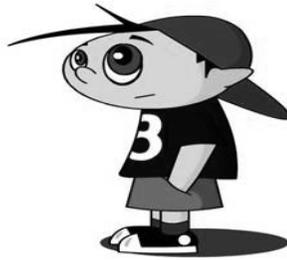


Multimedia Historia de la Facultad 8



*Trabajo de diploma para optar por el título de Ingeniero
en Ciencias Informáticas*

*Autores: Lisset Torreblanca Carvajal
Maykel Cobas Terrero*

Tutor: Lic. Héctor Matías González

*Ciudad de La Habana, junio de 2007.
"Año del 49 Aniversario de la Revolución."*

Declaración de autoría.

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de junio del año 2007.

Lisset Torreblanca

Maykel Cobas

Héctor Matías

Firma del Autor

Firma del Autor

Firma del Tutor

No hay más que asomarse a las puertas de la tecnología y la ciencia contemporáneas para preguntarnos si es posible vivir y conocer ese mundo del futuro sin un enorme caudal de preparación y conocimientos...

Fidel Castro

Agradecimientos.

A todos nuestros familiares, por su preocupación.

A Michel, por su ayuda en el diseño y el tiempo dedicado.

A nuestro tutor, por su ayuda.

A los compañeros de aula, por hacer más llevaderos todos estos años.

A todos los que, de alguna forma, han colaborado con la realización de este proyecto de tesis.

A todos los compañeros del quinto año, por ser fundadores de esta universidad y por haber vivido tantas cosas lindas juntos.

A nuestro eterno Comandante en Jefe, por ser el autor intelectual de esta obra tan linda y permitir que hayamos sido los primeros en cimentarla.

Gracias a todos.

Lisset:

A mi madre, por ayudar a hacer de mí quien soy.

Maykel:

A mi familia, amigos y educandos que han contribuido de una manera u otra en mi formación profesional e intelectual.

Dedicatoria.

*A nuestros padres, familiares, amigos
y a todo el que ha estado ahí,
ofreciéndonos su incondicional ayuda.*

Resumen.

Este proyecto de tesis propone desarrollar una aplicación informática de carácter informativo, haciendo uso de la tecnología multimedia en la que se recopile y centralice toda la información que haga referencia a la ardua actividad que se ha llevado en la facultad 8 de la Universidad de las Ciencias Informáticas durante sus 3 años de fundada. El objetivo que persigue esta aplicación es hacer llegar esta importante información a todos los integrantes de esta facultad y a todas las personas interesadas en conocer esta historia. Esta aplicación permitirá reunir todos los datos que se han generado todos estos años, los que se encuentran de forma desorganizada, y que estudiantes, profesores y trabajadores, de una forma interactiva y más amena, incrementen el conocimiento sobre su facultad. Haciendo uso del lenguaje OMMMA-L como extensión del lenguaje UML se desarrollan los modelos para el desarrollo de la aplicación: se analizan los requisitos funcionales y no funcionales de la aplicación y se realiza el diseño, el análisis y la implementación de acuerdo a éstos para lograr un mayor entendimiento para los desarrolladores del sistema. Finalmente, utilizando un mecanismo de estimación de tiempo, desarrollo y esfuerzo, se estima la duración y esfuerzo para el desarrollo del sistema analizando su factibilidad.

Índice

Introducción.....	1
Situación problemática.....	1
Problema científico.....	2
Objeto de estudio.....	2
Campo de acción.....	2
Idea a defender.....	2
Objetivo general.....	2
Objetivos específicos.....	2
Tareas.....	3
Estructuración del contenido.....	3
Capítulo1: Fundamentación teórica del tema.....	4
1.1 Introducción.....	4
1.2 Estado del arte.....	4
1.2 Análisis de otras soluciones existentes.....	5
1.4 Descripción del objeto de estudio.....	5
1.4.1 Descripción general.....	5
1.4.2 Identificación de la audiencia.....	6
1.5 Conceptos asociados al dominio del problema.....	7
1.5.1 Texto.....	7
1.5.2 Imagen.....	7
1.5.3 Sonido.....	7
1.5.4 Video.....	7
1.5.5 Animación.....	7
1.5.6 Multimedia.....	8
1.5.7 Hipertexto.....	8
1.5.8 Hipermedia.....	8
1.6 Análisis del modelo de arquitectura de información utilizada.....	8
1.6.1 Principios y normas del diseño.....	8
1.6.2 Estándares de interfaz de la aplicación.....	9
1.6.3 Estándares de codificación.....	10
1.6 Conclusiones.....	10
Capítulo2: Tendencias y tecnologías actuales a considerar.....	12
2.1 Introducción.....	12
2.2 Tecnologías de la información y las comunicaciones (TICs).....	12
2.3 La multimedia.....	13
2.4 Antecedentes de la tecnología multimedia.....	14
2.5 Actualidad sobre los sistemas multimedia.....	15
2.6 Beneficios de las aplicaciones multimedia.....	16

2.7 Metodologías.....	17
2.8 Metodologías para la tecnología multimedia.....	17
2.9 Comparación entre metodologías tradicionales y ágiles.....	18
2.10 Metodologías tradicionales.....	18
2.10.1 RMM (Metodología de Administración de Relaciones).....	18
2.10.2 RUP (El Proceso Unificado del Rational)	20
2.11 Metodologías ágiles.....	24
2.11.1 XP (Programación Extrema).....	24
2.12 Herramientas Case.....	31
2.13 Lenguajes utilizados por las metodologías.....	32
2.13.1 OMMMA-L (Lenguaje de Modelado Orientado a Objetos de Aplicaciones Multimedia) como extensión del lenguaje UML (Lenguaje de Modelado Unificado).....	32
2.14 Herramientas de autor.....	37
2.14.1 Macromedia Flash.....	37
2.14.2 Revolution.....	38
2.14.3 Macromedia Director.....	39
2.14.4 Scala Multimedia MM200.....	39
2.14.5 Neobook.....	40
2.15 Lenguajes utilizados por las herramientas.....	41
2.15.1 XML (Lenguaje de Marcas Extensible).....	41
2.16 Metodología, herramientas y lenguajes de programación utilizados para el desarrollo de la multimedia.....	41
2.17 Factibilidad de trabajar en Flash con base de datos en XML.....	42
2.18 Conclusiones.....	42
Capítulo 3: Descripción de la solución propuesta.....	44
3.1 Introducción.....	44
3.2 Especificación del contenido.....	44
3.3 Modelo de dominio para el sistema propuesto.....	45
3.3.1 Modelo de dominio.....	46
3.4 Solución Propuesta.....	46
3.5 Descripción del sistema propuesto.....	47
3.6 Descripción de la funcionalidad del sistema propuesto.....	47
3.7 Requerimientos funcionales.....	47
3.8 Requerimientos no funcionales.....	48
3.9 Actores del sistema.....	49
3.10 Descripción de los casos de uso del sistema.....	49
3.11 Modelo de casos de uso del sistema.....	52
3.12 Modelo de navegación del sistema.....	53
3.13 Conclusiones.....	54
Capítulo 4: Construcción de la solución propuesta.....	56
4.1 Introducción.....	56

4.2 Modelo de clases del diseño.....	56
4.2.1 Diagrama de clases del diseño.	56
4.2.2 Diagramas de presentación.	60
4.3 Diagrama de jerarquía de clases.....	68
4.4 Modelo de implementación.....	69
4.4.1 Diagramas de componentes.	69
4.5 Modelo de despliegue.	73
4.5.1 Diagrama de despliegue.	73
4.6 Diagramas de secuencia.	73
4.7 Descripción de la organización de los archivos XML de la aplicación.	74
4.8 Descripción del archivo XML de la aplicación.	74
4.8.1 Descripción del archivo XML General.	74
4.8.2 Descripción del archivo XML Galería.....	76
4.9 Conclusiones.	76
Capítulo 5: Estudio de factibilidad.....	78
5.1 Introducción.	78
5.2 Planificación.....	78
5.3 Puntos de Casos de Uso.	78
5.4 Salario.	83
5.5 Costos.....	83
5.4 Beneficios tangibles e intangibles.....	84
5.4.1 Tangibles.....	84
5.4.2 Intangibles.....	84
5.5 Análisis de costos y beneficios.	84
5.6 Conclusiones.	85
Conclusiones.	86
Recomendaciones.....	87
Referencias bibliográficas.....	88
Bibliografía	90
Glosario de términos.	91

Introducción.

Al calor de la Batalla de Ideas surge la universidad más grande del desarrollo de software de la historia de Cuba, la Universidad de Ciencias Informáticas, ésta agrupa un gran número de jóvenes provenientes de todos los municipios del país, los cuales dieron el paso al frente ante esta importante tarea, priorizada por la máxima dirección del país e ideada por nuestro Comandante en Jefe. Con la ayuda de las tecnologías de la información, estos estudiantes se preparan no sólo en la esfera de la Informática sino también en la superación política, en la pedagogía, en los valores humanos y en todos los aspectos que los hace revolucionarios preparados y comprometidos con la Revolución.

Para una mayor organización del trabajo científico que se desarrolla en la universidad, se adoptó el método de dividirla en 6 facultades el primer año, posteriormente, cuando comenzó el tercer año se decidió dividirla en 10 facultades, todas con un perfil diferente de acuerdo a las necesidades del país. Cada facultad se desarrolla según su perfil, tiene su propia estructura organizativa y los estudiantes que pertenecen a ésta hacen que avance cada día, con actividades productivas y recreativas, eventos científicos y es así como se va escribiendo la historia de todas las facultades. Pero todos estos elementos están aislados y no aparecen registrados de forma conjunta para que puedan ser conocidos por todos, por lo que surge la necesidad de realizar un proyecto en el cual toda esta información sea centralizada.

El perfil de la facultad 8 es el software educativo, debido a esto los autores de este proyecto de tesis decidieron que la forma más indicada es la realización de una multimedia informativa, la cual debe ayudar de forma amena, instructiva y provechosa a ampliar el conocimiento de estudiantes, trabajadores y profesores acerca de la historia de la facultad 8 hasta la actualidad.

Situación problemática.

La facultad 8 cumplirá 4 años de fundada y aún existen estudiantes, profesores y trabajadores que desconocen muchos aspectos importantes relacionados con la historia de la facultad como el comportamiento en la docencia, en los eventos científicos, en actividades políticas y recreativas a nivel de universidad entre otros muchos aspectos que, debido a que estos elementos no han sido plasmados en un en algún formato que permita almacenarlos y poder ser mostrados a todos los integrantes de la facultad y a toda persona interesada en el desarrollo de la facultad.

Problema científico.

La no existencia de un producto con tecnología multimedia que sirva como soporte para mostrar toda la información referente a la historia de la facultad 8.

Objeto de estudio.

El objeto de estudio lo constituye el proceso de producción de software. El proceso de gestión y desarrollo de una aplicación con tecnología multimedia y de carácter informativo que permita presentar toda esta información acerca de la historia de la facultad 8.

Campo de acción.

Proceso de desarrollo de un producto con tecnología multimedia para centralizar la información y que sirva como apoyo para el conocimiento de la historia de la facultad.

Idea a defender.

Si se desarrolla una aplicación utilizando tecnología multimedia en la cual se refleje la historia de la facultad 8, se unificarían los datos que se han generado a partir de las actividades realizadas y se colocará toda esta importante información que en la actualidad se encuentra aislada, al alcance de todos los que deseen conocer acerca del arduo trabajo que se ha hecho para que esta facultad se encuentre entre una de las mejores en todos los índices que miden un buen desempeño en todos estos años.

Objetivo general.

Desarrollar un producto con tecnología multimedia que centralice toda la información referente a la facultad y de forma amena, sencilla y eficiente muestre esta información a todos los interesados en conocer el historial de la facultad.

Objetivos específicos.

Para dar solución al problema de este proyecto de trabajo se plantean los siguientes: Obtener toda la documentación requerida que será utilizada durante el desarrollo del software.

- Realizar una exhaustiva búsqueda y recolección de toda la información sobre actividades, eventos y demás acontecimientos realizados en la facultad.

- Plasmar toda la información de la facultad en una aplicación informática usando la tecnología multimedia para la divulgación de ésta.
- Apoyar a la elevación del nivel de conocimientos de los integrantes y de todos los interesados en la historia de la facultad 8.
- Permitir que la aplicación ofrezca facilidad de actualización de esta información a medida que pasen los años.

Tareas.

Para dar cumplimiento a los objetivos general y específicos se proponen las siguientes:

- Realizar entrevistas a los profesores, trabajadores y estudiantes fundadores de la facultad.
- Recopilar información de todo lo referente a eventos, actividades y todo lo que se haya desarrollado en la facultad.
- Revisar toda la documentación obtenida y clasificarla.
- Auto prepararnos en el trabajo con herramientas como flash, firework, photoshop, xml y otras.
- Realizar el diseño del producto.
- Implementar la aplicación utilizando la tecnología multimedia.
- Acudir a todos los medios posibles de información dentro de la universidad, para divulgar la existencia del producto.

Estructuración del contenido.

Capítulo 1. Fundamentación teórica del tema: En este capítulo se hace alusión al estado del arte de la temática planteada, la descripción del objeto de estudio, se analiza otra solución existente a la temática que se trata así como la identificación de la audiencia y los principios y normas de diseño que se aplicaran.

Capítulo 2. Tendencias y tecnologías actuales a considerar: En este capítulo se hace un análisis de las tendencias actuales al desarrollo de software Multimedia. Se analiza la metodología a utilizar para el análisis y diseño del sistema, teniendo en cuenta las facilidades que puede aportar al trabajo. Y finalmente se exponen las propuestas de herramientas a utilizar para el desarrollo de la aplicación.

Capítulo 3. Descripción de la solución propuesta: En este capítulo se realiza el modelado del negocio del sistema a través del modelo de dominio. Se realiza una descripción de la información que se presentará en la aplicación, así como de todos los requisitos funcionales y no funcionales que ésta deberá cumplir.

Capítulo 4. Construcción de la solución propuesta: Incluye la definición del modelo de clases del sistema. Describe los diagramas de secuencia del modelo del diseño para cada realización de los casos de uso. Muestra el diagrama de clases del diseño y la descripción de cada una de las clases así como la descripción de los archivos XML donde queda almacenada la información.

Capítulo 5. Estudio de la factibilidad: Incluye todo el estudio de la factibilidad del producto a desarrollar así como una evaluación de los beneficios tangibles e intangibles de la utilización del mismo.

Capítulo 1

Fundamentación teórica del tema.

1.1 Introducción.

En este capítulo se dará una breve explicación sobre los antecedentes que ha tenido el tema que se presenta en este proyecto de tesis, las inconveniencias que han presentado estos primeros trabajos presentados y cómo, con la propuesta de una aplicación desarrollada con tecnología multimedia que, después de una exhaustiva investigación y recopilación de datos, presente toda esta información a los estudiantes y trabajadores de toda la universidad.

También se realizará una descripción detallada del objeto de estudio que se persigue para llevar a cabo este proyecto: en primer lugar se hará una descripción general del objeto de estudio exponiendo qué se quiere lograr y cómo, luego se profundizará un poco más con el análisis de la audiencia que tendrá acceso a este producto y el conocimiento que tengan estas personas para el posterior uso de éste.

Por último se analizará la arquitectura de información que se utilizará durante el desarrollo del producto como son los principios y normas de diseño y los estándares de interfaz de aplicación y codificación que serán aplicados.

1.2 Estado del arte.

La facultad 8 cumplirá muy pronto 3 años de fundada y hasta estos días sólo se tienen referencias aisladas de todo lo que ha acontecido desde su surgimiento, las informaciones se han recogido de modo general en documentos, presentaciones, y aplicaciones de poca envergadura que están propagadas por toda la facultad. Todos los datos que se van generando a medida que pasan los años y se van realizando las distintas actividades como son los juegos deportivos, los festivales culturales, las copas de programación, actividades políticas o recreativas, quedan almacenados en la computadoras de la facultad pero no de forma conjunta.

La dirección de la FEU de la facultad ha convocado, en varias ocasiones, a concursos en lo que se puede participar presentando aplicaciones en las cuales se pueda encontrar referencias sobre el trabajo realizado

en estos años, logrando así que se hayan realizado varios sitios Web en los que se han recopilado muchos datos, sin embargo no han llegado a cumplir las expectativas, porque no todo el personal de la facultad y demás estudiantes y trabajadores de la universidad tienen acceso a estos sitios. A pesar de que se ha hecho un gran esfuerzo por parte de todo el personal de la facultad para coleccionar toda esta información, no se ha logrado desarrollar un sistema informático que reúna la intensa actividad que ha llevado el colectivo desde hace más de tres años, por lo que se hace imprescindible desarrollar un producto que cumpla estas condiciones.

1.2 Análisis de otras soluciones existentes.

En la actualidad existen numerosos trabajos creados con el fin de recopilar y mostrar de forma eficaz cualquier tipo de información capaz de revelar parte de la historia de la facultad 8, entre ellos los más destacados son los sitios Web, pero los mismos presentan carencia de información totalmente válida, ya que sus creadores no contaban con la información exacta, además de no extender su trabajo en todas las esferas que forman parte del desarrollo acelerado de la facultad.

Los sitios Web durante mucho tiempo se han mantenido en los primeros lugares entre los grandes medios de publicación de la información, pero las aplicaciones que utilizan la tecnología multimedia gana cada vez más terreno en el ciberespacio al incrementar la motivación y la interactividad con el usuario.

1.4 Descripción del objeto de estudio.

1.4.1 Descripción general.

La facultad 8 de la Universidad de las Ciencias Informáticas lleva casi 3 años de fundada y desde entonces, todos sus integrantes han hecho el mayor esfuerzo por hacer que ésta compita entre una de las mejores facultades de la Universidad y sea un ejemplo perenne en la disciplina y la producción. Todo este arduo trabajo ha generado una gran cantidad de información, la cual no se ha logrado, en todos estos años, centralizar y presentar en un formato digital de forma que todos los usuarios tengan acceso a ésta.

Por esta razón, el objeto de estudio de este proyecto de tesis es el proceso de gestión y desarrollo de una aplicación desarrollada con tecnología multimedia y de carácter informativo que permita la presentación de toda esta información acerca de la historia de la facultad 8.

Un proceso de desarrollo de software es el conjunto de técnicas y procedimientos que nos permiten conocer los elementos necesarios para definir un proyecto de software.(BENDAHAN 1997)

El proceso de gestión y desarrollo de esta aplicación utilizando la tecnología multimedia se logrará con la recopilación, centralización, clasificación y presentación de toda la información referente a la facultad, la cual será de carácter informativo. Esta permitirá el libre acceso a esta información que será presentada en forma de imágenes, texto y sonido y que de forma sencilla hará que el usuario se sienta atraído y motivado cuando interactúe con la misma, además será de gran importancia porque permitirá que todas las personas de la universidad y fuera de ésta conozca el gran esfuerzo realizado en la facultad para que marche adelante y tenga buenos resultados.

1.4.2 Identificación de la audiencia.

La audiencia que tendrá acceso a este producto esta formada mayormente por estudiantes de la Universidad de Ciencias Informáticas, los cuales tiene un alto grado de preparación en el ambiente de la informática, pero también contará con personas como profesores y trabajadores de la Universidad que no necesariamente sean graduados en la esfera de la informática y que no tienen un gran dominio de todas las habilidades para usar programas en una computadora. Por esta razón el producto se desarrollará de forma sencilla, que no requiera de grandes conocimientos para ser usado, dispondrá de una navegación global, o sea que: a la información se podrá acceder desde cualquier pantalla y en cualquier dirección, sin necesidad de tener que pasar por temas en los que el usuario no esté interesado. Se desarrollará de forma que el usuario pueda acceder a la aplicación todas las veces que sean necesarias para tomar la información que necesite.

Es un producto que es necesario que sea desarrollado debido a la falta de información que se tiene en toda la universidad de este importante tema de la historia de la facultad 8. También es válido aclarar que los usuarios finales tendrán algún conocimiento del tema que van consultar y eso les hará más fácil la comprensión de la información que aparezca en la aplicación.

1.5 Conceptos asociados al dominio del problema.

1.5.1 Texto.

“Un texto es una composición de signos codificado en un sistema de escritura (como un alfabeto) que forma una unidad de sentido. Su tamaño puede ser variable, desde una obra literaria como "El Quijote" al mensaje de volcado de pila de Windows NT. También es texto una composición de caracteres imprimibles (con grafía) generados por un algoritmo de cifrado que, aunque no tienen sentido para cualquier persona si puede ser descifrado por su destinatario texto claro original.”(WIKIPEDIA 2007f)

1.5.2 Imagen.

En computación, representación visual de cosas en forma digital. Suele estar representadas por miles de píxeles que, vistos en conjunto, forman una fotografía, un gráfico, o pueden estar hechas por vectores. Las imágenes pueden guardarse en distintos formatos gráficos, cada uno con distintas posibilidades y limitaciones. Entre los formatos más populares: BMP (gráfico/fotográfico), GIF (gráfico/animaciones), JPG (fotográfico), etc.(ALEGSA 2007)

1.5.3 Sonido.

“El sonido es la sensación, en el órgano del oído, producida por el movimiento ondulatorio, debido a los cambios de presión en un medio elástico y generados por el movimiento vibratorio de un cuerpo sonoro.”(WIKIPEDIA 2007e)

1.5.4 Video.

“Es una grabación realizada con un grabador de video o algún otro dispositivo que capture imágenes en movimiento. También se refiere al hecho de mostrar imágenes y textos en el monitor de una computadora. El adaptador de video es el responsable de enviar esas señales a los dispositivos proyectores.”(NETWORK 2006)

1.5.5 Animación.

“Archivo de imágenes con movimiento, o sea, un archivo compuesto por una secuencia de imágenes que, al ser reproducido por un software determinado presenta a la vista una sensación de movimiento. Existen distintos tipos de animaciones: de dos y tres dimensiones (2D y 3D).”(LORENZO 2007)

1.5.6 Multimedia.

“Es un sistema que utiliza más de un medio de comunicación al mismo tiempo en la presentación de la información, como el texto, la imagen, la animación, el video y el sonido.”(WIKIPEDIA 2007c)

1.5.7 Hipertexto.

“Un hipertexto es un documento digital que se puede leer de manera no secuencial. Un hipertexto tiene los siguientes elementos: secciones, enlaces y anclajes. Las secciones o nodos son los componentes del hipertexto o hiperdocumento. Los enlaces son las uniones entre nodos que facilitan la lectura secuencial o no secuencial del documento. Los anclajes son los puntos de unión entre nodos.”(WIKIPEDIA 2007a)

1.5.8 Hipermedia.

“Hipermedia no es más que la unión del hipertexto con otras tecnologías de tipo multimedia, es decir, gráficos, sonidos y video.”(GARCÍA 2004)

1.6 Análisis del modelo de arquitectura de información utilizada.

1.6.1 Principios y normas del diseño.

A medida que se han ido tornando más complejos los procesos de los sistemas informáticos también ha ido adquiriendo popularidad el diseño de sus interfaces. Este es un tema importante en el mercado de la informática, donde los usuarios cada día optan más por aplicaciones con calidad visual y que le brinden interactividad para lograr una mayor facilidad para obtener la información que desean y la mejor comprensión de ésta.

En el desarrollo de este sistema con tecnología multimedia, para el diseño de la interfaz de usuario se tendrán muchos aspectos en cuenta para brindar a éste una mayor calidad visual como:

- Los colores de fondo que se utilizarán brindarán al usuario una vista fresca y sencilla permitiendo una interfaz amigable y que el usuario logre una mayor concentración sobre el contenido al que accederá.
- Las fuentes que se utilizarán tienen un estilo sencillo y colores claros, acordes con el color de fondo que tendrá la aplicación.
- La información será correctamente distribuida por cada una de las pantallas evitando que estén sobrecargadas.

- Las opciones generales se mantendrán siempre en un mismo lugar evitando que el usuario se sienta incómodo o inseguro en el acceso a la información.
- La aplicación no requerirá de muchos conocimientos sobre el uso de programas en computadoras para su utilización, o sea, será una aplicación sencilla.

1.6.2 Estándares de interfaz de la aplicación.

Para el logro de una interfaz visual agradable a la vista del usuario, inciden varios factores, los cuales en conjunto ayudan a los programadores a desarrollar sistemas que cumplan las expectativas de los éstos. Entre estos factores se encuentran:

La navegación.

La multimedia se desarrollará utilizando una navegación global, lo cual permitirá que el usuario acceda a las pantallas que desee sin tener que regresar por los contenidos para volver a la pantalla principal. Además, desde ésta se podrá ir a cualquiera de los módulos de la multimedia así como a las galerías de imágenes.

La tipografía.

En la multimedia, como anteriormente se mencionó, la fuente escogida será de características sencillas, para conseguir un efecto agradable a la vista de los usuarios, tratándose éstos de personas que generalmente utilizan las computadoras con periodicidad, esta fuente será Tempus Sans ITC.

Sonido.

Los temas de la multimedia serán emotivos y variados, ya que están relacionados con la juventud de nuestra universidad, con su arduo trabajo político, sus actividades políticas y recreativas, sus eventos, la cultura y el deporte y todo lo que se ha realizado en la facultad, por lo que se tomó el acuerdo de poner un sonido de fondo a la multimedia emotivo, el cual tiene formato MP3, formato de buena calidad.

La imagen.

Se utilizará un dibujo para la pantalla principal de la multimedia, el cual expresará, de forma muy evidente el tema a tratar en la multimedia, por ser el símbolo que representa a la facultad. Las demás imágenes que se utilizarán en la multimedia son de formato JPEG, éstas han sido fotos que se han tomado a lo largo de todos los cursos.

El color.

Los usuarios que accederán a la multimedia son, generalmente personas adaptadas a usar computadoras y como ya se ha mencionado los colores a ser usados en la multimedia son colores frescos, para ayudar a la vista de estas personas. El color azul será el predominante. El color azul oscuro se aplicará al fondo y las tonalidades que se le darán a los botones en contraste con el blanco brindarán una vista refrescante al usuario, sin embargo, los colores del dibujo que aparecerá a la izquierda en todas las pantallas tendrá los colores más vivos para que el usuario siempre recuerde el tema de la multimedia.

1.6.3 Estándares de codificación.

Uno de los instrumentos que facilitan la tarea de la codificación es la adopción de estándares de estilo y codificación. El uso de estos estándares tiene innumerables ventajas, entre ellas:

- Asegurar la legibilidad del código entre distintos programadores, facilitando el debugging del mismo.
- Proveer una guía para el encargado de mantenimiento/actualización del sistema, con código claro y bien documentado.
- Facilitar la portabilidad entre plataformas y aplicaciones.

Un tema muy importante son los comentarios, se aconseja el uso de comentarios en línea para facilitar la comprensión del código, sobre todo en procedimientos complejos. Los comentarios pueden ser con fin documental o bien como 'ayuda-memoria', se recomienda utilizar los estilos de C (`/* */`) y C++ (`//`). (DIGITAL 2006)

El estándar de codificación que se utilizará en el desarrollo de la multimedia es el que propuesto por el compilador de Macromedia Flash 8 para el lenguaje ActionScript.

1.6 Conclusiones.

Con el análisis de este capítulo se demuestra las ventajas de escoger la tecnología multimedia como la mejor opción para la presentación de la historia de la facultad sobre otras opciones como sitios Web y presentaciones en Power Point.

Con la descripción del objeto de estudio se tiene una visión más clara de que se quiere presentar toda esta información coleccionada en una aplicación desarrollada con tecnología multimedia, la cual ayudará a

ampliar los conocimientos de todas las personas de la facultad 8 y de toda la universidad acerca de su historia y llevar a cabo el desarrollo de este sistema mediante un proceso de desarrollo de software. También se definieron los principios y normas de diseño y los estándares de interfaz de aplicación y de codificación que tendrá el sistema.

Capítulo 2

Tendencias y tecnologías actuales a considerar.

2.1 Introducción.

La tecnología multimedia es el uso de diversos medios para el transporte de la información y desde que surgió esta tecnología, ha hecho un gran impacto en la esfera de la comunicación mundial. La información se ha presentado en varios formatos desde el surgimiento de las computadoras, sin embargo la multimedia realza la experiencia del usuario y la hace más fácil y más rápida para tomar la información. Es por esta razón que en este capítulo se analizará más detenidamente lo que representa la multimedia y toda su historia hasta nuestros días.

La creación de un producto informático siempre está regida por metodologías que ayudan a desarrollar un producto de forma sistemática y de mejor calidad. También para las tecnologías multimedia existen metodologías que, a medida que van pasando los años y las multimedias se tornan más sofisticadas, ellas, al igual, evolucionan y ofrecen más eficiencia a la hora de la construcción del software. Este tema es de suma importancia para el proyecto de tesis de desarrollo de una multimedia que brinde toda la información de la facultad, se profundizará en el estudio de estas metodologías y su utilización.

Las herramientas que se usan en el desarrollo de las multimedias son las llamadas herramientas de autor y éstas, al igual que las metodologías, han evolucionado muy rápidamente. Muchas de estas herramientas ofrecen muchas ventajas y en el curso del capítulo se estudiarán para tener un mayor conocimiento de ellas y sobre como utilizarlas.

2.2 Tecnologías de la información y las comunicaciones (TICs).

Las TICs (Tecnologías de la Información y Comunicaciones) son las tecnologías que se necesitan para la gestión y transformación de la información, y muy en particular el uso de ordenadores y programas que permiten crear, modificar, almacenar, proteger y recuperar esa información.

Las TICs agrupan un conjunto de sistemas necesarios para administrar la información, y especialmente los ordenadores y programas necesarios para convertirla, almacenarla, administrarla, transmitirla y encontrarla. Los primeros pasos hacia una sociedad de la información se remontan a la invención del telégrafo eléctrico, pasando posteriormente por el teléfono fijo, la radiotelefonía y, por último, la televisión, Internet, la telecomunicación móvil y el GPS (Sistema de Posicionamiento Global) pueden considerarse como nuevas tecnologías de la información y las comunicaciones.

La revolución tecnológica que vive la humanidad actualmente es debida en buena parte a los avances significativos en las tecnologías de la información y las comunicaciones. Los grandes cambios que caracterizan esencialmente esta nueva sociedad son: la generalización del uso de las tecnologías, las redes de comunicación, el rápido desenvolvimiento tecnológico y científico y la globalización de la información.

2.3 La multimedia.

Uno de los términos más novedosos de la nueva era de la informática es la multimedia, ésta tiene alcance a numerosos campos afines y distintos de la informática. Su surgimiento está remontado a principios de la década de los 80 y su desarrollo ha sido muy acelerado porque abre las puertas a la comunicación mundial a gran escala. Nace con la diversificación de las formas de comunicación entre las personas y con el desarrollo de las tecnologías en las cuales está fuertemente apoyada.

La tecnología multimedia es el sistema informático que se dedica a presentar texto, gráficos, video, animación y sonido. Además de combinar los elementos anteriores también debe interaccionar con el usuario, no deben ser presentaciones estáticas sino una experiencia interactiva y de variada información para el que haga uso del sistema.

Estos sistemas pueden almacenar información en forma textual, visual o auditiva y ejecutar paralelamente algún tipo de operación sobre la misma y suelen estar almacenados en CD-ROMs al igual que pueden ocupar lugar en páginas Web.

Es común encontrar aplicaciones con tecnología multimedia para juegos, pero las más importantes son las aplicaciones que presentan programas de aprendizaje un ejemplo son los diferentes cursos que se desarrollan en esta forma, y las aplicaciones de material de referencia, en las que se puede encontrar diversidad de información.

La recuperación de la información en los sistemas multimedia también ha ampliado las formas en que el usuario busca y recupera la información que necesita.

Ejemplos de aplicaciones que utilizan tecnología multimedia son: diccionarios, enciclopedias, publicaciones electrónicas, servicios de información referencial, etc.(FRANQUET 2003)

2.4 Antecedentes de la tecnología multimedia.

El término multimedia viene utilizándose desde la década de los sesenta para describir producciones y procesos muy diversos; desde un conjunto de proyectores de diapositivas con grabaciones de audio o vídeo hasta sistemas más sofisticados basados en reproductores de imágenes y sonidos en sincronización temporal. Nos referimos a artefactos creados antes de la popularización del ordenador y los soportes ópticos. En este sentido evolutivo, la capacidad multimedia en las computadoras constituye solamente la última etapa de un largo camino de progresos técnicos y tecnológicos. Todos los grupos humanos han construido herramientas y técnicas con el propósito innato de expresarse, de comunicarse, transmitir sus ideas y trascender los límites del espacio y del tiempo. A finales de la década de los ochenta, se empezó a trabajar en el Estado español con sistemas interactivos basados en los soportes ópticos. A finales de la década de los ochenta, se empezó a trabajar en el estado español con sistemas interactivos basados en los soportes ópticos. El videodisco analógico consiguió cierta aceptación en países como EE.UU. o Japón, en ámbitos docentes o profesionales, pero sobre todo se utilizó para contener películas cinematográficas. En 1992 se consolida el soporte del CD-ROM y augura una buena perspectiva para el sector emergente de las aplicaciones multimedia, aunque finalmente esto no será así. La escasez de ordenadores con capacidades multimedia entre la población, la proliferación de aplicaciones de poca calidad, las dificultades de comercialización y distribución, entre otros factores, no permitieron conseguir una buena penetración de obras culturales o educativas entre los usuarios.

Alrededor de los años 80, en algunos de los países más desarrollados en la industria de la informática comenzaron a incursionar en el desarrollo de las tecnologías multimedias, algunos de éstos fueron España, Japón y EE.UU., en los cuales el desarrollo de éstas fue quebrado debido a la escasez de herramientas que ayudaran a su desarrollo, a la proliferación de productos de mala calidad y a las dificultades de comercialización y distribución entre otros factores que no permitieron que los usuarios disfrutaran del beneficio de los servicios que brinda. Sin embargo hubo producciones lanzadas por potentes empresas multinacionales que sí tuvieron éxito, como fue la Enciclopedia Encarta de Microsoft y algunos juegos, a diferencia de otras que fueron un rotundo fracaso como lo fue la Enciclopedia Británica. A partir de este momento las grandes empresas comenzaron a tomar más en cuenta la producción de estos sistemas y se

hicieron contratos con empresas como IBM para comercializar y distribuir sus productos offline debido a que, se debía hacer una colaboración entre los productores de multimedias y una empresa de equipos informáticos capaces de soportar y gestionar el desarrollo de éstas. Hubo un gran auge en la producción y a pesar de verse afectada por diferentes problemas los primeros soportes fueron los CD-ROMs, además también influyó negativamente la falta de canales adecuados de comercialización y de servicios que permitieran que el usuario pudiera interactuar con el producto. Más tarde aparecieron los DVD que dieron un gran paso de avance para el soporte de los sistemas producidos.(FRANQUET 2003)

2.5 Actualidad sobre los sistemas multimedia.

Actualmente la mayoría de los usuarios en el mundo, acceden desde sus propios hogares a la red de redes y se ha hecho primordial el hecho de trasladar a la red, todo lo referente a trabajo y ocio, para brindar a las personas más comodidad y darles mayor grado de importancia y provecho a tan importante recurso, trayendo así una nueva era para las empresas que se basan en la producción de estos sistemas informáticos. El surgimiento de Internet, el desarrollo de éste y de la banda ancha han proporcionado un progreso vertiginoso en el sector de la multimedia. En los últimos años las grandes potencias han hecho de la tecnología el principal medio para el desarrollo de una sociedad, un fiel ejemplo lo representan las aplicaciones con tecnología multimedia que ya se pueden encontrar en los teléfonos móviles.

Nuestro país no queda rezagado en el desarrollo de productos multimedias y una de las primeras empresas de más prestigio ha sido la Consultoría BIOMUNDI/IDICT, fundada el 15 de agosto de 1992, específicamente la coordinación de Servicios de Inteligencia Empresarial que cuenta con un grupo de especialistas en diferentes disciplinas y brinda servicios de información y de Inteligencia Empresarial de alto valor dedicada a la identificación y solución de problemas de entidades cubanas y extrajeras. Esta brinda servicios de información como:

- Búsquedas de Información.
- Monitoreos de Información.
- Compendios Informativos.

Los Compendios Informativos son entregados en formato multimedia por esta Consultoría. En el 2003 se le hicieron 50 solicitudes de este tipo de producto, índice que lo convirtió en la organización más destacada. Las temáticas desarrolladas abarcaron un amplio espectro, desde el sector de las Biociencias hasta la esfera militar.

La tecnología multimedia podría describirse como la integración de las cualidades informativas y comunicativas de texto, sonido, imágenes fijas y video con la interactividad de la computadora, para comunicar, informar, crear habilidades o simplemente entretener. Los hiperenlaces y la posibilidad de conectar piezas de información en una red ofrecen nuevas posibilidades a los productores de información.

La parte “hiper” que es la que convierte a la multimedia en hipermedia es la que provee a esta tecnología en la verdadera revolución en la forma de presentar y recuperar la información. Es por ello que estas tecnologías brillan especialmente en el área de los servicios y productos de información, los cuales al incorporar estas técnicas adquieren un alto valor agregado: diccionarios, enciclopedias, publicaciones electrónicas en general, información referencial y servicios de información tipo quioscos, son ejemplos de tipos de aplicaciones multimedia.

Otra empresa iniciadora cubana dedicada a la producción de aplicaciones multimedia es CITMATEL (Empresa de Tecnologías de la Información y Servicios Telemáticos Avanzados). Tiene experiencia en la producción de ediciones electrónicas lo que ha permitido a la Editorial Multimedia de CITMATEL, desarrollar las más diversas obras sobre CD-ROM, DVD y WEB. Tiene en su aval la creación del primer CD-ROM y el primer DVD producidos en Cuba. Se encuentra entre las más importantes editoriales cubanas, con una amplia y variada edición de títulos reconocidos por su calidad, muchos de ellos premiados en ferias y exposiciones nacionales e internacionales. Ejemplo de algunos de sus productos son: “Deporte Cubano. Éxitos, secreto y estrellas.” que permite conocer los triunfos cubanos y a los protagonistas de tales hazañas y el por qué de sus éxitos, también podemos encontrar “CubAcusa” donde se podrá acceder a una cronología de los años 1959 al 2002 que contiene hechos terroristas realizados contra Cuba. El desarrollo de estas empresas es un indicador que mide el esfuerzo de nuestro país en el tema de las tecnologías y su adecuado uso en la sociedad.(CÁRDENAS 2003)

2.6 Beneficios de las aplicaciones multimedia.

- **Impacto:** al reunir efectos llamativos como sonidos, imágenes, animaciones en tercera dimensión para crear presentaciones de excelente calidad.
- **Flexibilidad:** el material digital puede ser fácil y rápidamente actualizado y presentado a través de varios medios.
- **Credibilidad:** al utilizar tecnología de punta proyecta la imagen de una empresa hacia nuevas dimensiones de comunicación.

- **Costo-Beneficio:** al aprovechar todos sus recursos e incorporarlos a la presentación multimedia. Los mismos recursos pueden ser utilizados para diversas finalidades y a través de diversos medios y se ahorran recursos en materiales impresos difíciles de actualizar y presentándola en innumerables ocasiones sin ninguna restricción.
- **Interactividad:** el usuario tienen el control y puede acceder a la información precisa que está buscando, aprendiendo de los temas de su interés e ignorando aquellos que domina y no está interesado, haciéndolo a su propio ritmo y en el momento en que lo decida.(TECHNOLOGY 2007)

2.7 Metodologías.

Metodología se refiere a los métodos de investigación en una ciencia. La metodología se entiende como una parte del proceso de investigación que permite sistematizar los métodos y las técnicas necesarios para llevarlo a cabo. Los métodos son vías que facilitan el descubrimiento de conocimientos seguros y confiables para solucionar los problemas que la vida nos plantea. Es el conjunto de procedimientos, técnicas, herramientas y soporte de documentos que guía a los desarrolladores de software a realizar un producto de calidad de acuerdo con las especificaciones del cliente. Las metodologías ayudan a desarrollar un producto de forma sistemática, a lograr mejores aplicaciones, a guiar al equipo de desarrolladores a llevar un proceso estándar. Existen herramientas software que brindan soporte a la aplicación de las técnicas de determinada metodología.

2.8 Metodologías para la tecnología multimedia.

Las metodologías aplicadas a los sistemas con tecnologías multimedias pueden ser tradicionales o ágiles. El esquema tradicional en el desarrollo de software probó ser efectivo y necesario en proyectos de gran tamaño respecto a tiempo y recursos. Las metodologías tradicionales se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir y las herramientas y notaciones que se usarán. Debido a su antigüedad éstas han recibido críticas por su carácter excesivamente burocrático, además, este enfoque no resulta ser el más adecuado para proyectos actuales donde el entorno del sistema puede ser muy cambiante y donde se exija reducir drásticamente el tiempo de desarrollo pero manteniendo alta calidad. Ante estas dificultades de tiempo y flexibilidad que presentan las tradicionales, emergen las metodologías ágiles como posible respuesta para llenar ese vacío en el desarrollo de productos informáticos.

Las metodologías ágiles están orientadas para proyectos de menor envergadura, o sea proyectos más pequeños. Estas constituyen una solución a los problemas que presentan las metodologías tradicionales debido a que ofrecen la solución a la medida para un entorno en el que los requisitos pueden ser variables y aportan una elevada simplificación en tiempo y recursos sin renunciar a las prácticas esenciales para asegurar la calidad del producto.(CANÓS 2004)

2.9 Comparación entre metodologías tradicionales y ágiles.

Las metodologías tradicionales están basadas en normas provenientes de estándares seguidos por el entorno de desarrollo, ofrecen resistencia a los cambios que puedan ocurrir en el proyecto, tienen un proceso controlado y con numerosas políticas, existe un contrato prefijado, el cliente no pertenece al equipo de desarrollo sino que interactúa con éste mediante reuniones, se trabaja en grandes grupo distribuidos, posee mucho roles y artefactos, la arquitectura del software es esencial y se expresa mediante modelos mientras que las metodologías ágiles están basadas en cálculos provenientes de prácticas de producción de código, están preparadas especialmente para cambios en el proyecto, tienen un proceso menos controlado, o sea de menos principios, no existe un contrato tradicional, es más flexible, el cliente es parte del equipo de desarrollo, se trabaja en pequeños grupos y concentrados en un mismo sitio, necesitan pocos roles y artefactos y hace menos énfasis en la arquitectura del software.(CANÓS 2004)

2.10 Metodologías tradicionales.

2.10.1 RMM (Metodología de Administración de Relaciones).

El método RMM fue la primera metodología para el diseño de multimedia y fue creada en 1995. Es un método para hipermedia que cubre todo el ciclo de desarrollo, desde el estudio de factibilidad hasta la evaluación del sistema, aunque sólo propone actividades y productos concretos para las fases de análisis y diseño. Esta orientada a problemas con datos dinámicos que cambian con mucha frecuencia, más que a entornos estáticos.

Esta metodología propone un proceso basado en 7 fases o etapas en las que el diseñador va modelando la estructura de la aplicación y las posibilidades de navegación de la misma. Está basada en el modelo Entidad-Relación y en HDM (Método de Diseño Hipermedia). Partiendo de ellos define un nuevo modelo, el

RMDM (Modelo de Datos de Administración de Relaciones), que propone un lenguaje que permitirá describir los objetos del dominio, sus interrelaciones y los mecanismos de navegación hipertexto de la aplicación. Los objetos del dominio se definen con la ayuda de entidades, atributos y relaciones asociativas. El modelo introduce el concepto de *slice* (trozo) con el fin de modelar los aspectos unidos a la presentación de las entidades. Un slice corresponde a un subconjunto de atributos de una misma entidad destinados a ser presentados de forma agrupada. Por otro lado, es necesario representar los enlaces. Aunque RMM toma a HDM como base, define sus propios enlaces y los divide en varios grupos:

- Enlaces no condicionales, pueden ser unidireccionales o bidireccionales. Serían los enlaces en los que, al partir del origen se pasa al destino sin necesidad de que el usuario indique ninguna condición.
- Enlaces condicionales, en ellos el usuario tiene que indicar alguna condición específica. Dependiendo de dicha condición el usuario irá a un slice u otro.
- Rutas guiadas, se activan automáticamente ante un evento o pasado un tiempo, el usuario no debe hacer nada.

Fases de RMM.

RMM propone un proceso dividido en etapas para el desarrollo de las aplicaciones multimedia. En este apartado se verán muy brevemente estas fases y sus objetivos.

Fase 1- Realizar el modelo E-R.

En esta fase se debe obtener un modelo Entidad-Relación del sistema, sin necesidad de entrar en detalles de navegación o de presentación al usuario. Se debe actuar de la misma forma que se actuaría para obtener un modelo E-R de una aplicación software clásica.

Fase 2- Realizar los diseños de slice.

Para cada entidad detectada en la fase anterior, se debe definir un diagrama de slices. Se deben detectar los slices para esa entidad, es decir, cómo se van a presentar los atributos de la entidad al usuario.

Fase 3- Diseñar la navegación.

Una vez que ya se han definido los slices, se debe diseñar cómo se pasará de una entidad a otra, es decir, hay que enriquecer el modelo Entidad-Relación obtenido en la primera fase con los enlaces entre entidades. En principio en el modelo no sería necesario al tener una única entidad.

Fase 4- Definir el protocolo de conversión.

En esta fase se debe definir el proceso a seguir para pasar del modelo RMDM a la plataforma de desarrollo concreta. En principio no se propone ninguna técnica estándar a seguir para ello.

Fase 5- Diseñar la interfaz.

En esta fase se diseñan las pantallas tal y como se van a mostrar al usuario. Por regla general cada slice se va a corresponder con una pantalla. En esta fase ya es necesario entrar en aspectos concretos del lenguaje de programación que se va a usar.

Fase 6- Implementar la aplicación.

En base al protocolo establecido en la fase 4 y al modelo RMDM obtenido, se implementa el sistema.

Fase 7- Probar la aplicación.

Una vez que se obtiene la aplicación ejecutable, se deben realizar las pruebas de funcionamiento a la misma. Para ello es necesario definir el test de prueba y estudiar sus resultados.

RMM sube un paso más con respecto a HDM. Propone una metodología basada en el modelo E-R y en HDM para representar las aplicaciones multimedia. Debido a esto es precisamente por lo que no ha tenido demasiada difusión. Es una técnica que se basa en el E-R cuando en su época todas las tendencias se dirigían a la orientación a objetos.

Algo ventajoso y que hace interesante a RMM es que propone un proceso estructurado y definido a seguir para el desarrollo de estas aplicaciones. En este proceso, sin embargo, se echan en falta las primeras etapas a tener en cuenta en cualquier proceso de desarrollo software, como la captura de requisitos. Además, el proceso que ofrece es demasiado abierto en sus fases como para considerarse como una herramienta de desarrollo adecuada, puesto que en la única fase en la que indica una técnica es en la que se hace uso del modelo RMDM, las otras fases quedan abiertas a la opción del diseñador. Y los mecanismos de acceso a la información son excesivamente simples por lo que es válido para problemas con pocas entidades, si éste llega a tener muchas no es eficiente.(CUARESMA 2001)

2.10.2 RUP (El Proceso Unificado del Rational)

Es un proceso de desarrollo de software que, junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico Proceso Unificado. El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

Sus principios de desarrollo son:

1- Adaptar el proceso.

El proceso deberá adaptarse a las características propias del proyecto u organización. El tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico.

2- Balancear prioridades.

Los requerimientos de los diversos inversores pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un balance que satisfaga los deseos de todos.

3- Colaboración entre equipos.

El desarrollo de software no lo hace una única persona sino múltiples equipos. Debe haber una comunicación fluida para coordinar requerimientos, desarrollo, evaluaciones, planes, resultados, etc.

4- Demostrar valor iterativamente.

Los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto.

5- Elevar el nivel de abstracción.

Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes 4GL o esquemas (frameworks) por nombrar algunos. Esto previene a los ingenieros de software ir directamente de los requisitos a la codificación de software a la medida del cliente. Un nivel alto de abstracción también permite discusiones sobre diversos niveles arquitectónicos. Éstos se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con UML.

6- Enfocarse en la calidad.

El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción.

Flujos de trabajo:

- **Modelamiento del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.

- **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- **Instalación:** Produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

El ciclo de vida de RUP se caracteriza por:

1- Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se 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 (cómo se llevan a cabo).

2- Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los Casos de Uso relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML.

3- Iterativo e Incremental: Aunque se puede sugerir que los flujos de trabajo se desarrollan en cascada, la "lectura" del gráfico tiene que ser vertical y horizontal (Anexo 1). RUP 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. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración.

Es práctico dividir el trabajo en partes más pequeñas o miniproyectos. Cada miniproyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los

incrementos, al crecimiento del producto. Cada iteración se realiza de forma planificada, es por eso que se dice que son miniproyectos.

Fases:

El ciclo de vida RUP es una implementación del desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semi-ordenadas. El ciclo de vida organiza las tareas en fases e iteraciones.

El RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante:

- **Concepción:** se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.
- **Elaboración:** se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- **Construcción:** se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.
- **Transición:** se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

Sus principales características son:

- 1- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- 2- Pretende implementar las mejores prácticas en Ingeniería de Software.
- 3- Desarrollo iterativo.
- 4- Administración de requisitos.
- 5- Uso de arquitectura basada en componentes.
- 6- Control de cambios.
- 7- Modelado visual del software.
- 8- Verificación de la calidad del software.

El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

La metodología RUP es más apropiada para proyectos grandes, dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas. En proyectos pequeños, es posible que no sea posible cubrir los costos de dedicación del equipo de profesionales necesario. (WIKIPEDIA 2007d)

2.11 Metodologías ágiles.

2.11.1 XP (Programación Extrema).

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Esta se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Las características esenciales de XP se dividen en tres partes: historias de usuarios, roles, procesos y prácticas.

Historias de Usuario.

Son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración.

Respecto a la información contenida en la historia de usuario, existen varias plantillas sugeridas pero no existe un consenso al respecto. En muchos casos sólo se propone utilizar un nombre y una descripción o sólo una descripción, o una estimación de esfuerzo en días. Kent Beck, el padre de XP, en su libro presenta un ejemplo de ficha en la cual pueden reconocerse los siguientes contenidos: fecha, tipo de actividad (nueva, corrección, mejora), prueba funcional, número de historia, prioridad técnica y del cliente, referencia a otra historia previa, riesgo, estimación técnica, descripción, notas y una lista de seguimiento con la fecha, estado de las cosas por terminar y comentarios.

Una de las interrogantes que también se presenta cuando se utilizan casos de uso es ¿cuál es el nivel de granularidad adecuado para una historia de usuario? Se dice que depende de la complejidad del sistema, debe haber al menos una historia por cada característica importante, y propone realizar una o dos historias por programador por mes. Si se tienen menos, probablemente sea conveniente dividir las historias, si se tienen más, lo mejor es disminuir el detalle y agruparlas. Para efectos de planificación, las historias pueden ser de una a tres semanas de tiempo de programación para no superar el tamaño de una iteración. No hay que preocuparse si en un principio no se identifican todas las historias de usuario. Al comienzo de cada iteración estarán registrados los cambios en las historias de usuario y según eso se planificará la siguiente iteración.

Roles

Los roles en XP son:

- 1- Programador: El programador escribe las pruebas unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.
- 2- Cliente: Escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio. El cliente es sólo uno dentro del proyecto, pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema.
- 3- Encargado de pruebas: Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- 4- Encargado de seguimiento: El encargado de seguimiento proporciona realimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones. También realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.
- 5- Entrenador: Es responsable del proceso global. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

6- Consultor: Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.

7- Gestor: Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente, creando las condiciones adecuadas. Su labor esencial es de coordinación.

Proceso XP.

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste en los siguientes pasos:

- 1- El cliente defina el valor de negocio a implementar.
- 2- El programador estime el esfuerzo necesario para su implementación.
- 3- El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
- 4- El programador construye ese valor de negocio.
- 5- Volver al paso 1.

En todas las iteraciones de este ciclo, tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la entrega, Iteraciones, Producción, Mantenimiento y Muerte del proyecto.

Fase I: Exploración.

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

Fase II: Planificación de la entrega.

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días. Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración.

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación.

Fase III: Iteraciones.

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible, ya que es el cliente quien decide qué historias se implementarán en cada iteración para maximizar el valor de negocio. Al final de la última iteración el sistema estará listo para entrar en producción.

Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores.

Fase IV: Producción.

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación por ejemplo, durante la fase de mantenimiento.

Fase V: Mantenimiento.

Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

Fase VI: Muerte del proyecto.

Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

Prácticas XP.

La principal suposición que se realiza en XP es la posibilidad de disminuir el costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de algunas prácticas.

El mayor beneficio de las prácticas se consigue con su aplicación conjunta y equilibrada puesto que se apoyan unas en otras. Algunas de estas prácticas son:

- 1- El juego de la planificación. Hay una comunicación frecuente el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas de cada iteración. Esta práctica se puede ilustrar

como un juego, donde existen dos tipos de jugadores: Cliente y Programador. El cliente establece la prioridad de cada historia de usuario, de acuerdo con el valor que aporta para el negocio. Los programadores estiman el esfuerzo asociado a cada historia de usuario. Se ordenan las historias de usuario según prioridad y esfuerzo, y se define el contenido de la entrega y/o iteración, apostando por enfrentar lo de más valor y riesgo cuanto antes. Este juego se realiza durante la planificación de la entrega, en la planificación de cada iteración y cuando sea necesario reconducir el proyecto.

2- Entregas pequeñas. Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más 3 meses.

3- Metáfora. En XP no se enfatiza la definición temprana de una arquitectura estable para el sistema. Dicha arquitectura se asume evolutiva y los posibles inconvenientes que se generarían por no contar con ella explícitamente en el comienzo del proyecto se solventan con la existencia de una metáfora. El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema. Martín Fowler explica que la práctica de la metáfora consiste en formar un conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema. Este conjunto de nombres ayuda a la nomenclatura de clases y métodos del sistema.

4- Diseño simple. Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente. Kent Beck dice que en cualquier momento el diseño adecuado para el software es aquel que: supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos.

5- Pruebas. La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse. En este contexto de desarrollo evolutivo y de énfasis en pruebas constantes, la automatización para apoyar esta actividad es crucial.

6- Refactorización. La refactorización es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. La refactorización mejora la estructura interna del código sin alterar su

comportamiento externo. Robert Martín señala que el diseño del sistema de software es una cosa viviente. No se puede imponer todo en un inicio, pero en el transcurso del tiempo este diseño evoluciona conforme cambia la funcionalidad del sistema. Para mantener un diseño apropiado, es necesario realizar actividades de cuidado continuo durante el ciclo de vida del proyecto. De hecho, este cuidado continuo sobre el diseño es incluso más importante que el diseño inicial. Un concepto pobre al inicio puede ser corregido con esta actividad continua, pero sin ella, un buen diseño inicial se degradará.

7- Programación en parejas. Toda la producción de código debe realizarse con trabajo en parejas de programadores. Según Cockburn y Williams en un estudio realizado para identificar los costos y beneficios de la programación en parejas, las principales ventajas de introducir este estilo de programación son: muchos errores son detectados conforme son introducidos en el código (inspecciones de código continuas), por consiguiente la tasa de errores del producto final es más baja, los diseños son mejores y el tamaño del código menor (continua discusión de ideas de los programadores), los problemas de programación se resuelven más rápido, se posibilita la transferencia de conocimientos de programación entre los miembros del equipo, varias personas entienden las diferentes partes del sistema, los programadores conversan mejorando así el flujo de información y la dinámica del equipo, y finalmente, los programadores disfrutan más su trabajo. Dichos beneficios se consiguen después de varios meses de practicar la programación en parejas. En los estudios realizados por Cockburn y Williams este lapso de tiempo varía de 3 a 4 meses.

8- Propiedad colectiva del código. Cualquier programador puede cambiar cualquier parte del código en cualquier momento. Esta práctica motiva a todos a contribuir con nuevas ideas en todos los segmentos del sistema, evitando a la vez que algún programador sea imprescindible para realizar cambios en alguna porción de código.

9- Integración continua. Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día. Todas las pruebas son ejecutadas y tienen que ser aprobadas para que el nuevo código sea incorporado definitivamente. La integración continua a menudo reduce la fragmentación de los esfuerzos de los desarrolladores por falta de comunicación sobre lo que puede ser reutilizado o compartido. Martín Fowler afirma que el desarrollo de un proceso disciplinado y automatizado es esencial para un proyecto controlado, el equipo de desarrollo está más preparado para modificar el código cuando sea necesario, debido a la confianza en la identificación y corrección de los errores de integración.

10- 40 horas por semana. Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo. Los proyectos que requieren trabajo extra para intentar cumplir con los plazos suelen al final ser entregados con retraso. En lugar de esto se puede realizar el juego de la planificación para cambiar el ámbito del proyecto o la fecha de entrega.

11- Cliente in-situ. El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Gran parte del éxito del proyecto XP se debe a que es el cliente quien conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita, ya que esta última toma mucho tiempo en generarse y puede tener más riesgo de ser mal interpretada. Se indica que se debe pagar un precio por perder la oportunidad de un cliente con alta disponibilidad. Algunas recomendaciones propuestas para dicha situación son las siguientes: intentar conseguir un representante que pueda estar siempre disponible y que actúe como interlocutor del cliente, contar con el cliente al menos en las reuniones de planificación, establecer visitas frecuentes de los programadores al cliente para validar el sistema, anticiparse a los problemas asociados estableciendo llamadas telefónicas frecuentes y conferencias, reforzando el compromiso de trabajo en equipo.

La mayoría de las prácticas propuestas por XP no son novedosas, sino que en alguna forma ya habían sido propuestas en ingeniería del software e incluso, demostrado su valor en la práctica para un análisis histórico de ideas que sirven como antecedentes a las utilizadas por las metodologías ágiles. El mérito de XP es integrarlas de una forma efectiva y complementarlas con otras ideas desde la perspectiva del negocio, los valores humanos y el trabajo en equipo.(LETELIER and PENADÉS 2005)

2.12 Herramientas Case.

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML1.1. Esta herramienta propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del

sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software.

Desarrollo iterativo.

Rational Rose utiliza un proceso de desarrollo iterativo controlado, donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Cada iteración comienza con una primera aproximación del análisis, diseño e implementación para identificar los riesgos del diseño, los cuales se utilizan para conducir la iteración, primero se identifican los riesgos y después se prueba la aplicación para que éstos se hagan mínimos. Cuando la implementación pasa todas las pruebas que se determinan en el proceso, ésta se revisa y se añaden los elementos modificados al modelo de análisis y diseño. Una vez que la actualización del modelo se ha modificado, se realiza la siguiente iteración.

Trabajo en Grupo.

Rose permite que haya varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo. También es posible descomponer el modelo en unidades controladas e integrarlas con un sistema para realizar el control de proyectos que permite mantener la integridad de dichas unidades.

Generador de Código.

Se puede generar código en distintos lenguajes de programación a partir de un diseño en UML.

Ingeniería Inversa.

Rational Rose proporciona mecanismos para realizar la denominada ingeniería inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño. (MARTÍNEZ 1998)

2.13 Lenguajes utilizados por las metodologías.

2.13.1 OMMMA-L (Lenguaje de Modelado Orientado a Objetos de Aplicaciones Multimedia) como extensión del lenguaje UML (Lenguaje de Modelado Unificado).

Lenguaje de Modelado Unificado.

El Lenguaje de Modelado Unificado (UML) es un lenguaje estándar para escribir planos de software. UML puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra gran cantidad de software. UML es una técnica para la especificación de sistemas en todas sus fases y es un lenguaje orientado a objetos.(ZAMITIZ 2003a)

Un modelo UML esta compuesto por tres clases de bloques de construcción:

- Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)
- Relaciones: relacionan los elementos entre sí.
- Diagramas: Son colecciones de elementos con sus relaciones.

Entre los diagramas se encuentran los de estructura estática, los de comportamiento y de implementación.

- Estáticos:
 - Diagramas de casos de uso.
 - Diagramas de clases.
 - Diagramas de objetos.
- Dinámicos.
 - Diagramas de interacción (secuencia y colaboración).
 - Diagramas de estados.
 - Diagramas de actividades.
- Implementación.
 - Diagramas de componentes.
 - Diagramas de despliegue.

Las fases del desarrollo de sistemas que soporta UML son: Análisis de requerimientos, Análisis, Diseño, Programación y Pruebas.

Análisis de Requerimientos.

UML tiene casos de uso para capturar los requerimientos del cliente. A través del modelado de casos de uso, los actores externos que tienen interés en el sistema son modelados con la funcionalidad que ellos requieren del sistema. Los actores y los casos de uso son modelados con relaciones y tienen asociaciones entre ellos o éstas son divididas en jerarquías. Los actores y casos de uso son descritos en un diagrama de casos de uso. Cada caso de uso es descrito en texto y especifica los requerimientos del cliente: lo que éste espera del sistema sin considerar la funcionalidad que se implementará.

Análisis.

La fase de análisis abarca las abstracciones primarias (clases y objetos) y mecanismos que están presentes en el dominio del problema. Las clases que se modelan son identificadas, con sus relaciones y descritas en un diagrama de clases. Las colaboraciones entre las clases para ejecutar los casos de uso también se consideran en esta fase a través de los modelos dinámicos en UML. Es importante notar que sólo se consideran clases que están en el dominio del problema (conceptos del mundo real) y todavía no se consideran clases que definen detalles y soluciones en el sistema de software, tales como clases para interfaces de usuario, bases de datos, comunicaciones, concurrencia, etc.

Diseño.

En la fase de diseño, el resultado del análisis es expandido a una solución técnica. Se agregan nuevas clases que proveen de la infraestructura técnica: interfaces de usuario, manejo de bases de datos para almacenar objetos en una base de datos, comunicaciones con otros sistemas, etc. Las clases de dominio del problema del análisis son agregadas en esta fase. El diseño resulta en especificaciones detalladas para la fase de programación.

Programación.

En esta fase las clases del diseño son convertidas a código en un lenguaje de programación orientado a objetos. Cuando se crean los modelos de análisis y diseño en UML, lo más aconsejable es trasladar mentalmente esos modelos a código.

Pruebas.

Normalmente, un sistema es tratado en pruebas de unidades, pruebas de integración, pruebas de sistema, pruebas de aceptación, etc. Las pruebas de unidades se realizan a clases individuales o a un grupo de

clases y son típicamente ejecutadas por el programador. Las pruebas de integración integran componentes y clases en orden para verificar que se ejecutan como se especificó. Las pruebas de sistema ven al sistema como una "caja negra" y validan que el sistema tenga la funcionalidad final que el usuario final espera. Las pruebas de aceptación conducidas por el cliente verifican que el sistema satisface los requerimientos y son similares a las pruebas de sistema.(ZAMITIZ 2003b)

Lenguaje de Modelado Orientado a Objetos de Aplicaciones Multimedia.

Esta metodología se lanza como una propuesta de extensión de UML para la integración de especificaciones de sistemas multimedia basados en el paradigma orientado a objetos, y MVC (Modelo Vista Controlador) para la interfaz de usuario. El MVC es un patrón de diseño de software que distingue un componente *modelo* sosteniendo la funcionalidad del núcleo y los datos, un componente *vista* para mostrar la información al usuario y un componente *controlador* para manipular los eventos de interacción. Separando así los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos de forma que las modificaciones al componente de la vista pueden ser hechas con un mínimo impacto en el componente del modelo de datos. Un mecanismo de propagación de cambios asegura la consistencia entre el modelo y la interfaz visual. Extendiendo el paradigma MVC para multimedia a las peculiaridades de comportamiento estático y dinámico identificadas anteriormente, obtenemos MVCmm, sobre el que se basa las especificaciones de OMMMA – L.

Esta metodología no presenta cambios con respecto a UML en el flujo de requisitos y casos de uso. Pero en el modelo de clases de objetos, que es con lo que se crea el modelo del dominio, sí, por ejemplo, en el diagrama de clases del modelo de objeto se agregan tres conceptos. Un objeto puede ser del tipo:

Escenario: cuando representa un conjunto de pantallas que muestran una información a través de objetos con similar funcionalidad.

Aplicación: cuando agrupa elementos de media y asocia sus funcionalidades como una entidad.

Media: cuando se hace referencia a sonido, texto, imágenes, animaciones, video, botones, etc.

OMMMA-L está sustentado en cuatro vistas fundamentales, donde cada una se asocia a un tipo de diagrama en particular. Estas vistas son:

- Vista Lógica: modelada a través del Diagrama de Clases de OMMMA-L, extendido del Diagrama de Clases de UML, utilizando las mismas notaciones, pero incorporando las clases correspondientes a las

medias: media continua y media discreta, generalizadas en una clase medias. Divide en dos áreas dicho diagrama: una para la jerarquía de los tipos de media y otra para la modelación de la estructura lógica del dominio de la aplicación.

- Vista de Presentación Espacial: modelada a través de los Diagramas de Presentación de OMMMA-L, los cuales son de nueva aparición en la extensión de UML, dado que este último no contiene un diagrama apropiado para esta tarea. Estos diagrama tienen el propósito de declarar las interfaces de usuario con un conjunto de estructuras delimitadas en tamaño y área, dividiéndose en objetos de visualización (texto, gráfico, video, animación) e interacción (scroll, barras de menú, botones, campos de entrada y salida, hipertextos con hipervínculos). Estos diagramas de presentación pueden ser divididos en capas virtuales de presentación donde en cada uno de ellas sólo se haga referencia a una clase específica de componentes (por ejemplo, una vista para los objetos de visualización y otra para los de interacción, u otro tipo de división para la representación de los intereses de los desarrolladores).
- Vista de Comportamiento Temporal Predefinido: modelada por el Diagrama de Secuencia de OMMMA-L, extendido a partir del diagrama de secuencia de UML. El Diagrama de secuencia modela una secuencia de una presentación predefinida dentro de una escena, donde todos los objetos dentro de un diagrama se relacionan al mismo eje del tiempo. En este diagrama se hace un refinamiento del eje del tiempo con la introducción de marcas de tiempo a través de diferentes tipos de intervalos; marcas de inicio y fin de ejecución que permite soportar su reusabilidad; marcas de activación y desactivación de demoras en objetos de tipo media, posibilitando la modelación de las tolerancias de la variación de las restricciones de sincronización para los objetos media; activación compuesta de objetos media para la agrupación de objetos concurrentemente activos.
- Vista de Control Interactivo: modelado a través del Diagrama de Estado, extendido a partir del diagrama de estado de UML, sin tácticamente ser igual a este último, pero con la diferencia semántica de que, en el orden de unir los controles interactivos y predefinidos, no interrumpidos de los objetos, las acciones internas de estados simples tienen que llevar nombres de diagrama de secuencia en vez de diagramas de estado empotrados; queriendo esto decir que el comportamiento especificado por el diagrama de secuencia se provoca automáticamente cuando se entra al estado correspondiente donde se hace referencia.

Actualmente, OMMMA – L se evalúa en diferentes escenarios, como proyectos industriales para la especificación de servicios de información multimedia, y se investiga características adicionales de sincronía

para su especificación en el lenguaje y la formalización de un modelo para la composición dentro y entre los diferentes diagramas de comportamiento.(FERNÁNDEZ and CATALÁ 2006)

2.14 Herramientas de autor.

Las herramientas de programación de aplicaciones con tecnología multimedia están diseñadas para administrar los elementos individualmente y permiten interactuar con los usuarios. Además de proporcionar un método para que los usuarios interactúen con el proyecto, la mayoría de las herramientas de desarrollo de multimedia ofrecen además facilidades para crear y editar texto e imágenes, y tienen extensiones para controlar los reproductores de videos, sonidos, etc. El conjunto de lo que se produce y la forma de presentarlo al observador es la interfaz de usuario. Esta interfaz puede definirse tanto por las reglas de lo que debe suceder con los datos introducidos por el usuario como por los gráficos que aparecen en la pantalla. Los programas que rigen los límites de lo que puede ocurrir es la plataforma o ambiente multimedia.

2.14.1 Macromedia Flash.

Flash es una herramienta de autor, que se refiere tanto al programa de edición multimedia como a Macromedia Flash Player, escrito y distribuido por Adobe. Este utiliza gráficos vectoriales e imágenes, sonido, código de programa, flujo de video y audio bidireccional. En sentido estricto, Flash es el entorno y Flash Player es el programa de máquina virtual utilizado para ejecutar los archivos generados con Flash. Generalmente los archivos generados por Flash tienen extensión .swf y pueden aparecer en páginas Web para ser vistos en navegadores o también pueden ser reproducidos en un reproductor de Flash. Estos son utilizados para presentar animaciones, sitios multimedia o anuncios en la Web.

Fue creada en 1997 siendo una simple herramienta de animación en su primera versión y a medida que han ido mejorando sus versiones, se ha convertido en una herramienta de desarrollo de sitios y aplicaciones interactivas en el actual mundo del ciberespacio brindando una interfaz visual amigable a los usuarios. Actualmente con Macromedia Flash se puede diseñar gráficos al igual que aplicaciones gestionadas por datos. A estas aplicaciones se les puede añadir imágenes, videos y sonidos, también tiene funciones ya implementadas que hacen más fácil el trabajo. Incluye el lenguaje de programación ActionScripts, éste se utiliza para controlar los objetos en la película Flash y crear elementos interactivos y de navegación creando

películas mucho más interactivas. Así en las otras versiones que surgieron éste fue incorporando un ActionScripts más rico en sintaxis y posibles aplicaciones.

En el 2003 surge la versión Flash MX 2004, en el que hubo cambios radicales y al que se le denominó ActionScripts 2.0, lenguaje de programación orientado a objetos. Macromedia Flash 8 también es una potente herramienta creada por Macromedia que ha superado las mejores expectativas: porque posee diseños más atractivos, optimización de fuentes, bibliotecas integradas, mayor potencia de animación, mayor potencia gráfica, mejoras en la importación de videos, asistente de ActionScripts, etc.

Macromedia ya lanzó la versión 3.0 de ActionScripts en una versión Beta, la cual todavía no está vigente en ninguna de las versiones anteriores, pero ésta incluye mejoras en el rendimiento además de expresiones regulares y nuevas formas de empaquetar las clases y disminuye el uso de recursos en las aplicaciones de Macromedia Flash.(WIKIPEDIA 2007)

2.14.2 Revolution.

Posee características de las herramientas de desarrollo rápido como Visual Basic o Delphi. Incluye soporte multiplataforma como Java y otros productos. Al igual que los Sistemas de Gestión de Bases de Datos como Oracle o Access tiene características que le permiten administrar y consultar datos. Su capacidad multimedia puede compararse con sistemas de autor reconocidos como Macromedia Director, y su filosofía es muy similar al formato de presentación como diapositivas de PowerPoint. El manejo de textos y sus capacidades hipertexto es comparable con los procesadores de textos o las herramientas de diseño Web. La capacidad de elaboración de gráficos y captura de pantallas es comparable a editores gráficos sencillos como Microsoft Paint. Y por último se basa en la metáfora de una pila de tarjetas indexadas que ha probado su efectividad en varios sistemas de autor como Hypercard y MetaCard.(RODRÍGUEZ. 2000)

Revolución es una herramienta desarrollada sobre MetaCard por la empresa Run Revolution Ltd. Es una respuesta a lo que en muchas ocasiones se ha lamentado de MetaCard: su austera interfaz. Tiene la demostración de lo que se puede construir con ésta y que responde a las demandas de una interfaz más visual a la misma herramienta. El entorno de trabajo deja de ser austero para pasar a ser totalmente visual en la disposición de las opciones y el modo de trabajo más cuidado desde el punto de vista estético. La incorporación de nueva documentación y sobre todo los asistentes que incorpora hacen de ésta una gran aplicación.

Esta aplicación también está disponible en las plataformas soportadas por MetaCard y tiene un lenguaje de

programación de alto nivel que se denomina "Transcript". Permite extender las posibilidades de MetaCard, las pilas (aplicaciones) que haya creado allí se pueden seguir editando dentro del entorno de Revolution solamente con abrirlas.(MELCHOR 2004)

2.14.3 Macromedia Director.

Programa de Adobe Systems Incorporated para la producción de películas ejecutables en Macromedia Shockwave, usando mapas de bits y en programación Lingo. Esta herramienta permite generar presentaciones multimedia que pueden ser distribuidas a través de CDs. Permite incorporar a las películas múltiples formatos, como imágenes, vídeos, sonidos o animaciones Flash. Incluye editores básicos para texto, mapa de bits y vectores.

El usuario es como el director de la película, es el que controla todos sus aspectos, de ahí su nombre. En esta se crea una película (*movie*), que para ello existen varias ventanas una para el reparto de "actores" (*Cast*), otra para el Montaje (*Score*), otra para los Guiones (*Scripts*) y otra para ver los resultados (*Stage*).

Además de tener incorporado el lenguaje Lingo, el cual es muy potente, también cuenta con pequeños programas llamados XTRAS desarrollados en el lenguaje C++, que proporcionan al usuario facilidad de trabajo.

Esta herramienta puede generar varios tipos de archivos y para verlos en cualquier computador sin tener Macromedia Director instalado, se debe salvar en un archivo ejecutable (.exe).(WIKIPEDIA 2007b)

2.14.4 Scala Multimedia MM200.

Scala Multimedia es un producto principalmente enfocado a la realización de presentaciones espectaculares, compitiendo en cierta medida con Director, pero que para nada se solapa con el mercado de Authorware y ToolBook. A diferencia de Director, Scala Multimedia es un producto que saca el máximo rendimiento a la máquina donde se ejecute. Hay que tener en cuenta que el objetivo perseguido por el producto es conseguir efectos espectaculares, muy parecidos a los que se utilizan en televisión.

MM200 es un producto que hace un uso intensivo de guiones para crear los efectos visuales y la correspondiente interactividad. Sin embargo, un aspecto a destacar es que mediante HumanTouch (su interfaz gráfica) se abstrae prácticamente toda la programación, siendo necesaria únicamente la utilización de menús y opciones para crear complejos efectos.

El producto incluye botones cuya funcionalidad ya ha sido programada, también se incluyen algunos cliparts, así como fondos de pantalla y animaciones. Junto a éstos también se distribuyen algunos efectos de sonido y cortes musicales. Por supuesto, todo ello de libre distribución. Sólo algunos guiones de ejemplo se entregan junto al producto, habiendo sido deseable que, al igual que sucede con ToolBook o Director, se incluyeran gran cantidad de guiones preescritos.(FERNÁNDEZ and CATALÁ 2006)

2.14.5 Neobook.

Neobook Profesional, poderosa Herramienta de autor Multimedia. Incluso personas sin conocimientos de programación pueden rápidamente crear y publicar diferentes tipos de software multimedia interactivo. Neobook Profesional permite la creación fácil de publicaciones multimedia interactivas basados en PC`s para ser distribuidos vía disquete, CD-Rom, redes, Internet, etc. El control de las páginas y otros elementos de la publicación electrónica son organizados en la pantalla y luego compilados dentro de los archivos EXE autosoportados o salva pantallas que pueden ser distribuidos libres del pago de royalties o regalías.

Uso de NeoBook:

- Material educativo interactivo.
- Boletín de novedades.
- Presentaciones y reportes.
- Catálogos y folletos de venta.
- Revistas electrónicas.
- Libros interactivos ilustrados.
- Textos multimediales (con hipertexto y video).
- Kioscos, pantallas comerciales interactivas.
- Material de entrenamiento, tutoriales.(ALEAN 2006)

2.15 Lenguajes utilizados por las herramientas.

2.15.1 XML (Lenguaje de Marcas Extensible).

XML es un lenguaje de marcas que ofrece un formato para la descripción de datos estructurados, el cual conserva todas las propiedades importantes del mencionado SGML (Lenguaje de Marcas Estándar Generalizado). Es decir, XML es un metalenguaje, porque con él se puede definir el lenguaje de presentación que desee el propio diseñador y, a diferencia del HTML (Lenguaje de Marcas de Hipertexto), que se centra en la representación de la información, XML se centra en la información en sí misma. La particularidad más importante del XML es que no posee etiquetas prefijadas con anterioridad, ya que es el propio diseñador el que las crea a su antojo, dependiendo del contenido del documento. Este lenguaje es un formato basado en texto, específicamente diseñado para almacenar y transmitir datos. Un documento XML se compone de elementos XML, cada uno de los cuales consta de una etiqueta de inicio, de una etiqueta de fin y de los datos comprendidos entre ambas etiquetas. Al igual que los documentos HTML, un documento XML contiene texto anotado por etiquetas.(EMAGISTER 2000) Sin embargo, a diferencia de HTML, XML admite un conjunto ilimitado de etiquetas, no para indicar el aspecto que debe tener algo, sino lo que significa. XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.(WIKIPEDIA 2007g)

2.16 Metodología, herramientas y lenguajes de programación utilizados para el desarrollo de la multimedia.

La metodología escogida para el desarrollo de la aplicación con tecnología multimedia es RUP con la herramienta CASE Rational Rose como apoyo para el trabajo con los modelos del sistema, esta metodología trabaja con el lenguaje UML y como extensión de este lenguaje se escogió OMMMA-L, lenguaje para el proceso de desarrollo de estas aplicaciones. La herramienta escogida para desarrollar la

aplicación es Macromedia Flash 8, el cual utiliza el metalenguaje XML como base de datos para el almacenamiento de la información que estará contenida en el sistema.

2.17 Factibilidad de trabajar en Flash con base de datos en XML.

XML, el lenguaje de marcas estándar para el intercambio de información entre aplicaciones, no es una excepción al soporte de ActionScript. El uso del objeto XML, destinado exclusivamente a la gestión de archivos y contenidos formateados en este estándar, permite a una película Flash importar y exportar fácilmente información desde y hacia lenguajes de servidor o bases de datos. XML se encarga de estructurar estos datos de forma tal que puedan ser leídos e interpretados sin problemas por cada una de las partes. Lectores de noticias, sistemas de gestión de weblogs y menús dinámicos son algunas de las aplicaciones donde el uso del lenguaje XML se hace prácticamente imprescindible. La integración de Flash con XML permite además establecer conexiones persistentes a través de los llamados XML sockets, que hacen posible una comunicación constante entre dos o más conexiones simultáneas. Siempre que estemos conectados a un servidor de este tipo, estaremos habilitados para recibir y enviar información en todo momento. Una nueva clase integrada en la interfaz de desarrollo de Flash, llamada XML Socket, permite conectar una aplicación a un servidor socket, una ventaja que hace posible el desarrollo de juegos en red o aplicaciones de conversación en tiempo real. A todas estas ventajas funcionales, se les suman las de mantenimiento, ya que al actualizar los datos desde un documento XML, no se tendrá que editar recurrentemente el archivo fuente (.fla) cada vez que se necesite modificar el contenido de una aplicación o sitio Web. Flash y XML, dos tecnologías ampliamente extendidas, se unen para lograr resultados de un dinamismo y atractivo visual incomparables. Sumando la versatilidad del lenguaje XML a la potencia de ActionScript 2.0, podrán obtenerse resultados de alto impacto visual y enorme funcionalidad.(VIRTUAL-FORMAC 2007)

2.18 Conclusiones.

Desde el surgimiento de la multimedia, ésta ha sido un paso importante en las comunicaciones entre las personas de todo el mundo. Sus antecedentes reflejan la gran importancia que tuvo desde un principio. Después de un estudio de estos antecedentes se fortalece más la idea de la realización de una multimedia para la presentación de la historia de la facultad 8.

En este capítulo también se hace un profundo estudio de las metodologías y herramientas que se utilizan para el desarrollo de aplicaciones con tecnología multimedia al igual que los lenguajes que utilizan las herramientas. El desarrollo de estas metodologías, herramientas y lenguajes, en todos estos años, desde que surgieron estas aplicaciones, ha sido muy importante para el perfeccionamiento de estos sistemas, como la calidad visual, el dinamismo y la interactividad que brinda a los usuarios.

Capítulo 3

Descripción de la solución propuesta.

3.1 Introducción.

En este capítulo se realizará una descripción y modelación de la solución propuesta a partir de la metodología seleccionada, que dé solución al proyecto de tesis presentado, que es el desarrollo de una multimedia que solucione la falta de conocimiento acerca de toda la historia de la facultad.

3.2 Especificación del contenido.

La aplicación aborda el tema de todas las actividades realizadas en la facultad desde su surgimiento. Primeramente muestra al usuario, después de la presentación, una introducción sobre la universidad y los orígenes de la facultad 8.

El módulo Facultad muestra información sobre la estructura que tiene ésta, o sea, los profesores que conforman su dirección y los antecedentes de la facultad como los docentes en los que ha radicado y el comportamiento de las matrículas por cada curso.

El módulo Cursos contiene toda la información referente a un curso docente como el comportamiento de la docencia y de la producción e investigación, todas las actividades que se han desarrollado, políticas, culturales, deportivas o recreativas, los eventos en los cuales los estudiantes participan.

El módulo FEU contiene información sobre esta organización, la estructura que ha tenido en cursos anteriores y la actual, como se desarrolla y el trabajo que lleva en la facultad para la organización y guía de todos los estudiantes. Además brinda la historia de todo lo realizado por esta organización desde el surgimiento de la facultad 8.

El módulo UJC es similar al módulo FEU, también contiene información sobre la estructura que tuvo y tiene su dirección en la facultad y sobre todo lo que ha logrado desde que se formó la facultad 8.

Cada uno de los módulos contiene una galería de imágenes que presenta imágenes que representan el contenido que se está abordando.

El contenido abordado en la multimedia, en general cumple con los objetivos planteados anteriormente de centralizar y mostrar toda la información generada desde el surgimiento de la facultad 8 en una aplicación multimedia para el conocimiento de estudiantes, profesores y trabajadores de esta facultad y para toda persona interesada en conocer acerca de este tema.

3.3 Modelo de dominio para el sistema propuesto.

El modelo de dominio se plantea cuando no se puede desarrollar el modelo de negocio, o sea, que no se pueda determinar las fronteras de las personas que lo inician, las que se benefician, las que realizan las actividades. Debido a que la aplicación planteada no define estas fronteras, se realiza el modelo de dominio con el objetivo de comprender y describir los conceptos dentro del contexto del problema para su mejor entendimiento.

Conceptos asociados al modelo de dominio.

Información: la información es un conjunto organizado de datos, que constituyen un mensaje sobre un determinado ente o fenómeno. Según otro punto de vista, la información es un fenómeno que proporciona significado o sentido a las cosas, e indica mediante códigos y conjuntos de datos, los modelos del pensamiento humano. La información por tanto, procesa y genera el conocimiento humano.

Cursos: Lapso dedicado durante un año o un semestre académico, para impartir las lecciones contenidas en un programa.

Organizaciones estudiantiles: Organizaciones en las que los estudiantes se reúnen y estructuran para el logro de un mejor funcionamiento, siguiendo deberes y derechos que deben cumplirse.

Facultad: Una facultad es una subdivisión de una universidad.

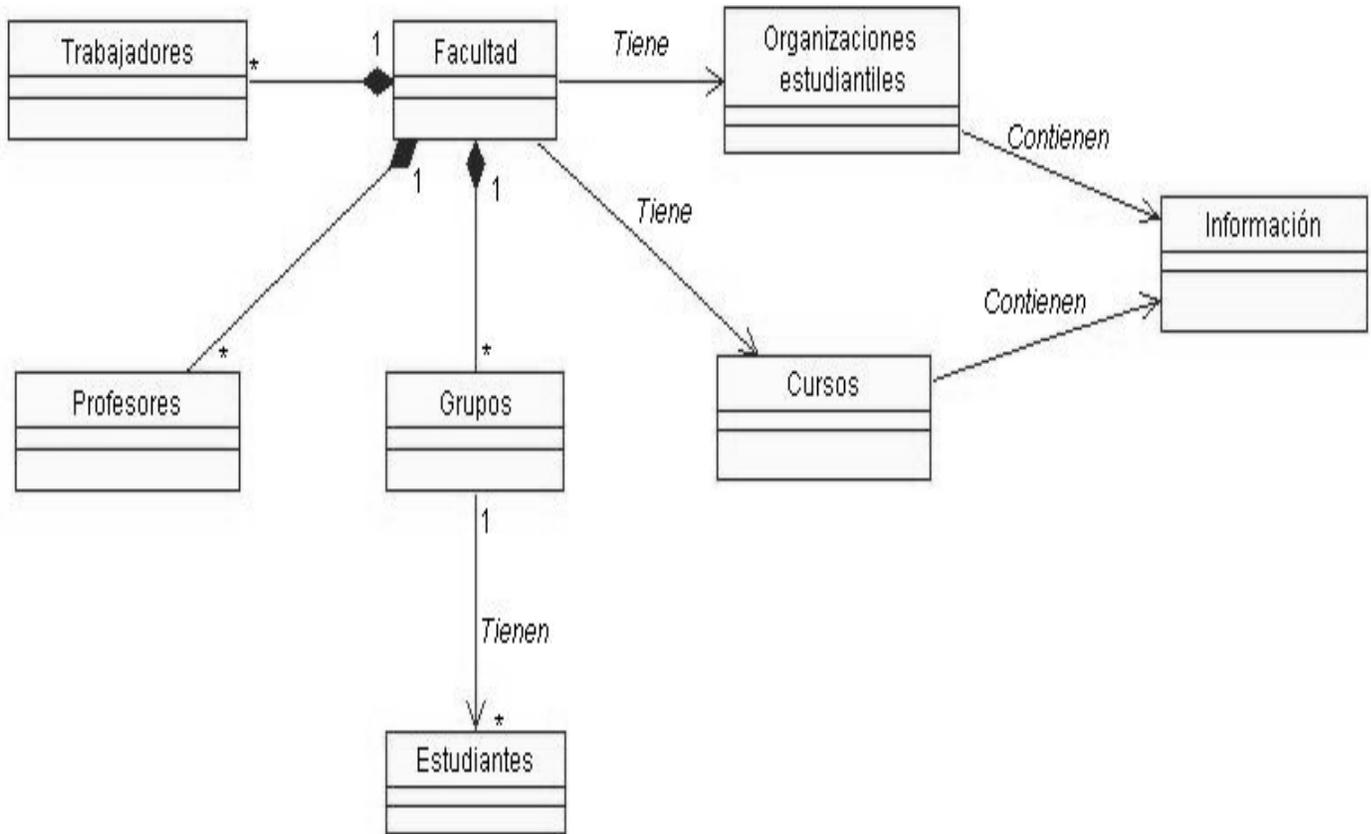
Grupos: Pequeñas partes en las que se divide la facultad para agrupar a los estudiantes según el año que estén cursando.

Estudiante: Un estudiante es aquella persona dedicada a la lectura, puesta en práctica y aprehensión de conocimientos sobre alguna materia o arte.

Profesor: Un profesor es una persona que enseña una determinada ciencia o arte pero al contrario que maestro no se le reconoce una habilidad extraordinaria en la materia que instruye. Sin embargo deben poseer habilidades pedagógicas para ser agentes efectivos del proceso de aprendizaje. El profesor, por tanto, parte de la base de que es la enseñanza su dedicación y profesión fundamental y que sus habilidades consisten en enseñar la materia de estudio de la mejor manera posible para el alumno.

Trabajador: Un trabajador es una persona que con la edad legal suficiente, y de forma voluntaria presta servicios retribuidos.

3.3.1 Modelo de dominio.



3.4 Solución Propuesta.

Se desarrollará una aplicación multimedia debido a las diferentes ventajas que brinda el uso de estos sistemas informáticos como, la motivación que crea en el usuario, su fácil uso individual o colectivo y su carácter interactivo que genera una intensa y continua actividad mental y con ésta se logra un mayor aprendizaje en menor tiempo. La multimedia que tiene propuesta este proyecto de tesis propone dar solución al poco conocimiento que tienen todos los que conforman el colectivo de la facultad 8 acerca de la

historia que ha desarrollado esta facultad desde que surgió en la Universidad de las Ciencias Informáticas. De acuerdo a todas las actividades que se han ido desarrollando a medida que han ido pasando los años se ha ido generando gran cantidad de información la que, después de una exhaustiva recopilación de ésta y una minuciosa selección, se presentará en dicha aplicación, resolviendo así la problemática presentada anteriormente.

3.5 Descripción del sistema propuesto.

La metodología escogida para su desarrollo es RUP, que trabaja con el lenguaje UML y como extensión de este lenguaje se escogió OMMMA-L, lenguaje para el desarrollo de aplicaciones con tecnología multimedia. Esta metodología no presenta cambios con respecto a UML en el flujo de requisitos y casos de uso. Pero en el modelo de clases de objetos, modelo con lo que se crea el modelo del dominio cambia el diagrama de clases del modelo de objeto donde se agregan tres conceptos: aplicaciones, medias y escenarios.

3.6 Descripción de la funcionalidad del sistema propuesto.

A continuación se comenzarán a analizar las funcionalidades y características del sistema. Para dar cumplimiento a este objetivo se identificarán los requerimientos funcionales y no funcionales, se identificarán los actores así como se modelarán las funcionalidades en términos de casos de uso del sistema con sus respectivas descripciones y se desarrollarán los diagramas de clases y de clases de diseño.

3.7 Requerimientos funcionales.

RF#	Función
R1	Mostrar información contenida en las pantallas.
R1.1	Mostrar presentación de la multimedia.
R1.2	Mostrar información de cada pantalla de la multimedia.
R1.3	Mostrar imágenes contenidas en la galería.
R2	Permitir la navegabilidad al usuario.
R2.1	Permitir que el usuario pueda acceder a cualquier módulo desde la pantalla que se encuentre
R2.2	Permitir al usuario acceder a cada una de las opciones que tenga una pantalla.
R3	Permitir al usuario realizar acciones generales.
R3.1	Permitir al usuario reproducir o pausar el sonido de fondo de la aplicación.

R3.2	Permitir al usuario interactuar con las palabras calientes.
R3.3	Permitir al usuario salir del sistema desde cualquier pantalla cuando lo desee.

3.8 Requerimientos no funcionales.

Requerimientos de apariencia o interfaz externa:

- Diseño sencillo, permitiendo el manejo del sistema sin muchos conocimientos de informática.
- Utilizar colores sencillos y frescos en el diseño de la interfaz de la aplicación.
- Diseño para resoluciones de 800x600, y otras resoluciones.
- Utilización de botones que lleven texto y ayuden en la navegación al usuario.
- Contrastar los colores de fondo con el color de las letras para lograr uniformidad y calidad en la visualización de la aplicación y una mayor motivación en el usuario.

Requerimientos de software.

- Requiere de Flash Player 8 para su correcto funcionamiento.

Requerimientos de hardware.

- Los requerimientos de hardware mínimos exigidos para Flash Player 8 sobre Windows son: Procesador Intel Pentium III de 450 MHz (o equivalente) y versiones posteriores y 128 MB de RAM. Para Macintosh son: 500 MHz PowerPC G3 y versiones posteriores y 128 MB de RAM.

Requerimiento de implementación.

- La base de datos debe realizarse en XML, para lograr que en un futuro, si se desea modificar algo, no haya que acceder a la aplicación y se haga más fácil.

Requerimientos de Rendimiento.

- El tiempo para la visualización de las medias no debe exceder los 5 segundos.
- El tiempo de ejecución de los vínculos entre las medias no debe superar los 5 segundos.

Requerimiento de Soporte.

- Para el correcto funcionamiento del software, la computadora donde se ejecutará ésta, deberá tener entre sus dispositivos una tarjeta de sonido y aditamentos para la reproducción de éste.

Requerimiento de Portabilidad.

- El software podrá ser usado bajo los sistemas operativos Windows y Linux, en este último debe existir el emulador para multimedia requerido para la ejecución de este tipo de aplicaciones.

Requerimientos no funcionales de Seguridad.

- Sólo la Universidad de las Ciencias Informáticas es la entidad autorizada para la distribución de la aplicación en cualquier ámbito de la universidad.

3.9 Actores del sistema.

Actores	Justificación
Usuarios	A quienes va dirigido el sistema.

3.10 Descripción de los casos de uso del sistema.

Caso de Uso	1 Mostrar información contenida en las pantallas.	
Actores	Estudiantes, profesores y trabajadores.(usuario)	
Resumen	El caso de uso se inicia cuando el usuario ejecuta la multimedia.	
Propósito	Permitir al usuario acceder a la información que se encuentra en la multimedia.	
Referencias	R1	
Precondiciones	El usuario tenga acceso a la multimedia.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema.	
1-El usuario ejecuta la multimedia.	1.1 El sistema muestra la presentación de la multimedia y culmina en la pantalla principal que contiene varios módulos:	
	-Cursos docentes (módulo 1)	
2-El usuario selecciona cualquiera de los módulos de la pantalla principal.	-FEU (módulo 2)	
	-UJC (módulo 3)	
	-Facultad (módulo 4)	
Módulo 1: Mostrar información que se brinda en cursos docentes.		

1-El usuario selecciona un curso docente.	1.1-El sistema carga el xml y muestra la introducción de dicho curso.
2-El usuario selecciona la opción Docencia del curso.	2.1-El sistema carga el xml y muestra al usuario la información seleccionada.
3-El usuario selecciona la opción Producción del curso.	3.1-El sistema carga el xml y muestra al usuario la información seleccionada.
4-El usuario selecciona la opción Cultura del curso.	4.1-El sistema carga el xml y muestra al usuario la información seleccionada.
5-El usuario selecciona la opción Deporte del curso.	5.1-El sistema carga el xml y muestra al usuario la información seleccionada.
6-El usuario selecciona la opción Eventos del curso.	6.1-El sistema carga el xml y muestra al usuario la información seleccionada.
7-El usuario selecciona la galería de imágenes de este curso.	7.1-El sistema carga el xml y muestra la galería de imágenes.
Módulo 2: Mostrar información que se brinda en FEU.	
1-El usuario selecciona el módulo FEU.	1.1-El sistema carga el xml y muestra la introducción de dicho módulo.
2-El usuario selecciona la opción Estructura.	2.1-El sistema carga el xml y muestra la información seleccionada.
3-El usuario selecciona la opción Dirección.	3.1-El sistema carga el xml y muestra la información seleccionada.
4-El usuario selecciona la opción Historia.	4.1-El sistema carga el xml y muestra la información seleccionada.
5-El usuario selecciona la galería de imágenes para este módulo.	5.1-El sistema carga el xml y muestra la galería de imágenes seleccionada.
Módulo 3: Mostrar información que se brinda en UJC.	
1-El usuario selecciona el módulo UJC.	1.1-El sistema carga el xml y muestra la introducción de dicho módulo.
2-El usuario selecciona la opción Estructura.	2.1-El sistema carga el xml y muestra la información seleccionada.
3-El usuario selecciona la opción Dirección.	3.1-El sistema carga el xml y muestra la información seleccionada.
4-El usuario selecciona la opción Historia.	4.1-El sistema carga el xml y muestra la información seleccionada.
5-El usuario selecciona la galería de imágenes para este módulo.	5.1-El sistema carga el xml y muestra la galería de imágenes seleccionada.
Módulo 4: Mostrar información que se brinda en Facultad.	
1-El usuario selecciona el módulo Facultad.	1.1-El sistema carga el xml y muestra la información de dicho módulo.
Prioridad	Crítico
Requerimientos no	
Funcionales	

Poscondiciones	
-----------------------	--

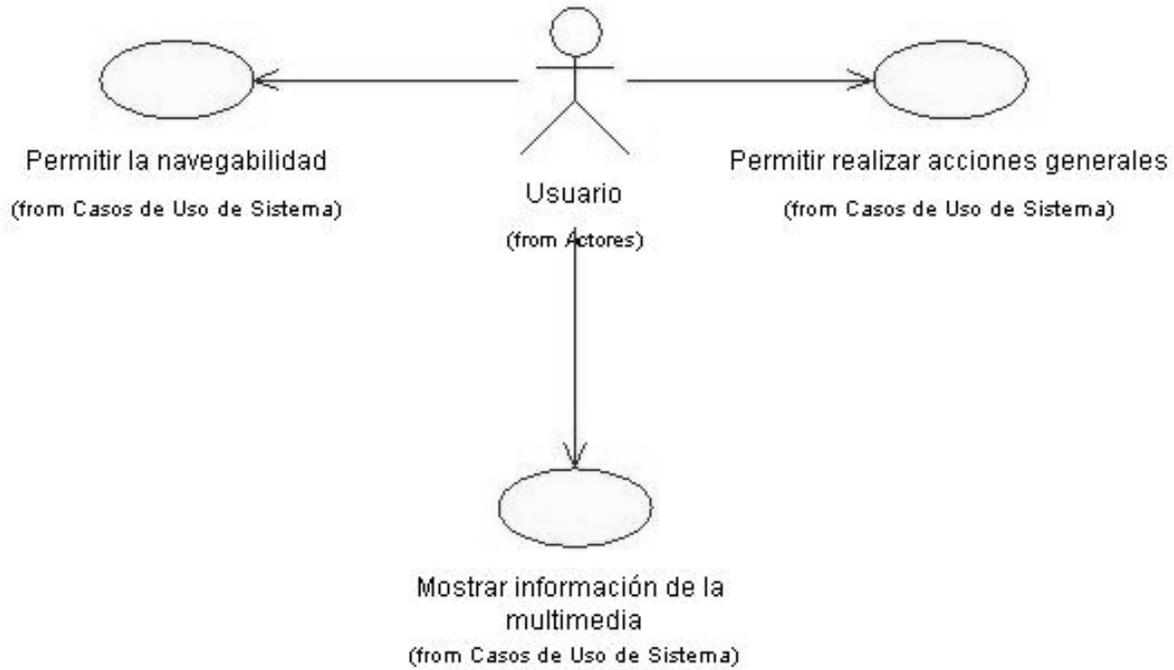
Caso de Uso	2 Permitir la navegabilidad al usuario.	
Actores	Estudiantes, profesores y trabajadores.(usuario)	
Resumen	El caso de uso se inicia cuando el usuario desea navegar por la multimedia.	
Propósito	Permitir al usuario acceder a todas las pantallas de la multimedia.	
Referencias	R2	
Precondiciones	El usuario debe seleccionar botones que le permitan navegar la por multimedia.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema.	
1-El usuario presiona algún botón para acceder a alguna pantalla de la multimedia. 2-El usuario presiona otro botón para seguir navegando por la multimedia. 3-El usuario accede a las opciones (botones) de la pantalla.	1.1-El sistema permite al usuario acceder a esta pantalla. 2.1-El sistema permite que el usuario acceda a la pantalla que desee el usuario desde donde se encuentre porque brinda una navegación global. 3.1-El sistema permite al usuario acceder a la información que brinda esta opción (botón).	
Prioridad	Crítico	
Requerimientos no Funcionales		
Poscondiciones		

Caso de Uso	3 Permitir al usuario realizar acciones generales.	
Actores	Estudiantes, profesores y trabajadores.(usuario)	
Resumen	El caso de uso se inicia cuando el usuario desea realizar alguna opción que brinda la multimedia.	
Propósito	Permitir al usuario acceder a todas las opciones que brinda la multimedia.	
Referencias	R3	
Precondiciones	El usuario debe seleccionar alguna de las opciones de la multimedia.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema.	
1-El usuario selecciona algunas de las opciones que brinda la multimedia.	1.1-El sistema muestra diversas opciones cuando el usuario esta interactuando con éste. - Reproducir o pausar el sonido de fondo. - Interactuar con las palabras calientes. - Salir del sistema.	
- Reproducir o pausar el sonido de fondo.		
1-El usuario selecciona la opción de controlar el sonido de fondo de la	1.1-El sistema ejecuta la opción solicitada por el usuario.	

aplicación.	
- Interactuar con las palabras calientes.	
2-El usuario selecciona alguna palabra caliente.	2.1-El sistema carga el xml y muestra el significado de la palabra caliente.
- Salir del sistema.	
3-El usuario selecciona la opción de salir de la aplicación.	3.1-El sistema ejecuta la acción seleccionada por el usuario. 3.2-Muestra una ventana para asegurar al usuario de su opción de salir.
4-El usuario acepta salir.	4.1-El sistema carga el xml y muestra los créditos.
Curso Alterno	
4-El usuario no acepta salir.	4.1-El sistema muestra la pantalla donde se encontraba el usuario.
Prioridad	Crítico
Requerimientos no Funcionales	
Poscondiciones	

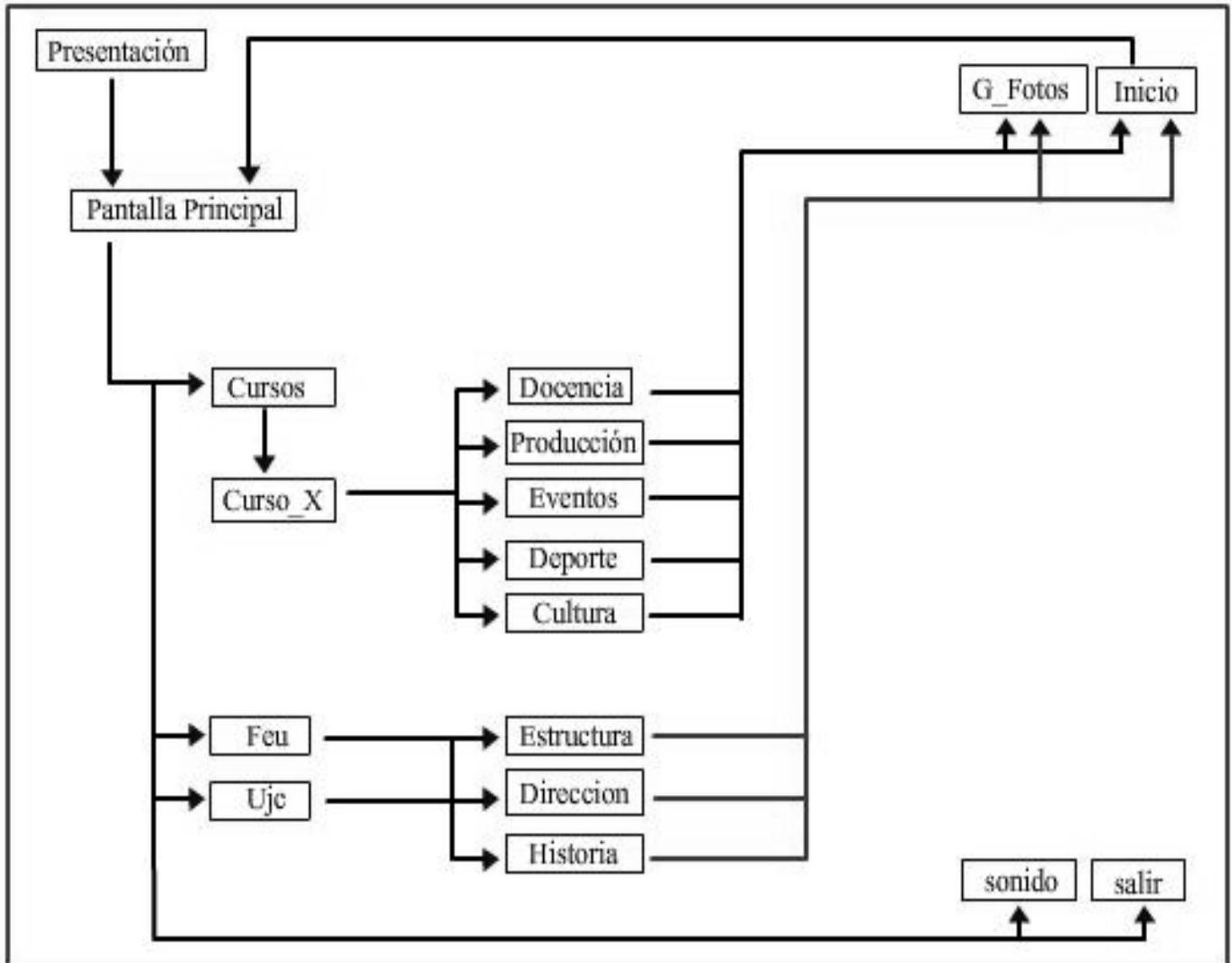
3.11 Modelo de casos de uso del sistema.

El modelado de Casos de Uso es la técnica más efectiva y a la vez la más simple para modelar los requisitos del sistema desde la perspectiva del usuario. Los Casos de Uso se utilizan para modelar cómo un sistema o negocio funciona actualmente, o cómo los usuarios desean que funcione. No es realmente una aproximación a la orientación a objetos; es realmente una forma de modelar procesos. Es, sin embargo, una manera muy buena de dirigirse hacia el análisis de sistemas orientado a objetos. Los casos de uso son generalmente el punto de partida del análisis orientado a objetos con UML.



3.12 Modelo de navegación del sistema.

Este representa la navegación que presenta la aplicación. La forma de navegar que tendrá el usuario en esta aplicación será global, o sea, que podrá acceder a todos los módulos desde la pantalla que se encuentre o volver a la pantalla principal. Cuando esté en presencia de la pantalla que muestra la galería, también tendrá la oportunidad de acceder a la pantalla principal y seleccionar el contenido que desea ver.



3.13 Conclusiones.

En este capítulo se muestra la modelación del dominio, del futuro proyecto que se tiene trazado, todas las aristas que representan el problema a resolver con la aplicación con tecnología multimedia que se desarrollará. Otras cuestiones importantes que se abordan en el capítulo son los requerimientos funcionales y no funcionales que tendrá la aplicación de acuerdo a lo que se requiere para que el usuario tenga en sus manos una aplicación que cumpla sus expectativas de búsqueda de información y agradable a la vista,

también los actores y casos de uso del sistema que son la parte esencial a tener en cuenta para el desarrollo del sistema. Como también se presentó una visión general del diagrama de casos de uso del sistema.

Capítulo 4

Construcción de la solución propuesta.

4.1 Introducción.

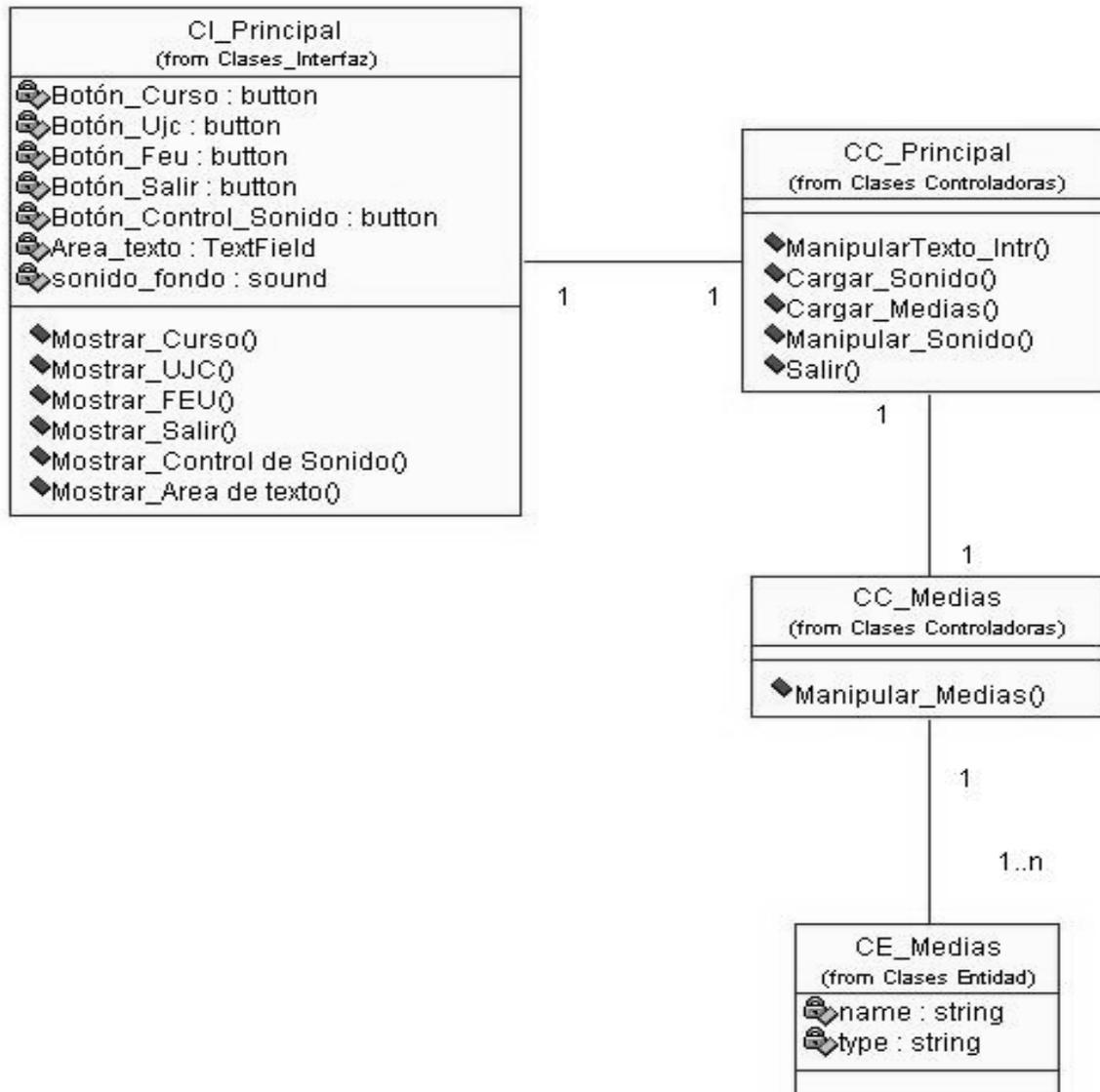
En este capítulo se desarrolla la ingeniería de software de la aplicación. Se analizará la presentación de la aplicación, es decir, como se distribuyen las funcionalidades de ésta por cada una de las pantallas que se muestran al usuario, como quedará estructurada la aplicación cuando se entregue al usuario, o sea, todos los elementos que la van a componer para su buen funcionamiento y como será la plataforma para el uso correcto de ésta. Además el tiempo que ésta demorará en dar respuesta a las peticiones que le hará el usuario mediante los diagramas de secuencia.

4.2 Modelo de clases del diseño.

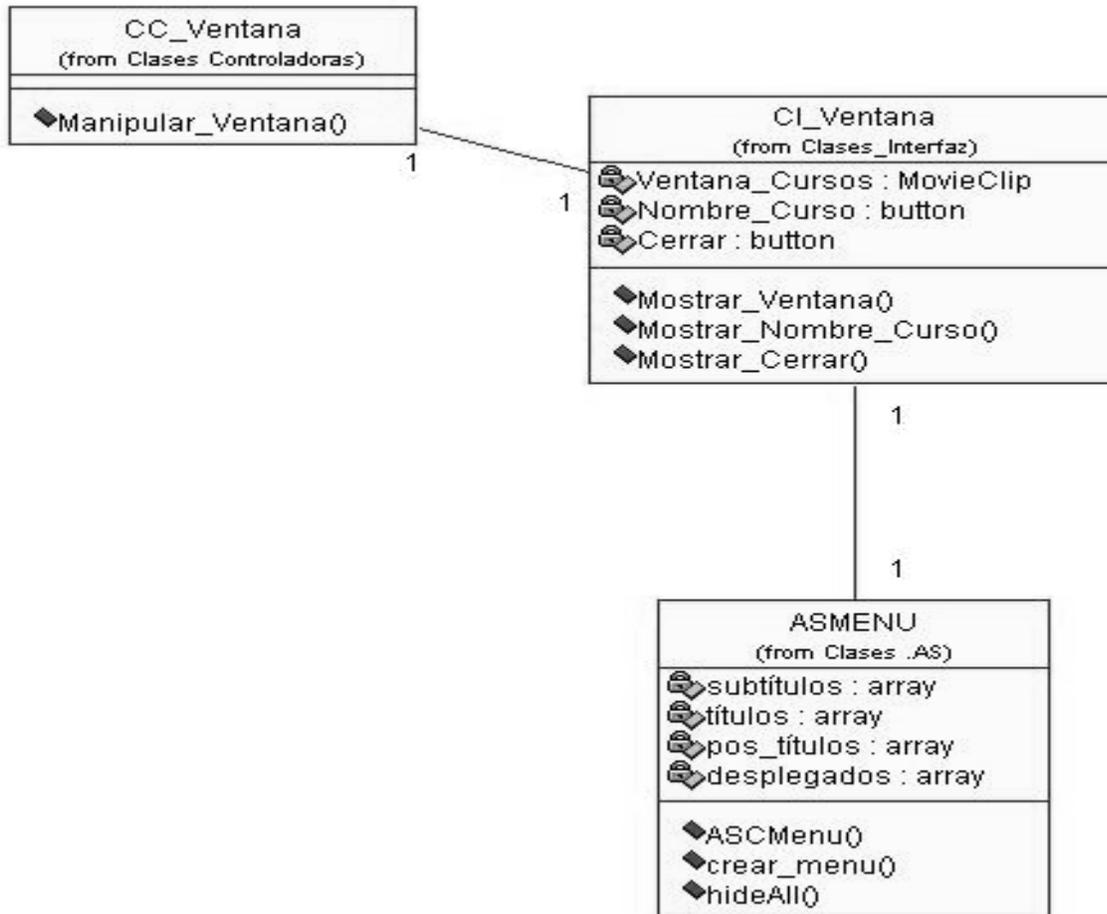
4.2.1 Diagrama de clases del diseño.

El Proceso Unificado del Rational o RUP como se ha hecho referencia anteriormente es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. UML plantea una vista estática y otra dinámica de los objetos que interactúan en el desarrollo de un sistema. Para la vista estática se utilizan los diagramas de clases y para la dinámica los de interacción o secuencia. A continuación se muestran los diagramas de clases del diseño.(JACOBSON 1995)

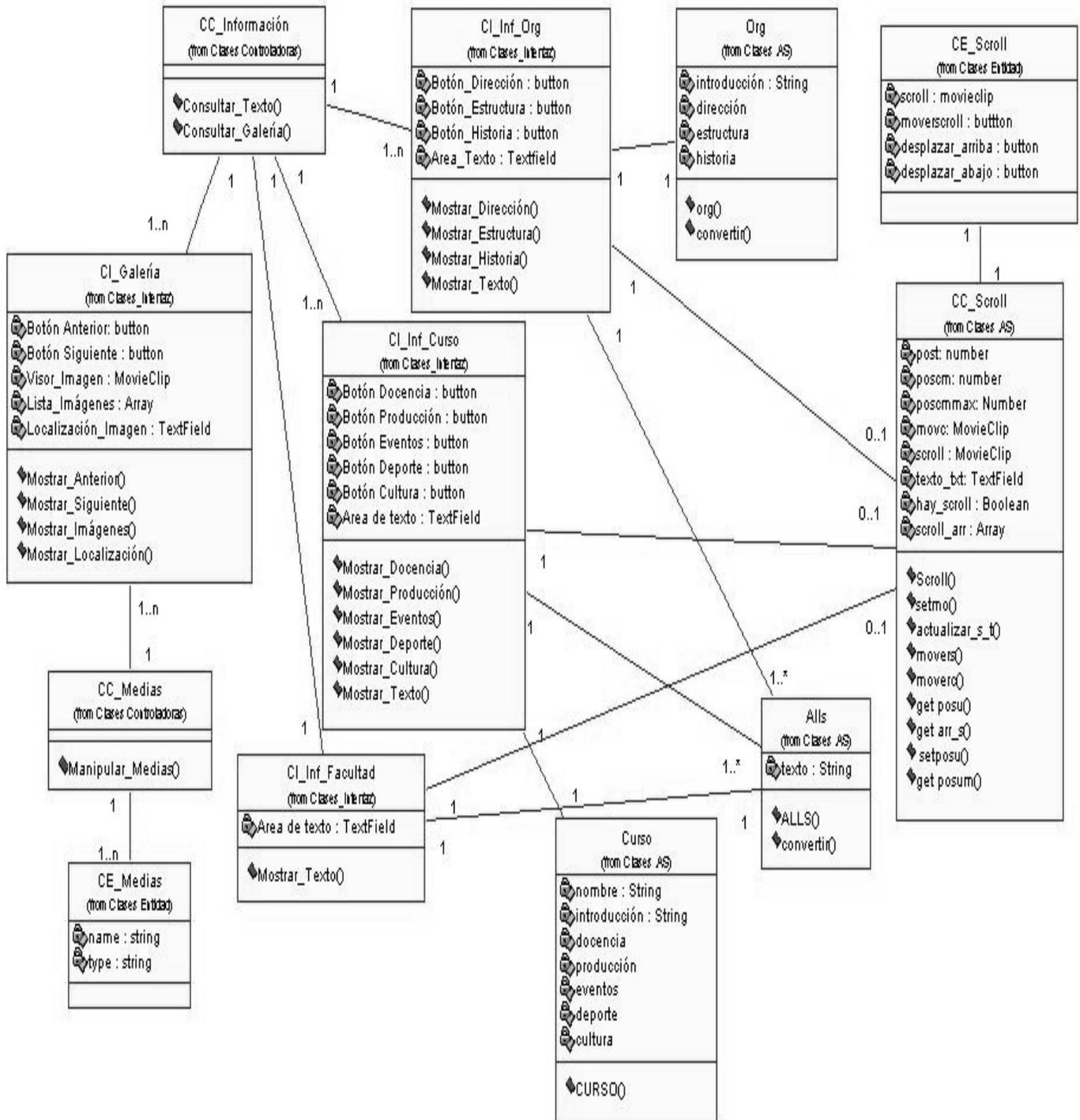
- Clases del diseño que intervienen en la interfaz principal.



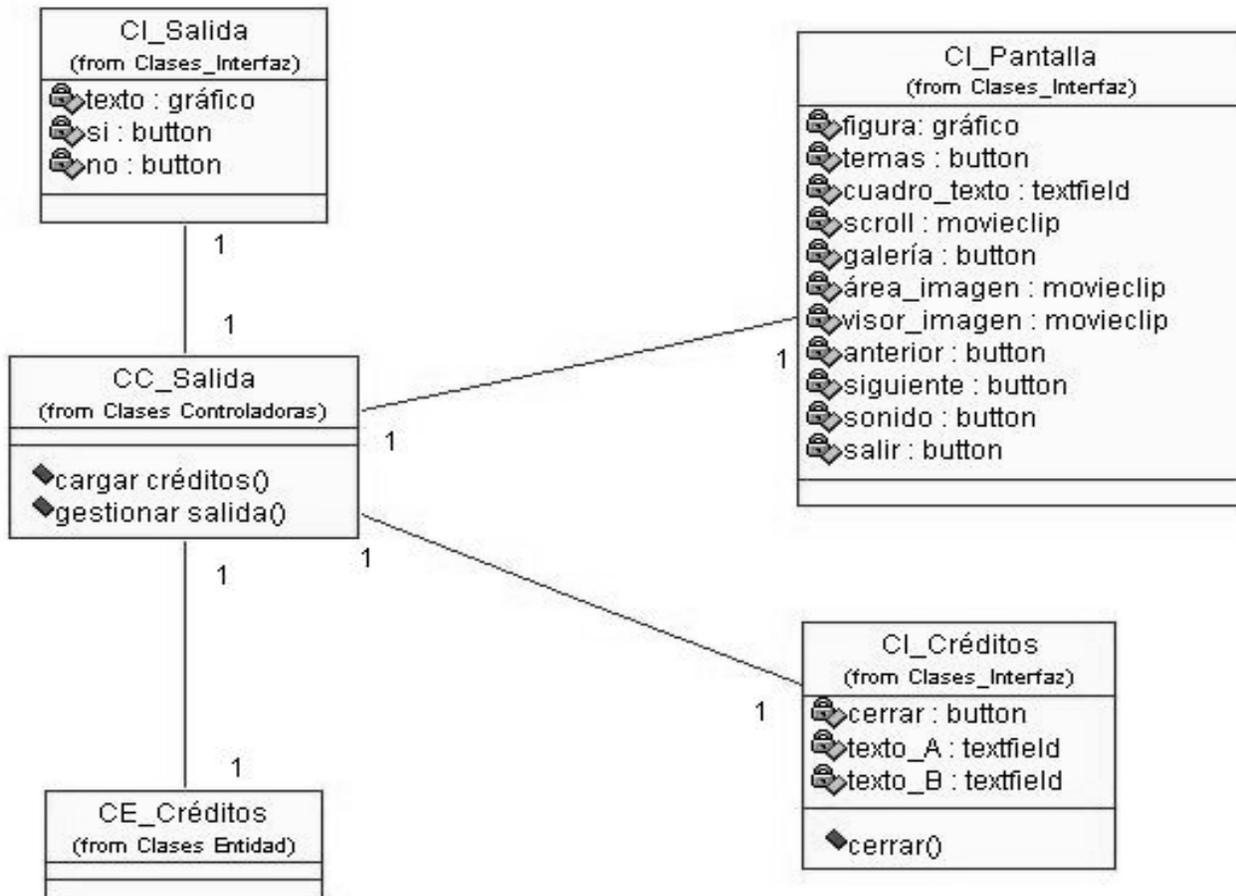
- Clases del diseño de la ventana Cursos.



- Clases del diseño que modelan y controlan toda la información.



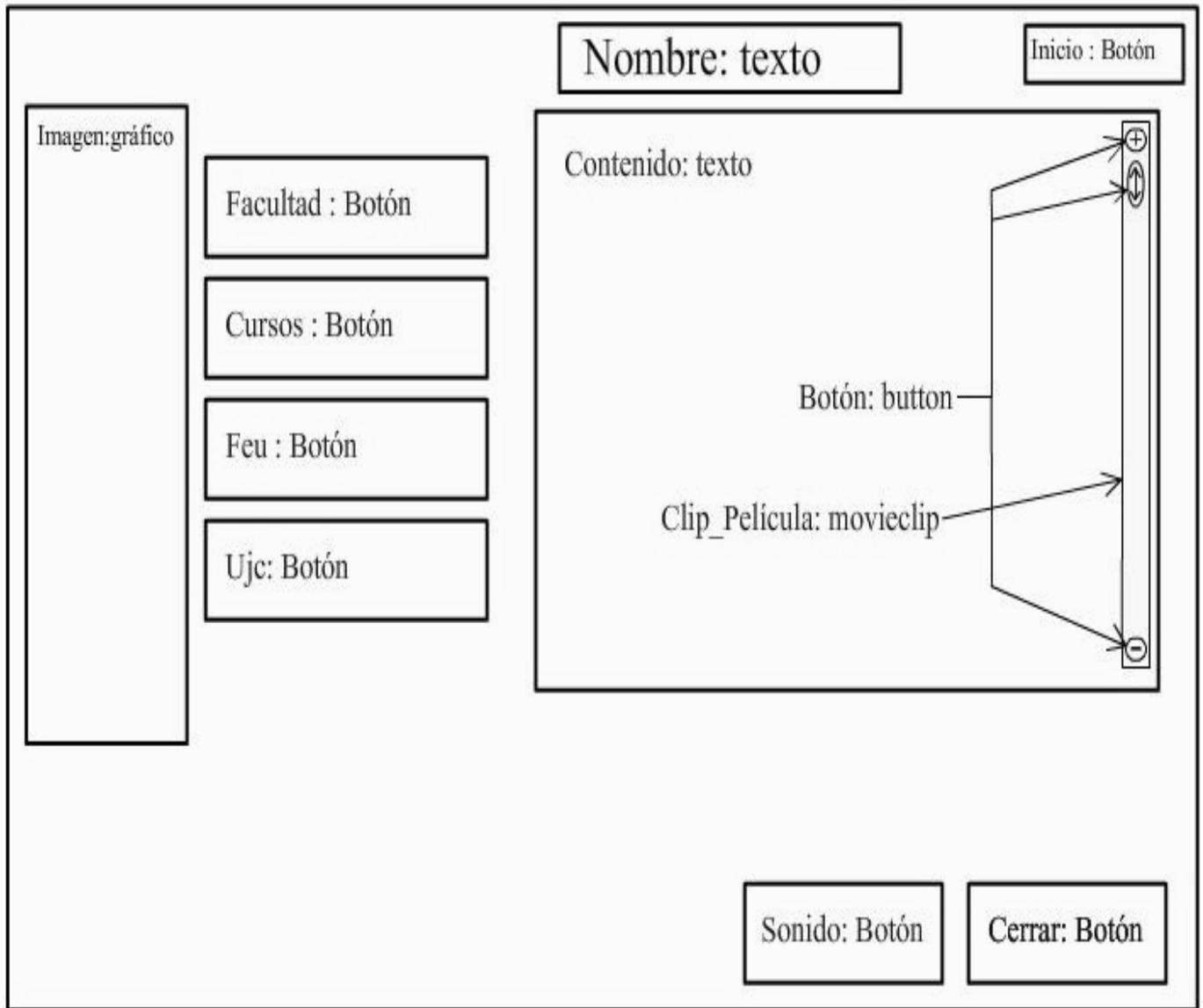
- Clases del diseño de la ventana Salir.



4.2.2 Diagramas de presentación.

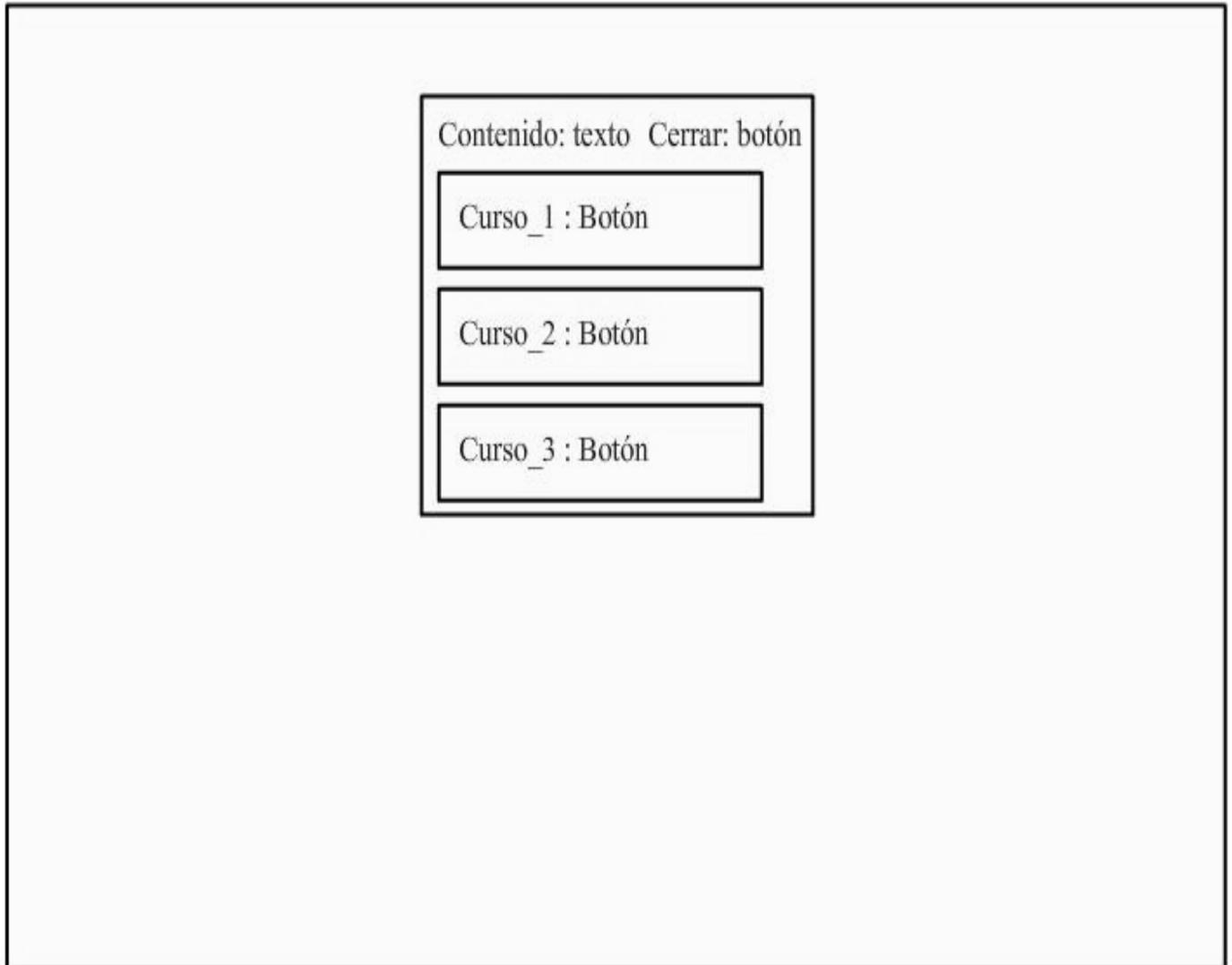
Artefacto nuevo dentro del lenguaje UML, es específico de OMMMA – L, y sirve, como se explicó en la fundamentación teórica, para describir la parte estática del modelo a través de una descripción intuitiva de la distribución espacial de objetos visuales de la interfaz de usuario. Aunque UML especifica propuesta de interfaz de usuario en sus requisitos no funcionales, no es un aspecto de fuerte medición, ni consideración en el análisis de la arquitectura del software.

-Diagrama de presentación Pantalla Principal.



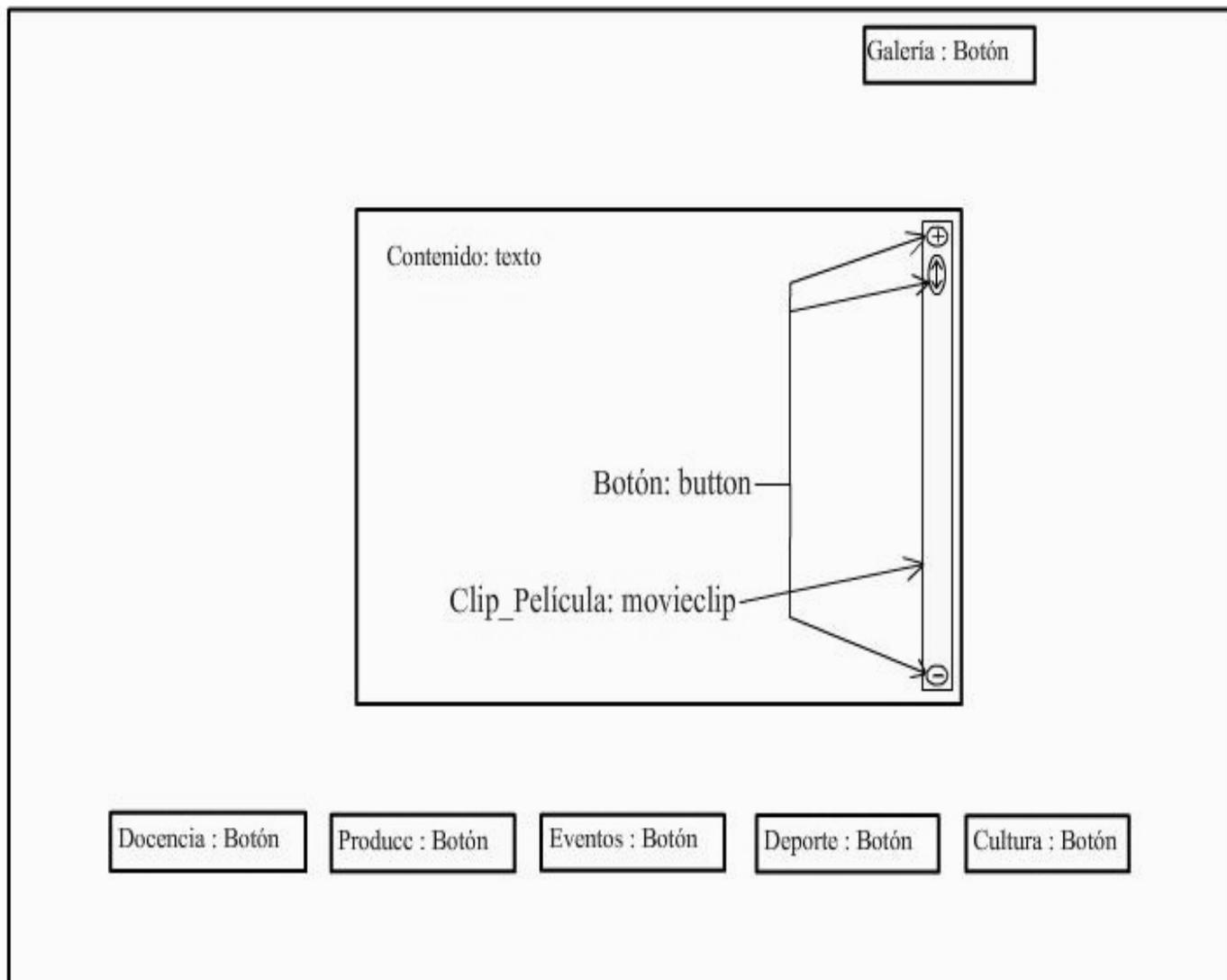
 → ambiente : sonido

-Diagrama de presentación contenido Cursos.



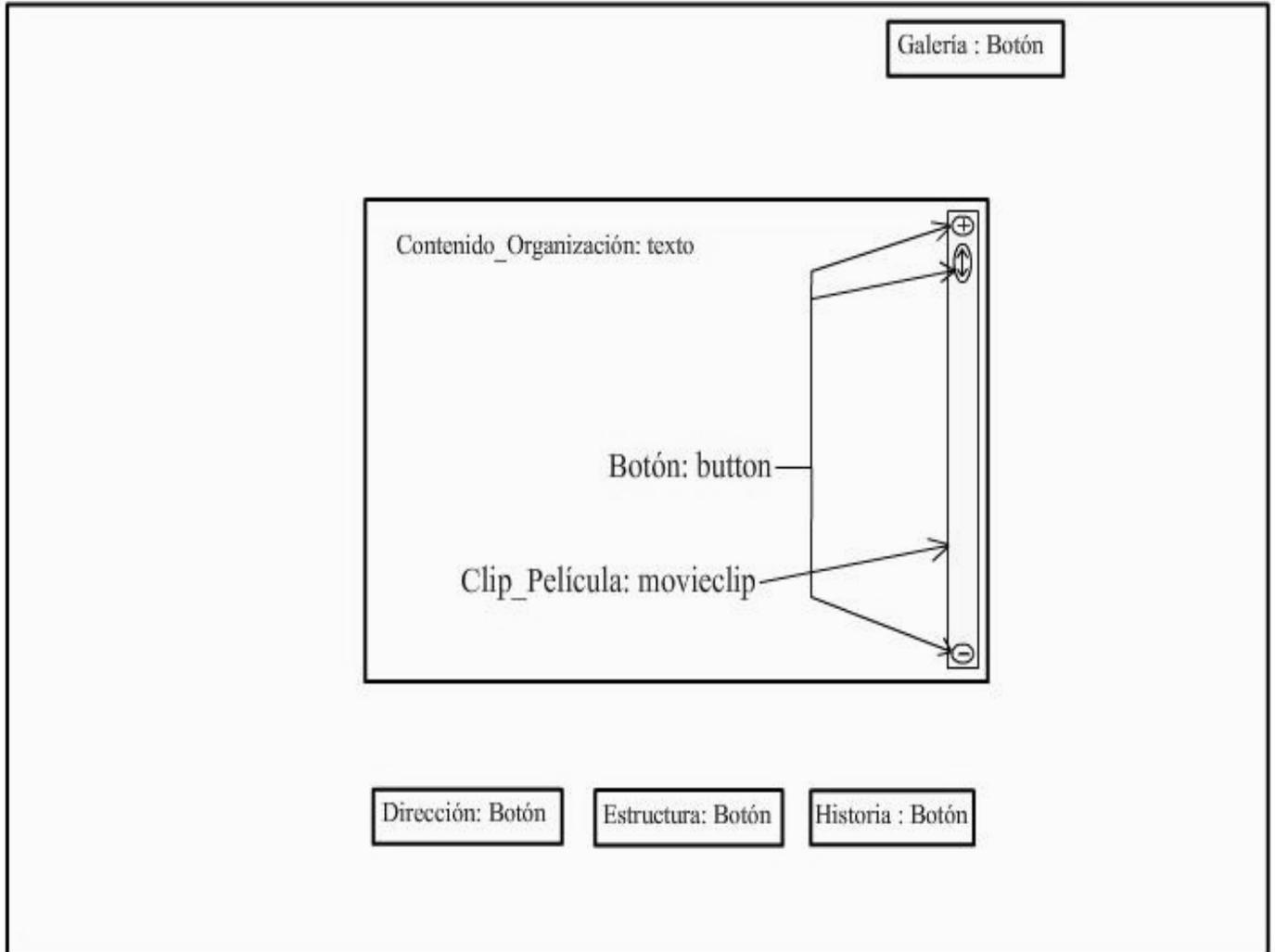
 → ambiente : sonido

- Diagrama de presentación módulo CURSO.



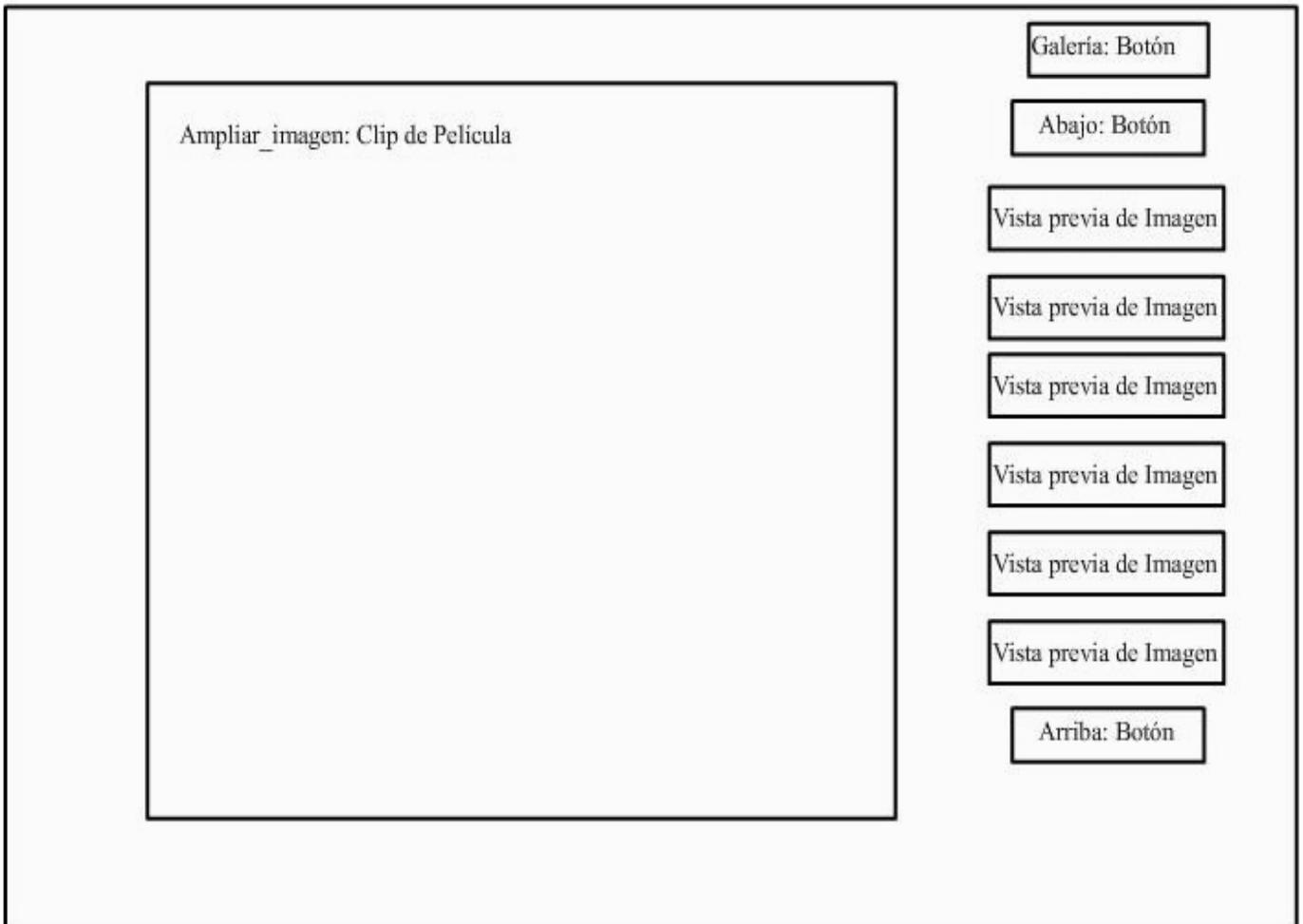
 → ambiente : sonido

-Diagrama de presentación para el contenido de una Organización Política.



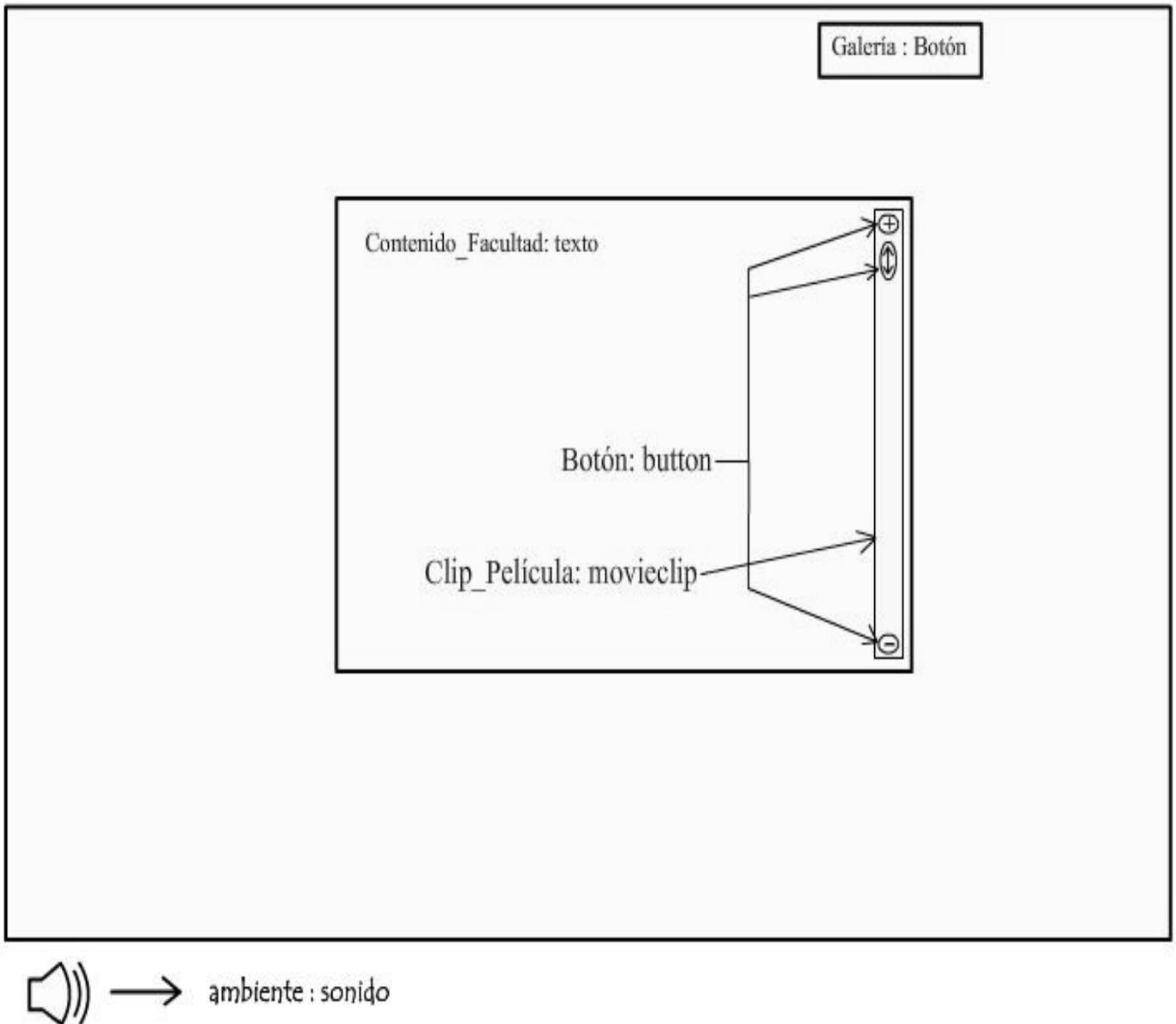
 → ambiente : sonido

-Diagrama de presentación Galería de imágenes.

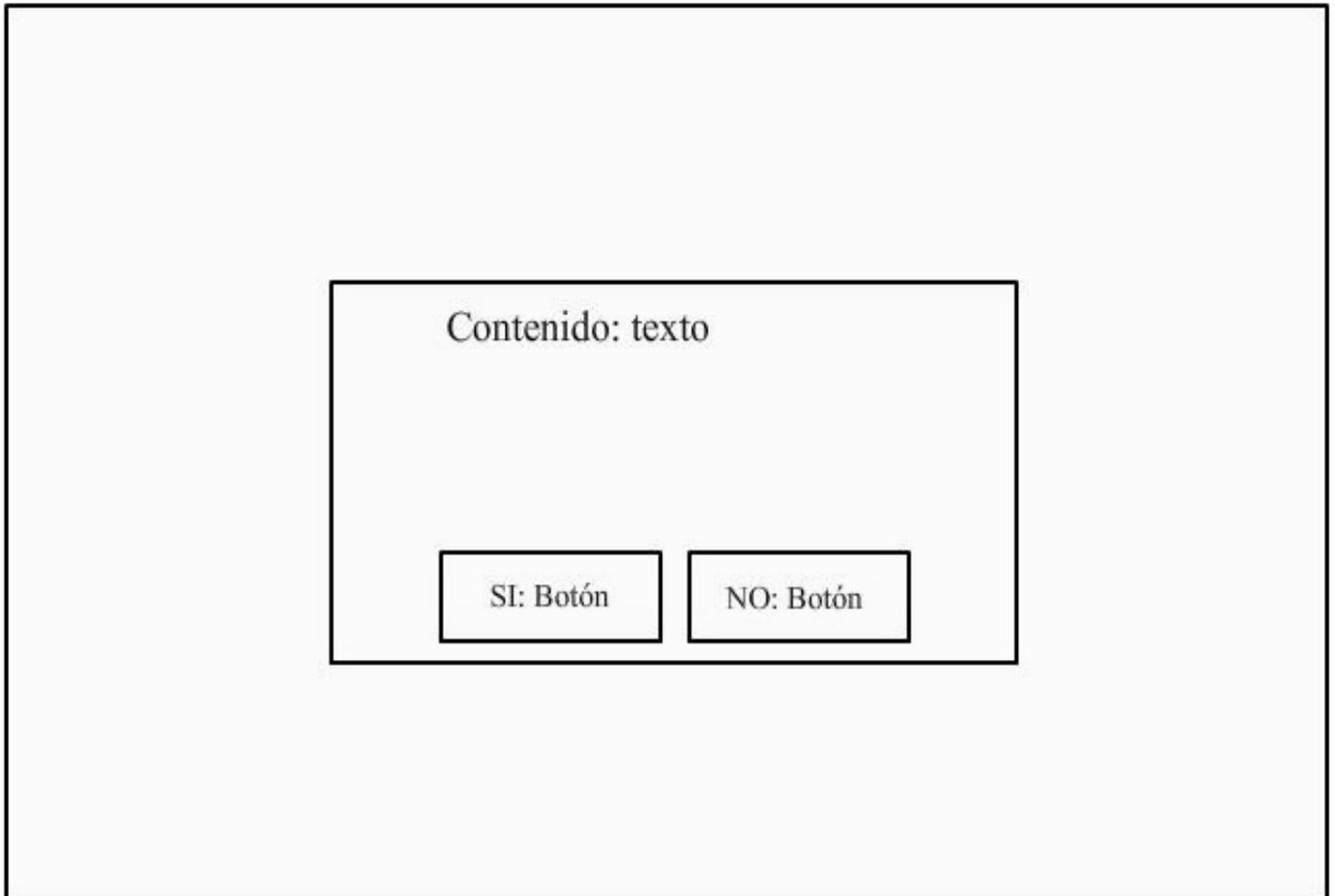


 → ambiente : sonido

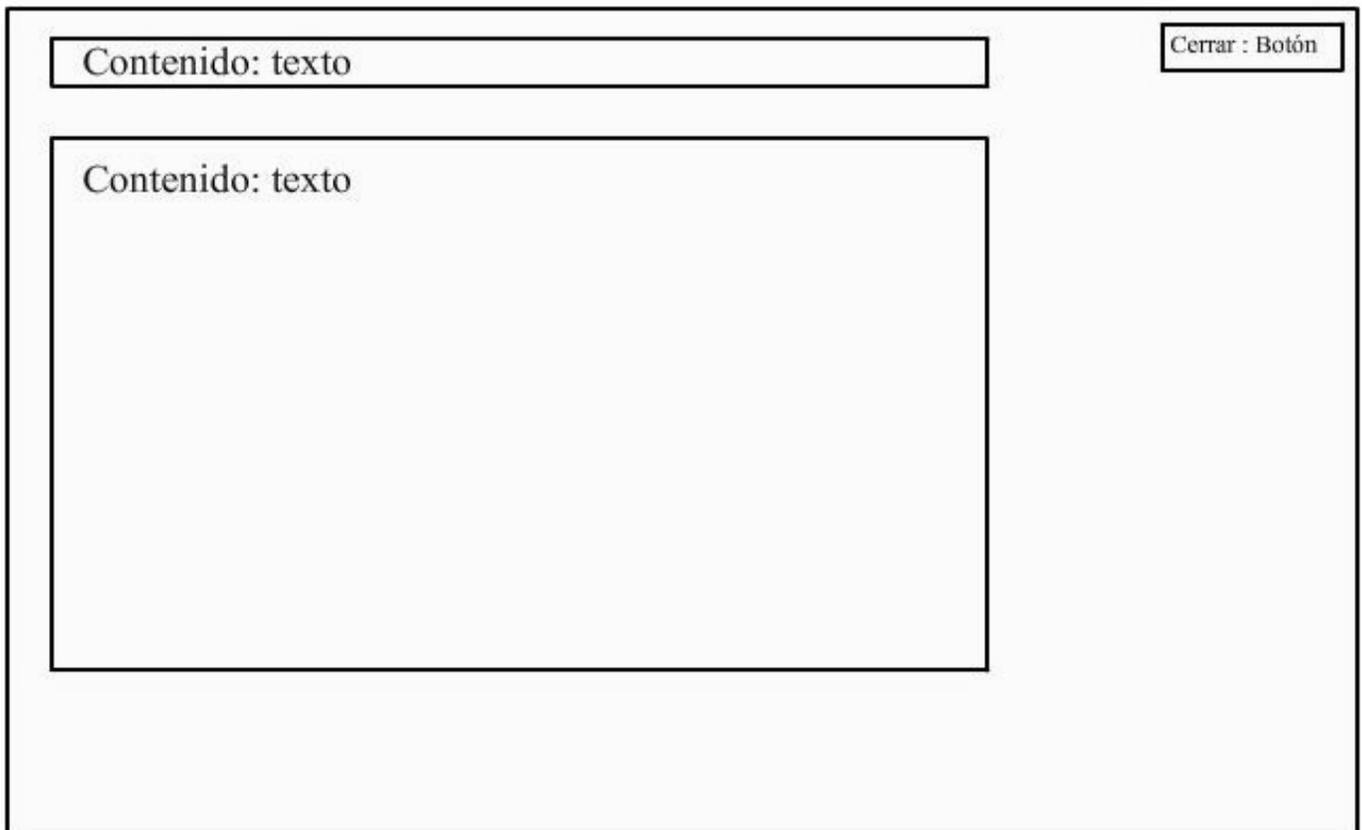
-Diagrama de presentación para el contenido de Facultad.



-Diagrama de presentación para la ventana Salir.



-Diagrama de presentación para la ventana Créditos.



4.3 Diagrama de jerarquía de clases.

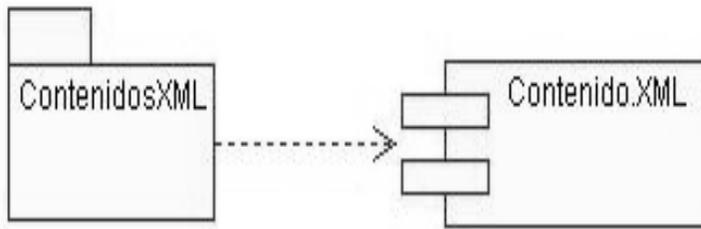
Diagrama de Clases de OMMMA-L, extendido del Diagrama de Clases de UML, utiliza las mismas notaciones, pero incorpora las clases correspondientes a las medias: media continua y media discreta, generalizadas en una clase medias. (Ver anexo 2)

4.4 Modelo de implementación.

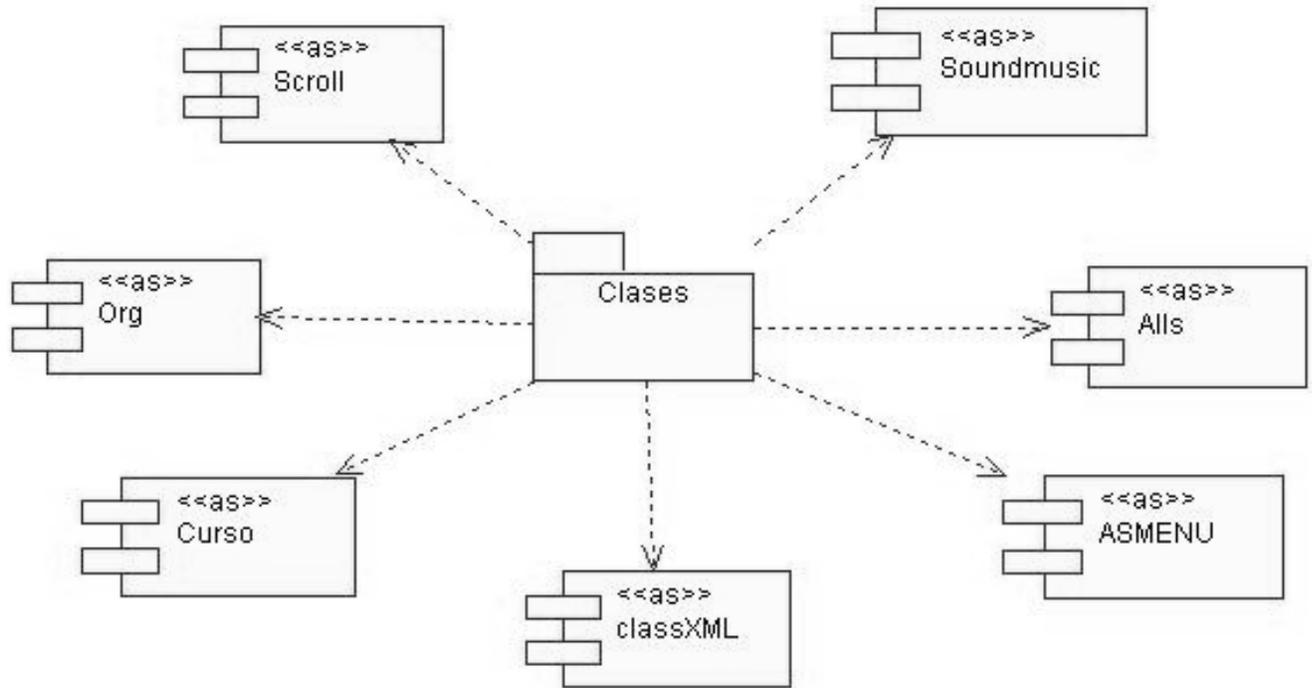
4.4.1 Diagramas de componentes.

Un diagrama de componentes representa la separación de un sistema de software en componentes físicos por ejemplo, archivos, cabeceras, módulos, paquetes, etc. En el diagrama de componentes se modelan componentes del sistema, a veces agrupados por paquetes, y las dependencias que existen entre componentes y paquetes de componentes.

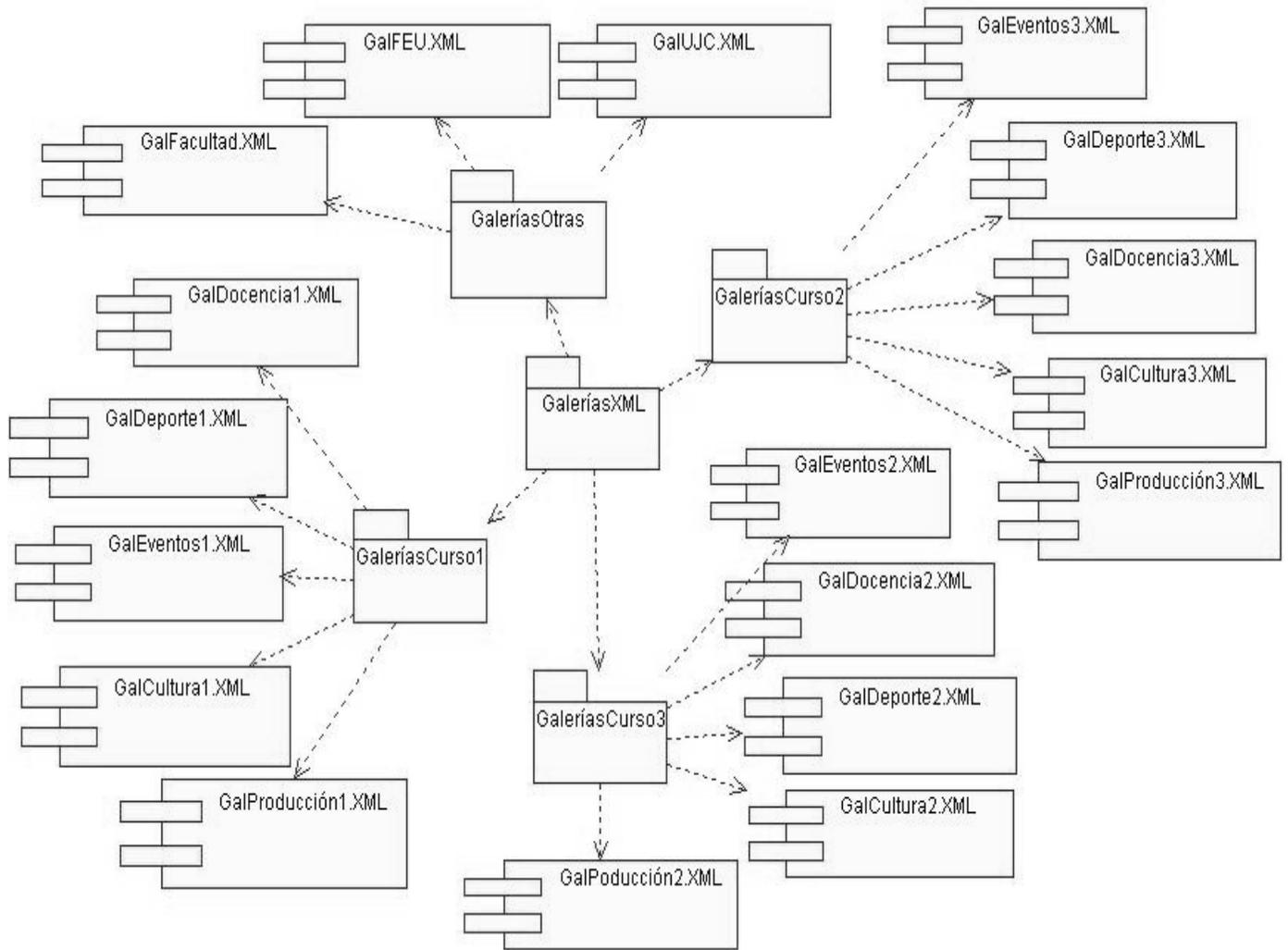
- Diagrama del componente Contenido.



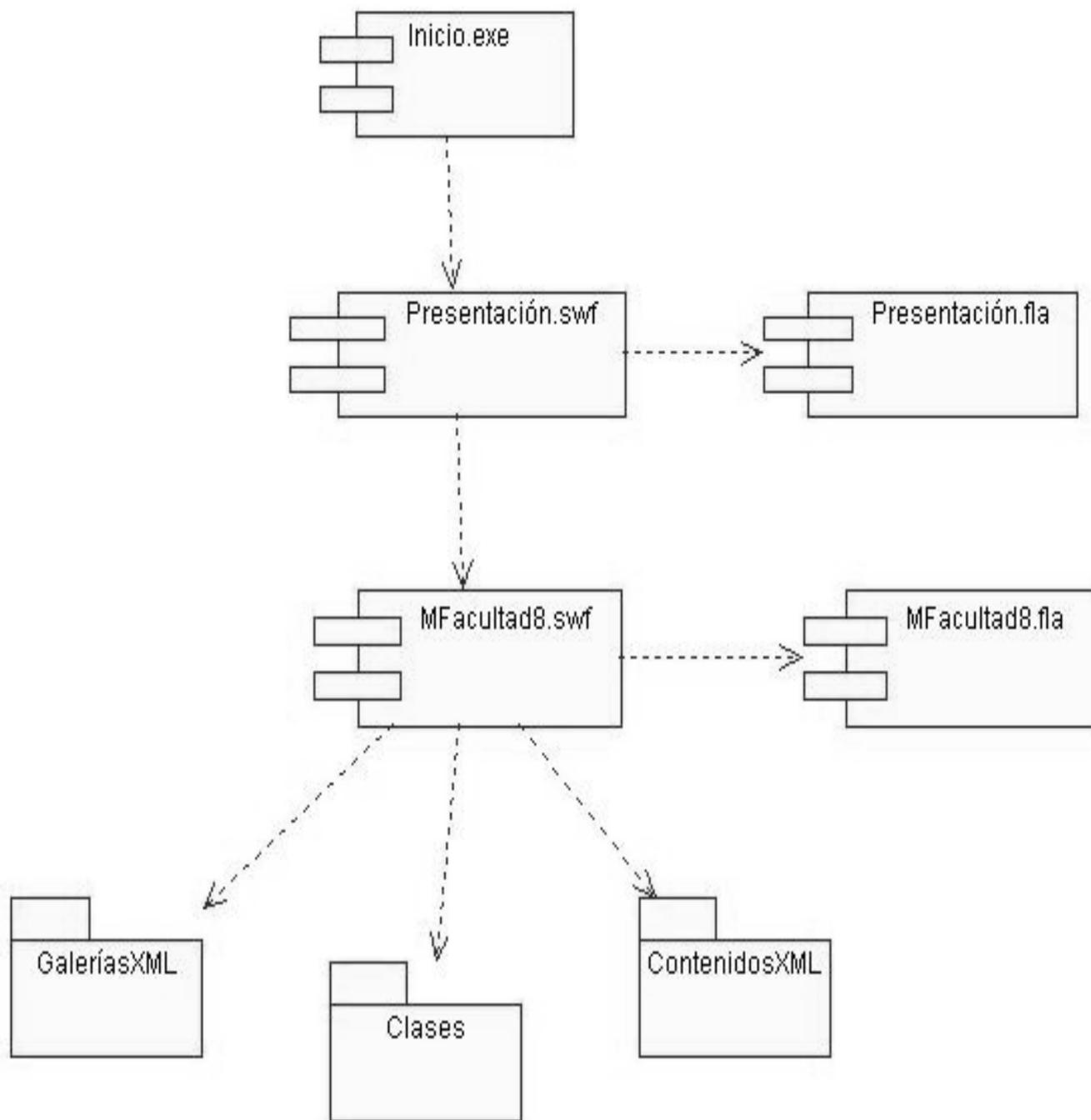
- Diagrama de los componentes Clases.



- Diagrama de paquetes de los componentes galerías.



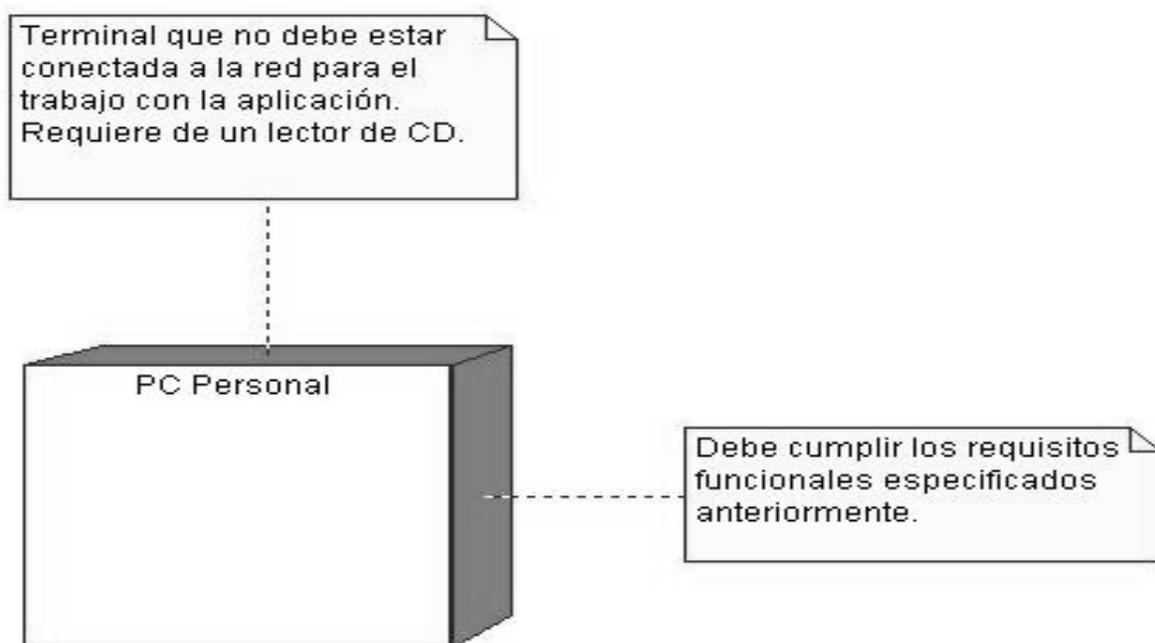
- Diagrama de componentes general.



4.5 Modelo de despliegue.

4.5.1 Diagrama de despliegue.

Un diagrama de despliegue describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. En el caso de la propuesta de la aplicación, se necesita sólo una computadora personal que cumpla los requisitos no funcionales mencionados anteriormente.



4.6 Diagramas de secuencia.

Los diagramas de secuencia son usados para especificar el comportamiento predefinido de una aplicación multimedia. Además de mantener su semántica original en el paso de mensajes para las interacciones. Estos se generan a partir de los diagramas de colaboración. Muestra las interacciones a través de un cerco o muro. (Ver anexo 3)

4.7 Descripción de la organización de los archivos XML de la aplicación.

La organización del archivo .xml Contenidos de la aplicación consiste en un solo archivo donde se ubicaron todos los contenidos que aparecen en la multimedia organizados por direcciones, por donde se hace la búsqueda para la consiguiente carga del texto que desee ver el usuario.

La organización de los archivos .xml Galerías de la aplicación consisten en varios archivos, guardados cada uno en la carpeta donde se guardan las imágenes del contenido al que se esté accediendo. Estos contienen las direcciones de las imágenes que aparecerán en pantalla cuando el usuario acceda a la galería de imágenes que desee.

4.8 Descripción del archivo XML de la aplicación.

4.8.1 Descripción del archivo XML General.

El principal objetivo de este archivo es almacenar el contenido a mostrar en las pantallas que ofrece la aplicación al usuario, en la cual éste puede encontrar toda la información que desea sobre la historia de la facultad. Este está dividido en cuatro módulos, los cuales, cada uno tiene información específica. El archivo está estructurado de la siguiente forma:

```
<facultad>
  <Introducción>
    <texto> Información referente a la estructura de la facultad </texto>
  </Introducción>
  <cursos>
    <introducción>
      <texto> breve introducción sobre este curso</texto>
    </introducción>
    <docencia>
      <texto> datos de docencia sobre este curso</texto>
```

```
</docencia>
<producción>
    <texto> datos de producción sobre este curso</texto>
</producción>
<eventos>
    <texto> datos sobre los eventos de este curso</texto>
</eventos>
<deporte>
    <texto>datos sobre los deportes en este curso</texto>
</deporte>
<cultura>
    <texto> información sobre la cultura de este año</texto>
</cultura>
</cursos>
<organizaciones políticas>
    <dirección>
        <texto> información sobre la organización</texto>
    </dirección>
    <estructura>
        <texto>información sobre la feu</texto>
    </estructura>
    <historia>
        <texto>datos de la historia de la feu</texto>
```

```
</historia>
</organizaciones políticas>
</facultad>
```

4.8.2 Descripción del archivo XML Galería.

Este XML almacena todas las direcciones de las imágenes de la pantalla Galería para todos los contenidos que guarda la aplicación, además de guardar el pie de foto correspondiente a cada imagen. El archivo está estructurado de la forma siguiente:

```
<contenido>
  <galería>
    <imágenes>
      <imagen id>dirección donde se encuentra la foto</imagen>
    </imágenes>
  </galería>
</contenido>
```

4.9 Conclusiones.

En este capítulo se realizó la ingeniería de software de la aplicación. Se desarrollaron los diagramas de presentación de la aplicación describiendo como se distribuyen las funcionalidades de ésta por cada una de las pantallas que se muestran al usuario, el diagrama de clases utilizadas y su relación con la base de datos XML, los diagramas de secuencia para el desarrollo de cada uno de los casos de uso descritos en el capítulo anterior y los diagramas de componente y despliegue para la aplicación describiendo el flujo de

implementación y como quedará desplegado el sistema cuando el cliente haga uso de éste. Además se exponen los principios del diseño utilizados en el diseño de la interfaz visual de la aplicación.

Capítulo 5

Estudio de factibilidad.

5.1 Introducción.

En la planificación de un proyecto siempre se deben tener en cuenta muchos factores que permiten que éste se lleve a cabo con los menores riesgos posibles pero los más importantes son: el costo, el tiempo de desarrollo y el esfuerzo humano que requiera la realización del mismo. Este capítulo se dedicará al estudio de la factibilidad del proyecto de tesis presentado mediante el modelo de COCOMO II, análisis de Puntos de Casos de Uso, el cual calcula el tiempo estimado mediante el análisis de varios factores.

5.2 Planificación.

La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto por Gustav Kemer de Objectory AB. Método que ha sido pulido por otros muchos actores. Para la planificación del proyecto de tesis se usará este método, el cual es muy eficiente debido a que analiza todos los factores que afectan el tiempo requerido para el proyecto.(PERALTA 2006)

5.3 Puntos de Casos de Uso.

La estimación mediante el análisis de Puntos de Casos de Uso se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. (PERALTA 2006)

Pasos:

1. Cálculo de Puntos de Casos de Uso sin ajustar.

Ecuación:

$UUCP = UAW + UUCW$ donde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

UUCW: Factor de Peso de los Casos de Uso sin ajustar.

- Calcular factor de Peso de los Actores sin ajustar (UAW).

Tipo de actor	Descripción	Factor de peso	Valor asignado	Comentario
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación.	1		
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2		
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	3	Usuario que interactúa con la interfaz gráfica.

Este valor se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos y en el caso de la multimedia es un actor complejo que tiene como valor de peso 3, o sea una persona que interactúa con el sistema mediante una interfaz gráfica, por tanto el factor de peso es igual a:

$$UAW=1*3=3$$

-Calcular factor de Peso de los Casos de Uso sin ajustar (UUCW).

Tipo de Caso de Uso.	Descripción	Factor de peso	Valor asignado	Comentario
Simple	El caso de uso contiene de 1 a 3 transacciones.	5	5	Los casos de uso tienen menos de 3 transacciones.
Medio	El caso de uso contiene de 4 a 7 transacciones.	10		
Complejo	El caso de uso contiene más de 8 transacciones.	15		

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos, que depende la cantidad de transacciones que tengan estos. Los casos de uso definidos poseen de 1 a 2 transacciones por lo que tienen peso 5 y se definieron 3 casos de uso simples, por lo tanto el factor de peso es igual a:

$$UUCW = 3 \times 5 = 15$$

Finalmente los Puntos de Casos de Uso sin ajustar resultan:

$$UUCP = UAW + UUCW = 3 + 15 = 18$$

2. Cálculo de Puntos de Casos de Uso ajustados.

Ecuación:

$$\mathbf{UCP = UUCP \times TCF \times EF}$$
 donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

Calcular el factor de complejidad técnica (TCF).

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

Factor	Descripción	Peso	Valor asignado	Comentario
T1	Sistema distribuido.	2	0	Sistema centralizado.
T2	Obj. de performance o tiempo de respuesta.	1	5	Velocidad normal.
T3	Eficiencia del usuario final.	1	5	Pocas restricciones de eficiencia.
T4	Procesamiento interno complejo.	1	0	No posee complejidad.
T5	Código reutilizable.	1	4	El código debe ser reutilizable.
T6	Facilidad de instalación.	0.5	5	Pocos requerimientos de

Factor	Descripción	Peso	Valor	Comentario
T7	Facilidad de uso.	0.5	5	Facilidad de instalación. Fácil de usar.
T8	Portabilidad.	2	3	Se requiere de portabilidad.
T9	Facilidad de cambio.	1	4	Requiere un bajo costo de mantenimiento.
T10	Concurrencia.	1	0	No hay concurrencia.
T11	Incluye objetivos especiales de seguridad.	1	0	Normal.
T12	Provee acceso directo a terceras partes.	1	0	No tiene acceso directo a terceras partes.
T13	Se requieren facilidades especiales de entrenamiento a usuarios.	1	1	No requiere de entrenamiento a usuarios.

El Factor de complejidad técnica se calcula mediante la siguiente ecuación:

$$TCF = 0.6 + 0.01 \times \sum (\text{Peso} \times \text{Valor asignado})$$

$$TCF = 0.6 + 0.01 \times \sum (2 \times 0 + 1 \times 5 + 1 \times 5 + 1 \times 0 + 1 \times 4 + 0.5 \times 5 + 0.5 \times 5 + 2 \times 3 + 1 \times 4 + 1 \times 0 + 1 \times 0 + 1 \times 0 + 1 \times 0)$$

$$TCF = 0.6 + 0.01 \times 29$$

$$TCF = 0.6 + 0.29$$

$$TCF = 0.89$$

-Calcular factor de ambiente (EF).

El cálculo del mismo es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5 en dependencia de las habilidades y el entrenamiento del grupo involucrado en el desarrollo lo que tiene un gran impacto en las estimaciones de tiempo.

Factor	Descripción	Peso	Valor asignado	Comentario
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	4	El equipo está familiarizado con el proyecto.
E2	Experiencia en la aplicación.	0.5	4	El equipo ya ha trabajado en esta aplicación.

E3	Experiencia en orientación a objetos.	1	4	El equipo ha trabajado orientado a objetos.
E4	Capacidad del analista líder.	0.5	5	Tiene buena experiencia.
E5	Motivación.	1	5	El equipo está motivado.
E6	Estabilidad de los requerimientos.	2	3	Se esperan cambios.
E7	Personal part-time.	-1	0	El equipo trabaja a tiempo completo.
E8	Dificultad del lenguaje de programación.	-1	3	Se usará el lenguaje ActionScript.

Ecuación:

$$EF = 1.4 - 0.03 \times \Sigma (\text{Peso} \times \text{Valor asignado})$$

$$EF = 1.4 - 0.03 \times 19.5$$

$$EF = 1.4 - 0.585$$

$$EF = 0.81$$

De los factores que afectan al Factor Ambiente de E1 a E6 ninguno está por debajo del valor medio (3) y de E7 y E8 ningún valor está por encima del valor medio por lo que la suma es 0 y se utiliza el factor de conversión 20 horas-hombres/puntos de casos de uso, o sea un punto de caso de uso toma 20 horas-hombres.

Finalmente los puntos de casos de uso resultan:

$$UCP = UUCP \times TCF \times EF$$

$$UCP = 18 \times 0.89 \times 0.81$$

$$UCP = 12.97$$

3. El esfuerzo en horas-hombre viene dado por:

$$E = UCP \times CF \text{ donde:}$$

E: esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: factor de conversión=20

$$E = 12.97 \times 20$$

$$E = 260 \text{ Horas-Hombres}$$

Actividad	Porcentaje	Horas-Hombre
Análisis	10.00%	65
Diseño	20.00%	130
Programación	40.00%	260
Pruebas	15.00%	97.5
Sobrecarga (otras act.)	15.00%	97.5
Total	100.00%	650

$$Et = 650 \text{ Horas-Hombres}$$

4. Esfuerzo en Mes-Hombres.

La jornada laboral de un día de trabajo cuenta con 8 horas y al mes son aproximadamente 192 horas de trabajo restando los días no laborales como sábados y domingos, por lo que:

$$Et = E(\text{Horas-Hombres}) / 192 \text{ horas-mes}$$

$$Et = 650 / 192 \text{ horas-mes}$$

$$Et = 3.38 \text{ Mes-Hombres}$$

Si en la aplicación trabajan dos hombres el tiempo de desarrollo está estimado en:

$$\text{Tiempo de desarrollo} = Et / \text{cantidad de hombres}$$

$$\text{Tiempo de desarrollo} = 3.38 / 2$$

$$\text{Tiempo de desarrollo} = 1.6 \approx 2$$

La aplicación se realizará por dos hombres en un tiempo estimado de dos meses.

5.4 Salario.

Los desarrolladores de la aplicación son ingenieros recién graduados de la Universidad de las Ciencias Informáticas, por lo que el salario mensual se toma de \$225.00.

5.5 Costos

Para dos desarrolladores que trabajaran en la aplicación tomando el salario mensual anterior, se obtiene que los costos generados por el proyecto ascienden a:

$$Ct = \text{Salario mensual} * \text{Cantidad de hombres} * \text{Tiempo de desarrollo}$$

Ct= 225*2*2

Ct=\$900

Ct=36 C.U.C

5.4 Beneficios tangibles e intangibles.

5.4.1 Tangibles.

Este proyecto es desarrollado para la ayuda al conocimiento de la historia de la facultad 8, para poner en las manos de todos los profesores, estudiantes y trabajadores los datos históricos de todos los eventos, actividades y promociones que se han producido en todas las esferas de la facultad. Los nuevos estudiantes que ingresen a la facultad contarán con esta aplicación, que les brindará una base para conocer todo lo que se ha generado con estos casi cuatro años de trabajo, que ha tenido para el bien y el avance de la facultad debido a que permitirá recoger en un formato digital todos los datos que históricamente ha ido recopilando la facultad sobre los eventos, actividades y todo lo demás. Además, según los costos calculados \$900(moneda nacional) y 36CUC (moneda convertible), la aplicación es factible si se comercializara.

5.4.2 Intangibles.

Ayudará a la ampliación del conocimiento de todos los integrantes de la facultad, permitirá que cualquier persona que desee conocer a cerca del trabajo realizado en ésta pueda basarse en la aplicación, para conocer los detalles de todo lo que sea llevado a cabo por las organizaciones estudiantiles y políticas que la componen, sus estudiantes, profesores y trabajadores.

5.5 Análisis de costos y beneficios.

El desarrollo de esta aplicación con tecnología multimedia no requiere de grandes gastos de tiempo ni recursos y los costos no son de gran envergadura, es una aplicación fácil de usar para los usuarios por su simple navegabilidad y la base de datos utilizada es XML, la cual brinda servicios de acceso rápido y prescinde de gestor de base de datos, además permitirá la recuperación y centralización de toda la información respecto a la facultad y mostrarla a todos los interesados en este tema aspectos por lo que es factible desarrollar esta aplicación.

5.6 Conclusiones.

Este capítulo se dedicó al estudio de la factibilidad del proyecto de tesis presentado mediante el modelo de COCOMO II, análisis de Puntos de Casos de Uso, el cual, después de un análisis detallado de cada uno de los factores que están implicados en el desarrollo del producto y de asignarle valores de acuerdo al peso que tienen, permitió calcular el tiempo de desarrollo y el costo que generará la aplicación con la participación de dos programadores. Además, este cálculo permitió hacer un estudio de los grandes beneficios tangibles e intangibles que acarrea el desarrollo de esta aplicación.

Conclusiones.

Después de la investigación realizada sobre el desarrollo de una aplicación utilizando tecnología multimedia y basada en la metodología RUP (Proceso Unificado de Desarrollo de Software) y el Lenguaje de Modelado (UML) y como extensión de éste el Lenguaje de Modelado Orientado a objetos de Aplicaciones Multimedia (OMMMA-L) se arriban a las conclusiones:

- Los antecedentes en la facultad 8 para la centralización de la información generada por todas las actividades que se han desarrollado todos estos años son muy escasos y con la investigación realizada en este trabajo se logró, después de realizar una búsqueda, recopilar toda esta información.
- Toda esta información almacenada en distintos lugares, después de una exhaustiva clasificación, se presentó en una aplicación, utilizando tecnología multimedia, de forma sencilla, amena y que el usuario puede interactuar con ella para adquirir la información que busca.
- Con el desarrollo de esta aplicación se logrará un aumento de los conocimientos de estudiantes, profesores y trabajadores de la facultad acerca de la historia de ésta, podrán conocer detalles del arduo trabajo que se realiza en la facultad para lograr que sea una de las mejores.
- Esta aplicación se desarrollará utilizando el lenguaje XML (Lenguaje de Marcas Extensible) de forma que, en el futuro pueda ser extensible. La información, registrada en la aplicación, se puede incrementar y actualizar fácilmente sin necesidad de modificar el diseño previamente propuesto.

Después de un largo estudio se ha llegado la conclusión que la propuesta de realizar la aplicación consuma el objetivo propuesto: centralizar toda la información referente a la facultad en una aplicación con tecnología multimedia y mostrar esta información a todos los interesados en conocer el historial de la facultad.

Recomendaciones.

La aplicación desarrollada cumple con las expectativas de centralizar toda la información acerca de la facultad 8 y brindarla a los usuarios, de forma sencilla, amena e interactiva, sin embargo se deben tener en cuenta algunas recomendaciones para el desarrollo de futuras aplicaciones o sistemas como:

- Volver a realizar una búsqueda exhaustiva de documentación que se ha generado en la facultad para lograr una mayor exquisitez en la información que se brinda a los usuarios.
- Perfeccionar los estudios de las herramientas utilizadas para el logro de una aplicación mucho más completa.
- Continuar estudios del lenguaje OMMMA-L para futuros desarrollos de aplicaciones con tecnología multimedia.
- Permitir al usuario la búsqueda de información más específicamente, o sea, hacer la aplicación lo más interactiva posible.
- Divulgar la información referente a la facultad 8 de la Universidad de las Ciencias Informáticas contenida en la aplicación propuesta, no sólo en la universidad, sino también llevarla a las escuelas de informática del país para lograr una mayor motivación de los estudiantes para el estudio de la Informática en la universidad.

Referencias bibliográficas

ALEAN. *NeoBook Pro (4.12) - Windows 95/98/2000/NT*, 2006. [Disponible en: <http://www.alean.com.ar/neosoftargentina.htm>]

ALEGSA. *Imagen*, 2007. [Disponible en: <http://www.alegsa.com.ar/Diccionario/diccionario.php>]

BENDAHAN, M. *Proceso de desarrollo de software.*, 1997. [Disponible en: <http://www.monografias.com/trabajos5/desof/desof.shtml>]

CANÓS, J. H.; P. LETELIER, *et al. Metodologías Ágiles en el Desarrollo de Software.*, 2004. [Disponible en: <http://www.willydev.net/descargas/prev/TodoAgil.Pdf>]

CÁRDENAS, G. C. *Los Compendios Informativos con tecnología multimedia como un servicio de alto valor añadido.*, 2003. [Disponible en: <http://www.congreso-info.cu/UserFiles/File/Info/Info2004/Ponencias/175.pdf>]

CUARESMA., M. J. E. *Metodologías para el desarrollo de sistemas de información global: análisis comparativo y propuesta.*, 2001. [Disponible en: <http://lsiweb.lsi.us.es/docs/informes/EstadoActual.pdf>]

DIGITAL, D. G. D. G. *Estándares de codificación de sistemas*, 2006. [Disponible en: <http://www.chubut.gov.ar/gobiernodigital/documentos/EstandaresCodificacion.pdf>]

EMAGISTER.COM. *XML - Lenguaje de Marcas Extensible.*, 2000. [Disponible en: <http://www.emagister.com/xml-lenguaje-marcas-extensible-cursos-317471.htm>]

FERNÁNDEZ, G. S. and S. D. CATALÁ. *MULTIMEDIA AUTO-APRENDE.*, 2006. p.

FRANQUET, R. *Multimedia: luces y sombras de un sector estratégico.*, 2003. [Disponible en: <http://www.audiovisualcat.net/publicaciones/Q15castfranquet.pdf>]

GARCÍA, C. M. *Hipermedia.*, 2004. [Disponible en: http://www.telecable.es/personales/carlosmg1/glosario_h.htm]

JACOBSON, I.; G. BOOCH, *et al. El proceso unificado de desarrollo de software.* 1995. p.

LETELIER, P. and M. D. C. PENADÉS. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*, 2005. [Disponible en: <http://www.willydev.net/descargas/masyxp.pdf>]

LORENZO, G. *Animación.*, 2007. [Disponible en: <http://www.lorenzoservidor.com.ar/info01/diccio-a-c.htm>]

MARTÍNEZ, G. M. *Ingeniería de SoftwareUML*, 1998. [Disponible en: <http://www.monografias.com/trabajos5/insof/insof.shtml>]

MELCHOR., M. A. I. *MetaCard y Revolution.*, 2004. [Disponible en: <http://www.disca.upv.es/magustim/mmmultiplataforma/book1.htm>]

NETWORK, M. P. *Video.*, 2006. [Disponible en: http://www.marcelopedra.com.ar/glosario_V.htm]

PERALTA, M. *ESTIMACIÓN DEL ESFUERZO BASADA EN CASOS DE USO*, 2006. [Disponible en: <http://www.google.com.cu/search?hl=es&q=ESTIMACION%20DEL+ESFUERZO+BASADA+EN+CASOS+DE+USO&meta>]

RODRÍGUEZ., L. A. *Evaluación de la herramienta de autor “Revolution” para el desarrollo de Hiperentornos Interactivos de Aprendizaje multiplataformas.*, 2000. [Disponible en: http://www.informaticahabana.com/evento_virtual/files/MUL072.pdf]

TECHNOLOGY, I. *Beneficios de una presentación multimedia*, 2007. [Disponible en: http://www.intellia.com.mx/esp/servicios/multimedia_y_publicidad_beneficios.php]

VIRTUAL-FORMAC. *Contenidos Dinámicos Con Actionscript 2.0 y Xml*, 2007. [Disponible en: http://www.virtual-formac.com/informatica/programacion/curso_contenidos_dinamicos_con_actionscript_2_0_y_xml-c6421.html]

WIKIPEDIA. *Adobe Flash.*, 2007 [Disponible en: http://es.wikipedia.org/wiki/Macromedia_Flash]

---. *Hipertexto.*, 2007a. [Disponible en: <http://es.wikipedia.org/wiki/Hipertexto>]

---. *Macromedia Director.*, 2007b. [Disponible en: http://es.wikipedia.org/wiki/Macromedia_Director]

---. *Multimedia.*, 2007c. [Disponible en: <http://es.wikipedia.org/wiki/Multimedia>]

---. *Proceso Unificado de Rational.*, 2007d. [Disponible en: http://es.wikipedia.org/wiki/Rational_Unified_Process]

---. *Sonido.*, 2007e. [Disponible en: <http://es.wikipedia.org/wiki/Sonido>]

---. *Texto.*, 2007f. [Disponible en: <http://es.wikipedia.org/wiki/Texto>]

---. *XML*, 2007g. [Disponible en: <http://es.wikipedia.org/wiki/XML>]

ZAMITIZ, C. A. R. *EL LENGUAJE UNIFICADO DE MODELADO (UML)*, 2003a. [Disponible en: <http://www.fi-b.unam.mx/pp/profesores/carlos/aydoo/uml.html>]

---. *EL LENGUAJE UNIFICADO DE MODELADO (UML)*, 2003b. [Disponible en: <http://www.fi-b.unam.mx/pp/profesores/carlos/aydoo/uml.html>]

Bibliografía

WIKIPEDIA. Información., 2007. [Disponible en: <http://es.wikipedia.org/wiki/Informaci%C3%B3n>

BENÍTEZ, I. C. and C. B. CARMONA. Curso, 2006. [Disponible en: http://www.umce.cl/~cipumce/cuadernos/facultad_de_historia/tecnologia_educativa/cuaderno_03/glosario_letrac.htm

WIKIPEDIA. Facultad, 2007. [Disponible en: <http://es.wikipedia.org/wiki/Facultad>

WIKIPEDIA. Trabajador, 2007. [Disponible en: <http://es.wikipedia.org/wiki/Trabajador>

WIKIPEDIA. Profesor, 2007. [Disponible en: <http://es.wikipedia.org/wiki/Profesor>

WIKIPEDIA. Estudiante, 2007. [Disponible en: <http://es.wikipedia.org/wiki/Estudiante>

WIKIPEDIA. GPS, 2007. [Disponible en: <http://es.wikipedia.org/wiki/GPS>

CITMATEL. CITMATEL, [Disponible en: <http://www.citmatel.inf.cu/empresa.php>

WIKIPEDIA. Lingo, 2007. [Disponible en: <http://es.wikipedia.org/wiki/Lingo>

WIKIPEDIA. Blog, 2007. [Disponible en: <http://es.wikipedia.org/wiki/Weblogs>

WIKIPEDIA. GIF, 2007. [Disponible en: <http://es.wikipedia.org/wiki/GIF>

WIKIPEDIA. BMP, 2007. [Disponible en: <http://es.wikipedia.org/wiki/BMP>

Glosario de términos.

A continuación se muestra el significado de varios términos usados en el documento que pueden dificultar la comprensión de su contenido.

BMP: En los sistemas operativos Windows, representan la sigla BitMaP (o también Bit Mapped Picture), o sea mapa de bits. Normalmente, se caracterizan por ser muy poco eficientes en su uso de espacio en disco, pero pueden mostrar un buen nivel de calidad.

GIF (Graphics Interchange Format): Es un formato gráfico utilizado ampliamente en la World Wide Web, tanto para imágenes como para animaciones.

MP3 (MPEG-1 Audio Layer 3): Es un formato de audio digital comprimido con pérdida desarrollado por el Moving Picture Experts Group (MPEG) para formar parte de la versión 1 (y posteriormente ampliado en la versión 2) del formato de vídeo MPEG. Su nombre es el acrónimo de MPEG-1 Audio Layer 3.

JPEG (Joint Photographic Experts Group): Algoritmo diseñado para comprimir imágenes con 24 bits de profundidad o en escala de grises. JPEG es también el formato de fichero que utiliza este algoritmo para comprimir imágenes.

GPS(Sistema de posicionamiento global): Es un Sistema Global de Navegación por Satélite (GNSS) el cual permite determinar en todo el mundo la posición de un objeto, una persona, un vehículo o una nave, con una precisión hasta de centímetros usando GPS diferencial, aunque lo habitual son unos pocos metros. El sistema fue desarrollado e instalado, y actualmente es operado, por el Departamento de Defensa de los Estados Unidos.

CITMATEL (Empresa de Tecnologías de la Información y Servicios Telemáticos Avanzados): Empresa que se distingue, por su excelencia y creciente proyección hacia el mercado externo e interno con una amplia diversidad de productos y servicios integrales de alto valor agregado. Participa activamente en el desarrollo científico y tecnológico de la sociedad cubana con relevante liderazgo en la introducción de tecnologías de primera línea en el país y como proveedores de Internet en Cuba.

IBM (International Business Machines): Es una empresa que fabrica y comercializa hardware, software y servicios relacionados con la informática.

Lingo: Es el lenguaje de programación que lleva incorporado Macromedia Director, un programa de autoría. Permite integrar con relativa facilidad texto, imágenes, sonidos y video digital, siendo una alternativa a lenguajes más tradicionales, como el C++, porque el desarrollo de la aplicación es mucho más rápido y flexible.

Weblogs: O cuaderno de bitácora (listado de sucesos), es un sitio Web periódicamente actualizado que recopila cronológicamente textos o artículos de uno o varios autores, apareciendo primero el más reciente, donde el autor conserva siempre la libertad de dejar publicado lo que crea pertinente.

Socket: designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiarse cualquier flujo de datos, de manera fiable y ordenada. Un socket queda definido por una dirección IP, un protocolo y un número de puerto.

Transacciones: Secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia.

RF: Requerimientos funcionales.

UJC: Unión de Jóvenes Comunistas.

FEU: Federación Estudiantil Universitaria.

CI: Clase Interfaz.

CC: Clase Controladora.

CE: Clase Entidad.