

INSTITUTO SUPERIOR POLITÉCNICO “JOSÉ ANTONIO ECHEVERRÍA”

FACULTAD DE INGENIERÍA INDUSTRIAL

INGENIERÍA EN INFORMÁTICA



**Herramienta para la estructuración y la gestión de contenidos de
aplicaciones multimedia educativas.**

Trabajo para optar por el título de ingeniería en Informática

Autor: Enrique Pérez Rodríguez

Tutores: Ing. Licet Gutiérrez Mompíe

Ing. Abel Ernesto Lorente Rodríguez

Ciudad de la Habana

Junio de 2005

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) y al Instituto Superior Politécnico José Antonio Echeverría (CUJAE) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los 20 días del mes de junio de 2005.

Firma del Autor

Firma del Tutor

Opinión del usuario del trabajo de diploma.

El Trabajo de Diploma, titulado: Herramienta para la estructuración y la gestión de contenidos de aplicaciones multimedia educativas, fue realizado en la UCI. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

- Totalmente
- Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes:

Como resultado de la implantación de este trabajo se reporta un efecto económico que asciende a <valor en MN o USD del efecto económico>

Y para que así conste, se firma la presente a los ____ días del mes de Junio del año 2005.

Representante de la entidad

Cargo

Firma

Cuño

Opinión del tutor del trabajo de diploma

Título: "Herramienta para la estructuración y la gestión de contenidos de aplicaciones multimedia educativas".

Autor: Enrique Pérez Rodríguez

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de

Firma

Fecha

Agradecimientos:

Resumen

El Ministerio de Educación (MINED) frecuentemente solicita a la Universidad de las Ciencias Informáticas (UCI) la elaboración de un gran número de aplicaciones multimedia educativas. Estas aplicaciones presentan una estructura común. Todas están integradas por módulos bien definidos: Presentación, Temas, Ejercicios, Biblioteca, Esquina del Maestro y Resultados.

El desarrollo del presente trabajo está encaminado a la automatización del proceso de desarrollo de las aplicaciones multimedia educativas solicitadas por el MINED en la UCI, específicamente a la parte de la estructuración y a la gestión de los contenidos de dichas aplicaciones.

El mismo ha surgido dada la necesidad que existe de lograr el desarrollo de estas aplicaciones multimedia educativas con calidad y eficiencia, utilizando el mínimo de recursos y personal y que a la vez esta actividad se realice de una manera rápida.

Se pretende desarrollar una herramienta de software que permita la automatización de la estructuración y la gestión de los contenidos de dichas aplicaciones. Las mismas además se agruparán en colecciones según las especificaciones generales de cada una de ellas. El desarrollo de esta herramienta deberá lograr que para su utilización no sea necesario el conocimiento de lenguajes de programación.

En el presente trabajo se recogen las diferentes etapas de desarrollo de dicha herramienta, así como el análisis previo a la misma que se ha realizado en cuanto a las tecnologías actuales en que se basa y el estudio de las herramientas que se encuentran en el mercado y que se utilizan para el desarrollo de aplicaciones multimedia.

Índice

Introducción.....	1
Capítulo 1: Fundamentación del tema.....	3
1.1 Introducción.	3
1.2 Descripción de los principales conceptos asociados al dominio del problema.	3
1.3 Objeto de estudio.....	5
1.4 Análisis comparativo de otras soluciones existentes con la propuesta.....	6
1.5 Descripción de las tendencias y tecnologías actuales sobre las que se apoya la propuesta.	11
1.5.1 Microsoft.NET	11
1.6 Servicios Web	15
1.7 Arquitectura de aplicaciones	17
1.8 El proceso unificado de desarrollo de software (RUP).....	20
1.9 Fundamentación del lenguaje, gestores de bases de datos y de otros <i>software</i> y <i>hardware</i> que serán utilizados.	22
1.10 Conclusiones	25
Capítulo 2: Descripción de la solución propuesta.....	25
2.1 Introducción	25
2.2 Reglas del negocio a considerar.....	26
2.3 Descripción de los procesos del negocio.....	26
2.3.1 Descripción de los actores y trabajadores del negocio.	26
2.3.2 Diagrama de casos de uso del negocio	27
2.3.3 Descripción de los Procesos del Negocio	27
2.3.4 Diagrama de Actividades	29
2.4 Diagrama de clases del modelo de objetos.	30
2.5 Requisitos Funcionales.....	31
2.6 Requisitos no funcionales.....	31
2.7 Descripción del sistema propuesta.	32
2.7.1 Concepción general del sistema	32
2.7.2 Modelo de casos de uso del sistema	34
Capítulo 3: Construcción de la solución propuesta	37
3.1 Introducción	37
3.2 Diagrama de Clases	37
3.3.1 Diagrama de clases persistentes.	45
3.3.2 Modelo de datos.....	47
3.4 Principios de diseño.....	51
3.4.1 Estándares en la interfaz de la aplicación.....	51
3.4.3 Tratamiento de excepciones.	52
3.4.3 Concepción general de la ayuda.....	52
3.5 Estándares de codificación	52
3.6 Modelo de despliegue.....	53
3.7 Modelo de implementación.....	54
3.8 Conclusiones.	70
Capítulo 4: Estudio de factibilidad.	71
4.1 Introducción.	71

4.2 Planificación.....	71
4.4 Beneficios tangibles e intangibles.....	79
4.5 Análisis de costos y beneficios.....	80
4.6 Conclusiones.....	80
Conclusiones.....	81
Recomendaciones.....	82
Bibliografía.....	83
Glosario de términos.....	84
Anexos.....	86

Introducción

En la elaboración de las aplicaciones multimedia educativas que han sido solicitadas por el MINED a la UCI participan varios equipos de trabajo: guionistas, diseñadores, especialistas de la dirección de audiovisual y programadores. Cada equipo se encarga de una tarea específica pero entre ellos existe una gran comunicación y coordinación ya que el trabajo de unos depende del de otros.

Los guionistas se encargan de la elaboración del guión (documento que contiene las pautas o especificaciones que deberán tenerse en cuenta para el desarrollo de la aplicación). El especialista de la dirección de audiovisual se encarga de la elaboración de elementos de media que contendrá la aplicación final (imágenes, video, sonido, etc.) Los diseñadores se encargan del diseño de la interfaz de la aplicación y de elaborar elementos que conformarán la misma como: animaciones, botones, etc.

Para el desarrollo de estas aplicaciones se emplean herramientas como flash y director de macromedia que son comunes para el desarrollo de aplicaciones multimedia y de probada eficiencia, sin embargo, son muy generales, o sea, no están enfocadas específicamente al desarrollo de aplicaciones multimedia educativas, por lo que se requiere programar cada aplicación desde el inicio aunque resulten muy parecidas. Otro inconveniente que presentan es la necesidad de conocer lenguajes de programación como action script y lingo para elaborar cada aplicación.

Por otra parte existen problemas de retraso en la entrega de estas aplicaciones dada la poca experiencia de los estudiantes que participan en la elaboración de las mismas y la poca relación que tienen con las herramientas de desarrollo empleadas.

Se necesita lograr el desarrollo de estas aplicaciones con un menor empleo de recursos con una mejor calidad y eficiencia.

Con el desarrollo de una herramienta de software que permita automatizar la producción de las aplicaciones multimedia educativas desarrolladas en la UCI para el MINED en forma de colecciones que permita la agrupación de estas según sus especificaciones, se logrará su obtención con una mayor calidad, mayor rapidez, menor esfuerzo y con un costo inferior no solo en tiempo, sino también en recursos humanos y materiales.

Con el presente trabajo se pretende:

Desarrollar una herramienta que permita la estructuración y gestión de los contenidos de los módulos: Presentación, Temas, Biblioteca y Esquina del Maestro pertenecientes a un producto para una colección dada.

Para lograr el objetivo propuesto es necesario

- 1- Realizar el análisis de las diferentes etapas de desarrollo en la elaboración de dicha herramienta así como el estudio de factibilidad de la misma.
- 2- Realizar un estudio de las tendencias y tecnologías actuales que se ajusten a la solución del problema planteado.
- 3- Analizar alternativas de solución que puedan ser usadas para resolver el problema planteado y realizar una comparación con la propuesta.

El presente trabajo consta de 4 capítulos. El primero de ellos: **Fundamentación del tema** que proporciona un acercamiento al funcionamiento de los procesos actuales desarrollados en la UCI en la confección de aplicaciones multimedia educativas para el MINED, sus deficiencias, así como el análisis de posibles soluciones a la problemática planteada incluyendo la comparación con herramientas de desarrollo de aplicaciones multimedia disponibles actualmente y el estudio de las tendencias y tecnologías actuales que pueden servir de base para una herramienta de desarrollo que facilite el proceso antes mencionado en la UCI.

El segundo capítulo: **Descripción de la solución propuesta** que trata sobre el sistema propuesto para darle solución a la problemática planteada y que abarca los aspectos relacionados con el mismo como son los requisitos funcionales y no funcionales, el diagrama de casos de uso del sistema etc. En este capítulo además se recoge un estudio

del negocio previo al sistema, explicando los procesos actuales del negocio. Se incluyen el diagrama de casos de uso del negocio y el diagrama de clases del modelo de objetos.

El tercer capítulo: **Construcción de la solución propuesta** abarca aspectos más específicos del sistema propuesto. Aquí se incluyen los diagramas de clases, el diseño de la base de datos y otros aspectos vinculados con la interfaz, la implementación del sistema etc.

El cuarto y último capítulo: **Estudio de factibilidad** recoge lo concerniente al estudio de factibilidad del sistema: planificación, costos, beneficios tangibles e intangibles, análisis de costos y beneficios.

Capítulo 1: Fundamentación del tema.

1.1 Introducción.

El presente capítulo constituye un acercamiento al proceso de desarrollo de aplicaciones multimedia educativas en la UCI. En el mismo se describen los principales conceptos asociados al dominio del problema así como el flujo de las actividades que intervienen en el proceso y los problemas que presenta.

Se muestra un análisis comparativo entre otras alternativas de solución a la problemática planteada y se realiza además un estudio de las tendencias y tecnologías actuales que pudieran servir como base para una herramienta de desarrollo que permita la automatización del proceso antes mencionado.

1.2 Descripción de los principales conceptos asociados al dominio del problema.

1- Guión

El guión constituye el elemento fundamental que servirá de guía a los diferentes equipos de trabajo que participan en el proceso de confección de las aplicaciones multimedia para el logro de los objetivos a través de sus pautas o especificaciones.

2- Colección

Paquete de productos agrupados por un conjunto de invariantes que constituyen el universo general de especificaciones para cualquier proyecto que se enmarque en la colección.

3- Producto

Los productos no son más que cursos con formato multimedia cuya estructura y contenidos son agrupados según sus especificaciones en colecciones.

4- Recursos mediáticos

Son elementos como: videos, sonido, imágenes, etc. que son utilizados tanto por los diseñadores como por los programadores en la confección de cada producto para cada colección. Estos son realizados por especialistas de la dirección de audiovisual.

5- Multimedia

En el campo de las Nuevas Tecnologías se puede acotar el concepto de multimedia al sistema que integra o combina diferentes medios: texto, imagen fija (dibujos, fotografías), sonidos (voz, música, efectos especiales), imagen en movimiento (animaciones, vídeos), a través de un único programa (software).

6- Hipertexto

El hipertexto es un documento donde solo se presenta información en bloques de texto unidos entre sí por nexos o vínculos que hacen que el lector elija o decida en cada momento el camino de lectura a seguir en función de los posibles itinerarios que le ofrece el programa.

7- Hipermedia

Cuando al hipertexto se le añaden dibujos, imágenes, sonidos, etc. aparece el concepto de hipermedia. Ambos son documentos no lineales, cuya información está unida por vínculos que configuran una red o malla de información, estando la diferencia entre ellos en que en el hipertexto solo hay información textual, mientras que el hipermedia incluye aparte del texto, imágenes y sonidos. [Día05].

1.3 Objeto de estudio.

EL objeto de estudio del presente trabajo es el desarrollo de las aplicaciones multimedia educativas solicitadas por el MINED en la UCI. En la elaboración de las mismas participan varios equipos de trabajo: guionistas, diseñadores, especialistas de la dirección de audiovisual y programadores. Cada equipo se encarga de una tarea específica pero entre ellos existe una gran comunicación y coordinación ya que el trabajo de unos depende del de otros.

Los guionistas se encargan de la elaboración del guión (documento que contiene las pautas o especificaciones que deberán tenerse en cuenta para el desarrollo de la aplicación). Los especialistas de la dirección de audiovisual se encargan de la elaboración de elementos de media que contendrá la aplicación final (imágenes, video, sonido, etc.) Los diseñadores se encargan del diseño de la interfaz de la aplicación así como de elaborar elementos que conformarán la aplicación como: animaciones, botones, etc. Finalmente los programadores, se encargan de engranar todos los elementos elaborados tanto por los especialistas de la dirección de audiovisual como por los diseñadores a la hora de programar la aplicación final.

Por otra parte para el desarrollo de estas aplicaciones se emplean herramientas como flash y director de macromedia que son comunes para el desarrollo de aplicaciones multimedia y de probada eficiencia, sin embargo, son muy generales, o sea, no están enfocadas específicamente al desarrollo de aplicaciones multimedia educativas, por lo que se requiere programar cada aplicación desde el inicio aunque resulten muy parecidas. Otro inconveniente que presentan es la necesidad de conocer lenguajes de programación como action script y lingo para elaborar cada aplicación.

Además existen problemas de retraso en la entrega de estas aplicaciones dada la poca experiencia de los estudiantes que participan en la elaboración de las mismas y la poca relación que tienen con las herramientas de desarrollo empleadas.

Las aplicaciones obtenidas finalmente carecen de flexibilidad y el proceso de mantenimiento de las mismas implica casi siempre tener que programar nuevamente la aplicación.

Para la confección de estas aplicaciones se están empleando gran cantidad de recursos tanto humano como materiales.

1.4 Análisis comparativo de otras soluciones existentes con la propuesta.

Para la producción de aplicaciones multimedia en general existen actualmente disponibles un gran número de herramientas. Muchas de ellas son muy usadas por su alta calidad y el gran número de prestaciones que ofrecen. A continuación se describen las características de algunas de ellas, analizando finalmente si es conveniente la utilización de alguna para el desarrollo de las aplicaciones multimedia educativas en la UCI para el MINED.

1.4.1 Herramientas de desarrollo basadas en tarjetas y páginas

1.4.1.1 ToolBook

“Toolbook es un programa que permite realizar aplicaciones Windows. Adicionalmente, posee el lenguaje de programación OpenScript orientado a objetos que enriquece extraordinariamente sus posibilidades en la generación de aplicaciones multimedia. Para facilitar las tareas de programación, ToolBook incorpora funciones de guiones propios y de grabación de otros con lo que se consiguen simplificaciones notables”

"La utilización de Toolbook en el desarrollo de programas educativos permite la creación de aplicaciones en las que, de forma sencilla y rápida se tiene la posibilidad de cambiar el flujo de la información según las necesidades del usuario, relacionar palabras, incluir cuestiones planteadas al usuario y marcadores que evalúen los conocimientos alcanzados, activar animaciones y vídeos aclarativos..." [Cab05].

Los sistemas de desarrollo basado en tarjetas y páginas proporcionan una presentación sencilla y fácil de entender para organizar los elementos de multimedia. Estos están

orientados a objeto: los objetos son botones, campo de texto, objetos gráficos, fondo, páginas o tarjetas y aun el proyecto mismo. Las características de los objetos se definen con propiedades (resaltado, negritas, rojos, escondidos, activo, bloqueado y así sucesivamente). Cada objeto puede contener un guión de programación, casi siempre una propiedad de ese objeto se activa cuando ocurre un evento (como click de un ratón) relacionado con él.

1.4.1.2 Visual Basic

Es el sistema de programación para Windows que se utiliza a menudo para organizar y presentar los elementos multimedia. Esta compuesto por controles (objetos) que residen en formas (o ventanas).

Utiliza un código de lenguaje con sintaxis similar a *BÁSICA* o a *GW-BASIC*. El programa es controlado por eventos, esto es, códigos que se asocian a objetos y que no se ejecutan hasta que son llamados a responder a los eventos creados por el usuario o el sistema, tal como hacer click con el ratón o al terminarse el tiempo de espera del sistema. Los controles se utilizan para crear la interfaces de usuario de una aplicación, incluyendo botones de orden, de opción, de verificación, cuadro de listas, cuadros combinados, cuadros de textos, barra de desplazamiento, marcos, cuadros de selección de archivos y directorios, relojes y barras de menú.

Una vez terminado su proyecto en *Visual Basic* puede convertirlo en archivo .EXE para que se ejecute como archivo de Windows independiente.

1.4.2 Herramientas de desarrollo basadas en íconos.

Las herramientas basadas en íconos y controladas por eventos brindan un enfoque de programación visual para organizar y presentar aplicaciones multimedia. Primero, se debe construir una estructura o diagrama de flujo de los eventos, tareas y decisiones, arrastrando los iconos adecuados de la biblioteca. Estos iconos pueden incluir selección de menú, imágenes gráficas, sonido y cálculos. El diagrama de flujo representa

gráficamente la lógica del proyecto. Cuando se construye la estructura se puede agregar su contenido: texto, gráficos, animación, sonido y películas de vídeo.

1.4.2.1 Authorware Professional .

Los autores sin antecedentes técnicos pueden crear aplicaciones avanzadas sin ningún guión. Colocando iconos en línea de flujo se puede crear secuencias de eventos y actividades, incluyendo toma de decisiones e interacciones del usuario. Authorware es útil como herramienta de diseño para crear secuencia de escenas porque permite cambiar las secuencias, agregar opciones y reestructurar las interacciones simplemente arrastrando y soltando los iconos.

Authorware ofrece más de doscientas variables del sistema y funciones para la captura, manipulación y despliegue de datos, y para controlar la operación del proyecto. Las variables incluyen elementos de interacción, decisión, tiempo, vídeo, gráficos, generales, archivo y de usuarios; las funciones incluyen tareas del tipo de matemáticas, cadenas, manejo de tiempo, vídeo, gráficos, generales, de archivo y del usuario.

Authorware proporciona vínculos para funciones de usuario externas escritas como DLLs en Windows.

1.4.3 Herramienta de desarrollo basadas en tiempo.

Las herramientas de desarrollo de multimedia basada en tiempo son las más comunes. Cada una utiliza su propio y único enfoque e interfaces de usuario para administrar eventos en el tiempo. Muchas emplean una línea de tiempo visual para dar secuencia a los eventos de una presentación de multimedia, a menudo desplegando capas con elementos en varios medios o en eventos a lo largo de una escala de incrementos tan precisos que alcanzan una el orden de un treintavo de segundo. Otras herramientas arreglan largas secuencias de marcos gráficos y agregan el componente de tiempo ajustando la duración de reproducción de cada marco. [Zam05].

1.4.3.1 Director 8

Es un producto único, fácil e intuitivo en su manejo. Al mismo tiempo, tiene la capacidad de trabajo que permite a los desarrolladores más expertos elaborar aplicaciones multimedia con interfaces tridimensionales, bases de datos, y conectividad con Internet. Pero además posee un lenguaje de elaboración de guiones llamado Lingo que le proporcionara un control total sobre cada uno de los aspectos del programa.

Director puede llevar a cabo todo esto a través de una metáfora que se ajusta muy bien al mundo del contenido multimedia: el mundo del cine y la animación. Cuando se crea un trabajo multimedia, al igual que cuando dirige una película, lo que hace es integrar sonidos con imágenes, movimientos que se realizan en el escenario, transiciones y efectos especiales. [Nyq00].

1.4.3.2 Flash MX 2004

Flash es la herramienta más utilizada para crear presentaciones multimedia para la Web. La idea detrás es la utilización de imágenes vectoriales, reutilización de recursos y animación interpolada; de manera de utilizar mínimo espacio.

Utilizando Flash MX 2004 se puede:

- 1- Producir animaciones o películas completas destinadas al Web, es decir, con la máxima compresión.
- 2- Incluir fácilmente en las películas archivos de sonido con la tecnología más moderna (MP3. etc.) y comprimirlos conjuntamente con los fotogramas.
- 3- Incluir la máxima interactividad en las páginas.
- 4- Producir proyectores ejecutables *standalone* para que las películas se puedan ver sin necesidad de un navegador, o sea, desde un CD-ROM.
- 5- Producir cambios en vivo (*in real time*) dentro de los elementos de la película.
- 6.- Conectar con las bases de datos de los servidores, para la actualización en tiempo real de los datos incluidos en la película.

7. Conectar con otras tecnologías para incluir en las películas datos que les sean propios, como las paginas ASP, JSP. etc.
8. Conectar con las aplicaciones que se ejecutan en los servidores de la Red de redes.
[Iss02].

1.4.4 Comparación

Las herramientas de desarrollo de aplicaciones multimedia descritas anteriormente son muy usadas actualmente y de una calidad probada. Algunas de ellas incluso incluyen lenguajes de programación que les proporcionan muchas funcionalidades. Sin embargo, todas están enfocadas al desarrollo de aplicaciones multimedia en general y persiguen otros propósitos, o sea, no están dirigidas al desarrollo de aplicaciones multimedia educativas, por lo que construir una aplicación de este tipo mediante el uso de estas herramientas resulta trabajoso.

La herramienta que se pretende desarrollar para la estructuración y la gestión de contenidos de las aplicaciones multimedia educativas desarrolladas en la UCI para el MINED brinda muchas comodidades para su uso ya que encapsula una serie de funcionalidades de las aplicaciones finales en componentes preelaborados. Permite la agrupación de productos con especificaciones comunes en colecciones, lo cuál permite que se compartan recursos comunes para varios productos. No requiere del conocimiento de lenguajes de programación para su uso. Permite definir la estructura y el diseño de las aplicaciones de manera visual, mediante el uso de controles predefinidos y elementos preelaborados como son animaciones en formato swf, imágenes, sonido, etc.

1.5 Descripción de las tendencias y tecnologías actuales sobre las que se apoya la propuesta.

1.5.1 Microsoft.NET

Microsoft.NET es el conjunto de nuevas tecnologías en las que *Microsoft* ha estado trabajando durante los últimos años con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados. Ésta es la llamada plataforma .NET.

Para crear aplicaciones para la plataforma .NET, *Microsoft* ha publicado el denominado *kit* de desarrollo de software conocido como *.NET Framework SDK*, que incluye las herramientas necesarias tanto para su desarrollo como para su distribución y ejecución y *Visual Studio.NET*, que permite hacer todo lo anterior desde una interfaz visual basada en ventanas.

“El concepto de *Microsoft.NET* también incluye al conjunto de nuevas aplicaciones que *Microsoft* y terceros han (o están) desarrollando para ser utilizadas en la plataforma .NET. Entre ellas podemos destacar aplicaciones desarrolladas por *Microsoft* tales como *Windows.NET*, *Hailstorm*, *Visual Studio.NET*, *MSN.NET*, *Office.NET*, y los nuevos servidores para empresas de *Microsoft* (*SQL Server.NET*, *Exchange.NET*, etc.)”[Gon01].

1.5.1.1 Common Language Runtime (CLR)

El CLR es el núcleo de la plataforma .NET. Es el motor encargado de gestionar la ejecución de las aplicaciones para ella desarrolladas y a las que ofrece numerosos servicios que simplifican su desarrollo y favorecen su fiabilidad y seguridad. Las principales características y servicios que ofrece el CLR son:

- **Modelo de programación consistente:** A todos los servicios y facilidades ofrecidos por el CLR se accede de la misma forma: a través de un modelo de programación orientado a

objetos, no como algunos sistemas operativos actuales (por ejemplo, los de la familia *Windows*) en los que a algunos servicios se accede a través de llamadas a funciones globales definidas en DLLs y a otros a través de objetos (objetos COM).

- **Eliminación del “infierno de las DLLs”:** En la plataforma .NET desaparece el problema conocido como “infierno de las DLLs que se da en los sistemas operativos actuales de la familia *Windows*, problema que consiste en que al sustituirse versiones viejas de DLLs compartidas por versiones nuevas puede que aplicaciones que fueron diseñadas para ser ejecutadas usando las viejas dejen de funcionar si las nuevas no son 100% compatibles con las anteriores. En la plataforma .NET las versiones nuevas de las DLLs pueden coexistir con las viejas, de modo que las aplicaciones diseñadas para ejecutarse usando las viejas podrán seguir usándolas tras la instalación de las nuevas.

- **Ejecución multiplataforma:** El CLR actúa como una máquina virtual, encargándose de ejecutar las aplicaciones diseñadas para la plataforma .NET. Es decir, cualquier plataforma para la que exista una versión del CLR podrá ejecutar cualquier aplicación .NET

- **Integración de lenguajes:** Desde cualquier lenguaje para el que exista un compilador que genere código para la plataforma .NET es posible utilizar código generado para la misma usando cualquier otro lenguaje tal y como si de código escrito usando el primero se tratase.

- **Gestión de memoria:** El CLR incluye un **recolector de basura** que evita que el programador tenga que tener en cuenta cuándo ha de destruir los objetos que dejen de serle útiles.

- **Seguridad de tipos:** El CLR facilita la detección de errores de programación difíciles de localizar comprobando que toda conversión de tipos que se realice durante la ejecución de una aplicación .NET se haga de modo que los tipos origen y destino sean compatibles.

- **Aislamiento de procesos:** El CLR asegura que desde código perteneciente a un determinado proceso no se pueda acceder a código o datos pertenecientes a otro, lo que evita errores de programación muy frecuentes e impide que unos procesos puedan atacar a otros.
- **Tratamiento de excepciones:** En el CLR todos los errores que se puedan producir durante la ejecución de una aplicación se propagan de igual manera: mediante excepciones.
- **Soporte multihilo:** El CLR es capaz de trabajar con aplicaciones divididas en múltiples hilos de ejecución que pueden ir evolucionando por separado en paralelo o intercalándose, según el número de procesadores de la máquina sobre la que se ejecuten.
- **Distribución transparente:** El CLR ofrece la infraestructura necesaria para crear objetos remotos y acceder a ellos de manera completamente transparente a su localización real, tal y como si se encontrasen en la máquina que los utiliza.
- **Seguridad avanzada:** El CLR proporciona mecanismos para restringir la ejecución de ciertos códigos o los permisos asignados a los mismos según su procedencia o el usuario que los ejecute
- **Interoperabilidad con código antiguo:** El CLR incorpora los mecanismos necesarios para poder acceder desde código escrito para la plataforma .NET a código escrito previamente a la aparición de la misma y, por tanto, no preparado para ser ejecutado dentro de ella.

“Como se puede deducir de las características comentadas, el CLR lo que hace es gestionar la ejecución de las aplicaciones diseñadas para la plataforma .NET. Por esta razón, al código de estas aplicaciones se le suele llamar **código gestionado**, y al código no escrito para ser ejecutado directamente en la plataforma .NET se le suele llamar **código no gestionado**” [Gon01].

1.5.1.2 Microsoft Intermediate Language (MSIL)

Todos los compiladores que generan código para la plataforma .NET en vez de generar código máquina para CPUs x86 o para algún otro tipo de CPU concreta, generan código escrito en el lenguaje intermedio conocido como *Microsoft Intermediate Language* (MSIL). El CLR da a las aplicaciones la sensación de que se están ejecutando sobre una máquina virtual, y precisamente MSIL es el código máquina de esa máquina virtual. Es decir, MSIL es el único código que es capaz de interpretar el CLR.

MSIL es un lenguaje de un nivel de abstracción mucho más alto que el de la mayoría de los códigos máquina de las CPUs existentes, e incluye instrucciones que permiten trabajar directamente con objetos (crearlos, destruirlos, inicializarlos, llamar a métodos virtuales, etc.), tablas y excepciones (lanzarlas, capturarlas y tratarlas)

La principal ventaja del MSIL es que facilita la ejecución multiplataforma y la integración entre lenguajes al ser independiente de la CPU y proporcionar un formato común para el código máquina generado por todos los compiladores que generen código para .NET. Sin embargo, dado que las CPUs no pueden ejecutar directamente MSIL, antes de ejecutarlo habrá que convertirlo al código nativo de la CPU sobre la que se vaya a ejecutar. De esto se encarga un componente del CLR conocido como compilador JIT (*Just-In-Time*) o *jitter* que va convirtiendo dinámicamente el código MSIL a ejecutar en código nativo según sea necesario.

1.5.1.3 La Librería de Clase Base (BCL)

La Librería de Clase Base (BCL) es una librería incluida en el *.NET Framework* formada por cientos de tipos de datos que permiten acceder a los servicios ofrecidos por el CLR y a las funcionalidades más frecuentemente usadas a la hora de escribir programas. Además, a partir de estas clases prefabricadas se pueden crear nuevas clases que mediante herencia extiendan su funcionalidad y se integren a la perfección con el resto de clases de la BCL.

Esta librería está escrita en MSIL, por lo que puede usarse desde cualquier lenguaje cuyo compilador genere MSIL. A través de las clases suministradas en ella es posible desarrollar cualquier tipo de aplicación, desde las tradicionales aplicaciones de ventanas, consola o servicio de *Windows NT* hasta los novedosos servicios web.

1.5.1.4 Seguridad

La faceta más importante de cualquier entorno de desarrollo de aplicaciones distribuida es cómo maneja la seguridad. Con .Net la seguridad comienza tan pronto como se carga una clase por el CLR, ya que el cargador de clases es parte del esquema de seguridad de .NET. Cuando se carga una clase en el entorno de ejecución .NET, se comprueban los factores relacionados con la seguridad tales como reglas de accesibilidad y requisitos de autoconsistencia. Además, los chequeos de seguridad aseguran que un trozo de código tenga las credenciales apropiadas para acceder a ciertos recursos. El código de seguridad asegura la determinación de los roles y la información de identidad. Estos chequeos de seguridad incluso se extienden a procesos y más allá de los límites de la máquina para asegurar que los datos sensibles no estén comprometidos en entornos de ejecución distribuidos. [Arc01].

1.6 Servicios Web

Un servicio web es simplemente un componente de servidor que puede ser invocado en Internet. Este componente de servidor normalmente realiza un servicio fundamental de negocios como la autenticación del usuario, la validación de tarjetas de crédito, el cálculo del precio de un seguro de derivados, etc.

Los servicios web permiten a las aplicaciones invocar servicios de negocios mediante un mecanismo basado en estándares (usando xml y http). En la invocación real de un servicio web toman parte tres tecnologías: el protocolo de conexión, el formato de mensaje y el mecanismo de invocación.

El protocolo de conexión es el mecanismo de transporte que se emplea para establecer la comunicación entre el cliente y el servidor. Por lo general, suele ser http, protocolo de

Internet basado en tcp/ip. El formato de mensaje es el formato que se emplea para invocar un servicio web. Un servicio web puede ser invocado mediante http puro o con un mensaje xml en formato específico llamado Protocolo de Acceso simple a objetos (SOAP).

La tercera tecnología que controla cómo se llaman a los componentes de servidor, no esta especificada por los estándares del servicio web. El programador que crea el servicio web elige la tecnología usada para llamar al código de negocio en el servidor. Esta puede ser por ejemplo COM+.

Las dos partes de un servicio web pueden describirse como el creador (cliente) y el consumidor (servidor). El creador desarrolla el componente de servidor y muestra este servicio a quien corresponda. Mostrar un servicio web significa publicar la url que los usuarios necesitan para invocar al servicio web. El consumidor puede usar el servicio expuesto enviando un mensaje de petición SOAP a la url publicada. Al recibir una petición SOAP escrita en xml, el componente de servidor tras el servicio web es invocado en el servidor creador. Los resultados de esta invocación toman el formato de un mensaje de respuesta SOAP y se envían de vuelta al consumidor.

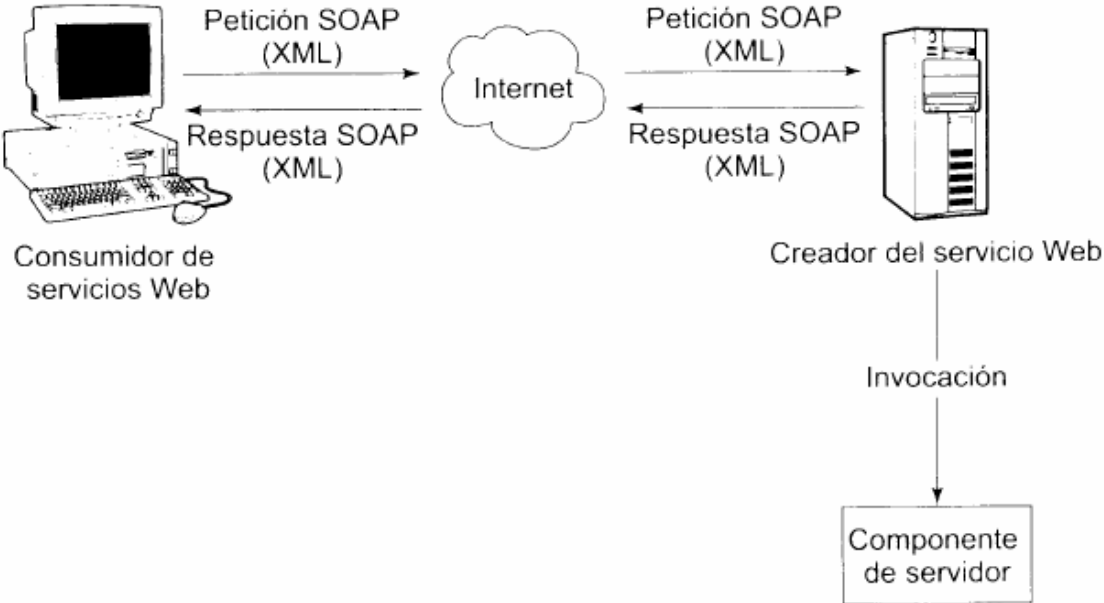


Figura 1 Representación del servicio web

El sistema operativo del cliente o del servidor, el lenguaje de programación usado para escribir el componente de servidor o el mecanismo utilizado para invocar el componente del servidor no tienen importancia para un servicio web. Un cliente escrito en C# que se ejecuta en Windows xp, por ejemplo, puede invocar un servicio web escrito en java ejecutándose en *Sun Solaris*. De hecho un cliente no tiene forma de saber qué tecnologías se usan para mostrar el servicio web.

“Las implicaciones de los servicios web son enormes. *Microsoft* comenzó la programación basada en componentes con la introducción de OLE (OCX), una versión depurada de los innovadores conceptos presentados por los controles de *Visual Basic* (VBL). Los OCX se basan en el Modelo de Objetos de Componentes de *Microsoft* (COM) y funciona perfectamente. Sin embargo COM y su contrapartida distribuida el modelo de objetos de componentes distribuido de *Microsoft* (DCOM) tienen un alcance limitado. Aparte de la familia de sistemas operativos Windows muy pocos sistemas admiten COM/ DCOM. Además aunque la familia Windows tiene una amplia aceptación para aplicaciones de escritorio, en la modalidad de servidor se usan una gran variedad de sistemas operativos, por ejemplo, muchas variedades del sistema operativo *UNIX*, como *Solaris* y *Linux* tienen una importante presencia. “

Los servicios web eliminan la necesidad de decidir el sistema operativo que se ejecuta en el servidor al integrar dos aplicaciones web. Al usar servicios web puede ensamblar aplicaciones usando componentes de otros servidores. Por ejemplo puede usar un servicio web de una compañía de tarjetas para validar tarjetas de crédito, un servicio web de una compañía distribuidora para determinar los gastos de envío y así sucesivamente. [Fer03]

1.7 Arquitectura de aplicaciones

1.7.1 Arquitectura de dos capas

La arquitectura tradicional de cliente/servidor también es conocida como arquitectura de dos capas. Requiere una interfaz de usuario que se instala y corre en una PC o estación de trabajo y que envía solicitudes a un servidor para ejecutar operaciones complejas. Por

ejemplo, una estación de trabajo utilizada como cliente puede correr una aplicación de interfaz de usuario que interroga a un servidor central de bases de datos.

En la mayoría de los casos, el cliente requiere recursos y servicios que el servidor le proporciona. Ambos procesos pueden existir en una máquina o en máquinas separadas que se comunican a través de una red. La aplicación lógica es dividida entre las aplicaciones del cliente y la base de datos.

1.7.2Arquitectura de tres capas

La arquitectura de tres capas se refiere a un diseño reciente que introduce una capa intermedia al proceso. Cada capa es un proceso separado y bien definido corriendo en plataformas separadas. La parte funcional de la arquitectura de tres capas generalmente es conocida como la capa intermedia o el servidor de aplicaciones. En éste ocurren la mayoría de los procesos.

1.7.3Servidor de Aplicaciones

El servidor de aplicaciones fue introducido como parte del diseño de tres capas. Es relativamente nuevo y aún no bien definido. Las empresas del mundo entero están esforzándose para producir su propia versión de lo que creen que es un servidor de aplicaciones. La definición más común de un servidor de aplicaciones es la de software corriendo en una capa intermedia entre un cliente pequeño y una base de datos. Generalmente se acepta que un servidor de aplicaciones maneja todas las transacciones lógicas y de conectividad que históricamente compartían el cliente y el servidor en un diseño cliente/servidor.

1.7.4 Análisis comparativo entre ambas arquitecturas

- **Ventajas de la arquitectura de dos capas**

- El desarrollo de aplicaciones en un ambiente de dos capas es mucho más rápido que en ambientes anteriores, pero no es necesariamente más rápido que con el nuevo ambiente de tres capas.

- Las herramientas para el desarrollo con dos capas son robustas.

- Las soluciones de dos capas trabajan bien en ambientes no dinámicos estables, pero no se ejecutan bien en organizaciones rápidamente cambiantes.

- **Desventajas de la arquitectura de dos capas**

- Los ambientes de dos capas requieren control excesivo de las versiones y demandan esfuerzo de distribución de la aplicación cuando se les hacen cambios. Esto se debe al hecho de que la mayoría de la aplicación lógica existe en la estación de trabajo del cliente.

- La seguridad del sistema en un diseño de dos capas es compleja y a menudo requiere administración de las bases de datos; esto es debido al número de dispositivos con acceso directo al ambiente de esas bases de datos.

- Las herramientas del cliente y de la base de datos, utilizadas en diseños de dos capas, constantemente están cambiando. La dependencia a largo plazo de cualquier herramienta, puede complicar el escalamiento futuro o las implementaciones.

- **Ventajas de la arquitectura de tres capas**

- Las llamadas de la interfaz del usuario en la estación de trabajo, al servidor de capa intermedia, son más flexibles que en el diseño de dos capas, ya que la estación sólo necesita transferir parámetros a la capa intermedia.

- Con la arquitectura de tres capas, la interfaz del cliente no es requerida para comprender o comunicarse con el receptor de los datos. Por lo tanto, esa estructura de los datos puede ser modificada sin cambiar la interfaz del usuario en la PC (*Personal Computer*).
- El código de la capa intermedia puede ser reutilizado por múltiples aplicaciones si está diseñado en formato modular. Esto puede reducir los esfuerzos de desarrollo y mantenimiento, así como los costos de migración.
- La separación de roles en tres capas, hace más fácil reemplazar o modificar una capa sin afectar a los módulos restantes.
- Separando la interfaz del usuario de la aplicación, libera de gran procesamiento a la estación de trabajo y permite que las actualizaciones de la aplicación sean centralizadas en el servidor de aplicaciones.

- **Desventajas de la arquitectura de tres capas**

- Los ambientes de tres capas pueden incrementar el tráfico en la red y requerir más tolerancia a las fallas. [Jan04].

1.8 El proceso unificado de desarrollo de software (RUP)

RUP es un proceso de desarrollo de software, o sea, el conjunto de actividades necesarias para transformar los requisitos del usuario en un sistema de software, y es además un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de *software*, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

Está basado en componentes, lo que quiere decir que el sistema *software* en construcción está formado por componentes *software* interconectados a través de interfaces. Este además utiliza el Lenguaje Unificado de Modelado (UML).

El ciclo de vida de RUP se caracteriza por:

- **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso. El usuario no solo representa a una persona sino que comprende además a otros sistemas.
- **Centrado en la arquitectura:** El concepto de arquitectura incluye los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa como la perciben los usuarios y los inversores, y se refleja en los casos de uso. Esta además se ve influida por otros factores como la plataforma en que debe funcionar el software, arquitectura *hardware*, sistema operativo, etc. En resumen, es una vista del diseño completo con las características más importantes resaltadas, dejando los detalles de lado.
- **Iterativo e incremental:** RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. Propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. [Jac00].

1.9 Fundamentación del lenguaje, gestores de bases de datos y de otros software y hardware que serán utilizados.

1.9.1C#

C# es un lenguaje de programación que toma las mejores características de lenguajes preexistentes como *Visual Basic*, *Java* o *C++* y las combina en uno solo. *Microsoft* ha escrito la mayor parte de la BCL usándolo, por lo que su compilador es el más depurado y optimizado de los incluidos en el *.NET Framework SDK*.

Entre sus características más sobresalientes se encuentran las siguientes:

- **Sencillez:** C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET. Por ejemplo:

- El código escrito en C# es autocontenido, lo que significa que no necesita de ficheros adicionales al propio fuente tales como ficheros de cabecera.

- El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile (no como en C++, por ejemplo), lo que facilita la portabilidad del código.

- No se incluyen elementos poco útiles de lenguajes como C++ tales como macros, herencia múltiple o la necesidad de un operador diferente del punto (.)

- **Modernidad:** C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como *Java* o *C++* hay que simular, como un tipo básico decimal que permita realizar operaciones de alta precisión con reales de 128 bits (muy útil en el mundo financiero), la inclusión de una instrucción *foreach* que permita recorrer colecciones con facilidad y es ampliable a tipos definidos por el usuario, la inclusión de un tipo básico *string* para representar cadenas o la distinción de un tipo *bool* específico para representar valores lógicos.

- **Orientación a objetos:** Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos. Una diferencia de este enfoque orientado a objetos respecto al de otros lenguajes como C++ es que el de C# es más puro en tanto que no admiten ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código.

C# soporta todas las características propias del paradigma de programación orientada a objetos: encapsulación, herencia y polimorfismo.

- **Orientación a componentes:** La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas.

- **Gestión automática de memoria:** Todo lenguaje de .NET tiene a su disposición el recolector de basura del CLR. Esto tiene el efecto en el lenguaje de que no es necesario incluir instrucciones de destrucción de objetos. Sin embargo, C# también proporciona un mecanismo de liberación de recursos determinista.

- **Seguridad de tipos:** C# incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, lo que permite evitar que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto y es especialmente necesario en un entorno gestionado por un recolector de basura.

- **Sistema de tipos unificado:** En C# todos los tipos de datos que se definan siempre derivarán, aunque sea de manera implícita, de una clase base común llamada *System.Object*, por lo que dispondrán de todos los miembros definidos en ésta clase (es decir, serán “objetos”).

- **Eficiente:** En principio, en C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, y a diferencia de

Java, en *C#* es posible saltarse dichas restricciones manipulando objetos a través de punteros.

- **Compatible:** Para facilitar la migración de programadores, *C#* no sólo mantiene una sintaxis muy similar a *C*, *C++* o *Java* que permite incluir directamente en código escrito en *C#* fragmentos de código escrito en estos lenguajes, sino que el CLR también ofrece la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos tales como las DLLs de la API Win32. También es posible acceder desde código escrito en *C#* a objetos COM. [Gon01].

1.9.2 Microsoft SQL Server 2000

Tal vez no es considerado como el mejor sistema gestor de bases de datos relacionales (RDBMS), pues es aventajado en algunos aspectos como la disponibilidad de versiones para múltiples plataformas por otros gestores como *Oracle* y *MySQL*, pero es uno de los RDBMS más rápidos disponibles actualmente.

Visual *C#* cuenta con un proveedor ADO.Net nativo para *SQLServer*. Además Microsoft lo ha desarrollado con el objetivo de explotar al máximo las características de los sistemas operativos Windows.

SQL Server cuenta con un lenguaje (Transact-SQL) para programar procedimientos almacenados y *triggers*, permite definir tablas, índices, vistas, etc., es distribuido y escalable, con soporte para 32 procesadores y 64 GB de RAM, es más fácil de usar que el *Oracle* y más potente que *MySQL*, brinda facilidades de replicación, seguridad administrada según perfiles configurables, que puede ser sobre cuentas de usuarios propias o integrada con las de Windows, ofrece además mecanismos de salva y restauración, y posibilidad de importar y exportar los datos en múltiples formatos [Mar01].

1.10 Conclusiones

Se ha determinado que las herramientas de software que se han analizado en este capítulo y que pueden ser utilizadas para el desarrollo de aplicaciones multimedia educativas y en particular para la estructuración y la gestión de los contenidos de las mismas, resultan insuficientes para resolver la problemática planteada, por lo que se propone como solución el desarrollo de una herramienta que agrupe algunas de las características de las mismas e incorpore un grupo nuevo de funcionalidades como por ejemplo el hecho de agrupar algunas de las tareas comunes a las aplicaciones multimedia educativas a desarrollar en componentes de fácil acceso por parte de los operadores de las mismas, permitir el agrupamiento de los productos en colecciones que tengan las mismas especificaciones, entre otras.

Después de haber analizado las características más importantes de las nuevas tecnologías existentes para el desarrollo de software se ha determinado esta herramienta de software que se pretende desarrollar en la UCI para la estructuración y gestión de contenidos de las aplicaciones multimedia educativas solicitadas por el MINED puede desarrollarse basada en .Net, utilizando como lenguaje de programación el C# y como gestor de base de datos SQL Server 2000. El diseño de la misma estará basado en una arquitectura de tres capas, aprovechando las múltiples ventajas que esta arquitectura brinda. Además para la capa intermedia de la aplicación se usarán varios servicios web que se encargarán de realizar las consultas a la capa de datos para satisfacer las peticiones del cliente.

Capítulo 2: Descripción de la solución propuesta.

2.1 Introducción

Este capítulo trata sobre la solución propuesta. Describe los procesos del negocio así como las reglas del negocio que se deben considerar. En él se describe además el sistema propuesto, los requisitos que debe cumplir y los diagramas de casos de uso asociados al mismo divididos por paquetes.

2.2 Reglas del negocio a considerar

1- Los guionistas podrán objetar las soluciones dadas por los diseñadores en cuanto a las especificaciones gráficas de la interfaz, si éstas resultan improcedentes desde el punto de vista pedagógico, por lo que el contacto multidisciplinario deberá constituir un principio de trabajo, de lo contrario deberán presentarse los esbozos de los elementos de diseño para ser presentados previamente al equipo de guionistas.

2- Los elementos elaborados por cada uno de los equipos de trabajo que participan en el proceso de elaboración de aplicaciones multimedia educativas serán colocados en una estructura de directorios previamente definida a la que tendrán acceso todos, por lo que cualquier integrante de uno de estos equipos podrá acceder a los recursos que necesite para llevar a cabo su trabajo. De esta forma se facilita la coordinación y la comunicación entre los diferentes equipos de trabajo, a la vez que se agiliza el proceso.

2.3 Descripción de los procesos del negocio

2.3.1 Descripción de los actores y trabajadores del negocio.

1- Cliente (Actor del negocio)

Persona o institución que solicite a la UCI la confección de una Aplicación Multimedia Educativa, por ejemplo: El MINED.

2- Guionista (Trabajador del negocio)

Persona o conjunto de personas que se encargan de la elaboración del guión.

3- Diseñador (Trabajador del negocio)

Persona o conjunto de personas que se encargan del diseño de la interfaz de cada producto así como de la elaboración de elementos que serán usados por los programadores en la elaboración del producto como: botones, animaciones etc.

4- Especialista de la dirección de audiovisual (Trabajador del negocio)

Persona o conjunto de personas que se encargan de la elaboración de elementos de media: imágenes, sonidos etc. que serán usados por los diseñadores y por los programadores en la confección de los elementos que conforman al producto final.

5- Programador (Trabajador del negocio)

Persona o conjunto de personas que se encargan de la programación del producto final, usando los elementos facilitados por los diseñadores y los especialistas de la dirección de audiovisual.

2.3.2 Diagrama de casos de uso del negocio

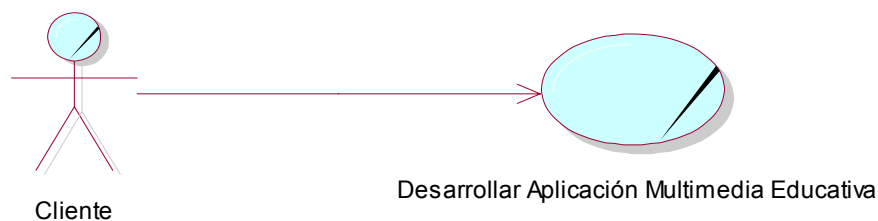


Figura 2 Diagrama de casos de uso del negocio

2.3.3 Descripción de los Procesos del Negocio

Una vez que un cliente solicita la confección de una Aplicación Multimedia Educativa, el guionista confecciona el guión que servirá como patrón para los demás equipos de trabajo que intervienen en el proceso (especialistas de la dirección de audiovisual, diseñadores y programadores).

Una vez elaborado el guión el especialista de la dirección de audiovisual comienza la confección y gestión de los recursos de media: videos, sonido y otros elementos de este tipo que serán usados tanto por los diseñadores como por los programadores para la confección de cada aplicación.

Los diseñadores son los encargados de la confección de otros elementos como son: botones, animaciones, etc. Además se encargan del diseño de la interfaz de cada aplicación.

Los recursos elaborados o gestionados por cualquier integrante de los equipos de diseño o de la dirección de audiovisual son colocados en una estructura de directorios a la que tendrán acceso todos para que cada cual utilice los recursos que necesita en su trabajo.

Por último los programadores usando los recursos elaborados tanto por los especialistas de la dirección de audiovisual como por los diseñadores se encargan de engranar todos estos elementos mediante la programación para obtener el producto final.

2.3.4 Diagrama de Actividades

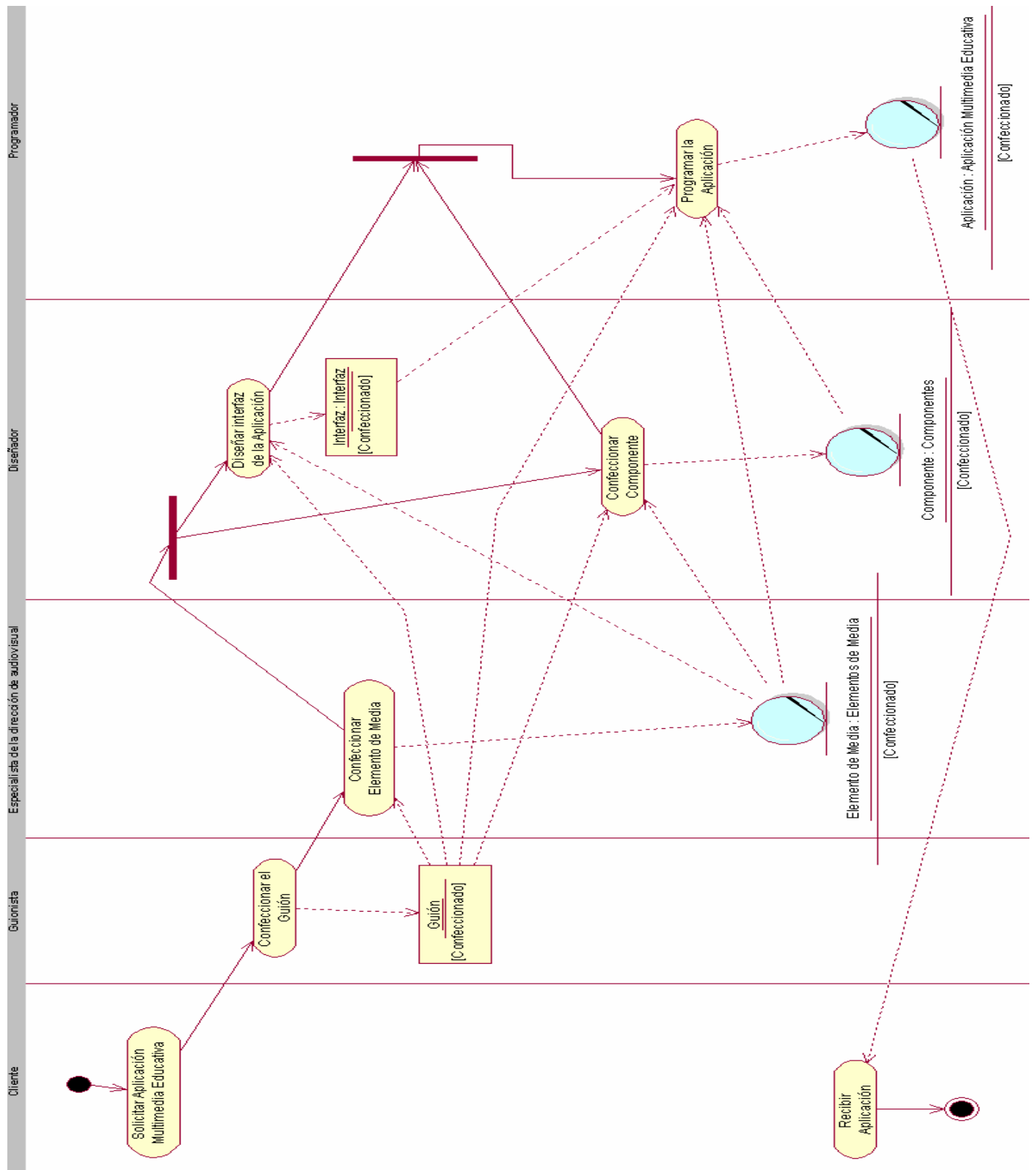


Figura 3 Diagrama de Actividades

2.4 Diagrama de clases del modelo de objetos.

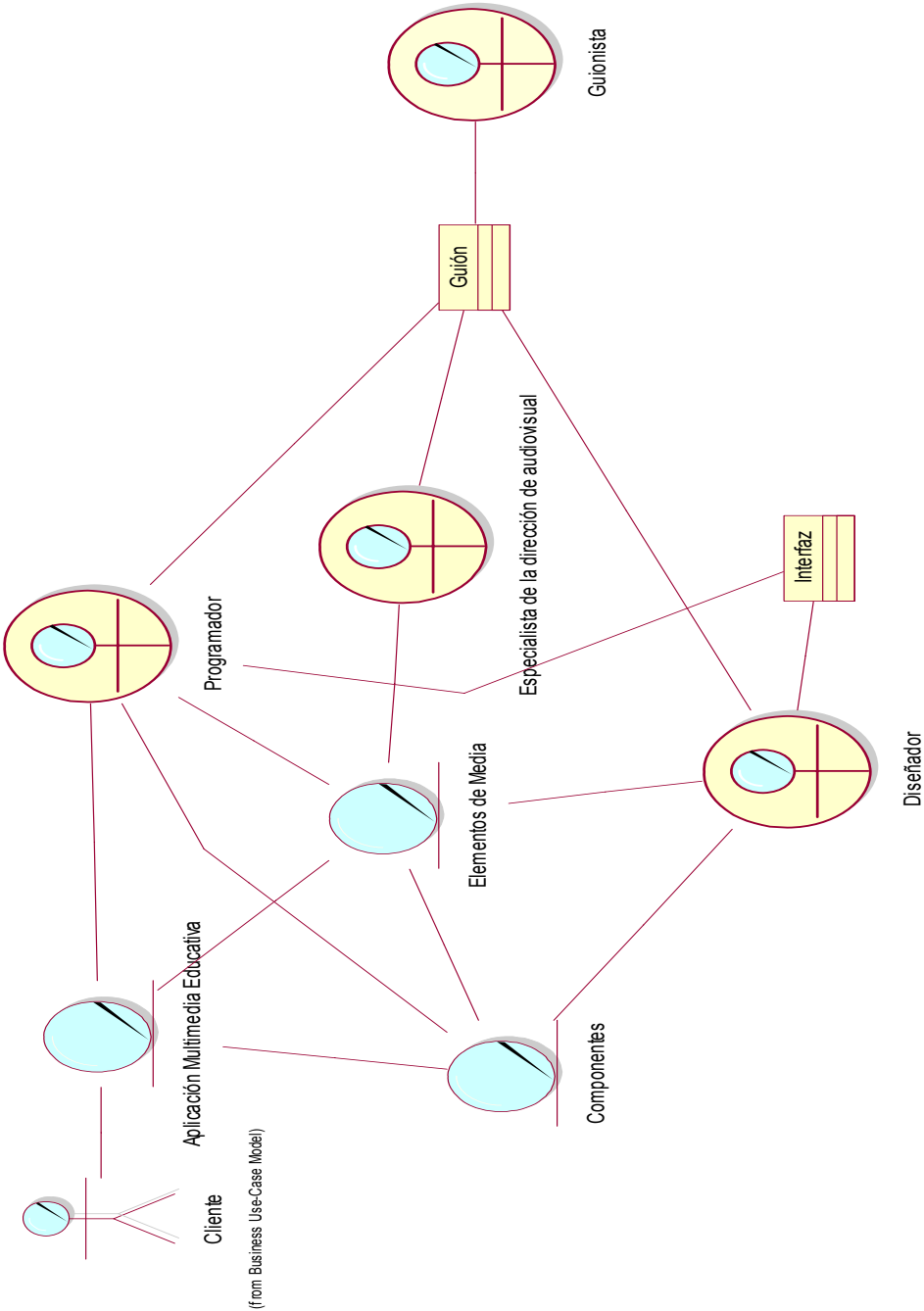


Figura 4 Diagrama de clases del modelo de objetos

2.5 Requisitos Funcionales

- 1- Gestionar Colecciones.
- 2- Gestionar productos para una colección dada.
- 3- Gestionar módulos para un producto.
- 4- Gestionar prototipos de pantalla para cada módulo de cada producto de cada colección.
- 5- Gestionar controles asociados a un prototipo de pantalla.
- 6- Gestionar Temas para un producto.
- 7- Gestionar epígrafes para un tema de un producto.
- 8- Gestionar palabras calientes.
- 9- Gestionar pantallas asociadas a un epígrafe.
- 10- Gestionar los controles asociados a una pantalla de un epígrafe
- 11- Gestionar las palabras del glosario para un producto.
- 12- Gestionar los elementos de media que estarán disponibles en la Biblioteca para cada producto.
- 13- Gestionar efemérides para cada producto.
- 14- Gestionar programas de las asignaturas.
- 15- Gestionar recomendaciones metodológicas para cada producto.
- 16- Gestionar artículos de actualización que podrán consultar los profesores.

2.6 Requisitos no funcionales.

- 1- La interfaz externa deberá ser lo más atractiva y clara posible para el usuario Debe ser fácil de comprender su funcionamiento.
- 2- El sistema debe ser eficiente, con una capacidad adecuada de procesamiento y cálculo, así como requiere de un tiempo de respuesta relativamente pequeño.
- 3- El sistema debe ser adaptable.
- 4- Los datos relacionados con los usuarios deben estar encryptados.
- 5- Es necesario establecer control de acceso a los servicios por parte de los usuarios.

- 6- El sistema debe correr sobre la plataforma *Microsoft .NET Framework 1.1* o versiones superiores.
- 7- Para el desarrollo y ejecución de esta aplicación se necesita conexión a la red local, por lo que se requiere tarjeta de red.
- 8- El servidor de Base de Datos debe soportar *Microsoft SQL Server 2000*.
- 9- Se debe garantizar la integridad de los datos manipulados por el sistema.

2.7 Descripción del sistema propuesta.

Nota: Para ver la descripción de los casos de uso del sistema ver anexos.

Actor del sistema: Operador

Justificación:

Es la persona que opera el sistema y puede realizar cualquier acción sobre el mismo excepto las de administración. Los requerimientos funcionales asociados al mismo son todos los descritos excepto:1, 2, 3, 4.

Actor del sistema: Administrador

Justificación:

Es un operador que además realiza las tareas de administración. Todos los requerimientos funcionales están asociados a él.

2.7.1 Concepción general del sistema

La herramienta que se propone desarrollar automatizará la estructuración y la gestión de los contenidos de las aplicaciones multimedia educativas desarrolladas en la UCI para el MINED, permitiendo agrupar esta información en colecciones dividida por productos, específicamente la información relacionada con los módulos: Presentación, Temas, Biblioteca y Esquina del Maestro de estas aplicaciones, así como la gestión de los prototipos de pantalla vinculados a estos módulos, las pantallas asociadas a un tema y un epígrafe y los elementos que los componen (controles).

Dicha herramienta se construirá sobre la plataforma.NET, utilizando el lenguaje de programación C#, aprovechando las amplias ventajas que el mismo ofrece y que fueron descritas en el capítulo 1. Para la gestión de los controles que componen los prototipos de pantalla y las pantallas de epígrafes se utilizan los controles de .NET aprovechando sus propiedades que se ajustan a las de los controles que deberán tener los productos finales obtenidos a partir de la utilización de la herramienta en cuestión. De esa forma para simular una etiqueta de un producto final se utiliza un label de .NET, para simular una imagen se usa un Picture Box y así sucesivamente. De esta forma la herramienta brinda un ambiente que permite la creación de prototipos de pantalla y pantallas de epígrafes de forma visual a través de controles predefinidos.

Todos los recursos que se utilizan para la confección de un producto: elementos de media y componentes elaborados por los diseñadores y los especialistas de la dirección de audiovisual, serán colocados en una estructura de directorios predeterminada y con carácter permanente.

El sistema además constará de políticas de seguridad otorgando a cada usuario los derechos que le corresponden. Existirán dos tipos de usuario: administradores y operadores. Los primeros son los encargados de las tareas administrativas dentro del sistema: gestión de usuarios, gestión de colecciones, gestión de productos, gestión de módulos y gestión de prototipos de pantalla. El resto de las tareas son compartidas por ambos usuarios.

Para garantizar la seguridad en este sentido se utiliza el cifrado de la contraseña del usuario mediante el algoritmo de cifrado MD5. Cada usuario que entra al sistema debe autenticarse antes de realizar alguna acción.

El gestor de base de datos utilizado para el almacenamiento de las colecciones y todos los elementos que las componen será SQL Server 2000. El sistema está basado en una arquitectura de tres capas. La capa intermedia consta de varios servicios web que se

encargan del acceso a los datos, separando así la capa cliente de la capa de datos lo cuál hace al sistema más flexible.

2.7.2 Modelo de casos de uso del sistema

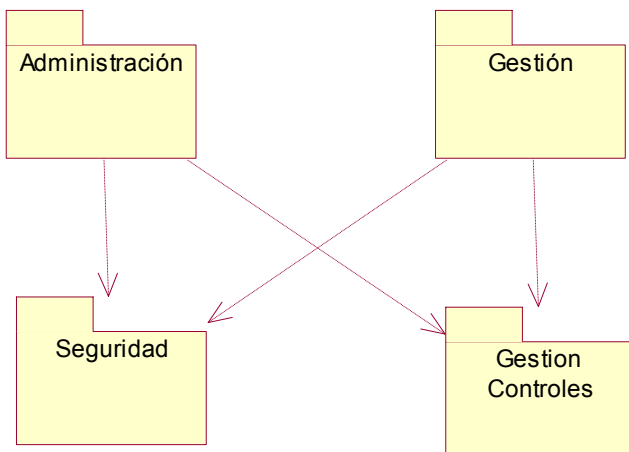


Figura 5 Diagrama de casos de uso por paquetes

2.7.3 Paquete de seguridad



Figura 6 Diagrama de casos de uso del paquete de seguridad

2.7.4 Paquete de administración

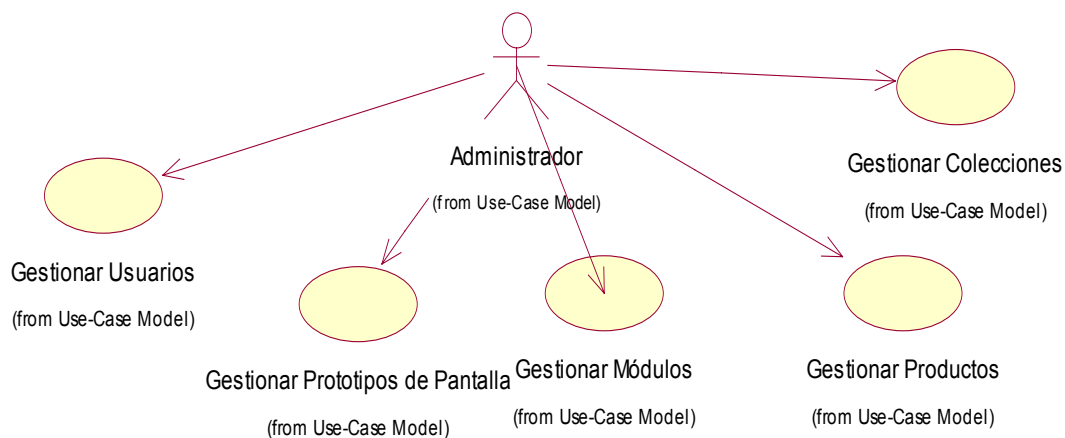


Figura 7 Diagrama de casos de uso del paquete de administración

2.7.5 Paquete de gestión

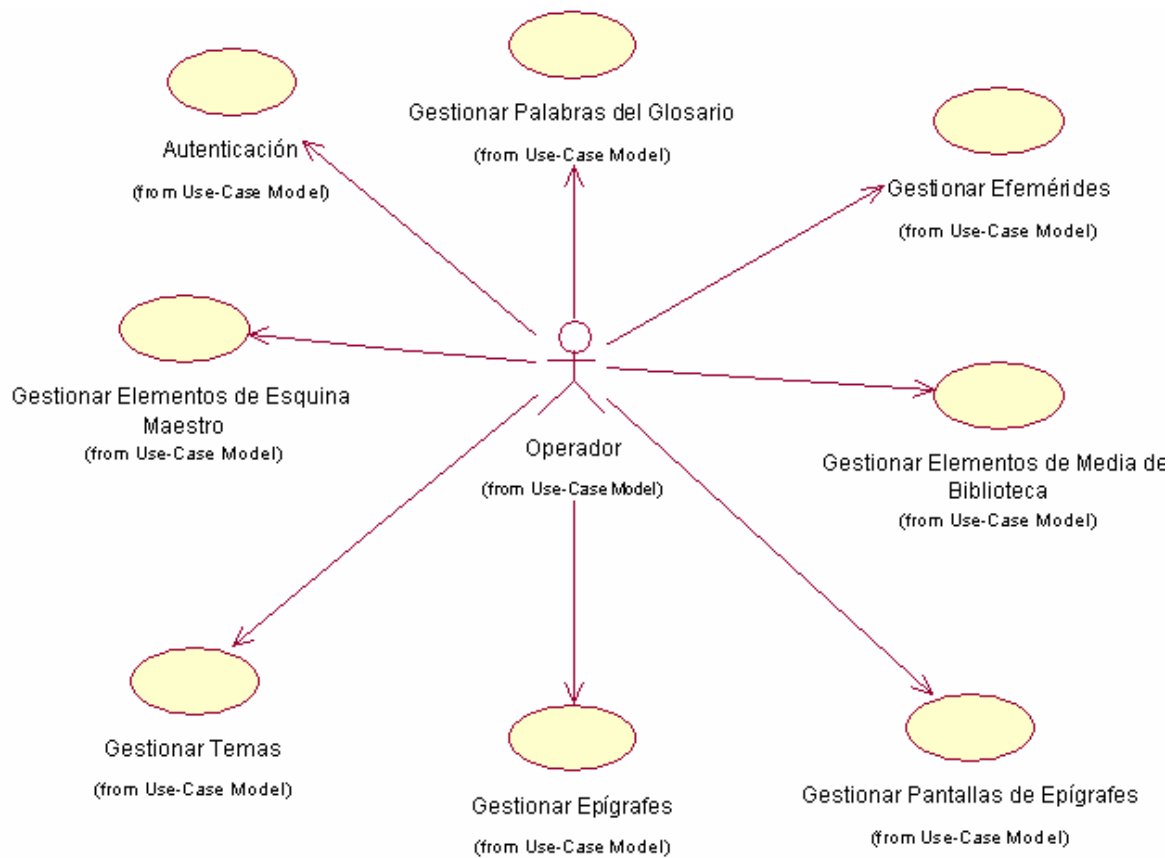


Figura 8 Diagrama de casos de uso del paquete de gestión

2.7.6 Paquete de gestión de controles

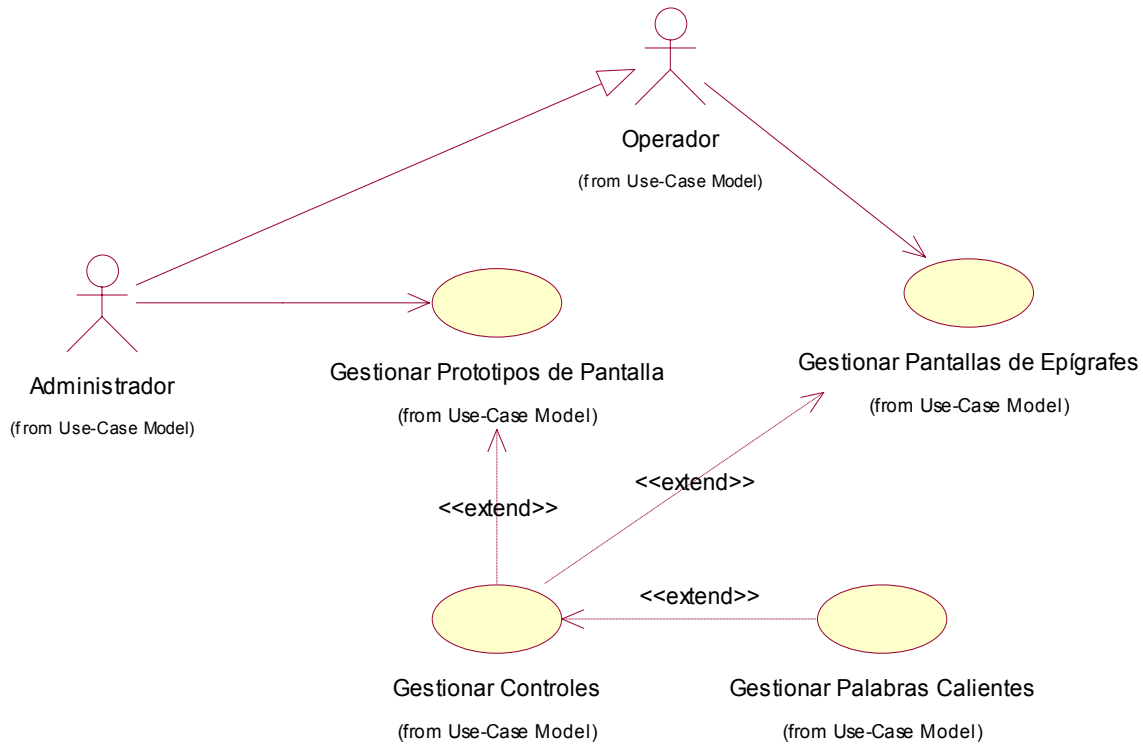


Figura 9 Diagrama de casos de uso del paquete de gestión de controles

2.8 Conclusiones

En este capítulo se han expuesto los principales conceptos asociados al negocio así como la descripción del sistema propuesto, especificando los requerimientos que el mismo debe tener logrando ajustar la funcionalidad del sistema con los problemas reales que presenta el negocio. El comportamiento del sistema ha quedado descrito mediante el diagrama de casos de uso del sistema y este refleja que el mismo da respuesta a la problemática planteada.

Capítulo 3: Construcción de la solución propuesta

3.1 Introducción

En este capítulo se profundiza en el sistema propuesto, en específico lo relacionado con la construcción del mismo. Se representan los diagramas de clases, el modelo de datos obtenido así como aspectos relacionados con la interfaz de la aplicación, el tratamiento de excepciones etc. De la misma forma se representa la arquitectura empleada en su solución y se representan el modelo de despliegue y el diagrama de componentes.

3.2 Diagrama de Clases

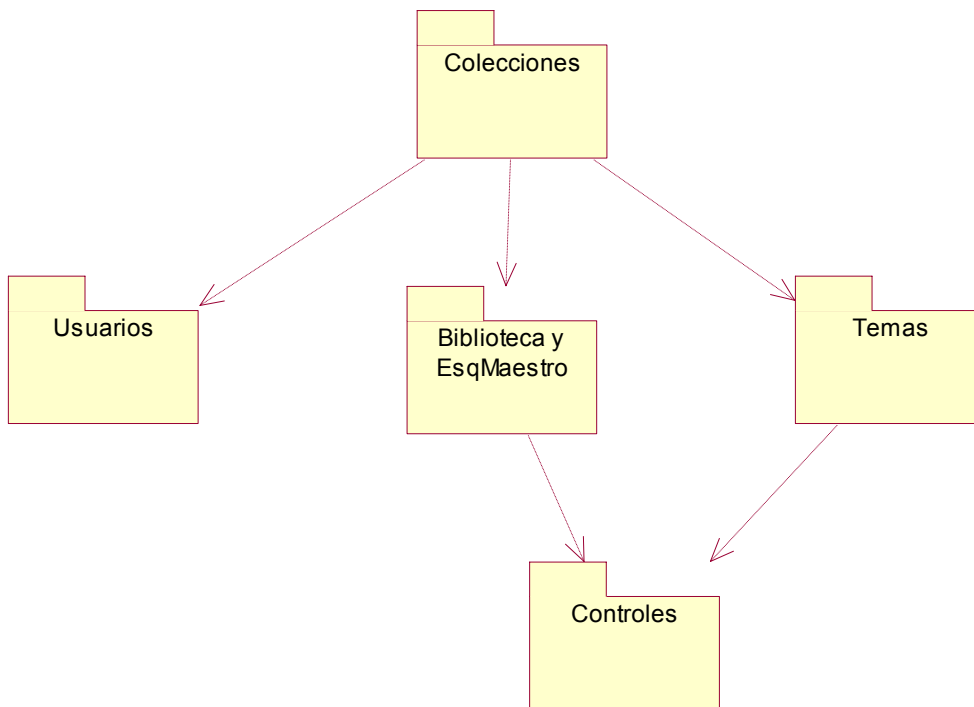


Figura 10 Diagrama de clases por paquetes

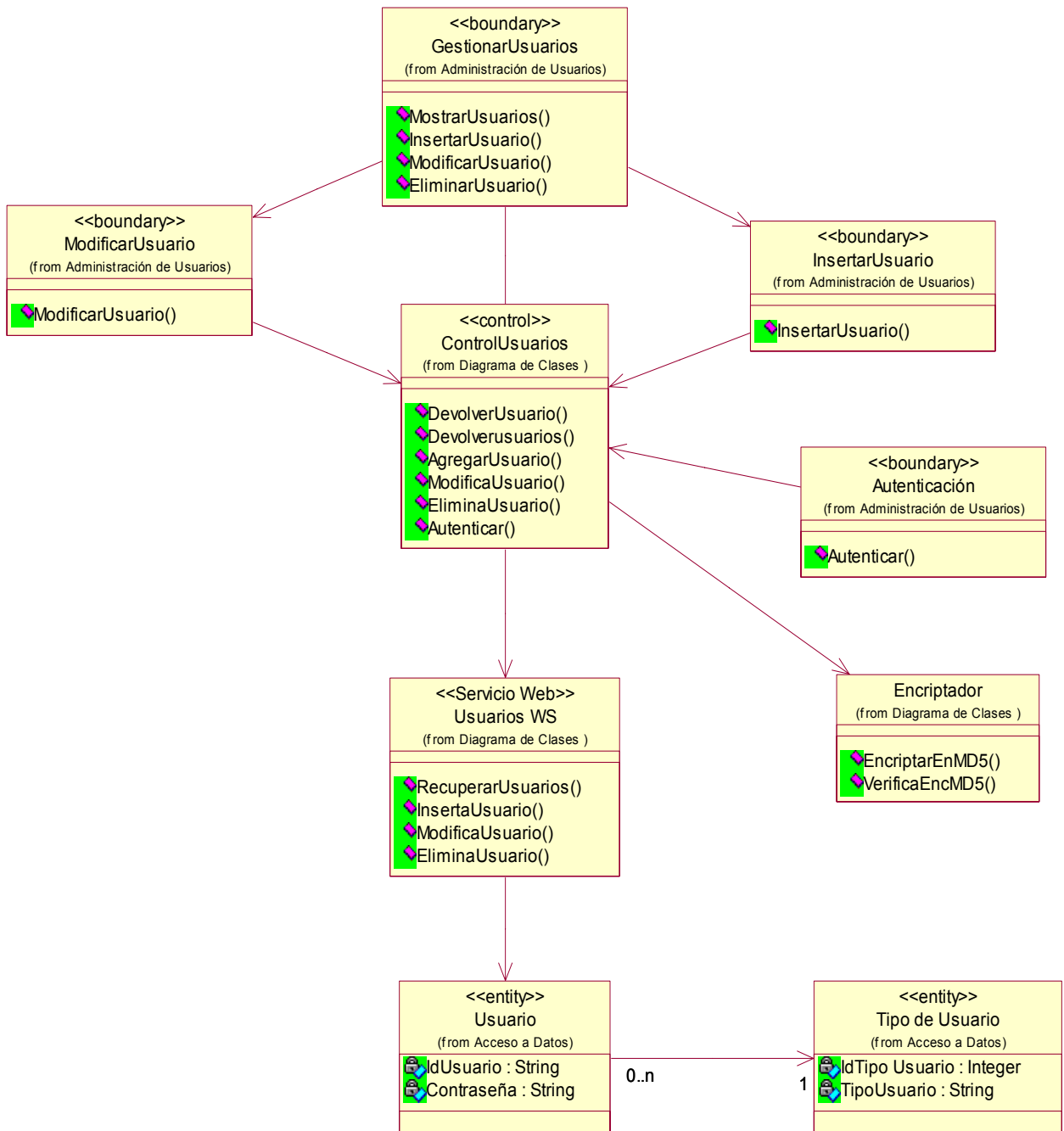


Figura 11 Diagrama de clases del paquete usuarios

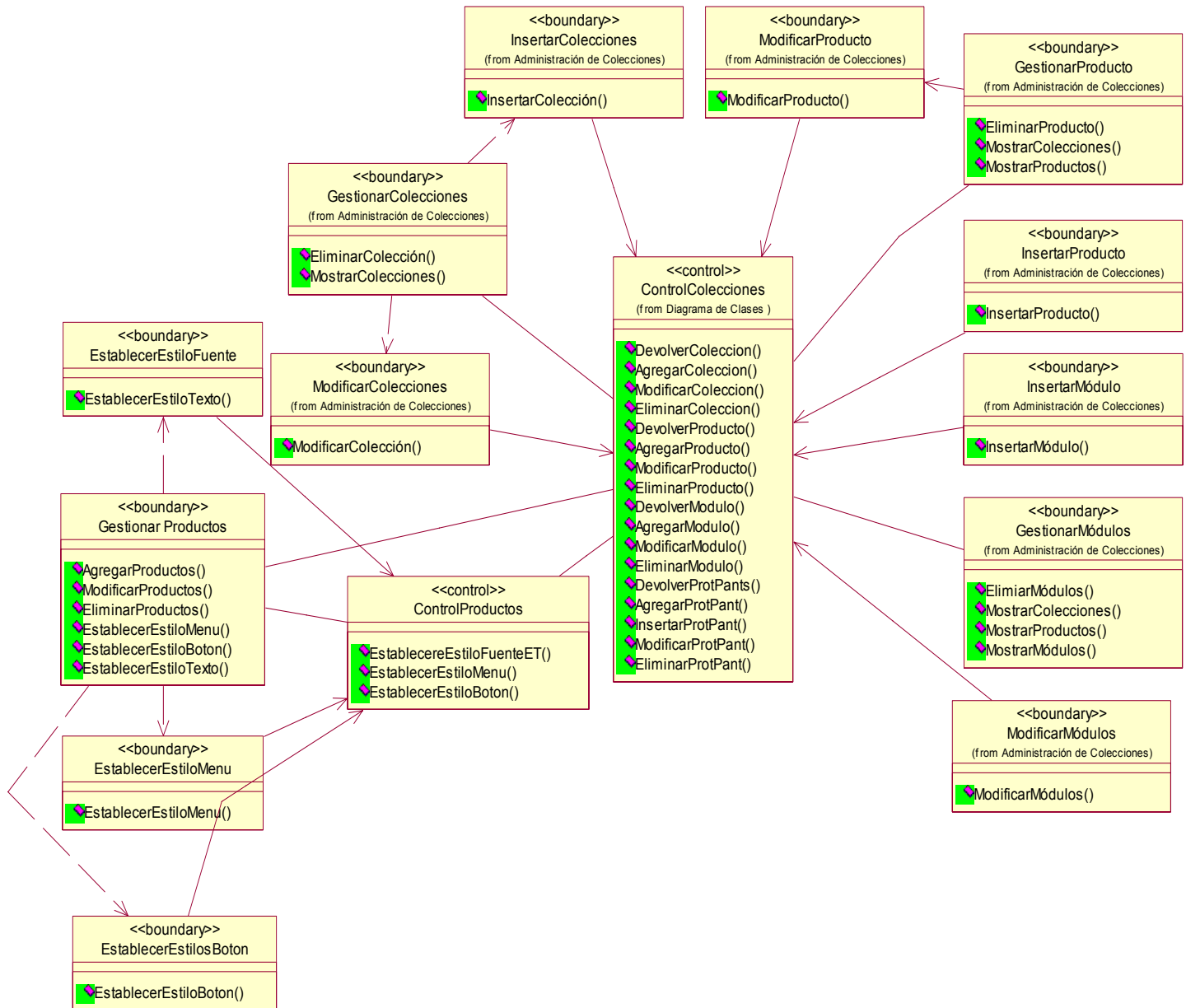


Figura 12 Diagrama de clases del paquete colecciones (sección1)

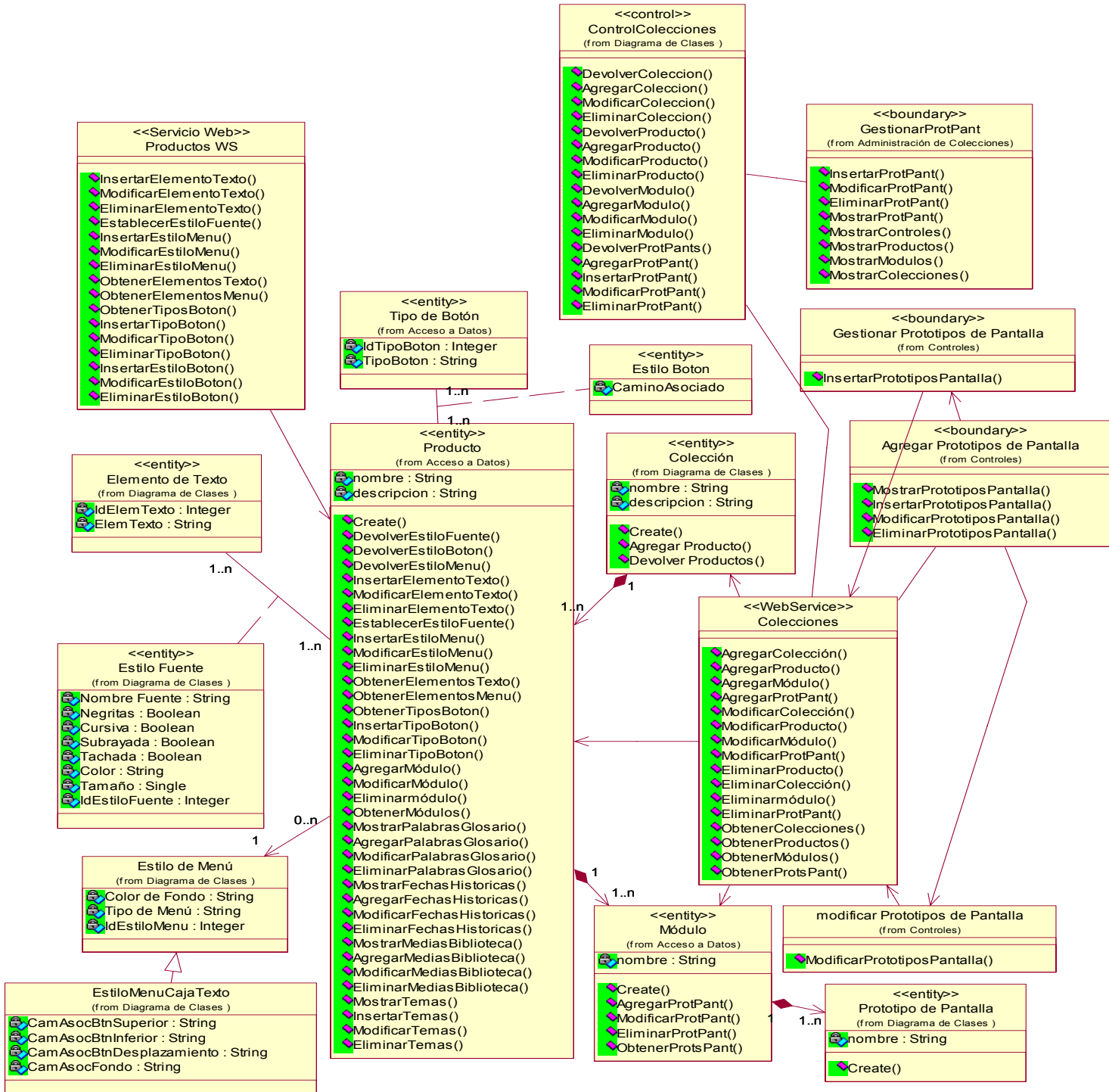


Figura 13 Diagrama de clases del paquete colecciones (sección2)

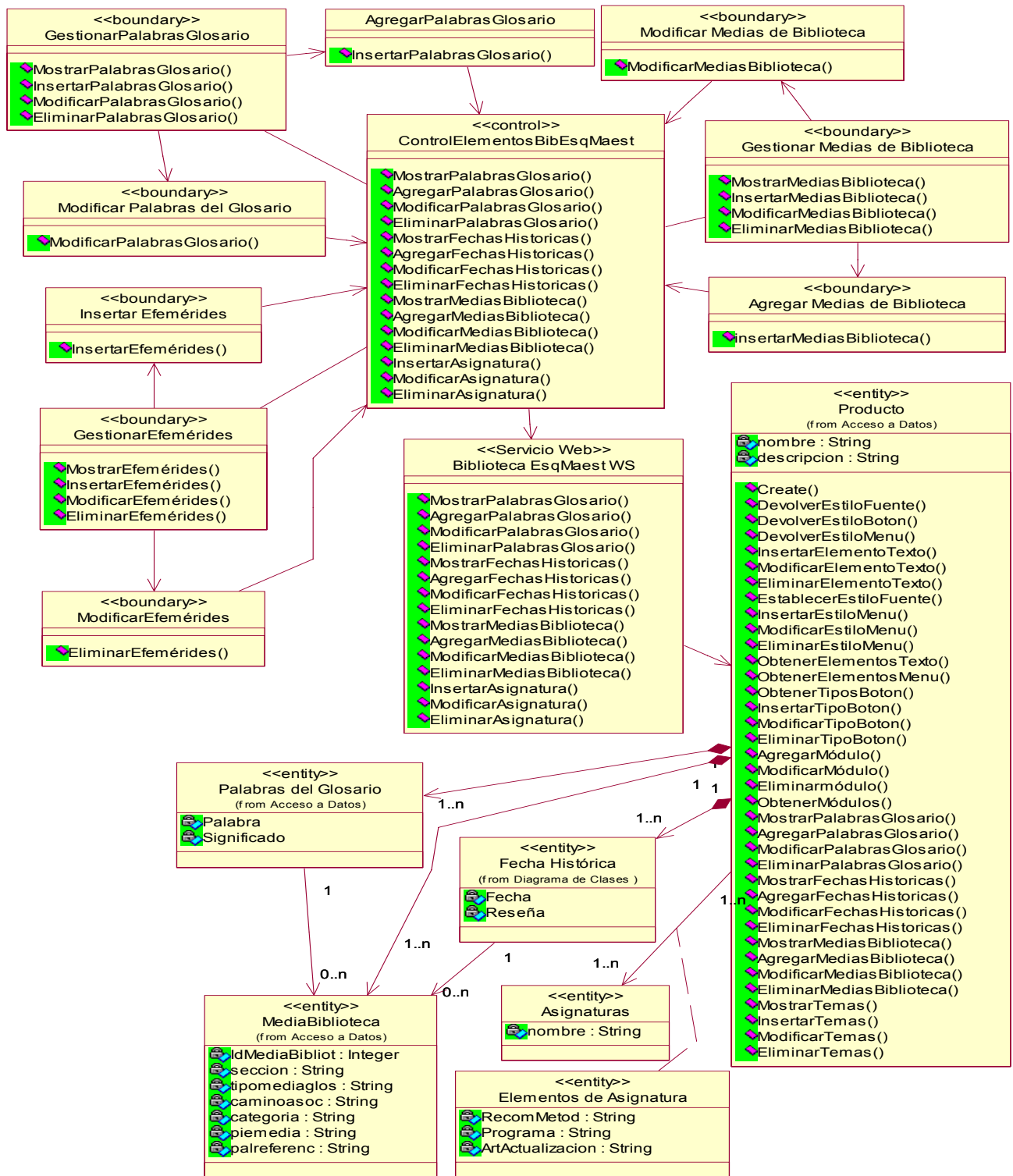


Figura 14 Diagrama de clases del paquete Biblioteca y EsqMaestro

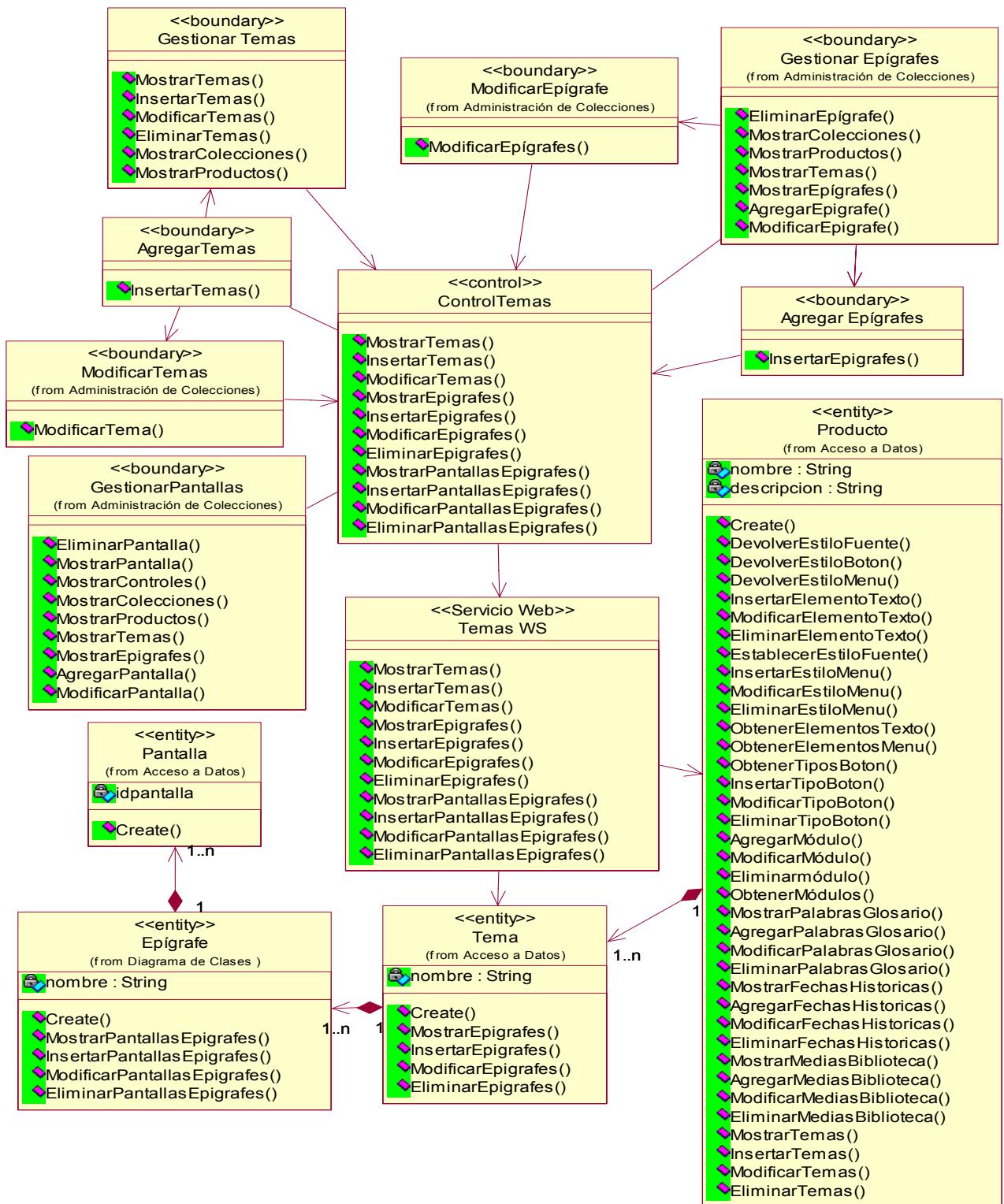


Figura 15 Diagrama de clases del paquete Temas

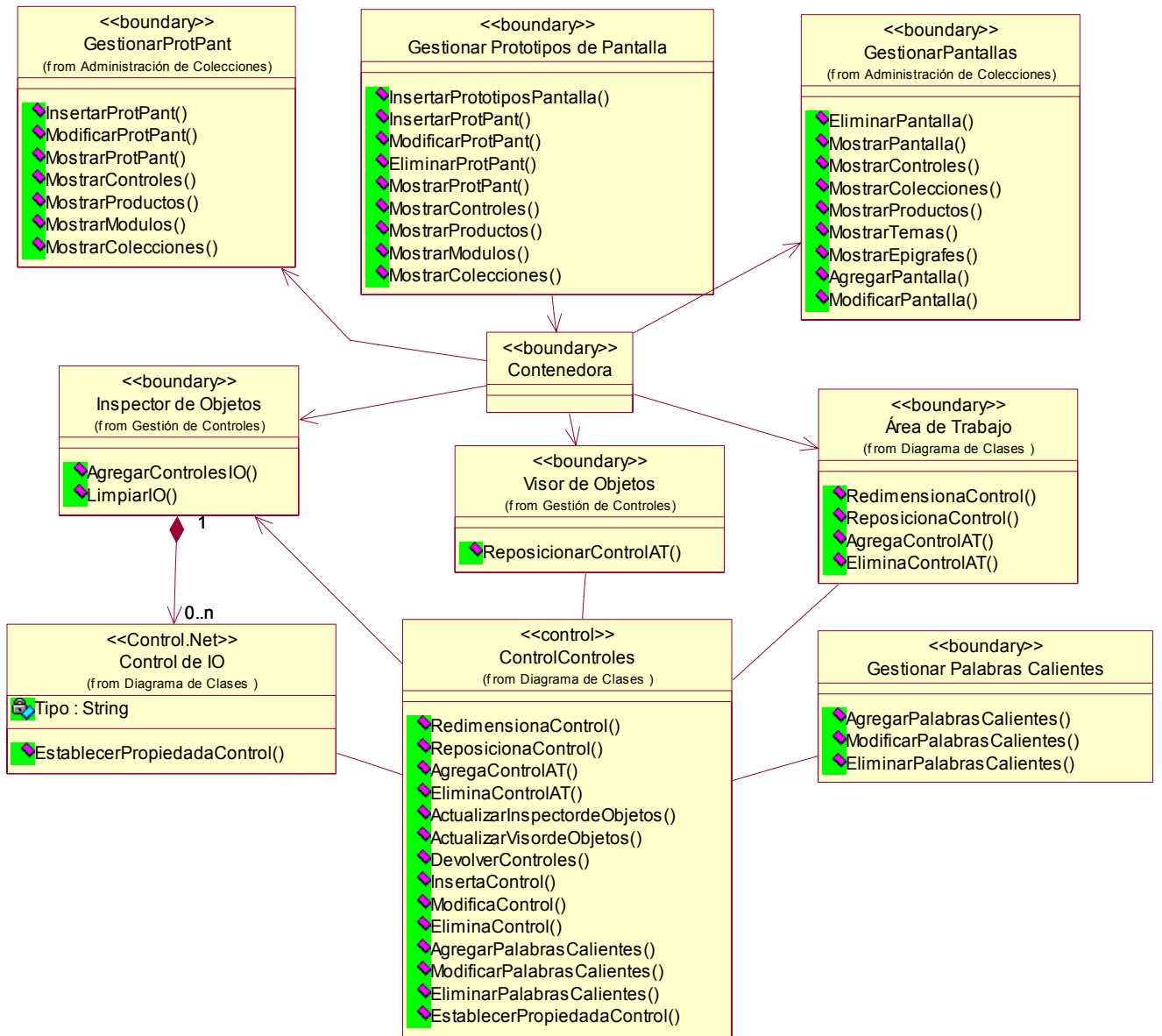


Figura 16 Diagrama de clases del paquete Controles (Sección1)

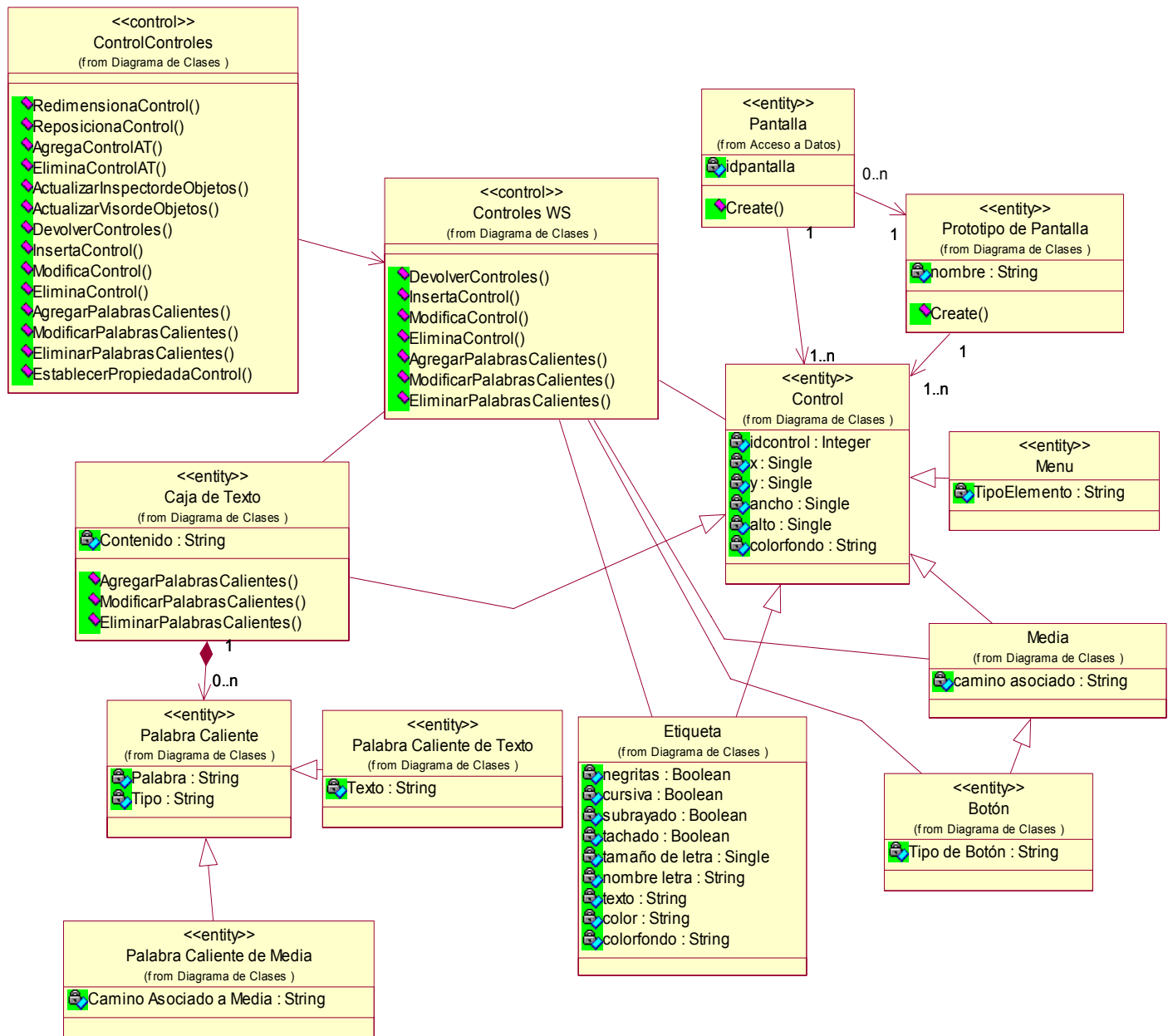


Figura 16 Diagrama de clases del paquete Controles (Sección2)

3.3 Diseño de la base de datos.

3.3.1 Diagrama de clases persistentes.

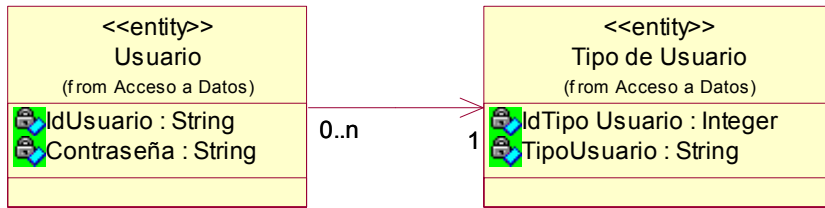


Figura 17 Diagrama de Clases Persistentes (Paquete Usuarios)

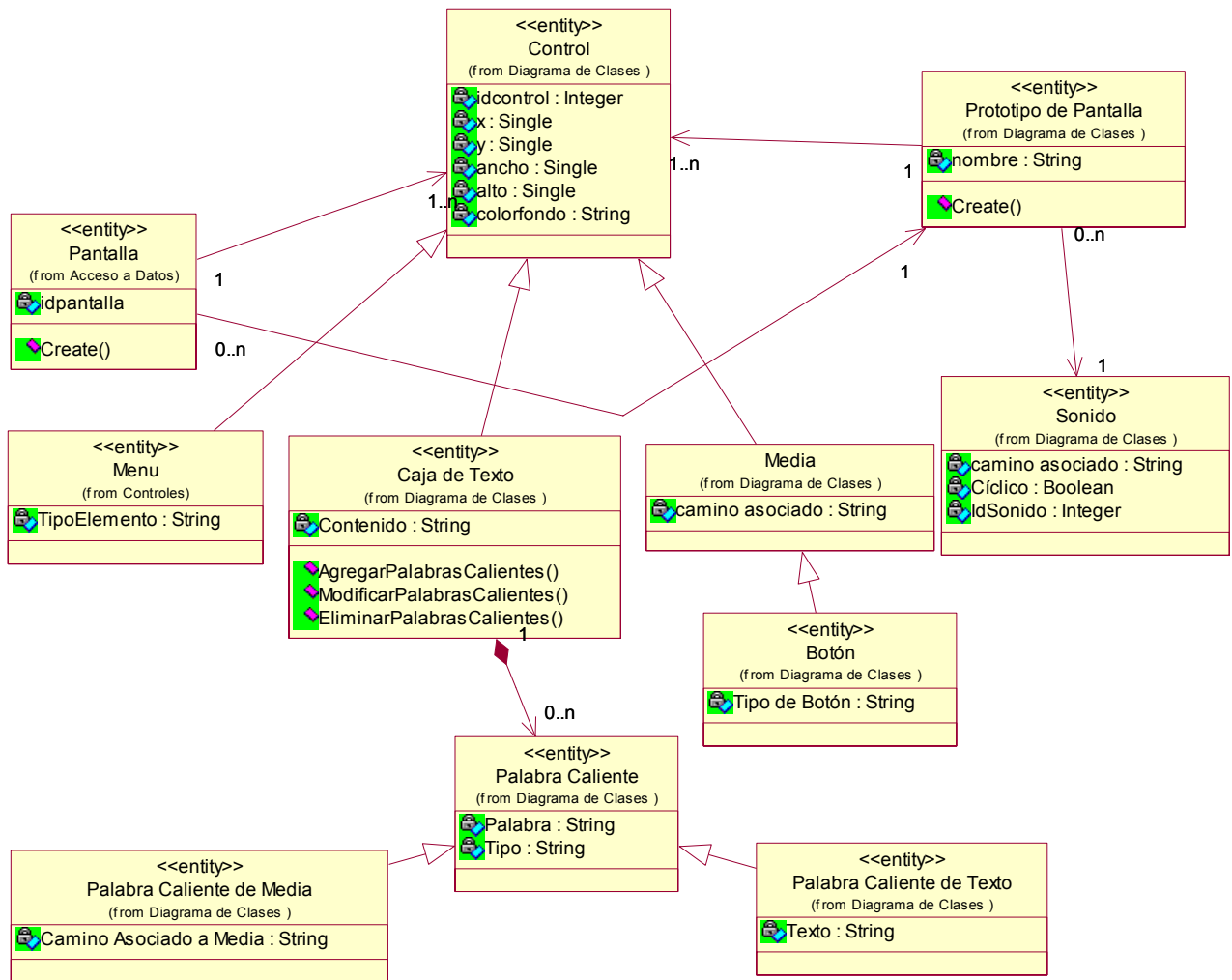


Figura 18 Diagrama de Clases Persistentes (Paquete Controles)

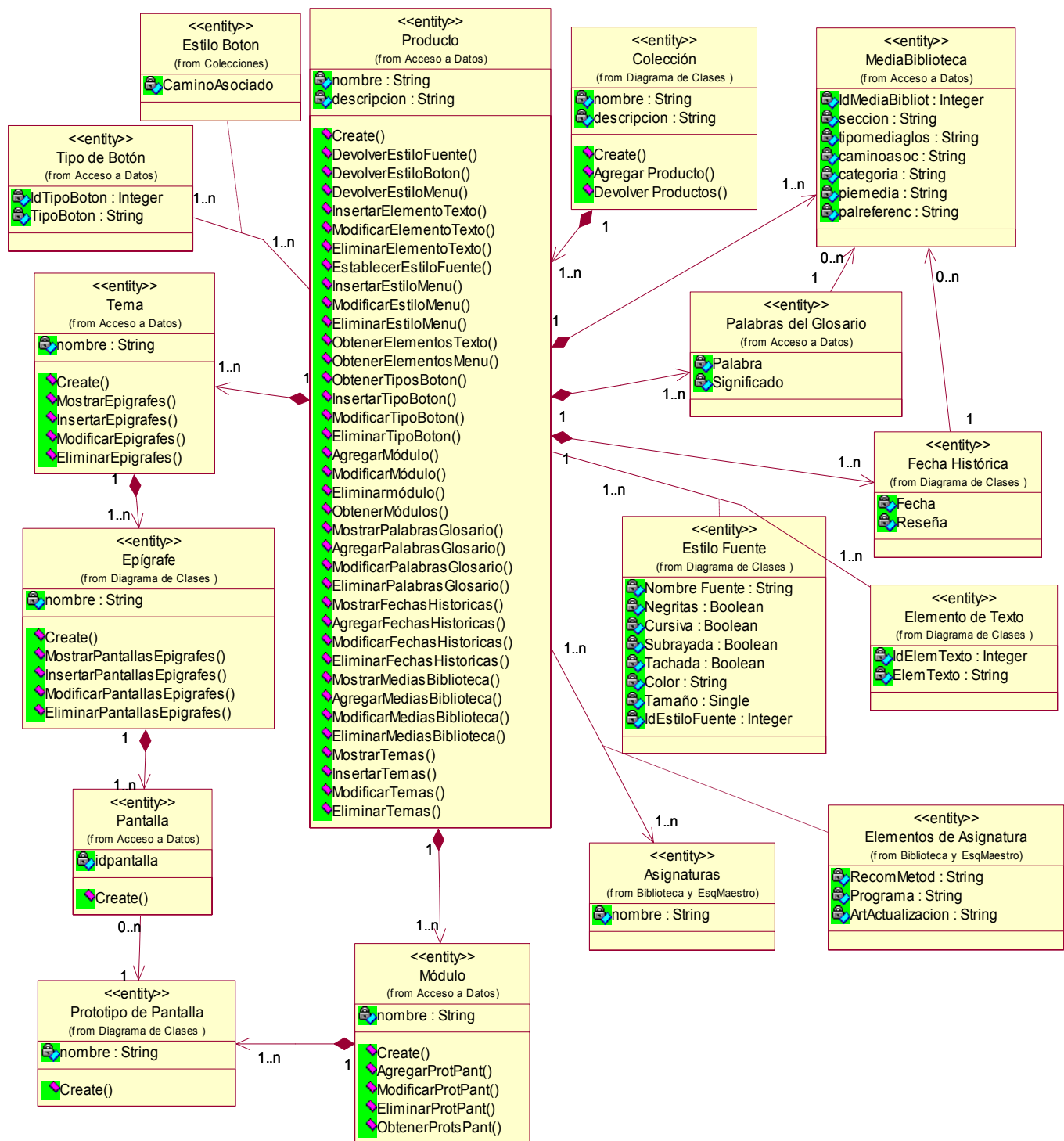


Figura 19 Diagrama de Clases Persistentes (Paquete Colecciones, Biblioteca, EsqMaest y Temas)

3.3.2 Modelo de datos

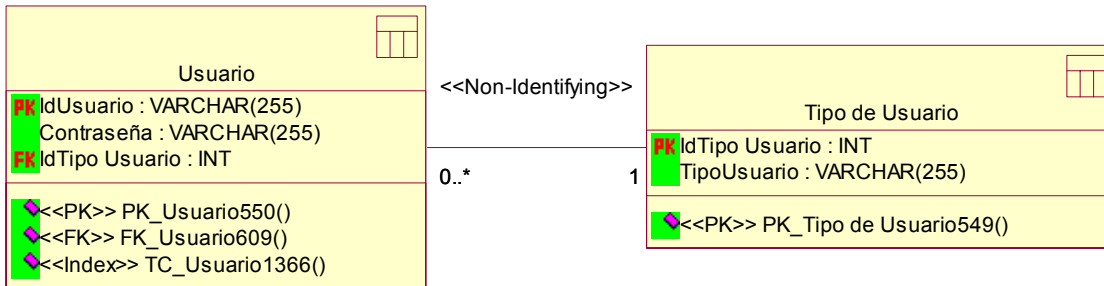


Figura 20 modelo de Datos (Paquete Usuarios)

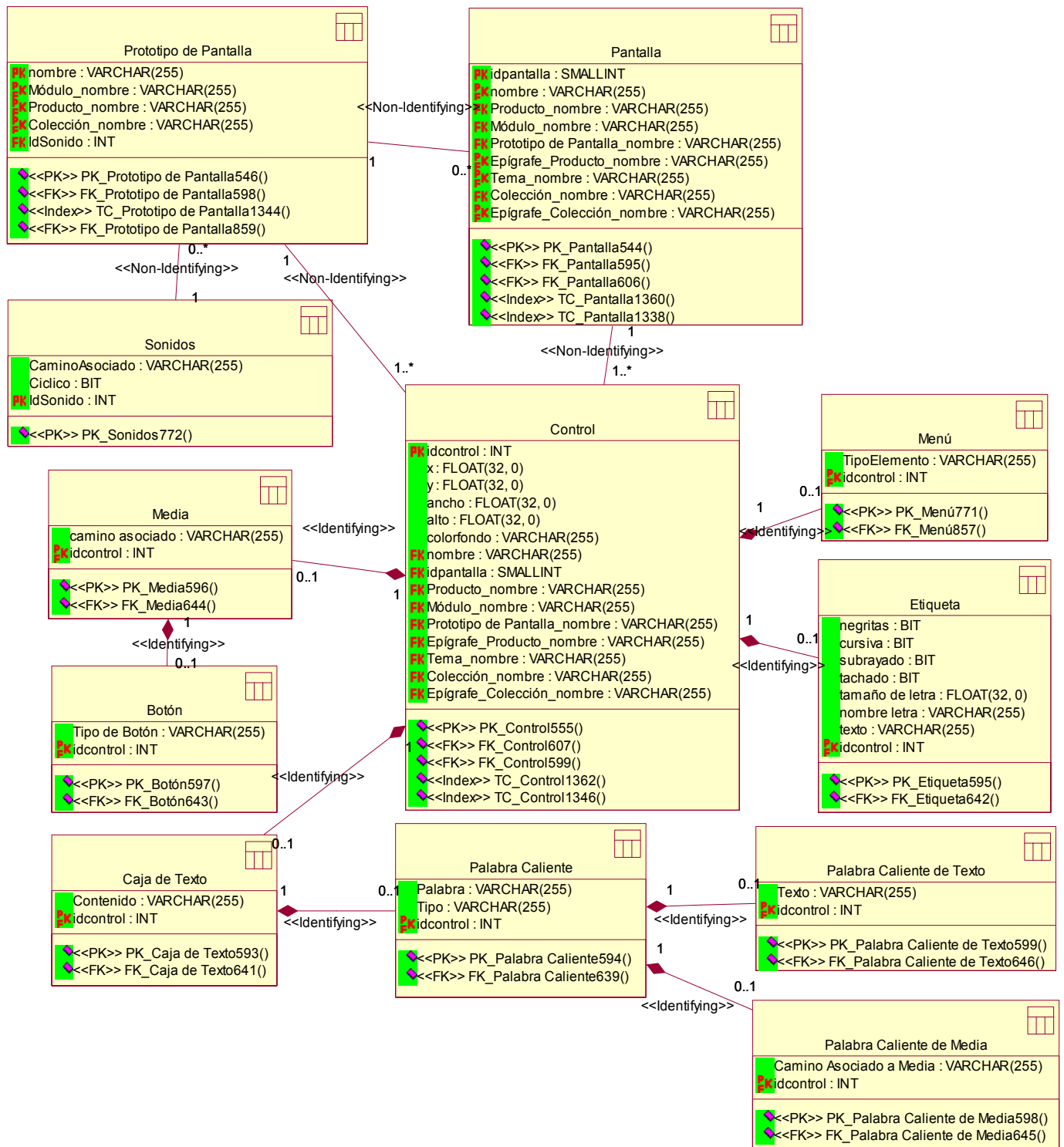


Figura 21 Modelo de datos (Paquete Control)

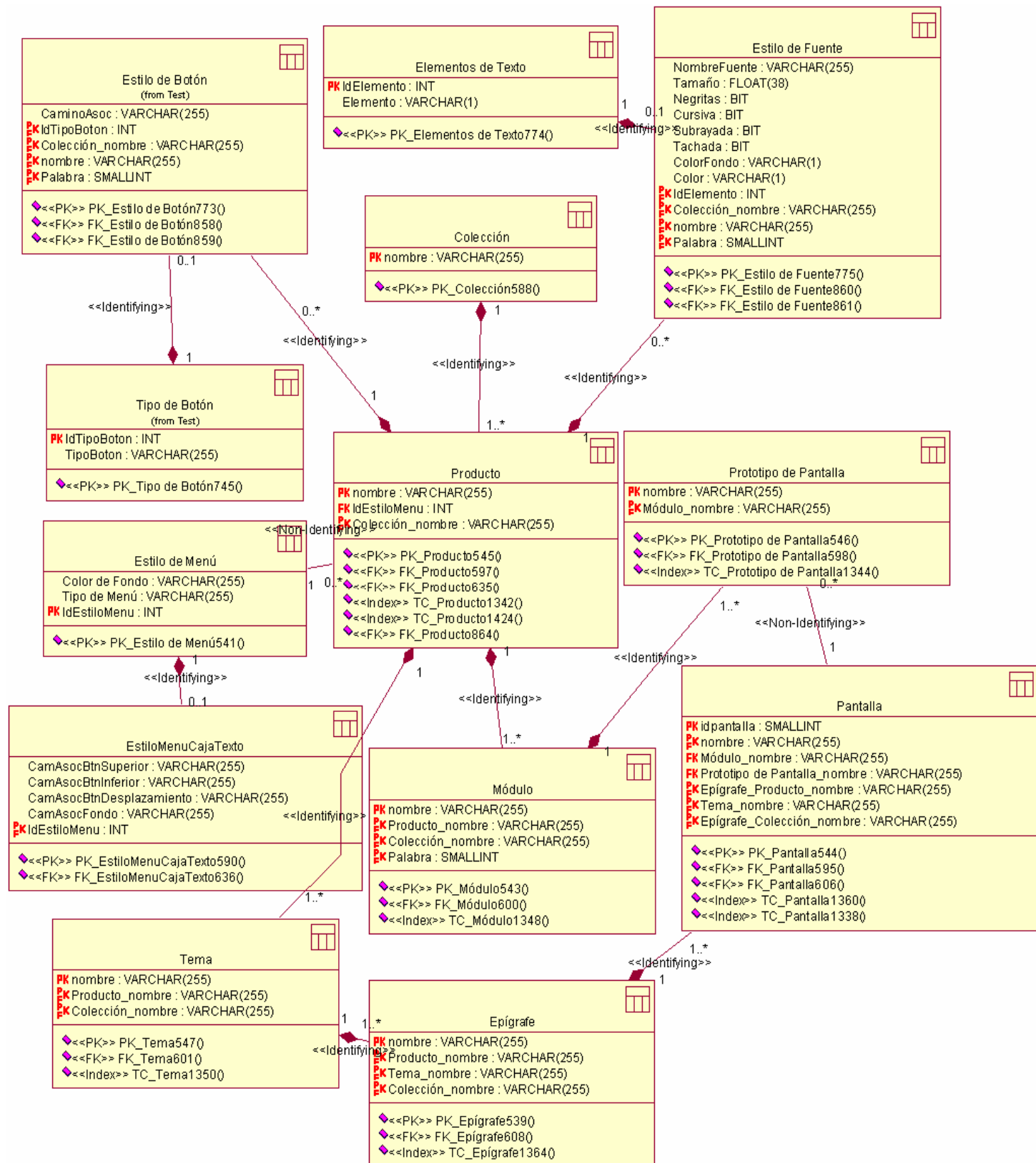


Figura 22 Modelo de datos (Paquetes Colecciones y Temas)

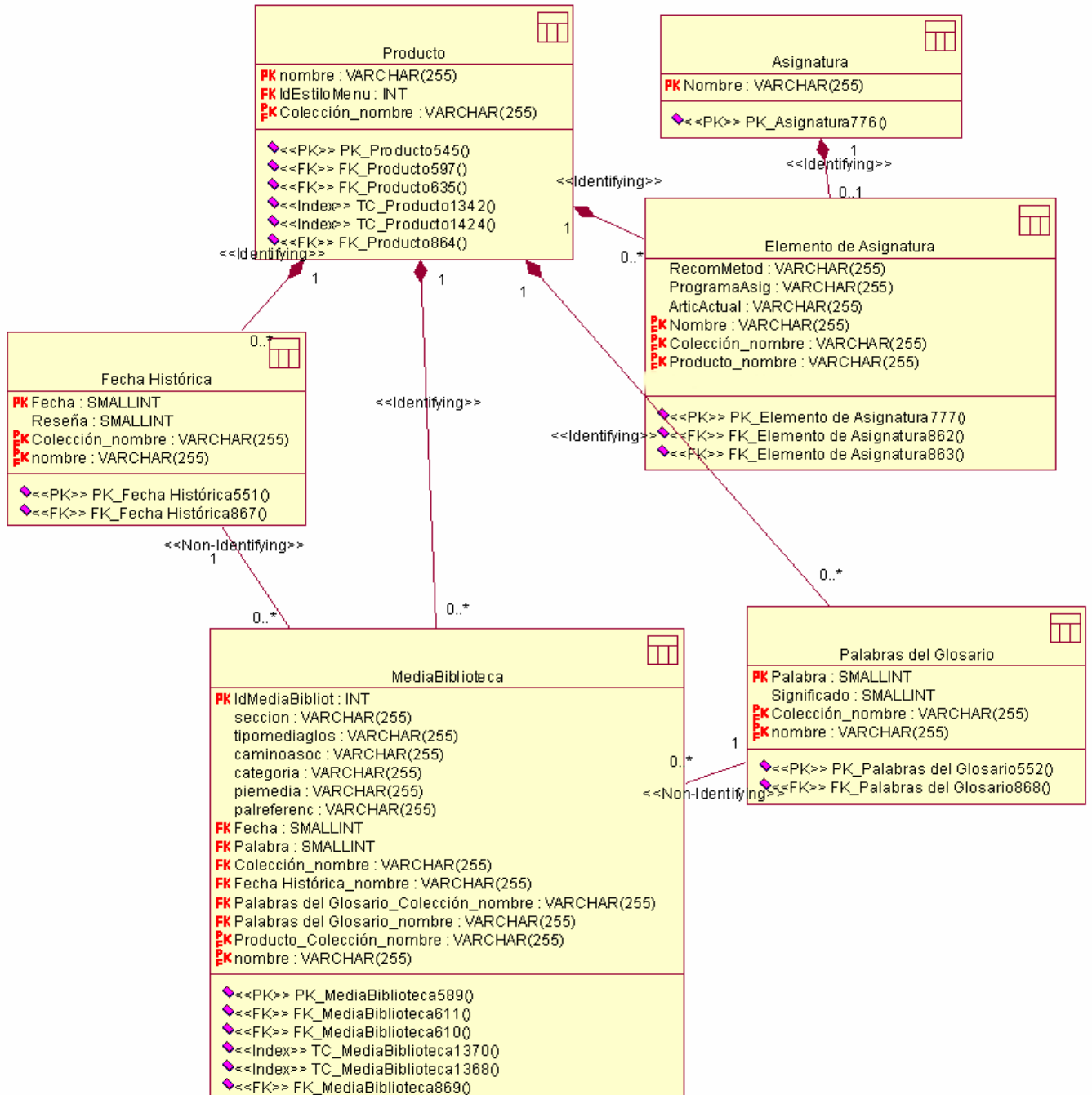


Figura 23 Modelo de datos (Paquete Biblioteca)

3.4 Principios de diseño

3.4.1 Estándares en la interfaz de la aplicación

Para garantizar un vínculo entre el usuario y el sistema realizado se ha hecho un esfuerzo por garantizar la máxima claridad posible en la interfaz procurando que sea intuitiva y que garantice un empleo óptimo del sistema y una comodidad en el trabajo de los usuarios finales. Las ventanas aparecen localizadas en el centro del escritorio.

Cada ventana está representada por un título acorde a su funcionalidad. Todos los botones están adornados con iconos que sirven de referencia a la función de los mismos. A continuación se presentan algunos de estos íconos:

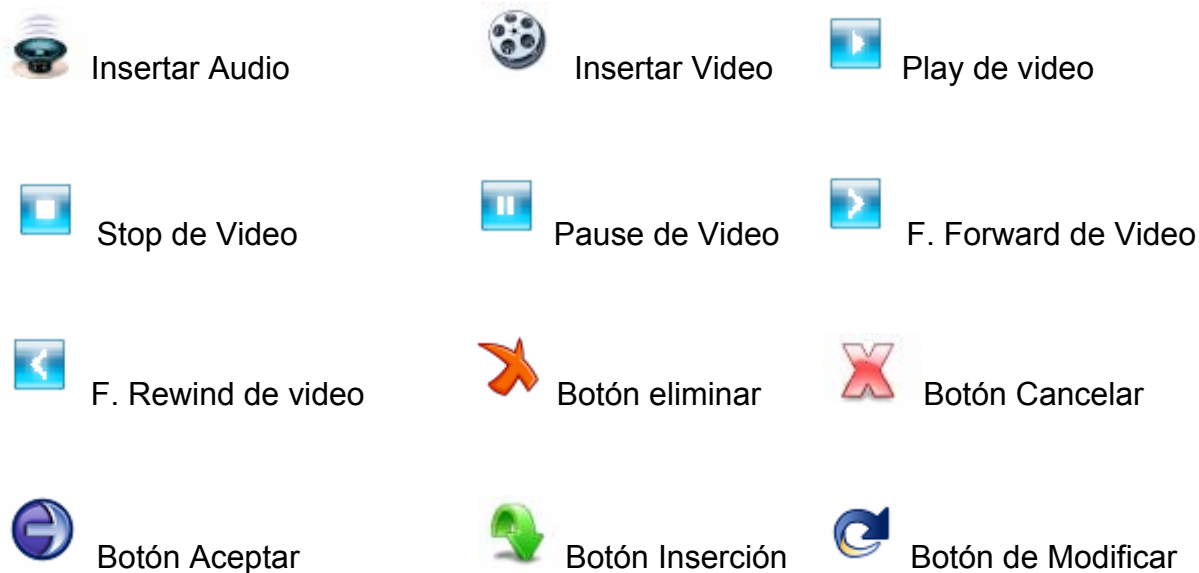


Figura 24 Iconos de la interfaz

3.4.3 Tratamiento de excepciones.

Para garantizar la consistencia del sistema se realiza un control exhaustivo de los errores. En el sistema casi toda la información se manipula de forma dinámica por lo que resulta indispensable llevar un estricto control sobre los posibles errores vinculados a la gestión de los datos con la base de datos. Se controlan errores de inserción de valores de llave duplicados. El gestor de base de datos (SQL Server 2000) garantiza la integridad referencial.

El sistema captura los errores emitidos y traduce el error a los usuarios, para esto se utilizan las librerías de clases de .Net (*System. Exception*).

3.4.3 Concepción general de la ayuda.

El sistema consta de una ayuda que permite a los usuarios del mismo conocer aspectos de su funcionamiento y donde se indicarán además posibles soluciones a problemas frecuentes con los que se pueden tropezar los mismos. A la misma se accederá a través del menú principal del sistema y desde cualquier otra ventana.

En la ayuda habrá una sección de consejos que tratarán de orientar al usuario en lo relativo al diseño de las colecciones, cómo deben quedar confeccionadas según los estándares dispuestos por los diseñadores y otros aspectos a tener en cuenta señalados en el guión.

3.5 Estándares de codificación

Para obtener una mayor claridad en el código y lograr así que el mantenimiento del sistema resulte más fácil se ha seguido un estándar en la codificación. A continuación se describen algunos de los aspectos considerados:

1- Al poner nombre a un elemento el primer carácter del nombre es un carácter alfabético, así la declaración de las clases comienza con T_ y los atributos con F_. Las clases de interfaz comienzan con frm_.

- 2- Cada palabra independiente de un nombre comienza con una letra mayúscula, por ejemplo: FindLastRecord y RedrawMyForm.
- 3- Los nombres de método y de función comienzan con un verbo, por ejemplo: InitNameArray o CloseDialog.
- 4- Los nombres de controlador de eventos empiezan con un nombre que describe el tipo de evento seguido por el sufijo "EventHandler", como en "MouseEventHandler".
- 5- Para los eventos que tienen un concepto de "antes" o "después", se utiliza un prefijo en tiempo presente o pasado, como en "ControlAdd" o "ControlAdded".
- 6- Para términos largos o utilizados con frecuencia, se utilizan abreviaturas para mantener las longitudes de los nombres dentro un límite razonable, por ejemplo, "HTML" en lugar de "Lenguaje de marcado de hipertexto", asegurando que las abreviaturas sean coherentes a lo largo de toda la aplicación.
- 7- Se evita utilizar nombres que en un entorno interno sean iguales que otros nombres de un entorno externo.

3.6 Modelo de despliegue.

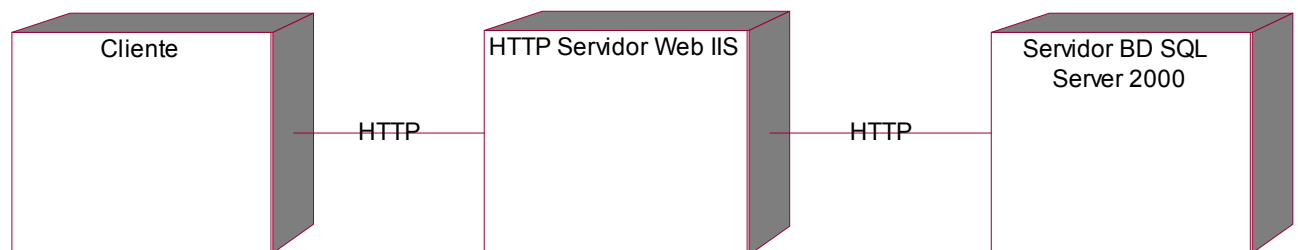


Figura 25 Diagrama de Despliegue

3.7 Modelo de implementación.

3.7.1 Diagramas de los componentes del sistema

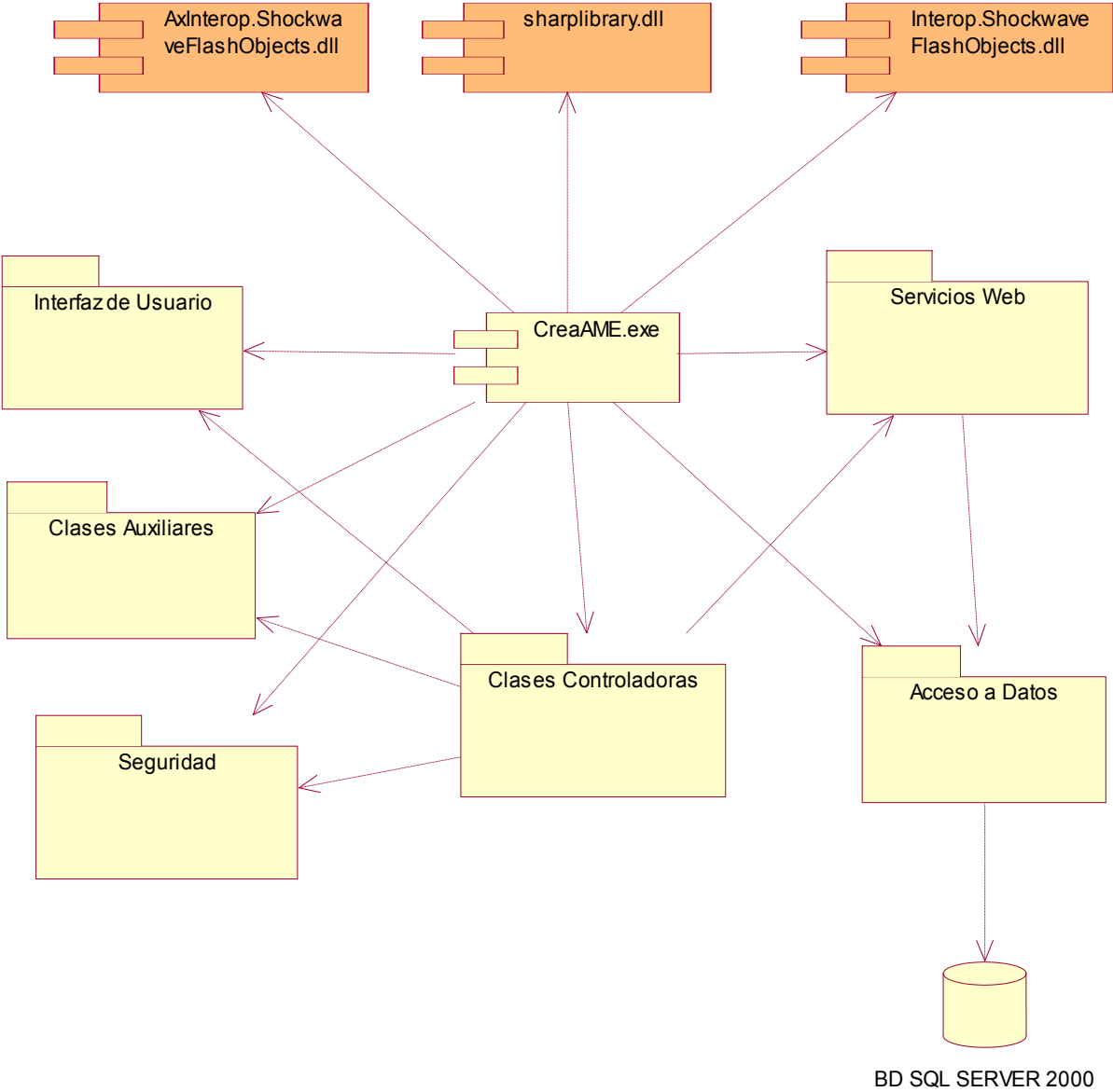


Figura 26 Diagrama de Componentes

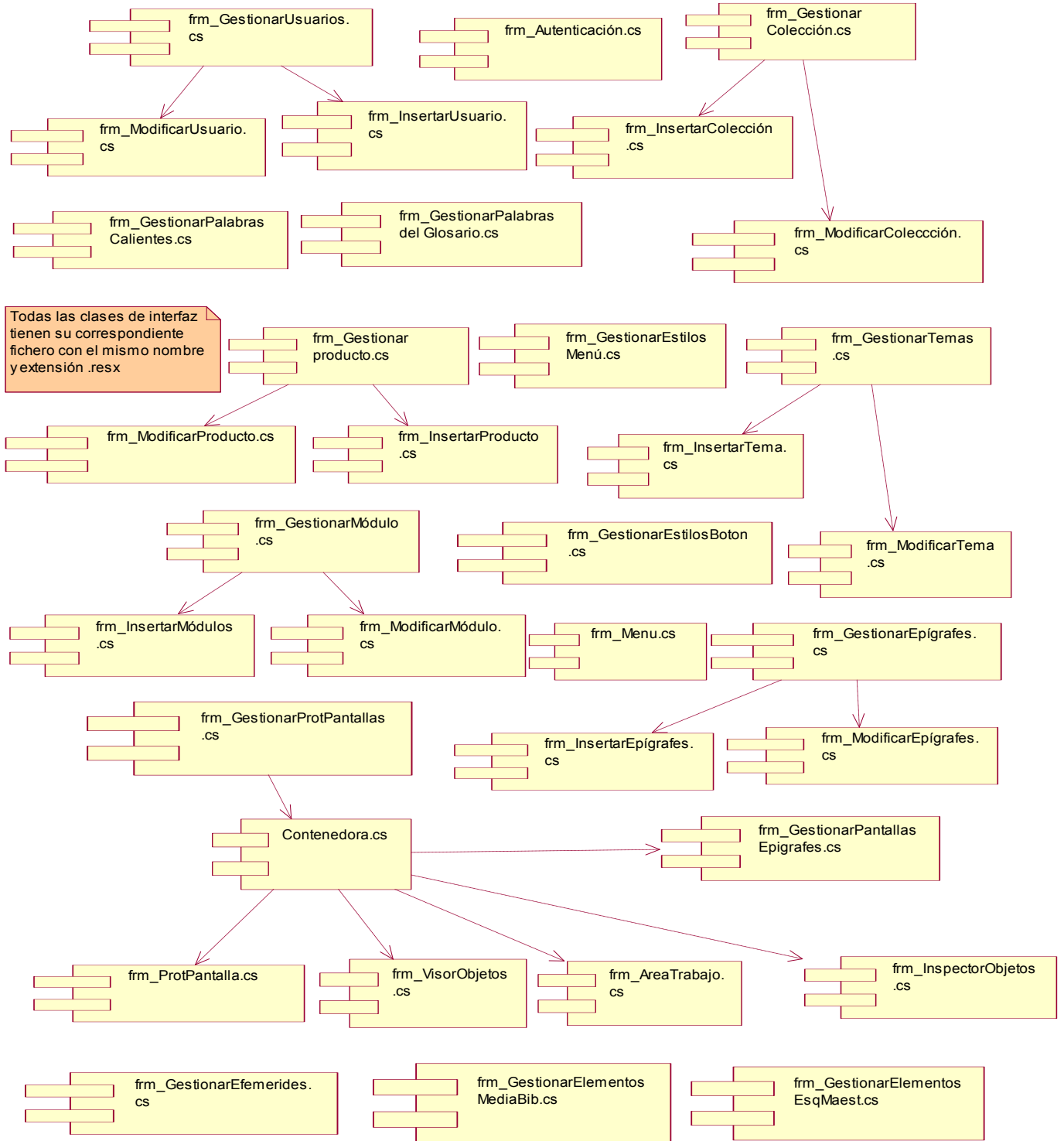


Figura 27 Diagrama de Componentes (Paquete Interfaz de Usuario)

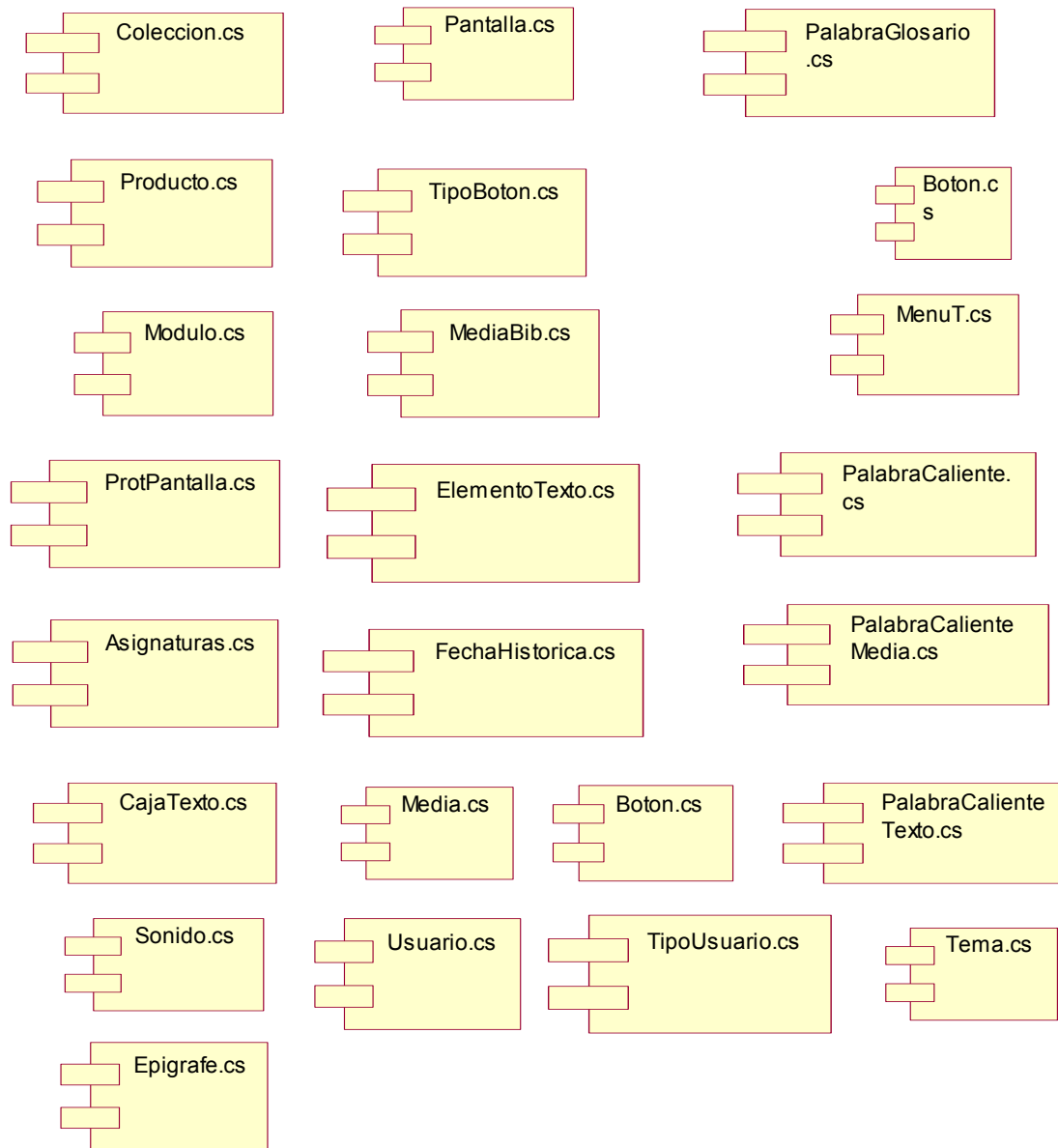


Figura 28 Diagrama de Componentes (Paquete Acceso a Datos)

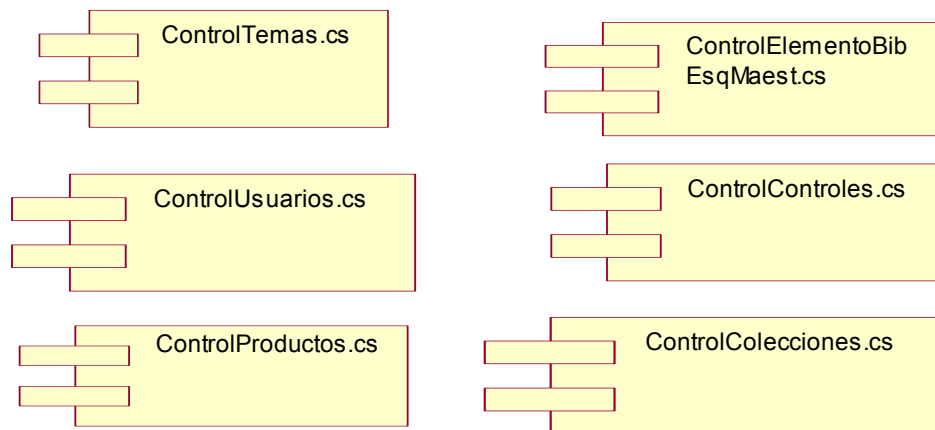


Figura 29 Diagrama de Componentes (Paquete Clases Controladoras)

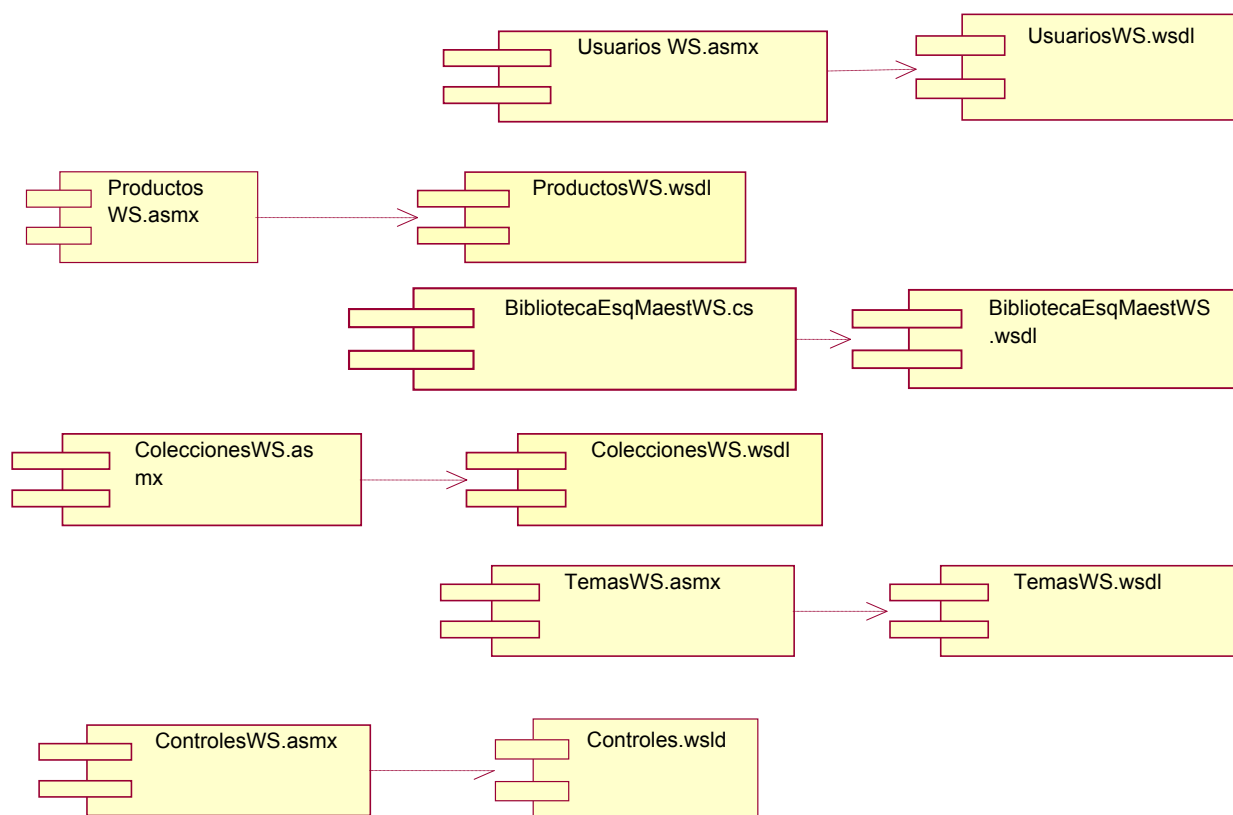


Figura 30 Diagrama de Componentes (Paquete Servicios Web)

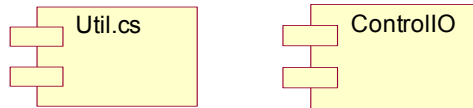


Figura 31 Diagrama de Componentes (Paquete Clases Auxiliares)

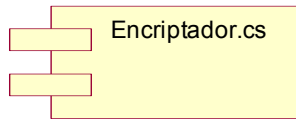


Figura 32 Diagrama de Componentes (Paquete Seguridad)

3.7.2 Descripción de los componentes

3.7.2.1 Acceso a Datos

Componente	Propósito	Contenido
Colección.cs	Permite la gestión de los datos de la tabla Colección en la Base de Datos.	T_Colección.cs (clase)
Producto.cs	Permite la gestión de los datos de la tabla Producto en la Base de Datos.	T_Producto (clase)
Modulo.cs	Permite la gestión de los datos de la tabla Modulo en la Base de Datos.	T_Modulo (clase)
ProtPantalla.cs	Permite la gestión de los datos de la tabla ProtPantalla en la Base de Datos.	T_ProtPantalla (clase)

Asignaturas.cs	Permite la gestión de los datos de la tabla Asignatura en la Base de Datos.	T_Asignaturas (clase)
CajaTexto.cs	Permite la gestión de los datos de la tabla CajaTexto en la Base de Datos.	T_CajaTexto (clase)
Sonido.cs	Permite la gestión de los datos de la tabla Sonido en la Base de Datos.	T_Sonido (clase)
Epigrafe.cs	Permite la gestión de los datos de la tabla Epigrafe en la Base de Datos.	T_Epigrafe (clase)
Pantalla.cs	Permite la gestión de los datos de la tabla Pantalla en la Base de Datos.	T_Pantalla (clase)
TipoBoton.cs	Permite la gestión de los datos de la tabla TipoBoton en la Base de Datos.	T_TipoBoton (clase)
MediaBib.cs	Permite la gestión de los datos de la tabla MediaBib en la Base de Datos.	T_MediaBib (clase)
ElementoTexto.cs	Permite la gestión de los datos de la tabla ElementoTexto en la	T_ElementoTexto (clase)

	Base de Datos.	
FechaHistorica.cs	Permite la gestión de los datos de la tabla FechaHistorica en la Base de Datos.	T_FechaHistorica (clase)
Media.cs	Permite la gestión de los datos de la tabla Media en la Base de Datos.	T_Media (clase)
Usuario.cs	Permite la gestión de los datos de la tabla Usuario en la Base de Datos.	T_Usuario (clase)
TipoUsuario	Permite la gestión de los datos de la tabla TipoUsuario en la Base de Datos.	T_TipoUsuario (clase)
Boton.cs	Permite la gestión de los datos de la tabla Boton en la Base de Datos.	T_Boton (clase)
Menu.cs	Permite la gestión de los datos de la tabla Menu en la Base de Datos.	T_Menu (clase)
PalabraCaliente.cs	Permite la gestión de los datos de la tabla PalabraCaliente en la Base de Datos.	T_PalabraCaliente (clase)
PalabraCalientetMedia.cs	Permite la gestión de los datos	T_PalabraCalienteMedia

	de la tabla PalabraCalienteMedia en la Base de Datos.	(clase)
PalabraCalienteTexto.cs	Permite la gestión de los datos de la tabla PalabraCalienteTexto en la Base de Datos.	T_PalabraCalienteTexto (clase)
Tema.cs	Permite la gestión de los datos de la tabla Tema en la Base de Datos.	T_Tema (clase)

Tabla1 Descripción de los componentes del paquete de acceso a datos

3.7.2.2 Interfaz de Usuario

Componente	Propósito	Contenido
frm_GestionarUsuarios.cs	Mostrar los usuarios del sistema y permitir el acceso a los formularios de insertar y modificar usuarios.	frm_GestionarUsuarios (clase)
frm_InsertarUsuario.cs	Agrupar a todos los controles visuales que permiten agregar usuarios.	frm_InsertarUsuario (clase)
frm_ModificarUsuario.cs	Agrupar a todos los controles visuales que permiten modificar usuarios.	frm_ModificarUsuarios (clase)

frm_Autenticacion.cs	Agrupar a todos los controles visuales que permiten la autenticación del usuario en el sistema.	frm_Autenticacion (clase)
frm_GestionarColeccion.cs	Mostrar las colecciones existentes y permitir el acceso a los formularios de insertar y modificar colección	frm_GestionarColeccion (clase)
frm_InsertarColeccion.cs	Agrupar a todos los controles visuales que permiten agregar colecciones.	frm_InsertarColeccion (clase)
frm_ModificarColeccion.cs	Agrupar a todos los controles visuales permiten modificar colecciones.	frm_ModificarColeccion (clase)
frm_GestionarPalabrasCalientes.cs	Agrupar a todos los controles visuales que permiten la gestión de palabras calientes.	frm_GestionarPalabrasCalientes (clase)
frm_GestionarPalabrasGlosario.cs	Agrupar a todos los controles visuales que permiten la gestión de palabras del glosario.	frm_GestionarPalabrasGlosario (clase)
frm_GestionarProducto.cs	Mostrar los productos existentes y permitir el acceso a los formularios de insertar y modificar producto	frm_GestionarProducto (clase)
frm_InsertarProducto.cs	Agrupar a todos los controles	frm_InsertarProducto

	visuales que permiten agregar productos.	(clase)
frm_ModificarProducto.cs	Agrupar a todos los controles visuales que permiten modificar productos.	frm_ModificarProducto (clase)
frm_GestionarEstilosMenu.cs	Agrupar a todos los controles visuales que permiten la gestión de estilos de menú.	frm_GestionarEstilosMenu (clase)
frm_GestionarTemas.cs	Mostrar los temas existentes y permitir el acceso a los formularios de insertar y modificar tema	frm_GestionarTemas (clase)
frm_InsertarTema.cs	Agrupar a todos los controles visuales que permiten agregar temas.	frm_InsertarTema (clase)
frm_ModificarTema.cs	Agrupar a todos los controles visuales que permiten modificar temas.	frm_ModificarTema (clase)
frm_GestionarModulos.cs	Mostrar los módulos existentes y permitir el acceso a los formularios de insertar y modificar módulo	frm_GestionarModulos (clase)
frm_InsertarModulo.cs	Agrupar a todos los controles visuales que permiten agregar	frm_InsertarModulo (clase)

	módulos.	
frm_ModificarModulo.cs	Agrupar a todos los controles visuales que permiten modificar módulos.	frm_ModificarModulo (clase)
frm_GestionarEpigrafes.cs	Mostrar los epígrafes existentes y permitir el acceso a los formularios de insertar y modificar epígrafes	frm_GestionarEpigrafes (clase)
frm_InsertarEpigrafe.cs	Agrupar a todos los controles visuales que permiten agregar epígrafes.	frm_InsertarEpigrafe (clase)
frm_ModificarEpigrafe.cs	Agrupar a todos los controles visuales que permiten modificar epígrafes.	frm_ModificarEpigrafe (clase)
frm_GestionarProtPantalla.cs	Mostrar los prototipos de pantalla existentes y permitir el acceso a los formularios de insertar y modificar prototipos de pantalla	frm_GestionarProtsPantall (clase)
frm_Menu.cs	Constituye la ventana desde la cuál se accederá a cada una de las pantallas de gestión.	frm_Menu (clase)
frm_Contenedora.cs	Constituye una ventana MDI que contiene varios <i>windowsforms</i> como el inspector	frm_Contenedora (clase)

	de objetos, el visor de objetos, etc. para la gestión de las pantallas y prototipos de pantallas así como de los controles correspondientes.	
frm_GestionarPantallasEpi grafes.cs	Es uno de los formularios contenidos en frm_Contenedora. Agrupa los controles visuales para la gestión de las pantallas.	frm_GestionarPantallas Epigrafes (clase)
frm_ProtPantalla.cs	Es uno de los formularios contenidos en frm_Contenedora. Agrupa los controles visuales para la gestión de los prototipos de pantalla.	frm_ProtPantalla
frm_VisorObjetos.cs	Es uno de los formularios contenidos en frm_Contenedora. Muestra los controles agregados al Área de Trabajo (frm_AreaTrabajo) y permite establecer la posición de los mismos en profundidad.	frm_VisorObjetos (clase)
frm_AreaTrabajo.cs	Es uno de los formularios contenidos en frm_Contenedora. Representa el área donde se colocan los controles de una pantalla o prototipo de pantalla.	frm_AreaTrabajo (Clase)
frm_InspectorObjetos.cs	Es uno de los formularios contenidos en	frm_InspectorObjetos (clase)

	frm_Contenedora. Representa un Inspector de Objetos que contiene controles para visualizar y establecer las propiedades de los controles colocados en el Área de Trabajo.	
frm_GestionarEfemerides.cs	Mostrar las efemérides y agrupar controles que permitan la gestión de las mismas.	frm_GestionarEfemerides (clase)
frm_GestionarElementosMediaBib.cs	Mostrar los elementos de media de la biblioteca y agrupar controles que permitan la gestión de los mismos.	frm_GestionarElementosMediaBib (clase)
frm_GestionarElementosEsqMaest.cs	Mostrar los elementos de la esquina del maestro (asignaturas, artículos de actualización, etc. y agrupar controles que permitan la gestión de los mismos.	frm_GestionarElementosEsqMaest (clase)

Tabla 2 Descripción de los componentes del paquete de interfaz de usuario

3.7.2.3 Clases Controladoras

Componente	Propósito	Contenido
ControlColecciones.cs	Manipula la información referente a las colecciones, productos, módulos y prototipos	T_ControlColecciones (clase)

	de pantalla y media entre la interfaz y el servicio web ColeccionesWS	
ControlProductos.cs	Manipula la información referente a los productos y media entre la interfaz y el servicio web ProductosWS.	T_ControlProductos (clase)
ControlTemas.cs	Manipula la información referente a los temas, epígrafes y pantallas y media entre la interfaz y el servicio web TemasWS.	T_ControlTemas (clase)
ControlElementosBibEsqMaest.cs	Manipula la información referente a los elementos de la biblioteca: Palabras de glosario, efemérides, etc. y media entre la interfaz y el servicio web BibliotecaEsqMaest WS.	T_ControlElementosBibEsqMaest (clase)
ControlControles.cs	Manipula la información referente a los controles que se colocan en el Área de Trabajo y que pertenecen a un prototipo de pantalla o pantalla y media entre la interfaz y el servicio web ControlesWS. Se encarga además restablecer las propiedades de los controles y	T_ControlControles (Clase)

	actualizarlos en el área de trabajo, así como de actualizar el Inspector de Objetos y el Visor de Objetos.	
ControlUsuarios.cs	Manipula la información referente a los usuarios y media entre la interfaz y el servicio web UsuariosWS. Se encarga de la autenticación del usuario.	T_ControlUsuarios (clase)

Tabla 3 Descripción de los componentes del paquete de clases controladoras

3.7.2.4 Seguridad

Componente	Propósito	Contenido
Encriptador.cs	Encrypta la contraseña de los usuarios y luego la verifica.	T_Encryptador (clase)

Tabla 3 Descripción de los componentes del paquete de seguridad

3.7.2.5 Servicios Web

Componente	Propósito	Contenido
ColeccionesWS.cs	Contiene y ejecuta las sentencias de SQL relacionadas con la recuperación, inserción, modificación y eliminación de colecciones, productos, módulos y prototipos de pantalla.	ColeccionesWS (servicio web)

ProductosWS.cs	Contiene y ejecuta las sentencias de SQL relacionadas con la recuperación, inserción, modificación y eliminación de los elementos de un producto: Estilos de Menú, Estilos de Botón, etc.	ProductosWS (servicio web)
TemasWS.cs	Contiene y ejecuta las sentencias de SQL relacionadas con la recuperación, inserción, modificación y eliminación de los elementos de temas, epígrafes y pantallas.	TemasWS (servicio web)
BibliotecaEsqMaestWS.cs	Contiene y ejecuta las sentencias de SQL relacionadas con la recuperación, inserción, modificación y eliminación de los elementos de biblioteca.	BibliotecaEsqMaestWS (servicio web)
ControlesWS.cs	Contiene y ejecuta las sentencias de SQL relacionadas con la recuperación, inserción, modificación y eliminación de los controles	ControlesWS (servicio web)
UsuariosWS.cs	Contiene y ejecuta las sentencias de SQL relacionadas	UsuariosWS (servicio web)

	con la recuperación, inserción, modificación y eliminación de usuarios	
--	--	--

Tabla 4 Descripción de los componentes del paquete de servicios web

3.7.2.6 Clases Auxiliares

Componente	Propósito	Contenido
Útiles.cs	Contiene métodos implementados que usan otras clases.	T_Utiles (clase)
ControlIO.cs	Constituyen controles que pertenecen al Inspector de Objetos y que a través de los mismos se muestran y establecen las propiedades de los controles agregados al Área de Trabajo.	T_ControlIO (clase)

Tabla 5 Descripción de los componentes del paquete de clases auxiliares

3.8 Conclusiones.

Después de analizar la descripción más detallada del sistema se puede concluir que es un sistema robusto, basado en una arquitectura eficiente dada por la implementación del mismo en diferentes capas donde cada una cumple una función determinada existiendo cierta independencia entre las diferentes partes que integran el sistema y permitiendo a la vez que el mismo resulte más flexible y de fácil mantenimiento. Por otra parte el sistema mantiene un control eficiente ante posibles errores a la vez que sigue un patrón de diseño

de la interfaz que hace al sistema fácil de manipular a través de una interfaz intuitiva y agradable.

Capítulo 4: Estudio de factibilidad.

4.1 Introducción.

En éste capítulo se realiza la estimación de costo en el estudio preliminar que tiene la realización del proyecto realizado. Basado en la herramienta COCOMO II (*Constructive Const Model*) se valoran una serie de resultados que permiten determinar un estimado de tiempo de duración del proyecto, cantidad de personas que se necesitan para desarrollarlo y el costo aproximado que tendrá, apoyándose en fórmulas de estimación y puntos de función establecidos por la herramienta.

Parte de este estudio de factibilidad lo constituyen los beneficios tangibles e intangibles, así como un análisis de costo beneficios que se encuentran después del cálculo de las estimaciones.

4.2 Planificación.

Entradas externas

Nombre de la entrada externa (EI)	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación
Colección	1	2	Simple
Producto	1	2	Simple

Estilos de Menú	2	7	Medio
Estilos de Fuente	2	10	Medio
Estilos de Botón	2	3	Simple
Palabras del glosario	2	9	Medio
Efemérides	2	9	Medio
Módulos	1	1	Simple
Temas	1	1	Simple
Epígrafes	1	1	Simple
Prototipos de Pantalla	7	24	Complejo
Pantallas de epígrafes	6	21	Complejo
Palabras Calientes	3	4	Simple
Usuarios	1	3	Simple
Autenticación	2	3	Simple
Elementos de Media de Biblioteca	1	7	Simple
Elementos de la Esquina del Maestro	2	4	Simple

Tabla 6 Cocomo entradas externas

Peticiones

Nombre de la petición (EQ)	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación según tabla no 2.
Temas por Producto	3	3	Simple
Epígrafes por Producto	4	4	Medio
Palabras del Glosario por Producto	3	3	Simple
Prototipos de Pantalla por Módulo	4	4	Medio
Pantallas por Epígrafe	5	5	Medio
Efemérides por Producto	3	3	Simple
Módulos por Producto	3	3	Simple
Productos por Colección	2	2	Simple

Tabla 7 Cocomo peticiones

Ficheros Internos

Nombre del fichero interno (ILF)	Cantidad de records	Cantidad de elementos de datos	Clasificación según tabla no 1.
Usuario	1	2	Simple
Tipo de Usuario	1	2	Simple

Colección	1	2	Simple
Producto	1	2	Simple
Estilo de Menú	1	3	Simple
Media de Biblioteca	1	7	Simple
Estilo de Fuente	1	8	Simple
Tipo de botón	1	2	Simple
Palabra de Glosario	1	2	Simple
Efemérides	1	2	Simple
Tema	1	1	Simple
Epígrafe	1	1	Simple
Módulo	1	1	Simple
Prototipo de Pantalla	1	1	Simple
Pantalla	1	1	Simple
Sonido	1	3	Simple
Mensaje	1	2	Simple
Mensaje con Media	1	2	Simple
Mensaje con texto	1	2	Simple
Control	1	6	Simple
Etiqueta	1	9	Simple
Control de Media	1	1	Simple
Caja de Texto	1	1	Simple
Menú	1	1	Simple
Palabra Caliente_Media	1	1	Simple
PalabraCaliente_Texto	1	1	Simple
Asignaturas	1	1	Simple
Elementos Asignatura	1	3	Simple

Tabla 8 Cocomo ficheros internos

Interfaces Externas

Nombre de la interfaz externa	Cantidad de records	Cantidad de elementos de datos	Clasificación(Simple, Media y compleja)
Colecciones WS	4	6	Simple
Productos WS	5	20	Medio
Usuarios WS	1	2	Simple
Temas WS	3	3	Simple
Controles WS	9	23	Complejo

Tabla 9 Cocomo interfaces externas

Puntos de Función desajustados

Elementos	Simple s	X Peso	Medios	X Peso	Complejos	X Peso	Subtotal de puntos de función
Ficheros lógicos internos	28	7	0	10	0	15	196
Ficheros de interfaces externas	3	5	1	7	1	10	32
Entradas externas	11	3	4	4	2	6	61
Salidas externas							
Peticiones	5	3	3	4		6	27
Total							316

Tabla 10 Cocomo puntos de función desajustados

Cálculo de las instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.

Características	Valor
Puntos de función desajustados	316
Lenguaje	C#
Instrucciones fuentes por puntos de función	59
Instrucciones fuentes	5,355

Tabla 11 Cálculo de las instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.

Multiplicadores de esfuerzo:

Multiplicador	Descripción	Valor
Complejidad y Confianza del Producto (RCPX)	La complejidad del producto es media.	1
Reutilización Requerida (RUSE)	Se implementa código reutilizable para su aprovechamiento en el proyecto.	1
Inconvenientes de la Plataforma (PDIF)	La plataforma es estable. Requerimientos bajos de almacenamiento y tiempo de ejecución.	0.87
Capacidad Personal (PERS)	La capacidad de los especialistas (analistas-programadores) es alta. La continuidad del personal es alta.	0.83
Experiencia Personal (PREX)	El equipo tiene poco dominio y conocimiento del lenguaje de programación, plataforma y herramientas de desarrollo utilizados. No ha desarrollado aplicaciones	1.33

	similares, casi ninguna experiencia.	
Facilidades (FCIL)	Se utilizan herramientas e instrumentos de programación modernos.	1
Planificación (SCED)	Los requerimientos de calendario de desarrollo son bajos.	1.14
7 EM = $\sum_{i=1}^7 E_{mi}$		1,09

Tabla 12 Cocomo multiplicadores de esfuerzo:

Factores de Escala

Factor	Descripción	Valor
PREC	Aspectos novedosos	3.72
FLEX	El sistema cuenta con alguna flexibilidad en relación con las especificaciones de los requerimientos preestablecidos y a las especificaciones de interfaz externa.	3.04
TEAM	Interacciones principalmente cooperativas. Mediana experiencia previa operando	2.19
RESL	La arquitectura es sólida y los riesgos generalmente se mitigan. Poca incertidumbre, riesgos no son críticos.	1.41
PMAT	Relación con el proceso de madurez del software. Nivel 3.	3.12
5		13,48

$\sum_{i=1} SFi$		
------------------	--	--

Tabla 13 Cocomo factores de escala

Valores calibrados:

$A=2.94; B=0.91; C=3.67; D=0.24$

$E = B + 0.01 * \sum SFi = 0.91+0.01*13,48= 1,04$

$F = D + 0.2 * (E - B)=0,24+0.2*(1.04-0.91)= 0,26$

Cálculo del esfuerzo (PM)

$PM = A * (MSLOC)^E * \Pi Emi = 2.94 * (18,644)^{1.04} * 1.09= 67,16 \text{ Hombres/Mes.}$

Cálculo del tiempo de desarrollo.

$TDEV = C * PM^F = 3.67 * (67,16)^{0.26} = 10,9 \approx 10 \text{ meses (Estimado)}$

Cálculo de la cantidad de hombres.

$CH = PM / TDEV = 67,16/10 = 6,716 \approx 6 \text{ hombres}$

Ajustando para la cantidad real de personas

$CH^* = 3 \text{ hombres.}$

$TEDV = PM/CH^* = 67,16 / 3 = 22 \text{ meses.}$

Cálculo del costo.

Asumiendo como salario promedio mensual (SP) \$225.00

$CHM = CH * SP = 3 * \$225.00 = \$ 675.00$

$\text{Costo} = CHM * PM = \$675 * 67,16 = \$ 45333.00$

Cálculos

Cálculo de:	Valor
Esfuerzo	67,16 Hombres/Mes
Tiempo de desarrollo	22Meses
Cantidad de hombres	3
Costo	\$45333
Salario medio	\$225

Tabla 14 Cocomo cálculos

4.4 Beneficios tangibles e intangibles.

4.4.1 Beneficios Tangibles

El desarrollo de la herramienta propuesta sin dudas traerá como consecuencias un ahorro considerable del tiempo empleado en la producción de aplicaciones multimedia educativas en la UCI para el MINED. Por otra parte se ahorrarán gran cantidad de recursos humanos y materiales que hasta el momento se usan en dicha labor.

Se reporta un costo de \$45333 por concepto de salario para los desarrolladores de dicha herramienta lo cuál se compensa con los beneficios que reporta especialmente el hecho de que permitirá apoyar el proceso de construcción de aplicaciones multimedia en un menor período de tiempo aumentando así la productividad de la UCI en este sentido.

4.4.2 Beneficios Intangibles

Mediante el desarrollo de la herramienta propuesta se logrará agilizar el proceso de creación de aplicaciones multimedia educativas en la UCI para satisfacer las solicitudes del MINED. Se permitirá tener un control centralizado de todos los recursos y componentes que se emplean en la construcción de las mismas a la vez que se hará más eficiente y cómodo el trabajo por parte de cada uno de los participantes en este proceso.

Se garantiza además un flujo de trabajo ordenado, donde cada trabajador tendrá un rol bien determinado. Se logrará además prescindir de personal capacitado en lenguajes de programación para el desarrollo de las aplicaciones multimedia y se disminuirá el esfuerzo

al permitir agrupar funcionalidades comunes a estas aplicaciones en componentes de fácil acceso por parte de los operadores de dicha herramienta evitando tener que repetir el mismo trabajo de programación para elementos iguales en aplicaciones diferentes.

4.5 Análisis de costos y beneficios.

La herramienta propuesta tiene un costo real, según los cálculos realizados en este capítulo, aproximado a los \$45333 pesos por concepto de salario contando con 3 hombres para su desarrollo en un tiempo alrededor de los 22 meses.

Un dato importante es que los desarrolladores con que se cuenta son estudiantes y no reciben salario por lo que analizando los beneficios tangibles e intangibles, en especial el ahorro en tiempo, la mayor productividad de aplicaciones multimedia educativas y la disminución del esfuerzo resulta evidente que es factible el desarrollo de dicha herramienta.

4.6 Conclusiones.

Una vez terminado el estudio de factibilidad del sistema, se estima un tiempo de 22 meses para su construcción por 3 hombres y su costo asciende a \$45333.00.

La herramienta propuesta trae consigo una serie de beneficios tangibles e intangibles dentro de los que se encuentran: ahorro en tiempo, mayor productividad de aplicaciones multimedia educativas, disminución del esfuerzo, ahorro de recursos humanos y materiales, por lo que se concluye que es factible el desarrollo de dicha herramienta.

Conclusiones

En el presente trabajo se ha hecho un análisis del desarrollo de aplicaciones multimedia educativas en la UCI para el MINED y se han determinado las dificultades que este proceso presenta. Se ha hecho un estudio de diferentes alternativas que pueden ser usadas para resolver dicho problema y se concluyó proponiendo el desarrollo de una herramienta que permita que permita la estructuración y gestión de los contenidos estas aplicaciones, para lo cuál se ha realizado un estudio de las tecnologías actuales que pueden ser empleadas para la implementación de la misma.

Este trabajo recoge las distintas etapas de desarrollo de la herramienta propuesta. Además se ha hecho un estudio de la factibilidad de dicha herramienta analizando sus beneficios tangibles e intangibles y su costo legando a la conclusión de que resulta factible la implementación del mismo.

Recomendaciones.

A pesar de que resulta satisfactorio desde todos los puntos de vista el desarrollo de la herramienta propuesta aún hay elementos que deben ser llevados a cabo para que la misma quede con calidad y para que ofrezca un mayor número de funcionalidades. Entre otros elementos se debe recomendar concluir con la etapa de pruebas dentro del proceso de desarrollo de la misma. Además es necesario lograr que dicha herramienta permita la estructuración y gestión de contenidos de todos los módulos que están definidos para las aplicaciones multimedia educativas desarrolladas en la UCI para el MINED, incluyendo el módulo de ejercicios y el de resultados. Se recomienda además agregarle nuevos componentes a dicha herramienta que ofrezcan un mayor número de funcionalidades a la vez que simplifique aún más el trabajo de los operadores de la misma.

Bibliografía

1. [Gon01]. González Seco, José Antonio. *El lenguaje de programación C#*. 2001.
<http://www.josanguapo.com/> (16/03/05).
2. [Arc01]. Archer, Tom. *A Fondo C#*. Primera edición en español. McGraw-Hill. 2001.
3. [Fer03]. Ferguson, Jeff, Patterson, Brian, Beres, Jason, Boutquin, Pierre, Gupta, Meeta. *La biblia de C#*. Edición española. Anaya. Madrid. 2003.
4. [Mar01]. Martin, Richard. *Designing SQL Server 2000 Databases for .net Enterprises Servers*. Edición digital. Syngress. 2001.
5. [Jan05]. *Aplicaciones basadas en web Internet/Intranet*.
<http://www.janium.com/phpws/index.php>. (24/03/05).
6. [Jac00]. Jacobson, Ivar, Booch, Grady, Rumbaugh, James. *El proceso Unificado de Desarrollo de Software*. Primera edición en español. Addison Wesley. 2000.
7. [Día05]. Díaz, Dionisio. *Multimedia en la enseñanza*.
<http://www.dionisiodiaz.com/multimensenanza/multimediaensenanza.html> (26/03/05).
8. [Mar05]. Martín, Maglio, Federico. *Concepto de Informática Educativa*.
<http://www.fmmeduccion.com.ar/Informatica/infoeduc.htm> . (26/03/05).
9. [Cab05]. Cabero, Almenara, Julio. *Creación de aplicaciones multimedias educativas con el programa: toolbok*. <http://tecnologiaedu.us.es/edutec/paginas/3.html>. (27/03/05).
10. [Zam05]. Zambrano, Rodríguez, Douglas Francisco. *Multimedia*.
<http://monografias.com>. (27/03/05).
11. [Nyg00]. Nyquits, R., John, Martin, Robert. *Director 8*. Anaya. 2000.
12. [Iss02]. Issi, Camy, Lázaro. *Flash MX*. Anaya. 2002.

Glosario de términos.

COM (*Component Object Model*):

Tecnología para construir sistemas software a partir de la interacción de objetos (componentes). El modelo de programación COM está basado en la distribución de código de clases en componentes binarios.

COM+

COM+, no es más que la integración de la arquitectura COM y MTS (*Microsoft Transaction Server*). Como COM, COM+, es basado sobre componentes binarios y programación basada en interfaces. Los Componentes COM+ pueden ser actualizados y extendidos una vez que estén en explotación sin afectar a las aplicaciones clientes que los usan en la producción.

MTS (*Microsoft Transaction Server*).

MTS es una pieza de software que fue creada para *Windows NT Server*. MTS permite a los objetos de la capa media correr sobre *Windows NT Server* y controlar las transacciones distribuidas, es decir, permite a los componentes ser esparcidos por la red y que se ejecuten en otras computadoras con sistema operativo *Windows NT Server*.

Excepción

Una excepción es un evento que ocurre durante la ejecución del programa que interrumpe el flujo normal de las sentencias. Son una forma clara para controlar los errores sin confundir el código con muchas instrucciones de control del error.

Interface:

Cuando una persona utiliza una herramienta o interactúa con cualquier sistema, existe un punto de contacto, existe un punto de contacto entre el sistema y la persona, por donde se transmite la información entre ellos, este espacio común donde se comunican las dos partes es lo que llamamos interface

URL (*Uniform Resource Locator*):

Es el sistema de direcciones usado en la Web y otros recursos de Internet. El URL contiene información sobre el método de acceso, el servidor al que se accede y la dirección o el fichero.

HTTP

Conjunto de reglas para intercambiar archivos (texto, gráfica, imágenes, sonido, video y otros archivos multimedia) en la World Wide Web.

TCP/IP

Se refieren a dos protocolos de red: *Transmission Control Protocol* (Protocolo de Control de Transmisión) e *Internet Protocol* (Protocolo de Internet). Los diferentes protocolos de la suite TCP/IP trabajan conjuntamente para proporcionar el transporte de datos dentro de Internet (o Intranet). En otras palabras, hacen posible que poder acceder a los distintos servicios de la Red.

Anexos

Descripción de los casos de uso

Nombre del caso de uso	Autenticación
Actores	Operador
Propósito	Permitir la autenticación de los usuarios para que el sistema le otorgue los derechos que les corresponden según el tipo de usuario al que pertenece.
Resumen	<p>Los usuarios entran al sistema su identificador (IdUsuario) y su contraseña. El sistema encrypta la contraseña y verifica los datos entrados.</p> <p>Si los datos entrados son válidos entonces mostrará al usuario la pantalla del menú principal del sistema, la que mostrará las opciones según el tipo de usuario.</p> <p>Si los datos suministrados son incorrectos, el sistema mostrará un mensaje de alerta en dependencia del error detectado.</p>
Referencias	
Precondiciones	
Poscondiciones	Si el usuario existe en la Base de Datos y su contraseña es correcta el usuario accede al menú principal con las opciones de menú según el tipo del usuario autenticado, de lo contrario se muestra un mensaje de alerta.
Requisitos especiales	(8)
Pantallas	



Figura 1 Pantalla de autenticación

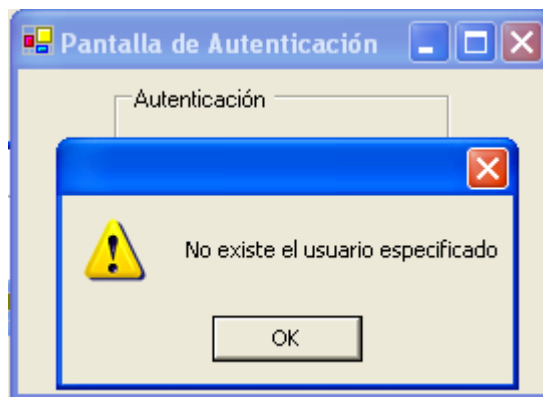
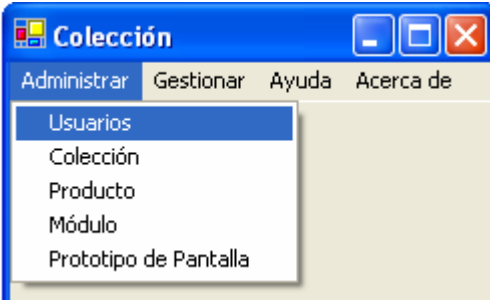


Figura 2 Mensaje de alerta 1



Figura 3 Mensaje de alerta 2

Nombre del caso de uso	Gestionar Usuarios
Actores	Administrador
Propósito	Permitir la gestión de los usuarios del sistema. (Creación de nuevos usuarios, modificación de sus datos y eliminación de los mismos)
Resumen	
<p>Una vez que el administrador entra al sistema y se autentica como tal se muestra la pantalla del menú principal con las opciones de menú que le corresponden como administrador. El administrador selecciona la opción de menú “Gestionar Usuarios” y posteriormente se muestra la ventana para la gestión de usuarios donde podrá elegir la acción que desea realizar sobre algún usuario del sistema.</p> <p>Según la opción elegida en esta ventana se mostrará una nueva ventana donde el administrador podrá realizar la acción correspondiente (insertar, modificar o eliminar usuarios).</p> <p>Si ocurre algún error en alguna de las operaciones realizadas el sistema mostrará un mensaje de alerta según corresponda.</p>	
Referencias	
Precondiciones	El administrador del sistema debe haberse autenticado y el sistema reconocerlo como tal.
Poscondiciones	Se inserta un nuevo usuario, se modifican sus datos o se elimina.
Requisitos especiales	(8)
Pantallas	
	
Figura 4 Pantalla de menú principal	

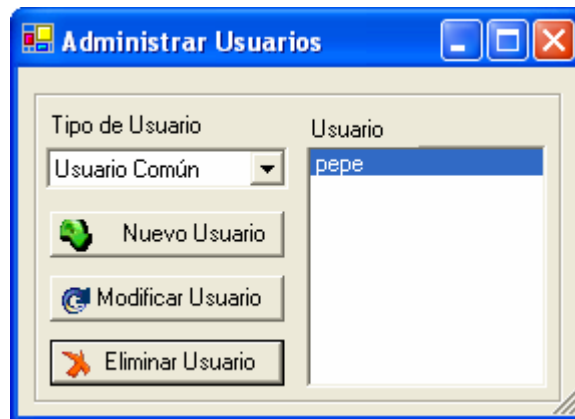


Figura 5 Pantalla de administración de usuarios

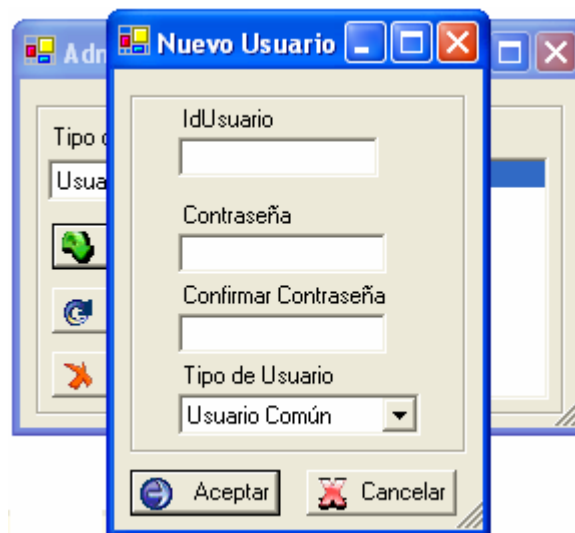


Figura 6 Pantalla de inserción de nuevos usuarios



Figura 7 Pantalla de modificación de usuarios

Nombre del caso de uso	Gestionar Colección
Actores	Administrador
Propósito	Permitir la gestión de colecciones.
Resumen	<p>El administrador selecciona la opción de menú “Gestionar/Colección” y posteriormente se muestra la ventana para la gestión de colecciones donde podrá elegir la acción que desea realizar sobre alguna colección.</p> <p>Según la opción elegida en esta ventana se mostrará una nueva ventana donde el administrador podrá realizar la acción correspondiente (insertar, modificar o eliminar colecciones).</p> <p>Si ocurre algún error en alguna de las operaciones realizadas el sistema mostrará un mensaje de alerta según corresponda.</p>
Referencias	Requisitos Funcionales: Gestionar Colecciones.
Precondiciones	El administrador del sistema debe haberse autenticado y el sistema reconocerlo como tal.

Poscondiciones	Se inserta una nueva colección, se modifican los datos de la colección especificada, o se elimina esta.
Requisitos especiales	

Pantallas



Figura 8 Pantalla de menú principal

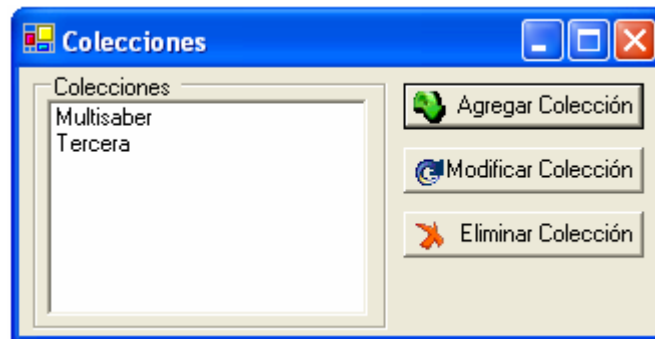


Figura 9 Pantalla de gestión de colecciones

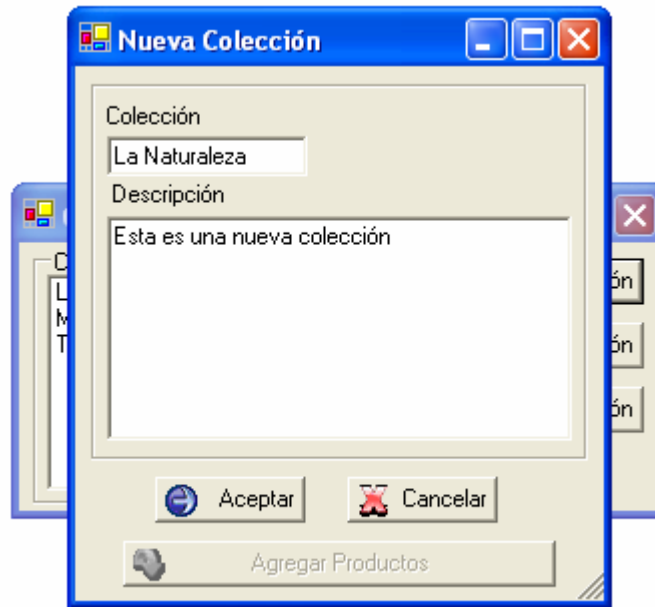


Figura 10 Pantalla de inserción de nuevas colecciones



Figura 11 Pantalla de modificación de colecciones

Nombre del caso de uso	Gestionar Producto
Actores	Administrador
Propósito	Permitir la gestión de productos.
Resumen	
<p>El administrador selecciona la opción de menú “Gestionar/Producto” y posteriormente se muestra la ventana para la gestión de productos. El administrador selecciona la colección a la que pertenece el producto en cuestión, listándose en cada momento los productos existentes para esa colección. El administrador podrá elegir la acción que desea realizar sobre algún producto.</p> <p>Según la opción elegida en esta ventana se mostrará una nueva ventana donde el administrador podrá realizar la acción correspondiente (insertar, modificar o eliminar productos).</p> <p>Desde la ventana de inserción y modificación de productos se accede a otras ventanas que permiten establecer propiedades a los productos tales como: estilo de menú, estilos de texto, etc.</p> <p>Si ocurre algún error en alguna de las operaciones realizadas el sistema mostrará un mensaje de alerta según corresponda.</p>	
Referencias	Requisitos Funcionales: Gestionar Productos.
Precondiciones	El administrador del sistema debe haberse autenticado y el sistema reconocerlo como tal.
Poscondiciones	Se inserta un nuevo producto, se modifican los datos del producto especificado, o se elimina el mismo.
Requisitos especiales	
Pantallas	

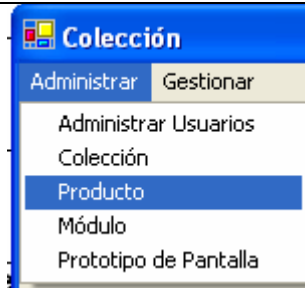


Figura 12 Pantalla de menú principal

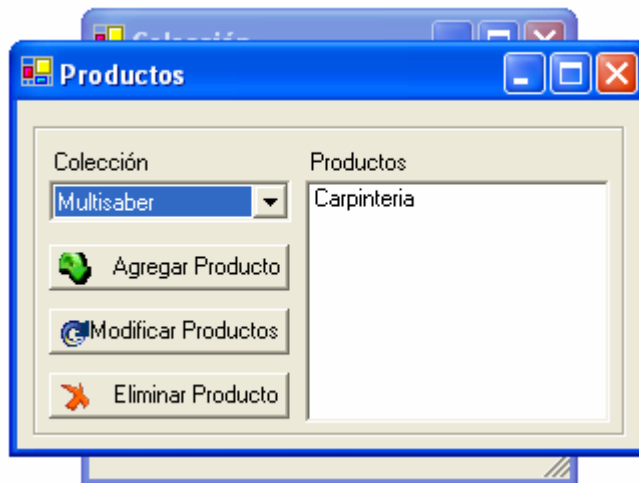


Figura 13 Pantalla de gestión de productos



Figura 14 Pantalla de inserción y modificación de nuevos productos

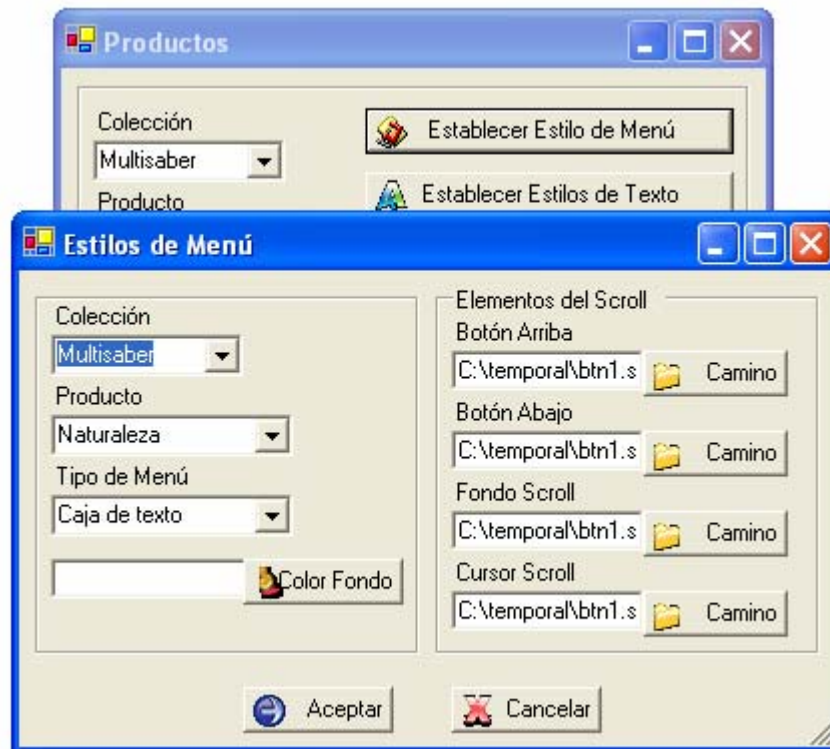


Figura 15 Pantalla de gestión de estilos de menú

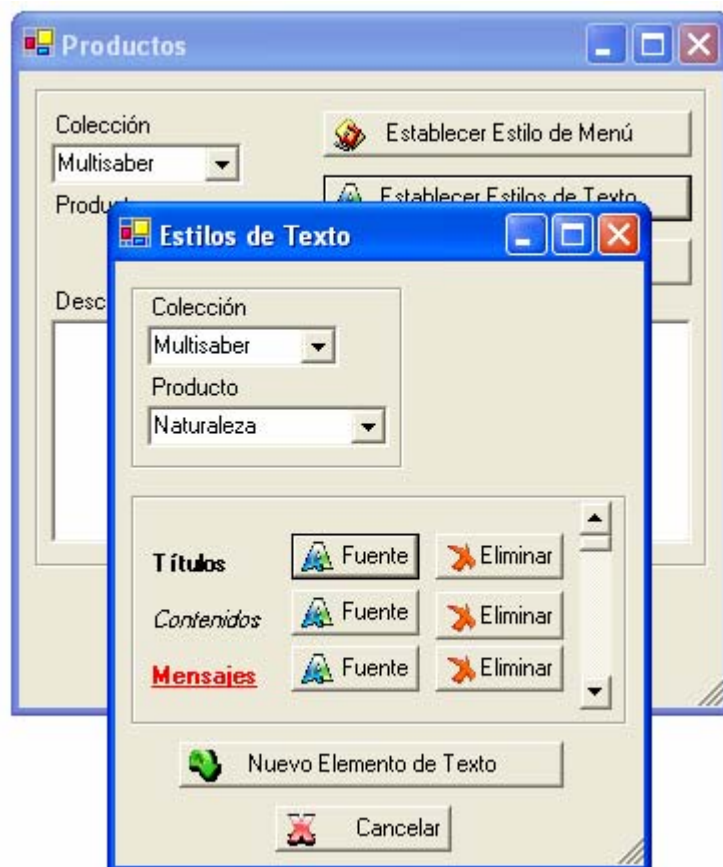


Figura 16 Pantalla de gestión de estilos de texto

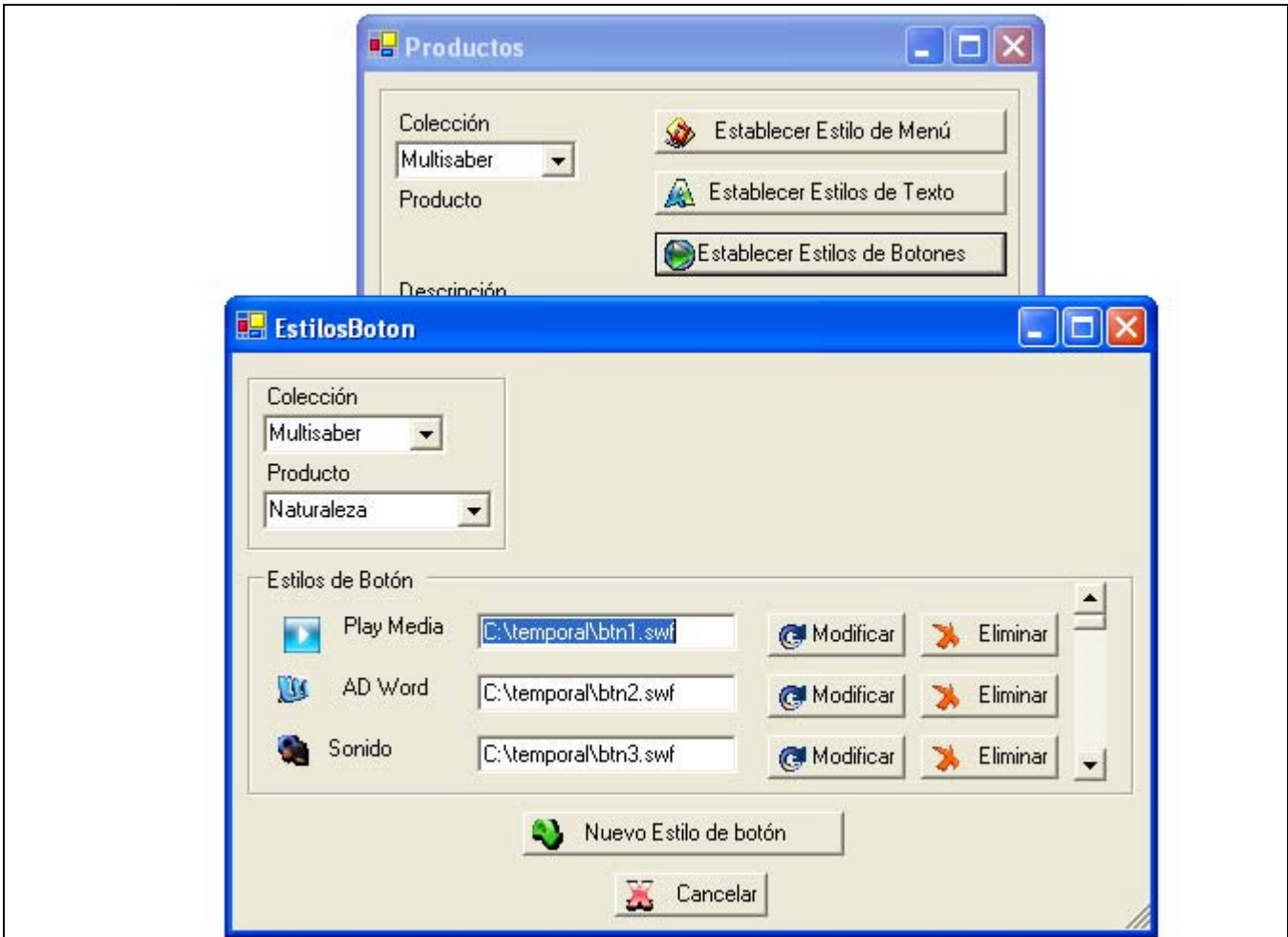


Figura 17 Pantalla de gestión de estilos de botón

Nombre del caso de uso	Gestionar Módulos
Actores	Administrador
Propósito	Permitir la gestión de módulos.
Resumen	
<p>El administrador selecciona la opción de menú “Gestionar/Módulo” y posteriormente se muestra la ventana para la gestión de módulos. En esta ventana el administrador especifica a qué colección y producto pertenece el módulo sobre el que realizará la acción deseada, listándose en cada selección los módulos que ya existen para la colección y productos</p>	

especificados.

Según la opción elegida en esta ventana se mostrará una nueva ventana donde el administrador podrá realizar la acción correspondiente (insertar, modificar o eliminar módulos).

Si ocurre algún error en alguna de las operaciones realizadas el sistema mostrará un mensaje de alerta según corresponda.

Referencias	Requisitos Funcionales: Gestionar Módulos.
Precondiciones	El administrador del sistema debe haberse autenticado y el sistema reconocerlo como tal. Que exista el producto al cuál se va a añadir dicho módulo.
Poscondiciones	Se agrega, modifica o elimina un Módulo.
Requisitos especiales	

Pantallas



Figura 18 Pantalla de menú principal



Figura 19 Pantalla de gestión de módulos

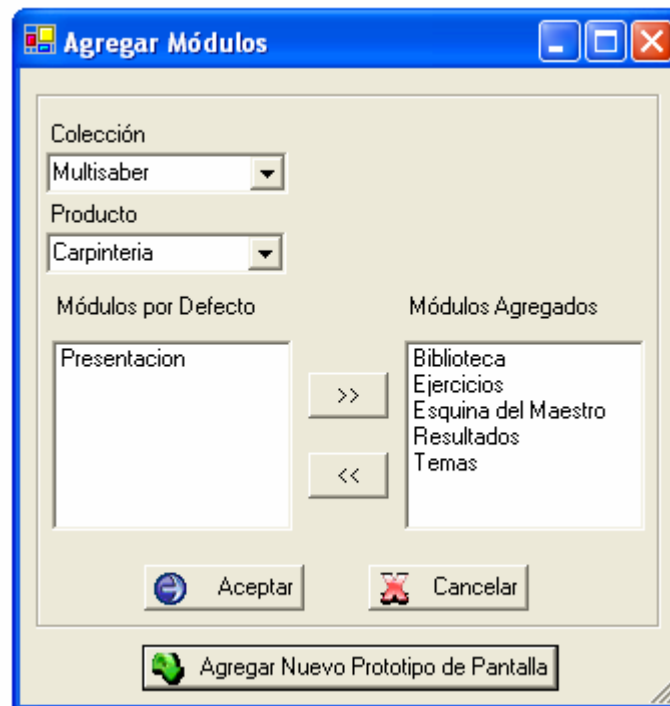


Figura 20 Pantalla de inserción de nuevos módulos



Figura 21 Pantalla de modificación de módulos

Nombre del caso de uso	Gestionar Prototipos de Pantalla
Actores	Administrador
Propósito	Permitir la gestión de prototipos de pantalla.
Resumen	
<p>El administrador selecciona la opción de menú “Gestionar/Prototipo de Pantalla” y posteriormente se muestra la ventana para la gestión de prototipos de pantalla. En esta ventana el administrador especifica a qué colección, producto y módulo pertenece el prototipo de pantalla sobre el que realizará la acción deseada, listándose en cada selección los prototipos de pantalla que ya existen para la colección, producto y módulo especificados. Según la opción elegida en esta ventana se mostrará una nueva ventana donde el administrador podrá realizar la acción correspondiente (insertar, modificar o eliminar prototipos de pantalla).</p> <p>Si ocurre algún error en alguna de las operaciones realizadas el sistema mostrará un mensaje de alerta según corresponda.</p>	

Referencias	Requisitos Funcionales: Gestionar Prototipos de Pantalla.
Precondiciones	El administrador del sistema debe haberse autenticado y el sistema reconocerlo como tal. Que exista el módulo al cuál se va a añadir dicho prototipo de pantalla.
Poscondiciones	Se agrega, modifica o elimina un prototipo de pantalla.

Requisitos especiales

Pantallas



Figura 22 Pantalla de menú principal



Figura 23 Pantalla de gestión de prototipos de pantalla

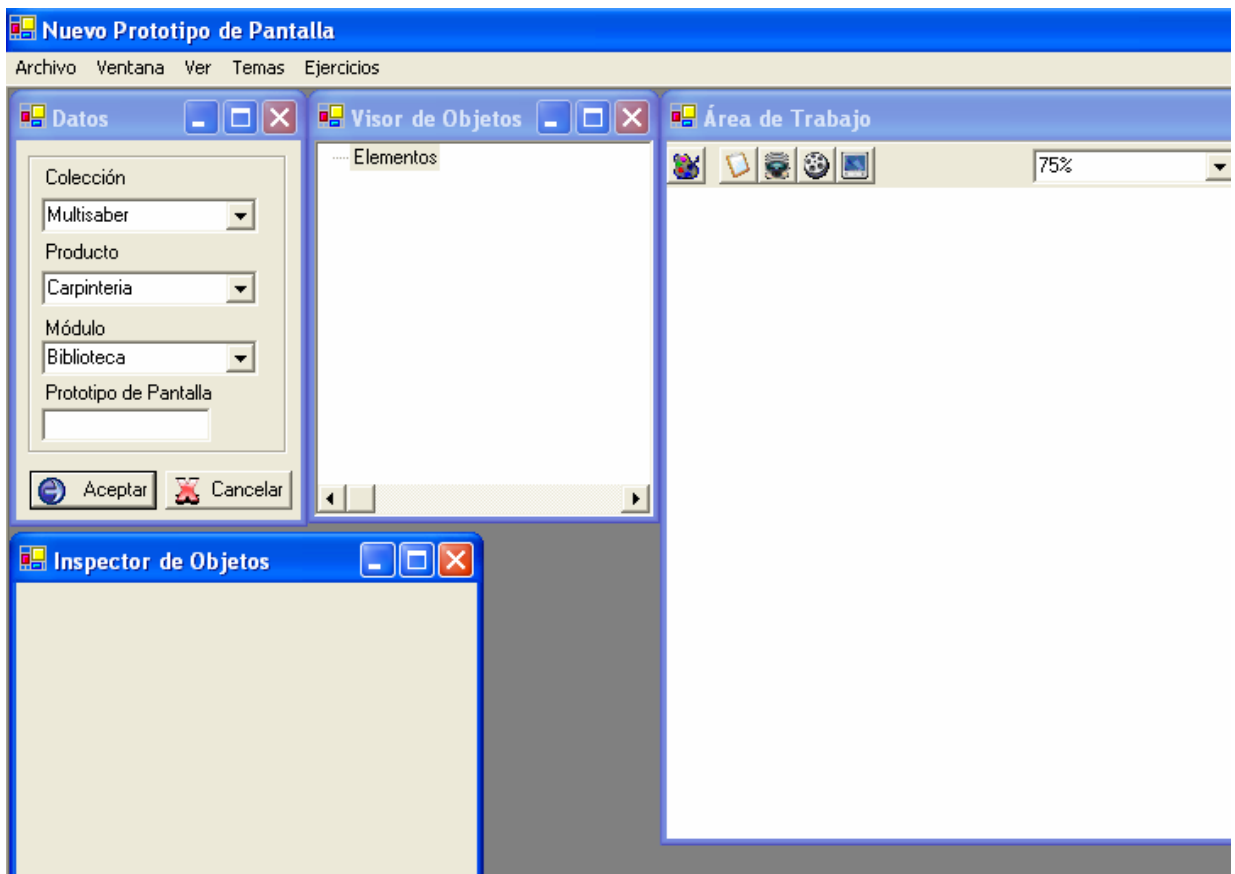


Figura 24 Pantalla de inserción de nuevos prototipos de pantalla

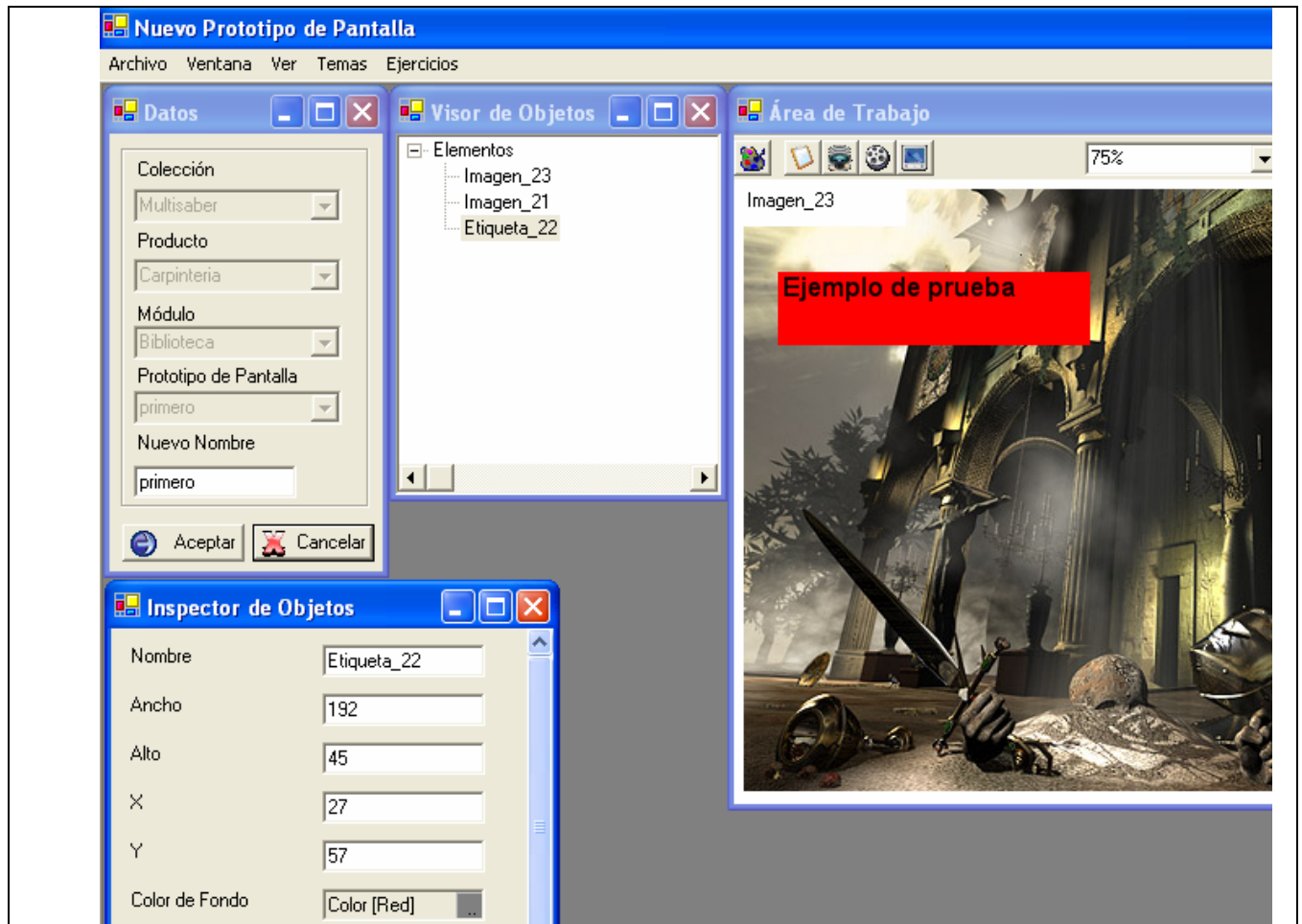


Figura 25 Pantalla de modificación de prototipos de pantalla

Nombre del caso de uso	Gestionar Controles
Actores	Operador
Propósito	Permitir la gestión de controles que pueden estar asociados a prototipos de pantalla, pantallas epígrafes, ejercicios, etc.
Resumen	<p>Cuando se muestra una de las ventanas de inserción o modificación de prototipos de pantalla o pantallas de epígrafes aparece una paleta de controles que permite la inserción de un control al “Área de Trabajo” que representa al prototipo de pantalla o pantalla de</p>

epígrafe en cuestión.

Para agregar un nuevo control solo basta con pulsar uno de los botones de la paleta que representa al control en cuestión y luego pulsar el “Área de Trabajo”. Una vez agregado un nuevo control se agrega su referencia en el “Visor de Objetos” (Ventana que muestra una referencia por cada control que aparece en el “Área de Trabajo” y que permite cambiar el orden en que aparecen los controles en profundidad) y se muestran sus propiedades en el “Inspector de Objetos”

Al pulsar el control aparece un contorno a su alrededor. Situando el puntero sobre ese contorno y arrastrándolo se modifican sus propiedades de ancho y alto. Para modificar la posición de un control dentro del “Área de Trabajo” basta con pulsarlo y arrastrarlo a la posición deseada. El resto de las propiedades asociadas a un control varían de acuerdo al tipo de control y se manipularán a través del “Inspector de Objetos”. Una vez modificado un control se actualizan sus propiedades en el “Inspector de Objetos”

Para eliminar un control basta con pulsarlo y presionar una de las teclas de borrar en el teclado. Una vez eliminado un control se limpia el “Inspector de Objetos” y se elimina su referencia del “Visor de Objetos”.

Referencias	Requisitos Funcionales: Gestionar Controles. Casos de Uso: Gestionar Prototipos de Pantalla (extensión) Gestionar Pantallas de Epígrafes (extensión)
Precondiciones	El operador debe haber accedido a la pantalla de inserción o modificación de prototipos de pantalla, o pantallas de epígrafes.
Poscondiciones	Se agrega, modifica o elimina un control al prototipo de pantalla o pantalla de epígrafe en cuestión.
Requisitos especiales	
Pantallas	



Figura 25 Gestión de Controles

Nombre del caso de uso	Gestionar Temas
Actores	Operador
Propósito	Permitir la gestión de temas.
Resumen	
<p>El operador selecciona la opción de menú “Gestionar/Temas” y posteriormente se muestra una ventana para la gestión de temas. En esta ventana el operador especifica a qué colección pertenece el tema sobre el que realizará la acción deseada, listándose en consecuencia los temas que ya existen para la colección y productos especificados.</p> <p>Según la opción elegida en esta ventana se mostrará una nueva ventana donde el operador</p>	

podrá realizar la acción correspondiente (insertar, modificar o eliminar temas).
 Si ocurre algún error en alguna de las operaciones realizadas el sistema mostrará un mensaje alerta según corresponda.

Referencias	Requisitos Funcionales: Gestionar Temas.
Precondiciones	El operador del sistema debe haberse autenticado. Que exista el producto al cuál se va a añadir dicho tema.
Poscondiciones	Se agrega, modifica o elimina un tema.
Requisitos especiales	

Pantallas



Figura 26 Pantalla de menú principal




Figura 27 Pantalla de gestión de temas



Figura 28 Pantalla de inserción de nuevos temas



Figura 29 Pantalla de modificación de temas

Nombre del caso de uso	Gestionar Epígrafes
Actores	Operador
Propósito	Permitir la gestión de epígrafes.
Resumen	
<p>El operador selecciona la opción de menú “Gestionar/Epígrafes” y posteriormente se muestra la ventana para la gestión de epígrafes. En esta ventana el operador especifica a qué colección, producto y tema pertenece el epígrafe sobre el que realizará la acción deseada, listándose en cada selección los epígrafes que ya existen para el tema de la colección y producto especificados.</p> <p>Según la opción elegida en esta ventana se mostrará una nueva ventana donde el operador podrá realizar la acción correspondiente (insertar, modificar o eliminar epígrafes).</p> <p>Si ocurre algún error en alguna de las operaciones realizadas el sistema mostrará un mensaje de alerta según corresponda.</p>	
Referencias	Requisitos Funcionales: Gestionar Epígrafes.
Precondiciones	El operador del sistema debe haberse autenticado. Que exista el tema al cuál se va a añadir dicho epígrafe.
Poscondiciones	Se agrega, modifica o elimina un epígrafe.
Requisitos especiales	
Pantallas	
	
Figura 30 Pantalla de menú principal	

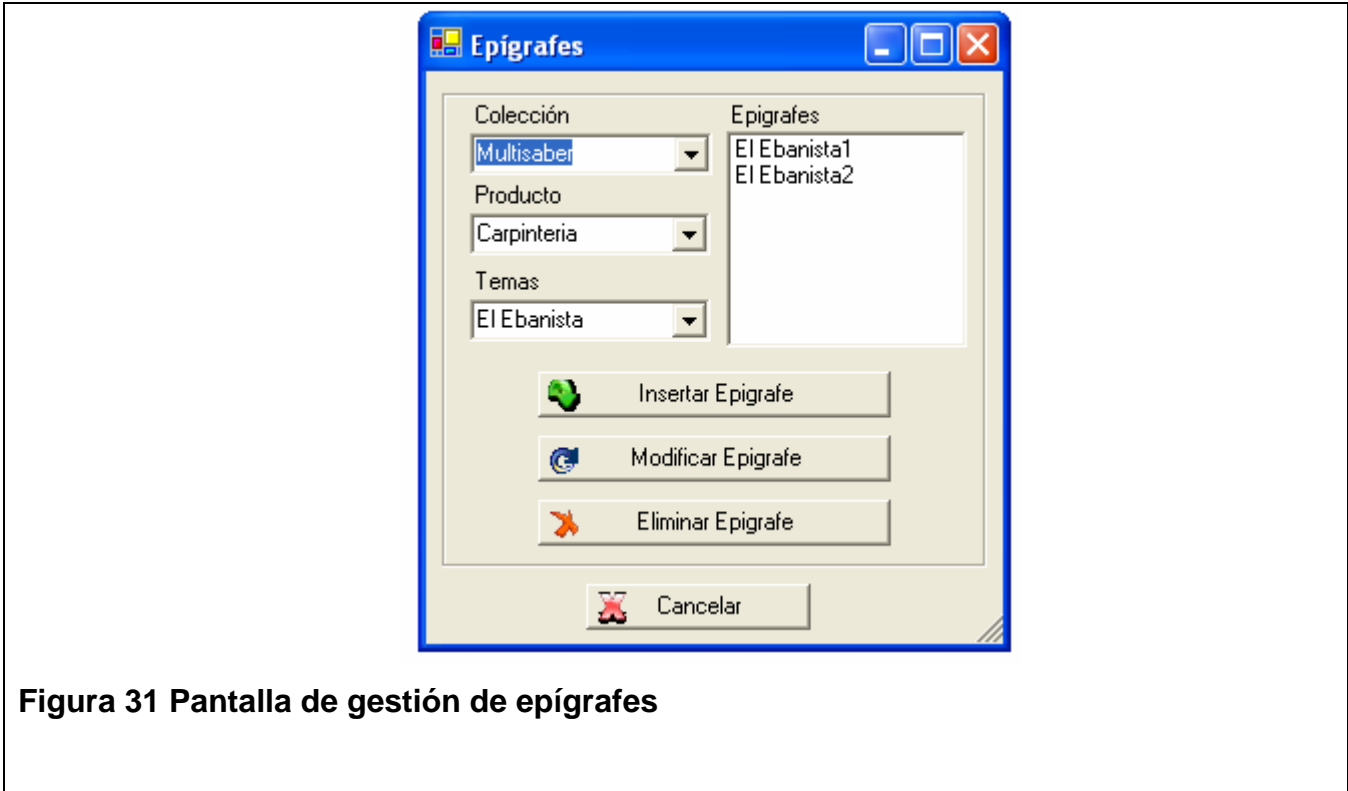


Figura 31 Pantalla de gestión de epígrafes

Nombre del caso de uso	Gestionar Pantallas de Epígrafes
Actores	Operador
Propósito	Permitir la gestión de pantallas de epígrafes.
Resumen	
<p>El operador selecciona la opción de menú “Gestionar/Pantallas de Epígrafes” y posteriormente se muestra la ventana para la gestión de pantallas. En la misma el operador selecciona a qué colección, producto, tema y epígrafe pertenece la pantalla sobre la que desea realizar la acción, actualizándose en cada selección la vista de la pantalla con los controles que le corresponden.</p> <p>Si va a crear una nueva pantalla deberá especificar además el prototipo de pantalla que le corresponde. Se brindan opciones para salvar los cambios hechos a una pantalla, eliminar la pantalla en curso o crear una nueva.</p>	
Referencias	Requisitos Funcionales:

	Gestionar Pantallas de Epígrafes.
Precondiciones	<p>El operador del sistema debe haberse autenticado.</p> <p>Que exista el epígrafe al que pertenece la pantalla en cuestión.</p> <p>Que existan prototipos de pantalla disponibles.</p>
Poscondiciones	Se agrega, modifica o elimina una pantalla.
Requisitos especiales	

Pantallas

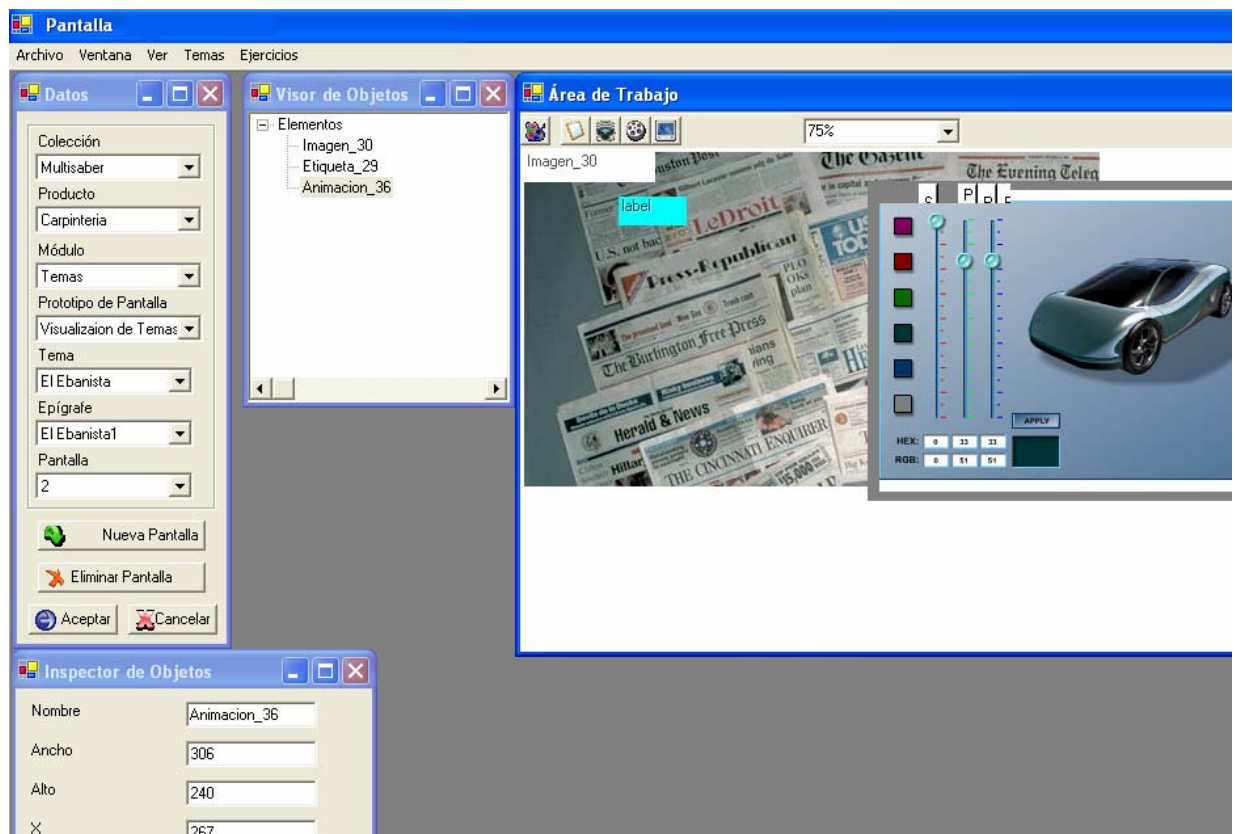


Figura 32 Pantalla de gestión de pantallas de epígrafes

Nombre del caso de uso	Gestionar Palabras Calientes
Actores	Operador
Propósito	Permitir la gestión de palabras calientes.
Resumen	
<p>El operador selecciona en la paleta de componentes del Área de Trabajo el componente caja de texto para introducir el contenido a mostrar en la misma y se abre una ventana de edición de contenido. En esta caja de texto aparece una paleta de componentes para establecer el font a las palabras seleccionadas. El operador selecciona la palabra haciendo doble clic sobre la misma y luego pulsa el componente de establecer palabra caliente. Se muestra una ventana donde podrá especificar la acción que se realizará al interactuar con esta palabra caliente (mostrar texto o media) y aquí podrá establecer el texto o el elemento de media asociado a la misma.</p>	
Referencias	<p>Requisitos Funcionales: Gestionar Palabras Calientes.</p> <p>Casos de Uso: Gestionar controles (Extensión)</p>
Precondiciones	El operador del sistema debe haberse autenticado.
Poscondiciones	Se agrega, modifica o elimina una palabra caliente.
Requisitos especiales	
Pantallas	

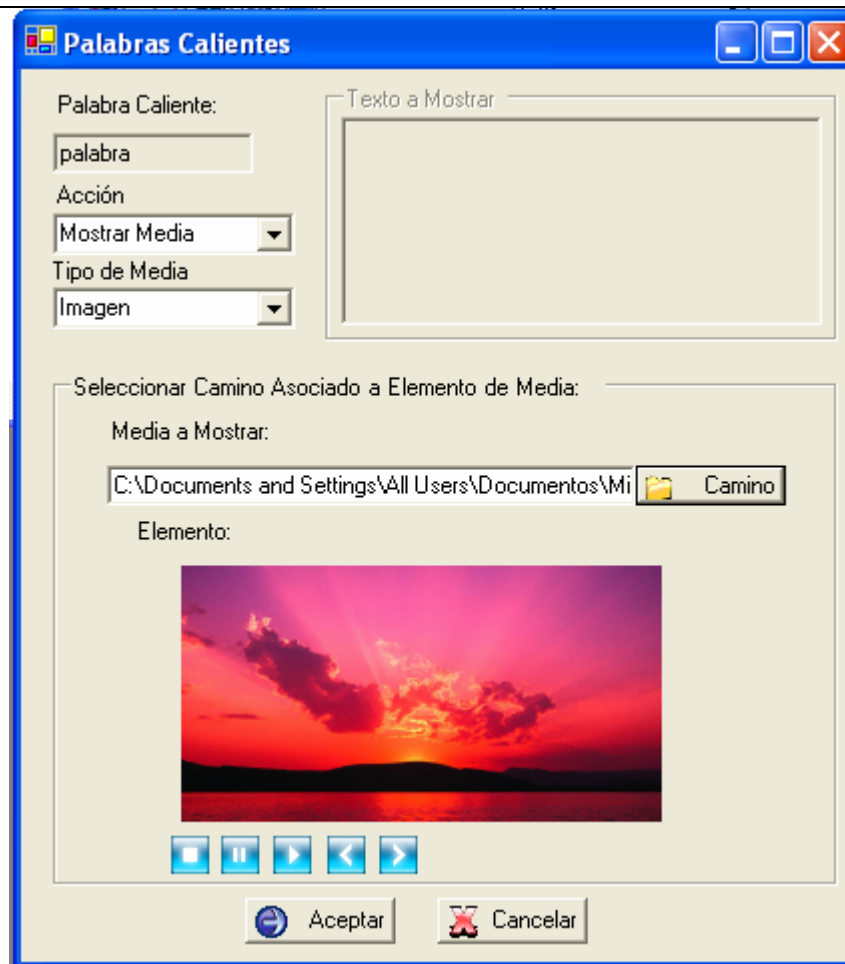


Figura 33 Pantalla de gestión de palabras calientes.

Nombre del caso de uso	Gestionar Elementos de Media de Biblioteca
Actores	Operador
Propósito	Permitir la gestión de los elementos de media de la biblioteca.
Resumen	El operador selecciona la opción de menú “Gestionar/Elementos de Media de Biblioteca” y

posteriormente se muestra la ventana para la gestión de los elementos de media de la biblioteca. El operador debe seleccionar la colección y el producto al que pertenece el elemento de biblioteca sobre el que desea realizar la acción y el tipo de elemento que es: imagen, animación, etc., mostrándose en cada selección los elementos de biblioteca que existen para los datos especificados. Se muestra una vista previa del elemento seleccionado y si es de tipo animación o sonido se habilitan botones para su reproducción. Desde aquí el operador podrá insertar un nuevo elemento, modificar uno existente, o eliminarlo.

Referencias	Requisitos Funcionales: Gestionar Elementos de media de biblioteca.
Precondiciones	El operador del sistema debe haberse autenticado.
Poscondiciones	Se agregan, modifican o eliminan los elementos de media de biblioteca.
Requisitos especiales	

Pantallas



Figura 34 Pantalla de gestión de elementos de media de biblioteca



Figura 35 Pantalla de edición de elementos de media de biblioteca

Nombre del caso de uso	Gestionar Efemérides
Actores	Operador
Propósito	Permitir la gestión de efemérides.
Resumen	
El operador selecciona la opción de menú “Gestionar/Efemérides y posteriormente se muestra la ventana para la gestión de las efemérides. El operador debe seleccionar la colección y el producto al que pertenece la efeméride sobre la que desea realizar la acción.	

Al seleccionar una fecha se listan los acontecimientos relacionados con la misma. Desde aquí el operador podrá insertar una nueva efeméride, modificar una existente, o eliminarla. El operador podrá además añadir elementos de media de biblioteca relacionados con la efeméride en cuestión.

Referencias	Requisitos Funcionales: Gestionar Efemérides.
Precondiciones	El operador del sistema debe haberse autenticado.
Poscondiciones	Se agregan, modifican o eliminan efemérides
Requisitos especiales	

Pantallas

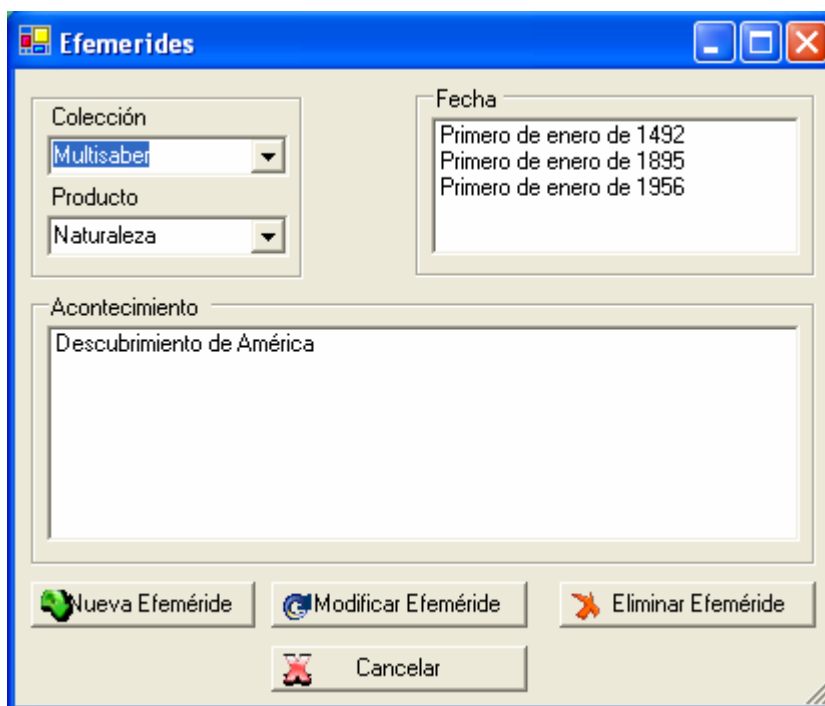


Figura 36 Pantalla de gestión de efemérides

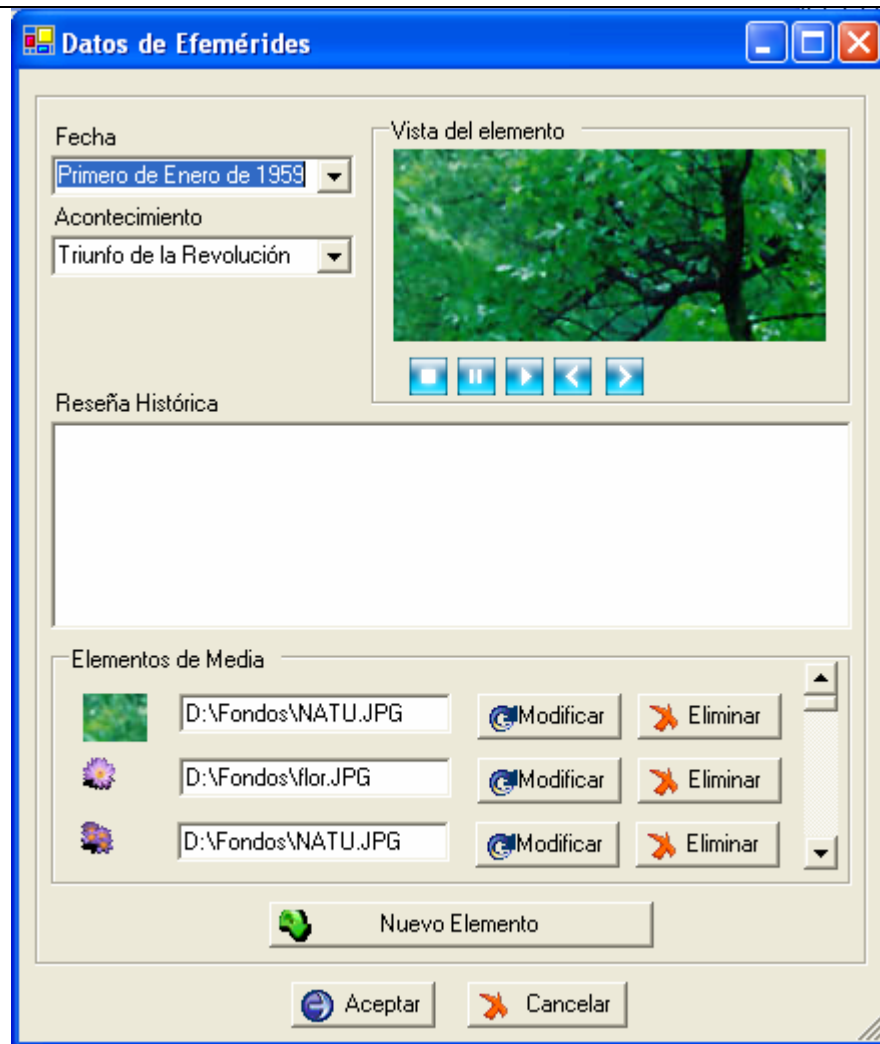


Figura 37 Pantalla de edición de efemérides

Nombre del caso de uso	Gestionar Palabras del Glosario
Actores	Operador
Propósito	Permitir la gestión de las palabras del glosario.
Resumen	El operador selecciona la opción de menú “Gestionar/Palabras del Glosario y posteriormente se muestra la ventana para la gestión de las palabras del glosario. El

operador debe seleccionar la colección y el producto al que pertenece la palabra sobre la que desea realizar la acción. Al seleccionar una palabra se muestra su significado. Desde aquí el operador podrá insertar una nueva palabra, modificar una existente, o eliminarla. El operador podrá además añadir elementos de media de biblioteca relacionados con la palabra en cuestión.

Referencias	Requisitos Funcionales: Gestionar Palabras del Glosario.
Precondiciones	El operador del sistema debe haberse autenticado.
Poscondiciones	Se agregan, modifican o eliminan palabras del glosario.
Requisitos especiales	

Pantallas



Figura 38 Pantalla de gestión de palabras del glosario



Figura 39 Pantalla de edición de palabras del glosario

Nombre del caso de uso	Gestionar Elementos de Esquina del Maestro
Actores	Operador
Propósito	Permitir la gestión de los programas de las asignaturas, recomendaciones metodológicas y artículos de actualización.
Resumen	El operador selecciona la opción de menú “Gestionar/Elementos de la Esquina del Maestro” y posteriormente se muestra la ventana para la gestión de las asignaturas. El operador selecciona la colección y el producto se listan las asignaturas vinculadas a estos

elementos. Desde aquí el operador podrá insertar nuevas asignaturas y definir el programa, las recomendaciones metodológicas y los artículos de actualización vinculados a estas. Los elementos antes mencionados son documentos Word que se encuentran en una estructura de directorios previamente definida.

<p>Referencias</p>	<p>Requisitos Funcionales:</p> <ul style="list-style-type: none"> - Gestionar recomendaciones metodológicas para cada producto. - Gestionar artículos de actualización que podrán consultar los profesores. - Gestionar programas de las asignaturas.
<p>Precondiciones</p>	<p>El operador del sistema debe haberse autenticado.</p>
<p>Poscondiciones</p>	<p>Se agregan, modifican o eliminan programas de las asignaturas, recomendaciones metodológicas y artículos de actualización.</p>
<p>Requisitos especiales</p>	

Pantallas

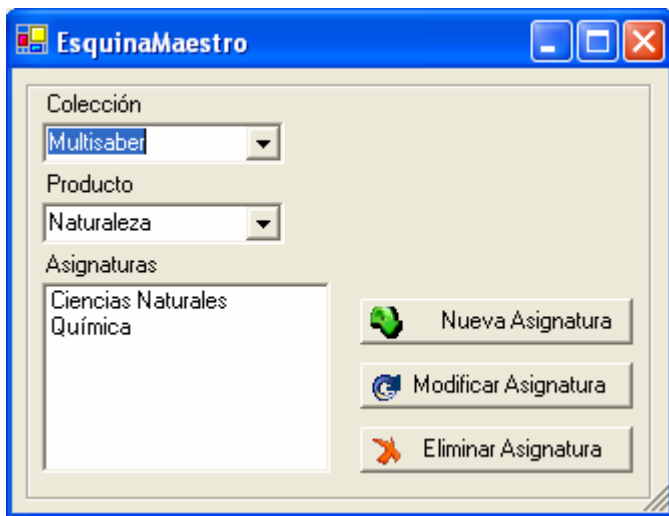


Figura 40 Pantalla de gestión de asignaturas

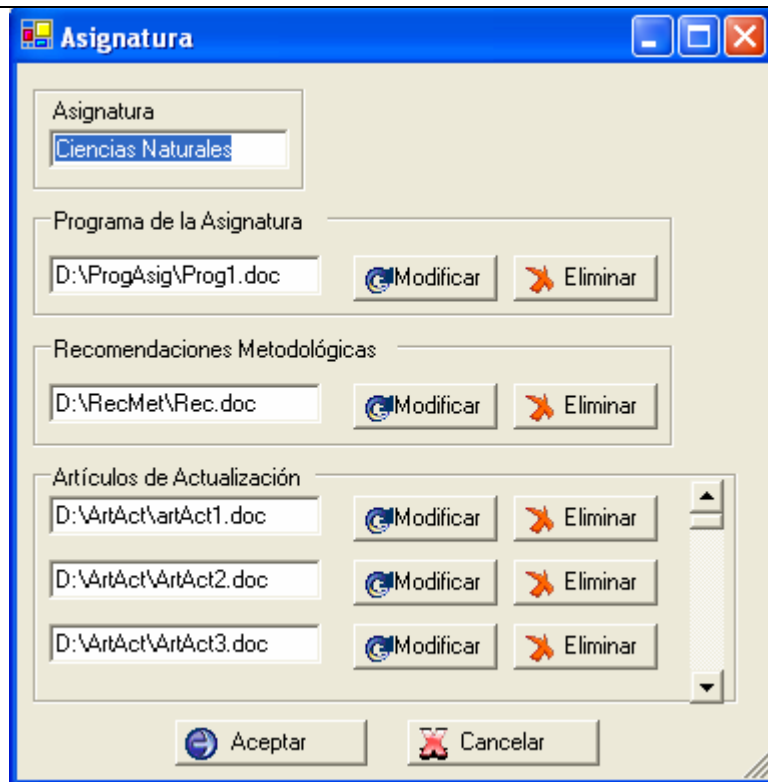


Figura 41 Pantalla de gestión de elementos de la esquina del maestro