

T
F
C
N
E
T



Arantza Jiménez Aranda

7 Enero de 2008

Consultor: David Gañan Jiménez



TFC.NET

Gracias a ti, mi niño, que has sabido entender a tu corta edad, que tu madre tenía que robarte tiempo para estudiar, sin llegar a entenderlo. Y a ti, mi niña, por ser tan especial.

Resumen

Hoy en día hay muchos ámbitos donde es necesario realizar listas jerárquicas con el fin de estructurar muy distintos tipos de información.

El proyecto “MisListas” está diseñado especialmente para usuarios domésticos, permitiéndoles a estos construir listas con distintos contenidos como: Listas de la compra, de tareas, citas, cumpleaños, reuniones, árboles genealógicos y en definitiva cualquier tipo de lista jerárquica que deseen guardar en nuestra aplicación.

El objetivo de este trabajo fin de carrera es el diseño e implementación de un conjunto ilimitado de listas, para que el usuario pueda ir creando distintos contenidos que posteriormente pueden ser consultados, modificados o filtrados.

Es fundamental crear una base sólida de características eficientes que cumpla la funcionalidad descrita, de forma que pueda ir ampliándose en un futuro para incorporar nuevas funcionalidades, entornos y propiedades. Con ellos vamos a conseguir que futuras aplicaciones reales cuesten menos tiempo de análisis y desarrollo, al disponer de unos cimientos claros sobre los que basarse.

Hemos diferenciado dos tipos de aplicaciones, con el fin de utilizar distintas tecnologías.

- Aplicación de escritorio que permitirá ir construyendo el árbol jerárquico.
- Aplicación web de consulta, filtrado, búsqueda y ordenación de las listas jerárquicas obtenidas.

La aplicación se ha desarrollado completamente utilizando la plataforma Microsoft .NET 2.0 y 3.0. con el entorno de desarrollo Microsoft Visual Studio 2005. El lenguaje de programación utilizado ha sido C# y como Sistema Gestor de Bases de Datos se ha utilizado Microsoft SQL Server Express 2005. Además se han utilizado distintas tecnologías como: Asp.net, acceso a datos ADO.NET, Windows Presentation Foundation para darle un aspecto gráfico mejorado y Ajax para el desarrollo web.

INDICE

<u>I. INTRODUCCIÓN</u>	5
1. JUSTIFICACIÓN DEL TFC	5
2. OBJETIVOS DEL TFC	6
3. ENFOQUE Y MÉTODO SEGUIDO	6
4. TECNOLOGÍA A UTILIZAR	7
5. PLANIFICACIÓN DEL PROYECTO	9
6. PRODUCTOS OBTENIDOS	11
7. BREVE INTRODUCCIÓN DE OTROS CAPÍTULOS	11
<u>II. CUERPO DE LA MEMORIA</u>	12
1. ANÁLISIS DEL PROYECTO	12
1.1. DESCRIPCIÓN	12
1.2. ANÁLISIS DE REQUISITOS Y FUNCIONALIDADES	12
1.3. DIAGRAMA DE CASOS DE USO	16
1.4. MODELO CONCEPTUAL DE CLASES	25
2. DISEÑO	27
2.1. DISEÑO DE LA ARQUITECTURA DEL SISTEMA	27
2.2. DISEÑO DE CLASES DEL SISTEMA	28
2.3. DISEÑO DE LA INTERFACE DE USUARIO	29
2.4. DISEÑO DE LA BASE DE DATOS	34
2.4.1. DIAGRAMA ENTIDAD - RELACIÓN	34
2.4.2. ESTRUCTURA DE LA BASE DE DATOS	35
3. IMPLEMENTACIÓN	36
<u>III. CONCLUSIÓN</u>	39
1. CONCLUSIÓN Y OBJETIVOS ALCANZADOS	39
2. AMPLIACIONES Y MEJORAS DEL PROYECTO	40
<u>IV. GLOSARIO</u>	41
<u>V. BIBLIOGRAFIA</u>	42
<u>ANEXO: MANUAL BASICO DE USO</u>	42

I. INTRODUCCIÓN

1.JUSTIFICACIÓN DEL TFC

El objetivo del trabajo fin de carrera es completar los estudios de Ingeniería Técnica de Informática de Gestión, para ello se plasma algunos los conocimientos adquiridos durante el estudio de dicha carrera, en función del tipo de proyecto elegido. En mi caso me ha resultado, especialmente atractiva la posibilidad de realizar este proyecto utilizando la plataforma .NET de Microsoft, como entorno orientado a objetos, y más concretamente, explorar el potencial que presta esta plataforma para crear entornos Webs con ASP.NET, así como acceso a datos a través de ADO.NET.

Puesto que el proyecto debe ser realizado de forma individual y en un periodo muy limitado de tiempo se han simplificado notablemente las funcionalidades de la aplicación, pudiendo esta ser mucho más completa, siendo desarrollada por el equipo de trabajo necesario para realizar un trabajo de estas características (director de proyecto, analistas, diseñadores, desarrolladores).

El tema del TFC escogido se basa en el desarrollo de una aplicación para gestionar listas jerárquicas, que permita anotar cualquier tipo de elemento que se desee, por ejemplo: lista de tareas pendientes, lista de la compra, contactos, eventos...

Dada la infinitud de utilidades que se le puede dar a este tipo de aplicación se ha pensado en desarrollar para que funcione bajo varios entornos.

Tendremos la parte de "gestión de listas" en un aplicación de escritorio, para que el usuario cree y actualice sus propias listas desde su PC. Y otra aplicación en entorno Web para que el usuario pueda consultar dichas listas desde cualquier ordenador con conexión a internet, previa identificación.

Pudiéndose, en una fase posterior, implementar para que funcione en versiones validas para Pocket PC.

2.OBJETIVOS DEL TFC

El objetivo principal del TFC ha sido el análisis, diseño e implementación de una solución completa, con posibilidades de ampliaciones futuras y totalmente funcional, utilizando la plataforma de Microsoft .NET.

Un objetivo no menos importante es el aprender a utilizar el mayor número posible de tecnologías ofrecidas en el mercado actual. En nuestro caso hemos utilizado tecnologías como WPF, Ajax, Ado.net, Asp.net... que en los siguientes capítulos se explicará de forma más detallada.

3.ENFOQUE Y MÉTODO SEGUIDO

El enfoque y el método seguido para realizar este proyecto, se corresponde con el ciclo de vida clásico de una aplicación. Se trata de un sistema en cascada que, en gran parte ha venido marcado por el ritmo oficial de curso.

Aunque se trata de un sistema en cascada, existe la posibilidad de retroceder en algún aspecto de alguna fase anterior. Esto se ha producido en varias ocasiones a lo largo de la realización de este proyecto, especialmente en la implementación. Se han encontrado detalles planteados en el diseño que se han tenido que modificar para obtener los resultados esperados.

4. TECNOLOGÍA A UTILIZAR

La .NET es un proyecto de Microsoft para crear una nueva plataforma de desarrollo. Basado en esta plataforma, Microsoft intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el Sistema Operativo hasta las herramientas de mercado. Provee los cimientos para la nueva generación de software. Les da a los desarrolladores las herramientas y tecnologías para hacer rápidamente soluciones de negocios que involucran distintas aplicaciones, dispositivos físicos y organizaciones.

Microsoft Visual Studio 2005, con Net Framework 2.0 y la versión 3.0 para utilizar las novedosas utilidades que nos presta Windows Presentation Foundation, incluido en dicha versión del Framework.

Windows Presentation Foundation – WPF es el nuevo modelo de programación de interfaces de usuario basadas en XML, lo que da lugar al XAML.

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.

Se utilizará la tecnología **ADO.NET** para acceso a datos.

ADO.NET ofrece varias ventajas sobre las anteriores versiones de ADO y sobre otros componentes de acceso a datos. Estas ventajas se incluyen en las siguientes categorías:

- ✦ Interoperabilidad: pueden aprovechar la flexibilidad y la amplia aceptación de XML.
- ✦ Mantenibilidad: Si la aplicación original se implementó en ADO.NET mediante conjuntos de datos (DataSets tipados), la posibilidad de una futura transformación resulta más sencilla. Dado que los niveles o capas pueden

transmitir datos por medio de conjuntos de datos con formato XML, la comunicación es relativamente fácil.

- ⊕ Programabilidad: los componentes de datos ADO.NET encapsulan funcionalidad de acceso a datos de diversas formas que ayudan a programar de modo más rápido y con menos errores.
- ⊕ Rendimiento Para las aplicaciones desconectadas, los conjuntos de datos ADO.NET ofrecen ventajas de rendimiento frente a los conjuntos de registros ADO desconectados.
- ⊕ Escalabilidad Para facilitar la escalabilidad, ADO.NET anima a los programadores a ahorrar recursos limitados. Las aplicaciones ADO.NET pueden utilizar un acceso desconectado a los datos, por lo que no retienen bloqueos ni conexiones activas con bases de datos durante largos periodos de tiempo.

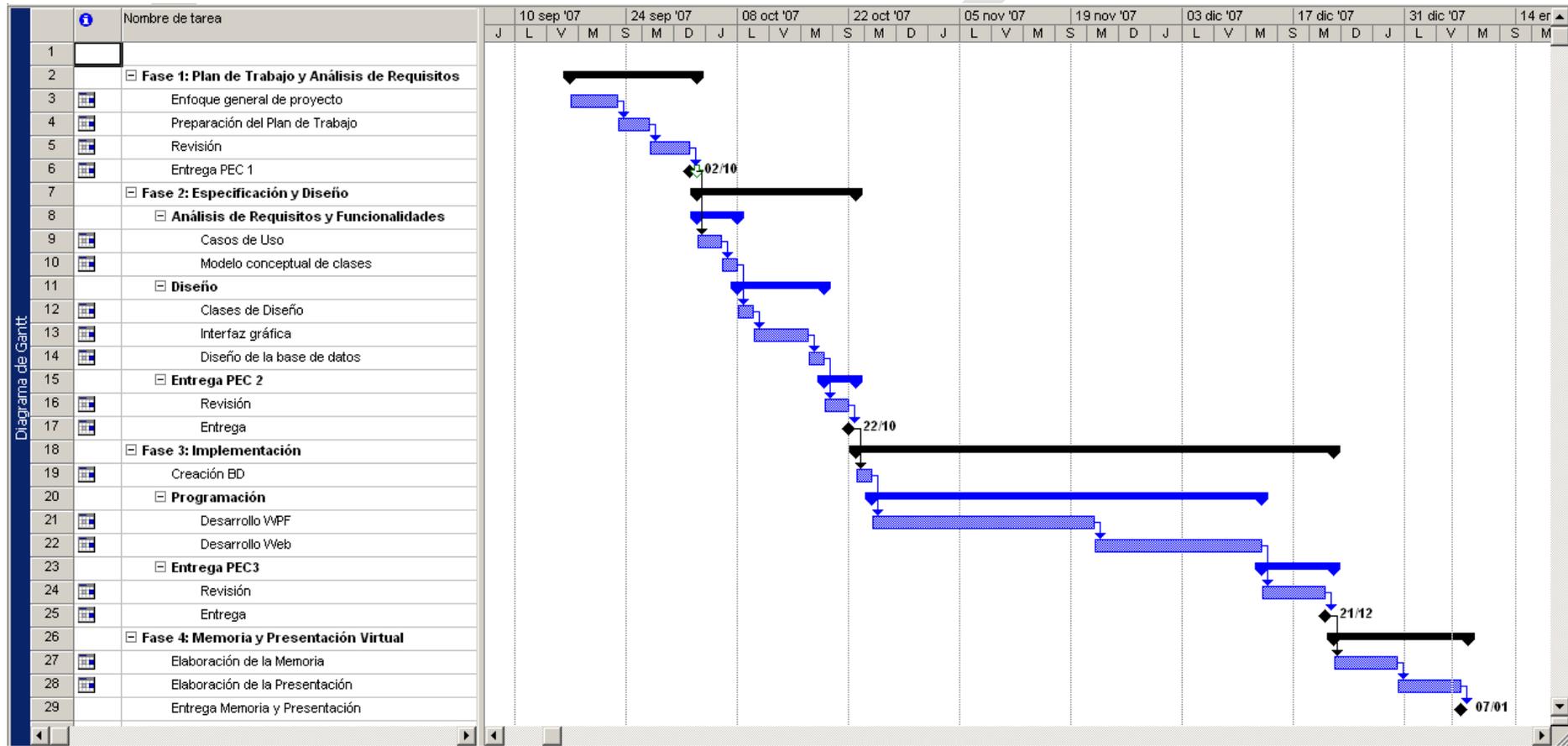
ASP.NET es un conjunto de tecnologías de desarrollo de aplicaciones web comercializado por Microsoft. Es usado por programadores para construir sitios web, aplicaciones web y servicios Web XML. Forma parte de la plataforma .NET de Microsoft y es la tecnología sucesora de la tecnología Active Server Pages (ASP).

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications), que forma parte del .NET Framework 3.5. Éstas se ejecutan en el cliente, es decir, en el navegador de los usuarios 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 completamente, sino que solamente se recarga la parte de la página que sea necesaria. Esto significa aumentar la interactividad, velocidad y usabilidad

5. PLANIFICACIÓN DEL PROYECTO

La planificación temporal de tareas del proyecto, contempla todos los trabajos a realizar en el TFC distribuidos temporalmente durante el presente cuatrimestre.

		Nombre de tarea	Duración	Comienzo
1				
2		<input type="checkbox"/> Fase 1: Plan de Trabajo y Análisis de Requisitos	16 días	lun 17/09/07
3		Enfoque general de proyecto	6 días	lun 17/09/07
4		Preparación del Plan de Trabajo	4 días	dom 23/09/07
5		Revisión	5 días	jue 27/09/07
6		Entrega PEC 1	1 día	mar 02/10/07
7		<input type="checkbox"/> Fase 2: Especificación y Diseño	20 días	mié 03/10/07
8		<input type="checkbox"/> Análisis de Requisitos y Funcionalidades	5 días	mié 03/10/07
9		Casos de Uso	3 días	mié 03/10/07
10		Modelo conceptual de clases	2 días	sáb 06/10/07
11		<input type="checkbox"/> Diseño	11 días	lun 08/10/07
12		Clases de Diseño	2 días	lun 08/10/07
13		Interfaz gráfica	7 días	mié 10/10/07
14		Diseño de la base de datos	2 días	mié 17/10/07
15		<input type="checkbox"/> Entrega PEC 2	4 días	vie 19/10/07
16		Revisión	3 días	vie 19/10/07
17		Entrega	1 día	lun 22/10/07
18		<input type="checkbox"/> Fase 3: Implementación	60 días	mar 23/10/07
19		Creación BD	2 días	mar 23/10/07
20		<input type="checkbox"/> Programación	49 días	jue 25/10/07
21		Desarrollo WPF	28 días	jue 25/10/07
22		Desarrollo Web	21 días	jue 22/11/07
23		<input type="checkbox"/> Entrega PEC 3	9 días	jue 13/12/07
24		Revisión	8 días	jue 13/12/07
25		Entrega	1 día	vie 21/12/07
26		<input type="checkbox"/> Fase 4: Memoria y Presentación Virtual	17 días	sáb 22/12/07
27		Elaboración de la Memoria	8 días	sáb 22/12/07
28		Elaboración de la Presentación	8 días	dom 30/12/07
29		Entrega Memoria y Presentación	1 día	lun 07/01/08



6.PRODUCTOS OBTENIDOS

Se han obtenido los siguientes productos:

- ✚ Análisis de requisitos y funcionalidades
- ✚ Diseño de la Aplicación
- ✚ Desarrollo de Aplicación de escritorio WPF
- ✚ Desarrollo de Aplicación Web Ajax
- ✚ Manual de usuario de la aplicación
- ✚ Memoria del proyecto
- ✚ Presentación virtual del proyecto

7.BREVE INTRODUCCIÓN DE OTROS CAPÍTULOS

En el resto de capítulos se detallan las diferentes etapas que se han seguido en la elaboración de este proyecto.

Los siguientes capítulos son:

Análisis: se explicarán los requerimientos funcionales de los diferentes módulos de la aplicación, mediante diagramas de casos de uso y otros diagramas siguiendo los modelos que nos presenta el UML.

Diseño: se explicará la arquitectura de software y de hardware necesarias para poder implantar la aplicación con éxito. También se mostrarán los diagramas de clases de diseño, diagramas de persistencia y el diagrama físico del diseño de la base de datos. Así como las interfaces de usuario (captura de pantallas) con el fin de poder observar las distintas funcionalidades de la aplicación de forma más visual.

II. CUERPO DE LA MEMORIA

1. ANÁLISIS DEL PROYECTO

1.1. DESCRIPCIÓN

“MisListas” se trata de una aplicación que permite la gestión de listas de todo tipo. La aplicación permite crear nuevas listas, añadiendo todos los elementos que se deseen bajo una relación de igualdad o subordinación respecto al elemento precedente.

La aplicación consta de un programa de escritorio en el que se realizan altas, bajas y modificaciones de dichas listas, así como de una parte web desde la cual previa identificación, los usuarios pueden acceder a las listas y consultarlas

1.2. ANÁLISIS DE REQUISITOS Y FUNCIONALIDADES

ANALISIS DE REQUISITOS:

La aplicación permite crear listas jerárquicas de cualquier tipo y de forma ilimitada. Cada lista puede estar compuesta por elementos que se refieran a distintos conceptos. El sistema permitiría crear una sucesión de elementos en una jerarquía, como puede verse en el ejemplo siguiente:

- Elemento 1
 - Elemento 2
 - Elemento 3
 - Elemento 4
 - Elemento 5
 - ✓ Elemento 6
 - ⊕ Elemento 7
 - ⊕ Elemento 8
 - ⊕ Elemento 9
 - ✓ Elemento 10
- Elemento 11
 - Elemento 12

Donde cada elemento puede corresponder a cualquier concepto distinto. Es decir: elemento 3 = actividad, elemento 4= pintura; elemento 5 = cumpleaños, etc.

El tipo de contenido que se encuentra dentro de cada elemento es indiferente. Dicho elemento es un "String" en todos los casos, al cual se le va a poder añadir los datos necesarios para poder dotar de más información a cada elemento, por ejemplo descripción, fecha de realización, categoría, prioridad, palabras clave, un apartado de nota por si el usuario quisiera añadir alguna observación relacionada con una determinada unidad de contenido e incluso alguna imagen asociada, etc.

Funcionalidades principales de la Aplicación de escritorio Windows (WPF)

- ✦ Alta de unidad de contenido (elemento de la lista)
- ✦ Edición de una unidad de contenido existente
- ✦ Eliminación de una unidad de contenido y todas las unidades que dependen de éste.

Funcionalidades principales de la Aplicación Web (ASP.NET Y AJAX)

- ✦ Consulta de las listas existentes
- ✦ Búsqueda por distintos criterios
- ✦ Filtro por categorías
- ✦ Ordenación por distintos criterios (fecha, prioridad...)

La aplicación que puede ser instalada en los puestos de una red con un servidor de datos y conexión a Internet. Para acceder a la aplicación web, debemos entrar en el navegador en www.mislistas.net, y previa identificación accedemos a nuestras listas.

1. **MÓDULO ESCRITORIO**

Esta parte de la aplicación será la encargada de realizar el mantenimiento de los ficheros de datos.

Las funcionalidades de este módulo son:

1. Añadir elementos:

A lo largo del proyecto utilizamos “elementos” o “ítems” para referirnos a un objeto que bien puede ser el nombre de una lista, o tratarse de un elemento de la misma.

Al añadir un elemento especificaremos si tiene otro elemento superior. Si no tiene ningún elemento superior se tratará de una nueva lista. Si el elemento tiene otro superior, en ese caso se tratará de un elemento de dicha lista, que a su vez puede tener otros elementos de igual nivel o inferior.

2. Añadir datos asociados

Cada elemento tiene una serie de datos asociados, datos que nos permiten enriquecer la información de dicho elemento. Estos datos también nos permitirán habilitar criterios de ordenación, filtrado y búsqueda

3. Editar elementos:

Permite editar un elemento seleccionado previamente.

4. Ver/Editar datos asociados:

Permite ver y/o editar los datos asociados a un elemento concreto previamente seleccionado.

5. Eliminar elemento:

Opción que permite dar de baja un elemento. Con objeto de mantener la integridad del sistema, se realizará la eliminación de todos los elementos dependientes (subordinados) así como los de los datos asociados.

6. Copiar elemento:

Opción que permite copiar el elemento asociado con todos sus datos asociados, así como todos sus elementos subordinados.

2. MÓDULO WEB

La aplicación se encontrará accesible a través de Internet para que el usuario/s de la aplicación escritorio pueda acceder a las listas creadas.

Las funcionalidades de este módulo son:

1. **Identificación de acceso:**

Esta opción permite que el usuario se identifique y pueda acceder al sistema previa validación.

2. **Consulta Listas:**

Permite visualizar las listas existentes, así como los datos asociados a la misma.

3. **Búsqueda:**

Permite buscar el concepto introducido como criterio de búsqueda entre los elementos existentes.

4. **Filtro por criterio:**

Permite filtrar la información existente de acuerdo con el criterio introducido.

5. **Ordenar por criterio:**

Ordena los elementos de la lista de acuerdo con el criterio de ordenación seleccionado.

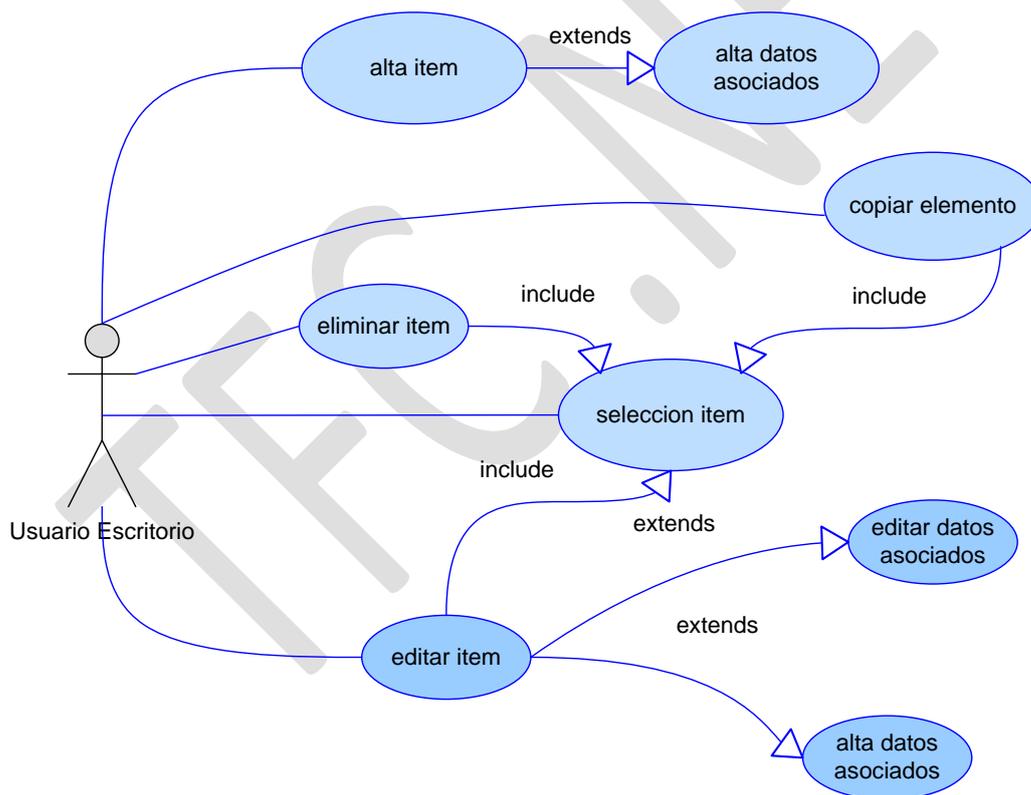
1.3. DIAGRAMA DE CASOS DE USO

A continuación describimos los casos de uso relacionados con las funcionalidades anteriormente descritas. Se distingue un único actor para cada uno de los módulos.

- ✦ Usuario local
- ✦ Usuario web

Cada funcionalidad descrita se corresponde prácticamente con cada caso de uso.

MODULO ESCRITORIO



Caso de uso: Selección Item	
Objetivo	Permite seleccionar un item existente, para la realización de una acción posterior.
Actores	Usuario de Escritorio
Precondiciones	El elemento a seleccionar existe.
Casos de uso relacionados	Incluye "Eliminar Item", "Editar Item" y "Copiar Elemento"
Post Condiciones	Se ha seleccionado un elemento concreto
Descripción	<ol style="list-style-type: none"> 1. Seleccionar el elemento 2. Aceptar

Caso de uso: Alta Item	
Objetivo	Nos permite añadir un nuevo elemento, que bien se puede añadir a una lista existente, o ser el primero de una nueva lista.
Actores	Usuario de Escritorio
Precondiciones	
Casos de uso relacionados	Extiende a Alta Datos asociados
Post Condiciones	Se ha creado un nuevo elemento en una nueva lista o en una ya existente.
Descripción	<ol style="list-style-type: none"> 1. Seleccionar crear un nuevo dato, <ol style="list-style-type: none"> a. Add Child si se va a crear un elemento dependiente del item seleccionado. b. Add Brother si se desea crear un elemento del mismo nivel del elemento seleccionado.

	<p>2. Completar los campos relacionados:</p> <ul style="list-style-type: none"> ✚ Descripción del elemento ✚ Referencia a su elemento superior en caso de tenerlo. ✚ Categoría a la que pertenece el elemento. <p>3. Guardar</p>
--	---

Caso de uso: Alta Datos Asociados	
Objetivo	Enriquecer con información adicional un elemento concreto de una lista, un elemento nuevo o uno existente.
Actores	Usuario de Escritorio
Precondiciones	Se ha seleccionado un elemento al que se le añaden nuevos datos.
Casos de uso relacionados	<p>Extiende de "Alta Item"</p> <p>Extiende de "Editar Item"</p>
Post Condiciones	Se han agregado datos relacionados a un elemento.
Descripción	<ol style="list-style-type: none"> 1. Seleccionar el ítem. 2. Completar aquellos datos que nos interese. 3. Guardar

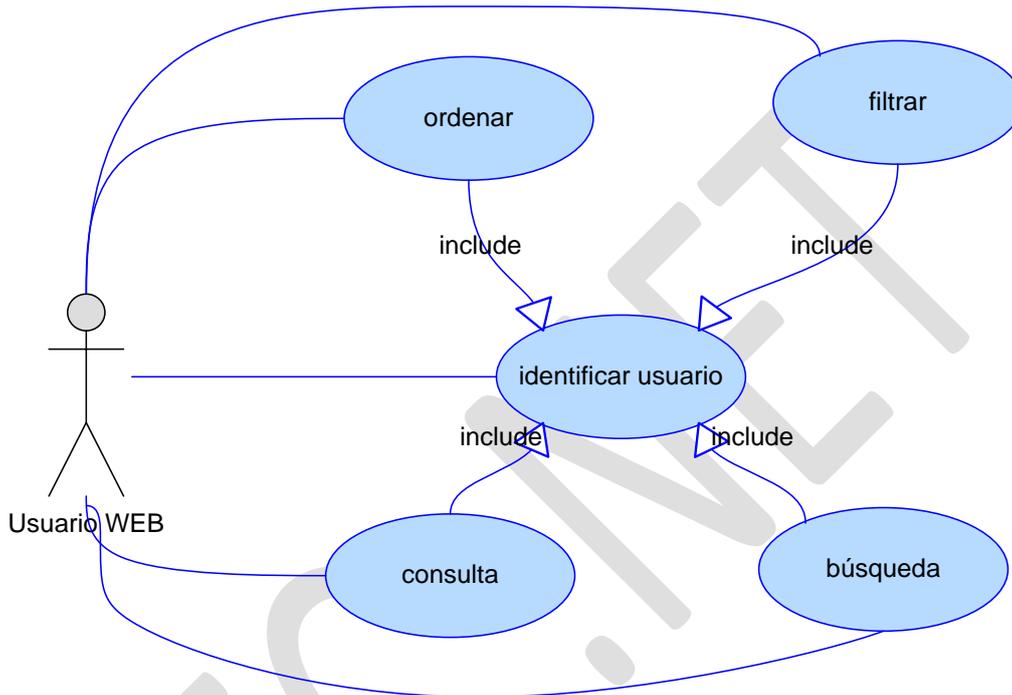
Caso de uso: Editar Item	
Objetivo	Nos permite editar un elemento existente en relación a la información de su elemento superior, de la descripción del mismo o de la categoría a la que pertenece.
Actores	Usuario de Escritorio
Precondiciones	Se ha seleccionado el elemento a modificar.
Casos de uso relacionados	Extiende a "Editar Datos Asociados" Incluye "Selección Item"
Post Condiciones	Se han modificado los campos.
Descripción	<ol style="list-style-type: none"> 1. Seleccionar el item a editar. 2. Modificar la información de su item superior, de la descripción del mismo o de la categoría a la que pertenece. 3. Guardar

Caso de uso: Editar Datos Asociados	
Objetivo	Nos permite editar los datos asociados de un elemento existente que ya tiene unos datos asociados al mismo.
Actores	Usuario de Escritorio
Precondiciones	Se ha seleccionado un item y tiene datos asociados ya creados.
Casos de uso relacionados	Extiende de "Editar Item"
Post Condiciones	Se han modificado los datos.
Descripción	<ol style="list-style-type: none"> 1. Seleccionar el dato a editar del item previamente identificado. 2. Modificar la información de los campos. 3. Guardar

Caso de uso: Eliminar Item	
Objetivo	Nos permite eliminar un elemento, así como todos los datos asociados y elementos subordinados.
Actores	Usuario de Escritorio
Precondiciones	Se ha seleccionado un elemento existente.
Casos de uso relacionados	Incluye "Selección Item"
Post Condiciones	Se elimina el elemento seleccionado así como sus datos relacionados y todos los elementos dependientes en cascada.
Descripción	<ol style="list-style-type: none"> 1. Seleccionar el item a eliminar. 2. Aceptar eliminar

Caso de uso: Copiar Elemento	
Objetivo	Permite copiar un elemento existente, sus datos asociados.
Actores	Usuario de Escritorio
Precondiciones	Se ha seleccionado el elemento a copiar.
Casos de uso relacionados	Incluye "Selección Item"
Post Condiciones	Se ha creado una copia del elemento y datos asociados.
Descripción	<ol style="list-style-type: none"> 1. Seleccionar el elemento a copiar. 2. Aceptar copiar

MODULO WEB



Caso de uso: Identificar Usuario	
Objetivo	Permite la identificación del usuario en el sistema, lo que le dará acceso al sistema.
Actores	Usuario WEB
Precondiciones	No hay
Casos de uso relacionados	Incluye "Búsqueda", "Filtrar", "Ordenar" y "Consultar"

Post Condiciones	El sistema validará o rechazará al usuario, por tanto, dará acceso o no al sistema.
Descripción	<ol style="list-style-type: none"> 1. El sistema solicita el nombre usuario y contraseña de acceso. 2. El usuario introduce los datos solicitados 3. <ol style="list-style-type: none"> a) El sistema valida los datos y le da acceso. b) El sistema rechaza los datos y le impide el acceso.

Caso de uso: Ordenar	
Objetivo	Permite ordena los elementos de la lista de acuerdo con el criterio de ordenación seleccionado.
Actores	Usuario WEB
Precondiciones	El usuario se ha identificado y tiene acceso al sistema
Casos de uso relacionados	Incluye "Identificar usuario"
Post Condiciones	Se han ordenado los elementos de acuerdo con el criterio determinado
Descripción	<ol style="list-style-type: none"> 1. Seleccionar el criterio de ordenación: prioridad, fecha, categoría y orden alfabético. 2. Aceptar 3. Se muestra en pantalla el listado ordenado.

Caso de uso: Filtrar	
Objetivo	Permite filtrar los elementos de la lista de acuerdo con el criterio de filtrado seleccionado.
Actores	Usuario WEB
Precondiciones	El usuario se ha identificado y tiene acceso al sistema
Casos de uso relacionados	Incluye "Identificar usuario"
Post Condiciones	Se han filtrado los elementos de acuerdo con el criterio determinado
Descripción	<ol style="list-style-type: none"> 1. Seleccionar el criterio a filtrar: prioridad, fecha, y categoría. 2. Aceptar 3. Se muestra en pantalla el listado de elementos que cumplen con el filtro.

Caso de uso: Consulta	
Objetivo	Permite la consulta de los elementos en forma de lista jerárquica.
Actores	Usuario WEB
Precondiciones	El usuario se ha identificado y tiene acceso al sistema
Casos de uso relacionados	Incluye "Identificar usuario"
Post Condiciones	Se muestran los elementos en forma de lista jerárquica
Descripción	<ol style="list-style-type: none"> 1. El sistema muestra por defecto esta forma de consulta.

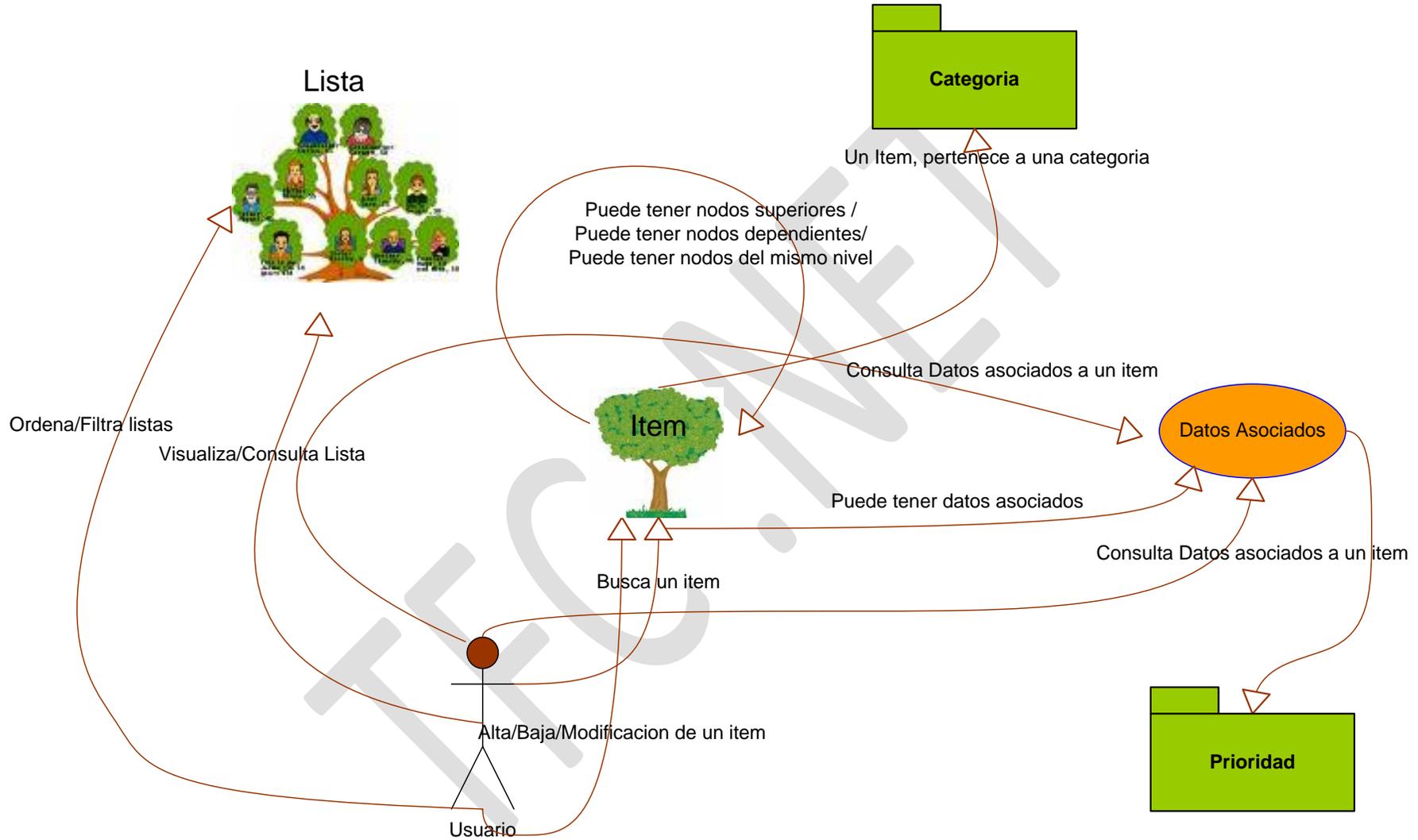
Caso de uso: Búsqueda	
Objetivo	Permite la búsqueda de un elemento concreto de acuerdo con el criterio de búsqueda descrito.
Actores	Usuario WEB
Precondiciones	El usuario se ha identificado y tiene acceso al sistema
Casos de uso relacionados	Incluye "Identificar usuario"
Post Condiciones	Se muestra el listado de los elementos que cumplen con el criterio de búsqueda introducido.
Descripción	<ol style="list-style-type: none">1. Se introduce como criterio de búsqueda una palabra que se buscará en el campo relativo a la descripción del elemento.2. Se visualiza la lista de elementos que cumplen con el criterio de búsqueda.

1.4. MODELO CONCEPTUAL DE CLASES

Los casos de uso son una importante herramienta para el análisis de requerimientos, pero realmente no están orientados a objetos. Un modelo conceptual representa cosas del mundo real, no componentes del software.

Para descomponer el dominio del problema identificamos los conceptos, los atributos y las asociaciones del dominio que se consideren importantes:

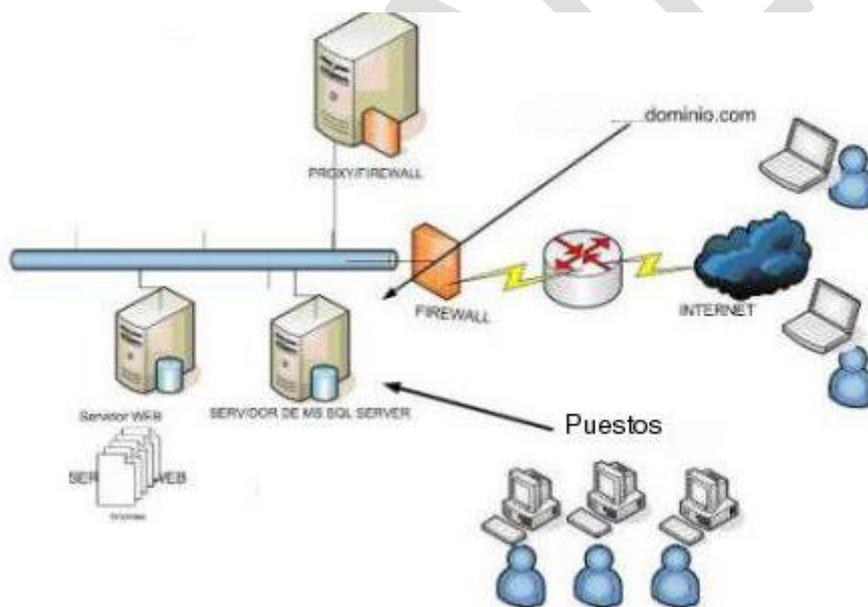
- ✚ Usuario – Alta/Baja/Modificación de un ítem en las listas
- ✚ Usuario – Visualiza o consulta listas de ítems.
- ✚ Usuario – Consulta datos asociados a un ítem
- ✚ Usuario – Alta/Baja/Modificación datos asociados a un ítem
- ✚ Usuario – Ordena listas de ítems.
- ✚ Usuario – Filtra listas de ítems por categoría
- ✚ Usuario – Busca un ítem
- ✚ Ítem – Puede tener datos asociados
- ✚ Ítem – Puede tener nodos superiores
- ✚ Ítem – Puede tener nodos dependientes
- ✚ Ítem – Puede tener nodos del mismo nivel
- ✚ Ítem – Pertenece a una categoría
- ✚ Ítem – Puede tener datos asociados
- ✚ Un Dato Asociado – es de una prioridad



2.DISEÑO

2.1. DISEÑO DE LA ARQUITECTURA DEL SISTEMA

El escenario donde se podrá implantar esta aplicación puede ser cualquiera, desde un local hasta una oficina con varias redes locales e intranets. Hemos de dotar al sistema para que este preparado para funcionar en el caso mas complejo. Requerimos por parte del servidor una máquina con SQL Server 2005 con las bases de datos y un servidor Web con ISS para el acceso a Internet y la implantación del entorno Web. Sería posible, aunque no recomendable, tener un único servidor para los tres sistemas.



En la figura anterior se puede observar que pueden existir varios ordenadores conectados a la red, que tendrán acceso a los diferentes servidores. Los usuarios pueden acceder a las aplicaciones (de escritorio o web) conectándose a los servidores.

Aplicación de escritorio

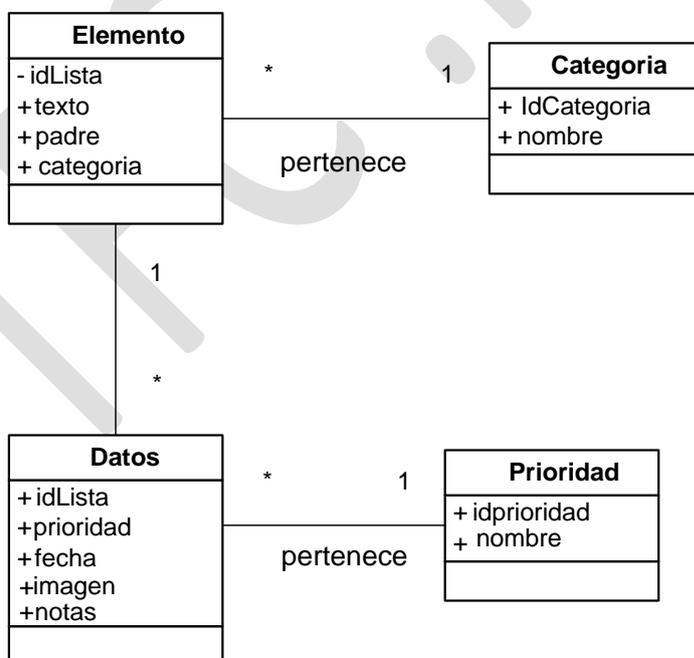
La arquitectura que utilizará la aplicación será del tipo cliente-servidor. La aplicación de escritorio se instalará en los ordenadores de la red local que se comunicarán con la base de datos instalada en el servidor de SQL Server 2005. En el servidor no será necesario instalar la aplicación.

Aplicación web

Para el caso de los ordenadores remotos que se conecten a la Web vía Internet, será necesario disponer de un Servidor Web con acceso a la base de datos y al entorno Web de la aplicación, por lo que se utilizará IIS. La aplicación ASP.NET se comunicará con el servidor SQL Server 2005 para acceder a la base de datos y generar la información necesaria en respuesta a las peticiones del cliente.

2.2. DISEÑO DE CLASES DEL SISTEMA

Del enunciado y de los casos de uso podemos deducir el siguiente diagrama de clases:



2.3. DISEÑO DE LA INTERFACE DE USUARIO

A continuación se irán detallando las pantallas de los diferentes módulos de la aplicación.

A. APLICACIÓN ESCRITORIO:

Pantalla principal: Esta pantalla se carga por defecto al abrir la aplicación.



En esta pantalla se visualizan las listas jerárquicas guardadas por el usuario.

En la parte superior se encuentra barra de menú con las siguientes opciones:

+ Crear nuevo elemento: Nos permite crear un nuevo elemento.

- ✚ **Crear elemento dependiente:** Crea un elemento dependiente del elemento seleccionado.
- ✚ **Crear elemento del mismo nivel:** Crea un elemento del mismo nivel del elemento seleccionado.
- ✚ **Ver datos:** Permite visualizar la lista general de elementos.
- ✚ **Editar datos:** Permite modificar el elemento seleccionado.
- ✚ **Copiar elemento:** Permite copiar el elemento seleccionado.
- ✚ **Eliminar elemento:** Permite eliminar el elemento seleccionado y todos sus elementos dependientes.

Pantalla Editar/Ver /Nuevo Item: Esta pantalla permite crear un nuevo elemento o editar los datos de uno previamente seleccionado.

MisListas - Versión 1.0

Descripción

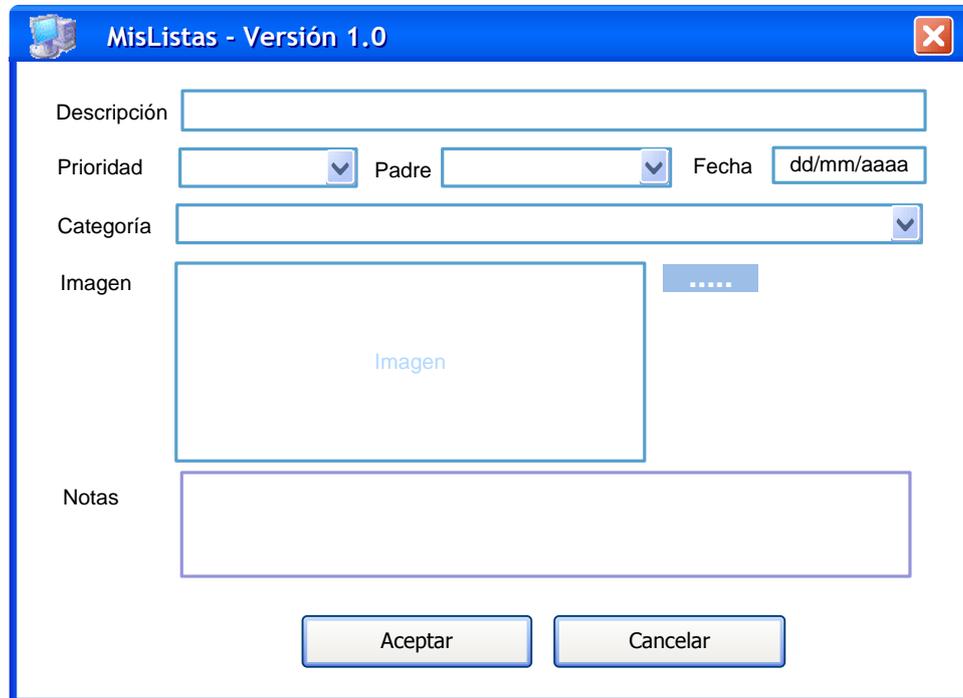
Padre

Categoría

Aceptar Cancelar

En el caso de querer enriquecer la información asociado a un Item, seleccionamos la opción "Ver datos", y nos muestra la pantalla siguiente.

Pantalla Editar/Ver/ Nuevo Datos Asociados: Esta pantalla permite enriquecer la información relativa a un Item con datos relacionados.

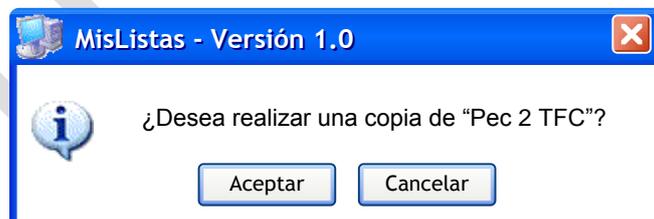


The screenshot shows a window titled "MisListas - Versión 1.0" with a close button in the top right corner. The form contains the following fields:

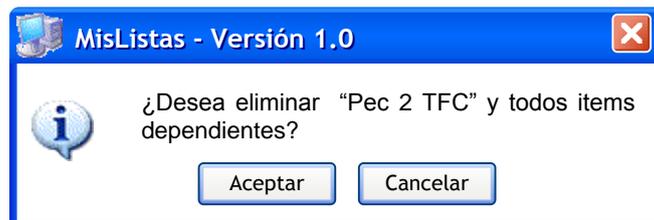
- Descripción:** A text input field.
- Prioridad:** A dropdown menu with a downward arrow.
- Padre:** A text input field.
- Fecha:** A date input field with the placeholder "dd/mm/aaaa".
- Categoría:** A dropdown menu with a downward arrow.
- Imagen:** A large rectangular area with the text "Imagen" in the center and a small blue button with six dots to its right.
- Notas:** A large text area for notes.

At the bottom of the window, there are two buttons: "Aceptar" and "Cancelar".

Mensaje de Confirmación de Copia: Este mensaje nos pide la confirmación de copia de un elemento concreto seleccionado.

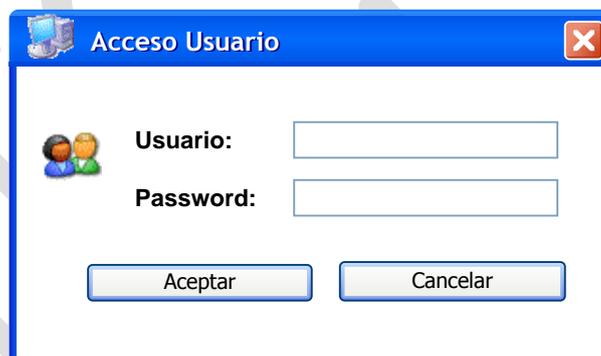


Mensaje de Confirmación de Eliminación: Este mensaje nos pide la confirmación de la eliminación de un elemento concreto seleccionado.

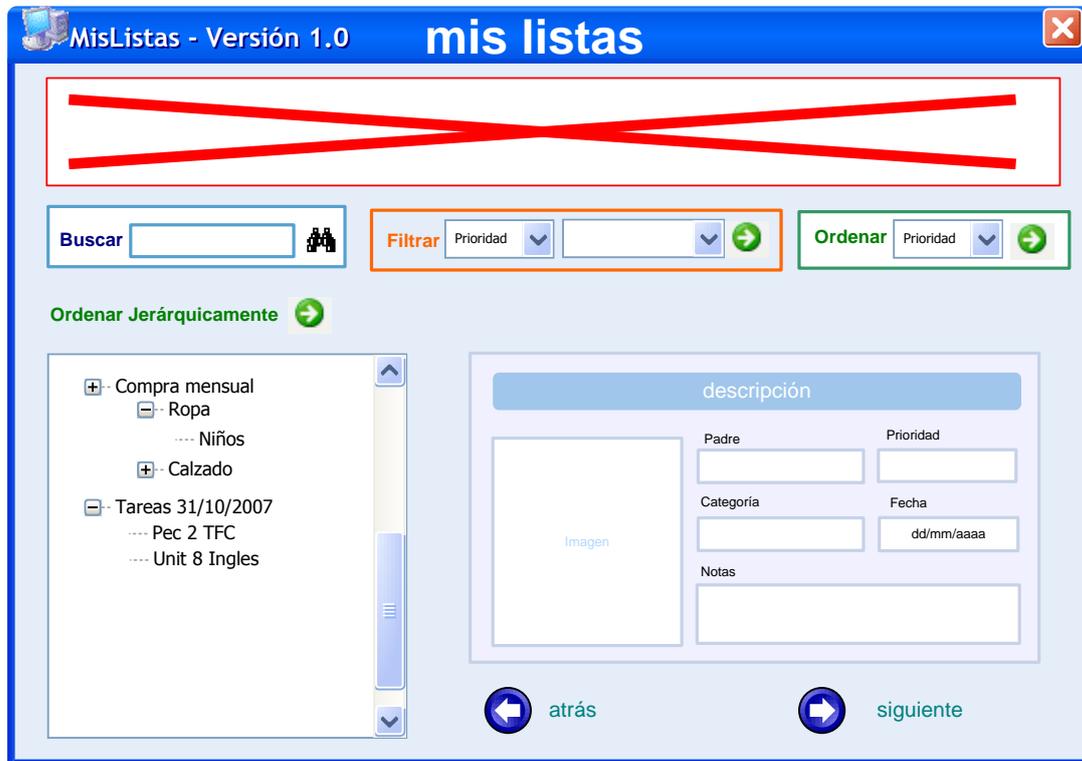


B. APLICACIÓN WEB:

Pantalla de identificación: El usuario se identifica y accede al sistema.



Pantalla Principal: Una vez identificado el usuario, se carga la página principal.



Dividimos la página en dos marcos:

En el **marco superior** encontramos una imagen/logo de la web, que servirá para cargar la página en su estado por defecto.

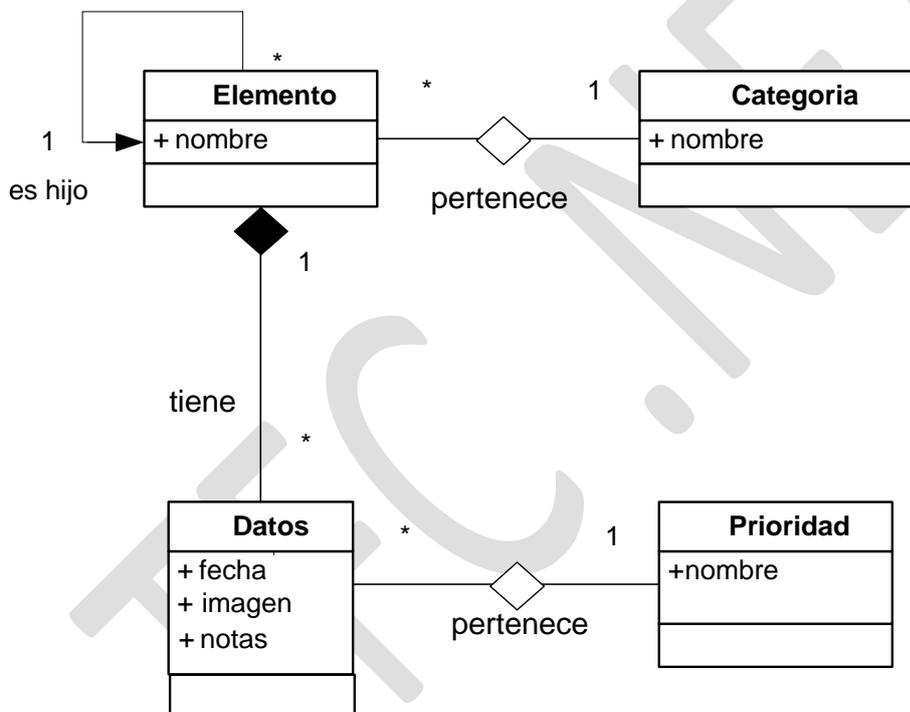
En el **marco inferior**, distinguimos tres zonas:

- En el margen superior encontramos los controles necesarios para presentar los elementos de las listas de tres formas distintas: filtrar, buscar u ordenar.
- En el margen izquierdo, se visualizan las listas jerárquicas guardadas por el usuario. Estas listas se mostrarán en función de la selección realizada de los controles del margen superior.
- En el margen derecho se muestran los detalles de un elemento concreto.

2.4. DISEÑO DE LA BASE DE DATOS

2.4.1. DIAGRAMA ENTIDAD - RELACIÓN

Para realizar el diagrama de entidad-relación hemos de decidir los conjuntos tanto de entidades como de relaciones, atendiendo al enunciado del problema, la estructura del diagrama ha quedado como sigue



2.4.2. ESTRUCTURA DE LA BASE DE DATOS

Con la transformación al modelo relacional, se obtienen las siguientes relaciones de las entidades y de las interrelaciones.

✚ **Tabla ELEMENTO**

IdLista, nombre, padre, categoría

- Primarykey: idLista
- Foreignkey: categoría (Categoría.IdCategoría)

✚ **Tabla CATEGORÍA**

IdCategoría, nombre

- Primarykey: idCategoría

✚ **Tabla DATOS**

IdLista, prioridad, fecha, imagen, notas

- Primarykey: idLista
- Foreignkey: idLista(Elemento.IdLista)
- ForeignKey: prioridad (Prioridad idPrioridad)

✚ **Tabla PRIORIDAD**

IdPrioridad, nombre

- Primarykey: idPrioridad

3.IMPLEMENTACIÓN

Creo importante comentar algunos aspectos de la implementación del proyecto, ya que es esta la fase en la que se ha invertido mayor tiempo y dedicación.

La base de datos que guarda la información de los datos necesarios para la aplicación está creada bajo SQL Server. Con objeto de facilitar las funciones que hay que realizar desde la aplicación se han generado funciones en la base de datos para que sean utilizadas desde cualquier aplicación.

Se muestra a continuación la función que se utiliza en la aplicación para obtener los elementos dependientes de otros para poder permitir desde la aplicación el borrado en cascada:

```
set ANSI_NULLS ON
set QUOTED_IDENTIFIER ON
GO
ALTER FUNCTION [dbo].[cascada] (@id INTEGER)
returns @t table ([idLista] int)
as
begin
declare @i int
set @i = @id
while exists(select * from dbo.elementos where [padre] = @i)
begin
insert into @t values(@i)
select @i = idLista
from dbo.elementos
where [padre] = @i
end
return
end
```

Para el acceso a datos desde las dos aplicaciones, se ha utilizado ADO.NET. En la mayoría de los casos se trabajó de forma desconectada utilizando siempre el mismo objeto connection.

Se muestra a continuación una función del módulo de acceso a datos de la clase datos.cs, que permite la obtención de los hijos de un nodo concreto, pasando el valor del mismo como parámetro de entrada de la función y devolviendo los hijos del mismo si los tuviera:

```

public TreeViewItem children(TreeViewItem nodoPadre)
{
    SqlDataAdapter daHijos;
    DataTable tblHijos = new DataTable();
    DataSet ds = new DataSet();
    daHijos = new SqlDataAdapter( "select idLista,
texto,padre,categoria from elementos where padre=" +
nodoPadre.Tag + " order by idLista", sqlCon);
    daHijos.Fill(ds, "Hijos");
    tblHijos = ds.Tables["Hijos"];
    foreach (DataRow drC in tblHijos.Rows)
    {
        TreeViewItem fChild = new TreeViewItem();

        Elemento e = new Elemento();
        e.setIdLista((int)drC["idLista"]);
        e.setTexto(drC["texto"].ToString());
        if (drC["padre"] != DBNull.Value)
            e.setPadre((int?)drC["padre"]);
        if (drC["categoria"] != DBNull.Value)
            e.setCategoria((int?)drC["categoria"]);
        fChild.DataContext = e;
        fChild.Header = drC["texto"].ToString();
        fChild.Tag = drC["idLista"];
        nodoPadre.Items.Add(fChild);
        children(fChild);
    }
    return nodoPadre;
}

```

Con el fin de poder demostrar algunos de los conocimientos obtenidos sobre WPF se ha añadido una pizarra editable y unas imágenes ajustables en su tamaño.

Veáse por ejemplo el siguiente código xaml que permite ajustar el tamaño de una imagen

```

<StackPanel Grid.Column="1" Grid.Row="1" HorizontalAlignment="Center">
    <TextBlock HorizontalAlignment="Left">
        Ajustar tamaño de Imagen
    </TextBlock>
    <Slider Name="imageSizeSlider" Width="150"
Orientation="Horizontal" HorizontalAlignment="Center"
Value="200" Minimum="0" Maximum="200"
TickPlacement="TopLeft" TickFrequency="10"/>
    <Image Source="/images/bolsa.jpg"
HorizontalAlignment="Left"
Margin="0,0,0,100"
Height="{Binding
ElementName=imageSizeSlider,Path=Value}"/>
</StackPanel>

```

En la aplicación Web se ha utilizado Ajax para evitar la recarga innecesaria de la página. Se muestra a continuación el updatePanel insertado con el fin de que evitar que la página se recargue al cambiar de elemento seleccionado:

```
<asp:UpdatePanel ID="UpdatePanel1" runat="server">
  <ContentTemplate>
    <asp:Panel ID="Panel2" runat="server" ScrollBars="Auto"
      Width="390px">
      <asp:TreeView ID="Listas" runat="server" Height="400px"
        Width="380px"
        ImageSet="XPFileExplorer" NodeIndent="15" Font-
          Names="@Arial Unicode MS" Font-Size="14pt" >
      <ParentNodeStyle Font-Bold="False" />
      <HoverNodeStyle Font-Underline="True" ForeColor="#6666AA" />
      <SelectedNodeStyle BackColor="#B5B5B5" Font-
        Underline="False" HorizontalPadding="0px"
        VerticalPadding="0px" />

      <NodeStyle Font-Names="Tahoma" Font-Size="12pt"
        ForeColor="Black" HorizontalPadding="2px"
        NodeSpacing="0px" VerticalPadding="2px" />

      </asp:TreeView>
    </asp:Panel>
  </ContentTemplate>
</asp:UpdatePanel>
```

Para la identificación de usuarios se ha utilizado la herramienta que ofrece Asp (proveedor memberShip).

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

La configuración de la aplicación se guarda en el fichero web.config. Podemos ver en el siguiente fragmento de código la configuración de la cadena de conexión con la base de datos:

```
<connectionStrings>

  <add name="MisListasConnectionString" connectionString="Data
    Source=.\SQLEXPRESS;Initial Catalog=MisListas;Integrated
    Security=True" providerName="System.Data.SqlClient" />

</connectionStrings>
```

III. CONCLUSIÓN

1. CONCLUSIÓN Y OBJETIVOS ALCANZADOS

La realización del proyecto “MisListas” para gestión de listas jerárquicas bajo la plataforma .NET de Microsoft ha sido en general, una experiencia profesional muy satisfactoria, a pesar de haber supuesto gran dedicación y esfuerzos.

El proyecto ha servido para afianzar y adquirir nuevos conocimientos sobre lo ya aprendido en la carrera, especialmente asignaturas como Ingeniería del software, base de datos, programación orientada a objetos y estructura de la información.

Además me ha permitido conocer algunas de las nuevas tecnologías que nos ofrece, hoy en día, el mercado, no sin grandes esfuerzos por encontrar información sobre éstas. En el mundo de las nuevas tecnologías estamos constantemente cambiando, mejorando y sustituyendo unas herramientas por otras, y es muy difícil (casi imposible) mantenerse informado de todas estas mejoras. Pero a lo largo del desarrollo de este proyecto y bajo el desconocimiento absoluto de algunas de estas tecnologías (por ejemplo WPF o Ajax) me he dado cuenta de que Internet es un gran “lugar” donde poder obtener e intercambiar información, y de esta forma ir aprendiendo. En general, estoy muy satisfecha del producto final obtenido y de los conocimientos adquiridos por medio de la realización del mismo.

Me hubiera gustado mucho haber podido realizar un proyecto libre, planteado por mí para poder darle alguna finalidad en mi trabajo, esto me hubiera motivado mucho más, pero entiendo que en una Universidad a distancia es difícil de plantear. Entiendo que los temas sean planteados por la Universidad. No obstante, en este semestre los dos escenarios planteados no me han resultado demasiado interesantes, dada mi actividad profesional, porque no los he visto muy aplicables en mi entorno laboral.

2.AMPLIACIONES Y MEJORAS DEL PROYECTO

Dada la ambigüedad del escenario propuesto, "Gestión de listas jerárquicas", quizás resulten discutibles las funcionalidades de la aplicación. Yo he desarrollado las que planteo en la etapa de análisis ya que fueron bien aceptadas en su momento, pero creo que se podría plantear de formas muy distintas.

Algunas opciones de la aplicación son mejorables o ampliables, aunque he tratado de que todas sean funcionales y ofrezcan al usuario la mayor cantidad de posibilidades de gestión de listas.

Las posibles mejoras o ampliaciones de la aplicación que considero son:

- ✚ Se podría añadir al dar de alta un elemento la posibilidad desde la misma pantalla de añadir los datos asociados.
- ✚ Añadir menús contextuales al pulsar el botón derecho del ratón sobre una lista, como alta, editar, añadir hijo, añadir hermano...
- ✚ También se podría hacer una aplicación para la gestión de estas listas para dispositivos móviles.
- ✚ Se podrían hacer listas por usuario estableciendo permisos de creación y consulta para las mismas, de esta forma cada usuario podría tener sus propias listas privadas y públicas. La aplicación de escritorio pediría, al igual que la web, la identificación del usuario y así podría ofrecer la información propia a cada uno.
- ✚ Se podría guardar de forma persistente las anotaciones escritas en la pizarra o dar alguna facilidad para que esta sea más usable.
- ✚ Las imágenes ajustables podrían estar relacionadas con la imagen del elemento seleccionado.

IV. GLOSARIO

UML: Lenguaje unificado de modelado. Es la notación utilizada en la Ingeniería del Software para documentar y construir un sistema de Software.

.NET: la plataforma .NET es una capa de software que está colocada entre las aplicaciones y el sistema operativo. Se trata de un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma y que permita un rápido desarrollo de aplicaciones.

ASP .NET: Plataforma de desarrollo orientado a aplicaciones Web, independiente del lenguaje de servidor escogido (C#, VB .net, J# y otros).

ADO.NET: solución de Microsoft para permitir el acceso a bases de datos desde la plataforma .NET.

SQL Server Express 2005: es un sistema gestor de bases de datos relacionales que actualmente Microsoft ofrece de forma gratuita. Esta basado en el lenguaje SQL.

C#: Lenguaje de programación creado por Microsoft basado en C++ y Java usado en el desarrollo de aplicaciones bajo Visual Studio.

Microsoft Visual Studio 2005: es un IDE de desarrollo de aplicaciones que permite el diseño de aplicaciones o soluciones Windows, Webs y móviles.

AJAX: Asynchronous JavaScript And XML, es una técnica de desarrollo web para crear aplicaciones interactivas.

WPF: Windows Presentation Foundation, es una tecnología para la creación de interfaces que ofrece demostrables beneficios a los usuarios que las utilizan. Además 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 las aplicaciones. Diseñadores y desarrolladores intercambian información sobre el diseño de las aplicaciones de forma mucho más inteligente.

V. BIBLIOGRAFIA

Enlaces utilizados para la investigación y el aprendizaje de las distintas herramientas:

He encontrado muy útil las webs que ofrece Microsoft para desarrolladores:

<http://msdn.microsoft.com/library/spa/default.asp?>

<http://msdn2.microsoft.com/es-es/default.aspx>

Curso de acceso a datos:

[http://msdn2.microsoft.com/es-es/library/ms370113\(VS.80\).aspx](http://msdn2.microsoft.com/es-es/library/ms370113(VS.80).aspx)

<http://msdn.microsoft.com/sql/express/default.aspx>

Y sus foros:

<https://forums.microsoft.com/MSDN-ES/User/MyForums.aspx?SiteID=11>

Ajax:

<http://www.asp.net/>

<http://www.codeplex.com/>

W.P.F.:

<http://netfx3.com/>

y sus laboratorios: <http://netfx3.com/labs/>

ejemplos: <http://wpf.netfx3.com/files/25/default.aspx>

Desarrollo en general con NET:

<http://www.elguille.net/>

<http://www.programar.net/>

<http://www.asp.net/> , <http://ajax.asp.net/>

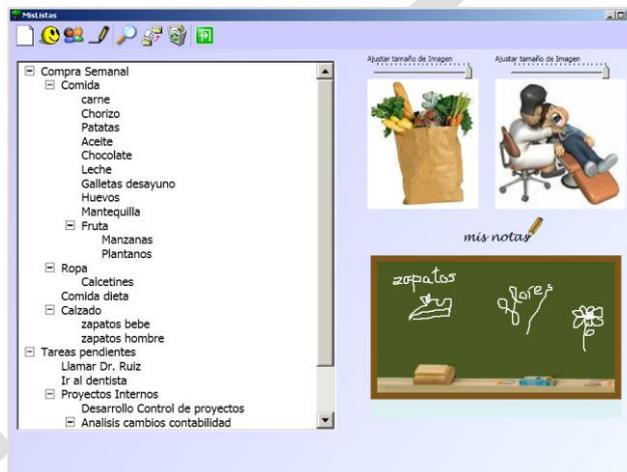
<http://www.lawebdelprogramador.com/>

**ANEXO:
MANUAL BÁSICO DE USO DE LA
SOLUCIÓN "MIS LISTAS"**

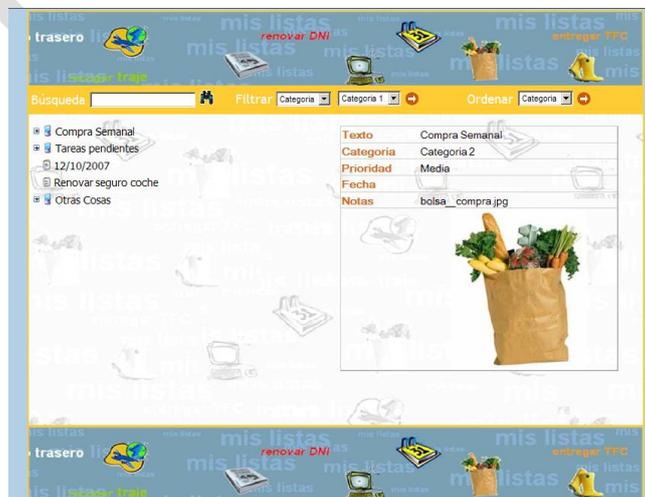
Descripción de la Aplicación:

La solución "Mis Listas" consta de dos aplicaciones, aplicación escritorio y la aplicación web. La solución nos permite crear listas jerárquicas mediante la utilización de la aplicación escritorio, y la de visualización mediante la aplicación escritorio o web.

Aplicación Escritorio



Aplicación Web



En los capítulos siguientes se explica en detalle las funcionalidades de cada una de ellas de forma gráfica y sencilla.

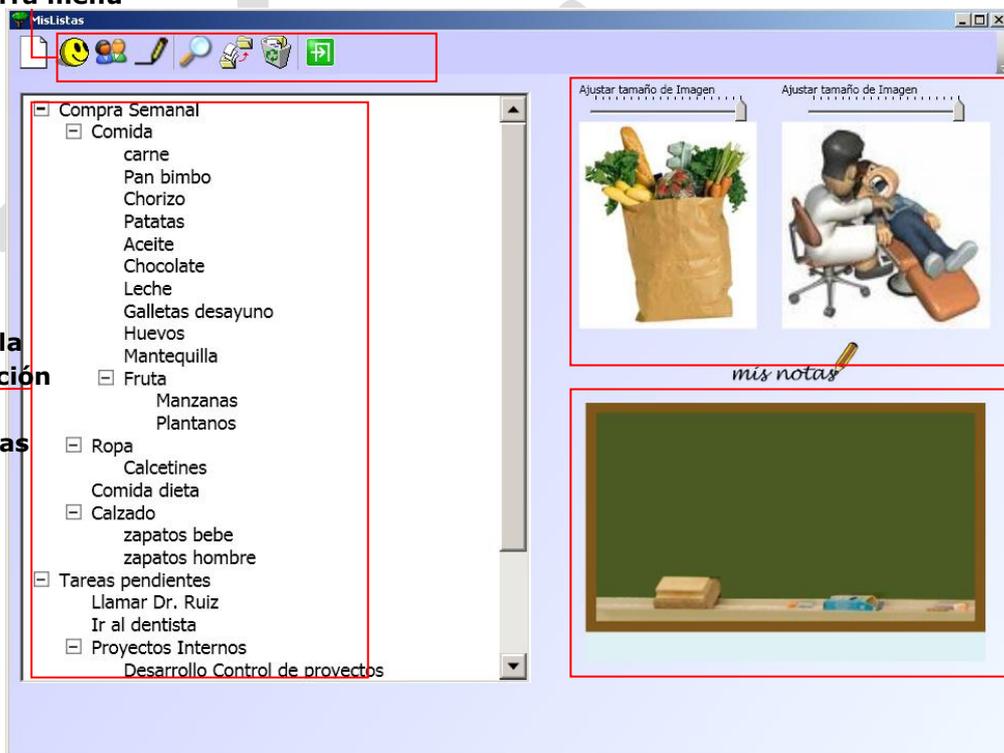
1. APLICACIÓN ESCRITORIO

Esta aplicación nos permite ir creando las listas jerárquicas que queramos. Una lista jerárquica significa ir creando un árbol de nodos donde ir agregando elementos de la misma categoría, inferior o superior a un nodo anterior, la aplicación nos permite además agregar información adicional: categoría, prioridad, adjuntar imágenes, etc..

Los **elementos fundamentales de la pantalla principal** son:

A. Barra menú

B. Pantalla visualización de listas jerárquicas

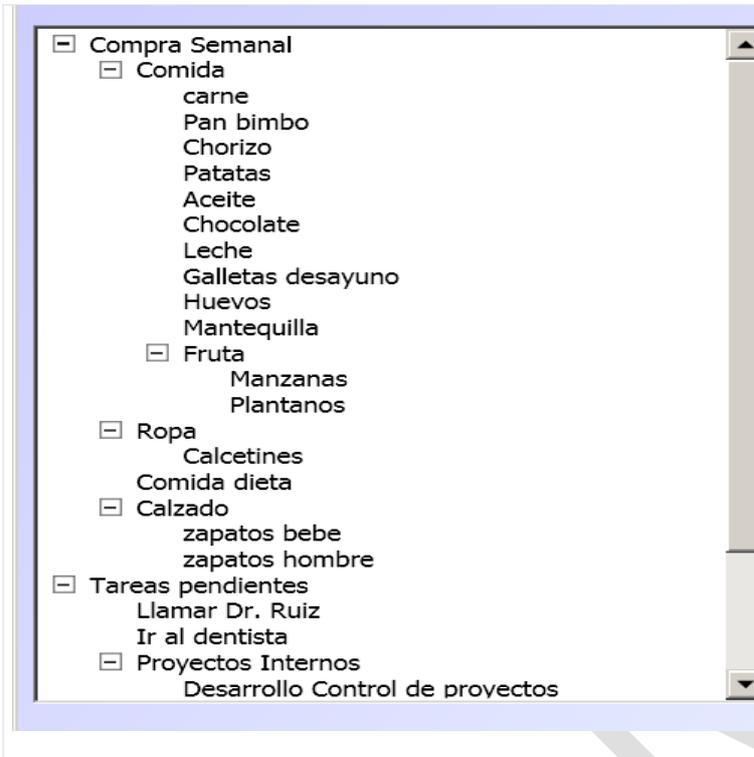


D. Zona animación

C. Pizarra temporal de anotación

<h2 style="text-align: center;">A. Barra Menú</h2>	
	<p>Add Item: Permite añadir un elemento nuevo</p>
	<p>Add Child: Permite añadir un elemento nuevo hijo del nodo seleccionado en el momento de pinchar el icono.</p>
	<p>Add Brother: Permite añadir un elemento nuevo del mismo nivel, y del mismo padre del nodo seleccionado en el momento de pinchar el icono.</p>
	<p>Editar Item: Permite editar los campos del Item seleccionado: nombre, categoría y nodo padre.</p>
	<p>Ver Datos: Permite visualizar los datos asociados al Item seleccionado: nombre, categoría, nodo padre, fecha, prioridad, notas y añadir/editar una imagen desde archivo.</p>
	<p>Copiar Item: Realiza una copia exacta del Item seleccionado. Este Item podrá ser editado posteriormente.</p>
	<p>Borrar: Opción que nos permite borrar el Item seleccionado y sus datos asociados en cascada.</p>
	<p>Salir: Opción que nos permite salir de la aplicación.</p>

B. Pantalla Visualización de Listas Jerárquicas



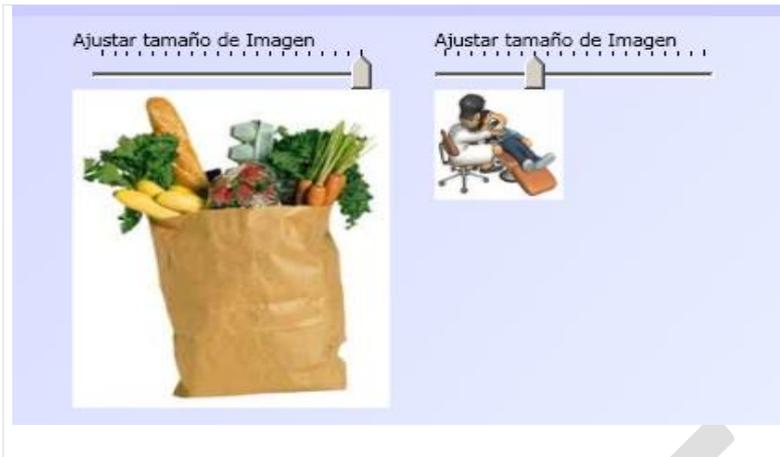
Se puede observar en la imagen cómo se han creado listas jerárquicas, donde los elementos se relacionan entre sí de acuerdo con una relación de superioridad, igualdad o inferioridad en su jerarquía.

C. Pizarra temporal de anotación



Se puede observar en la imagen la pizarra sobre la cual se puede escribir notas. Esta pizarra no se guarda, es sólo de uso temporal mientras se ejecuta la aplicación.

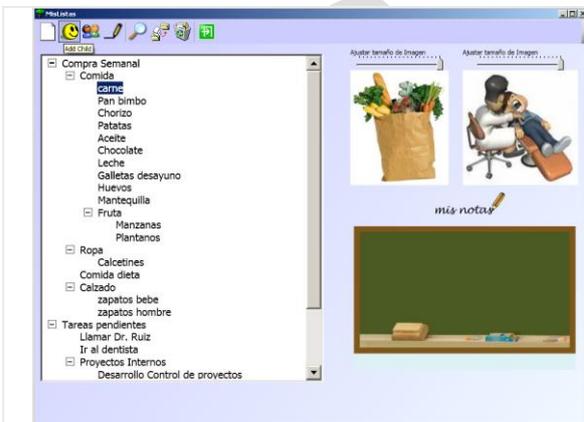
D. Zona de Animación



Se puede observar en la imagen la “zona de animación”, se trata únicamente de una zona decorativa donde se pueden agrandar y reducir las imágenes tal y como se observa.

La aplicación “paso a paso”:

1. Add Item/ Add Child / Add Brother:



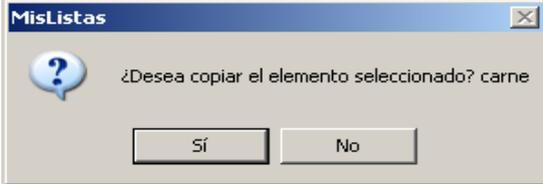
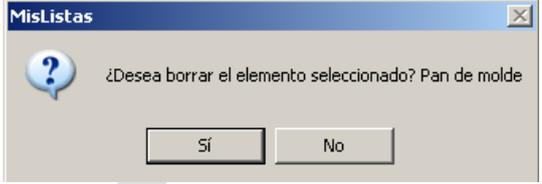
Nos situamos en un item concreto y seleccionamos algunas de las tres opciones “Add item”, “Add child” o “Add brother”.

El sistema nos muestra una pantalla emergente donde completar los datos relativos al “nombre” del nuevo item y su “categoría”. El nodo padre viene definido en función de la selección realizada previamente, pero permite su modificación.

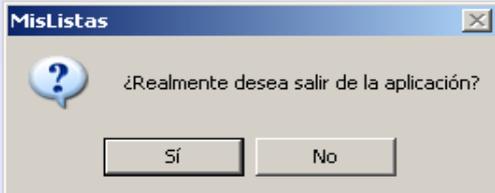
2. Editar Item/ Ver Datos Asociados

	
<p>Nos situamos en un item concreto y seleccionamos la opción “Editar item”, el sistema nos muestra sus datos básicos actuales y nos permite actualizarlos.</p>	<p>Nos situamos en un item concreto y seleccionamos la opción “Ver Datos”, el sistema nos muestra sus datos asociados y nos permite crearlos o actualizarlos. Estos datos son además de los básicos: La prioridad, la fecha y una imagen relacionada: al pinchar sobre ... junto a la imagen podemos acceder a c:/ y adjuntar la imagen que deseemos desde archivo.</p>

3. Copiar Item/ Borrar Item:

	
<p>Nos situamos en un item concreto y seleccionamos la opción “Copiar item”, el sistema nos duplica el dato. Esta funcionalidad es interesante para crear datos del mismo nivel en los que tan sólo se quiera modificar algún atributo.</p>	<p>Nos situamos en un item concreto y seleccionamos la opción “Borrar item”, el sistema borra el dato y todos sus datos asociados</p>

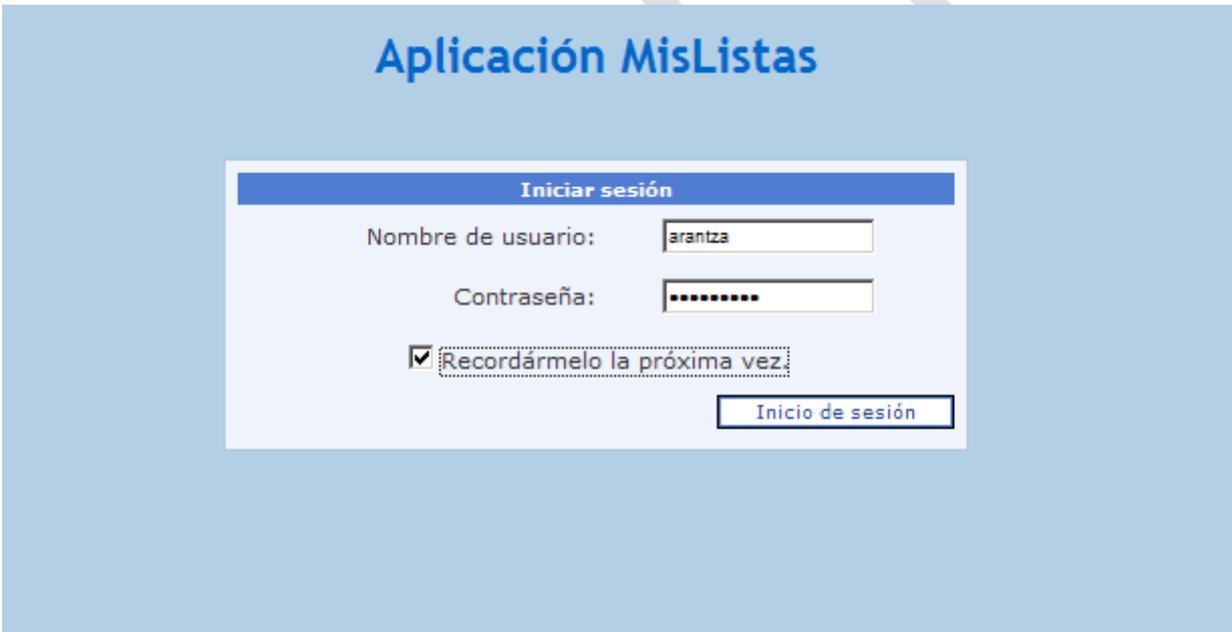
4. Salir de la Aplicación

	<p>Al seleccionamos la opción “Salir”, el sistema cierra la aplicación.</p>
---	--

2. APLICACIÓN WEB

La aplicación web, nos permite visualizar la información relativa a mis listas jerárquicas vía web, previa validación de "usuario" y "password".

pantalla de acceso:



Aplicación MisListas

Iniciar sesión

Nombre de usuario:

Contraseña:

Recordármelo la próxima vez

Existe un usuario ya creado:

- Nombre de usuario: arantza
- Contraseña": aran_aran

Los **elementos fundamentales de la pantalla principal** son:

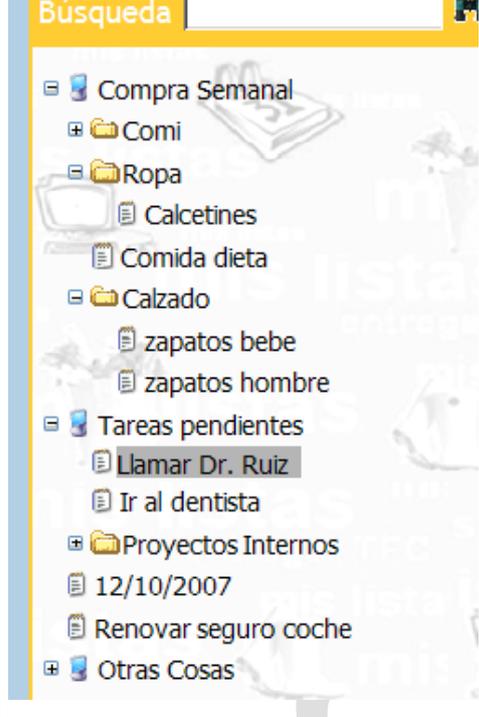
B. Filtros

A. Pantalla visualización de listas jerárquicas

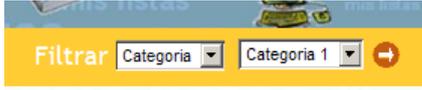
C. Información del item seleccionado

Texto	Compra Semanal
Categoría	Categoría 2
Prioridad	Media
Fecha	
Notas	bolsa_compra.jpg

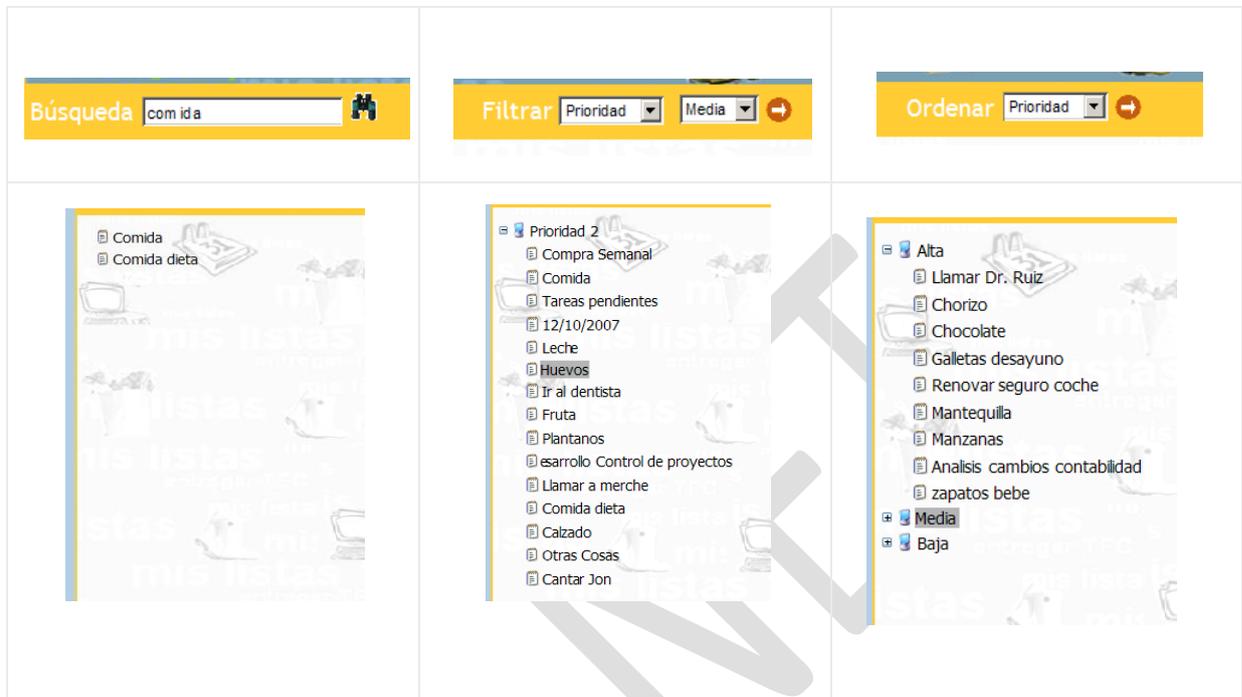
A. Pantalla Visualización de Listas Jerárquicas

 <p>Búsqueda</p> <ul style="list-style-type: none">[-] Compra Semanal<ul style="list-style-type: none">[+] Comi[-] Ropa<ul style="list-style-type: none">[-] Calcetines[-] Comida dieta[-] Calzado<ul style="list-style-type: none">[-] zapatos bebe[-] zapatos hombre[-] Tareas pendientes<ul style="list-style-type: none">[+] Llamar Dr. Ruiz[-] Ir al dentista[-] Proyectos Internos<ul style="list-style-type: none">[-] 12/10/2007[-] Renovar seguro coche[-] Otras Cosas	<p>Se puede observar en la imagen cómo se visualizan las listas jerárquicas existentes, donde los elementos se relacionan entre sí de acuerdo con una relación de superioridad, igualdad o inferioridad en su jerarquía.</p> <p>Esta Pantalla modificará su estructura en base a los filtros que se vayan seleccionando.</p>
--	--

B. Filtros. Criterios de ordenación y búsqueda.

	
	<p>Búsqueda: Al introducir un término el sistema busca este término en el "nombre" de los ítems y muestra un listado de aquellos que lo contengan.</p>
	<p>Filtrar: Permite filtrar en función de la categoría (1,2, 3..) o de la prioridad (alta, media, baja), mostrando una lista de aquellos elementos que cumplan con los criterios introducidos..</p>
	<p>Ordenar: Permite ordenar los datos agrupados en función de su categoría (1,2, 3..) o de su prioridad (alta, media, baja), mostrando una lista de los datos agrupados por uno de los dos conceptos.</p>

Al aplicar los filtros siguientes **obtenemos las pantallas** que se muestran inmediatamente debajo del filtro:



C. Información del Item seleccionado



Como se puede observar la pantalla muestra los datos asociados al item seleccionado (sobre el cual se encuentra el cursor)