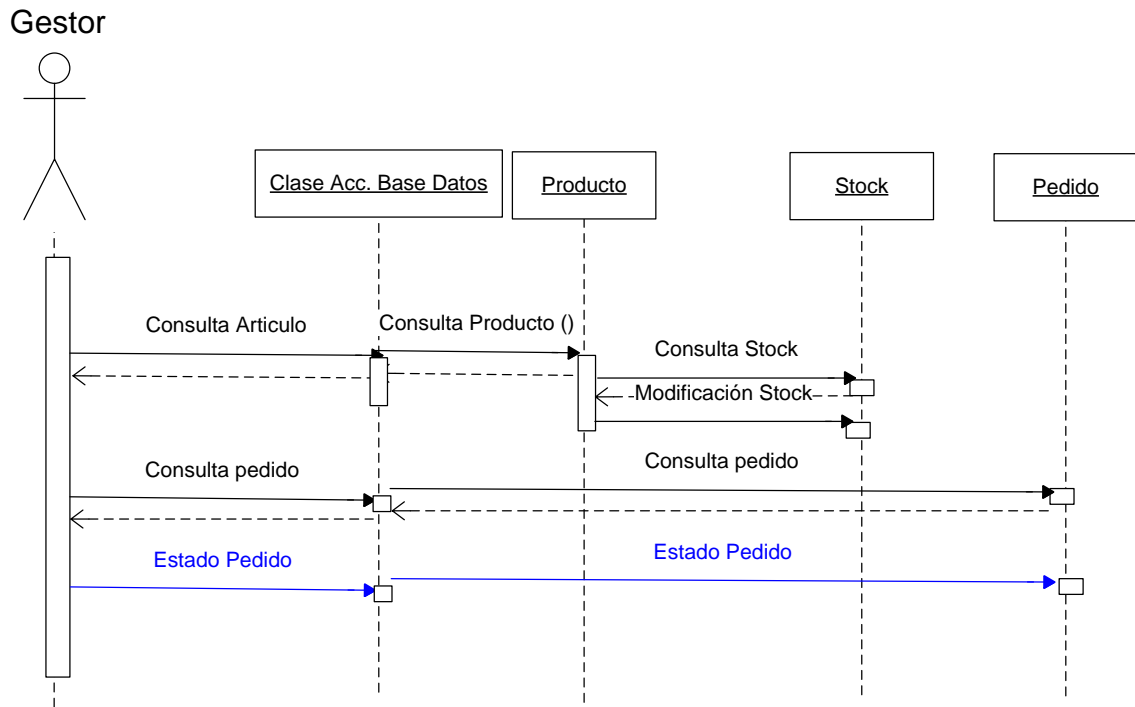
## 2. DIAGRAMAS DE SECUENCIA

A continuación utilizamos el diagrama de secuencia para determinar las operaciones que demandan los actores, y reflejaremos su interacción con los objetos del sistema: cliente, producto, categoría, etc. Se han agrupado los diagramas de secuencia por cada actor ([en azul se muestran las funciones ampliadas](#)):

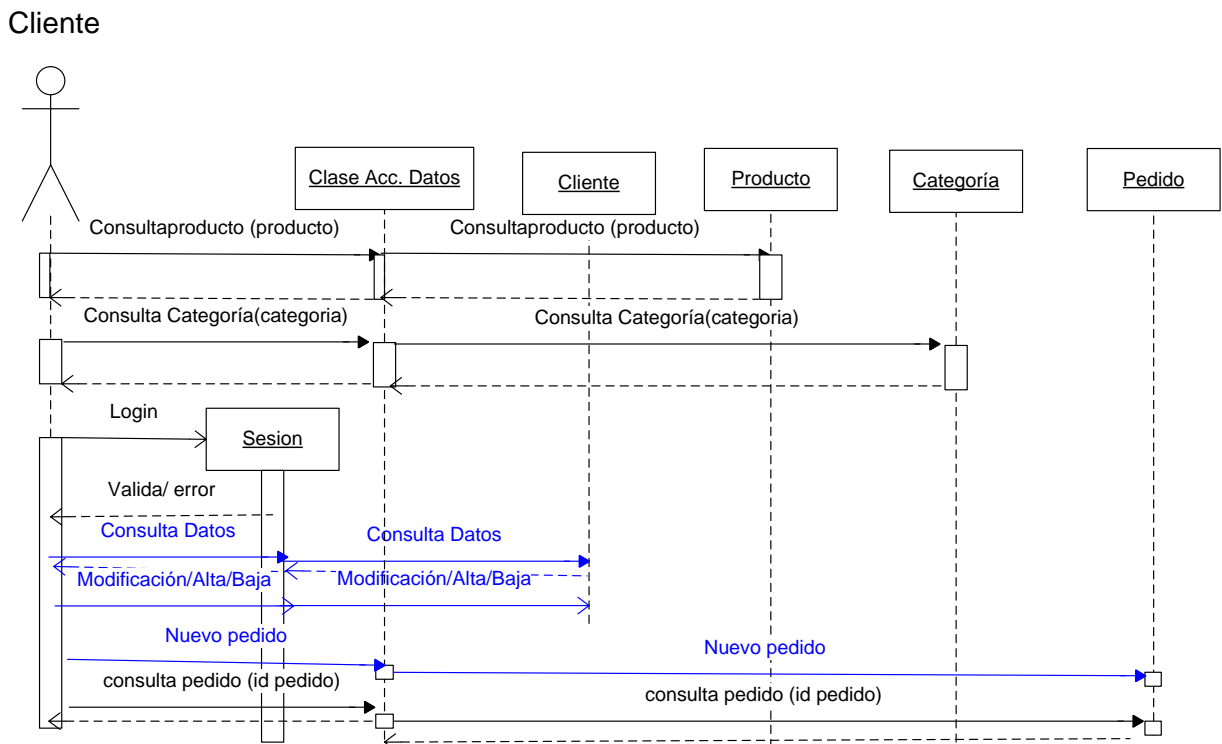
### A. Operador:



**B. Gestor:**

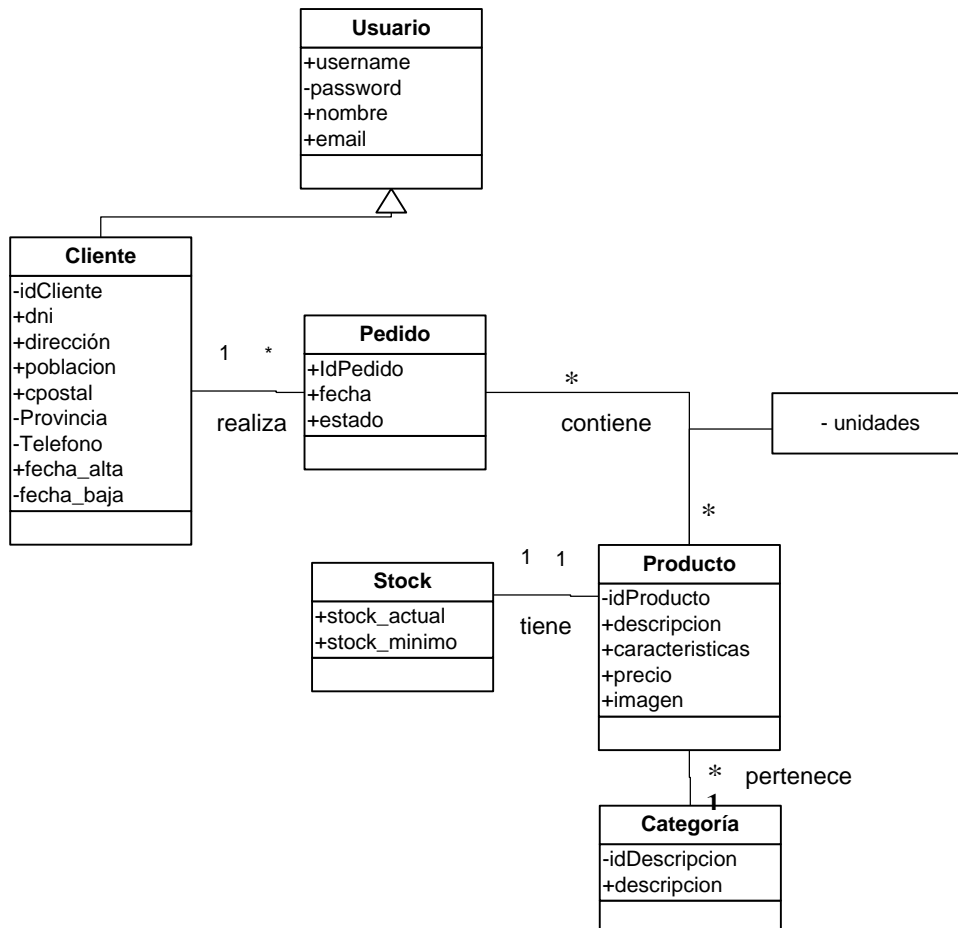


**C. Cliente:**



### 3. DIAGRAMA DE CLASES DEL DISEÑO

A partir del modelo conceptual desarrollado, elaboramos el diagrama de clases. En este diagrama se han eliminado los conceptos de Operador y Gestor, porque no se prevé que tengan que estar identificados ni registrados en el sistema, no obstante se contempla la clase "Usuario", de la que heredarían en el caso de querer incluir estas nuevas clases en un futuro. Mantendremos las asociaciones y atributos que existen en el modelo conceptual.



## 4. DISEÑO DE LA INTERFAZ GRAFICA

### I. FUNCIONALIDADES BÁSICAS:

#### ✚ APLICACIÓN GESTOR Y OPERADOR:

**Consulta de categorías:** mediante esta opción se visualizan las categorías definidas.



**Nueva Categoría:**



**Modificación de una categoría:**



**Alta y editar el producto:** opción que permite dar de baja o modificar un producto en el sistema.



**surf eCatalogo**

**Nuevo Producto**

Categoría:

Descripción:

Características:

imagen:

Precio:

Stock Actual:

Stock Mínimo:

Aceptar Cancelar

**surf eCatalogo**

**Editar Producto**

Categoría: Aletas

Descripción: Aleta piscina

Características: Aleta de 30 cm y plástico duro. Especial para piscina. En naranja



Precio (€): 30

Stock Actual: 24

Stock Mínimo: 10

Aceptar Cancelar

**Consulta cliente:**

**surf eCatalogo**

Cliente Pedidos Producto Stock Categoría

**Cientes**

Díaz Ramírez, Pedro  
 Fernández Aranda, Beatriz  
 García Álvarez, Arantza  
 González Jiménez, Miguel  
 Sánchez Marcos, Andrés

Ver Pedidos Nuevo Editar Eliminar

**Nuevo cliente:**



**surf eCatalogo**

**Nuevo Cliente**

Nombre

DNI

Población  C.P.

Provincia  Teléfono

email

FechaAlta  FechaBaja

username  password

Aceptar Cancelar

**Modificar los datos del cliente:**



**surf eCatalogo**

**Editar Cliente**

Nombre

DNI

Población  C.P.

Provincia  Teléfono

email

FechaAlta  FechaBaja

username  password

Aceptar Cancelar

**Listado de nivel de stock de productos:** muestra el nivel de stock del listado de productos, en la misma se pueden modificar los datos de entrada de nuevas unidades de stock.



**Consulta/listado pedidos pendientes de entregar:** opción que muestra los datos más relevantes de los pedidos pendientes de entrega.



**APLICACIÓN CLIENTE:**

**Consulta de productos:** permite al cliente localizar los productos por categorías.



**Detalle de productos:** muestra el detalle del producto seleccionado.



**Identificación del cliente:** esta opción permite al cliente identificarse para tener acceso a la parte de "Acceso clientes" nuestro sistema.



**Menú del área "acceso a clientes":** esta opción permite al cliente identificarse para tener acceso a la parte de "Acceso clientes" nuestro sistema.



## II. FUNCIONALIDADES AMPLIADAS:

### + APLICACIÓN OPERADOR Y GESTOR:

**Detalle de pedido:** opción que permite ver el detalle del pedido, y modifica el estado del pedido del cliente como entregado.



### + APLICACIÓN CLIENTE:

**Alta nuevo usuario:** opción que permite darse de alta un nuevo usuario.

La misma pantalla permite editar los datos del mismo el caso de querer modificar sus datos.



The screenshot shows a web browser window with the title 'surf eCatalogo' and the URL 'www.eCatalogosurf.com'. The page has a blue header with a search icon and the text 'surf eCatalogo'. Below the header, there are links for 'Acceso clientes' and 'Contacto'. The main content area is divided into two columns. The left column is titled 'Categorías' and lists 'Tablas de surf', 'Tabla de snowboard', 'Table skate', 'Trajes neopreno', and 'Aletas'. Below this list is a search bar with the text 'Buscar:' and a 'Registro' button. The right column is titled 'Nuevo usuario' and contains a registration form with the following fields: 'Nombre', 'DNI', 'Teléfono', 'Dirección', 'C.P.', 'Población', 'Provincia', 'email', 'username', and 'password'. At the bottom of the form are 'Aceptar' and 'Cancelar' buttons.

**Realización de nuevo pedido:** al pinchar sobre el "carrito de compra" se añade el producto al pedido realizado.



The screenshot shows a web browser window with the title 'surf eCatalogo' and the URL 'www.eCatalogosurf.com'. The page has a blue header with a search icon and the text 'surf eCatalogo'. Below the header, there are links for 'Acceso clientes' and 'Contacto'. The main content area is divided into two columns. The left column is titled 'Categorías' and lists 'Tablas de surf', 'Tabla de snowboard', 'Table skate', 'Trajes neopreno', and 'Aletas'. Below this list is a search bar with the text 'Buscar:' and a 'Registro' button. The right column is titled 'Aleta Piscina' and contains a product page for 'Aleta piscina'. The product is shown with an image of a pool fin and the following details: 'Características: Aleta de 30 cm y plástico duro. Especial para piscina. En naranja' and 'Precio (€) 30'. At the bottom right of the page is a shopping cart icon and the text 'Realizar Pedido'.

**Pantalla que muestra el detalle del pedido, y confirmar el pedido.**

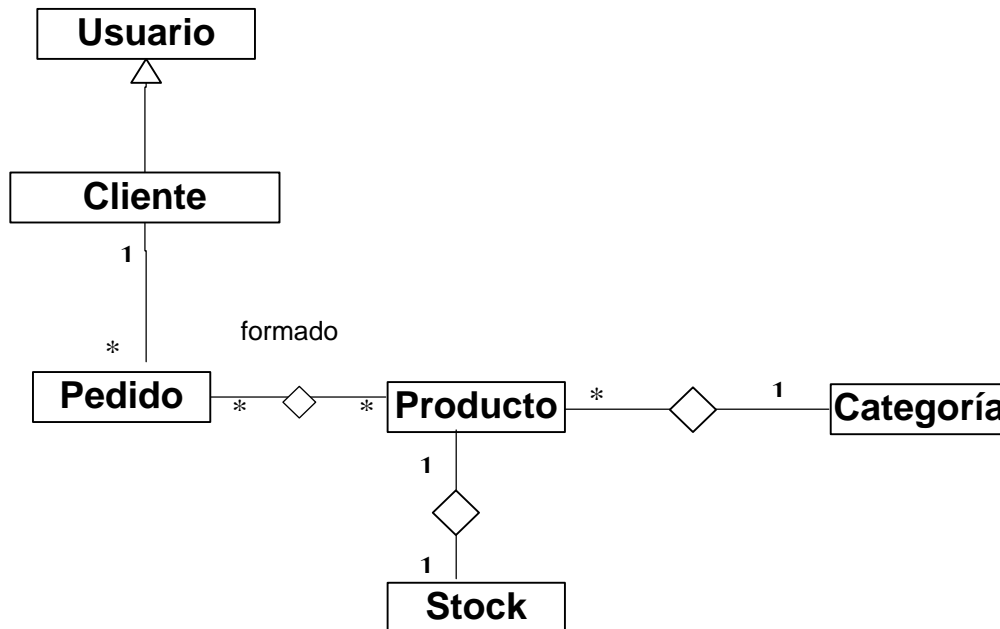


**Consulta de pedidos realizados:** el usuario puede consultar los pedidos realizados.





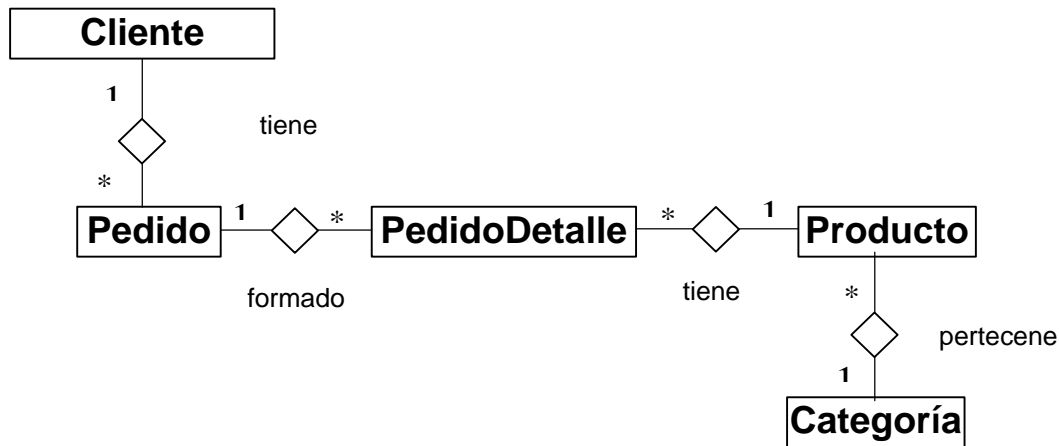
## 5. DISEÑO DE LA BASE DE DATOS



Para el diseño de la base de datos debemos definir previamente la estructura de los datos que debía tener la base de datos de nuestro sistema. De este modo, a través de nuestro diagrama de clases (modelo estático) hemos realizado el modelo de entidad-relación (ER) como podemos ver justamente en la figura superior. Una vez definido este modelo hemos suprimido la herencia, transformado las interrelaciones, entidades débiles, etc. para poder llegar al modelo relacional que esta a continuación y de esta forma tener el diseño definitivo del diagrama de persistencia.

En concreto:

- ✚ Se ha eliminado la clase usuario, integrándose todos sus atributos a la clase cliente.
- ✚ Se ha eliminado la clase stock, dada su relación unitaria con la clase producto, se han incorporado sus atributos a la clase producto.
- ✚ Se ha creado una clase DetallePedido dada la relación múltiple de pedido y producto.



### **CLIENTE**

IdCliente, username, password, dni, nombre, direccion, cpostal, poblacion, provincia, telefono, email, fechaAlta, fechaBaja.

Donde:

IdCliente es clave única de identificación de cliente (PrimaryKey)  
dni es clave alternativa

### **PEDIDO**

idPedido, cliente, fecha, estado

Donde:

idPedido es la clave única de identificación de pedido (PrimaryKey)  
Cliente es clave foranea a cliente (idCliente)

### **PEDIDODETALLE**

idPedido, idLinea, idProducto, uds, precioUd

Donde:

idPedido, idLinea es clave única de identificación de pedidodetalle (PrimaryKey)  
idProducto es clave foranea de producto (idProducto)

**PRODUCTO**

idProducto, descripción, características, predioUd, stockActual, stockMinimo, imagen, idCategoría

Donde:

idProducto es clave única de identificación de producto (PrimaryKey)

idCategoría es clave foránea de categoría (idCategoría)

**CATEGORÍA**

idCategoría, descripción

Donde:

IdCategoría es clave única de identificación de categoría (PrimaryKey)

## ***C. IMPLEMENTACIÓN***

---

Se ha obtenido el código de la aplicación a partir de la documentación elaborada en las fases previas del proyecto, de análisis y diseño.

La solución esta compuesta de dos aplicaciones distintas, dado que los escenarios en los que se ejecutan son diferentes. Cada una de ellas puede ser utilizada por múltiples usuarios. A continuación se muestran los aspectos más relevantes de la implementación de cada una de las diferentes aplicaciones.

### **+ Aplicación de escritorio de gestión y mantenimiento:**

La base de datos se ha creado mediante SQL Server y se mantiene mediante ADO.NET.

Se han utilizado dataset tipados en un parte de la aplicación por su simplicidad a la hora de establecer un acceso a datos rápido y eficaz.

En la interface de presentación de los datos a los usuarios se han distinguido dos tipos de programas en la aplicación, los programas de mantenimiento y los programas de consulta. Los programas de consulta son llamados desde los botones situados en la pantalla principal de la aplicación, éstos están diseñados bajo el mismo criterio que la principal, con Windows Presentation Foundation (WPF) para dar una buena apariencia a esta parte de la aplicación. Y seleccionando un elemento de la consulta se accede a los programas de mantenimiento diseñados mediante WindowForms.

Distinguimos en esta aplicación distintos tipos de clases:

- ❑ xaml: formularios windows presentation foundation
- ❑ xsc y xss: dataSet tipado para acceso a datos
- ❑ vb: windows forms

Tras valorar distintas herramientas que ofrecen terceras empresas (xceed, infragistics), se ha seleccionado la herramienta de Infragistics para los grids de consulta de la aplicación debido a que Microsoft aún no dispone de dicha herramienta

para la tecnología de Windows Presentation Foundation. El enlace de dichos grids con los datos se ha realizado mediante ADO.NET, esta vez sin usar dataset tipados.

A continuación se muestra un fragmento de código que muestra como se ha llevado a cabo dicho proceso, pudiéndose comprobar la sencillez del mismo:

```

Partial Public Class ListaStock
    Inherits Grid
    Private daProd As SqlDataAdapter

    Public Sub New()
        InitializeComponent()
        Dim ds As DataSet = CrearDataSet()
        Me.XamDataGrid1.DataSource = ds.Tables(0).DefaultView
    End Sub
    'Creamos una función para retornar el DataSet

    Public Function CrearDataSet() As DataSet

        Dim dsDatos As New DataSet
        Dim cn As New
        SqlConnection(Global.eCatalogo.Settings.Default.eCatalogoConnectionString)
        Dim sqlStock As SqlCommand = New SqlCommand("select
        idproducto,descripcion,stockminimo,stockactual from Producto where
        stockactual<=stockminimo", cn)

        Dim da As SqlDataAdapter = New SqlDataAdapter()
        da.SelectCommand = sqlStock
        da.Fill(dsDatos, "Producto")
        Return dsDatos
    End Function
End Class

```

### **Aplicación Cliente Web:**

Esta aplicación es accesible a través de Internet, se ha utilizado ASP.NET, AJAX y ADO.NET. Igualmente se accede a la misma base datos que la aplicación de escritorio, mediante ADO.NET.

Además de la base de datos eCatalogo para la gestión de nuestro catalogo, se ha creado automáticamente, la base de datos aspnetdb al utilizar el proveedor de memberShip.

Distinguimos en esta aplicación distintos tipos de clases:

- .master:** plantilla maestra para mantener un diseño común en las distintas páginas de la web
- .aspx:** páginas web de la aplicación
- .vb:** desarrollo de la programación adicional necesario para el buen funcionamiento de la aplicación web.

Además de estas tenemos el fichero web.config, donde se implementan los detalles de configuración generales de la aplicación web.

Por ejemplo tenemos el siguiente código perteneciente a la sección <system.web> que especifica el tipo de autenticación que vamos a usar:

```
<authentication mode="Forms" />
```

Y el fichero stylesheet.css que es la hoja de estilo general donde están definidos los distintos estilos que se utilizan en la aplicación web.

A continuación se muestra, a modo de ejemplo, parte del código usado para conectarse a las dos bases de datos, mediante dos cadenas de conexión:

```
'base datos gestion
<add name="LocalSqlServer" connectionString="Data
Source=.\SQLEXPRESS;Integrated Security=True;User
Instance=True;AttachDBFilename=|DataDirectory|aspnetdb.mdf"/>
'base datos membership provider
<add name="eCatalogoConnectionString" connectionString="Data
Source=.\SQLEXPRESS;Initial Catalog=eCatalogo;Integrated
Security=True;Pooling=False"
providerName="System.Data.SqlClient" />
```

### III. CONCLUSIÓN

---

#### **1. CONCLUSIÓN Y OBJETIVOS ALCANZADOS**

El resultado final del trabajo ha sido realmente muy enriquecedor ya que ha permitido tener una primera aproximación a la tecnología .NET de Microsoft, adquiriendo suficientes conocimientos como para desarrollar una aplicación Web de complejidad media. El proceso ha requerido mucho esfuerzo por la falta de información disponible al tratarse de tecnologías tan novedosas (especialmente WPF). No obstante, una vez familiarizada con estas tecnologías las nuevas posibilidades que ofrecen, han justificado el esfuerzo realizado. También ha servido como repaso y puesta en práctica de numerosos conceptos alcanzados en diversas asignaturas a lo largo de la carrera, especialmente de las asignaturas de programación, bases de datos y ingeniería del software.

Se puede decir que se han alcanzado todos los objetivos propuestos inicialmente, tanto por el desarrollo de la aplicación como por los conocimientos adquiridos. Por otro lado, la propia redacción de la memoria es un ejercicio recomendable para alcanzar una base sólida en redacción de documentación técnica.

#### **2. AMPLIACIONES Y MEJORAS DEL PROYECTO**

Se ha conseguido una aplicación que implementa todas las funcionalidades que se marcaron como objetivo durante la planificación del trabajo, utilizando para ello las tecnologías previstas. Sin embargo, existe toda una serie de mejoras o añadidos que se podrían realizar para mejorar el producto final.

En la parte de **la aplicación escritorio**, aplicación utilizada por los gestores y administradores del sistema:

- ✚ Los identificadores únicos actualmente se introducen de forma manual, sería recomendable se generaran de forma automática.
- ✚ Se debería ampliar las posibilidades en el mantenimiento y consulta de pedidos, actualmente se consulta y el grid permite su modificación, pero no se ha desarrollado el procedimiento que actualice el stock al lanzar el pedido.
- ✚ Con objeto de mejorar la utilización se propone vincular desde el acceso de clientes los pedidos. Es decir poder filtrar desde el acceso a clientes sus pedidos correspondientes, y poder actualizar su situación.

En la parte de la **aplicación web**, aplicación utilizado por los clientes:

- ✚ En lo relativo a la gestión de altas en clientes, actualmente las bases de datos de clientes de la aplicación escritorio y la aplicación web son independientes, esto supone que el cliente que se registra vía web mediante la herramienta membership que te proporciona asp.net no se da de alta automáticamente en la tabla de clientes. El sistema debería completarse para que las altas realizadas mediante la web se registraran en la tabla clientes, de la misma forma los campos de datos personales actualmente no se solicitan en la web, tan solo los relativos al usuario y al password.
- ✚ Falta desarrollar la parte de realizar un nuevo pedido, el sistema permite visualizar los pedidos ya creados, pero no lanzar uno nuevo.
- ✚ El buscador figura en el diseño, pero no esta operativo.



## IV. GLOSARIO

---

**ADO.NET** es la solución propuesta por Microsoft para permitir el acceso a bases de datos desde su plataforma .NET.

**AJAX:** Asynchronous JavaScript And XML (JavaScript y XML asíncronos, donde XML es un acrónimo de eXtensible Markup Language), es una técnica de desarrollo web para crear aplicaciones interactivas.

**ASP.NET** es el componente de Microsoft.Net totalmente orientado al desarrollo de aplicaciones Web.

**Microsoft.NET** es la arquitectura o conjunto de nuevas tecnologías lanzadas por Microsoft con el objetivo de mejorar tanto su sistema operativo como su modelo de componentes para obtener una plataforma con la que sea sencillo el desarrollo de software en forma de servicios web.

**Visual Basic .NET** es el lenguaje de programación creado por Microsoft utilizado en la aplicación.

**Web Forms** Es el marco de trabajo de las páginas ASP, que permite crear páginas Web programables como parte de una aplicación Web global.

**Windows Presentation Foundation:** la tecnología eficaz para la creación de interfaces que ofrecen beneficios apreciables a los usuarios y las organizaciones que se sirven de ellas. WPF ofrece una plataforma unificada para interfaces de usuario modernas, convierte a los diseñadores en participantes activos a la hora de crear estas interfaces y proporciona un modelo de programación común para aplicaciones

## V. BIBLIOGRAFIA

---

Enlaces consultados:

Principalmente he usado la ayuda de Visual Studio, con enlace a msdn :  
<http://www.microsoft.com/es/es/default.aspx>

Y otros enlaces con la finalidad de aprender a utilizar las nuevas tecnologías:

<http://www.codeplex.com/>  
<http://www.programar.net/>  
<http://www.asp.net/> , <http://ajax.asp.net/>  
<http://www.lawebdelprogramador.com/>

Información sobre el SQL Server 2005 Express:  
<http://msdn.microsoft.com/sql/express/default.aspx>

QuickStart Tutorials: <http://samples.gotdotnet.com/quickstart/>  
Descarga herramienta de W.P.F.: <http://www.infragistics.com/>