



Facultad 8

Curso de Podología para la Salud

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias
Informáticas

Autores:

Dayamis Ruíz Pereira.
Lissette Nuñez Maturel.

Tutor: Ing. Michel Miranda Cairo.
Cotutor: Frank David Ávalos Palomo.

Ciudad de La Habana
Junio del 2008
“Año 50 de la Revolución”

*A nuestros padres, familiares y amigos que nos han apoyado en todo momento y siempre confiaron en nosotros.
A la revolución cubana por la oportunidad que nos dieron y a Fidel por ser nuestro guía.*

AGRADECIMIENTOS

De Lisette:

A nuestro Comandante en Jefe por tener la gran idea de forjar la Universidad de la Ciencias Informáticas.

A la personita que más quiero en este mundo, a la que confió en mí y me supo sacar siempre adelante a pesar de todo, a quien me ayudó siempre sin preguntar por qué. A quien supo quitarse todo lo que tenía y mucho más para ofrecérmelo sin esperar nada a cambio. Gracias mamita querida. Este es tu sueño y el mío.....Gracias, Nelly Maturel Rodríguez, por hacer de mí lo que soy.

A mi querido novio Dixolín por estar siempre a mi lado cuando lo necesité, por corregir mis errores, por alegrarme la vida cada minuto y por soportar mis malacrianzas. Gracias por todo mi vida...

A mi hermano Alain y a mis sobrinitos.

A mi papá....

A toda mi familia en general.

A todos mis compañeros del 8502.

A Dani, a Dayamis mi compañera de tesis, a Isyed, a Leivis.....gracias por ser mis amigas durante este tiempo.

A mis tutores Frank y Michel por ayudarnos tanto.

De Dayamis:

A la Revolución por hacer posible que personas de todas las clases sociales tengan derecho de asistir a la Universidad.

A mi mamá querida, porque sin ella no hubiera llegado hasta donde estoy.

A mis hermanas **Damaisy** y **Dayana**, y a mi padrastro **Jesús**, por apoyarme siempre.

A mi novio **Julio**, por hacerme la vida en la Universidad más placentera con su amor y cariño.

A mi papá, a mis hermanos varones, a mis abuelos y a mi abuelita que aunque ya no esté siempre se preocupó por mí.

A mi suegra **María**, por ser mi madre aquí en La Habana y por estar siempre cuando la necesito.

A mi suegro **Julio**, por preocuparse por mis problemas.

A mi tío **Wilfredo**, a mis tías **Sara** y **Eldi**, a mis primas **Yahima** y **Yahilet**, y a mis primos **Serafín** y **Osmin** por haberme acogido en sus casas durante los cinco años de carrera.

A mi amiga **Isyed**, por ayudarme en los momentos difíciles en la Universidad.

A mi compañera de tesis, y a todas mis amistades, gracias por estar cuando las necesito.

A mis tutores Frank y Michel por darnos tanto apoyo y ánimos durante el desarrollo de la tesis.

A toda mi familia en general.



“Nunca consideres el estudio como una obligación sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.”

Albert Einstein

DECLARACIÓN DE AUTORÍA

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

Declaramos que todo lo anteriormente expuesto se ajusta a la verdad, y asumimos la responsabilidad moral y jurídica que se derive de este juramento profesional.

Para que así conste firmamos la presente declaración jurada de autoría en Ciudad de la Habana a los ____ días del mes de Junio del año 2008.

Autores: Lissette Nuñez Maturel.

Dayamis Ruiz Pereira.

Tutor: Ing. Michel Miranda Cairo.

Cotutor: Ing. Frank D. Ávalos Palomo

Resumen

La Podología es la rama de la ciencia médica, que tiene por objeto el tratamiento de las afecciones y deformidades de los pies. El estudio de esta especialidad en Cuba se ve afectado pues existe poca documentación y la información se encuentra dispersa, por lo que a la hora de buscar un tema específico se hace muy difícil y engorroso. El presente Trabajo de Diploma se enmarca en desarrollar, haciendo uso de la tecnología multimedia, un software educativo que centralice toda la información existente sobre la Podología. En este software se muestran un conjunto de temas relacionados con esta disciplina, además de una serie de ejercicios a realizar por los estudiantes, con el objetivo de que estos puedan ejercitarse y que sus resultados puedan ser guardados en un fichero para ser evaluados posteriormente por el profesor. Además posee un atlas de Morfología que ilustra la localización de todas las estructuras del pie. Para dar solución a los objetivos planteados se realizó un análisis sobre las tendencias, tecnologías y metodologías actuales de las aplicaciones con tecnología multimedia. Este documento recoge todo el proceso de análisis, diseño e implementación del producto, así como un estudio de la factibilidad para determinar los beneficios tangibles e intangibles del software desarrollado.

ÍNDICE

INTRODUCCIÓN.....1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.5

INTRODUCCIÓN.....5

1.1 ANÁLISIS DE OTRAS SOLUCIONES EXISTENTES5

1.2 CONCEPTOS CLAVES6

1.2.1 ¿QUÉ ES LA PODOLOGÍA?.....6

1.2.2 ¿QUÉ ES LA TECNOLOGÍA MULTIMEDIA?.....6

1.2.3 ¿QUÉ ES MULTIMEDIA EDUCATIVA?.....9

1.2.4 HIPERTEXTO.....9

1.2.5 HIPERMEDIA.....10

1.2.6 TEXTO.....10

1.2.7 SONIDO.....11

1.2.8 IMAGEN.....11

1.2.9 ANIMACIÓN.....11

1.2.10 VIDEO.....11

1.3 METODOLOGÍAS12

1.3.1 RMM: RELATIOSHIP MANAGEMENT METHODOLOGY.....12

1.3.2 OOHDm: OBJECT ORIENTED HYPERMEDIA DESIGN MODEL.....12

1.3.3 XP: PROGRAMACIÓN EXTREMA O EXTREME PROGRAMING.....13

1.3.4 RUP: PROCESO UNIFICADO DE SOFTWARE.....14

1.4 LENGUAJES DE MODELADO16

1.4.1 UML: UNIFIED MODELING LANGUAGE.....16

1.4.2 OMMMA – L: LENGUAJE DE MODELADO ORIENTADO A OBJETOS DE APLICACIONES MULTIMEDIA.....16

1.5 HERRAMIENTAS CASE18

1.5.1 RATIONAL ROSE.....18

1.5.2 VISUAL PARADIGM.....19

1.6 HERRAMIENTAS DE AUTOR20

1.6.1 MACROMEDIA DIRECTOR.....20

1.6.2 TOOLBOOK.....20

1.6.3. FLASH DEVELOP.....21

1.6.4 MTASC	21
1.6.5 MACROMEDIA FLASH.....	22
1.6.5.1 Macromedia Flash 8.0.....	23
1.7 HERRAMIENTAS PARA EL DISEÑO DE LA INTERFAZ.....	24
1.7.1 SCREENWEAVER 3 OS	24
1.7.2 SONY VEGA.....	25
1.8 LENGUAJES DE MARCAS.....	25
1.8.1 HTML	25
1.8.2 XML.....	26
1.9 LENGUAJE DE PROGRAMACIÓN.....	27
1.9.1 ACTIONSCRIPT.....	27
1.9.1.1 ActionScript 2.0:	28
1.10 TECNOLOGÍAS A UTILIZAR PARA LA SOLUCIÓN PROPUESTA	28
1.11 ANÁLISIS DEL MODELO PEDAGÓGICO.	29
1.12 CONCLUSIONES PARCIALES	30
CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....	31
INTRODUCCIÓN.....	31
2.1 ESPECIFICACIÓN DEL CONTENIDO	31
2.2 DESCRIPCIÓN DEL SISTEMA PROPUESTO.....	32
2.3 REQUERIMIENTOS	32
2.3.1. REQUERIMIENTOS FUNCIONALES:.....	33
2.3.2 REQUERIMIENTOS NO FUNCIONALES:.....	34
2.4 MODELO CONCEPTUAL.	35
2.4.1 MODELO DE NEGOCIO.....	35
2.4.2 ¿QUÉ ES EL MODELO DE DOMINIO?	36
2.4.3 ¿MODELO DE NEGOCIO O MODELO DE DOMINIO?	36
2.4.4 MODELO DE DOMINIO	37
2.4.4.1 Conceptos del dominio.....	37
2.5 MODELO DE CASO DE USO DEL SISTEMA:.....	38

2.5.1 DETERMINACIÓN Y JUSTIFICACIÓN DE LOS ACTORES DEL SISTEMA.....	39
2.5.2 DIAGRAMA DE CASOS DE USO.	40
2.5.3 DESCRIPCIÓN Y EXPANSIÓN DE LOS CASOS DE USO.	41
2.6 DIAGRAMA DE NAVEGACIÓN.....	47
2.6.1 DIAGRAMA DE NAVEGACIÓN GENERAL.	48
2.6.2 DIAGRAMA DE NAVEGACIÓN: EJERCICIOS.....	49
2.7 CONCLUSIONES PARCIALES	49
CAPÍTULO 3: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.....	51
INTRODUCCIÓN.....	51
3.1 DIAGRAMAS DE PRESENTACIÓN	51
3.2 DIAGRAMA DE JERARQUÍA DE CLASES.	64
3.3 DIAGRAMA DE CLASES DE DISEÑO	65
3.4 DIAGRAMAS DE SECUENCIA	70
3.5 DIAGRAMA DE DESPLIEGUE	71
3.6 MODELO DE IMPLEMENTACIÓN.....	72
3.6.1 <i>Diagrama de Componentes</i>	72
3.7 DESCRIPCIÓN DE ARCHIVOS XML.....	76
3.7.1 <i>Introducción</i>	76
3.7.2 <i>Capítulo 1</i>	76
3.7.3 <i>Capítulo 2</i>	76
3.7.4 <i>Capítulo 3</i>	76
3.7.5 <i>Capítulo 4</i>	76
3.7.6 <i>Capítulo 5</i>	77
3.7.7 <i>Capítulo 6</i>	77
3.7.8 <i>Ayuda</i>	77
3.7.9 <i>Glosario</i>	77
3.8 CONCLUSIONES PARCIALES.....	78
CAPÍTULO 4: ESTUDIO DE LA FACTIBILIDAD.	79
INTRODUCCIÓN.....	79
4.1 PLANIFICACIÓN	79
4.2 CÁLCULO DE PUNTOS DE CASOS DE USOS SIN AJUSTAR	79
4.2.1. <i>Factor de Peso de los Actores sin ajustar (UAW)</i>	80
4.2.2. <i>Factor de Peso de los Casos de Uso sin ajustar (UUCW)</i>	81

4.3 CÁLCULO DE PUNTOS DE CASOS DE USO AJUSTADOS	82
4.3.1 <i>Factor de complejidad técnica (TCF):</i>	82
4.3.2 <i>Factor ambiente (EF)</i>	83
4.4 DE LOS PUNTOS DE CASOS DE USO A LA ESTIMACIÓN DEL ESFUERZO.....	84
4.5 PARA CONVERTIR EL ESFUERZO A HOMBRES-MES	86
4.6 TIEMPO DE DESARROLLO	86
4.8 SALARIO	86
4.9 COSTO	87
4.10 BENEFICIOS TANGIBLES E INTANGIBLES.	87
4.10.1 <i>Tangibles</i>	87
4.10.2 <i>Intangibles</i>	87
4.11 ANÁLISIS COSTO-BENEFICIO.	88
4.12 CONCLUSIONES PARCIALES.	88
CONCLUSIONES GENERALES	89
RECOMENDACIONES	90
REFERENCIAS BIBLIOGRÁFICAS	91
BIBLIOGRAFÍA	93
ANEXOS	94
GLOSARIO DE TÉRMINOS.....	105

Índice de tablas.

TABLA 1. CRONOGRAMA DE ACTIVIDAD	3
TABLA 2.DETERMINACIÓN Y JUSTIFICACIÓN DEL ACTOR DEL SISTEMA.	39
TABLA 3.CASOS DE USO DEL SISTEMA	41
TABLA 4.DESCRIPCIÓN DEL CU: CARGAR PRESENTACIÓN.	41
TABLA 5.DESCRIPCIÓN DE CU: AUTENTICAR USUARIO.....	42
TABLA 6.DESCRIPCIÓN DE CU: MOSTRAR CONTENIDO	43
TABLA 7.DESCRIPCIÓN DE CU: GESTIONAR CALIFICACIÓN	44
TABLA 8.DESCRIPCIÓN DEL CU: CONTROLAR SALIDA	46
TABLA 9.DESCRIPCIÓN DEL CU: GESTIONAR AYUDA	47
TABLA 10.CRITERIO PARA CALCULAR EL FACTOR DE PESO DE LOS ACTORES SIN AJUSTAR.	80
TABLA 11.CRITERIO PARA CALCULAR EL FACTOR DE PESO DE LOS CASOS DE USO SIN AJUSTAR.	81
TABLA 12.FACTOR DE COMPLEJIDAD TÉCNICA	82
TABLA 13.FACTOR AMBIENTE	83
TABLA 14.DISTRIBUCIÓN DEL ESFUERZO ENTRE LAS DIFERENTES ACTIVIDADES DEL DESARROLLO DE SW.	86
TABLA 15.CONCLUSIONES PARCIALES	88

Índice de figuras.

FIGURA 1.MODELO DE DOMINIO	37
FIGURA 2.DIAGRAMA DE CASOS DE USO DEL SISTEMA	40
FIGURA 3.DIAGRAMA DE NAVEGACIÓN GENERAL	48
FIGURA 4.DIAGRAMA DE NAVEGACIÓN: EJERCICIOS	49
FIGURA 5.DIAGRAMA DE PRESENTACIÓN: GENERAL	52
FIGURA 6.DIAGRAMA DE PRESENTACIÓN: INICIO	53
FIGURA 7.DIAGRAMA DE PRESENTACIÓN: ÍNDICE	54
FIGURA 8.DIAGRAMA DE PRESENTACIÓN: TEXTOS	55
FIGURA 9.DIAGRAMA DE PRESENTACIÓN: GLOSARIO	56
FIGURA 10.DIAGRAMA DE PRESENTACIÓN: EJERCICIOS DE SELECCIÓN (V O F)	57
FIGURA 11.DIAGRAMA DE PRESENTACIÓN: EJERCICIOS DE ENVIAR.....	58
FIGURA 12.DIAGRAMA DE PRESENTACIÓN: EJERCICIOS DE CUADRO	59
FIGURA 13.DIAGRAMA DE PRESENTACIÓN: EJERCICIOS DE CHECKBOX.....	60
FIGURA 14.DIAGRAMA DE PRESENTACIÓN: EJERCICIOS DE COMPLETAMIENTO.....	61
FIGURA 15.DIAGRAMA DE PRESENTACIÓN: EJERCICIOS DE ENLAZAR COLUMNAS.....	62
FIGURA 16.DIAGRAMA DE PRESENTACIÓN: EJERCICIOS DE TABLAS	63
FIGURA 17.DIAGRAMA DE PRESENTACIÓN: ATLAS.....	64
FIGURA 18.DIAGRAMA DE JERARQUÍA DE CLASES	65
FIGURA 19.DIAGRAMA DE CLASES CU: AUTENTICAR USUARIO.....	66
FIGURA 20.DIAGRAMA DE CLASES CU: CARGAR PRESENTACIÓN.....	66
FIGURA 21.DIAGRAMA DE CLASES CU: MOSTRAR CONTENIDO	67
FIGURA 22.DIAGRAMA DE CLASES CU: GESTIONAR CALIFICACIÓN.....	68
FIGURA 23.DIAGRAMA DE CLASES CU: GESTIONAR AYUDA	69
FIGURA 24.DIAGRAMA DE CLASES CU: CONTROLAR SALIDA	70
FIGURA 25.DIAGRAMA DE DESPLIEGUE	72
FIGURA 26.DIAGRAMA DE COMPONENTES GENERAL	73
FIGURA 27.DIAGRAMA DE COMPONENTES DEL PAQUETE VENTANAS	74
FIGURA 28.DIAGRAMA DE COMPONENTES DEL PAQUETE XML	75
FIGURA 29.DIAGRAMA DE COMPONENTES DEL PAQUETE CLASES	75
FIGURA 30.MULTIMEDIA	94

FIGURA 31.ETAPAS DE LA METODOLOGÍA OOHDM.	94
FIGURA 32.METODOLOGÍA EXTREME PROGRAMING.....	95
FIGURA 33.RUP EN DOS DIMENSIONES.....	95
FIGURA 34.DIAGRAMA DE SECUENCIA CU: AUTENTICAR USUARIO	96
FIGURA 35.DIAGRAMA DE SECUENCIA CU: CARGAR PRESENTACIÓN.....	97
FIGURA 36.DIAGRAMA DE SECUENCIA CU: MOSTRAR CONTENIDO	98
FIGURA 37.DIAGRAMA DE SECUENCIA CU: GESTIONAR CALIFICACIÓN, SECCIÓN A: EJERCICIOS DE SELECCIÓN (V O F).....	99
FIGURE 38.DIAGRAMA DE SECUENCIA CU: GESTIONAR CALIFICACIÓN, SECCIÓN B: EJERCICIOS DE TABLAS	100
FIGURA 39.DIAGRAMA DE SECUENCIA CU: GESTIONAR CALIFICACIÓN, SECCIÓN C: EJERCICIOS DE CHECKBOX.	101
FIGURA 40.DIAGRAMA DE SECUENCIA CU: GESTIONAR CALIFICACIÓN, SECCIÓN D: EJERCICIOS DE COMPLETAMIENTO	102
FIGURA 41.DIAGRAMA DE SECUENCIA CU: GESTIONAR CALIFICACIÓN, SECCIÓN E: EJERCICIOS DE ENLAZAR COLUMNAS	103
FIGURA 42.DIAGRAMA DE SECUENCIA CU: GESTIONAR AYUDA	104
FIGURA 43.DIAGRAMA DE SECUENCIA CU: CONTROLAR SALIDA.....	104

Introducción

El propósito esencial de la enseñanza es la transmisión de información mediante la comunicación directa o soportada en medios auxiliares y se ha de considerar estrecha e inseparablemente vinculada a la educación.

El proceso de enseñanza-aprendizaje evoluciona constantemente. Las Tecnologías de Información y Comunicación (TICs) constituyen una herramienta de ineludible valía para la transformación socialmente positiva de dicho proceso. Desde finales de 1999 en nuestro país se ejecutan varios programas que utilizando las TICs revolucionan la educación para multiplicar los conocimientos de las nuevas generaciones. Uno de estos programas lo constituyen los sistemas con tecnología multimedia, tendencia que gana terreno día a día debido a su fácil manejo, portabilidad e interactividad.

Con un uso adecuado de la tecnología multimedia se logra que los alumnos capten mejor las ideas que se quieren transmitir y les es más fácil entender y aprender cualquier tema que se les haga llegar por estos medios. Además el proceso de aprendizaje se hace más dinámico y menos aburrido.

Cuba, ha ido incorporando esta tecnología en distintos sectores de la sociedad, como la educación y la salud. Dentro de esta última se encuentra la especialidad de Podología donde el uso de la tecnología multimedia representa uno de los instrumentos claves para la formación integral de sus profesionales.

Esta especialidad es la rama de la ciencia médica, que tiene por objeto el tratamiento de las afecciones y deformidades de los pies.

Debido a las ventajas de la tecnología multimedia y la necesidad imperante de lograr profesionales mejor preparados en la carrera de Podología se ha decidido realizar el Curso de Podología para la salud haciendo uso de esta nueva técnica.

En la actualidad existen dificultades con el aprendizaje de este tema, pues:

- Existe poca documentación.
- Los libros contienen grandes volúmenes de información, resultando aburridos y poco emotivos a la hora del estudio.

- El conocimiento que brindan los profesores está restringido al tiempo de un turno de clase, y si algún estudiante falta, se queda sin recibir los conocimientos.
- La información se encuentra dispersa, por lo que a la hora de buscar un tema específico se hace muy difícil y engorroso.
- No hay proyectos educativos que contemplen el uso de la tecnología multimedia.
- Tanto los profesores como los estudiantes tienen restricciones de acceso a una bibliografía adecuada y actualizada.

Problema científico: ¿Cómo centralizar toda la información referente a la Podología?

El problema científico se enmarca en el **objeto de estudio:** Proceso de desarrollo de un software educativo con tecnología multimedia.

El **campo de acción** es el proceso de desarrollo de un software educativo con tecnología multimedia sobre la Podología para la Salud.

Para resolver el problema se propone el siguiente **objetivo general:** Desarrollar un software con tecnología multimedia con fines educativos que centralice toda la información sobre la Podología.

Del objetivo general se han derivado algunos **objetivos específicos** que contribuirán al desarrollo exitoso del trabajo:

- Investigar sobre el estado del arte de las aplicaciones con tecnología multimedia.
- Desarrollar la aplicación educativa con tecnología multimedia.
- Crear un documento que recoja todo el proceso investigativo del desarrollo de la multimedia.

Idea a defender: Si se desarrolla el software con tecnología multimedia, entonces se centralizará toda la información sobre la Podología.

Para cumplir con los objetivos de esta investigación se desarrollarán las siguientes tareas:

- Investigar y recopilar toda la información necesaria acerca de la Podología y la tecnología multimedia.
- Seleccionar las metodologías, herramientas y lenguajes de modelado adecuados para el análisis y diseño de la aplicación.

- Seleccionar las herramientas de autor y lenguajes de programación adecuados para la implementación de la aplicación.
- Realizar el análisis, diseño e implementación de la aplicación educativa con tecnología multimedia.
- Redactar un documento que recoja toda la investigación sobre la tecnología multimedia.

Tabla 1. Cronograma de actividad

Tarea	Fecha inicio	Fecha cumplimiento	Responsable
Investigar y recopilar toda la información necesaria acerca de la Podología y la tecnología multimedia.	7 de diciembre de 2007	25 de enero de 2008	Lisette Nuñez. Dayamis Ruiz.
Seleccionar las metodologías, herramientas y lenguajes de modelado adecuados para el análisis y diseño de la aplicación.	4 de febrero de 2008	15 de febrero de 2008	Lisette Nuñez. Dayamis Ruiz.
Seleccionar las herramientas de autor y lenguajes de programación adecuados para la implementación de la aplicación.	20 de febrero de 2008	29 de febrero de 2008	Dayamis Ruiz Lisette Nuñez

Realizar el análisis, diseño e implementación de la aplicación educativa con tecnología multimedia.	1 de marzo de 2008	15 de mayo de 2008	Lissette Nuñez. Dayamis Ruiz.
---	--------------------	--------------------	----------------------------------

Capítulo 1: Fundamentación Teórica.

Introducción

En este capítulo se analizarán otras soluciones existentes que ayuden al estudio de la Podología en nuestro país y se abordarán conceptos claves como son: Podología, Hipermedia, Hipertexto, Tecnología Multimedia, Multimedia Educativa, entre otros, que serán necesarios para el conocimiento de los diferentes aspectos en que se fundamentará este Trabajo de Diploma. Se realizará un estudio de las diferentes tecnologías, metodologías y lenguajes que pueden ser utilizados para desarrollar la aplicación con tecnología multimedia. Por último se hará un análisis del modelo pedagógico empleado.

1.1 Análisis de otras soluciones existentes

En Internet existen diferentes portales dedicados al tema de Podología, como por ejemplo el sitio www.podologiaweb.com donde podólogos, estudiantes, pacientes y personas interesadas en conocer sobre las distintas condiciones que afectan al pie pueden consultar a podólogos, compartir técnicas, intercambiar ideas y dejar sus comentarios.

En varios portales dedicados a este tema se pueden encontrar artículos con carácter informativo de gran interés tanto para profesionales como para estudiantes o pacientes. Estos portales son:

- www.podologia.fmed.uba.ar
- www.podologiaclinica.cl

En el sitio Web www.podologia.cl se ofrecen diversos documentos e incluso se ofrecen test, pruebas de ensayo y guías de estudio para ayudar al estudio de la Podología, pero no se ofrece retroalimentación para los estudiantes y los profesores no pueden conocer las evaluaciones del estudiantado.

Para hacer uso de estos materiales o páginas Web se necesita de una conexión a Internet rápida que permita visualizarlos o descargarlos, acción que en ocasiones se tiene que pagar y si se identifica el número IP de la máquina como un IP de Cuba no lo permite descargar por las condiciones impuestas por el bloqueo de los EE.UU. a Cuba. Muchas veces el contenido de estos sitios está en inglés.

En la actualidad Cuba no cuenta con un producto en soporte multimedia que centralice toda la información existente acerca de la Podología y que permita la evaluación y retroalimentación de sus estudiantes. Es por ello que se decide desarrollar una aplicación con tecnología multimedia que sirva como material de apoyo para esta disciplina, producto que se corresponderá con el contenido y los

objetivos de aprendizaje que se persiguen. Además esta aplicación con tecnología multimedia presentará una interfaz amigable y de fácil acceso a su contenido, siendo este en idioma español.

1.2 Conceptos Claves

1.2.1 ¿Qué es la Podología?

La Podología (del griego podo, pie y logos, estudio) es una ciencia que tiene por objeto el estudio de las enfermedades y alteraciones que afectan al pie, cuando este estudio no rebasa los límites de la cirugía menor ambulatoria, es decir aquella que no precisa ingreso hospitalario. Dentro de la Podología existe un amplio campo de acción:

- Cirugía del pie. Cirugía de partes blandas, corrección de dedos en garra.
- Biomecánica de la marcha. Estudio, mediante instrumental informatizado, de los patrones de marcha normal y patológica, así como la distribución dinámica y estática de las presiones que recibe el pie.
- Ortopodología. El podólogo, después de una exploración biomecánica completa, puede confeccionar unos soportes plantares o plantillas totalmente personalizados.
- Quiropodología. Tratamiento quirúrgico del pie y no quirúrgico de las hiperqueratosis, helomas, verrugas y alteraciones ungueales.
- Podología preventiva.
- Podología pediátrica. Especializada en el pie del niño.
- Podología deportiva. Especializada en el gesto del deportista.
- Podología geriátrica. Especializada en el pie del anciano.

1.2.2 ¿Qué es la tecnología multimedia?

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, todo esto a través de un único programa (Ver anexo 31). Estos programas pueden tener diversos soportes, desde el propio ordenador personal, al CD-ROM, DVD, etc.

Algunos títulos que se manejan en torno a la multimedia son reveladores de la forma en que se la concibe:

- Multimedia: tecnología digital que integra diversos datos a través de la computadora.

- Multimedia: capacidad de interactividad.
- Multimedia: una alternativa en comunicación.
- Multimedia: como medio de difusión.

Con todo ello se afirma que la multimedia es un concepto que revolucionará a la computación tradicional e impactará a la informática con la integración de audio, imagen y datos.

Este concepto es tan antiguo como la comunicación humana ya que al expresarse en una charla normal se habla (sonido), se escribe (texto), se observa a nuestro interlocutor (video) y se acciona con gestos y movimientos de las manos (animación). Con el auge de las aplicaciones multimedia para computador este vocablo entró a formar parte del lenguaje habitual.

En resumen, multimedia podría ser denominada como una integración libre de tecnología que extiende y expande la forma en que interactuamos con una computadora, concepto que enriquece y amplía la interacción hombre-máquina, hoy en día lo vemos manifestado en diversas aplicaciones que incluyen enciclopedias históricas, aventuras científicas animadas y libros de cuentos y novelas interactivas. (DIAZ 1994)

Características de la multimedia

Un buen sistema multimedia debe tener en cuenta los siguientes aspectos (MA. VICTORIA AGUIAR PERERA):

- Interacción: debe dejar un margen o un espacio adecuado de interacción entre el alumno y el programa.
- Facilidad: que el usuario no tenga dificultades a la hora de manejarlo.
- Calidad técnica: los sonidos, los textos y los gráficos deben tener un mínimo de calidad de manera que facilite y no dificulte la recepción de la información.
- Capacidades y habilidades que debe fomentar: Creatividad, originalidad, flexibilidad de pensamiento, trabajo cooperativo, análisis/síntesis, planificación de objetivos, organización de las ideas, evaluación del proceso y, en menor medida del resultado, autoevaluación, exploración y búsqueda de información, reflexión.
- Motivación: debe despertar el interés e incrementar el grado de participación del alumno.
- Adecuación: debe adecuarse a las características del alumnado al que va dirigido.

- Atención a la diversidad: debe atender a la diversidad del alumnado que muchas veces presenta necesidades educativas especiales diferentes al grupo de clase.
- Enseñanza individualizada: debe facilitar el autoaprendizaje.
- Ayuda: debe resolver las dudas y facilitar la búsqueda de información tanto dentro del mismo programa como con el uso de otros materiales externos.
- Proceso: todo programa debe trabajar el proceso de elaboración de la actividad y no pedir exclusivamente la obtención del resultado final.
- Claridad: la información presentada debe ser clara.
- Rentabilidad temporal: el uso del programa debe ser rentable, es decir, debe ser adecuado en el tiempo de respuesta al tratamiento de la información, debe permitir la adquisición de los conocimientos en un tiempo razonable, etc.
- Eficaz: debe perseguir la consecución de los objetivos marcados en un principio.
- Estrategia didáctica: debe combinar diferentes estrategias como son la exploración guiada y el libre descubrimiento y, en menor medida, utilizar la enseñanza dirigida.
- Refuerzos: el alumno debe sentirse reforzado en todo momento tanto en los errores como en los aciertos.
- Accesible: debe permitir económicamente a los diferentes usuarios su adquisición.

Ventajas de la multimedia (SERNA 1999)

- Mejora el aprendizaje, ya que el alumno explora libremente, pregunta cuando lo necesita, repite temas hasta que los haya dominado. Se puede hablar de un aprendizaje personalizado.
- Incrementa la retención al presentar los contenidos a través de textos, imágenes, sonidos, etc., y todo ello unido a las simulaciones y a la posibilidad de interactuar.
- Aumenta la motivación y el gusto por aprender debido a la gran riqueza de animaciones y sonidos, que resultan muy atractivos para el alumnado.

Desventajas de la multimedia (AULAFACIL.COM 2000)

- Para que funcionen, dependen de la energía eléctrica permanente. Si esta falla, no hay manera de utilizarlos.
- Requiere un amplio conocimiento de las utilidades y formas de manipular cada equipo.
- Actualmente los costos de estos equipos son altos, pero la tendencia es que cada día bajen más los precios y aumente la calidad.

- Como todo equipo que funciona con energía eléctrica, requiere de cuidados especiales, ya que algunos de ellos son frágiles.
- Algunos equipos tienen la tendencia a crear adicción en su uso, por lo que es necesario dar charlas especiales a los alumnos sobre su adecuado uso.

1.2.3 ¿Qué es multimedia educativa?

Son aquellas aplicaciones multimedia que se utilizan con fines educativos. Todos los materiales didácticos multimedia orientan y regulan el aprendizaje de los estudiantes ya que, explícita o implícitamente, promueven determinadas actuaciones de los mismos encaminadas a este fin. Además, mediante sus códigos simbólicos, estructuración de la información e interactividad condicionan los procesos de aprendizaje. Aunque no siempre sus planteamientos pedagógicos sean innovadores, los programas educativos pueden desempeñar esta función ya que utilizan una tecnología actual y, en general, suelen permitir muy diversas formas de uso. Esta versatilidad abre amplias posibilidades de experimentación didáctica e innovación educativa en el aula.

Las principales funciones que pueden realizar los recursos educativos multimedia son las siguientes: informativa, instructiva o entrenadora, motivadora, evaluadora, entorno para la exploración y la experimentación, expresivo-comunicativa, metalingüística, lúdica, proveedora de recursos para procesar datos, innovadora, apoyo a la orientación escolar y profesional, apoyo a la organización y gestión de centros (DEWEY.UAB.ES 2004).

Aplicaciones de la multimedia educativa.

- Como apoyo al profesor.
- Para explorar información.
- Para la realización de proyectos de trabajo.

1.2.4 Hipertexto

Es un paradigma en la interfaz del usuario, cuyo fin es el de presentar documentos que puedan bifurcarse o ejecutarse cuando sea solicitado. La forma más habitual de hipertexto en documentos es la de hipervínculos o referencias cruzadas automáticas que van a otros documentos. Si el usuario selecciona un hipervínculo, hace que el programa de computador muestre el documento enlazado en un corto período. Otra forma de hipertexto es el stretchtext, que consiste en dos indicadores o

aceleradores y una pantalla. El primer indicador permite que lo escrito pueda moverse de arriba hacia abajo en la pantalla. El segundo indicador induce al texto a que cambie de tamaño por grados.

El hipertexto es una de las formas de la hipermedia. Cuenta con los siguientes elementos: nodos o secciones, enlaces o hipervínculos y anclajes. Los nodos son las partes que contienen información accesible para el usuario. Los enlaces son las uniones o vínculos que se establecen entre nodos y facilitan la lectura secuencial o no secuencial por los nodos del documento. Los anclajes son los puntos de activación de los enlaces.

1.2.5 Hipermedia.

Hipermedia es el término con que se designa al conjunto de métodos o procedimientos para escribir, diseñar o componer contenidos que tengan texto, video, audio, mapas u otros medios, y que además tenga la posibilidad de interactuar con los usuarios.

El enfoque hipermedia de estos contenidos, los califica especialmente como medios de comunicación e interacción humanas, en este sentido un espacio hipermedia, es un ámbito sin dimensiones físicas, que alberga, potencia y estructura actividades de las personas.

Entre los tipos de hipermedia se encuentran:

- Hipertexto.
- Hiperfilmes.
- Hipergrama.

El término Hipermedia, combinación de los conceptos HIPERtexto y multiMEDIA, hace referencia a una tecnología de construcción de (hiper)documentos que permite a los lectores encontrar fácilmente la información que realmente necesitan, de la manera que ellos decidan, a través de enlaces establecidos por el autor entre los diferentes elementos de información multimedia (texto, sonido, imagen, video, etc.) que conforman el documento (MARTÍNEZ *et al.* 1998).

1.2.6 Texto

Es una composición de signos codificados en un sistema de escritura (como un alfabeto) que forma una unidad de sentido. Su tamaño puede ser variable.

También es texto una composición de caracteres imprimibles generados por un algoritmo de cifrado que, aunque no tiene sentido para cualquier persona, si puede ser descifrado por su destinatario texto claro original.

En otras palabras un texto es un entramado de signos con una intención comunicativa que adquiere sentido en determinado contexto.

1.2.7 Sonido

El sonido es, desde el punto de vista físico, el movimiento ondulatorio en un medio elástico (normalmente el aire), debido a cambios rápidos de presión, generados por el movimiento vibratorio de un cuerpo sonoro. En general, se llama sonido a la sensación en el órgano del oído, producida por este movimiento.

1.2.8 Imagen

Una imagen (del latín imago) es una representación visual de un objeto mediante técnicas diferentes de diseño, pintura, fotografía y video.

1.2.9 Animación

La animación es aquella técnica para dar sensación de movimiento a imágenes o dibujos. Para realizar una animación existen numerosas técnicas que van más allá de los familiares dibujos animados. Los cuadros se pueden generar dibujando, pintando o fotografiando los minúsculos cambios hechos repetidamente a un modelo de la realidad o a un modelo tridimensional virtual; también es posible animar objetos de la realidad y actores.

Es decir, la animación es una representación sucesiva de una secuencia de imágenes que produce la sensación de estar viendo imágenes en movimiento. Para ello, a cada imagen de una animación se le modifica un pequeño detalle para mantener el movimiento tan fluido como sea posible.

1.2.10 Video

El video es una tecnología de captura electrónica, grabación, tratamiento, almacenamiento, transmisión y reconstrucción de una secuencia de imágenes que representan escenas en movimiento. La tecnología de video fue desarrollada por primera vez para los sistemas de televisión, pero se ha reforzado en muchos formatos para permitir la grabación de video de los consumidores.

1.3 Metodologías

Se define metodología como el conjunto de métodos que se siguen para alcanzar una gama de objetivos en una ciencia.

El modelo **HDM** (Hypertext Design Model) es el primer modelo multimedia que se publica, introduce la metodología de multimedia y el concepto de entidad, tipos de entidades y estructuras de acceso. Fue creado con el objetivo de realizar el diseño de una aplicación de hipertexto.

1.3.1 RMM: RelatioShip Management Methodology.

Es una metodología basada en los conceptos del Modelo de diseño de Hipertexto (HDM) es decir, en entidades y tipos de entidades. Su objetivo es mejorar la navegación a través de un análisis de las entidades del sistema. Incorpora el concepto de slice como agrupación de datos de varias pantallas en una entidad. Es la primera metodología que se publica completa para la creación de un software multimedia. Su problema principal es que no permite realizar consultas a partir de dos entidades por su ligadura al modelo entidad relación, obligando a la descomposición de relaciones uno a muchos, no obstante muestra su fortaleza en los procesos de análisis y diseño para multimedia. (LAMARCA LAPUENTE 2005)

1.3.2 OOHDM: Object Oriented Hypermedia Design Model.

OOHDM es una metodología orientada a objetos que propone un proceso de desarrollo de cinco fases donde se combinan notaciones gráficas UML con otras propias de la metodología. En una primera instancia debido al poco auge que tenía Internet, OOHDM era sólo para aplicaciones que incluían hipertexto y algo de multimedia (CD-ROM promocionales, enciclopedias, museos virtuales, etc.). Pero el gran desarrollo de Internet obligó su adaptación para el desarrollo de aplicaciones hipermedia en Internet, tales como comercio electrónico, motores de búsqueda y sitios educativos (GIORGIS *et al.*).

Las cinco etapas de OOHDM son: Obtención de requerimientos, Modelo Conceptual, Diseño navegacional, Diseño de Interfaz Abstracta e Implementación (Ver anexo 32). Propone en las fases de diseño conceptual o análisis de dominio, que utiliza el método del análisis del método OO, esquemas conceptuales de las clases y su modelado. Define la navegación a través de diferentes vistas del esquema conceptual, la fase de diseño de interfaz abstracta para el modelo de la interfaz de sistema con diagramas de cada clase, diagrama de configuración para eventos externos y diagramas de

estado para el comportamiento dinámico. En la fase de implementación, construye una aplicación completamente orientada a objetos.

1.3.3 XP: Programación extrema o Extreme Programming

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo y pequeños equipos de trabajo y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto (Ver anexo 33).

Características de XP, la metodología se basa en (SANCHEZ 2004):

- Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.
- Refabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

¿Qué es lo que propone XP?

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo.

Derechos del Cliente:

- Decidir qué se implementa.
- Saber el estado real y el progreso del proyecto.
- Añadir, cambiar o quitar requerimientos en cualquier momento.
- Obtener lo máximo de cada semana de trabajo.
- Obtener un sistema funcionando cada 3 ó 4 meses.

Derechos del Desarrollador:

- Decidir cómo se implementan los procesos.
- Crear el sistema con la mejor calidad posible.
- Pedir al cliente en cualquier momento aclaraciones de los requerimientos.
- Estimar el esfuerzo para implementar el sistema.
- Cambiar los requerimientos en base a nuevos descubrimientos.

Lo fundamental en este tipo de metodología es:

- La comunicación entre los usuarios y los desarrolladores.
- La simplicidad, al desarrollar y codificar los módulos del sistema.
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

1.3.4 RUP: Proceso Unificado de Software

Es un proceso de desarrollo creado por la Corporación “Rational Software”, ahora una división de IBM, como una plataforma adaptable de procesos para describir cómo crear productos efectivos a través de técnicas de alta fidelidad. Aunque RUP abarca un determinado número de actividades diferentes, está diseñado para poder ajustarse en la selección de procesos específicos destinados a un proyecto u organización de desarrollo en particular y es reconocida en medio de grandes equipos de trabajo que llevan a cabo el manejo de complicadas aplicaciones de software.

El ciclo de vida de **RUP** se caracteriza por (VALDÉS 2006):

- **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).
- **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.
- **Iterativo e Incremental:** 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.

Los creadores de este proceso, se basaron en los diagnósticos de las fallas de diferentes proyectos de software, identificaron las causas matrices, los procesos de ingeniería de software y las soluciones propuestas, construyendo un sistema basado en el conjunto de todas las formas óptimas de trabajo y modelando el proceso de desarrollo con las mismas técnicas de modelado de software, a través del paradigma Orientado a Objetos y el Lenguaje Unificado (UML).

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo. En el anexo 34: RUP en Dos Dimensiones se representa el proceso en el que se grafican los flujos de trabajo y las fases y muestra la dinámica expresada en iteraciones y puntos de control.

RUP divide en 4 fases el desarrollo del software:

- Inicio: El Objetivo en esta etapa es determinar la visión del proyecto.
- Elaboración: En esta etapa el objetivo es determinar la arquitectura óptima.
- Construcción: En esta etapa el objetivo es llegar a obtener la capacidad operacional inicial.
- Transición: Su objetivo es llegar a obtener la versión liberada del proyecto.

Como RUP es un proceso, en su modelación define como sus principales elementos: Trabajadores (“quién”), Actividades (“cómo”), Artefactos (“qué”), Flujo de actividades (“Cuándo”).

Entre sus principales ventajas podemos citar que RUP contiene diferentes elementos de planificación (plan de desarrollo, plan de iteración, plan de calidad, etc.) con los que se controla el desarrollo del software. A través de un predefinido esquema de escalabilidad y gestión de riesgos, se pueden identificar previamente problemas y fallos en etapas tempranas del desarrollo y prevenirlos ó corregirlos. RUP define en cada momento del ciclo de vida del proyecto, qué artefactos, con qué nivel de detalle, y por qué rol, se deben crear. Se definirán qué artefactos son necesarios para poder realizar una actividad y qué artefactos se deberán crear durante dicha actividad.(PALERMO)

RUP se aplica a una buena cantidad de productos y procesos de software en el mundo. No es específico para diseño hipermedia, sin embargo a través de la extensión de UML para multimedia, conocida por OMMMA – L, se presenta como algo eficientemente realizable.

1.4 Lenguajes de Modelado

1.4.1 UML: Unified Modeling Language

UML (Unified Modeling Language) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software.

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad, está respaldado por el OMG (Object Management Group). UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables (ORALLO 2001).

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado de Racional), pero no especifica en sí mismo qué metodología o proceso usar.

1.4.2 OMMMA – L: Lenguaje de Modelado Orientado a Objetos de Aplicaciones Multimedia.

Desarrollada por Stefan Sauer y Gregor Engels, profesores e investigadores del Departamento de Matemática y Ciencias de la Computación de la Universidad de Paderborn, Alemania. El Lenguaje de Modelado Orientado a Objetos de Aplicaciones Multimedia integra el comportamiento interactivo con el de procedimientos temporales para lograr la descripción de aplicaciones que reaccionan ante eventos externos y producen ejecuciones dinámicas predecibles en tiempo de ejecución, dando una muestra sólida de la integración temporal y la sincronización de diferentes objetos de media.

Representable a través de los modelos y artefactos, conservando la semántica de muchos de estos y creando nuevas interpretaciones afines a una especificación multimedia, **OMMMA-L** modela diversos aspectos de sistema basados en el paradigma Orientado a Objeto, utiliza el Lenguaje de Modelado Unificado y se integra dentro del Proceso Unificado de Ingeniería del Software. (VALDÉS 2006)

OMMMA-L 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, siendo este un patrón de diseño de software que separa 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.

OMMMA – L, modela la estructura a través de diagramas de objetos y clases, mientras que el comportamiento puede ser descrito en los diagramas de interacción, estado y actividad. Por último, la distribución espacial de media contemplada en el modelo vista, es descrita a través de un nuevo artefacto propuesto para el lenguaje: el diagrama de presentación.

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

- **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 diagramas 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 (barras de menú, botones, campos de entrada y salida, scrolls, 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, sintácticamente igual a este último, 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.

1.5 Herramientas CASE.

Las herramientas CASE (Computer Aided Software Engineering) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de los mismos en términos de tiempo y de dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

1.5.1 Rational Rose

Rational Rose es una herramienta CASE que permite el diseño detallado del software y la generación de código fuente (de programas y bases de datos) e ingeniería inversa (obtención del diseño a partir del código fuente), basado en modelos con soporte UML. Es una forma de ayuda para la comprensión del sistema y de sus distintos componentes. Su característica más significativa consiste en la creación de componentes, que contengan una serie de archivos dentro de los cuales se encuentran las distintas clases pertenecientes a dicho componente.

Rational Rose 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. Utiliza un proceso de desarrollo iterativo controlado donde el desarrollo se lleva a cabo en una secuencia de iteraciones.

Rational Rose proporciona múltiples ventajas: permite separar el desarrollo de la vida del software en etapas predefinidas como análisis, diseño e implementación y con esto facilita una mejor comprensión del sistema en su etapa de desarrollo y construcción (BLANCO and TOBALINA).

Esta herramienta contiene 4 vistas:

- Vista de Caso de Uso.
- Vista Lógica.
- Vista de Componente.
- Vista de Despliegue.

1.5.2 Visual Paradigm

Visual Paradigm es una herramienta CASE que utiliza UML como lenguaje de modelado y que funciona sobre múltiples plataformas. Dotada de una buena cantidad de productos o módulos para facilitar el trabajo durante la confección de un software, lo cual garantiza la calidad del producto final.

Entre otras facilidades ofrece (INTERNATIONAL):

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad en múltiples plataformas.
- Profesionalidad: Da la posibilidad de crear un gran conjunto de artefactos de las distintas fases del desarrollo del software.
- Contiene facilidades para redactar Especificaciones de Casos de Uso del Sistema.
- Sincronización entre Diagramas de Entidad Relación y Diagramas de Clases.

Además, brinda la posibilidad de una representación gráfica de los diagramas lo que permite ver el sistema desde múltiples perspectivas, como el de componentes, despliegue, secuencia, casos de uso, clase, actividad, estado, entre otros. Por otra parte, identifica requisitos y comunica información, se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos, además, permite ver las relaciones entre los componentes del diseño.

1.6 Herramientas de autor

Existen muchas herramientas para desarrollar aplicaciones con tecnología multimedia tales como Flash, Director, Toolbook, MTASC y Flash Develop, las cuales son utilizadas según la decisión del cliente o el desarrollador.

1.6.1 Macromedia Director

Director es una poderosa herramienta que sin apenas la necesidad de programar permite desarrollar nuestras propias aplicaciones (presentaciones sencillas, juegos más complicados, enciclopedias interactivas, etc.). Además del potente lenguaje incorporado (Lingo), una de sus principales ventajas esta en el uso de los llamados XTRAS. Se trata de "pequeños programas" desarrollados en lenguaje C++ por otros usuarios o terceras empresas, y que proporcionan al usuario infinidad de utilidades.

Macromedia Director incorpora un rango de nuevas capacidades para satisfacer las necesidades evolutivas del desarrollador actual, al mismo tiempo que conserva todo el poder de sus características centrales. Incluye una integración transparente y completa con la familia de productos Macromedia Flash MX, adopción de interfaces eficientes para el usuario de Macromedia MX, soporte para Mac OS X, nuevas y mejores eficiencias en el flujo de trabajo y la habilidad para crear contenido accesible para que las presentaciones enriquecidas de Director puedan ser disfrutadas por personas con discapacidades. El lenguaje de programación orientado a objetos de Director (Lingo) agiliza los tiempos de desarrollo y ayuda a integrar a sus producciones una interactividad única y de alto nivel.

Podemos decir también que Director es un gran "integrador" de recursos. Su versatilidad le permite incluir múltiples formatos de imagen, video y audio, así como también "movies" de Flash. Quizá sea por esto que sus herramientas no le permiten una gran ductilidad a la hora de crear sus propias interfaces gráficas. De hecho, es preferible crear estas interfaces en otro programa (Fireworks, Photoshop, Freehand, etc), y luego importarlas desde Director. La versatilidad de Director no se logra tanto por sus herramientas, sino por Lingo, el lenguaje que nos permite sacarle el máximo de posibilidades.(TORRES)

1.6.2 ToolBook

Ofrece interfaces gráfica Windows y un ambiente de programación orientado a objeto para construir proyectos, o libros, a fin de presentar gráficamente información, como dibujos, imágenes digitalizadas a color, textos, sonido y animaciones.

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

ToolBook tiene dos niveles de trabajo: el lector y el autor. A nivel autor usted utiliza órdenes para crear nuevos libros, crear y modificar objetivo en las páginas y escribir guiones. ToolBook ofrece opciones de vinculación para botones y palabras claves, de forma que se pueda crear guiones de navegación identificando la página a la que debe ir (ALMENARA *et al.*).

Herramientas libres:

1.6.3. Flash Develop

Flash Develop es una de las herramientas más utilizadas del mundo de Open Source en Flash. No es más que un IDE para programadores de Action Script (1, 2 y 3) y aplicaciones Flex (MXML), por lo que es una excelente aplicación, que basa su funcionamiento en MTASC, SWFMILL, FBUILDER y otras más. Flash Develop, en su versión actual (2.0.1), es una herramienta que le da un carácter profesional y muy potente a las aplicaciones Flash, por lo que su penetración en el mundo de los proyectos de Action Script es total por parte de la comunidad de Open Source en Flash.

Incorpora completamiento local y global de código, verificación de sintaxis, coloreado de palabras reservadas, control de ficheros del Proyecto, etc. (GUTIÉRREZ 2007).

1.6.4 MTASC

MTASC (Motion Twin Action Script Compiler): Es el primer compilador de Action Script libre que existe. Fue desarrollado por Alessandro Cugnola, creador también del lenguaje de programación.

Sobre MTASC están desarrollados muchos de los proyectos de Open Source Flash que existen, como: Flash Develop, FAMES, AMES, SEPY Action Script Editor, etc. Su funcionamiento se basa en interpretar sintaxis de ficheros de clases de Action Script (.as) y compilarlos a SWFs que más adelante Flash Player interpreta y ejecuta los scripts al igual que hace con los creados en Macromedia Flash. En su versión actual, solamente compila Action Script 1 y 2, debido a que su creador anunció discontinuar el proyecto de MTASC. Es muy útil y en algunos proyectos ha demostrado ahorrar semanas de trabajo al equipo de desarrollo sencillamente por aceleración de la demora de compilación.(GUTIÉRREZ 2007).

1.6.5 Macromedia Flash

Flash es una herramienta de edición con la que los diseñadores y desarrolladores pueden crear presentaciones, aplicaciones y otro tipo de contenido que permite la interacción del usuario. Los proyectos de Flash pueden abarcar desde simples animaciones hasta contenido de video, presentaciones complejas, aplicaciones y cualquier otra utilidad relacionada. En general, los fragmentos independientes de contenido creados con Flash se denominan aplicaciones, aunque se trate solamente de una animación básica. Se pueden crear aplicaciones de Flash con una amplia variedad de contenido multimedia que incluye imágenes, sonido, video y efectos especiales.

Una de las razones por las que Flash es una herramienta tan popular es que utiliza la tecnología de gráficos vectoriales (gráficos definidos como puntos y líneas en lugar de píxeles). Existen dos estándares fundamentales en Internet: raster (mapa de bits) y vectorial. *“Los archivos de gráficos vectoriales son muchos más compactos y eficientes comparados con los de mapa de bits. Los archivos creados por Flash son, por lo tanto, comparativamente pequeños, lo que es uno de los factores culpables del éxito de Flash”*. Además los archivos de gráficos vectoriales requieren mucha menos memoria y espacio de almacenamiento que las imágenes de mapa de bits, ya que se representan mediante fórmulas matemáticas en lugar de grandes conjuntos de datos. Las imágenes de mapa de bits son de un tamaño superior porque cada píxel requiere un fragmento de datos independiente que lo represente. (KRISTIAN BESLEY).

Flash ha consolidado sus capacidades multimedia gracias a un completo soporte para contenidos de video, y también incorpora su propio lenguaje de programación Actionscript.

Flash incluye muchas funciones que la convierten en una herramienta con numerosas prestaciones sin perder por ello la facilidad de uso. Entre dichas funciones destacan la posibilidad de arrastrar y soltar componentes de la interfaz de usuario creados previamente, comportamientos integrados que permiten añadir fácilmente código ActionScript al documento y varios efectos especiales que pueden incorporarse a los objetos multimedia.

Flash es independiente del navegador y el plugin es universal, por lo que las animaciones diseñadas con este programa se verán casi idénticamente en cualquier plataforma y navegador.

Flash es fácil de aprender, tiene un entorno amigable que nos invita a sentarnos y pasar horas y horas creando lo que nos dicte nuestra imaginación.

1.6.5.1 Macromedia Flash 8.0

Flash 8.0 es una potente herramienta creada por Macromedia que ha superado las mejores expectativas de sus creadores.

“Flash 8.0 ha conseguido hacer posible lo que más se echa en falta en Internet: Dinamismo, y con dinamismo no sólo nos referimos a las animaciones, sino que Flash permite crear aplicaciones interactivas que permiten al usuario ver la web como algo atractivo, no estático (en contraposición a la mayoría de las páginas, que están realizadas empleando el lenguaje HTML). Con Flash podremos crear de modo fácil y rápido animaciones de todo tipo”.(AULACLIC.ES 2006)

Las mejoras incorporadas en Flash Professional 8.0 ofrecen mayor rendimiento, flexibilidad y facilidad de uso, estas mejoras son:

- **Diseños más atractivos:** Permite el uso de efectos visuales que facilitan la creación de animaciones, presentaciones y formularios más atractivos y profesionales, así mismo, pone a nuestra disposición mecanismos para hacer este trabajo más cómodo y rápido, tales como la existencia de filtros y modos de mezcla añadidos en esta versión.
- **Optimización de fuentes:** Incorpora también opciones de legibilidad para fuentes pequeñas, haciendo la lectura de nuestros textos más agradables y de alta legibilidad. Además de poder modificar la optimización, permite también la selección de configuraciones preestablecidas para textos dinámicos y estáticos.
- **Bibliotecas integradas:** Ahora se puede buscar rápidamente cualquier objeto existente en nuestras películas, navegando por las bibliotecas de todos los archivos abiertos desde un único panel.
- **Mayor potencia de animación:** Permite un mayor control de las interpolaciones habilitando un modo de edición desde el que se podrá modificar independientemente la velocidad en la que se apliquen los diferentes cambios de rotación, forma, color, movimiento, etc, de nuestras interpolaciones.
- **Mayor potencia gráfica:** Evita la repetición innecesaria de la representación de objetos vectoriales simplemente señalando un objeto como mapa de bits. Aunque el objeto se convierta al formato de mapa de bits, los datos vectoriales se mantienen tal cual, con el fin de que, en todo momento, el objeto pueda convertirse de nuevo al formato vectorial.
- **Mejoras en la importación de video:** Para facilitar el resultado con formatos de video, Flash 8 incluye un códec independiente de calidad superior capaz de competir con los mejores códecs

de video actuales con un tamaño de archivo mucho más pequeño. Además de una gran posibilidad de revestimientos para los controles de éste en nuestra película.

- **Compatibilidad Metadatos:** Incluye los SWF en buscadores de internet con la nueva característica de definición de archivo con un título, una descripción y/o palabras clave para que los motores de búsqueda reflejen con más precisión el contenido representado por el archivo.
- **Asistente de ActionScript:** El Asistente de ActionScript ha vuelto. Fue eliminado en la versión anterior, pero se ha vuelto a recuperar, y de forma mejorada, en esta última. Este facilita la creación de scripts sin necesidad de tener un conocimiento profundo de ActionScript. El Asistente de script le ayuda a crear sus scripts seleccionando elementos del panel Acciones y le proporciona una interfaz de campos de texto, botones de opción y casillas de verificación que le proponen las variables correctas y otros componentes del lenguaje de creación de scripts.

Mejoras en ActionScript 2.0:

- El lenguaje ActionScript ha crecido y se ha desarrollado desde su introducción hace algunos años. Con cada nueva versión de Flash se han añadido a ActionScript palabras clave, objetos, métodos y otros elementos adicionales. Se han incluido mejoras relativas a ActionScript en los entornos de edición de Flash 8 y presenta nuevos elementos de lenguaje para mejorar la expresividad, como filtros y modos de mezcla, y el desarrollo de aplicaciones, como integración con JavaScript (ExternalInterface) y entrada y salida de archivos (FileReference y FileReferenceList).

1.7 Herramientas para el diseño de la interfaz

1.7.1 Screenweaver 3 OS

Screenweaver es una herramienta que extiende las capacidades de flash y permite salvar archivos al disco duro. Para la creación de aplicaciones de escritorio basadas en Flash, se usa Screenweaver que no es más que una aplicación que crea proyectores (.EXE) a partir de ficheros SWF. Incorpora su propio lenguaje de programación y le ofrece a Flash Player las funcionalidades de las aplicaciones comunes de escritorio que este carece por ser una tecnología enfocada a la WEB (GUTIÉRREZ 2007).

1.7.2 Sony Vega

Es un potente editor de video orientado a profesionales o a usuarios que busquen resultados profesionales. Aúna una potente edición de audio y video en una única y completa plataforma de creación.

En cuanto a la edición de video, Vegas ofrece captura de dispositivos de alta calidad, más de 300 efectos y transiciones, soporte para edición y creación de subtítulos, codificación MPEG2 (para DVD), importa archivos SWF (animaciones Flash), y mucho más.

Soporta en el tratamiento de audio un número ilimitado de pistas simultáneas. Posee soporte para canales 5.1, más de 30 efectos personalizables y la posibilidad de aumentar su capacidad y variedad gracias al soporte de plug-ins VST. (FORO3K.COM 2008)

1.8 Lenguajes de marcas

1.8.1 HTML

El HTML no es más que una aplicación del SGML (Standard Generalized Markup Language), que es un sistema para definir tipos de documentos estructurados y lenguajes de marcas para representar esos mismos documentos.

El HTML, Hyper Text Markup Language (Lenguaje de marcación de hipertexto), es el lenguaje de marcas de texto utilizado normalmente en la www (World Wide Web). HTML no es propiamente un lenguaje como C++, Visual Basic, etc., sino un sistema de etiquetas. HTML no presenta ningún compilador, por lo tanto algún error de sintaxis que se presente éste no lo detectará y se visualizará en la forma como éste lo entienda. El entorno para trabajar HTML es simplemente un procesador de texto, como el que ofrece los sistemas operativos Windows (bloc de notas), o el que ofrece MS Office (Word). El conjunto de etiquetas que se creen se deben guardar con la extensión .htm o .HTML.

Estos documentos pueden ser mostrados por los visores o "browsers" de páginas Web en Internet, como Netscape Navigator, Mosaic, Opera y Microsoft Internet Explorer.

También existe el HTML Dinámico (DHTML), que es una mejora de Microsoft de la versión 4.0 de HTML que le permite crear efectos especiales como, por ejemplo, texto que vuela desde la página palabra por palabra o efectos de transición al estilo de anuncio publicitario giratorio entre página y página. (RAVIOLI 2007)

1.8.2 XML

XML significa Extensible Markup Language o Lenguaje de Anotación Extensible. XML fue desarrollado por un grupo de trabajo bajo los auspicios del consorcio World Wide Web (W3C) a partir de 1996. Este fue constituido en 1994 con el objetivo de desarrollar protocolos comunes para la evolución de Internet. Se inició como un subconjunto de SGML (Structured Generalized Markup Language), un estándar ISO para documentos estructurados que es sumamente complejo para poder servir documentos en la Web. Es algo así como SGML simplificado, de forma que una aplicación no necesita comprender SGML completo para interpretar un documento, sino sólo el subconjunto que se defina. Los editores SGML, sin embargo, pueden comprender XML.

XML es un lenguaje de marcas que ofrece un formato para la descripción de datos estructurados, el cual conserva todas las propiedades importantes del antes mencionado SGML. Es decir, XML es un metalenguaje, dado que con él podemos definir nuestro propio lenguaje de presentación y, a diferencia del HTML, 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.

No es solo un lenguaje, es una forma de especificar lenguajes, de ahí lo de extensible. Todo lenguaje que se exprese de una forma determinada puede ser XML. Por lo tanto XML no es un lenguaje para hacer mejores páginas Web, sino un lenguaje para información auto-descrita, o al menos, auto-descritas si las etiquetas están bien puestas.

Por tanto, no debe uno pensarse que XML es para crear páginas Web, o algo parecido. XML es un lenguaje que cambia el paradigma de programación: de basada en funciones u objetos a la programación basada en el documento. Se puede usar para cambiar totalmente el paradigma de publicación; de un programa que recibe unas entradas y produce unas salidas, se pasa a un documento que genera otro documento, o bien programas que toman documentos y producen otros documentos. Por eso, también, y, en general, salvo en entornos de servicios Web, lo normal es que el XML se use en el servidor, y se sirva otro tipo de documentos, HTML, por ejemplo, que se obtienen a base de una serie de transformaciones. Precisamente, esto hace que los documentos XML se usen dentro de entornos de aplicaciones. Este entorno de aplicaciones permite publicar documentos XML, que, antes de ser enviados al cliente, sufrirán una serie de transformaciones para adaptarlo a los requisitos del mismo. (GUERVOS 2004)

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 dinámicos de noticias, sistemas de gestión de weblogs y foros son algunas de las aplicaciones donde el uso del lenguaje XML se hace prácticamente imprescindible. No obstante, su campo de acción no se limita únicamente a las aplicaciones de carácter dinámico. Una de las principales razones por la que se recomienda la integración de XML con Actionscript es evitar una recurrente edición del archivo fuente (.fla) cada vez que se necesite introducir algún cambio en el contenido.

1.9 Lenguaje de Programación

1.9.1 ActionScript

Action Script es el lenguaje de programación orientado a objetos para crear scripts en Flash. Puede ser utilizado para examinar o modificar las propiedades de los elementos de una película. Por ejemplo, podemos:

- Cambiar el color y la localización de un objeto.
- Reducir el volumen de un sonido.
- Especificar la tipografía de un bloque de texto, etc.

Con ActionScript podemos generar contenidos directamente desde la biblioteca de la película o duplicar contenido existente en el escenario. También provee de una amplia variedad de herramientas para enviar y recibir información del servidor (PROGRAMACION.NET 1999).

Mediante ActionScript se pueden crear programas que, por ejemplo, busquen en una base de datos o interactúen con un programa en otro lenguaje.

Permite añadir interactividad a los elementos multimedia del documento. Por ejemplo, se puede añadir código para que un botón muestre una nueva imagen cuando el usuario haga clic en el mismo. También se puede utilizar ActionScript para añadir lógica a las aplicaciones. Gracias a la lógica, la aplicación se comporta de distintas formas dependiendo de las acciones del usuario u otras condiciones. Flash incluye dos versiones de ActionScript, cada una adaptada a las necesidades específicas del editor.

Al igual que otros lenguajes de creación de scripts, ActionScript tiene sus propias reglas sintácticas, reserva palabras clave, proporciona operadores y permite utilizar variables para almacenar y recuperar

información. ActionScript incluye funciones y objetos incorporados y permite crear los suyos propios. Se basa en la especificación ECMAScript (ECMA-262), el estándar internacional del lenguaje de programación ECMAScript. Ofrece un subconjunto de las funciones de ECMAScript.

Los archivos AS son los archivos de ActionScript. Se pueden utilizar si se prefiere guardar parte o todo el código ActionScript fuera de los archivos FLA. Esto puede resultar útil para la organización del código, así como para proyectos en los que participan varios usuarios en distintas partes del contenido de Flash.

ActionScript incluye varias clases incorporadas, como la clase MovieClip o la clase Sound, entre otras. También puede crear clases personalizadas para definir categorías de objetos para las aplicaciones.

1.9.1.1 ActionScript 2.0:

ActionScript 2.0 permite utilizar un enfoque orientado a objetos para desarrollar aplicaciones a través de un conjunto adicional de elementos de lenguaje, lo que puede ser ventajoso para el desarrollo de la aplicación. Implementa varios nuevos conceptos y palabras claves de programación orientada a objetos, como por ejemplo clase, interfaz y paquetes. El modelo OOP (Object Oriented Programming) que proporciona ActionScript 2.0 es una "formalización sintáctica" del método de cadenas prototipo utilizado en versiones anteriores de Macromedia Flash para crear objetos y establecer la herencia. Con ActionScript 2.0, se puede crear clases personalizadas y ampliar las clases incorporadas en Flash. También permite especificar de forma explícita tipos de datos para variables, parámetros de función y tipos de devolución de funciones.

1.10 Tecnologías a utilizar para la solución propuesta

Después de un estudio realizado de las herramientas y metodologías más importantes para el desarrollo de la aplicación con tecnología multimedia se escogió para el análisis y diseño la metodología RUP pues es una metodología que propone cuatro fases de desarrollo muy bien definidas y cada una con un objetivo general muy relevante, además tiene múltiples ventajas: es iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura. Además los creadores de este proceso lo modelaron a través del paradigma Orientado a Objetos y el Lenguaje Unificado (UML).

Se seleccionó la herramienta CASE Rational Rose ya que permite completar una gran parte de las disciplinas (flujos fundamentales) del Proceso Unificado (RUP), además es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación de este lenguaje de modelado.

Como lenguaje de modelado se decidió usar OMMMA-L debido a que facilita el modelado de un gran rango de aspectos de aplicaciones multimedia de una forma integrada y comprensiva. Además se integra dentro del Proceso Unificado de ISW y utiliza el Lenguaje de Modelado Unificado.

Para realizar la implementación de la aplicación se escogió como herramienta principal la Macromedia Flash 8.0 debido a que es una de las más utilizadas para el desarrollo de aplicaciones con tecnología multimedia, reduce las animaciones a la mínima expresión en cuanto al espacio e incorpora potentes herramientas de animación y efectos de fácil uso. Se pueden exportar imágenes creadas al tradicional formato .swf o a estándares .gif a la animación por fotogramas y además se tienen conocimientos previos de dicha herramienta. Flash es el entorno de autoría más avanzado del mercado para la creación de aplicaciones con tecnología multimedia interactivas, porque las aplicaciones se pueden ejecutar en la Web, en Windows, Macintosh, Unix y Linux. Otro punto a su favor es su potencia en la creación de animaciones. Macromedia Flash 8.0 posee mejoras con respecto a otras versiones en cuanto a la facilidad de manejo, incorpora mayor potencia gráfica y en esta versión se recupera el asistente de ActionScript que había desaparecido en la versión anterior.

Para salvar archivos al disco rígido se escogió la herramienta Screenweaver 3 OS. El lenguaje de programación seleccionado fue Action Script 2.0, ya que es el lenguaje de programación encargado de crear los scripts en Flash, además permite crear clases personalizadas y ampliar las clases incorporadas en Flash. Se escogieron los lenguajes de marcas HTML para darle formato a los textos de la aplicación y XML 1.0 para cargar los textos de la aplicación. Para el diseño del software se escogió la herramienta Sony Vegas 5.0, debido a que permite modificar los sonidos que lleva la aplicación.

El uso de todas estas herramientas permitirá realizar un software eficiente, con la calidad requerida y que cumpla con los objetivos propuestos.

1.11 Análisis del modelo pedagógico.

Para el diseño de actividades se cumplió con las orientaciones metodológicas recibidas para la disciplina de Podología. Se confeccionaron actividades para trabajar con las diferentes funcionalidades del pie, partes, estructuras, etc.

Se orientaron ejercicios en el siguiente orden para cada tema por separado:

- Ejercicios para conocer e identificar el sistema tegumentario del pie.
- Ejercicios para conocer e identificar el sistema óseo del pie.
- Ejercicios para conocer e identificar el sistema articular del pie.
- Ejercicios para conocer e identificar el sistema muscular del pie.

- Ejercicios para conocer e identificar el sistema circulatorio del pie.
- Ejercicios para conocer e identificar el sistema nervioso del pie.

1.12 Conclusiones parciales

Con la realización de este capítulo que aborda la fundamentación teórica del tema se ha comprendido la importancia de realizar nuestro producto con eficiencia y rapidez al analizar otras soluciones existentes en el mundo y en nuestro país, ya que estas no resuelven el problema científico en cuestión. Se investigaron profundamente los conceptos relacionados con la tecnología multimedia y la podología, lo que conllevó a un mejor entendimiento del Trabajo de Diploma que se está realizando. Se realizó un exhaustivo análisis de las tecnologías actuales en el amplio mundo de los productos multimedia, dígame metodologías, lenguajes de modelado, herramientas y lenguajes de programación para así escoger cuál o cuáles de cada una de estas tecnologías es la adecuada para desarrollar nuestro producto y cumplir así con la calidad y eficiencia que se espera del mismo.

Capítulo 2: Descripción de la Solución Propuesta

Introducción

En este capítulo se definirán los requerimientos funcionales y no funcionales que debe tener el producto para que sea agradable, sencillo y atractivo a los usuarios. Los requisitos funcionales estarán implícitos en los diferentes casos de uso del sistema, e interactuarán con él o los actores, conformando el Diagrama de Casos de Uso del Sistema. Además se modelarán los objetos o eventos más importantes que ocurren en el contexto del sistema mediante un modelo de dominio, y se describirán los conceptos asociados. Por último se representará la navegación que se lleva a cabo durante la interacción del usuario con la aplicación.

2.1 Especificación del contenido

De modo general el software estará compuesto por seis capítulos donde se le dará información al usuario del contenido que este tiene, y además cada uno de ellos tendrá una serie de ejercicios evaluativos asociados, estos capítulos son:

- Capítulo 1: Sistema tegumentario del pie.
- Capítulo 2: Sistema óseo del pie.
- Capítulo 3: Sistema articular del pie.
- Capítulo 4: Sistema muscular del pie.
- Capítulo 5: Sistema circulatorio del pie.
- Capítulo 6: Sistema nervioso del pie.

En el capítulo 1 se define qué es el sistema tegumentario y se describen sus funciones, además contiene ejercicios evaluativos de verdadero y falso.

En el capítulo 2 se explica el papel fundamental que tiene el sistema óseo del pie en la función de sostener el peso del cuerpo y las partes por las que está compuesto, además de tener ejercicios de selección, completamiento y de enlazar columnas.

El capítulo 3 trata acerca de las categorías en que se pueden clasificar las articulaciones y las características de cada una, y además se proponen ejercicios de completamiento a través de cuadros.

El capítulo 4 explica donde están ubicados cada uno de los músculos del pie y sus funciones, también tiene ejercicios de completamiento.

En el capítulo 5 se describen los componentes del sistema circulatorio del pie y como transporta las sustancias, consta de una serie de ejercicios relacionados con el tema que son de tipo completamiento y enlace de columnas.

En el capítulo 6 se habla de las características, ubicación y funciones que tiene cada una de las partes del sistema nervioso del pie, y se presentan ejercicios de completamiento y de argumentación.

2.2 Descripción del sistema propuesto

La solución propuesta constituye la elaboración de un producto con tecnología multimedia compuesto por seis capítulos. Cada uno de estos temas o capítulos contienen una serie de ejercicios asociados para que los estudiantes puedan evaluarse, además de un atlas de morfología del pie. Las evaluaciones podrán ser consultadas por el profesor a través de un fichero evaluativo. Esta aplicación contribuirá a la centralización de la información existente sobre la Podología en nuestro país.

2.3 Requerimientos

Se define el término requerimiento como (MELLADO 2008):

- Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.
- Una representación documentada de una condición o capacidad como en 1 ó 2.

La IEEE Standard Glossary of Software Engineering Terminology define un requerimiento como Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.

Los requerimientos o requisitos se pueden clasificar en: **funcionales y no funcionales**.

2.3.1. Requerimientos Funcionales:

Son capacidades o condiciones que el sistema debe cumplir y se mantienen invariables sin importar con qué propiedades o cualidades se relacionen. Para la captura de los requisitos funcionales se debe preguntar: ¿Qué debe hacer el sistema?

RF 1: Mostrar presentación general del producto.

RF 2: Gestionar nombre del usuario.

RF 3: Mostrar índice de cada tema.

RF 4: Mostrar la información o contenido.

RF 5: Permitir ir a la pantalla principal desde cualquier parte del software, exceptuando dos escenarios.

RF6: Permitir navegar por el software sin resolver ejercicios no deseados.

RF 7: Mostrar cuadros de completamiento en los ejercicios para que el usuario los realice.

RF 8: Evaluar cada ejercicio elaborado por el usuario.

RF 9: Mostrar errores específicos de los ejercicios de completamiento en caso de que estén incorrectos.

RF 10: No permitir evaluar o calificar dos veces el mismo ejercicio, excepto cuando este sea de completamiento de tablas, u otro tipo de completamiento.

RF 11: Elaborar un fichero, que se salve en disco rígido, con la evaluación de todos los ejercicios elaborados por el usuario.

RF 12: Mostrar mensaje:

12.1 Mostrar mensaje de error cuando el usuario no introduce el nombre al sistema.

12.2 Mostrar mensaje de error cuando los ejercicios estén incompletos, excepto cuando el ejercicio sea de enviar o cuando sea de tipo checkbox.

12.3 Mostrar mensaje de error en los ejercicios de completamiento cuando el usuario introduce palabras con errores ortográficos.

12.4 Mostrar mensaje de error de cada ejercicio cuando estén incorrectos.

12.5 Mostrar mensaje de felicitación cuando se elabora un ejercicio correctamente.

12.6 Mostrar mensaje de ayuda en cualquier pantalla que el usuario desee.

RF 13: No permitir la navegación entre pantallas si el usuario ya ha trabajado en algún ejercicio de completamiento seleccionado y este no ha sido evaluado.

RF 14: Mostrar atlas de imágenes.

RF 15: Manipular fondo musical.

RF 16: Mostrar significado de las palabras claves.

RF 17: Permitir salir del software en cualquier momento.

RF 18: Mostrar créditos del producto.

2.3.2 Requerimientos no funcionales:

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Normalmente están vinculados a requerimientos funcionales, es decir una vez se conozca lo que el sistema debe hacer podemos determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

A continuación se muestra un listado con los requisitos no funcionales de la aplicación:

RNF 1. Apariencia o interfaz externa

1.1 El producto multimedia se ejecutará a pantalla completa.

1.2 El producto tendrá una pantalla de Presentación, donde se muestra su nombre.

1.3 Todas las pantallas de ejercicios cuentan con opciones similares (botón atrás, botón siguiente, botón salir, botón detener fondo musical, etc.).

1.4 La profundidad de color será de 24 bits.

RNF 2. Navegación

2.1 Desde cualquier pantalla se podrá acceder a la pantalla donde se encuentra el listado de temas (índice), excepto de un escenario y cuando aún no se tiene el nombre del usuario.

2.2 Desde cualquier pantalla se podrá salir del producto multimedia, con una previa confirmación del usuario.

RNF 3 .Servicios generales

3.1 Los servicios generales como: fondo musical, salida del sistema, etc., siempre estarán visibles al usuario durante toda la navegación que realice por las pantallas del producto.

RNF 4. Software

4.1 Se requiere un ordenador con el Flash Player8, en adelante, instalado.

RNF 5. Sistemas operativos

5.1 Microsoft Windows 98, Me.

5.2 Microsoft Windows NT, 2000, XP, Vista o superior.

2.4 Modelo conceptual.

2.4.1 Modelo de negocio

El modelo de negocio describe los procesos de un negocio y su interacción con elementos externos tales como socios y clientes, es decir, describe las funciones que el negocio pretende realizar y su objetivo básico es describir cómo el negocio es utilizado por sus clientes y socios.

En otras palabras los objetivos del modelamiento del negocio son (TELEFORMACION.UCI.CU 2007):

- Comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema.
- Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- Asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización.

- Derivar los requerimientos del sistema que va a soportar la organización.

Si los procesos están claramente definidos y no se van a introducir cambios entonces se justifica la realización de un modelo de negocio. Si se determina que no es necesario un modelo completo del negocio se realizará lo que se conoce como un modelamiento del dominio.

2.4.2 ¿Qué es el modelo de dominio?

Es una representación visual de los conceptos u objetivos del mundo real significativos para un problema o área de interés. Representa conceptos del mundo real, no de los componentes del software. Un modelo del dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema. Este tipo de modelo no incluyen las responsabilidades de las personas que ejecutan las actividades.

El modelo de dominio se describe mediante diagramas de clases de UML, donde se muestran y especifican las principales clases conceptuales (clases del dominio) que pueden intervenir en el sistema, y cómo se relacionan unas con otras mediante asociaciones.

2.4.3 ¿Modelo de negocio o Modelo de dominio?

Debido a la poca claridad en las fronteras del proceso del negocio se decide hacer un modelo de dominio que posibilite mostrar visualmente al usuario los principales conceptos que se manejan en el sistema a desarrollar. Para una mayor comprensión se realiza un diagrama de clases UML que describa las relaciones de las principales clases conceptuales del proceso de negocio.

2.4.4 Modelo de dominio

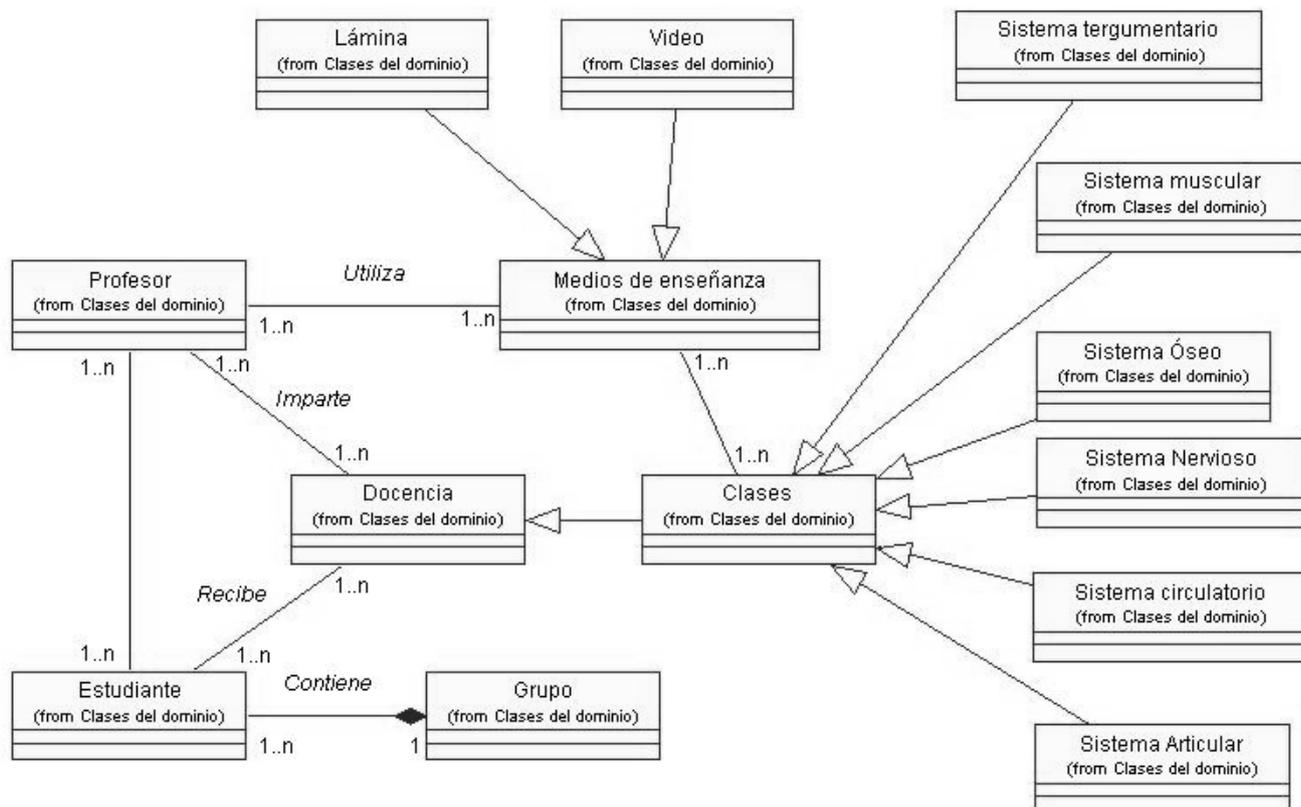


Figura 1. Modelo de Dominio

2.4.4.1 Conceptos del dominio

- **Profesor:** Persona que imparte clases a los estudiantes.
- **Estudiante:** Persona admitida en cualquier modalidad por una institución educativa, para recibir formación académica.
- **Grupo:** Es el conjunto de estudiantes.
- **Medio de Enseñanza:** Son todos aquellos instrumentos, materiales didácticos o maquetas que son utilizados en el proceso enseñanza-aprendizaje y que permiten el desarrollo de habilidades en el estudiante o le permiten ampliar su conocimiento acerca de un contenido.

- **Video:** Tecnología de captura electrónica, grabación, tratamiento, almacenamiento, transmisión y reconstrucción de una secuencia de imágenes que representan escenas en movimiento y que se utiliza como medio de enseñanza.
- **Lámina:** Media de tipo gráfico o imagen que se utiliza como medio de enseñanza.
- **Clase:** Actividad docente en la cual los estudiantes guiados por el profesor se enfrentan a la solución de problemas de su vida mediante tareas docentes en función de apropiarse de diversos contenidos y alcanzar determinados logros, basándose en métodos y estilo propios, en función de desarrollar competencias múltiples.
- **Docencia:** Conjunto de clases que se le imparten a los estudiantes. Función sustantiva de las instituciones de educación superior, mediante la cual se transmiten conocimientos, se desarrollan habilidades, se fomentan actitudes y hábitos y se induce la asunción de valores.
- **Sistema tegumentario:** Clase que trata sobre la piel y sus anexos, como son: uñas, pelos, glándulas sebáceas, sudoríparas y mamarías.
- **Sistema óseo:** Clase que trata temas sobre los huesos, como por ejemplo su función, ubicación e importancia.
- **Sistema articular:** Clase que trata sobre los ligamentos que mantienen unidos los huesos.
- **Sistema muscular:** Clase que trata sobre los músculos, su ubicación y función en el pie.
- **Sistema circulatorio:** Clase que trata sobre las arterias que componen este sistema en la región del pie, sus principales características y funciones.
- **Sistema nervioso:** Clase que trata sobre las partes en que se divide este sistema y la importancia y función que realizan cada una de ellas.

2.5 Modelo de Caso de uso del sistema:

El modelado de casos de uso es la técnica más concreta para modelar los requisitos del sistema. Los casos de uso se utilizan para modelar el funcionamiento o cómo el cliente desea que funcione el sistema. Utilizando el lenguaje OMMMA-L extendido de UML, se capturan los requisitos funcionales del sistema y se representan mediante un diagrama de casos de uso. Para ello se definen cuál o cuáles

serían él o los actores que van a interactuar con el sistema, y los casos de uso que van a representar las funcionalidades del mismo.

2.5.1 Determinación y justificación de los actores del sistema

Actor	Justificación
Usuario	Representa a la persona que va a interactuar con el sistema.

Tabla 2.Determinación y Justificación del actor del sistema.

2.5.2 Diagrama de Casos de Uso.

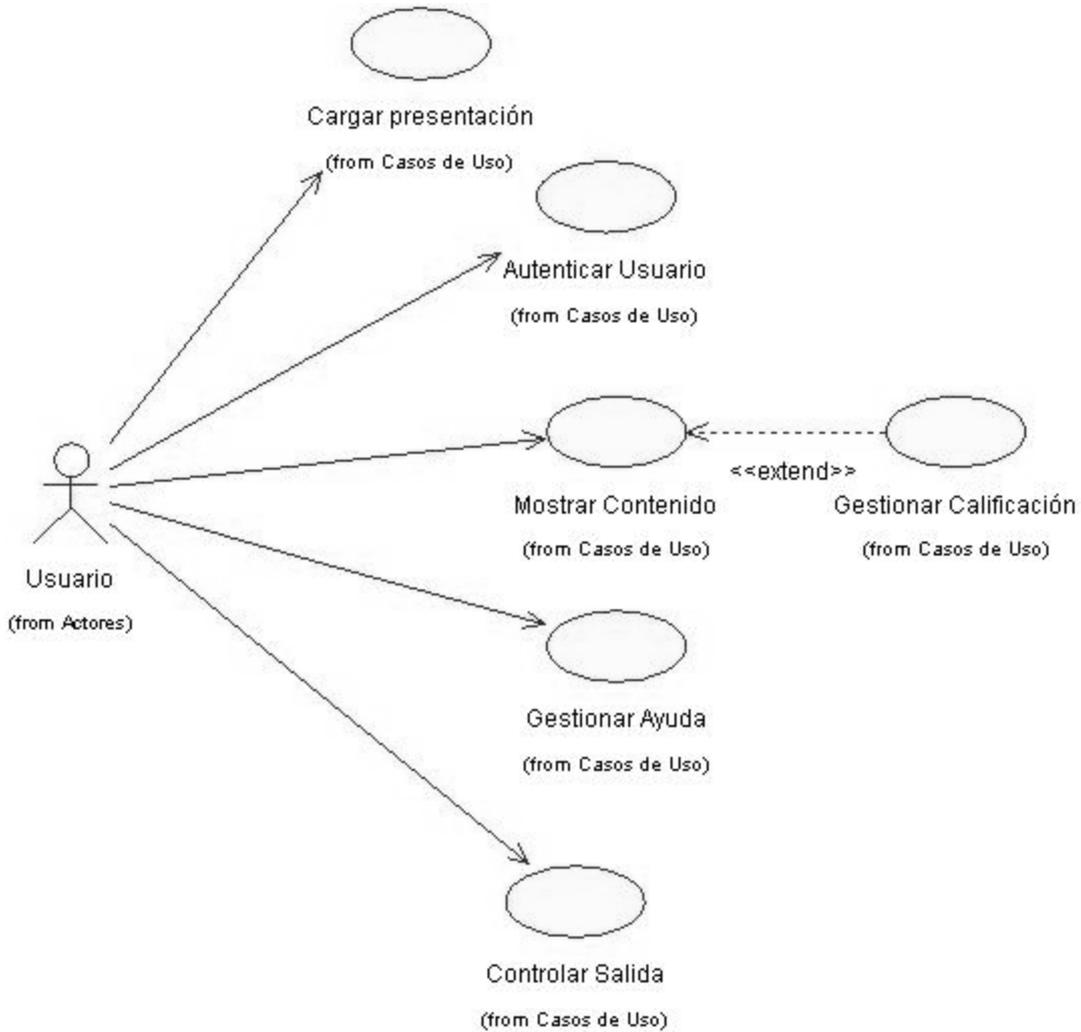


Figura 2. Diagrama de Casos de uso del Sistema

Referencia	Caso de Uso	Prioridad
CUS 1	Cargar presentación	Crítico
CUS 2	Autenticar usuario	Crítico
CUS 3	Mostrar contenido	Crítico
CUS 4	Gestionar Calificación	Crítico
CUS 5	Controlar salida del sistema	Crítico

CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

CUS 6	Gestionar Ayuda	Crítico
--------------	-----------------	---------

Tabla 3.Casos de Uso del Sistema

2.5.3 Descripción y expansión de los Casos de Uso.

Tabla 4.Descripción del CU: Cargar Presentación.

Nombre del Caso de Uso:	Cargar presentación
Actor:	Usuario(inicia)
Propósito:	Interactuar con el sistema.
Resumen:	El programa comienza con la presentación general de la aplicación, la cual no será de obligatoria visualización por parte del usuario, ya que este podrá pulsar cualquier tecla y saltar la presentación. El cursor del ratón no estará visible. Al concluir la presentación se dará paso automáticamente a la pantalla inicio del producto.
Referencias:	RF 1
Responsabilidad:	Mostrar la presentación del sistema.
CU Asociados:	
Flujo Normal de Eventos	
Acción del actor:	Respuesta del sistema:
1. El usuario solicita trabajar en la aplicación.	1.1 El sistema carga la presentación del Software para la Salud sobre Podología. 1.2 El sistema muestra la pantalla de inicio.
Cursos Alternos	
2. El usuario oprime cualquier tecla.	2.1. El sistema salta la presentación y va directamente a la pantalla de inicio.
Poscondiciones:	La presentación del producto fue ejecutada. Pantalla de inicio mostrada. Audio activado.

CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Tabla 5.Descripción de CU: Autenticar Usuario

Nombre del Caso de Uso:	Autenticar usuario
Actor:	Usuario(inicia)
Propósito:	Autenticarse en el sistema.
Resumen:	El caso de uso comienza cuando el usuario escribe su nombre, después el sistema controla el nombre del usuario para posteriormente mostrar el texto de la introducción.
Referencias:	RF 2, RF 12.1
Responsabilidad:	Controlar el nombre del usuario.
CU Asociados:	
Flujo Normal de Eventos	
Acción del actor:	Respuesta del sistema:
1. El usuario escribe su nombre en la pantalla de inicio.	1.1 El sistema controla el nombre del usuario. 1.2 El sistema muestra el texto de introducción de la aplicación.
Cursos Alternos	
2. El usuario no escribe su nombre en la pantalla de inicio.	2.1 El sistema le muestra un mensaje de error.
Poscondiciones:	Se puede navegar libremente.

CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Tabla 6.Descripción de CU: Mostrar Contenido

Nombre del Caso de Uso:	Mostrar contenido
Actor:	Usuario(inicia)
Propósito:	Mostrar el contenido de la aplicación.
Resumen:	El caso de uso se inicia cuando el usuario solicita visualizar los contenidos de una pantalla determinada, luego el sistema se encarga de obtener y mostrar el contenido solicitado.
Referencias:	RF 3, RF 4, RF 14, RF 16
Responsabilidad:	Mostrar contenido al usuario.
CU Asociados:	CU 4 << extend >>
Precondiciones:	
Flujo Normal de Eventos	
Acción del actor:	Respuesta del sistema:
1. El usuario desea visualizar el contenido seleccionado.	1.1 El sistema se encarga de mostrar el contenido correspondiente.
2. El usuario desea visualizar el significado de la palabra caliente.	2.1 El sistema se encarga de mostrar el significado de la palabra caliente.
Cursos Alternos	
Poscondiciones:	

CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Tabla 7.Descripción de CU: Gestionar Calificación

Nombre del Caso de Uso:	Gestionar Calificación
Actor:	Mostrar contenido (inicia)
Propósito:	Calificar un ejercicio elaborado por el usuario.
Resumen:	El caso de uso se inicia cuando se muestran los ejercicios y el usuario desea calificarlos, entonces el sistema se encarga de hacer la evaluación correspondiente.
Referencias:	RF 7, RF 8, RF 9, RF 10, RF 12.1, RF 12.2, RF 12.3, RF 12.4, RF 12.5, RF 13.
Responsabilidad:	Calificar un ejercicio elaborado.
CU Asociados:	
Precondiciones:	Debe haber trabajado en el ejercicio correspondiente.
Flujo Normal de Eventos	
Acción del actor:	Respuesta del sistema:
<p>1. El usuario introduce o selecciona las respuestas correspondientes.</p> <p>1.1 Si el ejercicio es de completamiento ver sección D.</p> <p>2. El usuario solicita calificar o enviar el ejercicio que ha elaborado.</p>	<p>2.1 El sistema se encarga de enviar o calificar el ejercicio correspondiente.</p> <p>2.2. Si el ejercicio es de selección, tablas, checkbox, completamiento de enlazar columnas ver Sección A, B, C, D, E respectivamente.</p>
Sección A: Ejercicios de selección (V ó F)	
	<p>2.2 Verifica que los ejercicios estén completos.</p> <p>2.3 Si la(s) respuesta(s) es (son) correcta(s), el sistema muestra un mensaje de felicitación.</p>

CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Sección B: Ejercicios de tablas	
	2.2 Verifica que los ejercicios estén completos.
Sección C: Ejercicios de checkbox	
	2.2 Si la(s) respuesta(s) es (son) correcta(s), el sistema muestra un mensaje de felicitación.
Sección D: Ejercicios de completamiento	
	2.2 Verifica que los ejercicios estén completos. 2.3 Verifica que no existan errores ortográficos. 2.4 Si la(s) respuesta(s) es (son) correcta(s), el sistema muestra un mensaje de felicitación.
Sección E: Ejercicios de enlazar columnas	
	2.2 Verifica que los ejercicios estén completos 2.3 Si la(s) respuesta(s) es (son) correcta(s), el sistema muestra un mensaje de felicitación.
Cursos Alternos	
Sección A: Ejercicios de selección (V ó F)	
	2.2.1 Muestra mensaje error indicando que los ejercicios están incompletos. 2.3.1 Si la(s) respuesta(s) es (son) incorrecta(s), el sistema muestra un mensaje de error.
Sección B: Ejercicios de tablas	
	2.2.1. Muestra mensaje error indicando que los ejercicios están incompletos.
Sección C: Ejercicios de checkbox	
	2.2.1 Si la(s) respuesta(s) es (son) incorrecta(s), el sistema muestra un mensaje de error.
Sección C: Ejercicios de completamiento	
1.1 Si desea navegar para otra pantalla.	1.1.1 El sistema le muestra un mensaje de error indicándole que debe calificar los ejercicios antes de navegar hacia otra pantalla. 2.2.1 Muestra mensaje error indicando que los ejercicios están incompletos.

CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

	<p>2.3.1 Mostrar mensaje de error indicando que existen errores ortográficos.</p> <p>2.4.1 Si la(s) respuesta(s) es (son) incorrecta(s), el sistema muestra un mensaje de error y muestra errores específicos.</p>
Sección E: Ejercicios de enlazar columnas	
	<p>2.2.1 Muestra mensaje error indicando que los ejercicios están incompletos.</p> <p>2.3.1 Si la(s) respuesta(s) es (son) incorrecta(s), el sistema muestra un mensaje de error y muestra errores específicos.</p>
Poscondiciones:	

Tabla 8. Descripción del CU: Controlar Salida

Nombre del Caso de Uso:	Controlar salida del sistema
Actor:	Usuario
Propósito:	Salir del sistema.
Resumen:	El caso de uso se inicia cuando el usuario solicita salir del sistema, entonces se muestra la pantalla de créditos.
Referencias:	RF 11, RF 17, RF 18
Responsabilidad:	Permitir la salida del sistema.
CU Asociados:	
Flujo Normal de Eventos	
Acción del actor:	Respuesta del sistema:
1. El usuario solicita salir del sistema.	1.1 El sistema crea y guarda el fichero evaluativo con el nombre e información del estudiante. 1.2 El sistema verifica que el usuario desee salir.
2. Confirma la salida.	2.1 El sistema va a la pantalla de créditos.
Cursos Alternos	
2.1. El usuario no desea salir del sistema.	2.1 El sistema sigue suministrando funcionalidades.
Poscondiciones:	

Tabla 9.Descripción del CU: Gestionar Ayuda

Nombre del Caso de Uso:	Gestionar Ayuda
Actor:	Usuario(inicia)
Propósito:	Buscar ayuda del sistema.
Resumen:	El caso de uso se inicia cuando el usuario solicita la opción de ayuda del sistema, entonces el sistema muestra la pantalla con la información correspondiente
Referencias:	RF 12.6
Responsabilidad:	Mostrar el contenido referido en esta opción.
CU Asociados:	
Flujo Normal de Eventos	
Acción del actor:	Respuesta del sistema:
1. El usuario solicita la opción de ayuda del sistema.	1.1. El sistema a partir de la solicitud realizada se encarga de obtener la información. 1.2. El sistema muestra la pantalla con la información correspondiente.
Cursos Alternos	
Poscondiciones:	

2.6 Diagrama de navegación

El diagrama de navegación responde el cómo se ven los temas, sub-temas o contenidos. Brinda una visión de lo que se desea lograr, y ayuda a crear una idea del camino que se debe seguir.

2.6.1 Diagrama de navegación general.

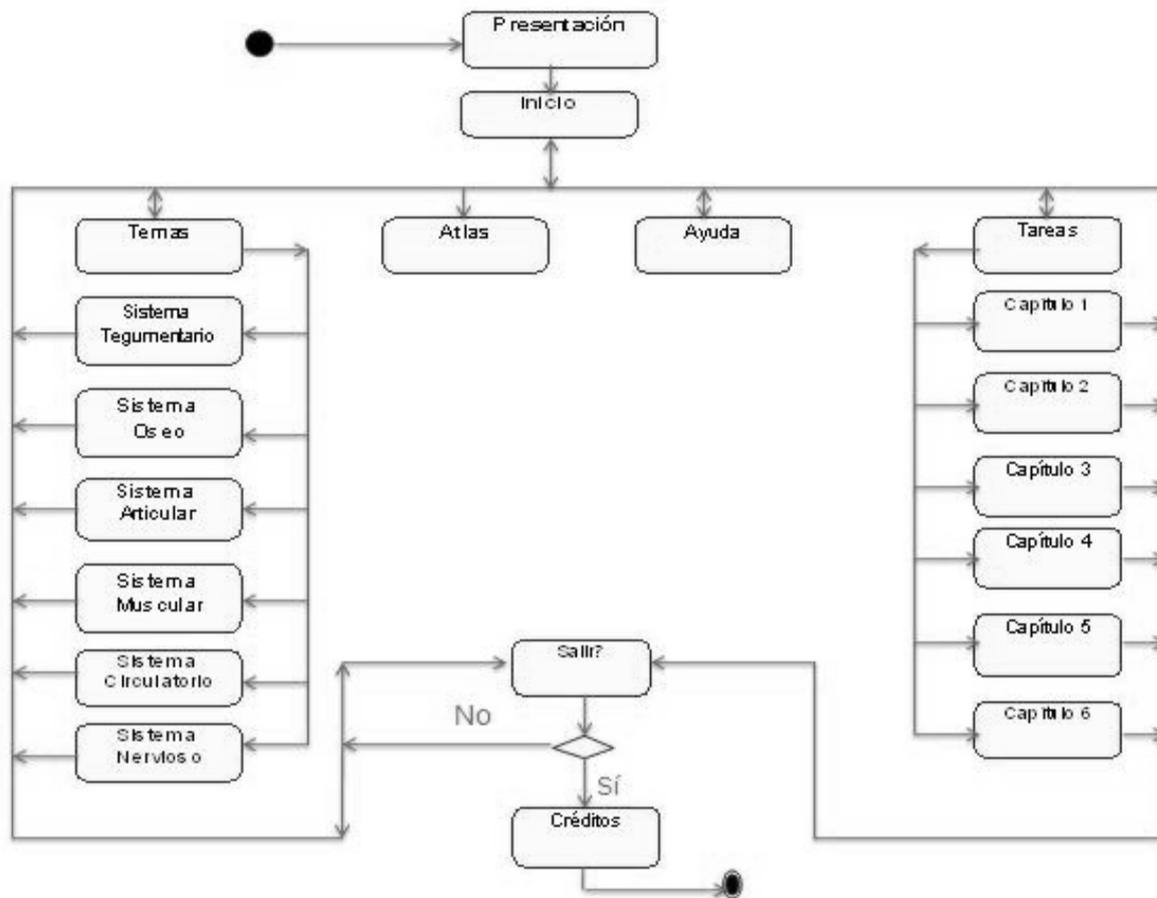


Figura 3. Diagrama de Navegación General

2.6.2 Diagrama de navegación: Ejercicios.

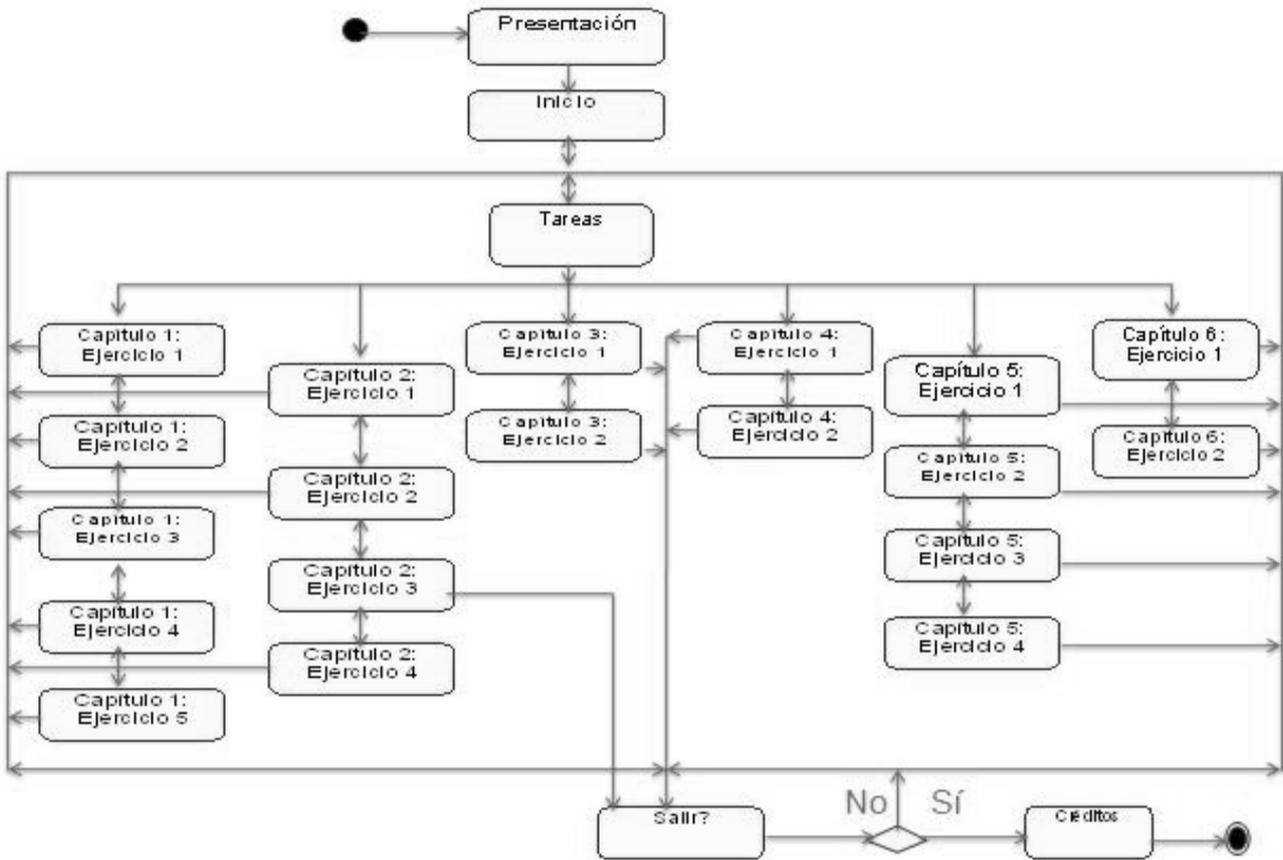


Figura 4. Diagrama de Navegación: Ejercicios

2.7 Conclusiones parciales

Lograr una comunicación efectiva entre los clientes y el equipo del proyecto, con el objetivo de llegar a un entendimiento de lo que hay que hacer, es la clave del éxito en la producción del software. Durante muchos años las aplicaciones han fallado porque existieron incongruencias entre lo que el usuario quería, lo que realmente necesitaba, lo que interpretaba cada miembro del proyecto y lo que realmente se obtenía, de ahí la importancia de modelar el negocio cuidadosamente y de adaptar los requerimientos de nuestro proyecto a lo que realmente desea el cliente.

CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

El modelo de dominio propuesto en este capítulo conllevó a comprender mejor la estructura y la dinámica de las organizaciones donde se va a implantar la aplicación.

La captura de requisitos es un paso fundamental en el desarrollo de software, pues nos muestran qué debe hacer el sistema y qué cualidades debe poseer. En este capítulo se definieron estos requerimientos, los cuales fueron reflejados en el Modelo de Casos de uso del Sistema. Por último se definió la navegación que debe poseer la aplicación con tecnología multimedia.

Capítulo 3: Construcción de la Solución Propuesta

Introducción.

En el presente capítulo se mostrará la construcción de la solución propuesta, a través de los flujos de trabajo de diseño e implementación. Se expondrán los diferentes Diagramas de Presentación que permitirán representar los elementos de cada uno de los escenarios con que cuenta la aplicación con tecnología multimedia. Se mostrará el Diagrama de Jerarquía de Clases correspondiente a la aplicación. Para darle cumplimiento a los artefactos generados en el flujo de trabajo de diseño se confeccionarán los Diagramas de Clases del Diseño de cada caso de uso, así como las realizaciones de estos mediante los Diagramas de Secuencia. Se representará el Diagrama de Despliegue, plasmando el hardware necesario para el funcionamiento de la aplicación y finalmente el Diagrama de Componente, visualizando los componentes físicos generados en la aplicación.

3.1 Diagramas de Presentación.

OMMMA-L presenta un nuevo diagrama: El Diagrama de Presentación. Estos diagramas describen la parte estática del modelo a través de una descripción intuitiva de la distribución espacial de objetos visuales de la interfaz de usuario. Este diagrama nos permitirá representar los elementos de cada uno de los escenarios con que cuenta el producto, de forma general. Utilizaremos tres objetos para describir los elementos de la pantalla: escenario: que representa las diferentes pantalla del software; aplicación: que agrupa los elementos de media y el objeto media: que hacen referencia a sonido, imagen, video, texto, etc. (CATALÁ *et al.* 2007)

Figura 5. Diagrama de Presentación: General

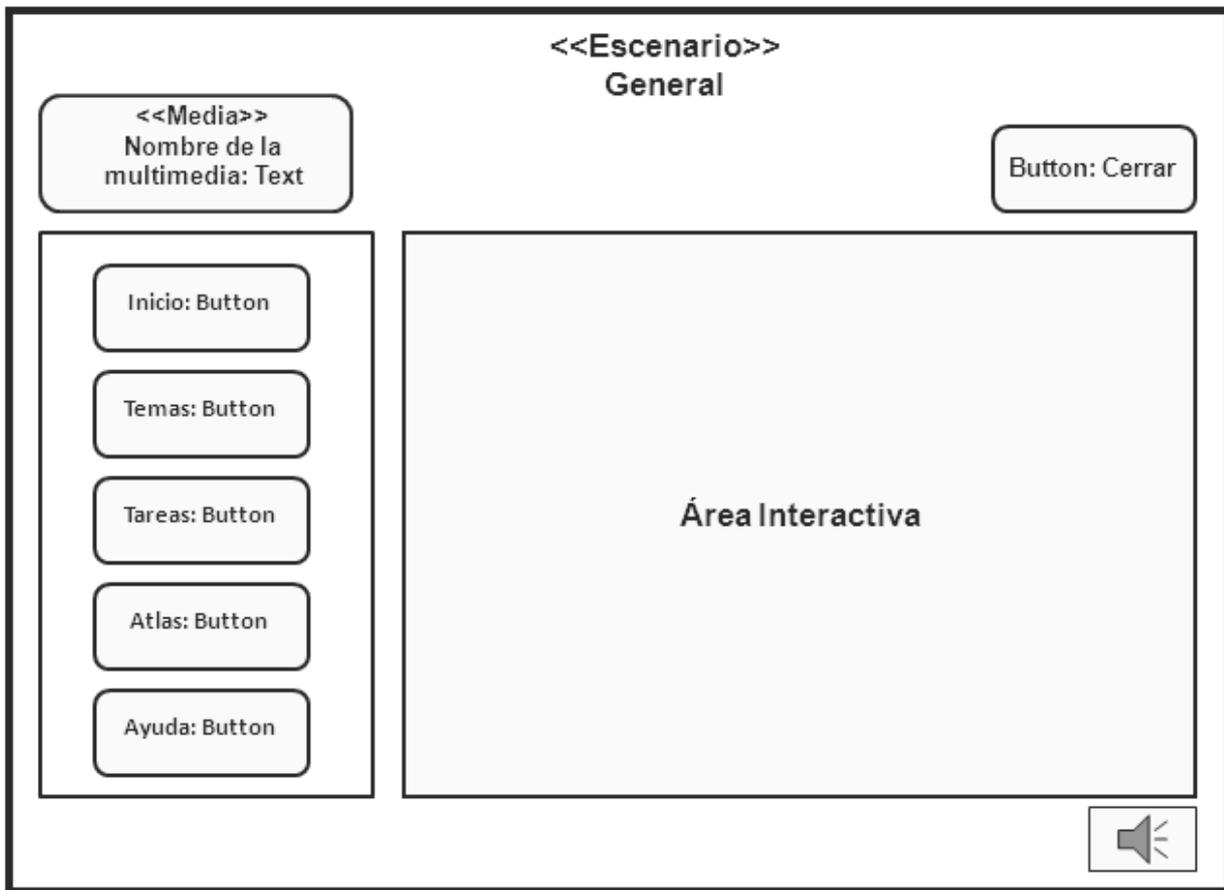


Figura 6. Diagrama de Presentación: Inicio

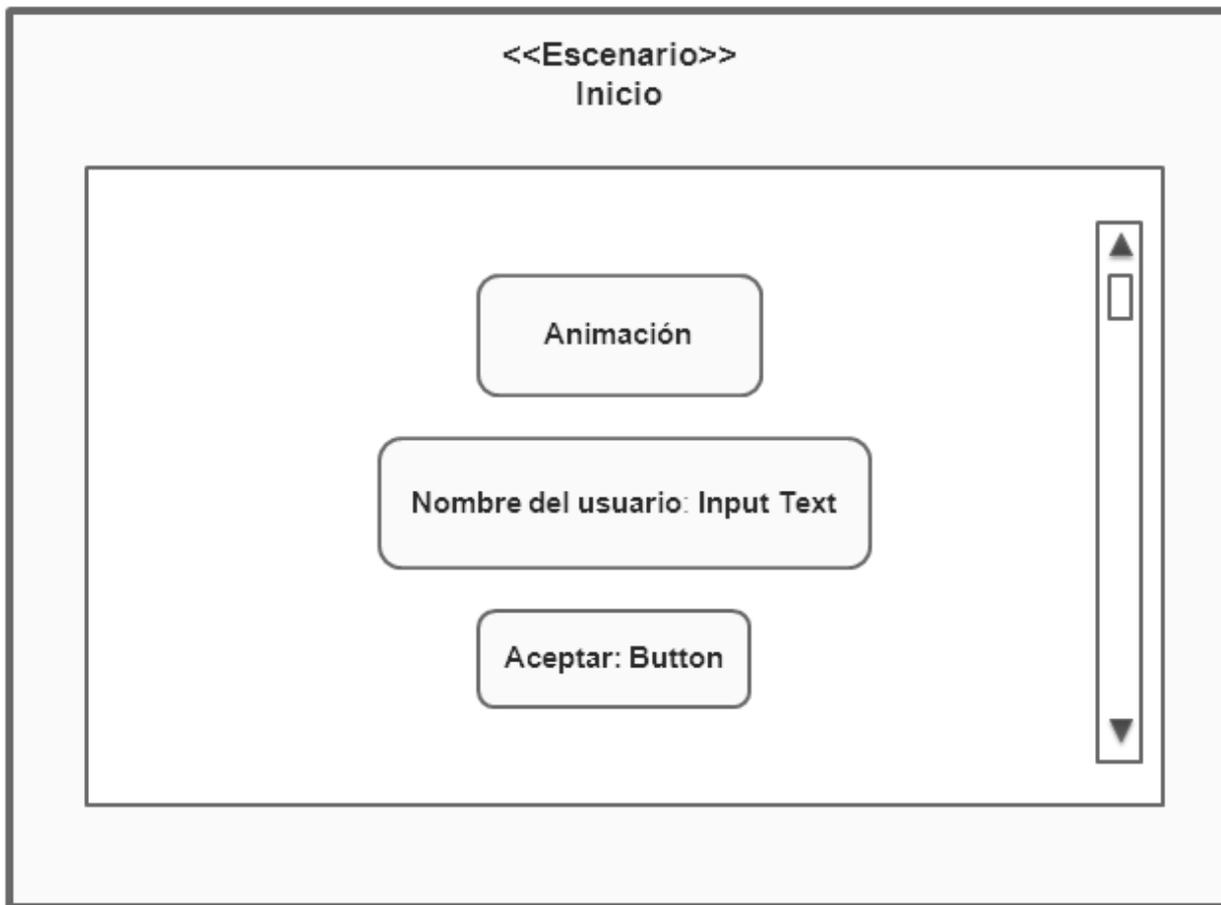


Figura 7. Diagrama de Presentación: Índice

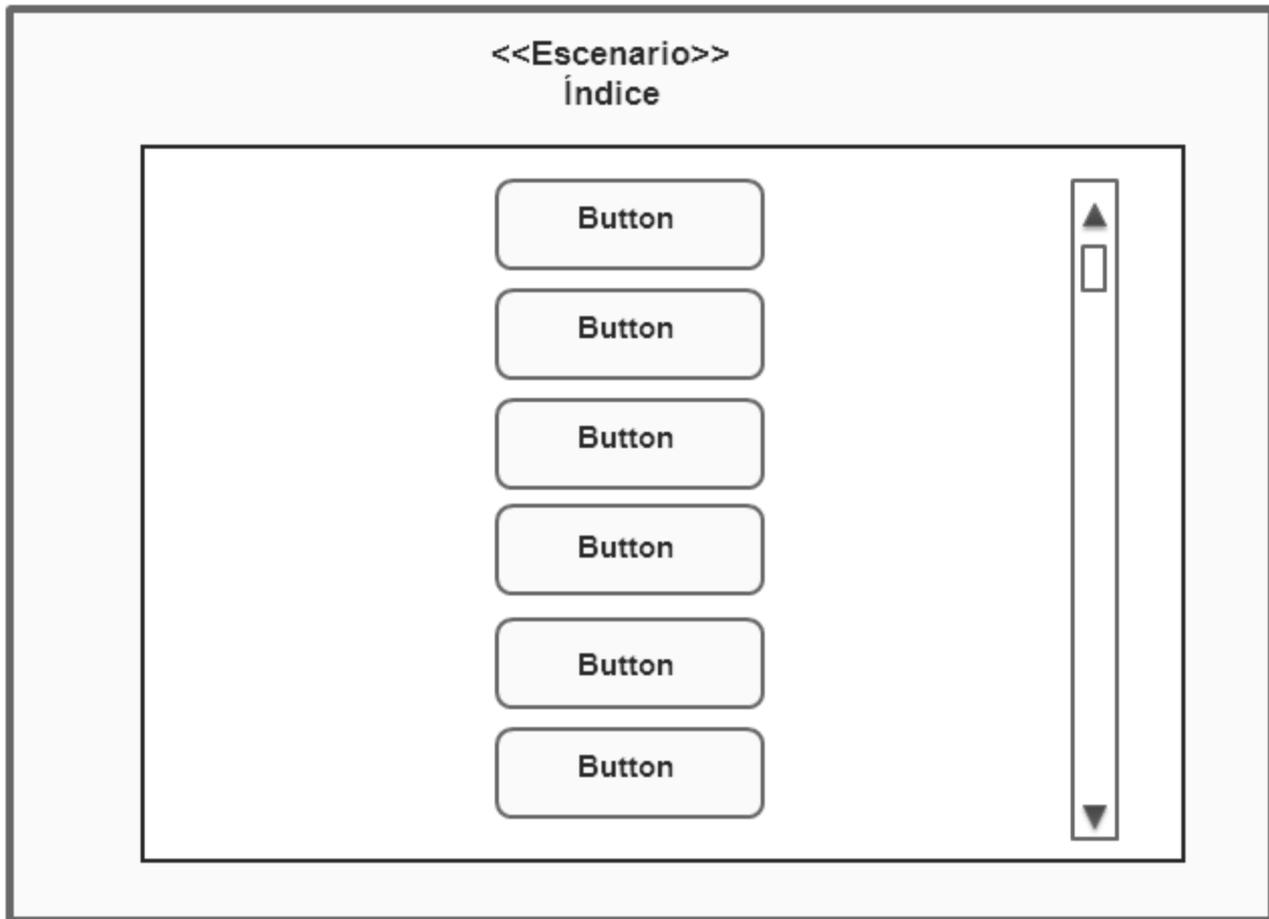


Figura 8. Diagrama de Presentación: Textos

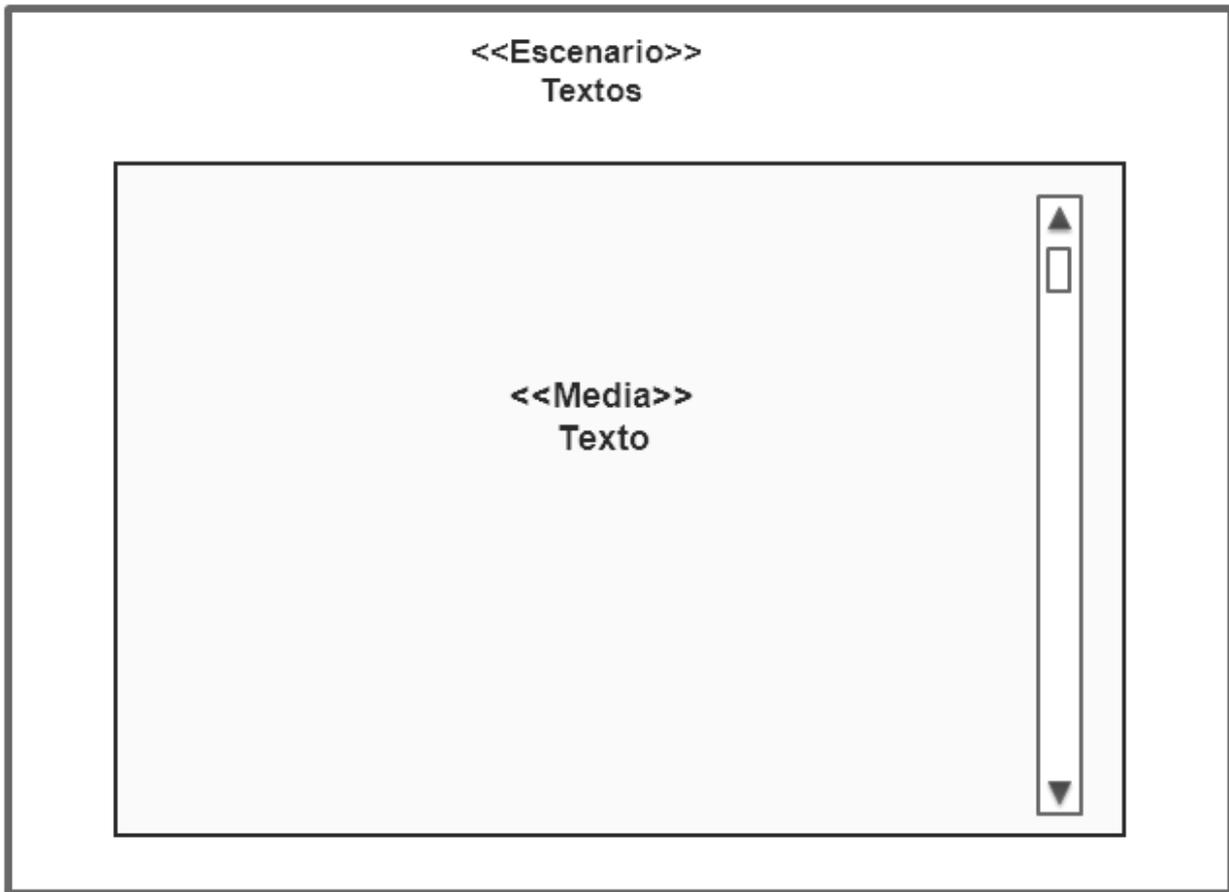


Figura 9. Diagrama de Presentación: Glosario

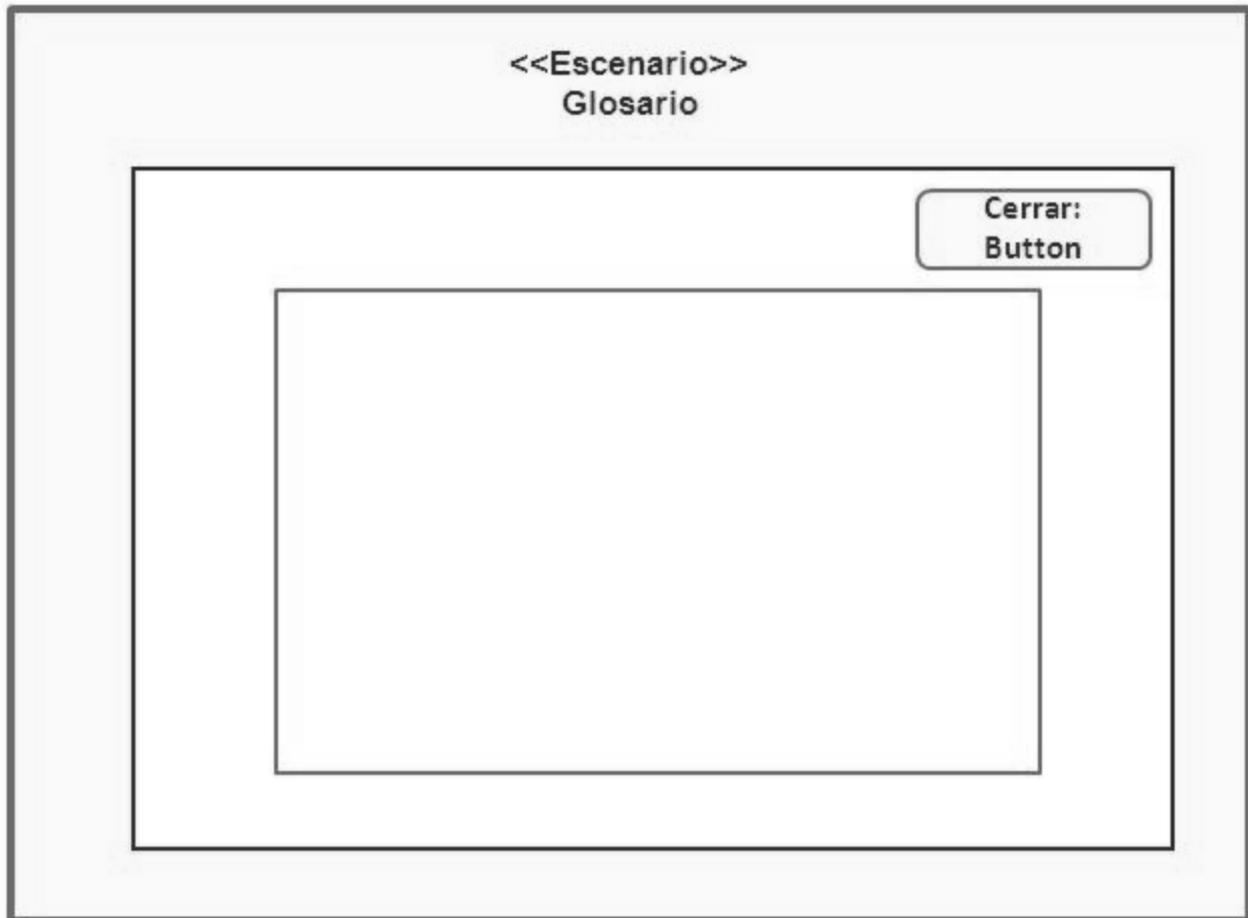


Figura 10. Diagrama de Presentación: Ejercicios de selección (V o F)

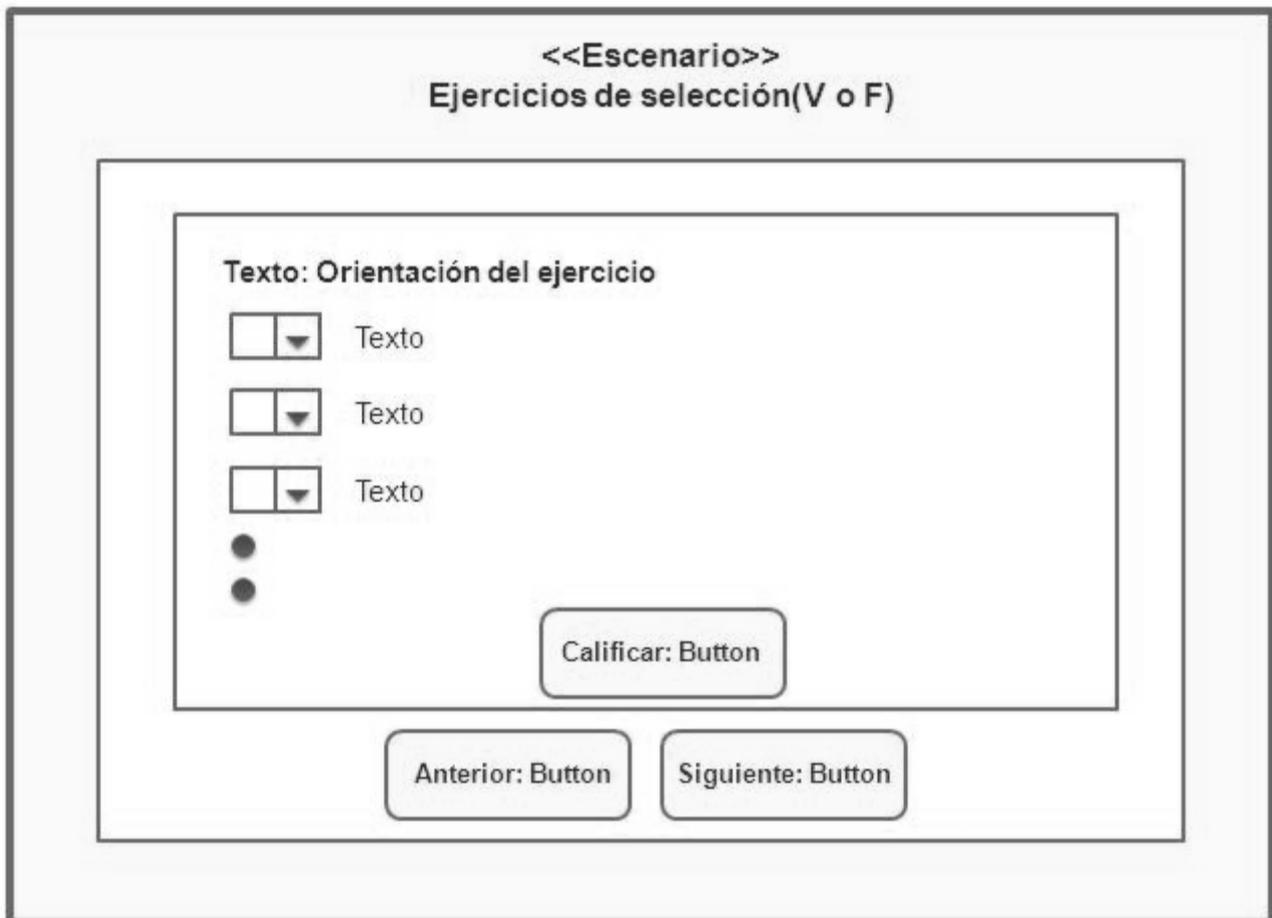


Figura 11. Diagrama de Presentación: Ejercicios de Enviar

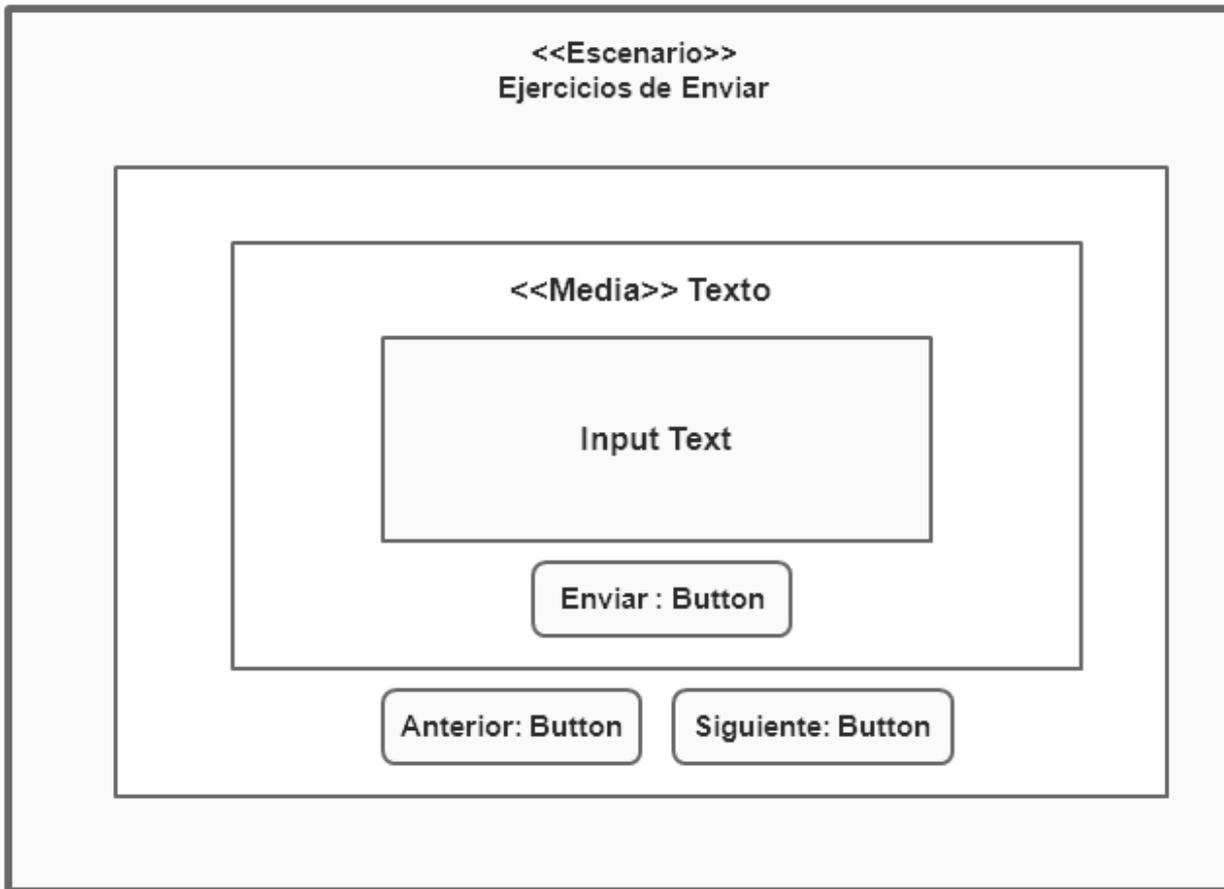


Figura 12. Diagrama de Presentación: Ejercicios de Cuadro

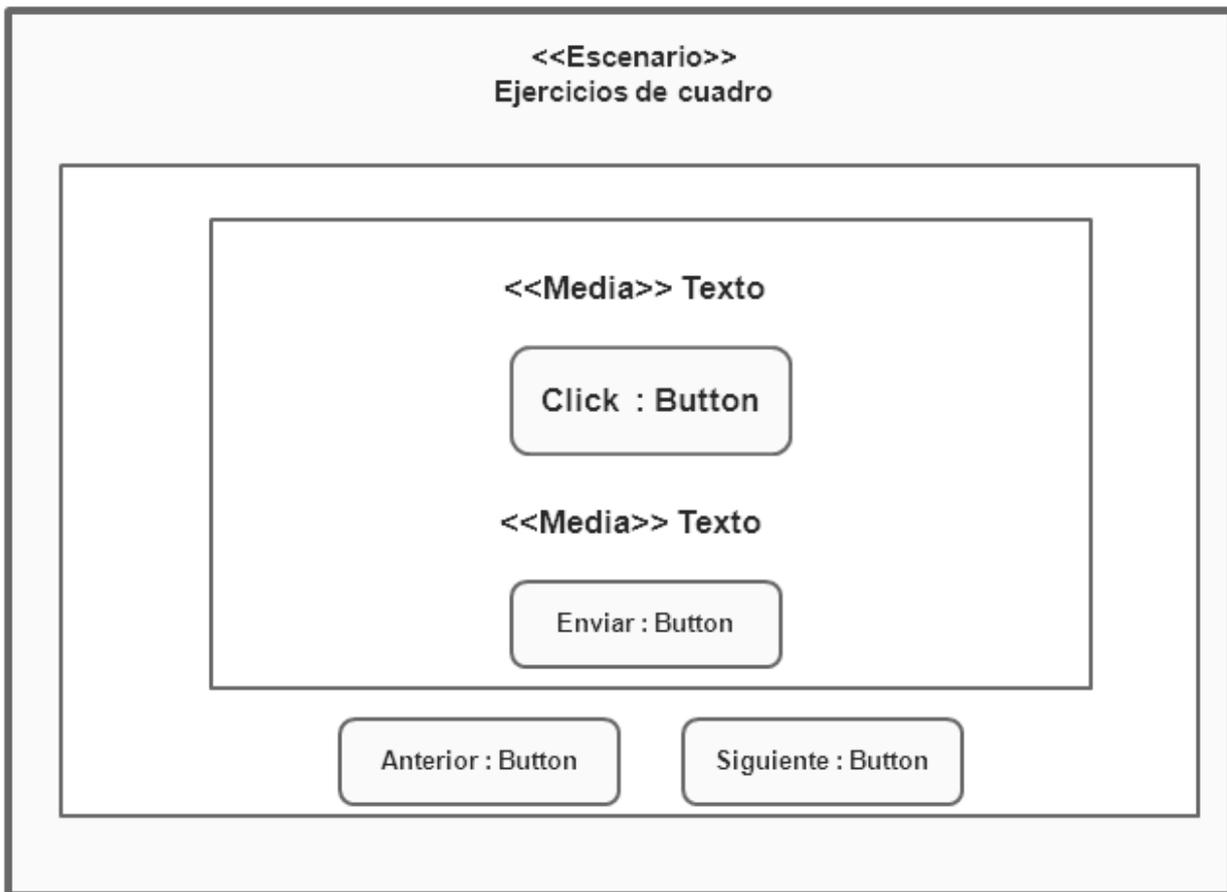


Figura 13. Diagrama de Presentación: Ejercicios de Checkbox

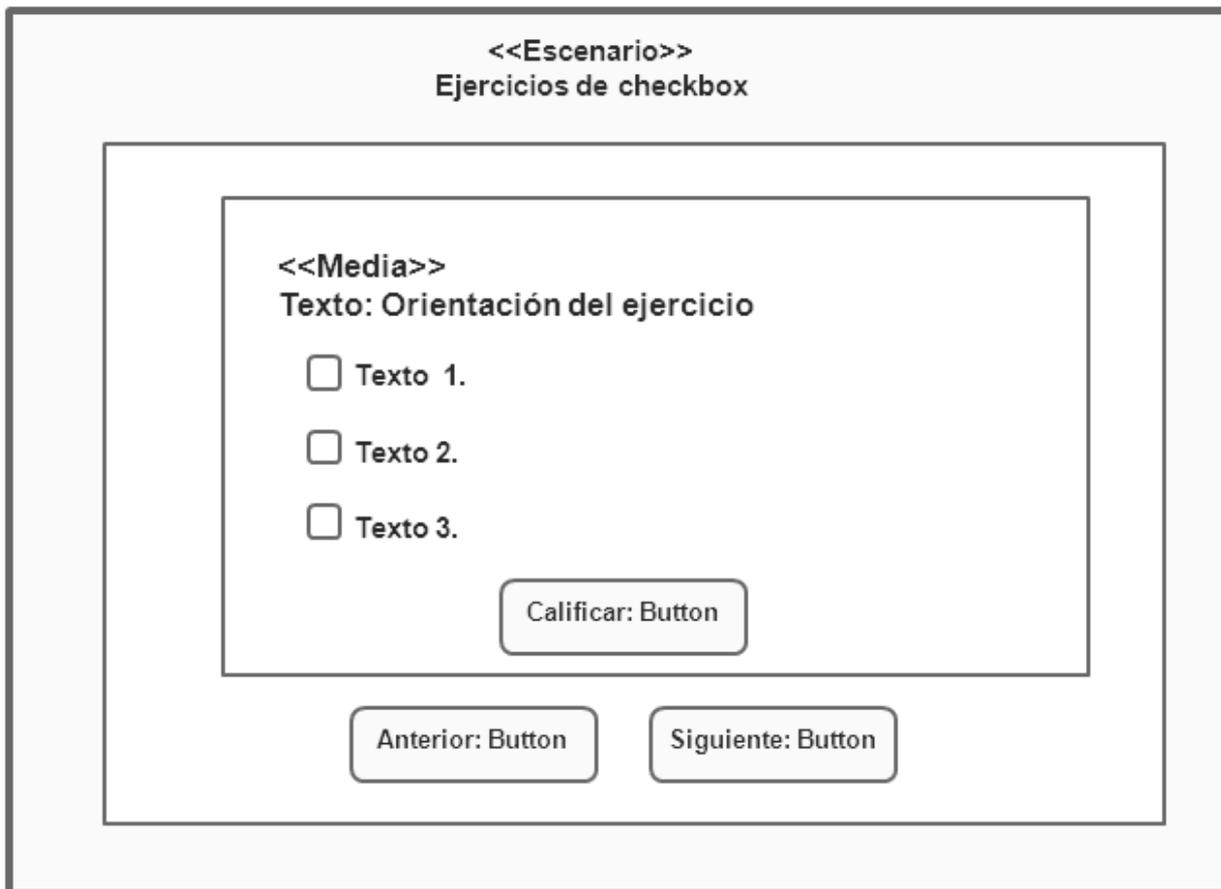


Figura 14. Diagrama de Presentación: Ejercicios de Completamiento

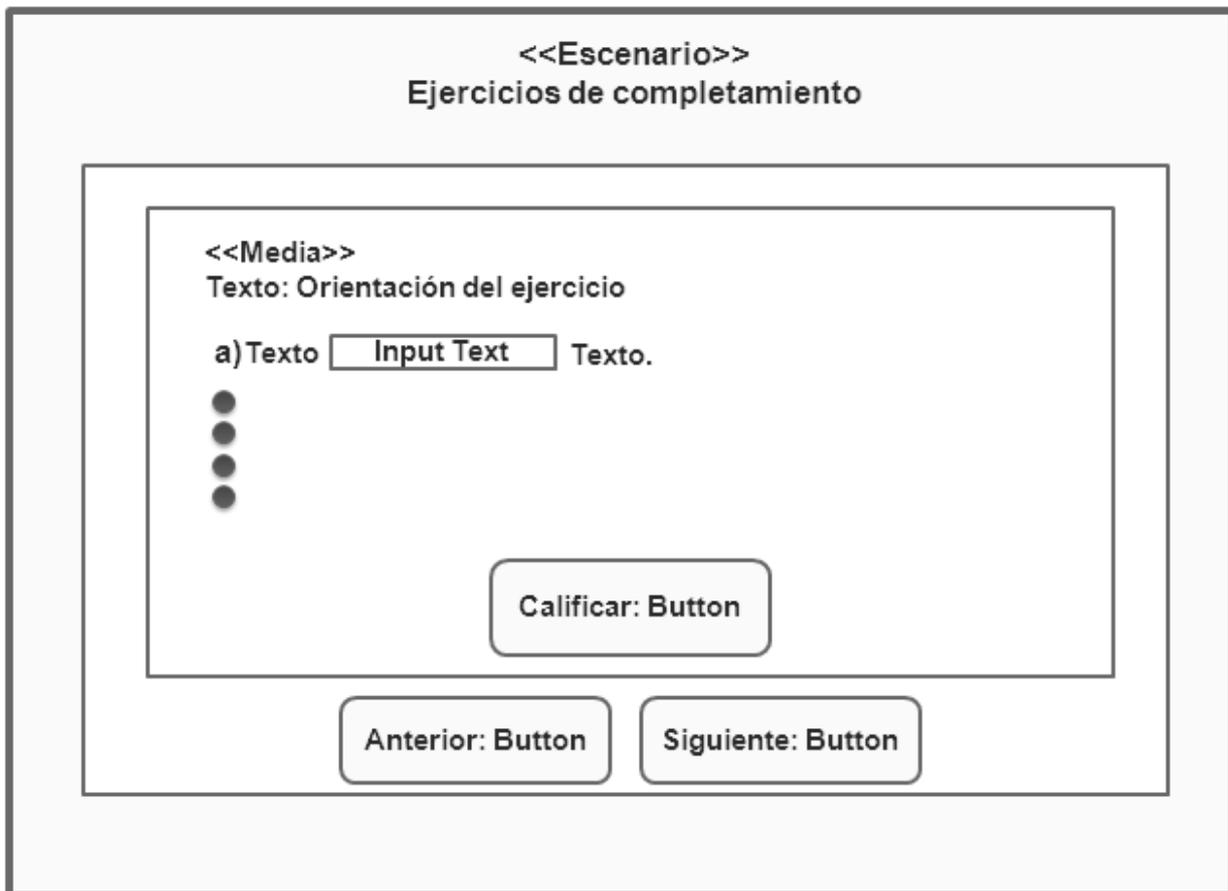


Figura 15. Diagrama de Presentación: Ejercicios de Enlazar Columnas

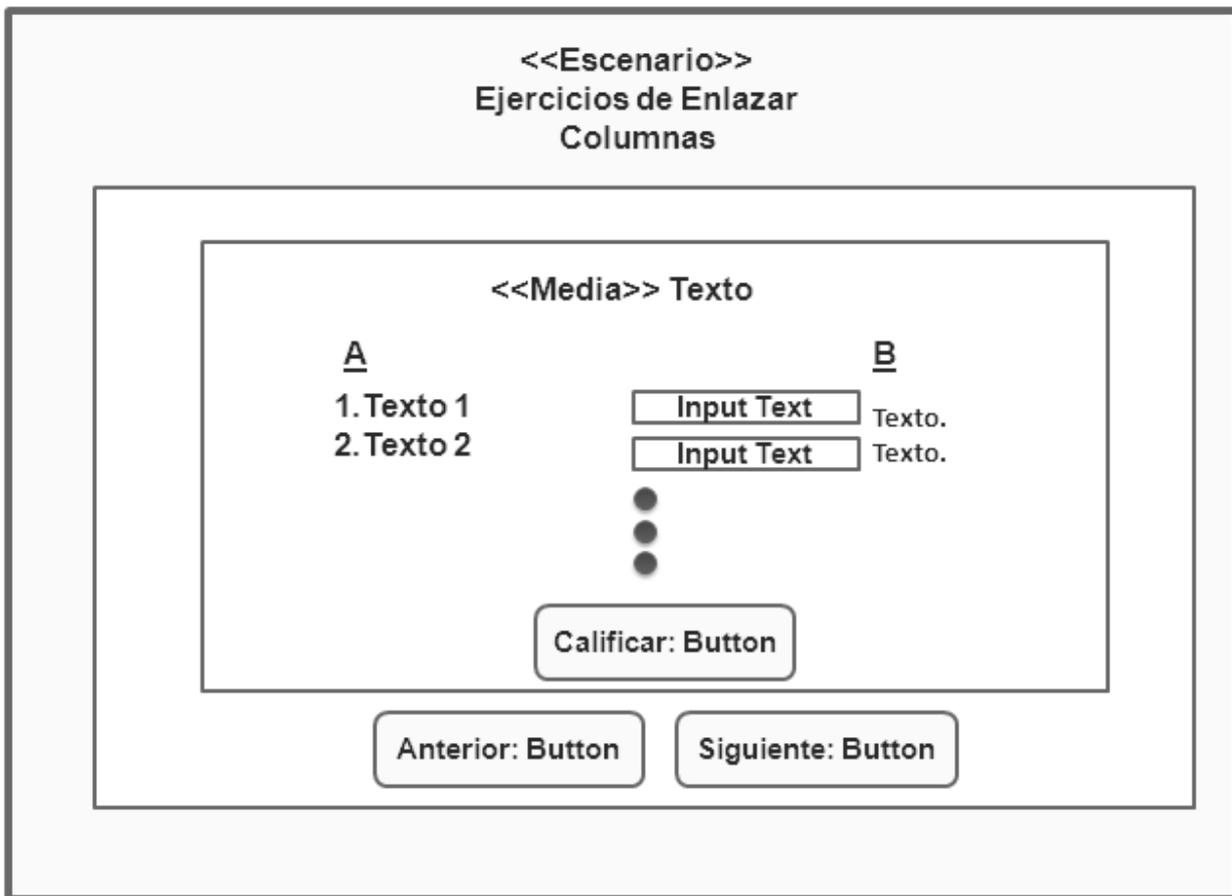


Figura 16. Diagrama de Presentación: Ejercicios de Tablas

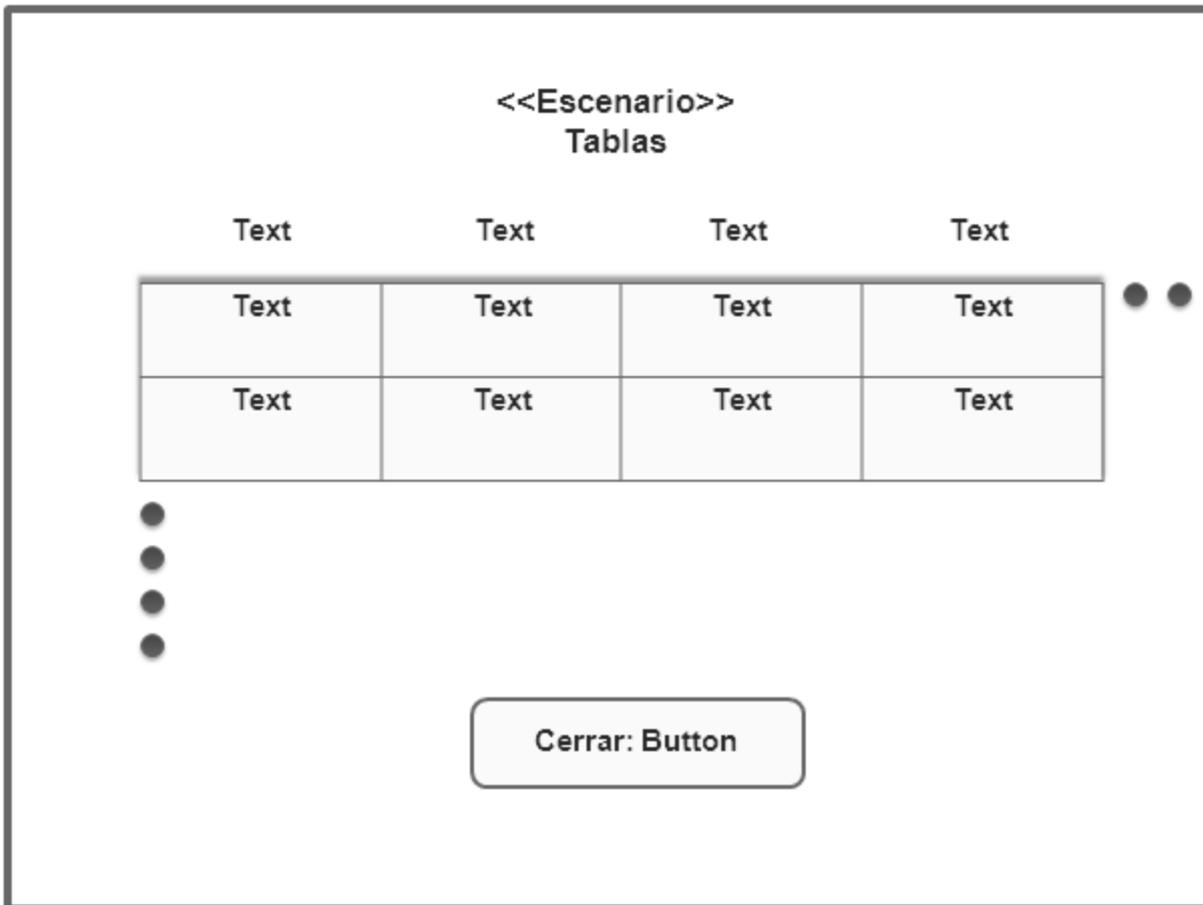
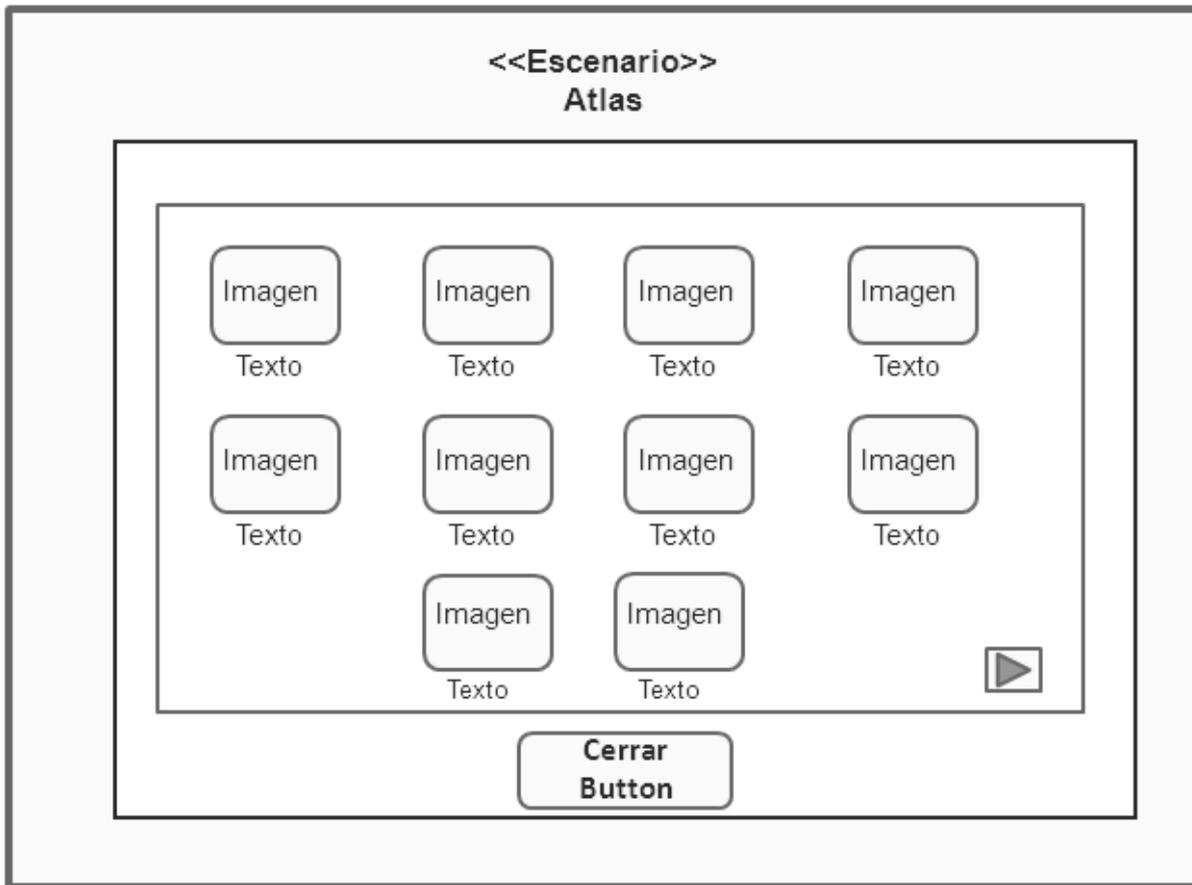


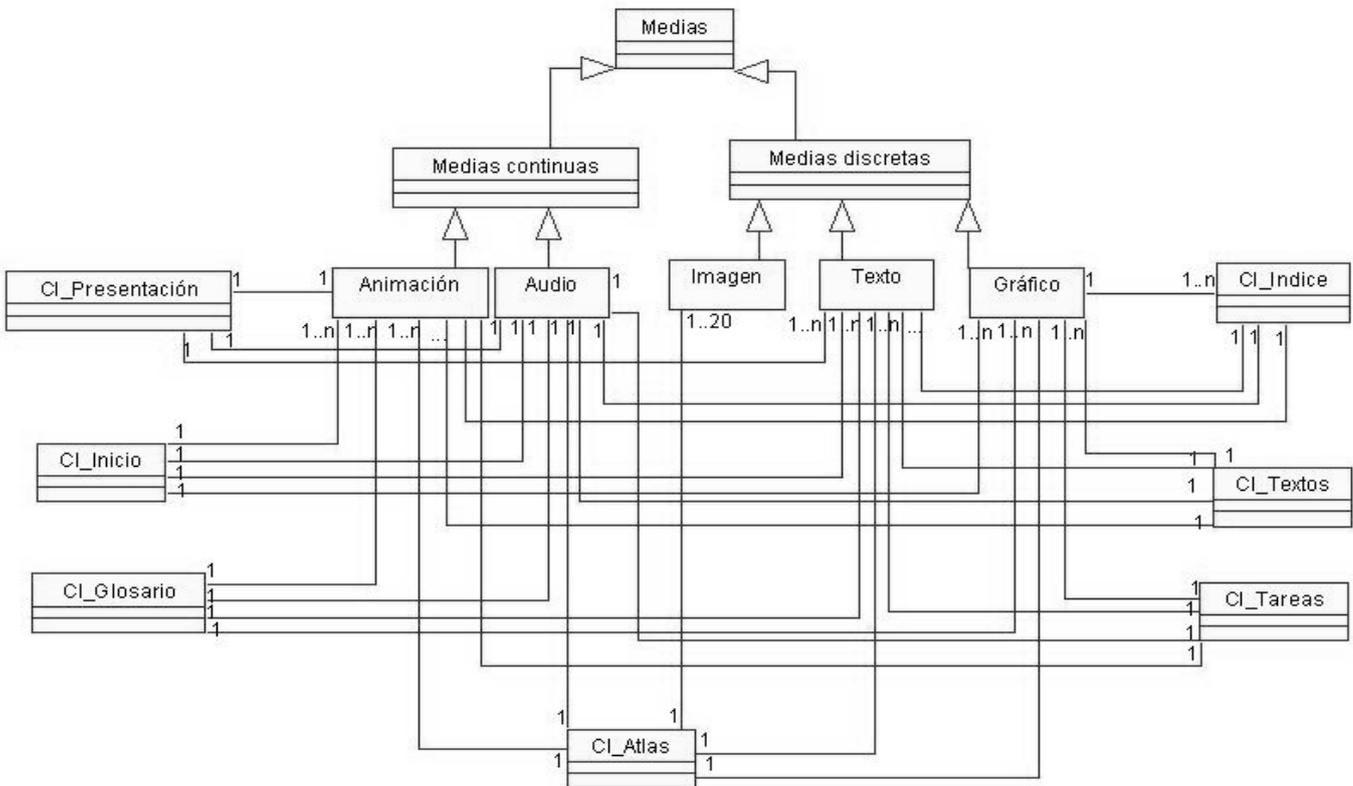
Figura 17. Diagrama de Presentación: Atlas



3.2 Diagrama de jerarquía de clases.

El Diagrama de Clases de OMMMA-L, extendido del Diagrama de Clases de UML propone adicionar en cada diagrama de clases elaborado, la jerarquía de media de la herramienta y enlazar a través de relaciones las clases del tipo correspondientes. Aquí se incorporan las clases correspondientes a las medias: media continua y media discreta, generalizadas en una clase medias.

Figura 18. Diagrama de Jerarquía de Clases



3.3 Diagrama de clases de diseño

El diagrama de clases de diseño describe los tipos de objetos que hay en el sistema y las diversas clases de relaciones (asociatividad, herencia, etc.) que existen entre ellos. Además muestra los atributos y operaciones de una clase y las restricciones a que se ven sujetos, según la forma en que se conecten los objetos. Un diagrama de clases está compuesto por los siguientes elementos: Clase (Atributos, métodos y visibilidad) y Relaciones (Herencia, Agregación, Asociación y Composición).

Los autores consideran que se debe usar este diagrama, con el objetivo de mostrar las distintas relaciones entre las clases de la aplicación, así como sus métodos y atributos. Esto conlleva a un mejor entendimiento de la parte lógica del sistema. Cada diagrama está compuesto por las clases que intervienen en la realización de un caso de uso.

Figura 19. Diagrama de Clases CU: Autenticar Usuario

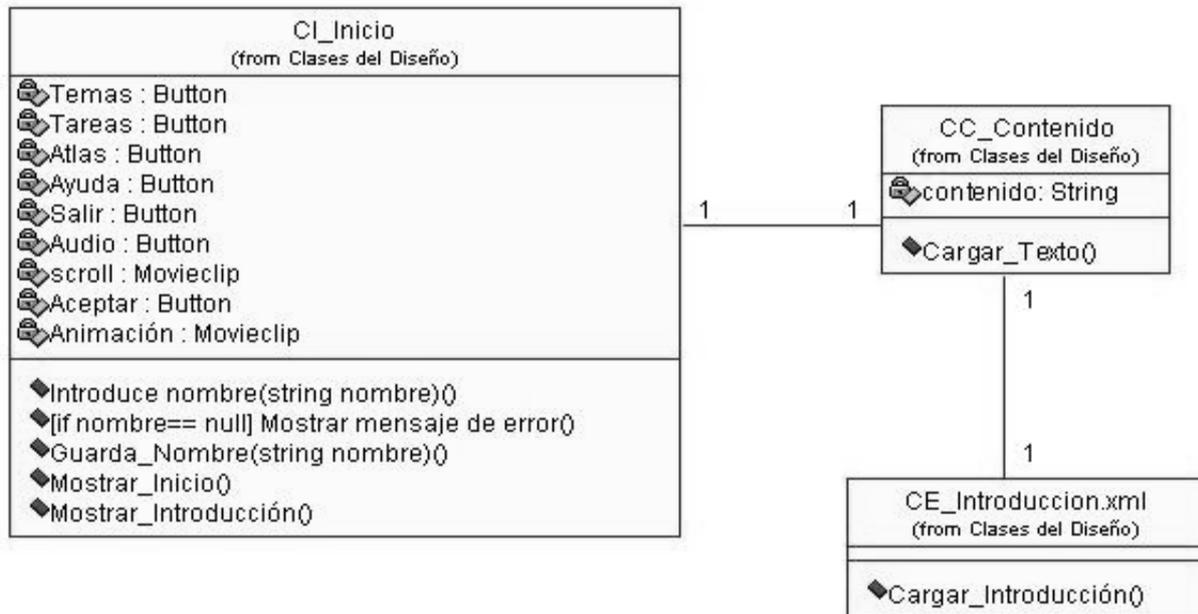


Figura 20. Diagrama de clases CU: Cargar Presentación

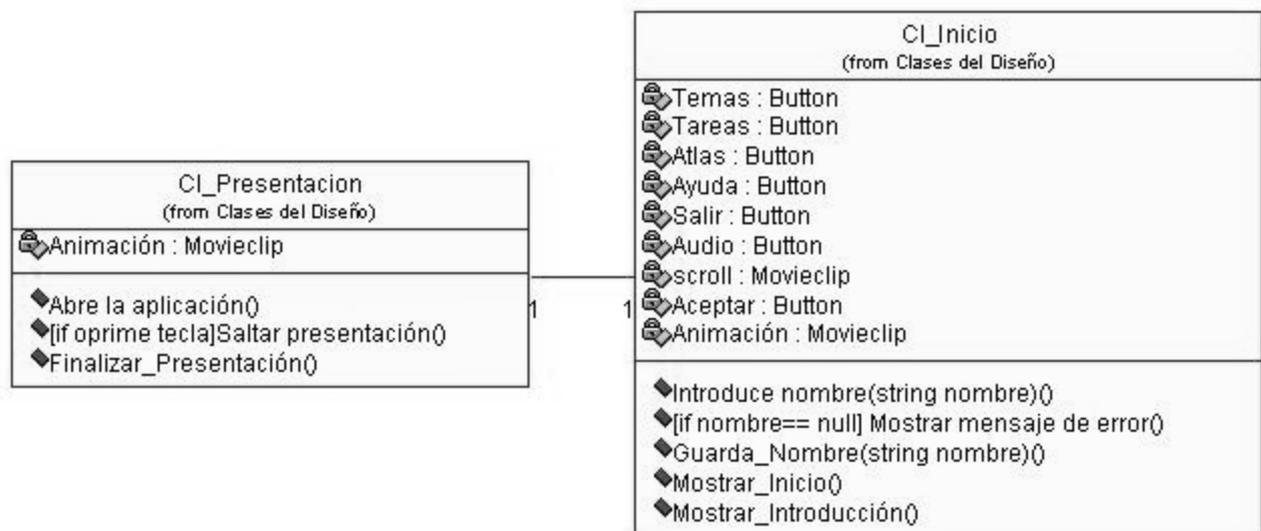


Figura 21. Diagrama de clases CU: Mostrar Contenido

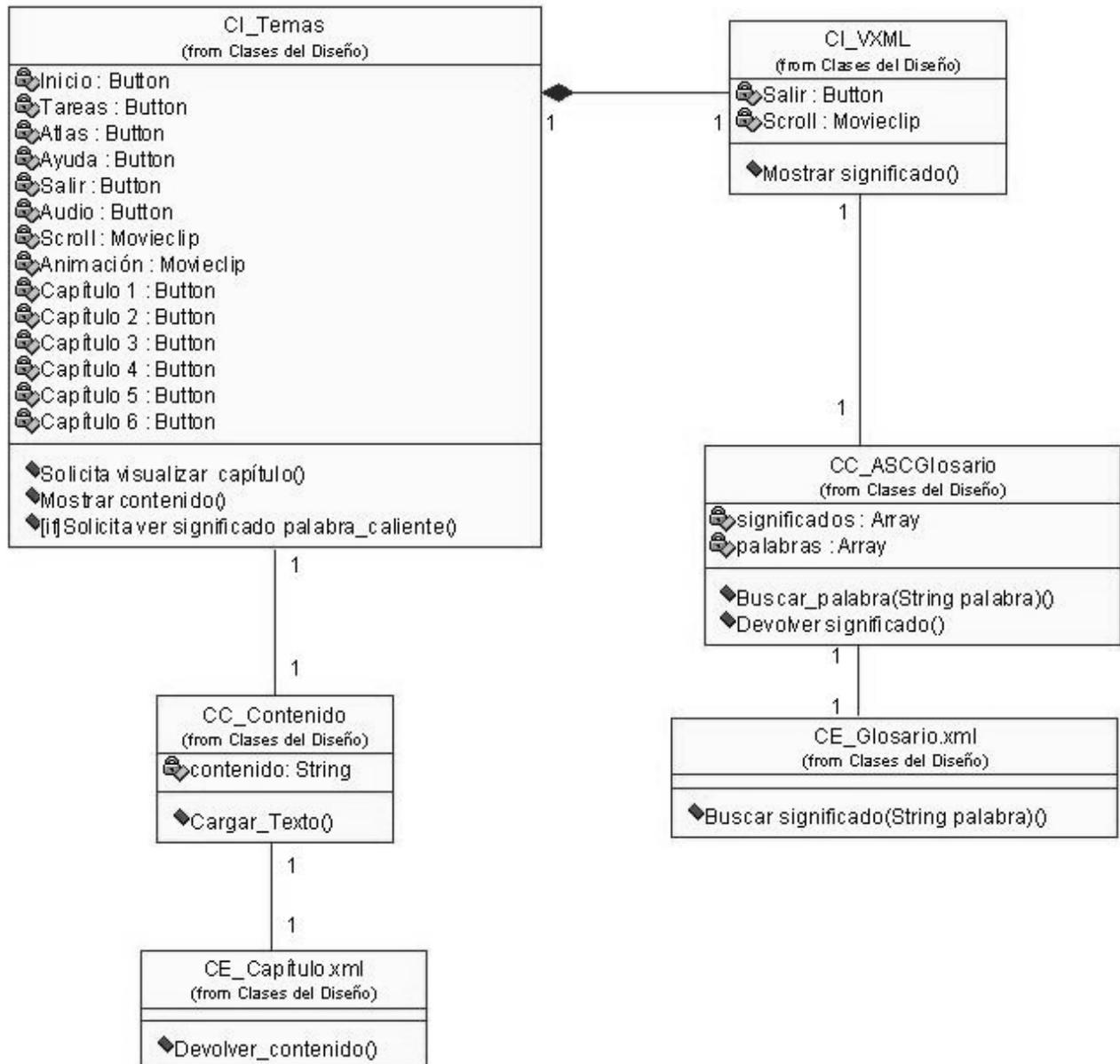


Figura 22. Diagrama de clases CU: Gestionar Calificación

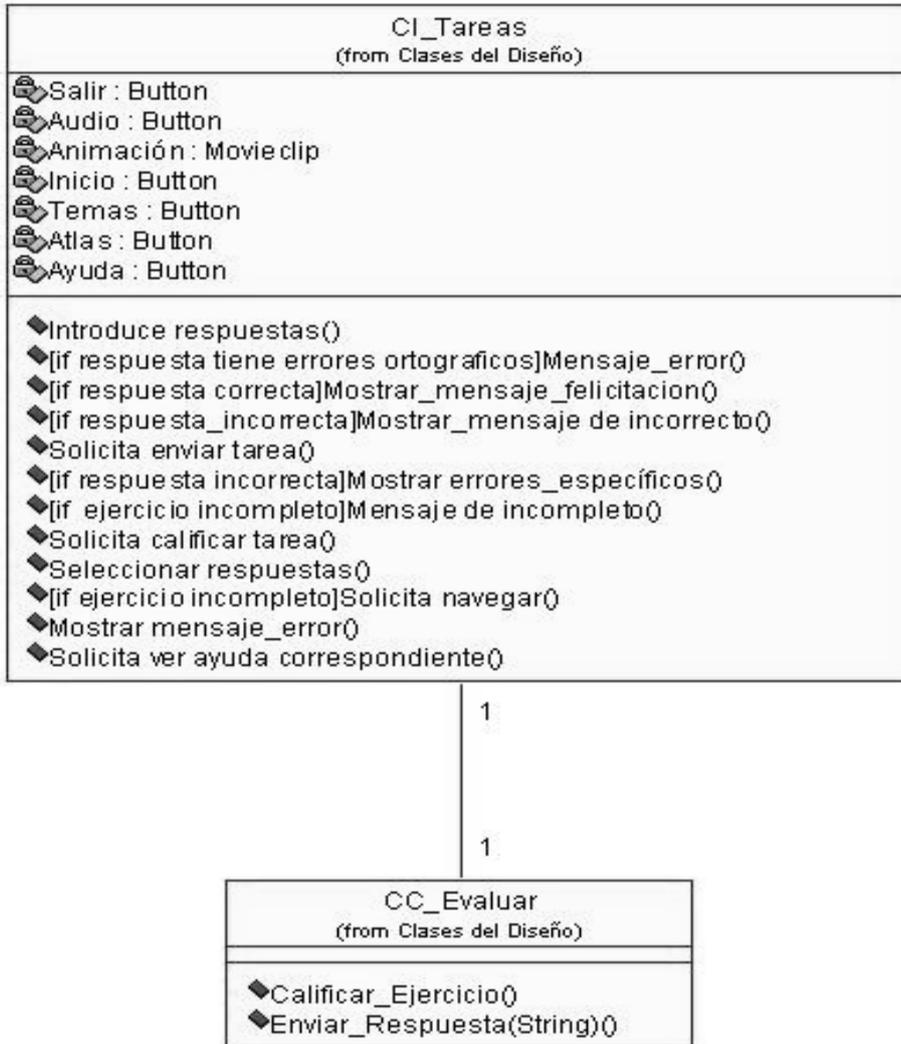


Figura 23. Diagrama de clases CU: Gestionar Ayuda

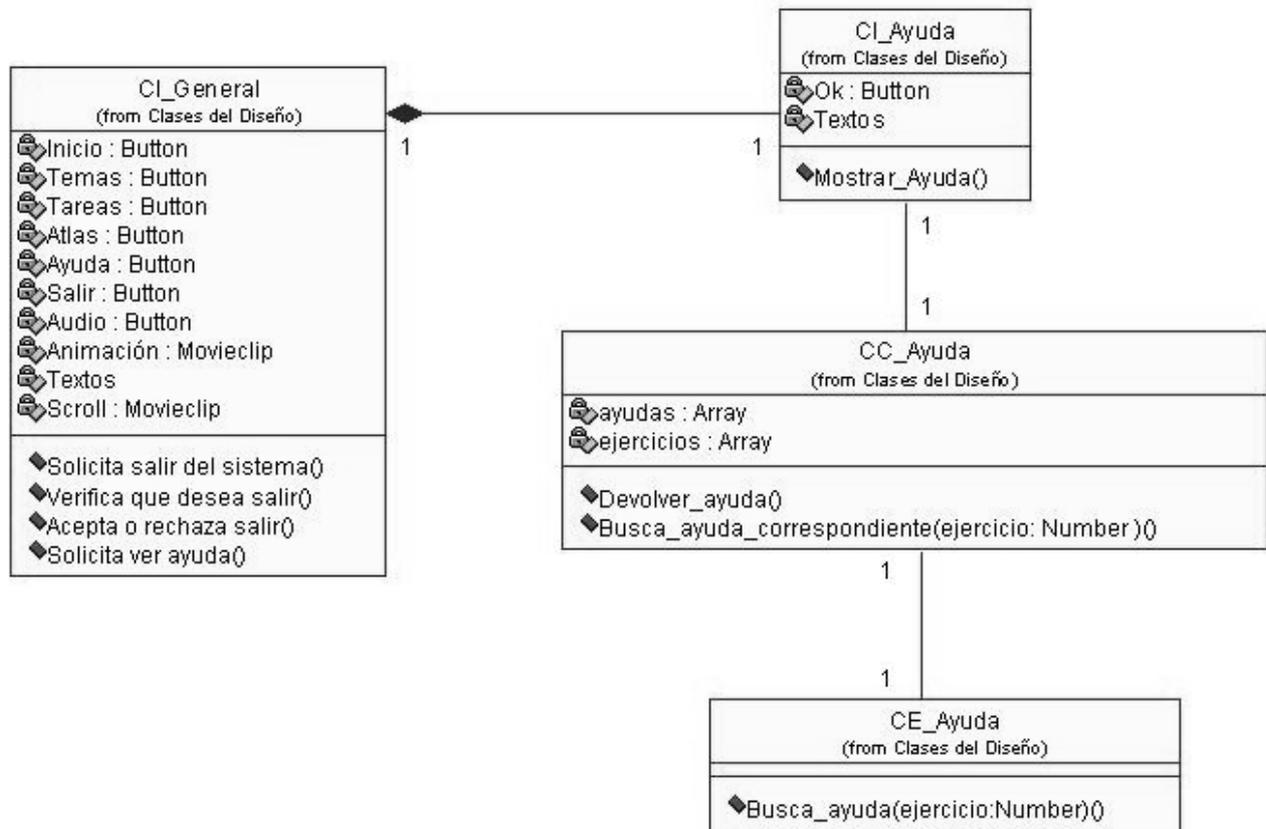
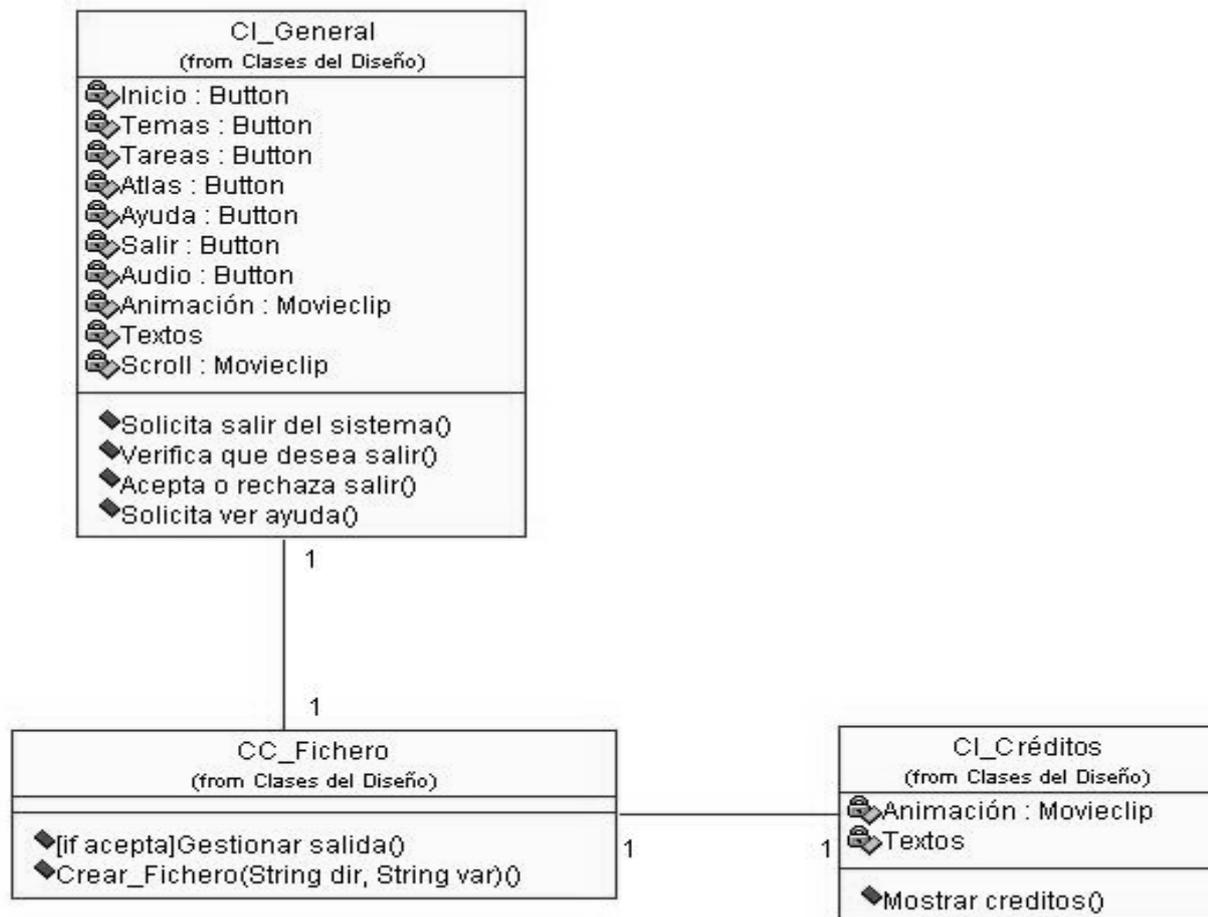


Figura 24. Diagrama de clases CU: Controlar Salida



3.4 Diagramas de secuencia.

El Diagrama de secuencia que plantea OMMMA - L modela el comportamiento temporal predefinido de una aplicación con tecnología multimedia, la parte dinámica del aspecto Modelo del MVC. 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 a la misma línea de tiempo.

Diagrama de secuencia CU: Autenticar Usuario. (Ver anexo 34)

Diagrama de secuencia CU: Cargar Presentación. (Ver anexo 35)

Diagrama de secuencia CU: Mostrar Contenido. (Ver anexo 36)

Diagrama de secuencia CU: Gestionar Calificación, Sección A: ejercicios de selección (V o F). (Ver anexo 37)

Diagrama de secuencia CU: Gestionar Calificación, Sección B: Ejercicios de tablas (Ver anexo 38)

Diagrama de secuencia CU: Gestionar Calificación, Sección C: Ejercicios de Checkbox (Ver anexo 39)

Diagrama de Secuencia CU: Gestionar Calificación, Sección D: Ejercicios de completamiento (Ver anexo 40)

Diagrama de secuencia CU: Gestionar Calificación, Sección E: Ejercicios de enlazar columnas (Ver anexo 41)

Diagrama de secuencia CU: Gestionar Ayuda. (Ver anexo 42)

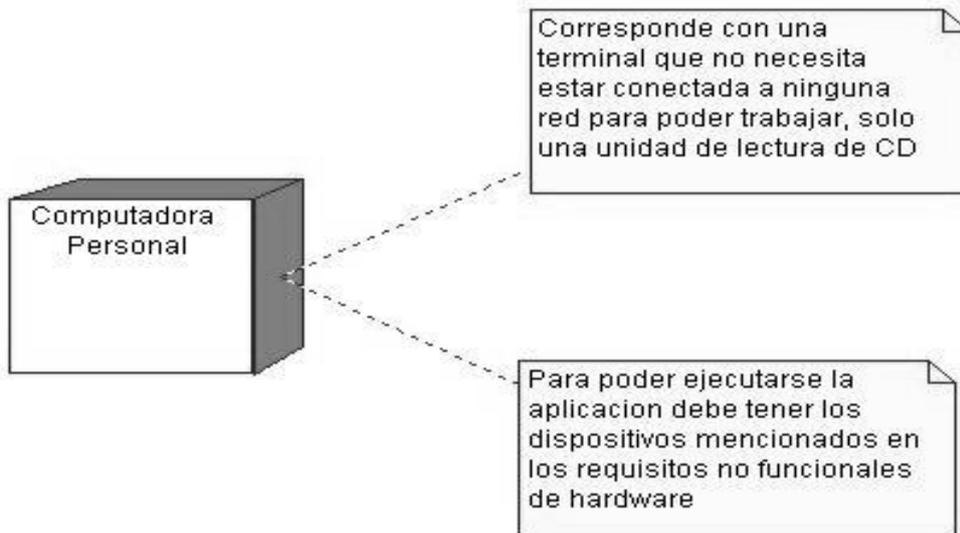
Diagrama de secuencia CU: Controlar Salida. (Ver anexo 43)

3.5 Diagrama de despliegue.

El modelo de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los links de comunicación entre ellos, y las instancias de los componentes y objetos que residen en ellos.

También se utiliza para visualizar la distribución de los componentes de software en los nodos físicos. A continuación se presenta el diagrama de despliegue correspondiente a la aplicación:

Figura 25. Diagrama de despliegue



3.6 Modelo de Implementación

El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes.

3.6.1 Diagrama de Componentes

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación.

Se utilizan para modelar la vista estática de un sistema y expresan la estructura de alto nivel del modelo de implementación. Muestran la organización y las dependencias lógicas entre un conjunto de componentes software, sean estos componentes de código fuente, librerías, binarios o ejecutables. A continuación se presenta el diagrama de componentes correspondiente a la aplicación:

Figura 26. Diagrama de Componentes General

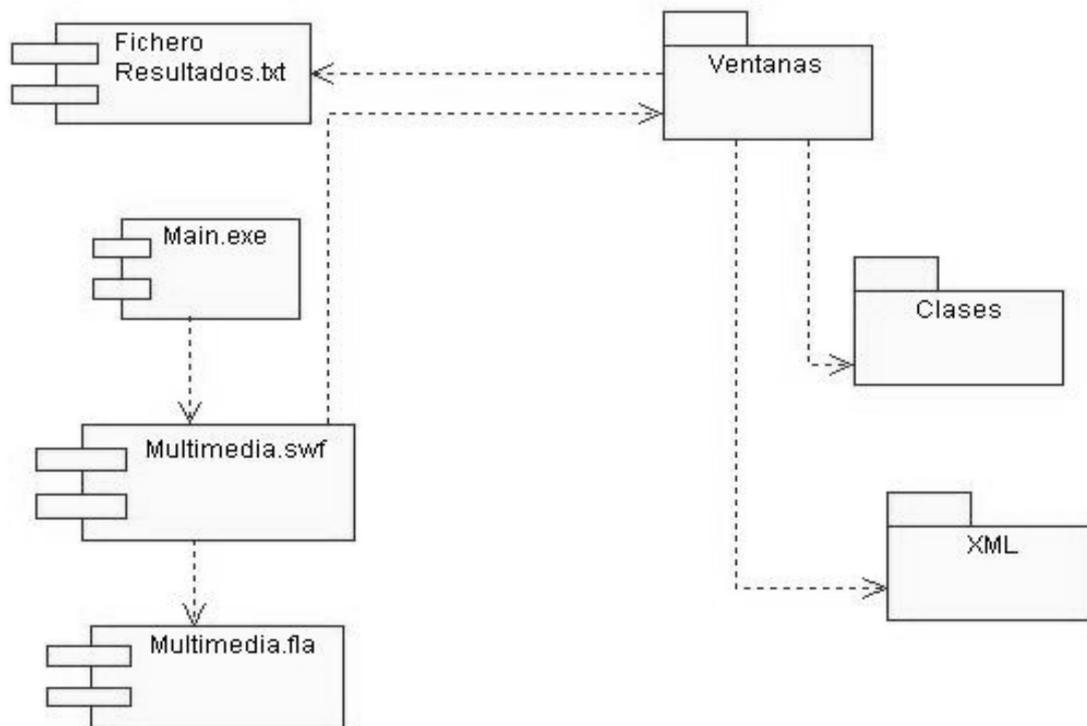


Figura 28. Diagrama de Componentes del paquete XML

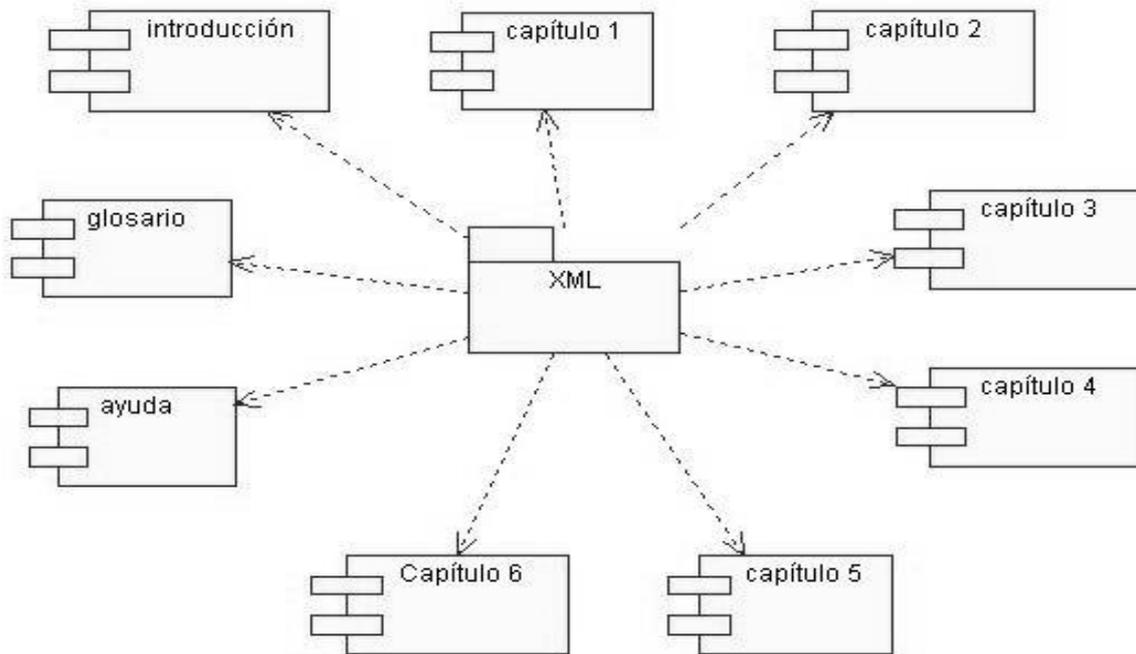
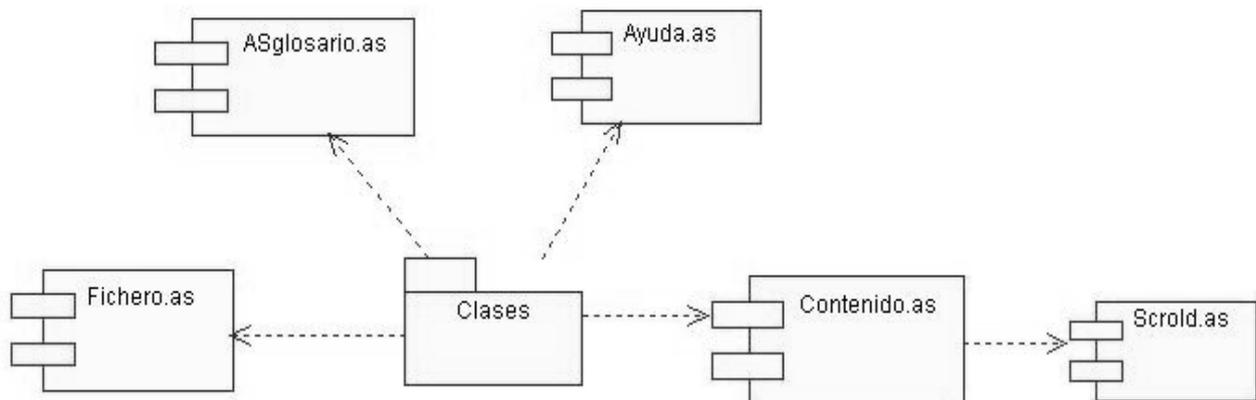


Figura 29. Diagrama de Componentes del paquete Clases



3.7 Descripción de archivos XML.

La aplicación cuenta con 9 archivos XML, los cuales son:

3.7.1 Introducción

Contiene toda la información referente a la introducción la de la aplicación con tecnología multimedia. La estructura básica de este archivo es la siguiente:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<info> [p class='e1'] Texto de la introducción [/p] </info>.
```

3.7.2 Capítulo 1

Contiene toda la información referente al capítulo 1 de la aplicación que trata sobre el sistema tegumentario del pie. La estructura básica de este archivo es la siguiente:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<info> [p class='e1'] Texto del Capítulo 1.[/p]</info>
```

3.7.3 Capítulo 2

Contiene toda la información referente al capítulo 2 de la aplicación que trata sobre el sistema óseo del pie. La estructura básica de este archivo es la siguiente:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<info> [p class='e1'] Texto del Capítulo 2.[/p]</info>
```

3.7.4 Capítulo 3

Este archivo contiene la información referente al capítulo 3 de la aplicación que trata sobre el sistema articular del pie. La estructura básica de este archivo es la siguiente:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<info> [p class='e1'] Texto del Capítulo 3.[/p]</info>
```

3.7.5 Capítulo 4

Este archivo contiene la información referente al capítulo 4 de la aplicación que trata sobre el sistema muscular del pie. La estructura básica de este archivo es la siguiente:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<info> [p class='e1'] Texto del Capítulo 4.[/p]</info>
```

3.7.6 Capítulo 5

Este archivo contiene la información referente al capítulo 5 de la aplicación que trata sobre el sistema circulatorio del pie. La estructura básica de este archivo es la siguiente:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<info> [p class='e1'] Texto del Capítulo 5.[/p]</info>
```

3.7.7 Capítulo 6

Este archivo contiene la información referente al capítulo 6 de la aplicación que trata sobre el sistema nervioso del pie. La estructura básica de este archivo es la siguiente:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<info> [p class='e1'] Texto del Capítulo 6.[/p]</info>
```

3.7.8 Ayuda

Este archivo contiene la información referente a la ayuda de la aplicación. La estructura básica de este archivo es la siguiente:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<ayuda> [p class='e6']
<ejercicio id="1"> Texto 1. </ejercicio>
<ejercicio id="2"> Texto 2. </ejercicio>
<ejercicio id="3"> Texto 3. </ejercicio>
<ejercicio id="4"> Texto 4. </ejercicio>
<ejercicio id="5"> Texto 5. </ejercicio>
<ejercicio id="6"> Texto 6. </ejercicio>
<ejercicio id="7"> Texto 7. </ejercicio>
[/p]
</ayuda>
```

3.7.9 Glosario

Este archivo contiene la información referente al glosario de la aplicación. La estructura básica de este archivo es la siguiente:

```
<glosario>[p class='e4']
<palabra id="Palabra ">Significado de la palabra .</palabra>
```

*

*

*

[/p]

</glosario>

3.8 Conclusiones parciales

En este capítulo se confeccionaron los artefactos correspondientes a los flujos de trabajo de diseño e implementación. Mediante los diagramas de presentación confeccionados se mostró cada uno de los elementos de la aplicación. Los diagramas de jerarquía de medias permitieron mostrar cada una de las medias, ya sea continua o discreta, que contienen los diferentes escenarios del producto. Los diagramas de clases y los diagramas de secuencia, permiten visualizar cada una de las clases que tendrá la aplicación, además de representar los métodos y atributos que estas poseen. Se confeccionó también el diagrama de componentes, mostrando los ficheros y clases, así como sus dependencias. Por último se describieron los archivos XML que tiene la aplicación.

Capítulo 4: Estudio de la Factibilidad.

Introducción.

Para la realización de un proyecto es importante estimar el esfuerzo humano, el tiempo de desarrollo que se requiere para la ejecución del mismo y también su costo. En este capítulo se realizará el estudio de factibilidad del sistema utilizando el método Análisis de Puntos de Casos de Uso. Este método permite documentar los requerimientos de un sistema en términos de Actores y Casos de Uso. Posteriormente se estimará el esfuerzo en hombres/mes necesarios para llevar a cabo la realización del producto, así como el tiempo de desarrollo en meses requerido para el sistema. Además se evaluará el costo total del proyecto y por último se expondrán los beneficios tangibles e intangibles que traerá implícitamente la realización de la aplicación con tecnología multimedia “Curso de Podología para la Salud”.

4.1 Planificación.

El objetivo de la planificación es establecer planes razonables para desarrollar la Ingeniería de Software y manejar los cambios de los proyectos de Software.

El proceso de gestión de proyecto de software comienza con un conjunto de actividades que, globalmente, se denominan planificación del proyecto. La primera de estas actividades es la de estimar los resultados del proyecto y los valores de costo, tiempo y recursos requeridos. Existen técnicas útiles para la estimación como son Análisis de Puntos de Función y COCOMO II y Análisis de Puntos de Casos de Uso.

La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner de Objectory AB y posteriormente refinado por muchos otros autores. 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. (ISW 2007)

A continuación, se detallan los pasos a seguir para la aplicación de éste método.

4.2 Cálculo de puntos de Casos de Usos sin Ajustar

El primer paso para la estimación consiste en el cálculo de los Puntos de Casos de Uso sin ajustar. Este valor, se calcula a partir de la siguiente ecuación:

$$\boxed{UUCP = UAW + UUCW}$$

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.

4.2.1. Factor de Peso de los Actores sin ajustar (UAW)

Este valor se calcula mediante un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema. Los criterios se muestran en la siguiente tabla:

Tipo de Actor	Descripción	Cantidad de Actores	Factor de Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación(API, Application Programming Interface)	0	1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	0	2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	1	3

Tabla 10.Criterio para calcular el Factor de Peso de los Actores sin ajustar.

Luego, el factor de Peso de los Actores sin ajustar, resulta:

$$UAW = \sum \text{Cantidad de actores}_i * \text{Factor de peso}_i$$

$$UAW = 0 * 1 + 0 * 2 + 1 * 3 = 3$$

Entonces:

$$\boxed{UAW = 3}$$

4.2.2. Factor de Peso de los Casos de Uso sin ajustar (UUCW)

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. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica, es decir, está representada por uno o más pasos del flujo de eventos principal del caso de uso, pudiendo existir más de una transacción dentro del mismo caso de uso. Los criterios se muestran en la siguiente tabla:

Tipo de Caso de Uso	Descripción	Cantidad de Casos de Uso	Factor de Peso
Simple	El caso de uso contiene de 1 a 3 transacciones	6	5
Medio	El caso de uso contiene de 4 a 7 transacciones	0	10
Complejo	El caso de uso contiene más de 8 transacciones	0	15

Tabla11. Criterio para calcular el Factor de Peso de los Casos de Uso sin ajustar.

Luego, el Factor de Peso de los Casos de Uso sin ajustar, resulta:

$$UUCW = \sum \text{Cantidad de Casos de Uso}_i * \text{Factor de peso}_i$$

$$UUCW = 6 * 5 + 0 * 10 + 0 * 15$$

Entonces:

$$\boxed{UUCW = 30}$$

Finalmente los Puntos de Casos de Uso sin ajustar resultan:

$$\boxed{UUCP = UAW + UUCW = 3 + 30 = 33}$$

4.3 Cálculo de Puntos de Casos de Uso ajustados

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

$$\text{UCP} = \text{UUCP} \times \text{TCF} \times \text{EF}$$

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

4.3.1 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. En la siguiente tabla se muestra el significado y el peso de cada uno de éstos factores:

Tabla 12. Factor de complejidad técnica

Factor	Descripción	Valor	Peso
T1	Sistema distribuido	0	2
T2	Objetivos de performance o tiempo de respuesta	4	1
T3	Eficiencia del usuario final	4	1
T4	Procesamiento interno complejo	2	1
T5	El código debe ser reutilizable	5	1
T6	Facilidad de instalación	5	0.5
T7	Facilidad de uso	5	0.5
T8	Portabilidad	3	2
T9	Facilidad de cambio	3	1
T10	Concurrencia	2	1
T11	Incluye objetivos especiales de seguridad	1	1
T12	Provee acceso directo a terceras partes	0	1
T13	Se requieren facilidades especiales de entrenamiento a usuarios	1	1

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

$$\text{TCF} = 0.6 + 0.01 * \Sigma (\text{Peso}i * \text{Valor asignado}i)$$

Luego:

$$\text{TCF} = 0.6 + 0.01 * (0 + 4 + 4 + 2 + 5 + 2.5 + 2.5 + 6 + 3 + 2 + 1 + 0 + 1)$$

$$\text{TCF} = 0.6 + 0.01 * 33$$

Entonces:

$$\text{TCF} = 0.93$$

4.3.2 Factor ambiente (EF)

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente. 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 la siguiente tabla se muestra el significado y el peso de cada uno de éstos factores:

Factor	Descripción	Valor	Peso
E1	Familiaridad con el modelo de proyecto utilizado.	4	1.5
E2	Experiencia en la aplicación.	5	0.5
E3	Eficiencia en la orientación a Objetos	4	1
E4	Capacidad del Analista Líder.	3	0.5
E5	Motivación.	5	1
E6	Estabilidad de los requerimientos.	3	2
E7	Personal part.-time	3	-1
E8	Dificultad del lenguaje de programación.	3	-1

Tabla 13. Factor Ambiente

El Factor de ambiente se calcula mediante la siguiente ecuación:

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

Luego:

$$EF = 1.4 - 0.03 * (6 + 2.5 + 4 + 1.5 + 5 + 6 - 3 - 3)$$

$$EF = 1.4 - 0.03 * (19)$$

$$EF = 1.4 - 0.57$$

Entonces:

$$EF = 0.83$$

Finalmente, los Puntos de Casos de Uso ajustados resultan:

$$UCP = UUCP * TCF * EF = 33 * 0.93 * 0.83 = 25.47$$

$$UCP = 25.47$$

4.4 De los puntos de casos de uso a la estimación del esfuerzo

Karner originalmente sugirió que cada Punto de Casos de Uso requiere 20 horas-hombre. Posteriormente, surgieron otros refinamientos que proponen una granularidad algo más fina, según el siguiente criterio:

Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores E1 a E6.

Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.

- Si el total es 2 ó menos, se utiliza el factor de conversión 20 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 20 horas-hombre.
- Si el total es 3 ó 4, se utiliza el factor de conversión 28 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 28 horas-hombre.

CAPÍTULO 4: ESTUDIO DE LA FACTIBILIDAD

- Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

El esfuerzo en horas-hombre viene dado por:

$$E = UCP \times CF$$

Donde:

E: esfuerzo estimado en horas/hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: Factor de conversión.

Luego:

CF= 20 horas/hombre

E = 25.47 * 20 horas/hombre = 509.4 horas/hombre

Entonces:

$$E = 509.4 \text{ horas/hombre}$$

Este método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los casos de uso.

Para una estimación más completa de la duración total del proyecto, hay que agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software.

Para ello se puede tener en cuenta el siguiente criterio, que estadísticamente se considera aceptable. El criterio plantea la distribución del esfuerzo entre las diferentes actividades de un proyecto, según la siguiente aproximación:

Actividad	Porcentaje	Horas /Hombre
Análisis	10 %	127.35
Diseño	20 %	254.7
Programación	40 %	509.4
Pruebas	15 %	191.025
Sobrecarga(otras actividades)	15 %	191.025
Total	100 %	1273.5

Tabla 14. Distribución del esfuerzo entre las diferentes actividades del desarrollo de SW.

4.5 Para convertir el esfuerzo a hombres-mes

Como la jornada laboral de un día de trabajo es de 8 horas y en un mes se trabaja un aproximado de 24 días, entonces una persona en 1 mes trabaja 192 horas, por tanto:

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

Quedaría, $E_{total} = 1273.5 \text{ horas-Hombres} / 192 \text{ horas-mes} = 6.63 \text{ mes-hombre} = 7 \text{ mes-hombre.}$

4.6 Tiempo de desarrollo

Si en el proyecto trabajan dos hombres entonces el tiempo de desarrollo es:

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

$\text{Tiempo de desarrollo} = 7 \text{ mes-hombres} / 2 \text{ hombres} = 3.5 \text{ meses} = 4 \text{ meses}$

El tiempo a emplear para el desarrollo de la aplicación es de 4 meses.

4.8 Salario

Para determinar el salario mensual se tiene en cuenta que los desarrolladores del sistema pueden ser ingenieros recién graduados, por lo que se toma como salario mensual: \$225.

4.9 Costo

Como anteriormente se definió el salario promedio de un ingeniero y son dos personas para desarrollar el proyecto, entonces para hallar el costo total, sería:

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

$Ct = \$225 * 2 * 4$

$Ct = \$ 1800 \text{ MN}$

$Ct = \$ 72 \text{ CUC}$

4.10 Beneficios tangibles e intangibles.

4.10.1 Tangibles.

Teniendo en cuenta que la aplicación con tecnología multimedia Curso de Podología para la Salud no es un producto desarrollado para la comercialización, sino que surge como una necesidad social debido a la creciente informatización de nuestro país, no es válido mencionar beneficios económicos. Se puede decir que el costo por desarrollar la aplicación es de \$ 1800 MN (moneda nacional) y 72 CUC (convertible), el cual es perfectamente reparable si en un futuro se comercializara dicha aplicación.

4.10.2 Intangibles

Como beneficios intangibles asociados al desarrollo de la multimedia educativa se mencionan los siguientes:

- Centralización de la información disponible sobre la Podología.
- Disminución de la cantidad de horas presenciales del profesor frente a los alumnos.
- Posibilidad de registros evaluativos automatizados para los profesores.
- Aumento de la calidad en la enseñanza de la Podología.
- Consolidación de los conocimientos adquiridos en clases para los estudiantes.
- Posibilidad de individualización de la enseñanza de los educandos a partir de sus resultados y de esta forma mejorar sus resultados docentes.
- Aumento de la calidad de la presentación de los contenidos a los alumnos.

4.11 Análisis costo-beneficio.

Dadas las características del software se puede asegurar que este no requiere demasiado tiempo de desarrollo ni grandes gastos de recursos ya que será de tamaño pequeño, con archivos XML que contienen toda la información referente al sistema.

Los ficheros que genera la aplicación pueden ser extraídos por el profesor sin problema ya que los mismos tienen acceso rápido y no se necesita ningún gestor de base de datos.

La navegabilidad y entorno de la aplicación cuenta con un diseño muy intuitivo que le permite al usuario navegar sin perderse en la aplicación y la interfaz está diseñada cuidadosamente y resulta agradable al entorno del mismo.

Por tanto, se considera que es factible desarrollar una aplicación con tecnología multimedia para centralizar la información existente sobre la Podología.

4.12 Conclusiones parciales.

Después de haber concluido el estudio de factibilidad se puntualizan los costos a incurrir, los recursos humanos implicados, el tiempo de desarrollo y los beneficios tangibles e intangibles que aporta la terminación del producto “Curso de Podología para la Salud”. Con la realización de este capítulo se expresa la ventaja que implica la realización de este producto.

Los resultados obtenidos al concluir los cálculos y estimaciones realizadas en este capítulo, se muestran en la siguiente tabla:

Parámetros	Valores
Esfuerzo	7 mes-hombre
Tiempo de desarrollo	4 meses
Cantidad de hombres	2 hombres
Salario	\$ 225
Costo total (MN)	\$ 1800 MN
Costo total (CUC)	\$ 72 CUC

Tabla 15.Conclusiones Parciales

Conclusiones generales

Durante el desarrollo de la aplicación con tecnología multimedia fueron cumplidos una serie de objetivos específicos y tareas trazadas en el inicio de la investigación que conllevaron al cumplimiento del objetivo general.

Se concluye que:

- Como resultado de realizar el análisis, diseño e implementación del producto se obtuvo una aplicación con tecnología multimedia sobre la Podología para la salud.
- Se creó un documento donde se recogió todo el proceso investigativo del desarrollo de la aplicación, incluido el estudio de la factibilidad de la misma.

De forma general se logró de forma factible y satisfactoria el desarrollo de un software educativo con tecnología multimedia que contribuirá a la centralización de la información sobre la Podología, además se cumplió con los requerimientos del cliente y los objetivos propuestos inicialmente.

Recomendaciones

La aplicación con tecnología multimedia cumple con los objetivos y requisitos planteados inicialmente, no obstante se pueden realizar algunas mejoras que contribuyan al perfeccionamiento del producto. Se recomienda que:

- La aplicación sea valorada por especialistas de software educativo en la UCI y expertos en la especialidad de Podología.
- Se realicen pruebas piloto en varias instituciones educacionales para que se puedan valorar sus resultados y tomar decisiones en torno a este.
- Se le dé la posibilidad al usuario de imprimir el contenido y los ejercicios para que este pueda consultarlo en cualquier momento u ofrecérselo a alguien que no halla trabajado en la aplicación.
- Profundizar en el estudio de las aplicaciones multimedia para implementar un producto que permita ver a los profesores un reporte de los ejercicios realizados por el estudiante sin necesidad de visualizar el fichero evaluativo.
- Se continúe el estudio de la Podología y las aplicaciones multimedia con el objetivo de encontrar nuevas funcionalidades para refinar e implementar un producto más completo.

Referencias Bibliográficas

ALMENARA, J. C.; C. B. REGAÑA, *et al.* CREACIÓN DE APLICACIONES MULTIMEDIAS EDUCATIVAS CON EL PROGRAMA: TOOLBOOK., 2008]. Disponible en: <http://tecnologiaedu.us.es/edutec/paginas/3.html>

AULACLIC.ES. *Curso de Flash 8*, 2006. [2008]. Disponible en: http://www.aulaclie.es/flash8/t_1_1.htm

AULAFACIL.COM. VENTAJAS, DESVENTAJAS Y DESCRIPCIÓN DE EQUIPOS MULTIMEDIA

2000. [Disponible en: <http://www.aulafacil.com/cursosenviados/cursomultimediaaprendizaje/Lecc-4.htm>

BLANCO, R. G. and S. P. TOBALINA. *Introducción a Rational Rose.* , 2008]. Disponible en: <http://www.lsi.upc.edu/~es-e/web/documents/lab/0304Q2/lessons/lese-2/LESE-2%20-%20Introduccion%20a%20Rational%20Rose.ppt>

CATALÁ, S. D.; C. M. S. REYES, *et al.* *Ingeniería para productos con tecnología multimedia*, 2007. [Disponible en: <http://uciencia.uci.cu/?q=node/405&ev=II%20Taller%20de%20Software%20Educativo%20y%20Multimedia>

DEWEY.UAB.ES. *MULTIMEDIA EDUCATIVO: CLASIFICACIÓN, FUNCIONES, VENTAJAS E INCONVENIENTES*, 2004. [Disponible en: <http://dewey.uab.es/P/MARQUES/funcion.htm>

DIAZ, C. C. *La tecnología multimedia*, 1994 [Disponible en: <http://iteso.mx/~carlosc/pagina/documentos/multidef.htm#caracter>

FORO3K.COM. *SONY VEGAS 7e PROFESIONAL EDITION ESPAÑOL*, 2008. [Disponible en: <http://www.foro3k.com/pirate-zone/75127-sony-vega-7e-profesional-en-espanol.html>

GIORGIS, R. S. D.; W. P. MUÑOZ, *et al.* *Propuesta de un modelo navegacional para el desarrollo de aplicaciones basadas en OOADM.* Disponible en: http://www.inf.ucv.cl/~rsoto/papersPUCV/Propuesta_de_un_modelo_navegacional.pdf

GUERVOS, J. J. M. *Introducción al lenguaje XML*, 2004. [Disponible en: <http://geneura.ugr.es/~jmerelo/xml/>

GUTIÉRREZ, J. A. D. *Acción Script 2.0 + Flash Develop 2*, 2007.

INTERNATIONAL, V. P. *Build Quality Applications Faster, Better and Cheaper. Introducción a los Sistemas y Herramientas CASE.* . Disponible en:

ISW, D. C. D. *Planificación y Estimación de Proyectos*, 2007. [Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=9007>

KRISTIAN BESLEY, S. B., AMANDA FARR. *Macromedia Flash MX*. Disponible en: <http://bibliodoc.uci.cu/pdf/reg02096.pdf>

REFERENCIAS BIBLIOGRÁFICAS

LAMARCA LAPUENTE, M. J. *Modelo RMM*, 2005.

MA. VICTORIA AGUIAR PERERA, J. C. F., JUAN JIMENEZ BETANCOURT. *Evaluación de programas multimedia*. Disponible en:
<http://www.uib.es/depart/gte/edutec/edutec01/edutec/comunic/TSE48.html>

MARTÍNEZ, J. M.; J. R. HILERA, *et al.* *Orientación a Objetos en la Documentación Hipermedia*, 1998. [Disponible en: <http://www.ati.es/gt/LATIGOO/OOp96/Ponen6/atio6p06.html>]

MELLADO, D. *Introducción a la Ingeniería de Requisitos de Seguridad*, 2008. [Disponible en: [http://www.calidaddelsoftware.com/eventos/SoloRqstos2006/Solo%20Requisitos%202006/ReqSeg\(Mellado\)SOLORQ.pdf](http://www.calidaddelsoftware.com/eventos/SoloRqstos2006/Solo%20Requisitos%202006/ReqSeg(Mellado)SOLORQ.pdf)]

ORALLO, E. H. *El Lenguaje Unificado de Modelado (UML)*, 2001. [2008]. Disponible en: <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>

PALERMO, U. D. *Ciclos de Vida de proyectos*. Disponible en: <http://writer.zoho.com/public/faguero/Adm---Ciclo-vida-RUP>

PROGRAMACION.NET. *Introducción a ActionScript* 1999. [Disponible en: <http://www.programacion.net/tutorial/actionscript/1/>]

RAVIOLI, P. *Lenguaje de programación para paginas web HTML*, 2007. [Disponible en: <http://www.monografias.com/trabajos7/html/html.shtml>]

SANCHEZ, M. A. M. *Metodologías De Desarrollo De Software*, 2004. [2008]. Disponible en: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html

SAUER, S. and G. ENGELS. *Extending UML for Modeling of Multimedia Applications*, 2008]. Disponible en: <http://www.itec.uni-klu.ac.at/%7Eharald/proseminar02/sauer1.pdf>

SERNA, J. M. R. A. Y. M. C. D. L. *Nuevas Tecnologías de la Información y la Comunicación aplicadas a la Educación*, 1999. [Disponible en: <http://www.dionisiodiaz.com/multimensenanza/multimediaensenanza.html#03>]

TELEFORMACION.UCI.CU. *Fase de inicio. Modelo de negocio*, 2007. [Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=6680>]

TORRES, M. A. G. *Manual de Macromedia director*. Disponible en: <http://www.programatium.com/manuales/director/director1.htm>

VALDÉS, S. *La constitución Volumen II*. La Habana, Cuba, Universidad de las Ciencias Informáticas, 2006. p.

Bibliografía

Aulaclíc.es. Aulaclíc.es. <http://www.aulaclíc.es>. [Online] marzo 2006.

http://www.aulaclíc.es/flash8/t_1_1.htm.

Avalos Palomo., Frank David. *Multimedia Educativa para los niños de la Enseñanza Primaria con disgrafía escolar*. Ciudad de la Habana : s.n., 2007.

Candebat Calderón, Karelys and Miranda Cairo, Michel. *Curso Optativo Interactivo Flash Básico*. Ciudad de la Habana : s.n., 2007.

CiberAula.com. CiberAula.com. [En línea] 2006. http://flash.ciberaula.com/articulo/actionscrip_t_xml/.

Corrales Díaz, Carlos. LA TECNOLOGIA MULTIMEDIA. <http://iteso.mx>. [Online] enero 1994. [Cited: abril 12, 2008.] <http://iteso.mx/~carlosc/pagina/documentos/multidef.htm#inicio>.

Der, Henst, Christian Van. Maestros del Web. <http://www.maestrosdelweb.com>. [Online] 1997. [Cited: febrero 20, 2008.] <http://www.maestrosdelweb.com/editorial/flash/>.

Díaz Catalá, Sergio, Souлары Reyes, Celia María and Gálvez George, Greisy. *Ingeniería para productos con tecnología multimedia*. Ciudad de la Habana : s.n., 2007.

Emagister.com. <http://www.emagister.com>. [Online] 1999/2000. [Cited: abril 5, 2008.]

<http://www.emagister.com/xml-lenguaje-marcas-extensible-cursos-317471.htm#programa>.

Issy Camy, Lázaro. *La Biblia de Flash*.

Mendoza Sanchez, María A.; Informatizate. [Online] noviembre 27, 2002. [Cited: febrero 15, 2008.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

Reynox . *Servicios informáticos*. [Online] [Cited: mayo 10, 2008.] <http://www.reynox.com>.

RUMBAUHG, J. *El Proceso Unificado de Desarrollo de Software*, 2000.

RUMBAUHG, J. and I. JACOBSON. *El Lenguaje Unificado de Modelado. Manual de referencia*. Madrid, 2000 p.

Zambrano Rodriguez, Douglas Francisco. Monografías.com. <http://www.monografías.com>. [Online] 1997. [Cited: marzo 24, 2008.] <http://www.monografías.com/trabajos10/mmedia/mmedia.shtml>.

Anexos

Figura 30. Multimedia

Multimedia es una combinación de formas de contenido:



Figura 31. Etapas de la metodología OOHDM.



Figura 1 Las cinco etapas de la metodología OOHDM.

Figura 32. Metodología Extreme Programing

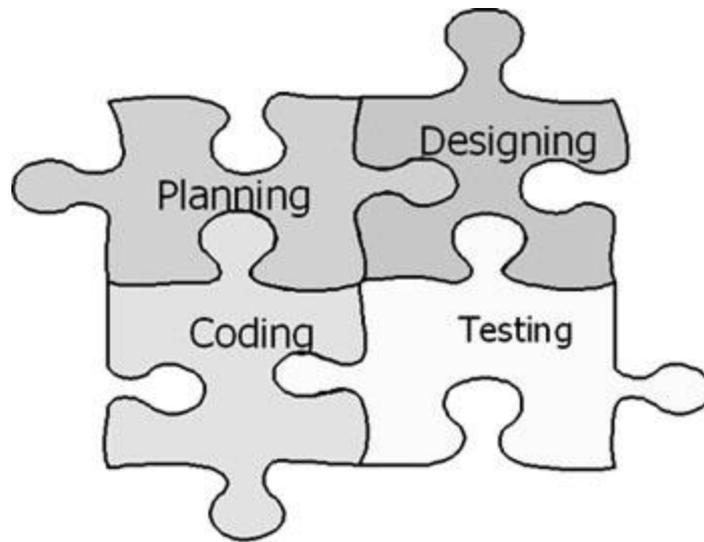


Figura 33. RUP en dos dimensiones

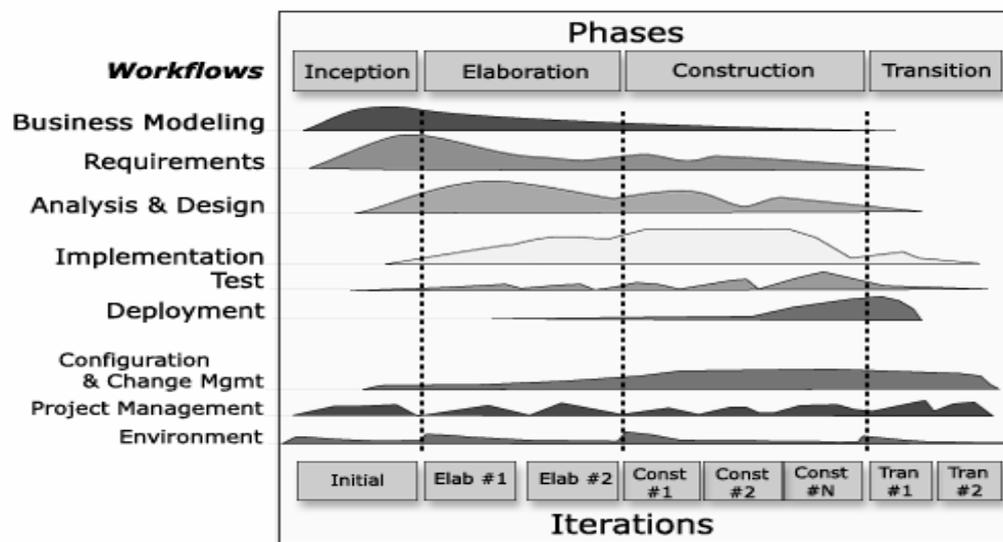


Figura 34. Diagrama de secuencia CU: Autenticar Usuario

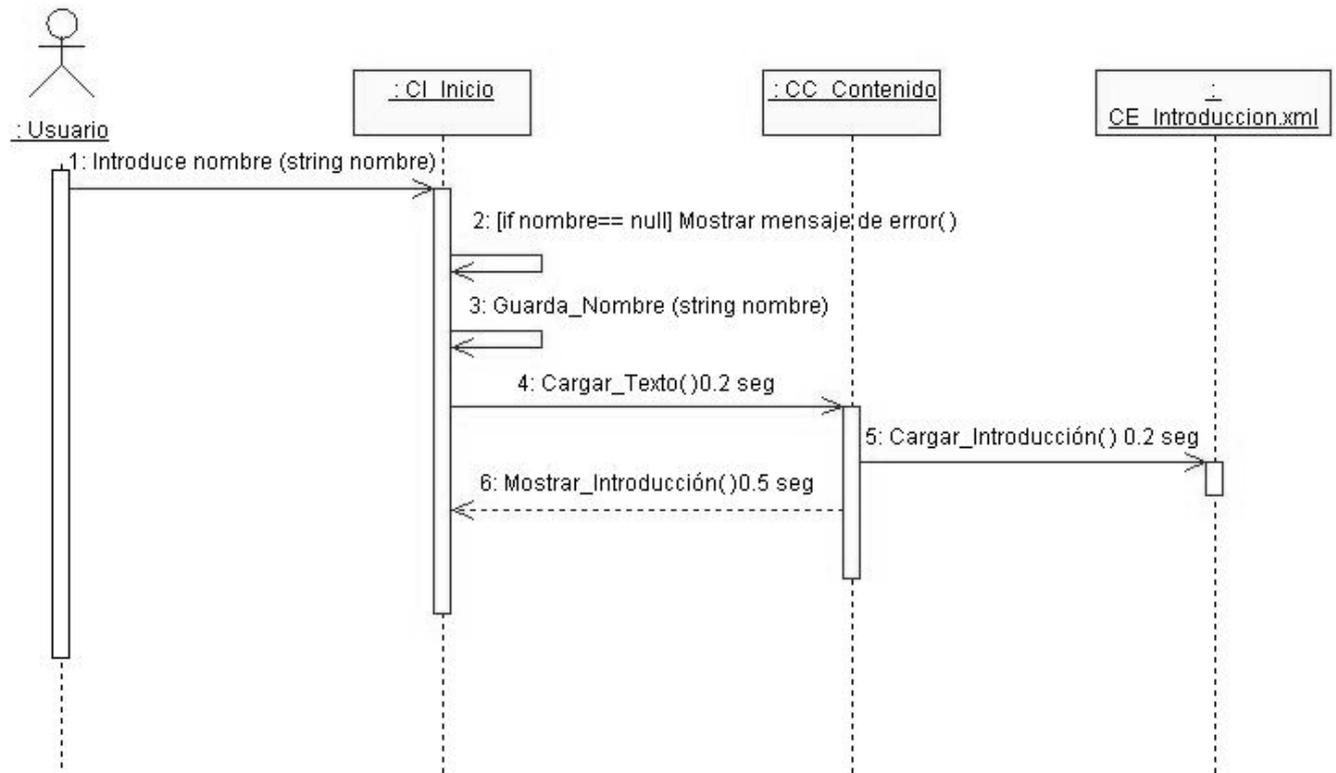


Figura 35. Diagrama de secuencia CU: Cargar Presentación

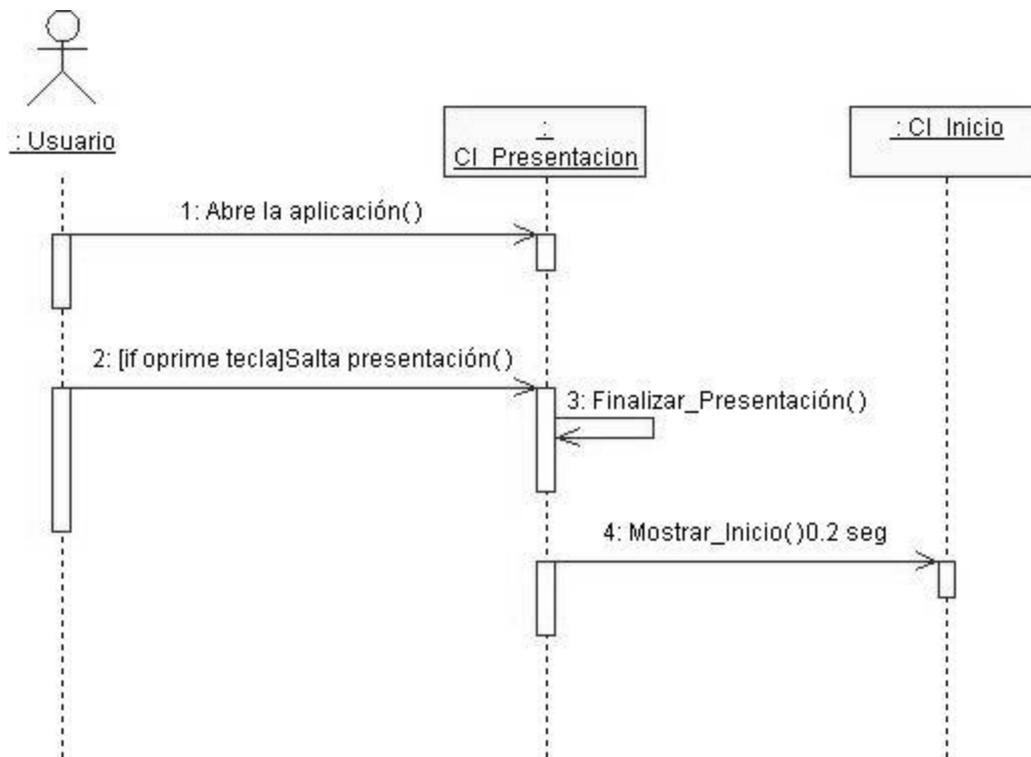


Figura 36. Diagrama de Secuencia CU: Mostrar Contenido

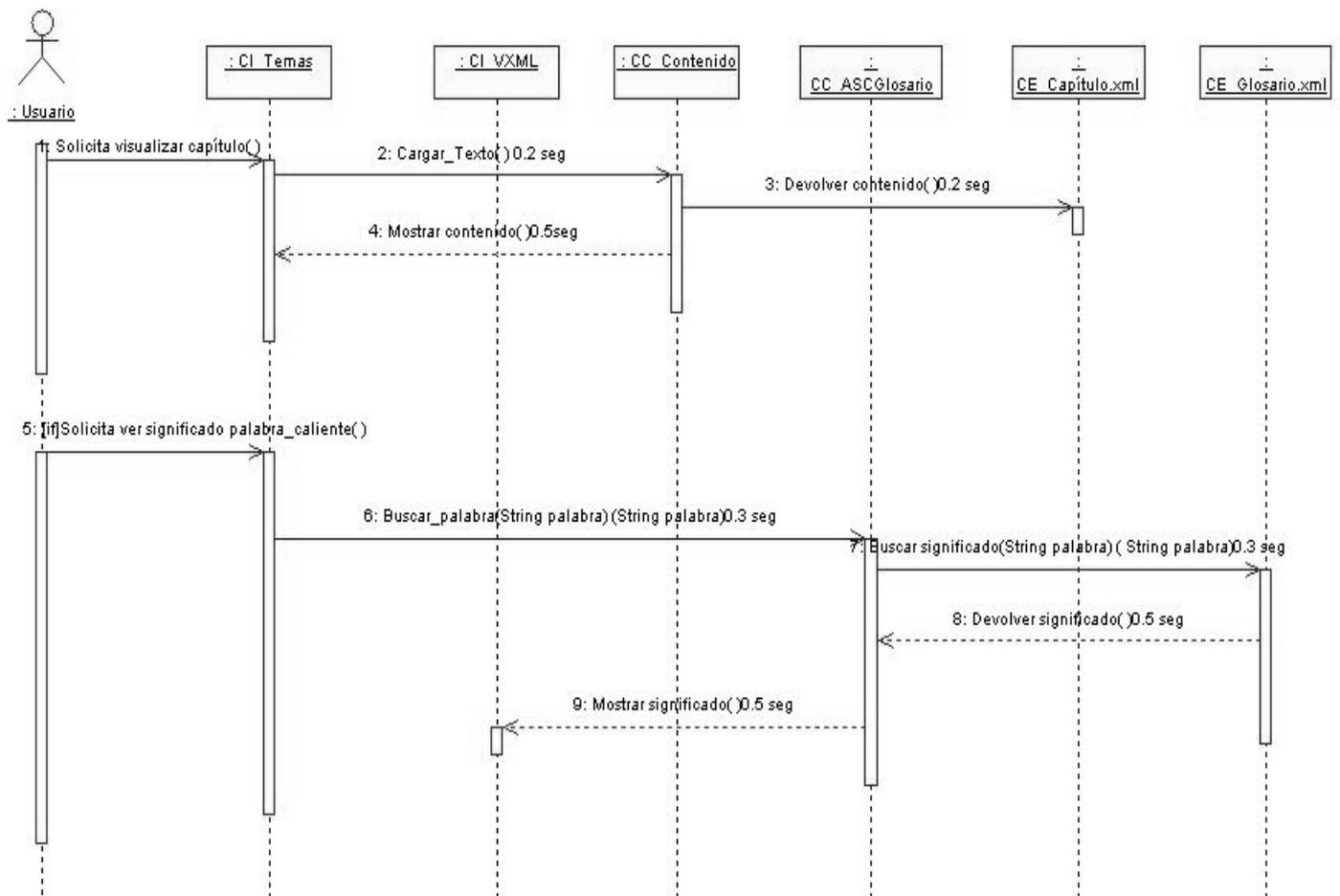


Figura 37. Diagrama de secuencia CU: Gestionar Calificación, Sección A: Ejercicios de Selección (V o F)

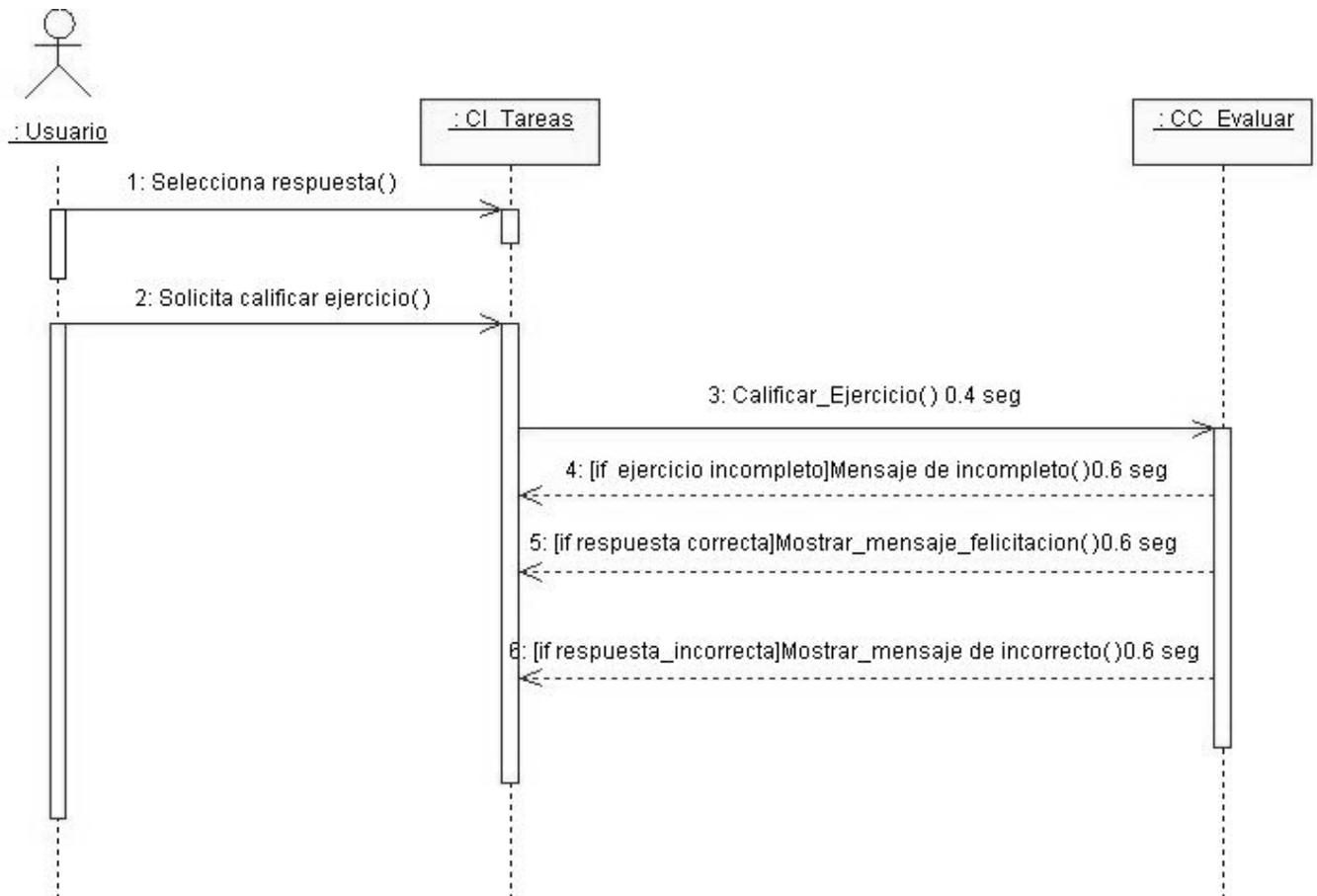


Figure 38. Diagrama de secuencia CU: Gestionar Calificación, Sección B: Ejercicios de tablas

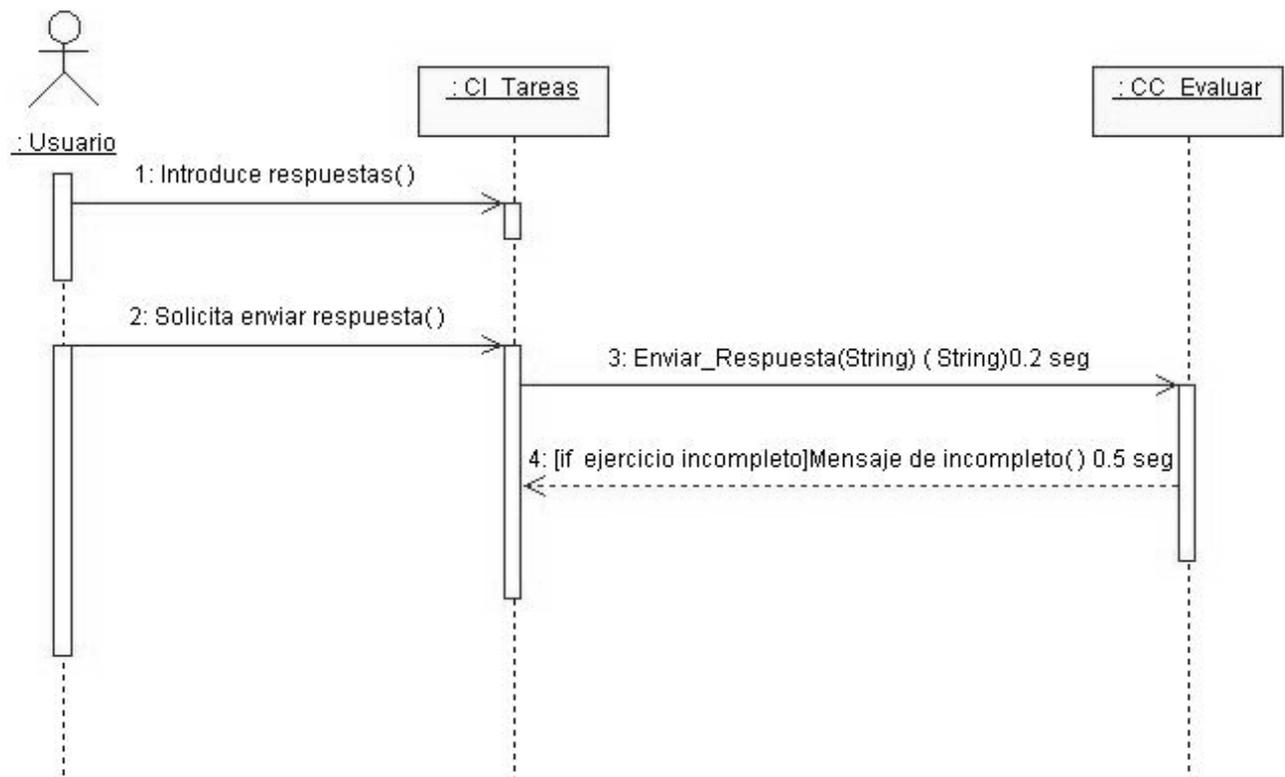


Figura 39. Diagrama de secuencia CU: Gestionar Calificación, Sección C: Ejercicios de checkbox.

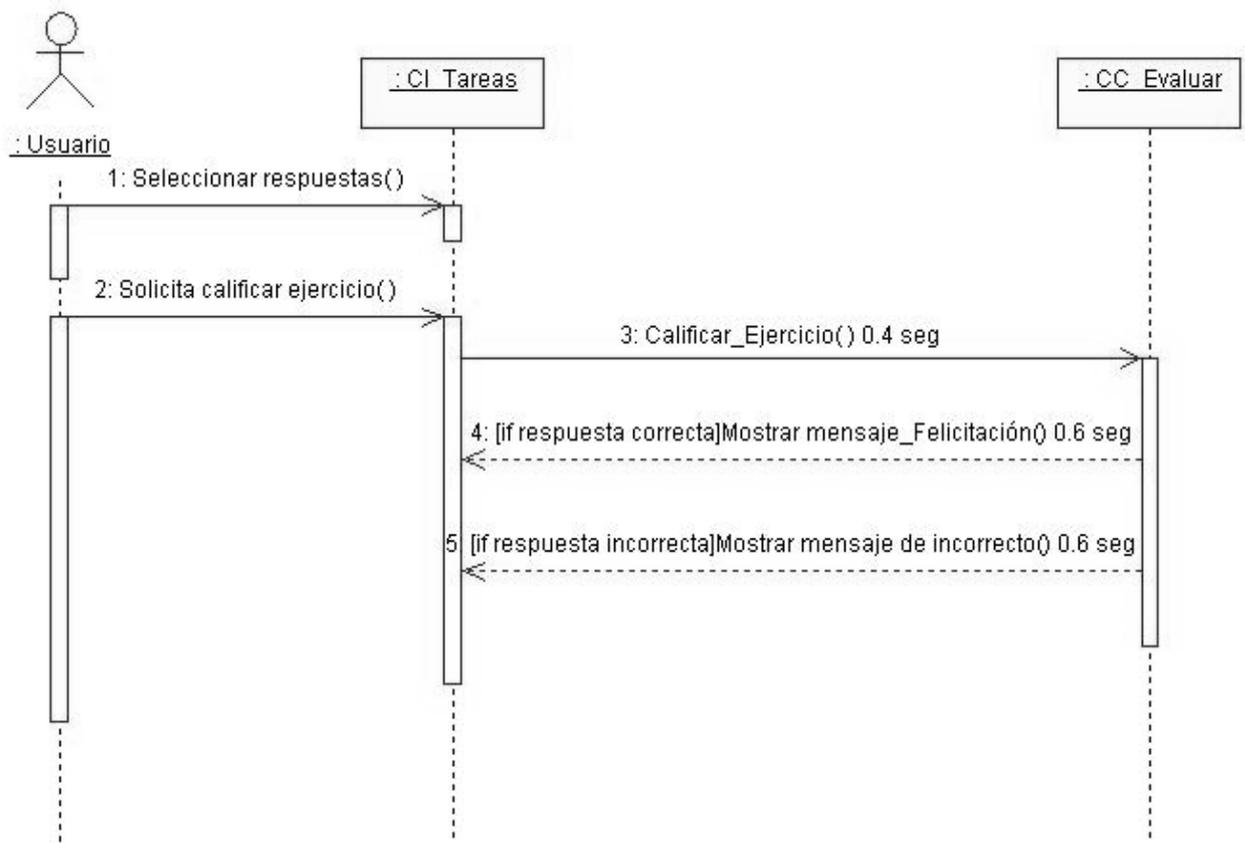


Figura 40. Diagrama de Secuencia CU: Gestionar Calificación, Sección D: Ejercicios de completamiento

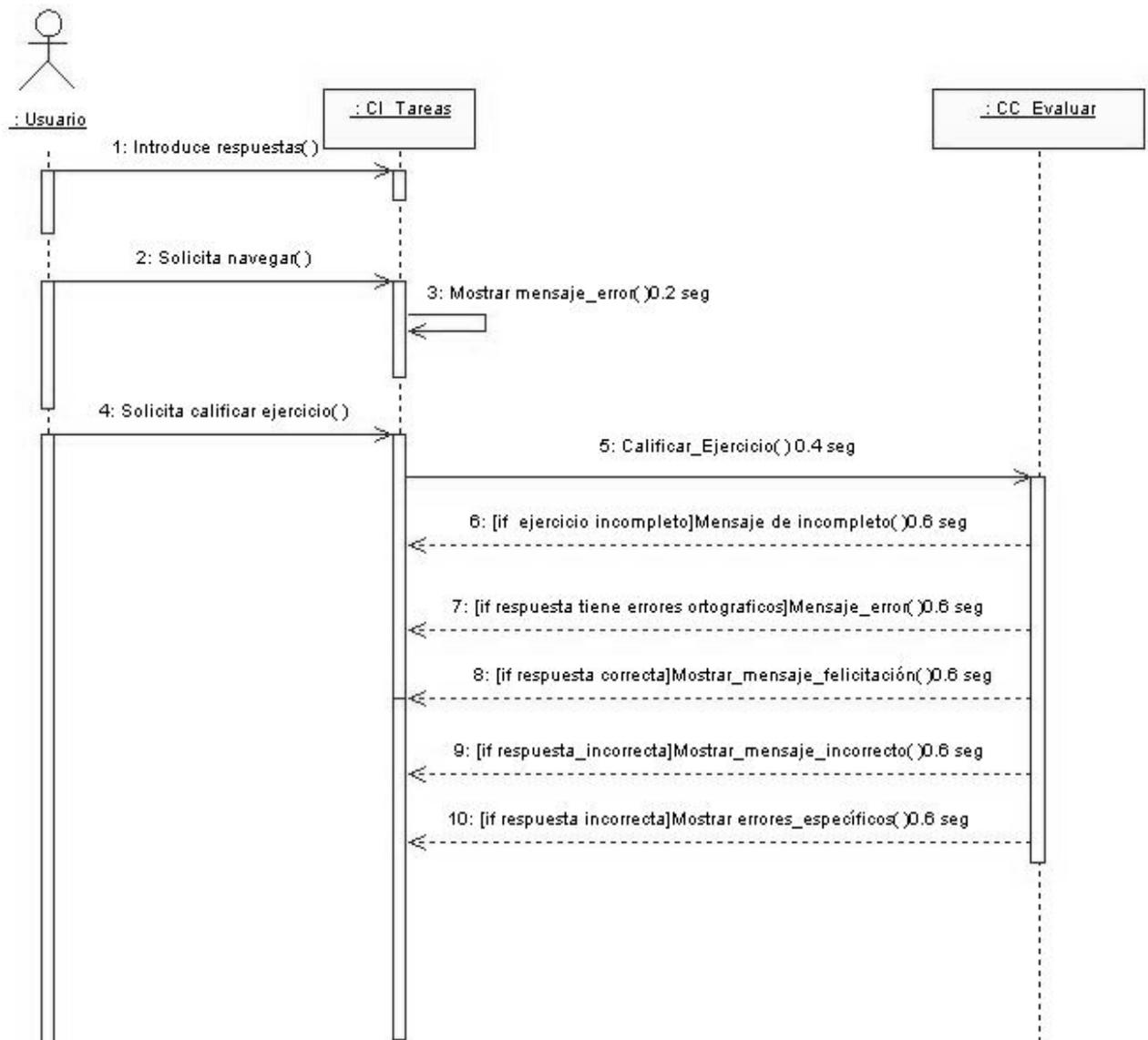


Figura 41. Diagrama de secuencia CU: Gestionar Calificación, Sección E: Ejercicios de enlazar columnas

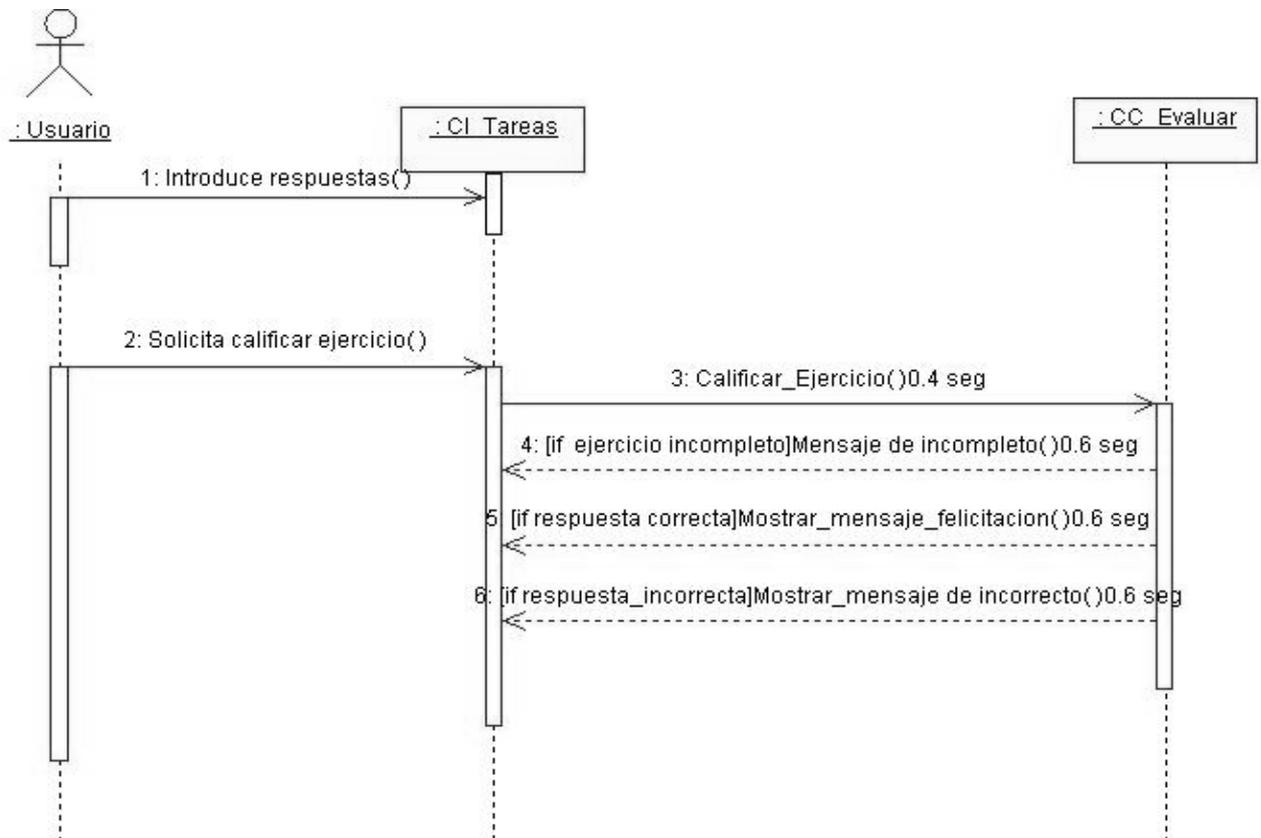


Figura 42. Diagrama de Secuencia CU: Gestionar Ayuda

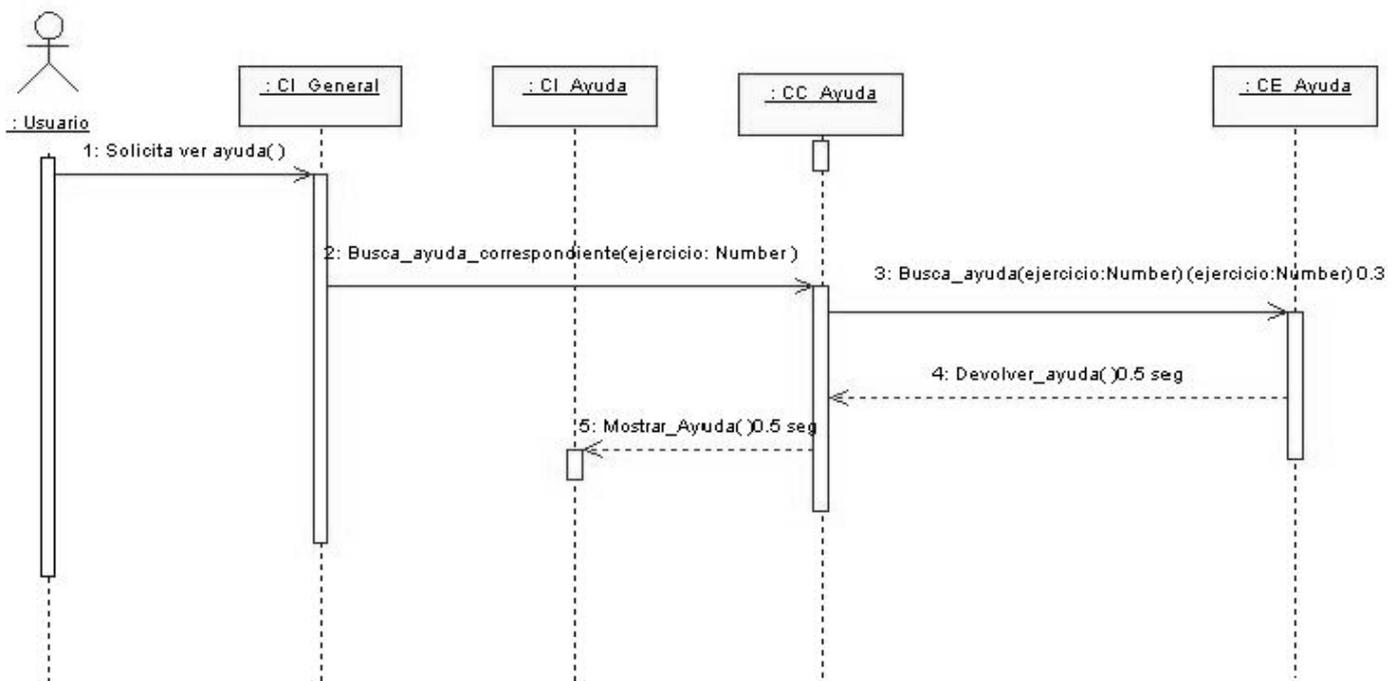
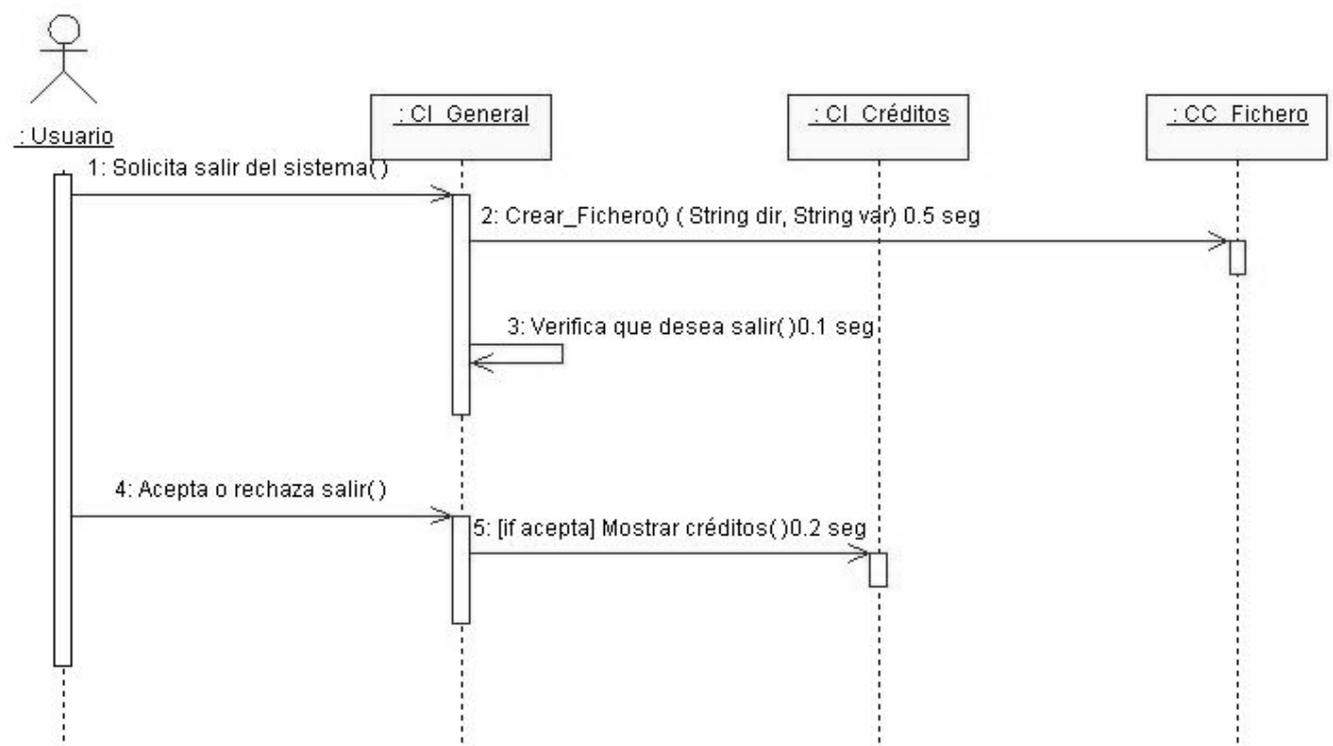


Figura 43. Diagrama de Secuencia CU: Controlar Salida



Glosario de términos

- **Actor:** Abstracción de las entidades externas a un sistema, subsistemas o clases que interactúan directamente con el sistema. Un actor participa en un caso de uso o en conjunto coherente de casos de usos para llevar a cabo un propósito global.
- **Artefacto:** Pieza de información utilizada o producida por un proceso de desarrollo de software como un documento externo o el producto de un trabajo. Un artefacto puede ser un modelo, elementos dentro del modelo, una descripción, o el software.
- **CASE** (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador): Conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases.
- **CC (Clase control):** Coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.
- **CD ROM** (Compact Disk- Read only memory): Denominación de un CD no regrabable que contiene datos para computadoras y también de audio. Las informaciones se leen en una unidad de CD ROM que puede funcionar en la computadora como dispositivo interno externo.
- **CE (Clase entidad):** Son clases que modelan información que posee larga vida y que es a menudo persistente.
- **CI (Clase interfaz):** Son clases que modelan la interacción entre el sistema y sus actores.
- **DVD (Digital Versatile Disc o "Disco Versátil Digital"):** Es un formato de almacenamiento óptico que puede ser usado para guardar datos, incluyendo películas con alta calidad de video y audio. Se asemeja a los CD en cuanto a sus dimensiones físicas, pero están codificados en un formato distinto y a una densidad mucho mayor. A diferencia de los CD, todos los DVD deben guardar los datos utilizando un sistema de archivos denominado UDF (Universal Disk Format).
- **.Fla:** Una extensión de archivo con la cual se puede trabajar en Macromedia Flash, es decir, es el código fuente de una animación.
- **Herramientas de autor:** Software que manejan elementos de media asociados a la programación para lograr la funcionalidad interactiva de un producto multimedia.
- **MVC:** (Modelo Vista Controlador) Patrón de diseño de software que separa 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.

- **Palabras calientes:** Palabras activas que difieren en cuanto a su formato al contenido donde se encuentran y que profundizan acerca de un concepto o contenido con tan solo hacer un click sobre ella.
- **SWF:** Se trata de la extensión de los archivos creados con Macromedia Flash, y significa (S)hock (W)ave (F)lash. Los archivos SWF no son editables, y son una compilación y compresión del archivo de autor (FLA) editable desde Flash.
- **TICs** (Tecnologías de la Información y Comunicaciones): Instrumentos y procesos utilizados para recuperar, almacenar, organizar, manejar, producir, presentar e intercambiar información por medios electrónicos y automáticos. En resumen, las TIC son aquellas tecnologías que permiten transmitir, procesar y difundir información de manera instantánea. Son consideradas la base para reducir la Brecha Digital sobre la que se tiene que construir una Sociedad de la Información y una Economía del Conocimiento.