

Universidad de las Ciencias Informáticas

Facultad 4



"Desarrollo del Portafolio Electrónico para la
Plataforma Educativa ZERA, e-Portafolio."

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

Autores:

Evelyn Gallego Frías
Eduar Smith Guerrero Sánchez

Tutor:

Ing. Nilber Barban Góngora.

*La Habana, 2011
Año 53 de la Revolución.*

Declaración de autoría.

Declaración de autoría.

Declaramos que somos los únicos autores del trabajo “Desarrollo del portafolio electrónico para la plataforma educativa ZERA, e Portafolio” y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Nombre del Autor
Evelyn Gallego Frías

Nombre del autor
Eduar Smith Guerrero Sánchez

Nombre del Tutor
Ing. Nilber Barbán Góngora



“El hombre debe transformarse al mismo tiempo que la producción progresa; no realizaríamos una tarea adecuada si fuéramos tan sólo productores de artículos, de materias primas y no fuéramos al mismo tiempo productores de hombres.”

Che

Agradecimientos

De Evelyn:

Mi primer agradecimiento para quien ha guiado mis pasos a pesar de estar ausente, por haber sido desde siempre el mejor ejemplo, por haberme educado y ser uno de los motivos por los cuales hoy me gradúo, por haberme enseñando a ser mejor ser humano. A mi papá.

A mi mamá por sus buenas enseñanzas, por haber estado siempre pendiente de mí, por convertirme en una buena persona.

A Mimi por su amor sin medida, por sus preocupaciones y su interés constante, por ser la “súper abuela” que todos quisieran tener.

A mi abuela Juana que supo sustituir el cariño de mi papá, y que día tras día se mantuvo atenta de mí.

A mis hermanas porque han servido de ejemplo y motivación para mí.

A mis tíos Angel y Julio y mis tías Oly, Arcidia y Amelia porque siempre confiaron en mí.

A mis primas y primos, en especial a Vilma y Yaíma.

A Annia, que desde el 1er día se convirtió en mi mejor amiga y en mi hermana, y que por 5 años ha sabido mantener una amistad que será insustituible. A Pedro Sáez por haber compartido tantos momentos buenos y malos, y por haberme enseñado un poco más sobre la vida...

A la familia que más voy a extrañar, mis amigos: Roger, Pedro González, Dianamary, Suly, Mile, Lili, Daril, la Mora, Peti, Lizandra, Aldo, el pikete de las fiestas...en fin, todos los que me hicieron creer que este sueño se haría realidad algún día.

A mi primer grupo el 8106, por ser el mejor de los grupos en los que estuve durante 5 años.

A mi compañero de tesis que ante todo es un excelente amigo.

A nuestro tutor Nilber, que es una excelente persona.

Agradecimientos

De Eduar:

Mi principal agradecimiento es para mi mamá que siempre estuvo presente para cualquier problema que se me presentara, porque sin ella no fuera quien soy hoy.

A mi papá por haber estado siempre pendiente de mí, y por darme su apoyo en cualquier momento.

A mis abuelos que son como unos padres para mí, por su enseñanza, apoyo y dedicación en todo momento.

Le agradezco a mi hermano por siempre darme consejos y apoyarme.

Le agradezco mi tío Norge que siempre confió en mi, y me dio su apoyo.

Al hermano Daril que es como mi hermano.

A Rubén que fue un amigo que siempre confió en mí.

A mi tutor Nilber que fue un amigo en cada momento de batalla.

A los profesores que me impartieron clases y los que he conocido en el transcurso de la carrera

A mi compañera de tesis que fue la mejor compañera de tesis, es una excelente amiga.

A Lily , los Pedros , Carlos , Annia , Yasmany , Milena ,Sulamis, Moraima, Yamila ,Leonardo , Jose, Lizandra , René , Santiago ,Ernesto ,Yohandy, Dianamary y a todas las personas que de una forma u otra hicieron posible que hoy alcanzara este sueño.

En general a todos mis amigos que formamos una familia para compartir en las buenas y las malas, al pikete de las fiestas.

Dedicatoria

A Fidel y a la Revolución Cubana.

A nuestros padres.

Resumen

La evaluación como parte del proceso de enseñanza-aprendizaje ha experimentado excelentes resultados en un marco donde los ambientes virtuales juegan un papel decisivo. Una de las vías para llevar a cabo este proceso lo constituyen los portafolios electrónicos. De forma general son empleados para almacenar los resultados de los estudiantes dentro de un entorno informático vinculado a la gestión del aprendizaje, así como las evidencias de su trayectoria durante un curso académico o una asignatura específica. La plataforma educativa ZERA desarrollada en la Universidad de las Ciencias Informáticas no cuenta con un sistema que les permita a los estudiantes acceder a las evidencias evaluadas por el docente y realizar acciones correctivas sobre ellas. Con el presente trabajo se persigue el desarrollo de un portafolio electrónico destinado a la plataforma educativa ZERA. Para ello se realiza un estudio del estado de los portafolios electrónicos a nivel internacional y dentro de nuestro país, así como de las herramientas y lenguajes de programación a emplear, y posteriormente se lleva a cabo el análisis, diseño e implementación del portafolio. A través del desarrollo del presente trabajo se generó un portafolio de evidencias, el cual brinda la posibilidad de acceder a las evaluaciones obtenidas, compartir evidencias con otros usuarios dentro de la plataforma, así como comentar evidencias que hayan sido compartidas con anterioridad. De esta forma el estudiante puede consultar sus evaluaciones, y establecer pautas para mejorarlas.

Palabras clave: ambiente virtual, evaluación, evidencias, portafolio electrónico.

Índice

Introducción	1
Capítulo 1 Fundamentación teórica	6
Introducción	6
1.1 Portafolio electrónico.	6
1.1.1 Propósitos del ePortafolio.	8
1.1.2 Funciones del portafolio electrónico.....	9
1.1.3 Estructura del ePortafolio.....	9
1.1.4 Tipos de ePortafolios	10
1.1.5 Ventajas y desventajas	10
1.2 Sistemas similares.....	11
1.2.1 Portafolios electrónicos desarrollados a nivel internacional.....	11
1.2.2 Los e-Portafolio en Cuba.	14
1.3 Estudio de las metodologías y estándares para el desarrollo del software.....	14
1.3.1 Rational Unified Process (RUP).....	15
1.3.2 Arquitectura.	17
1.4 Estándar IMS e-Portafolio.....	18
1.5 Selección de las herramientas y lenguaje de desarrollo.....	19
1.5.1 Lenguajes de programación.....	19
1.5.2 Lenguaje Unificado de Modelado.....	21
1.5.3 Framework.....	22
1.5.4 Sistema gestor de base de datos.....	23
1.5.5 Entorno de desarrollo integrado.....	24
1.5.6 Herramienta CASE para el modelado UML.....	24

Conclusiones parciales.....	26
Capítulo 2 Características del sistema.....	27
2.1 Modelo de Dominio.....	27
2.1.1 Análisis de los conceptos del Dominio.....	27
2.1.2 Diagrama del Modelo de Dominio.....	29
2.2 Descripción del Sistema Propuesto.....	30
2.3 Requerimientos del software.....	30
2.3.1 Requerimientos Funcionales.....	31
2.3.2 Requerimientos No Funcionales.....	32
2.4 Modelo de Casos de Uso del Sistema.....	33
2.4.1 Descripción de los actores del sistema.....	33
2.4.2 Diagrama de casos de uso del sistema.....	34
2.4.3 Descripción de los casos de uso del sistema.....	34
2.4.4 Patrones de Casos de uso.....	37
Conclusiones parciales.....	37
Capítulo 3: Análisis y Diseño.....	39
Introducción.....	39
3.1 Modelo de análisis.....	39
3.2 Diagrama de clases del análisis.....	39
3.3 Modelo de diseño.....	41
3.3.1 Diagrama de clases del diseño.....	41
3.4 Diagramas de secuencia del diseño.....	42
3.5 Patrón de diseño Modelo Vista Controlador.....	43
3.6 Diseño de la base de datos.....	43
3.6.1 Descripción de las tablas.....	44

Conclusiones parciales	52
Capítulo 4 Implementación y prueba.....	53
4.1 Modelo de implementación.....	53
4.2 Diagrama de despliegue.....	53
4.3 Diagrama de componentes.....	54
4.4 Pruebas de software.....	56
4.5 Pruebas de caja negra.....	56
4.6 Diseño de casos de prueba.....	57
4.7 Resultados obtenidos	62
Conclusiones parciales	63
Conclusiones Generales.....	64
Recomendaciones	66
Referencias Bibliográficas.....	67

Introducción

En la actualidad las Tecnologías de la Información y las Comunicaciones han experimentado un avance vertiginoso en una de las ramas sociales que más impacto tiene en el desarrollo de un individuo: la educación. La organización de la enseñanza y el proceso de aprendizaje han experimentado un gran auge en este sentido. La evaluación, como parte fundamental de todo el proceso de aprendizaje no se ha visto alejada de este increíble impacto en el que las TIC constituyen un medio para marcar una diferencia en cuanto a los métodos tradicionales de evaluación basados en ambientes presenciales.

La nueva concepción de aprendizaje, basada en el uso de las tecnologías, requiere que la evaluación sea uno de sus pilares fundamentales. De igual manera se hace necesario el aporte constante de información que sirva para medir el avance de los estudiantes, y que a la vez pueda ser utilizado por los profesores para llevar a cabo el proceso de enseñanza. En tal sentido es necesario contar con un instrumento que sea capaz de brindar las deficiencias que de forma paulatina se van observando en los estudiantes y poder encaminar acciones en pos de reducir o eliminar dichas dificultades.

Los ambientes virtuales en la actualidad ofrecen grandes posibilidades de evaluar el desempeño académico, ya sea de forma individual o colectiva, ya que propicia el registro de evaluaciones, y el seguimiento constante de las mismas.

Además del proceso de evaluar, uno de los elementos más importantes es la forma de obtener el resultado de dicha evaluación, o sea, tener acceso a las evidencias que ya han sido calificadas en un momento determinado. En tal sentido surge en el mundo de la educación el concepto de portafolios electrónicos (ePortafolio¹), concebidos como una colección organizada de información que describe o documenta logros y aprendizajes de una persona. Los ePortafolios son usados con disímiles finalidades, tales como la búsqueda de trabajo, acreditación de información previa, desarrollo profesional, certificación de competencias, entre otros propósitos.

Un ePortafolio se define por cuatro características básicas: (García Doval, 2005)

¹ A partir de este momento se empleará este término en el documento para hacer referencia a los portafolios electrónicos.

- ✓ Tienen unos objetivos determinados.
- ✓ Se desarrollan para una audiencia en particular.
- ✓ Contienen trabajos realizados, comúnmente conocidos, como evidencias.
- ✓ Incluyen reflexiones personales acerca de las evidencias incorporadas.

De forma general, los ePortafolios cumplen los mismos objetivos de los portafolios tradicionales, pero se ven beneficiados por las ventajas que brindan las tecnologías de la informática y las comunicaciones, lo cual les aporta accesibilidad a un mayor número de personas, una presentación y contenido mucho más interactivo, así como una capacidad de almacenamiento de información superior.

En los últimos tiempos los ePortafolios, han adquirido una connotación mayor. Buena parte de este impulso se debe a que su uso reduce en gran medida el engorroso trabajo de archivar, clasificar, y ordenar los materiales acumulados.

Entre las principales ventajas que muestran en cuanto a sus usos, se encuentran:

- ✓ Compartir experiencias en el proceso de aprendizaje con profesores y estudiantes.
- ✓ Fomentar el profesionalismo.
- ✓ Mantener un registro de los trabajos realizados durante un tiempo específico.
- ✓ Comentar y hacer observaciones respecto a los trabajos almacenados.

En Cuba, la Universidad de las Ciencias Informáticas (UCI), surgida bajo el calor de la batalla de ideas, ha mantenido desde su creación la misión de formar profesionales comprometidos con el proceso revolucionario cubano y con un vasto conocimiento de la rama de la informática. En tal sentido se ha creado un modelo de formación en el que se vincula el estudio con la producción y la investigación. La UCI es uno de los centros donde se propicia el desarrollo de software vinculado a la rama educacional. Reconocida por sus excelentes resultados en este sentido, asume en el año 2009 la creación de una plataforma para la gestión del aprendizaje, ZERA, cuyos orígenes se enmarcaban en una concepción pedagógica denominada hiperentornos de aprendizaje. Determinar necesidades cognitivas y brindar un seguimiento constante para alcanzar mejoras notables, forman parte de la concepción de los hiperentornos de aprendizaje. En el caso específico de ZERA, el módulo Resultados, concebido en un principio para perseguir esta finalidad, no contaba con todas las funcionalidades necesarias para llevar a cabo un proceso de evaluación del estudiante, ya que sólo almacenaba los resultados de los estudiantes

en la plataforma. Es decir, no contaba con los recursos necesarios para establecer el avance de los estudiantes en todas las actividades de un período académico. De igual manera no se encuentra disponible la opción de consultar las evaluaciones que posee un estudiante, ni la de consultar sus evidencias después de haber sido evaluadas.

En el año 2010 se realizó una propuesta de portafolio de evidencias, cuya idea fundamental se basaba en el establecimiento de una relación estrecha entre las calificaciones y las evidencias subidas a la plataforma por parte del usuario.

Teniendo en cuenta la situación antes planteada, se propone como **problema de investigación** ¿cómo los estudiantes pueden acceder a sus evaluaciones y evidencias dentro de la plataforma educativa ZERA? Para dar solución a la problemática anterior se plantea el siguiente **objetivo general**: Desarrollar un módulo para la plataforma educativa ZERA que permita consultar las evaluaciones, compartir, comentar y ver las evidencias creadas, funcionando de esta manera como un portafolio electrónico.

Este problema se enmarca en el **objeto de estudio** la evaluación en el proceso de enseñanza-aprendizaje.

Teniendo como **campo de acción** un portafolio de evidencias integrado a la plataforma educativa ZERA.

Para dar cumplimiento al objetivo general se plantean como **objetivos específicos**:

- ✓ Realizar el estudio del estado del arte.
- ✓ Definir un diseño del módulo que garantice un correcto funcionamiento del portafolio.
- ✓ Implementar el módulo e Portafolio teniendo en cuenta los requisitos capturados.
- ✓ Validar la implementación del portafolio de evidencias.

Se plantea como **idea a defender** que si se implementa un portafolio de evidencias para la plataforma ZERA, esta contará con un sistema donde se propicie la evaluación constante y se genere un ambiente colaborativo que brinde mejores resultados en el proceso de enseñanza- aprendizaje. De esta forma los estudiantes podrán compartir sus evidencias y comentar las que hayan sido compartidas para él. Además de contar con la posibilidad de almacenar sus evidencias.

Al finalizar el presente trabajo se espera sean alcanzados los siguientes resultados:

- ✓ Un portafolio electrónico vinculado a la plataforma educativa ZERA.
- ✓ El establecimiento de un ambiente colaborativo entre estudiantes y profesores que empleen la plataforma ZERA.

Para la realización del trabajo de diploma se proponen las tareas de investigación que a continuación se relacionan:

- ✓ Investigación acerca de la creación de portafolios electrónicos en el mundo, y análisis de cuáles son las similitudes y diferencias entre ellos.
- ✓ Documentación de toda la información referente al estudio del estado del arte.
- ✓ Investigación acerca de las metodologías, estándares, herramientas y lenguajes de desarrollo a utilizar.
- ✓ Realización de la captura de los requerimientos para conocer cómo debe funcionar el sistema.
- ✓ Confección de los Diagramas de casos de uso.
- ✓ Confección de los diagramas de interacción, guiados por los patrones de diseño.
- ✓ Realización de los diagramas de clases del diseño para los casos de uso.
- ✓ Realización del modelo de datos del módulo a desarrollar.
- ✓ Realización del modelo de clases del análisis.
- ✓ Análisis de la propuesta de arquitectura para la plataforma ZERA.
- ✓ Implementación de las clases del análisis, basado en los requerimientos.
- ✓ Integración de la solución obtenida al resto de la plataforma ZERA.
- ✓ Diseño de los casos de prueba.
- ✓ Realización de las pruebas al módulo y al código para medir la calidad del mismo.
- ✓ Documentación de los resultados obtenidos al realizar las pruebas al sistema creado.

Durante todo el transcurso de la investigación y realización del presente trabajo se tomarán en cuenta los métodos científicos y de investigación que se mencionan a continuación:

Métodos teóricos:

Dentro de los métodos teóricos se seleccionó el histórico-lógico, para conocer el avance y la evolución

que han adquirido los portafolios electrónicos, desde su inserción en la rama educacional, como una forma más de evaluar al estudiante. El método analítico-sintético fue utilizado para desarrollar el marco teórico, ya que proporciona un análisis amplio de los conceptos incluidos. Permitió además, definir con claridad los temas vinculados con los portafolios electrónicos. La modelación aportó los datos necesarios para realizar el diseño del módulo e Portafolio, a través de la abstracción y la posterior representación de la realidad.

Métodos Empíricos:

A través de la observación se pudieron recoger datos importantes durante el desarrollo del proyecto. Mediante el análisis estático se realizaron pruebas al software con el fin de descubrir errores y poder llegar satisfactoriamente a la implementación del portafolio electrónico para la plataforma educativa ZERA.

Estructura capitular:**Capítulo 1: Fundamentación teórica.**

Contiene lo referente a los elementos teóricos que soportan la investigación, se presentan las tecnologías y lenguajes de programación que se ajustan al desarrollo del trabajo, fundamentando su selección en base al estudio realizado. Además se describen las soluciones similares existentes.

Capítulo 2: Características del sistema

Se realiza el modelamiento del negocio y el levantamiento de los requisitos. Se obtiene como artefacto fundamental, el diagrama de casos de uso del sistema.

Capítulo 3: Análisis y Diseño del módulo e Portafolio.

Se realiza el análisis y el diseño del sistema, aportando una visión clara del producto.

Capítulo 4: Implementación y prueba del módulo e Portafolio.

En este capítulo son abordados todos los aspectos relacionados con el desarrollo. Se construyen los diagramas de secuencia y de despliegue, y se describe cómo está implementado el sistema.

Capítulo 1

Fundamentación teórica

Capítulo 1 Fundamentación teórica

Introducción

Los ePortafolios se han convertido en un instrumento para llevar a cabo la evaluación dentro de los entornos virtuales. Se clasifican en varios tipos, de acuerdo a sus funciones, y tienen propósitos específicos. A nivel internacional existen una gran cantidad de sistemas para la gestión del aprendizaje que incorporan la experiencia de los ePortafolios.

Teniendo en cuenta que en la plataforma ZERA no existe un sistema para consultar las evaluaciones de los estudiantes, a continuación se muestra un estudio realizado en cuanto al estado de los ePortafolios a nivel internacional, así como las herramientas, metodologías y lenguajes de desarrollo que darán paso a la implementación del ePortafolio para la plataforma ZERA.

1.1 Portafolio electrónico.

El empleo de los portafolios, de manera general, tiene sus orígenes en el mundo del arte, y de forma particular en la arquitectura y el diseño. Surgió de la necesidad de demostrar competencias profesionales en un marco laboral. Como otras tantas técnicas esta toma un matiz y concepto diferente en la rama educacional.

De forma general se piensa en el portafolio como una colección de trabajos (generalmente comentados) de considerable relevancia en la trayectoria de una persona. El Diccionario de la Real Academia de la Lengua Española lo define como:

Portafolio (del fr. *portefeuille*): *Cartera de mano para llevar libros, papeles, etcétera.*

En el campo de la educación el portafolio se convierte en un recurso para la enseñanza y la evaluación, que se basa en el análisis constante de la trayectoria docente de un estudiante, y de esta forma en su

avance y desarrollo cognitivo. Dentro del concepto de portafolio de manera general se destacan 2 tipos fundamentales, los portafolios tradicionales y los portafolios electrónicos.

Diversas han sido las definiciones dadas por diferentes autores para el término de ePortafolio. Entre las más completas se encuentra la siguiente:

(Lorenzo, 2005) Un portafolio electrónico es una colección de materiales digitalizados que incluyen demostraciones, recursos y logros que representan a un individuo, un grupo o una institución. Este texto puede comprender texto, gráficos o elementos multimedia que se pueden consultar en Internet o en otro medio electrónico como CD-ROM o un DVD. Un e-Portafolio es más que una simple colección- también sirve como herramienta para administrar, organizar y controlar los trabajos creados con diferentes aplicaciones. Los e-Portafolios incluyen reflexión personal y por lo regular promueven el intercambio de ideas y retroalimentación.

Así el ePortafolio no es más que la recopilación de evidencias digitales a través de documentos, imágenes y proyectos relacionados con la trayectoria ya sea estudiantil o docente del usuario que lo utiliza. Exhibe los esfuerzos de un estudiante durante un curso académico o el transcurso de una materia específica, así como sus logros y progresos. Constituye una herramienta en línea para recolectar, almacenar, compartir y actualizar información.

En la actualidad existen diferentes factores que intervienen de manera directa en el proceso educacional, uno de los más importantes es la evaluación, definida como la etapa cuya finalidad es comprobar de manera sistemática en qué medida se han alcanzado los objetivos planteados con antelación.

Según (Díaz Barriga, 2010) ...la definición moderna de "evaluación educativa" responde a la idea de control tanto hacia el estudiante como hacia el docente, el plan de estudios y la institución educativa. A partir de una aproximación a la historia del aprendizaje, el autor muestra cómo el examen no nace en la práctica educativa ni en la relación educativa, sino como un instrumento social, de vigilancia continua e ininterrumpida.

La evaluación por portafolios emergió a lo largo de los años ochenta, cuando se evidenció que la evaluación de carácter más tradicional no suplía la necesidad de valorar habilidades procedimentales y competencias vinculadas a un contexto específico. Esta forma de evaluación conlleva una nueva visión

del proceso de enseñanza-aprendizaje, integrándose a este, y propiciando que los propios estudiantes conozcan sus resultados y puedan llevar a cabo acciones para regular sus éxitos. De manera general, se podría lograr una mayor autonomía en el proceso de adquisición de habilidades, conocimientos, contenidos, y competencias (Mariel, 2008).

En palabras de (Cully, 2001):

“El portafolio electrónico incorpora la tecnología actual al proceso de evaluación. El portafolio ya no está limitado a la secuenciación lineal; es posible el empleo de capacidades no lineales debidas a los avances en la tecnología. El portafolio electrónico ayuda a condensar el material en formatos que son mucho más manejables como: CD-ROM, diskettes, sitios web, etc. Si los portafolios se encuentran en formato electrónico se posibilita una actualización y gestión del material mucho más sencilla. Los portafolios convencionales suelen materializarse en forma de archivadores y carpetas, y por lo tanto requiriendo copias en papel y otros formatos rígidos como parte del proceso de actualización. El portafolio electrónico elimina la necesidad de dichas copias, se accede a toda la información a través de la pantalla del ordenador” (la traducción es propia).

En el caso de la educación a distancia se ha visto que el ePortafolio es una opción con gran potencial, en cuanto a la información que proporciona relacionada con las evidencias de desarrollo de habilidades cognitivas, comunicativas e interactivas y la posibilidad de tener un cúmulo de información que permite seguir los avances de los alumnos en la construcción y recreación del conocimiento.

En la creación de un ePortafolio es muy importante tener en cuenta los siguientes aspectos:

- ✓ Validar los logros obtenidos por los estudiantes.
- ✓ Ayudar al estudiante a reflexionar sobre su propio trabajo.
- ✓ Proveer un espacio para la comunicación entre el estudiante y el profesor.
- ✓ Brindar la posibilidad de compartir y comentar sus propios trabajos.

1.1.1 Propósitos del ePortafolio.

Estimular a los estudiantes a:

- ✓ Fomentar el pensamiento crítico.
- ✓ Posibilitar los procesos de auto-reflexión.

- ✓ Proveer la responsabilidad de asumir su propio proceso de aprendizaje.
- ✓ Documentar el conocimiento adquirido.
- ✓ Fomentar la comunicación entre estudiantes y profesores.
- ✓ Desarrollar habilidades de comunicación.
- ✓ Brindar al profesor una visión del avance y desarrollo de sus estudiantes.
- ✓ Proveer al estudiante de una herramienta que le permita consultar su desarrollo y evolución a través de los resultados docentes.
- ✓ Evaluar tanto el proceso como el producto.
- ✓ Desarrollar destrezas colaborativas entre los estudiantes. (Prendes Espinosa M. P., 2008)

1.1.2 Funciones del portafolio electrónico

Entre las funciones de un ePortafolio se definen las siguientes:

- ✓ Busca mejorar la calidad docente y es medio de actualización.
- ✓ A través de la reflexión se describe y se analiza los resultados de su evolución profesional, toma conciencia de los factores que influyeron en el éxito o en el fracaso de sus actividades y define áreas y propósitos de mejoramiento.
- ✓ Es un excelente instrumento de auto evaluación y mejoramiento de la calidad.
- ✓ Permite definir los criterios fundamentales para la medición del desempeño estudiantil, obligándolo a pensar en su quehacer diario, a colocar materiales necesarios para clarificar y definir los patrones de calidad.

Es la base de una nueva cultura pedagógica institucional (Banks, 2004).

1.1.3 Estructura del ePortafolio.

En la mayoría de los casos los ePortafolios están estructurados por secciones de trabajo, donde cada una de ellas cumple una función específica para lograr proporcionar al usuario una herramienta íntegra y certera en el desarrollo del proceso de enseñanza-aprendizaje.

Los ePortafolios pueden incluir:

- ✓ Información personal.
- ✓ Historial educativo.

- ✓ Reconocimientos, premios y certificados.
- ✓ Reflexiones y comentarios.
- ✓ Trabajos, tareas, proyectos.
- ✓ Comentarios del profesor.
- ✓ Comentarios de otros estudiantes.
- ✓ Aspiraciones, planes futuros.
- ✓ Presentaciones, artículos.
- ✓ Actividades personales.
- ✓ Evidencias cargadas como respuesta a alguna tarea o proyecto.

Puede contener además fotografías digitales, audio, video, archivos de texto y/o combinaciones de estos formatos (González, 2008).

1.1.4 Tipos de ePortafolios

Existen 3 tipos de portafolios electrónicos según (García Doval, 2005) los cuales son:

Portafolio de procesos: Con este tipo de portafolio se plasma en un documento una secuencia de acontecimientos encaminadas al logro de un objetivo específico.

Portafolio de productos: Con este tipo de portafolio se pretende analizar los resultados de una acción o acciones en este caso académicas. Incluye considerar tanto resultados positivos como negativos y se busca un análisis global teniendo en cuenta los objetivos antes que la simple presentación de resultados exitosos.

Portafolio de muestras: Es donde se recogen los mejores resultados de un proceso. Su principal objetivo es de marketing personal.

En dependencia del tipo de usuario los ePortafolios también pueden ser clasificados en: portafolio del estudiante, portafolio docente.

1.1.5 Ventajas y desventajas

Entre las principales ventajas de un ePortafolio se encuentran:

- ✓ Es de fácil almacenamiento, por lo que su espacio deja de ser físico para convertirse en electrónico.

- ✓ Es de elaboración permanente, lo cual le brinda la posibilidad de ser actualizado o reeditado en cualquier momento.
- ✓ Las computadoras se han convertido en un medio más accesible, y pueden encontrarse instaladas en las escuelas, bibliotecas, salas, entre otros lugares.

Las desventajas que más resaltan son:

- ✓ El portafolio electrónico deberá formar parte de la cultura institucional para ser considerado una herramienta valiosa.
- ✓ El nivel de alfabetización tecnológica no siempre es el idóneo.
- ✓ Los docentes con poca experiencia en el uso y desarrollo del portafolio pueden considerarlo como algo demasiado complejo para llevarlo a cabo (López Carrasco, 2003).

1.2 Sistemas similares

La introducción de los ePortafolios en el ámbito educativo es un tema bastante amplio. A pesar de que el portafolio no surge con esta finalidad, se ha utilizado desde siempre, incluso algunos docentes sostienen que los portafolios no representan una novedad en la educación, porque tradicionalmente el docente siempre ha realizado una “recopilación de trabajos” de sus estudiantes.

Uno de los usos más frecuentes de los portafolios electrónicos es mediante la integración de este a una plataforma educativa virtual. Entendiéndose por plataforma educativa virtual un entorno informático donde se encuentran múltiples herramientas organizadas y optimizadas con fines docentes. Se corresponden con un Sistema de Administración del Aprendizaje (LMS). De manera general cuentan con una estructura modular, que propicia la creación de espacios de trabajo, destinados al intercambio de contenido e información. Incorporan herramientas de gestión de contenido, herramientas de comunicación y colaboración, seguimiento y evaluación, administración y asignación de permisos y otras herramientas complementarias, donde se encuentran los portafolios electrónicos.

1.2.1 Portafolios electrónicos desarrollados a nivel internacional.

Una de las plataformas más conocidas en el entorno educacional es Moodle, que constituye un paquete de software para la creación de cursos y sitios web basados en Internet. Está diseñado para dar soporte a un marco de educación social constructivista. Se distribuye gratuitamente como software libre bajo la licencia pública GNU. En pocas palabras es un sistema web que posibilita la creación de contenidos y su

distribución a través de medios electrónicos. Las características más relevantes que presenta son: Interoperabilidad, es escalable, y personalizable, económico y seguro, pedagógicamente flexible, permite la gestión de tareas y posee una gran facilidad de uso, así como de administración.

Moodle

Dentro de la plataforma Moodle se encuentran vinculadas una serie de herramientas destinadas a la creación de un portafolio electrónico. Entre las más utilizadas figuran: Exabis ePortfolio, Mahara, SPDC Portfolio y MyStuff. Cada uno de ellos presentan características mínimas que se resumen en: recolección, reflexión, organización y compartición feedback de evidencias.

Exabis acepta como evidencias: archivos de cualquier índole (“Files”), textos en línea (a los que llama “Notes”) y URLs (“External Links”). En MyStuff se observa un modelo de recolección muy versátil y sencillo pues todas las evidencias se introducen como “Notas” que admiten archivos adjuntos. Con esta estrategia, todas las evidencias van necesariamente acompañadas de un texto reflexivo. Estas cuatro herramientas permiten realizar una reflexión sobre las evidencias sea cual sea su tipo. Otra de las funcionalidades esencial en el caso de los e-portafolios es la posibilidad de organización y presentación de las evidencias. Un portafolio al que sólo tenga acceso la persona que lo elabora puede tener sentido en la medida en que el proceso de elaboración puede desempeñar un papel importante en el proceso de enseñanza-aprendizaje. Sin embargo, también es importante que las evidencias y los portafolios no queden encerrados en el ámbito privado, sino que puedan ser compartidos con otras personas. Exabis permite compartir las evidencias con otros usuarios. Con Exabis, el hecho de compartir un recurso implica necesariamente dar a los usuarios la posibilidad de añadir comentarios al mismo. Mediante SPDC pueden elegirse diferentes niveles de compartición: acceso público, usuarios registrados, usuarios individuales y, finalmente, con cualquiera de los grupos existentes en el sistema Moodle (pertenezca o no el usuario a ellos). Haya sido compartido o no, el docente tiene no sólo la posibilidad de añadir comentarios, sino incluso de realizar cambios en las reflexiones o comentarios que haya realizado el estudiante. Mahara ofrece diferentes formas de selección de las personas con las que compartir los portafolios: un portafolio puede ser público, compartido con todos los usuarios identificados del sistema, o bien con algunos usuarios seleccionados. Además, el sistema permite compartir también con grupos o con “amigos”. En el caso de MyStuff, la compartición puede realizarse tanto a nivel de portafolio como de evidencias

individuales. La compartición puede ser de dos tipos: Sólo lectura o Lectura/Comentarios y como en el caso de Mahara, se permite compartir durante cierto rango de fechas (Muñoz-Justica, 2008).

A pesar de que en Moodle lo que existen son herramientas para la creación de portafolios electrónicos, con el estudio desarrollado se pudieron extraer ideas para la implementación en cuanto a la forma en la que se almacenen las evidencias, además de en cuanto a la existencia de la opción de compartir dichas evidencias.

Sakai

El proyecto Sakai, que dio lugar al surgimiento de la plataforma Sakai, comenzó en el año 2004, con el inicio de la construcción de un sistema de gestión de cursos. El proyecto se convirtió en un entorno modular de código fuente abierto, cuya finalidad era integrar las funcionalidades del e-Learning a un marco virtual. Más de un tercio de las cien mejores universidades del mundo participan en el proyecto Sakai. Su objetivo es crear un entorno de colaboración y aprendizaje para el contexto universitario, que pueda competir con sus equivalentes comerciales Blackboard / WebCT y que mejore otras iniciativas de Código Abierto como Moodle.

Sakai cuenta con la posibilidad de que el alumno posea una página personal donde puede incluir trabajos, fotos, y otro tipo de información, acercándose a la idea esencial de un portafolio electrónico (Pascual Gimeno, 2006).

En cuanto a la plataforma SAKAI se tiene en cuenta la inclusión de trabajos realizados, que constituye la idea fundamental del portafolio electrónico.

WebCT

La plataforma WebCT (Web Course Tools), fue la elegida por la mayoría de las Universidades Españolas en un principio. Ha sido desarrollada en la Universidad de British Columbia (Canadá), se considera al profesor Murray Goldberg como su creador en 1995. Se trata de un sistema comercial de aprendizaje virtual en el que uno de los aspectos más atractivos es su flexibilidad para el diseño de cursos y que puede acoger a profesores con poca experiencia de cursos online, así como a otros con una amplia experiencia. Otras de las ventajas que presenta, es que permite realizar publicaciones de libros electrónicos. Desde poco después de su lanzamiento, WebCT se convirtió en la plataforma dominante en

el mundo académico, manteniéndose hoy día entre las más usadas tanto por funcionalidad como por número de implantaciones reales y contenidos disponibles.

El Blackboard / WebCT ePortfolio permite recopilar y organizar evidencias de diversas formas de medios de comunicación (tales como texto, imágenes, vídeo, audio) en un entorno digital. Es una herramienta muy flexible y permite que el usuario seleccione los artefactos que incluirá en su portafolio (Moreno Romero, 2008).

A raíz de este estudio se determina que el ePortfolio para la plataforma educativa ZERA debe permitirle al usuario ir almacenando sus evidencias dentro del propio ePortfolio, de esta forma podrá consultarlas cada vez que lo necesite.

1.2.2 Los e-Portafolio en Cuba.

Recientemente se ha comenzado a implementar el uso del Portafolio electrónico en la docencia universitaria en países de habla española. Con el objetivo de comenzar su empleo en la Universidad de la Habana, esta llevó a cabo la traducción del portafolio electrónico de la SPDC al español y se realizó su implementación en la plataforma interactiva Moodle de dicha institución. Se ha comenzado a utilizar el portafolio electrónico en la Facultad de Artes y Letras.

Cuando un estudiante se inscribe en un curso que dispone de este medio, automáticamente se crea su propio recurso, llamado "Mi portafolio". El profesor puede ver los portafolios de los alumnos y añadirles tareas, al acceder a "Los portafolios de mis estudiantes". Mediante "Mi Almacén", el alumno puede subir materiales a su Almacén de Archivos, que luego pueden pasar al portafolio del alumno cuando se añadan las reflexiones sobre el aprendizaje realizado. Este portafolio puede tener clave de indización, lo que permite hacer búsquedas temáticas, entre otras tareas (Fuentes Betancourt, 2007).

Hoy forma es evidente un avance a nivel nacional en relación con el uso de los portafolios electrónicos. A pesar de no ser una experiencia extendida a lo largo y ancho del país se han dado algunos pasos significativos en aras de impulsar la aplicación de métodos novedosos en la rama educacional.

1.3 Estudio de las metodologías y estándares para el desarrollo del software.

Un sistema informático está compuesto por hardware y software. En cuanto al hardware su producción se realiza sistemáticamente, y la base de conocimiento está claramente definida. Sin embargo respecto al software, su construcción y resultados, han presentado históricamente los siguientes problemas (Pressman):

- ✓ Los sistemas no responden a las expectativas de los usuarios.
- ✓ Los programas “fallan” con cierta frecuencia.
- ✓ Los costes del software son difíciles de prever y normalmente superan las estimaciones.
- ✓ La modificación del software es una tarea difícil y costosa.
- ✓ El software se suele presentar fuera del plazo establecido y con menos prestaciones de las consideradas inicialmente.
- ✓ Normalmente, es difícil cambiar de entorno hardware usando el mismo software.
- ✓ El aprovechamiento óptimo de los recursos (personas, tiempo, dinero, herramientas, etc.) no suele cumplirse.

De ahí la necesidad de establecer y usar principios de ingeniería orientados a obtener un software fiable, de manera económica y que funcione eficientemente. Un proceso de desarrollo de software tiene como propósito la producción eficaz y eficiente de un producto software que reúna los requisitos del cliente. Un producto software es intangible y por lo general muy abstracto, esto dificulta la definición del producto y sus requisitos, sobre todo cuando no se tiene precedentes en productos software similares. Esto hace que los requisitos sean difíciles de consolidar tempranamente. Así, los cambios en los requisitos son inevitables, no sólo después de entregado el producto sino también durante el proceso de desarrollo. El proceso de desarrollo de software no es único. No existe un proceso de software universal que sea efectivo para todos los contextos de proyectos de desarrollo. Debido a esta diversidad, es difícil automatizar todo un proceso de desarrollo de software.

1.3.1 Rational Unified Process (RUP).

El proceso de software propuesto por RUP tiene tres características esenciales: está dirigido por los casos de uso, centrado en la arquitectura, y es iterativo e incremental. En el caso específico del presente trabajo de diploma se hace necesario emplear esta metodología para especificar los requisitos del sistema y llegar hasta los casos de uso, los cuales juegan un papel fundamental para guiar el diseño,

implementación y prueba. En RUP los casos de uso constituyen un elemento integrador y una guía del trabajo.

A través de RUP el proceso se divide en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades.

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una línea base de la arquitectura.

Durante la fase de **inicio** se define el modelo del negocio y el alcance del proyecto. El empleo de RUP es muy útil para definir los riesgos a los que se podía enfrentar el proyecto, así como los casos de uso críticos, y poder establecer un plan para evitar o mitigar los riesgos en fases posteriores.

El propósito de la fase de **elaboración** es analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del proyecto y eliminar los mayores riesgos. Durante esta fase se emplean los artefactos generados durante la fase de inicio, lo cual aporta elementos necesarios para construir un prototipo de la arquitectura, que va evolucionando en iteraciones sucesivas hasta convertirse en el sistema final. RUP permite además demostrar que se han evitado los riesgos más graves. Para la realización del presente trabajo esta metodología permitirá definir, validar y cimentar la arquitectura, completar la visión existente referente al software, y crear un plan fiable para entrar en la fase posterior.

La finalidad principal de la fase de **construcción** es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones. Durante esta fase el programador o desarrollador podrá implementar los requisitos capturados, así como integrar y probar las características y componentes propuestos en fases anteriores, obteniendo de esta manera una versión aceptable del producto. RUP permite conseguir una calidad adecuada tan rápido como sea práctico, así como generar versiones funcionales, para pasar a la próxima fase.

Durante la fase de **transición** se pondrá el producto en manos de los usuarios finales, para lo que se requiere desarrollar nuevas versiones actualizadas del producto, completar la documentación, entrenar al

usuario en el manejo del producto, y en general tareas relacionadas con el ajuste, configuración, instalación y facilidad de uso del producto (Letelier, 2008).

A través de RUP los autores del presente trabajo conseguirán que el usuario se valga por sí mismo, así como un producto que sea capaz de cumplir con los requisitos especificados, funcionando de acuerdo a las necesidades del usuario.

Se selecciona la metodología RUP debido a las características antes expuestas ya que basa su trabajo principalmente en la documentación del software y expone un conjunto de actividades orientadas a visualizar, especificar, construir y documentar los artefactos necesarios para el desarrollo de software con la calidad que el cliente lo necesita. Además de que el cliente no forma parte del equipo de trabajo, siendo esta una de las principales ventajas por encima de otras metodologías como la Extreme Programming (XP)

1.3.2 Arquitectura.

De acuerdo con (Catalani, 2007) para el diseño de aplicaciones con sofisticadas interfaces se utiliza el patrón de diseño Modelo-Vista-Controlador. Buscando un poco de información histórica, se puede decir que este fue descrito por primera vez en 1979 por Trygve Reenskaug, trabajador de Smalltalk, en unos laboratorios de gran investigación de Xerox.

- ✓ **El Modelo** es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.
- ✓ **La Vista** genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.
- ✓ **El Controlador** es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

1.4 Estándar IMS e-Portafolio

La especificación IMS e-Portafolio define un modelo de datos estándar para portafolios electrónicos y por lo tanto aspira a garantizar su compatibilidad, tanto en el ámbito educativo como en el mundo laboral.

El estándar IMS e-Portafolio define los siguientes elementos de información a almacenar en el portafolio (ePortafolio, 2005):

- ✓ **accessForAll:** Preferencias de los usuarios en su interacción con el sistema o con contenidos.
- ✓ **activity:** Registro de las actuaciones de los usuarios en cursos en general, herramientas de evaluación, etc.
- ✓ **affiliation:** Descripciones de las organizaciones en las que el estudiante ha estado vinculado.
- ✓ **assertion:** Texto sobre el estudiante o sobre partes del portafolio.
- ✓ **competency:** Descripción de las competencias adquiridas por el usuario.
- ✓ **goal:** Competencias deseadas o expectativas del estudiante.
- ✓ **identification:** Contiene la identificación del propietario del portafolio.
- ✓ **interest:** Descripción de los hobbies o actividades recreativas del propietario del portafolio.
- ✓ **Product:** Referencias materiales producidas por el propietario del portafolio.
- ✓ **qcl:** Certificados o licencias ganadas por el propietario del portafolio.
- ✓ **reflexion:** Reflexiones o aserciones acerca de partes del portafolio.
- ✓ **relationship:** Define relaciones entre dos o más instancias del portafolio.
- ✓ **Rubric:** Guías sobre cómo el portafolio o partes del portafolio son evaluadas.

A través del estándar IMS ePortafolio se intenta conseguir una guía para gestionar la información contenida dentro de un portafolio digital o electrónico. De esta manera se establecen una serie de parámetros y funciones que no deben faltar en la confección de un portafolio.

1.5 Selección de las herramientas y lenguaje de desarrollo.

Se utilizará tecnología web para darle solución al problema por las ventajas que tiene en cuanto a compatibilidad y eficiencia, teniendo como finalidad que sea más flexible. Por otra parte se evitan instalaciones y configuraciones complicadas, el soporte al usuario se puede dar sin tener que estar físicamente, es decir a distancia.

1.5.1 Lenguajes de programación.

Del lado del cliente:

XHTML:

El lenguaje XHTML (Lenguaje de Marcado de Hipertexto Extensible) es una versión más estricta y limpia de HTML se crea con el objetivo de remplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas de XML. Permite una correcta interpretación de la información independientemente del dispositivo desde el que se accede a ella. El XHTML es utilizado en diversas aplicaciones digitales como son motores de búsquedas, sitios de información, bancos electrónicos, etc. (XHTML., 2008)

JavaScript

JavaScript es un lenguaje interpretado basado en prototipos. Es usado principalmente en la creación de páginas web. Permite un mejor desarrollo de interfaz de usuario, es un lenguaje basado en acciones. La mayor parte de la programación en este lenguaje está centrada en describir objetos. Además Javascript es de código libre, por lo que cualquier persona puede utilizarlo sin comprar una licencia.

Es necesario resaltar que hay dos tipos de JavaScript: por un lado está el que se ejecuta en el cliente, este es el Javascript propiamente dicho, aunque técnicamente se denomina Navigator JavaScript. Pero también existe un JavaScript que se ejecuta en el servidor, es más reciente y se denomina LiveWire Javascript. (ECMA, 2009)

Ajax

De acuerdo con (Eguíluz Pérez, 2008) Ajax es el acrónimo de Asynchronous JavaScript + XML. Ajax no es una tecnología, es realmente muchas tecnologías.

Cuando se combinan estas tecnologías en el modelo Ajax, las aplicaciones funcionan mucho más rápido, ya que las interfaces de usuario se pueden actualizar por partes sin tener que actualizar toda la página completa.

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

Las aplicaciones de formularios, que usan funciones de Ajax ofrecen las siguientes características:

- ✓ Elementos de interfaz de usuario interactivos que resultan familiares, por ejemplo indicadores de progreso, información sobre herramientas y ventanas emergentes.
- ✓ Mayor eficacia para la aplicación de formularios, porque una parte importante del procesamiento de una página web se puede realizar en el explorador.
- ✓ Actualizaciones parciales de la página, que actualizan sólo las partes de la página web que han cambiado.
- ✓ Clases de proxy generadas automáticamente que simplifican las llamadas a los métodos del servicio web desde el script de cliente.
- ✓ La capacidad de personalizar los controles de servidor para incluir capacidades de cliente.
- ✓ Soporte para los exploradores Microsoft Internet Explorer, Mozilla Firefox y Apple Safari.

Cascading Style Sheets (CSS)

Las hojas de estilo o CSS es un lenguaje utilizado para definir la presentación de un documento en formato HTML o XML, permite la separación de los contenidos de los documentos escritos en HTML, XML, XHTML, SVG, o XUL de la presentación del documento con las hojas de estilo. El objetivo fundamental del estilo en cascada es separar la apariencia con la estructura del documento, facilitando al desarrollador realizar modificaciones en la apariencia del documento sin tener que ver la estructura del mismo (Eguíluz Pérez, 2008).

Del lado del servidor:

Hypertext Preprocessor (PHP)

Es un lenguaje interpretado, incrustado en páginas HTML y ejecutado en el servidor. Da soporte a una gran cantidad de base de datos como son (InterBase, mSQL, MySQL, Oracle, Informix, PostgreSQL). También ofrece la integración con varias bibliotecas externas, que le permiten al desarrollador más seguridad y confiabilidad. Es legible y fácil de aprender, es de código abierto o libre, por lo que se puede hacer uso de él sin tener que comprar una licencia. (Alvarez, 2001).

1.5.2 Lenguaje Unificado de Modelado

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. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema, para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo y se puede aplicar para dar soporte a una metodología de desarrollo de software (Systems, 2008).

Para comprender qué es el UML, basta con analizar cada una de las palabras que lo componen, por separado.

Lenguaje: el UML es, precisamente, un lenguaje. Lo que implica que éste cuenta con una sintaxis y una semántica. Por lo tanto, al modelar un concepto en UML, existen reglas sobre cómo deben agruparse los elementos del lenguaje y el significado de esta agrupación.

Modelado: el UML es visual. Mediante su sintaxis se modelan distintos aspectos del mundo real, que permiten una mejor interpretación y entendimiento de éste.

Unificado: unifica varias técnicas de modelado en una única.

Ya que el UML proviene de técnicas orientadas a objetos, se crea con la fuerte intención de que este permita un correcto modelado orientado a objetos.

UML es fácil de aprender y usar, ya sea respecto a las técnicas más avanzadas, es decir, estereotipos y propiedades, así como algunos cambios en la anotación y semánticas.

A través de UML, se definirán los distintos diagramas que en cada una de las fases de desarrollo del portafolio permitirán establecer una visión cada vez más clara de las necesidades del producto. De tal manera podrá ser generado en un primer momento el Modelo de dominio, en aras de representar de forma visual las relaciones entre las diferentes clases inmersas en el ámbito de la creación del portafolio electrónico. A continuación se realizará el diagrama de casos de uso, diagramas de clases del análisis, diagramas de clases del diseño, diagrama de despliegue, diagrama de componentes, y otros diagramas útiles en la confección del ePortafolio para la plataforma educativa ZERA.

1.5.3 Framework.

El término framework se utiliza constantemente en el desarrollo de software. Un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Un framework representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

Symfony

Symfony es un framework que está desarrollado en PHP5, se puede utilizar en plataformas *nix (Unix, Linux) y Windows. Incorpora el patrón MVC, soporta AJAX, plantillas y un gran número de bases de datos. Facilita el desarrollo de las aplicaciones web (Eguíluz Pérez, 2008).

Características

- ✓ Fácil de instalar y configurar en sistemas Windows, Mac y Linux.
- ✓ Funciona con todas las bases de datos comunes (MySQL, PostgreSQL, SQLite, Oracle, MS SQL Server).
- ✓ Compatible solamente con PHP5 desde hace años, para asegurar el mayor rendimiento y acceso a las características más avanzadas de PHP.
- ✓ Basado en la premisa de *"convenir en vez de configurar"*, en la que el desarrollador solo debe configurar aquello que no es convencional.

- ✓ Preparado para aplicaciones empresariales, ya que se puede adaptar con facilidad a las políticas y arquitecturas propias de cada empresa u organización.
- ✓ Flexible hasta cualquier límite y extensible mediante un completo mecanismo de plugins.
- ✓ Publicado bajo licencia MIT de software libre y apoyado por una empresa comprometida con su desarrollo.
- ✓ Traducido a más de 40 idiomas y fácilmente traducible a cualquier otro idioma. (Eguíluz Pérez, 2008)

jQuery

jQuery no es más que un framework de Javascript que simplifica la manera de interactuar con HTML, manipular el DOM y sobre todo ofrece el uso de AJAX. Es compatible con todos los navegadores modernos. (G.V., 2010)

La librería jQuery en resumen aporta las siguientes ventajas:

- ✓ Ahorro de muchas líneas de código.
- ✓ Hace transparente el soporte de la aplicación para los navegadores principales.
- ✓ Provee un mecanismo para la captura de eventos.
- ✓ Provee un conjunto de funciones para animar el contenido de la página en forma muy sencilla.
- ✓ Integra funcionalidades para trabajar con AJAX.

1.5.4 Sistema gestor de base de datos.

PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es uno de los sistemas de gestión de bases de datos de código abierto más potente del mercado. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Tiene interfaces de programación nativo de C/C++, Java, .NET, Perl, Python, Ruby, Tcl, ODBC, entre otros, y la documentación de carácter excepcional. Funciona en todos los principales sistemas operativos, incluyendo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. (Postgresql, 2008)

1.5.5 Entorno de desarrollo integrado.

NetBeans

NetBeans es un proyecto de código abierto. Es un entorno de desarrollo, una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas.

Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. Es un producto libre y gratuito sin restricciones de uso (Netbeans, 2007).

Entre las características que proporciona se encuentran:

- ✓ Desarrollo de aplicaciones multiplataforma sobre: MacOS, Windows, Linux.
- ✓ Add-ons para desarrollo Móvil, desarrollo Web gráfico, integración con SOA, optimización de aplicaciones y desarrollo con C y C++.
- ✓ Cliente CVS integrado
- ✓ Crecimiento de plataforma por medio de plugins. Entre los plugins que existen se tienen los siguientes:
 - Herramientas java que sirven para la mejora de desarrollo de aplicaciones.
 - Herramientas de modelado UML
 - Herramientas XML, etc.
- ✓ Struts, JSF, EJB, WebServices, etc.

1.5.6 Herramienta CASE para el modelado UML.

Existen diferentes herramientas creadas para el desarrollo de la ingeniería de software. Su propósito no es más que desarrollar programas, empleando técnicas de diseño y metodologías bien definidas.

¿Qué se entiende por herramienta CASE? CASE o por sus siglas en inglés Computer Aided Software Engineering es el conjunto de ayudas para el desarrollo de un software desde la planificación, el diseño, hasta la generación del código fuente y la documentación (Giraldo, 2005)

Objetivos de las Herramientas CASE

- ✓ Mejorar la productividad en el desarrollo y mantenimiento del software.
- ✓ Aumentar la calidad del software.

- ✓ Reducir el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- ✓ Mejorar la planificación de un proyecto.
- ✓ Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
- ✓ Automatizar el desarrollo del software, la documentación, la generación de código, las pruebas de errores y la gestión del proyecto.
- ✓ Ayuda a la reutilización del software, portabilidad y estandarización de la documentación.
- ✓ Gestión global en todas las fases de desarrollo de software con una misma herramienta.
- ✓ Facilitar el uso de las distintas metodologías propias de la ingeniería del software.

Las herramientas CASE—UML pueden clasificarse en 3 tipos fundamentales:

Upper Case: planificación y requerimientos.

Middle Case: análisis y diseño.

Lower Case: generación de código, pruebas e implementación.

Entre las herramientas CASE orientadas a UML se destacan:

- ✓ ArgoUML.
- ✓ Poseidon.
- ✓ MagicDraw UML.
- ✓ Visual Paradigm.
- ✓ Borland Together.

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Esta herramienta tiene unas características gráficas muy cómodas que facilitan la realización de los diagramas de modelado que sigue el estándar de UML. Entre otras características importantes figuran:

Integración con diversas IDE's como son:

- ✓ NetBeans(de Sun)
- ✓ JDeveloper(de Oracle)

- ✓ Eclipse (de IBM)
- ✓ JBuilder (de Borland)

Ingeniería Inversa para:

- ✓ JAVA
- ✓ .NET
- ✓ XML
- ✓ Hibernate

Exportación de imágenes jpg, png y svg(w3g estándar) (Valdez Altamirano, 2009).

Conclusiones parciales.

Al concluir el presente capítulo se abordaron los aspectos fundamentales relacionados con los ePortafolio, exponiéndose de esta manera, concepto, funcionalidades, ventajas y desventajas, así como los tipos y la estructura de los mismos, todo lo cual propició una visión más detallada acerca de las características a implementar en el ePortafolio destinado a la plataforma educativa ZERA.

Se realizó un estudio del estado del arte, comprobando la existencia de ePortafolios a nivel internacional, donde se destacan los incorporados a la plataforma Moodle, Sakai y WebCT. Este estudio aportó ideas para la implementación del módulo ePortafolio, como la inclusión de un archivo de evidencias y la posibilidad de comentar evidencias evaluadas con anterioridad.

Para llevar a cabo el proceso de desarrollo de software se selecciona RUP como metodología a emplear ya que facilitará la documentación, el seguimiento y control de las actividades a desarrollar, logrando alcanzar un producto de excelente calidad. Se utilizará Visual Paradigm como herramienta de modelado, ya que emplea UML como lenguaje de modelado, es libre y multiplataforma. Dentro de los lenguajes de programación se escogen XHTML, Javascript, PHP, CCS y Ajax. Todo lo expuesto anteriormente junto con el estudio de la arquitectura Modelo-Vista-Controlador a utilizar, aportan los elementos necesarios para el desarrollo del portafolio electrónico para la plataforma educativa ZERA.

Capítulo 2

Características del sistema.

Capítulo 2 Características del sistema.

Introducción

Las características del sistema a implementar son el punto de partida para comenzar el proceso de desarrollo. Estas van a permitir una visión objetiva acerca del producto que se quiere obtener. Los conceptos más importantes del dominio del problema permiten estructurar el modelo del dominio, que da paso al análisis de las funcionalidades que deben estar presentes en el ePortafolio para la plataforma ZERA.

En el capítulo que se presenta a continuación se realiza una especificación de los requisitos tanto funcionales como no funcionales, así como una descripción de los actores, casos de uso, y las relaciones que se establecen entre ellos.

2.1 Modelo de Dominio

Al realizar un análisis detallado de los conceptos que intervienen en el dominio del problema planteado, se llega como resultado final, al modelo de dominio, donde se representa de forma visual el entorno real en el que se desarrolla.

2.1.1 Análisis de los conceptos del Dominio

- ✓ **Portafolio:** Se define como el espacio personal de cada estudiante dentro de la plataforma, donde puede consultar sus evaluaciones, acceder a las evidencias que posee, compartir evidencias con otros estudiantes, así como comentar aquellas que hayan sido compartidas para él.

- ✓ **Profesor:** Responsable del proceso de enseñanza-aprendizaje. Es el encargado de la educación de los estudiantes.
- ✓ **Estudiante:** Es el actor fundamental que interactúa con el portafolio. Posee un único portafolio donde guarda un archivo de sus evidencias.
- ✓ **Evidencia:** Se define como el archivo que da respuesta a una tarea orientada por el profesor, o que fue subida a la plataforma como parte del estudio independiente del estudiante.
- ✓ **Evidencia compartida:** Esta evidencia es aquella que algún estudiante desde su portafolio definió que los demás estudiantes pudieran consultarla. Las evidencias que se pueden compartir son aquellas que el estudiante realizó como parte de su estudio individual.
- ✓ **Evidencia comentada:** Es aquella evidencia que ha recibido algún comentario por parte de las personas que la consultaron.
- ✓ **Evaluación:** Proceso sistemático a través del cual, se comprueba el cumplimiento de los objetivos propuestos en una actividad específica y se corrigen los errores para llegar a un resultado final satisfactorio.
- ✓ **Módulo:** Conjunto de funcionalidades con características similares que responden a un mismo objetivo dentro de la creación del producto final.
- ✓ **Notificaciones:** Constituyen avisos, alertas que se emiten cuando se genera un cambio dentro los parámetros que involucran al usuario que las recibe. Se emiten para informar acerca de dichas transformaciones.
- ✓ **Archivo de evidencias:** Constituye un almacén de todas las evidencias que posee el usuario.
- ✓ **Comentario:** Criterio emitido como opinión respecto a algún documento o archivo que constituya una evidencia compartida.
- ✓ **Asignación:** Proceso, mediante el cual se seleccionan los ejercicios y los estudiantes a los que estos estarán asociados para la evaluación.

- ✓ **Hiperentorno de aprendizaje:** Sistema informático basado en tecnología hipertexto, en el que se mezclan diversas tipologías de software educativo con el propósito de atender las necesidades didácticas.
- ✓ **Usuario de hiperentorno:** Es el usuario que accede al hiperentorno de aprendizaje, puede comportarse como un estudiante, docente, tutor, administrador local o director.
- ✓ **Tareas:** Módulo que integra varios componentes como son: recorridos dirigidos, evidencias y/o ejercicios agrupados en una softarea.
- ✓ **Docente:** Módulo que reúne todas las acciones específicas que se le asignan al docente dentro del hiperentorno.

2.1.2 Diagrama del Modelo de Dominio

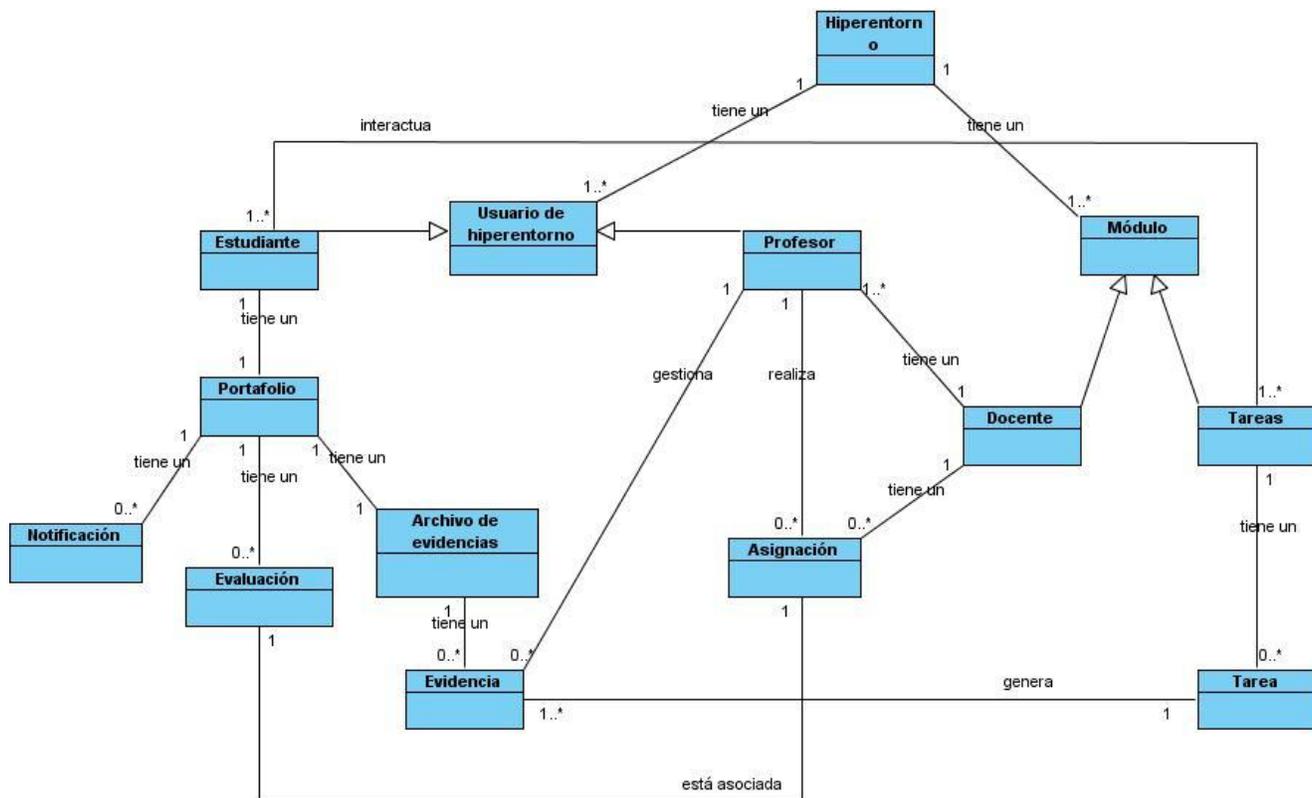


Imagen 1: Modelo de dominio

2.2 Descripción del Sistema Propuesto

Las evidencias constituyen una muestra tangible del desempeño de un estudiante durante todo un curso académico. Estas pueden tener diferentes formatos, en dependencia de los parámetros definidos por el profesor. Las evidencias son gestionadas directamente por los docentes, quienes se encargan de asignar a estas los estudiantes que deben realizarlas. En tal sentido el profesor puede incluir, modificar, ver o eliminar una evidencia. El estudiante sube un archivo a la plataforma como respuesta a una tarea orientada por el profesor, y este se almacena directamente en el portafolio del estudiante. La propuesta de portafolio llevada a cabo se divide en 4 secciones. La sección de Notificaciones incluye todas las novedades que posee el estudiante en materia de evaluaciones y evidencias desde su última entrada a la plataforma. En las notificaciones se muestran las últimas evaluaciones obtenidas y/o aviso de la existencia de evidencias que hayan sido compartidas para él. La sección de Mis calificaciones facilita la consulta de las calificaciones que el estudiante va obteniendo de forma paulatina. Dentro de la sección Evidencias Compartidas el estudiante accede a las evidencias que otros estudiantes hayan compartido para que él pueda consultarlas, y puede emitir algún criterio, es decir comentarla, puede además modificar algún comentario hecho con anterioridad, o eliminarlo. Por último se encuentra la sección Archivo de evidencias, donde se almacenan todas las evidencias que han sido subidas a la plataforma. De esta forma el estudiante puede compartir una evidencia, siempre y cuando esta haya sido evaluada.

2.3 Requerimientos del software

Para llevar a cabo el desarrollo de un software es necesario contar con una descripción detallada de los requerimientos, para conocer, qué debe hacer el sistema, y la forma en que debe hacerlo. Los requerimientos del software constituyen características o capacidades que el sistema debe cumplir. Estos pueden clasificarse en:

Requerimientos funcionales: Definen las funcionalidades que el sistema debe realizar.

Requerimientos no funcionales: Son propiedades, o características que el sistema debe tener. Define los elementos referidos a la usabilidad, el rendimiento, interfaces, fiabilidad, mantenimiento, seguridad, entre otros aspectos.

2.3.1 Requerimientos Funcionales

RF 1: Gestionar evidencia.

RF 1.1 Incluir evidencia. Brinda la posibilidad de introducir los siguientes datos para crear una evidencia: Nombre, descripción, fecha de expiración, formato y complejidad, además permite seleccionar los estudiantes que estarán asociados a dicha evidencia.

RF 1.2 Modificar evidencia. Brinda la posibilidad de modificar todos los datos gestionados al introducir una nueva evidencia.

RF 1.3 Ver evidencia. Muestra los datos asociados a las evidencias creadas.

RF 1.4 Eliminar evidencia. En un primer momento muestra un mensaje de confirmación, al acceder a aceptar la operación se elimina la evidencia.

RF 2: Mostrar notificaciones. Muestra las notificaciones del usuario referentes a evaluaciones y evidencias que hayan sido compartidas para él. Permite marcarlas todas, desmarcarlas y eliminarlas.

RF 2.1 Eliminar notificaciones. Muestra un mensaje de confirmación, y oculta la notificación seleccionada.

RF 3: Consultar evaluaciones.

RF 3.1 Buscar evaluaciones dados diferentes parámetros de búsqueda. Permite filtrar por 2 criterios de búsqueda, rango de fecha y tipo de actividad.

RF 3.2 Mostrar un listado de evaluaciones. Muestra todas las coincidencias encontradas al realizar la búsqueda de acuerdo a los datos introducidos.

RF 4: Compartir evidencia.

RF 4.1 Asignar estudiantes. Permite seleccionar los estudiantes que tendrán acceso a la evidencia que se está compartiendo.

RF 4.2 Descompartir evidencia. Permite cambiar el estado de la evidencia, por lo que los estudiantes a los cuales se les había compartido ya no la podrán ver.

RF 5: Comentar evidencia.

Consiste en emitir un criterio acerca de una evidencia compartida hacia el usuario que realiza el comentario.

RF 5.1 Eliminar comentario. Permite eliminar uno de los comentarios que posea la evidencia. Muestra un mensaje de confirmación antes de realizar la acción.

RF 5.2 Modificar comentario. Permite modificar un comentario realizado con anterioridad.

RF 6: Consultar evidencia compartida.

RF 6.1 Mostrar un listado de evidencias compartidas. Muestra todas las evidencias compartidas para el usuario autenticado. Al realizar el filtrado muestra solamente las que coinciden con el criterio de búsqueda.

RF 6.2 Buscar evidencias teniendo en cuenta parámetros de búsqueda. Se brinda la opción de filtrar las evidencias de acuerdo al nombre, y al autor de la misma.

RF 7: Consultar archivo de evidencias.

RF 7.1 Mostrar un listado de evidencias. Muestra todas las evidencias que posee el usuario autenticado. Desde esta sección puede compartir las evidencias. Por cada evidencia muestra el nombre y el autor.

2.3.2 Requerimientos No Funcionales

✓ **Seguridad**

La plataforma deberá proteger el acceso a la información para evitar que esta sea alterada por agentes no autorizados, restringiendo el acceso a la misma, a través del otorgamiento de roles y privilegios, y la asignación de perfiles. En el caso específico del portafolio, sólo se encontrará disponible para usuarios con el rol de estudiante, el cual deberá registrarse previamente para acceder a este, a través del uso de

una contraseña. Con respecto a la seguridad del código: utilizar ofusadores de código fuente. Asegurar la plataforma de los ataques de hackers, tales como Cross Site Scripting (XSS), SQL Injection, Cross-Site Request Forgeries (CSRF), mediante la escritura de código seguro.

✓ **Interfaz, usabilidad y rendimiento.**

Interfaz práctica y amigable para los usuarios finales. Capaz de poseer un funcionamiento intuitivo y que a la vez requiera poca información para lograr su correcto funcionamiento. Claridad y organización de la información, ejecución rápida y precisa de las acciones, tratando de minimizar los pasos a dar en cada proceso.

✓ **Interfaces de Hardware**

Las estaciones de trabajo que brindarán el servicio cliente del sistema deberán presentar potencias mayores o iguales a la de una PC Pentium 4, presentando al menos 512 Mb de memoria RAM, y 200 MB de espacio libre en disco.

2.4 Modelo de Casos de Uso del Sistema

2.4.1 Descripción de los actores del sistema

Actor	Descripción
Estudiante	Realiza todas las acciones sobre el portafolio. Puede consultar sus notificaciones, ver las evaluaciones obtenidas, compartir evidencias, así como comentar aquellas que hayan sido compartidas para él.
Docente	Es el responsable de la gestión de las evidencias.

Tabla 1.1: Descripción de los actores del sistema

2.4.2 Diagrama de casos de uso del sistema

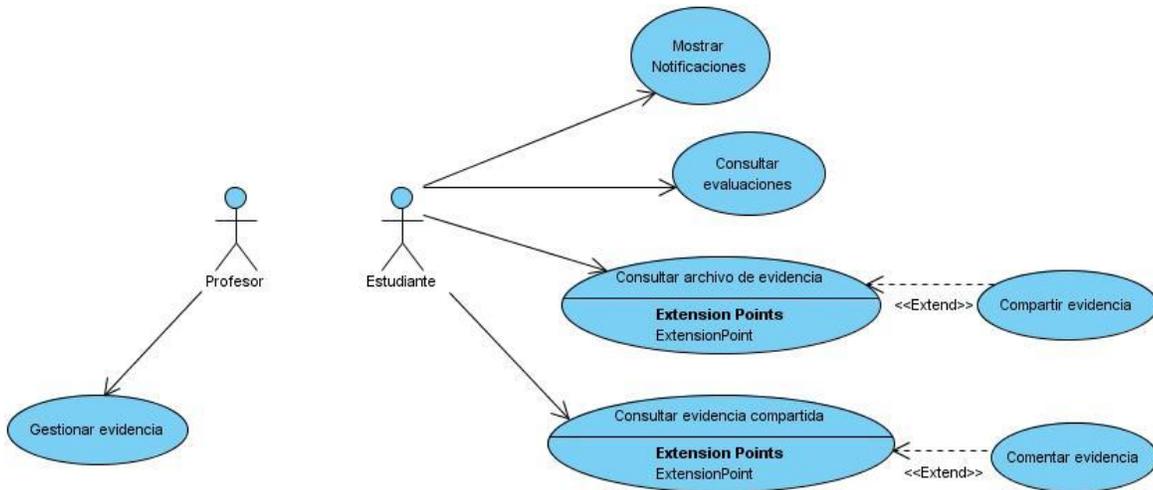


Imagen 2: Diagrama de casos de uso del sistema

La imagen muestra la relación que existe entre los actores y los casos de uso definidos en el problema planteado. Se observa la existencia de 2 casos de uso extendidos.

2.4.3 Descripción de los casos de uso del sistema

Caso de uso: Consultar evaluaciones.

Resumen

El caso de uso se inicia cuando el actor (estudiante) selecciona la opción de consultar sus evaluaciones, selecciona el tipo de evaluación que desea consultar, a la cual puede llegar a través de un sistema de filtros donde contará con los criterios de tipo de evaluación y rango de fechas. El sistema muestra adicionalmente el promedio del grupo del estudiante autenticado y el promedio del estudiante.

Acciones del actor	Respuesta del sistema
1 El caso de uso se inicia cuando el actor accede a la opción de Consultar Mis calificaciones.	
	2 Muestra un listado con 5 evaluaciones del estudiante, teniendo en cuenta el

	<p>criterio: la primera será la última evaluada. De cada una se muestran los siguientes datos:</p> <ul style="list-style-type: none"> ✓ Nombre del elemento evaluado. ✓ Tipo de actividad. ✓ Fecha de evaluación. ✓ Calificación obtenida. <p>Muestra además el promedio general del grupo al que pertenece el usuario y el promedio individual del usuario. Brinda la posibilidad de añadir los siguientes filtros:</p> <ul style="list-style-type: none"> • Tipo de actividad • Rango de fechas. <p>Permite</p> <ul style="list-style-type: none"> • Navegar con el paginado.
3	Selecciona el filtro deseado.
4	Selecciona los datos asociados al filtro.
5	Selecciona la opción Aplicar filtro
	6 Valida los datos y muestra un listado de coincidencias.
7	Consulta el resumen de los datos de las evaluaciones mostradas.
	8 Permite: <ul style="list-style-type: none"> • Realizar una nueva búsqueda
9	Selecciona una de las opciones que le permite salir de la vista actual.
10	El caso de uso termina.

Flujo alternativo

2 .a El actor no tiene evaluaciones.

Acciones del actor

Respuesta del sistema

4. a.1 Muestra un mensaje de información.

4. a.2 Regresa al paso 2 del Flujo básico.

3 .a El actor selecciona la opción de navegar por el paginado

Acciones del actor

Respuesta del sistema

3. a.1 Regresa al paso 2 del Flujo básico.

4 .a El actor selecciona el filtro de fecha y la fecha de inicio es mayor a la fecha final.

Acciones del actor

Respuesta del sistema

4. a.1 Muestra un mensaje de error

4. a.2 Regresa al paso 4 del Flujo básico.

4 .b El actor selecciona el filtro de tipo de actividad y no selecciona ninguno de los tipos mostrados.

Acciones del actor

Respuesta del sistema

4. b.1 Muestra un mensaje de error.

4. b.2 Regresa al paso 4 del Flujo básico.

6 .a El sistema no encuentra ninguna coincidencia.

Acciones del actor

Respuesta del sistema

6. a.1 Muestra un mensaje de información

6. a.2 Regresa al paso 2 del Flujo básico.

9 .a El actor selecciona la opción de realizar una nueva búsqueda.

Acciones del actor

Respuesta del sistema

6. a.1 Regresa al paso 2 del Flujo Básico.

Para ver las descripciones completas de los casos de uso, consulte el Anexo 1.

2.4.4 Patrones de Casos de uso

Los patrones de los casos de uso son empleados generalmente como plantillas que describen cómo deberían ser estructurados y organizados los casos de uso. Constituyen mejores prácticas para modelar casos de uso. Al aplicarlos aumenta la productividad, se pueden reutilizar soluciones ya propuestas, y se evita que aparezcan errores repetidos. Uno de los patrones más utilizados en la actualidad son los CRUD², permiten gestionar una entidad al agrupar las funciones de incluir, ver, modificar o eliminar sus datos, en una sola acción. En el presente trabajo se emplea este patrón en el caso de uso Gestionar evidencia.

Otro de los patrones utilizados en los casos de uso para la implementación del portafolio electrónico es el patrón Extensión concreta o Inclusión, fue empleado en los casos de uso Compartir evidencia y Comentar evidencia, que constituyen extensiones de los casos de uso Consultar archivo de evidencias, y Consultar evidencias compartidas respectivamente.

A través de estas técnicas se especifica y modela mejor el sistema. Permiten que los casos de uso sean más claros, y que al aparecer algún tipo de problema ya identificado, este pueda ser solucionado en el menor tiempo posible. En fin la idea es reutilizar todo lo que va surgiendo y establecerlo como un estándar.

Conclusiones parciales

² Los patrones CRUD deben su nombre a las siglas en inglés de Create, Read, Update, Delete.

En el capítulo que concluye se realizó un modelamiento del dominio del problema propuesto, a raíz del cual fueron definidos los 7 casos de uso que darán paso a la implementación del módulo ePortafolio. Se determinaron los requerimientos no funcionales de hardware, seguridad así como interfaz de usuario.

A raíz de los casos de uso se creó el diagrama de casos de uso del sistema y una vez creado se describen estos para lograr un mayor entendimiento de las funcionalidades que debe cumplir el ePortafolio. De esta forma es posible entrar en una fase posterior donde se realizará el análisis y diseño del sistema propuesto.

Capítulo 3 Análisis y Diseño

Capítulo 3: Análisis y Diseño.

Introducción

A raíz del flujo de trabajo análisis y diseño propuesto por la metodología empleada en el desarrollo del ePortafolio para la plataforma educativa ZERA se generan diferentes artefactos que dan al traste con la posterior etapa de implementación.

A continuación se expone el resultado de la realización del modelo de análisis, presentando de esta forma los diagramas de clases del análisis. Se realizan además los diagramas vinculados al diseño de la aplicación y se define todo lo referente al modelo de datos, diseño de la base de datos, y la explicación de las tablas que contiene dicho modelo.

3.1 Modelo de análisis

El modelo de análisis contiene clases y objetos, proporcionando una comprensión detallada de los requisitos. Durante el análisis se muestran los siguientes estereotipos de clases:

Clases Interfaz: Se encargan de manejar la interacción entre el sistema y sus actores.

Clases Controladora: Coordina el trabajo para la realización de los casos de uso. Poseen un control sobre las acciones a realizar.

Clase Entidad: Representa la información, que generalmente es de larga duración, y a menudo persistente.

3.2 Diagrama de clases del análisis

A continuación se muestran las clases del análisis por cada caso de uso definido en el desarrollo del portafolio electrónico para la plataforma educativa ZERA.



Imagen 3: Diagrama de clases del análisis CU_Consultar evaluaciones.

En la imagen se muestra la relación que se establece entre el actor estudiante al interactuar con la clase interfaz Consultar Evaluaciones, esta a su vez establece comunicación con la clase controladora que lleva el mismo nombre y por último la clase controladora solicita las evaluaciones a la clase entidad.

Para ver los restantes diagramas de clases del análisis, consultar el Anexo 2

Diagramas de colaboración del análisis. .

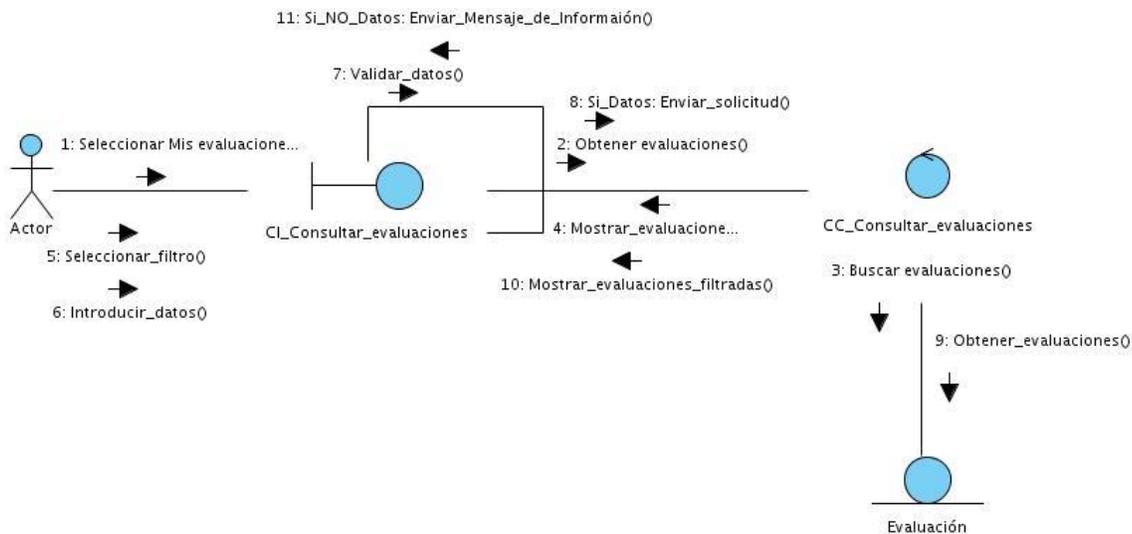


Imagen 4: Diagrama de Colaboración _CU Consultar evaluaciones.

La imagen muestra el proceso a través del cual un estudiante consulta sus evaluaciones. Como se puede observar el estudiante accede a la opción Mis evaluaciones, inmediatamente la clase interfaz solicita a la clase controladora que obtenga las evaluaciones de dicho estudiante, la controladora busca dentro de la entidad y por último la interfaz muestra la evaluación en cuestión. El próximo paso es la selección de un

filtro, donde el usuario lo selecciona e introduce los datos asociados al mismo. Los datos son validados, y si están correctos se envía la solicitud y se obtienen las evaluaciones. Si los datos son incorrectos, se muestra un mensaje de error.

Para ver los restantes Diagramas de Colaboración del análisis, consultar Anexo 3.

3.3 Modelo de diseño

El diseño se encarga de formular los modelos que se centran en los requisitos no funcionales y en el dominio de la solución, preparando al sistema para la posterior implementación y prueba. Constituye el eje principal durante la fase de elaboración, permitiendo crear una arquitectura estable y sólida y propiciando un entendimiento mayor acerca de los requisitos.

3.3.1 Diagrama de clases del diseño

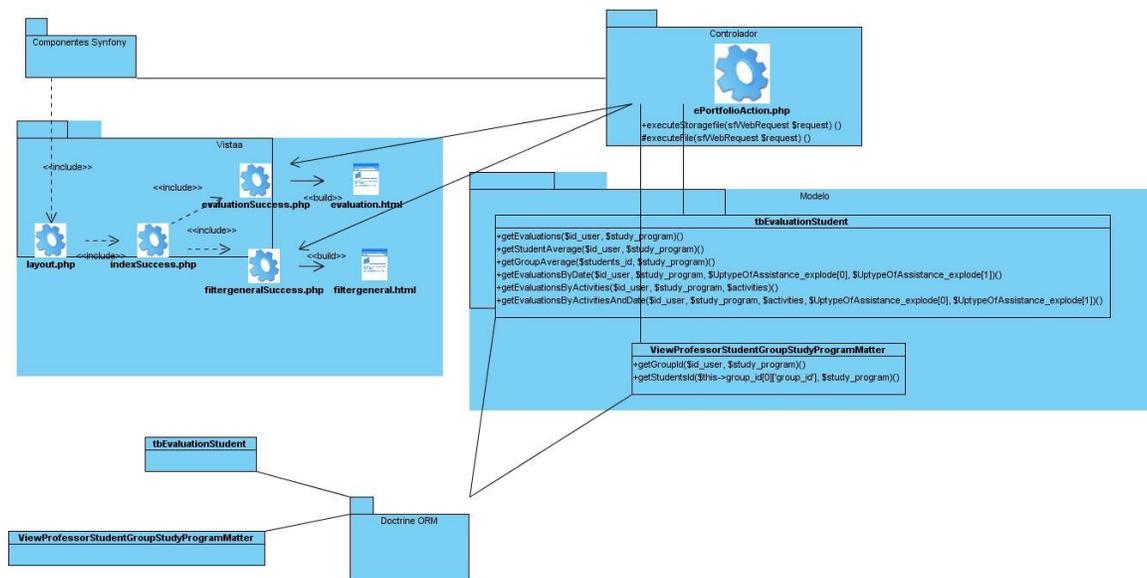


Imagen 5: Diagrama de clases del diseño CU_Consultar evaluaciones.

En la imagen mostrada se observan las capas pertenecientes a la arquitectura modelo-vista-controlador. En el diseño de las vistas se muestra cómo llegará el usuario hasta la página (en este caso) de las evaluaciones. Estas clases se conectan con la página controladora action.php. De igual manera se observa la relación de la clase anteriormente señalada con las tablas de la base de datos que intervienen en el desarrollo de este caso de uso.

Para ver los restantes diagramas de clases del diseño, consultar Anexo 4

3.4 Diagramas de secuencia del diseño.

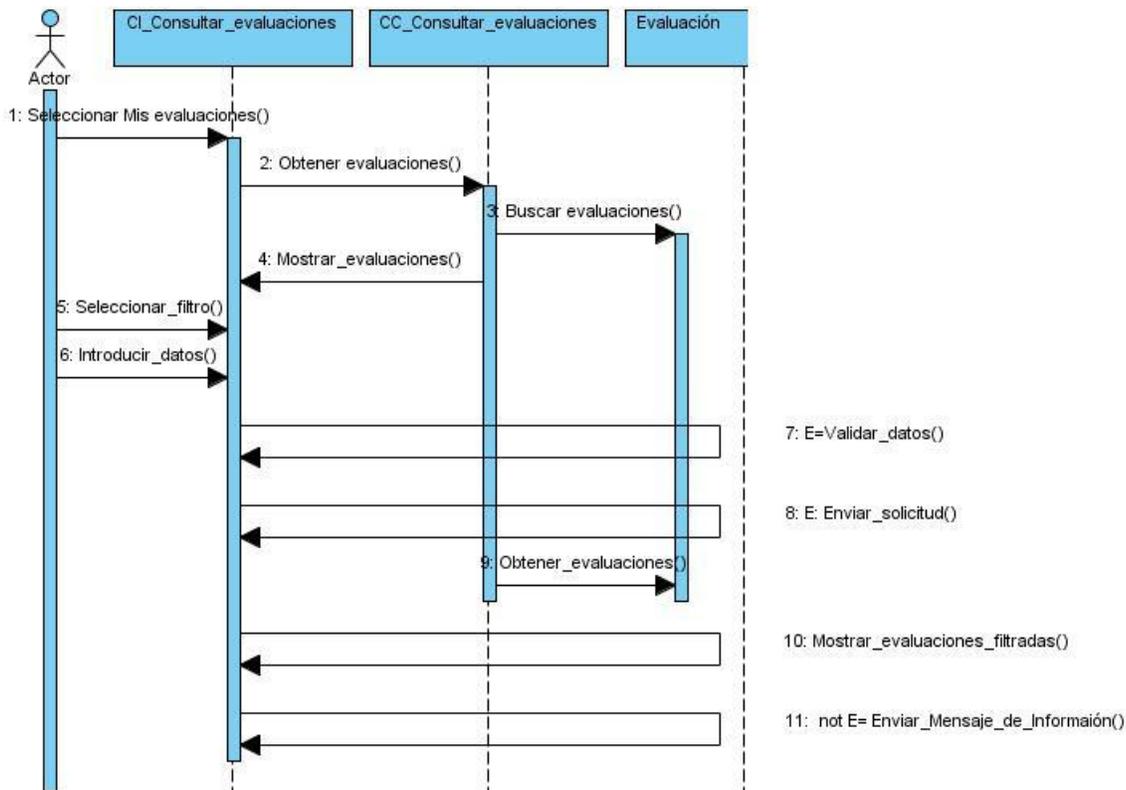


Imagen 8: Diagrama de secuencia del diseño CU_Consultar evaluaciones.

El presente diagrama de secuencia fue generado a partir del diagrama de colaboración mostrado. Como se puede observar modela los aspectos dinámicos de este caso de uso, es decir los pasos que se van llevando a cabo para cumplir con la funcionalidad. No es más que la interacción entre las diferentes clases.

Para ver los restantes diagramas de secuencia, consultar Anexo 5.

3.5 Patrón de diseño Modelo Vista Controlador.

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC, que está formado por tres niveles:

- ❖ El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- ❖ La vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- ❖ El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones.

El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación (De Dios, 2009).

3.6 Diseño de la base de datos

Al realizar el diseño de la base de datos se logra el modelo de datos donde se representan de forma lógica y física los datos persistentes.

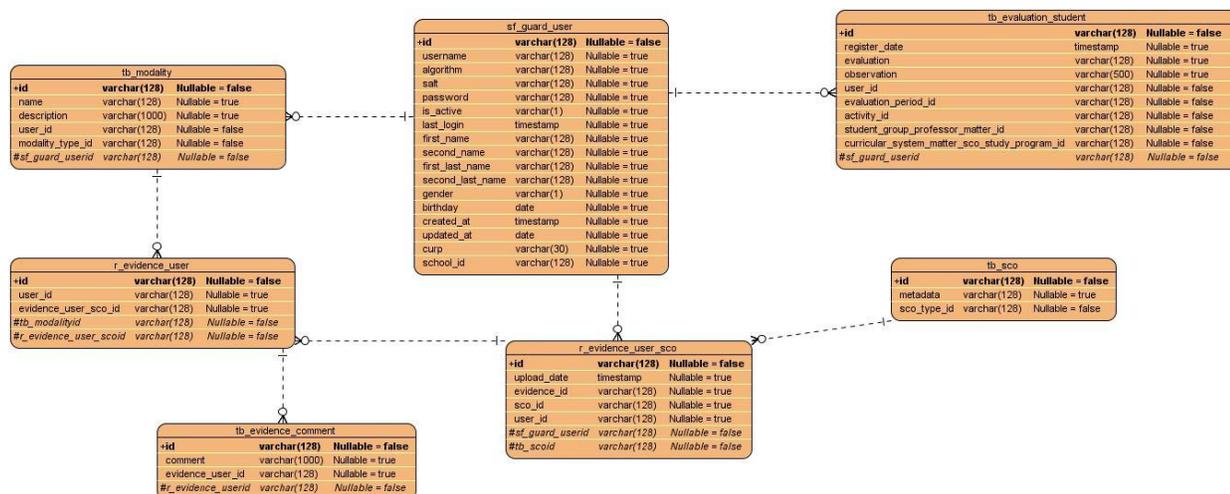


Imagen 9: Diagrama Entidad-Relación.

En la imagen 9 se relacionan las tablas de la base de datos general que tienen una vinculación estrecha con la creación del ePortafolio es decir las tablas que brindan los datos utilizados para desarrollar las funcionalidades. Por cada tabla se muestran los atributos, su tipo de dato, así como la especificación acerca de si su valor puede ser nulo o vacío. Además pueden verse las relaciones que se establecen entre las tablas (one to one, one to many, many to many).

3.6.1 Descripción de las tablas.

Tabla 1: tb_evaluation_student

tb_evaluation_student		
Descripción: Esta tabla almacena los datos de la evaluación del estudiante		
Atributo	Tipo	Descripción
register_date	TIMESTAMP/DATE	Este atributo se refiere a la fecha de registro de la evaluación.
evaluation	VARCHAR(128)	Este atributo constituye la evaluación que recibe el estudiante después de haber realizado una tarea.
observation	VARCHAR(500)	Constituye una breve descripción acerca de la evaluación realizada.
user_id	VARCHAR(128)	Este atributo constituye

		un identificador que se asigna a cada usuario registrado.
evaluation_period_id	VARCHAR(128)	Este atributo constituye el id que tiene asignado el período evaluativo relacionado con el usuario autenticado.
activity_id	VARCHAR(128)	Este atributo no es más que el identificador de la actividad a la cual se le asigna la evaluación.
student_group_professor_matter_id	VARCHAR(128)	Representa la relación entre el estudiante, el grupo, el profesor que va a emitir la evaluación y el id de la materia en la que se encuentra.
curricular_system_matter_sco_student_program_id	VARCHAR(128)	Representa la relación entre sistema curricular, la materia y el identificador del programa de estudio.

Tabla 2: r_evidence_user_sco

r_evidence_user_sco		
Descripción: La tabla muestra los aspectos relacionados con la vinculación de los usuarios, las evidencias y los sco.		
Atributo	Tipo	Descripción
upload_date	TIMESTAMP/DATE	Representa la fecha en la que se carga una evidencia.
evidence_id	VARCHAR(128)	Representa el identificador único que posee la evidencia.
sco_id	VARCHAR(128)	Identificador proveniente de los objetos que posibilitan compartir un contenido específico.
user_id	VARCHAR(128)	Este atributo constituye un identificador que se asigna a cada usuario registrado.

Tabla 3: r_evidence_user

r_evidence_user		
Descripción: La tabla muestra los aspectos relacionados con la vinculación de los usuarios y las evidencias.		
Atributo	Tipo	Descripción

evidence_user_sco_id	VARCHAR(128)	Este atributo representa la relación que existe entre una evidencia, el usuario que la posee y el identificador del sco.
user_id	VARCHAR(128)	Este atributo constituye un identificador que se asigna a cada usuario registrado.

Tabla 4: tb_evidence_comment

tb_evidence_comment		
Descripción: La tabla muestra los aspectos relacionados con la vinculación de los comentarios y las evidencias.		
Atributo	Tipo	Descripción
comment	TEXT	Este atributo constituye un juicio emitido acerca de algún tema en particular, específicamente una evidencia.
evidence_user_id	VARCHAR (128)	Este atributo representa la relación que se establece entre la evidencia y el usuario que la posee.

Tabla 5: tb_message

tb_message
Descripción: Describe todos los aspectos referentes a los mensajes.

Atributo	Tipo	Descripción
subject	VARCHAR(500)	Atributo que permite guardar el asunto del mensaje.
send_to	TEXT	Atributo que permite guardar para quien es el mensaje.
body	TEXT	Atributo que permite guardar el contenido o cuerpo de un mensaje.
send_date	TIMESTAMP/DATE	Atributo que permite guardar la fecha y la hora en la que se envía un mensaje
state_id	VARCHAR(128)	Etiqueta extranjera que identifica al objeto (estado) en la tabla.
user_id	VARCHAR(128)	Este atributo constituye un identificador que se asigna a cada usuario registrado.

Tabla 6: r_guard_user_message

r_guard_user_message		
Descripción: En la siguiente tabla se agrupa la información de la relación existente entre la tabla mensaje y usuario.		
Atributo	Tipo	Descripción

message_id	VARCHAR(128)	Etiqueta extranjera que identifica al objeto (mensaje) en la tabla.
user_id	VARCHAR(128)	Este atributo constituye un identificador que se asigna a cada usuario registrado.

Tabla 7: tb_group

tb_group		
Descripción: En la siguiente tabla se agrupa la información correspondiente a los grupos.		
Atributo	Tipo	Descripción
name	VARCHAR(128)	Atributo que representa el nombre del grupo, es decir constituye su identificador.

Tabla 8: nom_complexity

nom_complexity		
Descripción: En la siguiente tabla se agrupa la información del nomenclador complejidad.		
Atributo	Tipo	Descripción
complexity	VARCHAR(10)	Atributo que permite guardar el tipo de complejidad que puede alcanzar un ejercicio.

		Ejemplo: bajo, medio o alto.
--	--	------------------------------

Tabla 9: sf_guard_user

sf_guard_user		
Descripción: En la siguiente tabla se agrupa la información referente al usuario.		
Atributo	Tipo	Descripción
username	VARCHAR(128)	Atributo que representa una cadena de caracteres que identifican a cada usuario de manera única. A través de este se realiza la autenticación.
password	VARCHAR(128)	Atributo que representa una cadena de caracteres que constituye la clave para que el usuario entre en la plataforma. Cada usuario posee una contraseña única.
is_active	VARCHAR(1)	Atributo que permite ver si la sesión está activa o no
last_login	TIMESTAMP/DATE	Representa la fecha (incluida la hora) de la última vez que fue logueado el usuario.
first_name	VARCHAR(30)	Atributo que permitirá guardar el segundo nombre de un usuario.
second_name	VARCHAR(30)	Atributo que permitirá guardar el segundo

		nombre de un usuario.
first_last_name	VARCHAR(30)	Atributo que permitirá guardar el segundo nombre de un usuario.
second_last_name	VARCHAR(30)	Atributo que permitirá guardar el segundo nombre de un usuario.
gender	VARCHAR(1)	Atributo que permitirá guardar el sexo de un usuario.
birthday	DATE	Atributo que permitirá guardar la fecha de nacimiento de un usuario.
created_at	TIMESTAMP/DATE	representa los tipos de datos donde es necesario registrar además de la fecha la hora, identificados con el dominio: d_fecha hora.
updated_at	DATE	Representa la fecha en la que se realiza una actualización. .
curp	VARCHAR(30)	Atributo que permitirá guardar el segundo nombre de un usuario.
school_id	VARCHAR(128)	Atributo que almacena el identificador de la escuela en cuestión.

Conclusiones parciales

En el capítulo que recién concluye se generaron diferentes artefactos, como resultado del flujo de trabajo Análisis y Diseño propuesto por la metodología RUP. En tal sentido se documentan los diagramas de clases del análisis, proporcionando un mayor entendimiento de los requisitos del problema. Los diagramas de colaboración del análisis fueron muy útiles para tener una visión acerca de la comunicación entre las diferentes clases involucradas en el modelo de análisis. A través del modelo de diseño se logró preparar el sistema para la posterior implementación y prueba. Como resultado de esta etapa se obtuvieron los diagramas de clases del diseño, así como los diagramas de secuencia. La descripción de la base de datos y las tablas asociadas a él, contribuyó a establecer una relación directa con las entidades que aparecen en dicho modelo, en aras de lograr una implementación exitosa del portafolio.

Capítulo 4

Implementación y prueba.

Capítulo 4 Implementación y prueba.

Introducción

Una vez finalizado el análisis y diseño del sistema propuesto, el equipo de trabajo se encuentra listo para entrar en una fase posterior donde se obtendrá un producto final, capaz de cumplir con las funcionalidades definidas. En el capítulo que se presenta a continuación se documenta todo lo referente al proceso de implementación, y a la posterior fase de prueba del sistema. El diagrama de componentes y el diagrama de despliegue, constituyen artefactos tangibles que se generan en esta etapa. Además se llevan a cabo las pruebas de caja negra, a través del diseño de los casos de prueba.

4.1 Modelo de implementación.

La implementación constituye el centro de atención durante la fase de construcción. Se parte fundamentalmente del resultado obtenido durante el flujo de trabajo de análisis y diseño. El objetivo fundamental durante esta etapa es desarrollar la arquitectura y lograr estructurar el sistema final. Durante esta fase se organiza el código, se implementan los elementos del diseño, y se integran todos estos elementos para obtener un resultado satisfactorio que responda a lo definido durante las fases anteriores. Al llevar a cabo el modelo de implementación se establece la estructura de los elementos de implementación, basándose en las responsabilidades asignadas a los subsistemas de implementación y su contenido.

4.2 Diagrama de despliegue.

El diagrama de despliegue describe la arquitectura física del sistema durante la ejecución, en término de componentes de software, dispositivos y procesadores. Está integrado por nodos que son objetos físicos

que representan algún tipo de recurso computacional, tales como impresoras, dispositivos de comunicación, computadoras con procesadores entre otros elementos.

A continuación se presenta el diagrama de despliegue propuesto para el presente sistema:

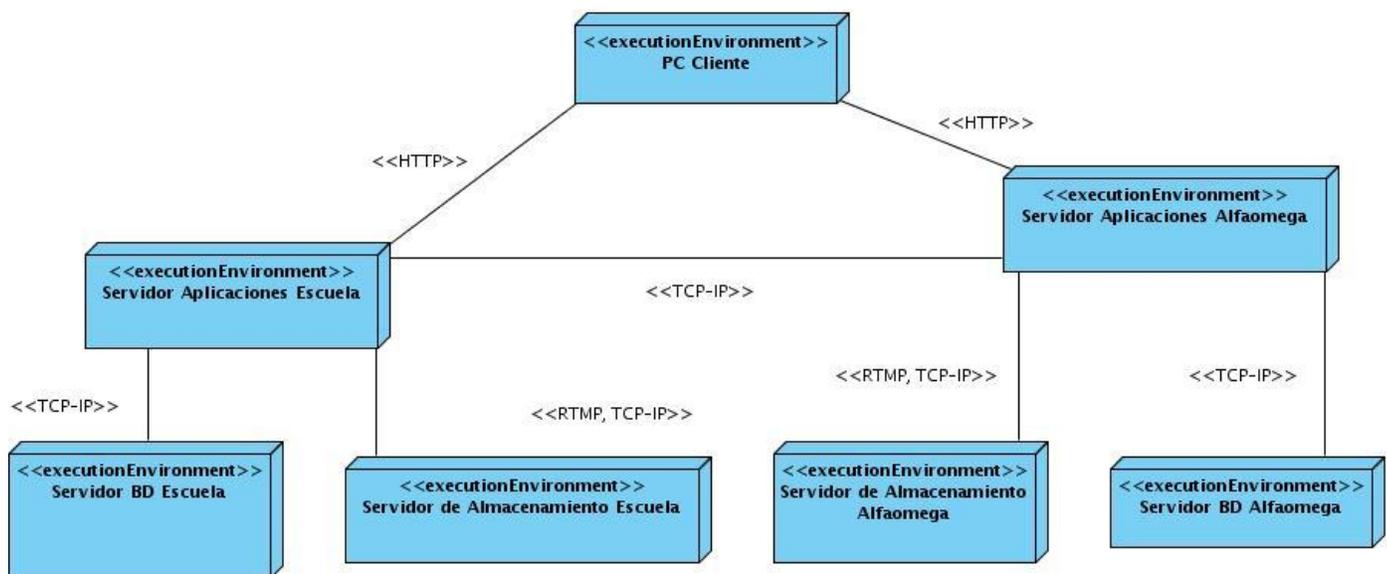


Imagen 10: Diagrama de despliegue.

La imagen muestra la existencia de estaciones de trabajo, las cuales se conectan a través del protocolo HTTP al servidor de aplicaciones de la escuela y al servidor central de aplicaciones de Alfaomega. Estos 2 servidores a su vez se conectan entre sí a través del protocolo TCP_IP. Cada uno de ellos tiene asociados un servidor que es propio de la escuela y un servidor de almacenamiento de la escuela, lo mismo sucede con el servidor de aplicaciones de Alfaomega. Es importante señalar que entre el servidor de aplicaciones y el servidor de almacenamiento se establece una conexión a través de los protocolos TCP_IP y el protocolo RTMP, este último facilita el acceso a las medias.

4.3 Diagrama de componentes.

Los diagramas de componentes modelan la vista estática de un sistema. El uso más importante de estos es mostrar la estructura de nivel elevado del modelo de implementación enfatizando en los subsistemas de implementación y sus dependencias, así como los subsistemas de implementación en capas.

A continuación se presenta el diagrama de componentes propuesto para el presente sistema.

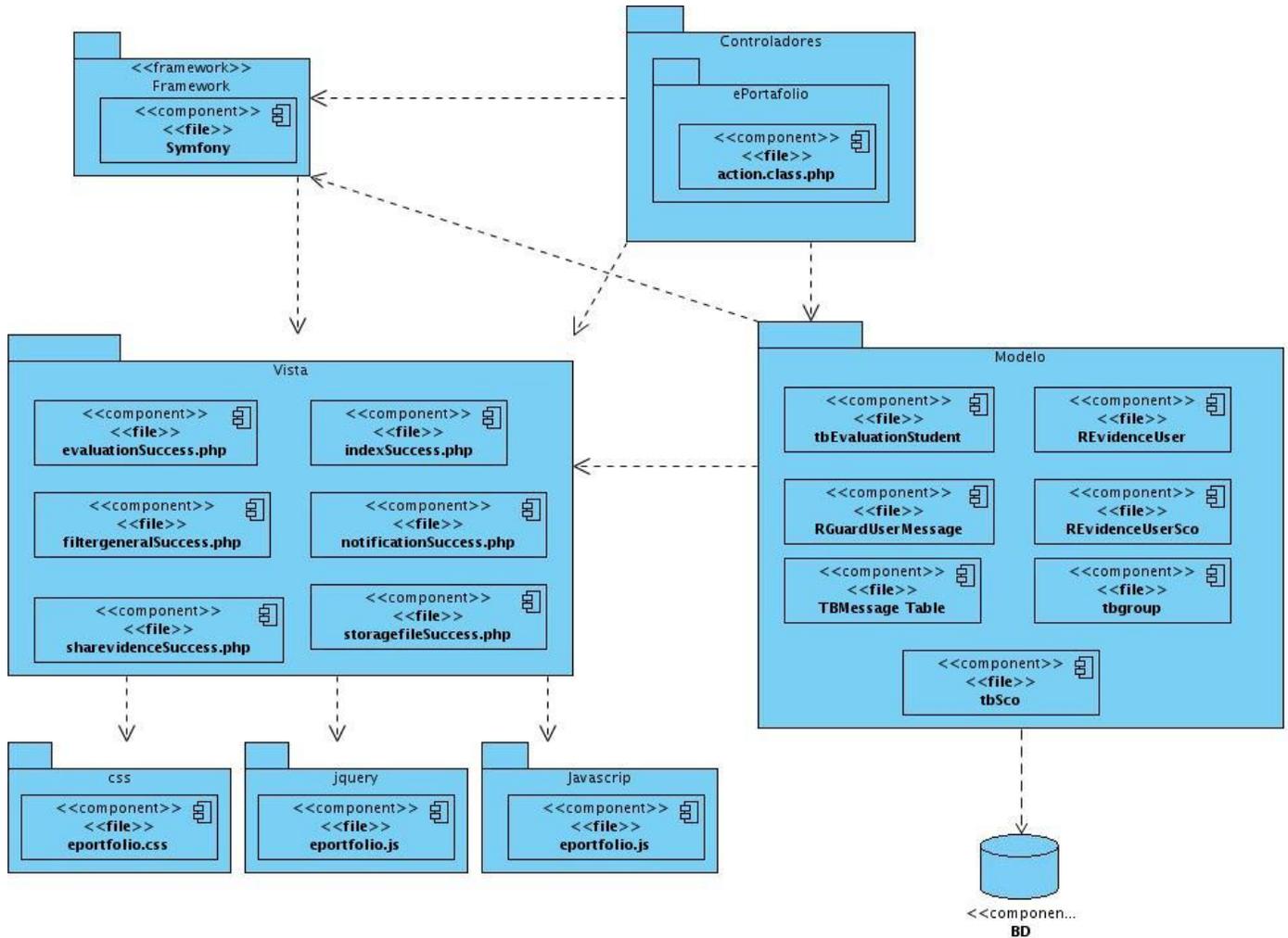


Imagen 11: Diagrama de componentes ePortafolio.

La imagen describe, siguiendo el patrón modelo vista controlador, la existencia de varios subsistemas de implementación, donde cada paquete cuenta con la existencia de diversos componentes, los cuales tienen una estrecha relación entre sí, lo que les brinda una mejor comunicación a la hora de implementar el portafolio electrónico.

4.4 Pruebas de software.

Al realizar un software es muy importante tener una vía para validar que lo que se realizó realmente es correcto o cumple con las expectativas del cliente o usuario final. Para ello es necesario conocer las funcionalidades específicas y determinar si realmente cumplen el objetivo para lo cual fueron diseñadas, demostrando que las funciones son operativas. De igual manera se pueden realizar pruebas internas en el código que se genera producto de la implementación, y así asegurar que todos los elementos existentes cumplan una función y dan lugar al producto final. Todo esto se resume en las pruebas de caja negra y las pruebas de caja blanca

4.5 Pruebas de caja negra.

Las pruebas de caja negra se centran fundamentalmente en los requisitos funcionales del software. Constituyen un enfoque complementario que intenta descubrir diferentes tipos de errores a los identificados en las pruebas de caja blanca. A través de las mismas es muy usual encontrar funciones incorrectas o ausentes, errores de interfaz, errores de rendimiento, así como errores de inicialización o terminación.

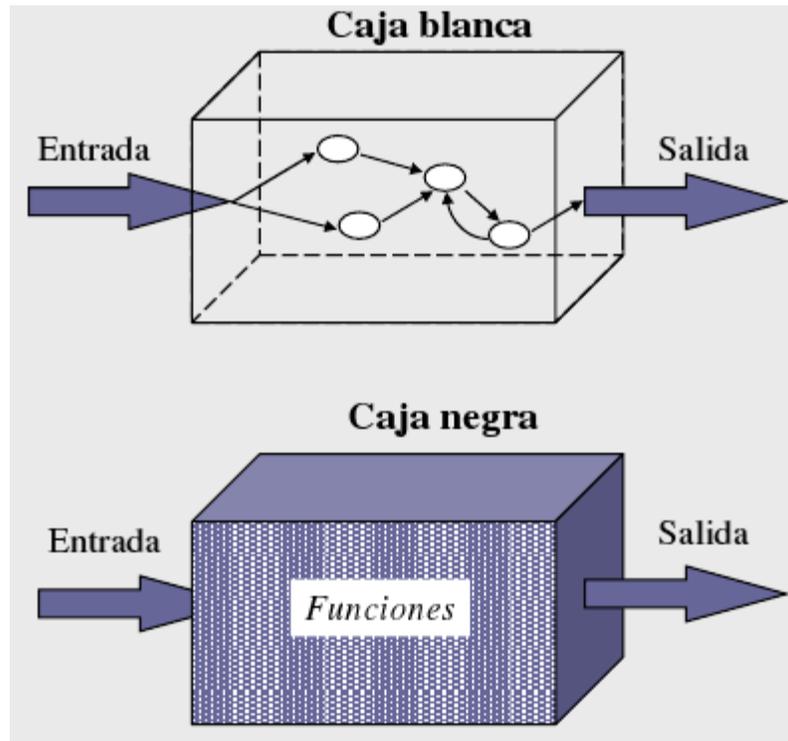


Imagen 12: Muestra gráfica del funcionamiento de las pruebas de caja negra y caja blanca.

4.6 Diseño de casos de prueba.

SC Consultar evaluaciones.

ID del escenario	Escenario	Añadir filtro	Fecha de inicio (rango de fecha)	Fecha fin (rango de fecha)	Actividad en el aula	Actividad en la plataforma	Reactivos en aula	Reactivos en plataforma	Evidencias	Todas	Respuesta del sistema	Flujo central
EC 1	Se accede a la opción de consultar Mis										Muestra un listado con 5 evaluaciones del estudiante, teniendo en cuenta el criterio: la	Escritorio de trabajo/e Portafoli

<p>calificaciones.</p>										<p>primera será la última evaluada. De cada una se muestran los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre del elemento evaluado. • Tipo de actividad. • Fecha de evaluación. • Calificación obtenida. <p>Muestra además el promedio general del grupo al que pertenece el usuario y el promedio individual del usuario. Brinda la posibilidad de añadir los siguientes filtros:</p> <ul style="list-style-type: none"> • Tipo de actividad • Rango de fechas. <p>Permite</p> <ul style="list-style-type: none"> • Navegar con el paginador 	<p>o/ Mis calificaciones</p>
------------------------	--	--	--	--	--	--	--	--	--	---	------------------------------

											o.	
[EC 2]	Selecciona un filtro e introduce los datos para realizar una búsqueda específica de información.	V	V	V	V	V	V	V	V	V	Valida los datos y muestra un listado de posibles coincidencias.	Escritorio de trabajo/e Portafolio/Mis calificaciones /Aplicar filtro.
		V	V	V	V	V	V	V	V	V		
		V	V	V	V	V	V	V	V	V		
		V	V	V	V	V	V	V	V	V		
		V	V	V	V	V	V	V	V	V		
		V	V	V	V	V	V	V	V	V		
		V	V	V	V	V	V	V	V	V		
EC 3	Se introducen datos incorrectos en el rango de fecha.	NA	Muestra un mensaje de información	Escritorio de trabajo/e Portafolio/Mis calificaciones / Aplicar filtro								
		NA	I	NA								
		NA	NA	I	NA	NA	NA	NA	NA	NA		
EC 4	Se selecciona un filtro y no se	NA	Muestra un mensaje informativo, donde se le indica al usuario	Escritorio de trabajo/e								
		NA										

	introduce ningún dato.	NA	que debe introducir datos en el filtro seleccionado.	Portafolio/Mis calificaciones / Aplicar filtro								
EC 5	No se encuentran coincidencias	V	V	V	V	V	V	V	V	V	Muestra un mensaje informativo.	Escritorio de trabajo/Portafolio/Mis calificaciones /Aplicar filtro
		V	V	V	V	V	V	V	V	V		
		V	V	V	V	V	V	V	V	V		
		V	V	V	V	V	V	V	V	V		
		V	V	V	V	V	V	V	V	V		
		V	V	V	V	V	V	V	V	V		
		V	V	V	V	V	V	V	V	V		
		V	V	V	V	V	V	V	V	V		
EC6	Selecciona la opción de realizar una nueva búsqueda	V	V	V	V	V	V	V	V	V	Permite volver a filtrar las evaluaciones y consultar los datos de las coincidencias encontradas.	Escritorio de trabajo/Portafolio/Mis calificaciones
		V	V	V	V	V	V	V	V	V		
		V	V	V	V	V	V	V	V	V		
		V	V	V	V	V	V	V	V	V		
		V	V	V	V	V	V	V	V	V		
		V	V	V	V	V	V	V	V	V		

	V	V	V	V	V	V	V	V	V	V
	V	V	V	V	V	V	V	V	V	V

Descripción de las variables.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Añadir filtro	Campo de selección	NO	
2	Fecha de inicio	Lista desplegable	NO	Sólo números (una fecha)
3	Fecha de fin	Lista desplegable	NO	Sólo números (una fecha)
4	Actividad en el aula	Campo de selección	SI	
5	Actividad en la plataforma	Campo de selección	SI	
6	Reactivos en el aula	Campo de selección	SI	
7	Reactivos en la plataforma	Campo de selección	SI	
8	Evidencias	Campo de selección	SI	
9	Todas	Campo de selección	SI	

4.7 Resultados obtenidos

Tipos No Conformidades	1era Iteración	2da Iteración
Formato (A y D)		
Error Técnico (D)		
Correspondencia con otro artefacto (A y D)	1	
Ortografía (A y D)		
Redacción (A y D)	2	
Validación (A)		
Opciones que no funcionan (A)		
Error de Interfaz (A)		
Error Idioma (A)	2	
Funcionalidad (A)	3	
Excepciones (A)		
Otros (A y D)		
Seguridad (A)		
Diseño CP (D)		
Recomendación (A y D)		
Total	8	

Tabla 10 Descripción de No Conformidades detectadas.

En la tabla se muestra las 8 No conformidades encontradas en la 1ra iteración de la revisión de la calidad del producto. En estos momentos ya se encuentran resueltas las 8 NC aportándole una mayor calidad y funcionalidad al ePortafolio.

Conclusiones parciales

Al concluir la implementación del ePortafolio para la plataforma educativa ZERA se obtiene un producto capaz de realizar las funciones definidas durante la captura de los requerimientos. Como parte de esta etapa el diagrama de despliegue confeccionado permitió establecer de manera gráfica la forma en la que se encontrarán los elementos físicos que intervienen en la instalación del portafolio electrónico como parte de la plataforma educativa ZERA. Se realizaron además los casos de prueba para llevar a cabo las pruebas al software y asegurar la calidad y correcto funcionamiento del mismo.

Conclusiones Generales

Conclusiones Generales.

Al concluir el análisis, diseño, implementación y prueba del módulo ePortafolio para la administración de las evidencias de los estudiantes dentro de la plataforma ZERA se determinan los resultados alcanzados y el cumplimiento de los objetivos propuestos.

Se realizó un estudio acerca de la existencia de portafolios electrónicos en otras plataformas educativas a nivel internacional, y la puesta en práctica en nuestro país. Además se determinaron las herramientas necesarias para llevar a cabo el desarrollo de un portafolio para la plataforma educativa ZERA. A partir de lo antes expuesto se procedió a la descripción del sistema, análisis, diseño y posterior implementación y prueba del mismo, arribando a las siguientes conclusiones:

- ❖ Como herramientas para el desarrollo se definieron las siguientes: como lenguaje de programación PHP 5, como framework Symfony, para el modelado UML se seleccionó Visual Paradigm. El gestor de base de datos utilizado fue PostgreSQL, y la metodología de desarrollo de software Rational Unified Process (RUP).
- ❖ Se determinaron las características principales del sistema a desarrollar teniendo en cuenta los requisitos capturados, lo cual permitió definir los casos de uso y generar el diagrama de casos de uso del sistema, dando una idea precisa acerca de lo que se implementaría posteriormente.
- ❖ Se realizaron los diagramas de clases del análisis, así como los diagramas de colaboración. A su vez fueron generados los diagramas de clases del diseño, dando una idea precisa acerca de cómo debería ser implementado el portafolio electrónico.
- ❖ Se implementó el módulo ePortafolio para la plataforma educativa ZERA tomando los resultados de las fases anteriores.
- ❖ Fueron realizadas pruebas de caja negra al portafolio electrónico, basadas en los casos de prueba, aportando una mayor calidad al producto. Se encontraron 8 No Conformidades que fueron corregidas permitiendo erradicar deficiencias y generar un portafolio que cumple con las

funcionalidades requeridas por el cliente.

- ❖ Se generó un portafolio electrónico, el cual se encuentra integrado a la plataforma educativa ZERA.

El presente trabajo permitió la realización de una fuerte investigación acerca de los portafolios electrónicos, para poder llegar a la implementación con una idea precisa acerca del funcionamiento de estos.

Recomendaciones

Recomendaciones

Al finalizar el presente trabajo de diploma se recomienda:

- ✓ Realizar una futura implementación del portafolio que pueda ser empleada por los docentes, ya que en la actualidad sólo se encuentra disponible para los estudiantes. Esto permitirá que el profesor pueda realizar todas las acciones sobre las evidencias desde su portafolio, sin tener que acudir a otras secciones.

Referencias Bibliográficas

Referencias Bibliográficas

AJAX. (n.d.). <http://msdn.microsoft.com>. Retrieved diciembre 13, 2010, from <http://msdn.microsoft.com/es-es/library/bb398874.aspx>.

Alvarez, M. A. (2001, Mayo 9). Retrieved diciembre 14, 2010, from <http://www.desarrolloweb.com/articulos/392.php>.

Banks, B. (2004, junio). <http://www.excellencegateway.org.uk>. Retrieved enero 2011, from http://www.excellencegateway.org.uk/media/ferl_and_aclearn/ferl/resources/organisations/fd%20learning/e-portfoliopaper.pdf

Catalani, E. (2007, agosto 20). <http://exequielc.wordpress.com>. Retrieved marzo 20, 2011, from <http://exequielc.wordpress.com/2007/08/20/arquitectura-modelovistacontrolador/>

Cully, C. (2001). Retrieved from <http://www.ohiolink.edu/etd/view.cgi?ucin990130595>

de la Fuente Gamiz, A. (n.d.). Retrieved enero 21, 2011, from <http://www.docentestic.es/index.php/aplicaciones-y-manuales/2-eva/2-mahara>

Diaz Barriga, A. (2010, julio 24). <http://www.suite101.net>. Retrieved enero 23, 2011, from <http://www.suite101.net/content/el-concepto-de-evaluacion-educativa-segun-ngel-diaz-barriga-a21804>

ECMA, I. (2009, diciembre). Retrieved diciembre 18, 2010, from <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf>.

Eguíluz Pérez, J. (2008). Retrieved enero 17, 2011, from <http://www.symfony.es/que-es-symfony/>.

Eguíluz Pérez, J. (2008). Retrieved enero 22, 2011, from <http://librosweb.es/css/index.html>

Eguíluz Pérez, J. (2008). <http://www.librosweb.es>. Retrieved enero 2011, from http://www.librosweb.es/ajax/pdf/introduccion_ajax.pdf

ePortafolio, I. (2005). Retrieved enero 17, 2011, from <http://www.imsglobal.org/ep/>

Fuentes Betancourt, J. (2007, agosto 16). Retrieved enero 19, 2011, from <http://www.journal.lapen.org.mx/sep07/FUENTES%20Final.pdf>

G.V., D. (2010, abril 27). Retrieved enero 21, 2011, from <http://vitaminaweb.com/tutorial-rapido-de-jquery/>

García Doval, F. (2005). Retrieved diciembre 18, 2010, from <http://www.um.es/glosasdidacticas/GD1/10.pdf>

García Hernández, E. (2000). www.quadernsdigitals.net. Retrieved enero 15, 2011

Giraldo, L. (2005, septiembre 24). Retrieved enero 18, 2011, from http://hugolopez.phi.com.co/docs/download/file=Giraldo-Zapata-Herramientas%20de%20ISW.pdf,_id=17.

González, F. (2008, Agosto). <http://portaleducativo.edu.ve>. Retrieved enero 2011, from http://portaleducativo.edu.ve/Recursos_didacticos/Usos_educativo_tic/Modulo_IV/MIV_MA_MC_05.pdf

Kruchten, P. (n.d.). Retrieved enero 18, 2011, from <http://www.google.com/books>

Letelier, P. (2008). <http://www.slideshare.net>. Retrieved from <http://www.slideshare.net/oscar8711/introduccion-a-rup-presentation>

López Carrasco, M. A. (2003). Retrieved enero 21, 2011, from <http://www.ccee.edu.uy/ensenian/formdoc/lopez.pdf>

Lorenzo, G. a. (2005, Julio). Retrieved enero 18, 2011, from <http://www.educause.edu/>.

Mariel, S. G. (2008). <http://es.scribd.com>. Retrieved enero 23, 2011, from <http://es.scribd.com/doc/4798851/La-evaluacion-por-portafolios-digitales-una-alternativa-en-la-evaluacion-para-integrar-las-TIC-al-curriculo-Desafios-y-posibilidades>

- Molnar, G. (n.d.). *www.chasque.net*. Retrieved febrero 20, 2011, from http://www.chasque.net/gamolnar/evaluacion_educativa/evaluacion.01.html
- Moreno Romero, L. M. (2008). *www.eatis.org*. Retrieved enero 2011, from www.eatis.org/eatis2010/portal/paper/memoria/html/files/83.pdf
- Muñoz-Justica, J. (2008, octubre 25). Retrieved enero 21, 2011, from <http://www.moodlemoot.net/2008/file.php/1/abstracts/107-363-1-PB.pdf>.
- Navas Gómez Tagle, M. a. (n.d.). Retrieved enero 20, 2011, from www.ciie.cfie.ipn.mx/2domemorias/documents/c/c14/c14_57.pdf.
- Netbeans. (2007). *www.netbeans.org*. Retrieved enero 18, 2011, from http://netbeans.org/index_es.html.
- Pascual Gimeno, N. (2006, octubre). *www.recercat.net*. Retrieved enero 20, 2011, from www.recercat.net/bitstream/2072/8979/1/Pascual.pdf.
- Postgresql. (2008). Retrieved enero 20, 2011, from http://www.postgresql-es.org/sobre_postgresql
- Prendes Espinosa, M. P. (n.d.). Retrieved enero 15, 2011, from www.sav.us.es/pixelbit/pixelbit/articulos/n32/2.pdf.
- Prendes Espinosa, M. P. (2008, marzo). *www.sav.us.es*. Retrieved enero 20, 2011, from www.sav.us.es/pixelbit/pixelbit/articulos/n32/2.pdf
- Pressman, R. (n.d.). Retrieved enero 21, 2011, from <http://www.scribd.com/doc/7978336/Ingenieria-de-Software-Un-Enfoque-Practico-Pressman-5th-Ed>.
- Systems, P. S. (2008) Retrieved enero 22, 2011, from <http://lucas.hispalinux.es/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf>.
- Valdez Altamirano, I. A. (2009). *www.ubicuos.com*. Retrieved enero 18, 2011
- XHTML. (2008, febrero 7). Retrieved enero 17, 2011, from <http://www.w3c.es/divulgacion/guiasbreves/XHTML#uso> .