



Universidad de las Ciencias Informáticas
Facultad 5

Multimedia interactiva para el Bloque Básico
General del Perfil de Calidad de Software.

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

Autor: Raidel Ramón de Armas Gurri

Tutor: MsC. Yamilis Fernández Pérez

Ciudad de la Habana, junio de 2007

Año 50 de la Revolución

Declaración de Autoría.

Declaro que soy el único autor de este trabajo y autorizo al Perfil de Calidad de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Raidel Ramón de Armas Gurri

MsC. Yamilis Fernández Pérez

Firma Autor

Firma Tutor

Dedicatoria

*A mi padre y amigo.
A mi madre y abuela siempre presentes.
A mi cariñosa hermana.*

Agradecimientos

A mi familia, por haberme apoyado en todo momento, por haber confiado en mí en el transcurso de la carrera.

A mis padres por todo su amor y cariño. Por haberme inspirado a superarme profesionalmente.

A mi amigo Enrique por su bondad, apoyo incondicional y confianza en la realización de este proyecto de tesis.

A Marinos por su preocupación y asesoría en cuanto el diseño gráfico de la multimedia.

A Dayany por brindarme su apoyo e ideas en los momentos que la necesite.

A Yamilis, mi tutor, por su apoyo y ayuda.

A mis compañeros de estudio, que han sido como mi familia, quienes me han brindado cariño, apoyo y confianza durante la carrera.

Agradecerle a la Revolución por haberme brindado la oportunidad de estudiar en la 1era. Universidad surgida en el calor de la batalla de ideas.

A la Universidad por haberme ofrecido la oportunidad de superarme durante los cinco años de la carrera.

A todos aquellos que de una forma u otra influyeron tanto en mi carrera como en la realización de este proyecto de tesis.

Resumen

Las Tecnologías de la Información y las Comunicaciones (TICs) han logrado un cambio significativo en la sociedad con respecto a la educación, brindando una serie de beneficios que logran mejorar el proceso de enseñanza-aprendizaje como herramientas educativas. Esas nuevas posibilidades de acceso, almacenamiento, y utilización de la información han dado paso al surgimiento de novedosas formas de enseñanza con técnicas multimedia.

Por esa razón el presente trabajo de diploma se centra en la creación de un producto multimedia interactivo como material complementario de apoyo a los cursos que se imparten dentro del Bloque Básico del Perfil de Calidad en la Universidad de Ciencias Informáticas proporcionándole al estudiante un ambiente ameno en su proceso de aprendizaje. Este documento también contiene información acerca de los principales conceptos relacionados con los cursos que se imparten en este Bloque Básico como son el de Introducción a la Calidad y al Proceso de Desarrollo de Software además sobre el análisis, diseño, implementación e información teórica necesaria para la creación de este tipo de aplicación. Obteniéndose una caracterización de las tendencias actuales para la creación de software multimedia, así como una evaluación de las plataformas y herramientas de desarrollo más utilizadas en la actualidad para estas aplicaciones.

Palabras Claves:

Calidad de Software, Proceso de Desarrollo de Software, Multimedia y Metodología.

Índice

Introducción	1
Capítulo 1. Fundamentación Teórica	6
1.2.1 Control de Calidad	7
1.2.2 Aseguramiento de la Calidad	7
1.2.3 Calidad Total	8
1.2.4 Calidad de Software	9
1.2.4.1 Factores que determinan la Calidad del Software	10
1.2.4.2 Aseguramiento de Calidad del Software	11
1.2.5 Proceso de Desarrollo de Software (PDS).....	12
1.2.5.1 Modelos de Proceso Software	16
1.2.5.2 Metodologías para Desarrollo de Software	17
1.3 Análisis crítico de sistemas existentes	18
1.4 Multimedia. Evolución, características y conceptos relacionados	19
1.5 Multimedia. Tendencias y tecnologías actuales	21
1.5.1 Metodologías y lenguajes de modelado para el desarrollo de multimedia	21
1.5.2 Herramientas para el desarrollo de multimedia	24
1.5.3 Propuesta de herramienta y metodología a utilizar	28
Capítulo 2. Descripción de la Solución Propuesta	30
2.1 Introducción	30
2.2 Especificación del contenido curso Introducción a la Calidad de Software	31
2.2.1 Mapa Conceptual de Calidad de Software	32
2.3 Especificación del contenido curso PDS	34
2.3.1 Mapa Conceptual de PDS.....	35
2.4 Identificación de la Audiencia	37
2.5 Modelo Conceptual.....	37
2.5.1 Modelo de Dominio General	37
2.5.2 Análisis de los Conceptos del Dominio	39
2.6 Descripción de la Funcionalidad	39
2.6.1 Requisitos Funcionales.....	39
2.6.2 Requisitos no Funcionales	42
2.7 Modelo de Caso de Uso del Sistema	44
2.7.1 Diagrama de Casos de Uso	44
2.7.2 Determinación y justificación de los Actores del Sistema	45
2.7.3 Casos de Uso	45
2.7.4 Descripción de Casos de Uso del Sistema.....	46
2.8 Mapas de Navegación	49
Capítulo 3. Construcción de la Solución Propuesta	55
3.1 Introducción	55
3.2 Principios de Diseño	56
3.3 Diagramas de Presentación.....	57

3.4 Modelo de Implementación.....	64
3.4.1 Diagrama de Componentes	64
3.4.2 Diagrama de Despliegue.....	66
3.5 Descripción y estructura de principales archivos XML	67
Conclusiones	73
Recomendaciones	74
Bibliografía	75
Glosario de Términos	78
Anexos	81

Introducción

No es ningún misterio que para ser competitivo y tener éxito, las organizaciones deben entender lo que es la calidad en sus productos, servicios y procesos con el objetivo de que estos respondan de manera eficiente y satisfactoria las necesidades establecidas por los clientes.

La **calidad** es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad (*Martínez*) y esta ha evolucionado a lo largo del tiempo sufriendo sucesivas transformaciones hasta llegar a su forma más evolucionada que es la Calidad Total, donde en un primer momento se habla de **control de calidad**: primera etapa en la gestión de la Calidad que se basa en técnicas de inspección aplicadas a la Producción. Posteriormente nace el **aseguramiento de la calidad**: fase que persigue garantizar un nivel continuo de la calidad del producto o servicio proporcionado. Finalmente llegándose a lo que hoy en día se conoce como **calidad total**: un sistema de gestión empresarial íntimamente relacionado con el concepto de Mejora Continua y que incluye las dos fases anteriores.

En el ámbito del **software** la calidad es un conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia donde se deben responder a los requerimientos especificados cubriendo las necesidades y expectativas del cliente o usuario dentro de los tiempos, costos y recursos planificados (*Gutiérrez Márquez, 2006*). Para lograr esto, se hace necesario que el equipo de desarrolladores adopte un **proceso de desarrollo de software (PDS)**, es decir, un marco de trabajo que defina las actividades necesarias para garantizar, técnica y administrativamente, que un software pueda ser construido o mantenido de manera organizada, disciplinada y previsible. Por lo que un mejoramiento de la calidad de este proceso no solamente eleva la calidad del producto, sino también aumenta la eficiencia de costos y tiempo, la posibilidad de tener éxitos en proyectos y la dominación de los riesgos del proceso. Para finalmente traer confianza y satisfacción por parte del cliente.

En la actualidad se puede decir que existen aún problemas en cuanto a la producción de software debido a la situación actual en el PDS a nivel mundial a pesar de la indudable evolución en la calidad de su desarrollo. Donde cada vez más clientes, quedan insatisfechos exigiendo soluciones más novedosas y con mayor calidad. Ejemplo de estos problemas se reflejan en el artículo publicado por la revista “*SD Time*” en el año 2006, donde sólo el 35% de los proyectos que se comenzaron se consideraron exitosos, cumpliendo con el tiempo requerido, el presupuesto y los requisitos exigidos por el cliente. A pesar de mostrarse una marcada mejora con respecto al año 1994 cuando solo el 16.2% fue exitoso todavía no se llega a niveles que hagan de la producción de software una industria segura y de éxito. El 46% de los proyectos comenzados fueron rechazados o no aceptados por los clientes (en 1994, 52.7%) y el 19% nunca se terminaron o fueron cancelados (31.1% en 1994). (*Rubstein, 2007*)

Pudiéndose decir que:

- La industria del software no ha acabado de salir de su fase artesanal.
- Planes y presupuestos excedidos sistemáticamente, al no estar basados en estimaciones realistas.
- Si hay plazos rígidos, se sacrifica funcionalidad y calidad del producto para el cumplimiento del plan y fecha de entrega.
- Cuando los proyectos están fuera del plan, las revisiones o pruebas se recortan o eliminan.

En Cuba **La Universidad de Ciencias Informáticas (UCI)** es una institución encargada de la formación de profesionales altamente calificados que están directamente vinculados a proyectos productivos y que lucha por insertarse en el mercado mundial en la producción y desarrollo de software. La industria Cubana del Software y en especial la UCI, no se encuentra ajena a ninguno de estos problemas mencionados anteriormente. Existiendo problemas con la productividad de los trabajadores, los tiempos de entrega de los productos, las documentaciones, la calidad de las pruebas y la falta de comunicación efectiva entre los usuarios, desarrolladores, administradores, clientes e investigadores. (*Febles, 2001*)

Estos problemas se deben principalmente a que la disciplina ingeniería de software es relativamente reciente y muchos de sus conceptos importantes están aun inmaduros. Según encuestas realizadas por (Suárez Gijón, Díaz Magdariaga, 2007) en muchos de los proyectos productivos de la UCI existen deficiencias en los procesos de desarrollo de software. Fundamentalmente esto es debido al desconocimiento o no aplicación de los nuevos métodos, metodologías, normas y estándares para la gestión de la calidad de software y el PDS como son las normas ISO, IEEE, CMMI, la metodología RUP, XP, etc. Además el personal encargado de realizar el proceso de desarrollo de software, cuenta con poca experiencia en la producción debido a que la UCI se inicia en este mercado. Por lo que se da como resultado que la calidad del software que se produce sufre y experimenta malas consecuencias debido a que en muchas ocasiones los proyectos que se encuentran retrasados en la entrega, sacrifican funcionalidades, acortan el proceso de revisiones y pruebas, llegando en ocasiones también a ser suprimidos.

De ahí el surgimiento de este trabajo como necesidad de dar respuesta a las situaciones antes expuestas, por lo que el **problema** a resolver queda formulado de la siguiente forma: **¿Cómo apoyar en el proceso de enseñanza-aprendizaje de calidad de software y PDS a los estudiantes de la UCI?**

Para ello el **objeto de estudio** lo constituye el proceso de desarrollo de multimedias. Derivándose que el **campo de acción** que abarca esta investigación es el proceso de desarrollo de multimedia interactiva para el apoyo al proceso de enseñanza-aprendizaje de los conceptos básicos de calidad de software y PDS.

Esta indagación tiene como propósito **defender la idea** de que con el desarrollo de esta multimedia interactiva se puede contribuir a disminuir las insuficiencias que existen en el aprendizaje de los conceptos generales de calidad de software y PDS.

Precisamente se decidió que el **objetivo** del presente trabajo consistiera en:

Elaborar una multimedia interactiva para la comprensión de los conceptos esenciales de calidad de software y PDS.

Para dar cumplimiento al objetivo trazado es necesario cumplir con las siguientes **tareas de la investigación:**

- Seleccionar la metodología de análisis y diseño de sistemas informáticos, que facilite la creación y garantice la calidad del sistema.
- Analizar y estudiar cómo se encuentran en el ámbito internacional las herramientas que se utilizan para confeccionar multimedias como la que se pretende desarrollar.
- Recopilar y analizar los contenidos, diferentes aspectos y conceptos vinculados a la calidad de software y el PDS, de manera tal que se corresponda con el programa del bloque básico general del perfil de calidad en la UCI.
- Confeccionar mapas conceptuales que contengan los conceptos necesarios para la comprensión del tema de calidad de software y PDS.
- Diseñar e implementar una multimedia que contenga los contenidos recopilados y los mapas conceptuales, asociando a estos contenidos recopilados diversos medios (videos, juegos y autoevaluaciones) que ayudan a la comprensión de dichos contenidos.

El desarrollo de la investigación del presente trabajo esta constituido por 3 capítulos de los cuales a continuación se da un pequeño resumen.

Capítulo 1. Fundamentación teórica: Resume los principales conceptos que se relacionan con el tema propuesto en este trabajo y de los vinculados al objeto de estudio con el fin de lograr una mejor comprensión del problema al que se enfrenta. Se describen además las metodologías, lenguajes de

programación y sistemas para realizar el análisis y diseño de la aplicación multimedia, así como las tendencias y tecnologías actuales de este tipo de aplicaciones.

Capítulo 2. Descripción de la solución propuesta: Realiza un análisis de lo que debe hacer el sistema. Definiendo el dominio en donde el mismo está enmarcado, los mapas de navegación y el levantamiento de los requisitos funcionales y no funcionales. Además identifica procesos y se modela el sistema a través de la descripción de los casos de uso del sistema.

Capítulo 3. Descripción de la solución propuesta: Dedicado a la construcción del sistema a desarrollar donde se incluyen los diagramas de presentación. Además se aborda el modelo de implementación el cual incluye el diagrama de componentes y de despliegue.

Capítulo 1. Fundamentación Teórica

1.1 Introducción

El presente capítulo tiene como objetivo dar una panorámica general acerca de los principales conceptos que se relacionan con el tema propuesto en este trabajo. Además se dan algunas características que sirven de punto de partida a la solución del problema antes mencionado, definiendo algunos conceptos que serán de utilidad en la comprensión de la solución propuesta.

1.2 Introducción a la calidad y al PDS. Antecedentes, evolución y principales conceptos

Desde los tiempos más remotos, el hombre al construir sus armas, elaborar sus alimentos y fabricar sus vestidos observa las características de ese producto y enseguida busca la manera de cómo mejorarlo (*Figueroa*). Por lo que la calidad surge de la propia necesidad de contar con un producto fiable.

En la búsqueda de lograr una mayor calidad esta ha evolucionado sufriendo diversas transformaciones y atravesando por diferentes etapas. En la que primeramente se contaba con una calidad de manera muy artesanal en la cual se hacían las cosas bien no importando el esfuerzo necesario para hacerlo, donde solo el propio artesano tenía la responsabilidad de inspeccionar el producto terminado, con el interés de mantener una buena reputación y pago por su trabajo bien hecho.

La llegada de la revolución industrial cambia un poco la manera en que se realiza la producción con la creación de fábricas de producción masiva, donde lo que se trata es de hacer muchas cosas, no importando que sean de calidad, confundándose la alta producción con calidad, satisfaciéndose una gran demanda de bienes (*Pérez*).

Al estallar la segunda guerra mundial lo más importante era asegurar la eficiencia del armamento, con la producción más rápida posible, para así garantizar una gran cantidad de armamento eficaz en la cantidad y en el momento preciso.

En la etapa de post guerra en Japón se trabaja por lograr una producción satisfactoria, en un primer momento para minimizar costos mediante la calidad, satisfacer al cliente y lograr mayor competencia en el mercado mundial. Mientras en el resto del mundo la tendencia estaba enfocada a producir en cantidad, olvidándose de la calidad, pues la principal tarea era satisfacer la gran demanda causada por los grandes desastres que trajeron consigo la segunda guerra mundial.

1.2.1 Control de Calidad

Un tiempo después de que el mundo se recuperase de los daños causados por la segunda guerra mundial ya se puede decir que se habla de un **control de calidad**, siendo esta la primera etapa en la **gestión de la calidad**. Se caracteriza por la realización de inspecciones a la producción comprobando si una determinada materia prima, semielaborado o algún producto terminado cumple con determinadas especificaciones establecidas previamente, para así evitar la salida de bienes defectuosos satisfaciéndose las necesidades técnicas del producto (Gonzalez, 2003). Es un poco errónea esta manera de ver la calidad ya que dichas inspecciones tienen lugar cuando la materia prima se ha recibido, cuando un proceso productivo ha concluido o cuando el producto final está terminado.

1.2.2 Aseguramiento de la Calidad

Al incrementarse el desarrollo tecnológico surgen industrias como la Nuclear, la Aeronáutica, la de Defensa, entre muchas otras. Donde prevenir los fallos de calidad es más rentable que tener que corregirlos o lamentarlos, incorporándose el concepto de prevención, garantía o aseguramiento a la **gestión de la calidad** que se desarrolla en esos tipos de Industrias. De esta manera surge el

aseguramiento de la calidad (Gonzalez, 2003). Fase que enriquece la función de la calidad en las empresas industriales la cual es realizada por personal más calificado. Esta etapa no pretende sustituir a su anterior (control de la calidad), si no más bien lo que hace es absorberla sirviendo como herramienta de gestión para apropiarse aún más de la confianza del cliente al cual se le ofrece un determinado servicio.

1.2.3 Calidad Total

Todas estas necesidades han llevado a que en la actualidad se tenga una nueva percepción de lo que es calidad, dando paso al surgimiento de una nueva etapa denominada **calidad total**. Esta es la etapa más evolucionada dentro de las sucesivas transformaciones que ha sufrido el término calidad a lo largo del tiempo. Es un sistema de gestión empresarial íntimamente relacionado con el concepto de mejora continua y que incluye las dos fases anteriores (Pérez) Fig. 1. Implantar este proceso en una empresa tiene un principio pero nunca un fin e involucra a todos los miembros, actividades y procesos llevados a cabo en la organización, incluso al cliente y al proveedor cuando esto sea posible. Con esta estrategia de calidad lo que se busca es garantizar, a largo plazo, la supervivencia y el crecimiento de una organización, mediante el aseguramiento permanente de la satisfacción de los clientes y la eliminación de todo tipo de desperdicios, respondiendo a la necesidad de transformar los productos, servicios, procesos, estructuras y cultura de la empresa, para asegurar su futuro.

Evolución histórica

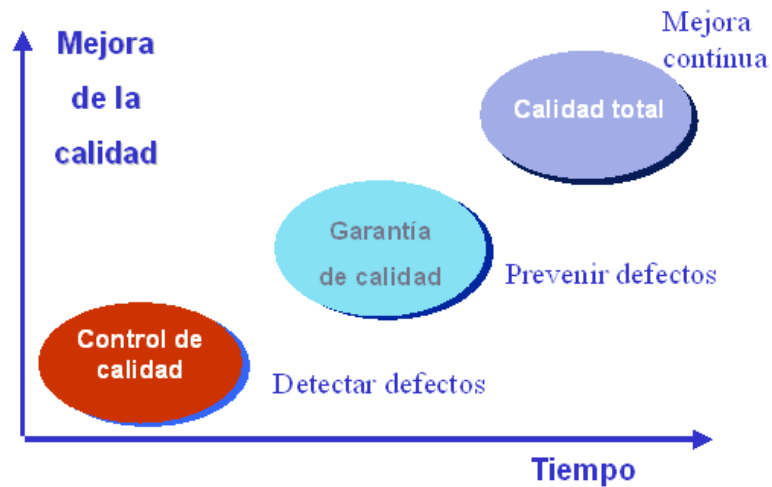


Fig.1 Evolución Histórica de la Gestión de Calidad (DMI, 2007).

1.2.4 Calidad de Software

A continuación se dan algunas de las definiciones de calidad de software:

Grado con el que un sistema, componente o proceso cumple los requerimientos especificados, las necesidades y expectativas del cliente o usuario (*IEE, 1990*).

Nivel de fiabilidad, robustez y eficiencia del software referido a todo su comportamiento en todo el período de vigencia (*SEDNA, 2006*).

Conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas (*ISO 8402, 1994*).

Haciendo una comparación entre las anteriores definiciones se puede decir que Roger S. Pressman hace una definición más completa de lo que a calidad de software se refiere definiéndola como: La concordancia del software producido con los requisitos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requisitos implícitos no establecidos formalmente, que desea el usuario (*PRESSMAN, 1998*).

Los **requisitos del software** son la base de las medidas de calidad, siendo estos los parámetros que debe cumplir software para su correcto funcionamiento, por lo que una falta de concordancia con los requisitos es una falta de calidad. Al igual existen algunos **requisitos implícitos** que a menudo no se mencionan, o se mencionan de forma incompleta y que también pueden implicar una falta de calidad. Hay que hacer un esfuerzo continuo para cada contexto de uso a la hora de la entrega por parte del usuario, es decir, es necesario comprender las necesidades reales de los usuarios con tantos detalles como sea posible, por lo que la calidad del software todavía es un logro importante a alcanzar.

1.2.4.1 Factores que determinan la Calidad del Software

James McCall descompone el concepto de calidad en tres usos o capacidades importantes para un producto de software y cada capacidad en una serie de factores que determinan la calidad en cada una de ellas.

Estos Factores se centran en tres grandes grupos (*McCall, 1977*):

- **Sus características operativas**
 - **Corrección:** El grado en que una aplicación satisface sus especificaciones y consigue los objetivos encomendados por el cliente.
 - **Fiabilidad:** El grado con que una aplicación lleva a cabo las operaciones especificadas y con la precisión requerida.
 - **Eficiencia:** La cantidad de recursos hardware y software que necesita una aplicación para realizar las operaciones con los tiempos de respuesta adecuados.

- **Seguridad:** El grado con que puede controlarse el acceso al software o a los datos de personal no autorizado.
- **Facilidad de Uso:** El esfuerzo requerido para aprender el manejo de una aplicación, trabajar con ella, introducir datos y conseguir resultados.

- **Capacidad de Soportar los Cambios**
 - **Facilidad de mantenimiento:** El esfuerzo requerido para localizar y reparar errores.
 - **Flexibilidad:** El esfuerzo requerido para modificar una aplicación en funcionamiento.
 - **Facilidad de prueba:** El esfuerzo requerido para probar una aplicación de forma que cumpla con lo especificado en los requisitos.

- **Adaptabilidad a nuevos entornos**
 - **Portabilidad:** El esfuerzo requerido para transferir la aplicación a otro hardware o sistema operativo.
 - **Reusabilidad:** El grado en que partes de una aplicación pueden utilizarse en otras aplicaciones.
 - **Interoperabilidad:** El esfuerzo necesario para comunicar la aplicación con otras aplicaciones o sistemas informáticos.

1.2.4.2 Aseguramiento de Calidad del Software

Para llevar a cabo la elaboración de un proyecto de software se debe tener en cuenta una de sus principales etapas que es el aseguramiento de la calidad del software conocido comúnmente como Software Quality Assurance (SQA) que no es más que un conjunto de actividades planificadas y sistemáticas necesarias para confiar en que el producto software cumplirá los requisitos dados de calidad.

Las reglas del grupo de SQA tratan de ayudar al equipo de ingeniería del software en la consecución de un producto final de alta calidad. El Instituto de Ingeniería del Software recomienda un conjunto de

actividades de SQA que se enfrentan con la planificación de garantía de calidad, supervisión, mantenimiento de registros, análisis e informes (*PRESSMAN, 1998*).

Estas son algunas de las principales actividades que realizan un grupo independiente de SQA (*PRESSMAN, 1998*):

- Participación en el desarrollo de la descripción del proceso de software del proyecto.
- Revisión de las actividades de ingeniería del software para verificar su ajuste al proceso de software definido.
- Auditoria de los productos de software designados para verificar el ajuste con los definidos como parte del proceso del software.
- Asegurar que las desviaciones del trabajo y los productos del software se documenten y se manejen de acuerdo con un procedimiento establecido.

Este aseguramiento se diseña para cada aplicación antes de comenzar a desarrollarla y no después, esta presente en los métodos y herramientas de análisis, diseño, programación y prueba, al igual que en las inspecciones técnicas formales en todos los pasos del proceso de desarrollo del software, en el control de la documentación del software y de los cambios realizados y en los mecanismos de medida.

1.2.5 Proceso de Desarrollo de Software (PDS).

Para la obtención de un software con calidad hay que tener presente la utilización de metodologías y procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad en el proceso de construcción de un software, a la vez que eleve la productividad, tanto para la labor de desarrollo como para el control de la calidad del mismo.

Un PDS es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código; el código es

probado, documentado y certificado para su uso operativo. Concretamente define **quién** está haciendo **Qué, Cuándo y Cómo** para alcanzar un determinado objetivo (Jacobson, 2000).

Cada una de estas etapas de Diseño, Implementación y Prueba, están parcialmente ordenadas con la intención de lograr un objetivo. En este caso es la obtención de un producto de software con calidad y que cumpla con las especificaciones establecidas dentro de los tiempos, costos y recursos acordados con el cliente Fig.2.

Todo este proceso de desarrollo también requiere por otro lado de un conjunto de conceptos, una metodología y un lenguaje propio. El mismo es conocido como **ciclo de vida del software**.

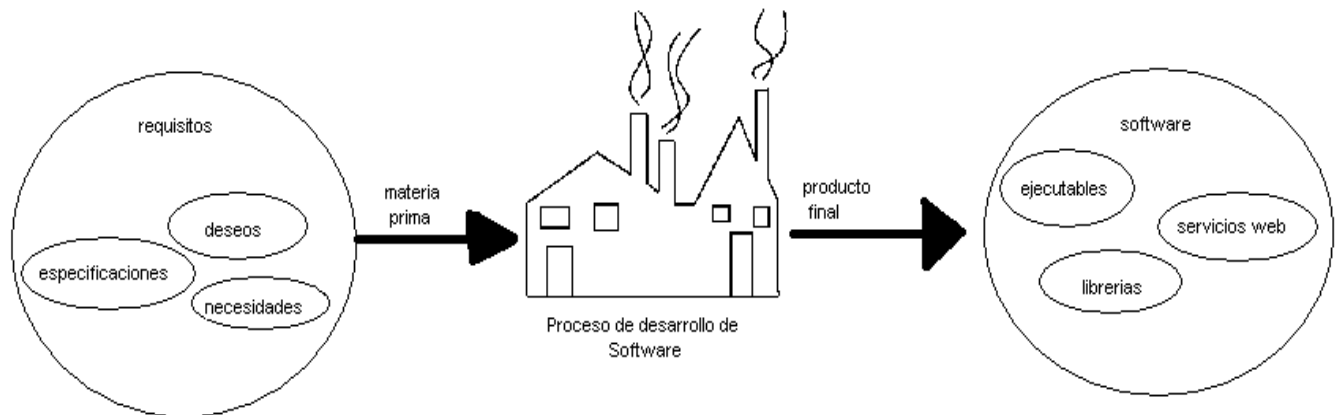


Fig. 2 PDS. Recibe como materia prima los requisitos y da como resultado un producto software con calidad.

El PDS no es único por lo que no existe un proceso de desarrollo universal que sea efectivo para los diferentes contextos de proyectos de desarrollo. Debido a esta gran diversidad resulta difícil y complejo automatizar todo un PDS, pero a pesar de la variedad de propuestas de procesos de software, existe un conjunto de actividades fundamentales que le son comunes y se encuentran presentes en cada uno de ellos (Sommerville, 2002):

Especificación de software: Se debe definir la funcionalidad y restricciones operacionales que debe cumplir el software.

Diseño e Implementación: Se diseña y construye el software de acuerdo a la especificación.

Validación: El software debe validarse, para asegurar que cumpla con lo que quiere el cliente.

Evolución: El software debe evolucionar, para adaptarse a las necesidades del cliente.

Además de estas actividades fundamentales existen otro grupo de actividades denominadas protectoras que se aplican a lo largo de todo el PDS. Las cuales son (*PRESSMAN, 1998*):

- Seguimiento y control de proyecto de software.
- Revisiones técnicas formales.
- Garantía de calidad del software.
- Gestión de configuración del software.
- Preparación y producción de documentos.
- Gestión de reutilización.
- Mediciones.
- Gestión de riesgos.

Un PDS también es caracterizado como lo que se muestra en la *Fig. 3*.

A continuación se describen los elementos involucrados en la *Fig. 3* (*PRESSMAN, 1998*):

- **Un marco de trabajo común del proceso**, definiendo un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos de software, con independencia del tamaño o complejidad.
- **Un conjunto de tareas**, cada uno es una colección de tareas de ingeniería del software, hitos de proyectos, entregas y productos de trabajo del software, y puntos de garantía de calidad, que permiten que las actividades del marco de trabajo se adapten a las características del proyecto de software y los requisitos del equipo del proyecto.
- **Las actividades de protección**, tales como garantía de calidad del software, gestión de configuración del software y medición, abarcan el modelo del proceso. Las actividades de protección son independientes de cualquier actividad del marco de trabajo y aparecen durante todo el proceso.

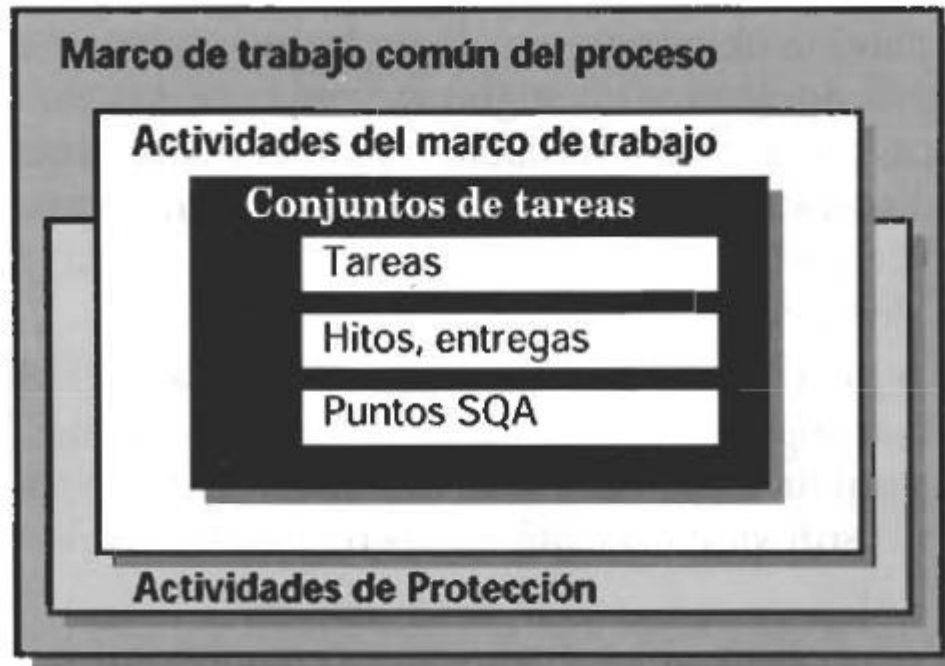


Fig. 3 Elementos del proceso del software (PRESSMAN, 1998).

Es necesario que un PDS sirva como guía a todos los involucrados en el proyecto como son: desarrolladores, directivos, ejecutivos e incluso a los clientes o usuarios finales. Por lo que un proceso antiguo no sirve, se necesita uno que sea el mejor proceso que se pueda reunir en ese momento de la historia y que este ampliamente disponible de forma que todos los interesados puedan comprender su papel en el proceso que se encuentran implicados.

La disposición de un PDS bien definido e implementado en una empresa aumenta su rendimiento, la calidad de su proceso de producción y sus productos finales trayendo consigo más capacidad de negocio.

1.2.5.1 Modelos de Proceso Software

Un Modelo de Proceso Software es una representación simplificada de un proceso de software representada desde una perspectiva específica. Por su naturaleza los modelos son simplificados, por lo tanto un modelo de proceso de software es una abstracción de un proceso real (*Sommerville, 2002*).

Los modelos genéricos no son descripciones definitivas de procesos de software. Sin embargo, llegan a ser abstracciones útiles que pueden ser utilizadas para explicar diferentes enfoques del desarrollo de software.

A continuación se definen algunos de los modelos:

Codificar y corregir (Code-and-Fix): Se trata de primero implementar algo de código y luego pensar acerca de requisitos, diseño, validación y mantenimiento.

Modelo en cascada: Este modelo toma las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución y las representa como fases separadas del proceso, donde cada fase tiene como resultado documentos que deben ser aprobados por el usuario y una fase no comienza hasta que termine la fase anterior, además, generalmente se incluye la corrección de los problemas encontrados en fases previas. En la práctica, este modelo no es lineal, e involucra varias iteraciones e interacciones entre las distintas fases de desarrollo y debe usarse si se entienden a plenitud los requisitos.

Desarrollo evolutivo: La idea de este modelo es el desarrollo de una implantación del sistema inicial, exponerla a los comentarios del usuario, refinarla en N versiones hasta que se desarrolle el sistema adecuado obteniéndose una rápida realimentación del usuario, ya que las actividades de especificación, desarrollo y pruebas se ejecutan en cada iteración.

Desarrollo formal de sistemas: Se basa en transformaciones formales de los requisitos hasta llegar a un programa ejecutable y el mantenimiento se realiza sobre la especificación (no sobre el código)

fuelle) y es realizado por repetición del proceso (no mediante parches sobre la implementación) permitiendo demostrar la corrección del sistema durante el proceso de transformación.

Desarrollo basado en reutilización: Modelo fuertemente orientado a la reutilización de código disminuyendo el costo, esfuerzo y riesgo de desarrollo reduciéndose el tiempo de entrega.

Procesos iterativos: Se clasifican en dos grupos especialmente diseñados para el soporte de las iteraciones.

- **Desarrollo Incremental:** Una forma de reducir la repetición del trabajo en el proceso de desarrollo y dar oportunidad de retrasar la toma de decisiones en los requisitos hasta adquirir mas experiencia con el sistema es una combinación del modelo de cascada y modelo evolutivo.
- **Desarrollo en Espiral:** El ciclo de desarrollo se representa como una espiral, en lugar de una serie de actividades sucesivas con retrospectiva de una actividad a otra, tomando en consideración el riesgo, siendo el mismo una actividad importante en la administración del proyecto.

Un modelo no es más adecuado que otro solo que cada proyecto de software requiere de una forma particular de abordar el problema. Las propuestas actuales promueven procesos iterativos, donde en cada iteración pueda utilizarse uno u otro modelo de proceso.

1.2.5.2 Metodologías para Desarrollo de Software

Suele denominarse **metodología** a un proceso de software detallado y completo. Las metodologías se basan en una combinación de los modelos de proceso de software (cascada, evolutivo, incremental, etc.). Adicionalmente una metodología debe definir con precisión los artefactos, roles y actividades involucradas, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto, guías para uso de herramientas de apoyo, etc. Habitualmente se utiliza el término **método** para referirse a técnicas, notaciones y guías asociadas, que son aplicables a una (o algunas) actividades del proceso de desarrollo, por ejemplo, suele hablarse de métodos de análisis y diseño.

La comparación y clasificación de metodologías no es una tarea sencilla debido a la diversidad de propuestas y diferencias en el grado de detalle, información disponible y alcance de cada una de ellas. Si se toma como criterio las notaciones utilizadas para especificar artefactos producidos en actividades de análisis y diseño, se pueden clasificar las metodologías según el paradigma de programación en dos grupos:

- **Metodologías estructuradas:** las cuales son particularmente apropiadas para proyectos que utilizan en su implementación lenguajes de 3ra y 4ta generación.
- **Metodologías orientadas a objetos:** unidas a la evolución de los lenguajes de programación orientadas a objetos como c++ y Smalltalk dando lugar al Unified Modeling Lenguaje (UML) la notación orientada a objeto es más popular en la actualidad.

Por otra parte, considerando su filosofía de desarrollo, aquellas metodologías con mayor énfasis en la planificación, control del proyecto, especificación precisa de requisitos y modelado reciben el apelativo de **metodologías tradicionales o pesadas**. Mientras las denominadas **metodologías ágiles** están más orientadas a la generación de código con ciclos muy cortos de desarrollo, se dirigen a equipos de desarrollo pequeños, especialmente haciendo hincapié en aspectos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso.

1.3 Análisis crítico de sistemas existentes

No existe ninguna herramienta de autor en específico para la enseñanza de calidad y proceso de desarrollo de software (PDS), pero no obstante existen en internet varios sitios web con material docente, tutoriales y cursos online montados, al igual que entornos virtuales de aprendizaje que tratan conceptos relacionados con los temas de Calidad y PDS.

Ya más específicamente para el estudio de ingeniería de software en forma general existen algunas herramientas como:

- **Problems and Programmers y SimSE:** Dos herramientas creadas por un grupo de desarrolladores de la Universidad Irvine de California en forma de juego para la enseñanza-aprendizaje de Ingeniería del Software, donde en el juego se simulan procesos teniendo el estudiante la oportunidad de practicar como si fuera un proyecto real pero en un menor tiempo. En estos juegos se propone dar un acercamiento a lo que se puedan enfrentar realmente los estudiantes una vez ya graduados.

En la UCI el estudio de estos temas es realizado a través de Cursos Optativos del Perfil de Calidad montados en la plataforma Moodle (entorno virtual de aprendizaje online de libre distribución que permite la gestión de cursos). Los mismos están necesitados de una mejor estructuración y de herramientas que apoyen al estudiante en el apropiamiento del contenido haciendo más atractivo e intuitivo su proceso de aprendizaje.

1.4 Multimedia. Evolución, características y conceptos relacionados

La comunicación desarrolla, a partir de los 70s, en la educación, la instrucción, la capacitación y la publicidad, el concepto operativo de multimedia. Por tal concepto se entiende la integración de diversos medios (visuales y auditivos) para la elaboración y envío de mensajes por diversos canales, potencializando la efectividad de la comunicación, a través de la redundancia; De esta forma, la comunicación resulta más atractiva, afecta e impacta a más capacidades de recepción de la persona y aumenta la posibilidad de eliminar el ruido que puede impedir la recepción del mensaje (*Corrales Díaz, 1994*).

En el ámbito de la computación el término multimedia es más nuevo y designa el uso de varios recursos o medios como audio, video, animaciones, texto y gráficas en una computadora ofreciendo posibilidades de creatividad mediante estos sistemas de computación. Esta se inicia en 1984 año cuando Apple Computer lanzó la Macintosh, la primera computadora con amplias capacidades de

reproducción de sonidos equivalentes a los de un radio, característica unida a su sistema operativo y programas que se desarrollaron en la forma que se conoce como ambiente Windows muy apropiados para el diseño gráfico y la edición.

La tecnología de multimedia toma auge en los video-juegos, a partir de 1992, cuando se integran audio, video, gráficas, animación y texto al mismo tiempo. Esta permite navegar y buscar la información que se desea sobre un tema, sin tener que recorrer todo el programa y que la información no sea lineal sino asociativa donde se extiende y expande la forma en que se interactúa con una computadora.

La tecnología multimedia podría describirse como la integración de las cualidades informativas y comunicativas de texto, sonido, imágenes fijas y vídeo con la interactividad de la computadora, para comunicar, informar, crear habilidades o simplemente entretener. Los hiperenlaces y la posibilidad de conectar piezas de información en una red ofrecen nuevas posibilidades a los productores de información. La parte “**hiper**” que es la que convierte a la multimedia en **hipermedia** es la que provee a esta tecnología la verdadera revolución en la forma de presentar y recuperar la información (*Cossío*), la cual permite generar áreas, dentro de una pantalla, sensible al mouse, al toque o a una tecla. De modo que el usuario navegue, interactúe con los archivos o partes del programa conforme a sus intereses personales, regrese a la parte original o se adentre en la exploración de otra parte del programa, sin necesidad de recorrerlo todo. Es por eso que estas tecnologías brillan especialmente en el área de los servicios y productos de información.

En la **educación**, la multimedia se utiliza para producir los cursos de aprendizaje computarizado y los libros de consulta como son las enciclopedias, dejando al usuario pasar con una serie de presentaciones, de texto sobre un asunto particular y de ilustraciones asociadas en varios formatos de información como pueden ser las imágenes, los textos y los videos . Por lo que cuando en una multimedia logra combinar adecuadamente los medios, se mejora notablemente la atención, la comprensión y el aprendizaje, ya que se emplean varios sentidos para comprender un mismo objeto o concepto.

1.5 Multimedia. Tendencias y tecnologías actuales

El avance tecnológico es algo que esta en constante aumento y con el paso del tiempo la multimedia se ha ido desarrollando y se ha convertido en el más reciente fenómeno tecnológico y cultural, que está contribuyendo a cambiar muchos conceptos de la vida y del trabajo cotidiano, aportando muchas ventajas y facilidades en el desarrollo diario.

Unos de los avances donde se utilizan recursos de multimedia y que dan muchos resultados son la combinación de las teleconferencias vía internet y los entornos virtuales de aprendizaje, los cuales aumentan posibilidades de cultura, educación, capacitación, información e instrucción, de modo interactivo. Fomentándose así la teleformación que no es más que cursos en línea o a distancia que administran y representan contenidos educativos en diferentes formatos aprovechándose las ventajas que nos brindan las Tecnologías de Información y las Comunicaciones.

Aunque existen avances, los desarrollos de multimedia enfrentan obstáculos de normatividad tecnológica en torno a la compatibilidad, contando con alrededor de más de 30 arquitecturas diferentes e incompatibles entre sí. Por lo que el desarrollo de productos multimedia no es nada fácil y mucho menos cuando se intenta adecuar estos para que sean funcionales en cualquier equipo. Además no solo se basa en crear nuevos productos, si no que también implica adecuar estos a normas y estándares de compatibilidad que haga posible que los programas desarrollados puedan ser usados en diferentes tecnologías con una plataforma que tiende a ser uniforme.

1.5.1 Metodologías y lenguajes de modelado para el desarrollo de multimedia

Mientras que en otras metodologías de desarrollo se trata con especial interés los aspectos de almacenamiento y funcionalidad, precisamente en las aplicaciones multimedia el objetivo principal es

difundir información almacenada en diferentes medios como imágenes, vídeos, sonido, etc. de manera que lleguen al usuario no necesariamente experto en informática de una forma sencilla, fácil e intuitiva.

Por esta razón, a principios de los 90, se comienza a estudiar la necesidad de crear metodologías que guíen a los desarrolladores y que aseguren la calidad de los productos multimedia.

Algunas de estas metodologías y lenguajes de modelado más conocidos son:

La Metodología **MultiMet** orientada a las etapas de concepción más que a la descripción de la modelación del producto como tal. Aunque es capaz de guiar las acciones circundantes a la fabricación, deja un hueco en el conocimiento necesario para la estructura programática del software y el flujo de procesos durante el mismo, así como instrumentos que faciliten el análisis, diseño e implementación (*Santos Fabelo, 2008*).

La metodología **CEDISAC (Centro de Diseño de Sistemas Automatizados)** desarrollada y utilizada en la empresa CITMATEL por el MSc. Rafael Barrera Yanes; obtenida a partir de la experiencia acumulada durante varios años (1992-1996). Constituye un proceso para construir una obra de comunicación audiovisual interactivo, basada en la praxis obtenida.

El modelo **HDM (Hypertext Design Model)** método desarrollado para definir la estructura y la navegación propia de las aplicaciones multimedia. Se basa en el modelo Entidad-Relación e introduce nuevos elementos, como las unidades o los enlaces. HDM no supone una metodología para el desarrollo de aplicaciones multimedia, es simplemente una técnica de modelado. Los elementos definidos por HDM sirven para definir este tipo de aplicaciones, pero resultan insuficientes para guiar al diseñador en el proceso de desarrollo de las mismas, sentando las bases para futuras propuestas de desarrollo, ofreciendo ideas como la separación de lo conceptual, información que se almacena, y de la presentación e información que se presenta (*Garzoto Paolini, 1993*).

El modelo **RMM (RelatioShip Management Methodology)** se define como un proceso de análisis, diseño y desarrollo de aplicaciones multimedia. Los elementos principales de RMM son el modelo E-R (Entidad-Relación) y el modelo **RMDM (Relationship Management Data Model)** basado en el modelo HDM. El modelo RMDM propone un lenguaje que permite describir los objetos del dominio, sus interrelaciones y los mecanismos de navegación hipertexto de la aplicación. Según sus autores, está orientada a problemas con datos dinámicos que cambian con mucha frecuencia, más que a entornos estáticos (*Izakowitz, 1995*).

El modelo **OOHDM (Object Oriented Hypermedia Design Model)** es una propuesta basada en el diseño haciendo una separación clara entre lo conceptual, lo navegacional y lo visual. Esta independencia hace que el mantenimiento de la aplicación sea mucho más sencillo. Además, es la primera propuesta que hace un estudio profundo de los aspectos de interfaz. También orientada a objetos, hace uso de un diagrama tan estandarizado como el de clases, para representar el aspecto de la navegación a través de las clases navegacionales: índices, enlaces y nodos (*Rossi, 1997*).

La metodología **SOHDM (Scenario-based Object-oriented Hypermedia Design Methodology)** es hasta ahora la única propuesta que tiene en cuenta aspectos como la especificación de requisitos haciendo uso de los escenarios y cubre todas las fases del proceso de desarrollo, obviando la implantación y las pruebas (*Lee Heeseok, 1998*).

La metodología **HFPM (Hypermedia Flexible Process Modeling Strategy)** engloba todas las fases del proceso de desarrollo, va desde el análisis hasta el desarrollo de la documentación y el mantenimiento. Además divide y detalla cada una de las tareas que comprende cada fase (*Olsina, 1998*).

La metodología **OO/Pattern Approach** propuesta bastante similar a HFPM pues ambas proponen el uso de patrones y es orientada a objetos para el diseño navegacional y la interfaz. Sin embargo, esta propuesta, a diferencia de FPM, no cubre el ciclo completo de desarrollo (*Thomson, 1998*).

La metodología **RUP (Proceso Unificado de Software)** es un proceso desarrollado por Philippe Kruchten, Ivar Jacobson y otros de la Corporación “Rational Software”, ahora una división de IBM, cuyo objetivo es producir software de alta calidad, y que cumpla con los requisitos de los usuarios dentro de una planificación y presupuestos establecidos. 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.

RUP no es específico para diseño de multimedia, sin embargo a través de la extensión de UML para multimedia, conocida por **OMMMA – L** la que 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 nuevos que incorpora como son los mapas de navegación y los diagramas de presentación, 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 (*Labori de la Rosa, et al., 2007*).

1.5.2 Herramientas para el desarrollo de multimedia

El aumento de las ofertas y necesidades de formación mediante cursos distribuidos a través de la Web, así como el número de profesores, educadores y expertos que utilizan los servicios de Internet para desarrollar su actividad profesional ha potenciado la investigación y el desarrollo, por parte de instituciones, universidades y empresas comerciales, de herramientas cada vez más potentes y fáciles de utilizar por el usuario. Estas herramientas abarcan tanto aquellas destinadas a la creación de materiales multimedia, como los editores de páginas Web, software de comunicación y trabajo colaborativo o las diseñadas específicamente para la distribución de cursos a través de Internet.

Las características técnicas son uno de los aspectos claves que se deben tener en cuenta a la hora de seleccionar las herramientas a utilizar para la confección de una multimedia. Estas características

deben adaptarse a las necesidades y posibilidades de cada organización o institución que decide crear un entorno de aprendizaje.

La flexibilidad y capacidad de adaptarse al cambio debe ser una característica de estas herramientas y vienen determinada por (*Milgrom, Simbandumwe, 1997*):

- **Posibilitar el acceso remoto.** Tanto los profesores como los alumnos pueden acceder remotamente al curso en cualquier momento desde cualquier lugar con conexión a Internet.
- **Utilizar un navegador.** Los usuarios acceden a la información a través de navegadores existentes en el mercado (como Netscape, Explorer o Mozilla). Utilizando el protocolo de comunicación http.
- **Multiplataforma.** Utilización de estándares para poder ser visualizados en cualquier ordenador.
- **Interfaz gráfica.** Posibilitan la integración de diferentes elementos multimedia: texto, gráficos, vídeo, sonidos, animaciones, etc.
- **Actualización de la información.** La información contenida en las páginas web puede ser modificada y actualizada de forma relativamente sencilla.
- **Diferentes niveles de usuarios.** Presenta tres niveles de usuario con privilegios distintos: el administrador, que se encarga del mantenimiento del servidor y de la creación de los cursos; el diseñador, es la figura del profesor el cual diseña, elabora materiales y se responsabiliza del desarrollo del curso; y el alumno que es el estudiante matriculado o interesado en recibir los cursos.

Desde la parte pedagógica, las principales características de estas herramientas (*de Benito Crosetti, 2000*):

- **Seguimiento del progreso del estudiante.** Proporcionan diferentes tipos de información que permiten al profesor realizar un seguimiento sobre el progreso de aprendizaje del alumno.
- **Comunicación interpersonal.** Posibilitar el intercambio de información y el diálogo y discusión entre todas las personas implicadas en el proceso.

- **Trabajo colaborativo.** Posibilitar el trabajo colaborativo entre los alumnos a través de diferentes aplicaciones que les permiten compartir información, trabajar con documentos conjuntos, facilitar la solución de problemas y la toma de decisiones.
- **Gestión y administración de los alumnos.** Permitir llevar a cabo todas aquellas actividades relacionadas con la gestión académica de los alumnos como matriculación, establecer privilegios de acceso, la creación de grupos y acceso a la información sobre el alumno.
- **Creación de ejercicios de evaluación y autoevaluación.** Proporcionar al profesor la información sobre la adquisición de conocimientos o destrezas por parte del alumno y la efectividad del diseño del proceso de enseñanza. Proporcionar al alumno ejercicios de autoevaluación.
- **Interacción.** En los entornos de enseñanza basados en la web podemos diferenciar tres niveles de interacción entre: profesor-alumno, alumno-alumno y alumno-contenidos de aprendizaje.

A continuación se dan algunas de las herramientas mas usadas para la creación de multimedia:

- **Adobe Director:** Es una aplicación creada por la Empresa Macromedia la que fue adquirida por Adobe Systems. Permite al usuario crear aplicaciones multimedia tal como si fuera el director de una película controlando sus aspectos, por lo que tiene su nombre muy de acorde a su interfaz. Básicamente se trata de crear una película (Movie). Donde para crear este Movie tiene diversas ventanas como la del reparto de actores (Cast), la de los guiones (Script), otra para el montaje (Score) y la de ver los resultados (Stage). Originalmente diseñado para crear secuencias de animación, pero la adición de un lenguaje de programación script denominado lingo le atribuyen nuevas y mejores funcionalidades logrando incorporar y relacionar imágenes, audios y videos, y diferentes formatos de archivos que permitan la integración de estos medios (*Zambrano Rodriguez, 2008*). Puede extender su funcionalidad a través de los plug-in llamados Xtras los cuales pueden ser desarrollados por el propio usuario. Los tipos de archivos creados por director pueden ser vistos en un navegador que utilice el plug-in Shockwave o al igual que también se pueden crear ejecutables denominados proyectores que pueden ser visualizados sin necesidad de tenerlo instalado.

- **Adobe Flash:** Es una aplicación que trabaja sobre fotogramas en una línea de tiempo destinada a la producción de animaciones y tecnologías multimedia. Anteriormente pertenecía a la Empresa Macromedia la que fue adquirida por Adobe Systems la que ahora se encargada de su distribución. Esta tecnología se a convertido en un popular método de añadir animación, interactividad y enriquecer el contenido de las paginas web (*Zambrano Rodriguez, 2008*). Puede manipular gráficos vectoriales e imágenes de bitmap, flujo de video, audio bidireccional y contiene un lenguaje script de programación orientado a objeto denominado ActionScript tratándose de ver el entorno de programación como el mundo real donde cada objeto tiene sus propiedades como color, forma y ubicación. Los archivos flash tienen generalmente la extensión .swf y pueden ser vistos en un navegador web y algunos teléfonos móviles o reproducidos independientemente por un reproductor flash player que es la máquina virtual utilizada para ejecutar estos archivos, además de que pueden ser exportados a un ejecutable .exe. Entre las características más importantes que tiene flash tenemos que se pueden crear animaciones complejas sin que ocupen mucho espacio en disco, debido al uso de vectores que se modifican fácilmente. Además se puede decir que es un estándar, debido a que lo utilizan más del 95% de los usuarios que acceden a la Web con navegadores desde cualquier sistema operativo, siendo también multiplataforma.

- **Toolbook:** Es una herramienta de desarrollo de aplicaciones Multimedia basadas en un lenguaje orientado a objetos llamado OpenSript, desarrollado en 1985 por una compañía norteamericana llamada Asymetrix. Este sistema de autor permiten diseñar una amplia variedad de actividades de distinta índole, combinando textos, imágenes, sonidos, animaciones, vínculos a sitios web y autoevaluaciones en distintos formatos. Toolbook interpreta el diseño de una multimedia como un libro donde cada una de sus posibles pantallas que contienen información serían las páginas (*Zambrano Rodriguez, 2008*). Además permite asignar un mismo fondo (background) de la aplicación con una misma imagen, color o conjunto de objetos para optimizar los recursos de la aplicación y el sistema.

- **UIRA:** Proyecto creado para ampliar las alternativas de Adobe Flash debido a que estas son muy escasas en Linux. Partiendo del código de Flash4linux (F4l) se fueron rescribiendo varios módulos para solucionar diversos *bugs* y añadirles más funcionalidades. F4l siempre ha tenido problemas a la hora de crear Flash, se pueden hacer animaciones pero todavía tiene grandes dificultades para generar un archivo .swf. Partiendo de ese código se obtuvo QFlash, el mismo puede crear animaciones básicas, añadir textos, polígonos, formas básicas, cuadro de texto, etc. Se llegó a un acuerdo general entre los proyectos de F4l y Qflash y se decidió crear un nuevo proyecto surgiendo de esta forma UIRA. Actualmente el proyecto está en pleno desarrollo, por ahora solo se está trabajando en buscar un diseño óptimo que evite que se tenga que deshacer el trabajo por mal diseño. La parte gráfica se desarrolla con las librerías Qt4 y también usa el motor gráfico Amanith y para el editor ActionScript se usa las librerías cintilla. *(Mestre Dumenigo, 2007)*
- **Ktoon:** Es una herramienta para desarrollar animaciones 2D realizada por animadores de Tonnka Films. Este proyecto está cubierto por licencia GPL usando C++, OpenGL y Qt como recursos de programación desde Kdevelop. Este se divide en dos módulos, el primero es de ilustración y un segundo módulo de animación. El de ilustración permite hacer los componentes gráficos y está muy vinculado a la edición de los componentes. El módulo de animación permite hacer edición pero no de las gráficas, sino del proyecto en general con respecto al tiempo *(Mestre Dumenigo, 2007)*.

1.5.3 Propuesta de herramienta y metodología a utilizar

Como se puede ver, existen diversas herramientas (tanto comerciales como gratuitas) para la creación de entornos de enseñanza-aprendizaje basados en software multimedia. Frente a la proliferación de estas herramientas, es un problema real determinar que herramientas serán más adecuadas para lograr un objetivo educativo específico. Para ello, se debe determinar, por una parte, cuáles son las necesidades y, por otra, cuáles son las posibilidades de las herramientas con las que se dispone.

Después de haberse hecho un análisis de las herramientas mas importantes para el desarrollo de multimedias se escogió como herramienta principal para la creación del producto a Macromedia Flash 8 ya que esta herramienta brinda un grupo de facilidades y posibilidades muy aprovechables desde el punto de vista del diseño, facilidad de uso, portabilidad y rapidez en la creación de productos.

A partir de analizar las diversas metodologías que existen para el desarrollo de multimedias se decidió utilizar RUP aprovechando las posibilidades y la incorporación de los nuevos artefactos como los mapas de navegación y los diagramas de presentación, que nos brinda la extensión de UML para multimedias, conocida como OMMMA-L muy apropiadas para la modelación, análisis y diseño de estos tipos de aplicaciones cuyo objetivo principal es difundir información almacenada en diferentes medios como imágenes, vídeos y sonido.

Capítulo 2. Descripción de la Solución Propuesta

2.1 Introducción

En el presente capítulo o para la descripción del sistema se ha formalizado el trabajo siguiendo la metodología RUP. Realizándose un modelo conceptual de la propuesta al no poderse determinar un proceso de negocio con fronteras bien definidas el cual está compuesto por el diagrama de clases del **modelo de dominio**, para lograr una mejor comprensión por parte de los usuarios, clientes y desarrolladores, de los conceptos dentro del contexto en donde esta enmarcado el sistema, especificando las principales clases conceptuales que pueden intervenir, estos representarán los objetos que existen o eventos que suceden en el entorno en el que trabajará el sistema. Del mismo modo se muestran los **mapas de navegación** del sistema, nuevo artefacto que incorpora OMMMA-L a RUP como extensión de UML para el desarrollo de aplicaciones multimedia para tener una estructura o visión global del sistema compuesto por nodos, que son las diferentes pantallas de la aplicación y sus relaciones que indican la posibilidad de navegación entre un nodo y otro. Se presenta también la **descripción de funcionalidad**, ya sean requisitos funcionales como no funcionales que permitirá definir que es lo que el sistema debe hacer. Además se presenta el **modelo de casos de uso del sistema**, que de forma análoga a un sistema tradicional, los casos de uso identifican procesos dentro del desarrollo del software que son generados por un actor u otros casos de uso y describen el flujo de acciones a ocurrir durante el tiempo de vida en que esta en ejecución la aplicación o de sus procesos o subprocesos internos, lo que interpretados desde la visión o punto de vista multimedia, estos casos de usos están orientados a las acciones que ocurren durante la modificación del comportamiento interactivo del sistema o dentro de él.

2.2 Especificación del contenido curso Introducción a la Calidad de Software

El contenido que se abordará dentro del curso introducción a la Calidad de Software esta dividido en 5 temas fundamentales, un juego y un mapa conceptual sobre cómo es el estudio y lo que aporta la Calidad de Software. Cada uno de los temas está dividido en varios epígrafes.

El Tema 1 se introducirá en lo qué es la calidad de software de manera general desde sus inicios centrándose en los siguientes epígrafes:

- Evolución Histórica. Calidad Total.
- Conceptos de Calidad de Software

El Tema 2 tiene como objetivo dar una panorámica de que es lo que abarca y trata el Perfil de Software en la UCI centrándose en su:

- Importancia y principales características.
- Asignaturas que deben ser cursadas por año explicación general de cada una de ellas.

El Tema 3 tiene como objetivo profundizar en el estudio del Aseguramiento de calidad de software centrándose en:

- El Plan de aseguramiento de calidad de software.
- El Proceso de mejoramiento continuo de la calidad.

El Tema 4 tiene como objetivo hacer un análisis de la calidad de proceso y la calidad del producto centrándose en:

- La Calidad aplicada al desarrollo de software.
- La Verificación vs. Validación.

El Tema 5 tiene como objetivo dar una introducción a los modelos de Calidad ISO, CMMI y SPICE centrándose en sus:

- Características generales y estado actual.
- Una profundización en CMMI.

2.2.1 Mapa Conceptual de Calidad de Software

A continuación en la *Fig. 3* se mostrará el mapa conceptual referente a lo que se estudia en el curso de Introducción a la Calidad de Software.

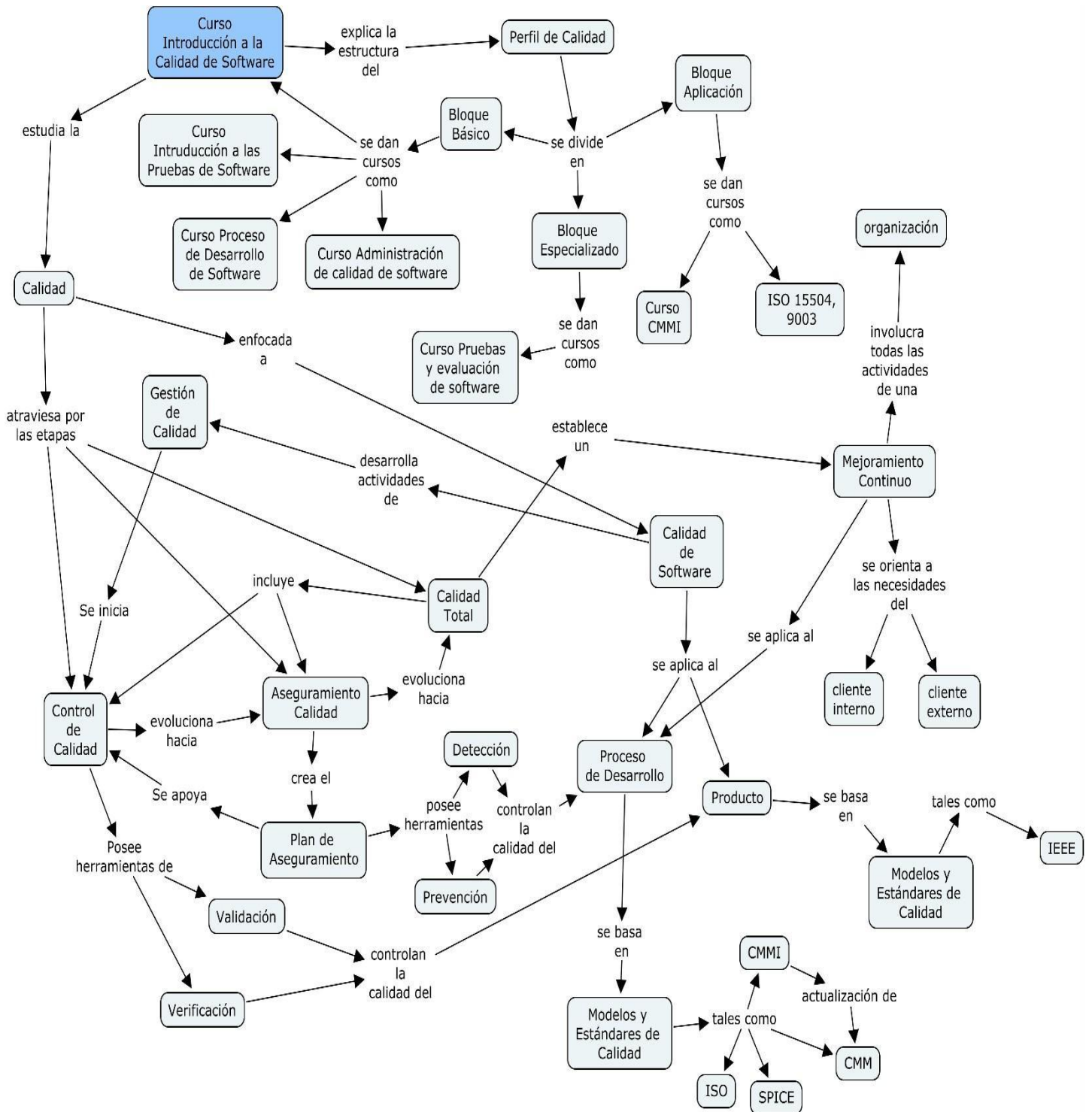


Fig. 3 Mapa Conceptual del curso Introducción a la Calidad de Software.

2.3 Especificación del contenido curso PDS

El contenido que se abordará dentro del curso PDS esta dividido en 4 temas fundamentales, un juego y un mapa conceptual sobre cómo es el estudio y lo que aporta el PDS. Cada uno de los temas esta dividido en varios epígrafes.

El tema 1 tiene como objetivo dar una introducción a lo que es el PDS, estando centrado en los siguientes epígrafes:

- El proceso de desarrollo del software.
- Modelos de proceso software.
- Metodologías para desarrollo de software.

El tema 2 tiene como objetivo profundizar en el estudio de las metodologías Tradicionales haciéndose énfasis en la metodología tradicional RUP centrándose en sus:

- Características Fundamentales.
- Fases y Flujo de trabajo.
- Herramientas Rational.

El tema 3 tiene como objetivo profundizar en el estudio de las metodologías ágiles haciéndose énfasis en la metodología ágil Extreme Programming (XP) centrándose en sus:

- Prácticas Básicas y Actividades.
- Ciclo de vida.
- Actores y Artefactos.

El tema 4 tiene como objetivo un estudio sobre Microsoft Solución Framework centrándose en los siguientes epígrafes.

- El modelo de proceso de Microsoft.
- El modelo de equipo de proyecto de Microsoft.
- Las disciplinas de MSF.

2.3.1 Mapa Conceptual de PDS

A continuación en la *Fig. 4* se mostrará el mapa conceptual, referente a lo que se estudia en un el curso de PDS.

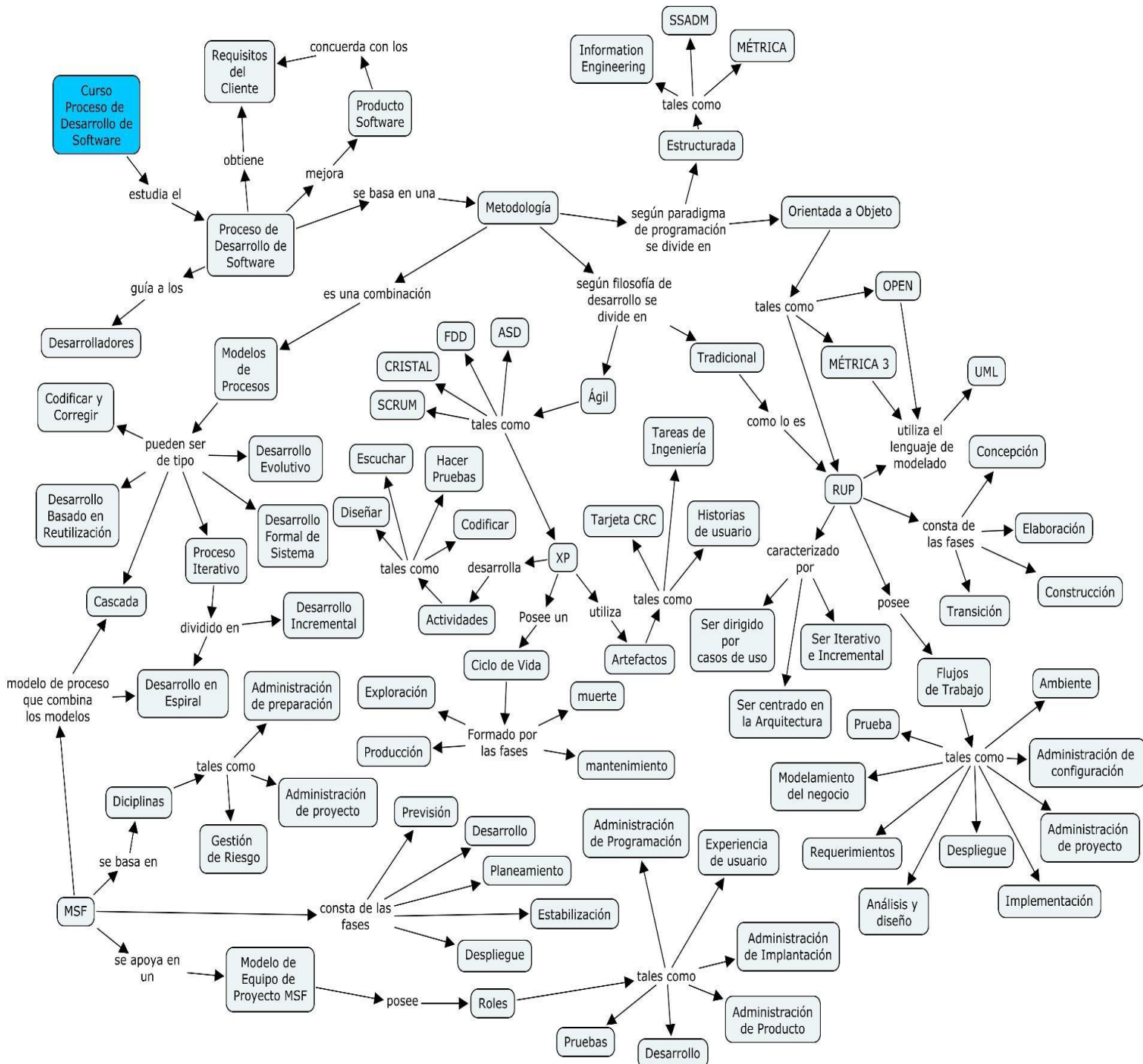


Fig. 4 Mapa Conceptual del curso PDS.

2.4 Identificación de la Audiencia

El producto multimedia de apoyo al bloque básico del perfil de calidad, va dirigido fundamentalmente a los estudiantes de primero y segundo año que tienen muy poca o ninguna idea de lo que es calidad de software y PDS, ya que estos cursos no están dentro del plan de estudio de ninguno de los niveles inferiores y que dan acceso al ingreso a la universidad. Estos cursos son los primeros que se imparten y de obligatorio conocimiento de los estudiantes, para entender los principios esenciales de calidad de software y recibir acreditación de este perfil por parte de la UCI una vez graduados.

2.5 Modelo Conceptual

Se realiza un modelo conceptual de la propuesta al no poderse determinar un proceso de negocio con fronteras bien definidas compuesto por el modelo de dominio. Identificándose las principales clases conceptuales que pueden intervenir, estos representarán los objetos que existen o eventos que suceden en el entorno en el que trabajará el sistema.

2.5.1 Modelo de Dominio General

A continuación en la *Fig. 5* se muestra el modelo de dominio, para lograr una mejor comprensión por parte de los usuarios, clientes y desarrolladores, de los conceptos dentro del contexto en donde esta enmarcado el sistema.

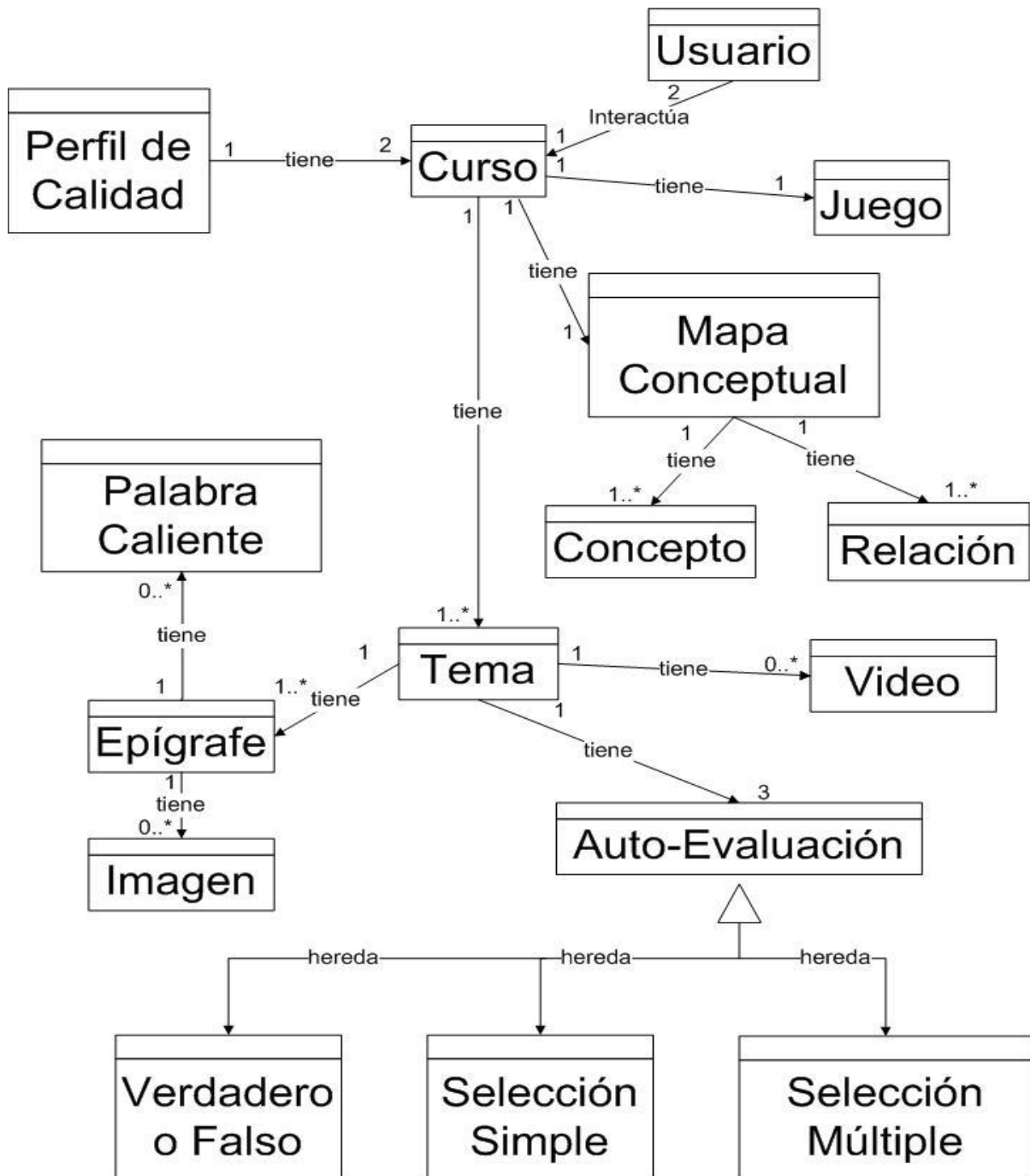


Fig. 5 Modelo de Dominio General

2.5.2 Análisis de los Conceptos del Dominio

A continuación se hace un análisis referente al significado que tiene cada concepto del modelo de dominio dentro del contexto donde el sistema está enmarcado.

Perfil de Calidad: Línea de formación profesional del ingeniero informático.

Usuario: Se le denomina a cualquier persona que interactúa con el sistema.

Curso: Asignatura que se incorporara a la aplicación, relacionada con el perfil.

Tema: Contenido que se trata en una asignatura.

Epígrafe: Contenido que se trata en un tema.

Video: Video conferencia sobre un tema en particular.

Autoevaluación: Medición de los conocimientos adquiridos en un tema.

Imagen: Gráfico o Imagen referente a un epígrafe.

Palabra Caliente: hipertexto que abre una ventana flotante que profundiza en el significado de alguna palabra o concepto tratado en un epígrafe.

Juego: Entretenimiento en la búsqueda de palabras o conceptos que se tratan en un curso.

Mapa Conceptual: Mapa estructural de conceptos relacionados entre sí, referente a lo que se estudia en un determinado Curso.

Concepto: Significado de un determinado concepto del mapa conceptual.

Relación: Tipo de relación existente entre un concepto y otro.

2.6 Descripción de la Funcionalidad

Con la descripción de la funcionalidad se permitirá definir qué es lo que el sistema debe hacer.

2.6.1 Requisitos Funcionales

Los requisitos funcionales recogidos en la *Tabla 1* son los parámetros que tiene que cumplir el software para su correcto funcionamiento.

Tabla 1. Requisitos Funcionales

Ref. #	Función
R1	Mostrar presentación del producto.
R2	Permitir saltar la presentación del Producto.
R3	Mostrar pantalla principal del programa, donde el usuario debe recibir información general acerca de las especificaciones y objetivos del perfil.
R4	Mostrar menú de cursos, el cual al pasarle con el cursor del mouse por encima de cada botón de curso debe mostrarse un tooltip con el título del curso, sus especificaciones y objetivos.
R5	Reproducir música de fondo automática de forma predeterminada.
R6	Permitir subir volumen, bajar volumen, parar y reanudar sonido de fondo.
R7	Permitir reiniciar la aplicación mostrando de nuevo la presentación del producto.
R8	Permitir escoger un curso.
R9	Permitir salir del sistema posterior a una confirmación.
R10	Mostrar créditos del producto.
R11	Mostrar pantalla principal del curso, donde el usuario debe recibir información sobre la definición del concepto de lo que trata el curso.
R12	Mostrar menú de temas, mapa conceptual y juego del curso, el cual al pasarle con el cursor del mouse por encima a cada botón debe mostrar un tooltip con el título del tema e información de lo que se aborda en el mismo.
R13	Permitir navegación hacia otro curso.
R14	Mostrar un seguimiento de la navegación dentro del curso.
R15	Mostrar pantalla principal del tema donde el sistema muestra automáticamente información del primer epígrafe del tema.
R16	Mostrar localización de los demás epígrafes.
R17	Permitir mostrar pantalla principal del curso.
R18	Permitir mostrar pantalla principal de los videos.
R19	Permitir mostrar pantalla principal de las autoevaluaciones.
R20	Permitir mostrar un epígrafe del tema.

Ref. #	Función
R21	Mostrar el contenido de un epígrafe que puede estar enriquecido con palabras calientes e imágenes, donde el sistema debe colorear de una manera distinta las palabras calientes al contenido texto del epígrafe.
R22	Permitir mostrar contenido de una palabra caliente en ventana flotante.
R23	Mostrar pantalla principal de autoevaluación, donde el sistema muestra automáticamente la primera autoevaluación.
R24	Permitir responder pregunta
R25	Mostrar mensaje de respuesta correcta.
R26	Mostrar mensaje de respuesta incorrecta.
R27	Permitir avanzar hacia otra pregunta, donde en caso de no hacerse por el usuario el sistema avanza automáticamente a la próxima autoevaluación pasado un breve tiempo o muestra la pantalla principal del tema si es la última autoevaluación.
R28	Mostrar pantalla principal de los videos, donde el sistema reproduce de manera automática el primer video de la lista de videos, en caso de no existir videos el sistema muestra el mensaje correspondiente y pasado un tiempo breve muestra la pantalla principal del tema.
R29	Permitir subir volumen, bajar volumen, parar, reanudar, pausar, desplazar, ver siguiente, ver anterior, poner a pantalla completa y restablecer la pantalla de los videos.
R30	Mostrar un listado con los videos del tema.
R31	Mostrar título del video en reproducción.
R32	Mostrar pantalla principal del Juego.
R33	Permitir jugar.
R34	Permitir ver solución del juego.
R35	Mostrar mensaje de felicitación al ganar el juego.
R36	Mostrar pantalla principal del mapa conceptual.
R37	Permitir ampliar tamaño de mapa conceptual.
R38	Permitir disminuir tamaño de mapa conceptual.
R39	Permitir mostrar contenido de un concepto en ventana flotante
R40	Permitir mover de lugar el mapa conceptual.

2.6.2 Requisitos no Funcionales

Los siguientes requisitos no funcionales o implícitos que a menudo no se mencionan son con los que el software debe cumplir para alargar su vida útil y lograr una mejor aceptación por parte del usuario.

Sistema Operativo

El archivo con extensión .swf que se genera de cualquier aplicación hecha con Macromedia Flash es independiente del sistema, debido a que es un formato estándar que soporta la mayoría de los navegadores web, el mismo deberá ejecutarse en cualquier sistema operativo sin problemas, solamente instalándose el plug-in del Flash Player 8 para poder soportarlo.

Resolución de pantalla y profundidad de colores

En el producto no se imponen los requisitos de resolución de pantalla, cursores o profundidad de colores, si no que este utiliza la configuración existente en la PC antes de entrar al programa, sin embargo, la resolución para una óptima visualización del producto debe estar encuadrada en 800 x 600 pixeles con una profundidad de color de 32 bit.

Apariencia e interfaz externa

El producto debe tener una interfaz amigable con un sencillo diseño y de fácil manejo por parte de los usuarios para brindarles todas las facilidades necesarias y garantizar que su uso sea lo mas cómodo y agradable posible.

Todos los objetos interactivos del producto tendrán tooltips a manera de mostrar su utilidad y cambios de cursos en su evento onRollOver.

El producto debe funcionar tanto desde un CD, un disco duro (HDD), una memoria flash y un navegador Web siempre que el mismo esté publicado en algún servidor web remoto.

El producto debe utilizar botones que expresan su función ya sea con textos, con tooltip o de manera intuitiva.

El color de la letra debe estar en con contraste con el fondo del área de texto para facilitar su lectura.

El idioma del producto debe ser en su totalidad español con excepción de algunas palabras técnicas.

Implementación

Todos los textos de los epígrafes, tooltips de botones de menú, imágenes, videos, y parámetros del juego deben cargarse de manera dinámica pudiéndose cambiar su información.

Los datos se almacenarán usando ficheros XML por su fácil manejo y edición evitándose así el uso de un servidor o gestor de base de datos.

La herramienta de diseño gráfico e implementación de la aplicación debe ser Macromedia Flash 8.0, Macromedia Firewall 8.0 para el trabajo con las imágenes, Adobe Premier y Macromedia Flash Video Encoder 8.0 para la edición y codificación de los videos.

Se requiere del reproductor Flash Player 8.0 o superior para la visualización del producto.

Como lenguaje de programación para la implementación de la aplicación se utilizará ActionScript 2.0. y el lenguaje de etiquetas XML para el almacenamiento del contenido de texto y algunas configuraciones.

Para la modelación del sistema se utilizará la herramienta CASE Rational Rose Enterprise Edition 2003 y Microsoft Office Visio 2007.

Usabilidad

El producto debe permitir el acceso de cualquier usuario y lograr una interfaz y navegación funcionales ya que ha sido diseñado con ese propósito, permitiendo el acceso a personal tanto experto como el de conocimientos básicos en informática. Pudiendo cualquier usuario interactuar con el producto sin presentar dificultades en la asimilación de la herramienta.

Hardware

Para la utilización del producto por parte de los usuarios se requiere como mínimo una computadora con las siguientes características:

- Procesador Pentium 3 a 800 MHz de velocidad de procesamiento o superior.
- 128 MB de RAM o superior.
- Espacio de 1 GB disponible en el disco duro.
- Periféricos: teclado, mouse, lector CD, audífonos o speaker y monitor.

2.7 Modelo de Caso de Uso del Sistema

Usado para identificar procesos dentro del desarrollo del software que son generados por un actor u otros casos de uso y describen el flujo de acciones a ocurrir durante el tiempo de vida en que esta en ejecución la aplicación o de sus procesos o subprocessos internos. Desde el punto de vista multimedia, están orientados a las acciones que ocurren durante la modificación del comportamiento interactivo del sistema.

2.7.1 Diagrama de Casos de Uso

En la Fig. 6 se muestra el diagrama de casos de uso que identifica los procesos y darán solución al sistema propuesto.

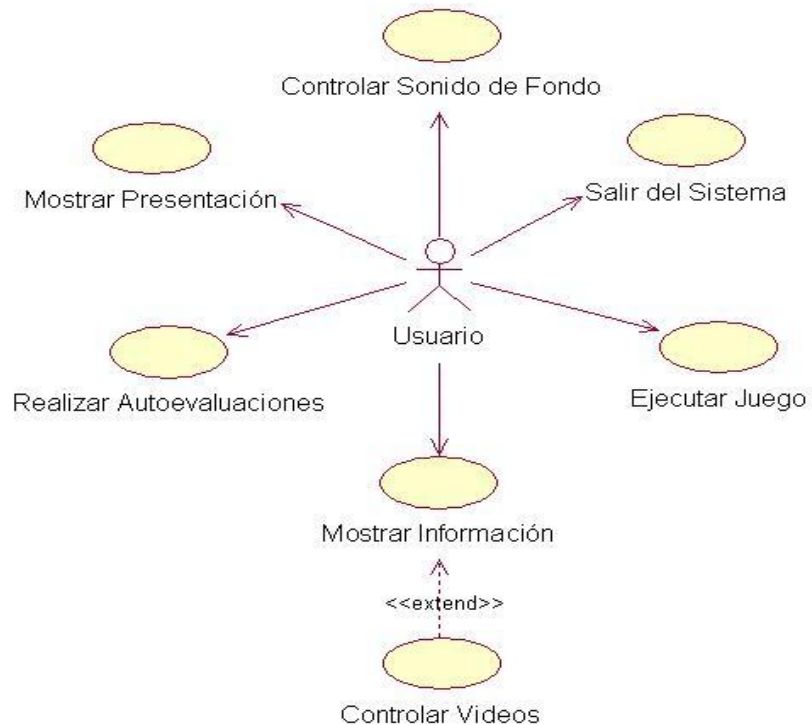


Fig.6 Diagrama de Casos de Uso.

2.7.2 Determinación y justificación de los Actores del Sistema.

A continuación en la *Tabla 2* se justifica la interacción del actor usuario con los casos de uso del sistema.

Tabla 2. Justificación de los Actores del Sistema.

Actor	Justificación
Usuario	Representa a una persona que va a utilizar el sistema para buscar información sobre los cursos de introducción a la calidad y PDS.

2.7.3 Casos de Uso

La *Tabla 3* contiene una relación de los casos de uso y su prioridad dentro del sistema.

Tabla 3. Prioridad de los Casos de Uso del Sistema.

CUS #	Nombre	Prioridad
CUS 1	Mostrar Presentación	Secundario
CUS 2	Controlar Sonido	Secundario
CUS 3	Salir del Sistema	Crítico
CUS 4	Mostrar Información	Crítico
CUS 5	Controlar Videos	Secundario
CUS 6	Realizar Autoevaluaciones	Crítico
CUS 7	Ejecutar Juego	Crítico

2.7.4 Descripción de Casos de Uso del Sistema

En las tablas que aparecen a continuación se hace un breve resumen de las acciones que ocurren dentro de cada caso de uso que interviene en el sistema.

Tabla 4. Resumen del Caso de Uso Cargar Presentación.

Caso de Uso:	Cargar Presentación
Actor:	Usuario
Propósito:	Mostrar la presentación de la multimedia.
Resumen:	Al ejecutar la aplicación el usuario, el sistema carga la presentación del producto, donde será opcional su visualización, la que puede ser interrumpida en cualquier instante dando clic en el botón saltar presentación. Al concluir la presentación de la aplicación se pasa automáticamente a la pantalla principal del producto.
Referencias:	R1,R2

Tabla 5. Resumen del Caso de Uso Controlar Sonido Fondo.

Caso de Uso:	Controlar Sonido Fondo
Actor:	Usuario
Propósito:	Controlar la reproducción de sonido de fondo de la multimedia.
Resumen:	La reproducción de sonido de fondo comienza de manera automática al concluir el caso de uso cargar presentación. Pero puede ser controlada por el usuario, en cualquier momento en que esté en ejecución la aplicación y el usuario desee subir volumen, bajar volumen, parar y reanudar el sonido de fondo.
Referencias:	R5,R6

Tabla 6. Resumen del Caso de Uso Salir del sistema.

Caso de Uso:	Salir del sistema
Actor:	Usuario
Propósito:	Cerrar la multimedia después de haberse mostrado los créditos.
Resumen:	Al usuario decidir salir del sistema, se muestra una pantalla de confirmación de salida. En caso de ser confirmada se muestran los créditos y luego se cierra la aplicación, de no confirmarse, se vuelve a la pantalla donde se encontraba anteriormente.
Referencias:	R9, R10

Tabla 7. Resumen del Caso de Uso Mostrar Información.

Caso de Uso:	Mostrar Información
Actor:	Usuario
Propósito:	Mostrar la información que solicite el usuario.
Resumen:	Es iniciado el caso de uso al mostrarse la información contenida en la pantalla principal de la aplicación, también cuando el usuario solicita acceder a la información de los cursos que ofrece la multimedia y al contenido de los diferentes temas, videos, mapa conceptual, conceptos, epígrafes y palabras calientes que tienen estos cursos, encargándose el sistema de obtener y mostrar la información solicitada.
Referencias:	R7, R8, R9, R11, R12, R13, R14, R15, R16, R17, R18, R19, R20, R21, R22, R28, R30, R31, R36, R37, R38, R39, R40.

Tabla 8. Resumen del Caso de Uso Controlar Videos.

Caso de Uso:	Controlar Videos
Actor:	Usuario
Propósito:	Controlar la reproducción de los videos de la multimedia.
Resumen:	La reproducción del los videos comienza de manera automática con el primer video de la lista al usuario acceder al modulo videos de un tema. Pero esta reproducción puede ser controlada por el usuario pudiendo subir volumen, bajar volumen, parar, reanudar, pausar, desplazar, ver siguiente, ver anterior, poner a pantalla completa y restablecer la pantalla de los videos.
Referencias:	R29, R31

Tabla 9. Resumen del Caso de Uso Realizar Autoevaluaciones.

Caso de Uso:	Realizar Autoevaluaciones
Actor:	Usuario
Propósito:	Auto medir el conocimiento adquirido en un determinado tema.
Resumen:	Primeramente se muestra de manera automática la autoevaluación verdadero o falso al usuario acceder al modulo autoevaluación de un tema. Una vez respondida la pregunta se pasa de manera automática o controlada por el usuario a la siguiente que es de selección múltiple, culminando este caso de uso con una pregunta de selección simple.
Referencias:	R7, R9, R13, R14, R17, R18, R20, R23, R24, R25, R26, R27

Tabla 10. Resumen del Caso de Uso Ejecutar Juego.

Caso de Uso:	Ejecutar Juego.
Actor:	Usuario
Propósito:	Entretenerse en la búsqueda de palabras y conceptos relacionados con un determinado curso.
Resumen:	Juego de sopa de letras compuesta de una caja de letras de 14 x 14 letras donde el usuario se entretiene en localizar un conjunto de 6 palabras dadas.
Referencias:	R7, R9, R17, R32, R33, R34, 35.

2.8 Mapas de Navegación

En las figuras que aparecen a continuación se presentan los mapas de navegación para tener una estructura o visión global del sistema compuesto por nodos, que son las diferentes pantallas de la aplicación y sus relaciones que indican la posibilidad de navegación entre un nodo y otro.

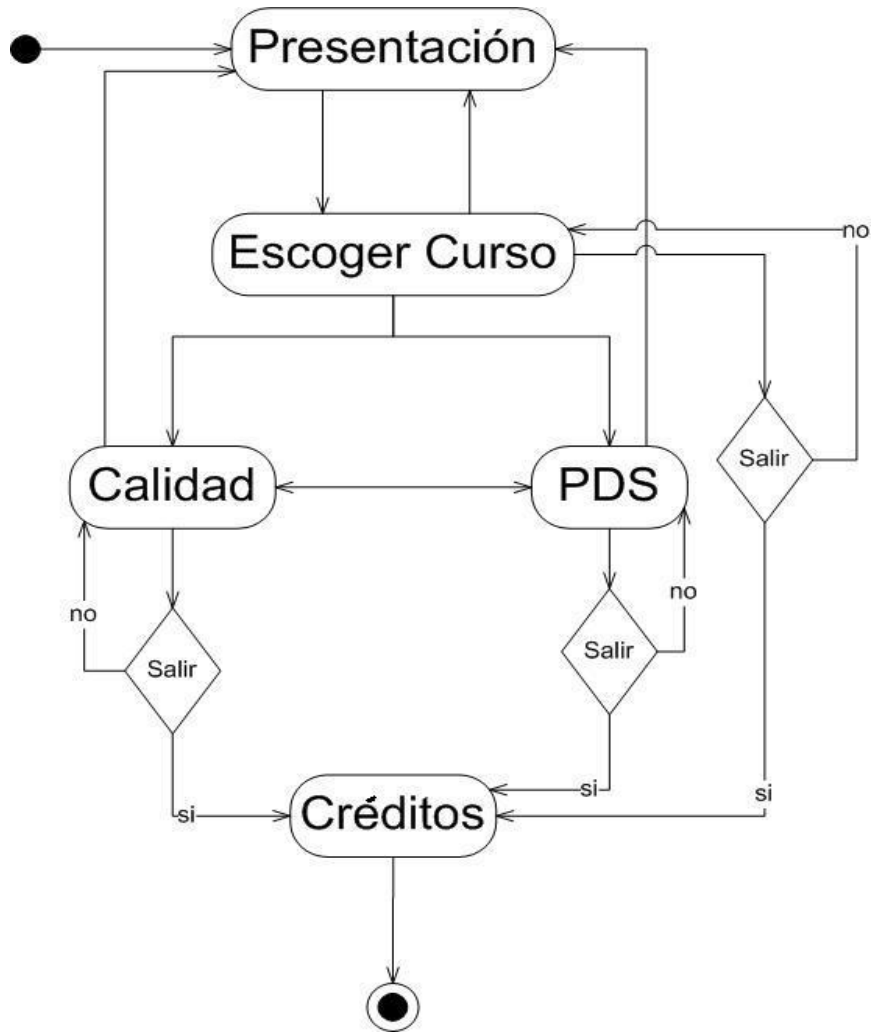


Fig. 7 Mapa de navegación general.

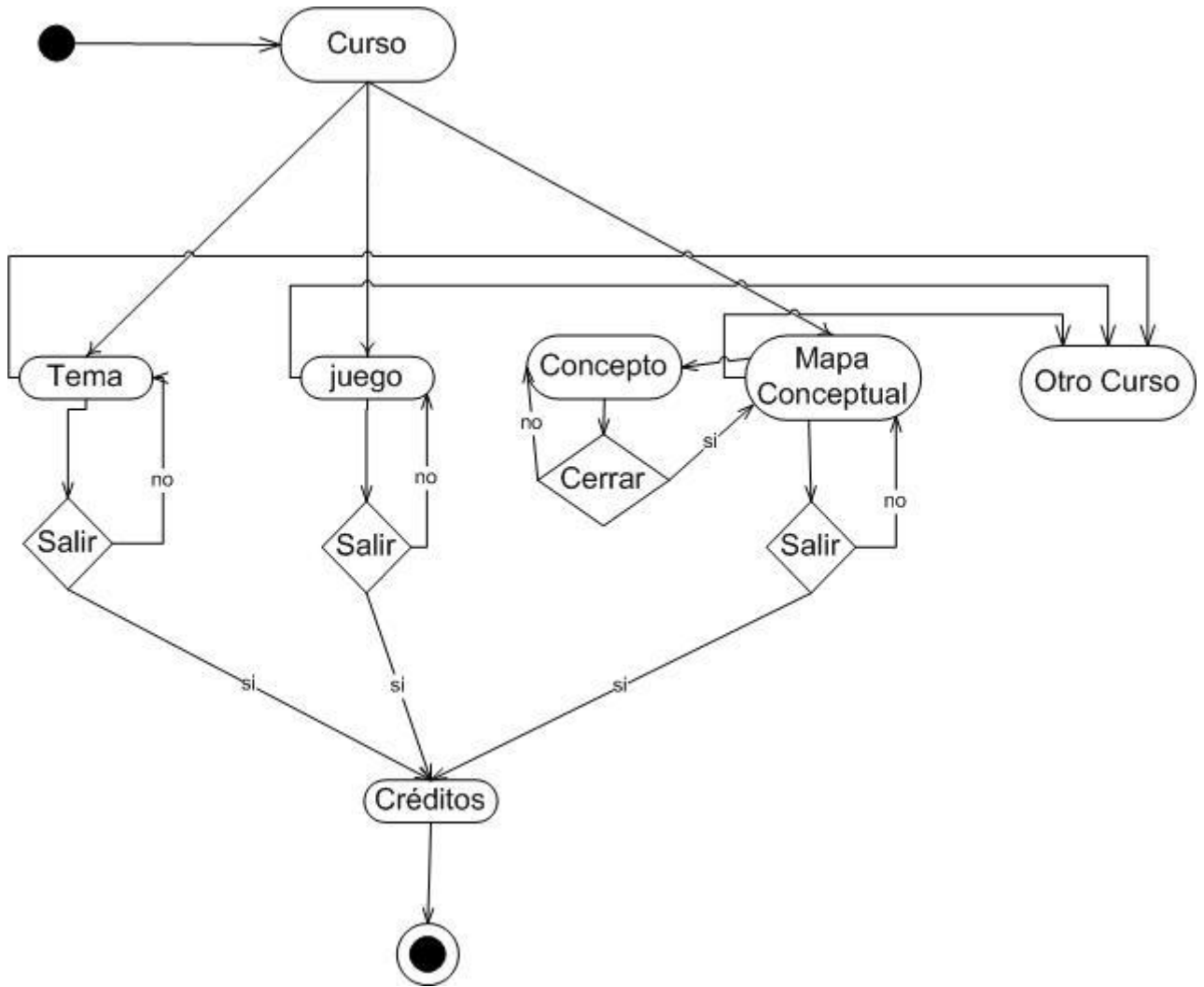


Fig.8 Mapa de navegación del módulo Curso.

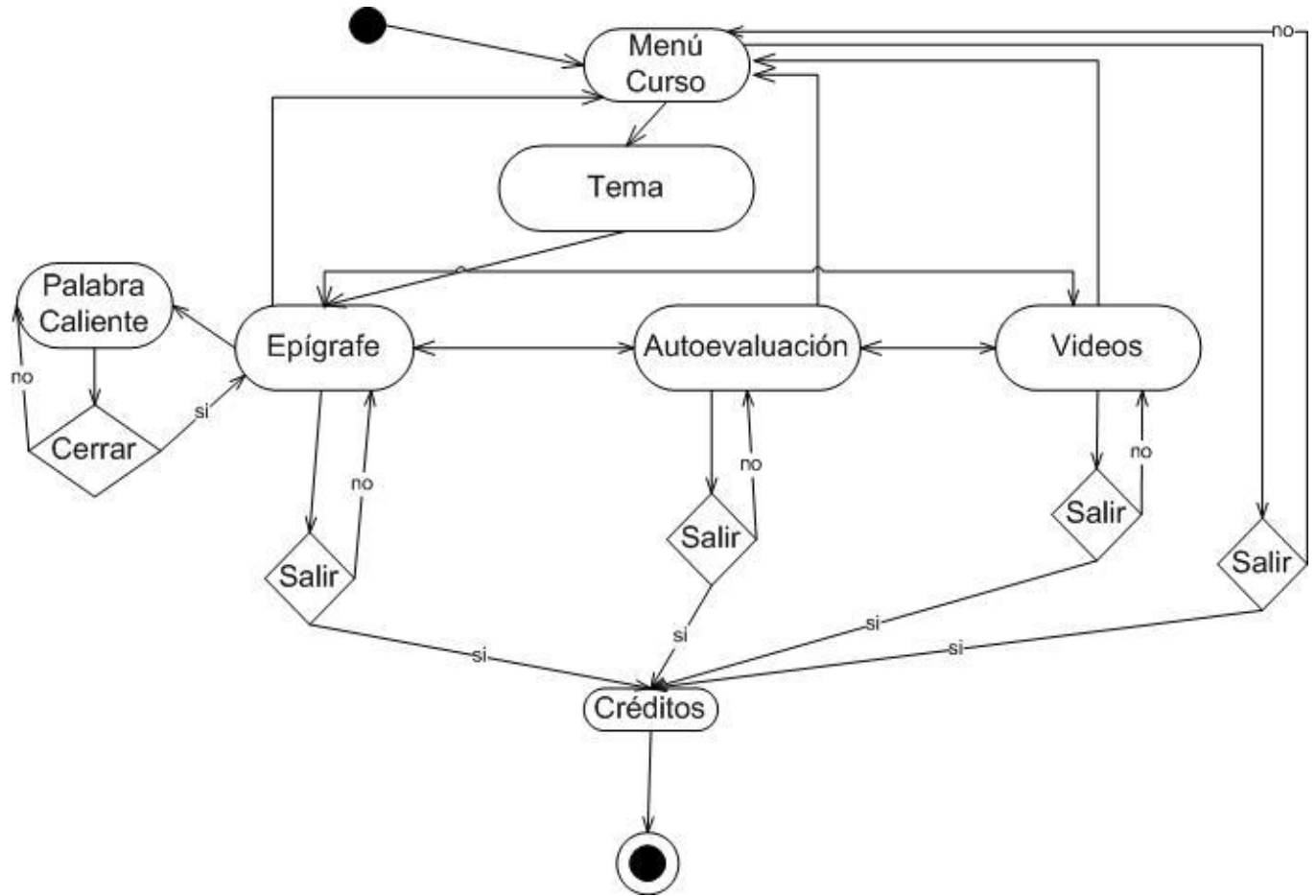


Fig. 9 Mapa de navegación del módulo tema.

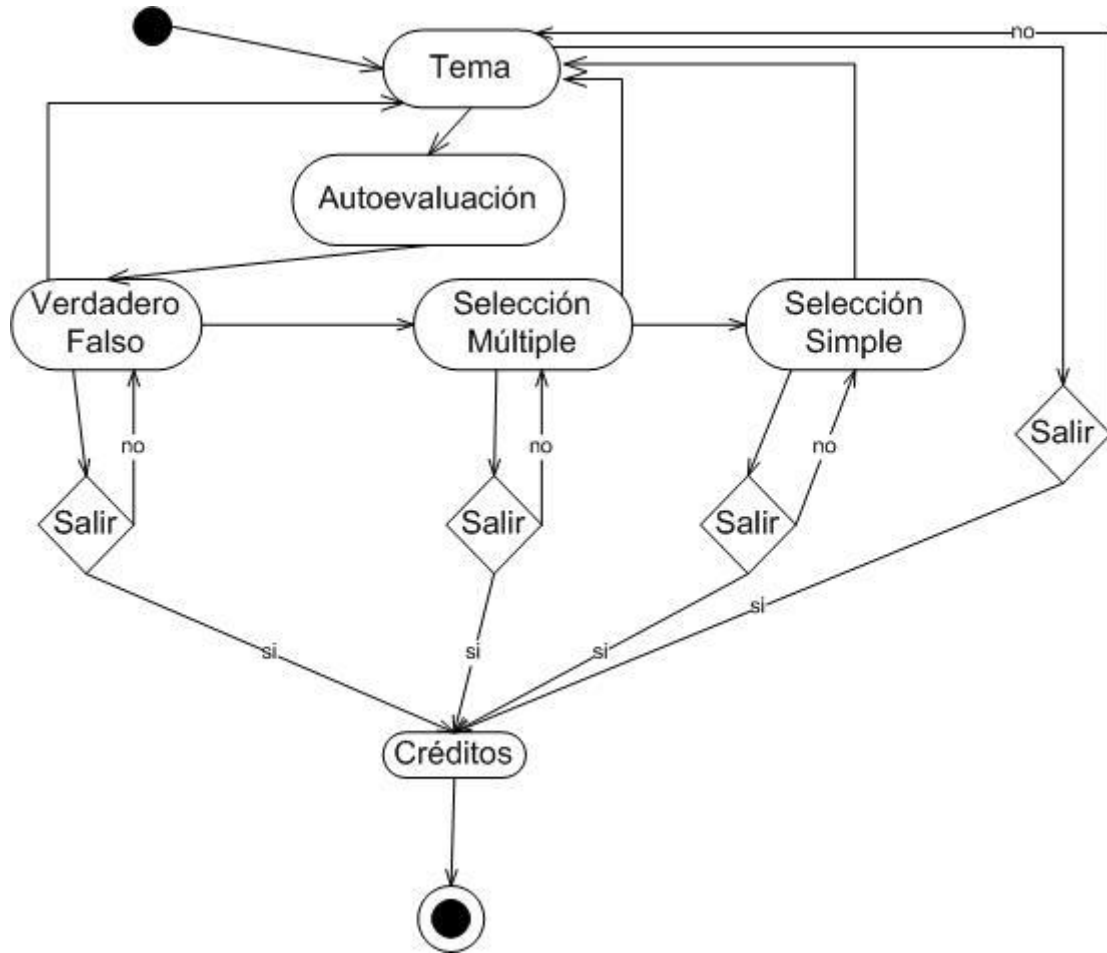


Fig. 10 Mapa de navegación del módulo autoevaluación.

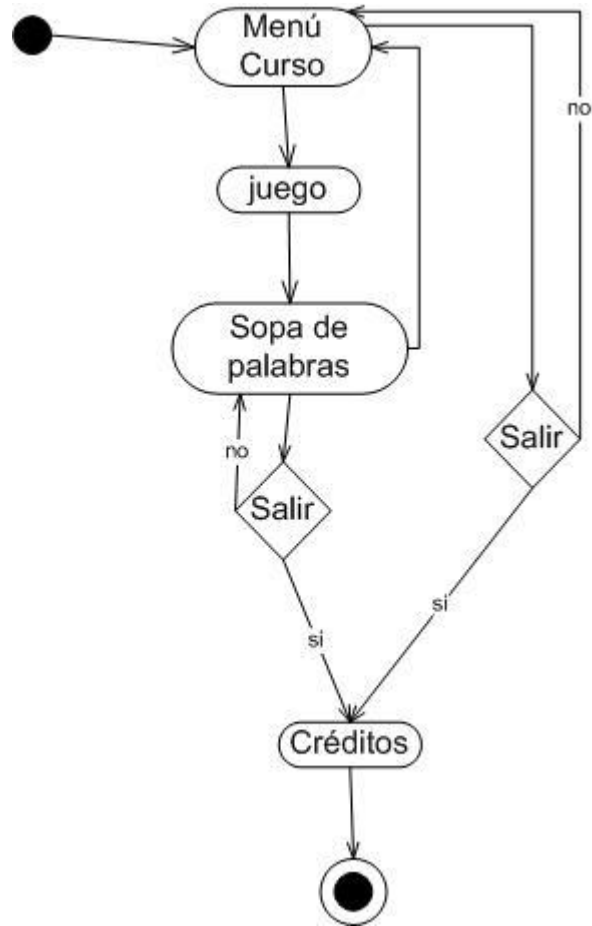


Fig. 11 *Mapa de navegación del modulo juego.*

Capítulo 3. Construcción de la Solución Propuesta

3.1 Introducción

En el presente capítulo se describe la construcción de la solución propuesta mediante la realización de los **diagramas de presentación** nuevo artefacto incorporado dentro del lenguaje UML a partir de la extensión del mismo para aplicaciones multimedia, planteada por OMMMA-L que sirve para describir y definir la parte estática a través de una distribución espacial de objetos visuales de la interfaz de usuario, como un conjunto de estructuras delimitadas en tamaño y espacio, dividiéndose en objetos de visualización (texto, imágenes, video, etc.) e interacción (barras de menú, botones, hipertextos con hipervínculos, etc.). Además se implementa el sistema en términos de componentes, representando el **diagrama de componentes**: separación del sistema en componentes físicos, es decir, ficheros de código fuente, ficheros de código binario, módulos y paquetes; y cómo estos se organizan y acoplan de acuerdo a los nodos específicos en el **modelo de despliegue**: despliega los componentes a construir entre los diferentes nodos físicos donde funcionará la aplicación como pudieran ser maquinas servidoras y maquinas clientes. Por último también se hace una **descripción de los principales XML**: utilizados para guardar algunas configuraciones y el contenido, garantizándose así la independencia de los datos mostrados en la aplicación haciéndose lectura del contenido que se necesita mostrar en cada momento.

3.2 Principios de Diseño

En términos de diseño se ha llegado a una primera propuesta de solución. La que puede estar sujeta a cambios en una etapa posterior de su desarrollo atendida por especialistas de diseño.

El diseño del producto se apropia de rasgos que caracterizan a la propia actividad de calidad de software como son la eficiencia, la confiabilidad, y la seguridad. Por lo que se ha utilizado una pauta cromática que comprende la gama de azules y grises. La selección de los azules se ha hecho debido al carácter corporativo que este comunica, pudiendo decirse además que es un color elegante, serio, que ayuda a comunicar tecnología de informática y con el cual se identifica la universidad, al igual que los grises que es un color mucho más serio y frío que acentúa las nuevas tendencias tecnológicas contemporáneas mientras que el blanco y el rojo se han utilizado para resaltar y enfatizar zonas de importancia.

La tipografía se ha seleccionado teniendo en cuenta el soporte en que va hacer mostrado el producto. Utilizándose la verdana que es una tipografía de tipo sans serif caracterizada por ser rectas de terminaciones limpias, claras y legibles muy apropiadas para ser visualizada en cualquier soporte digital como son pantallas, monitores y proyecciones data show. Además es una tipografía que connota tecnología y modernidad la cual se ajusta al término fundamental que es la calidad de software.

La distribución de los elementos dentro de la interfaz gráfica del producto se ha hecho teniendo en cuenta una jerarquización de la información que se ha querido mostrar. En la misma se ha utilizado elementos en planos superiores e inferiores y otros centrados en cuadros y mostrados en ventanas flotantes que contienen la información necesaria para evitar una sobrecarga, al igual que se mantienen las principales opciones en el mismo lugar para una mejor interacción y adaptabilidad del usuario con la aplicación.

3.3 Diagramas de Presentación

En las figuras que aparecen a continuación se muestran las principales pantallas con que cuenta la aplicación que sirven para describir y definir la parte estática del sistema a través de una distribución espacial de objetos visuales de la interfaz de usuario, como un conjunto de estructuras delimitadas en tamaño y espacio, dividiéndose en objetos de visualización (texto, imágenes, video, etc.) e interacción (barras de menú, botones, hipertextos con hipervínculos, etc.).

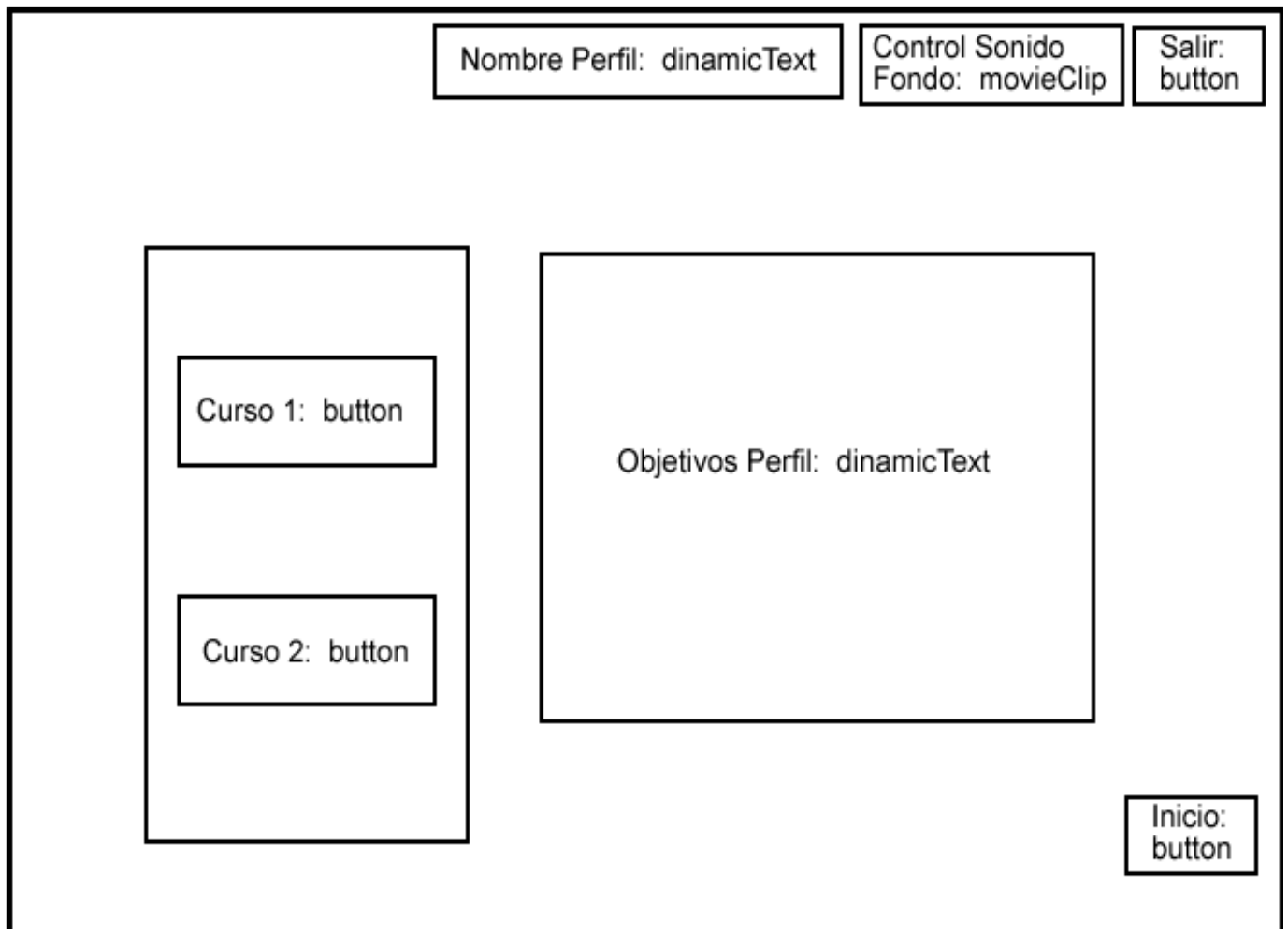


Fig. 12 Pantalla principal de la aplicación.

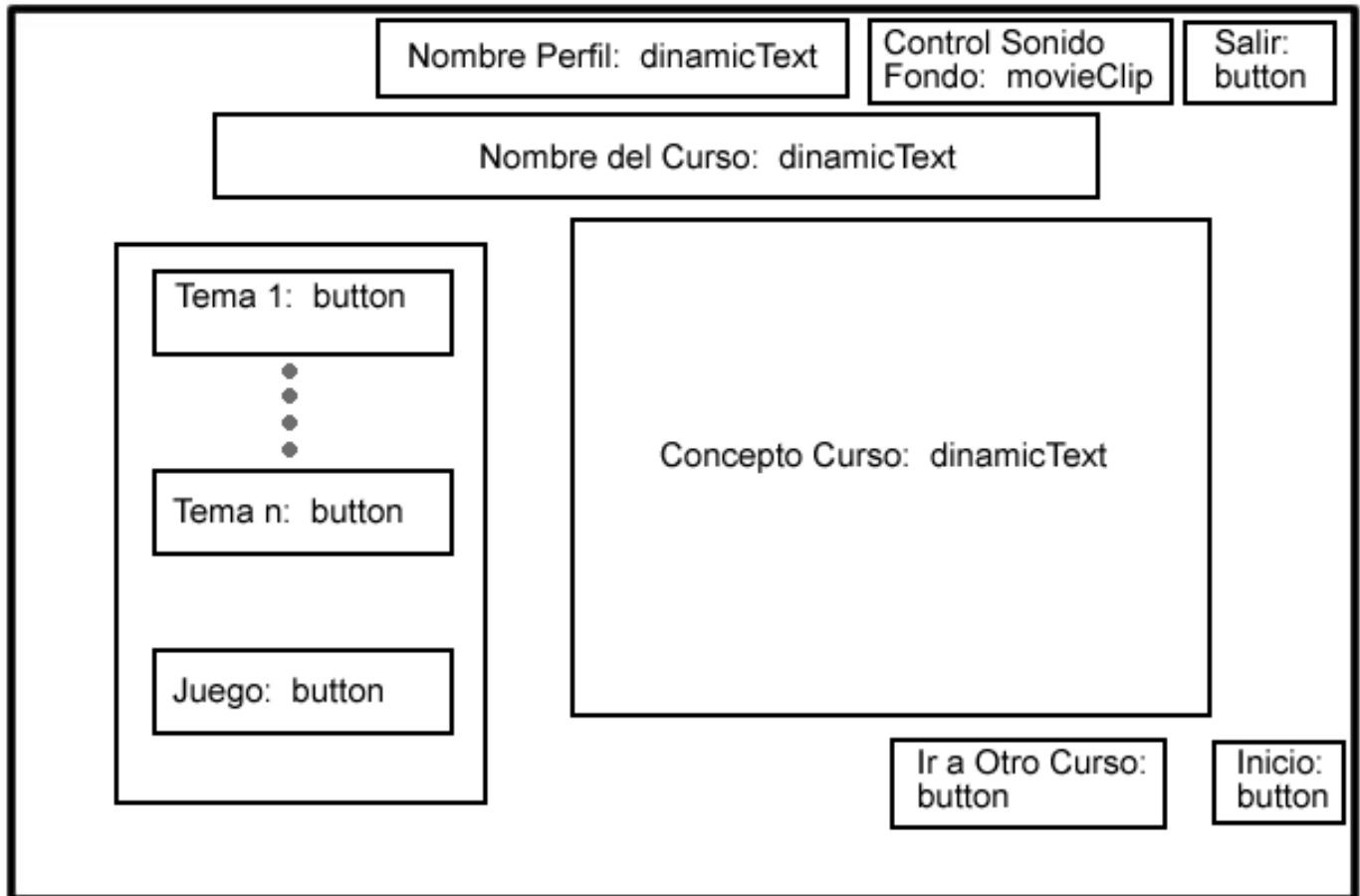


Fig. 13 *Pantalla principal de un curso.*

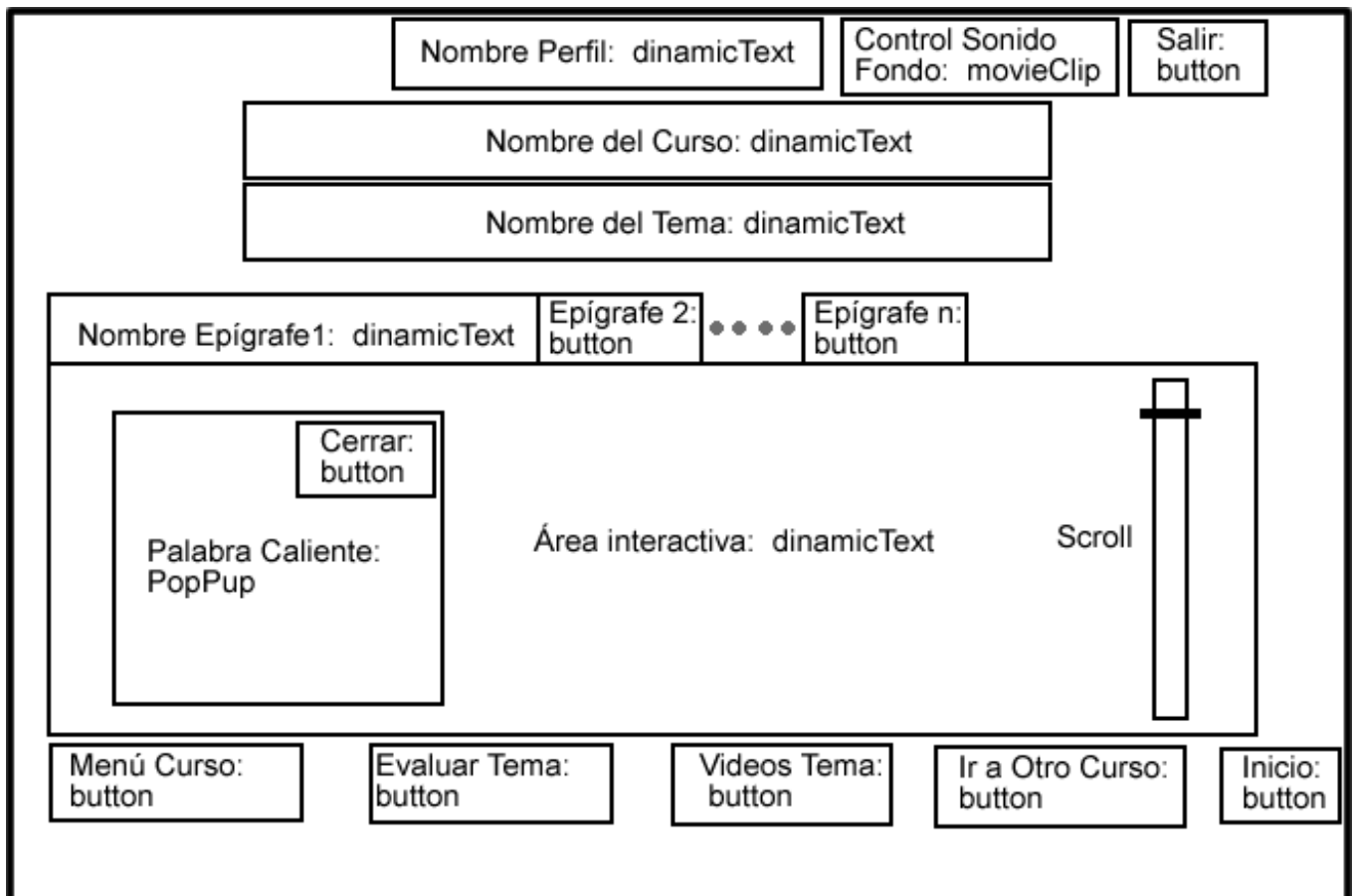


Fig. 14 *Pantalla principal de un tema.*

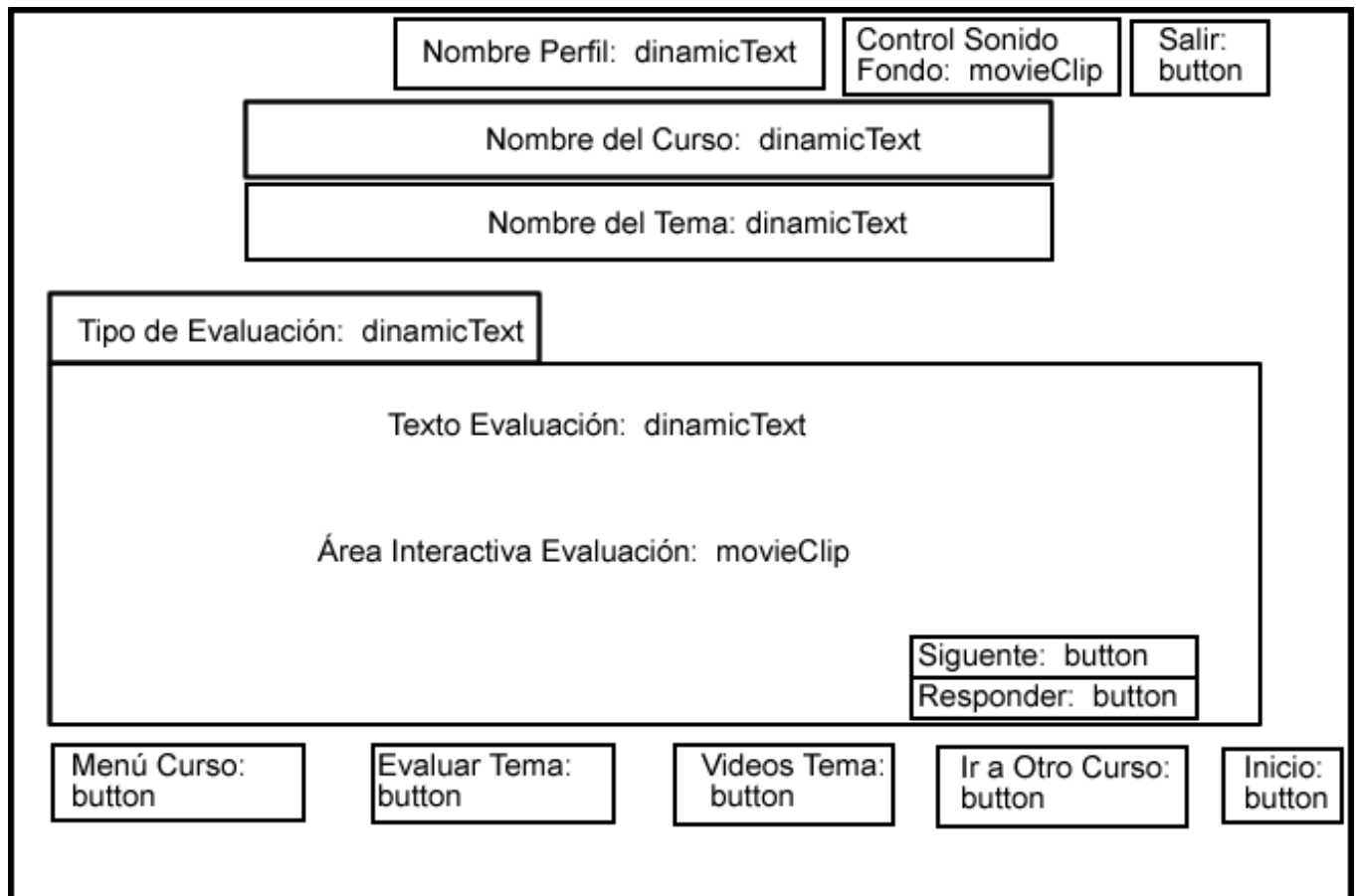


Fig. 15 *Pantalla principal de una autoevaluación.*

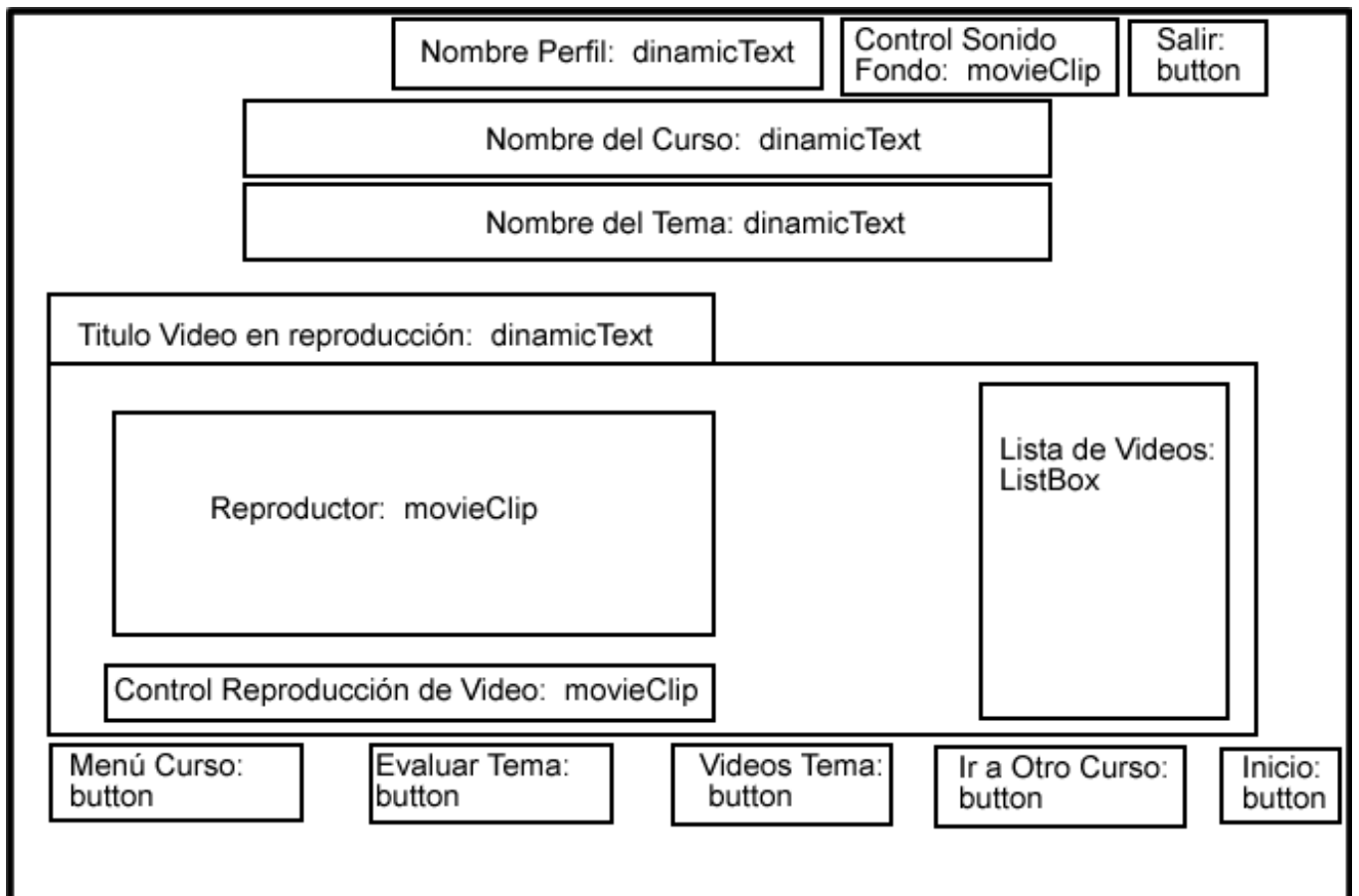


Fig. 16 Pantalla principal de videos.

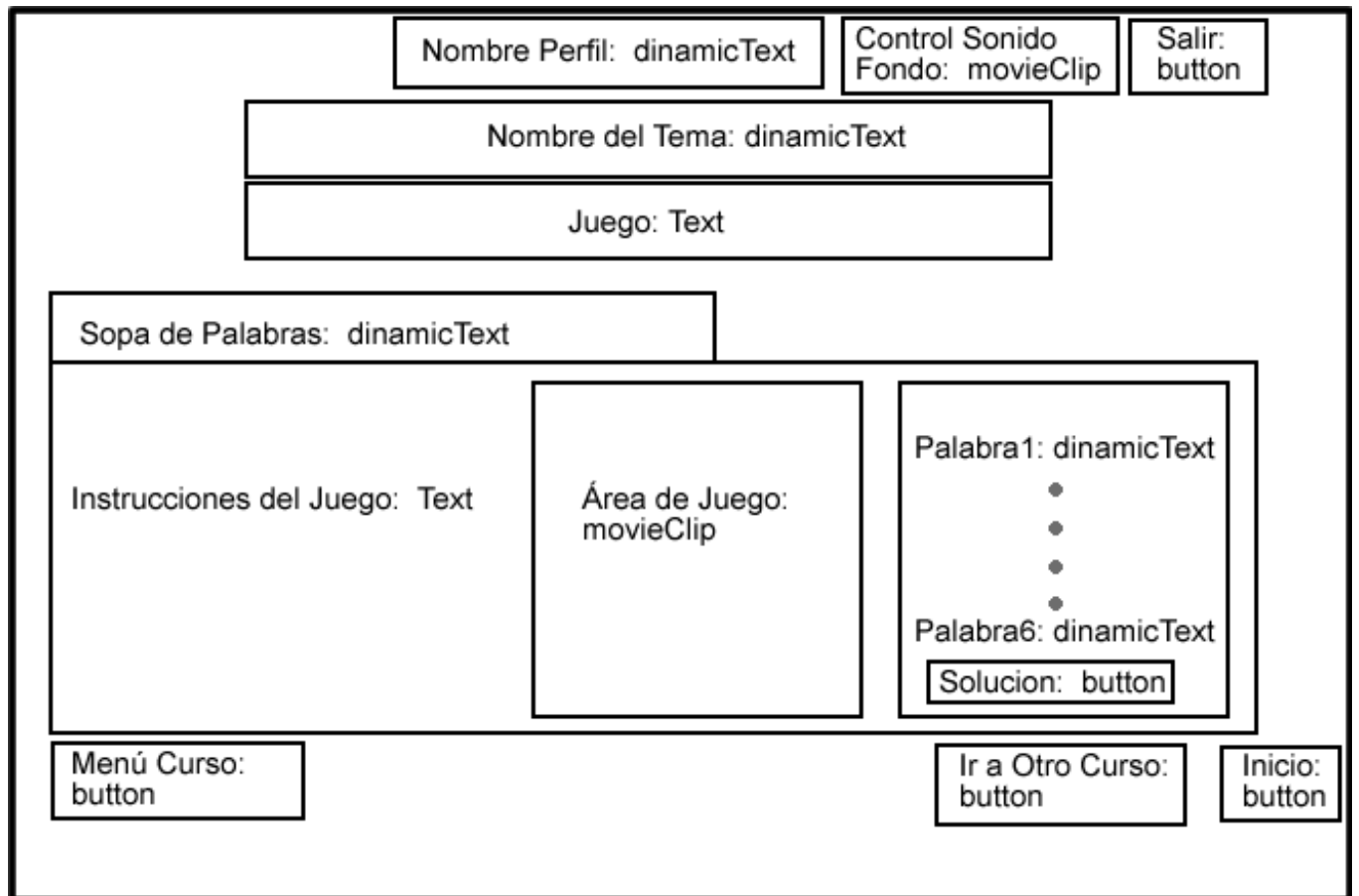


Fig. 17 Pantalla principal de un juego.

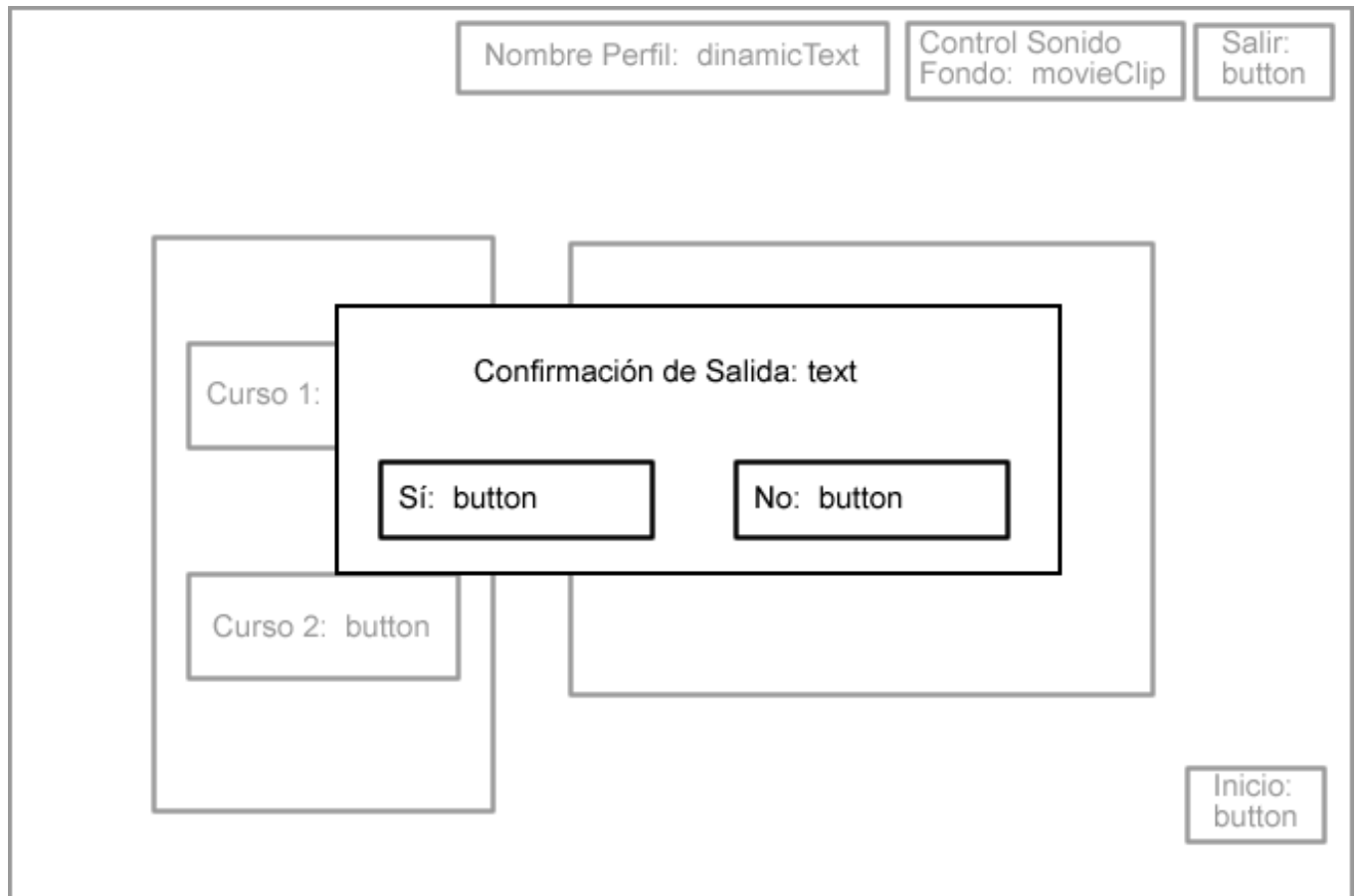


Fig. 18 Pantalla confirmación de salida.

3.4 Modelo de Implementación

La vista de implementación modela los componentes de un sistema a partir de los cuales se construye la aplicación, así como la dependencia entre los componentes, para poder determinar el impacto de un cambio en dicho sistema. Además describe como se organizan y se relacionan definiendo un componente como un empaquetamiento físico de los elementos de un modelo.

3.4.1 Diagrama de Componentes

En la *Fig. 19* se muestra la separación del sistema en componentes físicos, es decir, ficheros de código fuente, ficheros de código binario, módulos y paquetes. Para simplificar el diagrama cada tipo de archivo ejecutable con extensión `.swf` es generado por un archivo de código fuente con extensión `.fla` del mismo nombre que no esta reflejado en el diagrama.

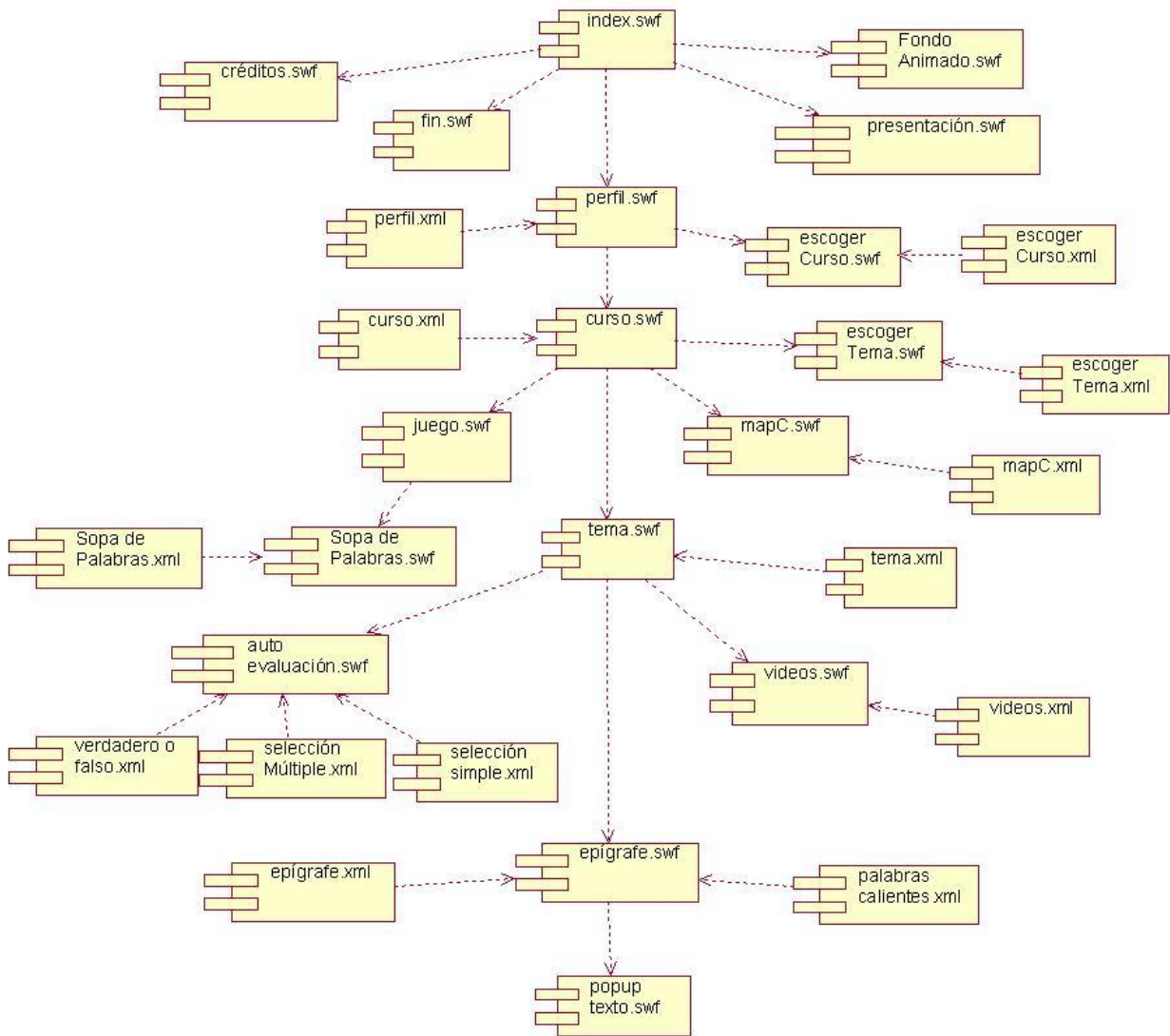


Fig. 19 Diagrama de Componentes General.

3.4.2 Diagrama de Despliegue

En la *Fig.20* se muestran los componentes a construir entre los diferentes nodos físicos en que funcionará la aplicación como son máquinas servidoras y clientes, permitiendo comprender la correspondencia entre la arquitectura software y la arquitectura hardware.

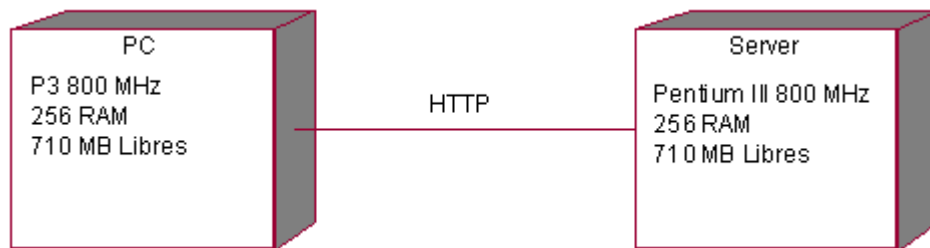


Fig. 20 Diagrama de Despliegue.

3.5 Descripción y estructura de principales archivos XML

A continuación se describen los diferentes archivos XML que posee la aplicación utilizados con el objetivo de que se cargue dinámicamente el contenido del mismo, garantizándose la independencia de los datos haciéndose lectura del contenido que se necesita mostrar en cada momento. Para poder darles un estilo y formato a los párrafos y oraciones que contiene los XMLs se tiene que codificar en HTML dentro de las etiquetas XML que delimitan ese contenido, al igual que soportan las etiquetas de nueva línea de HTML: **
**.

Archivo XML escoger curso

Contiene información que se muestra en la pantalla principal de la aplicación y el contenido de los tooltips de los botones de todos los cursos en esa pantalla.

```
<escogerCurso>
  <concepto>Especificaciones y Objetivos del Perfil</concepto>
  <curso1>Nombre y Objetivos del Curso 1</curso1>
  .
  .
  .
</escogerCurso>
```

Archivos XML escoger temas

Contienen información que se muestra en la pantalla principal de un curso y el contenido de los tooltips de los botones de todos los temas, del mapa conceptual y del juego en esa pantalla.

```
<escogerTema>
  <concepto>Concepto de lo que trata el curso</concepto>
  <tema1>Título del tema e información de lo que en el se aborda</tema1>
  .
  .
```

```

<juego>tipo de juego y características</juego>
<mapaC>Especificaciones del mapa conceptual<mapaC>
</escogerTema>

```

Archivos XML epígrafe

Contienen la información que se muestra en el epígrafe de un tema en específico. Para poder poner imágenes en el epígrafe tiene que ser dentro de las etiquetas XML **<contenido></contenido>** este soporta la etiqueta HTML **** al igual que para hacer referencia a alguna palabra caliente que este presente en el XML de palabras calientes se debe hacer con esta etiqueta **palabra caliente**. Además también contiene la dirección de ubicación de los archivos de los demás epígrafes.

```
<Epígrafe>
```

```

<nombre>Título del Epígrafe </nombre>
<contenido>Contenido Interactivo del Epígrafe</contenido>
<botonEpigrafe url= "dirección del swf del epígrafe">texto del botón
</botonEpígrafe>

```

```
</Epígrafe>
```

Archivos XML palabras calientes

Contienen todas las palabras calientes de un epígrafe con sus respectivos significados.

```
<PalabrasCalientes>
```

```

<cantPalabras>5</cantPalabras>
<palabra palabra="palabra caliente">Texto Significado palabra
caliente</palabra>

```

```
.  
. .  
. . .  
</PalabrasCalientes>
```

Archivos XML de videos

Contienen todos los títulos de los videos de un tema en específico, el nombre de cómo se tienen que llamar y enumerar los videos que están en la carpeta videos en el mismo directorio del XML y además da la posibilidad de habilitar o no la visualización del reproductor de video.

```
<Videos>  
  <CantidadVideos sufijo="video" habilitar = "true">3</CantidadVideos>  
  <video2>Texto Video </video2>  
  .  
  .  
  .  
</Videos>
```

Archivos XML de autoevaluación verdadero o falso

Contienen información sobre el tipo de autoevaluación verdadero o falso donde hay que especificarle el estado del texto.

```
<VerdaderFalso>  
  <texto estado="false">Texto del ejercicio verdadero o falso</texto>  
</VerdaderFalso>
```

Archivos XML de autoevaluación selección múltiple

Contienen información sobre el tipo de autoevaluación selección múltiple donde se especifica la cantidad de checkboxes con su correspondiente estado.

```
<selecciónMúltiple>
  <texto>Texto del ejercicio selección múltiple</texto>
  <CantCheckBox>6</CantCheckBox>
  <checkBox1 estado="true">Texto checkbox 1</checkBox1>
  <checkBox2 estado="false">Texto checkbox 2</checkBox2>
  .
  .
  .
</selecciónMúltiple>
```

Archivos XML de autoevaluación selección simple

Contienen información sobre la autoevaluación selección simple donde se especifica la cantidad de radioButtons y la posición del radiobutton que va hacer verdadero siendo los demás falsos por defecto.

```
<selecciónSimple>
  <texto>Texto del ejercicio selección Simple</texto>
  <cantRadioButton verdadero = "2">5</cantRadioButton>
  <radioButton1>Texto radioButton 1</radioButton1>
  <radioButton2>Texto radioButton 2</radioButton2>
  .
  .
  .
</selecciónSimple>
```

Archivos XML de juego sopa de palabras

Contienen los parámetros de un juego de sopa de palabras de 14 por 14 letras y 6 palabras a buscar de no más de 14 letras, al igual que la posición en el eje de las X y las Y de cada una de las letras de las palabras a buscar separadas por una coma empezando por la posición 0 hasta 13 dentro de la caja de letras de 14 por 14 letras de la sopa de palabras.

<Sopa>

<ListaPalabrasSopa>

<PalabraSopa1>ABCDEFGHJKLMN</PalabraSopa1>

.

.

.

<PalabraSopa14>TRRJRDOUOJHFIC</PalabraSopa14>

</ListaPalabrasSopa>

<ListaPalabrasBuscar>

<PalabraBuscar1>Palabra a buscar 1</PalabraBuscar1>

.

.

.

<PalabraBuscar6>Palabra a buscar 6</PalabraBuscar6>

</ListaPalabrasBuscar>

<ListaPalabrasPosiciónX>

<PalabraPosiciónX1></PalabraPosiciónX1>

.

.

.

<PalabraPosiciónX6></PalabraPosiciónX6>

</ListaPalabrasPosiciónX>

<ListaPalabrasPosiciónY>

<PalabraPosiciónY1></PalabraPosiciónY1>

.
.
.
<PalabraPosiciónY6></PalabraPosiciónY6>
</ListaPalabrasPosiciónY>
</Sopa>

Conclusiones

Luego del estudio del arte realizado y analizado las tendencias actuales, los lenguajes y metodologías a utilizar para el desarrollo de aplicaciones multimedia. Se creó un producto multimedia con contenidos informativos, interactivos, autoevaluativos y de entretenimientos para el Bloque Básico del Perfil de Calidad de la Universidad de Ciencias Informáticas, que apoya de una manera amena y agradable el proceso de aprendizaje a los estudiantes que se inician en el estudio de la Calidad de Software y el PDS.

El producto multimedia se ha desarrollado basado en la metodología RUP y los distintos artefactos que incorpora OMMMA-L ha esta metodología como extensión de UML específicamente para el desarrollo de aplicaciones multimedia. Poseído además este producto de un diseño sencillo, agradable a la vista y de fácil manejo por parte de los usuarios.

Resulta igualmente fácil mantenimiento, debido a lo dinámico de todo su contenido ya que integra la utilización de ficheros XML para gestionar y agrupar los datos mostrados en el producto multimedia en volúmenes compactos de información. Permitiendo con esto la incorporación de nuevos contenidos y la modificación de los ya existentes en caso de que se necesite.

Por todos los planteamientos descritos anteriormente se puede concluir que se cumplió satisfactoriamente los objetivos descritos al inicio del presente trabajo.

Recomendaciones

De manera general el software producido cumple con el objetivo planteado, sin embargo, se hacen las siguientes recomendaciones:

- Una actualización periódica de los contenidos, autoevaluaciones, videos y parámetros de los juegos de los cursos debido al carácter dinámico que estos presentan.
- Lograr que la estructura que presentan los mapas conceptuales sea obtenida de forma dinámica desde un archivo XML.
- Incorporarle más juegos a los cursos.
- Realizar un módulo de administración que permita a los profesores modificar el contenido de los cursos de acuerdo a sus intereses de una manera más sencilla, debido a que en su estado actual ese proceso resulta un poco engorroso. Lo antes planteado constituye un riesgo que puede echar a perder la información por una mala manipulación de los archivos XML que contienen la configuración y contenido de los cursos, además que se hace necesario tener conocimientos de XML y HTML.

Bibliografía

8402, ISO. 1994. *Administración de la calidad y garantía de la calidad.* 1994.

Corrales Díaz, Carlos. 1994. *La Tecnología Multimedia. Una nueva Tecnología de Comunicación e Información. Características, concepciones y aplicaciones.* Jalisco : s.n., 1994.

Cossío Cárdenas, Gema. *Los Compendios Informativos con tecnología multimedia como un servicio de alto valor añadido.*

de Benito Crosetti, Barbará. 2000. Tools for Developing Interactive Academic Web Courses. [En línea] Junio de 2000.

DMI. 2007. DMI. [En línea] 2007. [Citado el: 15 de Abril de 2008.] <http://dmi.uib.es/~bbuades/calidad/sld009.htm>.

Febles, Ailin. 2001. *Case Corporativo para el proceso de control de cambios. Tesis presentada en opción al título de Máster en Informática Aplicada.* Ciudad Habana : s.n., 2001.

Figuroa, Víctor. Sistemas de Gestión de la Calidad. [En línea] www.uach.cl/facultad/economicas/tema_administración007.

Garzoto Paolini, Paolo. 1993. *HDM-A Model Based Approach tu Hypermedia Application Desing.* *ACM Transaction on Information System.* 1993.

Gonzalez, Carlos. 2003. Conceptos Generales de calidad total. [En línea] 2003. <http://www.monografias.com/trabajos11/conge/conge.shtml>.

Group Report: There s Less Development Chaos Today SDTimes Software Development. **Rubstein, D. 2007.** 2007.

Gutiérrez Márquez, Andrés. 2006. *Calidad en las Organizaciones. Fundamentos, Análisis y Reflexiones.* 2006.

IEE, 610. 1990. Computer Dictionary, Compilation of IEEE Standard Computer Glossaries. *Institute of Electrical and Electronics Engineers Staff.* [En línea] 1990.

Izakowitz, Tomas. 1995. *Methodology for structured hypermedia design.* 1995.

- Jacobson, I, Grady, B y James, R. 2000.** *El Proceso Unificado de Desarrollo de Software.* 2000.
- Labori de la Rosa, Kenia y Rondón Robaúl, Izary. 2007.** *Metodologías para el desarrollo de software multimedia: Análisis comparativo y propuesta.* 2007.
- Lee Heeseok, C. L y Cheonsoo, Yoo. 1998.** *A Scenario-based object-oriented methodology for developing hipermedia information systems.* 1998.
- Martínez, Alicia.** *Calidad Enfocada al Desarrollo de Software.*
- McCall, James. 1977.** *Factors in Software Quality. Technical Report, General Electric.* 1977.
- Mestre Dumenigo, David A. 2007.** *Análisis de herramientas libres para el desarrollo de productos educativos multimedia.* Ciudad Habana : s.n., 2007.
- Milgrom, E, y otros. 1997.** *Evaluation of Web-based Tools for Building Distance Education Systems. Journal of Interactive Instruction Devepment.* 1997.
- Olsina, L. 1998.** *Buildin a Web-based information system applying the hypermdia flexible process modeling strategy.* 1998.
- Pérez, Memo.** La Calidad Total. [En línea] <http://Gestopolitis.com>.
- PRESSMAN, Roger S. 1998.** *Ingeniería de software. Un enfoque práctico.* s.l. : McGraw-Hill Interamericana de España S.A., 1998.
- Rossi, Gustavo y Garrido, Alejandra. 1997.** *Desing Reuse in Hipermedia Aplicarions Development.* 1997.
- Santos Fabelo, María del Carmen. 2008.** Prevención de las discografías escolares. [En línea] 2008. <http://www.monografias.com/trabajos55/disgrafia-escolar/disgrafia-escolar2.shtml>.
- Salaverría R.** Aproximación al concepto de multimedia desde los planos comunicativo e instrumental.
- SEDNA, P. 2006.** LA REVOLUCION INDUSTRIAL. [En línea] 2006. http://www.portalplanetasedna.com.ar/revolucion_industrial.htm.
- Simbandumwe, Jean-Paul.** Tools for Developin Interactive Academic Web Courses. [En línea] <http://www.umanitoba.ca/ip/tools/courseware>.
- Sommerville, I. 2002.** *Ingeniería de Software.* s.l. : Pearson Education , 2002.

Suárez Gijón Yaima Bety, Díaz Magdariaga Janet. 2007. *Mejora de Calidad en el proceso de desarrollo de software dirigido por casos de uso.* 2007.

Thomson, J. 1998. *Algorithmically detectable design patterns for hypermedia collections.* 1998.

Zambrano Rodríguez, Douglas Francisco. 2008. Multimedia. [En línea] 2008.
<http://www.monografias.com/trabajos10/mmedia/mmedia.shtml>.

Glosario de Términos

ActionScript: es un lenguaje de programación orientado a objetos (OOP), utilizado en especial en aplicaciones Web animadas realizadas en el entorno Macromedia Flash.

Actor: Abstracción de las entidades externas a un sistema, subsistemas o clases siempre y cuando interactúen directamente con el software. Un actor participa en un caso de uso o en conjunto coherente de casos de usos con el fin de llevar a cabo un propósito global.

Aplicación (sistema): sistema que ofrece a un usuario final un conjunto coherente de casos de usos.

Artefacto: pieza de información tangible que es creada, modificada y usada por los trabajadores al realizar actividades; representa un área de responsabilidad, y es candidata a ser tomada en cuenta para el control de la configuración. Un artefacto puede ser un modelo, un elemento de un modelo, o un documento.

CD (CD-ROM) Disco compacto con posibilidad de grabar todo tipo de datos informáticos.

Cliente: persona, organización grupo de personas que encarga la construcción de un sistema, ya sea empezando desde cero, o mediante el refinamiento de versiones sucesivas.

Componente: Es una parte física reemplazable de un sistema que empaqueta su implementación y es conforme a un conjunto de interfaces a las que proporciona su realización. Los componentes existentes en el dominio de la implementación son unidades físicas en los computadores que se pueden conectar con otros componentes, sustituir, trasladar, archivar, etc.

Despliegue: ocurre cuando varios trabajos mas o menos independientes (flujos de control, procesos) se distribuyen entre diferentes dispositivos hardware.

Diagrama: representación gráfica de una colección de elementos de modelado. Un diagrama muestra representaciones de elementos semánticos del modelo, pero su significado no se ve afectado por la forma en que son representados.

Dominio: Área de conocimiento o actividad caracterizada por un conjunto de conceptos y terminologías comprendidos por los practicantes de ese dominio.

FLA: Formato de fichero donde la herramienta de autor Flash guarda el documento de código fuente.

HTML. Lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

Hipermedia: Es un término usado como lógica extensión del término Hipertexto, en el cual audio, video, texto e hipervínculos generalmente no secuenciales, se entrelazan para formar un continuo de información, que puede considerarse como virtualmente infinito desde la perspectiva de Internet.

Hipertexto: Es el texto compuesto por fragmentos de otros textos con nexos electrónicos que los conectan entre si. Es un medio informático que relaciona información. Su rasgo distintivo es el de representar la información de forma lineal.

Microsoft: Compañía de software más grande del mundo. Fue fundada en 1975 por Paul Allen y Bill Gates. Aunque también se conoce por sus lenguajes de programación y aplicaciones para computadores personales, el éxito sobresaliente de Microsoft se debe a sus sistemas operativos DOS y Windows.

Modelo: captura una vista de un sistema del mundo real. Es una abstracción de dicho sistema, considerando un cierto propósito. El modelo describe completamente aquellos aspectos del sistema que son relevantes a su propósito, y a un apropiado nivel de detalle.

Multimedia: Es un sistema que utiliza más de un medio de comunicación al mismo tiempo en la presentación de la información, como texto, imagen, animación, vídeo y sonido (Salaverría).

Requisito: condición o capacidad que debe cumplir un sistema.

SWF: ShockWave Flash. Extensión de archivo de animación digital creado con Macromedia Flash y exportado con Macromedia Shockwave que puede ser visualizado independientemente, o desde una obra hecha con Director, o por un visor o browser de páginas Web en Internet.

Pantalla: Es la agrupación visual de elementos de medias contenidas en una vista determinada.

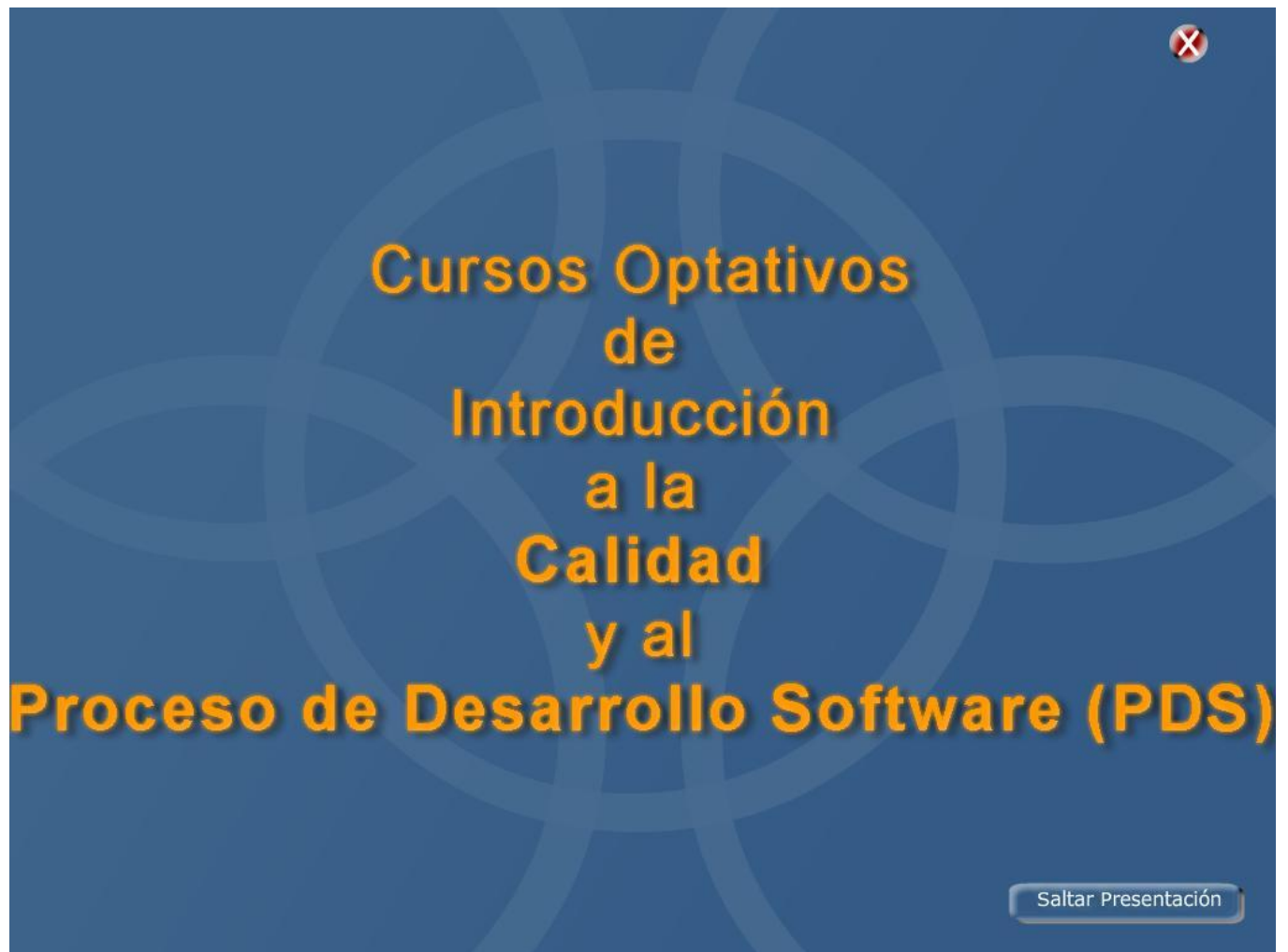
Plugin (Plug-in) Pequeño programa que añade alguna función a otro programa, habitualmente de mayor tamaño. Un programa puede tener uno o más conectores. Son muy utilizados en los programas navegadores para ampliar sus funcionalidades.

UML. Se le denomina así al Lenguaje Unificado de Modelado, basados en los primeros métodos de Programación Orientada a Objetos (POO) y está pensado para realizar análisis completos para desarrollo de aplicaciones de unas dimensiones amplias.

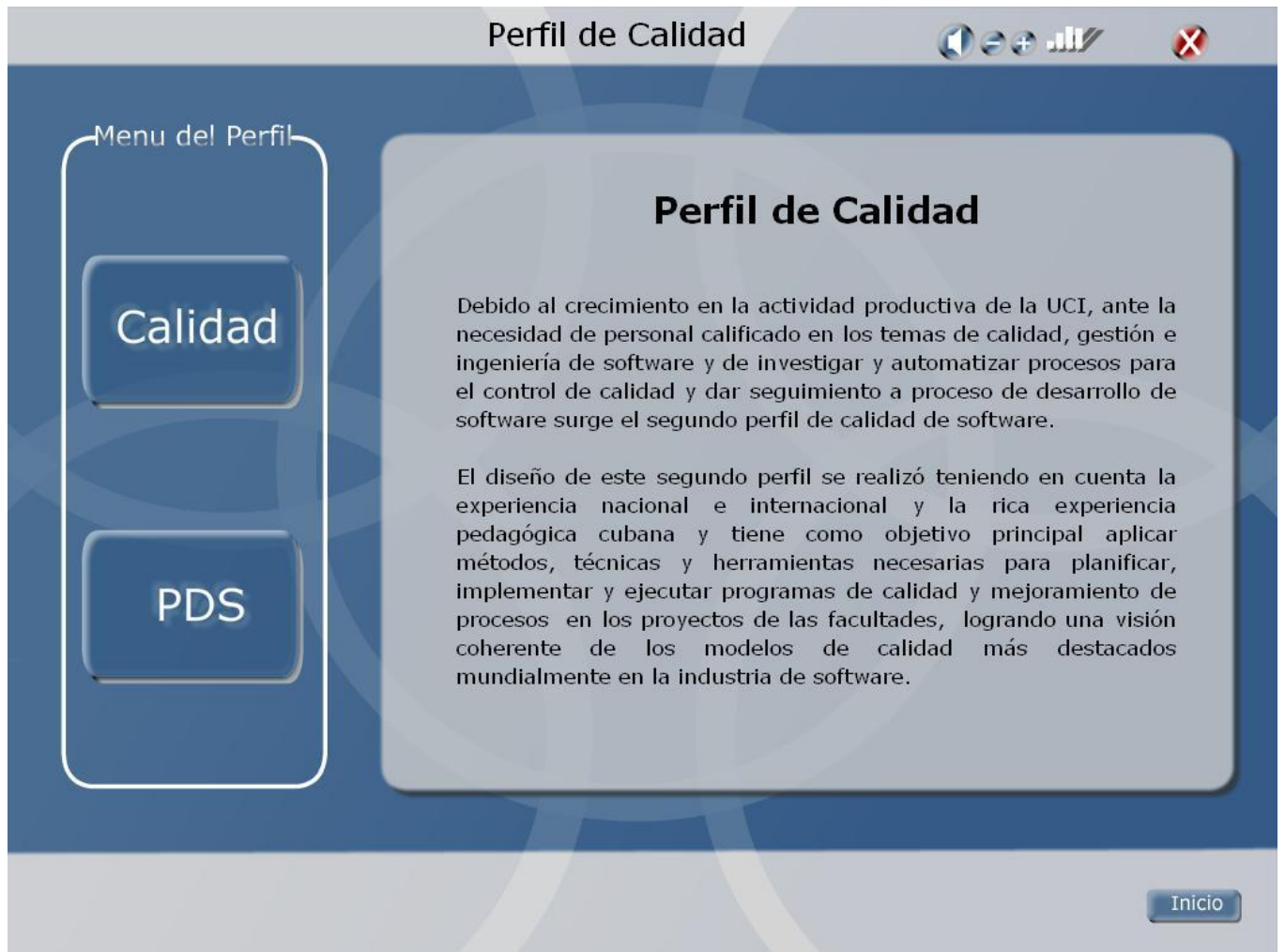
XML (Extensive Markup Language). Parecido a HTML pero más moderno y flexible. Se creó en 1998 por el World Wide Web Consortium (conocido por W3C) como sustituto del anterior, pensando principalmente en los negocios en la red. Es muy simple de utilizar y con unas características de hiperenlaces muy potentes gracias a las especificaciones XML (Extended Linking Language).

Anexos

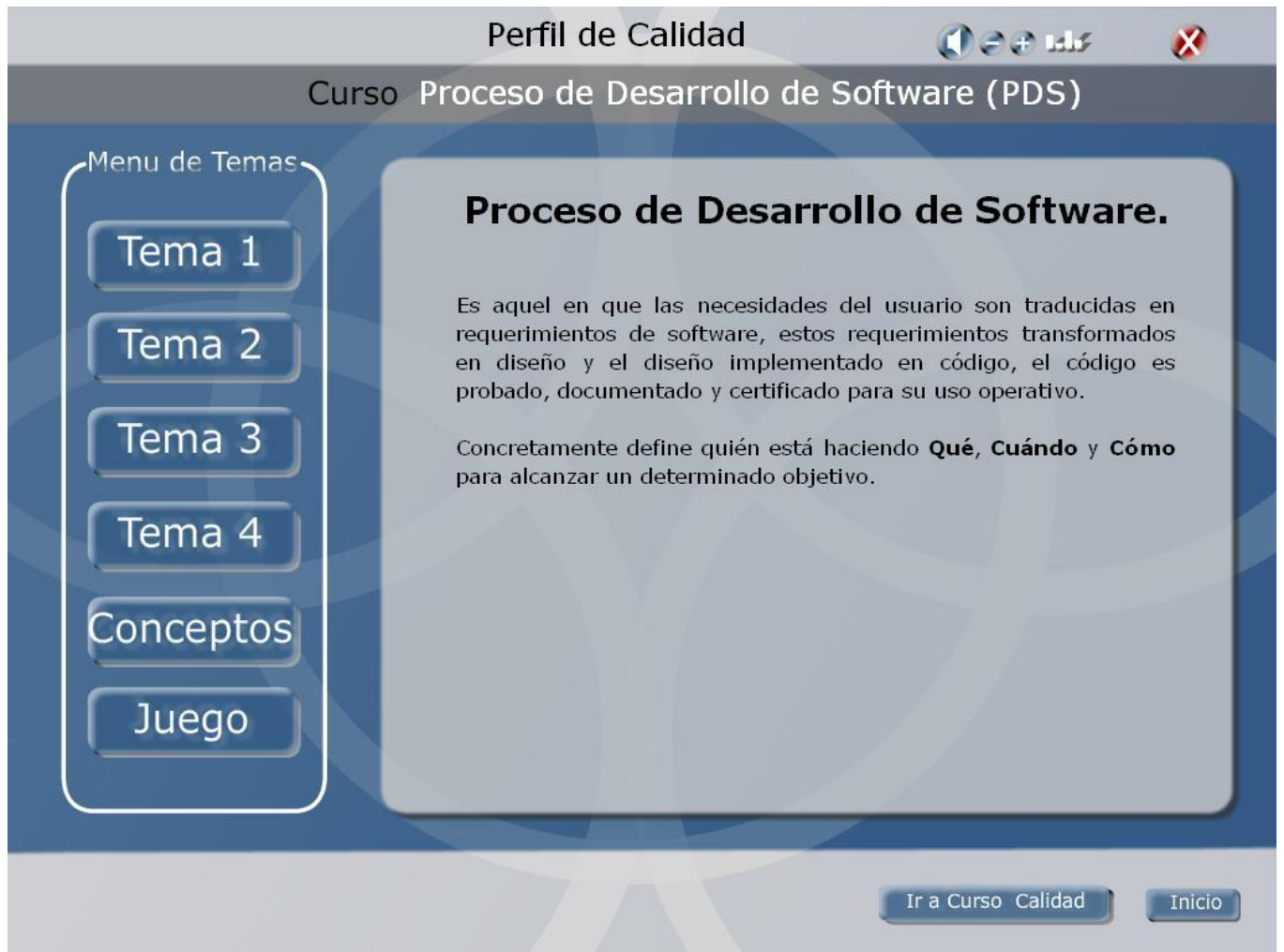
En los anexos que aparecen a continuación se muestran algunas imágenes de la interfaz del producto.



Anexo. 1 Presentación del Producto.



Anexo. 2 Pantalla Principal del Producto.



Anexo. 3 Pantalla Principal de un Curso del Producto.

The screenshot displays a web-based course interface. At the top, the title bar reads "Perfil de Calidad" with standard window controls. Below it, the course title is "Curso Proceso de Desarrollo de Software (PDS)" and the current topic is "Tema 1 Introducción al PDS". The main content area is titled "Metodologías para desarrollo software" and includes a sub-section "Demás epígrafes del tema". A text box on the left contains the phrase "proceso de software detallado y completo". The main text discusses the definition of a methodology, its components, and the challenges of selecting one. At the bottom, there are five navigation buttons: "Menu Curso PDS", "Auto Evaluar Tema", "Videos del Tema", "Ir a Curso Calidad", and "Inicio".

Perfil de Calidad

Curso Proceso de Desarrollo de Software (PDS)

Tema 1 Introducción al PDS

Metodologías para desarrollo software

Demás epígrafes del tema

proceso de software detallado y completo

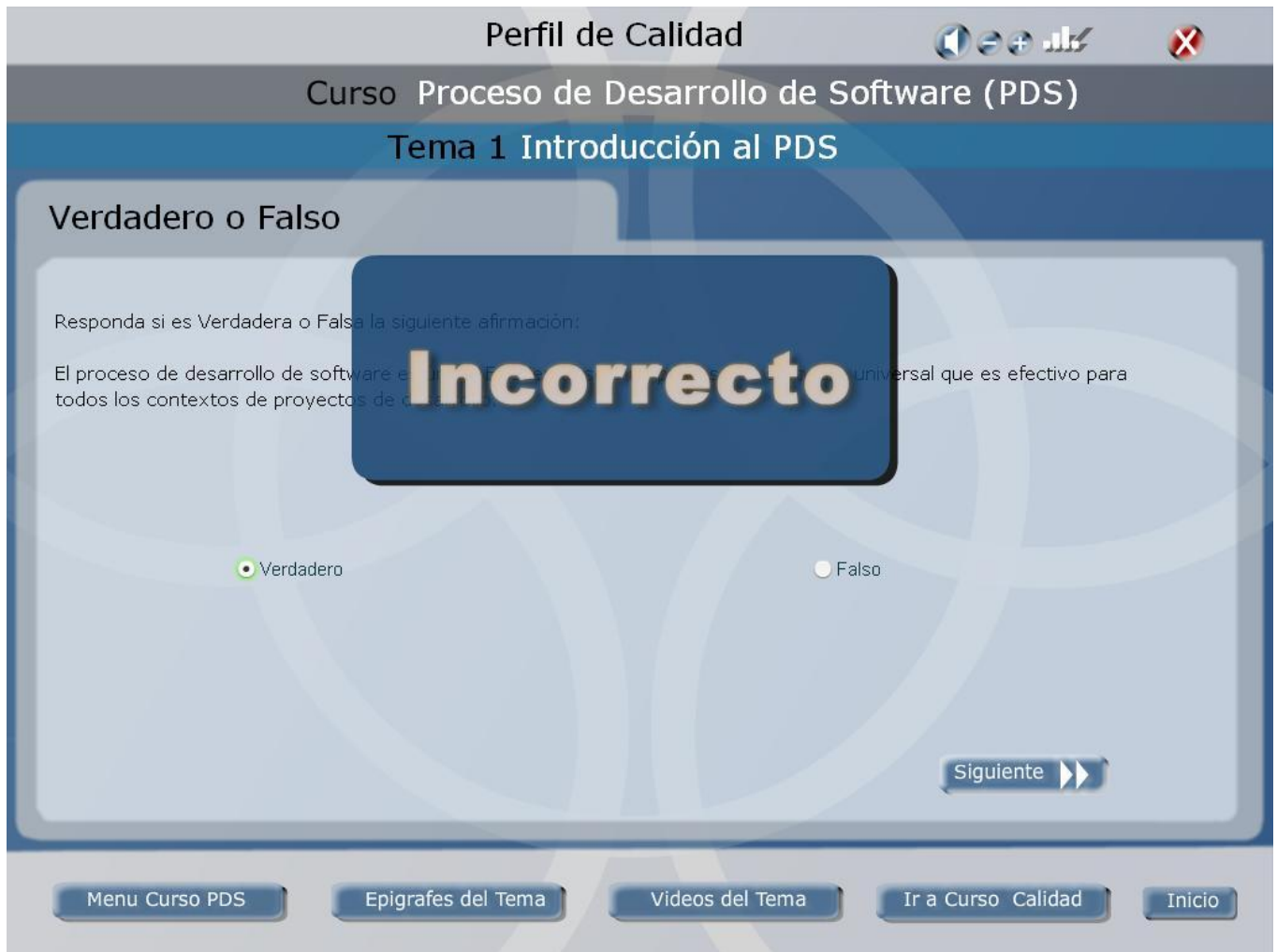
una **Metodología** se basa en una combinación de los modelos de evolutivo, incremental, etc.) Imponiendo un proceso disciplinado e con el fin de hacerlo: más predecible y eficiente. Adicionalmente inir con precisión los artefactos, roles y actividades involucrados, recomendadas, guías de adaptación de la metodología al proyecto, tas de apoyo, etc. Habitualmente se utiliza el término **método** que s) actividades del proceso de desarrollo, por ejemplo, suele hablarse eño.

ón de metodologías no es una tarea sencilla debido a la diversidad de propuestas y diferencias en el grado de detalle, información disponible y alcance de cada una de ellas.

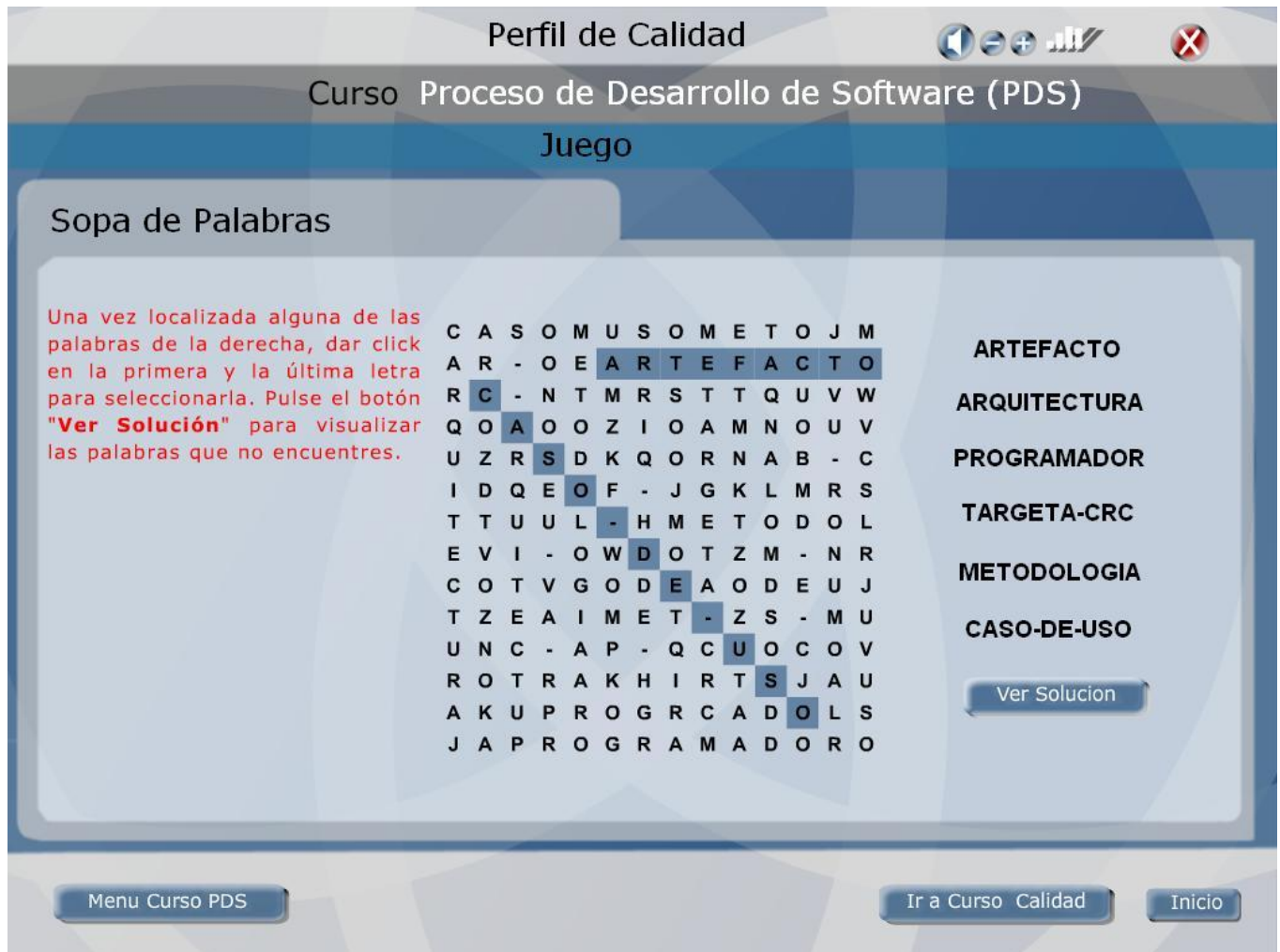
A grandes rasgos, si se toma como criterio las notaciones utilizadas para especificar artefactos producidos en actividades de análisis y diseño, podemos clasificar las metodologías según el paradigma de programación en dos grupos: **Metodologías Estructuradas** y **Metodologías**

Menu Curso PDS Auto Evaluar Tema Videos del Tema Ir a Curso Calidad Inicio

Anexo. 4 Pantalla Principal de un Tema de un Curso del Producto.



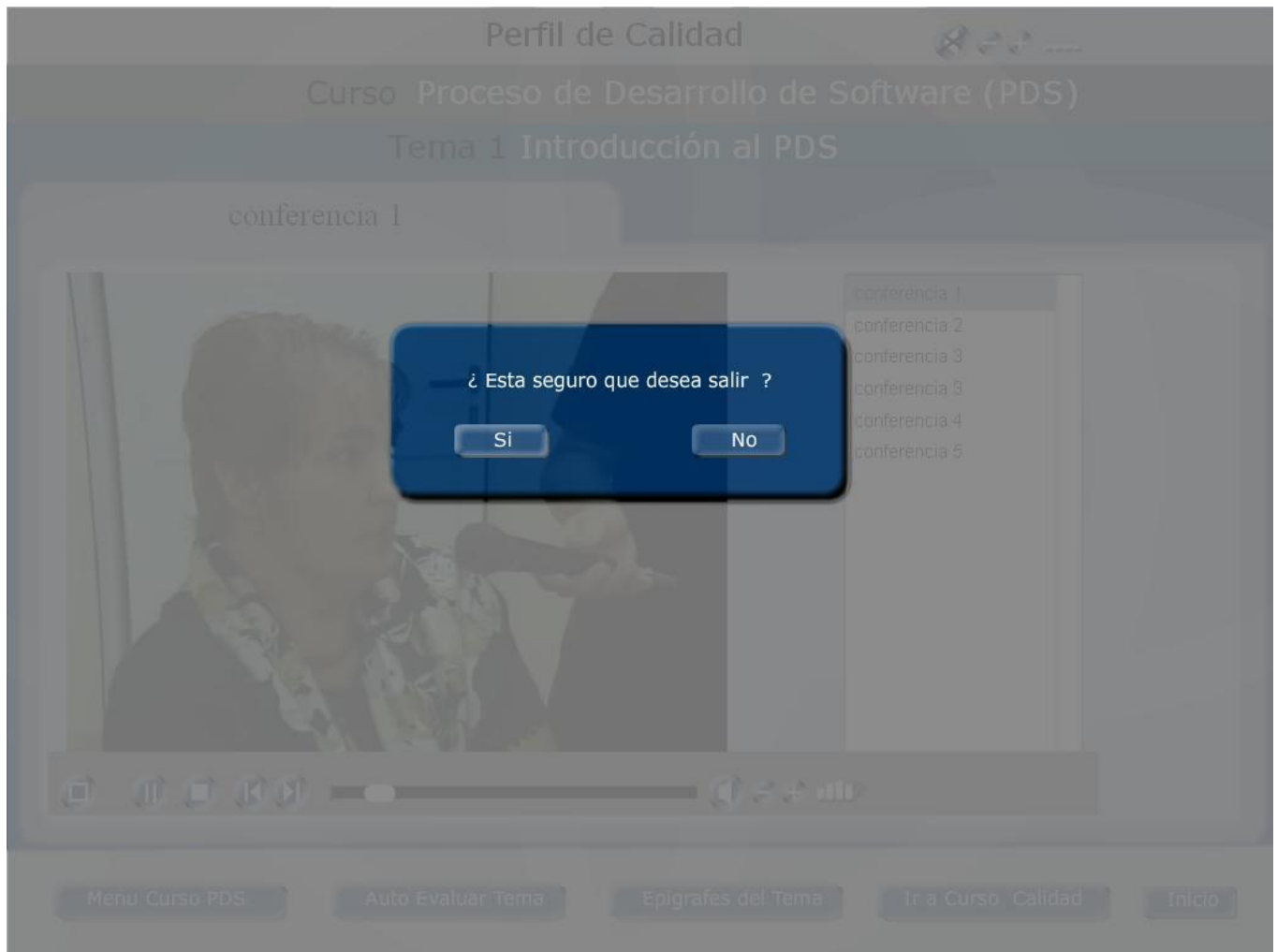
Anexo. 5 Pantalla de una Autoevaluación de un Tema de un Curso del Producto.



Anexo. 6 Pantalla Principal de un Juego de un Curso del Producto.



Anexo. 7 Pantalla Principal de Videos de un Tema de un Curso del Producto.



Anexo. 8 Pantalla Confirmación de Salida del Producto.

Perfil de Calidad de Software de la Universidad de las Ciencias Informáticas, Trabajo para optar por el título de Ingeniero en Ciencias Informáticas.

Diseño Original y Autor:

Raidel Ramon de Armas Gurri

Programador:

Raidel Ramon de Armas Gurri

Tutor

MSc. Yamilis Fernández Pérez

Especialista de Contenido Calidad

Ing. Tayche Capote García

Especialista de Contenido PDS

Ing. Ailec Granda Dihigo

Anexo. 9 Créditos del Producto.