

# Trabajo Fin de Carrera en Tecnologías .Net

**José María Artal Bañón**

ITIG

**David Gañán Jiménez**

11 de Junio de 2007

## **Agradecimientos**

---

A Concha por todo su apoyo durante estos años; sin ella volver a estudiar hubiera sido bastante más difícil.

A mis padres, hermanos y sobrinos por enseñarme el verdadero significado una familia unida.

A David Iglesias por su apoyo profesional y críticas constructivas aportando siempre un punto de vista interesante.

## Resumen

---

El objetivo de este Trabajo Fin de Carrera ha sido profundizar en las Tecnologías .Net desarrollando una solución software relativamente sencilla: la gestión de un Catálogo Online de Productos (en este caso en concreto los productos son miniaturas a escala para pintar).

El desarrollo de una solución software sencilla, permite orientar este aprendizaje a un punto de vista completamente práctico; se han utilizado diversas tecnologías como WebServices, WPF, Patrones de Diseño, etc. Además el lenguaje de programación es C# aprovechando este proyecto para aprenderlo. Se han aplicado los conocimientos adquiridos en diversas (Ingeniería del Software, Técnicas de Desarrollo de Software, Bases de Datos, Programación Orientada a Objetos, etc.) asignaturas de ITIG y se han aplicado de una manera global y conjunta a lo largo del proyecto.

Esta memoria describe las etapas para crear la aplicación: captura de requisitos, análisis, diseño e implementación; también se incluye un pequeño resumen de las tecnologías utilizadas. Por último se exponen unas conclusiones sobre el resultado obtenido y el conocimiento adquirido.

## Índice de Capítulos

<i>Agradecimientos</i>	<i>I</i>
<i>Resumen</i>	<i>II</i>
<b>1. Introducción</b>	<b>1</b>
1.1. Justificación del Proyecto	1
1.2. Objetivos	1
1.3. Enfoque y Método seguido	2
1.4. Planificación del proyecto	4
1.5. Productos obtenidos	6
1.6. Sigüientes capítulos	6
<b>2. Especificación del Proyecto</b>	<b>7</b>
2.1. Definición de los subsistemas	7
2.2. Requisitos funcionales	7
2.3. Composición del software	8
2.4. Usuarios de la aplicación	8
2.5. Tecnologías Utilizadas	10
2.5.1. ADO.Net	10
2.5.2. WebServices	10
2.5.3. Windows Presentation Foundation	11
2.5.4. Patrones de Diseño MVC	12
<b>3. Análisis de Requisitos</b>	<b>14</b>
3.1. Diagrama Estático de Clases	14
3.2. Casos de Uso del Subsistema de Consulta	14
3.3. Casos de Uso del Subsistema de Mantenimiento	16
<b>4. Diseño</b>	<b>23</b>
4.1. Arquitectura de la aplicación	23
4.2. Diagramas de Clases del Diseño y Diagramas de secuencia	24
4.3. Diseño de la persistencia	28
4.4. Interface de Usuario	29
<b>5. Implementación</b>	<b>40</b>
5.1. WebServices	40
5.2. Windows Presentation Foundation	40
5.3. Patrones de Diseño MVC – MonoRail	41
5.4. ADO.net	41

<b>6. Conclusiones</b>	<b>42</b>
<b>7. Glosario y Bibliografía</b>	<b>43</b>

## Índice de Ilustraciones

ILUSTRACIÓN 1 ENTORNOS Y CLIENTES .....	2
ILUSTRACIÓN 2 DIAGRAMA DE CLASES ESTÁTICO .....	14
ILUSTRACIÓN 3 CASO DE USO CONSULTA DE OFERTAS .....	14
ILUSTRACIÓN 4 CASO DE USO SUBSISTEMA DE MANTENIMIENTO.....	16
ILUSTRACIÓN 5 ARQUITECTURA DE LA APLICACIÓN .....	23
ILUSTRACIÓN 6 DIAGRAMA DE CLASES SUBSISTEMA DE CONSULTAS.....	25
ILUSTRACIÓN 7 DIAGRAMA DE COLABORACIÓN CONSULTA DE OFERTAS .....	25
ILUSTRACIÓN 8 DIAGRAMA DE CLASES SUBSISTEMA DE MATENIMIENTO .....	27
ILUSTRACIÓN 9 DIAGRAM DE LA BASE DE DATOS .....	28
ILUSTRACIÓN 10 PANTALLA DE BIENVENIDA.....	29
ILUSTRACIÓN 11 PANTALLA DE BÚSQUEDA DE FIGURAS .....	30
ILUSTRACIÓN 12 PANTALLA DE BÚSQUEDA DE OFERTAS.....	30
ILUSTRACIÓN 13 PANTALLA DE LISTADO DE FIGURAS .....	31
ILUSTRACIÓN 14 PANTALLA DE LISTADO DE OFERTAS .....	31
ILUSTRACIÓN 15 PANTALLA DE OFERTA NUEVA .....	32
ILUSTRACIÓN 16 PANTALLA DE BORRAR OFERTA.....	33
ILUSTRACIÓN 17 PANTALLA DE MODIFICAR OFERTA.....	33
ILUSTRACIÓN 18 PANTALLA DE EMPRESA NUEVA .....	34
ILUSTRACIÓN 19 PANTALLA DE BORRAR EMPRESA .....	35
ILUSTRACIÓN 20 PANTALLA DE MOFICIAR EMPRESA.....	35
ILUSTRACIÓN 21 PANTALLA DE FIGURA NUEVA .....	36
ILUSTRACIÓN 22 PANTALLA DE BORRAR FIGURA .....	37
ILUSTRACIÓN 23 PANTALLA DE MODIFICAR FIGURA .....	38

## 1. Introducción

---

Este capítulo ofrece una perspectiva general del proyecto recogiendo puntos importantes como la justificación del proyecto y sus objetivos. También incluye la planificación del proyecto y una descripción de los productos obtenidos.

### 1.1. Justificación del Proyecto

La principal razón para la elección de este proyecto ha sido la oportunidad de aprender y profundizar en un grupo de tecnologías de vanguardia en el mercado como es .Net. Este aprendizaje ha sido eminentemente práctico: se ha desarrollado una solución de gestión sencilla (un catálogo online) utilizando múltiples tecnologías: ADO.net, Patrones de Diseño, WebServices y Windows Presentation Foundation (WPF).

En la Gestión del Catálogo de Figuras, el principal objetivo es la utilización de nuevas tecnologías .Net en una solución real; la implementación de funcionalidades ha pasado a un segundo plano: si hubiera sido un problema real, se hubiera simplificado la solución orientándola sobretodo al desarrollo de funcionalidades. Con este enfoque las fases de análisis y diseño también se han simplificado, orientándonos sobre todo a la implementación de las tecnologías escogidas (con una importante labor de investigación y estudio).

### 1.2. Objetivos

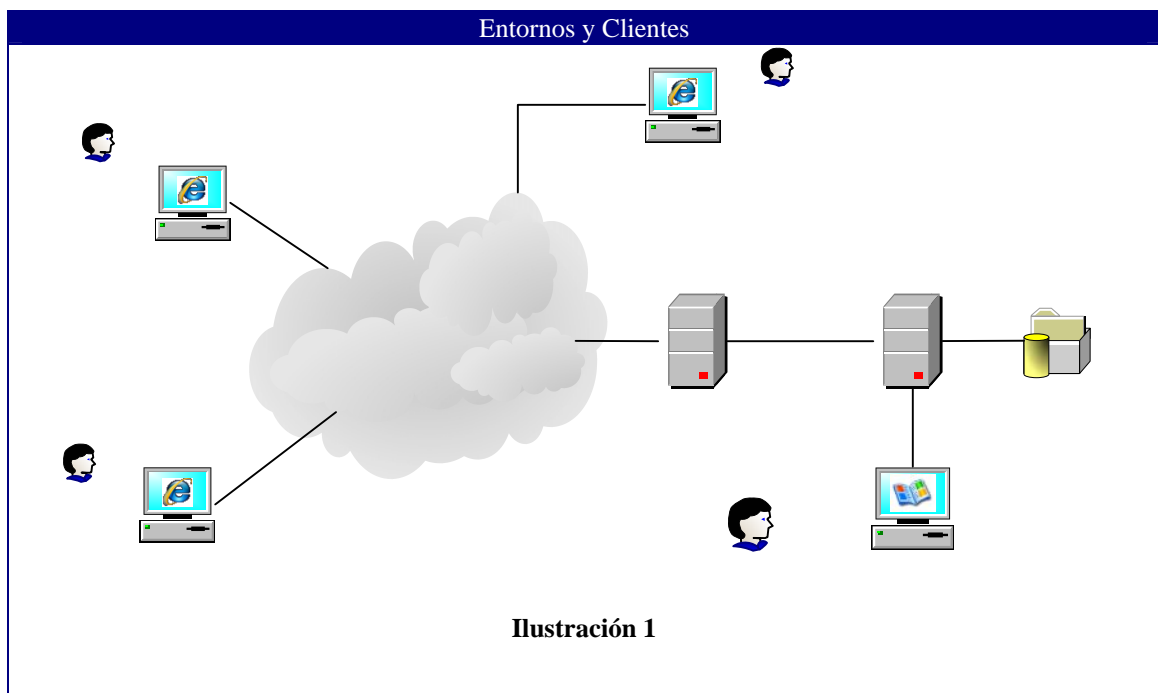
El objetivo principal de este Trabajo Fin de Carrera es: profundizar en el conocimiento de la tecnología .Net de Microsoft. Este conocimiento debe tener un carácter práctico.

Para alcanzar este objetivo, se ha desarrollado una solución software de Gestión de un Catálogo que ponga en práctica los conocimientos,

herramientas y métodos aprendidos en las asignaturas de la Ingeniería Técnica de Informática de Gestión en la UOC y que utilice tecnologías .Net así como Patrones de Diseño MVC.

### 1.3. Enfoque y Método seguido

Para el estudio y aplicación de tecnologías .Net se ha desarrollado la Gestión del Catálogo On line de una tienda de miniaturas para pintar. La tienda ofrece los productos a nuestros Clientes y estos nos llaman por teléfono para realizar la compra: la gestión del Catálogo ofrece información sobre los productos; no es una solución de comercio electrónico completa donde el cliente realice una compra online (se deja abierta esta vía para posibles futuros proyectos o ampliaciones de funcionalidades).



El gestor de catálogos tiene dos clientes bien diferenciados:

- Un navegador Web que permite a nuestros Clientes realizar consultas sobre los productos del catálogo y ven las ofertas.
- Una aplicación cliente que se instala en un puesto de gestión donde se mantienen los productos (altas, bajas y modificaciones) y se actualizan las ofertas.

Es una aplicación en tres niveles basada en patrones MVC (Model – View – Controller) que para la presentación en el cliente de mantenimiento utilizará la tecnología Windows Presentation Foundation. Se desarrollarán Web Services para atender las peticiones de consulta y mantenimiento.



#### 1.4. Planificación del proyecto

Para la planificación del proyecto se han tomado en cuenta los siguientes factores:

- **Ajuste a las fechas de entrega de PECs:** se han ajustado las fechas al calendario propuesto para el TFC en el aula de la asignatura.
- **No se han puesto disponibles los fines de semana:** dota de flexibilidad al proyecto ya que en caso de no cumplir los plazos semanales, se pueden aprovechar los fines de semana para realizar las tareas correctoras necesarias con intención de ajustarnos a la planificación.

La planificación propuesta del proyecto es la siguiente (siguiente página):

Id	Nombre de tarea	Duración	Comienzo	Fin	eb '07	05 mar '07				26 mar '07			16 abr '07			07 may '07			28 may '07			18 jun '07			09	
					D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X				
1	Selección del TFC	3 días	jue 01/03/07	lun 05/03/07																						
2	<b>Definición del TFC</b>	<b>6 días?</b>	<b>mar 06/03/07</b>	<b>mar 13/03/07</b>																						
3	Descripción del Proyecto	2 días	mar 06/03/07	mié 07/03/07																						
4	Planificación del Proyecto	2 días	jue 08/03/07	vie 09/03/07																						
5	Primer borrador	1 día?	lun 12/03/07	lun 12/03/07																						
6	Revisión y mejoras	1 día?	mar 13/03/07	mar 13/03/07																						
7	Entrega Definición del proyecto	0 días	mar 13/03/07	mar 13/03/07																						
8	<b>Especificación y diseño.</b>	<b>20 días</b>	<b>mié 14/03/07</b>	<b>mar 10/04/07</b>																						
9	<b>Subsistema de Consultas</b>	<b>6 días</b>	<b>mié 14/03/07</b>	<b>mié 21/03/07</b>																						
10	Diagrama de Clases	2 días	mié 14/03/07	jue 15/03/07																						
11	Casos de Uso	2 días	vie 16/03/07	lun 19/03/07																						
12	Definición del interfaz	2 días	mar 20/03/07	mié 21/03/07																						
13	<b>Subsistema de Mantenimiento</b>	<b>4 días</b>	<b>jue 22/03/07</b>	<b>mar 27/03/07</b>																						
14	Diagrama de Clases	2 días	jue 22/03/07	vie 23/03/07																						
15	Casos de Uso	2 días	lun 26/03/07	mar 27/03/07																						
16	<b>Subsistema Cliente</b>	<b>6 días</b>	<b>mié 28/03/07</b>	<b>mié 04/04/07</b>																						
17	Diagramas de Clase	2 días	mié 28/03/07	jue 29/03/07																						
18	Casos de Uso	2 días	vie 30/03/07	lun 02/04/07																						
19	Definición del interfaz	2 días	mar 03/04/07	mié 04/04/07																						
20	Diseño de la persistencia	2 días	jue 05/04/07	vie 06/04/07																						
21	Generación de la documentación	2 días	lun 09/04/07	mar 10/04/07																						
22	Entrega de la Especificación y Diseño	0 días	mar 10/04/07	mar 10/04/07																						
23	<b>Implementación y Pruebas</b>	<b>34 días</b>	<b>mié 11/04/07</b>	<b>lun 28/05/07</b>																						
24	Subsistema de Consultas	8 días	mié 11/04/07	vie 20/04/07																						
25	Subsistema de Mantenimiento	8 días	lun 23/04/07	mié 02/05/07																						
26	Subsistema Cliente	8 días	jue 03/05/07	lun 14/05/07																						
27	Pruebas Funcionales y de Integración	5 días	mar 15/05/07	lun 21/05/07																						
28	Generación de la documentación	5 días	mar 22/05/07	lun 28/05/07																						
29	Entrega de Implementación y Pruebas	0 días	lun 28/05/07	lun 28/05/07																						
30	<b>Memoria y presentación</b>	<b>10 días</b>	<b>mar 29/05/07</b>	<b>lun 11/06/07</b>																						
31	Generación de la Memoria	5 días	mar 29/05/07	lun 04/06/07																						
32	Presentación del Proyecto	5 días	mar 05/06/07	lun 11/06/07																						
33	<b>Entrega Final</b>	<b>0 días</b>	<b>lun 11/06/07</b>	<b>lun 11/06/07</b>																						

## 1.5. Productos obtenidos

A la conclusión del desarrollo del proyecto, hay dos productos bien diferenciados:

- **Memoria del Proyecto:** que recoge el trabajo y las- decisiones tomadas lo largo de las distintas fases del proyecto.
- **Solución software de Gestión del Catálogo Online:** solución completa de la gestión de un catálogo online de figuras. Está compuesto por varios módulos y utilizando tecnologías .Net. Este catálogo esta compuesto por varios proyectos en .Net correspondientes: acceso a datos, subsistema de consultas, subsistema de mantenimiento, cliente Web y cliente de mantenimiento.

## 1.6. Siguietes capítulos

En los siguientes capítulos de la memoria recogen el trabajo del desarrollo normal de un proyecto:

- **Especificación del Proyecto:** donde se definirán las distintas piezas que compondrán la solución. Se incluye un capítulo a explicando las tecnologías utilizadas en el proyecto.
- **Análisis de Requisitos:** con el diagrama de clases estático y los casos de uso correspondientes.
- **Diseño de la Aplicación:** donde se presenta la arquitectura de la solución, los diagramas de clases de diseño y diagramas de secuencia cuando sean necesarios.

## 2. Especificación del Proyecto

---

En este capítulo se concretan las funcionalidades que va a tener el Gestor del Catálogo Online: la definición de los subsistemas, cuáles son los requisitos funcionales, qué piezas van a componer la solución y quiénes son los usuarios de la aplicación. Además debido a la naturaleza del proyecto, se incluye un apartado exponiendo las tecnologías utilizadas.

### 2.1. Definición de los subsistemas

Para la Gestión del Catálogo de productos, se han establecido los siguientes subsistemas:

- **Subsistema de Consulta:** gestionará todas las peticiones de consulta sobre nuestros productos. Estas peticiones se realizarán desde clientes Web.
- **Subsistema de Mantenimiento:** responsable de las peticiones altas, bajas y modificaciones de los productos y las ofertas, así como del acceso a los datos para su modificación.
- **Subsistema Cliente:** responsable de presentar al cliente el interface necesario para el mantenimiento de los productos y ofertas del catálogo.

### 2.2. Requisitos funcionales

Estos son los requisitos funcionales de la solución, agrupados por cada subsistema.

#### ***Subsistema de Consulta***

Proporcionar un interface Web para que posibles Clientes puedan realizar consultas sobre nuestros productos desde su ordenador personal a través de Internet. Estas consultas podrán ser por tipo de producto, categoría, precio o disponibilidad.

Proporcionar un interface Web para que posibles Clientes puedan acceder a las ofertas especiales de ciertos productos con descuento.

### ***Subsistema de Mantenimiento***

Proporcionar los servicios Web de mantenimiento del catálogo (alta baja y modificación de productos) para que el usuario administrador, mediante el subsistema cliente, realice las tareas necesarias del mantenimiento.

Proporcionar los servicios Web para la gestión de ofertas del catálogo, mediante el subsistema cliente.

### ***Subsistema Cliente***

Responsable del interface gráfico de la aplicación de mantenimiento del usuario final. Desde este interface se realizarán las tareas de mantenimiento del catálogo de productos y ofertas.

## **2.3. Composición del software**

La solución de Gestión de Catálogos consta de las siguientes partes:

- **Gestor de Base de Datos:** donde se almacenarán los distintos productos con sus características y precios así como las ofertas disponibles. El motor elegido es MS SQL Server 2005.
- **Web Services:** que proporcionarán el acceso a los datos y realizarán las consultas a contra la base de datos; también será responsable del mantenimiento de los productos. Se diferenciarán los servicios de consultas respecto de los de mantenimiento para mejorar la seguridad de la solución. Se desarrollará en C# y el servidor Web elegido es IIS versión 6.
- **Cliente Web:** que presentará la pantalla de consulta así como los resultados obtenidos. Debe ser Microsoft Internet Explorer 6 o superior.
- **Cliente Pesado:** desde dónde el usuario se realizará las tareas de mantenimiento. Para su interface se utilizará WPF

## **2.4. Usuarios de la aplicación**

La aplicación constará de dos usuarios distintos:

- **Usuario Administrador:** responsable del mantenimiento del catálogo de productos y de las ofertas.
- **Usuario de Consulta:** usuario anónimo que desde Internet preguntará al catálogo sobre nuestros productos y ofertas.

## **2.5. Tecnologías Utilizadas**

El objetivo de este proyecto es profundizar en las Tecnologías .Net y en Patrones de Diseño (MVC). Este apartado recoge una breve introducción a las tecnologías utilizadas en el proyecto: ADO.net para el acceso a datos, WebServices para las distintas capas de la aplicación, Windows Presentation Foundation para definir el interface del cliente pesado y Patrones de Diseño MVC para el cliente Web.

### **2.5.1. ADO.Net**

Es la última tecnología de Microsoft para el acceso a datos. Se puede acceder a distintos orígenes tales como SQL Server, XML y 'data sources' de OLE DB y ODBC. ADO.Net es similar a ADO aunque nativo en .Net. Además funciona en una 'arquitectura de datos desconectada':

- Se obtienen los datos del servidor y se cachean en local.
- Se hace el tratamiento de los datos.
- Se conecta de nuevo al servidor para subir las actualizaciones o leer nuevos datos.

Este tipo de arquitectura reduce el consumo de recursos tanto en el servidor como en el cliente permitiendo una mejor escalabilidad: la mayoría de las aplicaciones consultan datos y los muestran en pantalla.

### **2.5.2. WebServices**

Permiten construir objetos cuyos métodos pueden ser llamados remotamente vía http (y en consecuencia desde Internet). Un desarrollador puede crear nuevos WebServices invocando los servicios de otros WebServices. En el caso del catálogo online los subsistemas de Consulta y Mantenimiento ofrecen servicios de consulta y mantenimiento a distintos clientes (Web y pesado). Para realizar estas operaciones utiliza el WebService AccesoADatos responsable de la conexión y acceso a las base de datos.

Una aplicación puede integrar varios 'WebServices' repartidos por todo el mundo a través de Internet, ofreciendo una dimensión a la Red: no solo se consulta e intercambia información sino también se pueden ejecutar aplicaciones.

SOAP (Simple Object Access Protocol) es un protocolo ligero y para el intercambio de mensajes que se asienta en XML, HTTP, y SMTP y fue propuesto al W3C en 1999. .Net implementa la arquitectura para que a través del intercambio de mensajes (SOAP) se pueda acceder a las propiedades y métodos de los objetos. El WSDL (Web Service Description Language) es un esquema XML que se utiliza para describir el interface de un WebService y está desarrollado por varias empresas (Microsoft, IBM, etc.)

#### **Creación de un WebService Servidor**

Para crear un WebService es necesario utilizar las clases que se encuentran dentro del espacio de nombres `System.Web.Services`. Solo es necesario añadir el atributo `[WebMethod]` al método correspondiente.

#### **Uso de un WebService por un cliente**

Cuando se desarrolla el cliente del WebService, los servicios remotos se invocan como si fueran locales. En realidad la comunicación entre un cliente y un WebService se realiza mediante un Proxy. Este Proxy encapsula las llamadas del cliente en mensajes SOAP que se envían al servidor y recuperan las respuestas como resultado.

Para crear este Proxy hay que invocar la utilidad WSDL con parámetro la URL del WebService. Crear un fichero con código fuente encargado de la encapsulación de los mensajes.

#### **2.5.3. Windows Presentation Foundation**

Es el nuevo framework a nivel de presentación para Windows XP y Windows Vista. Integra los controles (Botones, Etiquetas, Listas



Desplegables, etc.) con el tratamiento de imágenes y elementos gráficos en 3D tratándolos de la misma manera.

WPF proporciona el modelo de contenido permitiendo que cualquier control albergue otro tipo: es posible combinar elementos de una lista desplegable y que contenga iconos, imágenes y texto.

Para determinar como se presentan los controles en una ventana (si debe agrandarse o no según se maximiza la ventana, como realizar los resizes, etc.) WPF implementa varios 'contenedores de objetos' con distintos comportamientos según el tipo de contenedor, independientes del tipo de controles que albergan.

WPF también permite el enlace datos con controles, sincronizando elementos visuales con cambios en los datos. Las animaciones aumentan la sensación de interacción del usuario con las aplicaciones.

Para la definición del interface de usuario aparece un nuevo lenguaje XAML (eXtensible Application Markup Language): la ventana, los contenedores, controles y propiedades se definen en un árbol XML asociado a cada clase (ventana).

#### **2.5.4. Patrones de Diseño MVC**

Para el desarrollo del cliente ligero (cliente Web) se ha utilizado un patrón de diseño MVC (Model View Controller) que divide el cliente en tres partes específicas y especializadas:

- Modelo: correspondiente a los datos de la Aplicación.
- Vista: responsable de toda la parte de presentación e interface de usuario.
- Controlador: responde a eventos (acciones del usuario) que realizan cambios en el Modelo o la Vista.

Existen varios patrones de diseño para .Net como Spring.Net y Maverick.Net (de Apache) pero nos hemos decantado por MonoRail de Castle Project. Se ha escogido este patrón por su sencillez de uso. En el caso concreto de

MonoRails, el proyecto del cliente Web se divide en tres carpetas (Model View y Controllers); cada controlador tiene un grupo asociado de vistas. Para visualizar las vistas (ficheros con extensión vm) se ha utilizado el renderizador Nvelocity, recomendado e incluido en el proyecto Castle Project.

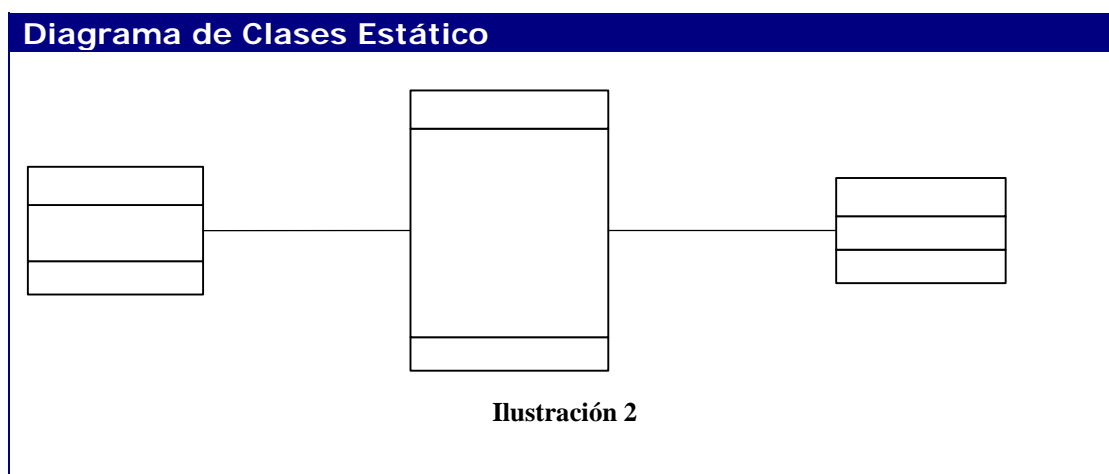
### 3. Análisis de Requisitos

El análisis de requisitos se presentará el diagrama de clases estático para el catálogo online, los casos de uso para cada subsistema y los requisitos tecnológicos.

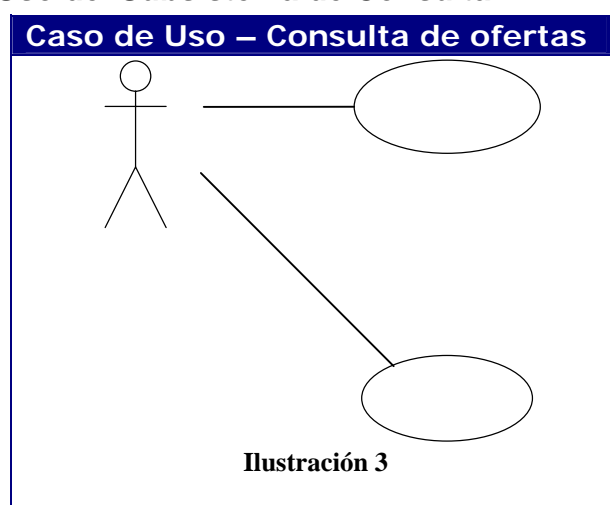
#### 3.1. Diagrama Estático de Clases

Como producto para la gestión del catálogo se han escogido miniaturas para pintar. Estas miniaturas, de diferentes escalas (28 mm, 56 mm, etc.) pueden estar fabricadas con resina o metal y pertenecen a una categoría: históricas, cine, videojuegos, fantasía, etc.

El diagrama de clases propuesto sería:



#### 3.2. Casos de Uso del Subsistema de Consulta



- **Cliente:** persona ajena a la empresa que está interesada en nuestro catálogo de figuras y sus ofertas.
- **Buscar Figura:** ofrece los criterios de búsqueda de figuras, así como los resultados de las búsquedas.
- **Buscar Oferta:** presenta las ofertas al cliente y le permite realizar búsquedas sobre las figuras de las ofertas.

### **Buscar Figura:**

---

**Descripción:** buscar una figura dentro de nuestro catálogo a través de la Web.

**Precondición:** que el usuario haya llegado al sitio Web.

**Postcondición:** mostrar todas las figuras en nuestro catálogo coincidentes con los criterios de búsqueda

**Disparador:** seleccionar [buscar figuras](#) en la página de inicio.

#### **Flujo Básico:**

- 1.- El cliente elige buscar figuras en la página principal.
- 2.- Aparece la página de criterios de búsqueda.
- 3.- El cliente introduce los criterios para buscar figuras y pulsa **Buscar**.
- 4.- Aparecen en pantalla las figuras que coinciden.

### **Buscar Oferta:**

---

**Descripción:** buscar una oferta dentro de nuestro catálogo a través de la Web.

**Precondición:** que el usuario haya llegado al sitio Web.

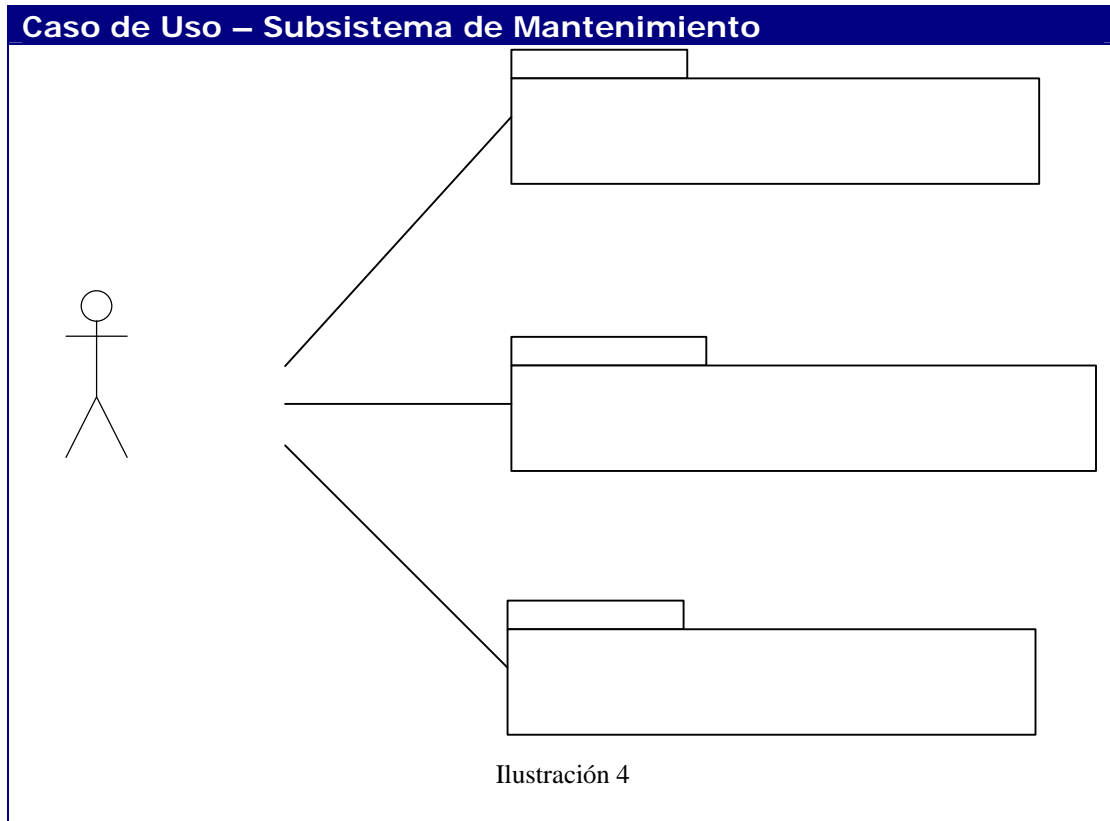
**Postcondición:** mostrar todas las ofertas en nuestro catálogo coincidentes con los criterios de búsqueda

**Disparador:** seleccionar [buscar ofertas](#) en la página de inicio.

#### **Flujo Básico:**

- 1.- El cliente elige buscar figuras en la página principal.
- 2.- Aparece la página de criterios de búsqueda.
- 3.- El cliente introduce los criterios para buscar ofertas y pulsa **Buscar**.
- 4.- Aparecen en pantalla las ofertas que coinciden.

### 3.3. Casos de Uso del Subsistema de Mantenimiento



- **Administrador:** persona que mantiene la información del catálogo online.
- **Gestionar Oferta:** paquete que contiene los casos de uso de alta, baja y modificación de ofertas.
- **Gestión de Figuras:** contiene los casos de uso del alta, baja y modificación de las figuras del catálogo.
- **Gestión de Empresas:** paquete que contiene los casos de uso de alta baja y modificación de empresas.

#### Añadir Oferta

**Descripción:** añade una oferta de una figura. Una oferta es una figura con un precio especial rebajado.

**Precondición:** que el usuario pertenezca al grupo autorizado para ejecutar la aplicación de mantenimiento y no exista una oferta sobre esa figura.

**Poscondición:** se añade una nueva oferta al catálogo.

**Disparador:** el usuario selecciona **Añadir Oferta** del menú principal.

**Flujo Principal:**

- 1.- El usuario pulsa **Añadir Oferta** en el menú principal.
- 2.- Aparece la pantalla de **Añadir Oferta**.
- 3.- El usuario selecciona la **Empresa** en el combo de empresa.
- 4.- El usuario selecciona la **Categoría** de la figura en el combo de categoría
- 5.- El usuario selecciona el **Nombre** de la figura en el combo nombre.
- 6.- Se muestra el valor del **PVP** de la figura.
- 7.- El usuario introduce el **PVP Oferta** para la oferta.
- 8.- El usuario pulsa añadir y se añade la oferta al catálogo.

**Flujo Alternativo:**

- 8.- El usuario pulsa cancelar y se cierra la pantalla.

## **Borrar Oferta**

---

**Descripción:** borra una oferta del catálogo online.

**Precondición:** que el usuario pertenezca al grupo autorizado para ejecutar la aplicación de mantenimiento y que exista la oferta que se quiere borrar.

**Poscondición:** se borra una oferta existente del catálogo.

**Disparador:** el usuario selecciona **Borrar Oferta** del menú principal.

**Flujo Principal:**

- 1.- El usuario pulsa **Borrar Oferta** en el menú principal.
- 2.- Aparece la pantalla de **Borrar Oferta**.
- 3.- El usuario selecciona la **Empresa** en el combo de empresa.
- 4.- El usuario selecciona la **Categoría** de la figura en el combo de categoría
- 5.- El usuario selecciona el **Nombre** de la figura en el combo nombre.
- 6.- Se muestra el valor del **PVP Oferta** de la figura.
- 7.- El usuario pulsa **Borrar**.
- 8.- Se pide confirmación al usuario sobre el borrado y el usuario acepta.

**Flujo Alternativo:**

- 7.- El usuario pulsa **Cancelar**.

## **Actualizar Oferta**

---

**Descripción:** cambia el precio de una oferta de una figura.

**Precondición:** que el usuario pertenezca al grupo autorizado para ejecutar la aplicación de mantenimiento y que exista la oferta que se quiere modificar.

**Poscondición:** se modifica una oferta existente del catálogo.

**Disparador:** el usuario selecciona **Actualizar Oferta** del menú principal.

### **Flujo Principal:**

- 1.- El usuario pulsa **Actualizar Oferta** en el menú principal.
- 2.- Aparece la pantalla de **Actualizar Oferta**.
- 3.- El usuario selecciona la **Empresa** en el combo de empresa.
- 4.- El usuario selecciona la **Categoría** de la figura en el combo de categoría
- 5.- El usuario selecciona el **Nombre** de la figura en el combo nombre.
- 6.- Se muestra el valor del **PVP Oferta** de la figura.
- 7.- El usuario introduce el **Nuevo PVP Oferta** para la oferta.
- 8.- El usuario pulsa **Actualizar** y se modifica la oferta al catálogo.

### **Flujo Alternativo:**

- 8.- El usuario pulsa **Cancelar** y se cierra la pantalla.

## **Añadir Empresa**

---

**Descripción:** añade una empresa proveedora de figuras.

**Precondición:** que el usuario pertenezca al grupo autorizado para ejecutar la aplicación de mantenimiento y que la empresa a dar de alta no exista.

**Poscondición:** se añade una nueva empresa proveedora al catálogo.

**Disparador:** el usuario selecciona **Añadir Empresa** del menú principal.

### **Flujo Principal:**

- 1.- El usuario pulsa **Añadir Empresa** en el menú principal.
- 2.- Aparece la pantalla de **Añadir Empresa**.
- 3.- El usuario introduce el nombre de la empresa en el campo **Empresa**.
- 4.- El usuario introduce la dirección de la empresa en el campo **Dirección**.

5.- El usuario pulsa **Añadir** y se añade la empresa.

**Flujo Alternativo:**

5.- El usuario pulsa **Cancelar**; se cierra esta ventana.

## **Borrar Empresa**

---

**Descripción:** borra una empresa proveedora de figuras.

**Precondición:** que el usuario pertenezca al grupo autorizado para ejecutar la aplicación de mantenimiento y que la empresa a borrar exista.

**Poscondición:** borra una empresa proveedora.

**Disparador:** el usuario selecciona **Borrar Empresa** del menú principal.

**Flujo Principal:**

- 1.- El usuario pulsa **Borrar Empresa** en el menú principal.
- 2.- Aparece la pantalla de **Borrar Empresa**.
- 3.- El usuario selecciona del combo **Empresa** la empresa que quiere borrar.
- 4.- El usuario pulsa **Borrar**.
- 5- Se pide confirmación al usuario sobre el borrado y el usuario acepta.

**Flujo Alternativo:**

5.- El usuario pulsa **Cancelar** y se cierra.

## **Actualizar Empresa**

---

**Descripción:** modificar los datos de una empresa proveedora de figuras.

**Precondición:** que el usuario pertenezca al grupo autorizado para ejecutar la aplicación de mantenimiento y que la empresa a modificar exista.

**Poscondición:** se modifican los datos de una empresa.

**Disparador:** el usuario selecciona **Actualizar Empresa** del menú principal.

**Flujo Principal:**

- 1.- El usuario pulsa **Actualizar Empresa** en el menú principal.
- 2.- Aparece la pantalla de **Actualizar Empresa**.
- 3.- El usuario selecciona del combo **Empresa** la empresa que quiere borrar.
- 4.- El usuario cambia la dirección de la empresa en el campo **Dirección**.
- 5- El usuario pulsa **Actualizar**.

**Flujo Alternativo:**

5.- El usuario pulsa cancelar y se cierra.



## **Añadir figura**

---

**Descripción:** añade una figura al catálogo online.

**Precondición:** que el usuario pertenezca al grupo autorizado para ejecutar la aplicación de mantenimiento y no exista ya figura que se desea añadir.

**Poscondición:** se añade una nueva figura al catálogo.

**Disparador:** el usuario selecciona **Añadir Figura** del menú principal.

### **Flujo Principal:**

- 1.- El usuario pulsa **Añadir Figura** en el menú principal.
- 2.- Aparece la pantalla de **Añadir Figura**.
- 3.- El usuario introduce el **Nombre** de la figura en el campo nombre.
- 4.- El usuario introduce la **Descripción** el campo descripción.
- 5.- El usuario selecciona la **Empresa** en el combo de empresa.
- 6.- El usuario selecciona la **Categoría** de la figura en el combo de categoría.
- 7.- El usuario selecciona la **Material** de la figura en el combo de material.
- 8.- El usuario introduce la **Escala** de la figura en el combo escala.
- 9.- El usuario introduce la **Cantidad** de la figura en el campo cantidad.
- 10.- El usuario introduce el **Precio** de la figura en el campo precio.
- 11.- El usuario pulsa **Añadir** y se añade la figura al catálogo.

### **Flujo Alternativo:**

- 11.- El usuario pulsa **Cancelar** y se cierra la pantalla.

## **Borrar figura**

---

**Descripción:** borra una figura al catálogo online.

**Precondición:** que el usuario pertenezca al grupo autorizado para ejecutar la aplicación de mantenimiento y que exista ya figura que se desea borrar.

**Poscondición:** se borra una figura del catálogo.

**Disparador:** el usuario selecciona **Borrar Figura** del menú principal.

### **Flujo Principal:**

- 1.- El usuario pulsa **Borrar Figura** en el menú principal.

- 2.- Aparece la pantalla de **Borrar Figura**.
- 3.- El usuario elige la **Empresa** de la figura que quiere borrar en el combo empresa.
- 4.- El usuario elige la **Categoría** de la figura que quiere borrar en el combo categoría.
- 5.- El usuario elige el **Nombre** de la figura que quiere borrar en el combo nombre.
- 6.- Se muestran los datos correspondientes a la figura seleccionada: **Nombre, Descripción, Empresa, Categoría, Cantidad, Material, Escala y Precio**.
- 7.- El usuario pulsa **Borrar** y se borra la figura al catálogo.
- 8.- Se pide confirmación del borrado y el usuario acepta.

#### **Flujo Alternativo:**

- 8.- El usuario pulsa **Cancelar** y se cierra la pantalla.

#### **Actualizar figura**

---

**Descripción:** modifica una figura al catálogo online.

**Precondición:** que el usuario pertenezca al grupo autorizado para ejecutar la aplicación de mantenimiento y que exista la figura que se desea modificar.

**Poscondición:** se modifica una figura del catálogo.

**Disparador:** el usuario selecciona **Actualizar Figura** del menú principal.

#### **Flujo Principal:**

- 1.- El usuario pulsa **Actualizar Figura** en el menú principal.
- 2.- Aparece la pantalla de **Actualizar Figura**.
- 3.- El usuario elige la **Empresa** de la figura que quiere modificar en el combo empresa.
- 4.- El usuario elige la **Categoría** de la figura que quiere modificar en el combo categoría.
- 5.- El usuario elige el **Nombre** de la figura que quiere modificar en el combo nombre.

6.- Se muestran los datos correspondientes a la figura seleccionada:  
**Nombre, Descripción, Empresa, Categoría, Cantidad, Material, Escala y Precio.**

7.- El usuario modifica los datos correspondientes de la figura seleccionada.

8.- El usuario pulsa **Actualizar** y se modifica la figura del catálogo.

**Flujo Alternativo:**

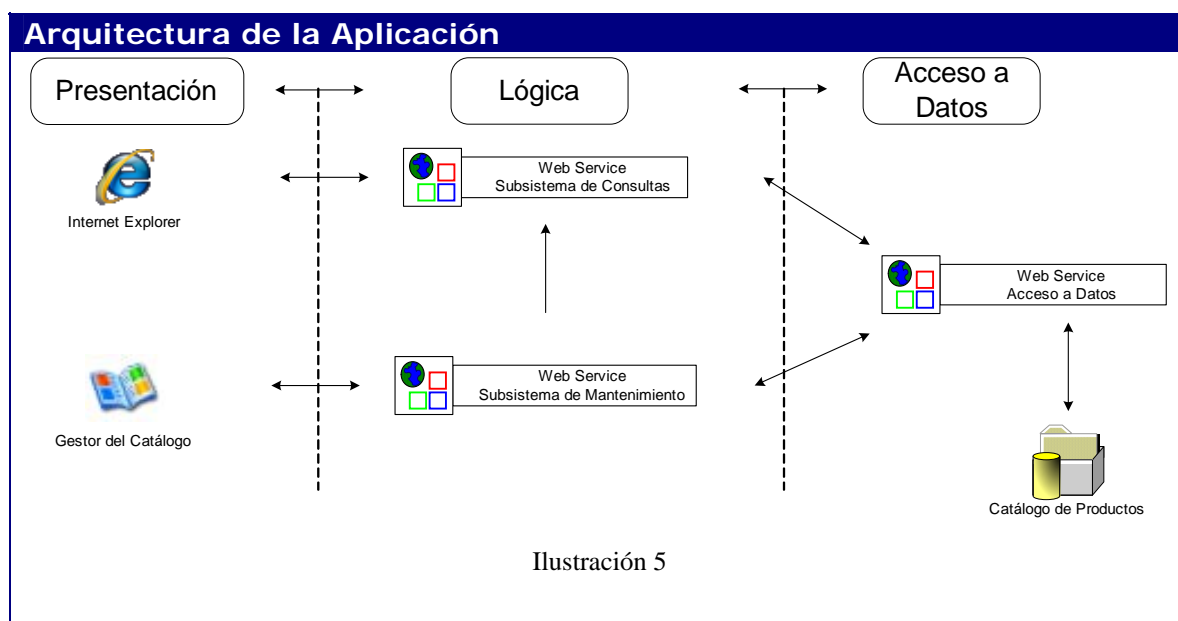
8.- El usuario pulsa **Cancelar** y se cierra la pantalla.

## 4. Diseño

En el apartado de Diseño se presentará la arquitectura del sistema, las clases de diseño de cada subsistema, el diagrama de secuencia del subsistema de consultas, el diseño del interface gráfico y el diseño de la base de datos.

### 4.1. Arquitectura de la aplicación

La arquitectura consta de tres niveles de la aplicación se presenta en la siguiente figura:



#### *Capa de Presentación*

Se ejecuta en el puesto final y hay dos interfaces de usuario:

- El **Navegador Web** Internet Explorer para realizar consultas sobre las ofertas y productos; solicita la lista de los productos y ofertas al Subsistema de Consultas.
- El **Gestor del Catálogo** desde el que se realizan las tareas de mantenimiento del catálogo: altas, bajas y modificaciones de productos y ofertas. Realiza las solicitudes al Subsistema de Mantenimiento.

#### *Capa de Lógica de Negocio*

Servidor IIS donde se sirven las páginas Web a los clientes y se ejecutan los Web Services de consulta y mantenimiento:

- **Subsistema de Consultas:** este Web service es responsable realizar las consultas a la base de datos y devolver la página al cliente.
- **Subsistema de Mantenimiento:** atiende las peticiones del Gestor del Catálogo. Durante los procesos de mantenimiento, también lanza peticiones al Subsistema de Consultas.

#### ***Capa de Acceso a Datos:***

Independiza el repositorio de los datos respecto de la aplicación. Contiene las clases de Entidad y es responsable de la conexión con la base de datos. Proporciona el acceso a los datos a la capa de lógica de negocio.

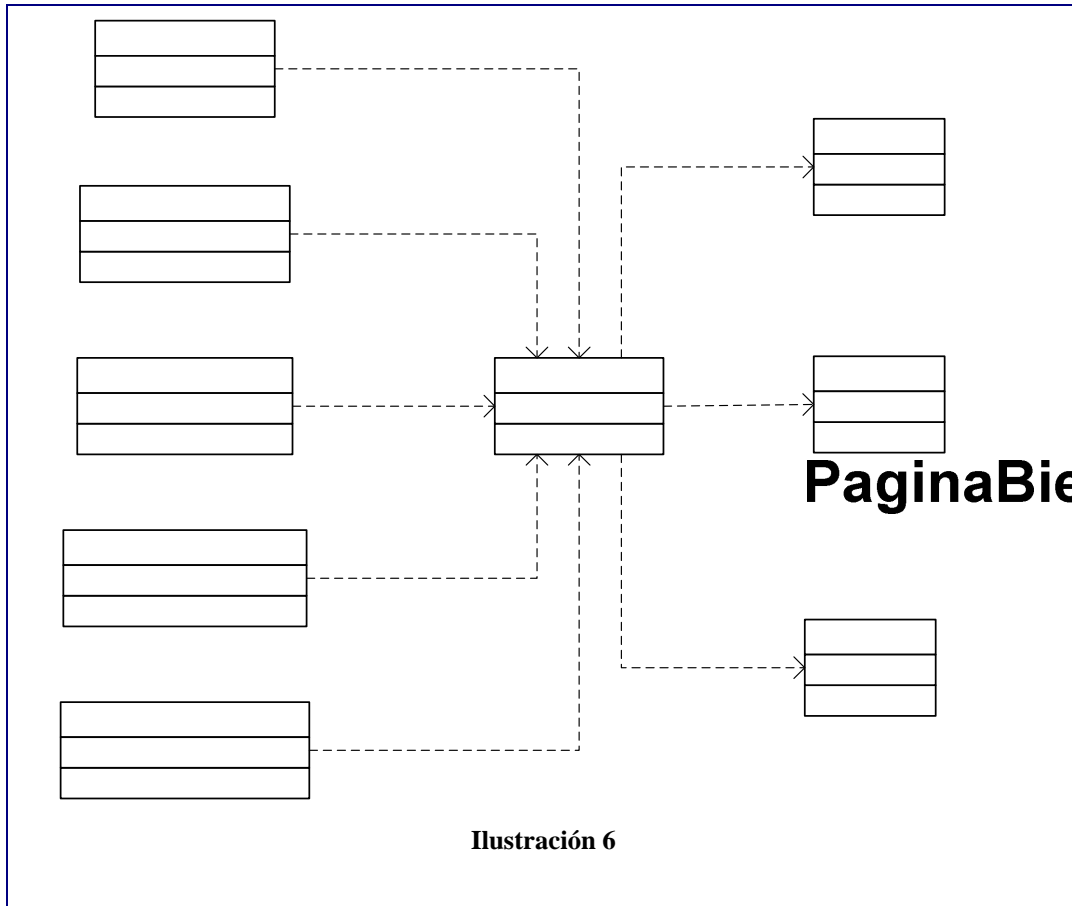
## **4.2. Diagramas de Clases del Diseño y Diagramas de secuencia**

Se presentan los diagramas de clases de diseño y de secuencia para cada subsistema.

### ***Subsistema de consultas***

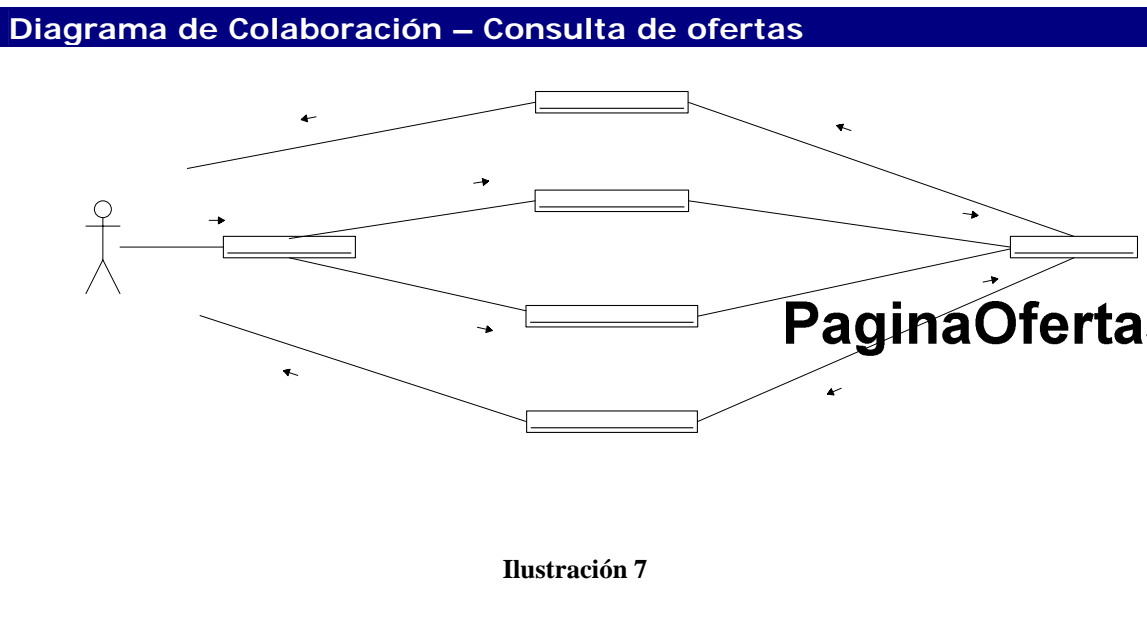
El diagrama de clases del diseño propuesto para el subsistema de consultas es el siguiente:

#### **Diagrama de clases – Subsistema de Consultas**



PaginaBienvenida

A continuación se presenta el **diagrama de colaboración** de **PaginaOfertasBuscador** y **Consultas de Ofertas**. Se decide presentar el diagrama de colaboración porque recoge el flujo que sigue el cliente a través de las páginas Web de la solución.



PaginaOfertasMostr

PaginaProductosMostr

Descripción: un Cliente desde Internet se conecta a nuestra Web y realiza una consulta sobre nuestras ofertas.

Actores: Cliente que consulta nuestro catálogo.

Precondiciones: ninguna.

Postcondiciones: se muestra al Cliente los productos seleccionados según su criterio.

Disparador: el Cliente se conecta a nuestra Web.

Flujo:

1. El Cliente se conecta a nuestra Web y aparece la pantalla de bienvenida con las ofertas de nuestro catálogo.
2. El Cliente selecciona buscar figuras en nuestro catálogo.
  - 2.1. Se muestra un formulario con los criterios de búsqueda de una figura.
  - 2.2. El Cliente define los criterios de búsqueda de la figura y solicita la búsqueda.
  - 2.3. Aparecen las figuras que cumplen los criterios de búsqueda.
3. El Cliente selecciona buscar una oferta.
  - 3.1. Se muestra un formulario con los criterios de búsqueda de ofertas.
  - 3.2. El Cliente define los criterios de búsqueda de la oferta y solicita la búsqueda.
  - 3.3. Aparecen las ofertas que cumplen los criterios de búsqueda.

### ***Subsistema de Mantenimiento***

Se encarga del mantenimiento de productos y ofertas: proporciona los servicios para el alta, baja y modificación en nuestro catálogo.

## **Diagrama de clases – Subsistema de Mantenimiento**

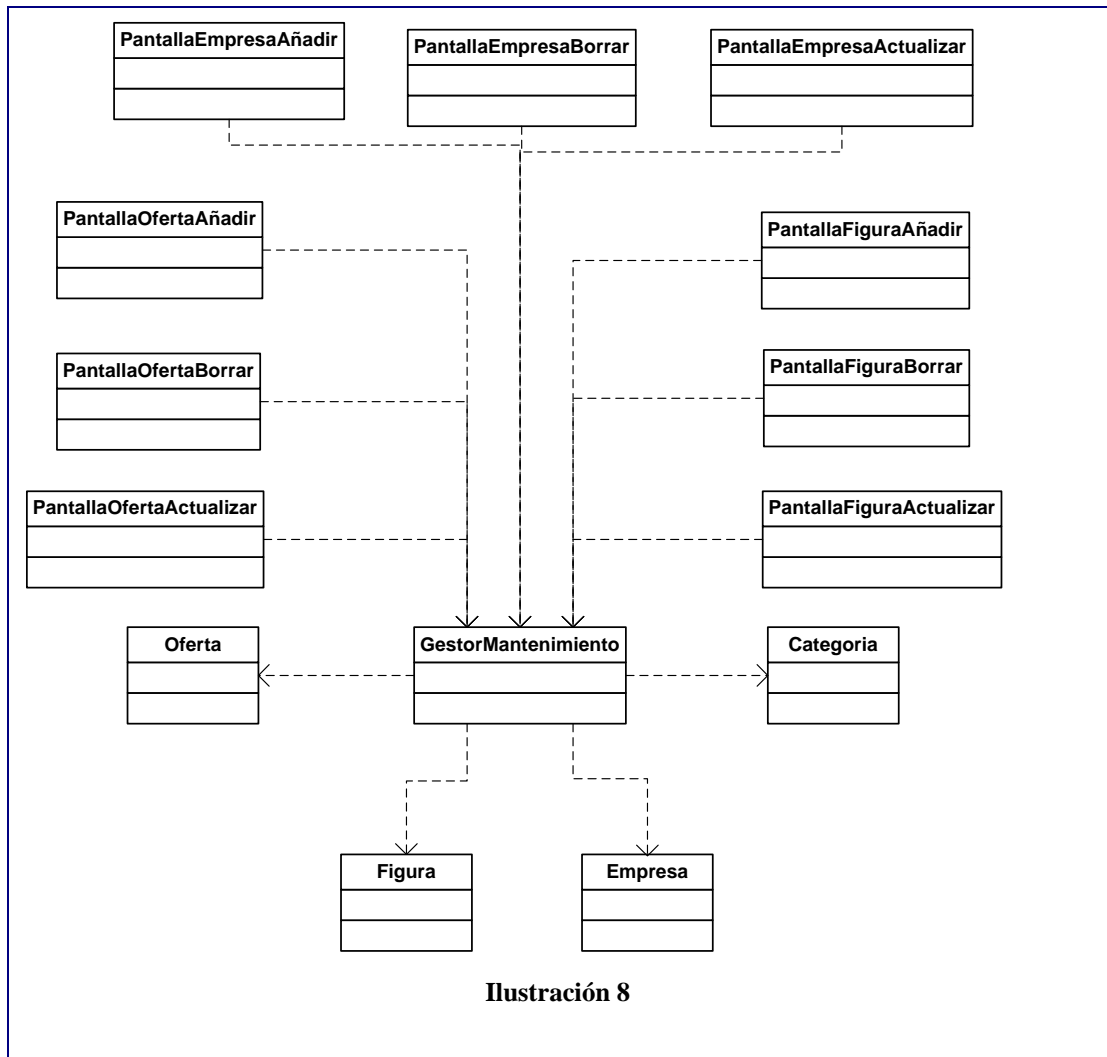
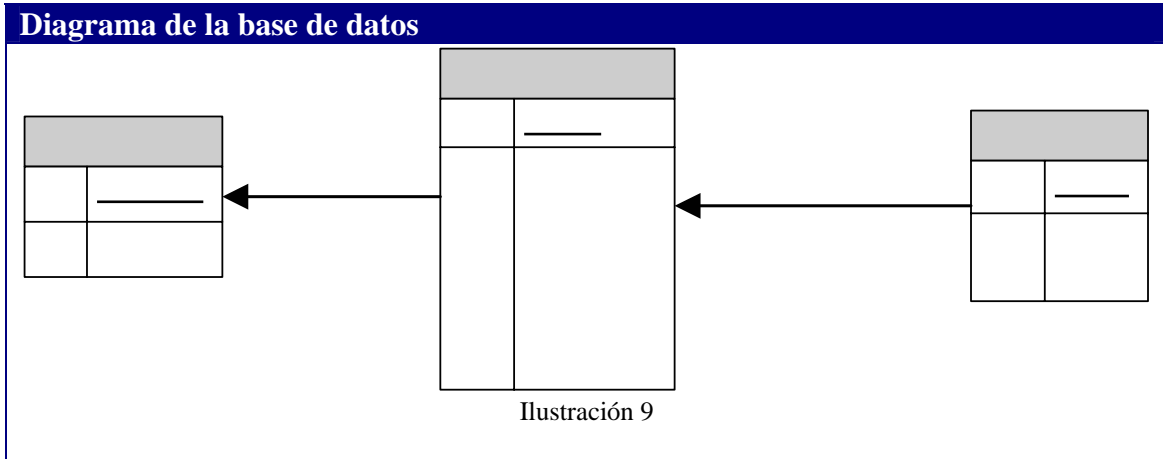


Ilustración 8



### 4.3. Diseño de la persistencia

El diagrama de persistencia de la base de datos es bastante simple:



El modelo relacional quedaría de la siguiente manera:

EMPRESAS (Empresa, Dirección);

FIGURAS (Figura, Material, Descripción, Cantidad, Precio, Escala, Categoría, Empresa) donde Empresa hace referencia a EMPRESAS.

OFERTAS (Oferta, Figura, Precio) donde Figura hace referencia a FIGURAS.

Empresas

PK Empresa

Direccion

#### 4.4. Interface de Usuario

A continuación se presentan las pantallas del interface de usuario.

##### **Servidor Web**

---

Estas son las páginas que se presentan a nuestros clientes cuando contactan con nuestro Web.

##### **Pantalla de Bienvenida**

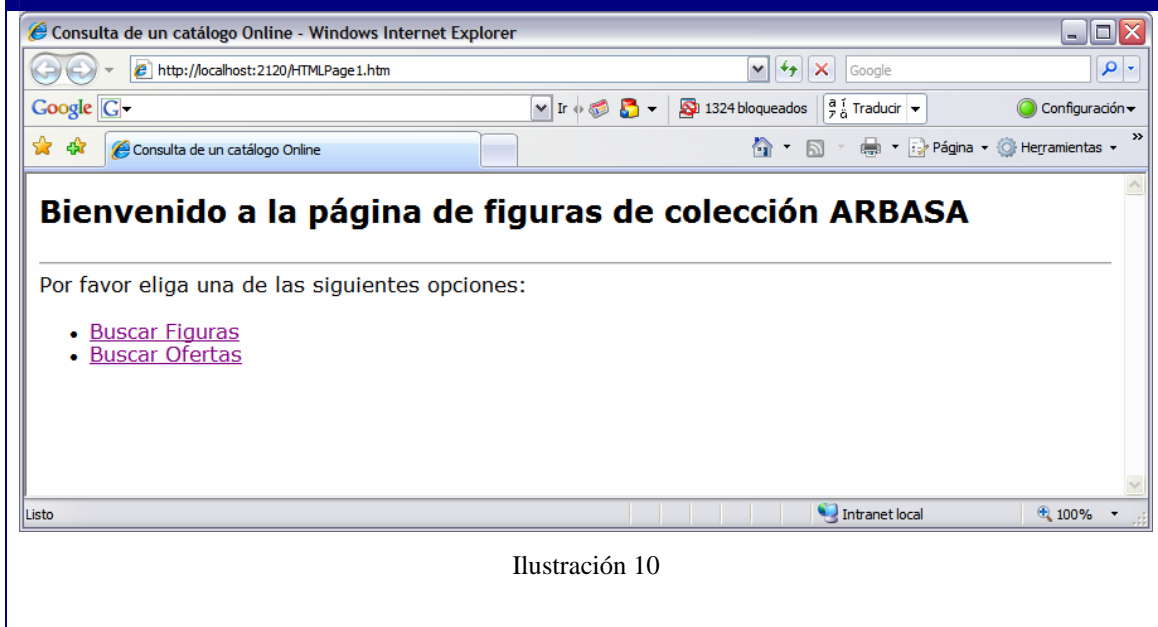
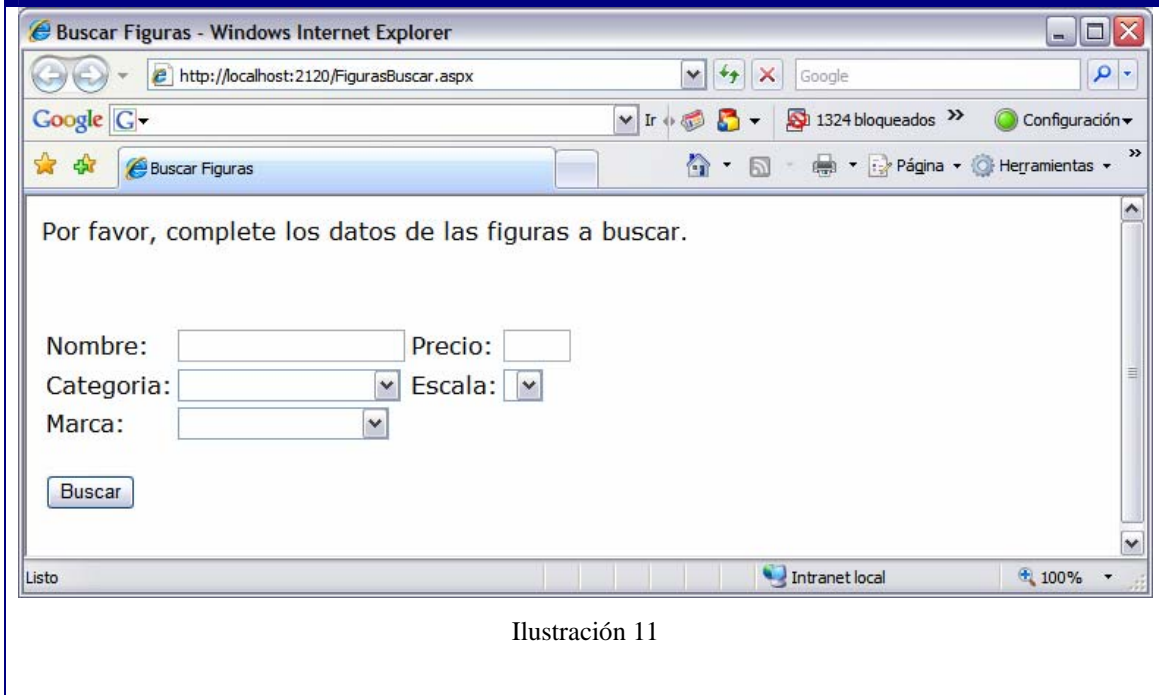


Ilustración 10

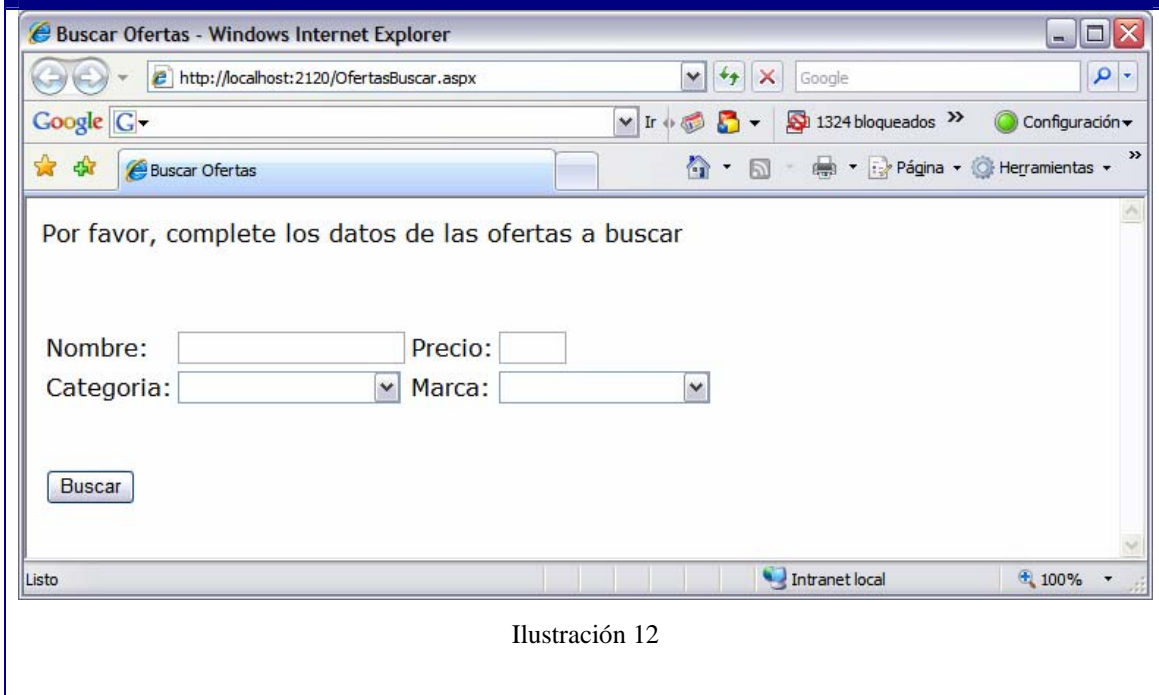
En esta pantalla se le presentan dos posibilidades a nuestros clientes: buscar figuras o buscar ofertas.

### Búsqueda de Figuras



Esta pantalla ofrece los posibles criterios de búsqueda de figuras.

### Búsqueda de Ofertas



Esta pantalla ofrece los posibles criterios de búsqueda de ofertas a nuestros posibles clientes.

### Listado de Figuras

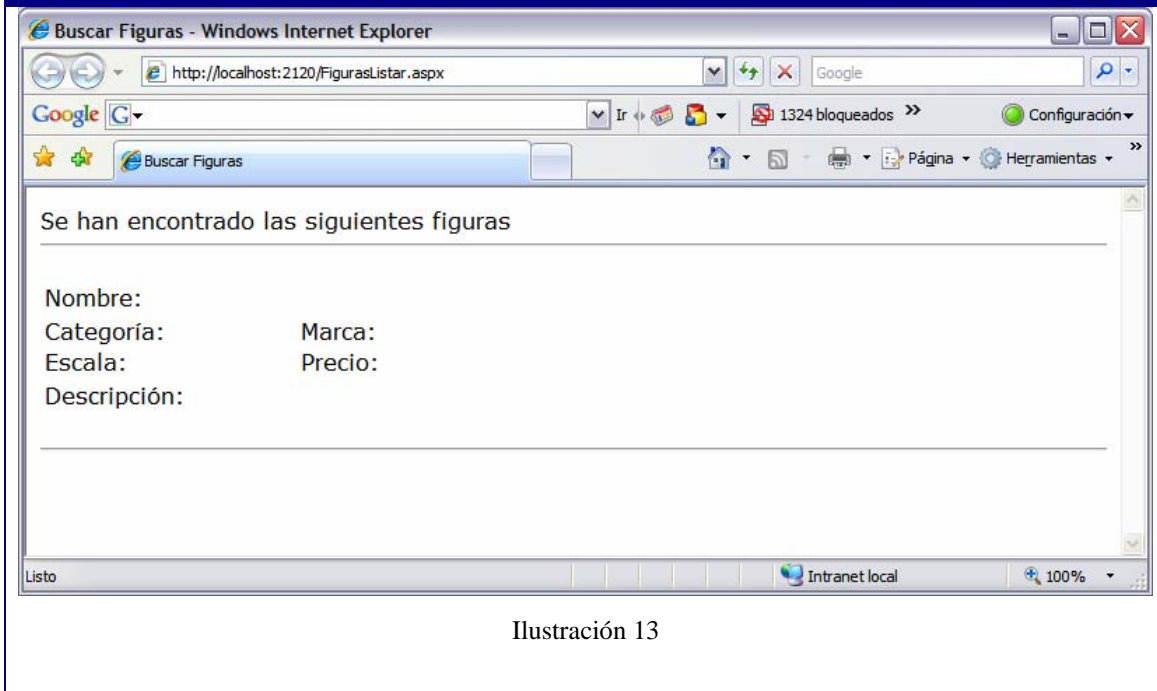


Ilustración 13

Esta pantalla informa de las figuras encontradas que cumplen los criterios puestos por el cliente.

### Listado de ofertas

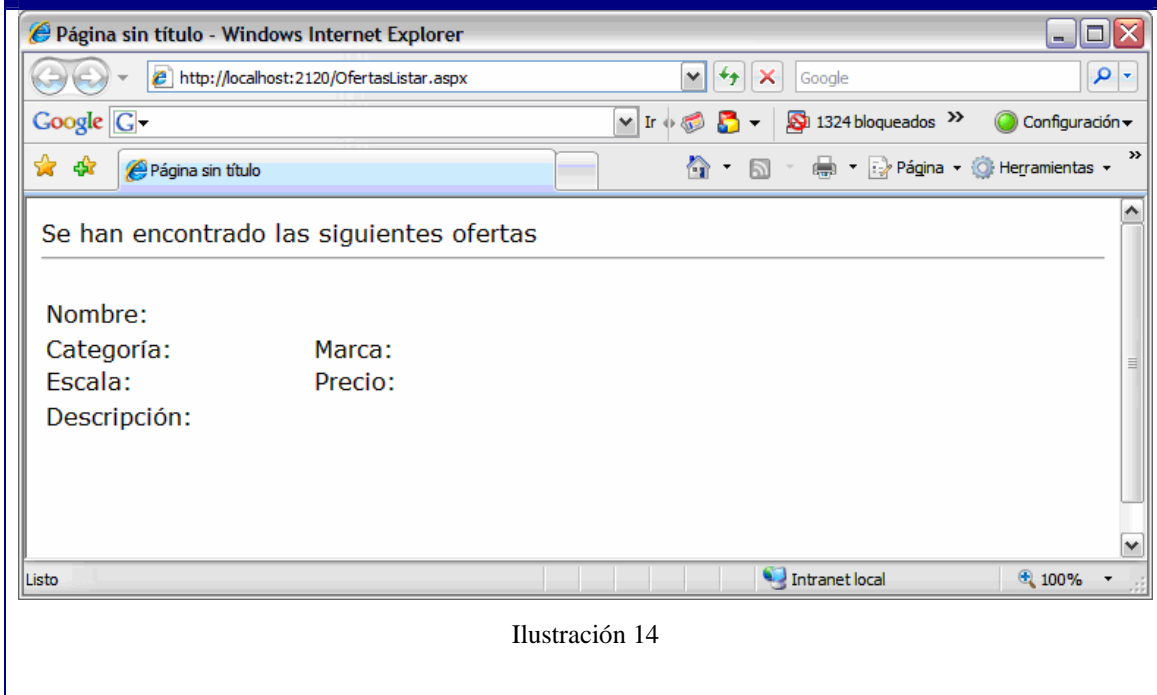


Ilustración 14

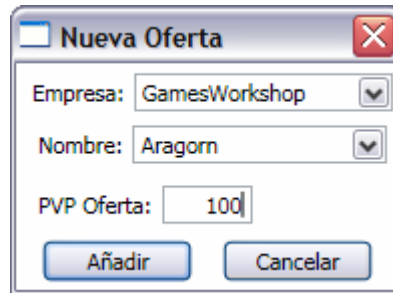
Muestra las ofertas encontradas.

**Cliente de Mantenimiento**

A continuación se presentan las pantallas correspondientes al mantenimiento.

**Nueva Oferta**

Pone una figura a un precio especial, creando una oferta:



*Ilustración 15*

Nombre	Tipo	Descripción y validaciones
Empresa	Combo	Lista de las empresas proveedoras de las figuras.
Nombre	Combo	Lista de figuras disponibles de la empresa seleccionada.
PVP Oferta	Caja de texto	Permite introducir el valor de la figura en oferta
Añadir	Botón	Modifica los datos de la figura seleccionada
Cancelar	Botón	Cierra la pantalla actual.

**Borrar Oferta**

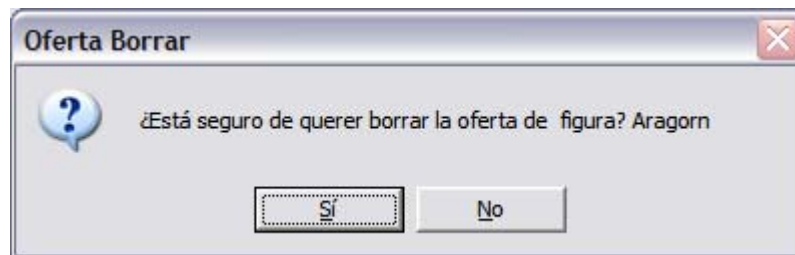
Permite eliminar una oferta de una figura:



Ilustración 16

Nombre	Tipo	Descripción y validaciones
Nombre	Combo	Lista de figuras disponibles
PVP Oferta	Etiqueta	Muestra el valor del precio de la oferta.
Borrar	Botón	Modifica los datos de la figura seleccionada
Cancelar	Botón	Cierra la pantalla actual.

Antes de realizar el borrado de la oferta se pide confirmación al usuario:



### Modificar Oferta

Permite cambiar los datos de una oferta:

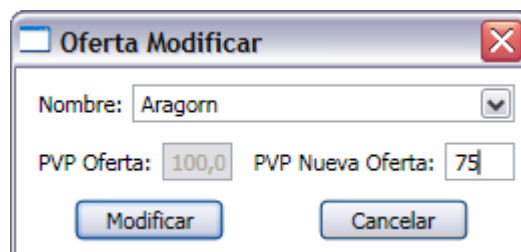


Ilustración 17

Nombre	Tipo	Descripción y validaciones
Nombre	Combo	Lista de figuras disponibles

PVP Oferta	Etiqueta	Muestra el valor del precio de la oferta.
Nuevo Oferta	PVP Caja de texto	Permite introducir el nuevo precio de la oferta
Borrar	Botón	Modifica los datos de la figura seleccionada
Cancelar	Botón	Cierra la pantalla actual.

### Añadir empresa

En esta pantalla el administrador introduce una nueva empresa:



Ilustración 18

Nombre	Tipo	Descripción y validaciones
Empresa	Caja de texto	Nombre de la empresa que se va a dar de alta.
Dirección	Caja de texto	Dirección de la empresa que se que da de alta.
Añadir	Botón	Añade los datos de la empresa.
Cancelar	Botón	Cierra la pantalla actual.

### Borrar empresa

Permite borrar una empresa del catálogo.

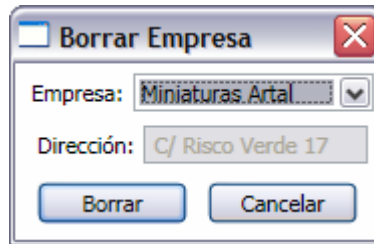
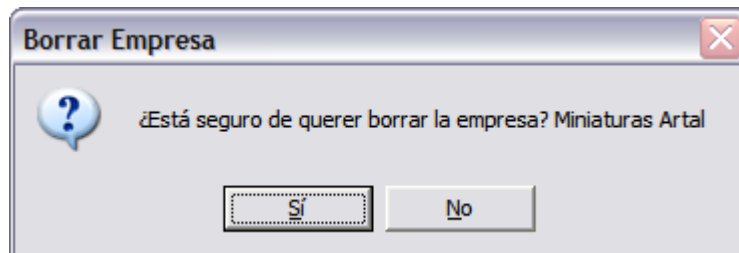


Ilustración 19

Nombre	Tipo	Descripción y validaciones
Empresa	Combo	Lista de las empresas que se pueden borrar.
Dirección	Caja de texto	Dirección de la empresa que se va a borrar.
Borrar	Botón	Borra la empresa seleccionada.
Cancelar	Botón	Cierra la pantalla actual.

Cuando se desea borrar una empresa se pide confirmación al usuario mediante la siguiente pantalla:



### Modificar empresa

Permite cambiar los datos de una empresa.



Ilustración 200



Nombre	Tipo	Descripción y validaciones
Empresa	Combo	Lista de las empresas que se pueden borrar.
Dirección	Caja de texto	Dirección de la empresa que se va a cambiar.
Modificar	Botón	Actualiza los datos de la empresa seleccionada.
Cancelar	Botón	Cierra la pantalla actual.

### Añadir figura

Introduce una nueva figura en el sistema:

Ilustración 21

Nombre	Tipo	Descripción y validaciones
Nombre	Caja de texto	de Nombre de la figura.
Descripción	Caja de texto	de Características de la figura.
Empresa	Combo	Lista de las empresas proveedoras de las figuras.
Categoría	Combo	Categoría o temática de la figura.
Cantidad	Caja de	de Stock de la figura.

	texto	
Escala	Combo	Escala de la figura.
Precio	Caja de texto	PVP de la figura.
Material	Combo	Material del que está hecha la figura.
Aceptar	Botón	Añade la figura al catálogo.
Cancelar	Botón	Cierra la pantalla actual.

### Borrar figura

Busca una figura dentro de la base de datos y ofrece la posibilidad de borrarla

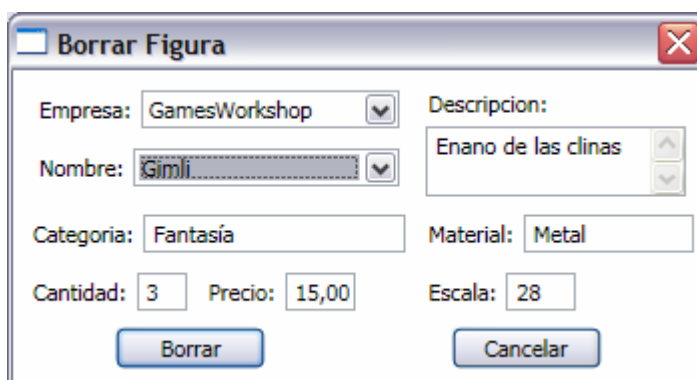
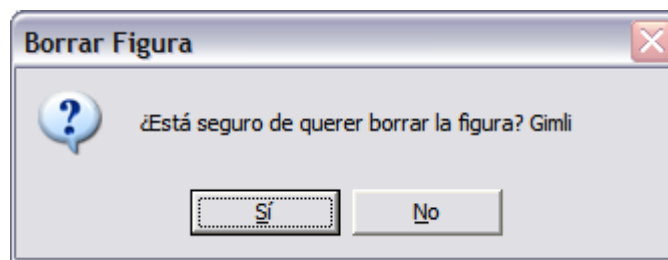


Ilustración 22

Nombre	Tipo	Descripción y validaciones
Empresa	Combo	Lista de las empresas proveedoras de las figuras
Nombre	Combo	Lista de figuras disponibles
Descripción	Caja de texto	Características de la figura
Empresa	Caja de texto	Lista de las empresas proveedoras de las figuras
Categoría	Caja de texto	Categoría o temática de la figura
Escala	Caja de texto	Escala de la figura.
Categoría	Caja de texto	Categoría o temática de la figura
Cantidad	Caja de texto	Stock de la figura

Precio	Caja de texto	PVP de la figura
Material	Caja de texto	Material del que está hecha la figura.
Borrar	Botón	Borra la figura encontrada
Cancelar	Botón	Cierra la pantalla

La pantalla de confirmación de borrado es la siguiente:



### Modificar figura

Busca una figura dentro del sistema y permite modificar sus datos:

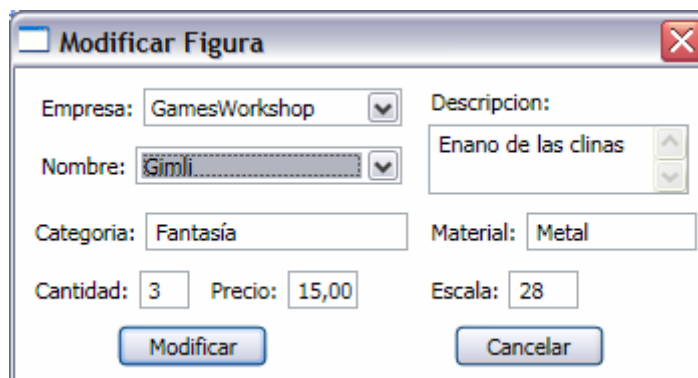


Ilustración 23

Nombre	Tipo	Descripción y validaciones
Empresa	Combo	Lista de las empresas proveedoras de las figuras
Nombre	Combo	Lista de figuras disponibles
Descripción	Etiqueta	Características de la figura
Categoría	Caja de texto	Categoría o temática de la figura

---

Cantidad	Caja de texto	Stock de la figura
Precio	Caja de texto	PVP de la figura
Material	Caja de texto	Material del que está hecha la figura.
Escala	Caja de texto	Escala de la figura.
Actualizar	Botón	Modifica los datos de la figura seleccionada
Cancelar	Botón	Cierra la pantalla actual.

---

## 5. Implementación

---

En este capítulo se destacan los aspectos importantes en la codificación en C# de los distintos módulos así como la integración con las tecnologías seleccionadas.

### 5.1. WebServices

Se ha utilizado en el Subsistema de AccesoADatos así como en los Subsistemas de Consulta y de Mantenimiento. La utilización de WebServices es realmente sencilla. Solo hay que incluir el atributo [WebMethod] en el método público que queramos ofrecer.

Un detalle importante a destacar es que un WebMethod puede devolver una clase desarrollada por nosotros, pero de manera no trivial. El primer requisito es incluir el atributo [XmlInclude(typeof(NombreClasePropia))] en la definición del método.

El cliente que use este WebService debe crear una clase Proxy con el comando WSDL. La clase Proxy creada con este comando permite invocar los servicios remotos del WebService como si fueran locales. El inconveniente está en que esta clase crea internamente su propia clase `NombreClasePropia` que es distinta de la clase `NombreClasePropia` que se codificó originalmente. La clase `NombreClasePropia` de la clase Proxy solo contiene un mapeo de los atributos y no permite utilizar los métodos correspondientes de `NombreClasePropia`.

### 5.2. Windows Presentation Foundation

Se ha utilizado para la definición del interface gráfico del cliente de mantenimiento. Al principio utilice la opción de arrastrar los componentes y que Visual Studio creara el correspondiente fichero XAML. A corto plazo

funciona pero en cuanto se realiza un reajuste del tamaño de la ventana los controles se descolocan. El principal problema es que por defecto no se utilizan 'layers' por lo que los objetos toman la política del redimensionamiento por defecto.

Usando 'layers' los controles se van metiendo dentro de estos contenedores y quedan perfectamente alineados. Su estructura es parecida a la de una tabla donde se definen filas y columnas. Destacar que un control contenedor puede a su vez contener otros controles contenedores.

Los ficheros `NombreClase.xaml` contienen la definición de los controles así como la regencia al fichero `NombreClase.cs` donde se definen los métodos de las clases. Hay que tener en cuenta que si se renombra una clase o un fichero las referencias deben seguir teniendo integridad.

### 5.3. Patrones de Diseño MVC – MonoRail

Se ha utilizado en el desarrollo del cliente Web. Únicamente destacar que es importante mantener el nombre de las clases y los ficheros. A través de los propios ejemplos que proporciona la solución se han solucionado las dudas que han surgido a lo largo del desarrollo. Únicamente destacar el formato de las url's que llevan la extensión `.rails` tal como se puede ver en los enlaces de las páginas.

### 5.4. ADO.net

Para realizar el acceso a los datos se ha utilizado ADO.net en la capa AccesoADatos. Se ha utilizado para conectarnos a la base de datos del catálogo. Para realizar las consultas y actualizaciones se han construido las correspondientes sentencias SQL.

## 6. Conclusiones

---

Gracias a este proyecto me adquirido un conocimiento básico de distintas tecnologías .Net y de Patrones de Diseño. Creo que es un proyecto difícil técnicamente hablando porque algunas de las tecnologías acaban de salir y ciertas herramientas de desarrollo están en fase Beta.

Principalmente me ha aportado conocimientos en áreas bastante vanguardistas como puede ser WPF, WebServices y Patrones de Diseño. Este conocimiento ha sido eminentemente práctico ya que he tenido que aplicarlo en una solución software real; aunque no me convierte en un experto en las materias, creo que me proporcionado el lenguaje suficiente para poder entender soluciones que usan estas tecnologías. También he aprovechado para introducirme en el lenguaje C#.

Además de los nuevos conocimientos adquiridos, ha sido muy positivo aplicar materias de las asignaturas de la carrera así como la intensa labor de investigación y búsqueda de materiales a través de bibliotecas virtuales y artículos técnicos. A nivel personal creo que el tremendo esfuerzo de horas empleadas merece mucho la pena y siempre compensa.





Como futura ampliación del proyecto se puede pensar en el desarrollo de una solución de comercio electrónico: este catálogo online ofrece información sobre productos pero para comprarlos habría que ir a la tienda o encargarlo por teléfono.

## 7. Glosario y Bibliografía

A continuación se ofrecen la definición de una serie de términos utilizados a lo largo del proyecto:

Nombre	Descripción y validaciones
ASP.Net	Active Server Pages: Conjunto de tecnologías desarrolladas por Microsoft para el desarrollo de páginas Web.
ADO.Net	Active Data Object: Librerías de acceso a datos de framework de .NET
Patrones MVC	Model – View – Controller: Patrón de diseño que se para el modelo de datos, la interacción del usuario y la lógica de negocio.
WebServices	Colección de estándares y protocolos que permiten intercambiar datos y aplicaciones entre ordenadores independientemente su sistema operativo.
WPF	Windows Presentation Foundation: Subsistema gráfico de .Net 3.0.

### Bibliografía y Referencias Web

	C# Cookbook, 2nd Edition de Jay Hilyard, Stephen Teilhet Editorial: O'Reilly
	Programing ASP.NET, 3rd Edition de Dan Hurwitz, Jesse Liberty Editorial: O'Reilly
	Programing C#, 4th Edition de Jesse Liberty Editorial: O'Reilly
	Programing Windows Presentation Foundation de Ian Griffiths, Chris Sells Editorial: O'Reilly
CodeProject	<a href="http://www.codeproject.com/">http://www.codeproject.com/</a>
MonoRail	<a href="http://www.castleproject.org/monorail/index.html">http://www.castleproject.org/monorail/index.html</a>
Microsoft WPF	<a href="http://msdn2.microsoft.com/es-es/default.aspx">http://msdn2.microsoft.com/es-es/default.aspx</a> , <a href="http://wpf.netfx3.com/">http://wpf.netfx3.com/</a>