

Universidad de las Ciencias Informáticas

Facultad 5

Perfil “Realidad Virtual”



**Diseño e Implementación de una plataforma
básica sobre Web para prácticas virtuales
interactivas**

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

Autor: Roger Florat Gutierrez

Tutor: Ing. Ariel Viera Díaz

Co. Tutor: Lic. Luis G. Viciedo Carabaloso

Ciudad de la Habana Julio del 2007

“Año 49 de la Revolución”

DECLARACIÓN DE AUTORÍA

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

Y para que así conste firmo la presente a los ____ días del mes de _____ del 2007.

Roger Florat Gutierrez

Ariel Viera Díaz

.....
Firma del Autor

.....
Firma del Tutor

Agradecimientos

A mi familia...

A mis padres: Roselio y Dulce, por haber depositado en mi toda su confianza y amor, porque han sabido guiarme por el buen camino y sembrando en mí los valores que me caracterizan, ya que son ejemplos de firmeza, audacia e inteligencia, dándome siempre su apoyo incondicional.

A mi hermana: Damarys por todo su apoyo y ternura...

A mis amigos y compañeros de todos los tiempos, a todos aquellos que se detuvieron al menos un instante para regalarme una sonrisa, un gesto, una frase de apoyo...

A todas aquellas personas que de una forma u otra contribuyeron al desarrollo de este trabajo.

A la Revolución, por darme la posibilidad de hacer realidad este sueño... Muchas gracias Fidel.

Dedicatoria

A mis padres por la confianza que siempre depositaron en mí, por su apoyo por su cariño.

A mi hermana por ser tan especial.

A mi familia...

Resumen

Este trabajo esta basado en la creación de una primera versión de una herramienta de autor sobre Web que permite el diseño de una secuencia de actividades para prácticas de laboratorios virtuales.

Por tanto el objetivo de este trabajo consiste en desarrollar una aplicación Web, segura y de interfaz amigable, que permita crear de forma interactiva una secuencia de actividades para prácticas de laboratorios virtuales sobre Web, las que serán utilizadas en un entorno de aprendizaje para el control y evaluación de tareas que son diseñadas por el profesor.

Este documento plasma los resultados del estudio realizado para la construcción del sistema, se explican los conceptos relacionados con el mismo, se hace un análisis de la propuesta y de las tecnologías existentes siguiendo la filosofía del software libre.

Para su desarrollo se siguieron los pasos que proponen el Proceso Unificado del Software. Está implementado específicamente en el lenguaje PHP, como gestor de base de datos se utilizó MySQL y para dar una mayor interactividad y hacer la aplicación más creativa y llamativa utilizaremos XML, CSS y Ajax por ser tecnologías de punta con múltiples ventajas.

Índice

Introducción	1
Capítulo 1 Fundamentación Teórica	4
Introducción.....	4
1.1 ¿Qué es educación virtual?	4
1.2 ¿Qué es un Laboratorio Virtual?	6
1.3 ¿Que es una Herramienta de Autor?	14
1.4 Entornos Virtuales de Aprendizajes	15
1.5 Tendencias y tecnologías actuales	17
1.5.1 Modelo Cliente Servidor	17
1.5.2 Tecnologías para el desarrollo de aplicaciones Web.....	21
1.5.3 Sistemas de Gestión de Bases de Datos.....	30
1.5.4 Servidores Web.....	34
1.5.5 Tendencias actuales	36
1.6 Fundamentación de la metodología a utilizar	39
1.7 Elección de las herramientas que serán utilizadas.	41
Conclusiones.....	42
Capítulo 2 Características del Sistema	43
Introducción.....	43
2.1 Modelación del Negocio.....	43
2.2 Modelo de Dominio	45
2.3 Captura de requisitos	46
2.4 Modelo de casos de uso del sistema	49
2.5 Glosario de Términos.....	56
Conclusiones	57
Capítulo 3 Descripción de la Solución Propuesta	58
Introducción.....	58
3.1 Patrones de diseño	59
3.2 Modelo de Diseño	59
3.3 Diseño de la Base de Datos.....	70
3.4 Principios de diseño	72
3.5 Descripción preliminar del modelo de pruebas	73
3.6 Modelo de despliegue	74
Conclusiones.....	75
Conclusiones Generales	76
Recomendaciones	77
Referencias bibliográficas	78
Glosario de términos	80

Introducción

La introducción de las Tecnologías de la Información y las Comunicaciones (TIC) en el sistema educativo, ha marcado una nueva etapa de desarrollo en los procesos de enseñanza universitaria, y especialmente en las carreras de ingeniería.

La vinculación de estas tecnologías en la educación ha ampliado las oportunidades para transformar y mejorar los procesos enseñanza y aprendizaje utilizando nuevos modelos educativos.

Dentro de estos nuevos modelos educativos podemos encontrar los laboratorios virtuales, los cuales como medio de enseñanza proporciona al estudiante, conceptos, habilidades y destrezas en el uso de la computadora, y en el desarrollo cognitivo para que se logre un aprendizaje significativo.

Estos laboratorios permiten observar gran diversidad de fenómenos; simples procesos de observación, fenómenos imposibles de apreciar en la naturaleza, acerca y facilita la realización de experiencias a un mayor número de alumnos, aunque alumno y laboratorio no coincidan en el espacio. Permite simular fenómenos y modelos físicos, conceptos abstractos, mundos hipotéticos, controlar la escala de tiempo, etc., ocultando el modelo matemático y mostrando el fenómeno simulado de forma interactiva.

En el proyecto Laboratorios Virtuales se está desarrollando una plataforma para la educación a distancia, en el cual se propone una herramienta de autor sobre Web que permita el diseño de secuencias de actividades para prácticas de laboratorio virtuales y dentro de la cual existe la necesidad de que el profesor de manera interactiva diseñe un conjunto o secuencias de tareas de manera on-line para el desarrollo de estas prácticas de laboratorio.

Después de hacer un análisis de la **situación problemática** (SP) expuesta anteriormente, definimos como **problema científico**: ¿Cómo combinar un conjunto de diferentes tecnologías que permitan un diseño interactivo de secuencias de tareas para prácticas de laboratorios virtuales sobre Web?

Nuestro problema se enmarca en la investigación de las tecnologías y herramientas de comunicación para el desarrollo de prácticas de laboratorios virtuales sobre Web como **objeto de estudio** (OE).

El **objetivo general** es implementar de manera interactiva secuencias de tareas para el desarrollo de prácticas de laboratorios virtuales sobre Web.

El **campo de acción** esta dirigido al análisis y diseño de la arquitectura básica para el desarrollo de una herramienta de autor para prácticas virtuales sobre Web.

De acuerdo a esta propuesta se desarrollaran las siguientes **tareas**:

- Realizar un análisis y estudio bibliográfico del tema a nivel nacional e internacional.
- Seleccionar las tecnologías que más se ajusten a los requerimientos.
- Analizar y discutir la arquitectura de la aplicación.
- Descripción de los procesos que se van a implementar en la aplicación.
- Declaración de los ciclos de desarrollo.
- Desarrollar los diagramas que describen el diseño Web de la aplicación.
- Diseño y normalización de la Base de Datos
- Diseño de la interfaz.
- Implementación de la aplicación.

Estructuración del contenido

Capítulo 1 “Fundamentación Teórica”: En este capítulo se explica todo el estado del arte de la tesis, se realiza una breve descripción de que es Educación Virtual y Laboratorios Virtuales. Se explican las metodologías, lenguajes y herramientas utilizadas para el desarrollo del módulo correspondiente.

Capítulo 2 “Características del Sistema”: Se identifican los actores y trabajadores del sistema, se definen conceptos que serán agrupados en un Modelo de Dominio y se enumeran los requisitos funcionales y no funcionales con que debe contar el sistema.

Capítulo 3 “Descripción de la solución propuesta”: Aborda aspectos relacionados con la construcción de la solución propuesta, se modelan los diagramas de clases de diseño y de secuencia, se plantea el modelo de datos, y se especifican los principios para el diseño gráfico y la implementación.



Fundamentación Teórica

Introducción

En el presente capítulo se brinda una visión general de los principales aspectos relacionados con la Educación Virtual y los Laboratorios Virtuales, sus principales conceptos y características.

También se hace un análisis de las tecnologías y tendencias que existen en la actualidad a nivel mundial y que pudieran ser útiles en el desarrollo de la propuesta y de la solución. Se tienen en cuenta los servidores Web, los lenguajes de programación para la Web, los Sistemas Gestores de Bases de Datos mayormente utilizados a escala internacional, las distintas metodologías de desarrollo de software, así como el lenguaje de marcado extensible (XML) y las hojas de estilo en cascada (CSS).

1.1 ¿Qué es educación virtual?

Se entiende por *educación virtual* o *e-learning* la utilización de nuevas tecnologías y herramientas de aprendizaje como complemento a procesos sincrónicos y asincrónicos de comunicación y enseñanza, es un sistema y modalidad educativa que surge de la necesidad propia de la educación y tecnología educativa.

Es una modalidad del proceso enseñanza aprendizaje, que parte de la virtud inteligente imaginativa del hombre, hasta el punto de dar un efecto a la realidad, en la interrelación

con las nuevas tecnologías, sin límite de tiempo–espacio que induce a constantes actualizaciones e innovaciones del conocimiento. [1]

Son aquellas formas de estudio que no son guiadas o controladas directamente por la presencia de un profesor en el aula, pero se beneficia de la planeación y guía de los tutores a través de un medio de comunicación que permita la interrelación profesor-alumno. [2]

Es una estrategia educativa, basada en el uso intensivo de las nuevas tecnologías, estructuras operativas flexibles y métodos pedagógicos altamente eficientes en el proceso enseñanza-aprendizaje, que permite que las condiciones de tiempo, espacio, ocupación o edad de los estudiantes no sean factores limitantes o condicionantes para el aprendizaje. [2]

Principios de la educación virtual:

- Interactivos, los usuarios pueden adoptar un papel activo en relación al ritmo de aprendizaje.
- Multimedia, se incorpora a textos, imágenes fijas, animaciones, videos, sonidos.
- Abierta, permite una actualización de los contenidos y las actividades de forma permanente, algo que los libros de textos no poseen.
- Sincrónicos y asincrónicos, los alumnos pueden participar en las tareas y actividades en el mismo momento independientemente y en cualquier lugar (Sincrónico). O bien, la realización del trabajo y estudio individual en el tiempo particular de cada alumno (asincrónico).
- Accesibles, no existen limitaciones geográficas ya que utiliza todas las potencialidades de la red de Internet, de manera que los mercados de formación son abiertas.

- Con recursos on-line, que los alumnos pueden recuperar en sus propios ordenadores personales.
- Distribuidos, no tienen porque estar centrado en un solo lugar, sino accesible en cualquier lugar del mundo, los recursos y materiales didácticos.
- Con un alto seguimiento, el trabajo y actividad de los alumnos, ya que organiza tareas a tiempo a remitir.
- Comunicación horizontal, entre los alumnos, ya que la formación y colaboración parte de las técnicas de formación.

1.2 ¿Qué es un Laboratorio Virtual?

Entiéndase por laboratorio virtual la creación de un entorno que integró mediante la utilización de las tecnologías Web todos los recursos necesarios para la realización de las prácticas específicas de una asignatura en cuestión, ya sea para su utilización en el laboratorio como fuera de él.

Se entiende por laboratorio virtual un sitio informático que simula una situación de aprendizaje comúnmente realizada en un espacio físico llamado laboratorio.

Los laboratorios virtuales son imitaciones digitales de prácticas de laboratorio o de campo, reducidas a la pantalla de una computadora. [3]

En la educación, el concepto de laboratorios virtuales ha sido utilizado para designar una simulación de lo que la/el estudiante percibe, algunas acciones semejantes a estar presente en un laboratorio real, apoyándose en el uso de tecnologías digitales. [4]

Una de las definiciones de “laboratorios virtuales” que se ha aplicado a la enseñanza a distancia es la de Monge-Nájera et al. (1999), que las definen como “simulaciones de prácticas manipulativas que pueden ser hechas por la/el estudiante lejos de la universidad y el docente”. [3]

Niveles de complejidad de un laboratorio virtual:

- El nivel más sencillo es el que tiene básicamente un texto y dibujos sin movimiento. Ejemplos de este nivel son el Digital Frog de 1995 (<http://www.digitalfrog.com/products/frog.html>) Digital Frog permite hacer una disección simulada de una rana, evitando los problemas éticos y psicológicos de hacerlo con un animal real. [3]
- En un segundo nivel de complejidad, existen laboratorios que usan animaciones usando el formato GIF, compatible con Internet. Un ejemplo es el Laboratorio Virtual de Reproducción de la UNED de 1997 (Monge-Nájera 1998), en el cual se puede seleccionar organismos para ver una animación que muestra su secuencia reproductiva, incluyendo imágenes de microscopio electrónico. El Laboratorio Virtual de Depredadores y Presas (UNED 2002) permite variar la proporción de organismos en un ambiente y ver el efecto sobre la población. [3]
- El tercer nivel corresponde a los laboratorios que usan videos para mostrar prácticas verdaderas. Ejemplos de este nivel son el Laboratorio Virtual de Digestión desarrollado por la UNED en 1997 (Monge-Nájera 1998) y el Digital Frog 2, versión mejorada del ya mencionado, en que además de las imágenes fijas hay videos (<http://www.digitalfrog.com/products/frog.html>). [3]
- En el cuarto nivel de complejidad están aquellos laboratorios en los cuales se ven en la pantalla objetos o escenas que pueden ser manipulados por la (el) estudiante. Ejemplo de este nivel es el laboratorio Virtual Drosophila Project japonés y Mouse genetics (http://www.explorescience.com/activities/Activity_page.cfm?ActivityID=39). [3]
- En el quinto nivel se tienen laboratorios en que la falta de certeza en las mediciones y la variabilidad aleatoria de algunos parámetros limitan el control que tiene el usuario. Para explicar esto se puede usar un símil: en las simulaciones actuales, se puede lanzar una canica digital a un frasco y si se apunta en la dirección correcta, cae dentro. En el futuro, factores como el viento y la oscuridad en la habitación digital afectarán la trayectoria: será más difícil

acertar. Estos laboratorios que imitan, por ejemplo, la falibilidad de la puntería humana, están siendo desarrollados en Québec (VPLab, http://www.licef.teluq.quebec.ca/gmec/lvphysique/LVP_EDMEDIA.htm). [3]

- El sexto nivel, permiten al usuario mirar una imagen y percibir en su piel las sensaciones correspondientes (<http://www.vrinnovations.com/>), aunque tiene muy poco control sobre la secuencia de eventos. [3]
- El séptimo nivel corresponde a la nueva tecnología que permite una interacción con otra persona conectada a la red Internet, usando casco y traje (<http://www.cnn.com/2001/TECH/computing/01/19/virtual.sex.idg/>). [3]
- El octavo nivel, que una vez desarrollado podría permitir una buena relación costo / beneficio en los laboratorios virtuales, se relacionaría con reproducir experiencias mediante implantes eléctricos dentro del cuerpo humano (Jacomma, R. 2002 Virtual sex is coming. The bad news is it's being developed by the English. <http://www.seattleweekly.com/features/0103/tech-jacomma.shtml>). A pesar del enorme potencial de esta tecnología para el campo educativo, su desarrollo es todavía tan preliminar que es difícil juzgar si tendrá uso práctico en este y otros. [3]

Ventajas de los Laboratorios Virtuales:

Un laboratorio virtual (LV) es un sistema computacional que pretende aproximar el ambiente de un laboratorio tradicional (LT).

Los experimentos se realizan paso a paso, siguiendo un procedimiento similar al de un LT: se visualizan instrumentos y fenómenos mediante objetos dinámicos (applets de Java o Flash, cgi-bin, JavaScripts,...), imágenes o animaciones. Se obtienen resultados numéricos y gráficos, tratándose éstos matemáticamente para la obtención de los objetivos perseguidos en la planificación docente de las asignaturas. A continuación, destacamos algunas ventajas importantes de los LV.

- Acerca y facilita a un mayor número de alumnos la realización de experiencias, aunque alumno y laboratorio no coincidan en el espacio. El estudiante accede a los equipos del laboratorio a través de un navegador, pudiendo experimentar sin riesgo alguno, y, además, se flexibiliza el horario de prácticas y evita la saturación por el solapamiento con otras asignaturas.
- Reducen el coste del montaje y mantenimiento de los LT, siendo una alternativa barata y eficiente, donde el estudiante simula los fenómenos a estudiar como si los observase en el LT.
- Es una herramienta de autoaprendizaje, donde el alumno altera las variables de entrada, configura nuevos experimentos, aprende el manejo de instrumentos, personaliza el experimento, etc. La simulación en el LV, permite obtener una visión más intuitiva de aquellos fenómenos que en su realización manual no aportan suficiente claridad gráfica. El uso de LV da lugar a cambios fundamentales en el proceso habitual de enseñanza, en el que se suele comenzar por el modelo matemático. La simulación interactiva de forma aislada posee poco valor didáctico, ésta debe ser embebida dentro de un conjunto de elementos multimedia que guíen al alumno eficazmente en el proceso de aprendizaje. Se trata de utilizar la capacidad de procesamiento y cálculo del ordenador, incrementando la diversidad didáctica, como complemento eficaz de las metodologías más convencionales.
- Los estudiantes aprenden mediante prueba y error, sin miedo a sufrir o provocar un accidente, sin avergonzarse de realizar varias veces la misma práctica, ya que pueden repetirlas sin límite; sin temor a dañar alguna herramienta o equipo. Pueden asistir al laboratorio cuando ellos quieran, y elegir las áreas del laboratorio más significativas para realizar prácticas sobre su trabajo.

En Internet encontramos multitud de simulaciones de procesos físicos (en forma de applets de Java y/o Flash). Con estos objetos dinámicos, el docente puede preparar actividades de aprendizaje que los alumnos han de ejecutar, contestando al mismo tiempo las cuestiones que se les plantean.

Algunos inconvenientes:

Los LV a parte de unas series de características y ventajas que proporcionan no puede sustituir la experiencia práctica altamente enriquecedora del LT. Ha de ser una herramienta complementaria para formar a la persona y obtener un mayor rendimiento.

En el LV se corre el riesgo de que el alumno se comporte como un mero espectador. Es importante que las actividades en el LV, vengan acompañadas de un guión que explique el concepto a estudiar, así como las ecuaciones del modelo utilizado.

Es necesario que el estudiante realice una actividad ordenada y progresiva, conducente a alcanzar objetivos básicos concretos. El alumno no utiliza elementos reales en el LV, lo que provoca una pérdida parcial de la visión de la realidad. Además, no siempre se dispone de la simulación adecuada para el tema que el profesor desea trabajar.

Requisitos que debe cumplir un Laboratorio Virtual:

- Ser autocontenido
- Ser interactivo
- Combinar imágenes bidimensionales y tridimensionales
- Tener animación tridimensional, video y sonido
- Incluir ejercicios (cuya calificación puede ser enviada automáticamente al docente)
- Instalación automática
- Que la navegación no sea necesariamente lineal
- Posibilidad de guardar notas sin necesidad de procesador de textos externo.
- Un buscador

El desarrollo de los LV en Cuba:

La introducción de las nuevas Tecnologías de la Información y las Comunicaciones (TIC) en nuestro país constituyen una nueva herramienta para el desarrollo de nuevas formas de enseñanza y aprendizaje. La utilización de estas tecnologías en la creación de espacios virtuales que simulen procesos reales ha marcado una nueva etapa de desarrollo en los procesos de enseñanza universitaria, y especialmente en las carreras de ingeniería.

A continuación se citan algunos ejemplos:

- Laboratorio virtual en anestesiología.

En el Hospital General Docente "Aleida Fernández Chardiet" Güines, La Habana se desarrolló un trabajo sobre anestesiología aplicada a la especialidad de cirugía oral y maxilofacial que contribuye al conocimiento y aprendizaje de alumnos, técnicos y profesionales a través de un software que permite clasificar imágenes captadas y procesadas en diferentes equipos (scanner, cámara digital, etc.). En las imágenes clasificadas se describen aspectos de utilidad para el entrenamiento del personal en formación en las diferentes técnicas de anestesia locorregional. La utilización de este medio de enseñanza en soporte electrónico ofrece la posibilidad de actualizar, adecuar, describir, procesar, representar y modificar, gráficos, textos e imágenes de forma rápida y ahorrando cuantiosos recursos. (Laboratorio virtual en anestesiología)

- Sistema Interactivo Didáctico para la Enseñanza de la Física.

Se explican las particularidades de un sistema sustentado en la interacción del alumno con materiales didácticos, dirigido esencialmente a los estudiantes de la asignatura Física de las carreras de ingeniería de la Universidad Central "Marta Abreu" de Las Villas, Santa Clara, Villa Clara, Cuba. El sistema se aplica desde el año 2002 con muy buenos resultados ya que garantiza la información que puede necesitar el estudiante de estas carreras para lograr su preparación teórica necesaria para desarrollar las prácticas de laboratorio de Física General. En el sistema propuesto el alumno puede utilizar la información de forma

presencial o en la modalidad virtual. El material apoya el desarrollo de las conferencias, clases prácticas, seminarios, y las actividades de estudio independiente; en él se muestra gran cantidad de simulaciones lo cual permite sustituir prácticas reales de elevado costo. En cada práctica virtual aparece un resumen, donde se especifica si es filmada o elaborada a través de applets, aparecen además los fundamentos teóricos y las orientaciones metodológicas necesarias para su realización. (Sistema Interactivo Didáctico para la Enseñanza de la Física.)

- Dosis Calculador.

Es un entrenador virtual de cálculo de dosis para clases prácticas de Farmacología Veterinaria. Este trabajo se realizó en el Laboratorio de Computación de la Facultad de Medicina Veterinaria en la Universidad de Granma el cual trajo consigo la elaboración de un software DosisCalculator como una experiencia innovadora a utilizar en la asignatura de Farmacología Veterinaria.

- MultiH 3.0 virtual

La Hidráulica como ciencia empírica requiere para su estudio del apoyo del laboratorio. Los laboratorios físicos para este fin son complejos y resultan costos, teniendo además el inconveniente de estar limitado en espacio, tiempo y alternativas. Las nuevas TICs constituyen una herramienta útil para el desarrollo de nuevas formas de la enseñanza y el aprendizaje ; dentro de ellas , los procesos virtuales que simulen los procesos reales casos de estudio constituyen una fortaleza que ninguna Universidad debe pasar por alto .

MultiH 3.0 simula en cada instrumento de medición las mismas características que su similar real y por tanto incorpora en la medición los mismos errores sistemáticos y aleatorios que en el proceso real. Esto permite que cada repetición de la misma medición sea semejante a la anterior pero no igual.

MultiH 3.0 contiene una poderosa ayuda en línea con teoría y las ecuaciones empíricas de cada uno de los procesos que se ejecutan. Además contiene un detallado procedimiento experimental para la guía de cada practicante. Los

formatos de cada práctica, para la toma de información y procesamiento de los datos están insertados en páginas de texto.

En la versión para estudiantes de Ingeniería o Licenciatura (IL) el banco de tuberías tiene un diámetro y un material de tubería fijo en cada uno de sus cinco niveles, igual que un banco real; el sistema bombas simples y serie-paralelo esta conformado por equipos iguales.

En la versión para postgrados (P) hay diámetros de tuberías desde 12.5 mm hasta 100 mm y tres diferentes materiales para cada diámetro, equipos iguales o diferentes en el banco de bombas; los tanques de aforo tienen áreas diferentes, los manómetros en U diferentes opciones de líquidos manométricos y las bombas que abastecen a los canales paralelos y al banco de tuberías tienen diferentes características Q-H.

Con los experimentos de Laboratorios se han logrado resultados superiores en la formación del profesional partiendo de lo educativo, lo instructivo y el vínculo con la vida laboral, social y cultural; organizando así un trabajo metodológico dirigido a recuperar contenido y valores en el proyecto Educativo Integrador en el cual nuestro país se encuentra emergido.

1.3 ¿Que es una Herramienta de Autor?

Las herramientas de autor o metamedios son aplicaciones que permiten un trabajo multimedia y constructivista para generar un entorno de aprendizaje dinámico, dentro de las funcionalidades que este tipo de herramientas presentan se puede destacar la posibilidad de crear actividades o pequeñas aplicaciones desde la misma herramienta.

Las herramientas de autor proveen generalmente módulos desde los cuáles se pueden organizar actividades o se puede interconectar pequeños componentes y se pueden adecuar a los objetivos, los conocimientos y habilidades que se busque desarrollar por parte del autor.

Las herramientas de autor más básicas son aquellas que solamente permiten un conjunto limitado de acciones para que el usuario interactúe con el sistema, como por ejemplo, crear mapas sensibles o ir de una diapositiva a la siguiente, ejemplo de estas herramientas puede ser Kiss u OpenOffice.org Impress o mapas Web sensibles en conjuntos de páginas html con javascript.

Las herramientas de autor (también denominados entornos de autor o lenguajes visuales) son aplicaciones informáticas que permiten elaborar sistemas multimedia. Ofrecen un entorno de trabajo que permite una programación basada en iconos, objetos y menús de opciones, los cuales posibilitan al usuario realizar un producto multimedia (como, por ejemplo, un libro electrónico) sin necesidad de escribir una sola línea en un lenguaje de programación. Los iconos u objetos se asocian a las exigencias del creador, de tal modo que existen iconos para reproducir sonidos, mostrar imágenes (gráficos, animaciones, fotografías, vídeos), controlar dispositivos y/o tiempos, activar otros programas, crear botones interactivos, etc.

Por otra parte, los menús asociados a cada icono u objeto permiten la selección de opciones sin necesidad de recurrir a un lenguaje de programación, al menos para las posibilidades que la propia herramienta incluye directamente. Para programar acciones que son imposibles de realizar mediante los menús de cada icono u objeto, la herramienta de autor posee un lenguaje específico que permite superar las limitaciones o restricciones del programa.

Este tipo de software puede ser aplicado al desarrollo de programas educativos ya que permite la creación de aplicaciones en las que, de forma sencilla y rápida se tiene la posibilidad de cambiar el flujo de la información según las necesidades del alumno, relacionar palabras, incluir cuestionarios y marcadores que evalúen los conocimientos alcanzados, activar animaciones y vídeos explicativos, incorporar sonidos y lenguaje hablado, etcétera.

1.4 Entornos Virtuales de Aprendizajes

Los nuevos avances de la informática y las comunicaciones han generado un nuevo espacio social, la creación de un entorno virtual para desarrollar el área de la educación, el cual posibilita nuevos procesos de aprendizaje y transmisión del conocimiento a través de las redes modernas de comunicaciones.

Este entorno cada día adquiere más importancia, porque para ser activo en el nuevo espacio social se requieren nuevos conocimientos y destrezas que habrán de ser aprendidos en los procesos educativos. Mediante el mismo se crea un nuevo sistema de centros educativos, a distancia y en red, así como nuevos escenarios, instrumentos y métodos para los procesos educativos.

Este método influye en el conocimiento humano debido al grado de interactividad, que hace que el usuario se motive. En lugar del lápiz, el cuaderno, los lápices de colores, y los libros de texto, los actuales utensilios educativos del aula virtual está formado por la pantalla, el ratón, el teclado, la multimedia, la pizarra electrónica, el software y los videojuegos.

El profesor, puede ahora proponer una serie de problemas previamente diseñados, controlar a distancia lo que hacen los alumnos, corregirles interviniendo en su pantalla, sugerirles ideas, motivarles, pero todo ello en un nuevo medio que no es físico, sino electrónico.

Se puede decir que un entorno virtual requiere de mucha fuerza de voluntad, concentración y un saber-hacer específico. El entorno ha de ser evolutivo, presentando

sólo los elementos imprescindibles al alumno, de manera que otros nuevos se vayan incorporando a medida que se necesitan, preferiblemente con la jerarquía que él prefiera.

El entorno ha de ser neutro, como la pizarra en el aula: no importa si es verde o gris, pero no ha de molestar ni ser protagonista, porque es sólo un medio: lo más útil, eficaz, sencillo y desapercibido posible. Claro está, iniciativas formativas diferentes exigen configuraciones diferentes, y no se trata sólo de una cuestión estética o de imagen corporativa.

Muchas instituciones están desarrollando programas de teleformación. Diversas universidades, como la de Princeton en Estados Unidos, están creando campus virtuales, y han surgido universidades específicas del Entorno cibernético, como la Universidad Abierta de Cataluña que recién visité en España o el Instituto Tecnológico de Monterrey en México, que ha coordinado actividades con al Fundación Ventura de nuestro país.

Durante los próximos años proliferarán iniciativas de este tipo y no cabe duda de que otra clase de agentes sociales como organizaciones religiosas, instituciones internacionales o plurinacionales se incorporarán al proceso de creación de escenarios educativos en aulística virtual. [10]

1.5 Tendencias y tecnologías actuales

En la actualidad la humanidad atraviesa por un momento único y crucial en la historia de la informática, no se puede ignorar ser protagonista de la mayor revolución tecnológica antes vista. Incontables proyectos se llevan a cabo en diferentes ramas socioeconómicas como en la educación semipresencial, la biomedicina y en el campo empresarial; en el cual se han desarrollado diversas herramientas, que son perfeccionadas cada día con las nuevas tecnologías.

En este epígrafe se muestra un análisis de las principales tecnologías que se consultaron para la realización de este trabajo.

1.5.1 Modelo Cliente Servidor

El término cliente/servidor es originalmente aplicado a la arquitectura de software que describe el procesamiento entre dos o más programas: una aplicación y un servicio soportante; puede definirse como:

Cualquier combinación de sistemas que pueden colaborar entre si para dar a los usuarios toda la información que ellos necesiten sin que tengan que saber donde esta ubicada.

Es una arquitectura de procesamientos cooperativo donde uno de los componentes pide servicios a otro.

Es un procesamiento de datos de índole colaborativo entre dos o más computadoras conectadas a una red.

IBM define al modelo Cliente/Servidor. "Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores".

Es un modelo para construir sistemas de información, que se sustenta en la idea de repartir el tratamiento de la información y los datos por todo el sistema informático, permitiendo mejorar el rendimiento del sistema global de información.

¿Qué es un cliente?

Es el que inicia un requerimiento de servicio. El requerimiento inicial puede convertirse en múltiples requerimientos de trabajo a través de redes LAN o WAN. La ubicación de los datos o de las aplicaciones es totalmente transparente para el cliente.

¿Qué es un servidor?

Es cualquier recurso de cómputo dedicado a responder a los requerimientos del cliente. Los servidores pueden estar conectados a los clientes a través de redes LANs o WANs, para proveer de múltiples servicios a los clientes y ciudadanos tales como impresión, acceso a bases de datos, fax, procesamiento de imágenes, etc.

Características del modelo cliente/servidor

- El Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.
- Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma.
- Un servidor da servicio a múltiples clientes en forma concurrente.
- Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
- La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no

muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.

- El servidor no necesita tanta potencia de procesamiento, parte del proceso se reparte con los clientes.
- Se reduce el tráfico de red considerablemente. Idealmente, el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre.

La arquitectura cliente-servidor llamado modelo cliente-servidor o servidor-cliente es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realiza se efectúe con la mayor eficiencia, y permita simplificarlas.

La estrategia tradicional de utilizar aplicaciones compactas causa gran cantidad de problemas de integración en sistemas software complejos como pueden ser los sistemas de gestión de una empresa o los sistemas de información integrados consistentes en más de una aplicación. Estas aplicaciones suelen encontrarse con importantes problemas de escalabilidad, disponibilidad, seguridad, integración.

Para solventar estos problemas se ha generalizado la división de las aplicaciones en capas que normalmente serán tres: una capa que servirá para guardar los datos (base de datos), una capa para centralizar la lógica de negocio (modelo) y por último una interfaz gráfica que facilite al usuario el uso del sistema.

La programación por capas es un estilo de programación en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario.

La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles y en caso de algún cambio sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.

Además permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles.

En el diseño de sistemas informáticos actual se suele usar las arquitecturas multinivel o Programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten).

El diseño más en boga actualmente es el diseño en tres niveles (o en tres capas).

- Capa de presentación.
- Capa lógica.
- Capa de datos.

En una arquitectura de tres niveles, los términos capas y niveles no significan lo mismo ni son similares.

El término "capa" hace referencia a la forma como una solución es segmentada desde el punto de vista lógico: Presentación/ Lógica de Negocio/ Datos.

En cambio, el término "nivel", corresponde a la forma en que las capas lógicas se encuentran distribuidas de forma física. Por ejemplos:

- Una solución de tres capas (presentación, lógica, datos) que residen en un solo ordenador (Presentación+lógica+datos). Se dice, que la arquitectura de la solución es de tres capas y un nivel.
- Una solución de tres capas (presentación, lógica, datos) que residen en dos ordenadores (presentación+lógica, lógica+datos). Se dice que la arquitectura de la solución es de tres capas y dos niveles.
- Una solución de tres capas (presentación, lógica, datos) que residen en tres ordenadores (presentación, lógica, datos). La arquitectura que la define es: solución de tres capas y tres niveles.

Ventajas de la arquitectura 3 capas

- Centralización de los aspectos de seguridad y transaccionalidad, que serían responsabilidad del modelo.
- No replicación de lógica de negocio en los clientes: esto permite que las modificaciones y mejoras sean automáticamente aprovechadas por el conjunto de los usuarios, reduciendo los costes de mantenimiento.
- Mayor sencillez de los clientes.

1.5.2 Tecnologías para el desarrollo de aplicaciones Web

Para el desarrollo de aplicaciones Web se han generado múltiples tecnologías entre las que se encuentra los lenguajes de lado servidor que son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Por otro lado, los lenguajes de lado cliente (entre los cuales no sólo se encuentra el HTML sino también el Java y el JavaScript los cuales son simplemente incluidos en el código HTML) son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pretratamiento.

Programación del lado del cliente

Existen diversas tecnologías y/o lenguajes para llevar a cabo la programación del lado del cliente, a continuación se describen dos lenguajes de programación principalmente por su amplia utilización: Javascript y Visual Basic Script (VBScript). También se encuentran los controles ActiveX y los applets de Java que mejoran la apariencia de una página Web.

Java Script

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

Con Javascript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

Javascript es el siguiente paso, después del HTML, que puede dar un programador de la Web que decida mejorar sus páginas y la potencia de sus proyectos. Es un lenguaje de programación bastante **sencillo y pensado para hacer las cosas con rapidez**, a veces con ligereza. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con sólo un poco de práctica.

Entre las acciones típicas que se pueden realizar en Javascript tenemos dos vertientes. Por un lado los **efectos especiales** sobre páginas Web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, javascript nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear **páginas interactivas** con programas como calculadoras, agendas, o tablas de cálculo.

Javascript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. Además, Javascript pone a disposición del programador todos los elementos que forman la página Web, para que éste pueda acceder a ellos y modificarlos dinámicamente.

Con Javascript el programador, que se convierte en el verdadero dueño y controlador de cada cosa que ocurre en la página cuando la está visualizando el cliente.

Visual Basic Script

Es un lenguaje de programación de scripts del lado del cliente, pero sólo compatible con Internet Explorer. Es por ello que su utilización está desaconsejada a favor de Javascript.

Está basado en Visual Basic, un popular lenguaje para crear aplicaciones Windows. Tanto su sintaxis como la manera de trabajar están muy inspirados en él. Sin embargo, no todo lo que se puede hacer en Visual Basic lo podremos hacer en Visual Basic Script, pues este último es una versión reducida del primero.

El modo de funcionamiento de Visual Basic Script para construir efectos especiales en páginas Web es muy similar al utilizado en Javascript y los recursos a los que se puede acceder también son los mismos: el navegador.

Como decimos, no debemos utilizar este lenguaje en la mayoría de las ocasiones, aunque un caso donde tendría sentido utilizar Visual Basic Script sería la construcción de una Intranet donde sepamos con toda seguridad que los navegadores que se van a conectar serán siempre Internet Explorer. En este caso, un programador habitual de Visual Basic tendría más facilidades para realizar los scripts utilizando Visual Basic Script en lugar de Javascript.

Programación del lado del servidor

Para desarrollar páginas dinámicas del lado del servidor existen varias tecnologías y/o lenguajes como Common Gateway Interface (CGI) comúnmente escritos en Perl, Active Server Pages (ASP), Hypertext Preprocessor (PHP), Java Server Pages (JSP), Python entre muchos otros.

PHP

Es un lenguaje de programación usado generalmente para la creación de contenido para sitios Web. Las siglas significan "**PHP** Hypertext **P**re-processor" (inicialmente PHP Tools, o, *Personal Home Page Tools*), y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web.

El fácil uso y la similitud con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores experimentados crear aplicaciones complejas con una curva de aprendizaje muy suave.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de Aplicaciones Web muy robustas.

PHP corre en (casi) cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en algo así como en 25 plataformas, incluyendo diferentes versiones de Unix, Windows (95,98,NT,ME,2000,XP,etc) y Macs. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al OS.

PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen yTHTTPD. Otra alternativa es configurarlo como modulo CGI.

Ventajas de PHP

- Es un lenguaje multiplataforma.
- Modulable.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Nos permite crear los formularios para la Web.
- Miles de ejemplos y código fuente disponible.
- Perfecta integración del Apache-PHP-MySQL.
- Sintaxis clara y bien definida.
- Soporte para varios servidores Web.
- Soporte nativo para prácticamente cualquier Base de Datos.

Nuevas funcionalidades del PHP5

- Presenta un ambiente de programación orientada a objetos mucho más completo.
- Con PHP4 había mil maneras de trabajar con los errores, ahora PHP5 trae un manejo de Excepciones.
- PHP5 viene, por defecto, con la base de datos SQLite habilitado. Esto permitirá distribuir aplicaciones que corren inmediatamente con ésta base de datos.

- El manejo de los documentos XML no ha sido muy establecido en PHP4, en cambio, PHP5 trae una nueva extensión de XML basada en Gnome XML Parser y libXML 2. Y también PHP5 ha introducido un nuevo analizador de XML, se trata del SimpleXML, el cual es bastante fácil de usar.

Perl

Acrónimo de (Practical Extracting and Reporting Language), que viene a indicar que se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros. Es un lenguaje de programación muy utilizado para construir aplicaciones CGI para la Web.

No establece ninguna filosofía de programación concreta. No se puede decir que sea orientado a objetos, modular o estructurado aunque soporta directamente todos estos paradigmas; su punto fuerte son las labores de procesamiento de textos y archivos.

No es ni un compilador ni un intérprete, está en un punto intermedio, cuando mandamos a ejecutar un programa en Perl, se compila el código fuente a un código intermedio en memoria que se optimiza como si se fuera a elaborar un programa ejecutable pero es ejecutado por un motor, como si se tratase de un intérprete.

Es un lenguaje de programación basado en scripts portable a casi cualquier plataforma. Es muy utilizado para escribir CGIs. Uno de sus elementos más potentes son las expresiones regulares, que a partir de su versión en Perl han sido adoptadas por otros lenguajes y plataformas como .NET o JavaScript.

Perl es free software, y está licenciado bajo la Artistic License y la GNU General Public License. Está disponible para la mayoría de sistemas operativos. Está especialmente extendido en Unix y en sistemas similares (como en Linux, FreeBSD, y Mac OS X), y su popularidad está creciendo en sistemas Microsoft Windows.

JSP

Acrónimo de (Java Server Pages) es una tecnología para generar páginas Web de forma dinámica en el servidor, desarrollado por Sun Microsystems, basado en scripts que utilizan una variante del lenguaje java.

La tecnología JSP, o de Java Server Pages, es una tecnología Java que permite a los programadores generar dinámicamente HTML, XML o algún otro tipo de página Web.

La principal ventaja de JSP frente a otros lenguajes es que permite integrarse con clases Java (.class) lo que permite separar en niveles las aplicaciones Web, almacenando en clases java las partes que consumen más recursos así como las que requieren más seguridad, y dejando la parte encargada de formatear el documento html en el archivo jsp.

Además Java se caracteriza por ser un lenguaje que puede ejecutarse en cualquier sistema, lo que sumado a jsp le da mucha versatilidad.

Sin embargo JSP no se puede considerar un script al 100% ya que antes de ejecutarse el servidor Web compila el script y genera un servlet, por lo tanto se puede decir que aunque este proceso sea transparente para el programador no deja de ser una aplicación compilada. La ventaja de esto es algo más de rapidez y disponer del API de Java en su totalidad.

Python

Es un lenguaje de programación interpretado e interactivo, capaz de ejecutarse en una gran cantidad de plataformas.

Permite dividir el programa en módulos reutilizables desde otros programas y viene con una gran colección de módulos estándar que se pueden utilizar como base de los programas (o como ejemplos para empezar a aprender Python). También hay módulos incluidos que proporcionan E/S de ficheros, llamadas al sistema, sockets y hasta interfaces a GUI (interfaz gráfica con el usuario) como Tk.

Python es un lenguaje interpretado, lo que ahorra un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar. El intérprete se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo del programa. También es una calculadora muy útil.

Asp

ASP (Active Server Pages) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página Web, utilizando el lenguaje Visual Basic Script o Jscript (JavaScript de Microsoft).

La mayor desventaja que presenta este lenguaje es que solo se puede implementar en los Servidores Web de su desarrollador: Microsoft. Actualmente se ha presentado ya la segunda versión de ASP: el ASP.NET, que comprende algunas mejoras en cuanto a posibilidades del lenguaje y rapidez con la que funciona. ASP.NET tiene algunas diferencias en cuanto a sintaxis con el ASP, de modo que se ha de tratar de distinta manera uno de otro. Para implementarlo es necesario montar en el Servidor la Plataforma .NET.

Características de ASP

- Las páginas pueden ser programadas en Visualscript, Jscript y Perl
- Uso bajo Licencia propietario
- Solo Servidores y/o equipos con sistema Operativo Windows de Microsoft.
- Invoca más frecuentemente los objetos.
- Realiza numerosas tareas sirviéndose de componentes (objetos) que deben ser comprados (o programados) por el servidor a determinadas empresas especializadas.
- Buen Nivel de seguridad.
- Conocimientos previos básicos de programación en Visual.
- Precisa que el servidor funcione sobre Windows NT, Windows 2000 o superiores.
- Para emular en Sistemas operativos Open Source deben cancelarse la licencia (ejemplo: chilisoft).
- Código cerrado, solo visto por sus desarrolladores.
- Buena documentación

1.5.3 Sistemas de Gestión de Bases de Datos

Los SGBD son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. [17]

Un sistema gestor de base de datos, es un sistema software capaz de proporcionar un sistema de almacenamiento de datos de forma mas o menos ordenada y transparente al usuario (paradigma), proporcionando a su vez, mecanismos para el manejo coherente de esos datos (Interfaz). En este (y en la mayoría de los casos actuales) tratamos con un sistema basado en paradigma relacional, de gran solera, pero ampliamente usado hoy en día, con un interfaz de acceso SQL, también mayoritariamente usado.

Objetivos

- Abstracción de la información
- Independencia
- Redundancia mínima
- Consistencia
- Seguridad
- Integridad
- Respaldo y recuperación
- Control de la concurrencia

Base Dato: Es un conjunto de información almacenada en memoria auxiliar que permite acceso directo y un conjunto de programas que manipulan esos datos.

MySQL

Es un sistema de gestión de base de datos, multihilo y multiusuario con más de seis millones de instalaciones. MySQL como software libre presenta un esquema de licenciamiento dual.

Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi (vía dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby, REALbasic (Mac), FreeBASIC, y Tcl; cada uno de estos utiliza una API específica. También existe un interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL.

MySQL es muy utilizado en aplicaciones Web como MediaWiki o Drupal, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación Web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

MySQL funciona sobre múltiples plataformas, incluyendo AIX, BSD, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell Netware, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista y otras versiones de Windows.

Las siguientes características son implementadas únicamente por MySQL:

- Múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, BDB, Memory/heap, MySql Cluster, Federated, Archive, CSV, Blackhole y Example en 5.x), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

PostgreSQL

Es un motor de base de datos, es servidor de base de datos relacional libre, liberado bajo la licencia BSD. Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2.

Algunas de sus principales características son:

Las funciones permiten subir bloques de código que se ejecuten en el servidor. Estas funciones pueden escribirse en una variedad de lenguajes, algunos de los más importantes son PL/pgSQL, C, C++ y Java.

Puede definirse si las funciones serán ejecutadas con los permisos del llamador o del usuario que definió la función....

Mediante un sistema denominado MVCC (Acceso concurrente multiversión) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

PostgreSQL provee nativamente soporte para:

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas)
- Direcciones IP (IPv4 e IPv6).
- Bloques de direcciones estilo CIDR.
- Direcciones MAC.
- Arreglos.

Microsoft SQL Server:

Es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Entre sus características figuran:

- Soporte de transacciones.
- Gran estabilidad.
- Gran seguridad.
- Escalabilidad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo accedan a la información.
- Además permite administrar información de otros servidores de datos.

Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle o Sybase ASE.

Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para la mayoría de las plataformas de desarrollo, incluyendo .NET.

Microsoft SQL Server, al contrario de su más cercana competencia, no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft.

1.5.4 Servidores Web

Un servidor Web es un programa que implementa el protocolo HTTP (hypertext transferprotocol) y se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas Web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. [20]

Actualmente existen 2 servidores Web que tienen predominio total o casi total según estadísticas y estudios realizados por diferentes instituciones Apache e Internet Information Services (IIS).

Apache

Apache Web Server, es un servidor de páginas Web desarrollado por la Apache Software Foundation, uno de los más populares servidores. Según algunas estimaciones es utilizado para hosting por más del 50% de los sitios Web en todo el mundo.

Apache fue desarrollado para proveer un alto grado de calidad y fortaleza para las implementaciones que utilizan el protocolo HTTP, es un servidor de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras.

Dentro de sus puntos fuertes se encuentran:

- Tiene interfaz con todos los sistemas de autenticación.
- Facilita la integración como "plug-ins" de los lenguajes de programación de páginas Web dinámicas más comunes.
- Tiene integración en estándar del protocolo de seguridad SSL.(más utilizado)
- Provee interfaz a todas las bases de datos.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido.

Internet Information Services:

Internet Information Services (IIS) es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del *Option Pack* para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a un computador en un servidor de Internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas Web tanto local como remotamente (servidor Web).

El servidor Web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

Existen otros servidores Web que pueden usarse como alternativa a este que es de propiedad de Microsoft Corporation, por ejemplo Apache, Cherokee que son desarrollados en Software Libre y otros muchos.

1.5.5 Tendencias actuales

En este epígrafe se muestra un análisis de las principales tecnologías que se consultaron a la hora de realizar este trabajo.

CSS

Las hojas de estilo en cascada (*Cascading Style Sheets*, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).

Este permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo.

Funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne.

h1 {color: red;} h1 es el selector {color: red;} es la declaración

Las principales ventajas de utilizar CSS:

- Control centralizado de la presentación de un sitio Web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio Web remoto, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.

- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser "leída" por un sintetizador de voz.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

XML

Es un Lenguaje de Marcado Extensible (eXtensible Markup Language) muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.

Entre las tecnologías XML disponibles se pueden destacar:

[XSL](#): Lenguaje Extensible de Hojas de Estilo, cuyo objetivo principal es mostrar cómo debería estar estructurado el contenido, cómo debería ser diseñado el contenido de origen y cómo debería ser paginado en un medio de presentación como puede ser una ventana de un navegador Web o un dispositivo de mano, o un conjunto de páginas de un catálogo, informe o libro.

[XPath](#): Lenguaje de Rutas XML, es un lenguaje para acceder a partes de un documento XML.

[XLink](#): Lenguaje de Enlace XML, es un lenguaje que permite insertar elementos en documentos XML para crear enlaces entre recursos XML.

[XPather](#): Lenguaje de Direccionamiento XML, es un lenguaje que permite el acceso a la estructura interna de un documento XML, esto es, a sus elementos, atributos y contenido.

[XQL](#): Lenguaje de Consulta XML, es un lenguaje que facilita la extracción de datos desde documentos XML. Ofrece la posibilidad de realizar consultas flexibles para extraer datos de documentos XML en la Web.

Este lenguaje se creó para que cumpliera varios objetivos:

- Que fuera idéntico a la hora de servir, recibir, y procesar la información del HTML para aprovechar toda la tecnología implantada de este.
- Que fuera normal y conciso desde el punto de vista de los datos y la manera de guardarlos.
- Que fuera extensible, para que lo puedan utilizar en todos los campos del conocimiento.
- Que fuese fácil de leer y editar e implementar.
- Que fuese fácil de implantar, programar y aplicar a los distintos sistemas.

AJAX

Acrónimo de Asynchronous JavaScript And XML (JavaScript y XML asíncronos), es una técnica de desarrollo Web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

AJAX es una combinación de tres tecnologías ya existentes:

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor Web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.

- XML es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.

1.6 Fundamentación de la metodología a utilizar

RUP

RUP (Rational Unified Process), es un proceso de desarrollo de software que junto a UML (Unified Modeling Language), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Para desarrollar un software se necesita una forma ordenada de trabajo. Un proceso que integre las múltiples facetas del desarrollo. Se necesita un método común, un proceso que:

- Proporcione una guía para ordenar las actividades de un equipo.
- Dirija las tareas de cada desarrollador por separado y del equipo como un todo.
- Especifique los artefactos que deben desarrollarse.
- Ofrezca criterios para el control y la medición de los productos y actividades de proyectos.

“El Proceso Unificado es un proceso de desarrollo de Software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organización, diferentes niveles de aptitud y diferentes tamaños de proyecto...El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes de software interconectados a través de interfaces bien definidas.”

Sus principales características son:

- Asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Desarrollo iterativo.
- Administración de requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software.

UML

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Se ha convertido en el estándar de facto de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Tiene como objetivo brindar un material de apoyo que le permita al lector poder definir diagramas propios como también entender diagramas ya existentes.

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas

notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real.

1.7 Elección de las herramientas que serán utilizadas.

Con las descripciones dadas en los epígrafes anteriores y con un conocimiento del problema en cuestión se puede definitivamente escoger las herramientas que serán utilizadas para implementar el software que propone este trabajo.

Tomando como base los elementos expuestos de algunas de las herramientas y tecnologías disponibles actualmente, se decidió utilizar PHP como lenguaje de programación por su rapidez, eficiencia, compatibilidad y portabilidad del código apoyado en el uso de PHP Data Objects (PDO) extensión que define una interfase liviana y consistente para el acceso a Bases de datos en PHP, además de ser un lenguaje multiplataforma, poseer una amplia documentación y ser gratuito.

Se utilizó MySQL como Gestor de Base de Datos por su fortaleza, rapidez, escalabilidad, funcionar sobre múltiples plataformas, ser libre y estar muy relacionado con el lenguaje PHP.

Como servidor Web se selecciono Apache por su fiabilidad, extensibilidad, seguridad, ser totalmente gratuito, estar distribuido bajo la licencia Apache Software License, poseer una increíble velocidad de respuesta soportando mayores peticiones por segundo que su homólogo mas cercano IIS y para dar una mayor interactividad y hacer la aplicación mas creativa y llamativa utilizaremos XML, CSS y Ajax por ser tecnologías de punta con múltiples ventajas.

Conclusiones

En este capítulo se describen las características fundamentales de la metodología, lenguaje y herramientas que se emplearon en la modelación del sistema. Se abordaron además las características principales de las tecnologías, lenguajes de programación y gestores de base de datos más utilizados actualmente en el desarrollo de aplicaciones Web, analizando sus características, ventajas y desventajas. Se decidió por el uso de PHP como lenguaje de programación del lado del servidor, MySQL como gestor de bases de datos y Apache como servidor HTTP.



Características del Sistema

Introducción

En el presente capítulo se realiza una descripción del negocio, se identifican los actores y trabajadores del sistema, se definen conceptos que serán agrupados en un Modelo de Dominio y se enumeran los requisitos funcionales y no funcionales con que debe contar el sistema, lo que permite hacer una concepción general del sistema, e identificar mediante un Diagrama de Caso de Uso, las relaciones de los actores que interactúan con el mismo.

2.1 Modelación del Negocio

El módulo propuesto en este trabajo pretende automatizar el proceso de realización de secuencias de actividades en un ambiente en línea para reforzar y ampliar el aprendizaje conceptual de los estudiantes. En su primera versión, el sistema debe dar la posibilidad de crear una secuencia, guardar una secuencia y abrir una existente.

Teniendo en cuenta que la información referida es variada y que la actividad de creación de una secuencia de actividades involucra a varios procesos, se describen a continuación como será su funcionamiento y el actor que interactúa con el mismo.

El profesor es el responsable de la creación de las secuencias de actividades además de poder eliminarlas y modificarlas.

Para la creación de una secuencia de actividades se cuenta con una interfaz gráfica que consta de tres partes:

1. Área de Trabajo: donde se crean y se editan las secuencias de actividades.
2. Barra de Herramientas: para gestionar las secuencias de actividades (Abrir, Nuevo, Guardar).
3. Herramientas: son componentes que permiten la creación de una secuencia de actividades (Foro, Chat, etc.).

Para la creación de una secuencia el profesor tiene que comprobar que en el Área de Trabajo no exista ninguna en caso contrario, se hace click en Nuevo en la Barra de Herramientas para que todo lo que exista en el Área de Trabajo sea eliminado y se puede crear una nueva secuencia de actividades.

Para crear una secuencia se arrastran los componentes hacia el Área de Trabajo y se organizan según el orden de prioridad que les da el profesor y haciendo doble click sobre cada uno se muestra una ventana de propiedades, la que le permite al profesor editar las propiedades del componente.

Si se desea eliminar una componente se arrastra hasta la Papelera y cuando se allá terminado de crear la secuencia hacemos click en Guardar en la Barra de Herramientas.

Si el profesor desea abrir una secuencia existente hacemos click en Abrir en la Barra de Herramientas seleccionamos la secuencia y esta se carga en el Área de Trabajo.

Se debe tener en cuenta que este es un módulo que depende en gran medida del resto de los módulos del sistema, por lo que han salido a relucir requerimientos funcionales, y por tanto casos de uso, que son responsabilidad de otros módulos y que son necesarios para el funcionamiento del que nos ocupa.

2.2 Modelo de Dominio

Debido al bajo nivel de estructuración que presenta el negocio que se está estudiando y que este está altamente centrado en tecnologías informáticas, se propone un modelo del dominio, ya que nos permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Esto ayuda a los usuarios, clientes y desarrolladores e interesados, a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema.

Descripción del modelo del dominio

El modelo del dominio se describe mediante diagramas UML, específicamente con un diagrama de clases conceptuales significativas en el dominio del problema.

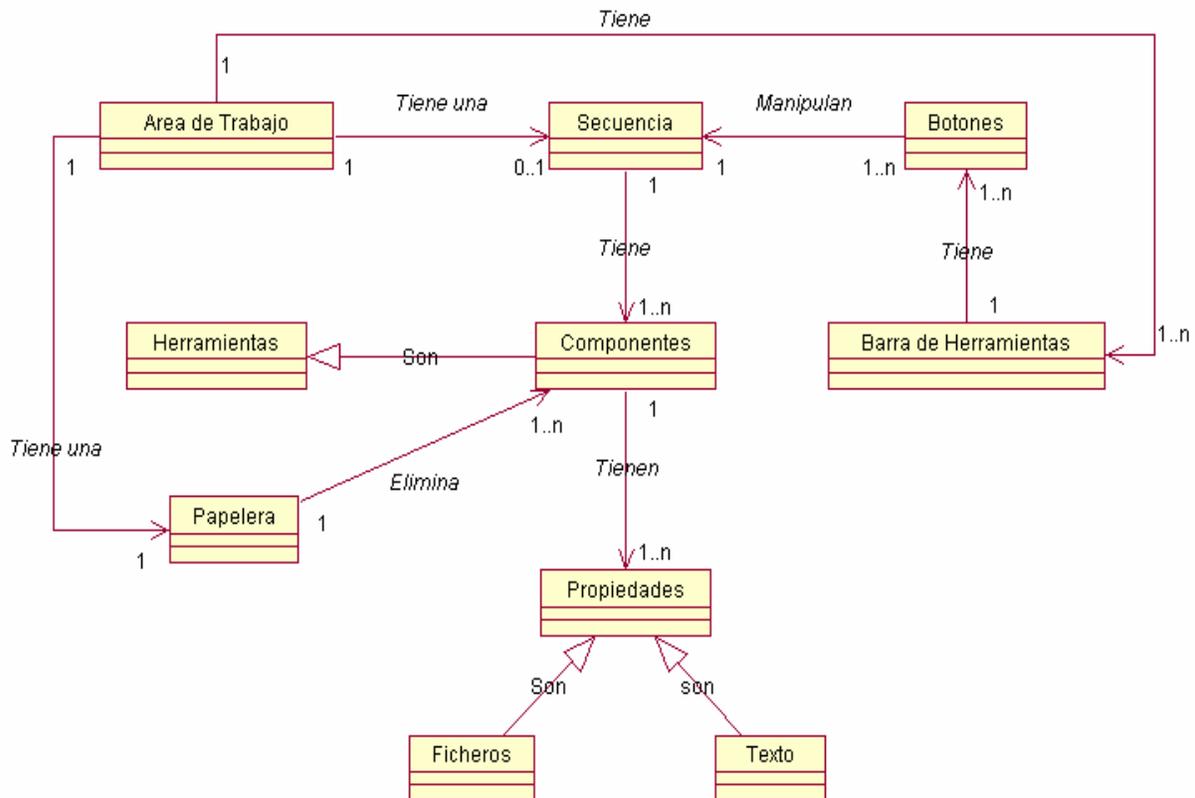


Fig. 2.1 Modelo de Dominio

2.3 Captura de requisitos

A continuación se expondrán los requisitos funcionales y no funcionales del sistema.

Requerimientos Funcionales

Los requerimientos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física. Por lo tanto los requerimientos funcionales especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto.

1- Crear una secuencia

1.1 Editar una secuencia

1.2 Eliminar una secuencia

1.3 Guardar una secuencia

1.4 Modificar los componentes de la secuencia

1.5 Eliminar los componentes de la secuencia

1.6 Actualizar los datos de los componentes

2- Abrir una secuencia

2.1 Crear la secuencia en el Área de Trabajo

2.2 Eliminar la secuencia

2.3 Guardar la secuencia

2.4 Modificar los componentes de la secuencia

2.5 Eliminar los componentes de la secuencia

2.6 Actualizar los datos de los componentes

3- Salvar una secuencia.

3.1 Comprobar que exista una secuencia

3.2 Mostrar la ventana de dialogo Salvar

3.3 Guardar los datos de la secuencia

4- Crear componentes

4.1 Adicionar un componente al Área de Trabajo

4.2 Editar las propiedades del componente

4.3 Elimina el componente

4.4 Actualizar los datos del componente

4.5 Validar los datos

5- Configurar componente

5.1 Mostrar las propiedades del componente

5.2 Validar los datos entrados

5.3 Guardar los datos

Requerimientos no Funcionales

Usabilidad: El sistema podrá ser usado por cualquier tipo de personas que posean conocimientos básicos en el manejo de la computadora y el ambiente Web en sentido general.

Rendimiento: La herramienta propuesta debe ser eficiente, rápida y el tiempo de respuesta debe ser el mínimo posible y estará implementado sobre una tecnología Web, facilitando su uso a través de la red.

Software: Todas las herramientas que se utilicen en el desarrollo del sistema deben estar respaldadas por licencias bajo las condiciones de software libre.

Seguridad: Solo deben acceder a esta área los que estén autorizados para ello. Mantener la integridad de la información, es decir que no sea transformada durante su almacenamiento o transporte.

Portabilidad: El sistema correrá no solo sobre Windows sino también sobre Linux o Unix, de forma tal que no haya dificultad y se pueda llevar a cabo esta acción sin necesidad de efectuar cambios significativos.

Requerimiento de ayuda y documentación en línea: La herramienta no posee ayuda solo cuenta con ciertas aclaraciones sobre algunas opciones del sistema que garantizaran el buen desempeño de los usuarios a la hora de interactuar con el mismo.

Requerimiento de Soporte: Sistema Operativo Linux (Red hat, Debian, Ubuntu). Se recomienda hacer uso de servidores estándares de amplia distribución, con buenas prestaciones, que se integre con eficiencia a PHP y que garantice la estabilidad y rapidez de la aplicación. (Ejemplo: Apache Web Server 2.0 o superior). Servidor de Base Datos, MYSQL 4.1 o superior. Por parte del cliente se requiere un Navegador Web estándar con capacidad de interpretación de JavaScript (Mozilla Firefox, Internet Explorer, etc.).

2.4 Modelo de casos de uso del sistema

El modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones. Representa un esquema donde se recogen las funcionalidades del negocio que se automatizan y determina cómo será utilizado desde el punto de vista del usuario (Actor), pues se construye sobre la base de sus necesidades.

Los actores representan los usuarios del sistema y otras aplicaciones que interactúan con él, es decir, representan terceros fuera del sistema, que interactúan con éste. Estos suelen corresponderse con trabajadores o actores del negocio. Los actores definidos en el sistema son los siguientes:

Actores del Sistema	Justificación
Profesor	Es el encargado de crear una secuencia de actividades

Tabla 1: Actor del sistema

Diagrama de Casos de Uso del sistema

El diagrama de casos de uso del sistema es un diagrama donde se representa la relación existente entre los actores y los casos de uso del sistema.

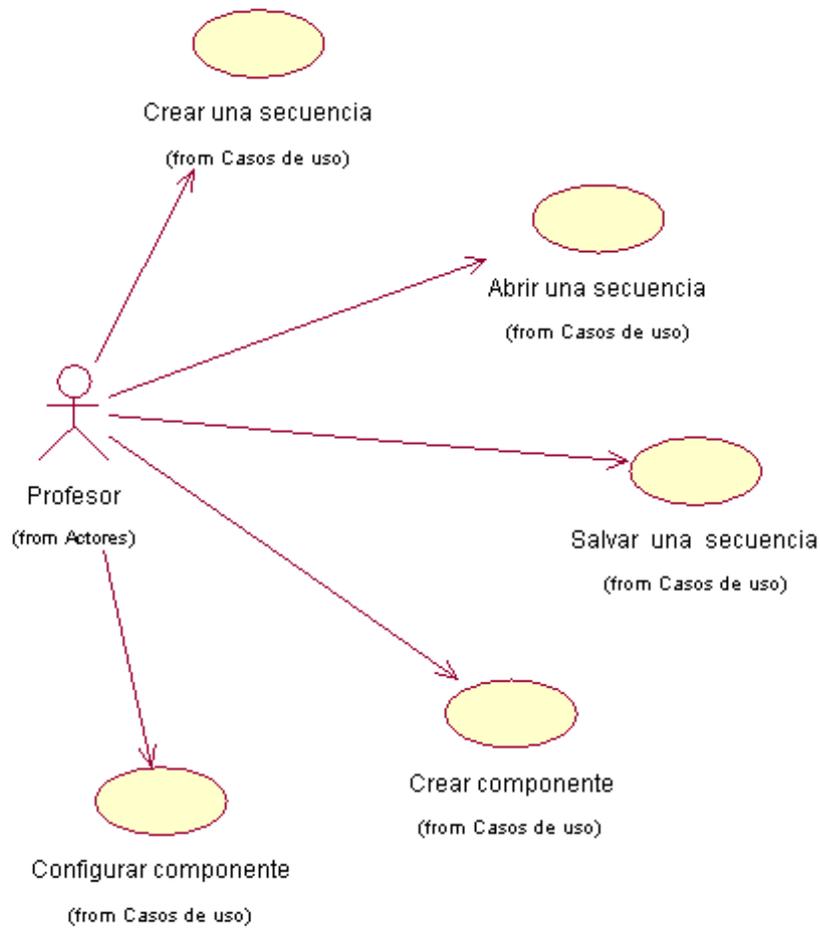


Fig. 2.2 Diagramas de Caso de Uso del Sistema

Descripción de los Casos de Uso del Sistema

A continuación se realiza una descripción de los casos de uso del sistema:

Nombre del Caso de Uso		Crear componente
Actores	Profesor (inicia)	
Propósito	Crear una actividad	
Resumen	El Caso de Uso se inicia cuando el profesor arrastra un componente hacia el Área de Trabajo.	
Referencias	R4	
Precondiciones		
Poscondiciones		
Curso Normal de los Eventos		
Acciones del Actor		Respuesta del Sistema
1. El Profesor arrastra el componente hacia el Área de Trabajo		2. El sistema mediante una forma grafica representa la acción de arrastrar.
3. El Profesor suelta el componente en el Área de Trabajo		4. Crea en el Área de Trabajo una representación grafica del componente.
Flujo alternativo		
Acciones del Actor		Respuesta del Sistema
		3. Si el profesor no suelta el componente en el Área de Trabajo este no se crea.
Prioridad:	Crítico	

Tabla 2: Descripción del caso de uso Crear componente

Nombre del Caso de Uso		Configurar componente
Actores	Profesor (inicia)	
Propósito	Crear una actividad	
Resumen	El Caso de Uso se inicia cuando el profesor hace doble click sobre el componente y aparece la ventana de propiedades para que el profesor entre los datos para crear una actividad.	
Referencias	R5	
Precondiciones	El componente debe estar creado	
Poscondiciones		
Curso Normal de los Eventos		
Acciones del Actor		Respuesta del Sistema
1. El Profesor hace doble click sobre el componente.		2. El sistema inmediatamente abre la ventana de propiedades.
3. El profesor entra los datos.		4. El sistema valida los datos entrados por el profesor.
5. El profesor cierra la ventana de propiedades.		6. El sistema guarda los datos entrados por el profesor.
Flujo alternativo		
Acciones del Actor		Respuesta del Sistema
		4. Muestra un mensaje de error al producirse alguno.
Prioridad:	Crítico	

Tabla 3: Descripción del caso de uso Configurar componente

Nombre del Caso de Uso		Abrir una secuencia	
Actores	Profesor (inicia)		
Propósito	Abrir una secuencia de actividades existentes.		
Resumen	El caso de uso se inicia cuando el profesor habré una secuencia de actividades existente y se muestra en el Área de Trabajo.		
Referencias	R2		
Precondiciones			
Poscondiciones	Debe crearse una secuencia de actividades		
Curso Normal de los Eventos			
Acciones del Actor		Respuesta del Sistema	
1. El Profesor hace click en el botón Abrir		2. El sistema inmediatamente abre una ventana de dialogo.	
3. El profesor selecciona una secuencia y hace click en el botón Abrir.		4. El sistema carga la secuencia. 4.1 Comprueba que la secuencia sea valida. 5. Representa los componentes gráficamente en el Área de Trabajo.	
Flujo alternativo			
Acciones del Actor		Respuesta del Sistema	
		4.1 Muestra un mensaje de error si la secuencia no es valida.	
Prioridad:	Crítico		

Tabla 4: Descripción del caso de uso Abrir una secuencia

Nombre del Caso de Uso		Crear una secuencia
Actores	Profesor (inicia)	
Propósito	Crear una secuencia de actividades.	
Resumen	El caso de uso se inicia cuando el profesor desea crear una nueva secuencia y hace click en el botón Nuevo.	
Referencias	R1	
Precondiciones	Debe existir una secuencia de actividades en el Área de Trabajo	
Poscondiciones		
Curso Normal de los Eventos		
Acciones del Actor		Respuesta del Sistema
1. El Profesor hace click en el botón Nuevo		2. El sistema comprueba que exista alguna secuencia de actividades en el Área de Trabajo. 2.1 Si existe el sistema emite un mensaje que le da la opción de eliminar la secuencia.
3. El Profesor hace clic en el botón aceptar para que sea eliminada la secuencia.		4. El sistema elimina la secuencia.
Flujo alternativo		
Acciones del Actor		Respuesta del Sistema
3. El profesor hace clic en el botón cancelar.		4. El sistema no elimina la secuencia.
Prioridad:	Crítico	

Tabla 5: Descripción del caso de uso Crear una secuencia

Nombre del Caso de Uso		Salvar una secuencia
Actores	Profesor (inicia)	
Propósito	Salvar una secuencia existente en el Área de Trabajo.	
Resumen	El caso de uso se inicia cuando el profesor desea salvar una secuencia de actividades y hace click en el botón Salvar.	
Referencias	R3	
Precondiciones	Debe existir una secuencia de actividades en el Área de trabajo	
Poscondiciones		
Curso Normal de los Eventos		
Acciones del Actor		Respuesta del Sistema
1. El Profesor hace click en el botón Salvar		2. El sistema inmediatamente muestra una ventana de dialogo.
3. El profesor escribe el nombre con el cual desea salvar la secuencia y hace click en el botón Salvar.		4. El sistema comprueba que exista una secuencia de actividades. 4.1 Si existe la secuencia el sistema guarda los datos de la secuencia.
Flujo alternativo		
Acciones del Actor		Respuesta del Sistema
		4.1 El sistema muestra un mensaje.
Prioridad:	Crítico	

Tabla 6: Descripción del caso de uso Salvar una secuencia

2.5 Glosario de Términos

Área de Trabajo: donde se crean y se editan las secuencias de actividades.

Barra de Herramientas: En un programa de una interfaz gráfica de usuario, fila, columna o bloque de botones o iconos en pantalla. Al hacer clic en ellos, los botones o iconos activan determinadas funciones, o tareas, del programa (Abrir, Nuevo, Salvar).

Herramientas: son componentes que se usan para la creación de una secuencia de actividades (Foro, Chat, Verdadero/Falso).

Secuencia: es una relación de componentes en un orden específico.

Componente: es una actividad de una secuencia, que tiene diferentes propiedades.

Propiedades: Característica o parámetro de una clase de objetos o dispositivos.

Texto: Es un ensamblado de signos con una intención comunicativa que adquiere sentido en determinado contexto.

Ficheros: son contenidos elaborados por el profesor o el estudiante en formato (PDF, PPT, WORD, TXT etc.).

Botones: son representaciones gráficas que al dar click sobre ellos realizan una acción determinada.

Papelera: es el lugar donde se eliminan los componentes que son arrastrados hacia ella en Área de Trabajo.

Conclusiones

En el presente capítulo se enumeraron los requisitos funcionales del sistema a partir del análisis de los procesos del negocio y se representaron mediante un Diagrama de Casos de Uso, se identificaron los actores y trabajadores del sistema y se llevo a cabo la descripción de los Casos de Uso.



Descripción de la Solución Propuesta

Introducción

El presente capítulo expone la construcción de la solución propuesta, a través de los flujos de trabajo de diseño e implementación. Primeramente se presenta el modelo de diseño, donde son expuestas las realizaciones de los casos de uso definidos en el capítulo anterior, mediante diagramas de clases del diseño y diagramas de secuencia. Aparece además descrito el diagrama de clases persistentes, así como el modelo de datos obtenido a partir de éste y el diagrama de despliegue. Se especifican los principios del diseño gráfico, los estándares de codificación y el tratamiento de excepciones.

3.1 Patrones de diseño

En la terminología de objetos, el patrón es una descripción de un problema y su solución, este recibe un nombre y puede emplearse en otros contextos.

En el diseño de la aplicación se tuvieron en cuenta principalmente los patrones *Factory*, *Singleton*, *Experto* y *Creador*. El primero de ellos nos permite desde una clase crear instancias de otras clases en tiempo de ejecución, el *Singleton* indica la manera de proceder a la creación de la única instancia de la clase, el *Experto* en información es el principio básico de asignación de responsabilidades, nos indica que la responsabilidad de la creación de un objeto debe recaer sobre la clase que conoce toda la información necesaria para crearlo. El patrón *Creador* nos ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases.

3.2 Modelo de Diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar, constituyendo una entrada principal en la actividad de implementación.

En este modelo, los casos de uso son realizados por las clases del diseño y sus objetos, lo cual se denota por la realización de casos de uso del diseño que describe cómo se realizan estos en particular.

A continuación se muestran los diagramas de clases del diseño y diagramas de secuencia para la realización de los casos de usos descritos en el capítulo anterior.

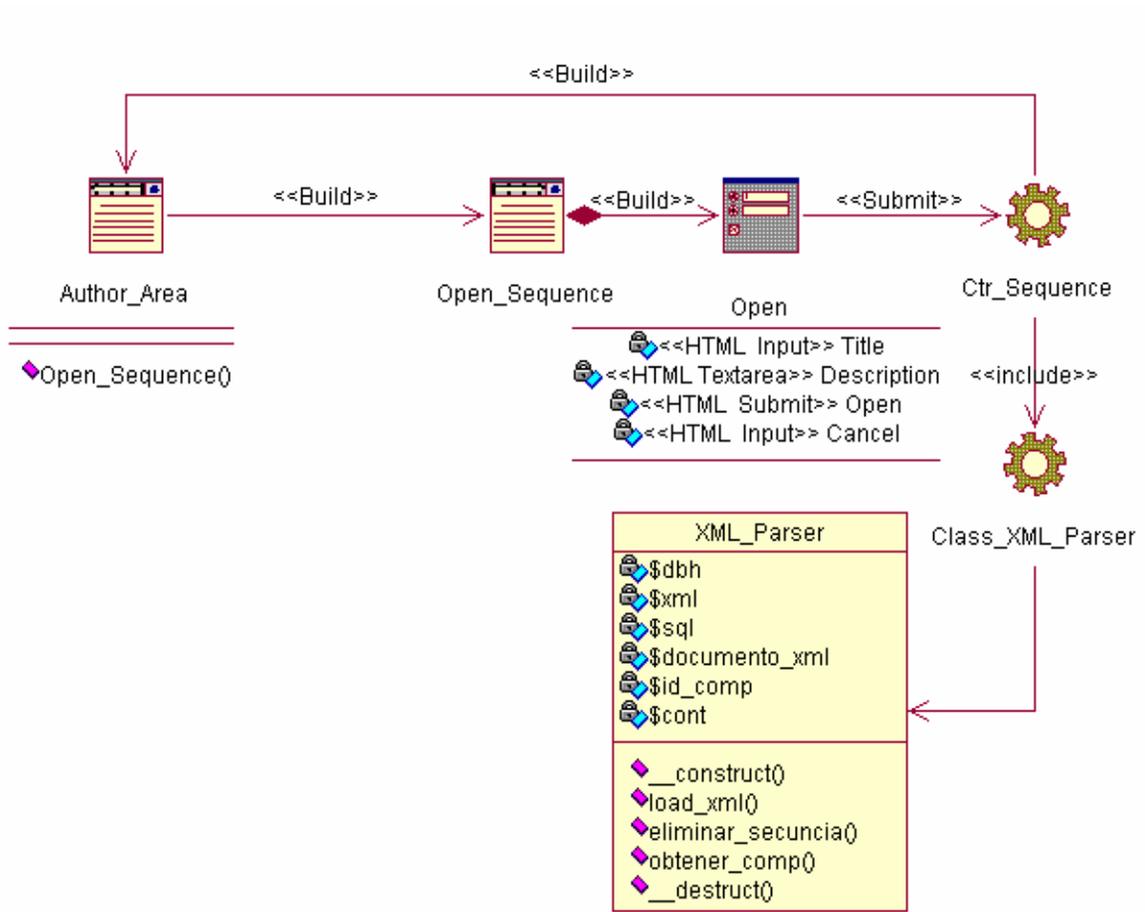


Fig. 3.1 Diagrama de Clases del Diseño Abrir_Secuencia

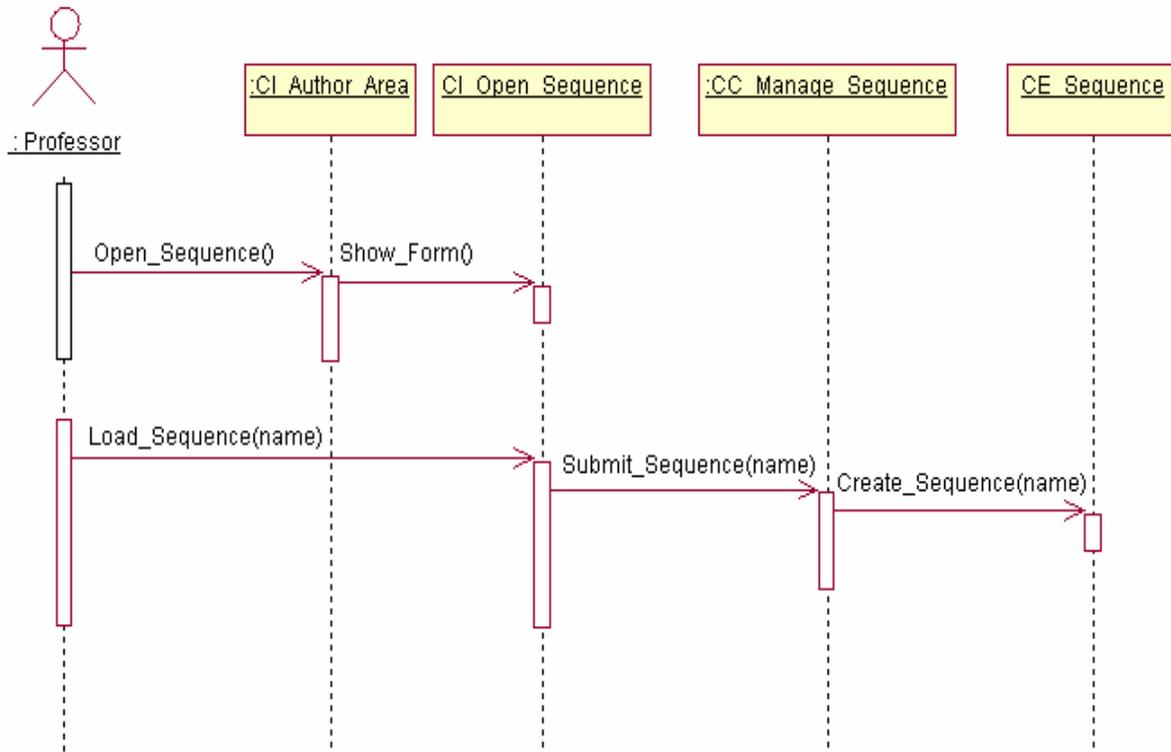


Fig. 3.2 Diagrama de Secuencia Abrir_Secuencia

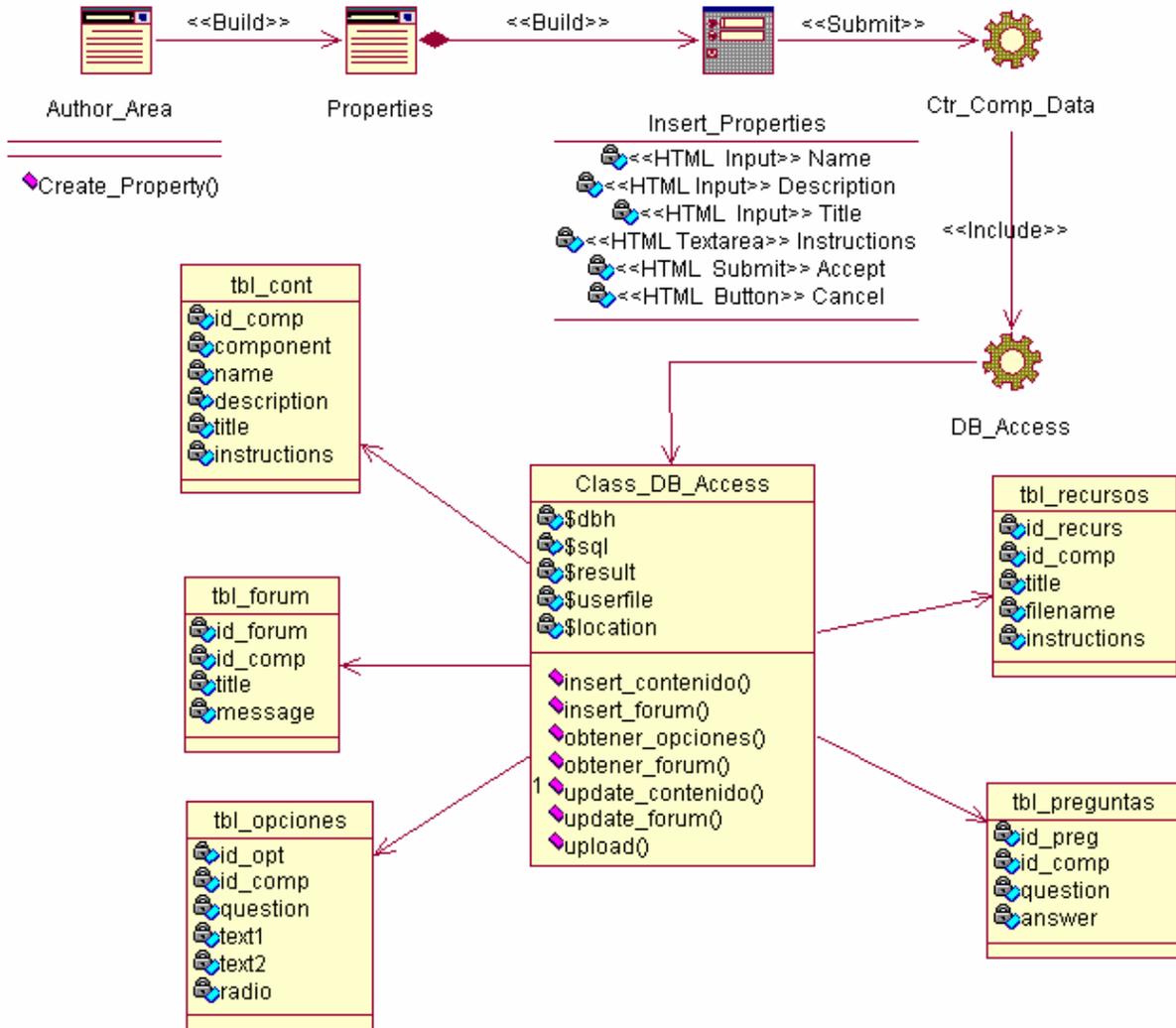


Fig. 3.3 Diagrama de Clases del Diseño Configurar Componente

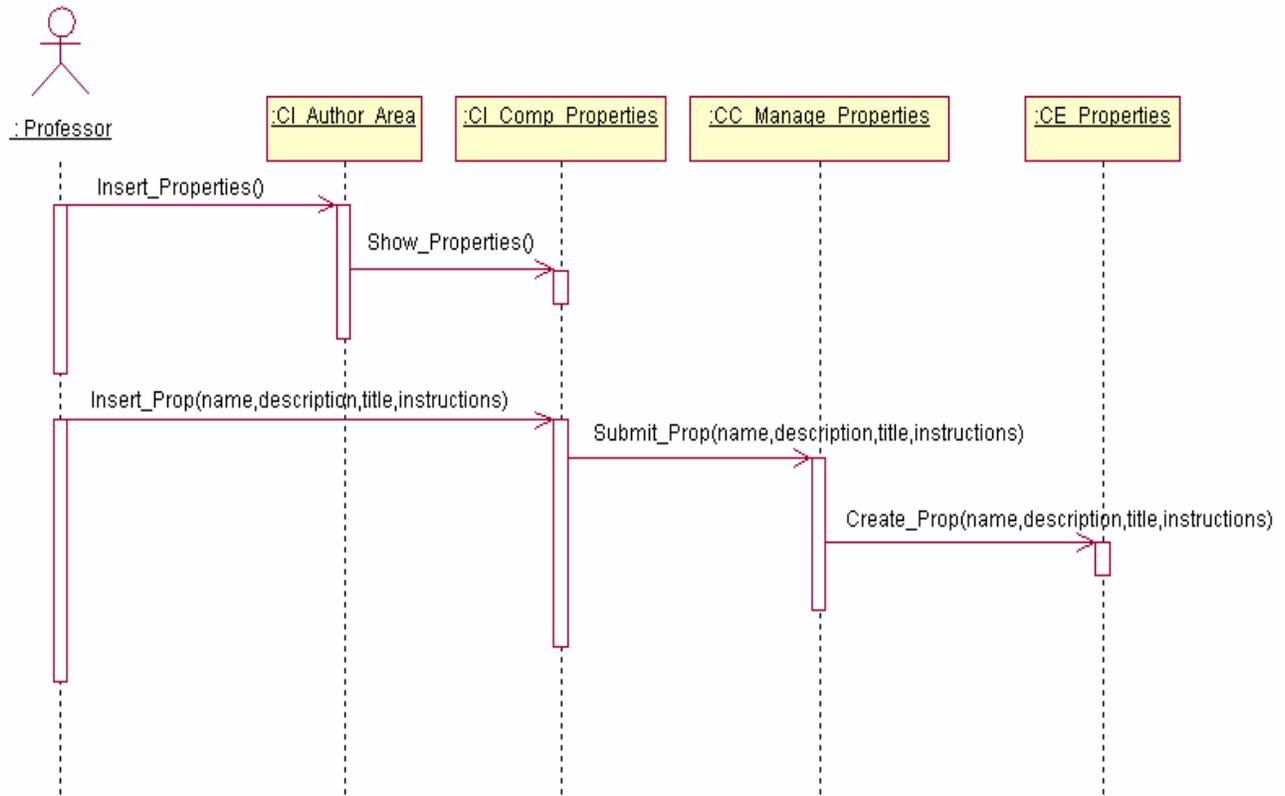


Fig. 3.4 Diagrama de Secuencia Configurar Componente

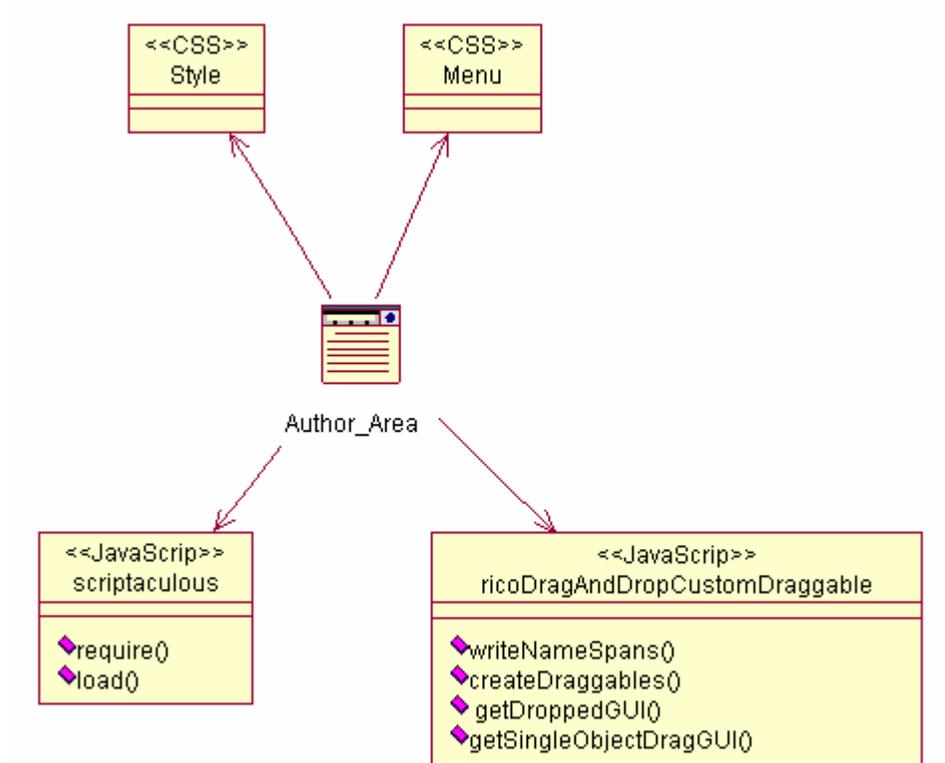


Fig. 3.5 Diagrama de Clases del Diseño Crear Componentes

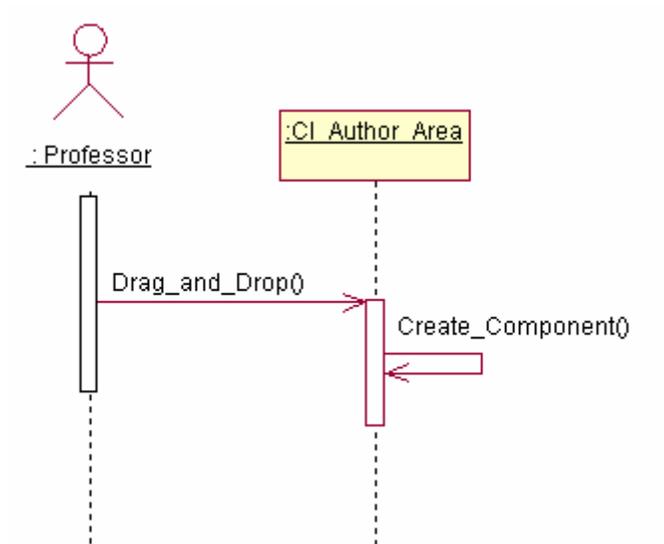


Fig. 3.6 Diagrama de Secuencia Crear Componentes

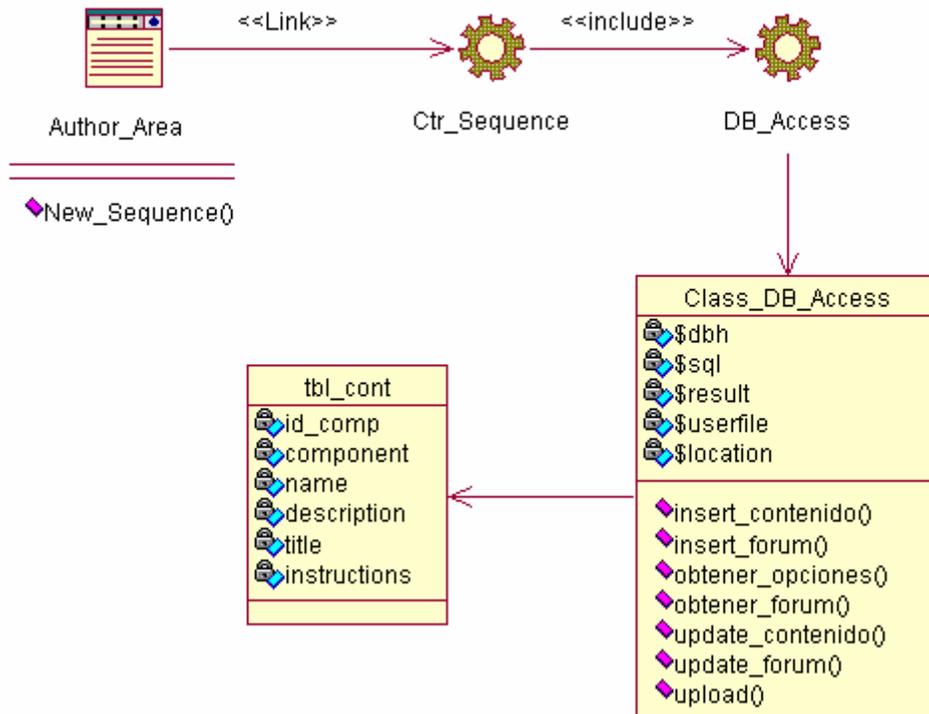


Fig. 3.7 Diagrama de Clases del Diseño Crear una Secuencia

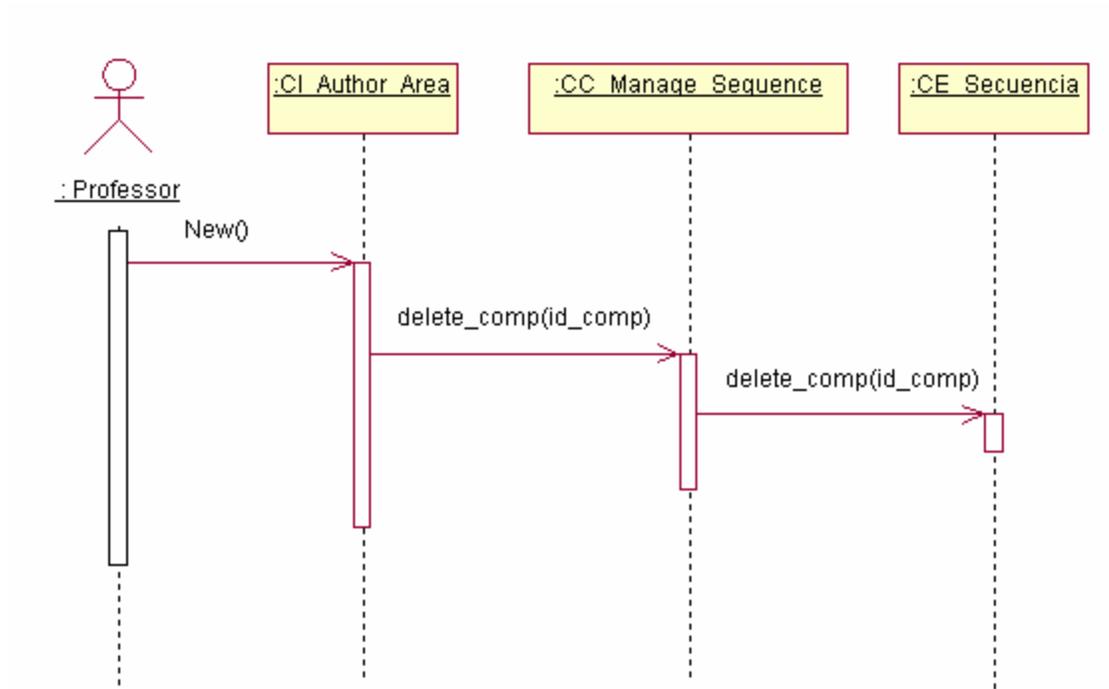


Fig. 3.8 Diagrama de Secuencia Crear una Secuencia

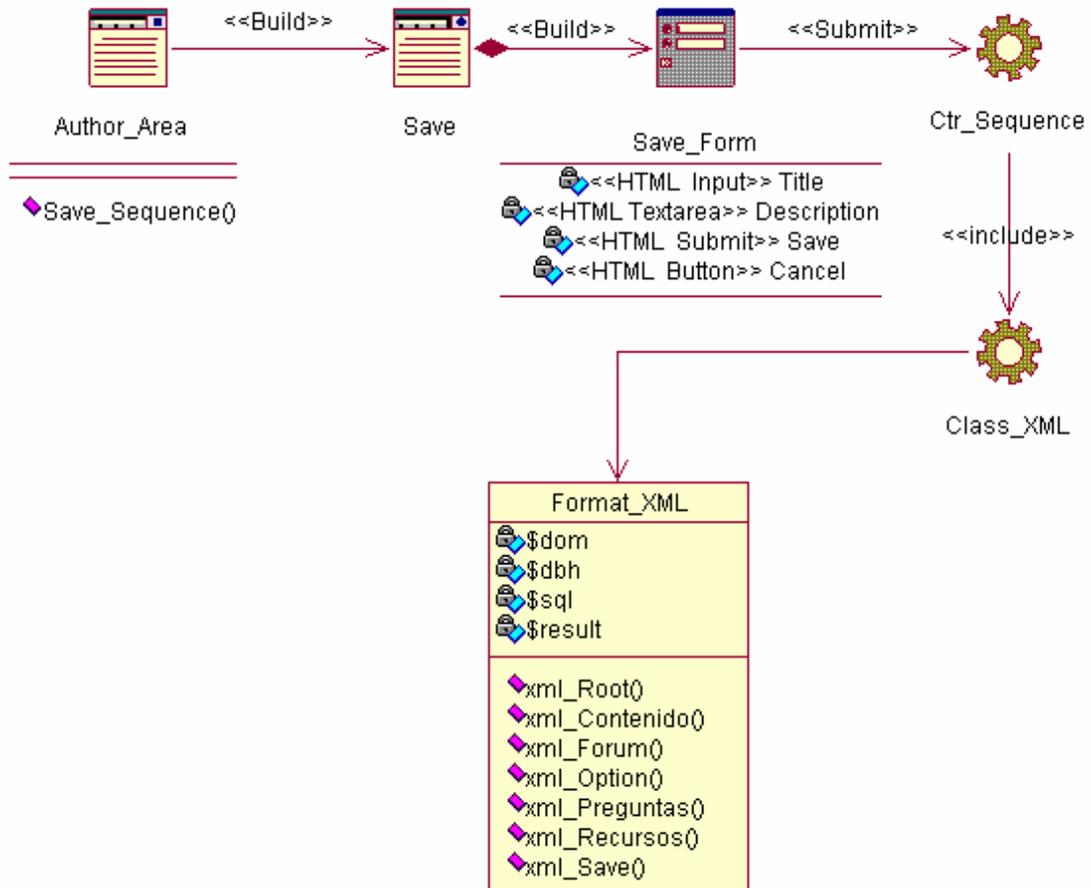


Fig. 3.9 Diagrama de Clases del Diseño Salvar Secuencia

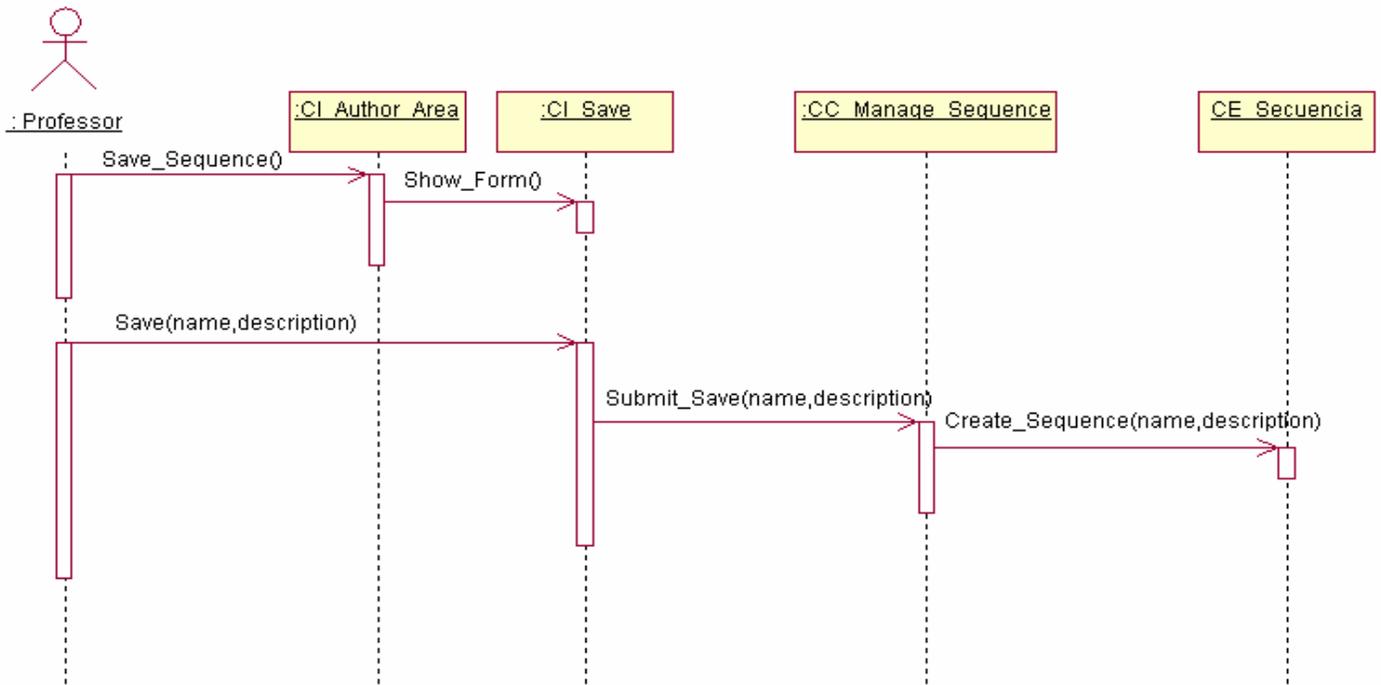


Fig. 3.10 Diagrama de Secuencias Salvar Secuencia

3.3 Diseño de la Base de Datos

Para el diseño de la base de datos del sistema se utilizó el diagrama de clases persistentes y el modelo de datos, representando las clases que simbolizan los datos que se obtienen y almacenan durante los procesos de la aplicación, siendo estos los que pueden modelarse a través de un diagrama de clases persistentes, lo que permitirá ver la relación entre los datos, y completar el modelamiento de la lógica del negocio. A continuación se muestra el diagrama de clases persistentes del sistema, donde aparecen todas las entidades que se manejan en él.

Diagrama de Clases Persistentes.

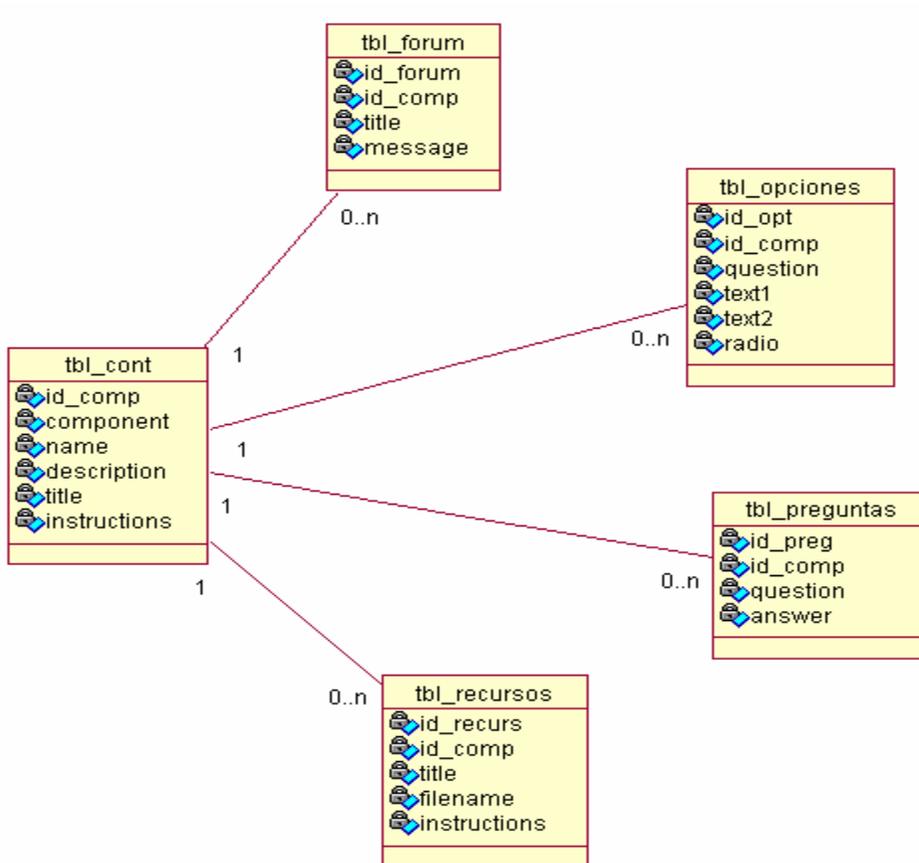


Fig. 3.11 Diagrama de Clases Persistentes.

Modelo de Datos

El modelo de datos se obtuvo a partir del diagrama de clases persistentes.

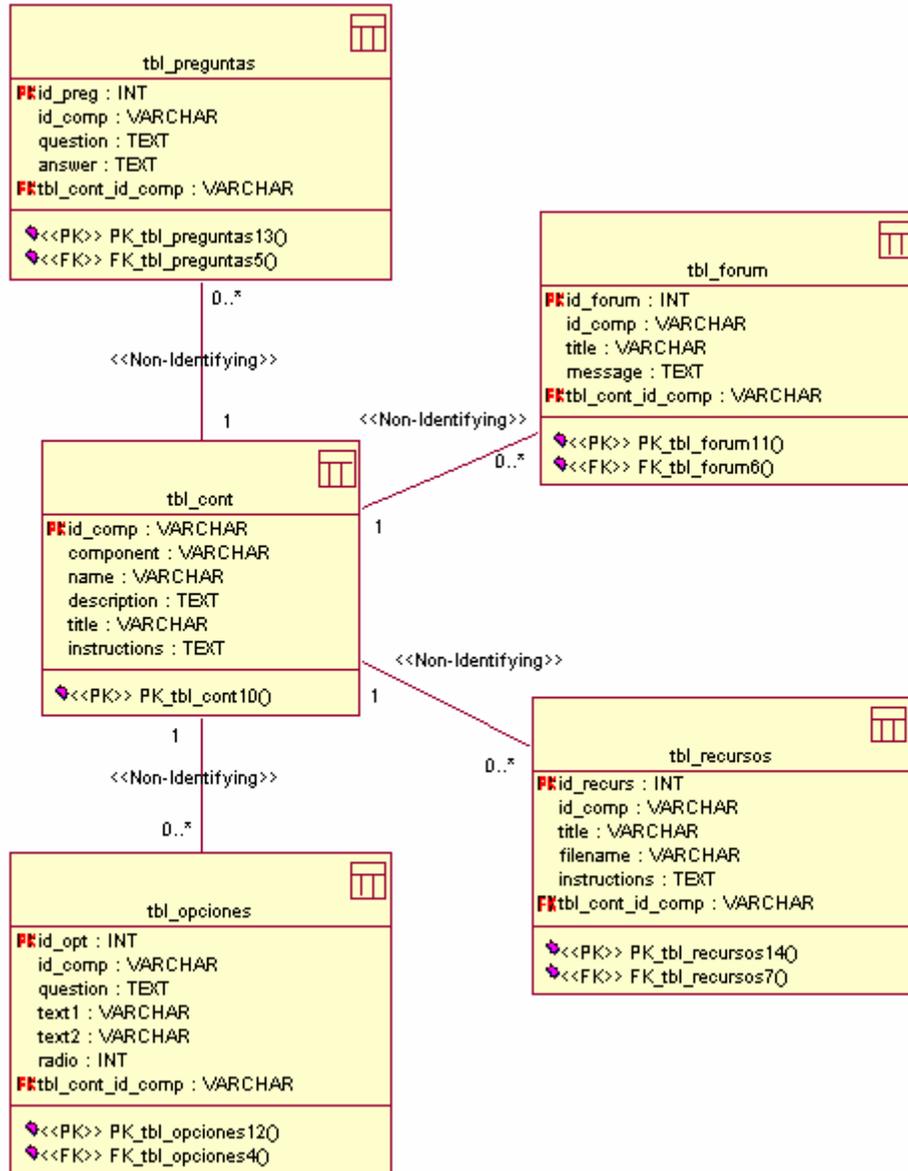


Fig. 3.12 Diagrama del modelo de datos.

3.4 Principios de diseño

Interfaz gráfica de usuario (En inglés Graphic User Interface, también conocido con su acrónimo GUI) es un método para facilitar la interacción del usuario con el ordenador o la computadora a través de la utilización de un conjunto de imágenes y objetos pictóricos (iconos, ventanas, tipografía), que representan la información y las acciones disponibles en la interfaz.

En el diseño de la interfaz del sistema se tuvieron en cuenta aspectos necesarios como una buena navegabilidad, usabilidad, distribución de los contenidos, diseño centrado en el usuario, utilización de estándares Web y un diseño consistente en todas las páginas, para lograr que el usuario se sienta cómodo y logre acostumbrarse rápidamente a la aplicación.

Para su construcción se utilizaron principalmente los colores negro, azul, gris, blanco y amarillo en sus diferentes tonalidades. Se utilizó el azul porque es un color relajante y da muestra de estabilidad y profundidad, el gris por ser un color neutral y provocar la sensación de estabilidad, elegancia y frescura.

Para los textos de los componentes que aparecen en las páginas (labels, nombres de los componentes, botones), se emplearon los estilos de letra: Arial y Verdana, 11pt con color negro teniendo en cuenta las tipografías estándar en diferentes plataformas y sistemas operativos.

Tratamiento de excepciones

Para prevenir errores por parte del usuario, garantizar la estabilidad y confiabilidad de un sistema se realizan validaciones siempre que se introduzcan datos al sistema, de esta forma se impide que el mismo pueda presentar comportamientos inesperados o falle al tratar de procesar datos incorrectos.

Se deben brindar las opciones estrictamente necesarias a la hora de efectuar cualquier operación, además, se realizará en la aplicación de manera general, un tratamiento de excepciones para que los errores cometidos en el lado del cliente, sean atendidos sin

necesidad de llegar al servidor, mostrándole así determinados mensajes que indican las imprecisiones cometidas y además se utilizan mensajes de confirmación, para acciones que son irreversibles como es el caso de las eliminaciones.

Estándares de codificación

Los estándares de codificación son reglas específicas a un lenguaje que reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores.

Para lograr un mejor entendimiento del código en la implementación de la aplicación fue necesario establecer un estándar de codificación. En el caso de los objetos que se utilizan como por ejemplo los textfield su nombre estará asociado al valor que va a contener. Los nombres de las variables utilizadas comienzan en minúscula y son cortos, claros y describen su propósito. El uso de comentarios para facilitar la comprensión del código, sobre todo en procedimientos complejos. Los comentarios pueden ser con fin documental o bien como 'ayuda-memoria'. Se utiliza los estilos de C (*/* */*) y C++ (*//*). Se utilizan los tag `<?php` y `?>` para iniciar y terminar un bloque de código PHP. Las clases serán colocadas en un archivo .php aparte, donde sólo se colocará el código de la clase. Para el caso de los strings o cadenas de texto se usan comillas simples a menos que se necesite hacer interpolación de variables que permiten las dobles. Las llaves para abrir y cerrar un método o un bloque de control de flujo se indentan al mismo nivel del bloque al que pertenecen

3.5 Descripción preliminar del modelo de pruebas

El modelo de pruebas es una colección de casos de prueba, procedimientos de prueba y componentes de prueba, que permite probar los componentes ejecutables en el modelo de implementación. Entre los casos de prueba se puede distinguir dos tipos comúnmente utilizados: las llamadas pruebas de “caja negra” y las de “caja blanca”.

Una prueba de “caja negra” es una prueba del comportamiento observable externamente del sistema, mientras que una prueba de “caja blanca” prueba la interacción interna entre los componentes del sistema.

En el caso de nuestro sistema, se propone la realización de una prueba de “caja negra” para cada caso de uso, para probar la interacción entre el usuario y el sistema, que se satisfagan las precondiciones y poscondiciones, y que se siga la secuencia de acciones intermedias especificadas por el caso de uso. También se propone la realización de una prueba de “caja blanca” para cada realización de caso de uso.

3.6 Modelo de despliegue

El modelo de despliegue es un modelo de objetos, que describe la distribución física del sistema, muestra como están distribuidos los componentes de software entre los distintos nodos de cómputo. Permite comprender la correspondencia entre la arquitectura software y la arquitectura hardware

A continuación se muestra el diagrama de despliegue de la aplicación.

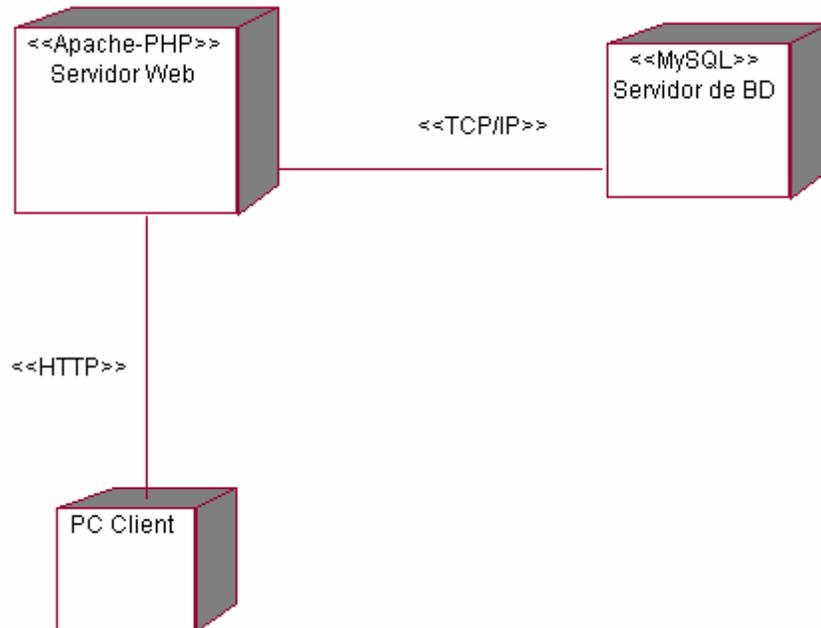


Fig. 3.13 Diagrama de Despliegue

Conclusiones

En este capítulo se ha llevado a cabo la descripción de las clases y demás elementos necesarios para la implementación. Los diagramas elaborados en el capítulo, contribuyeron a lograr un diseño detallado de la aplicación a desarrollar: diagrama de base datos, diagramas de secuencia, diagramas de clases que describen paso a paso el funcionamiento del sitio y diagrama de despliegue que describe la distribución entre los nodos.

Se describen además, los aspectos a tener en cuenta para la confección del diseño de la interfaz de la aplicación, los estándares de codificación a emplear durante la fase de construcción y se hizo una breve descripción del modelo de pruebas que se propone realizar al sistema para comprobar que cumpla los requerimientos que le dieron origen.

Conclusiones Generales

A partir de la investigación realizada para dar cumplimiento a los objetivos de este proyecto, en correspondencia con las exigencias del cliente, fue necesario hacer un estudio de las técnicas, tecnologías y tendencias actuales en relación con el tema de los laboratorios virtuales, utilizando el Proceso Unificado de Desarrollo de Software (RUP) como metodología para lograr una mejor comprensión de los requisitos, llegándose a las siguientes conclusiones:

- A través del estudio que se llevó a cabo se detectó la necesidad de que el profesor diseñe un conjunto o secuencias de tareas de manera on-line para el desarrollo de prácticas de laboratorios virtuales.
- Como respuesta al problema originado, se creó una herramienta de autor sobre Web que permite el diseño de secuencias de tareas para prácticas de laboratorio virtuales.
- El sistema desarrollado está implementado en PHP como lenguaje de programación, como gestor de bases de datos se utilizó MySQL y Apache como servidor HTTP.
- La utilización de esta aplicación contribuye al fortalecimiento del proceso de enseñanza y aprendizaje de la Universidad.

Con la propuesta y el estudio realizado se materializan los objetivos planteados al inicio de esta investigación: Crear una herramienta de autor sobre Web que permite el diseño de secuencias de tareas para prácticas de laboratorio virtuales.

Recomendaciones

Tomando como base la investigación realizada y la experiencia acumulada durante la realización de este trabajo, se proponen las siguientes recomendaciones:

- Incorporar nuevas funcionalidades que le permitan al profesor desarrollar secuencias de actividades con un mayor nivel de complejidad.
- Elaborar un sistema de ayuda sobre Web, en el cual se describa cada una de las funcionalidades de la aplicación y sus propósitos.
- Continuar el estudio con el objetivo de refinar e implementar una herramienta más completa y general.

Referencias bibliográficas

- [1] Tintaya, A. Eliseo
Desafíos y fundamentos de educación virtual <http://www.monografias.com/trabajos13/educvirt/educvirt.shtml> [2006]
- [2] Vázquez García, Constantino y otros
Educación Virtual
<http://www.monografias.com/trabajos24/educacion-virtual/educacion-virtual.shtml> [2007]
- [3] Monge-Nájera, Julián y otros
La evolución de los Laboratorios Virtuales durante una experiencia de seis años con estudiantes a distancia.
<http://www.biologia.ucr.ac.cr/~rbt/public/evolab6.doc> [2006]
- [4] Álvarez Aguilar, Magdalena, Julián Monge-Nájera
Evolución tecnológica de los laboratorios virtuales en la Universidad Estatal a Distancia.
<http://www.uned.ac.cr/Biblioteca/global/administracion/costos/articulos/AguilarMonge.pdf>
[2007]
- [5] Ramírez, Guillermo
Algunas Consideraciones acerca de la educación virtual
http://www.colegiovirtual.org/pr04_page.html [2007]
- [6] Sánchez, Liurkis Dilút. 2006. Principales Impactos de la Sede Universitaria Municipal en el Territorio. *Principales Impactos de la Sede Universitaria Municipal en el Territorio.*
<http://www.monografias.com>[2007]
- [7] Carlos A. Alejandro Alfonso, José M. Perdomo Vázquez. 2000.
Sistema Interactivo Didáctico para la Enseñanza de la Física. Sistema Interactivo Didáctico para la Enseñanza Física.
http://www.quadernsdigitals.net/datos_web/hemeroteca/r_1/nr_636/a_8620/8620.html.
[2007]
- [8] Dr. Alcides León Méndez, M. M. 2004.
Centro de Investigaciones Hidráulicas (C.I.H).
<http://www.cujae.edu.cu/centros/cih/content/index.htm> [2007]

- [9] RODRIGUEZ CALZADILLA, Orlando L e IRIARTE NAVARRO, Leonel.
Laboratorio virtual en anestesiología. Rev. Cubana Estomatol.
http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S003475072001000100002&lng=es [2007]
- [10] DÍAZ, DR. RAFAEL EMILIO BELLO
Educación virtual: Aulas sin Paredes
<http://www.educar.org/articulos/educacionvirtual.asp> [2007]
- [11] Definición arquitectura cliente servidor
<http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-clienteservidor.shtml>. [2007]
- [12] Arquitectura 3 Niveles
http://es.wikipedia.org/wiki/Arquitectura_de_tres_niveles[2007]
- [13] Tecnologías
<http://www.desarrolloweb.com/manuales/15/> [2007]
- [14]. ASP.net ventajas
<http://es.gotdotnet.com/quickstart/asplus/doc/whatisaspx.aspx> [2007]
- [15] PHP Definición.
<http://www.php.net/manual/es/preface.php#contributors> [2007]
- [16]. PHP Ventajas
<http://es.wikipedia.org/wiki/Php> [2007]
- [17] Sistemas de gestión de base de datos (SGBD)
http://es.wikipedia.org/wiki/Sistemas_Gestores_de_Bases_de_Datos [2007]
- [18]Microsoft SQL Server
http://es.wikipedia.org/wiki/SQL_Server [2007]
- [19] PostgreSql
<http://www.postgresql.org.mx/?q=node/6> [2007]
- [20]Servidor Web
http://es.wikipedia.org/wiki/Servidor_web [2007]
- [21] Apache
http://es.wikipedia.org/wiki/Servidor_HTTP_Apache [2007]
- [22] IIS
<http://es.wikipedia.org/wiki/IIS> [2007]

Glosario de términos

A

Aplicación: Es el programa que el usuario activa para trabajar en el ordenador. Existen muchos programas de ordenador que pueden clasificarse como aplicación. Generalmente se les conoce como Software.

Aula virtual: Espacio de aprendizaje en línea donde alumnos y tutores interactúan.

E

Brecha digital: Riesgo de que determinadas personas, grupos y colectivos sociales queden al margen de la sociedad de la información.

C

Campus virtual: Parte de una universidad o facultad que ofrece facilidades educacionales a cualquier hora y desde, potencialmente, cualquier sitio a través de Internet.

Chat o Sala de Chat: Comunicación entre miembros de un grupo en Internet usando texto. Los mensajes se envían entre los participantes en tiempo real, como en una conversación, al escribir oraciones breves.

Comunicación asíncrona: Acción de aprendizaje en la que las personas no están en línea al mismo tiempo, por lo que no pueden tener comunicación sin un cierto espacio de tiempo. Ejemplos: cursos en línea, cursos en CD-ROM, presentaciones Web, clases en vídeo, presentaciones de audio y video, tutorías, grupos de discusión en línea, y el correo electrónico.

Comunicación Sincrónica: Comunicación que permite a los participantes interactuar simultáneamente en tiempo real a través de métodos como el Chat, pizarras electrónicas o videoconferencia.

Contenido: Propiedad intelectual y conocimiento a ser impartido. Los diferentes tipos de contenidos de E-learning incluyen texto, audio, vídeo, animación y simulación.

E

E-Learning: Uso de nuevas tecnologías multimedia y de Internet para mejorar la calidad del aprendizaje mediante el acceso a recursos y servicios, y a colaboraciones e intercambios a larga distancia.

Entorno de aprendizaje virtual: Se refiere a los espacios que componen la interacción on-line con cualquier objetivo, incluyendo el aprendizaje de estudiantes y tutores.

Educación virtual: se entiende por educación virtual o e-learning la utilización de nuevas tecnologías y herramientas de aprendizaje como complemento a procesos sincrónicos y asincrónicos de comunicación y enseñanza, es un sistema y modalidad educativa que surge de la necesidad propia de la educación y tecnología educativa.

F

Formación abierta y a distancia: Posibilidad de efectuar el aprendizaje a distancia, lejos de las aulas y con un alto grado de autonomía, con la ayuda de diversos sistemas, entre los que actualmente destaca el e-learning.

G

Gestión del conocimiento: Organizar y almacenar el conocimiento individual de los trabajadores y grupos de una organización y hacerlo disponible a otros en la organización. La información es almacenada en una base de datos especial llamada base de conocimiento.

H

Herramientas de gestión de contenidos: Aquellas que permiten a los diseñadores de los cursos preparar, subir a la red y gestionar los contenidos de los cursos, páginas, texto, imágenes, animaciones, audio, video, etc.

Herramientas de seguimiento y evaluación: Aquellas que permiten a los profesores y tutores generar exámenes, y realizar el seguimiento de la actividad de los alumnos.

L

Laboratorios virtuales: son imitaciones digitales de prácticas de laboratorio o de campo, reducidas a la pantalla de una computadora.

M

Multimedia: Término que se refiere a los distintos medios a los que se puede acceder, o que se pueden controlar, desde un ordenador (vídeo, sonido, animación, texto, gráficos...).

O

Objeto de aprendizaje: Unidad reusable de información independiente de los medios.
Bloque modular de contenido de teleformación.

R

RUP (Rational Unified Process): Proceso Unificado de Racional es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

S

SGBD: Un Sistema Gestor o Manejador de Bases de Datos (SGBD) es un conjunto de programas que permite a los usuarios crear y mantener una BD, por lo tanto, el SGBD es un software de propósito general que facilita el proceso de definir, construir y manipular la BD para diversas aplicaciones. Pueden ser de propósito general o específico.

T

TIC (Tecnologías de la Información y la Comunicación): Esta expresión engloba el conjunto de tecnologías que conforman la sociedad de la información: informática, Internet, multimedia, etcétera, y los sistemas de telecomunicaciones que permiten su distribución.

Tutor / asesor a Distancia Profesional académico que, utilizando elementos didácticos y manejando las herramientas de telecomunicación a su alcance es capaz de conocer a su asesorado de tal manera que puede identificar sus necesidades, guiar sus actividades, orientar su aprendizaje fomentando el autodidactismo y estimulando la actitud analítica, crítica, constructiva y colaborativa.

U

UML (Unified Modeling Language): Lenguaje de Modelación Unificado es un lenguaje gráfico para especificar, construir, visualizar y documentar las partes o artefactos (información que se utiliza o produce mediante un proceso de software).

V

Virtualidad: Característica de aquello que parece ser real pero no lo es. Posibilidad de que algo lo sea. Cuando se habla de Aprendizaje o Educación Virtuales por tanto, se está utilizando un término incorrecto. Es preferible utilizar los términos Formación basada en Internet, en nuevas tecnologías, en línea, etc.

X

XML (Extensible Markup Language / Lenguaje de marcación extensible): Lenguaje de codificación de última generación, que permite a los diseñadores de webs programar sus propios comandos de marcación. Estos comandos podrán ser usados posteriormente como si fueran comandos HTML estándares.